Deutsche Geodätische Kommission
der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 727

Claus Nagel

Spatio-Semantic Modelling of Indoor Environments for Indoor Navigation

München 2014

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325 ISBN 978-3-7696-5139-3

Deutsche Geodätische Kommission
der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 727

Spatio-Semantic Modelling
# of Indoor Environments for Indoor Navigation

Von der Fakultät VI – Planen Bauen Umwelt
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation

von

M.Sc. Claus Nagel
aus Karlsruhe

D 83

München 2014

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325 ISBN 978-3-7696-5139-3

Adresse der Deutschen Geodätischen Kommission:

DGK

Deutsche Geodätische Kommission

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 – 89 – 23 031 1113 ● Telefax +49 – 89 – 23 031 - 1283/ - 1100
e-mail hornik@dgfi.badw.de ● http://www.dgk.badw.de

Vorsitzender: Prof. Dr. Dr. Harald Schuh

1. Gutachter: Prof. Dr. Thomas H. Kolbe
2. Gutachter: Pof. Dr. Frank Neitzel

Tag der wissenschaftlichen Aussprache: 21.03.2014

Diese Dissertation ist auf dem Server der Deutschen Geodätischen Kommission unter <http://dgk.badw.de/>
sowie auf dem Dokumentenserver der Technischen Universität Berlin unter
<http://opus4.kobv.de/opus4-tuberlin/frontdoor/index/index/docId/5537> elektronisch publiziert

ISSN 0065-5325

ISBN 978-3-7696-5139-3

To Jennifer
for your continual support and love to me

and to Oscar and Emil
for lighting up my life

Acknowledgements

In doing this research and writing this thesis, I enjoyed support and encouragement from many people without whom this thesis would have not been written and to whom I want to express my sincere gratitude.

First, I want to thank Prof. Dr. Thomas H. Kolbe who supervised my thesis and initiated and lead the research project at Technische Universität Berlin within many aspects of my research were done. I thank him for the fruitful discussions, his criticism, concern, valuable scientific remarks, but also for the freedom to develop my own ideas and concepts and his continuous support and encouragement. I thank Prof. Dr. Frank Neitzel for supervising and helping me finish my thesis.

I further want to thank all colleagues who helped me through discussions as well as valuable comments and remarks to gain insight into the diverse aspects of the research topic. This first includes the members of the Department for Geodesy and Geoinformation Science at the Technische Universität Berlin, especially Thomas Becker and Robert Kaden for scientific debates, joint articles, and for carefully reviewing several chapters of my thesis. I thank Prof. Ki-Joune Li from the Pusan National University, South Korea, and Prof. Jiyeong Lee from the University of Seoul, South Korea, for fruitful discussions on indoor spatial theory within the Indoor Spatial Awareness research project and for our joint standardization efforts on IndoorGML at the Open Geospatial Consortium. I am grateful to Prof. Dr. Sisi Zlatanova and Liu Liu from the Delft University of Technology, The Netherlands, for sharing alternative scientific views on indoor space modelling and for open and constructive discussions. I want to thank all the members of the Special Interest Group 3D, and especially PD Dr. Gerhard Gröger from the University of Bonn, Germany, not just for our joint and wonderful work on CityGML but also for the valuable and encouraging feedback that I received on geometric-topological questions of my research work at several SIG 3D meetings. I also owe my thanks to my new colleagues at virtualcitySYSTEMS GmbH, Berlin, especially Ingolf Jung and Dr. Lutz Ross, for being 3D GIS enthusiasts just like me and for giving me the freedom and support to finish my thesis. Thanks to Gavin Brown, whose supervisor of his Master’s Thesis I had the pleasure to be, and to Aftab Khan for creative and enjoyable discussions on navigation constraints. Not at least I want to thank the students of the GIS project in winter term 2011/12 at the Technische Universität Berlin for their efforts in acquiring Wi-Fi sensor measurements and 3D building models of the Technische Universität Berlin that served as basis for the chapter 7 of my thesis: Devika Kakkar, Katrin Arendholz, K.J. Sowmiya Narayanan, Felipe Carvajal, Olga Kuvshinnikova, Anna Thieme, Dimitris Karakostis, Vasileia Apostolidou, Carsten Malchow, Camilo León Sánchez, Maria Popkova, and Charalampos Gaitanis.

My special thanks go to Karl-Heinz Häfele and Dr. Joachim Benner from the Karlsruhe Institute of Technology, Germany, for introducing me into the world of 3D GIS, Semantic 3D City Models, and Building Information Modelling during my Master’s Thesis and for inspiring discussions ever since.

I am grateful to Rosemarie Kunkel for being the heart and soul of the Department for Geodesy and Geoinformation Science at the Technische Universität Berlin and for guiding me through the administrative labyrinths towards the doctoral degree.

This thesis would have not been possible without the lasting patience, support, and love of my family. All my thanks go to my mom and dad and to my parents-in-law who always supported my wife and me in busy times. All my love goes to my wife Jennifer for her belief in me and her continual support and love.

Contents

Table of Abbreviations

0D	zero dimensional
1D	one dimensional
2D	two dimensional
3D	three dimensional
4IM	4-Intersection Model
9IM	9-Intersection Model
ADT	Abstract Data Type
AEC/FM	Architecture, Engineering, Construction and Facility Management
AOA	Angle of Arrival
B-Rep	Boundary Representation
BIM	Building Information Model
BISDM	ESRI Building Interior Space Data Model
CAD	Computer-aided Design
CityGML	OGC City Geography Markup Language
CNM	Combinatorial Network Model
COLLADA	Collaborative Design Activity
CSG	Constructive Solid Geometry
DE-9IM	Dimensionally Extended 9-Intersection Model
DGP	Direct Path Graph
DHE	Dual-Half Edge
DR	Dead Reckoning
ER	Entity Relationship
ESRI	Environmental Systems Research Institute
FDS	Formal Data Structure
FME	Feature Manipulation Engine
GDI-DE	Spatial Data Infrastructure Germany
GDF	Geographic Data Files
GFM	General Feature Model
GIS	Geographic Information System
GML	Geography Markup Language
GNM	Geometric Network Model
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IAI	International Alliance for Interoperability
IDW	Inverse Distance Weighting
IFC	Industry Foundation Classes
INO	Indoor Navigation Ontology
INSM	Indoor Navigation Space Model
INSPIRE	Infrastructure for Spatial Information in the European Community
IR	Infrared
ISA	Indoor Spatial Awareness
ISO	International Organization for Standardization
ISO/PAS	ISO Publicly Available Specification
ISO/TC 211	ISO Technical Committee 211
ITS	Intelligent Transport Systems
IVS	Indoor Visibility Structure
KML	Keyhole Markup Language
LBS	Location-Based Services
LOD	Level of Detail
MAC	Media Access Control
MDA	Model-Driven Architecture
MIME	Multipurpose Internet Mail Extensions
MLSEM	Multilayered Space-Event Model
NBIMS	U.S. National BIM Standard
NDM	Oracle Network Data Model
NRS	Node-Relation Structure
OCL	Object Constraint Language
OGC	Open Geospatial Consortium
ONALIN	Ontology and Algorithm for Indoor Routing
OpenLS	OpenGIS Location Services
OWL	Web Ontology Language
QR Code	Quick Response Code
RCC	Region Connection Calculus
RFID	Radio-Frequency Identification

RSSI	Received Signal Strength Indicator
SIG 3D	Special Interest Group 3D
S-MAT	Straight Medial Axis Transformation
SQL	Structured Query Language
SSM	Simplified Spatial Model
STEP	Standard for the Exchange of Product Model Data
SWG	OGC Standards Working Group
TEN	Tetrahedral Network
TDOA	Time Difference of Arrival
TOA	Time of Arrival
UDM	Urban Data Model
UIM	Urban Information Model
UML	Uniform Modelling Language
UNO	User Navigation Ontology
URI	Uniform Resource Identifier
UWB	Ultra-Wideband
W3DS	OGC Web 3D Service
WFS	OGC Web Feature Service
WIRG	Weighted Indoor Routing Graph
WMS	OGC Web Map Service
WVS	OGC Web View Service
WGS 84	World Geodetic System 1984
WMS	OGC Web Map Service
X3D	Extensible 3D
XML	Extensible Markup Language

Summary

Outdoor navigation services have established over the past decade and are ubiquitously available today. We have become accustomed to everyday mobile devices such as smartphones and car navigation systems that help us plan our routes or provide us with personalized and added value information about our current position. Localization technologies based on global navigation satellite systems as well as the acquisition and availability of geoinformation about navigable road networks for large parts of the world are two main drivers for this development. Systems for navigating people or vehicles in indoor spaces however are not as widely spread to date but considerably lag behind existing outdoor solutions. The fact that GPS is not available indoors is often seen as a key reason, and substantial work in academia and industry has been done in developing alternative localization technologies. But there is also a need for a standardized model of the navigation space providing rich, complete, and accurate geoinformation about the indoor environment in order to address the multitude of challenges in localization, path planning, tracking, and guidance facing indoor navigation.

This thesis presents an approach to the *spatio-semantic modelling of indoor space* that aims at answering this need. Based on a comprehensive survey of related work, the multiple and different conceptual and technical challenges and requirements to indoor navigation are analysed and elaborated. Whereas existing approaches often tailor the complexity of the navigation task a priori to a specific and rigid navigation setting, the main research goal of this thesis is to define a generic framework for indoor navigation that satisfies the requirements and overcomes limitations in related work. Against this background, a *Multilayered Space-Event Model* (MLSEM) is developed that allows for the modelling, integration and joint consideration of different and multiple representations of navigable and non-navigable indoor spaces for various modes of locomotion such as walking, driving or even flying. The notion of indoor space hereby goes beyond the built-up environment but also comprises logical and thematic spaces such as security or disaster zones as well as sensor spaces reflecting the diverse indoor localization technologies and methods. The MLSEM facilitates the ad-hoc selection and combination of available localization technologies supported by the mobile end-user device and of appropriate representations of navigable indoor spaces according to the context of individual navigation users as well as global and user-dependent navigation constraints.

A second principal contribution of the research work is the embedding of the MLSEM in a sound mathematical framework. A correct, consistent, and complete mathematical formalization is to be seen as a key prerequisite for the definition of a universal view of indoor space that integrates existing approaches in literature at a foundational level. The formalization draws from fields such as algebraic topology, manifold theory, and graph theory in order to represent indoor space entities by their 2- or 3-dimensional real-world shape and their one-to-one mapping onto a graph-based conceptualization. A novel space layer algebra is proposed for the manipulation of complex indoor space models. Finally, the thesis develops a formal conceptual data model for the MLSEM in conformance with the ISO 19100 standards family for geographic information modelling. By this means, indoor space data can be exchanged according to the MLSEM between computer systems and applications in a standardized way in order to enable navigation and location-based services in indoor environments.

The result of this thesis is a generic, flexible, and context-aware modelling framework for indoor space that supports the complementary navigation tasks of localization, path planning, tracking, and guidance. The sound definition of indoor space at a conceptual, mathematical, and data exchange level exceeds and explains alternative approaches to indoor space modelling. The MLSEM is currently at the core of an international standardization activity called *IndoorGML* carried out at the Open Geospatial Consortium, which aims at making indoor navigation services as available and successful as in outdoor environments.

Zusammenfassung

Navigationsdienste im Freien sind seit vielen Jahren etabliert und heutzutage überall verfügbar. Alltägliche mobile Geräte wie Smartphones oder Autonavigationsgeräte unterstützen uns wie selbstverständlich bei der Wegeplanung oder stellen personalisierte Mehrwertinformationen über unseren aktuellen Standort bereit. Diese Entwicklung ist vor allem Lokalisierungstechnologien auf Basis von GNS-Systemen sowie der flächendeckenden Erfassung und Bereitstellung von Geoinformationen über navigierbare Straßen zu verdanken. Navigationssysteme für Personen oder Fahrzeuge im Innenraum sind hingegen nicht so weit verbreitet und bleiben hinter den Möglichkeiten bestehender Lösungen im Freien zurück. Als Hauptursache wird vielfach die Nichtverfügbarkeit von GPS in Gebäuden angeführt, weshalb insbesondere die Entwicklung von alternativen Lokalisierungstechnologien im Fokus von Arbeiten in Wissenschaft und Industrie steht. Ebenso wichtig ist jedoch ein standardisiertes Modell über den navigierbaren Raum, das reichhaltige, vollständige und genaue Geoinformationen über den Innenraum bereitstellt, um so die vielfältigen Herausforderungen an die Innenraumnavigation auf dem Gebiet der Lokalisierung, Wegeplanung und -leitung, sowie Nachverfolgung von bewegten Objekten zu adressieren.

Die vorliegende Doktorarbeit stellt ein *räumlich-semantisches Modell für den Innenraum* vor, das diesen Bedarf decken soll. Auf Grundlage einer umfassenden Studie verwandter Arbeiten werden die vielfältigen und unterschiedlichen konzeptuellen und technischen Anforderungen an die Innenraumnavigation analysiert und herausgearbeitet. Während bestehende Ansätze die Komplexität der Navigationsaufgabe oftmals a priori auf einen spezifischen und starren Anwendungsfall reduzieren, besteht ein wesentliches Forschungsziel dieser Arbeit in der Definition eines generischen Rahmenwerks für die Innenraumnavigation, das den Anforderungen genügt und die Schwächen anderer Arbeiten überwindet. Vor diesem Hintergrund wird ein *Multilayered Space-Event Modell* (MLSEM) entwickelt, das sowohl die Modellierung, die Integration als auch die gemeinsame Betrachtung unterschiedlicher Repräsentationen navigierbarer und nicht-navigierbarer Innenräume für verschiedene Fortbewegungsarten wie Laufen, Fahren oder sogar Fliegen ermöglicht. Der Begriff des Innenraums geht hierbei über den gebauten Raum hinaus und umfasst auch logische oder thematische Räume wie Sicherheits- und Unfallzonen und Sensorräume, welche die diversen Lokalisierungstechnologien und –methoden in Gebäuden abbilden. Das MLSEM ermöglicht die Ad-hoc-Auswahl und Kombination der verfügbaren und vom Endgerät des Nutzers unterstützen Lokalisierungstechnologien sowie derjenigen navigierbaren Innenräume, die den individuellen Nutzerkontext sowie globale und nutzerabhängige Navigationsbeschränkungen widerspiegeln.

Ein zweiter wesentlicher Beitrag der Arbeit besteht in der korrekten, widerspruchsfreien und vollständigen mathematischen Formalisierung des MLSEM. Eine solche Formalisierung stellt eine wichtige Voraussetzung für ein allgemeingültiges Verständnis des Innenraums dar, auf dessen Grundlage bestehende Ansätze in der Literatur zusammengeführt werden können. Die Formalisierung bedient sich Teilgebiete wie der algebraischen Topologie, der Theorie der Mannigfaltigkeiten, und der Graphentheorie, um die Objekte des Innenraums sowohl gemäß ihrer realen Form in 2D oder 3D zu modellieren als auch eineindeutig auf eine graphbasierte Repräsentation abzubilden. Weiterhin wird erstmals eine Space-Layer Algebra zur Manipulation komplexer Innenräume vorgestellt. Schließlich entwickelt die Arbeit ein formales konzeptuelles Datenmodell für das MLSEM in Übereinstimmung mit der ISO 19100 Normenreihe für die geographische Informationsmodellierung. Hierdurch können Innenraumdaten gemäß dem MLSEM zwischen Computersystemen und Anwendungen standardisiert ausgetauscht werden, um Navigations- und standortbezogene Dienste im Innenraum umzusetzen.

Das Ergebnis dieser Arbeit ist ein generisches, flexibles und kontextbezogenes Rahmenwerk für die Modellieung von Innenräumen, das die sich ergänzenden Navigationsaufgaben der Lokalisierung, Wegeplanung und -leitung, sowie Nachverfolgung von bewegten Objekten unterstützt. Die Definition des Innenraums auf konzeptueller, mathematischer und Datenaustauschebene erweitert und erklärt alternative Ansätze der Innenraummodellierung. Das MLSEM bildet derzeit den Kern der Standardisierungsaktivität zu *IndoorGML* im Open Geospatial Consortium, die zum Ziel hat, dass Navigationsdienste im Innenraum ebenso verfügbar und erfolgreich werden wie im Freien.

Chapter 1
Introduction

1.1 Motivation

Outdoor navigation services have established over the past decade and are ubiquitously available today. We have become accustomed to everyday mobile devices such as smartphones and car navigation systems that help us find nearby shopping malls, restaurants, or museums and get us to airports, public transport and train stations, or further destinations. Despite apparent distinctions in technologies and techniques underlying current outdoor navigation systems and services, there are two main drivers for this development. First, global localization systems such as GPS have become publicly available and provide localization information with sufficient degree of accuracy. Second, accurate and complete models of the outdoor space have been acquired which supply navigation data and content such as road networks for large parts of the world. Both aspects are accompanied by the evolution of mobile devices and navigation systems that, on the one hand, have become location-aware and, on the other hand, enable ubiquitous access to navigation space models and navigation-relevant information such as traffic jams or points of interest. This progress has also given rise to additional *location-based services* (LBS) for outdoor environments besides navigation which exploit the current location of the mobile user in order to offer personalized and added value information about the user's context. Typical examples include *"Where am I?"* and *"Who or what is near me?"* services that underpin a wide range of applications in different domains. Next-generation services are enabled by the fact that mobile users nowadays not only reactively consume LBS applications but also proactively and collaboratively produce and share location-tagged information and content.

We averagely spend 80% up to 90% of our time indoors (e.g., Jenkins et al. 1992, Brasche & Bischof 2005). It hence is natural to aim at extending the scope of location-based services to the indoor space which renders an area of intensive research in academia and industry. Promising applications for indoor environments such as navigation and emergency services, logistics, health care monitoring, and people or goods management are addressed. Selected applications are presented in the following in order to highlight some of their information needs (cf. Lacroix 2013).

Indoor navigation. Indoor venues such as shopping malls and airports are continuously growing larger and more complex. Although humans are generally good at assigning meaning to spatial structures while moving through and interacting with the indoor environment and thus have cognitive abilities to locate themselves, we often get lost in indoor space especially in case of incomplete knowledge about the indoor environment. There is a need for the possibility of virtually exploring and querying complex venues in order to discover our desired destination (e.g., the nearest food store in a shopping mall, the location of a product within the food store, the check-in counters at airports, or the public toilets inside a train station), and of being guided to this destination according to our individual needs and preferences as well as movement restrictions. Similar to outdoor navigation, this requires the mobile device of the user to be capable of acquiring its location with sufficient accuracy as well as access to a rich indoor space model. Indoor navigation is the most commonly discussed indoor location-based service in literature.

Emergency response. In emergency situations like fire incidents, immediate interventions such as guiding people to the nearest emergency exits as well as routing rescue personnel and fire fighter forces to injured people and to fire spots could save human lives. Although this actually renders a navigation task, the requirements and constraints in emergency situations are high and very specific. The indoor space model has to be up-to-date and has to support dynamic changes. For example, disaster areas have to be captured and continuously updated in the model in order to understand the demolition state of the building and to determine safe escape routes. Likewise, sections of the building that are inaccessible in normal situations may become available (e.g., emergency exits) but also movement restrictions may be imposed (e.g., to exclude elevators from routes or to control the flow of people). Rich information about obstacles (e.g., the location and material of walls) is required to assess whether obstacles can be removed or torn down in order to provide additional paths. The localization infrastructure needs to report precise and accurate location information to avoid people getting lost and disoriented (e.g., due to smoke) and to keep track of the fire fighter forces.

Services for elderly and disabled people. The world population of people over age of 65 is rapidly growing. Technology and LBS applications (e.g., health care monitoring) contribute to a “smart environment” that aids the elderly in being independent. Besides a smart home, also public venues such as train stations or public buildings need to offer corresponding services to avert the social exclusion of elderly people from public facilities and services. For example, elderly people often suffer from physical limitations in mobility and thus must be guided through the indoor space along barrier-free and safe paths which may substantially differ from the shortest or fastest paths. This requires a fine-granular representation of the built environment that captures non-passable obstacles and barriers at a small enough scale. Moreover, precise location and orientation information (e.g., based on special landmarks) has to be provided in order to assist elderly people along their route. Both the handling of the mobile device and the presentation of information have to conform to the needs of this user group. Similar observations can be made in the context of disabled people (e.g., wheelchair users).

Tracking of people and goods. Location-based services are not only of interest for the consumer market but also for enterprises. For example, tracking services could support the operator of a shopping mall in understanding the movement patterns of customers and hence in optimizing the layout of the mall. Moreover, such services would facilitate to push personalized product advertisements to the mobile device of people as they travel past a store (or, alternatively, information about artefacts in a museum). Tracking services rely on precise location fixes as well as an accurate and complete mapping of the indoor space which allows the trajectories of people and goods to be analysed both spatially and semantically.

Autonomous mobile robots. Mobile robotics is a highly active research area where a considerable body of theory has been developed to date. Since robots can be said to be blind and deaf per default, location-awareness is one of the crucial aspects for their autonomous capabilities. The focus of scientific works is therefore mostly put on the combination of different sensor technologies and the development of probabilistic methods that would allow the robot to self-locate itself with high accuracy. A second fundamental issue in mobile robotics is the indoor space model. Significant work has been undertaken to enable robots to deal with unknown environments by mounting sensors which continuously scan and interpret the environment in order to instantly map the indoor space while the robot is moving. Alternatively, robots can be provided with a complete model of the indoor space which, for instance, allows the robot to validate location estimates relative to the model and to more easily plan or follow predefined space trajectories.

The industry has recognized the potential of indoor LBS and there is a quickly growing market for corresponding applications. To a great extent, these applications are realized as internal enterprise applications or target a niche market if publicly accessible at all. However, at least when Google entered the market in 2011 with its indoor maps and an accompanying indoor localization solution for a variety of public venues like airports, museums, shopping malls, or stadiums, indoor LBS became main stream. As of January 2013, Google claims to hold 10,000 indoor maps across ten different countries.[1] A similar world-wide and public service was launched by Microsoft in June 2012 based on indoor maps provided by Nokia. The maps cover similar venues and finding closest facilities and services (e.g., toilets, cash machines) inside the venues is also supported. As of July 2012, Nokia claims to have mapped 4,600 venues in 38 countries with a 2000% increase in just 16 months[2] which nicely illustrates the rapid growth of the market. Also large data providers for outdoor maps such as NAVTEQ have released their first indoor contents.

Nevertheless, although promising applications can be envisioned and proprietary systems become available, location-based services for indoor space are still less well developed and considerably lag behind established outdoor services. Regarding the two identified drivers for outdoor LBS, the fact that GPS is not available inside buildings is most often cited in literature as being the main reason. Substantial work in academia and industry has therefore been done in developing alternative localization techniques and methods. However, and as shown above, an indoor space model that provides accurate, complete, and rich information about the complex interior environment likewise renders a fundamental prerequisite and even is said to be at the core of any indoor LBS system in literature

[1] See http://maps.google.com/help/maps/indoormaps/.

[2] See http://conversations.nokia.com/2012/07/16/nokia-leads-the-way-with-indoor-mapping/.

(e.g., Worboys 2011, Liu & Zlatanova 2011b). In their well-received reference book on indoor LBS, (Kolodziej & Hjelm 2006) expose two main problems of indoor space models (referred to as *world models*) in current location-based service solutions:

1. *"World models implicitly underlie the mapping infrastructure, yet they are rarely set in a proper theoretical framework by going back to the basics of what a location (space) can be in pure mathematics and, more importantly, symbolically (semantics)."* (Kolodziej & Hjelm 2006, p. 180),

2. *"One of the problems to date with attempting to solve the LBS computing problem is that every proposed solution has its own approach, data structures, processes, and the like. There is little if any standardization between the various approaches. Moreover, there is no one standardized view of the world that would unlock the potential of LBS computing. Standardization can be achieved at the foundational level by defining a universal view of the geographical space."* (Kolodziej & Hjelm 2006, p. 221).

(Kolodziej & Hjelm 2006) hence couple the success of indoor LBS with a formal and standardized representation of the spatial and semantic aspects of indoor space. However, the identified problems have not been sufficiently solved in literature up to now (cf. chapter 2). And the proprietary and non-standardized systems available on the market which have been presented above contribute to the problems rather than to their solution.

The motivation for this thesis is therefore to fill the gap identified by (Kolodziej & Hjelm 2006) by defining a spatio-semantic model for describing and structuring indoor space. The conceptual design of the indoor space model shall be built on a solid and consistent mathematical basis, and its validity and applicability shall be argued in the context of indoor navigation. Moreover, the indoor space model shall be realized in conformance with existing international standards on the modelling and exchange of geographic information so that it can be implemented by indoor navigation applications in a standardized way.

The field of indoor navigation faces a large number of challenges which significantly differ from the outdoor world. In the following section, the most relevant challenges are reviewed in detail and their implications on the modelling of indoor space are discussed. Based on this discussion, the research goals and objectives of this thesis are elaborated in the chapters 1.3 and 1.4.

1.2 Challenges to Indoor Navigation

The task of navigation generally comprises 1) the *localization* of a person or object, 2) the *planning* of *paths* between locations, and 3) the *tracking and guidance* along the path (Becker et al. 2009a, Worboys 2011). Localization (or positioning) determines the actual location of a person or object with respect to a given spatial reference system and localization technology. Path planning deals with finding the best path (e.g., the fastest, the shortest, or the cheapest) from a start to an end location and requires geoinformation about the navigation space in order to derive suitable route sections. The naming of locations in path queries additionally involves an addressing or georeferencing schema which maps locations onto human-friendly names and textual descriptions or onto unique geometric coordinates. The process of tracking compares the actual position of a person or object travelling through the navigation space with its target position along the path, and employs strategies and actions for minimizing the deviations between both. The guidance of moving persons along the path is supported by visual aids and presentations of the route as well as position communication and route instructions using appropriate media and channels (Nagel et al. 2010). It hence partly renders a human-device interaction task.

Each of these navigation aspects depends to a great extent on the *navigation context* which means the setting and interrelated conditions in which the navigation takes place. This very general understanding can be refined by differentiating between user-centric and environmental contextual information being relevant to the task of navigation (e.g., Abowd et al. 1999, Mokbel & Levandoski 2009). The *user context* comprises, for example, the state of the user (e.g., physical and perceptual capabilities), the user interface (e.g., capabilities of the mobile end-user device), as well as user-dependent navigation conditions (e.g., preferences or access rights). Whereas the user context is distinct for each individual or object, the *environmental context* determines the global navigation conditions and hence defines a framework that is independent from the local user-centric view. Amongst others, environmental contextual information includes navigation constraints following from the physical surroundings (e.g., architectural or geometric constraints) as well as from logical and thematic aspects (e.g., security zones or temporal access constraints), the available localization technologies, and the navigation scenario (e.g., navigation under normal conditions or in an evacuation situation). Both notions span a 2-dimensional context space with each dimension substantially affecting the task of navigation. In the field of context-aware computing and applications, even higher-dimensional context spaces are proposed which additionally consider, for example, the execution context of applications and resources (Schmidt et al. 1999, Mokbel & Levandoski 2009) or the time context (Chen & Kotz 2000).

A navigation system has to implement suitable methods for all identified navigation aspects. In the following, solutions applied in outdoor navigation as well as challenging tasks in indoor navigation are discussed separately for each aspect.

Localization. Most current outdoor navigation solutions employ absolute localization technologies based on global navigation satellite systems (GNSS) which provide location information by means of quantitative, measureable coordinates in a global spatial reference system (typically WGS 84), and hence identify absolute locations on the Earth’s surface. Alternative techniques measure the absolute position of a navigation user in cellular radio networks (e.g., mobile phone network) with respect to fixed reference points. Often such systems are combined with relative positioning methods such as dead reckoning (DR) which calculates the location of the navigation user relative to a previous position fix using measurements of the heading and distance travelled. However, and as stated in the previous section, GNSS based systems are generally not available indoors due to the absence of line of sight to satellites. Cellular network positioning methods commonly lack precision and accuracy in indoor environments, and dead reckoning is subject to integration drifts and significant cumulative errors (Kolodziej & Hjelm 2006).

Alternative indoor localization solutions are therefore being developed based on technologies like pseudo-satellites, Wi-Fi, Bluetooth, radio-frequency identification (RFID), infrared (IR), ultra-wideband (UWB), ultrasound, barcodes and quick response (QR) codes, or inertial navigation sensors as well as corresponding positioning algorithms. However, there is no commonly agreed standard system available yet, and usually no single technology provides sufficient degree of accuracy and precision as well as a continuous coverage of indoor space. Current approaches thus often apply specialized solutions or vendor-driven rigid compositions of a limited subset of technologies tailored to specific application demands in order to meet these requirements (e.g., Hightower et al. 2002, Retscher 2007). Instead, a comprehensive support for different and multiple localization technologies and methods is required to exploit their complementary strengths, but this also poses challenges. First, each technology is based on different types of sensors and usually is associated with its own local spatial reference system. Their integration hence requires a common representation model which abstracts from individual technologies. For example, most sensor types show comparable spatial characteristics such as visibility area, coverage area, or signal propagation area, which are suitable to serve as a common abstraction. The absolute position of a navigation user can then be derived from the known locality of sensors or senders and their covered areas. Second, each technology not only requires different installations within an interior built environment but also corresponding capabilities of the mobile end-user device. For example, even though both Wi-Fi and RFID infrastructures might be available in the same building and are supported by an indoor localization system, only Wi-Fi is feasible for localization and tracking if a specific end-user device is just Wi-Fi-enabled but not equipped with an RFID sensor. This renders a configuration problem whose degree of combinatorial complexity increases with the growing number of localization technologies available in an indoor environment and their varying support from different types of end-user devices.

A localization method benefits from an indoor space model in that only the latter is feasible to give both spatial context and semantic meaning to locations and hence to reason about location estimates. The positioning results thus need to be aligned with and communicated within the navigation space model. Positions are typically defined in the spatial reference system of the localization method (either a local or global coordinate system) which not necessarily coincides with the reference frame of the indoor space model. Moreover, the addressing schema for identifying locations is typically associated with its own spatial reference system which is often not a coordinate system but rather the natural reference system of the navigation user (e.g., in case of humans, mostly names, postal addresses, administrative areas, points of interest, etc.). Hence, multiple spatial reference systems may be involved in the navigation task and location information needs to be translated between these systems (Becker & Dürr 2005).

Path planning. Navigation relies on planning appropriate paths from a source to a destination location. This implies algorithms for finding optimal routes through the navigation space. Often time-dependent or length-dependent optimal search strategies are employed which deliver the fastest or shortest path between two locations. However, selecting the optimal path is subject to multiple criteria and factors which largely depend on both the user and the environmental navigation context, and thus may result in different paths even if the start and end locations are kept stable. For example, path searches for wheelchair users or mobile robots need to avoid impassable obstacles such as stairs, whereas stairs have to receive high priority in the context of pedestrians who prefer walking over using elevators. Likewise, and as illustrated in the previous section, in emergency situations additional passages may be usable which are not available in normal situations but also parts of the usual navigation space may be blocked. Thus, context-sensitive and adaptive search strategies need to be available which additionally consider user preferences and constraints, environmental and logical constraints, situation types, travel costs, simplicity, etc. (Goetz & Zipf 2011, Worboys 2011). Moreover, aspects such as route changes triggered by the user or by environmental state changes require the dynamic re-calculation of paths and thus need to be considered by navigation systems (Delling et al. 2009, Liu & Zlatanova 2011b).

Most current outdoor navigation systems use digital geographic maps of the outdoor environment which are available from various sources such as aerial and satellite imagery or existing map data serving as traditional base for many GIS applications. The availability of GNSS has additionally promoted the acquisition of outdoor data, and the resulting space models usually share the same global spatial reference system which simplifies localization tasks. In car navigation systems the physical road network is abstracted by a linear network structure which feeds path finding algorithms, whereas a digital road map is used for the display and communication of position, orientation, and route instructions. Modern car navigation solutions employ 3-dimensional road maps and models of urban spaces which facilitate the 3-dimensional presentation of the built environment or selected landmarks in order to support the visual recognition of places and the self-localization capabilities of humans. The space models generally support georeferencing of locations through lists of addresses and named places with their coordinates, and are capable of considering user preferences or environmental contextual information such as navigation constraints (e.g., speed limits, one-way roads, prohibited maneuver) or the current traffic situation (e.g., constructions sites, traffic jams) in path planning tasks.

Indoor spaces decisively differ from road spaces. They have complex 3-dimensional spatial structures and nested configurations, and involve multiple floors and levels with the architectural layout being the most obvious navigation constraint (e.g., rooms, corridors, doors, or walls). The possible movement is therefore richer, and users can travel more freely through the interior built environment than a linear network suggests. Models of indoor space for the purpose of navigation thus have to adequately represent the complex physical setting of the built-up space. In order to enable path finding, information about neighbouring places and the possibility of bodily movement between places needs to be available. Since most path finding algorithms generally presuppose some sort of network structure, additional graph-based conceptualizations of the indoor space have to be provided. However, the distinction between navigable and non-navigable spaces and appropriate route sections not just follows from the built reality, and thus not necessarily coincides with the decomposition of the interior environment along architectural constraints. In fact, whether indoor spaces are navigable also strongly depends on the type of locomotion of the moving person or object such as walking (e.g., pedestrians), driving (e.g., wheelchair users or mobile robots), and flying (e.g., autonomous flying vehicles such as quadrocopters). For example, obstacles or areas being insurmountable in the context of driving may be easily passable for pedestrians and flying vehicles. Thus, each mode of locomotion leads to a separate and distinct decomposition of indoor space into navigable spaces, which obviously affects the derivation of corresponding network representations (Khan & Kolbe 2012). This decomposition has to further consider navigation constraints imposed by the environmental context which result in non-navigable areas having their own spatial extent but which again may deviate from the architectural layout. For example, security zones possibly span several rooms or split a single room into accessible and non-accessible parts, and may even affect outdoor areas. Likewise, the decomposition of indoor space along the sensor characteristics of a localization infrastructure into, for example, sensor coverage areas is also obviously independent from the architectural structure. It follows that indoor space models need to support complementary space decompositions following from arbitrary contextual criteria and have to provide a suitable level of granularity for each space representation. A rich knowledge about the structural building elements and the various types of spatial entities that inhabit the indoor space (e.g., fixed or movable obstacles) as well as their spatial and functional properties and relationships is a necessary prerequisite for this.

The referencing of indoor locations in path queries has to support unique identifiers such as room numbers, descriptive names, or geometric coordinates, and has to be aligned with the spatial reference system and the granularity of the indoor space model, for example, to be able to name individual places in large halls (e.g., check-in counters or coffee booths in airport halls) or to identify spatial subdivisions within a single room that may reflect navigable and non-navigable areas for wheelchair users. Besides path queries, further location-based tasks such as nearest neighbour queries or range queries play a role in navigating through indoor spaces and correspondingly have to be addressed by the indoor space model (Becker & Dürr 2005). A nearest neighbour query aims at finding the k objects being closest to a given position (typically the current location of the user). To answer nearest neighbour queries, a notion of distance as well as a corresponding distance function needs to be modelled. The same is true for realizing the planning of shortest and fastest paths between two locations. Range queries, on the other hand, result in all objects being spatially contained within a given region such as all rooms on a given building floor. They hence reveal the hierarchical and nested configuration of indoor places and build upon a notion of containment that has to be provided by the indoor space model. Range queries are also important for the generation of routing instructions that adhere to and reflect the building hierarchy such as *"Enter the building, go to the fifth floor, and then to room 5126"*.

Similar to indoor localization, the task of path planning faces a high degree of combinatorial complexity in indoor environments due to different types of users and their mode of locomotion on the one side and the environmental context on the other side, and strongly depends on a rich model of indoor space.

Tracking and guidance. The tracking of moving persons and objects aims at tracing the trajectory of the movement or at determining deviations from a planned route. The process of tracking builds upon location estimates of the user from the available localization infrastructure and technology at given points in time and suitable algorithms for best matching the current location against the underlying navigation space model. Lacking precision and accuracy of the localization method or the space model as well as the transformation of location information between the involved spatial reference systems are common potential sources of error in this process (e.g., Liao et al. 2003). For outdoor navigation, corresponding methods and sophisticated map matching techniques involving probabilistic matching strategies are well established and successfully employed in navigation systems. The absolute position precision using publicly available GNSS based systems typically lies within the range of a few metres and thus suffices for applications such as car navigation. In micro-scale indoor environments, the required precision of the location determination may be higher depending on the spatial layout of the navigable indoor spaces and the user context, for example, to keep navigation users on track in open halls and narrow passages, or to safely navigate visually impaired persons or autonomous mobile robots through the interior environment. As mentioned in the previous section, the precise tracking of mobile robots based on highly specialized positioning technologies installed in the interior built environment or mounted on the robot has been a field of long study (e.g., Kruse & Wahl 1998, Fod et al. 2002). But also approaches for tracking people based on sparse and noisy sensor data (e.g., based on Wi-Fi, Bluetooth, or RFID) combined with probabilistic methods for deriving precise location estimates have been proposed, which then rely on an accurate indoor space model (e.g., Liao et al. 2003, Jensen et al. 2009). The tracking in indoor environments hence poses challenges to both the localization technology and the indoor space model, and a commonly accepted solution has not been presented to date. For navigation systems utilizing, for example, the self-localization capabilities of humans, also less precise indoor positioning methods may already be sufficient.

Route guidance supports human wayfinding from a source to a destination location, and typically means the provision of route instructions through visual or textual aids and descriptors. The guidance may take any form of presentation using digital or print media as well as further communications channels, and may be static or dynamic. Modern devices for car navigation, for example, commonly display the route together with symbolized route instructions on top of a cartographic representation of the digital road map and optionally offer spoken commands both of which are dynamically generated and updated while travelling. The generation of route instructions for car navigation is hereby strongly supported by the linear and strictly constrained movement in road spaces as well as clearly determinable decision and action points (e.g., road intersections, motorway exits and ramps, etc.). The complex structure of the interior built environment as well as the possibility to move rather freely renders the appropriate presentation of an indoor path as well as the provision of instructions along that path a much more challenging task. This is also impeded by the fact that the human spatial perception and knowledge acquisition in indoor spaces differ from the outdoor world which is partly due to substantial environmental differences (Raubal et al. 1999, Gilliéron & Merminod 2003). For example, the role of landmarks is different indoors. Whereas outdoor landmarks are typically visible from various places and hence provide a fixed reference frame independent from a travelled route, indoor landmarks are smaller in size and shape (e.g., fountains, lobbies, or walls painted in noticeable colours) and suffer from limited lines of sight due to the architectural complexity. Moreover, street names or numbering conventions for houses along a street are salient cues in outdoor spaces which give a cognitive structure to the space and promote spatial reasoning, for example, by deducing the direction of movement from ascending or descending house numbers. Such cues or meaningful reference systems are usually rarely present in indoor spaces. People thus tend to learn a specific route or a given sequence of route commands, especially in case of unknown interior environments, rather than building a cognitive map of the indoor space (Giudice et al. 2010).

Visual presentations of the layout topology of buildings are therefore important aids in guiding people indoors. Current research questions include the type and amount of salient building features and cues to be displayed, the definition of a common set of descriptors and symbols for route instructions, the proper cartographic representation of the interior environment using, for example, layered 2-dimensional floor plans up to 3-dimensional views, or the presentation technique which ranges from static paper-based maps to handheld displays facilitating real-time assistance or augmented reality solutions. Currently there is no consensus on these questions and the investigation and understanding of human spatial learning and wayfinding processes is commonly seen as a necessary prerequisite for the acceptance of visual aids (Giudice et al. 2010, Lorenz et al. 2010). Further challenges of indoor route guidance address the automatic generation of human-understandable descriptive route instructions or the use of appropriate media according to the physical and perceptual capabilities of the navigation user (cf. Lorenz et al. 2006, Anagnostopoulos et al. 2005). A key factor common to all these challenges is the underlying indoor space model that serves as basis for the derivation of guidance information and hence has to provide a rich semantic and spatial description of the interior built environment.

Navigation context. The importance and impact of the navigation context on the individual aspects of indoor navigation is documented in the above discussion and has also been recognized in many research works (e.g., Gilliéron & Merminod 2003, Anagnostopoulos et al. 2005, Stoffel et al. 2007, Dudas et al. 2009, Goetz & Zipf 2011, Afyouni et al. 2010, Yuan & Schneider 2011). An indoor navigation system has to cope with the many use cases and configurations which constitute the navigation context and mainly result from the different and varying localization techniques and infrastructures available in the indoor environment and their ad-hoc selection through end-user devices, as well as from complementary indoor space decompositions and navigation restrictions following from environmental and user-dependent constraints and preferences (Becker et al. 2009b). The support for contextual information hence requires a formal model which captures the knowledge about the navigation context and facilitates reasoning and inference about this knowledge, and is further confronted with dynamic changes of the user or environmental state (Brown et al. 2012). For example, the access rights of a navigation user may change while travelling through the interior environment (e.g., when passing a security gate at an airport) which directly affects the navigable and restricted spaces that may be considered in subsequent path searches. Likewise, if a building is equipped with different localization systems and the sensors of one or more system (partly) break down then the positioning of moving persons or objects must still be possible with the remaining infrastructure. Or the navigation scenario may change from normal conditions to an evacuation situation which may require re-routing all persons to the nearest exists. The examples also demonstrate the need for a strong coupling of the contextual information with the indoor space model.

1.3 Research Scope and Goals

The research in this thesis concentrates on the modelling of indoor space for the purpose of indoor navigation. The crucial role of an indoor space model within the task of indoor navigation and its relevance in the context of the individual navigation aspects has been revealed in the previous section.

The overall goal of this thesis is to design a *spatio-semantic model of indoor space* that meets the multiple challenges to indoor space modelling and thus can serve as solid foundation for the implementation of indoor navigation systems. This goal can be refined into three subgoals.

Conceptual model of indoor space. The first subgoal is to define a *conceptual model* of indoor space which allows for structuring the indoor space along arbitrary notions of space and for consistently describing the semantic and spatial aspects of spaces. Commonly, indoor space is merely understood as the built environment in which people usually behave (cf. Afyouni et al. 2012). Although, of course, the built architectural reality sets the frame in which the bodily movement takes place, the discussion in the previous section clearly documents that indoor space is also subject to alternative structuring and partitioning schemas in the context of indoor navigation which are not necessarily aligned with the spatial location and extent of the architectural entities. Thus, a main aim of this thesis is to define a generic notion of indoor space that abstracts from the built reality and thereby allows for different but complementary space partitionings. Based on this universal view, further concepts are to be developed for the integration of the different space representations in a common model that simultaneously supports route planning, multiple localization methods, navigation contexts and constraints, and different locomotion types. This development has to be preceded and supported by a review of previous and salient work on indoor space modelling in order to assess to which extent and by which means the identified challenges have already been addressed or possibly been solved in literature. From this review, essential requirements are to be deduced that set the outline for the conceptual definition of the spatio-semantic indoor space model in this thesis.

Mathematical model of indoor space. As a second subgoal, a *mathematical model* is to be defined which formalizes the conceptual model and embeds its conceptual entities in a sound mathematical framework. A correct, consistent, and complete mathematical formalization is to be seen as a key prerequisite for the definition of a universal view of indoor space. A main focus is hereby on the formal definition of the geometric-topological aspects of indoor space and the deterministic derivation of graph-based conceptualizations which, for example, can be fed into path search algorithms.

Computer representation. The third subgoal aims at designing a *computer representation* which maps the conceptual and mathematical model onto a data model and data exchange format based on international standards from the geoinformation community. The adherence to international standards admits an open and vendor-neutral computer representation. By this means, the spatio-semantic indoor space model can be implemented by and exchanged between computer systems in a standardized way. The separate subgoals of this thesis hence address different aspects of the main problems of indoor space models in the field of indoor location-based services as motivated in chapter 1.1 and together aim at overcoming them.

The research of this thesis builds upon results from previous research conducted at the Institute for Geodesy and Geoinformation Science of the Technische Universität Berlin. From 2007 to 2011, an ambitious multinational research project entitled *"Indoor Spatial Awareness"* (ISA) was carried out whose vision was to provide a basis for an indoor spatial theory and to develop data models and application systems for enabling indoor spatial awareness in the context of location-based services. The ISA research initiative was supported and funded by the Ministry of Land, Transportation, and Maritime Affairs of the South Korean government, and both research institutions and companies from South Korea, the United States, Germany, and Denmark participated in this project (cf. Li & Lee 2010). The author was involved in the ISA project from its very beginning as member of a research team from the Technische Universität Berlin led by Prof. Dr. Thomas H. Kolbe and substantially contributed to the research work of this team. In the course of the ISA project, this team proposed a novel conceptual framework called *Multilayered Space-Event Model* (MLSEM) for the modelling of indoor space which considers the aspects of route planning for different modes of locomotion and navigation contexts on the one hand and of localization and tracking based on various localization techniques on the other hand. It hence addresses many of the challenges to indoor navigation identified in the previous section. The previous research on the MLSEM has been introduced to the scientific community in two subsequent peer-reviewed publications, namely (Becker et al. 2009a) and (Becker et al. 2009b), both of which served as basis and starting point for the research work presented in this thesis. The primary aim was hereby to realize the above mentioned research goals on top of the MLSEM. A detailed discussion and presentation of the MLSEM is provided in the subsequent chapters of this thesis and thus is omitted here.

As a second deliverable of the ISA project, a candidate discussion paper on the MLSEM (Nagel et al. 2010) was submitted to the *Open Geospatial Consortium* (OGC) which is one of the most important open, non-proprietary, and consensus-based standardization bodies at an international level in the geoinformation community. The aim of this discussion paper was to generate momentum for the development of an international standard on indoor navigation with the MLSEM at its core. In December 2010, the OGC Technical and Planning Committee approved and released the discussion paper which started an on-going standardization process within OGC. A further non-research goal of this thesis is therefore to contribute to this standardization process which relates back to the postulation put forward by (Kolodziej & Hjelm 2006) according to which standardization is a fundamental prerequisite for the success of the field of indoor navigation.

1.4 Research Hypotheses and Questions

The research in this thesis evolved along the following research hypotheses and questions.

Hypothesis 1.1. The conceptual model of the MLSEM as defined in the previous publications is complete.

The previous work on the MLSEM already defines a generic conceptual model for the spatial and semantic description of indoor environments. But it is not clear whether the proposed conceptualization suffices the needs and challenges to indoor space modelling in the context of indoor navigation (cf. chapter 1.2). The hypothesis is therefore verified by testing the concepts against these challenges and against related work. For requirements that are not met by the MLSEM, extensions to the conceptual model are investigated and proposed in this thesis. The mathematical definition and computer representation of the MLSEM are targeted by the following hypothesis.

Hypothesis 1.2. The MLSEM can be formalized in a mathematically sound manner and can be mapped onto a data model that conforms to existing modelling standards for geographic information.

The mathematical formalization is not sufficiently addressed in the previous work on the MLSEM. Likewise, the MLSEM has not been completely realized as data model so far since important concepts of its conceptual model have been neglected. Both aspects therefore render a main research task of this thesis.

Hypothesis 1.3. The MLSEM enables indoor navigation systems. It is a conceptual superset of many existing approaches to indoor space modelling for indoor navigation and can be used to explain these approaches.

The first part of this hypothesis substantiates the main goal of this research. The second part is tested within the thesis along the following research questions which address the three fundamental aspects of indoor navigation (cf. chapter 1.2) as well as related tasks of an indoor space model. Since these questions are at least partially answered in alternative approaches to indoor space modelling, they must also be covered by the MLSEM in order for it to be considered as a conceptual superset of these approaches.

Question 1.4. How are localization, route planning, as well as tracking and guidance in indoor environments supported by the MLSEM?

Question 1.5. How is the modelling of navigable spaces as well as non-navigable areas and obstacles for different locomotion types and user contexts supported by the MLSEM?

Question 1.6. How are multiple and different localization technologies and methods supported by the MLSEM?

Question 1.7. How is the modelling of hierarchical and nested structures of the built environment supported by the MLSEM?

Question 1.8. How is the context of navigation of a moving person or object represented in the MLSEM? How does the MLSEM support the ad-hoc selection of appropriate space representations according to the context of navigation?

Question 1.9. How can user-dependent and environmental navigation constraints be conceptually modelled and included in the MLSEM? How to support dynamic aspects such as temporal navigation constraints?

Geoinformation about the interior built environment of buildings and facilities becomes more readily available in the context of urban and building information modelling. Data about buildings is typically represented according to different building modelling standards and is stored and exchanged using different data formats. The resulting building models typically differ in the dimensionality of the geometric representation of the interior built environment (either 2-dimensional or 3-dimensional) and the spatio-semantic information. Nevertheless, they are to be seen as a valuable information source that feeds into an indoor space model for indoor navigation. The following hypothesis targets the need to support such input models.

Hypothesis 1.10. The MLSEM is valid in both two and three dimensions and can be populated from existing (and even purely geometric) building models.

In the previous publications on the MLSEM, the spatial description of indoor space has been defined for three dimensions only, thus neglecting 2-dimensional representations of the interior built environment. Its extension to two dimensions is looked at in thesis in order to verify the above hypothesis. Moreover, the relation of the MLSEM to standards from the field of building modelling such as IFC, CityGML, and ESRI BISDM has to be investigated. Since indoor navigation is typically not the primary purpose of such building models, this involves the following research question.

Question 1.11. What is the relation of the MLSEM to existing models providing geoinformation about the interior built environment such as 2-dimensional floor plans or 3-dimensional building models? How to link such models without duplicating or replacing their concepts?

A complementary group of international standards addresses the specification and realization of location-based service architectures. These standards mainly aim at defining the distributed system architecture as well as the components and their interfaces that are required to build an LBS computing environment. Since the scope of this thesis is to develop an indoor space model that underlies indoor navigation services, the role of this model within a location-based service infrastructure as well as its implications on service interfaces has to be investigated. Moreover, some of these standards also define a navigation space model. The applicability of the MLSEM within these standards is therefore another issue. Both aspects are summarized in the following research question.

Question 1.12. What is the relation of the MLSEM to existing international standards on location-based service architectures?

1.5 Organization of the Thesis

The thesis is organized into eight chapters whose contents are outlined in the following.

Chapter 2 identifies fundamental requirements for the modelling of indoor environments for the purpose of indoor navigation and hence sets the basis for the definition of a spatio-semantic indoor space model in this thesis. The requirements are deduced from a comparative analysis and review of related work on indoor space modelling. Since the mathematical formalization of the spatio-semantic indoor space model is one of the main goals of this thesis, basic mathematical notions and concepts which serve as foundation for this purpose are presented afterwards.

Chapter 3 develops the spatio-semantic indoor space model based on the previous work on the Multilayered Space Event-Model and against the requirements formulated in chapter 2. In this core chapter of the thesis, both the conceptual definition and the sound mathematical formalization of the model are elaborated in depth. Two fundamental concepts, namely *space cell* and *space layer*, are defined and mathematically embedded which realize a generic model that allows for structuring the indoor space according to arbitrary notions of space and for describing its semantic and geometric-topological aspects as well as graph-based conceptualization. Based on this formal model, further concepts such as the integration and joint consideration of multiple space layers in a common model based on a combinatorial *multilayered graph* structure and the modelling of hierarchical space structures are defined and formalized. A formal *space layer algebra* is developed which allows for applying well-defined binary operations on space layers. It is shown in detail how the concepts developed in this chapter relate to the challenges and requirements from chapter 2 and thus support the individual tasks in indoor navigation.

Chapter 4 is dedicated to the mapping of the indoor space model onto a conceptual data model and physical data exchange format. General spatial representation schemes for the computer modelling of geometric objects are presented and discussed, and geometric-topological data models proposed in the field of GIS are reviewed against their capability to fully map the spatio-semantic aspects of the developed indoor space model. Focus is then put on the ISO 19100 series of international standards for the modelling of geographic information issued by ISO/TC 211, and their feasibility in the context of the developed indoor space model is argued. Based on this, an ISO-conformant and complete conceptual data model is designed, formally expressed in UML, and presented in detail. In a subsequent engineering step, a data encoding and exchange format is derived. The chapter concludes with a discussion on how the developed conceptual data model embeds into existing standards on location-based service architectures.

Chapter 5 addresses the consideration of environmental and user-dependent navigation constraints within the developed indoor space model. Navigation constraints are a crucial aspect of indoor navigation and their relevance is commonly agreed in literature. However, conceptual models for the representation and evaluation of navigation constraints rarely exist. Based on a literature review, different categories of navigation constraints (e.g., *can*, *may*, *must*, and *should* constraints) are identified and concepts for describing the preconditions under which the constraints apply are developed and classified into being *temporal*, *physical*, or *logical*. From this, a conceptual data model is designed that fits into the framework elaborated in chapter 4 and allows for explicitly expressing navigation constraints on single or multiple space entities and for algorithmically evaluating them against the context of individual navigation users. The feasibility of the developed model is then demonstrated along typical use cases for navigation constraints in indoor environments.

Chapter 6 elaborates on the relation of the developed indoor space model to selected modelling approaches and standards for 2-dimensional floor plans and 3-dimensional building models. It is shown how the spatial and semantic description of the architectural entities within such models can be used to populate a topographic view on indoor space that satisfies the conceptual and mathematical definition of the developed indoor space model. Focus is hereby put on the different spatial modelling paradigms that are applied in the building models and their consequences in the mapping process. The mapping results are compared and proposals for minimizing their differences are presented. Moreover, elements of the conceptual model (if available) of the building modelling approaches are identified with conceptual model elements of the developed indoor space model.

Chapter 7 illustrates the application of the developed indoor space model along two examples. The first example presents the XML encoding of an artificial 2-dimensional indoor scene according to the data model evolved in chapter 4. The XML structures capturing the most important concepts for the spatio-semantic description of indoor space from chapter 3 are discussed in detail in order to show the complete and consistent computer-based mapping of these concepts which is a prerequisite for their lossless exchange between computer systems (e.g., in an LBS computing environment). The second example deals with the acquisition of a 3-dimensional indoor space model that is suitable for indoor navigation for a real world building. Moreover, a database schema for a relational spatial database is presented that realizes the conceptual data model from chapter 4 and thus allows for the storage and management of indoor space models as defined in this thesis.

Chapter 8 summarizes the research and draws conclusions with respect to the stated research goals as well as hypotheses and questions. It reviews and evaluates the results of the research, lists and discusses contributions to the field of indoor navigation as well as related fields, and identifies and outlines future research.

In addition to these eight chapters of the thesis, five appendices with supplementary material are provided. *Appendix A* introduces basic notions and concepts from point-set and algebraic topology as well as manifold theory which form the basis of the mathematical formalization in chapter 3. Although the content is assumed to be known, the terms, symbols, and definitions used in literature differ. The appendix hence provides a consistent and comprehensive presentation of the theory applied in this thesis. A brief summary overview of the theory is given in chapter 2. *Appendix B* presents the GML application schema that was derived from the conceptual data model developed in chapter 4. *Appendix C* contains the formal SQL definition of the relational database schema introduced in chapter 6. *Appendix D* illustrates the conceptual data model of the MLSEM as defined in previous publications (cf. Becker et al. 2009b, Nagel et al. 2010). *Appendix E* finally shows the initial conceptual data model for navigation constraints as proposed in (Brown et al. 2012).

Chapter 2
Analysis of Related Work

The modelling of indoor space for the purpose of indoor navigation is a field of intensive research in academia and industry. Numerous approaches have been proposed to date ranging from models that reflect the human spatial perception of indoor space in order to support human wayfinding to a representation of indoor space that is suitable for use with autonomous mobile robots and artificial intelligence systems. The approaches differ in the methods and techniques being applied for the spatial description of the interior environment as well as in the richness of the semantic information about the environment.

This chapter is dedicated to a survey of related work which aims at clarifying the fundamental concepts and principles as well as information needs for indoor space models. In a first step, a common classification scheme is introduced which identifies opposing methods for the modelling of the built environment and the spatial entities therein. Along this scheme, the role of quantitative, qualitative, and semantic facts about the spatial entities and their relationships in the context of indoor navigation is elaborated. In a second step, existing approaches to indoor space modelling are discussed and analysed with respect to their conceptualization of indoor space, their spatio-semantic model expressivity, and their ability to support the general tasks in navigation, namely localization, route planning, tracking, and guidance. The review identifies shortcomings as well as strengths of related work in addressing the challenges to indoor navigation as presented in chapter 1.2. Besides navigation space models, also standards from the field of urban and building information modelling are presented and examined in this chapter. On the one hand, it is shown that building models typically lack the required concepts to directly support indoor navigation. On the other hand, spatio-semantic information about the built-up space and the spatial entities therein which is required in the context of indoor navigation can often be derived from these models to a large extent. Based on the findings, general requirements are postulated which need to be addressed in the development of an indoor space model in order to meet the identified challenges and to support a general view on indoor space.

A second aim of this chapter is to give an overview of basic mathematical notions and concepts from topology, manifold theory, and graph theory which serve as foundation for the subsequent chapters of this thesis.

2.1 Classification of Indoor Space Models

Models of indoor space are commonly classified into *symbolic* and *geometric space models* (e.g., Leonhardt 1998, Becker & Dürr 2005, Ye et al. 2007, Li & Lee 2009b, Baras et al. 2010). This classification reflects two fundamental approaches to the representation of indoor environments. Symbolic space models represent physical places and indoor objects through sets of abstract symbols (e.g., human-friendly descriptive names), and express qualitative spatial relationships between them. In geometric approaches, spatial locations and objects are rather modelled as n-dimensional geometric figures which are given by a set of coordinates within a well-defined coordinate reference system. The metric properties of the geometric figures allow for quantifying the spatial relationships between model elements. Both modelling approaches are orthogonal and show complementary strengths and weaknesses. A third class of indoor space models describes the indoor environment from a conceptual perspective that is inferred from the types of spatial entities and their spatial and non-spatial relationships (e.g., Hu & Lee 2004, Bhatt et al. 2009, Raubal et al. 1999, Yang & Worboys 2011). Models from this class are said to be *semantic space models*, as they aim at defining the meaning of the identified concepts. Finally, *hybrid space models* combine symbolic, geometric, and semantic representations to benefit from their respective advantages. The following sections give an overview of the different modelling paradigms.

2.1.1 Symbolic Space Models

Symbolic space models provide a qualitative description of the interior built environment. Indoor spaces are referred to by abstract symbols which are commonly given as descriptive labels or identifiers such as room numbers or names (e.g., *"Room 5126"* and *"Living Room"*). The spatial structure of an indoor environment is hence mapped onto a conceptual structure which employs a human-friendly naming system. Symbolic spaces typically identify the physical places and structural components of a building which enable the movement through space like, for example, rooms, doors, corridors, floors, elevators, etc. Therefore, each space symbol has an implicit spatial extent but does not explicitly represent the size and shape of its associated space. Representative examples of symbolic space models have been proposed by (Brumitt & Shafer 2001), (Hoppenot et al. 2003), and (Baras et al. 2010).

Space symbols can be organized into sets according to the architectural subdivision of the building interior. As a simple example, a building can be expressed as set of all its room symbols. A floor of the building is then given by a subset containing only room symbols on that floor. (Becker & Dürr 2005) and (Ye et al. 2007) show how this *set-based* modelling approach facilitates the formal evaluation of spatial relationships between symbolic spaces using basic set operations. The non-empty intersection of two sets of symbols, for example, determines a spatial *overlap* between the corresponding physical spaces, whereas set inclusion implies spatial *containment*. A set-based symbolic model can also be used to express *connectedness* relationships by modelling fine-grained neighbourhood sets containing the symbols of connected physical spaces. This introduces a qualitative notion of *closeness* between a symbol and a given set. Since a set of symbols denotes a physical space in its own right (e.g., the floor of a building) it can also be represented by its own space symbol (e.g., the floor number). The spatial containment relationship then imposes a partial order on the set of all space symbols. In case the symbolic spaces are constrained to be non-overlapping, the resulting partially ordered set forms a tree structure with the root node being the symbolic space that spatially contains all others. In the more general case where overlapping symbolic spaces are allowed, the partially ordered set can be viewed as lattice in which any two space symbols have a least upper bound and a greatest lower bound. Both the tree and the lattice structure hence make the spatial *containment hierarchy* of physical spaces explicit. Hierarchical symbolic space models have been employed by, e.g., (Leonhardt 1998), (Ye et al. 2007), and (Li & Lee 2009a). Besides the spatial or structural organization of symbolic spaces, (Richter et al. 2009) propose functional and organizational hierarchies as additional dimensions.

A second family of symbolic space models discussed in literature applies *graph-based* models for the encoding of space symbols and their spatial relationships instead of sets and subsets (e.g., Hu & Lee 2004, Baras et al. 2010). Graph-based conceptualizations of space have a long tradition in architectural analysis as well as in human cognitive science and artificial intelligence (for a survey see Franz et al. 2005). In the context of navigation, *place graphs* and *visibility graphs* can be identified as two distinct concepts for the representation of space which underlie most research works. For both types of graphs, every symbolic space is mapped onto a separate node. The edges in place graphs then denote the *connectedness* between symbolic spaces, and hence implicitly between the associated physical spaces. Connectedness often is further differentiated into *adjacency*, *connectivity*, and *accessibility*, which induces different semantics for graph edges (Worboys 2011). The edges of an adjacency graph simply express that physical spaces are adjacent, e.g., that rooms share a common boundary wall. In connectivity graphs, every edge additionally implies the possibility to physically move between the spaces. If spaces are inaccessible or the movement between spaces is not permitted, then the corresponding symbolic nodes are not linked by edges in accessibility graphs. Access or movement restrictions may be the result of spatial facts but may also reflect logical constraints such as security or safety constraints. It obviously follows that accessibility graphs are subgraphs of connectivity graphs which themselves are subgraphs of adjacency graphs.

Visibility graphs, on the other hand, translate the *mutual visibility* relationship between physical spaces into edges between the symbolic space nodes (Franz et al. 2005). This visibility relationship naturally is close to the human perception of indoor space. Visibility graphs have thus been proposed in several approaches to indoor space modelling (e.g., Stoffel et al. 2007, Yuan & Schneider 2010a, Liu & Zlatanova 2011a). In the architecture domain, (Hillier et al. 1983) and (Hillier & Hanson 1984) have proposed the *space syntax* analysis as a set of techniques for describing and quantifying the interrelationships between spatial organization patterns at urban or building scale and the underlying social structures in order to understand space from a functional perspective in terms of what humans do in it. The built environment is decomposed into a near minimal set of 2-dimensional convex subspaces such that all points are directly visible from all other points within a single space. The fewest and longest lines that pass through all the spaces then generate an *axial map* of the spatial configuration. Each axial map can be connected into a graph where the nodes describe the lines of sight or straight movement and the binary edges denote their mere intersections (e.g., Desyllas & Duxbury 2001). In this sense, axial maps capture the global constituents of a spatial layout as perceived by humans while moving through the space. Based on obtained graph measures, space syntax theory and axial maps have been successfully applied in studying human movement patterns but are also used in in other fields such as the analysis of traffic flows or crime distribution (e.g., Turner et al. 2005).

Graph-based symbolic space models are purely topological structures. Based on the explicit representation of connectedness relationships between spaces, place graphs facilitate path queries in indoor environments. A path query results in a sequence of symbols, which corresponds to a traversal of the graph from a start to a goal symbol (Hoppenot et al. 2003). The descriptive place labels support the process of human wayfinding, and visibility graphs ensure the intervisibility of consecutive symbolic spaces along the path. Nodes and edges of graph-based representations can be attributed to carry additional contextual information such as navigation rules or restrictions. Moreover, edge weights allow for introducing a qualitative notion of distance between symbols. Symbolic place graphs commonly imply a segmentation of indoor space into non-overlapping physical spaces. Spatial containment hierarchies between physical spaces are hence not reflected. However, the containment relationship enables range queries between symbolic indoor spaces. Set-based symbolic models naturally support range queries but require the modelling and evaluation of a potentially large number of neighbourhood sets to provide path planning capabilities. For this reason, lattice- or tree-based hierarchical organizations of indoor space have been combined with graph-based symbolic representations (e.g., Becker & Dürr 2005, Ye et al. 2007). (Baras et al. 2010) demonstrate how symbolic place graphs can be annotated to also reflect spatial containment hierarchies.

Purely name-based symbolic space models inherently address human indoor navigation. Descriptive labels for indoor spaces support human location-awareness, and provide a human-friendly reference frame for the naming and addressing of locations in path or range queries. The granularity of the space representation is however limited by the description level of indoor spaces. Symbolic space models express the qualitative fact of connectedness and containment of symbolic spaces, but lack metric information which is required for a quantitative notion of distance or orientation between symbols. Shortest distance queries as well as guidance along the path based on metric information are hence not supported. Moreover, most approaches applying symbolic space models do not present formal methods for the automatic derivation and spatial verification of space symbols.

2.1.2 Geometric Space Models

Geometric space models describe the interior built environment as n-dimensional metric space (typically Euclidean n-space, cf. definition A.27) with one or more well-defined coordinate reference systems. The environment is decomposed into a finite number of non-overlapping *cells* which are given as 2-dimensional or 3-dimensional geometric figures. In contrast to purely name-based symbolic models, cells are represented by a set of coordinate tuples. Two general methods for the decomposition of space can be distinguished: *regular* cell decompositions map the indoor environment onto a regular array of predefined cells having equal size and shape, whereas *irregular* cell decompositions aim at computing cells whose union is exactly the free space. Cells participating in an irregular decomposition can be of different shape and size.

Prominent examples for regular geometric space models are *grid-based models* which mostly decompose the indoor space into rectangular (or cuboidal) cells. The grid is commonly chosen to be continuous, so that the entire indoor space is covered by cells. Each cell can then be marked whether it is occupied by an obstacle or free space. Obstacles comprise built structures and objects that obstruct movement such as walls or immovable furniture. A regular grid implicitly represents a graph structure where every grid cell is a node with rectilinear connections to its surrounding neighbours. The edges hence express *adjacency* relationships between cells. Due to its metric embeddedness, the resulting *grid-based graph* can be searched for shortest paths. Occupancy grids have been applied in the field of artificial intelligence as static or dynamically changing space representation for autonomous robots (e.g., Moravec & Elfes 1985, Thrun & Bü 1996, Biswas et al. 2002), as well as in human indoor navigation (e.g., Bandi & Thalmann 1998, Lyardet et al. 2006, Li et al. 2010, Afyouni et al. 2010, Yuan & Schneider 2011).

The advantage of regular grid-based models is that they are considerably easy to build, represent, and maintain even in large-scale environments (Thrun & Bü 1996). However, the structural rigidity of grids only allows for an approximate representation of indoor space whose accuracy is determined a priori by the grid resolution. For example, cells on the boundary of arbitrarily shaped obstacles suffer from partial membership if they cover the surrounding free space as well, and thus their classification into either obstacle or free space results in jagged obstacle boundaries. Passages or obstacles being smaller than the predefined cell size are possibly not represented at all. Fine-resolution grids capture the details of the environment more accurately, but require an exponentially growing number of cells and thus suffer from their increasing computational space and time complexity. Hierarchical 2-dimensional *quadtrees* address this conflict by recursively decomposing the grid into quadrants until all cells in one quadrant distinctively denote obstacle or free space (Ali & Abidi 1988, Lyardet et al. 2006). The resulting multi-resolution grid implicitly expresses a *containment hierarchy*. Tree-based grid structures are more space efficient than fine-resolution grids, and path queries benefit from the pre-sorted cell organization. *Octrees* achieve a similar hierarchical structure in three dimensions, and much work has been done in investigating further variants and optimizations of hierarchical grid structures (e.g., Eppstein et al. 2005).

Irregular geometric space models only partition the free indoor space and apply exact cell decomposition methods in order to overcome the approximate space representation of regular approaches. Often *trapezoidal decompositions* or *triangulations* are proposed, where the boundaries of built structures and obstacles are required to be boundaries of the resulting cells which thus differ in shape and size. Examples for triangulation-based approaches are presented by (Lamarche & Donikian 2004) and (Demyen & Buro 2006) who employ a constrained Delaunay triangulation of 2-dimensional floor plans to automatically compute the irregular subdivision of free space. In a subsequent step, (Lamarche & Donikian 2004) show how to simplify this subdivision and to minimize the number of cells by merging the triangles into convex cells. Similar to grid-based models, the irregular cell decomposition can be translated into a geometrically embedded *adjacency* graph which facilitates shortest path searches. Every cell is commonly mapped onto a node and an edge represents a free boundary segment shared by two adjacent cells. Alternatively, nodes are considered the midpoints of such segments which are connected by linear paths traversing each cell (Lamarche & Donikian 2004). A second type of irregular geometric space models decomposes the free space into a *generalized Voronoi diagram*. The boundaries of the Voronoi cells are exactly the locus of points equidistant to the two nearest obstacles, and form the edges of a graph whose nodes are the points where two or more cell boundaries meet. This graph is a 2-dimensional medial-axis transform of the free space and provides obstacle-free pathways through indoor space. It therefore underlies many approaches in the field of robot path and motion planning (e.g., Choset et al. 2000, Liao et al. 2003, Wallgrün 2010). A natural extension is the *generalized Voronoi graph* which is the 1-dimensional locus of points in an n-dimensional metric space equidistant to n obstacles (Choset et al. 2000).

Unlike regular approaches, irregular geometric space models provide an accurate representation of the free space in complex indoor environments. Although obstacles and built structures are not explicitly represented as cells themselves, their exact boundaries as well as narrow passages between them are preserved. Since the resolution of irregular cell decompositions is only determined by the complexity of the indoor environment, usually fewer cells are required which results in more compact adjacency graphs compared to regular models. When merging adjacent irregular cells, containment hierarchies can be established. However, irregular approaches are more expensive to construct and maintain in large environments. In order to benefit from their complementary advantages, (Thrun & Bü 1996) propose an integrated approach that generates an irregular subdivision on top of a regular grid which classifies the free space.

Geometric space models facilitate the precise quantification of position, distance, and orientation, and hence provide the means for shortest path computations as well as range and nearest neighbour queries. A key prerequisite is the unambiguous recognition of cells with respect to the given coordinate reference system. In case various coordinate reference systems are used (e.g., separate local reference frames for each floor of a building), unambiguity requires well-defined transformations between them, or otherwise distance measurements become meaningless. The geometric representation of the indoor environment can also be used for visualizations to efficiently guide humans along their path or to display the results of navigation queries. The discretization of indoor space into non-overlapping cells is well suited to evaluate and derive the connectedness between cells, and hierarchical containment relationships can be built on top. Since the space partitioning is strictly geometric, the resulting cells typically do not coincide with symbolic places such as rooms or corridors but have finer granularity. On the one hand, this is an advantage over symbolic space models, as it allows the explicit modelling of indoor areas for which there is no symbolic label. On the other hand, such areas can only be addressed and named through their coordinates which are machine-understandable but non-intuitive for humans. For this reason, purely geometric space models are mostly adopted in mobile robot navigation. In contrast to most symbolic approaches, the decomposition of indoor space along geometric methods (e.g., regular grid, constrained Delaunay triangulation, generalized Voronoi diagram, etc.) has been mathematically formalized and thus can usually be derived in an automatic and deterministic way.

2.1.3 Semantic Space Models

Both symbolic and geometric space models describe the spatial configuration of a specific indoor environment. Whereas geometric models employ an explicit coordinate-based representation of a given indoor space, symbolic models implicitly define the physical places within that space using qualitative labels. The labels provide an intuitive meaning to the associated spaces. For example, *"Room 5126"* implies a place in a building which typically is bounded by built borders and likely is located on the fifth floor. *"Entrance hall"* is intuitively distinct and rather associated with an open or semi-open space located on the ground floor. From the textual descriptions, we can even deduce action possibilities offered by the spaces. An entrance hall is a place where people enter the building and which commonly is open to the public, while access to a room may be restricted to authorized personnel.

The exemplified spatial understanding of symbolic labels as well as the qualitative reasoning about meaning and location is based on common-sense geospatial knowledge. However, neither symbolic nor geometric approaches provide a formal representation of this knowledge. Semantic space models define space from a conceptual perspective in contrast to the qualitative respectively quantitative views of space in symbolic and geometric space models. They infer abstract concepts from the relevant types and aspects of spatial entities that inhabit indoor spaces and define their meaning as well as their properties and relationships (Worboys 2011). This conceptualization of indoor space is not tailored to specific indoor settings but provides both an abstraction of the physical built reality and a semantic reference frame for the entities therein (e.g., symbolic labels may relate to this semantic frame). Thus, semantic approaches aim at making the implicit spatial knowledge explicit and enable knowledge to be shared and reused.

Semantic space models classify the spatial and structural entities of indoor space and determine distinctions. This classification renders a *taxonomy* and arranges the types (or classes) of entities in a hierarchical structure along *is-a* relationships. For example, a classification scheme for navigational purposes could organize the indoor space into built structures supporting movement (e.g., rooms, corridors, floors, etc.) or constraining movement (e.g., walls, obstacles, doors, etc.). Further semantic relationships commonly represented in conceptual views of space identify *part-whole structures*, referred to as *part-of* relationships, or general *associations* which denote membership or *has-a* relationships. Part-whole structures allow for denoting semantic decomposition hierarchies (e.g., a room is part of a floor which itself is part of a building) and induce a partial ordering on the involved entity types. Qualitative spatial relationships are often introduced to express and constrain the spatial configuration of the indoor entities. Further association relationships as well as properties may account for terminological, geometrical, temporal, functional, or further thematically different perspectives of indoor spaces.

In computer science, two distinct approaches to the formal specification and representation of semantic knowledge are commonly applied: *conceptual (data) models* and *ontologies*[3]. Conceptual models have risen in the field of software engineering and information science as important technique to map the real world aspects (i.e., concepts with their properties and relationships) of a given domain of interest in an explicit, unambiguous, and machine-readable way (cf. Booch 2007). Conceptual models are widely adopted in the GIS domain and provide the necessary means to describe semantic knowledge about indoor environments. An alternative formal knowledge representation is that of ontologies which have emerged from the field of artificial intelligence and underpin a large number of semantic models. Similar to a conceptual model, an ontology is an abstract and simplified but also commonly accepted view of some domain whose representational primitives are typically classes (or "things"), properties, and relationships (Bhatt et al. 2009, Giudice et al. 2010). The assumptions about the primitives are mostly described as unary and binary predicates in first-order logic. Besides representation, ontologies hence provide reasoning and inference capabilities about semantic makeups and spatial configurations (Worboys 2011). Although this is often cited as the main difference between both approaches, logic-based languages are also available for conceptual models in order to support reasoning (e.g., the Object Constraint Language OCL, cf. OMG 2012). In a survey on the relation of conceptual models and ontologies, (Atkinson et al. 2006) thus conclude that ontologies are to be seen as subset of conceptual models, but with the distinctly different intention of having universal scope. Conceptual models are commonly notated using formal modelling languages and graphical representations for the semantic artefacts. The Unified Modelling Language (UML, cf. Booch et al. 1999, OMG 2011) can be viewed as the de-facto industry standard for the description of conceptual models, but also Entity-Relationship diagrams (ER, cf. Chen 2002) are still being frequently used for this purpose. In the context of authoring ontologies and sharing knowledge over the internet, the Web Ontology Language (OWL, cf. W3C 2004) has emerged as flagship language for the notation of ontologies.

[3] The term "ontology" originates from philosophy, where it is the study of the nature and organization of being.

The discovery of entities and their relationships in a semantic space model is often aligned with the fixed structure that is visibly observable and supports the indoor space. In a much-cited conceptual model, (Hu & Lee 2004) classify indoor space into two types of entities, namely *locations* and *exits*. A location is defined as bounded geographic area and abstracts from rooms, corridors, floors, etc. Locations can have an arbitrary number of attributes and are related to one or more exists, where an exit denotes a point on the boundary of the location from which the location can be entered or left (e.g., doors, openings, etc.). Based on the definition of qualitative spatial relationships such as *directly reachable* between exists and *directly under* between locations, (Hu & Lee 2004) show how semantic hierarchies of locations and exits can be derived which facilitate range and path queries. For example, the location hierarchy uses a tree structure to encode the knowledge that locations mapped onto inner nodes have to be passed in order to reach locations at leaf nodes. (Bittner 2001) identifies *boundaries* as the ontologically salient features in built environments, and distinguishes *bona-fide* boundaries that physically exist in real world and arise from physical discontinuities (e.g., surfaces of walls) from *fiat* boundaries that only exist virtually and result from human cognitive acts (e.g., the intangible boundaries between parking slots). The boundaries partition the built environment into *regions* along the main ontological axiom that distinct spatial objects of the same ontological kind (i.e., bona-fide or fiat) cannot overlap. Regions reflect structural entities such as rooms and corridors as well as subdivisions thereof. Also (Bittner 2001) further constrains the configuration of spatial objects through a set of qualitative spatial relationships (*full overlap*, *partial overlap*, and *no overlap*) in order to support path analyses and region hierarchies.

In addition to the structural components in indoor environments, the ontology proposed by (Bhatt et al. 2009) characterizes abstract *spatial artefacts* as virtual spaces that have a spatial extent and exhibit relationships with both strictly physical entities and other artefacts. Indoor space is classified into *object space* (i.e., the space occupied by a physical entity), and into *operational*, *functional*, and *range spaces* which represent spatial artefacts. The operational space denotes the region of space needed to perform the intrinsic function of an object (e.g., the space needed for opening and closing a door), the functional space is where an agent must be located to physically interact with the object (e.g., to open or close the door), and the range space is the region of space within the vicinity or coverage of a sensor (e.g., motion detector or RFID sensor). With spatial artefacts it is possible to express complex constraints, for example, that the operational spaces of two objects may not overlap or that every region of space must be covered by the range space of at least one sensor, with the latter being especially important in indoor positioning tasks. Based on mereotopological relationships between spaces expressed in Region Connection Calculus (RCC, cf. Randell et al. 1992), (Bhatt et al. 2009) perform reasoning about spatial constraints as well as semantic consistency and soundness checks.

Another large area of research on the semantic representation of space builds upon the human perception, that is, how people assign meaning to spatial structures while moving through and interacting with the indoor environment (Raubal et al. 1999). In the field of cognitive science, (Johnson 1987) defines *image schemas* as experiential abstractions of recurring cognitive patterns for the purpose of spatial understanding and qualitative reasoning. According to him, people apply image-schematic structures to understand (possibly unknown) spatial situations and to relate previous experience with current spatial perceptions. In this sense, people have cognitive abilities to locate themselves even with incomplete knowledge about the built environment. This qualitative reasoning process is mostly based on layout topology rather than precise metric information. For example, a basic physical schema is CONTAINER which embodies the idea of *containment* and spatially separates an inside from an outside through borders (built or by fiat). Typical examples for containers are rooms inside a building or the building itself. Containers are commonly associated with the SURFACE schema which corresponds to perceived boundaries such as the wall or floor surfaces of a room. The LINK schema captures a mutual *connection* between two entities including spatial, temporal, or functional relationships, and can be used to describe a motion pattern through PATH. In addition to image schemas, (Gibson 1986) states that the human interaction with the environment is triggered by *affordances* that reflect actions or information offered by objects and things. Affordances contribute to the experiential view of space as people apply them to the cognition of things based on past knowledge and experience.

The theory of image schemas and affordances has been employed and expanded for human wayfinding and navigation. For example, (Raubal et al. 1999) propose a wayfinding graph which encodes states of knowledge and views (i.e., pictures) at decision points, as well as action and information possibilities that trigger state transitions. (Rüetschi 2007) applies image schemas in the modelling of transfers in public transport within indoor environments. In addition to CONTAINER, he introduces REGION as an open space which is perceived as unit but is not completely bounded by built borders, and requires both containers and regions to be spatially disjoint. A GATEWAY is defined as subtype of the LINK schema affording walk-through-ability, and is, for example, experienced when walking through a door. Groupings of spaces are facilitated through the AGGREGATE schema in order to express *containment hierarchies*. However, although image schemas are accepted in literature, there is no commonly agreed formalization to date (see Walton & Worboys 2009 for an overview). With focus on built environments, (Walton & Worboys 2009) introduce a mature method for representing and visualizing topological and physical image schemas based on bigraph algebra (cf. Milner 2008). This approach facilitates the integrated modelling of spatial containment (called *place graph*) and general linking relationships (called *link graph*) between physical schemas. However, the bigraphical model lacks the possibility to represent further topological relationships such as overlap or adjacency between places which hinders path queries. (Yang & Worboys 2011) have proposed a domain ontology for indoor space which formally maps image schemas and affordances onto ontological concepts and relationships (e.g., the ontology defines concepts such as *Container*, *Surface*, *Passage*, *Connector*, *Link*, *Path*, and *Affordance*).

Semantic space models represent indoor environments on a conceptual level. They provide meaning to the observation of indoor space, and offer a classification of the diverse types of observed spatial entities. Both semantic and spatial properties, relationships, and constraints are defined at the conceptual level, and thus facilitate analyses and validations of real indoor settings which are to be seen as instantiations of the abstract concepts. However, the (semi-)automatic instantiation of conceptual models also poses a challenge as it may require sophisticated extraction and classification methods, for example, to derive the semantic information from legacy CAD-based building floor plans. Moreover, the identification of the relevant types of spatial entities and their spatial and non-spatial relationships is domain-driven and task-dependent. Conceptual views of the indoor environment therefore can differ with respect to the assessed and captured concepts, their meanings, and their level of granularity. This may lead to semantic impedance between conceptualizations (i.e., the degree of difficulty when translating a concept across domains and views), or even hinder semantic interoperability. (Giudice et al. 2010) and (Yang & Worboys 2011) therefore suggest a hierarchy of interrelated ontologies with an *upper ontology* (containing only high-level space concepts), a set of *domain ontologies* or *"microworlds"* (i.e., limited ontologies focusing on the structure and artefacts of indoor space), and *task ontologies* (e.g., navigation, building planning). The authors expect such a hierarchy to create a common understanding and terminology of indoor space, and to serve as integration framework for existing approaches to indoor navigation.

2.1.4 Hybrid Space Models

While many tasks in indoor navigation need precise metric information, the importance of an accurate geometric representation of the built environment and its interior structures is to be seen on par with its qualitative description. For example, shortest-length path computations depend on metric distances but the additional consideration of user preferences (e.g., to use stairs rather than elevators) or user constraints (e.g., to avoid stairs for wheelchair users or mobile robots) in path queries requires semantic knowledge about the environment (e.g., to distinguish stairs from elevators). On the other hand, pure semantics will not suffice to decide whether a wheelchair user can drive through narrow doors or passages along the path, what however can be measured in a geometric model. Since human perception of and interaction with indoor spaces is mainly based on qualitative understanding and reasoning (cf. chapter 2.1.3), indoor space models for human navigation consequently combine and link geometric, topological, symbolic, and semantic information (see also Leonhardt 1998, Becker & Dürr 2005, Ye et al. 2007, Ruppel et al. 2008, Giudice et al. 2010). Approaches in mobile robot navigation long neglected the need for semantics and conceptual modelling but rather relied upon purely geometric and topological maps of the environment. Of course, precise geometric maps still play a pervasive role but the inclusion of semantics has come into focus, for example, in fields like human-robot interaction, autonomous spatial exploration and learning, but also in path planning and localization tasks (e.g., Anguelov et al. 2004, Stachniss et al. 2007).

Hybrid space models accommodate aspects of geometric, symbolic, and semantic space models. A challenge hereby is to define a common denominator for integrating the different types of indoor space models. Most approaches discussed in the previous sections agree in structuring the built interior environment into *spatial regions*, and express *connectedness* and *containment* relationships between them. If understood as (partly or virtually) bounded portion of indoor space that inherently enables or obstructs movement, the notion of a spatial region provides an overarching spatial concept that abstracts from symbolic places in symbolic space models, cells in geometric approaches, and similar concepts defined in sematic models (e.g. "locations" in Hu & Lee 2004, or "regions" in Bittner 2001 and Bhatt et al. 2009). Moreover, it conforms to the human perception of indoor space as embodied through the physical CONTAINER schema. Symbolic and geometric space descriptions can be aligned with a region by representing them as thematic and spatial properties. For the purpose of navigation, connectedness and containment are the two most elementary types of qualitative spatial relationships that need to be expressed between physical regions (see also Becker & Dürr 2005, Kolodziej & Hjelm 2006, Ye et al. 2007). Connectedness information is required for path searches and implies mutually *non-overlapping* regions. The containment relationship adds support for hierarchically nested regions and hence accounts for the structural complexity of the interior built environment. In contrast to connectedness, it builds upon *overlapping* regions. Human spatial cognition and wayfinding is known to be hierarchic (e.g., Richter et al. 2009, Stoffel et al. 2009), whereas containment hierarchies are mostly applied for scalability reasons in robot navigation. It follows that spatial regions as well as their connectedness and containment constitute a minimum conceptual overlap between geometric, symbolic, and semantic space models.

Formal hybrid models not only integrate the different aspects of indoor space on a conceptual level, but also provide a sound formalization of this integration. However, this is often hindered by the fact that indoor space models lack a mathematical mapping between the geometric, topological, symbolic, and semantic dimensions of indoor space. A well elaborated example of a formal hybrid model is presented by (Bhatt et al. 2009) who propose a framework of modular ontologies for what they name as *conceptual*, *quality*, and *quantity space*. The conceptual space reflects the semantics of the indoor environment, namely the types of entities with their symbolic and non-spatial properties and functional characteristics. The quality space is concerned with the qualitative spatial information such as topological relationships between physical regions, which are defined independently from the conceptual space. Finally, the quantity space is about the precise 2-dimensional respectively 3-dimensional geometry of the spatial entities. Corresponding instances in each space module are then formally linked in an integration view based on ε-connection theory for ontologies (cf. Stuckenschmidt et al. 2009). The proposed ontology modules closely follow the classification scheme for indoor space models as introduced in this chapter. A benefit of keeping the different space representations separate is that spatio-semantic reasoning and validation can be performed for individual modules as well as their integrated view. However, functional dependencies between the decoupled modules are not considered. For example, changing the spatial configuration of entities directly affects their topological relationships. The framework of (Bhatt et al. 2009) lacks formal methods for mapping such spatial facts between the separate modules.

2.2 Survey of Existing Hybrid Space Models for Indoor Navigation

Most current modelling proposals for indoor environments in the field of indoor navigation fall into the category of hybrid space models. This is a logical consequence of combining the strengths of geometric, symbolic, and semantic space models in order to overcome their respective weaknesses. Nevertheless, most proposals substantially differ in the emphasis they place on the separate aspects of space, in the semantic entities they define, and in the methods and the level of granularity they apply to their space representations. These differences are illustrated along a survey of existing hybrid modelling approaches in the following. However, due to the large number of proposals in both academia and industry, only selected approaches that are regularly referenced in related work are reviewed in this thesis. The approaches are arranged into four categories, namely *grid-based*, *cell-based*, *dual-graph-based*, and *conceptual-based* approaches, and discussed in separate subsections. For each category, conclusions about their shortcomings and strengths are drawn at the end of each subsection.

2.2.1 Grid-based approaches

A popular grid-based model is proposed by (Bandi & Thalmann 1998). In this model, the indoor space is discretized into a continuous 3-dimensional grid of uniform cuboidal cells which are tagged as *obstacle* or *non-obstacle*. The grid is treated as graph structure with each cell (except those at the grid borders) having links to its 26 neighbours. Based on this representation, obstacle-free shortest paths are computed by evaluating the possibility to move between neighbouring cells, which hence results in a discrete sequence of cells. In order to accelerate path searches, only *surface cells* on the upper surface of objects on which humans walk are considered. Obstacles are further classified into *insurmountable* and *surmountable* ones. Insurmountable obstacles obstruct movement and comprise objects that humans cannot step on or over (e.g., walls, pillars), gaps that cannot be crossed, or passages being too low or too narrow for humans to pass through. Surmountable obstacles, on the other hand, are accessible or passable for humans (e.g., stairs, steps, small furniture). Whether an obstacle falls into one or the other category is determined by its reachability which depends on the horizontal respectively vertical foot span of the navigating person. Figure 1 illustrates the different obstacle types as well as a 3-dimensional setting of surface cells and insurmountable cells. The model of (Bandi & Thalmann 1998) is also applicable in two dimensions.

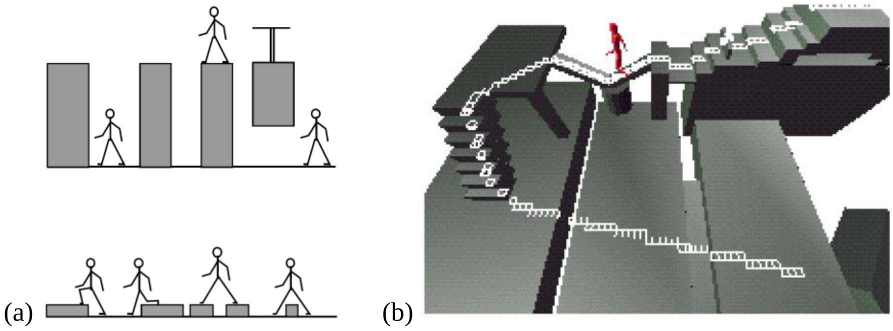


Figure 1: Insurmountable and surmountable obstacles (a); 3*-dimensional path based on surface and obstacle cells (b) (Bandi & Thalmann 1998).*

The 3-dimensional grid-based model presented in (Yuan & Schneider 2010b) and (Yuan & Schneider 2011) approximates the interior built structure as well as all contained obstacles using a LEGO-based representation inspired by the correspondent toy bricks. Similar to (Bandi & Thalmann 1998), the entire indoor space is mapped onto a continuous grid of LEGO cubes all of which have same-sized base areas. However, the cubes are not uniform but have their own heights depending on the space or object they represent as shown in the following figure.

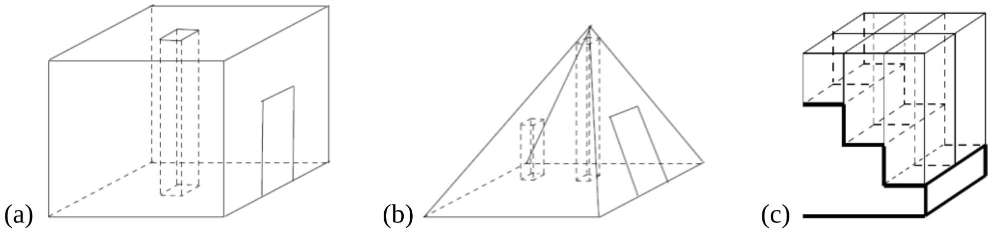


Figure 2: LEGO cube in a cuboidal cell (a), cubes with different heights in a pyramid-shaped cell (b), and cubes representing stairs (c) (Yuan & Schneider 2011).

(Yuan & Schneider 2010b) semantically differentiate three types of LEGO cubes: *plane_cubes*, *stair_cubes*, and *obstacle_cubes*. The free and navigable space between a floor and a ceiling is represented through *plane_cubes* whose height is the distance between the floor and the ceiling (cf. figure 2a and b). Stairs are modelled by a set of ascending or descending *stair_cubes* with differing heights (cf. figure 2c), and *obstacle_cubes* approximate objects whose occupied space is inaccessible for humans, such as walls, furniture, or objects suspended from the ceiling. In a subsequent step, cubes of the same type are merged into strictly rectangular blocks which are used to assess the maximum free space between obstacles. The resulting blocks can be spatially disjoint, adjacent, or overlapping. In the latter two cases, the maximum accessible width and height between two blocks is captured by a *connector* surface which is simply the common boundary in case the blocks are adjacent or the diagonal cross-section of their intersection volume otherwise. The connectivity information is translated into a graph in which nodes denote blocks and edges represent connectors. This connectivity graph is already feasible to derive discrete block-by-block paths through the interior environment. In order to more efficiently support shortest path queries, (Yuan & Schneider 2010b) propose a second graph structure, called *LEGO graph*, which results from mapping the connectors onto nodes. Two nodes are linked if the connectors are reachable from the same block. The distance between two connectors is encoded as edge attribute and is computed as Euclidean length of the straight line connecting the midpoints of the connectors. Figure 3 depicts the 2-dimensional LEGO representation of a simple building floor as well as the corresponding connectivity graph and LEGO graph. In (Yuan & Schneider 2011), the authors demonstrate how the LEGO model can be used to evaluate the accessibility of paths for arbitrarily shaped users (e.g., pedestrians, wheelchair users, mobile robots) based on the volumetric size of the user as well as the information about connectors and maximum overlaps between blocks.

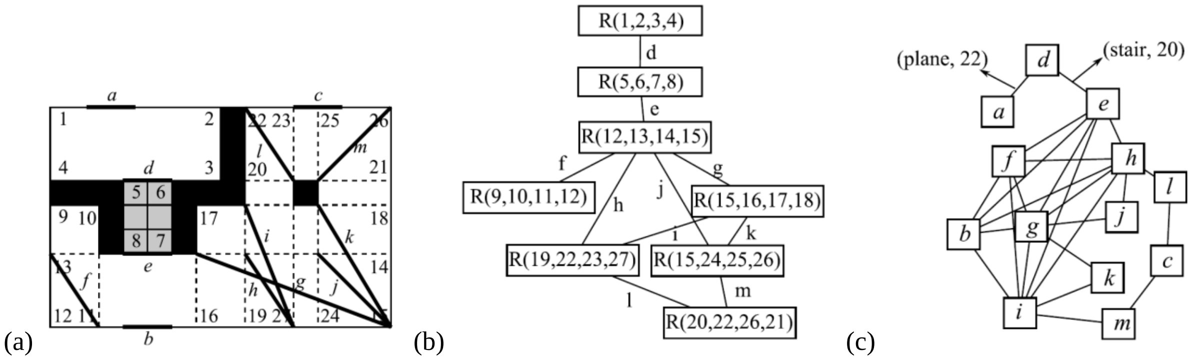


Figure 3: Floor plan with obstacles, stairs, and connector surfaces (a), the graph reflecting the connectivity of the blocks (b), and the corresponding LEGO graph (c) (Yuan & Schneider 2010b).

An alternative grid-based model supporting different levels of granularity is proposed by (Afyouni et al. 2010). At the finest level, the indoor space is modelled as 2-dimensional continuous grid of uniform cells. The grid is understood as cell adjacency graph in which each cell has rectilinear connections to its eight neighbours (cf. Bandi & Thalmann 1998). From this *micro level*, a coarser *meso level* representation is obtained by merging grid cells belonging to the same spatial unit such as rooms or corridors to larger non-uniform and non-overlapping cells. The meso level is also mapped onto a graph structure that captures both cell adjacency and connectivity information. Finally, at the coarsest *macro level*, all cells are related to a building floor as shown in the following figure.

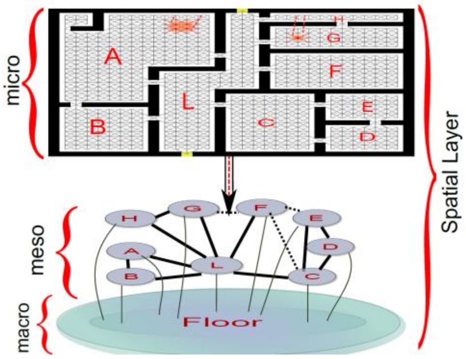


Figure 4: Multi-level representation of indoor space according to (Afyouni et al. 2010).

The approach of (Afyouni et al. 2010) exemplifies the derivation of a spatial containment hierarchy on top of a grid-based model, and thus is capable of supporting path queries at different scales. Besides the multi-granular organization of indoor space, (Afyouni et al. 2010) propose a complementary *feature layer* to integrate fixed obstacles (e.g., walls, doors, furniture, sensors). The feature layer is however not restricted to obstacles but also accounts for mobile objects (e.g., users, mobile devices) as well as continuous phenomena (e.g., noise or fire diffusion) inside the indoor environment. The conceptual representation of features is inspired by the work of (Bhatt et al. 2009) (cf. chapter 2.1.3) and differs between a feature's *object*, *operational*, *functional*, and *range space* each of which is spatially expressed as grid cells. An additional *action layer* is introduced to capture the action affordances offered by the features (cf. Gibson 1986, chapter 2.1.3). The proposed layer structure is meant to support context-dependent navigation. For example, mobile objects having different modes of locomotion are reflected with different operational and functional spaces. Moreover, action possibilities may change depending on the navigation scenario as well as the user context and preferences. However, (Afyouni et al. 2010) remain at a descriptive level and neither formalize the additional layers nor their interactions in order to evaluate the contextual information in a navigation task.

The presented examples demonstrate the integration of semantics into a geometric grid-based representation of the indoor environment. The resulting hybrid models already address several challenges to indoor navigation as identified in chapter 1.2. Shortest path searches as well as nearest neighbour and range queries are facilitated through the implicit grid-based adjacency graph and its metric embeddedness. Based on the continuous mapping of indoor space onto cells, the spatial extents of navigable and non-navigable regions are explicitly represented which supports the computation of obstacle-free paths. The non-overlapping grid cells also allow for the evaluation of geometric navigation constraints such as the accessibility of spaces with respect to the width, length, and height of the navigating users and their mode of locomotion. The cell-based discretization of indoor space can be formalized mathematically and generated automatically in both two and three dimensions. However, a major drawback of grid-based models is their approximation of the indoor environment. First, the rigidity of the grid results in indoor spaces and objects not being captured accurately (cf. chapter 2.1.2). Second, the resolution of semantic and symbolic aspects is coupled with the resolution of the grid. Semantic properties and relationships therefore cannot be assigned to a spatial entity directly but have to be expressed for the cells which represent the entity. This hinders unique symbolic labels (e.g., for naming places in path queries or route commands) as well as semantic queries (e.g., finding all fireproof doors or walls made of a certain material in evacuation scenarios). Moreover, if an entity is not mapped by a grid cell, it cannot be represented semantically at all. In order to overcome these issues, (Afyouni et al. 2010) introduce their feature layer and propose the merging of cells belonging to the same architectural unit, however facing the above mentioned drawbacks. The grid resolution also affects the computational efficiency of path searches. Fine-resolution grids exponentially increase the number of cells and adjacency relationships, and hence require strategies for reducing the search space. Whereas (Bandi & Thalmann 1998) only consider surface cells, (Yuan & Schneider 2010b) on the one hand reduce the number of cells by letting them span the entire height of indoor spaces, and on the other hand compactify the grid-based graph by creating larger cell blocks according to semantic and geometric criteria. Although both approaches apply a 3-dimensional grid, the path searches are hence restricted to two and a half dimensions and thus, for example, cannot account for flying objects. The 2-dimensional model of (Afyouni et al. 2010) is based on a multi-level granularity but lacks formal methods and algorithms to combine the different levels in path computations.

Finally, most grid-based approaches model the built environment only and neglect the decomposition of indoor space according to complementary criteria such as sensor coverage or logical aspects. The need for an additional sensor space representation in order to support indoor positioning tasks is at least identified by (Afyouni et al. 2010) who map the range spaces of fixed sensors onto the grid. However, consequential challenges like the integration of multiple and different position technologies or the context-dependent ad-hoc selection of positioning methods are not addressed in their proposal.

2.2.2 Cell-based approaches

An early approach to the subdivision of indoor space along the architectural structure rather than using geometric decomposition methods has been presented by (Gilliéron & Merminod 2003). In this model, architectural entities like corridors, rooms, stairs, elevators, or points of interest as well as their shapes and topological relationships are extracted from legacy 2-dimensional CAD floor plans. A connectivity graph, called *node/link model*, is established in which entities such as rooms or corridors are mapped onto nodes, and whose edges denote links which enable the movement between two nodes (e.g., doors, stairs, or elevators). As shown in figure 5, large corridors are decomposed into more than one node to account for several links supported by the corridor.

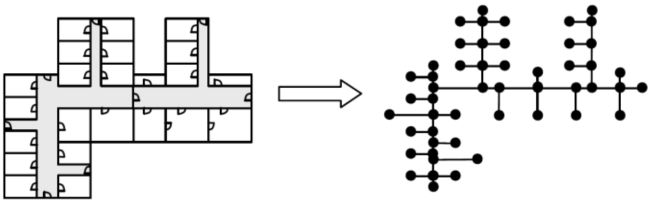


Figure 5: Transformation of a CAD floor plan (left) into a node/link model (right) (Gilliéron & Merminod 2003).

The graph elements are semantically enriched with additional properties required for navigation purposes such as information about access rights or reachability for handicapped persons. Like with irregular geometric models (e.g., Lamarche & Donikian 2004, cf. chapter 2.1.2), the spatial entities are geometrically described by non-overlapping cells of irregular shape and size which decompose the free indoor space but neglect the contained obstacles. However, in contrast to purely geometric methods, the cell geometries reflect the real world shape of the architectural entities which results in a more compact graph structure and more naturally follows the building layout. Based on a cost function which, for example, takes travel time, distance or access rights as parameters, shortest or best path searches are realized on top of the node/link model.

A similar approach is proposed by (Lorenz et al. 2006) and (Lorenz & Ohlbach 2006). Following the ontology of (Bittner 2001) (cf. chapter 2.1.3), the free indoor space is partitioned along bona-fide and fiat boundaries into disjoint 2-dimensional cells which represent navigable regions and mainly coincide with architectural entities such as rooms and corridors. A complementary graph structure is derived by mapping cells onto nodes, and by representing passages between cells (e.g., resulting from doors or elevators) as edges. The nodes are assigned the cell centres as spatial location in an underlying reference system in order to geometrically embed the graph and to facilitate the computation of distances. Similar to the approach of (Gilliéron & Merminod 2003), large corridors or rooms may be split into several cells. The following figure 6 illustrates the cell-based model of (Lorenz et al. 2006). However, a formal method to fully automate the cell decomposition is not presented.

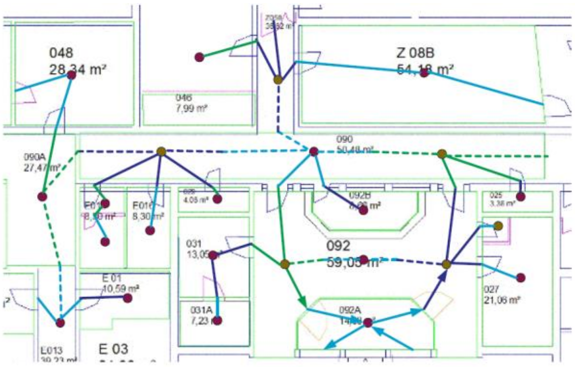


Figure 6: Example floor plan and corresponding graph-based conceptualization. Dashed lines indicate edges between subdivisions of a room or corridor (Lorenz et al. 2006).

In addition to the cell connectivity graph, (Lorenz et al. 2006) introduce a multi-level hierarchical graph structure to express spatial containment relationships. For example, rooms are aggregated to wings, wings to floors, and floors to the entire building at different levels of the hierarchy. Each hierarchy level is associated with its own graph, where nodes (edges) at a lower level are clustered into nodes (edges) of the upper level (cf. figure 7). The nodes are assigned semantic properties and symbolic labels in order to support semantic path queries and human-understandable path descriptions. The graph model hence supports path planning at different scales and the generation of route commands such as *"go to the fifth floor, and then navigate the left wing to the target room 5126"*. A salient aspect of the model is that doors of rooms and corridors are organized in sorted sets according to their angular distribution around the cell centre. This information can be used to generate commands such as *"take the second door to the right"* along a path. Moreover, the authors propose fuzzy sets to mathematically represent several degrees of qualitative relationships in human language, for example, to express that a door is *somewhat opposite*, *fairly opposite*, or *directly opposite* another door.

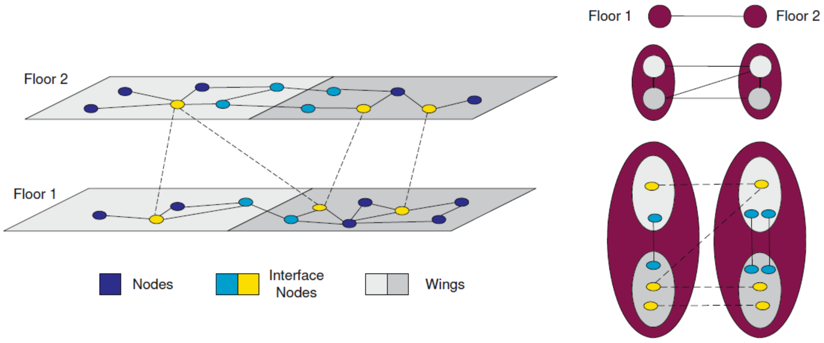


Figure 7: Hierarchical graph structure according to (Lorenz et al. 2006) for three levels (floors, wings, rooms).

In a formally well-developed cell-based model, (Stoffel et al. 2007) decompose the indoor space into a mesh of *spatial regions*. A spatial region is mathematically defined as 2-dimensional simple polygon that is bounded by a closed polygonal chain of straight non-intersecting line segments, and which is embedded in a given coordinate reference system. Spatial regions may have arbitrary shapes and map architectural features such as rooms and corridors. They are forced to be non-overlapping and to jointly make up the entire indoor environment. However, similar to the above discussed models, the spatial regions only model the free space but do not explicitly represent obstacles or non-navigable areas inside the interior environment. Inspired by the concepts of GATEWAYS (Rüetschi 2007) and *exits* (Hu & Lee 2004, for both see chapter 2.1.3), (Stoffel et al. 2007) define *boundary nodes* as points on the shared boundary of two spatial regions which enable access or egress. A boundary node thus denotes the connectivity between two spatial regions and conceptually abstracts from doors and other openings. Both spatial regions and boundary nodes are augmented with symbolic identifiers as well as semantic or metric properties such as the door width. A flat connectivity graph facilitating shortest path searches is constituted with nodes indicating spatial regions and edges reflecting boundary points. Comparable to (Lorenz et al. 2006), the model supports containment hierarchies between spatial regions. The containment relationship is mathematically defined so that it renders a tree structure, called hierarchical *region graph*, and hence induces a partial ordering on the set of spatial regions. In a subsequent paper, (Stoffel et al. 2009) present an algorithm to automatically construct the hierarchy from 2-dimensional floor plans. The resulting indoor space representation is depicted in the following figure 8.

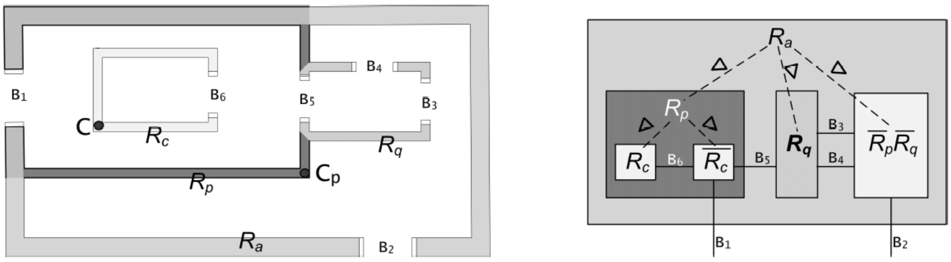


Figure 8: A 2-dimensional setting of nested spatial regions with boundary nodes (left) and the corresponding region graph expressing spatial containment and connectivity relations (right) (Stoffel et al. 2007).

According to (Stoffel et al. 2007), the derived graphs are still too coarse for human navigation, as concave spatial regions (e.g., an L-shaped corridor) cannot be perceived as a whole and thus need several path descriptions in textual route guidance. In order to automatically generate and provide such descriptions, (Stoffel et al. 2007) decompose concave spatial regions into non-overlapping convex sub-regions according to visibility criteria as shown in figure 9. Based on this partitioning, a complementary *visibility graph* linking boundary nodes is defined to guide persons through the interior of concave spatial regions. Intervisible boundary nodes (e.g., those in the same sub-region) are simply connected by straight lines, whereas paths between non-intervisible boundary nodes in different sub-regions require intermediate nodes (e.g., midpoints of the cutting lines dividing the concave spatial region). In contrast to the approaches presented so far, (Stoffel et al. 2007) further formalize the accessibility of spatial regions as well as user preferences. Accessibility constraints such as access rights or time restrictions are expressed on boundary nodes or region graphs as *Boolean* (hard) *constraints* yielding true or false, based on which graph elements are filtered before applying path searches. Predicate logic is used to formulate user preferences (e.g., using staircases rather than elevators) as *soft constraints* which, for example, affect the cost of traversing edges but do not alter the graph itself.

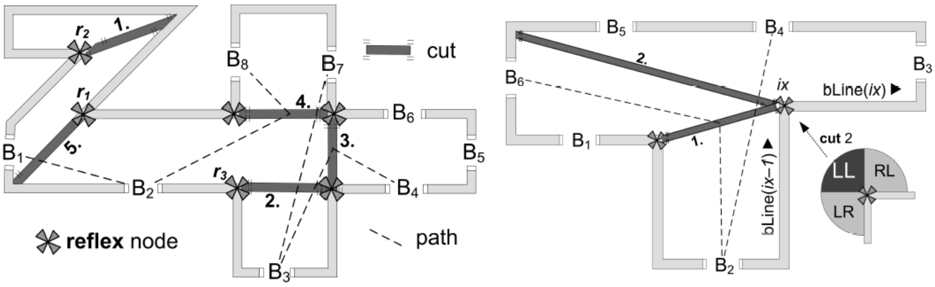


Figure 9: Decomposition of concave into convex spatial regions according to visibility criteria and derived visibility graph connecting the boundary nodes (Stoffel et al. 2007).

The *iNav* model developed by (Yuan & Schneider 2010a) aims at eliminating the circuitous, non-optimal geometric routes which result from assigning the cell centres as spatial location of corresponding nodes in cell connectivity graphs like the one proposed by (Lorenz et al. 2006). For this purpose, the free indoor space is decomposed into 2-dimensional non-overlapping spatial regions which are conceptually classified into *simple cells*, *complex cells*, *open cells*, and *connectors*. Both simple and complex cells are completely enclosed by built boundaries such as walls, whereas open cells are partly surrounded by intangible boundaries (cf. Bittner 2001). Connectors are objects which connect floors in a building like elevators or stairs. The access to cells is enabled through *access points* which represent any type of opening such as doors or (parts of) intangible boundaries. In contrast to simple cells which only possess a single access point, complex and open cells as well as connectors may exhibit multiple access points. Direct *path segments* denote the shortest path between two access points associated with the same cell, and are constructed by connecting the locations of access points through straight line segments. However, for concave-shaped cells, the straight line may intersect the cell boundary. In such cases, (Yuan & Schneider 2010a) propose the introduction of *intermediate points* on the concave cell boundary, and recursively partition the shortest path into segments between intermediate points until no segment intersects the cell boundary and the resulting shortest path lies completely inside the cell or on its boundaries. Figure 10 exemplifies this situation for a complex cell.

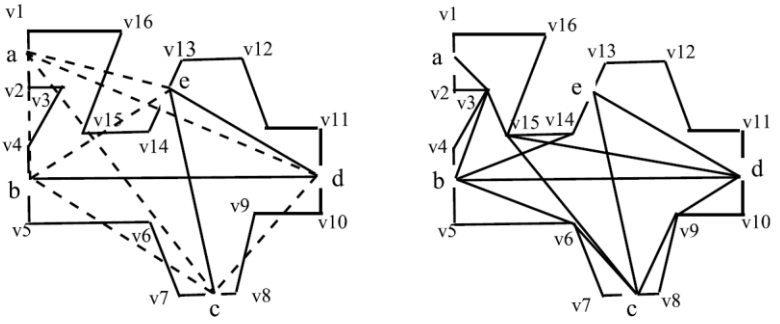


Figure 10: Two access points (like b and c) cannot be linked through straight line segments (left) which requires their partitioning along intermediate points on the concave cell boundary (right) (Yuan & Schneider 2010a).

For shortest path computations, a *direct path graph* (DPG) is introduced whose node set comprises all access and intermediate points, and whose edges are determined by the set of path segments (cf. figure 11). The DPG captures the connectivity between access points and not between cells, and hence provides direct paths from one door to the next. Accessibility information is modelled as semantic attributes on nodes and edges, and is evaluated during path searches. The DPG actually encodes the *mutual visibility* between its nodes and only differs by the location of intermediate points from the visibility graph proposed by (Stoffel et al. 2007). Since the cells are not explicitly represented in the DPG, symbolic cell labels for naming the start and goal of path queries (e.g., "from the *entrance hall* to *room 5126*") are not available, but have to be best matched with access points or intermediate points, which requires an extra computational step. The authors claim that the DPG offers length-optimal paths compared to paths involving the cell centre but fail to present a formal proof. Moreover, the process of decomposing the indoor space into cells is not further specified. In a follow-up work, (Yuan & Schneider 2010c) develop algorithms for supporting range queries based on the DPG.

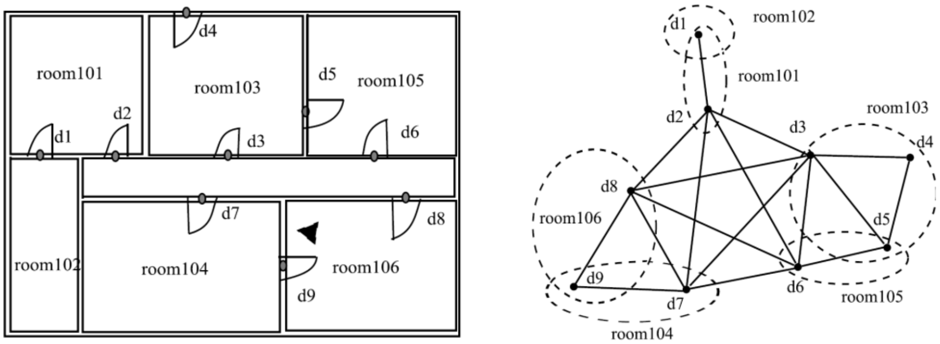


Figure 11: Example floor plan (left) and corresponding direct path graph (right) (Yuan & Schneider 2010a).

Similar ideas are followed in (Liu & Zlatanova 2011a), (Liu & Zlatanova 2011b), and (Liu & Zlatanova 2012), who propose a two-level routing strategy through the indoor environment. The first level addresses the sequence of rooms and corridors that has to be traversed from a start to a goal location. This step utilizes a place graph encoding cell connectivity information. For this purpose, an *Indoor Navigation Space Model* (INSM) is proposed which defines the topographic spaces inside a building from a conceptual point of view and hereby builds upon the previous work on the MLSEM. The spaces are primarily classified into *NavigableSpaceCell*, *Obstacle*, and

Opening which denote the free respectively non-navigable areas inside a building as well as openings providing passage between the free areas such as doors. Navigable spaces are further subtyped into *horizontal* and *vertical units* with the latter representing ramps, stairs, or elevators. Moreover, *vertical* and *horizontal connector spaces* as well as *end spaces* are identified based on the number and types of neighbouring spaces. A salient aspect of the INSM is the provision of semantic rules for identifying and validating the corresponding types of spaces and for deriving them from building modelling standards such as CityGML or IFC (cf. chapter 2.3). Based on the semantic entities, a *Logical Graph* is then built with nodes representing *NavigableSpaceCells* and edges denoting their connectivity based on *Openings*. This graph allows for determining purely symbolic paths through the environment. An example instance of the INSM is shown in the following figure.

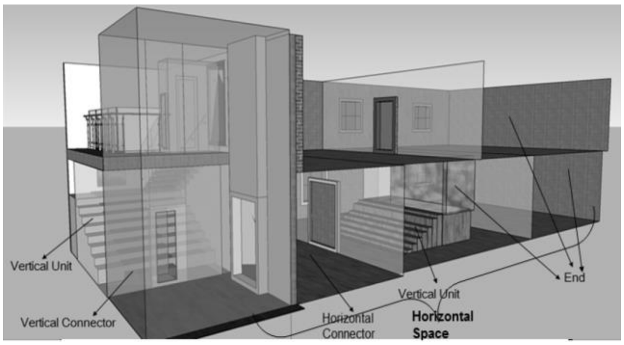


Figure 12: Example classification of indoor spaces according to the INSM (Liu & Zlatanova 2012).

In a second step, routes inside navigable space cells are determined based on an optimised *door-to-door* path finding algorithm which reveals the closest openings between subsequent cells. Similar to (Stoffel et al. 2007) and (Yuan & Schneider 2010a), a *visibility graph* is constructed by mapping openings onto nodes and connecting them along intervisibility criteria, which involves adding intermediate nodes on the boundary of concave cells. In contrast to the DPG, the number of additional nodes is minimized. (Liu & Zlatanova 2011b) also apply a visibility partitioning to stairs in order to connect different building floors in their visibility graph. The resulting navigation structure is named *Indoor Visibility Structure* (IVS). Since the door-to-door path generally traverses along the closest openings between cells, it may result in different paths if the start and goal location are swapped (cf. figure 13). The IVS is thus not feasible for shortest path computations between several cells. However, the authors assume that choosing the closest exit more naturally reflects human behaviour in escape situations.

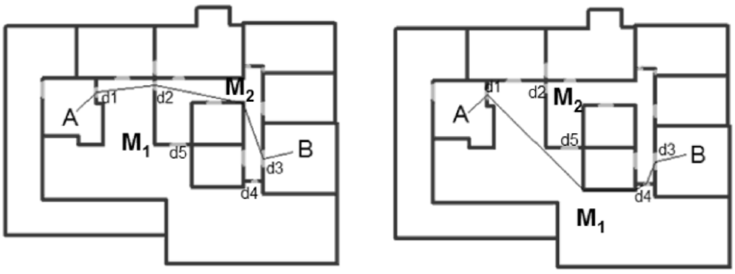


Figure 13: Door-to-door path according to (Liu & Zlatanova 2011a) from room A to room B (left) and vice versa (right).

Although the IVS can be acquired a priori for an entire building as realized in (Stoffel et al. 2007) and (Yuan & Schneider 2010a), the authors put focus on the real-time derivation and dynamic modification of the graphs in order to support emergency and evacuation scenarios. For example, if rooms are impassable due to fire, the room sequence needs to be re-computed on the first coarse-grained level. The door-to-door path inside cells is computed upon entering the cells. Moreover, fixed and moving obstacles inside cells (e.g., smoke diffusion or even crowds of people) may require a dynamic update of the visibility relationships between exits which possibly produces new paths. While (Liu & Zlatanova 2011b) discuss requirements for dynamic routing strategies, an implementation on top of the IVS is not presented.

(Goetz & Zipf 2011) propose a 2-dimensional cell-based approach to establish a context-dependent and length-optimal indoor routing graph. The cell decomposition of the indoor environment closely follows the approach of (Yuan & Schneider 2010a), and only differs in the graph-based representation of corridors. Whereas (Yuan &

Schneider 2010a) also apply the DPG to corridors, (Goetz & Zipf 2011) approximate a corridor through its centreline and orthogonally link access points to nodes along that centreline as shown in figure 14a (see already Gilliéron & Merminod 2003). For typical layouts of large corridors the authors mathematically prove that this representation requires less graph elements, and thus helps to increase the computational performance of path searches. In contrast to (Yuan & Schneider 2010a), also obstacles inside rooms are reflected in the graph by adding nodes to bypass the obstacles which however requires manual editing (cf. figure 14b). The authors present the hierarchical subdivision of rooms into smaller cells (e.g., to reflect check-in counters or smoking areas inside entrance halls at airports) as a new conceptual element while obviously neglecting the previous research on containment relationships between indoor spatial areas. A *weighted indoor routing graph* (WIRG) is proposed which maps access points as well as distinct areas inside rooms onto nodes, and whose edges express *mutual visibility* through straight line segments which enable shortest path computations (cf. figure 14c). However, unlike (Stoffel et al. 2007), (Yuan & Schneider 2010a), and (Liu & Zlatanova 2011a), the impact of concave cell boundaries on the intervisibility is not discussed.

An interesting aspect of the WIRG is the organization of symbolic labels, traveling conditions or requirements, as well as additional semantic information into sets, and the formal definition of functional mappings between these sets and the graph elements. For example, the mapping $g: E \times TC \times R \rightarrow \mathbb{R}_+$ results in a non-negative weight given an edge $e \in E$, a travel condition $tc \in TC = \{healthy, elderly, wheelchair, \dots\}$, and a routing requirement $r \in R = \{shortest\_route, quickest\_route, most\_interesting\_route, \dots\}$. An edge e representing a stair may hence receive $g(e, healthy, shortest\_route) = 5.2$ as well as $g(e, elderly, shortest\_route) = 19.9$ as context-dependent weights which can be considered by cost functions in path search algorithms. Although this approach provides a mathematical formalization of user-contextual aspects, the derivation of most interesting routes or alternative paths for wheelchair users and elderly people from the proposed indoor space model is left unaddressed. Moreover, the approach of (Goetz & Zipf 2011) faces the same issues as identified in the discussion of the underlying *iNav* model.

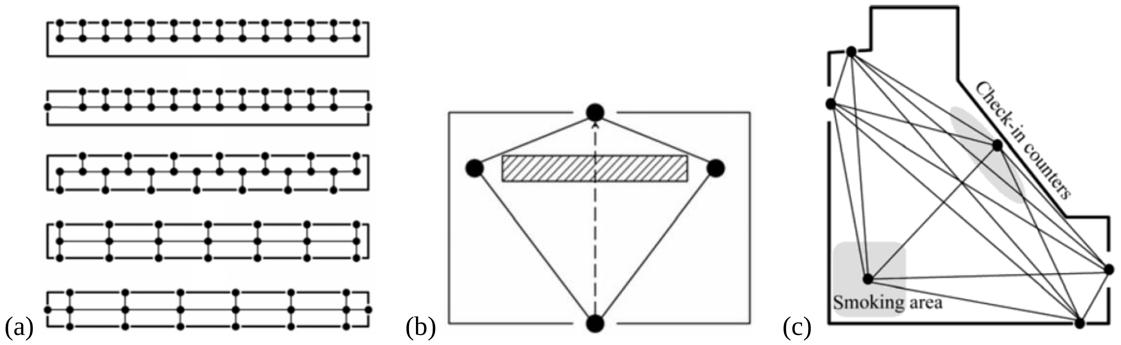


Figure 14: Different layouts of corridors and corresponding graphs (a), additional nodes for bypassing obstacles (b), and visibility graph linking subdivisions of an airport entrance hall (c) (Goetz & Zipf 2011).

(Meijers et al. 2005) present a cell-based indoor space model which provides a rich semantic classification of the building interior. The model builds upon a 3-dimensional geometric representation of indoor space through a mesh of polygons. The polygons form the boundaries of non-overlapping volumetric cells which occupy the free indoor space. The semantic classification takes polygons as smallest unit, and agrees with (Bittner 2001) who identifies boundaries as salient semantic entities, and also matches the human observation expressed through the SURFACE and CONTAINER image schemas. Polygons are classified according to their *persistency*, *existence*, *accessibility*, and *direction of movement*. Examples for persistent surfaces are built walls whereas walls that can be folded or temporarily removed are non-persistent. If the wall contains a door or window it is said to be access-granting with *full*, *semi* (e.g., door requires a key), and *limited* (e.g., only passable in rare situations such as emergency cases) representing different degrees of accessibility. If there is only an opening then this surface is classified as non-existent or *virtual*. Finally, access-granting polygons are distinguished into uni- and bi-directional passages. The different types of semantic entities are illustrated in figure 15. Cells are defined as smallest amount of bounded space in a building that has a specific function, and need to be completely enclosed by polygons (if required through virtual polygons). They are further classified into *end sections*, *connector sections*, and *non-accessible sections*. An end section only has a single entrance point (i.e., access-granting polygon) and represents simple rooms. Connector sections provide more than one entrance point such as corridors, elevators, or stairs. The concepts show analogies with simple, complex, and connector cells as defined by the *iNav* model (Yuan & Schneider 2010a). However, non-accessible sections additionally facilitate the representation of columns or fixed obstacles inside rooms, and sections can be aggregated to form *complex sections* (e.g., storeys). From the semantic entities, a *connectivity graph* is derived by mapping end sections onto nodes, granting polygons onto edges, and connector sections onto single nodes or, alternatively, onto subgraphs in case they exhibit complex shapes (cf. Gilliéron & Merminod 2003, Goetz & Zipf 2011). It follows that obstacles or non-granting polygons (i.e., walls) are not explicitly represented in the final graph structure, and thus are not considered in path searches. Since the graph is only derived from the conceptual model, it is a purely topological structure and lacks quantitative information. Therefore, a metric embedding is proposed by (Meijers et al. 2005) which assigns nodes the centre points of cells.

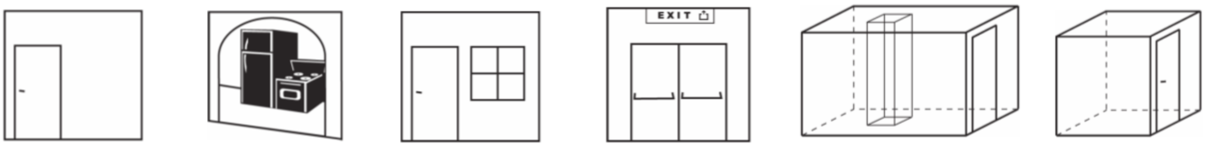


Figure 15: From left to right: persistent polygon (door), virtual polygon (opening), granting polygon with full access (door), granting polygon with limited access (emergency door), non-accessible section (column), and end section (room) (Meijers et al. 2005).

Another semantically enriched representation of the interior environment is proposed by (Gröger & Plümer 2010). The primary focus of their approach is put on the generation of geometric-topologically consistent 3-dimensional indoor models. The authors introduce an attributed grammar as mathematical formalism that governs the model generation. The grammar extends previous works on *shape* and *split grammars* that have been successfully applied in the field of procedural modelling of buildings (e.g., Wonka et al. 2003, Müller et al. 2006) by additionally representing topological relationships and ensuring their consistency throughout the generation process. The non-terminal symbols of the grammar are 3-dimensional cells called *Boxes* which are geometrically described through prisms and *Rectangles* which represent the 2-dimensional rectangular boundary surfaces of the cells. The boxes are semantically classified into *Building*, *BuildingPart*, *Storey*, *StoreyPart*, *Room*, *Hall*, and *Staircase*, whereas rectangles are used to model *Walls*, *Ceilings*, *Floors* as well as openings such as *Doors* and *Windows*. In contrast to (Meijers et al. 2005), obstacle spaces cannot be expressed since boxes are restricted to free and navigable spaces. Based on a set of parameterized rules, the non-terminal symbols of the grammar are carried to terminal symbols. For example, the most essential rule splits a given box into two boxes. Thus, starting from a box representing the outer hull of the entire building, this split rule provides a pattern for the step-wise decomposition and refinement of the building into floors, wings, stairs, corridors, and rooms. Further rules are presented to semantically tag boundary surfaces or to place doors and windows into surfaces. The authors introduce a *constraint store* which facilitates the reasoning about topological relationships between symbols in order to avoid topologically inconsistent rule applications (e.g., to avoid penetrations of boxes or that a box is split at a door or window). This constraint store is constantly updated and differs from previous approaches using split grammars in that it is not locally restricted to a single rule but rather affects all rules globally. The following figure 16 shows an example for the step-wise generation of the interior structures of a building. On the right the corresponding derivation tree resulting from the rule applications is depicted.

Once the building model has been constructed, a *route graph* for indoor navigation is built from the derivation tree. For this purpose, terminal boxes such as rooms, halls, and staircases are mapped onto nodes which are linked by an edge if they share a common rectangle that represents a wall and contains a door. Due to the geometric-topological consistency of the model, graph edges can only occur between topologically adjacent boxes. Moreover, the semantics are used to finally derive a *connectivity graph*. Similar to the approach of (Meijers et al. 2005), the resulting route graph however lacks elements representing the non-navigable surfaces. A geometric embedding of the route graph in order to quantify the connectivity information is not addressed by (Gröger & Plümer 2010). And although the derivation tree partially encodes spatial containment in case a box is split into two boxes, this knowledge is not used in an explicit hierarchical graph structure of the interior environment.

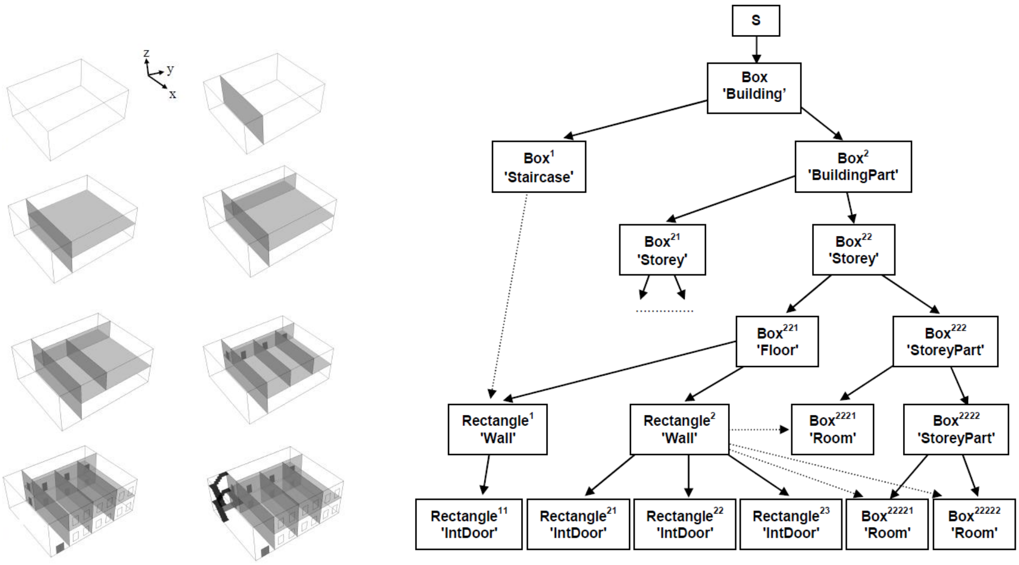


Figure 16: Step-wise generation of a building with interior structures (left) and corresponding derivation tree (right) (Gröger & Plümer 2010)

In summary, all of the presented cell-based hybrid approaches decompose the indoor environment along architectural constraints and symbolic places. The resulting cells spatially represent the abstract types of entities that inhabit indoor space (e.g., rooms, corridors, floors, elevators, or doors) and thus are much more aligned with the conceptual view of indoor space than grid-based approaches or irregular geometric models. Accordingly, the applied partitioning schemas facilitate the coherent integration of semantic, geometric, and symbolic representations. Place graphs and visibility graphs are explicitly constructed on top of the cell decomposition mostly in order to support shortest path searches. Hence, the graphs not only reflect the building topology qualitatively but are also metrically embedded to account for quantitative aspects. All approaches require the cells to be mutually non-overlapping, but (Lorenz & Ohlbach 2006) and (Stoffel et al. 2007) give examples for additionally aggregating cells on different levels of containment hierarchies. Since the space partitioning aligns closely with the human perception of indoor space, cell-based approaches simplify the use of symbolic labels in path queries as well as the generation of human-understandable path descriptions compared to grid-based approaches. In case contextual navigation information is supported, it is commonly given as properties of graph elements (e.g., Stoffel et al. 2007, Yuan & Schneider 2010a, Goetz & Zipf 2011). Only few approaches explicitly model both the free indoor space which supports bodily movement as well as structural elements like walls or pillars as well as obstacles which obstruct the movement. However, if obstacles are only represented implicitly then the indoor space model lacks expressivity. For example, queries such as finding walls of a given material that could be torn down in emergency cases in order to create new passages cannot be performed as walls are neither described semantically, nor geometrically, nor are they translated onto graph elements.

Two major challenges to cell-based approaches are the level of granularity of the navigation graph as well as its consistent and formally defined derivation from the cell structure. The proposed place graphs essentially encode room-to-room connectivity. This granularity is commonly well-suited for pedestrians but different user groups (e.g., elderly or handicapped people) or types of locomotion (e.g., wheelchair, mobile robots) may require more fine-grained cell decompositions and graph representations in order to precisely denote navigable and non-navigable sections within rooms or corridors. (Gilliéron & Merminod 2003) and (Goetz & Zipf 2011) map large corridors onto fine-grained subgraphs but only to better grasp their architectural layout, and fail to relate the elements of the subgraph to an underlying cell structure. The model of (Lorenz et al. 2006) generally foresees arbitrary subspacings of rooms and corridors into smaller disjoint cells but lacks the flexibility to support multiple and different cell decompositions and levels of granularity beyond containment hierarchies. Moreover, the graph construction from subspace cells is not sufficiently formalized. Visibility graphs, on the other hand, as proposed by (Stoffel et al. 2007), (Yuan & Schneider 2010a), and (Liu & Zlatanova 2011b), provide fine-grained graphs inside rooms which however do not imply whether the resulting path is suitable for different types of users or locomotion. Moreover, visibility graphs cannot capture spatial containment hierarchies and hence lack support for multi-level path queries. The linkage of concave room corners as proposed by most approaches renders paths directly along walls which are rather impractical and unnatural (e.g., for wheelchair users or mobile robots, but even for pedestrians). Only (Stoffel et al. 2007) formally base their visibility graph on a corresponding and consistent decomposition of spatial regions into smaller cells. However, multiple cell decompositions reflecting different navigation contexts are also not supported in their model.

The discussed cell-based hybrid space models have in common that they only provide a partitioning of the built environment. Mostly, the indoor space is captured by 2-dimensional cells only, and 3-dimensional aspects of indoor space which may affect navigation such as low ceilings, suspended obstacles, or passages being too low to pass through are not considered. Moreover, the models neither account for complementary sensor spaces or logical spaces nor their integrated modelling, and challenges regarding the localization or tracking of moving persons or objects (cf. chapter 1.2) remain unanswered.

2.2.3 Dual-graph-based approaches

Dual-graph-based approaches can be viewed as subfamily of cell-based hybrid models. They are presented and discussed separately here because they explicitly address one of the main challenges to cell-based approaches, namely the formally defined derivation of graph structures from the cell decomposition of indoor space. All following approaches build upon the mathematical notion of *Poincaré duality* from algebraic topology which also underlies the previous work on the MLSEM (Becker et al. 2009a, Becker et al. 2009b, Nagel et al. 2010). In the following, the Poincaré duality is described informally and as applied in the related work whereas its formal definition is given in chapter 2.5.1.4.

An important work in the dual representation of the interior built environment is presented by (Lee 2001) and further elaborated on in (Lee 2004b) and (Lee & Kwan 2005). The approach utilizes the Poincaré duality to translate 3-dimensional spatial objects and their relationships into a *dual graph* structure. The indoor space is decomposed into 3-dimensional cells which, similar to other cell-based approaches, represent the free space and are aligned with architectural entities such as rooms and corridors. The cells are required to render a topological cell complex (cf. definition A.48), and thus need to be mutually non-overlapping and at most are allowed to touch at their boundaries which are described by lower dimensional cells. The cell complex is said to represent the *primal space* as it follows from the 3-dimensional volumetric shape of the physical built reality. Applying the Poincaré duality to this cell complex carries the primal space representation into *dual space*, intuitively by reversing the dimensions of cells, i.e., 3-dimensional primal cells are paired with 0-dimensional dual cells, primal 2-cells are mapped onto dual 1-cells, and so on. (Lee 2001) lets the dual 0-cells form the set of nodes of a graph structure, with the dual 1-cells constituting its set of edges. If two 3-cells share a common 2-cell on their boundaries in primal space, then their counterparts in the dual graph are linked by an edge. Thus, the dual graph encodes the topological *adjacency* relationships between the architectural entities in primal space, and the Poincaré duality provides the formal tool to derive this graph. The following figure 17 sketches the general idea of the Poincaré duality transformation as applied by (Lee 2001).

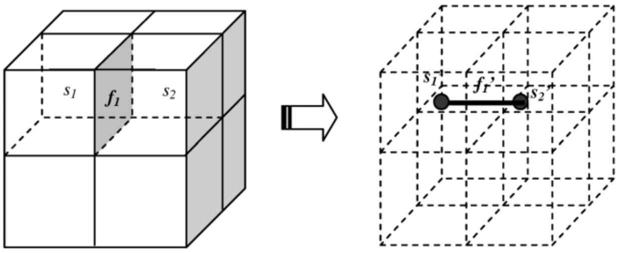


Figure 17: Duality transformation between primal space (left) and dual space (right) (Lee 2004a).

On top of this duality transformation, (Lee 2001) proposes a conceptual model called *Node-Relation Structure* (NRS). The NRS defines a *combinatorial network model* (CNM) of the indoor space that represents the qualitative adjacency relationships between spatial entities using the dual graph. In addition, a subgraph denoting *connectivity* is derived by pruning edges between nodes which cannot be accessed from each other through doors or openings. In order to support shortest path searches, the CNM is embedded in a complementary *geometric network model* (GNM). A key to this step is a straight medial axis transformation (S-MAT) of 2-dimensional polygons into 1-dimensional linear features for approximating large and arbitrarily shaped corridors by their centreline. In (Lee 2004b), a corresponding S-MAT algorithm is presented which simplifies alternative medial axis transforms (e.g., the generalized Voronoi diagram) as it only produces straight line segments. Simple rooms are represented by their centre points and are connected to the centrelines of corridors through orthogonal line segments as well as additional nodes along the centrelines (cf. figure 18, see also Gilliéron & Merminod 2003, Goetz & Zipf 2011). Thus, a single node in the CNM may be reflected by a subgraph in the GNM. Such nodes are called *master nodes* and induce a hierarchical relationship between the CNM and the GNM. However, this hierarchy is not underpinned by a corresponding cell partitioning and thus is less expressive than the containment hierarchies proposed by (Lorenz & Ohlbach 2006) or (Stoffel et al. 2007).

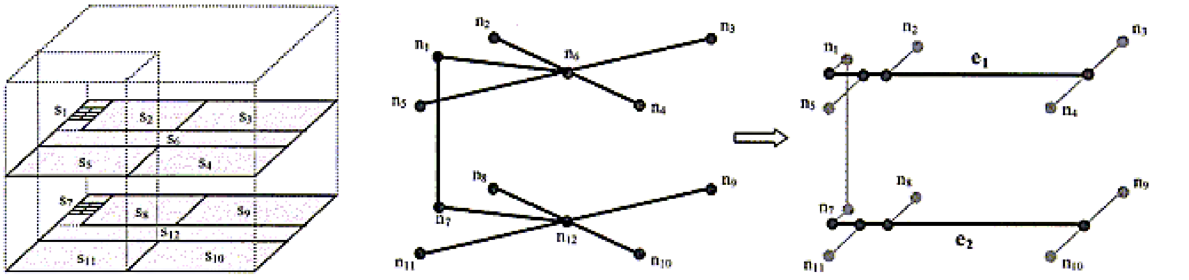


Figure 18: Example 3-dimensional setting (left), resulting CNM with two master nodes n_6 and n_{12} (middle), and corresponding GNM (right) (Lee 2004b).

The NRS has been applied in several research works in the field of indoor navigation and the spatial analysis of interior built environments. For example, (Lee 2004a) proposes a 3-dimensional indoor geocoding technique for the analysis of human activities through space and time which combines indoor positioning data with the building structure given as NRS (see also Park & Lee 2008 and Lee 2009). In (Lee & Zlatanova 2008), the NRS is embedded in a conceptual framework for an urban-scale emergency response system as essential component for the computation of shortest evacuation paths. The path search considers environmental factors such as the building damage status, toxicity status, and power status as well as traffic capacities of halls, corridors, or stairs, all of which are given as additional properties on graph elements. The authors demonstrate the coupling of the NRS with urban networks for outdoor navigation (e.g., street networks, underground transportation networks) in order to guide people away from disaster sites (cf. figure 19). (Li & He 2008) attempt to enrich the NRS with semantic information by combining it with the conceptual model of (Meijers et al. 2005) in a common framework.

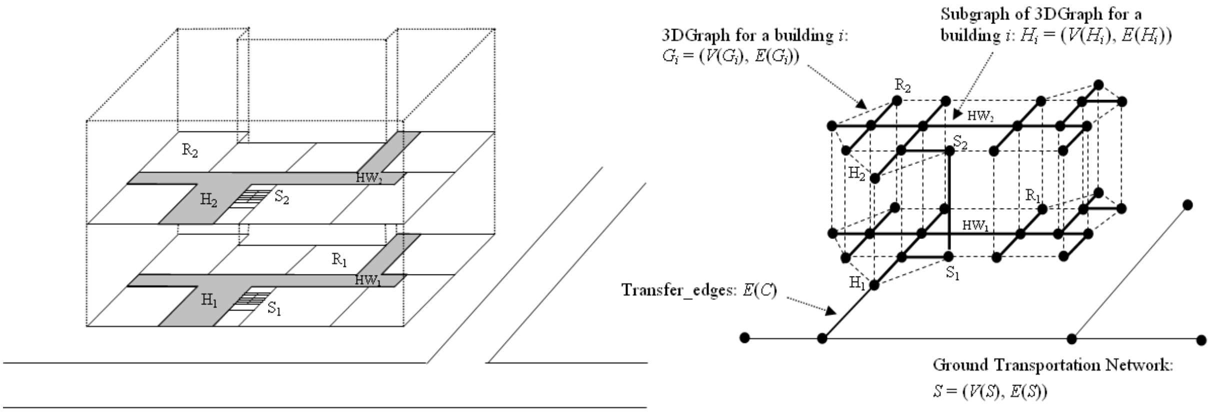

R2
HW2
H2
S2
R1
HW1
H1
S1
3DGraph for a building i:
$G_i = (V(G_i), E(G_i))$
Subgraph of 3DGraph for a building i: $H_i = (V(H_i), E(H_i))$
Transfer_edges: $E(C)$
Ground Transportation Network:
$S = (V(S), E(S))$

Figure 19: NRS resulting for an example building and graph-based link to the street network (Lee & Zlatanova 2008).

Another cell-based indoor space model relying on the Poincaré duality is proposed by (Jensen et al. 2009) and (Jensen et al. 2010). The authors apply the dual space transformation to 2-dimensional cells that represent spatial regions such as rooms, stairs, or corridors. Similar to (Lee 2001), the resulting dual adjacency graph is thinned to a connectivity graph. Moreover, the direction of movement permitted by doors and exits is translated onto a set of directed edges which results in an additional *accessibility graph* representation. Further access constraints such as security-based access restrictions or time-based access constraints are however not reflected. An example layout of a building floor together with the corresponding connectivity and accessibility graphs is depicted in figure 20. In contrast to all cell-based hybrid approaches discussed so far, (Jensen et al. 2009) propose a complementary cell decomposition of indoor space along sensor characteristics to facilitate positioning and tracking tasks in indoor environments. The partitioning into sensor cells is done so that an indoor object cannot move from one sensor cell to another without being observed from a sensor of the localization technology. The shape and size of sensor cells hence depends on the localization technology and the deployment of sensors. (Jensen et al. 2009) exemplify their approach based on RFID readers that are mounted at doors. The resulting sensor cells mainly coincide with the layout of rooms and corridors but may also span several rooms. The partitioning is captured by a separate *deployment graph* which encodes the topological relationships between sensor cells and which can be explained through Poincaré duality. The graph edges are labelled with symbolic tags identifying the RFID readers which register the movement between two cells (cf. figure 21). Based on their structuring of indoor space into architectural cells and sensor cells, (Jensen et al. 2009) develop algorithms for the efficient and effective tracking of indoor objects. However, the integrated modelling of arbitrary sensor spaces as well as additional indoor navigation tasks such as path planning or guidance along a path are not addressed by (Jensen et al. 2009).

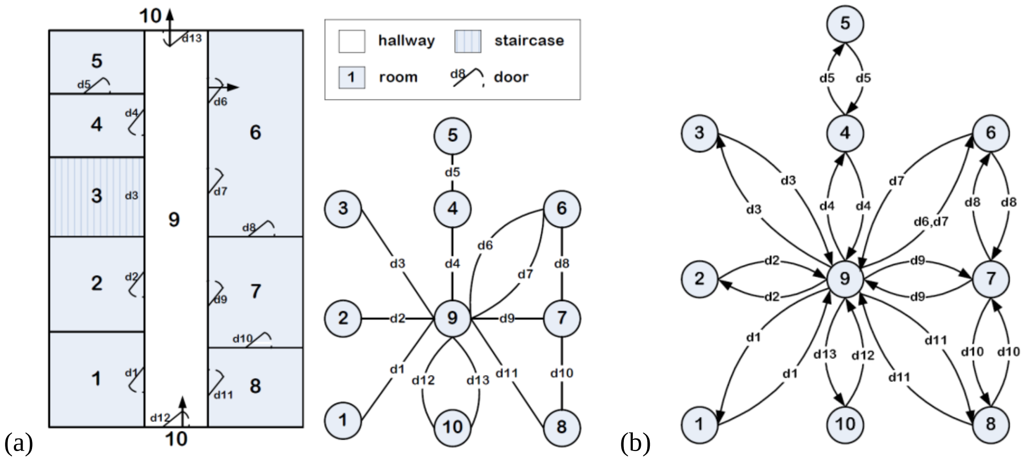


Figure 20: Example floor plan with corresponding connectivity graph (a) and accessibility graph (b) (Jensen et al. 2009).

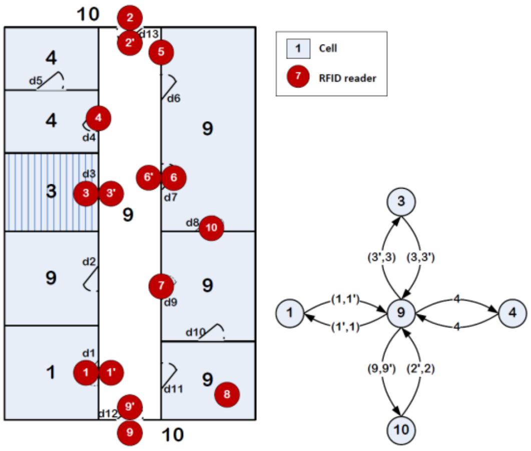


Figure 21: Deployment of RFID readers and corresponding deployment graph (Jensen et al. 2009).

In a series of papers, (Boguslawski & Gold 2009), (Boguslawski & Gold 2010), and (Boguslawski & Gold 2011)) develop a *dual-half edge* (DHE) data structure for the simultaneous representation and storage of primal and dual 3-dimensional cell decompositions of the interior built environment. The building interior is modelled in primal space as a set of non-overlapping polyhedral cells that are adjacent and connected, and geometrically describe the volumetric shape of free indoor spaces such as rooms and corridors. The dual representation of this cell complex is given by a dual graph whose nodes represent the centre points of the primal 3-cells and whose edges capture their topological adjacency relationships. The DHE structure is an extension of work done in the field of 2-dimensional computational geometry, and has its roots in approaches for the representation of subdivisions of 2-manifolds such as the *quad-edge* structure of (Guibas & Stolfi 1985), the *half-edge* proposed by (Mäntylä 1988), and the *augmented quad-edge* approach of (Ledoux & Gold 2007). To represent a 3-cell of the cell complex in primal space, the DHE only stores its edges and vertices. A geometric edge is decomposed into a pair of half-edges each pointing to its associated vertex and its paired half-edge that forms the opposite of the edge. Two more pointers are defined, one to the next half-edge around the associated vertex, and one to the next half-edge around the same face. The faces on the boundary of the 3-cell are hence implicitly represented through loops of edges which follow from navigating along this pointer structure. In the same way, the loop of faces implicitly bounds the 3-cell. The dual representation results from permanently pairing each half-edge with a dual counterpart that provides equivalent pointers in dual space. (Boguslawski & Gold 2009) make this pairing the atom of the DHE data structure. It thus renders a primal and a dual graph structure which can be navigated separately but also facilitates the transition between both spaces at every graph element. The resulting indoor space representation is depicted in the following figure based on a spatial setting originally used by (Lee 2001).

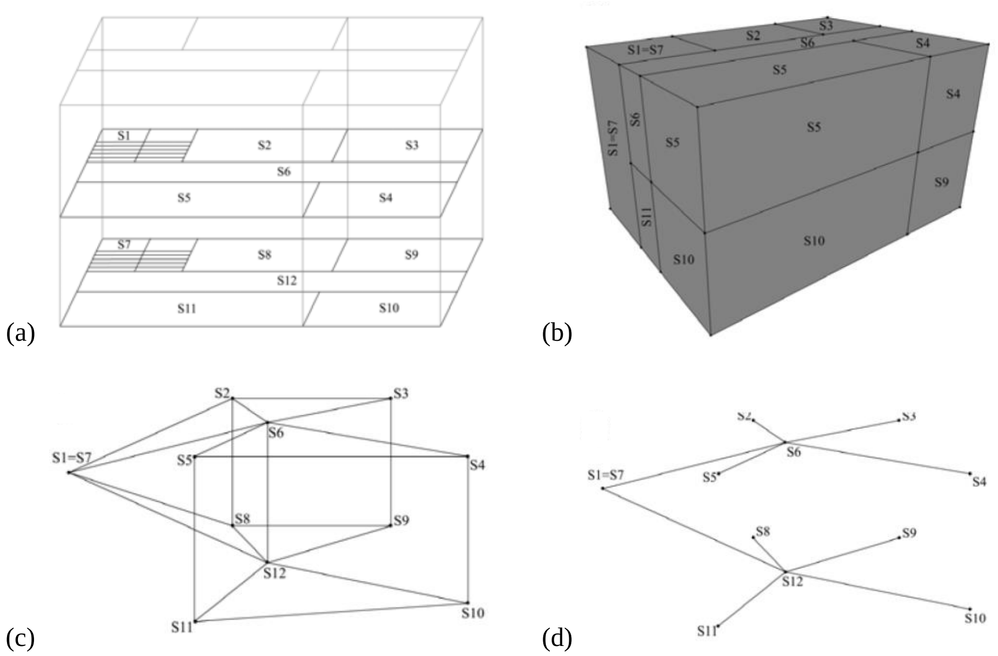


Figure 22: Example building interior after (Lee 2001) (a), volumetric model of rooms (b), corresponding dual graph (c), and connectivity graph (d) (Boguslawski & Gold 2009).

(Boguslawski & Gold 2009) implement path planning algorithms on their data structure and also propose the translation of the dual adjacency graph into a *connectivity* graph (cf. figure 22d) for this purpose. However, since edges cannot be simply removed without affecting the intactness of the DHE structure, the authors introduce edge weights as additional attributes which account for *strong connections* (e.g., doors or other passages) and *weak connections*. The latter edges actually mean no passage, but could, for example, be evaluated in emergency situations to search for walls that can be torn down. A complementary geometric embedding of the dual graph similar to the GNM (cf. Lee 2001) or hierarchical containment relationships between cells are not foreseen by the DHE. Thus, if a corridor should be represented by several dual nodes it has to be split into accordingly many cells in primal space. (Boguslawski & Gold 2011) semantically classify the surfaces separating the single corridor sections as *virtual doors*. An interesting aspect of the DHE which goes beyond the model of (Lee 2001) is the formal definition of an extended set of *Euler operators* that allow for constructing and modifying the cell complex of the building interior while at the same time maintaining its sound dual-half edge representation (cf. Boguslawski & Gold 2010, figure 23). In (Boguslawski & Gold 2011), the authors apply their model to the representation of the surrounding terrain in order to facilitate indoor-outdoor path searches. This is achieved by representing the outdoor environment through additional thin 3-dimensional cells.

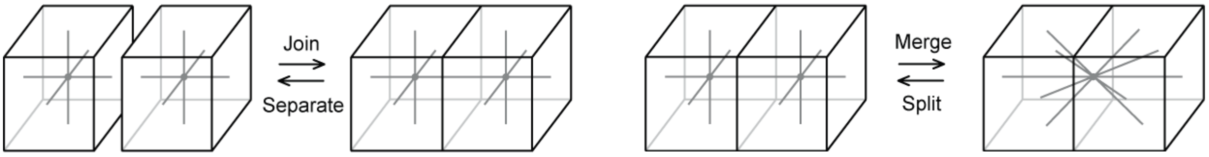


Figure 23: Example Euler operators as defined by (Boguslawski & Gold 2011) on their DHE structure.

It is important to emphasize that the dimension-reversing duality relation between the primal cell decomposition of a topological manifold and its dual is an elementary topological fact. Thus, the Poincaré duality also underlies the work of (Boguslawski & Gold 2009). The key difference of the DHE approach compared to the NRS model of (Lee 2001) is that it incorporates both subdivisions in a single data structure. In contrast, the NRS has to be extracted from a given primal cell complex in a subsequent step which is also true for the model proposed by (Jensen et al. 2009) and the former publications on the MSLEM.

The strength of the discussed dual-graph-based methods is that they are based on a consistent mathematical framework. Since the knowledge about the primal cell decomposition implies the knowledge about its dual (and vice versa), it obviously follows that the derivation of the dual adjacency graph is deterministic and can be proven to be correct. However, the research for this thesis revealed issues in the presented dual-graph-based approaches which are discussed in detail in chapter 3. Whereas most cell-based approaches are restricted to two dimensions, the Poincaré duality generally holds in both two and three dimensions and thus supports the sound definition of 3-dimensional indoor space models. The duality relation however is not feasible to express containment relationships between spatial entities as this requires overlapping cells in primal space which violates the notion of a topological cell complex. Thus, spatial containment hierarchies as well as the further challenges to cell-based approaches identified in the previous chapter need to be addressed explicitly by hybrid space models that build upon the Poincaré duality. For the presented dual-graph-based approaches it can be summarized that the integrated modelling of complementary space representations beyond the physical built reality is not sufficiently addressed, and neither are different levels of granularity of the cell partitioning in order to denote spatial containment or navigable and non-navigable spatial regions for different user groups and types of locomotion.

2.2.4 Conceptual-based approaches

The survey of existing hybrid approaches concludes with examples of research works putting strong focus on the conceptual and semantic modelling of indoor space and the definition of navigation ontologies rather than on geometric-topological space aspects and their mappings onto graph-based representations. Conceptual models are increasingly being developed in the field of indoor navigation and the key questions raised concern the knowledge about the indoor environment and its constituent elements that is required to support navigation tasks, the level of granularity of the knowledge representation as well as the reasoning and inference about this knowledge.

(Anagnostopoulos et al. 2005), (Tsetsos et al. 2006), and (Kolomvatsos et al. 2009) develop an extensive framework called *OntoNav* for the semantically enriched navigation in indoor environments. It employs a consequently user-centric view and aims at providing users with navigation paths and guiding descriptions according to their *physical* and *perceptual capabilities* (e.g., mode of locomotion, ability to see, familiarity with the environment) as well as their *routing preferences* (e.g., fastest or least-effort-based routes, points of interests). OntoNav defines an *Indoor Navigation Ontology* (INO) which captures the spatial and structural concepts of indoor space necessary to fulfil these goals. The taxonomy of this ontology is inspired by the semantic differentiation of *locations* and *exits* as proposed by (Hu & Lee 2004). The following figure 24 shows the INO of OntoNav formally notated in OWL.

The basic INO concepts are *Space*, *Path_Element* and *Obstacle*. Spaces decompose the interior environment into rooms, corridors, floors, and buildings at different hierarchical levels. A path between two spaces is represented through a sequence of path elements whose representation is twofold. First, a *Passage* models a spatial path element through or along a user may pass and is further classified into *horizontal* (e.g., doors, ramps, corridor segments) and *vertical* (e.g., stairs, escalator, elevator) passages both of which can be *motor-driven*. Second, a *Path_Point* provides a point-based abstraction of spaces, passages and their exits, as well as special waypoints (e.g., turning points) along the path. Obstacles preventing the movement of users are explicitly represented and comprise physical constraints (e.g., physical objects blocking the way) and logical constraints (e.g., security policies, non-functional elevator). Each class is further enriched with semantic properties and exhibits relationships to other concepts.

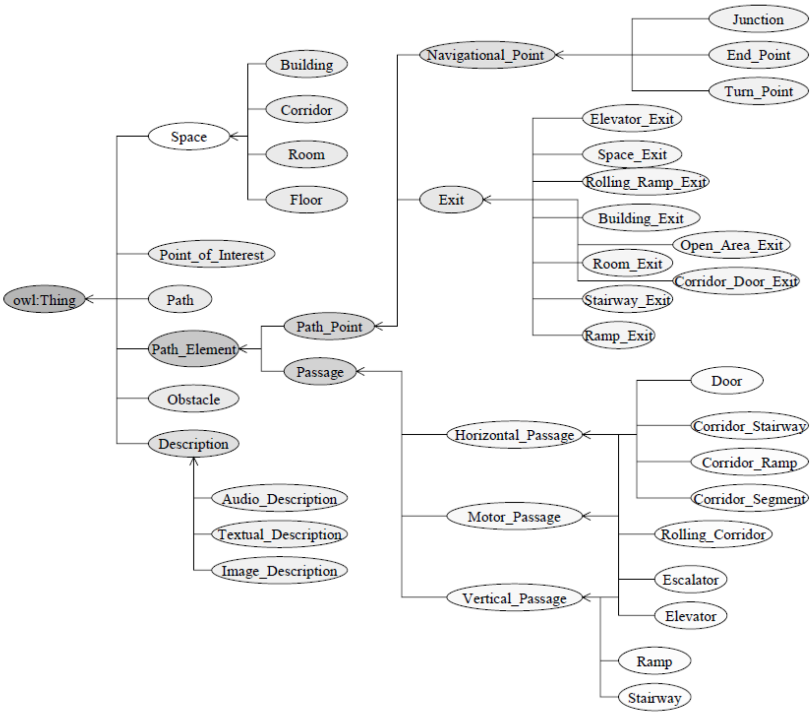


Figure 24: The Indoor Navigation Ontology of OntoNav (Tsetsos et al. 2006).

The INO is accompanied by a *User Navigation Ontology* (UNO) that models the navigation-oriented user profile (cf. figure 25) which describes, amongst others, the mental and physical capabilities as well as the preferences of a particular user. Conditional statements are then used to establish a rule base which governs user-dependent path searches and reasoning processes (e.g., that stairs are excluded from path searches given that the user is handicapped). Guidance is realized through user-tailored path descriptions (e.g., audio messages for blind people) which are conceptually modelled as *Description* class in the INO. Although the authors propose the construction of navigation graphs based on the conceptual point-based representation of path elements, they do not present formally sound methods for the instantiation of their semantic concepts or the derivation of graph structures from, for example, building floor plans. Moreover, OntoNav neither conceptually nor practically addresses challenges in positioning and tracking of navigation users as identified in chapter 1.2.

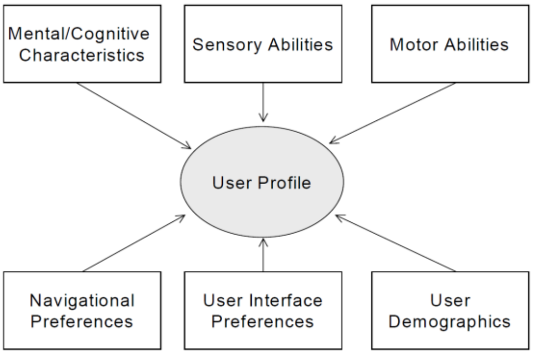


Figure 25: The User Navigation Ontology of OntoNav (Tsetsos et al. 2006).

(Dudas et al. 2009) propose an ontology and algorithm for indoor routing (ONALIN) which are meant to provide indoor routes for individuals taking into consideration the needs and preferences of different user groups such as the physical, cognitive, or sensory impaired. The ontological description of the physical built environment incorporates concepts of the *American Disability Act* (ADA)[4] standards which pose requirements on the layout of public buildings to ensure accessibility for disabled people. The resulting indoor space ontology is depicted in figure 26 and shows considerable agreement with OntoNav. The key concept is similarly called *PathElement* and represents the types of navigable spatial elements that make up the navigation path such as horizontal and vertical *Passages*,

[4] The ADA standards are issued by the U.S. Department of Justice and can be freely obtained from http://www.ada.gov/.

Exits, and *Entrances*, as well as their conceptual mapping onto a *Network* structure. *Rooms* are understood as points of interest and reflect the targets and goals of path searches (i.e., symbolic places). Moreover, the ontology includes separate classes for the modelling of *Obstacles* and *Users* with their route preferences. Based on the network representation, (Dudas et al. 2009) present a two-stage path search algorithm which first looks for *feasible routes* by pruning the network structure to match the user's capabilities, and afterwards computes *comfortable routes*. However, like with OntoNav, ONALIN lacks details about instantiating the abstract types of semantic entities and the graph structure. Moreover, rules for pruning the network or for finding comfortable routes are not formally specified.

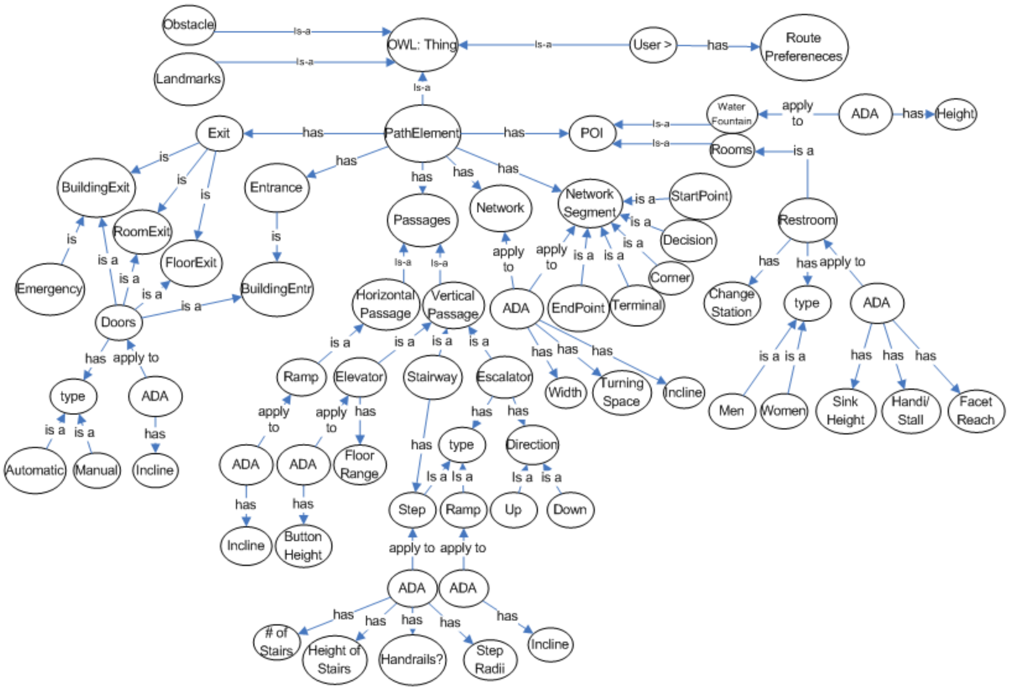


Figure 26: Indoor space ontology as defined by ONALIN (Dudas et al. 2009).

A surface-based conceptual model that integrates geometric, semantic, and navigational aspects of the interior environment is defined by (Slingsby & Raper 2008). The indoor space is geometrically described by a set of surfaces which represent the ground surfaces of 3-dimensional volumetric spatial entities (e.g., rooms, corridors, or stairs). The surfaces are layered on top of each other in three dimensions to account for the several floors in a building. The surface morphology is expressed through constraints such as breaklines in order to generate a topologically consistent and connected representation (cf. figure 27a). The conceptual model distinguishes between *spaces*, *barriers*, *portals*, and *teleports* which semantically classify different parts of the surfaces. A space is a spatial region with a specific function and coincides with the notion of a cell in cell-based approaches, whereas portals denote the entry and exit points of spaces (e.g., doors and windows). Barriers represent obstacle objects such as walls or fences, and teleports abstract from elevators, ramps, or stairs. Following the model of (Meijers et al. 2005), each semantic feature is enriched with additional attributes providing information about its persistency, accessibility, direction of movement, or structure. Moreover, users are explicitly modelled as conceptual entity and carry attributes such as access rights or their negotiable height for steps (which, for example, tends to zero for wheelchair users). An example instance of the model considering both indoor and outdoor surfaces is shown in figure 27b. However, the conceptual model of (Slingsby & Raper 2008) is only given descriptively and has not been formalized. Although discrete and connected navigable spaces in indoor space are represented, aspects such as path planning, localization of navigation users, tracking, or guidance based on this representation are not discussed by the authors.

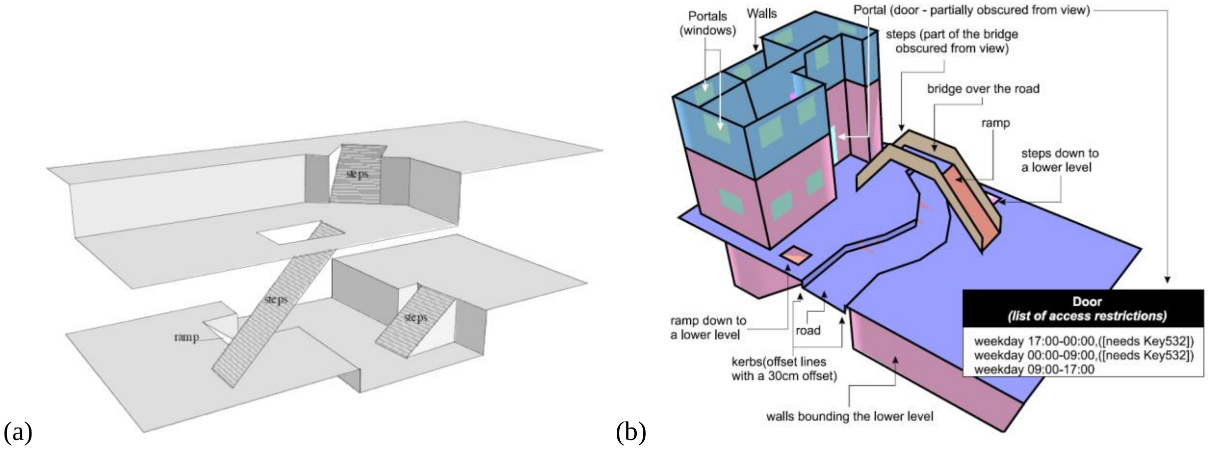

steps
steps
steps
ramp
(a)
Portals (windows)
Walls
Portal (door - partially obscured from view)
steps (part of the bridge obscured from view)
bridge over the road
ramp
steps down to a lower level
ramp down to a lower level
road
kerbs(offset lines with a 30cm offset)
walls bounding the lower level
Door
(list of access restrictions)
weekday 17:00-00:00,([needs Key532])
weekday 00:00-09:00,([needs Key532])
weekday 09:00-17:00
(b)

Figure 27: Example surface-based representation of an indoor scene (a) and extension with outdoor surfaces and access restrictions (b) (Slingsby & Raper 2008).

(Yang & Worboys 2011) have started work on developing a unified ontology for navigation in indoor and outdoor spaces. The proposed concepts are intentionally aligned with the theory of image schemas and affordances as introduced in chapter 2.1.3, and are based on the hypothesis that the human perception of indoor and outdoor space is essentially the same, and that so are the action possibilities offered by some of the entities therein. The authors thus regard image schemas and affordances as the proper instrument to bridge the gap between ontologies for either space. The indoor space is structured by (Yang & Worboys 2011) through the following ontology.

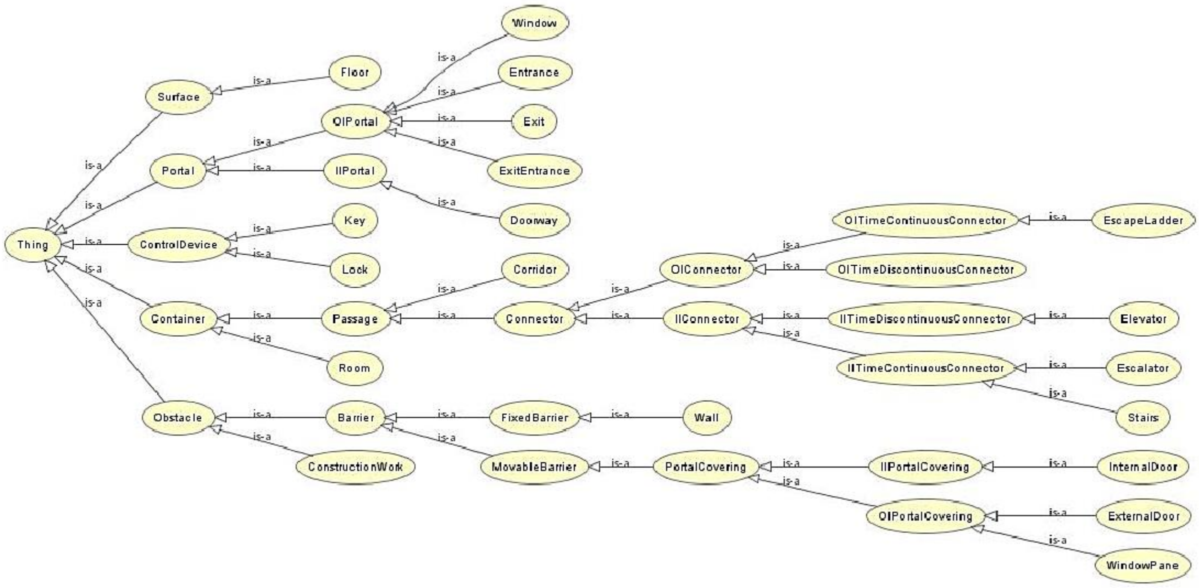


Figure 28: Indoor space ontology as proposed by (Yang & Worboys 2011).

The central concepts defined by (Yang & Worboys 2011) are *Container*, *Surface*, *Portal*, and *Obstacle*. A container is a spatial region that is perceived as a unit and that has the affordance of containing things. It is further subtyped into *Room* and *Passage* with a passage being a way or channel supporting free movement (e.g., a corridor or a segment thereof). A *Connector* is a special type of a passage which connects two containers and assists in transitioning between them (e.g., stairs, elevators). Portals are openings affording the passage from one container to another such as doors or openings in walls. The *Obstacle* concept abstracts from all fixed or movable barriers that may impede or obstruct movement from one place to another. Whereas all these concepts are also defined in the approaches discussed above (either identically or in a similar way), the notion of a *Surface* as spatial boundary of a container that extends in two dimensions and supports other objects (e.g., floor surfaces of rooms) is only represented as such by (Slingsby & Raper 2008) and (Meijers et al. 2005). The ontology of (Yang & Worboys 2011) further includes concepts which connect outdoor and indoor space such as an *OIConnector* (e.g., escape ladder) or *OIPortal* (e.g., windows or entrances and exits of buildings). The proposed concepts are also feasible to explain structural entities relevant to outdoor navigation. For example, a city or its buildings are containers. Roads are conceptually to be seen as passages, and since their pavement supports vehicles it matches the surface schema. An example for an outdoor obstacle is a road construction which impedes traffic. The ontology hence suggests a common indoor-outdoor language.

(Yang & Worboys 2011) distinguish between the structural conceptualization of space and the definition of concepts which are only required for the task of navigation (cf. chapter 2.1.3). The indoor space ontology is seen as an upper-level *domain ontology* supporting several tasks, one of them being navigation. The proposed navigation *task ontology* is illustrated in the following figure 29.

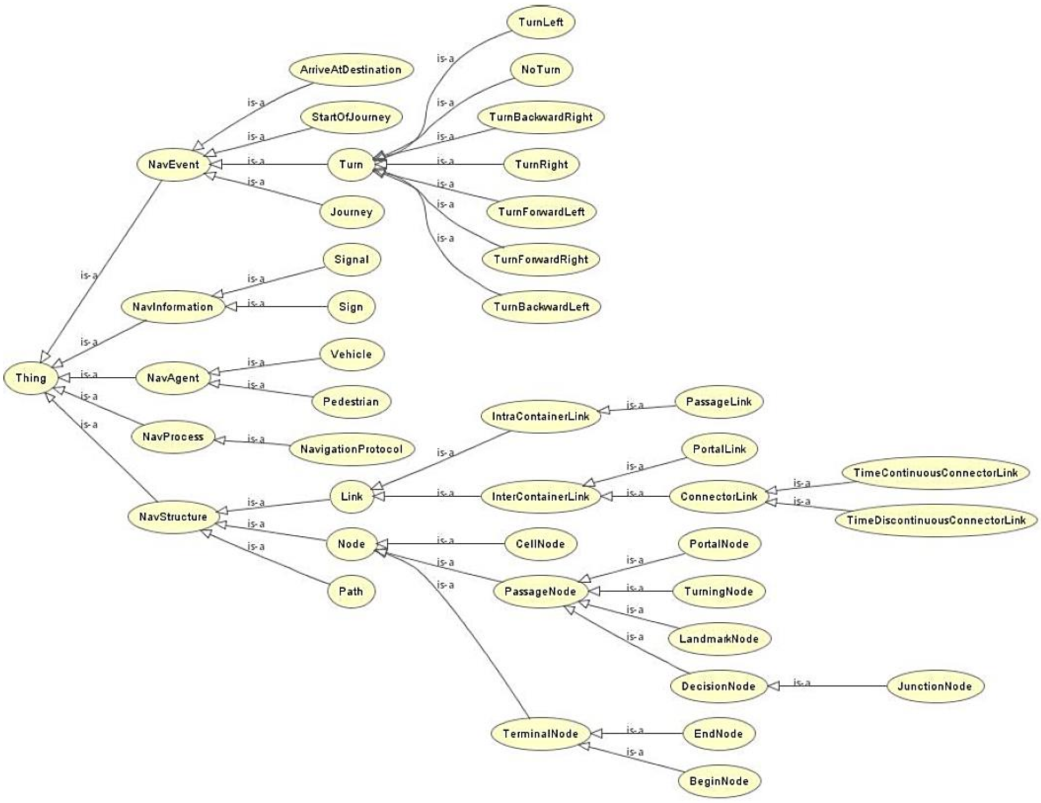


Figure 29: Navigation task ontology as proposed by (Yang & Worboys 2011).

The navigation ontology maps the conceptual entities defined in the upper-level indoor space ontology onto a semantic *Node/Link* view represented by the *NavStructure* concept. A *CellNode* translates containers to nodes, which aligns with the graph-based representation of non-overlapping cells in most cell-based approaches. *InterContainerLinks* express connectedness relationships between containers realized through portals and connectors, whereas qualitative hierarchical containment relationships following from the decomposition of a container into smaller passages (e.g., segments of a corridor) are denoted by a distinct link type called *IntraContainerLink*. Similar to (Tsetsos et al. 2006), additional waypoints such as turning points, decision points, landmarks, or terminal points are included in the navigation ontology as subtypes of *Node*. The node/link structure is proposed to serve as input for path search algorithms, and the resulting sequence of nodes and links along which a user travels or moves is explicitly represented through the *Path* class. In order to account for user-dependent contextual information in navigation tasks, the group of users is conceptually mapped onto *NavAgent* and further classified into pedestrians and vehicles (e.g., wheelchair users, mobile robots). (Yang & Worboys 2011) also model objects affording additional navigation information (e.g., signals or signs) as well as navigational events that occur while traveling through the interior environment such as turns in movement or the arrival at the destination based on which path descriptions and route instructions can be generated. The ontologies and their interrelationships are formally defined but the work of (Yang & Worboys 2011) does neither specify qualitative reasoning or inference strategies about the represented knowledge, nor methods for the coherent linkage of model elements in both ontologies.

In contrast to grid- and cell-based hybrid approaches, the illustrated conceptual-based approaches essentially apply logical criteria instead of geometric facts to both the conceptualization of indoor space and its ontological structuring. The proposed conceptual models substantially overlap in the semantic definitions and taxonomies of their identified spatial entities, and hence contribute to a common understanding of the minimum information needs about the interior built environment for the purpose of indoor navigation. It is noteworthy that the explicit representation of obstacles and non-navigable spaces resulting from spatial or user-contextual constraints is considered by all conceptual models and hence plays a more central role than in cell-based models which mainly capture the free indoor space only. Moreover, most approaches support arbitrary decompositions of spatial units into smaller parts at the conceptual level and do not restrict this partitioning to be aligned with architectural constraints. Thus, path elements and spatial regions can be described in a more fine-grained way than the room-to-room connectivity of most cell-based models. Accompanying point-based abstractions of spatial entities as well as links denoting transition possibilities are identified as fundamental spatial concepts, but lack a formal specification in terms of graph theory. The conceptual-based approaches introduce an extra level of model expressivity by representing knowledge about the different groups of users travelling through indoor space as well as their navigation context, which considerably goes beyond adding attributes on graph elements as proposed in other approaches. (Tsetsos et al. 2006) demonstrate a mathematical sound way for reasoning about this user knowledge in their OntoNav framework.

However, the quantitative dimension of indoor space is not sufficiently represented in either approach, and precise definitions of what semantic concepts such as spaces, containers, passages, path elements, or obstacles as well as their point- and link-based abstractions mean in a geometric-topological sense are missing. Path searches accounting for quantitative aspects are hence not supported and visual guidance along the path is, at least, hindered. Furthermore, and alike almost all of the presented hybrid approaches, the formal integration of complementary indoor space conceptualizations that do not follow the physical built reality and how it is perceived by humans, but rather describe decompositions of indoor space along sensor characteristics or logical aspects, is neither approached nor solved.

2.3 Building Models

The modelling of the interior built environment is not only addressed in the context of indoor navigation but naturally also in fields such as urban and building information modelling. Models from these fields aim at representing the building as designed or observed, with varying spatial and semantic information about building parts, structural building elements, the building interior, and objects fixed to or contained inside the building down to the level of single power plugs. The expressivity of such building models with respect to the built reality is therefore usually greater than that of the presented indoor navigation models. Although building models are not primarily designed for the task of indoor navigation, they nevertheless provide rich information about the indoor space which can serve as important source of information in the context of indoor navigation. For this reason, important representatives of building models are presented in the following sections and are classified according to the scheme from chapter 2.1.

2.3.1 Building Information Models (BIM)

Building information modelling refers to the collaborative process of generating and maintaining building data during the entire project life-cycle of buildings and construction facilities including all phases from earliest conception and design, through construction, functional life and maintenance, to demolition and disposal (Eastman 2008). This process results in and relies upon *Building Information Models* (BIM)[5] which are commonly understood as shared knowledge resource providing highly accurate and valuable information about the built environment and its structures as reliable basis for decision making in the different disciplines participating in the building life-cycle management such as architecture, engineering, construction, and facility management (AEC/FM) as well as computer-aided architectural design (CAD)[6]. A BIM model represents the physical and functional characteristics of a facility within a computer and incorporates the spatial and thematic properties of all building components, their well-defined meaning, as well as their structural, spatial, and logical interrelationships. A multitude of appli-

[5] BIM is also used interchangeably as acronym for the process of Building Information Modelling in literature.

[6] The acronym CAD stands for computer-aided design in general which is the use of computer technology for the task of drafting and designing “things”, whereas the drafting of architectural drawings is often referred to by the acronym CAAD. In the course of this thesis, the acronym CAD is consistently being used for both notions.

cations for BIM models is quickly emerging in the field of building informatics which go beyond traditional engineering and visualization needs of the AEC/FM industry and include applications domains such as environmental simulations and energy assessment, greenhouse planning, civil engineering, and disaster management.

BIM models have evolved from traditional ways of documenting facilities with paper-based drawings or their digital counterparts in legacy 2-dimensional CAD software systems mainly because of two shortcomings. First, an adequate sketch of the 3-dimensional setting of the built environment based on 2-dimensional building plans requires multiple and layered views, resulting in highly redundant and hence error-prone data. Moreover, many simulations and analyses throughout the life-cycle phases of a building (e.g., structural stress analysis) require true 3-dimensional data. Second, CAD drawings depict the building layout through graphical elements such as lines and arcs which are annotated with symbols and textual descriptions. Although this representation is understandable for human professionals, it impedes the automatic interpretation of building components and the spatial reasoning about the indoor space by machines. A BIM model involves the object-oriented description of a facility where each building component is mapped onto a semantic entity carrying attributes and relationships. The notion of a building component comprises solid-shape structural elements such as walls, beams, columns, or slabs as well as abstract void spaces such as the intangible volume of a room. The geometry of building components is usually given in three dimensions, for example, as volumetric object or through its projection onto a 2-dimensional footprint or profile. Current BIM tools also support the parametric description of objects, i.e. objects unfold at a given location in space according to a set of rules embedded in them. These rules may only concern the shape and size of the object, or additionally govern its physical and functional relationships to surrounding objects. Due to the consideration of the building life-time, BIM models are said to be living in four dimensions. The consideration of additional project processes and documentations even renders an n-dimensional model space (cf. Aouad et al. 2007). This integrated n-dimensional knowledge representation about the interior built environment is to be seen as the main difference to recent object-oriented 3-dimensional CAD models of building designs.

BIM models are typically created manually during the planning or construction phase by architects or civil engineers and hence represent a facility *as designed* following a generative modelling approach (Nagel et al. 2009). Although they are maintained throughout the building life-cycle, they are mostly available only for newly planned or recently constructed buildings. However, the U.S. National BIM Standard (NBIMS) has made the provision of BIM models compulsory at early stages of building projects in the public sector. Similar initiatives can be observed on national or organizational levels (e.g., in Denmark and Finland), as well as in literature (e.g., Benner et al. 2010). This development is expected to further boost the acceptance and availability of BIM models.

Starting from 1995, the International Alliance for Interoperability (IAI) pioneered an international standardization attempt involving organizations from 19 countries to define a single building model as one authoritative semantic definition of building elements, their properties and interrelationships (Howell & Batcheler 2005). This effort succeeded with the release of the *Industry Foundation Classes* (IFC), a vendor-neutral open information model and commonly used format for capturing and sharing data in the BIM domain. IFC is based on the ISO STEP standards family (ISO 10303) for the representation and exchange of product model data between CAD systems, and is itself registered by ISO as ISO/PAS 16739 to be approved as international ISO standard. The IFC specification is further developed and maintained by the non-profit and industry-led organization buildingSMART[7], and is embedded in the Open BIM initiative which aims at developing open standards and workflows as basis for an universal approach to the collaborative design, realization, and operation of buildings. The latest official version of IFC has been released in 2007 as IFC2x Edition 3 Technical Corrigendum 1 (IFC2x3, cf. Liebich et al. 2007), whereas the next future release named IFC4 is currently under preparation. The IFC information model is given in the EXPRESS (ISO 10303-11:1994) data definition language which is a formal part of the STEP standard and which can be viewed as enhancement of the entity-relationship model. The data exchange between IFC compliant software applications uses clear text encodings based on the STEP physical file format (ISO 10303-21:1994). An XML-based encoding of IFC2x3 (ifcXML) is additionally available from buildingSMART.

IFC defines several hundred entities arranged in a taxonomic hierarchy that render a rich semantic view on all aspects of a building design such as spatial and physical building elements as well as distribution and service elements (e.g., plumbing, heating, or electrical systems). The main components constituting the structure and ar-

[7] buildingSMART is the successor organization of IAI; see http://www.buildingsmart.org/ for more information.

chitectural layout of the built environment are grouped under the name *shared building elements* (*IfcSharedBldgElements*) as they carry building information relevant to all application domains accessing the BIM model. This element group hence provides a conceptual model of indoor space which is depicted in the following figure 30 as informal UML-based representation.

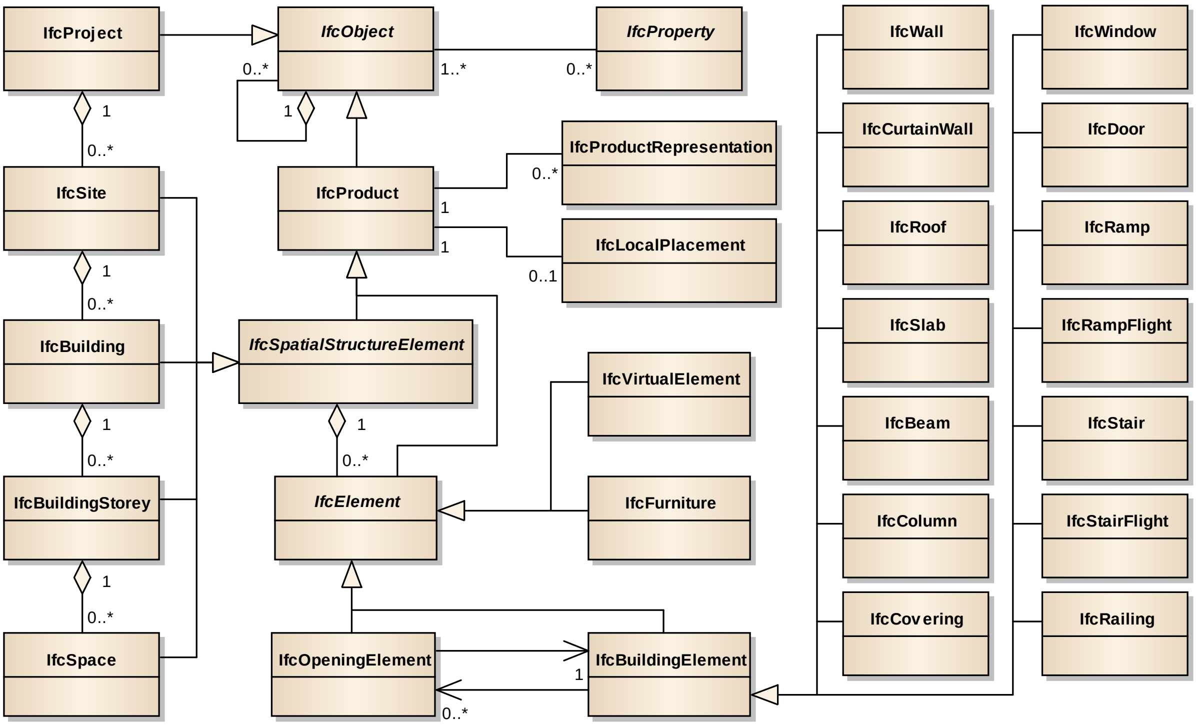


Figure 30: Informal UML diagram of the IFC shared building elements (excerpt) (after Benner et al. 2005).

The indoor space is decomposed into spatial elements (*IfcSpatialStructureElement*) structuring an entire building (*IfcBuilding*) into floors (*IfcBuildingStorey*) which may contain several spaces (*IfcSpace*). This part-whole structure denotes a spatial containment hierarchy. A space is understood as bounded volume or area enclosed by built or virtual boundaries which provides a specific function (e.g., rooms, corridors). Spaces are allowed to span over several connected spaces in order to express containment relationships on a single building floor. Each spatial element can be populated with physical building elements (*IfcBuildingElement*) which represent the primary parts of the building construction and its space separating system. Examples for building elements are walls (*IfcWall*), slabs (*IfcSlab*), roofs (*IfcRoof*), columns (*IfcColumn*), or stairs (*IfcStair*), as well as openings such as doors (*IfcDoor*) and windows (*IfcWindow*). Intangible boundaries of building elements can be expressed through virtual elements (*IfcVirtualElement*). The building elements also account for fixed objects inside spaces, whereas movable objects are mainly captured by *IfcFurniture* and comprise tables, desks, or chairs.

All spatial and physical elements are transitively derived from *IfcProduct* which serves as abstract root concept for any semantic object that exhibits a geometric or spatial context. Each such element can have an explicit geometric and/or topological representation of its 3-dimensional volumetric shape (*IfcShapeRepresentation* via *IfcProductRepresentation*). For this reason, an IFC model is also referred to as *volumetric elements model* in literature (e.g., Nagel et al. 2009). The geometry is commonly placed in a local Cartesian engineering coordinate system (*IfcLocalPlacement*) which may be defined relative to the placement of another product or enclosing element hierarchy. Every product has a unique identifier to distinguish it from other products in a world-wide context, and possesses thematic properties which further classify or characterize the element. For example, rooms may be identified by their number or name, and windows and doors may provide information about their opening direction, operation type (e.g. double swing), hinge location, thermal properties or whether they are fireproof. The construction material (*IfcMaterial*) of each building component can be specified as a whole or in layers (e.g., to depict the assembly of different materials), and also comprises visual appearances such as colours or textures. Relationships between entities are modelled explicitly through relationship classes. For example, *IfcRelDecomposes* denotes a part-whole relationship and expresses exclusive spatial or logical containment, whereas *IfcRelConnects* denotes the logical or physical connectedness between two building entities.

It obviously follows that the conceptual description of indoor space in IFC largely conforms to most of the concepts defined in cell-based and conceptual-based hybrid approaches presented in chapter 2.2, and if evaluated against the classification scheme developed in chapter 2.1, IFC could be easily assigned to the category of hybrid space models. The subdivision of the interior built environment into *IfcSpaces* essentially is a cell partitioning of the free space which includes hierarchical containment relationships. The ontological space representation explicitly models all objects required by most navigation approaches (e.g., rooms, corridors, floors, exists, stairs, physical and virtual boundaries, obstacles, etc.), and the rich semantics associated with the spatial entities covers most of the proposed symbolic and semantic properties and relationships. Since IFC is designed as comprehensive building model, it thus naturally exceeds the expressivity of many space models for indoor navigation regarding the representation of the built-up space. However, and for the same reason, IFC lacks essential components required to answer the challenges to indoor navigation as enlisted in chapter 1.2. For example, a graph-based conceptualization of indoor spaces is not provided, and the space partitioning in IFC follows the architectural layout and hence does not account for complementary decomposition criteria such as the mode of locomotion, sensor characteristics, or logical aspects. The meaning of entity relationships is not always appropriate for navigation purposes. For instance, *IfcRelConnects* relationships facilitate to denote that two wall entities are connected. However, this connectivity information obviously does not imply the possibility of bodily movement between both entities. The representation of the navigation context including user-contextual information and constraints as well as environmental navigation constraints is also not foreseen in IFC.

The thematic differentiation of indoor space provided by IFC-based BIM models is nevertheless already suitable for the symbolic naming and addressing of places as well as the generation of route descriptions and instructions in navigation tasks. The geometric description of spatial entities adds precise quantification to the indoor space and enables visual guidance. Although building elements are mostly represented as solid shapes, the derivation of consistent 2-dimensional building plans is still essential in the AEC/FM industry and hence supported by BIM models. Therefore, BIM models are to be seen as a rich source of information to populate and feed the presented approaches to the modelling of indoor space for indoor navigation, even if they are defined in a 2-dimensional setting. However, the relation to building models or the automatic derivation of spatio-semantic facts from building models is often neither discussed nor investigated in many current proposals.

2.3.2 Urban Information Models (UIM)

Urban Information Models (UIM) transfer the basic idea of BIM to the urban scale, and aim at establishing a shared digital knowledge resource providing geospatial information about natural and man-made real world objects in the urban context being relevant to decision making processes, sophisticated analyses, and display tasks in the field of geoinformatics and a variety of application domains such as environmental simulations, urban planning and data mining, city life-cycle and facility management, energy demand assessment, disaster management, and homeland security. Key components of UIM models include digital surface and terrain models, the built environment (e.g., buildings and sites, transportation infrastructures, underground facilities and utilities, artificial landscapes), and the natural environment and phenomena (e.g. water bodies, vegetation, weather). Due to the consideration of transportation infrastructures at a large scale, UIM models already successfully contribute to the field of outdoor navigation (e.g., as information source for car and pedestrian navigation systems).

The urban entities are described with respect to their spatial, semantic, and functional aspects as well as their ontological structure which adheres to taxonomies, logical decomposition hierarchies, and relationships that are given or can be observed in the real world (Kolbe 2009). The process of urban information modelling means the gathering, management, and continuation of this geospatial information at the urban level based on manual or (semi-)automatic processes. This involves manual design as well as methods for the observation and measurement of topographic features from fields such as photogrammetry, remote sensing, and engineering surveying. UIM models hence represent the urban context *as observed* (Nagel et al. 2009). The spatial representation of the shape and location of the urban artefacts is usually translated into a world coordinate reference system to reflect the real position of an object on the Earth’s surface. Since observation methods mostly capture the geometry and radiometry of urban features, a challenge to the acquisition of UIM models is the semantic qualification of the registered data in subsequent steps which requires techniques for object recognition, interpretation, and refinement, and often refers to existing geospatial knowledge (e.g., cadastral information systems) (cf., Baltsavias 2004, Brenner 2005, Kolbe et al. 2009).

A subfamily of UIM models are semantic 3D city models. In contrast to traditional 2-dimensional geo-data models being used in the GIS domain, they represent the spatial aspects of topographic entities in three dimensions and account for the complex semantic structures of the urban fabric. Similar to BIM, semantic 3D city models hence enable applications which rely on 3-dimensional, semantically rich urban information. 3D city models have established themselves as digital counterpart of real urban spaces and are constantly being acquired and set up worldwide, supporting a general trend towards modelling not only (parts of) cities but also their regional or country-wide context. For example, in the Netherlands, a national 3-dimensional information model for the country-wide large scale representation of buildings, roads, water, land use, land cover, bridges, and tunnels etc. has been implemented (van den Brink et al. 2012, Stoter et al. 2010). The European INSPIRE[8] initiative even aims at creating a continent-wide spatial data infrastructure for sharing environmental geospatial information about elevation, land cover, land use, transportation networks, cadastral information, buildings, etc. in two up to three dimensions. In literature, additional dimensions for UIM models such as time and scale (van Oosterom & Stoter 2010) as well as an n-dimensional model space analogous to BIM models are being discussed (Hamilton et al. 2005).

From 2002 onwards, more than 70 international members from industry, academia, and public administration founded and joined the Special Interest Group 3D (SIG 3D)[9] of the Spatial Data Infrastructure Germany (GDI-DE) in order to achieve a common understanding and definition of the urban entities, properties, and relationships relevant to the multitude of applications across the different disciplines in the UIM domain. This work resulted in the *City Geography Markup Language* (CityGML), an application-independent open information model and XML-based data format enabling the representation, storage, and sharing of semantic 3D city and landscape models. Since 2004, CityGML is being jointly developed by the SIG 3D and the Open Geospatial Consortium (OGC)[10], an international non-profit standards organization of more than 460 companies, government agencies, research organizations, and universities collaborating in the consensus-based development of publicly available and vendor-neutral geospatial standards. CityGML was adopted by the OGC membership as international OGC standard in 2008, and was released in its current version 2.0 in 2012 (Gröger et al. 2012, Gröger & Plümer 2012a).[11]

CityGML is designed as a universal topographic information model describing the semantics, geometry, topology, and appearance of the most important urban and regional entities required for a broad range of UIM applications. The conceptual model of CityGML hereby employs the ISO/TC 211 19100 standards family for the conceptual modelling of *geographic features*. According to ISO 19109:2005 and the General Feature Model (GFM) specified therein, geographic features are abstractions of real world phenomena which are associated with a location relative to the Earth's surface and which may have an arbitrary number of spatial and non-spatial properties. The GFM is a metamodel for the definition of conceptual feature models from a universe of discourse, and applies object-oriented modelling principles in order to express taxonomies, aggregation hierarchies, and associations between features. The standard conceptual modelling language proposed by the GFM is UML. The value domain of spatial properties is defined by the ISO Spatial Schema (ISO 19107:2003) which facilitates the geometric-topological representation of features in up to three dimensions. The data exchange format of CityGML results from the rule-based mapping of its conceptual model onto an application schema for the Geography Markup Language (GML version 3.1.1, Cox et al. 2012). GML provides an ISO 19118:2011 compliant XML encoding for the transport and storage of geographic information that adheres to the conceptual modelling framework of the ISO 19100 series, and is itself a joint development of ISO and OGC (ISO 19136:2007). A more detailed presentation of the mentioned ISO standards is given in chapter 4.3.

CityGML is conceptually partitioned into *modules* whose architectural overview is given in figure 31. The vertical modules are dedicated to different thematic topics in semantic 3D city and landscape models, and thus define separate feature models for the representation of sites (i.e. buildings, tunnels, bridges), relief (i.e. digital terrain models), transportation, vegetation, land use and cover, water bodies, etc. The horizontal modules provide basic content and model structures common to all thematic modules such as the modelling of appearance information or extensions. A salient aspect of CityGML is its support for the multi-scale representation of urban features in five

[8] See http://inspire.jrc.ec.europa.eu/ for more information.

[9] See http://www.sig3d.org/ for more information.

[10] See http://www.opengeospatial.org/ for more information.

[11] The author is actively involved in the CityGML development within OGC and the SIG 3D since 2007. Since 2008, the author is co-editor of the CityGML specification document. Moreover, since 2009, the author is vice chair of the CityGML Standards Working Group (SWG) in OGC which is in charge of future revisions of the standard.

consecutive levels of detail (LOD), which facilitates an increasing semantic differentiation and geometric granularity of the feature representation from the coarsest LOD0 to the finest LOD4. The different LODs account for varying resolutions in existing data acquisition processes and, for example, allow for the efficient processing and visualization of features in different scales appropriate to an application in question (Kolbe 2009).

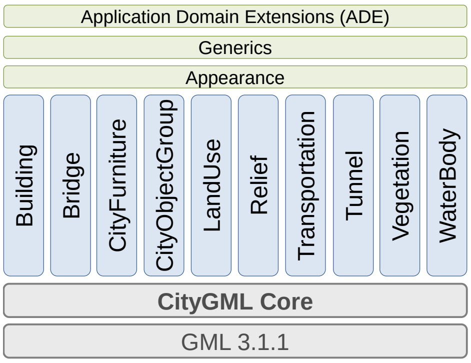


Figure 31: Modularization of the CityGML data model.

The modelling of buildings and facilities is provided through the *Building* module of CityGML. Semantic concepts as well as geometric representations of interior built structures and indoor spaces are only available in LOD4 of the conceptual building model, and hence this LOD implicitly underlies the following discussion. An excerpt of the building model in UML is shown below. The complete normative specification is given in (Gröger et al. 2012).

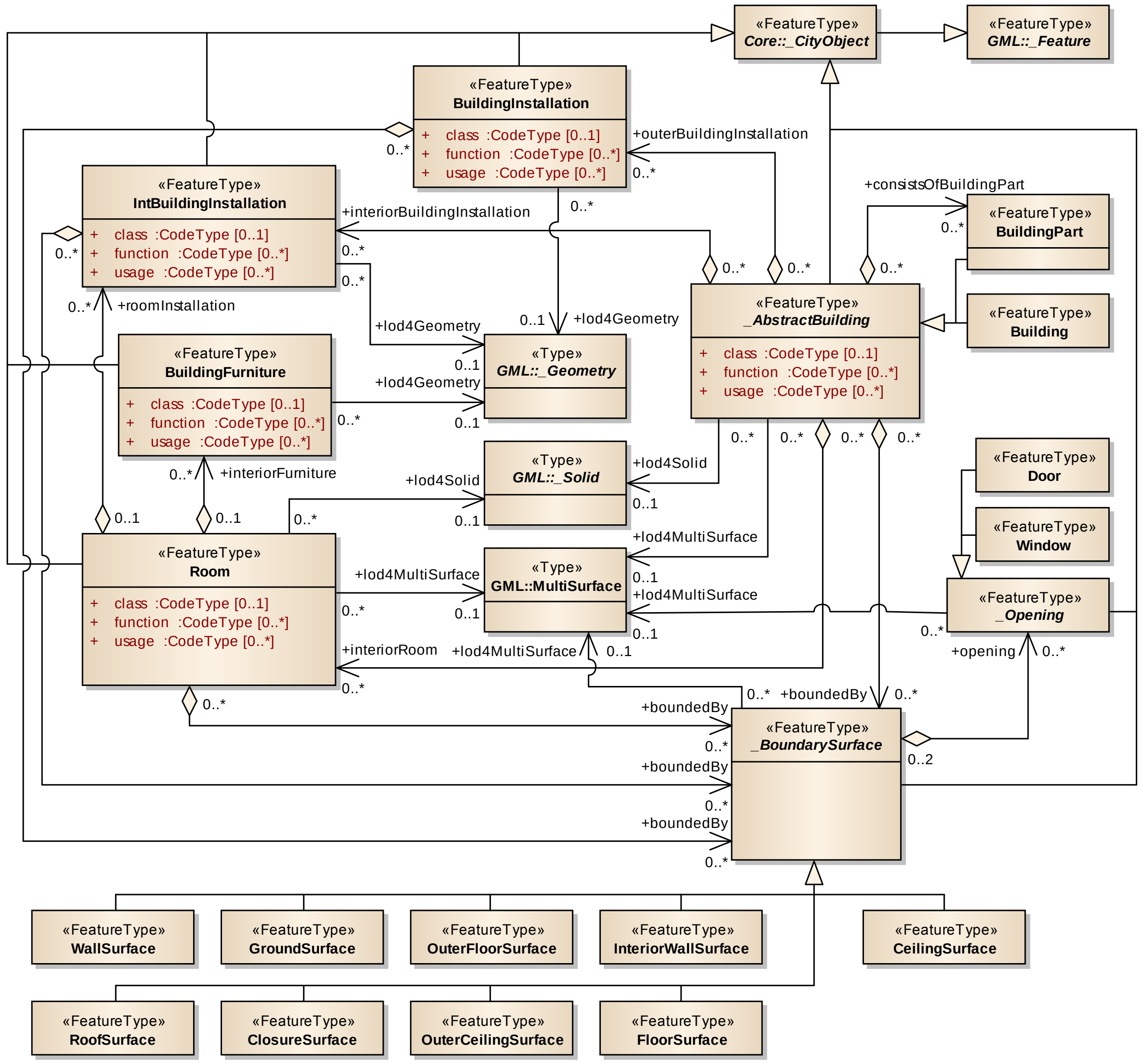

«FeatureType» Core::_CityObject
«FeatureType» GML::_Feature
«FeatureType» BuildingInstallation
+ class :CodeType [0..1]
+ function :CodeType [0..*]
+ usage :CodeType [0..*]
+outerBuildingInstallation
«FeatureType» IntBuildingInstallation
+ class :CodeType [0..1]
+ function :CodeType [0..*]
+ usage :CodeType [0..*]
+interiorBuildingInstallation
+consistsOfBuildingPart
«FeatureType» BuildingPart
«FeatureType» Building
«FeatureType» _AbstractBuilding
+ class :CodeType [0..1]
+ function :CodeType [0..*]
+ usage :CodeType [0..*]
+roomInstallation
«FeatureType» BuildingFurniture
+ class :CodeType [0..1]
+ function :CodeType [0..*]
+ usage :CodeType [0..*]
+lod4Geometry
«Type» GML::_Geometry
+interiorFurniture
+lod4Solid
«Type» GML::_Solid
«FeatureType» Door
«FeatureType» Window
«FeatureType» _Opening
+lod4MultiSurface
«Type» GML::MultiSurface
«FeatureType» Room
+ class :CodeType [0..1]
+ function :CodeType [0..*]
+ usage :CodeType [0..*]
+interiorRoom
+opening
+boundedBy
«FeatureType» _BoundarySurface
0..1
0..*
0..2
«FeatureType» WallSurface
«FeatureType» GroundSurface
«FeatureType» OuterFloorSurface
«FeatureType» InteriorWallSurface
«FeatureType» CeilingSurface
«FeatureType» RoofSurface
«FeatureType» ClosureSurface
«FeatureType» OuterCeilingSurface
«FeatureType» FloorSurface

Figure 32: Excerpt of the CityGML Building model (Gröger et al. 2012).

The central concept of the building model is *_AbstractBuilding* whose two subclasses are used to represent an entire building (*Building*) as well as its structural building parts (*BuildingPart*) that differ from the building in, for example, their number of floors or their roof type (e.g., wings of a building). Since *_AbstractBuilding* may have several building parts, recursive aggregation hierarchies of arbitrary depth are possible. The indoor space of an *_AbstractBuilding* is decomposed into non-overlapping free spaces (*Room*) which are defined as bounded volumes having a specific function (e.g., rooms, corridors). The boundary surfaces enclosing the *Room* volume are semantically classified through the abstract *BoundarySurface* concept which differentiates between built boundaries (*CeilingSurface*, *InteriorWallSurface*, *FloorSurface*) and virtual ones (*ClosureSurface*). The same concept also identifies the boundary surfaces of the outer building shell but in this case is restricted to the use of *RoofSurface*, *WallSurface*, *GroundSurface*, *OuterCeilingSurface*, *OuterFloorSurface*, and *ClosureSurface*. Openings in boundary surface are explicitly represented through the *Window* and *Door* feature types. A *Room* may have interior building installations (*IntBuildingInstallation*) which comprise, for example, beams, columns, and further fixed structural elements inside the free space, whereas movable objects such as furniture are captured by *BuildingFurniture*. Installations affecting the outer shell of a building (e.g., dormers, balconies) are modelled as *BuildingInstallation*, and both outer and interior installations may be further decomposed into boundary surfaces.

All semantic entities are transitively derived from *_CityObject* which is the top-level concept of the urban space taxonomy defined in CityGML. *_CityObject* itself is derived from the abstract GML superclass *_Feature* which implements the notion of a geographic feature in the sense of ISO 19109:2005 and hence classifies each city object as such. The 3-dimensional spatial representation of city objects is facilitated through a subset of the GML geometry model in accordance with the ISO 19107:2003 standard. City objects are hereby described by their observable surface geometries which are composed to volumetric shapes applying an accumulative modelling principle. CityGML is thus said to be a *surface-based model* (cf. Nagel et al. 2009). Topological relationships between object geometries are limited to adjacency properties which are modelled by reusing the common surface geometries along which two city objects touch. Each city object can be assigned a unique identifier and provides thematic attributes which facilitate the modelling of symbolic information such as the name and description as well as the *class*, *function*, and *usage* of the object. Observable properties of surfaces that can be registered by specific sensors like RGB or infrared cameras can be represented as appearance information in addition to semantics and spatial properties. Appearances include texture images and material definitions (e.g., colours) but are not restricted to visual data.

CityGML can be classified as hybrid space model, and the information about the interior built environment captured by the building model of CityGML obviously meets most of the conceptual requirements for indoor space models in the context of indoor navigation as discussed in chapter 2.2. The free space partitioning into *Rooms* conforms to cell-based hybrid approaches, and the surface-based representation of building entities agrees with the ontological structuring of indoor space along *bona-fide* and *fiat boundaries* as proposed by (Bittner 2001) and similar approaches (e.g.,Meijers et al. 2005, Slingsby & Raper 2008). Moreover, the space representation aligns with the human cognition of indoor space as embodied through the CONTAINER and SURFACE image schemas, and explicitly allows for modelling of architectural entities (e.g., rooms, corridors, doors, windows, stairs, etc.) as well as fixed and movable obstacles. However, and as stated above, CityGML is not specifically tailored to the purpose of indoor navigation and thus also neglects important navigational concepts. For example, the space representation follows from observations of built entities and hence does not include complementary space decompositions reflecting the mode of locomotion of the navigation user or alternative sensor spaces. Moreover, there is no explicit support for the qualitative modelling of connectedness and spatial containment relationships. Connectedness information between free spaces can only be implicitly deduced from the semantic associations relating rooms, boundary surfaces, and openings, or alternatively from their spatial configuration. In contrast to IFC, there is no concept for the representation of building floors, and spatial containment hierarchies can at most be expressed through CityGML’s notion of *CityObjectGroups* (cf. figure 31). A *CityObjectGroup* facilitates the grouping of city objects according to arbitrary spatial and non-spatial criteria, and may have its own defined spatial representation. Therefore it is suitable for denoting, for example, indoor spaces spanning several rooms and is in fact proposed by the CityGML specification for the modelling of floors (Gröger et al. 2012). Finally, CityGML does not include a graph-based conceptualization of indoor space, and lacks the consideration of user-contextual information and navigation constraints.

Nevertheless, and similar to BIM models, the spatio-semantic description of CityGML buildings may serve as input for symbolic, semantic, and spatial aspects of the presented indoor navigation models. CityGML-based UIM models thus contribute as rich information source to the field of indoor navigation. Due to the large scale representation of the urban and regional context and the use of a world coordinate reference system, they even provide the potential for a seamless combination of outdoor and indoor spatial information in navigation tasks (see also Lee & Zlatanova 2008).

2.3.3 ESRI Building Interior Space Data Model (BISDM)

The ESRI Building Interior Space Data Model (BISDM)[12] is a conceptual data model for the representation of 2-dimensional building *floor plans* which aims at supporting the planning and management of indoor spaces in the field of GIS on top of the proprietary ESRI software suite. The data model is meant as basic structure to support different perspectives on buildings such as architecture, construction, landscape-level planning, facility management, environmental management, and security/emergency preparedness. The intention of the BISDM development is hereby not to establish another international standard for building modelling but rather to provide a practical implementation for indoor space models which maps into existing standards such as CityGML or IFC wherever applicable. In contrast to floor plans given in terms of purely geometric objects (e.g., legacy CAD models), the BISDM defines a conceptual framework that translates the elements of a floor plan to semantic entities with thematic and spatial properties as well as relationships.

The BISDM is jointly developed by the company ESRI and the BISDM Group which is an independent and open committee with members from both the private and the public sectors. The work on the BISDM was launched in 2007 and the latest version 3.0 of the conceptual data model was released in 2011. An excerpt of this conceptual data model is depicted in figure 33 as informal UML-based representation. Note that the BISDM is expressed in terms of ESRI objects and features, and thus does not employ the GIS standards from the ISO 19100 standards family.[13]

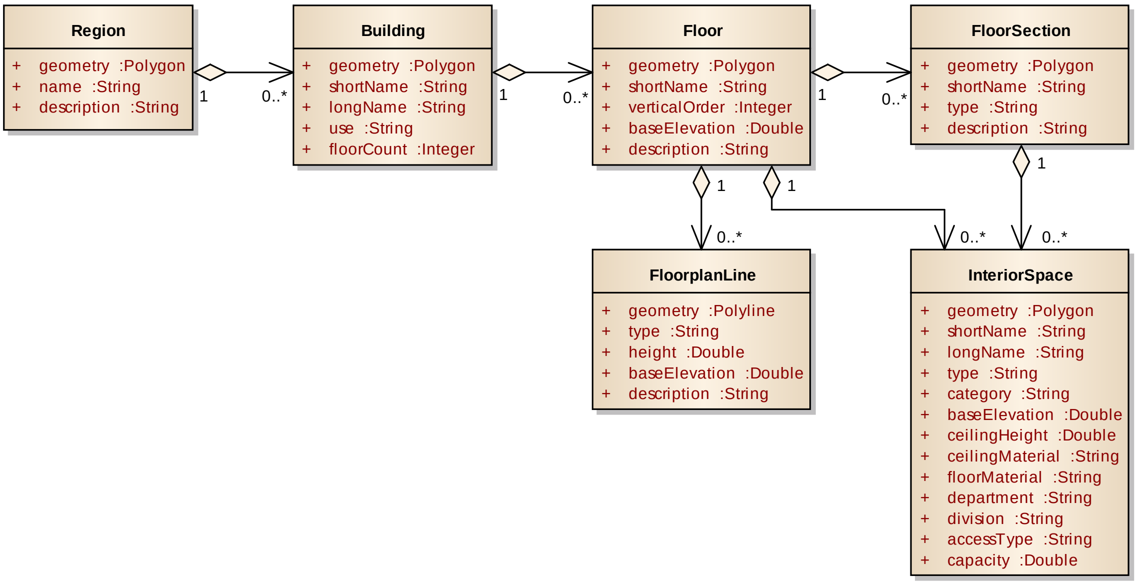


Figure 33: Informal UML diagram of the ESRI BISDM (excerpt).

The BISDM conceptually structures the interior of a *Building* along its contained *Floor* entities. Each floor is spatially described by a 2-dimensional polygon and may be assigned an elevation value through the property *baseElevation* in order to reliably stack several floors on top of each other. In case the elevation is not known, the *verticalOrder* attribute can be used instead. An instance of *Floor* can be decomposed into several sections (*FloorSection*) that represent logical or physical subdivisions such as wings, zones, or organizational units. A *FloorSection* is also represented as polygon which needs to be spatially contained within the *Floor* geometry. The *InteriorSpace* concept reflects the free spaces on a floor (or within a section) and therefore is used to model entities such as rooms, hallways, or staircases which are semantically differentiated through the properties *type* and *category*. Similar to *Floor*, every instance of *InteriorSpace* can be assigned a *baseElevation*. In addition, the most common

[12] See http://www.bisdm.org/.

[13] A discussion of the ESRI feature model and its differences to the ISO 19100 standards family is out of scope of this thesis.

height of the space can be given as *ceilingHeight* property. By this means, a 3-dimensional view can be retrieved by simple extrusion operations. The material of both the ceiling and the floor can be encoded for the space as character strings through *ceilingMaterial* and *floorMaterial*. Further semantic information includes *accessType* restrictions (e.g., to denote public or private spaces), the total number of occupants allowed in this space (*capacity*), or the *department* and company *division* the space is associated with. All building elements on a floor such as walls, columns, doors, or windows use the concept *FloorplanLine* to describe their boundaries as a collection of straight line segments. *FloorplanLine* hence captures the lines that compose the floor plan. The BISDM does not impose a specific modelling paradigm for the representation of element boundaries. A wall, for instance, can be given by those lines that result from a projection of its visible or observable boundary surfaces. Alternatively, the boundary lines may completely enclose the footprint of the physical wall element or abstract from it using a centreline representation. The *type* property of *FloorplanLine* is used to classify the boundary line. Moreover, every *FloorplanLine* can be assigned a *baseElevation* and a *height* value. Note that the boundary of a structural building element can be expressed by more than one *FloorplanLine* instance. A conceptual entity for representing the building element itself is however not provided.

The proposed subdivision of the interior built environment along architectural entities captured by *InteriorSpace* obviously renders a partitioning of the free space into mutually non-overlapping cells with the *FloorplanLine* objects representing both their *bona-fida* and *fiat boundaries*. Spatial containment relationships can be expressed which may also denote logical groupings of interior spaces. The BISDM can hence be classified as cell-based hybrid indoor space model whose spatial and semantic expressiveness largely satisfies the information needs of most of the navigation space models from chapter 2.2. However, and similar to both IFC and CityGML, complementary decompositions of the interior environment which are not aligned with the architectural layout of the building are not foreseen.

Starting from version 3.0, the BISDM includes a graph-based conceptualization of indoor space and additionally offers indoor routing capabilities based on the ESRI software suite. Precisely, corridors and hallways can be abstracted by their centrelines (called *FloorLines*) which are combined to form a geometric network on each floor. The *FloorLines* are meant to connect the doors between different *InteriorSpace* objects such as rooms. However, the interior spaces themselves are not represented in the resulting geometric network. Thus, the network structure only allows for routing people or objects along the corridors from doors to doors but not within rooms, which lags behind the expressivity of the indoor space models discussed in chapter 2.2. Besides *FloorLines*, the BISDM network model defines *FloorTransitions* which connect *FloorLines* on different floors in order to enable multi-level paths. The nested hierarchical structure of the interior environment is however not encoded in the navigation graph and thus not available for path planning or route guidance. The graph elements can be enriched with simple navigation constraints. For example, *FloorLines* can be marked with Boolean flags to encode whether or not they can be navigated by predefined user groups (e.g., wheelchair users). Likewise, restricted turns involving one up to three edges as well as time-based costs for individual edges can be modelled.

The following figure 34 illustrates the BISDM navigation graph for an example indoor setting (taken from http://proceedings.esri.com/library/userconf/feduc11/papers/tech/bisdm_ver3_final_feduc_2011.pdf).

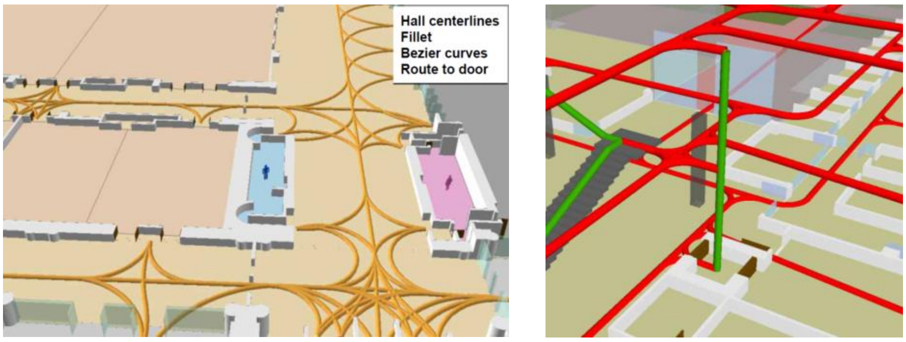


Figure 34: BISDM navigation graph based on FloorLines (yellow edges on the left) and FloorTransitions (green edges on the right).

2.3.4 Geometric and Graphical Building Models

Besides IFC, CityGML, and ESRI BISDM, also other standards and formats are being used for the representation of buildings and their interior built environment. With legacy 2-*dimensional* or 3-*dimensional CAD models* from the field of building design, architecture, and construction, an important group of geometric floor plan and building models has already been discussed in the context of BIM in chapter 2.3.1. Often, building models are also provided in *computer graphics formats* whose primary focus is on visualization and rendering. Examples are standardized formats such as X3D (ISO/IEC 19775-1:2008), VRML (ISO/IEC 14772-1:1997 and ISO/IEC 14772-2:2004) or COLLADA (ISO/PAS 17506:2012, Barnes & Finch 2008) but also include proprietary formats like 3D PDF or 3D Studio Max. A third group is given by *geovizualization standards and formats* such as OGC KML (Wilson 2008) which likewise aim at the visualization of (geographic) objects including buildings.

Indoor space models represented according to one of these standards or formats can be classified as purely geometric space models and thus lack the semantic and symbolic properties and structures which are required in the context of indoor navigation (cf. chapter 2.1.4). Moreover, visualization formats target at and are optimized for an efficient encoding of the geometric and graphical aspects of the visible objects within the interior built environment. In contrast to geometric space models for indoor navigation as discussed in chapter 2.1.2, they hence neglect the explicit representation of the spatial location and extent of architectural entities such as rooms, corridors, walls, and obstacles but merely describe the geometry and appearance of visible surfaces of the built structure (e.g., wall, floor, and ceiling surfaces that are visible from inside a room). The semantic qualification of visualization models thus requires techniques from object recognition and interpretation, and is often impeded by geometric-topological inconsistencies (e.g. overlaps, self-intersection, overshoots, undershoots) as well as inaccurate and incomplete geometric descriptions (cf. Nagel et al. 2009). Although geometric CAD models naturally represent the geometry of architectural entities, they likewise are subject to semantic interpretation in order to serve as input for indoor navigation. Due to these reasons, a more detailed description of the standards and formats for geometric and graphical building models is beyond the scope of this thesis.

2.4 Requirements for Indoor Navigation

Following on from the presentation of challenges to indoor navigation in chapter 1.2 and the subsequent survey of existing approaches to indoor space modelling in the previous sections, this section elaborates on general requirements for indoor space models for indoor navigation. Obvious and commonly agreed needs such as rich hybrid information about the indoor space (e.g., quantitative, qualitative, and symbolic information) are not listed separately.

Notion of indoor space. Most approaches to indoor space modelling for the purpose of indoor navigation presented in chapter 2.2 built upon a notion of indoor space that only addresses the *topographic space* and thus the description of the interior built environment, its fixed physical structure and the spatial entities therein (Becker et al. 2009a, Worboys 2011). The indoor space models being used range from layered 2-dimensional building floor plans to 3-dimensional models of the interior built-up space with differing richness and expressivity in their geometric, topological, semantic, and symbolic space representations. The subdivision of the topographic space is commonly aligned with the structural building elements and architectural constraints in order to derive physical regions of free space facilitating path planning and symbolic addressing.

Often, the sensor characteristics of a given indoor positioning technology such as Wi-Fi, RFID, or Bluetooth are mixed within the topographic space representation (e.g., as attributes on topographic spaces or as decomposition criterion for topographic spaces) to enable the localization of persons or objects. Although both modelling aspects are essential to indoor navigation as they address two main navigation tasks, their mixed representation within a single topographic space model faces severe disadvantages (Nagel et al. 2010). First, changes to the building layout (e.g., closing a door permanently or subdividing a room into two parts through a new wall) obviously affect path searches but not necessarily have an impact on the deployment or the sensor characteristics of an available localization infrastructure. Likewise, changes to the localization infrastructure do not influence the building topology. However, such changes cannot be captured separately but rather affect the entire space model in case topographic and sensor-specific aspects are combined and aligned in a single representation. Second, the resulting indoor space models typically lack the flexibility that would allow a context-dependent ad-hoc selection of different localization technologies. A large number of the presented topographic space models even completely neglects the additional representation of sensor characteristics and hence fails to answer substantial challenges to indoor navigation as identified in chapter 1.2.

If the topographic space is understood as the built-up space constituted by the architectural entities, then the structuring of indoor space along the sensor characteristics of a localization infrastructure consequently renders a complementary *sensor space*. Thus, topographic space and sensor space are to be seen as alternative notions of indoor space which result in substantially different and independent space representations. In fact, every localization technology installed in an interior environment, and even each individual sensor or transmitter of such an infrastructure defines a separate sensor space. The decoupled modelling of topographic space and possibly multiple sensor spaces is a prerequisite for overcoming the issues of their mixed representation, and may serve as basis for the support for different and multiple localizations technologies in an indoor navigation system. In their ontological classification of indoor space, (Bhatt et al. 2009) also name topographic object spaces and complementary sensor range spaces as distinct spatial concepts, and the approach of (Jensen et al. 2009) comes closest to the idea of their separate modelling.

Both topographic and sensor spaces are subject to physical observations and measurements. An indoor space partitioning however may also result from additional thematic or logical criteria such as environmental contextual information or navigation constraints. Examples such as security zones, disaster areas, or fire detector zones are rather to be seen as virtual entities which can be assigned a spatial extent but need not follow the architectural layout or the partitioning schema of topographic space (cf. chapter 1.2). The spatial configurations of virtual entities following from logical space decompositions hence determine separate *logical spaces* which again are complementary to the decomposition of indoor space along physical criteria (Becker et al. 2009b). However, if environmental contextual information is considered at all, none of the presented approaches explicitly represents logical spaces as such, but rather maps them onto properties of physical places or their equivalences in graph-based conceptualizations. This modelling approach generally faces the same weakness as the mixed representation of topographic and sensor space since logical space information and topographic space entities are enforced to be strictly coupled and aligned. For example, if a security zone only covers parts of a room, the room topography is commonly partitioned into correspondingly many subdivisions in order to attributively represent their access restrictions instead of representing both the room and the security zone as spatial entities in their own right. It obviously follows that any changes to the building layout or the complementary logical space partitioning may possible require a restructuring of the entire model.

While path planning requires a model of the topographic indoor space as well as additional thematic or logical decompositions in order to obtain navigable route sections, localization methods require corresponding sensor space representations. An indoor space model for the purpose of indoor navigation hence must support alternative space representations for complementary notions of indoor space as well as their common evaluation in different navigation tasks.

Multi-granular representation of indoor space. Besides alternative notions of indoor space, the space representation has to account for navigable and non-navigable areas within physical places and architectural units at an appropriate level of granularity in order to derive navigable route sections for different types of users and alternative modes of locomotion (e.g., walking, driving, or flying). Grid-based approaches naturally apply a fine-grained but rigid cell decomposition of the topographic space and (Yuan & Schneider 2011) demonstrate how to geometrically evaluate the navigability of grid cells for arbitrarily shaped users in path queries. Cell-based and conceptual-based hybrid approaches also recognize the need for subspacing the free topographic spaces into smaller regions at least in case of large halls or long corridors. In most cases, the subspacing follows purely spatial considerations though (e.g., medial axis transforms or straight lines of sight) and is commonly proposed in order to map complex-shaped architectural entities onto several elements in a navigation graph. The resulting graph structures already facilitate the planning of more fine-grained paths and space trajectories as well as the generation of more precise route descriptions.

However, complementary decompositions of indoor space for pedestrians, wheelchair users, driving or flying autonomous robots, or along additional user-centric or environmental criteria result in multiple and different configurations of navigable and non-navigable spaces. A single and fixed granularity of the space representation together with attributes for topographic spaces denoting their navigability as proposed by most research works is hence not sufficient to account for the various configurations. In fact, none of the presented indoor space models supports multiple and complementary subdivisions of topographic space according to arbitrary criteria. Moreover, range queries as well as multi-level path queries and route descriptions rely on multiple levels of detail that additionally capture the hierarchical structuring of the interior built environment into, for example, floors and wings. The fine-grained representation of navigable and non-navigable spaces apparently needs to be related to spatial aggregates on coarser levels of the hierarchy along containment and overlap relationships. A large number of the presented grid-based and cell-based approaches however restrict their space decompositions to non-overlapping cells and thus are not capable of representing spatial hierarchies.

The multi-granular representation of indoor space is to be seen as an essential requirement for indoor navigation, and renders a prerequisite for the challenge of context-dependent path planning. The identification of navigable spaces is hereby supported by rich spatial and semantic information about the interior environment including the explicit modelling of navigation obstacles. In order to detect low passages or obstacles suspended from ceilings both of which may obstruct movement in specific contexts (e.g., in case of flying objects) their accurate 3-dimensional geometric description needs to be available.

Mathematically sound graph-based conceptualization of indoor space. A graph-based conceptualization reflects the organization of indoor space through a combinatorial model which can be fed to a range of existing algorithms for answering navigation queries such as path queries, range queries, or nearest neighbour queries. The derivation of graph structures from the indoor space representation hence renders a core requirement in the context of indoor navigation, and corresponding mappings are proposed by almost all modelling approaches.

The navigation graph primarily has to encode the possibility to move between topographic or logical (sub)spaces in order to enable path searches. The majority of approaches apply place graphs for this purpose. Although the relation between the physical places and the nodes in the place graph as well as the translation of connectedness relationships onto edges is intuitively understood in most cases, the mappings from 2-dimensional or 3-dimensional models of the interior environment to graph-based conceptualizations are rarely defined in a formal mathematical framework but often remain on a descriptive level. A single place is usually associated with a single graph node but this simple rule is already violated in case of complex corridors which are typically mapped onto several graph elements without providing corresponding subspaces (cf. previous requirement). Actually, most approaches already lack a formal definition of what places and their connectedness mean in a mathematical sense, and rather depend on a priori semantic and functional knowledge about the spatial entities. This often impedes a (semi-)automatic and deterministic derivation and verification of place graphs, and the same interior built environment may even be captured by different graph structures for the same approach in case a sound functional dependency between the representation of the building topography and the corresponding place graph is missing (Nagel et al. 2010). Similar issues can be observed for approaches employing visibility criteria instead of connectedness to build their navigation graph, and in the context of mapping hierarchical space structures onto an appropriate graph-based representation. The latter is often defined decoupled from a place graph of the indoor space although it encodes the same spatial entities at least on one level of its hierarchy.

The most formally developed methods for the derivation of place graphs are presented in the dual-graph-based works of (Lee 2001) and (Boguslawski & Gold 2009). Both proposals though lack a sound framework for the hierarchical graph-based conceptualization of indoor space, for which a formal definition is given by (Stoffel et al. 2007). However, none of the presented approaches supports a mathematical sound derivation of an integrated graph structure that maps different and multiple space concepts and decompositions as postulated in the above requirements.

Navigation context. Limited work has been undertaken in literature to formalize a conceptual model for user or environmental contextual information and navigation constraints. As mentioned above, the environmental context is mostly proposed to be mapped as attributes of corresponding topographic spaces or their graph representation which is disadvantageous in case the contextual information is not aligned with the physical places. Moreover, complex constraints involving more than one spatial entity as well as combined constraints applying more than one constraint to a single spatial entity are rarely discussed. Examples for complex constraints are prohibited maneuvers which play an important role in outdoor car navigation systems as they declare an ordered sequence of connected road and junction elements to be non-navigable, which is also relevant to indoor navigation applications. Combined constraints, on the other hand, are required to express the combination of different types of constraints such as temporal constraints and access restrictions on architectural units. For example, a combined constraint on an office room could denote that the office is only accessible between 9am and 5pm and may only be entered by authorized personnel. Most approaches acknowledging the need for contextual information focus on the reasoning about this knowledge (e.g., Anagnostopoulos et al. 2005, Stoffel et al. 2007) but do not address its formal presentation or the ontological structuring and classification of navigation constraints. Moreover, dynamic aspects and changes of the navigation context are neglected in most research works. For example, areas monitored by fire detectors may temporarily affect the navigable space. In case of a fire incident, the monitored area (which is rather a logical than a topographic space) has to be avoided in planning escape routes for normal users, and additional navigation constraints such as security zones may need to be temporarily overruled in order to find the paths to the closest exits. In contrast, users with a different context such as fire fighter forces need to be directly navigated to the fire spot and corresponding paths may include obstacles such as closed doors or walls which can be easily torn down.

Supporting the navigation context mainly requires an indoor space model to facilitate the ad-hoc selection and simultaneous consideration of the navigable and non-navigable topographic spaces matching the mode of locomotion of the user, the logical spaces representing environmental contextual information affecting or constraining the movement of the user, the sensor spaces of the localization technologies supported both by the environment and the mobile end-user device, as well as additional spatial or thematic user constraints and preferences. However, in most approaches the navigation context is restricted to one or a limited subset of possible configurations in order to match a given use case and navigation scenario. By this means, the combinatorial complexity is a priori reduced to a single and rigid navigation setting. Localization and tracking methods as well as indoor space models and route planning strategies are then tailored to that specific navigation context. Although the approaches are naturally well suited for their specific setting, they lack the flexibility and methods to support additional configurations such as further modes of locomotion, additional localization technologies, or multiple navigation constraints.

The minimum requirements for an indoor navigation systems and its underlying model of indoor space are summarized in the following list:

- Support for different and multiple localization methods and infrastructures based on arbitrary indoor sensor technologies;
- Support for different and multiple representations of navigable and non-navigable spaces according to various modes of locomotion and navigation contexts;
- Support for different and multiple representations of logical and thematic spaces that are not necessarily aligned with the physical built structure or subject to physical observations;
- Support for hierarchical and nested structures of space;
- Support for the representation and evaluation of both environmental and user-contextual navigation constraints;
- Support for the ad-hoc selection and combination of available localization technologies supported by the mobile end-user device and appropriate representations of (non-)navigable and logical spaces matching the individual navigation context of a moving person or object; and
- Mathematical sound framework defining the fundamental notion of space and its deterministic graph-based conceptualization.

It can be concluded from the survey of existing approaches to indoor navigation that although the postulated challenges and requirements are often addressed and partially answered there is no one model of indoor space which fully facilitates localization and path planning in multiple navigation contexts.

2.5 Mathematical Background

A sound mathematical framework has been identified as an essential requirement to indoor space modelling for indoor navigation in the previous section and also renders a main goal of this thesis. The work undertaken in this thesis mainly draws from mathematical fields such as topology, manifold theory, and graph theory. In the following sections, basic notions and concepts from these fields are recapped and important terms, symbols, and definitions are fixed. The definitions and results presented here are mostly adapted from (Munkres 1984), (Hatcher 2008), (Lee 2011), (Jänich 2012), and (Morris 2012). Since definitions and terminology differ in literature, a consistent and comprehensive presentation of the applied theory is given in appendix A. The chapters 2.5.1.1 to 2.5.1.3 provide a brief summary overview of this appendix.

2.5.1 Basic Notions and Concepts from Topology

Topology is the branch of mathematics studying the characteristics of geometric figures that remain invariant under a given class of elastic deformations such as bending, stretching, twisting, and shrinking but not cutting or tearing apart. Whereas geometry mostly deals with the shape and metric properties of figures (e.g., length, distance, angle) and only considers two figures to be equivalent if they can be moved into each other by a rigid motion, topology can rather be viewed as *qualitative* geometry which considers two figures to be equivalent if they can be transformed into one another by continuous deformations (hence, topology is also called "rubber sheet geometry" or "the mathematics of continuity"). The French mathematician *Henri Poincaré* (1854 – 1912) who fundamentally contributed to the field of topology gave the following graphical and intuitive explanation:

> *"Imagine any sort of model and a copy of it done by an awkward artist: the proportions are altered, lines are drawn by a trembling hand and are subject to excessive deviation and go off in unexpected directions. From the point of view of metric or even projective geometry these figures are not equivalent, but they appear as such from the point of view of geometry of position [that is, topology]."* (Monastyrsky 2008, p. 95).

From the different branches of topology which have evolved over time and which are distinguished in literature, this thesis is mainly concerned with *point-set topology* and *algebraic topology*. Point-set topology is closely linked to set theory and deals with the definition of the abstract nature and characteristics of *topological spaces* in general (cf. chapter 2.5.1.1). It hence provides the basis for all other branches of topology. Algebraic topology associates algebraic structures with topological spaces in order to systematically study the properties of a narrower class of spaces called *topological manifolds* (cf. chapter 2.5.1.2). *Combinatorial topology* as subset of algebraic topology applies combinatorial methods to construct such spaces from simpler ones. The resulting combinatorial structure allows for reasoning about the invariant properties classifying the topological space itself and helps to accelerate their algorithmic computation (cf. chapter 2.5.1.3). One of the earliest theorems in algebraic topology is the *Poincaré duality theorem* on the duality of combinatorial structures on manifold spaces. Since this theorem is at the core of the mathematical formalization work in this thesis, it is presented in more detail in chapter 2.5.1.4.

Especially in the field of geographic information science, the practical use of topology is twofold (Zlatanova 2000). First, topology provides the tool to define the *intrinsic properties of spatial objects* (e.g., points, curves, surfaces, solids) that form the primitives for describing the spatial aspects of real-world objects and phenomena. Second, topology is used as formalism for categorizing the invariant *topological relationships* between two such objects (e.g., overlap, adjacency) which are induced by their spatial configuration and categorized based on the intersection of their interiors, boundaries, and exteriors as defined in point-set topology.[14] Mathematical models for classifying the topological relationships are the well-known 4-intersection model (4IM) as well as its extension to the (dimensionally extended) 9-intersection model (DE-9IM) (cf. appendix A.6).

In this thesis, both aspects play a role within the mathematical formalization of the spatio-semantic indoor space model. On the one hand, topological manifolds are used to model the partitioning of indoor space into smaller, well-behaved spaces reflecting, for example, the navigable free spaces and non-navigable obstacle spaces within the interior built environment or the coverage areas of sensors from a given localization technology. Based on methods from algebraic topology, a combinatorial structure, namely a CW complex, is imposed on the manifold spaces in order to describe them in a computationally efficient manner and to derive both a dual structure and a graph-based representation of indoor space based on the Poincaré duality. On the other hand, the notions of *connectedness* and *containment* that are relevant in navigation tasks and that occur between the spaces are precisely defined based on classification schemes for topological relationships.

2.5.1.1 Topological Spaces

In point-set topology, a *topological space* is defined as pair $(X, \mathcal{T})$ where X is a set and $\mathcal{T}$ is a family of subsets of X such that 1) X and the empty set $\emptyset$ belong to $\mathcal{T}$, 2) the union of any (finite or infinite) collection of sets in $\mathcal{T}$

[14] Also alternative methods such as Region Connection Calculus (RCC) have been proposed for the classification of spatial relationships, not least because the concept of a space as defined in point-set topology is seen to be generalized far beyond an intuitive meaning (e.g., Gotts et al. 1996). However, the set-theoretic definition of topological relationships is largely accepted in literature and implemented in GIS systems.

belongs to $\mathcal{T}$, and 3) the intersection of any finite collection of sets in $\mathcal{T}$ belongs to $\mathcal{T}$. $\mathcal{T}$ is called the *topology* on X. The elements of a topological space are said to be *points* regardless of their actual nature.

The members of $\mathcal{T}$ are called *open sets*. For every point $x \in X$, an open subset $O \in \mathcal{T}$ containing x is called an (*open*) *neighbourhood* of x. The complements to the open sets in X are said to be *closed sets*. A point $x \in X$ is said to be a *limit point* of a subset S of X, if every open neighbourhood of x contains a point of S different from x. A subset S of X is closed iff (if and only if) it contains all its limit points.

Given a subset A of X, the *closure* of A, denoted $\bar{A}$, is the smallest closed set containing A. The *interior* $Int(A)$ is the union of all open sets which lie entirely in A. The *exterior* $Ext(A) = X \setminus \bar{A}$ is the complement of the closure of A in X. The *boundary* $\partial A = \bar{A} \setminus Int(A)$ is the closure of A without the interior of A. It follows that the interior, exterior, and boundary are disjoint sets. A subset Y of X can be viewed as topological space in its own right. Precisely, if Y is a non-empty subset of a topological space $(X, \mathcal{T})$ then the topological space $(Y, \mathcal{T}_Y)$ is said to be a *subspace* of $(X, \mathcal{T})$ with the *subspace* (or *induced*) *topology* $\mathcal{T}_Y$ whose open sets are of the form $\mathcal{T}_Y = \{O \cap Y : O \in \mathcal{T}\}$.

The notion of *neighbourhood* is one of the most fundamental concepts in point-set topology and expresses a qualitative *closeness* relation between a point and a set. In order to measure the closeness between points or neighbourhoods in a quantitative way, a *metric* is required as additional structure on the topological space. A natural example is the *Euclidean metric* which is based on the standard inner product of a vector in the n-dimensional real vector space $\mathbb{R}^n$ and the derived *Euclidean distance* between any two vectors $x, y \in \mathbb{R}^n$ given as $d(x,y) = \|x - y\| = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$. The Euclidean metric induces a topology $\mathcal{T}_{\mathbb{E}}$ on $\mathbb{R}^n$ as consequence of defining a neighbourhood of a point $x \in \mathbb{R}^n$ to be a set of the form $B^n(x,r) = \{y \in \mathbb{R}^n \mid \|y - x\| < r\}$ with $r \in \mathbb{R}^+$. The set $B^n(x,r)$ is called an *open n-ball* centred at x. Every open set $O \in \mathcal{T}_{\mathbb{E}}$ can be generated as the union of open n-balls which makes the collection of open n-balls a *basis* $\mathfrak{B}$ for the topology $\mathcal{T}_{\mathbb{E}}$. Conversely, the set $\bar{B}^n(x,r) = \{y \in \mathbb{R}^n \mid \|y - x\| \leq r\}$ is said to be the *closed n-ball* about x which is closed in $\mathcal{T}_{\mathbb{E}}$. The topology $\mathcal{T}_{\mathbb{E}}$ induced by the Euclidean metric is called the *Euclidean* or *standard topology* on $\mathbb{R}^n$. The topological space $(\mathbb{R}^n, \mathcal{T}_{\mathbb{E}})$ is referred to as *n-dimensional Euclidean space* and is commonly denoted as just $\mathbb{R}^n$ with the Euclidean structure being understood.

Euclidean n-space $\mathbb{R}^n$ as well as subspaces thereof with the induced topology are *second-countable Hausdorff* topological spaces. The property of being second-countable requires both a countable basis $\mathfrak{B}$ and that any two distinct points in the space are separated by open neighbourhoods. A subset A of $\mathbb{R}^n$ is said to be *bounded* if it can be covered by a single open n-ball, i.e. $A \subseteq B^n(x,r)$ for $x \in X$ and $r > 0$. The boundedness property provides a measure of how far the set extends in space. In metric spaces, boundedness is implied by *compactness* which can be seen as the topological generalization of the notion of finiteness (Lee 2011). The *Heine-Borel theorem* states that a subspace $A \subset \mathbb{R}^n$ is compact iff it is closed and bounded. Open n-balls in $\mathbb{R}^n$ as well as $\mathbb{R}^n$ itself are examples for non-compact spaces. A topological space $(X, \mathcal{T})$ is said to be *disconnected* iff there are non-empty disjoint open subsets $A, B \subset X$ whose union represents X, and *connected* otherwise.

Two topological spaces $(X, \mathcal{T})$ and $(Y, \mathcal{T}_Y)$ are topologically *equivalent* if there exists a *homeomorphism* of one onto the other (written as $X \cong Y$). Homeomorphism is the most fundamental relation in topology and requires a function $f: X \rightarrow Y$ to be bijective and both f and f^{-1} to be continuous. Continuity is expressed in terms of open sets such that for each open set U in Y the inverse image $f^{-1}(U)$ has to be open in X, and thus $f^{-1}(U) \in \mathcal{T}$. A property of a topological space which is preserved under homeomorphisms is called *topological property* or *topological invariant*. Examples for invariant properties are compactness and neighbourhoods. A both injective and continuous map $f: X \rightarrow Y$ is called an *embedding* of the topological space $(X, \mathcal{T})$ into $(Y, \mathcal{T}_Y)$ if it induces a homeomorphism between X and the subspace $f(X) \subseteq Y$. Since the image $f(X)$ is contained in Y, the embedding $f: X \rightarrow Y$ allows X to be treated as subspace of Y with the topology induced by Y.

2.5.1.2 Topological Manifolds

An *n-dimensional topological manifold M*, or simply *n-manifold*, is a second-countable Hausdorff space in which each point has an open neighbourhood homeomorphic to either an open subset of $\mathbb{R}^n$ or to an open subset of the half-space $\mathbb{R}^n_+ = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_n \geq 0\}$, for some non-negative integer n. The prototypical example of an n-manifold is $\mathbb{R}^n$ itself with the standard topology. The set of points in M which have an open neighbourhood homeomorphic to an open n-ball is called the *interior* of M denoted by *Int*(M). The *boundary* ∂M of M is the set $\partial M = M \setminus Int(M)$. In case the set of boundary points is empty, M is said to be an n*-dimensional manifold without boundary*. The boundary of an n-manifold with boundary is exactly an $(n-1)$-manifold without boundary.

The *manifold interior* and *boundary* are not to be confused with the notion of *topological interior* and *boundary* as defined in general point-set topology (cf. previous section). The latter notion is only applicable if the manifold is viewed as subset of a larger ambient topological space but not necessarily coincides with its manifold interior and boundary. A manifold without boundary is called *closed* if it is compact, and *open* if it is non-compact. Again, both terms must not be confused with the set-theoretic notion of open and closed subsets of a topological space.

This thesis is concerned only with topological n-manifolds living in $\mathbb{R}^2$ respectively $\mathbb{R}^3$, with $0 \leq n \leq 3$, and with the induced standard topology. Examples are shown in figure 35. The circle (closed loop) is a 1-manifold and the open disk is a 2-manifold both of which have an empty boundary. The closed unit 3-ball $\overline{\mathbb{B}}^3$ about the origin is a 3-manifold with boundary whose boundary is the unit 2-dimensional sphere $\mathbb{S}^2$.

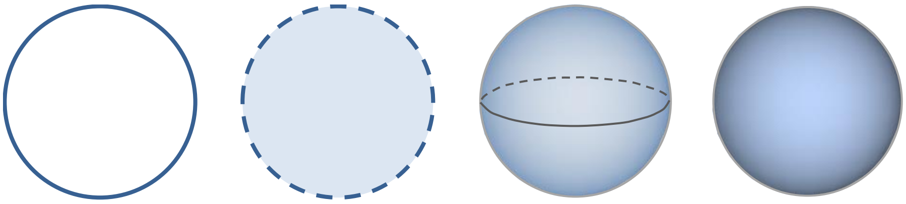


Figure 35: Examples of topological manifolds (from left to right: the 1*-sphere* $\mathbb{S}^1$*, the open* 2*-disk* $\mathbb{B}^2$*, the* 2*-sphere* $\mathbb{S}^2$*, and the* 3*-ball* $\overline{\mathbb{B}}^3$*).*

Counterexamples of non-manifold spaces are the union of the x-axis and the y-axis in $\mathbb{R}^2$, the conical surface in $\mathbb{R}^3$, and the space described by two cuboids sharing a common edge in $\mathbb{R}^3$, all with their Euclidean topology.

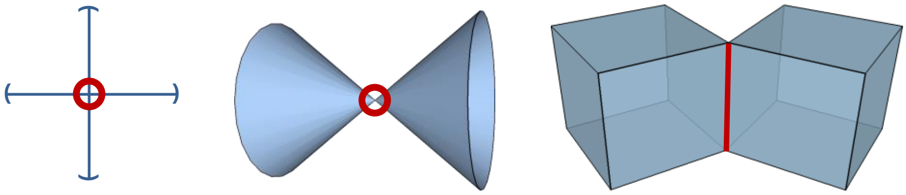


Figure 36: Examples of non-manifold spaces which are not locally Euclidean at every point (depicted in red).

2-dimensional topological manifolds are synonymously called *surfaces* in literature and play an important role in the context of spatial modelling. Fundamental examples are the sphere $\mathbb{S}^2$ and the torus $\mathbb{T}^2$ which belong to the family of *compact, connected* 2*-manifolds without boundary* whose members are equivalently called *closed surfaces*. Closed surfaces can be classified up to homeomorphism by their *genus* and their *orientability* which hence are topological invariants. The genus is equal to the maximum number of disjoint simple closed curves which can be cut from the closed surface without disconnecting it. For example, the sphere has genus 0 whereas the torus has genus 1. If a connected surface is orientable, it is possible to distinguish two sides that are commonly called front-side and back-side. This is not possible in the non-orientable case and non-orientable closed surfaces cannot be embedded in $\mathbb{R}^3$ without self-intersections (Gröger & Plümer 2011a). Famous examples for non-orientable surfaces are the Möbius strip and the Klein bottle. Members of the family of *compact surfaces with boundary* can be obtained from closed surfaces by cutting one or more *holes* (i.e., open disks) into the latter. This yields one or more disjoint circles as boundary components for the resulting compact surface. Examples are the closed disk $\overline{\mathbb{B}}^2$ (sphere with one hole) or the annulus (sphere with two holes). The number of boundary components is a topological property.

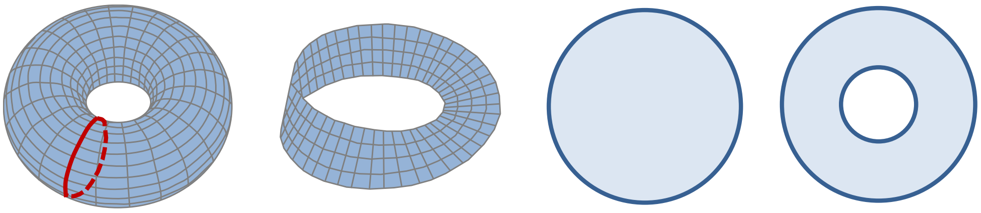


Figure 37: Examples of surfaces (from left to right: the torus $\mathbb{T}^2$ with one cutting curve drawn in red, the Möbius strip, the closed 2-disk $\overline{\mathbb{B}}^2$ having one boundary component, and the annulus having two boundary components).

Compact, orientable surfaces with or without boundary can be classified based on their *Euler characteristic* (cf. theorem A.85 and theorem A.87). Precisely, for a member M of the family of compact, orientable surfaces the Euler characteristic $\chi(M)$ is given as $\chi(M) = 2 - 2g - h$ where g denotes the genus and h the number of holes of the surface. For example, the 2-sphere $\mathbb{S}^2$ has Euler characteristic $\chi(\mathbb{S}^2) = 2 - 2 \cdot 0 - 0 = 2$ whereas the closed disk has $\chi(\overline{\mathbb{B}}^2\ ) = 2 - 2 \cdot 0 - 1 = 1$. The Euler characteristic renders a topological invariant.

3-manifolds are essential objects for mathematically describing 3-dimensional solid (volumetric) spatial objects embedded in $\mathbb{R}^3$. In literature, a *single-shell manifold solid* (or simply *manifold solid*) is commonly defined to be a non-empty, connected, compact 3-manifold with boundary embedded in $\mathbb{R}^3$ whose boundary is a closed, oriented 2-manifold tamely embedded in $\mathbb{R}^3$ (e.g., Mäntylä 1988 and appendix A). The orientation of the boundary induces an orientation on the manifold solid itself. It follows from the *Jordan-Brouwer separation theorem* that the boundary surface separates $\mathbb{R}^3$ into exactly two disjoint connected components. One component obviously is equal to the interior of the manifold solid whereas the other denotes the unbounded ambient space of the manifold solid in $\mathbb{R}^3$ (commonly called the *exterior*). A *tame* embedding of the boundary surface ensures that the ambient space is simply connected, which rules out pathological cases that do not conform to an intuitive spatial understanding such as the *Alexander horned sphere*. A *k-shell manifold solid* is a non-empty compact 3-manifold embedded in $\mathbb{R}^3$ with $k \geq 1$ boundary components whose disjoint union is a compact, oriented 2-manifold without boundary tamely embedded in $\mathbb{R}^3$. A single-shell manifold solid obviously is the special case of a k-shell manifold solid for $k = 1$. A *connected k-shell manifold solid M* is a *k-shell manifold solid* whose interior $Int(M)$ is restricted to be connected. In case $k > 1$, M is said to have $k - 1$ internal voids.

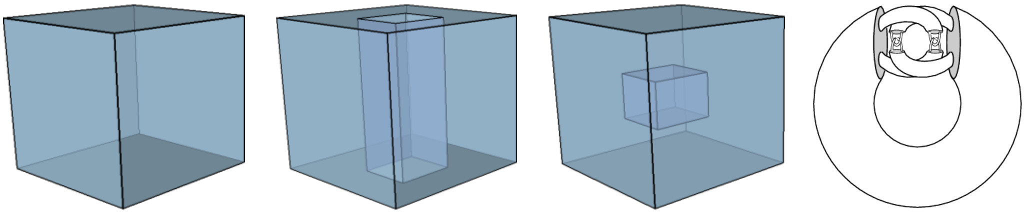


Figure 38: Examples of manifold solids (from left to right: a 1-shell manifold solid being homeomorphic to $\overline{\mathbb{B}}^3$, a 1-shell manifold solid with through hole, and a connected 2-shell manifold solid with one internal void). The Alexander horned sphere (right) is no manifold solid although it is likewise homeomorphic to $\overline{\mathbb{B}}^3$.

The members of the family of k-shell manifold solids can be classified up to homeomorphism based on their first three *Betti numbers* (cf. theorem A.94). In general, the Betti numbers allow for counting the number of holes of different dimensions in an orientable, compact n-manifold M with the first Betti number $\beta_0(M)$ denoting the number of connected components, $\beta_1(M)$ denoting the number of 2-dimensional holes, and $\beta_2(M)$ denoting the number of 3-dimensional holes. It can be shown that the alternating sum of the Betti numbers equals the Euler characteristic of such spaces (cf. theorem A.95). For example, the Euler characteristic of a 1-shell manifold solid being homeomorphic to $\overline{\mathbb{B}}^3$ results in $\chi(\overline{\mathbb{B}}^3) = \beta_0(M) - \beta_1(M) + \beta_2(M) = 1 - 0 + 0 = 1$. For a connected 2-shell manifold solid M with one internal void (cf. right of figure 38) the Euler characteristic evaluates to $\chi(M) = 1 - 0 + 1 = 2$.

2.5.1.3 *Cell complexes and CW complexes*

In algebraic topology, topological spaces are often constructed by gluing together subsets nicely along their boundary. The primary building blocks are *cells* of dimension n. An *open n-cell* is any topological space homeomorphic to the open unit n-ball $\mathbb{B}^n$. Analogously, a *closed n-cell* is homeomorphic to the closed unit n-ball $\overline{\mathbb{B}}^n$.

A *cell complex* is given by a pair $(X, \mathcal{E})$ where X is a topological Hausdorff space together with a partition $\mathcal{E}$ of X into subspaces that are open cells of various dimensions and whose disjoint union is X (called *cell decomposition* of X). For each open n-cell $e \in \mathcal{E}$ and $n \geq 0$, there exists a continuous map $\phi_e: \overline{\mathbb{B}}^n \to X$ (called the *characteristic map* for e) that restricts to a homeomorphism from $Int(\overline{\mathbb{B}}^n)$ onto e, and maps $\partial\overline{\mathbb{B}}^n$ into the union of all cells of $\mathcal{E}$ of dimension strictly less than n. More intuitively, cell complexes can be defined by an inductive construction process. Starting with the empty set $\emptyset$, a discrete space X_0 is built from attaching one or more disjoint 0-cells. X_1 results from the union of 1-cells whose boundaries are identified to lie on these points. Attaching 2-cells forms a new space X_2 by mapping the boundary of each 2-cell into the collection of cells of lower dimension, and so on. The subspace $X_n \subseteq X$ consisting of the union of all cells in $\mathcal{E}$ of dimension less or equal to n is said to be the n-*skeleton* of the cell complex.

A *CW complex* is a cell complex for which the closure of each cell is contained in a union of finitely many lower dimensional cells and which has the weak topology, i.e. each closed subset has a closed intersection with every cell closure. Cell complexes having a finite number of cells are inherently CW complexes. A CW complex $(X, \mathcal{E})$ is said to be *regular* if the image of the characteristic map ϕ_e of each open cell $e \in \mathcal{E}$ is equal to a closed cell and each open cell meets every 0-cell in $\mathcal{E}$ at most once. It is said to be *proper* iff the non-empty intersection of any two closed cells in X is also a closed cell. Different CW decompositions of the unit 2-sphere $\mathbb{S}^2$ are illustrated in the following figure.

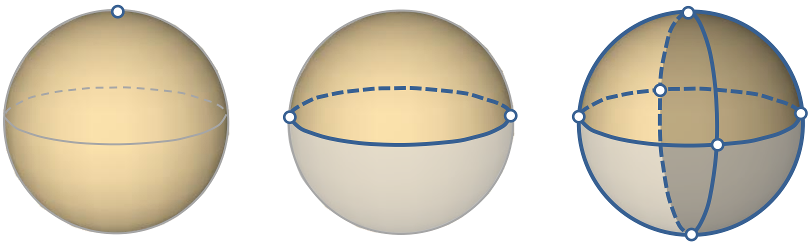


Figure 39: Different CW decompositions of the 2*-sphere* $\mathbb{S}^2$ *(from left to right: minimal CW complex with one* 0*-cell and one* 2*-cell, regular CW complex with two cells per dimension, and proper CW complex containing six* 0*-cells, twelve* 1*-cells, and eight* 2*-cells).*

Following from one of the earliest results in surface theory it can be shown that the Euler characteristic of the topological space underlying a finite CW complex is equal to the alternating sum of the numbers of cells in each dimension. Precisely, for a finite CW complex $(X, \mathcal{E})$ of dimension n, the Euler characteristic of X is given by $\chi(X) = \sum_{k=0}^{n}(-1)^k c_k$, where c_k denotes the number of k-cells in the CW decomposition of X. For example, for the minimal CW decomposition of the 2-sphere $\mathbb{S}^2$ as shown on the left of figure 39 we get $\chi(\mathbb{S}^2) = 1 - 0 + 1 = 2$ (which is also yielded for the further examples in figure 39). Since the Euler characteristic is a topological invariant, this result must, of course, agree with the result of the formula $2 - 2\mathrm{g} - \mathrm{h}$ respectively the alternating sum of the Betti numbers of the topological space as introduced in the previous section. Thus, there is simple and efficient validity check for the cell decomposition of a topological space.

2.5.1.4 Poincaré Duality

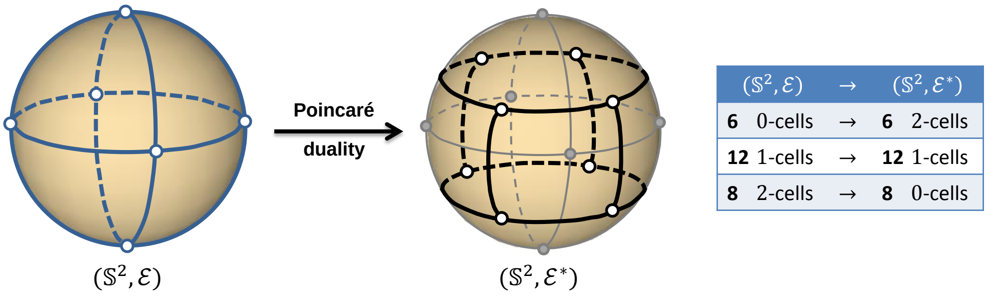
In general, the notion of *duality* takes numerous meanings and definitions in different contexts in mathematics. However, common to all of them is the idea of translating or pairing objects between two classes or domains in a bijective way. Precisely, given an object A of a certain class X, one can identify an object A^* from another class Y and vice-versa. A^* is the *dual* object of A if $A^{**} = A$, and $*$ is said to be an *involution* operation. The properties of an object with respect to its domain are completely determined by its *dual* counterpart. Put simply, both objects represent the same thing but from two different points of view (Ledoux & Gold 2007). In order to study certain properties of an object, the translation of the object into its dual may often be more suitable. One of the earliest geometric examples is the duality of the *Platonic solids* (e.g., the cube and the octahedron form a dual pair). A famous example from computational geometry with relevance in the field of GIS is the duality between the *Delaunay triangulation* of a set of points in the Euclidean plane and the *Voronoi diagram* for these points.

The *Poincaré duality* establishes a duality between combinatorial structures on topological manifolds. In its modern form, it states an isomorphism between the homology and cohomology groups of manifolds as follows.

Theorem 2.1 (Poincaré duality theorem). For an orientable closed manifold M of dimension n, the kth cohomology group of M is isomorphic to the $(n-k)$th homology group of M for all integers k, i.e. $H_k(M;\ \mathbb{Z}) \cong H^{n-k}(M;\ \mathbb{Z})$ (Munkres 1984, Hatcher 2008).

In cellular homology, the Poincaré duality can be interpreted in terms of dual cell structures. This notion is also applied in this thesis. Precisely, given a CW decomposition $\mathcal{E}$ of M, there exists a dual CW decomposition $\mathcal{E}^*$ of M. Every k-cell e_k of $\mathcal{E}$ corresponds to an $(n-k)$-cell e_k^* in $\mathcal{E}^*$ and thus $\mathcal{E}_k = \mathcal{E}_{n-k}^*$. The cellular boundary map $\partial\colon \mathcal{E}_k \to \mathcal{E}_{k-1}$ which maps the boundary of k-cells into the $(k-1)$-skeleton of $\mathcal{E}$ becomes the cellular coboundary map $\delta\colon \mathcal{E}_{n-k}^* \to \mathcal{E}_{n-k+1}^*$ in the dual CW decomposition $\mathcal{E}^*$ (cf. Hatcher 2008). The cells of the primal and dual CW decomposition of the manifold are in bijective correspondence. Since every k-cell of the primal decomposition is paired with an $(n-k)$-cell of the dual decomposition, the Poincaré duality intuitively *reverses dimensions*.

An example for the proper CW decomposition of the 2-sphere consisting of eight 2-cells, twelve 1-cells, and six 0-cells as well as its dual is shown in figure 40. Each 2-cell in $\mathcal{E}$ is mapped to a dual 0-cell in $\mathcal{E}^*$, each 1-cell in $\mathcal{E}$ crosses a dual 1-cell in $\mathcal{E}^*$, and each 0-cell of $\mathcal{E}$ is contained in a dual 2-cell in $\mathcal{E}^*$.


Poincaré
duality
$(\mathbb{S}^2, \mathcal{E})$
$(\mathbb{S}^2, \mathcal{E}^*)$

$(\mathbb{S}^2, \mathcal{E})$	→	$(\mathbb{S}^2, \mathcal{E}^*)$
6 0-cells	→	**6** 2-cells
12 1-cells	→	**12** 1-cells
8 2-cells	→	**8** 0-cells

Figure 40: Primal cell decomposition $(\mathbb{S}^2, \mathcal{E})$ of the 2-sphere (left) and Poincaré dual $(\mathbb{S}^2, \mathcal{E}^*)$ (right). The primal cell decomposition is repeated on the right in dark grey.

2.5.2 Graph Theory

A graph-based conceptualization of indoor space is defined in almost all approaches to indoor space modelling (cf. chapter 2.2) in order to enable, for example, path searches based on existing graph traversal algorithms. Graph structures are also applied in this thesis, and thus a brief overview of those terms and concepts from graph theory which are relevant in the further course of the thesis is provided in the following.

Based on the notion of CW complexes, a graph G can be defined to be the 1-skeleton of an n-dimensional CW complex (Lee 2011). Each 0-cell of the complex is called a *vertex*, and each 1-cell is said to be an *edge*. A graph G is commonly denoted as pair (V, E) where V is the set of vertices and E the set of edges. It follows from the definition of a CW complex that the boundary of each edge consists of one or two vertices which are said to be *incident*. An edge being incident to only one vertex is called a *self-loop*. Two edges are *adjacent* if they are incident to a common vertex. Likewise, two vertices are *adjacent* if they are on the boundary of a common edge. The *degree* of a vertex is the number of its incident edges. A graph is said to be *finite* if the associated CW complex is finite, and *connected* if the underlying topological space of the associated CW complex is connected. It is *directed* if all edges in E have a direction associated with them (i.e., the incident vertices are ordered) and *undirected* otherwise. A *path* is a sequence of edges where two consecutive edges are incident to a common vertex. The first vertex of the path is called *start vertex*, whereas the last vertex (in case the sequence of edges is finite) is said to be the *end vertex*. If the start and end vertex are identical, then the path is called a *cycle*. A path with no repeated vertices is *simple* and consequently homeomorphic to $\overline{\mathbb{B}}^1$.

A graph whose vertices can be partitioned into k disjoint subsets, with $k \geq 1$, such that no two vertices within the same subset are adjacent is said to be a *k-partite graph*. Given an undirected graph, a *clique* is defined as subset of its vertices such that every two vertices in the subset are adjacent. Two graphs G and H are called *isomorphic* (written as $G \simeq H$) if there exists a bijection $f\colon V(G) \to V(H)$ such that any two vertices $u, v \in V(G)$ are adjacent iff $f(u)$ and $f(v)$ are adjacent in H.

An *embedding* of a graph G into Euclidean n-space results from associating each vertex with a point in $\mathbb{R}^n$ and each edge with a simple space curve being homeomorphic to $\overline{\mathbb{B}}^1$ (Gröger & Plümer 2011a).The boundary points of the space curve are the points in $\mathbb{R}^n$ associated with the vertices incident to the edge. Space curves are neither allowed to intersect at an interior point nor to include points that are associated with other vertices. A graph embedded according to these rules into the plane $\mathbb{R}^2$ is also called *planar graph*. Not every graph is planar but at least can be embedded in $\mathbb{R}^3$.

Chapter 3
Multilayered Space-Event Model (MLSEM)

After having identified the multiple challenges to indoor navigation and motivated by the strengths and weaknesses of related work, this chapter develops a spatio-semantic indoor space model for indoor navigation that aims at addressing the challenges and at meeting the requirements postulated in chapter 2.4. The focus of the presentation in this chapter is hereby on the conceptual definition of the model elements and their complete, consistent, and correct mathematical formalization. As stated in the research scope of this thesis (cf. chapter 1.3), the work undertaken here is based upon and continues the previous work on the *Multilayered Space-Event Model* (MLSEM) as introduced in (Becker et al. 2009a), (Becker et al. 2009b), and (Nagel et al. 2010), and thus is presented and discussed in the following under that same name.

The presentation of the MLSEM follows three main consecutive steps. In a first step, two fundamental concepts for the modelling of indoor space, namely *space cell* and *space layer*, are defined that realize a generic model for structuring the indoor space according to arbitrary notions of space. The discussion of both concepts is organized into 1) the semantic conceptualization of space, 2) the geometric-topological representation of space, and 3) the sound embedding in a mathematical framework. A fundamental aspect is hereby the *dual representation* of indoor space that introduces a deterministic one-to-one relation between the 2-dimensional respectively 3-dimensional description of space and its graph-based conceptualization. Inspired by existing dual-graph-based approaches presented in chapter 2.2.3, the *Poincaré duality* is used as mathematical formalism for the dual representation which is developed in depth in the course of this chapter. Issues in related work applying the Poincaré duality as well as in the previous publications on the MLSEM are revealed in this discussion.

The second step elaborates on the integration of complementary indoor space representations (i.e., *space layers*) in a common model in order to facilitate their joint consideration in navigation tasks. For this purpose, the concept of a *multilayered graph* is introduced and formalized which links the graph-based conceptualizations of the separate space layers in a single combinatorial structure. It is shown in detail how this core concept of the MLSEM supports the tasks of path planning, localization, and tracking based on the derivation of valid and unique *joint states* of navigation. Moreover, the modelling of hierarchical space structures is presented carefully along the notions of *subspace cells* and *subspace layers* as well as their conceptual and mathematical definition.

In a final step, a formal *space layer algebra* is defined which allows for algebraic expressions to be built up from space layers and well-defined binary operations in order to derive an integrated or mutually exclusive view on indoor space. It is then discussed how this space layer algebra in addition to the developed concepts can be used in supporting multiple and different navigation contexts.

3.1 Structured Space Model

The structuring of indoor space according to different notions of space and partitioning schemas has been identified as an essential requirement to indoor navigation models in chapter 2.4. The principal challenge posed by this requirement is that of integrating the multiple and distinct space representations that result from decomposing indoor space along arbitrary partitioning criteria. For example, an obvious choice for structuring indoor space reflects the topographic built-up space and is a consequence of applying the space separating system of the built structure as partitioning schema. The resulting space partitions hence follow the architectural constraints and describe the layout of the interior environment. Alternative partitioning schemas for topographic space yielding contrary space partitions include the decomposition of architectural entities such as rooms and corridors into navigable regions for different modes of locomotion or the representation of visibility areas of indoor landmarks. Complementary notions of indoor space such as sensor space or logical space render substantially different space representations that may denote the subdivision of indoor space due to sensor characteristics of a given localization technology like signal coverage or signal strength, or the partitioning along contextual navigation constraints into security or emergency zones. Although the partitioning criteria as well as the semantics of the resulting space models differ in the examples, common to all is the fact that the resulting space partitions can be expressed through individual geometric shapes embedded in indoor space. In this sense, indoor space can be understood as the common spatial reference frame for the distinct space models.

The MLSEM builds upon this general notion of indoor space and employs a cell-based conceptualization of space in order to capture the spatial structuring along various partitioning schemas. Each partitioning schema is hereby mapped onto a separate configuration of cells which is modelled independently from further indoor space representations. This modelling approach thus facilitates the decomposition of indoor space into smaller units along specific criteria without influencing complementary space partitionings. The two core concepts of this *structured space model* are called *space cell* and *space layer*. Whereas a space cell defines a generic model for a partition of indoor space and hence abstracts from a specific notion of space and partitioning schema, the collection of all space cells that follow from a given partitioning schema is said to form an individual space layer. The complete view on indoor space comprising topographic space, sensor space, logical space, and further space concepts is then built up inductively from the collection of space layers as they all reference and describe the same real world space. The structured space model represents the geometric-topological, symbolic, and semantic aspects of both space cells and layers in a well-defined way, and can therefore be classified as cell-based hybrid approach to indoor space modelling.

The structured space model is a core component of the MLSEM in order to answer the requirement of supporting multi-granular space representations and different notions of indoor space, and is presented in detail in the following subchapters.

3.1.1 Space Cell

A *space cell* is the central concept and smallest building block for modelling and representing indoor space within the structured space model, and describes space from a conceptual, quantitative, and qualitative perspective.

3.1.1.1 Conceptualization of Space

A space cell is a subset of indoor space along a given partitioning schema. Its semantic meaning depends on the underlying notion of space and the applied partitioning criteria. For example, a space cell in topographic space may represent a room, a corridor, or any other physical entity within the built skeleton of a building or facility, as well as any more fine-grained subspace thereof, whereas a space cell in the sensor space representation of a Wi-Fi based positioning system may reflect the region of indoor space within the propagation range or signal strength band of a given transmitter. Likewise, a space cell may express the proximity range of an RFID sensor belonging to an RFID-based localization infrastructure as well as arbitrary virtual space partitions following from logical considerations.

As we travel through indoor space we intuitively move from one space cell to another. Since separate space concepts yield alternative decompositions of indoor space, we even move through several space cells at the same time which cover the same spatial region but belong to different space representations. For example, a topographic space cell capturing the indoor space occupied by a room might overlap with, or be covered by, further space cells denoting the Wi-Fi radio signal coverage of the room or the fire detector zone monitoring the room. Moreover, and in contrast to other cell-based approaches, space cells are not restricted to the modelling of free or navigable space but may also represent non-navigable spaces. On the one hand, this comprises objects in topographic space which physically obstruct movement such as walls or obstacles. On the other hand, space cells may also denote areas which logically constrain navigation such as security zones. Each space cell has a conceptual *boundary* that limits its spatial coverage and semantic scope. This implies that entering or leaving a space cell while travelling through indoor space requires crossing its boundary. The boundary of a space cell therefore needs to be partly intangible so that it can be traversed by moving persons or objects, but may also be partly impassable, for example, in case it reflects physical discontinuities such as wall or floor surfaces. Tangible boundaries mostly occur for topographic space cells, whereas sensor space cells are inherently enclosed by virtual borders which do not impede the bodily movement. A space cell can be enriched with thematic properties to account for symbolic and semantic information associated with the portion of space covered by the space cell (e.g., room number, MAC address of a Wi-Fi transmitter, security level of a security zone, etc.).

As elaborated in chapter 2.2, the idea of structuring indoor space into cells is common to most approaches to indoor space modelling but is also typically limited to the representation of the interior built environment along a single and rigid partitioning schema. The notion of a space cell can therefore be viewed as a higher-level abstraction which subsumes the different cell-based approaches and also provides a common basis for the integration of different semantic classifications of indoor space as presented in chapter 2.1.3. For example, *locations* and *regions* as proposed by (Hu & Lee 2004) and (Bittner 2001) can obviously be mapped onto topographic space cells. (Bittner 2001) also differentiates between *bona-fide* and *fiat* boundaries of spatial regions but only applies them as partitioning schema to extract regions. Moreover, fiat boundaries are restricted to virtual borders that result from human cognitive acts, whereas the boundaries of space cells may even be imperceptible for humans (e.g., physical signal characteristics or logical facts not being aligned with the built reality). The structuring of indoor space along the human perception into CONTAINER and SURFACE image schemas as introduced by (Johnson 1987) can likewise be translated into topographic space cells and their boundaries, with the SURFACE schema again expressing only perceivable boundaries of a CONTAINER. The modelling of physical and virtual spaces as *spatial artefacts* and their classification into *object*, *operational*, *functional*, and *range spaces* according to (Bhatt et al. 2009) conforms to the notion of physical indoor space (e.g., topographic and sensor space) and logical indoor space as well as the representation of corresponding spaces through different types of space cells.

It is important to note that purely geometric partitioning schemas are already sufficient and explicitly supported by the structured space model as well. Thus, topographic space cells may be arranged in a continuous regular grid underpinning grid-based hybrid approaches (cf. chapter 2.2.1) or denote the irregular cell decomposition of indoor space following from geometric rather than semantic decomposition methods (cf. chapter 2.1.2).

3.1.1.2 Geometric-topological Space Representation

Each space cell is spatially modelled according to the geometric-topological model shown in figure 41. The model proposes four distinct representations depicted as quadrants I to IV in figure 41 which result from separate geometric and topological descriptions of the space cell in both primal and dual space. The primal space representation is concerned with the real world shape and location of the space cell as subset of the indoor space, and is given by a 2-dimensional or 3-dimensional geometric figure (quadrant I) and its corresponding topological description (quadrant II). In dual space, the space cell is mapped onto a single 0-dimensional node whose representation is again split into a geometric (quadrant IV) and a topological (quadrant III) description. Several space cells can be linked through their dual node representation in order to establish geometric and topological graph structures.

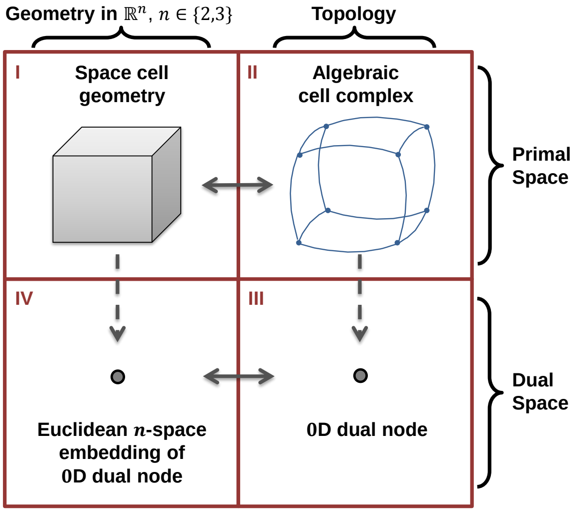


Figure 41: The geometric-topological representation model for space cells. The primal space representation (quadrants I and II) is depicted for a 3*-dimensional space cell but likewise holds in two dimensions.*

The vertical subdivision of the geometric-topological model into geometry space (quadrants I and IV) and topology space (quadrants II and III) allows for the derivation of quantitative aspects of a space cell based on its geometric representation by a set of coordinate tuples as well as the separate but interrelated description of its qualitative or topological properties. Together with the semantic notion of a space cell as discussed above, this separation into geometry and topology space agrees with the idea of defining explicit models for *conceptual*, *quantitative*, and *qualitative space* as proposed by (Bhatt et al. 2009).

The possibility of representing the shape and location of a space cell in two or three dimensions yields a high flexibility and goes beyond the expressivity of almost all presented approaches to indoor space modelling which are limited to either a 2-dimensional or a 3-dimensional setting. This flexibility is however important, for example, with respect to existing building data which serves as input to populate the navigation space model. Most approaches implicitly built on 2-dimensional building floor plans and are defined in two dimensions only, thus neglecting the support for 3-dimensional building models, or vice versa. The structured space model is not restricted to one alternative but supports both ways of representing indoor space.

The horizontal partitioning into primal space (quadrants I and II) and dual space (quadrants III and IV) proposes a deterministic one-to-one relation between a space cell as spatial region in indoor space and its node representation in a corresponding graph-based conceptualization of space. In contrast to other cell-based approaches where the mapping of cells onto nodes is implicitly understood but rarely developed in a formal way (cf. chapter 2.4), the schema shown in figure 41 provides a consistent relation between both representations that is furthermore applicable to arbitrary space concepts beyond topographic space due to the generic notion of space cells. The pairing of the primal and dual space representations of a space cell builds upon the Poincaré duality in topology space which thus provides its formal theoretical foundation (cf. chapter 3.1.2.4 for a detailed description).

3.1.1.3 Mathematical Formalization

The informal conceptual description of space cells presented above is used in the following to give space cells a mathematically formalized definition that provides a sound and consistent basis for the modelling of indoor space according to the structured space model.

Definition 3.1 (Space cell). A *space cell* S is a quintuple $(GM, TP, v_{TP}, v_{GM}, A)$ where GM and TP are the geometric respectively topological descriptions of S in primal space, v_{GM} and v_{TP} are the geometric respectively topological descriptions of S in dual space, and A is a set of symbolic or semantic attributes associated with S itself.

It obviously follows that GM, TP, v_{TP}, and v_{GM} correspond to the quadrants I, II, III, and IV of the geometric-topological representation schema depicted in figure 41. Both the primal and dual space representation of a space cell is consistently embedded in an ambient n-dimensional Euclidean space $\mathbb{R}^n$ (cf. definition A.27) with $2 \leq n \leq 3$. This ambient space is not to be confused with the indoor space in question. In the context of the structured space model, indoor space is rather to be understood as that space which is collectively made of the space cells, and thus is itself a subset of $\mathbb{R}^n$.

The primal space geometry GM of a space cell is a 2-dimensional or 3-dimensional bounded regular semi-analytic subset of $\mathbb{R}^n$ (cf. Kresse & Danko 2012, Requicha 1980, and chapter 4.1). In order to ensure a geometric configuration that conforms to an intuitive spatial understanding, it has to additionally render a topological manifold M with boundary of the same dimension. This leads to the following formal definition for the primal space geometry.

Definition 3.2 (Primal space geometry of a space cell). The *primal space geometry* $GM(S)$ of a space cell S is a non-empty n-dimensional subset of Euclidean space $\mathbb{R}^n$, with $GM(S) \subset \mathbb{R}^n$ and $2 \leq n \leq 3$, and is required to form

(i) a connected, orientable, compact surface with boundary in case $n = 2$, and
(ii) a connected k-shell manifold (cf. definition A.93) solid otherwise.

A space cell is said to be n-dimensional. Since the dimension of $GM(S)$ is equal to the dimension of the ambient space $\mathbb{R}^n$, the co-dimension of $GM(S)$ is zero. According to the triangulation theorem for low-dimensional manifolds (cf. theorem A.74), every compact topological manifold of dimension $n \leq 3$ is a trianguable space and thus admits a finite CW decomposition. This fact is utilized in the definition of the primal topological description TP of a space cell.

Definition 3.3 (Primal space topology of a space cell). Let M be the n-dimensional topological manifold described by $GM(S)$ of a space cell S. Then the *primal space topology* $TP(S)$ of S is a finite CW complex $(M, \mathcal{E})$ of dimension n whose underlying topological space is M. The CW decomposition $\mathcal{E}$ of M is restricted to contain a single n-cell which is attached to all lower dimensional cells in $\mathcal{E}$.

It obviously follows that the primal space occupied by a space cell is a topological space in its own right but also a compact subset of the ambient space $\mathbb{R}^n$ rendering it a subspace endowed with the induced Euclidean topology. The cellular decomposition $\mathcal{E}$ in primal topology space provides a combinatorial model of the space cell which facilitates the efficient algorithmic computation of and reasoning about the topological properties of space cells. The above definition of $TP(S)$ does not imply a particular choice of geometric realization for individual k-cells $e_k \in \mathcal{E}$ as long as this realization preservers the boundary map $\partial: \mathcal{E}_k \rightarrow \mathcal{E}_{k-1}$ which sends the boundary ∂e_k into the union of finitely many lower dimensional cells. Specifically, a k-dimensional subset $A \subset GM(S)$ is a geometric realization of e_k if it includes the realizations of all cells on the cell boundary $\partial e_k = \bar{e}_k \setminus e_k$ in $\mathcal{E}_{k-1}$, and thus $A \cong \bar{e}_k$. The subset A is said to be the *geometric carrier* of e_k (cf. Farin et al. 2002), and it follows that $GM(S)$ is the geometric carrier of $TP(S)$.

Independent of the dimension of the space cell in primal space, its dual space representations v_{TP} and v_{GM} describe discrete 0-dimensional subspaces of $\mathbb{R}^n$ whose definitions are as follows.

Definition 3.4 (Dual space topology of a space cell). The *dual space topology* $v_{TP}(S)$ of a space cell S is a finite 0-dimensional CW complex (X, δ) whose underlying space X is a discrete and connected 0-dimensional manifold and whose CW decomposition δ is a singleton set of exactly one 0-cell.

Definition 3.5 (Dual space geometry of a space cell). The *dual space geometry* $v_{GM}(S)$ of a space cell S is a discrete 0-dimensional point in Euclidean space $\mathbb{R}^n$, $2 \leq n \leq 3$, that is restricted to lie within the interior of the primal space geometry $GM(S)$ and thus $v_{GM}(S) \in Int(GM(S)) \subset \mathbb{R}^n$.

Symbolic or semantic space information is modelled as set of attributes for each space cell. An attribute is hereby understood as name-value pair in order to be able to explicitly model the meaning of the attribute in addition to its value. Example attributes are symbolic identifiers such as room numbers, place names, or sensor IDs as well as physical or logical characteristics of the partition of indoor space occupied by the space cell. The same attribute can be shared by several space cells. For example, since space cells are allowed to cover (parts of) the same topographic room, several space cells may consequently carry the same room number as symbolic attribute.

Definition 3.6 (Set of attributes of a space cell). The *set of attributes* $A(S)$ of a space cell S is the set $A(S) = \{a_1, \dots, a_n\}$ where each attribute $a_i \in A(S)$ is given by a 2-tuple (n_i, v_i) which maps the name of the attribute onto n_i and its value onto v_i.

The above definitions realize the mathematical embedding of the geometric-topological model of space cells as presented in figure 41. The generic understanding of a space cell as non-empty subset of Euclidean space is independent from a particular choice of notion of the underlying indoor space, and thus provides a common abstraction schema for arbitrary space concepts. As stated above, the Poincaré duality shall be utilized to formally express the relation between the primal and dual space representations of a space cell. However, this first requires a mathematical model for a set of space cells which is presented in the following section.

3.1.2 Space Layer

A *space layer* is a conceptual view on indoor space that reflects the decomposition of indoor space according to a given notion of space and partitioning schema, and describes space through a collection of space cells as well as their consistent graph-based representation.

3.1.2.1 Conceptualization of Space

A space layer is built from aggregating finitely many space cells resulting from a specific space partitioning to a larger space. The view on indoor space captured by a space layer therefore depends on the semantics of the aggregated space cells. For example, a *topographic space layer* may describe the interior built-up space by assembling topographic space cells representing the decomposition of indoor space into structural elements and architectural units such as rooms, corridors, stairs, elevators, walls, columns, etc. Since alternative decompositions of topographic space follow distinct partitioning criteria (e.g., according to different types of locomotion), they have to be modelled on separate topographic space layers. Likewise, each subdivision of indoor space into signal propagation areas of sensors belonging to individual localization technologies such as Wi-Fi and RFID is to be reflected on separate *sensor space layers*, whereas the spatial configuration of virtual security zones obviously has to be mapped by a *logical space layer*. It follows that space layers facilitate the conceptual separation of distinct notions of space and corresponding indoor space models, and changes to the layout of one space layer do not impact other space layers. For example, modifying the space cells on a Wi-Fi space layer in order to reflect a change to the Wi-Fi infrastructure in a building neither influences a topographic space layer nor complementary sensor and logical space layers. The concept of a space layer hence overcomes disadvantages of mixing different space concepts in a single space representation (cf. chapter 2.4).

The aggregation of space cells on a space layer has to meet two main conditions. First, the space cells are required to be *mutually non-overlapping*, i.e. two space cells may be disjoint or are allowed to touch at their boundaries while their interiors may not intersect. This requirement agrees with the ontological axiom introduced by (Bittner 2001) according to which distinct bona-fide or fiat spatial objects of the same ontological kind cannot overlap. The axiom is also applicable for a space layer since its space cells result from the same decomposition criteria and thus are of ontologically equal kind. Moreover, to be mutually non-overlapping is a prerequisite for evaluating qualitative *connectedness* relationships between spatial regions (cf. chapter 2.1.4). Most of the presented indoor space models therefore implicitly or explicitly rely on this requirement. Whereas in topographic space mutually non-overlapping space cells rather naturally follow from the architectural layout of the structural building elements and their space separating system, the propagation areas of, for example, Wi-Fi transmitters are usually meant to overlap in order to ensure full signal coverage within a building. Most approaches including the conceptual space model of (Bittner 2001) do not account for sensor spaces and hence do not address this issue. Within the structured space model, all space layers are equally required to represent non-overlapping space cells which generalizes the ontological axiom of (Bittner 2001) to arbitrary space concepts. In case of overlapping sensor propagation areas this means that the areas must be mapped onto non-overlapping space cells each having homogenous signal properties (cf. figure 46 in chapter 3.1.2.3). A second consequence is that spatial *containment* relationships cannot be expressed on a single space layer as they would require overlapping space cells. However, space cells on different space layers are allowed to spatially overlap as they reflect different partitioning schemas. This, again, is consistent with and generalizes the ontological axiom of (Bittner 2001) that spatial objects of ontologically different kind can be co-located.

A second condition for the aggregation of space cells is that the aggregation needs to be *jointly exhaustive* so that every point in Euclidean space can be precisely identified to lie either within one space cell or on the common boundary of neighbouring space cells. This rule consequently forces the spatial configuration of space cells to render a complete tiling of Euclidean space. A similar condition is imposed by (Stoffel et al. 2007) who however restrict the tiling to the scope of a floor plan which is to be seen as a subspace of the Euclidean plane. The full coverage of Euclidean space requires the introduction of a further conceptual space cell which is called *outer* in the context of the structured space model. Outer can be informally defined as the complement of all primal space cell representations on a given space layer and therefore is unique per space layer. Thus, in contrast to all other space cells, the space occupied by outer is topologically unbounded. The semantic meaning of outer depends on the space layer in question. On a sensor space layer it may spatially represent regions without signal perception. On a topographic space layer, the entire building space is typically covered by space cells so that outer represents the outdoor space in such cases. The notion of an outer space is rarely mentioned in the presented proposals for indoor space models. In the further course of this chapter it will be shown that outer plays an important role in the context of the Poincaré duality. Although it hence needs to be addressed by all approaches building upon this theorem, none of the presented dual-graph-based models provides a formal definition (cf. chapter 2.2.3) which also holds for the previous publications on the MLSEM and the structured space model (Becker et al. 2009a, Becker et al. 2009b, Nagel et al. 2010).

3.1.2.2 Geometric-topological Space Representation

Since a space layer is the result of a union of space cells, its spatial representation follows the geometric-topological model of a single space cell. Therefore, a space layer is systematically represented in both primal and dual space according to the four quadrants illustrated in figure 42.

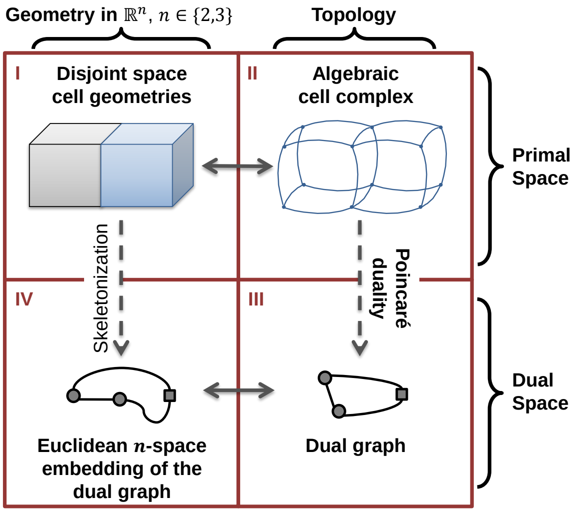

Geometry in $\mathbb{R}^n$, $n \in \{2,3\}$
Topology
I
Disjoint space cell geometries
II
Algebraic cell complex
Primal Space
Skeletonization
Poincaré duality
IV
III
Euclidean n-space embedding of the dual graph
Dual graph
Dual Space

Figure 42: The geometric-topological representation model for space layers (after Becker et al. 2009a). The primal space representation (quadrants I and II) is depicted for 3*-dimensional space cells but likewise holds in two dimensions. The square node in dual space represents the dual node of the outer space cell.*

The primal space representation of a space layer is given by the union of all space cell geometries including outer (quadrant I) as well as the associated topological description (quadrant II). The space cells need to share the same dimension, and since they are requested to be mutually non-overlapping, their spatial configuration results in an algebraic cell complex in primal topology space. Applying the Poincaré duality to this structure yields a dual cell complex with the dimensions of topological cells being reversed. Thus, in dual topology space, every space cell is represented by a 0-cell and its boundary components are mapped onto 1-cells. Following the NRS model proposed by (Lee 2001) (cf. chapter 2.2.3), a topological *dual graph* is established with the 0-cells forming the set of nodes and the 1-cells denoting the edges linking the nodes (quadrant III). The Poincaré duality ensures that whenever two space cells touch and share an $(n-1)$-dimensional part of their boundary in primal space then their node representations are linked by an edge. The dual geometry representation of a space layer (quadrant IV) is the Euclidean space embedding of the dual graph which associates each node with a point and each edge with a simple space curve joining the points of the incident nodes. The space curves may be the result of a skeletonization process of the primal space geometries but are not restricted to this.

It follows that both the pairing of the primal space representation of a space cell with a node in dual space (cf. chapter 3.1.1) and the graph-based conceptualization of a space layer are inherent consequences of the Poincaré duality in topology space (quadrant II and III). The structured space model hence falls into the category of dual-graph-based approaches (cf. chapter 2.2.3). Like with complementary proposals from (Lee 2001), (Jensen et al. 2009), and (Boguslawski & Gold 2011), the dual graph is an *adjacency graph* that encodes the topological adjacency relationships between space cells in primal space. The Euclidean space embedding adds metric information to the purely topological graph and thus yields a metric graph or, equivalently, a *geometric network* (quadrant IV). Both graph structures (quadrant III and IV) are isomorphic due to the one-to-one correspondence between their node and edge sets. Both (Lee 2001) and (Jensen et al. 2009) also differ between the topological dimension of the dual graph and its metric embedding. However, (Lee 2001) obtains the geometric network from skeletonization processes only and allows additional nodes and edges rendering the topological and metric graph structures non-isomorphic, whereas the geometric network as proposed by (Jensen et al. 2009) follows from semantic rules and thus does not preserve the dual graph structure either. In contrast, the approach of (Boguslawski & Gold 2011) neglects the topological graph representation and only models its metric embedding with the edges of the metric dual graph being limited to straight line segments.

The dual space representation of a space layer is unique for a given configuration of space cells. Due to the space cells being mutually non-overlapping and jointly exhaustive, moving from one space cell to another on the same space layer requires crossing their shared boundary in primal space. Equivalently, each edge of the dual graph denotes such a movement and essentially describes the transition between the states of physically being in either space cell which are represented by the incident nodes. The dual graph is therefore also to be seen as a *state diagram*. A moving person or object is exactly in the spatial region covered by one space cell of the layer at a given point in time. Thus, only the corresponding state within the state diagram is *active* at that point in time. The size of the active space cell in primal geometry space is a measure for the uncertainty of the absolute position within Euclidean space which obviously gains in precision with smaller cell sizes.

The conceptual model of space layers provides a generic approach for the modelling of indoor space which is independent from the underlying notion of space and allows for the consistent derivation of adjacency relationships between space cells as well as the common interpretation of states and transitions. A purely geometric description of indoor space (quadrant I) already suffices to derive the further space representations in a deterministic way.

3.1.2.3 Example Topographic and Sensor Space Layer

Before elaborating on the mathematical formalization, the conceptual modelling of space layers is exemplified for a simple indoor space setting as sketched in figure 43. The indoor scene shows two rooms and a corridor as well as the generalized spherical signal coverage areas of two Wi-Fi transmitters equipped in the rooms. The propagation areas and architectural entities partly occupy the same indoor space, and thus their spatial representations overlap.

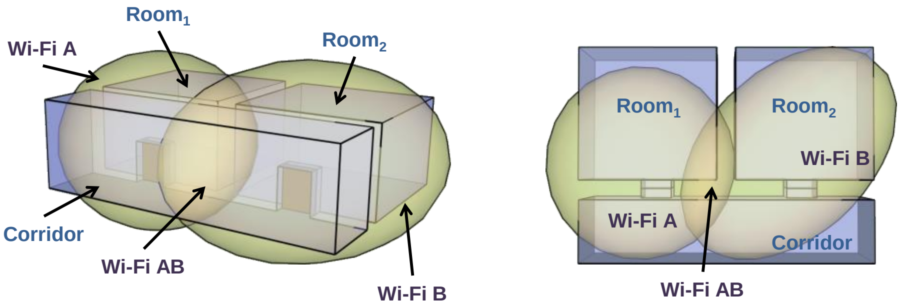


Figure 43: Example 3*-dimensional indoor scene in a slanted view (left) and from above (right).*

For the structured space model, each notion of space is modelled on a separate space layer. The topographic space layer reflecting the interior built environment is depicted in figure 44. The space partitioning along the architectural entities results in three topographic space cells each of which captures the 3-dimensional navigable space inside a room respectively the corridor in primal space. The room cells share a common boundary surface representing a door with the corridor cell but are not connected to each other. The space surrounding the architectural entities is the *outer space* of this topographic space layer. In dual space, the Poincaré duality carries each space cell as well as the outer space to a separate dual node. Since both rooms are adjacent to the corridor, their dual nodes are linked to the node representing the corridor. Likewise, the dual nodes of all space cells are linked to the outer node (depicted as dashed edges) as each space cell has a common boundary with the outer space. The resulting dual topological graph is shown in the middle of figure 44. The right part illustrates a possible Euclidean space embedding of this graph.

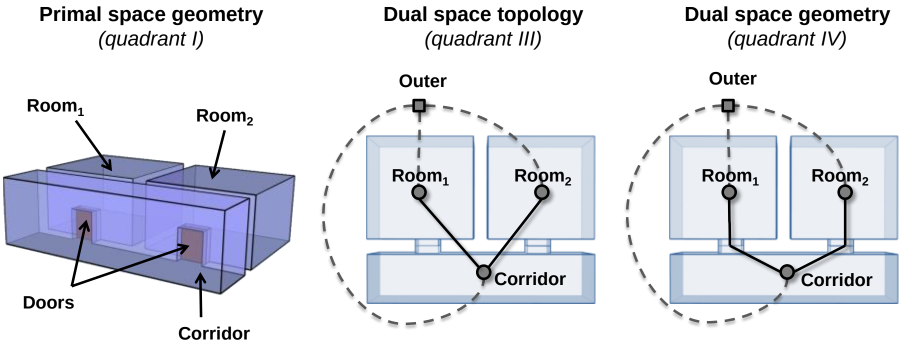


Figure 44: Primal space geometry of the topographic space layer (left), the resulting dual graph (middle), and a possible Euclidean space embedding (right). The formal derivation of the dual graph is elaborated in chapter 3.1.2.4.

The dual graph of the topographic space layer provides a combinatorial model of the interior built environment which conforms to the intuitive understanding of the indoor scene. It is suitable for path planning based on its encoding of topographic regions and their adjacency. Further addressing criteria such as room numbers may be introduced as thematic attributes on each space cell. The Euclidean space embedding results in a geometric network of the topographic space. The metric information adds quantitative properties such as distance or angle and thus enables, for example, best path calculations. The space curves associated with the edges of the geometric network facilitate the provision of geometric trajectories which more naturally follow the layout of the building. For example, such space curves may denote precise traveling paths for mobile robots. The point geometry associated with a node has to lie within the 3-dimensional geometry of the space cell. Often centroid representations are proposed but this is not enforced by the structured space model. The primal space geometry of a topographic space layer facilitates the visualization of the interior built environment and thus supports the visual guidance of moving persons along their route.

In figure 45, the different space representations of the topographic space layer are arranged in the four quadrants of the geometric-topological model shown in figure 42. Further partitioning schemas for this example indoor setting are possible and lead to a separate topographic space layer each.

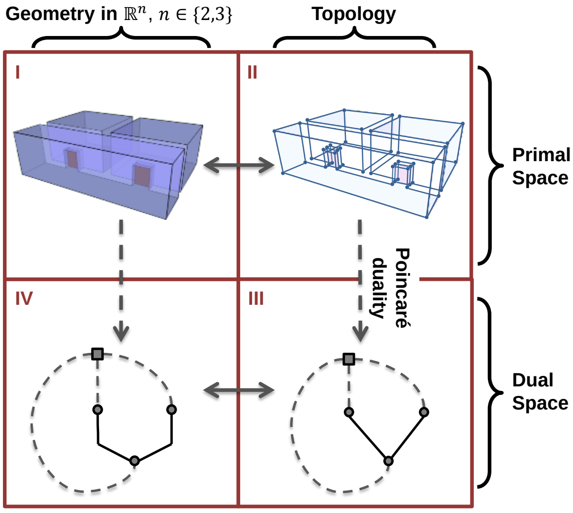


Figure 45: The four distinct space representations of the example topographic space layer.

The alternative view on indoor space resulting from its partitioning along the signal coverage of the two Wi-Fi transmitters (called A and B in the following) is illustrated in figure 46 and captured on a complementary sensor space layer. The overlapping coverage areas have to be mapped by three non-overlapping space cells each representing a region of homogeneous signal reception (only A, only B, both A and B). The surrounding outer space denotes regions without Wi-Fi reception. The primal space representation of the sensor space layer is depicted in the left part of figure 46. The three separate space cells are shown in an exploded view in the middle of the figure, and the resulting dual graph is shown on the right. The geometric network is not represented separately.

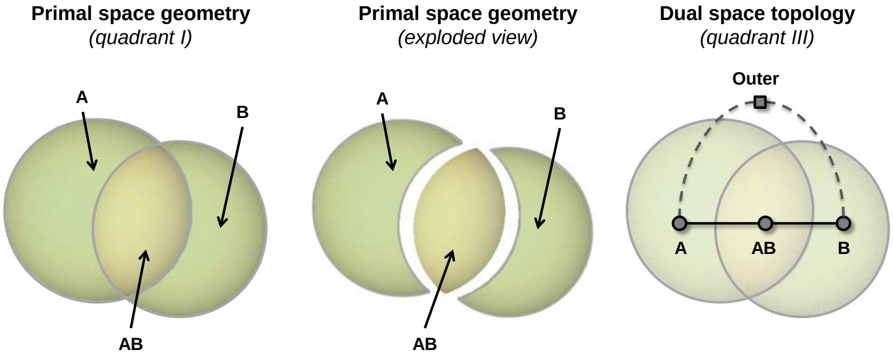


Figure 46: Primal space geometry of the Wi-Fi sensor space layer (left), exploded view of the three space cells in primal space (middle), and the resulting dual graph (right).

For this sensor space layer, the dual graph denotes the possible transitions between the signal coverage areas of both transmitters as well as the loss of signal reception. It hence enables the tracking of a moving person or object and the localization in the vicinity of a transmitter. Overlapping coverage areas generally help to increase the precision as they result in space cells of smaller size. The geometric network representation of a sensor space layer may account for the sensor deployment within indoor space by associating nodes with the fixed absolute location of sensors, and also adds quantification of distance to the adjacency information of the topological graph. The Wi-Fi space cells are arranged according to the geometric-topological representation schema of space layers in figure 47.

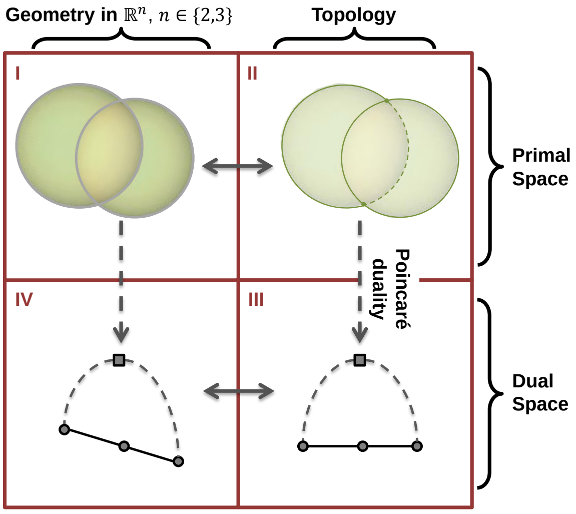


Figure 47: The four distinct space representations of the example Wi-Fi sensor space layer.

The combination of both space layers in an integrated view that simultaneously facilitates path planning and localization is the topic of chapter 3.2.

3.1.2.4 Mathematical Formalization

The conceptual model of space layers is set in a mathematical framework in this section. The formalization builds upon the definition of a space cell as provided in chapter 3.1.1.3, and additionally develops formal models for a *space cell complex*, for the *outer space cell*, and for *boundary cells* which are presented in the following.

Space cell complex. A *space cell complex* aggregates a set of non-overlapping space cells to a larger space. It hence captures the spatial configuration of space cells according to a given notion of space and partitioning schema.

Definition 3.7 (Space cell complex). A *space cell complex* $\mathcal{C} = \{S_\alpha\}_{\alpha \in I_\mathcal{C}}$ is a set of finitely many, pairwise non-overlapping n-dimensional space cells S_α with $S_i, S_j \in \mathcal{C}$: $Int(GM(S_i)) \cap Int\left(GM(S_j)\right) = \emptyset$, $i \neq j$ and $2 \leq n \leq 3$. A space cell complex may be an empty set and thus $|\mathcal{C}| \geq 0$.

Since all space cells in $\mathcal{C}$ share the same dimension n, the space cell complex itself is said to be n-dimensional. The geometric-topological representation of a space cell complex in primal space is derived from its contained space cells and is defined as follows.

Definition 3.8 (Primal space geometry of a space cell complex). The *primal space geometry* $GM(\mathcal{C})$ of an n-dimensional space cell complex $\mathcal{C}$ is the union $GM(\mathcal{C}) = \bigcup_{\alpha \in I_\mathcal{C}} GM(S_\alpha) \subset \mathbb{R}^n$.

Definition 3.9 (Primal space topology of a space cell complex). Let X be the topological space rendered by $GM(\mathcal{C})$ of an n-dimensional space cell complex $\mathcal{C}$, and equivalently $X = \bigcup_{\alpha \in I_\mathcal{C}} TP(S_\alpha)$. Then the *primal space topology* $TP(\mathcal{C})$ is a finite n-dimensional CW complex $(X, \mathcal{E})$ topologically embedded in $\mathbb{R}^n$ with $GM(\mathcal{C})$ being its geometric carrier. The cell decomposition $\mathcal{E}$ follows from inductively attaching the n-cells in the primal topology space description TP of all space cells in $\mathcal{C}$ along their common boundaries. Thus, for each $S \in \mathcal{C}$ there exists a continuous *cellular map* $\varphi_S: TP(S) \to TP(\mathcal{C})$ that, according to definition A.54, carries the k-skeleton of $TP(S)$

into the k-skeleton of $TP(\mathcal{C})$, with $TP(S)_k \subseteq TP(\mathcal{C})_k$, $0 \leq k \leq n$, and whose restriction to each open cell is a homeomorphism.

The following figure 48 presents a simple example of a space cell complex. On the left of figure 48, a 2-dimensional space cell in primal space is shown which is assumed to describe a room within a building. The aggregation of four such room cells (R_1 to R_4) to a space cell complex representing an excerpt of a building floor plan is sketched on the right figure 48.

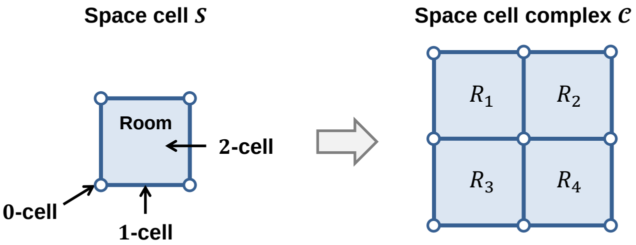


Figure 48: A single 2*-dimensional space cell representing a room (left) and an aggregation of four room cells to a space cell complex (right).*

In the above example, the primal space geometry $GM(S)$ of each room cell S is a rectangular region in the Euclidean plane $\mathbb{R}^2$. This geometric object is homeomorphic to a closed disk $\overline{\mathbb{B}}^2$ and thus satisfies condition (i) of definition 3.2. The primal space topology $TP(S)$ is a CW pair $(\overline{\mathbb{B}}^2, \delta)$ with δ decomposing $\overline{\mathbb{B}}^2$ into a single 2-cell, four 1-cells, and four 0-cells (cf. left part of figure 48). The primal space geometry $GM(\mathcal{C})$ of the space cell complex $\mathcal{C}$ on the right of figure 48 is the union of the primal geometries of the four space cells and again renders a rectangular region in $\mathbb{R}^2$ being homeomorphic to $\overline{\mathbb{B}}^2$. Its primal space topology $TP(\mathcal{C})$ decomposes $\overline{\mathbb{B}}^2$ into four 2-cells, twelve 1-cells, and nine 0-cells and results from gluing the 2-cells in the CW decompositions of the single rooms along their common boundaries.

For the remainder of this chapter, the geometric configuration of $GM(\mathcal{C})$ of a space cell complex $\mathcal{C}$ is assumed to be a topological n-manifold M that is homeomorphic to an orientable compact surface with boundary for $n = 2$, or to a k-shell manifold solid otherwise. At least for topographic space this restriction conforms to an intuitive understanding of the spatial configuration of space cells. In contrast to a single space cell, a space cell complex itself needs not be connected but may have k disjoint connected components with $1 \leq k \leq |\mathcal{C}|$. This accounts for the fact that the space cells in $\mathcal{C}$ only need to be mutually non-overlapping which includes spatially disjoint space cells. M is orientable and can be oriented through a consistent choice of orientation for each of its components (e.g., Mäntylä 1988). The discussion of non-manifold configurations of space cells is postponed to chapter 3.1.3.3.

If ∂M is connected then, according to the Jordan-Brouwer separation theorem (cf. theorem A.90), it separates $\mathbb{R}^n$ into $Int(M)$ and an unbounded component B with $\mathbb{R}^n \backslash \partial M = Int(M) \cup B$ and $Int(M) \cap B = \emptyset$. The component B itself can be viewed as a connected non-compact n-dimensional manifold U with $Int(U) = B$ and ∂M being its manifold boundary. The same holds in case M has k connected components and only differs in the fact that U has k boundary components but remains connected. Depending on whether U is a 2-dimensional or 3-dimensional manifold, it is also called *universal face* or *universal solid* in literature (e.g., ISO 19107:2003). (Gröger & Plümer 2011a) refer to U as *Out face* in the context of 2-dimensional respectively 2.8-dimensional maps. Since U represents the complement space of the space cell complex in $\mathbb{R}^n$, it obviously follows that each point in $\mathbb{R}^n$ can be identified to lie within one of the disjoint sets $Int(M)$, $Int(U)$, and ∂M (equivalently ∂U), which thus are jointly exhaustive with respect to $\mathbb{R}^n$.

The sets $Int(M)$, $Int(U)$, and $\partial M = \partial U$ are exemplified below for the 2-dimensional space cell complex $\mathcal{C}$ introduced in figure 48. The unbounded universal face $\mathbb{R}^2 \setminus Int(M)$ with $M \cong GM(\mathcal{C})$ is depicted as hatched area surrounding the four rooms.

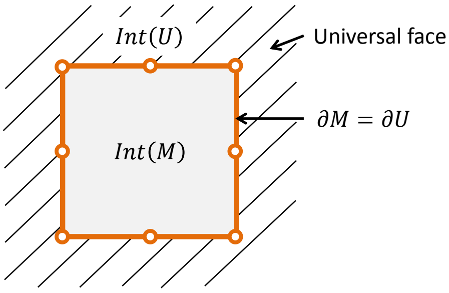


Figure 49: The interior, boundary, and exterior sets of the 2*-dimensional space cell complex from figure 48.*

A space cell complex is feasible to describe the configuration of space cells on a space layer in primal space (quadrants I and II in figure 42). The universal manifold complements the required tiling of Euclidean space $\mathbb{R}^n$. We wish to employ the Poincaré duality as formal tool from algebraic topology to derive the dual representation of space cells and the dual adjacency graph of the space layer in topology space (quadrant III in figure 41 and figure 42).

Compactification of the ambient space $\mathbb{R}^n$. As stated in theorem 2.1, the Poincaré duality is defined for orientable closed n-manifolds. However, the primal topological space rendered by a single space cell or a space cell complex does not meet this requirement. A closed manifold per definition A.44 is a compact manifold without boundary. Although both a space cell and a space cell complex are closed and bounded subsets of $\mathbb{R}^n$ which makes them compact according to the Heine-Borel theorem (cf. theorem A.30), each has a manifold boundary (cf. figure 48). Omitting the boundary yields bounded but open subsets of $\mathbb{R}^n$ which are non-compact. While the ambient Euclidean space $\mathbb{R}^n$ is an n-manifold without boundary, it is topologically unbounded and therefore also non-compact. These boundaryless but non-compact examples are open manifolds and thus do not satisfy the Poincaré duality. It is important to note that the spatial configuration of primal cells in the models proposed by (Lee 2001), (Jensen et al. 2009), and (Boguslawski & Gold 2011) equivalently renders closed and bounded subsets of Euclidean space which can be essentially mapped onto a 2-dimensional or 3-dimensional space cell complex. This consequently means that the cell configurations do not admit a Poincaré duality transformation in the sense of theorem 2.1 either. However, this issue is not explicitly addressed in any of these approaches.

The spatial layout of space cells as well as their arrangement in a space cell complex reflects physical or logical facts about the indoor environment and thus cannot be changed without interfering with these facts. This only leaves the non-compact ambient space $\mathbb{R}^n$ of the space cell complex for resolution. In order to retrieve a closed manifold satisfying the Poincaré duality, $\mathbb{R}^n$ has to be *compactified*. In general point-set topology, a compactification is the result of making a non-compact topological space X compact by embedding it in a compact space (cf. definition A.31) which opens up the space for the additional properties and structures of compact spaces. The compactified space is commonly denoted $\hat{X}$. The smallest compactification is the *Alexandroff* or *one-point compactification* (cf. definition A.32 and appendix A.1 for its formal discussion). Its idea is to adjoin a single point to the non-compact space such that the resulting space is compact. This extra single point is commonly called *point at infinity* and denoted $\{\infty\}$. For example, the real line $(-\infty, +\infty) = \mathbb{R}^1$ with the usual topology can be intuitively compactified by bending its opposite ends stretching to negative and positive infinity towards each other and adding a point at (unsigned) infinity which results in a compact circle being homeomorphic to the 1-sphere $\mathbb{S}^1$ (cf. figure 265). The same procedure can be applied to $\mathbb{R}^2$ and yields a compact space homeomorphic to $\mathbb{S}^2$ (cf. left part of figure 39).

Embedding of $TP(\mathcal{C})$ in the one-point compactification of $\mathbb{R}^n$. In general, the one-point compactification of Euclidean n-space $\mathbb{R}^n$ is homeomorphic to the n-sphere $\mathbb{S}^n$ (cf. proposition A.33). Since $\mathbb{S}^n$ is an orientable, compact and boundaryless manifold it meets the requirements of the Poincaré duality. In order to apply the Poincaré duality to an n-dimensional space cell complex $\mathcal{C}$ it therefore has to be topologically embedded in the sphere $\mathbb{S}^n$.

Definition 3.10 (Topological embedding of a space cell complex in $\mathbb{S}^n$). Let M denote the underlying topological space of an n-dimensional space complex $\mathcal{C}$ in $\mathbb{R}^n$. Then an embedding of M in the n-sphere $\mathbb{S}^n$ is given by the both injective and continuous map $f_E: M \rightarrow \mathbb{S}^n$ with $M \cong f_E(M) \subset \mathbb{S}^n$.

Let $(M, \mathcal{E})$ be the corresponding CW pair of the primal space topology $TP(\mathcal{C})$ with its CW decomposition $\mathcal{E}$ reflecting the configuration of space cells in $\mathcal{C}$. This combinatorial structure of M is preserved under f_E. The restriction $f_E|_e: e \to \mathbb{S}^n$ of f_E to a single k-cell $e \in \mathcal{E}$ results in an equivalent CW decomposition $\mathcal{E}'$ of $f_E(M)$. The topological cells in either structure are in one-to-one correspondence and the image of $f_E|_e$ for e is homeomorphic to e, i.e. $f_E|_e(e) \; \cong e$. Thus, the topological properties of the CW complex $(M, \mathcal{E})$ are invariant under f_E and the resulting CW pair $(f_E(M), \mathcal{E}')$ is topologically isomorphic to $TP(\mathcal{C})$.

The Jordan-Brouwer separation theorem has a natural extension by replacing $\mathbb{R}^n$ with its one-point compactification $\mathbb{S}^n$. Correspondingly, if $f_E(\partial M)$ is connected it separates $\mathbb{S}^n$ into two disjoint components B_1 and B_2 with $\mathbb{S}^n \backslash f_E(\partial M) = B_1 \cup B_2$ and $B_1 \cap B_2 = \emptyset$. Let $B_1 = f_E(Int(M))$, then the complement space of $f_E(M)$ is given by B_2. On the n-sphere, the complement space is a connected, compact n-manifold with boundary whose boundary is $f_E(\partial M)$. It therefore is a compactification $\widehat{U}$ of the universal face respectively the universal solid depending on its dimension.

The topological embedding $f_E: M \to \mathbb{S}^2$ of the example 2-dimensional space cell complex $\mathcal{C}$ from figure 48 in the one-point compactification of $\mathbb{R}^2$ is shown in figure 50. For one 0-cell e_0, one 1-cell e_1, and one 2-cell e_2 of the primal space topology $TP(\mathcal{C})$ the homeomorphic images $e_0' = f_E|_e(e_0)$, $e_1' = f_E|_e(e_1)$, and $e_2' = f_E|_e(e_2)$ are highlighted in order to illustrate the structure-preserving nature of f_E.

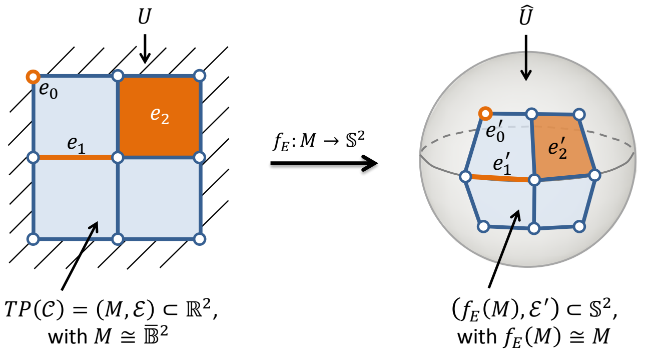

U
e_0
e_2
e_1
$f_E: M \to \mathbb{S}^2$
$\widehat{U}$
e_0'
e_2'
e_1'
$TP(\mathcal{C}) = (M, \mathcal{E}) \subset \mathbb{R}^2$,
with $M \cong \overline{\mathbb{B}}^2$
$(f_E(M), \mathcal{E}') \subset \mathbb{S}^2$,
with $f_E(M) \cong M$

Figure 50: Topological embedding of the primal space topology $TP(\mathcal{C})$ of the 2-dimensional space cell complex from figure 48 on the 2-sphere $\mathbb{S}^2$.

Outer space cell. Taking these facts as basis, the concept of *outer* as discussed in chapter 3.1.2.1 can now be formally defined for the structured space model.

Definition 3.11 (Outer space cell). The *outer space cell* S_{out} is an n-dimensional space cell $S_{out} = (GM, TP, v_{TP}, v_{GM}, A)$ where GM and TP are the geometric respectively topological descriptions of S_{out} in primal space, v_{GM} and v_{TP} are the geometric respectively topological descriptions of S_{out} in dual space, and A is a set of symbolic or semantic attributes associated with S_{out} itself.

Definition 3.12 (Primal space geometry of the outer space cell). Let $\mathcal{C}$ be an n-dimensional space cell complex. Then the *primal space geometry* $GM(S_{out})$ of the outer space cell S_{out} is the complement in $\mathbb{R}^n$ of all primal space cell geometries contained in $\mathcal{C}$, and thus $GM(S_{out}) = \mathbb{R}^n \setminus Int(\bigcup_{\alpha \in I_\mathcal{C}} GM(S_\alpha)) = \mathbb{R}^n \setminus Int(GM(\mathcal{C}))$. The primal space geometry is required to form a connected space.

Definition 3.13 (Primal space topology of the outer space cell). Let $\mathcal{C}$ be an n-dimensional space cell complex and $f_E: M \to \mathbb{S}^n$ an embedding of its underlying topological space M in the n-sphere $\mathbb{S}^n$ according to definition 3.10. Then the *primal space topology* $TP(S_{out})$ of the outer space cell S_{out} is a finite n-dimensional CW complex $(\widehat{U}, \delta)$ of the compact space $\widehat{U} = \mathbb{S}^n \setminus f_E(Int(M))$ which is required to be connected and whose topological boundary is $f_E(\partial M)$. The CW decomposition δ of $\widehat{U}$ is restricted to contain a single n-cell which is attached to all lower dimensional cells in δ. The subcomplex $A \subset TP(S_{out})$ decomposing the boundary $f_E(\partial M)$ has to be equal to the corresponding subcomplex $B \subset (f_E(M), \mathcal{E}')$.

It follows that the primal space representation of the outer space cell S_{out} depends on a given space layer complex $\mathcal{C}$. S_{out} differs from the notion of a space cell in that $GM(S_{out})$ needs not be manifold (cf. conditions (i) and (ii) of definition 3.2). However, S_{out} is likewise defined to be connected in primal space in order to ensure that it can be decomposed into a single n-cell by $TP(S_{out})$ and thus be mapped onto a single 0-cell in dual topology space.[15] Note that whereas $GM(S_{out})$ is a subset of $\mathbb{R}^n$, the ambient space of $TP(S_{out})$ is $\mathbb{S}^n$. Nevertheless, $GM(S_{out})$ serves as geometric carrier for $TP(S_{out})$. The definitions of the dual space geometry and topology of S_{out} as well as its set of attributes are identical to those of a space cell and hence are not repeated here.

In figure 51, both $GM(S_{out})$ and $TP(S_{out})$ are depicted for the 2-dimensional space cell complex $\mathcal{C}$ from figure 48. Since the topological space M underlying $TP(\mathcal{C})$ is homeomorphic to $\overline{\mathbb{B}}^2$ in this example, the compactified universal face $\widehat{U}$ underlying $TP(S_{out})$ results from removing the interior $Int(f_E(M)) \cong \mathbb{B}^2$ from $\mathbb{S}^2$ which is topologically equivalent to cutting a hole in the 2-sphere. Thus, $\widehat{U}$ itself is homeomorphic to $\overline{\mathbb{B}}^2$ and $TP(S_{out})$ is a CW complex $(\overline{\mathbb{B}}^2, \delta)$ with a single 2-cell as well as those eight 0-cells and eight 1-cells that lie on the common boundary $f_E(\partial M)$ of $\widehat{U}$ and $f_E(M)$.

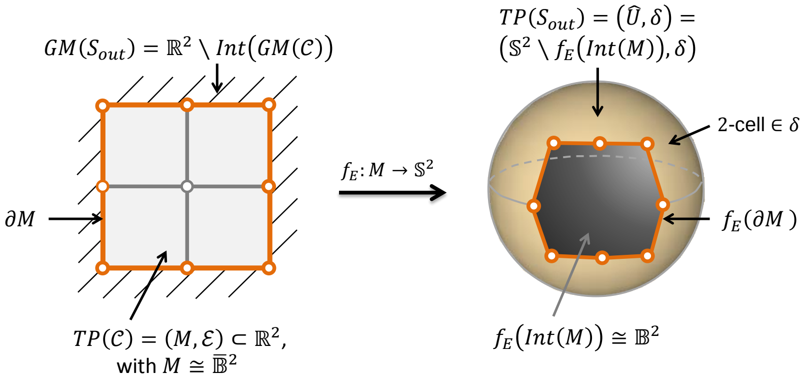

$GM(S_{out}) = \mathbb{R}^2 \setminus Int(GM(\mathcal{C}))$
∂M
$TP(\mathcal{C}) = (M, \mathcal{E}) \subset \mathbb{R}^2$,
with $M \cong \overline{\mathbb{B}}^2$
$f_E: M \rightarrow \mathbb{S}^2$
$TP(S_{out}) = (\widehat{U}, \delta) =$
$(\mathbb{S}^2 \setminus f_E(Int(M)), \delta)$
2-cell ∈ δ
$f_E(\partial M)$
$f_E(Int(M)) \cong \mathbb{B}^2$

Figure 51: Primal space geometry (left) and topology (right) of the outer space cell S_{out} associated with the 2-dimensional space cell complex from figure 48.

Duality transformation of a space cell complex. It follows from the above definitions that the union of $\widehat{U}$ with the topological embedding $f_E(M)$ of $\mathcal{C}$ results in $\mathbb{S}^n$. A finite CW decomposition $\hat{\mathcal{E}}$ of $\mathbb{S}^n$ which satisfies the Poincaré duality can therefore be obtained from gluing the cells in $TP(S_{out})$ to the cells in the image of $TP(\mathcal{C})$ on the n-sphere.

Definition 3.14 (CW decomposition $\hat{\mathcal{E}}$ of $\mathbb{S}^n$). Let $(f_E(M), \mathcal{E}')$ be the CW pair denoting the cell decomposition of the topological embedding $f_E(M)$ of an n-dimensional space cell complex $\mathcal{C}$ in $\mathbb{S}^n$ with $TP(\mathcal{C}) \cong (f_E(M), \mathcal{E}')$. Then a finite CW decomposition $\hat{\mathcal{E}}$ of $\mathbb{S}^\text{n}$ results from attaching the cells in $TP(S_{out})$ to $\mathcal{E}'$ by identifying cells on the common boundary $f_E(\partial M)$ along the two cellular maps $\varphi: \mathcal{E}' \rightarrow \hat{\mathcal{E}}$ and $\sigma: TP(S_{out}) \rightarrow \hat{\mathcal{E}}$ whose restrictions to each open cell are required to be homeomorphisms.

This cell structure on $\mathbb{S}^n$ can then be translated into a dual cell structure.

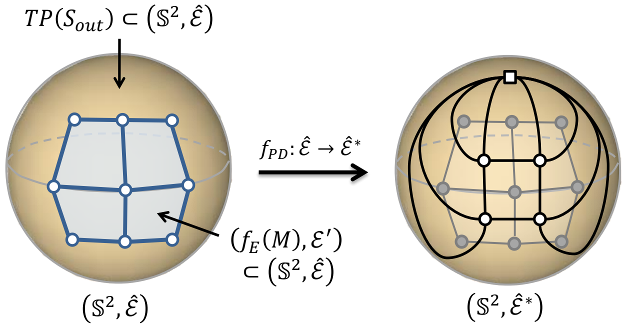
Definition 3.15 (Duality transformation of the CW decomposition $\hat{\mathcal{E}}$ of $\mathbb{S}^n$). Let $(\mathbb{S}^n, \hat{\mathcal{E}})$ be the CW complex as given by definition 3.14. The bijective map $f_{PD}: \hat{\mathcal{E}} \rightarrow \hat{\mathcal{E}}^*$ takes $\hat{\mathcal{E}}$ into the dual CW decomposition $\hat{\mathcal{E}}^*$ of $\mathbb{S}^n$ according to the Poincaré duality theorem 2.1.

As discussed in chapter 2.5.1.4, the image of f_{PD} for a k-cell e_k in $\hat{\mathcal{E}}$, with $0 \leq k \leq n$ and n being the dimension of the space cell complex, is an $(n-k)$-cell e_k^* in the dual decomposition $\hat{\mathcal{E}}^*$, and thus $\hat{\mathcal{E}}_k = \hat{\mathcal{E}}_{n-k}^*$. Consequently, the inverse image of e_k^* is given as $f_{PD}^{-1}(e_k^*) = e_k^{**} = e_k$. The bijective map f_{PD} can hence be interpreted as switch operator pairing both cell structures on $\mathbb{S}^n$ in primal and dual space.

The following figure illustrates the duality transformation for the embedding of the example space cell complex $\mathcal{C}$ from figure 48 on the 2-sphere $\mathbb{S}^2$. On the left, the CW complex $(\mathbb{S}^2, \hat{\mathcal{E}})$ is shown which decomposes $\mathbb{S}^2$ into five 2-cells, twelve 1-cells, and nine 0-cells. The right part of the figure shows the result of the Poincaré duality on

[15] A more detailed discussion on the implications of the connectedness condition of S_{out} together with supporting examples follows in chapter 3.1.3.

$(\mathbb{S}^2,\hat{\mathcal{E}})$. The five 2-cells in $\hat{\mathcal{E}}$ are mapped to five dual 0-cells in $\hat{\mathcal{E}}^*$, the twelve primal 1-cells result in twelve dual 1-cells, and the nine primal 0-cells are carried to nine dual 2-cells.


$TP(S_{out}) \subset (\mathbb{S}^2,\hat{\mathcal{E}})$
$f_{PD}\colon \hat{\mathcal{E}} \to \hat{\mathcal{E}}^*$
$(f_E(M),\mathcal{E}')$
$\subset (\mathbb{S}^2,\hat{\mathcal{E}})$
$(\mathbb{S}^2,\hat{\mathcal{E}})$
$(\mathbb{S}^2,\hat{\mathcal{E}}^*)$

$(\mathbb{S}^2,\hat{\mathcal{E}})$	$\xrightarrow{f_{PD}}$	$(\mathbb{S}^2,\hat{\mathcal{E}}^*)$
9 0-cells	→	**9** 2-cells
12 1-cells	→	**12** 1-cells
5 2-cells	→	**5** 0-cells

Figure 52: Poincaré duality transformation of the 2*-dimensional space cell complex from figure 48 on the* 2*-sphere* $\mathbb{S}^2$*. The primal cell decomposition is repeated on the right in dark grey. The square node represents the dual node of the outer space cell.*

The duality map f_{PD} works for the embedding of the primal space topology $TP(\mathcal{C})$ on $\mathbb{S}^n$. When additionally considering the map f_E and its restriction $f_E|_e$ then the pairing given by f_{PD} can be extended to $TP(\mathcal{C})$ itself which lives in $\mathbb{R}^n$.

Definition 3.16 (Duality transformation of a space cell complex). Let $(M,\mathcal{E})$ be the CW pair of the primal space topology $TP(\mathcal{C})$ of a space cell complex $\mathcal{C}$, and let $\hat{\mathcal{E}}$ and $\hat{\mathcal{E}}^*$ denote the primal and dual cell decomposition of $\mathbb{S}^n$ paired by f_{PD} as given in definition 3.15. Then the composition $g\colon f_E|_e \circ f_{PD}$ maps the cells in $\mathcal{E}$ onto their dual counterparts in $\hat{\mathcal{E}}^*$. Since $f_E|_e$ is a continuous injection from $\mathcal{E}$ to $\hat{\mathcal{E}}$, and f_{PD} is a continuous bijection between $\hat{\mathcal{E}}$ and $\hat{\mathcal{E}}^*$ the inverse $g^{-1}\colon (f_E|_e \circ f_{PD})^{-1}$ reverses this mapping.

The map g therefore pairs the cellular decompositions of a space cell complex in $\mathbb{R}^n$ and $\mathbb{S}^n$. This result allows for formally defining the consistent one-to-one relation between the n-dimensional topological description of a space cell in primal space and its mapping onto a single dual node as proposed by the geometric-topological representation schema for space cells introduced in chapter 3.1.1.2 (cf. quadrants II and III in figure 41).

Precisely, for a given n-dimensional space cell complex $\mathcal{C}$ and its associated outer space cell $S_{out}(\mathcal{C})$, we define a map $f_{TP}\colon \{TP(\mathcal{C}), TP(S_{out})\} \to \hat{\mathcal{E}}^*$ which takes the k-cells in the primal space descriptions $TP(\mathcal{C})$ respectively $TP(S_{out})$ to distinct $(n-k)$-cells in the dual cell decomposition $\hat{\mathcal{E}}^*$ of $\mathbb{S}^n$, with $0 \leq k \leq n$. The restriction $f_{TP}|_{e_n}\colon e_n \to \hat{\mathcal{E}}^*$ then takes the n-cell e_n in the primal topological description $TP(S)$ of a space cell $S \in \{\mathcal{C}, S_{out}\}$ to a distinct 0-cell in $\hat{\mathcal{E}}^*$ constituting $v_{TP}(S)$ as follows

$$v_{TP}(S) = f_{TP}|_{e_n}\left(Int\big(TP(S)\big)\right) = \begin{cases} g\left(Int\big(TP(S)\big)\right), & S \in \mathcal{C}; \\ f_{PD}\left(Int\big(TP(S)\big)\right), & S = S_{out}. \end{cases} \tag{3.17}$$

Since the n-cell in $TP(S_{out})$ is already contained in $\hat{\mathcal{E}}$ it suffices to consider f_{PD} instead of g for the outer space cell S_{out}. The reverse relation is denoted by

$$Int\big(TP(S)\big) = f_{TP}|_{e_n}{}^{-1}\big(v_{TP}(S)\big) = \begin{cases} g^{-1}\big(v_{TP}(S)\big), & S \in \mathcal{C}; \\ f_{PD}^{-1}\big(v_{TP}(S)\big), & S = S_{out}. \end{cases} \tag{3.18}$$

Figure 53 summarizes the results for the 2-dimensional building floor plan from figure 48. For the space cell R_4 as well as for the outer space cell S_{out} the one-to-one mapping between their representations in primal and dual topology space is exemplified.

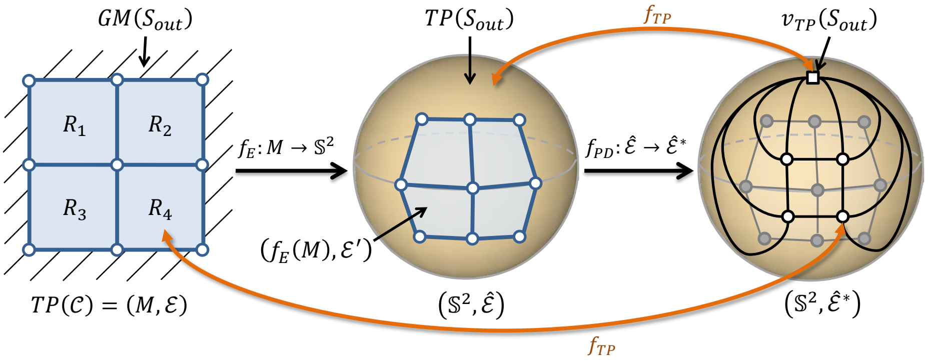

$GM(S_{out})$
$TP(S_{out})$
f_{TP}
$v_{TP}(S_{out})$
R_1
R_2
R_3
R_4
$f_E: M \to \mathbb{S}^2$
$f_{PD}: \hat{\mathcal{E}} \to \hat{\mathcal{E}}^*$
$(f_E(M), \mathcal{E}')$
$TP(\mathcal{C}) = (M, \mathcal{E})$
$(\mathbb{S}^2, \hat{\mathcal{E}})$
$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$
f_{TP}

Figure 53: One-to-one mapping between the primal and dual space topology of a space cell complex and its associated outer space cell.

The disjoint union $\bigcup_{\alpha \in I_{\mathcal{C}}} v_{TP}(S_\alpha) \cup v_{TP}(S_{out})$ is exactly the 0-skeleton $\hat{\mathcal{E}}_0^*$ in dual space. Each $(n-1)$-cell in the boundary $\partial(TP(S))$ of a space cell is equivalently send to a distinct 1-cell in $\hat{\mathcal{E}}^*$. As shown in chapter 2.5.1.4, the boundary map $\partial: \hat{\mathcal{E}}_n \to \hat{\mathcal{E}}_{n-1}$ in primal space becomes the coboundary map $\delta: \hat{\mathcal{E}}_0^* \to \hat{\mathcal{E}}_1^*$ in dual space. Thus, if two space cells share a common $(n-1)$-cell e_{n-1} on their boundary in $\hat{\mathcal{E}}$ then their dual nodes are bounding the corresponding 1-cell e_{n-1}^* in $\hat{\mathcal{E}}^*$. The dual 1-skeleton $\hat{\mathcal{E}}_1^*$ hence captures the adjacency relations between space cells in primal space and realizes the dual adjacency graph of a space layer (cf. quadrant III in figure 42).

Space layer. The machinery developed is used in the following to define a space layer of the structured space model.

Definition 3.19 (Space layer). A *space layer* L is a sextuple $(\mathcal{C}, S_{out}, G_{TP}, G_{GM}, \mathcal{B}, A)$ where $\mathcal{C}$ is an n-dimensional space cell complex in $\mathbb{R}^n$ with $2 \le n \le 3$ and $S_{out}(\mathcal{C})$ is the outer space cell associated with $\mathcal{C}$. Both $\mathcal{C}$ and $S_{out}(\mathcal{C})$ provide the primal representation of L in geometry and topology space. G_{TP} and G_{GM} are the graph-based conceptualizations of L in dual topology respectively geometry space. $\mathcal{B}$ is a set of boundary cells modelling the common boundary of space cells, and A is a set of symbolic or semantic attributes associated with L itself.

A space layer shares the same dimension as $\mathcal{C}$. Whereas both $\mathcal{C}$ and S_{out} populate the quadrants I and II of a space layer as depicted in figure 42, G_{TP} and G_{GM} correspond to quadrants III and IV respectively. The definition of the set of symbolic and semantic attributes A is identical to that of the attribute set of a space cell (cf. definition 3.6). Reasonable semantic attributes for a space layer are, for example, the underlying notion of space (e.g., topographic space, sensor space, logical space, etc.) and the applied partitioning schema (e.g., topographic space for pedestrians, sensor space representing Wi-Fi coverage, etc.).

Definition 3.20 (Dual space topology of a space layer, intra-layer graph). Let $(\mathbb{S}^n, \hat{\mathcal{E}}^*)$ be the duality transformation of the n-dimensional space complex $\mathcal{C}(L)$ of the space layer L according to definition 3.16. Then the *dual space topology* $G_{TP}(L) = (V, E)$ of L is a dual graph, equivalently called *intra-layer graph*, given as subset of the 1-skeleton $\hat{\mathcal{E}}_1^*$ so that

(i) $V(G_{TP})$ is constituted by the set of 0-cells e_0^* in $\hat{\mathcal{E}}_1^*$, and

(ii) $E(G_{TP})$ is the set of all 1-cells e_1^* in $\hat{\mathcal{E}}_1^*$ being incident to two distinct nodes in $V(G_{TP})$, and thus $E(G_{TP}) = \{e_1^* \in \hat{\mathcal{E}}_1^* \mid \partial e_1^* \cong \mathbb{S}^0\}$.

Obviously, the node set $V(G_{TP})$ can be equivalently rephrased to be the disjoint union of the dual topological representations of all space cells contained in the space cell complex $\mathcal{C}(L)$ including $S_{out}(L)$, and thus $V(G_{TP}) = \bigcup_{\alpha \in I_{\mathcal{C}(L)}} v_{TP}(S_\alpha) \cup v_{TP}(S_{out}(L))$.

The intra-layer graph resulting for the 2-dimensional building floor plan discussed in the previous examples of this section is sketched in figure 54. In this case, the intra-layer graph is equal to the 1-skeleton $\hat{\mathcal{E}}_1^*$ of the Poincaré transform of $(\mathbb{S}^n, \hat{\mathcal{E}})$ (cf. right of figure 53). Note that the intra-layer graph does not contain the dual cells from $\hat{\mathcal{E}}_1^*$ having dimension greater than 1. Thus, the initial CW decomposition $\hat{\mathcal{E}}$ of $\mathbb{S}^n$ in primal space (cf. middle of figure 53) cannot be reconstructed from the intra-layer graph anymore. This fact is however harmless in the context of the structured space model since only the one-to-one pairing between the primal n-cells and $(n-1)$-cells in $\hat{\mathcal{E}}$ with the dual 0-cells respectively 1-cells in $\hat{\mathcal{E}}_1^*$ is required.

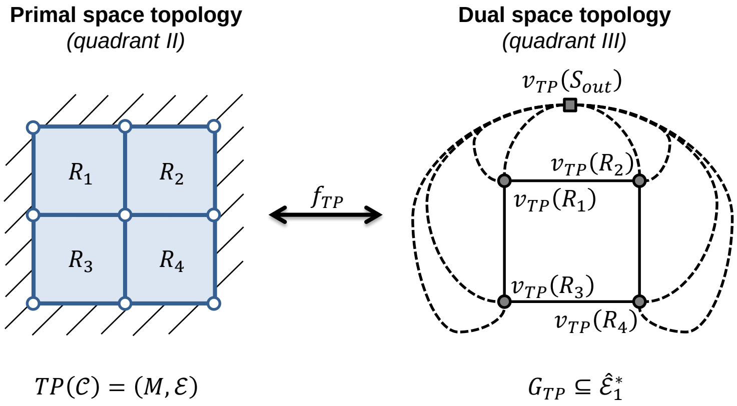


Figure 54: Intra-layer graph resulting for the 2*-dimensional building floor plan from figure 48.*

According to definition A.51 of a CW complex $(X, \mathcal{E})$, the cell boundary for each 1-cell $e_1 \in \mathcal{E}$ given by $\partial e_1 = \bar{e}_1 \setminus e_1$ is defined to be a continuous image of $\mathbb{S}^0$. If the 1-skeleton X_1 is understood as graph structure then this definition means that each edge can be incident to either one or two nodes. Condition (ii) of the above definition 3.20 of the intra-layer graph $G_{TP}(L)$ however restricts the boundary ∂e_1 to be a homeomorphic image of $\mathbb{S}^0$ and consequently requires all edges to link two different nodes $v_1, v_2 \in V(G_{TP}(L))$. This restriction is introduced due to semantic considerations rather than for mathematical reasons. The dual graph of a space layer L is meant to encode qualitative adjacency relationships between space cells in primal space which are to be evaluated in the context of navigation. A dual edge being incident to a single dual node would therefore carry the information that the corresponding space cell is adjacent to itself. This information is however implicitly true for every space cell, and thus cyclic edges can be omitted from $G_{TP}(L)$ without loss of information. The discussion of different spatial configurations of space cells in chapter 3.1.3 illustrates examples for which the Poincaré duality transformation necessarily yields dual edges linking a single dual node with itself which therefore fall under condition (ii) of definition 3.20.

Definition 3.21 (Dual space geometry of a space layer). The *dual space geometry* $G_{GM}(L)$ of a space layer L is the Euclidean space embedding of the dual graph $G_{TP}(L)$ given by the both continuous and injective map $f_{GM}: G_{TP} \rightarrow \mathbb{R}^n$ with n being the dimension of L. Every node in $V(G_{TP})$ is assigned a distinct point in $\mathbb{R}^n$ and every edge in $E(G_{TP})$ is associated with a space curve such that

(i) there exists a continuous map $\phi_e: \bar{\mathbb{B}}^1 \rightarrow \mathbb{R}^n$ for each $e \in E(G_{TP})$ that restricts to a homeomorphism from $Int(\bar{\mathbb{B}}^1)$ onto e and maps $\partial\bar{\mathbb{B}}^1$ onto $\bar{e} \setminus e$,

(ii) no space curve contains points associated with nodes not being incident to the corresponding edge, and

(iii) the intersection of the interior of two space curves is the empty set.

It follows that the restriction $f_{GM}|_v: V(G_{TP}) \rightarrow V(G_{GM})$ to the nodes in $V(G_{TP})$ is a bijection with any two nodes $n, m \in V(G_{TP})$ being adjacent if and only if $f_{GM}|_v(n)$ and $f_{GM}|_v(m)$ are adjacent in G_{GM}. Therefore, the graphs G_{TP} and G_{GM} are isomorphic, and $G_{TP} \simeq G_{GM}$. As stated in chapter 3.1.1.1, a space curve may result from a medial-axis transform of the primal space geometry of a space cell but arbitrary further embeddings are possible.

The embedding f_{GM} formally links the dual node representation of a space cell with its geometric representation in dual space as follows

$$\forall S \in \{\mathcal{C}(L), S_{out}(L)\}: v_{GM}(S) = f_{GM}|_v(v_{TP}(S)). \tag{3.22}$$

Since f_{GM} is a continuous injection, the reverse mapping also holds:

$$\forall S \in \{\mathcal{C}(L), S_{out}(L)\}: v_{TP}(S) = f_{GM}|_v{}^{-1}(v_{GM}(S)). \tag{3.23}$$

Boundary cells. The developed formalism assures that space cells are mapped onto distinct nodes in the dual graph structure of a space layer L. Thus, graph traversal algorithms can access the spatial and semantic properties of space cell S from its dual nodes in $G_{TP}(L)$ respectively $G_{GM}(L)$. In contrast, the dual edges as defined so far merely encode topological adjacency between space cells sharing a part of their boundary in primal space but are not associated with a formal model of this common boundary. However, the space cell boundary has been identified as an essential conceptual entity in its own right in chapter 3.1.1.1. For example, the bodily movement between topographic space cells requires that parts of the space cell boundary are intangible (e.g., the free space in walls due to doors or passages) whereas other parts might be tangible and hence constrain the movement (e.g., surfaces of walls). The transition between two places may additionally be hindered by geometric facts (e.g., a boundary being too narrow for wheelchair users) or logical restrictions (e.g., access constraints). Thus, the knowledge about navigable and non-navigable parts of the space cell boundary is a prerequisite for navigation queries and goes beyond pure topological adjacency. In order to make this spatial and semantic knowledge available within the structured space model, a formal model for a single *boundary cell* is introduced in the following. This model obviously needs to be strongly coupled with the definitions introduced so far in order to be consistent.

Definition 3.24 (Boundary cell). A *boundary cell* B is a quintuple $(GM, TP, e_{TP}, e_{GM}, A)$ where GM and TP are the geometric respectively topological descriptions of B in primal space, e_{TP} and e_{GM} are the topological respectively geometric descriptions of B in dual space, and A is a set of symbolic or semantic attributes associated with B itself.

Definition 3.25 (Primal space topology of a boundary cell). Let L be an n-dimensional space layer and $\mathcal{C}(L)$ its space cell complex. Further, let $TP\big(\mathcal{C}(L)\big) = (X, \mathcal{E})$ be the finite CW complex of dimension n that renders the primal space topology of $\mathcal{C}(L)$ (cf. definition 3.9). Then the *primal space topology* $TP(B)$ of a boundary cell B is a finite subcomplex of the $(n-1)$-skeleton $\mathcal{E}_{n-1}$ and contains a single $(n-1)$-cell e_{n-1} so that $TP(B) = \bar{e}_{n-1} \subset \mathcal{E}_{n-1}$. Moreover, $TP(B)$ has to satisfy one of the two following conditions:

(i) The coboundary of e_{n-1} contains precisely two n-cells $e_n^1, e_n^2 \in \mathcal{E}$ with $e_n^1 \neq e_n^2$, or

(ii) $TP(B)$ is a subset of the boundary of $TP\big(\mathcal{C}(L)\big)$, and thus $TP(B) \subseteq \partial TP\big(\mathcal{C}(L)\big)$.

Definition 3.26 (Primal space geometry of a boundary cell). Let L be an n-dimensional space layer and $\mathcal{C}(L)$ its space cell complex. Then the *primal space geometry* $GM(B)$ of a boundary cell B is the $(n-1)$-dimensional geometric carrier of $TP(B)$ in $GM(\mathcal{C}(L))$ with $GM(B) \subset GM(\mathcal{C}(L)) \subset \mathbb{R}^n$.

A boundary cell B on an n-dimensional space layer L is said to be $(n-1)$-dimensional. The conditions (i) and (ii) of definition 3.25 ensure that a boundary cell represents (a part of) the common boundary of precisely two space cells $S_1, S_2 \in \{\mathcal{C}(L), S_{out}(L)\}$. They hence formulate the primal space equivalence of the condition (ii) of definition 3.20 according to which each dual edge in the intra-layer graph $G_{TP}(L)$ of L has to be incident to two distinct dual nodes. The primal space geometry $GM(B)$ necessarily is a subset of $\partial GM(S_1) \cap \partial GM(S_2)$ and provides a quantitative description for the common boundary of S_1 and S_2 which can be used, for example, to geometrically evaluate whether arbitrarily shaped users having a geometric description can physically move through this boundary part. (Yuan & Schneider 2011) propose a similar concept called *connector* surface for their 3-dimensional grid-based LEGO model (cf. chapter 2.2.1) and introduce geometric methods for checking the accessibility of connectors which can be equivalently transferred to the primal geometric description of boundary cells. However, a connector surface in the LEGO model is a purely geometric object and thus only carries the information provided by $GM(B)$.

Per definition 3.16, the $(n-1)$-cell in $TP(B)$ is send to a 1-cell in the dual CW complex $(\mathbb{S}^n, \hat{\mathcal{E}}^*)$. This dual 1-cell is exactly the dual topological space representation $e_{TP}(B)$ of the boundary cell. Formally, a bijective pairing between $TP(B)$ and $e_{TP}(B)$ can be expressed by restricting the injective map $f_{TP}: \{TP(\mathcal{C}), TP(S_{out})\} \to \hat{\mathcal{E}}^*$ introduced above to the domain of $(n-1)$-cells. Thus, $f_{TP}|_{e_{n-1}}: e_{n-1} \to \hat{\mathcal{E}}^*$ yields

$$e_{TP}(B) = f_{TP}|_{e_{n-1}}\Big(Int\big(TP(B)\big)\Big), \text{ and} \tag{3.27}$$

$$Int\big(TP(B)\big) = f_{TP}|_{e_{n-1}}{}^{-1}\big(e_{TP}(B)\big). \tag{3.28}$$

Due to conditions (i) and (ii) of definition 3.25, $e_{TP}(B)$ is a valid dual edge contained in the intra-layer graph $G_{TP}(L)$ of the corresponding space layer L. The dual space geometry $e_{GM}(B)$ of B then is the Euclidean space embedding of $e_{TP}(B)$ in $G_{GM}(L)$ according to the map $f_{GM}: G_{TP} \to \mathbb{R}^n$ given in definition 3.21 and its bijective restriction $f_{GM}|_e: E(G_{TP}) \to E(G_{GM})$ to the edges in $E(G_{TP})$. Formally,

$$e_{GM}(B) = f_{GM}|_e\big(e_{TP}(B)\big), \text{ and} \tag{3.29}$$

$$e_{TP}(B) = f_{GM}|_e^{-1}\big(e_{GM}(B)\big). \tag{3.30}$$

The set of symbolic and semantic attributes $A(B)$ of the boundary cell is identically defined as the attribute set of a space cell (cf. definition 3.6). Since the attributes are associated with the boundary cell which itself is mapped onto a dual edge in $G_{TP}(L)$ respectively $G_{GM}(L)$, they are available for graph traversal algorithms.

In figure 55, the model for boundary cells is illustrated along the example building floor plan from figure 48. On the left, the primal topological representations of two boundary cells B_1 and B_2 are highlighted which are given as 1-dimensional subcomplexes of the CW decomposition $TP(\mathcal{C})$ of the space cell complex $\mathcal{C}$. Whereas B_1 separates the room R_1 from the outer space and thus $TP(B_1)$ satisfies condition (ii) of definition 3.25, B_2 describes the common boundary of the rooms R_3 and R_4. Correspondingly, the 2-cells in $TP(R_3)$ and $TP(R_4)$ are in the coboundary of the 1-cell contained in $TP(B_1)$ as required by condition (i) of definition 3.25. The dual edges resulting for both boundary cells link the dual nodes of the corresponding space cells as shown in the dual graph depicted on the right. The boundary cells may be enriched with semantic information (e.g., whether they represent a wall or a passable boundary such as a door) which is then available from both space representations.

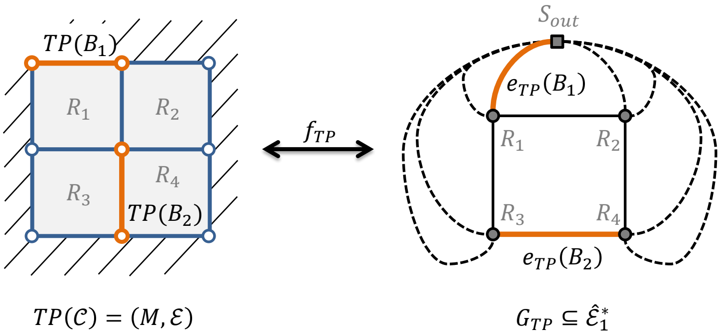


Figure 55: Two boundary cells B_1 and B_2 in primal topology space (left) and dual topology space (right). Note that the remaining ten boundary cells for the 2*-dimensional building floor plan from figure 48 are not illustrated for readability.*

Finally, a space layer is linked with the model of boundary cells in the following definition which concludes its mathematical formalization.

Definition 3.31 (Set of boundary cells of a space layer). The *set of boundary cells* $\mathcal{B}(L) = \{B_\alpha\}_{\alpha \in I_\mathcal{C}}$ of a space layer L is a set of finitely many boundary cells B_α with $B_i, B_j \in \mathcal{B}: Int\big(TP(B_i)\big) \neq Int\left(TP\big(B_j\big)\right), i \neq j$ and $|\mathcal{B}| = \big|E\big(G_{TP}(L)\big)\big|$.

According to this definition, every $(n-1)$-cell on the common boundary between two space cells $S_1, S_2 \in \{\mathcal{C}(L), S_{out}(L)\}$ is described by exactly one boundary cell $B \in \mathcal{B}(L)$. It necessarily follows that the number of boundary cells in $\mathcal{B}(L)$ is equal to the number of dual edges in $G_{TP}(L)$.

Minimal space layer. When assuming the space cell complex $\mathcal{C}$ of a space layer to be the empty set, the *minimal possible representation* of a space layer is retrieved.

Definition 3.32 (Minimal space layer). The *minimal n-dimensional space layer* L_{min} is a space layer whose space cell complex $\mathcal{C}(L_{min})$ is the empty set, and thus $\mathcal{C}(L_{min}) = GM\big(\mathcal{C}(L_{min})\big) = TP\big(\mathcal{C}(L_{min})\big) = \emptyset$. The outer space cell $S_{out}(L_{min})$ is then given in primal space by $GM(S_{out}(L_{min})) = \mathbb{R}^n$ and $TP(S_{out}(L_{min})) \cong \mathbb{S}^n$. By choosing a minimal CW decomposition of $\mathbb{S}^n$ (one 0-cell and one n-cell), it follows that $V\big(G_{TP}(L_{min})\big) = \{v_{TP}(S_{out})\}$ and $E(G_{TP}(L_{min})) = \emptyset$. Consequently, $\mathcal{B}(L_{min}) = \emptyset$. Finally, the minimal space layer does not carry semantic attributes, and thus $A(L_{min}) = \emptyset$.

Intuitively, the minimal space layer L_{min} of the structured space model is a representation of the boundaryless outer space in both primal and dual space. Any space layer reflecting an arbitrary indoor space partitioning can be built from L_{min} by iteratively adding space cells to $\mathcal{C}(L_{min})$.

This again emphasizes the need for an outer space complementing the spatial configuration of space cells in order to be able to apply the Poincaré duality transformation in the sense of theorem 2.1. From the alternative dual-graph-based approaches presented, only (Jensen et al. 2009) identify an *"exterior of the indoor space"* and represent this exterior by a single node in their dual graph. This representation generally conforms to the dual topology space of S_{out} within the structured space model given as $v_{TP}(S_{out}) = f_{PD}(Int(TP(S_{out})))$. However, (Jensen et al. 2009) neither present a formal definition for their notion of exterior nor for its node-based mapping. The model of (Boguslawski & Gold 2011) also introduces an *"exterior shell"* which encloses the cells representing the interior built environment. But the authors use this term to refer to the outer shell of a building which can be observed from outside the building. The outer building shell however can be described as bounded subset of Euclidean space and hence does not suffice to retrieve an orientable, closed manifold satisfying the Poincaré duality. Both in (Lee 2001) and the previous publications on the MLSEM, a consistent notion of outer as well as its impact on the applicability of the Poincaré duality is not discussed.

Weak intra-layer graph. If follows that the dual graphs presented by (Boguslawski & Gold 2011) and (Lee 2001) do not contain a dual node representing the outer space. In order to explain their results based on the mathematical realization of the structured space model, a *weak intra-layer graph* of a given space layer L is defined as follows.

Definition 3.33 (Weak intra-layer graph). Let $G_{TP}(L)$ be the intra-layer graph of a space layer L, and $S_{out}(L)$ the outer space cell associated with L. Then the *weak intra-layer graph* $G'_{TP}(L) = (V, E)$ is a subgraph of $G_{TP}(L)$ whose node set $V(G'_{TP})$ is a subset of $V(G_{TP})$ excluding the dual node of S_{out}, and thus $V(G'_{TP}) = V(G_{TP}) \backslash v_{TP}(S_{out})$.

The weak version of the intra-layer graph for the example building floor plan discussed above (cf. figure 54) is shown below. Note that a further pruning of graph elements, e.g. based on semantic criteria (e.g., Lee 2001 and Boguslawski & Gold 2011), is possible but out of scope of this thesis.

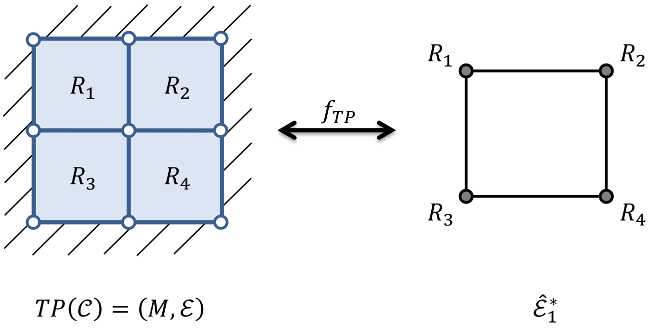


Figure 56: Weak intra-layer graph resulting for the 2-dimensional building floor plan from figure 48.

Summary. In order to visually relate the developed mathematical model with the geometric-topological representation schema for space cells and space layers, the most important mathematical facts are arranged according to the four quadrants in figure 57. The figure nicely illustrates the formal linkage between the separate space representations.

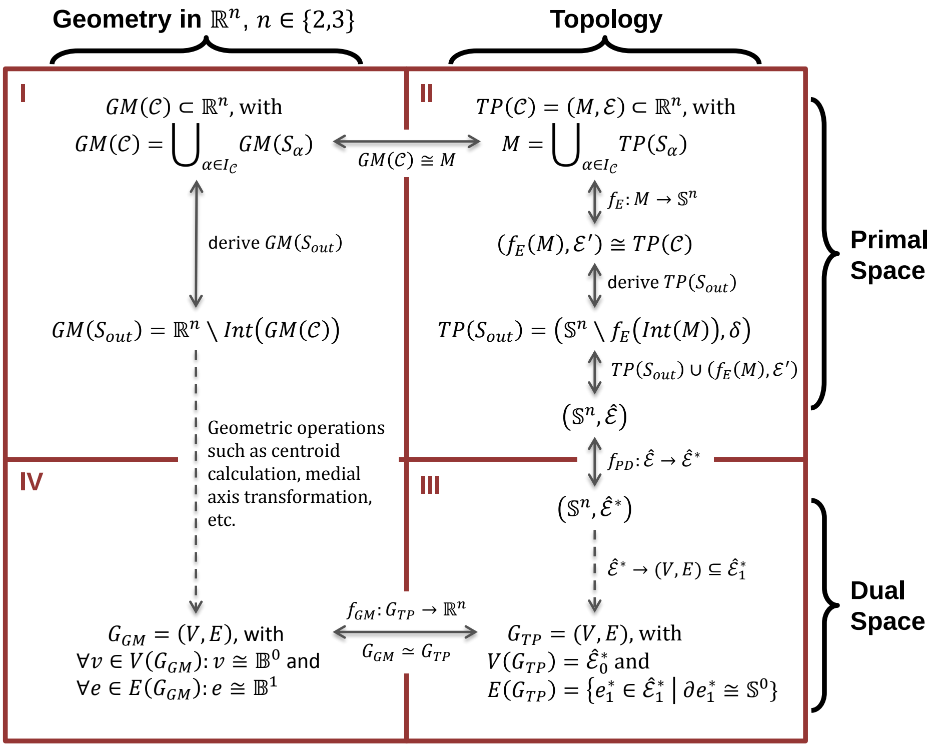

Geometry in $\mathbb{R}^n$, $n \in \{2,3\}$
Topology
I
$GM(\mathcal{C}) \subset \mathbb{R}^n$, with
$GM(\mathcal{C}) = \bigcup_{\alpha \in I_\mathcal{C}} GM(S_\alpha)$
II
$TP(\mathcal{C}) = (M, \mathcal{E}) \subset \mathbb{R}^n$, with
$M = \bigcup_{\alpha \in I_\mathcal{C}} TP(S_\alpha)$
$GM(\mathcal{C}) \cong M$
$f_E: M \to \mathbb{S}^n$
derive $GM(S_{out})$
$(f_E(M), \mathcal{E}') \cong TP(\mathcal{C})$
derive $TP(S_{out})$
$GM(S_{out}) = \mathbb{R}^n \setminus Int(GM(\mathcal{C}))$
$TP(S_{out}) = (\mathbb{S}^n \setminus f_E(Int(M)), \delta)$
$TP(S_{out}) \cup (f_E(M), \mathcal{E}')$
$(\mathbb{S}^n, \hat{\mathcal{E}})$
Primal Space
Geometric operations such as centroid calculation, medial axis transformation, etc.
IV
III
$f_{PD}: \hat{\mathcal{E}} \to \hat{\mathcal{E}}^*$
$(\mathbb{S}^n, \hat{\mathcal{E}}^*)$
$\hat{\mathcal{E}}^* \to (V, E) \subseteq \hat{\mathcal{E}}_1^*$
$f_{GM}: G_{TP} \to \mathbb{R}^n$
$G_{GM} \simeq G_{TP}$
$G_{GM} = (V, E)$, with
$\forall v \in V(G_{GM}): v \cong \mathbb{B}^0$ and
$\forall e \in E(G_{GM}): e \cong \mathbb{B}^1$
$G_{TP} = (V, E)$, with
$V(G_{TP}) = \hat{\mathcal{E}}_0^*$ and
$E(G_{TP}) = \{e_1^* \in \hat{\mathcal{E}}_1^* \mid \partial e_1^* \cong \mathbb{S}^0\}$
Dual Space

Figure 57: Summary of the mathematical model arranged in the four quadrants of the geometric-topological representation schema for space cells and layers. The dashed lines indicate mappings that are not reversible.

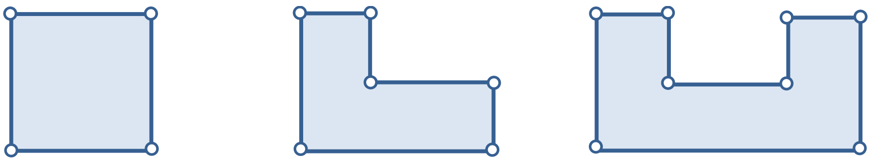
3.1.3 Discussion of Different Spatial Configurations of Space Cells

According to definition 3.3, the manifold space underlying the primal geometry representation $GM(S)$ of an n-dimensional space cell S must be decomposed by $TP(S)$ into exactly one topological n-cell and finitely many lower dimensional cells in primal topology space. This restriction to the CW decomposition of $TP(S)$ ensures that the Poincaré duality pairs each space cell with a single dual 0-cell captured by $v_{TP}(S)$. For the same reason, an equivalent condition has been expressed for the outer space cell S_{out} (cf. definition 3.13). In this chapter, different geometric layouts of space cells as well as different spatial configurations of space cells in a space cell complex are investigated with respect to their impact on the CW decompositions of the space cells including S_{out} and on the structure of the resulting dual graph. The discussion is split in the following into 2-dimensional respectively 3-dimensional manifold configurations of space cells as well as non-manifold settings.

3.1.3.1 Two-dimensional Manifold Configurations

In two dimensions, the primal space representation of space cells needs to be homeomorphic to a connected, orientable, compact surface with boundary in $\mathbb{R}^2$ (cf. condition (i) of definition 3.2). Geometrically, a space cell can be represented by a simple polygon with zero or more interior holes. In a topological sense, 2-dimensional space cells are the result of cutting $k \geq 1$ holes being open disks from the 2-sphere $\mathbb{S}^2$ and thus only differ in the number k of their boundary components.

Space cells with one boundary component. Prototypical examples for 2-dimensional space cells having one boundary component are depicted in figure 58. The geometric layout of space cells may be convex as shown in figure 58a or concave as illustrated by the L-shaped and U-shaped space cells in figure 58b and c.

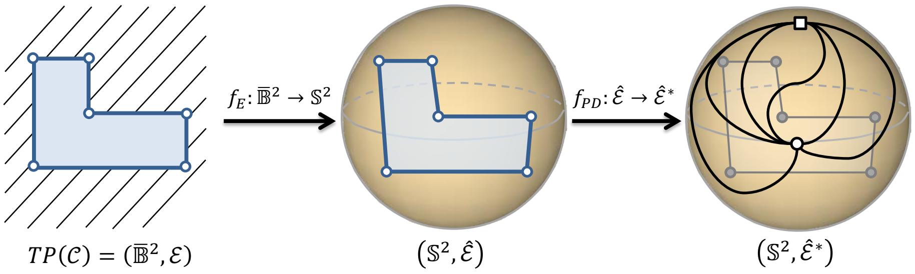


(a) $\chi(\mathcal{E}) = 4 - 4 + 1 = 1$ (b) $\chi(\mathcal{E}) = 6 - 6 + 1 = 1$ (c) $\chi(\mathcal{E}) = 8 - 8 + 1 = 1$

Figure 58: Examples of 2-dimensional space cells with their primal space topology and corresponding Euler characteristic.

A space cell S with a single boundary component is homeomorphic to the closed 2-disk $\bar{\mathbb{B}}^2$. It is an obvious consequence that S admits a finite CW complex $TP(S) = (\bar{\mathbb{B}}^2, \mathcal{E})$ that contains a single 2-cell covering the interior $Int(TP(S))$ because a 2-cell per definition A.46 is homeomorphic to the open 2-disk $\mathbb{B}^2$ which itself is the interior of $\bar{\mathbb{B}}^2$. In order to combinatorially check whether $\mathcal{E}$ is a valid cell decomposition of $\bar{\mathbb{B}}^2$ we simply compare the Euler characteristics of $\bar{\mathbb{B}}^2$ and $\mathcal{E}$. According to the classification of surfaces in appendix A.5.2, the Euler characteristic χ of a compact orientable surface with boundary is given as $\chi = 2 - 2g - h$, where g is the *genus* of the surface and h the number of *holes* (cf. theorem A.87). Since $\bar{\mathbb{B}}^2$ is obtained from the sphere $\mathbb{S}^2$ by cutting one hole (i.e., removing an open disk), we get $\chi(\bar{\mathbb{B}}^2) = 2 - 0 - 1 = 1$. The Euler characteristic of the cell decomposition $\mathcal{E}$ is the alternating sum of the numbers of k-cells in $\mathcal{E}$ and results in $\chi(\mathcal{E}) = \sum_{k=0}^{n} (-1)^k e_k$ (cf. proposition A.83). Consider the space cell S shown in figure 58a whose primal space topology $TP(S)$ decomposes $\bar{\mathbb{B}}^2$ into four 0-cells, four 1-cells, and one 2-cell. Thus, we get $\chi(\mathcal{E}) = 4 - 4 + 1 = 1$ for the Euler characteristic. And since $\chi(\mathcal{E}) = \chi(\bar{\mathbb{B}}^2)$ it follows that $TP(S)$ is a valid cell decomposition of $\bar{\mathbb{B}}^2$.

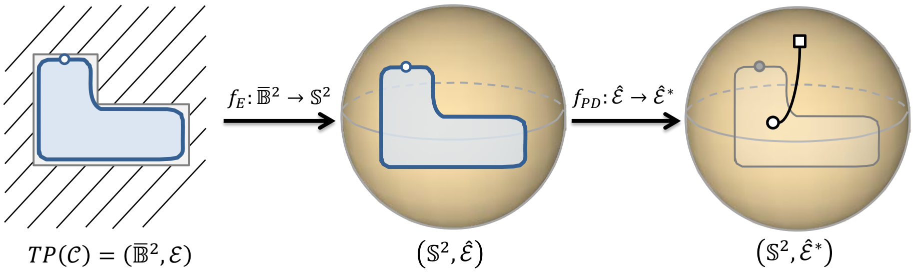
Suppose S is the only space cell in a corresponding space cell complex. Then the compactified universal face captured by S_{out} results from cutting $Int(TP(S))$ from $\mathbb{S}^2$ which likewise makes S_{out} a space cell with a single boundary component. Thus $TP(S_{out}) \cong \bar{\mathbb{B}}^2$, and what has been said about the CW decomposition of $TP(S)$ also holds for $TP(S_{out})$. The following figure 59 shows the Poincaré duality transformation for the L-shaped space cell from figure 58b and its associated outer space S_{out}.


$f_E: \bar{\mathbb{B}}^2 \to \mathbb{S}^2$
$f_{PD}: \hat{\mathcal{E}} \to \hat{\mathcal{E}}^*$
$TP(\mathcal{C}) = (\bar{\mathbb{B}}^2, \mathcal{E})$
$(\mathbb{S}^2, \hat{\mathcal{E}})$
$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$

$(\mathbb{S}^2, \hat{\mathcal{E}})$		$\xrightarrow{f_{PD}}$	$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$	
6	0-cells	→	**6**	2-cells
6	1-cells	→	**6**	1-cells
2	2-cells	→	**2**	0-cells

Figure 59: Embedding of the 2*-dimensional L-shaped space cell on* $\mathbb{S}^2$ *and Poincaré duality transformation.*

Each of the six edges in the resulting dual graph encodes the fact that S is topologically adjacent to S_{out} which actually can be viewed as being redundant information. In fact, the same information is also transported by a dual graph which links the nodes of both space cells through a single edge. Since a dual edge is bijectively related to a primal 1-cell on the common boundary of $TP(S)$ and $TP(S_{out})$, the number cells on the common boundary needs to be reduced in order to prune the dual graph. Figure 60 illustrates this for the above example by decomposing the L-shaped space cell into one 2-cell, one 1-cell, and one 0-cell which is the minimal CW decomposition for $\bar{\mathbb{B}}^2$ and any homeomorphic topological space.


$f_E: \bar{\mathbb{B}}^2 \to \mathbb{S}^2$
$f_{PD}: \hat{\mathcal{E}} \to \hat{\mathcal{E}}^*$
$TP(\mathcal{C}) = (\bar{\mathbb{B}}^2, \mathcal{E})$
$(\mathbb{S}^2, \hat{\mathcal{E}})$
$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$

$(\mathbb{S}^2, \hat{\mathcal{E}})$		$\xrightarrow{f_{PD}}$	$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$	
1	0-cell	→	**1**	2-cell
1	1-cell	→	**1**	1-cell
2	2-cells	→	**2**	0-cells

Figure 60: Poincaré dual for a minimal CW decomposition of the 2*-dimensional L-shaped space cell.*

It is important to note that in general the primal space topology $TP(S)$ of an n-dimensional space cell S may contain arbitrarily but finitely many lower dimensional cells as long as it remains homeomorphic to the manifold space realized by $GM(S)$. This means that the space cell in the above example can be represented by $GM(S)$ as L-shaped figure in primal geometry space while it is decomposed by $TP(S)$ into one k-cell per dimension which obviously does not match the number of corner points and straight line segments in $GM(S)$. Only $TP(S)$ is relevant for the Poincaré duality transformation and thus for the number of edges in the resulting dual graph. Moreover, topologically connected geometric objects on the common boundary of two n-dimensional space cells S_1 and S_2

can be described by arbitrarily but finitely many lower dimensional topological cells in $TP(S_1)$ and $TP(S_2)$. This is a direct consequence of the geometric-topological model for space cells and space layers as introduced in the previous chapters and its separate representation of the primal geometry and topology space. This modelling flexibility and expressivity has to be seen as strength of the structured space model. For example, suppose the L-shaped space cell represents a topographic room and the line segments on its boundary the walls enclosing the room. Then its dual representation as depicted in figure 59 may be advantageous as it carries the individual walls to separate edges of the dual graph which makes their thematic attributes (e.g., material information) available in path queries. Likewise, connected line segments representing walls of the same material may also be mapped onto a single dual edge in order to prune the dual graph while maintaining its topological information. A dual representation following figure 60 is typically more appropriate for sensor space cells as they are bounded by physical signal characteristics rather than real word entities and hence a single edge representing this boundary often suffices.

A counterexample to this flexibility is given by the dual-half edge model proposed by (Boguslawski & Gold 2011) that builds upon a single data structure which rigidly couples the geometric-topological representation of cells in both primal and dual space. Thus, in their model an edge in the dual graph always corresponds to a single geometric object in primal space, and the mapping of several topologically connected geometries onto a single topological cell that would be carried to a single edge in the dual graph is not supported.

Space cells with more than one boundary component. Figure 61 sketches examples for space cells having more than one boundary component. In primal space, the space cells are geometrically described through simple polygons with one or more disjoint interior holes with the number of holes being one less than the number of boundary components. The topological space described by two or more disjoint simple polygons (with or without interior holes) is also a compact surface having more than one boundary component. However, space cells are restricted to form connected topological spaces in primal space per definition 3.2, and thus disjoint simple polygons have to be mapped onto a separate space cell each.

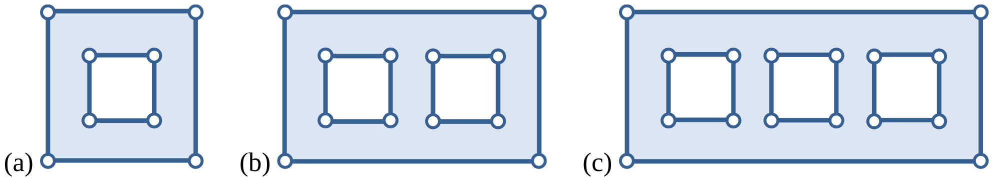


Figure 61: Examples of 2*-dimensional space cells with more than one boundary component (annulus (a), surface with two (b) and three (c) interior holes).*

The geometric figure shown on the left of figure 61 topologically can be thought of the space that results from cutting two open disks from $\mathbb{S}^2$ and also is referred to as annulus. It has genus zero and may not be confused with the torus $\mathbb{T}^2$ living in $\mathbb{R}^3$. Due to the values $g = 0$ and $h = 2$, its Euler characteristics evaluates to zero. An intuitive CW complex $(X, \mathcal{E})$ as sketched in figure 61 which decomposes the annulus into eight 0-cells, eight 1-cells, and one 2-cell however yields an Euler characteristic of one and hence is not feasible to describe the annulus. The reason is that the 1-skeleton X_1 of this CW complex is not connected and thus its underlying space itself is not connected (cf. proposition A.58) which however is not true for the annulus. Therefore, an additional 1-cell has to be introduced in order to bridge the boundary components and to make the 1-skeleton X_1 connected. In literature, this edge is often called *bridge edge* for this reason (e.g., Congli & Tsuzuki 2004, Boguslawski & Gold 2011). The following figure 62 illustrates the resulting CW complex whose Euler characteristic $\chi = 8 - 9 + 1 = 0$ now is equal to that of the annulus. Also the minimal CW decomposition of the annulus containing two 0-cells, three 1-cells, and a single 2-cell is shown.

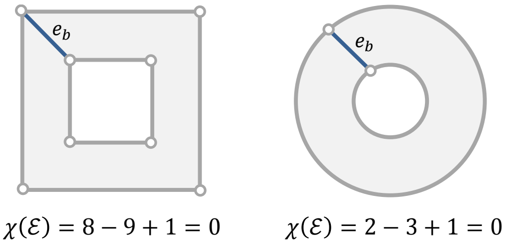
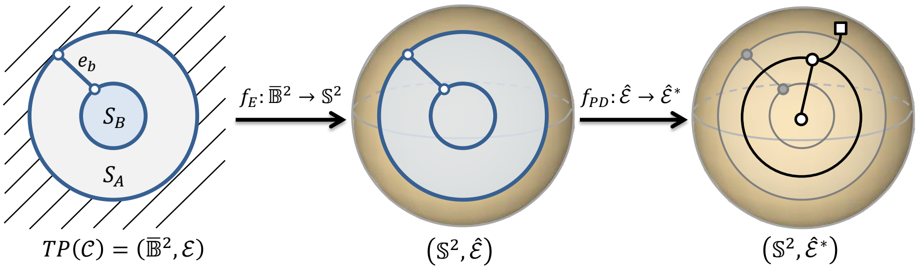

e_b
e_b
$\chi(\mathcal{E}) = 8 - 9 + 1 = 0$
$\chi(\mathcal{E}) = 2 - 3 + 1 = 0$

Figure 62: CW decomposition of the annulus involving a bridge edge e_b *(left) and minimal CW decomposition (right).*

It follows that the annulus is a valid space cell S_A of the structured space model. But it also has a deliberate impact on the topological representation $TP(S_{out})$ of the outer space cell. The compactified universal face $\mathbb{S}^2 \backslash Int(TP(S_A))$ as complement of the interior of the annulus on the 2-sphere is given by two disjoint closed 2-disks. Although each closed 2-disk is a prototypical example of a 2-dimensional space cell (cf. previous section), the complement space itself cannot be represented by S_{out} as it violates the connectedness condition in definition 3.13 and thus cannot not be mapped onto a CW complex containing only a single 2-cell. Therefore, the interior void of the annulus has to be explicitly modelled as additional space cell S_B which intuitively fills the hole. Then the union $S_A \cup S_B$ is homeomorphic to $\overline{\mathbb{B}}^2$ and the outer space $TP(S_{out})$ follows as described in the previous section.[16] In topographic space, the interior void commonly corresponds to a semantic entity. For example, assume the annulus represents a room. Then the interior of the annulus could reflect the navigable area of the room whereas its hole could denote a non-navigable area due to an immovable obstacle such as a column. In this case the semantic meaning of the interior void differs from that of S_{out} which also speaks in favour of their modelling as two separate space cells. The Poincaré duality transformation for S_A, S_B, and S_{out} is illustrated in the following figure based on the minimal CW decomposition of the annulus.


e_b
S_B
S_A
$TP(\mathcal{C}) = (\overline{\mathbb{B}}^2, \mathcal{E})$
$f_E: \overline{\mathbb{B}}^2 \to \mathbb{S}^2$
$(\mathbb{S}^2, \hat{\mathcal{E}})$
$f_{PD}: \hat{\mathcal{E}} \to \hat{\mathcal{E}}^*$
$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$

$(\mathbb{S}^2, \hat{\mathcal{E}})$		$\xrightarrow{f_{PD}}$	$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$	
2	0-cells	→	**2**	2-cells
3	1-cells	→	**3**	1-cells
3	2-cells	→	**3**	0-cells

Figure 63: Poincaré duality transformation for the annulus.

Since S_B is completely enclosed by S_A, its dual node $v_{TP}(S_B)$ is only linked to $v_{TP}(S_A)$ but not to $v_{TP}(S_{out})$ in the resulting dual representation. An important consequence of the introduced bridge edge in $TP(S_A)$ is that its dual counterpart connects $v_{TP}(S_A)$ with itself in a cycle. From a topological point of view this is consistent since the bridge edge cuts into the interior of S_A and thus forms a boundary of the 2-cell in $TP(S_A)$ to itself. However, and as already mentioned in chapter 3.1.2.4, the information encoded by the dual bridge edge is that S_A is connected to itself which holds for all space cells. Thus, dual bridge edges do not satisfy the condition (ii) of definition 3.20 and correspondingly are omitted from the dual graph $L(G_{TP})$ of a space layer L.

If a space cell S has two interior holes then it intuitively takes two bridge edges to make the 1-skeleton of $TP(S)$ connected. Thus, the minimum number of bridge edges in $TP(S)$ corresponds to the number of 2-dimensional interior holes of its underlying topological space. In order to formally express this relation for surfaces having an arbitrary number of interior holes, the second Betti number of the surface can be utilized. According to theorem A.94 (cf. appendix A.5.3), the *Betti numbers* $\beta_k(X)$ of a topological space X informally count the number of unconnected k-dimensional holes of a topological space. Based on the second Betti number $\beta_1(X)$ denoting the number of of 2-dimensional holes, the number of additional bridge edges in $TP(S)$ can be determined as follows.

Proposition 3.34 (Bridge edges in 2-dimensional space cells). Let S be a 2-dimensional space cell, and let $\beta_1(X)$ be the second Betti number of the topological space X underlying $GM(S)$. Then it requires precisely $\beta_1(X)$ additional 1-cells (*bridge edges*) to make the 1-skeleton of $TP(S)$ connected. Each bridge edge e_1 in $TP(S)$ is carried to a dual 1-cell e_1^* whose boundary $\overline{e_1^*} \setminus e_1^*$ is $v_{TP}(S)$, and thus $\overline{e_1^*} \cong \mathbb{S}^1$. It therefore connects $v_{TP}(S)$ in a cycle.

Complexes of 2-dimensional space cells. Suppose the spatial configuration $GM(\mathcal{C})$ of a 2-dimensional space cell complex $\mathcal{C}$ realizes a topological 2-manifold, then this manifold space can be classified as orientable compact surface with k boundary components, $k \geq 1$. The scenarios discussed in the context of single space cells can

[16] An alternative solution is to relax the connectedness condition for the primal space representation of S_{out} and correspondingly allow its CW decomposition $TP(S_{out})$ to contain more than one 2-cell. However, this is to be seen disadvantageous as S_{out} in such cases not necessarily represents a semantically homogeneous partition of space any more.

consequently be transferred to the CW decomposition $TP(\mathcal{C})$ of the space cell complex as well as the topological description $TP(S_{out})$ of its associated outer space cell as exemplified in figure 64.

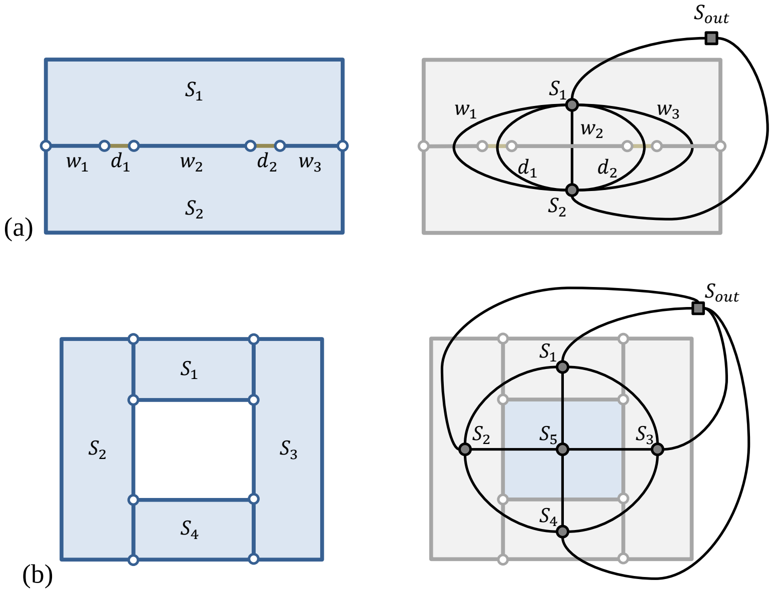
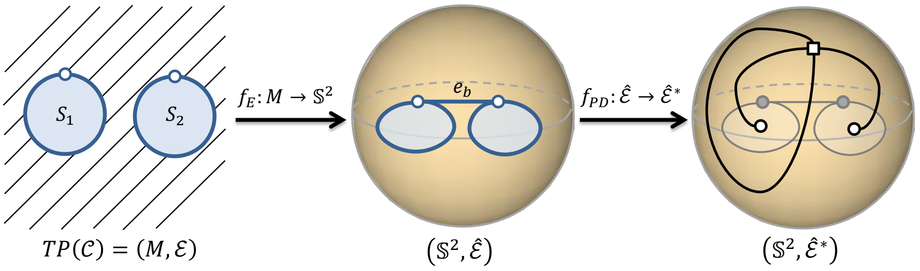


Figure 64: Examples of 2-dimensional space cell complexes (left) and the corresponding dual graphs (right).

Figure 64a shows two adjacent space cells whose resulting space cell complex is homeomorphic to the closed disk $\overline{\mathbb{B}}^2$ and thus agrees with the case of a single space cell with one boundary component. Therefore, S_{out} is also homeomorphic to $\overline{\mathbb{B}}^2$ which results in the Poincaré duality transformation illustrated on the right of figure 64a. Whereas both space cells share a single 1-cell with $TP(S_{out})$, their common boundary is decomposed into five 1-cells. Assume the space cells represent two neighboured rooms with the common boundary being a paper-thin model of the separating wall which contains two doors. Then the representation of the two doors as individual 1-cells d_1 and d_2 carries them onto distinct edges in the dual graph and allows for their separate evaluation in path queries. In figure 64a, the wall is split into the three 1-cells w_1 to w_3 enclosing the doors and is likewise translated into three dual edges. This cell decomposition possibly comes closest to the real world configuration. However, since the primal topological representation is decoupled from the geometric setting in the structured space model, also different topological descriptions are possible. For example, the 1-cells d_1 and d_2 could also make up the entire boundary between the rooms or, alternatively, the wall could be mapped onto just one additional 1-cell. As discussed in chapter 2.2, there are good reasons for explicitly representing wall entities in the final navigation graph in order to make them available for path search algorithms.

The topological space described by the space cell complex depicted in figure 64b is an annulus. It thus follows that the inner hole needs to be filled by one (or more) separate space cell since otherwise the outer space would be disconnected. On a topographic space layer this extra space cell could, for example, semantically denote an inner courtyard embraced by the four space cell. Alternatively, the four space cell could also represent a more fine-grained subdivision of a single room with the interior void reflecting a navigation obstacle (cf. above discussion). Additional bridge edges are not required in this example because the individual space cells themselves are obviously homeomorphic to $\overline{\mathbb{B}}^2$.

In contrast to a single space cell, the topological space underlying the geometric configuration of a space cell complex needs not be connected. Figure 65 demonstrates this case along two disconnected space cells which might represent the signal coverage areas of two sensors on a sensor space layer.


S_1
S_2
$TP(\mathcal{C}) = (M, \mathcal{E})$
$f_E: M \rightarrow \mathbb{S}^2$
e_b
$(\mathbb{S}^2, \hat{\mathcal{E}})$
$f_{PD}: \hat{\mathcal{E}} \rightarrow \hat{\mathcal{E}}^*$
$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$

$(\mathbb{S}^2, \hat{\mathcal{E}})$		$\xrightarrow{f_{PD}}$	$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$	
2	0-cells	→	**2**	2-cells
3	1-cells	→	**3**	1-cells
3	2-cells	→	**3**	0-cells

Figure 65: Poincaré duality transformation for two disconnected 2*-dimensional space cells.*

Again, the topological representation $TP(S)$ of each sensor space cell in figure 65 is homeomorphic to the closed disk $\mathbb{B}^2$. Since the compactified universal face of the space cell complex $\mathcal{C}$ results from removing $Int(TP(\mathcal{C}))$ from $\mathbb{S}^2$, this means cutting the open disk $\mathbb{B}^2 \cong Int\big(TP(S)\big)$ twice from $\mathbb{S}^2$ in this example. This operation obviously yields the annulus as underlying topological space for S_{out} with either of its two circular boundary components corresponding to a boundary $\partial TP(S)$ of a sensor space cell. In a topological sense, both boundary components cannot be distinguished into being exterior or interior. The CW decomposition $TP(S_{out})$ involves a bridge edge to make the boundary of the annulus connected. Assume that each sensor space cell is represented through one k-cell per dimension in $TP(\mathcal{C})$. When attaching $TP(S_{out})$ to $TP(\mathcal{C})$ in order to retrieve a single CW complex $(\mathbb{S}^2, \hat{\mathcal{E}})$ of the 2-sphere satisfying the Poincaré duality (cf. definition 3.14) then the bridge edge in $TP(S_{out})$ finally links the two 0-cells on the common boundary of $TP(S_{out})$ with the sensor space cells. We can verify this result by comparing the Euler characteristics of $\hat{\mathcal{E}}$ and $\mathbb{S}^2$ which gives $\chi\big(\hat{\mathcal{E}}\big) = 2 - 3 + 3 = 2$ and $\chi(\mathbb{S}^2) = 2$ (cf. theorem A.85). So the bridge edge cannot be removed from $(\mathbb{S}^2, \hat{\mathcal{E}})$ without invalidating it. The resulting CW complex $(\mathbb{S}^2, \hat{\mathcal{E}})$ is shown in the middle of figure 65, whereas the Poincaré dual $(\mathbb{S}^2, \hat{\mathcal{E}}^*)$ is illustrated on the right. In accordance with proposition 3.34, the dual counterpart of the bridge edge in $\hat{\mathcal{E}}^*$ connects $v_{TP}(S_{out})$ in a cycle and is omitted from the resulting dual navigation graph of a space layer without losing adjacency information. In general, in case of k disconnected space cells, S_{out} will be an orientable compact surface with k boundary components, and thus $k - 1$ bridge edges are required for $TP(S_{out})$ according to proposition 3.34.

Assume the signal coverage area of a sensor is reflected by separate sensor space cells that result from applying different signal strength bands as partitioning criteria. In figure 66, this is demonstrated for the circular reception area of a single Wi-Fi transmitter which is decomposed into three non-overlapping space cells S_1, S_2, and S_3. The space cells are supposed to cover those indoor space regions where the received signal strength ranges from -90 to -120 dBm, from -60 to -90 dBm, and from -30 to -60 dBm respectively.

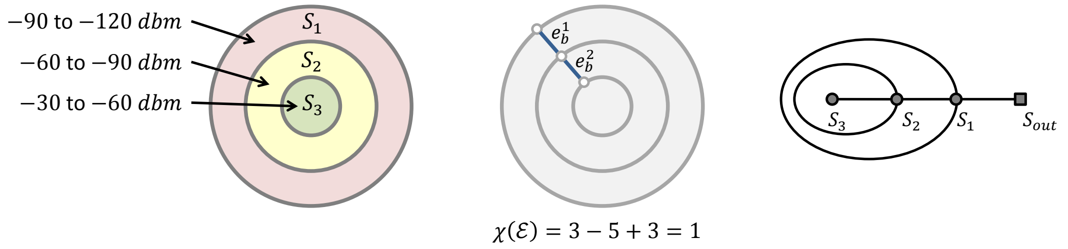


Figure 66: A space cell complex representing three signal strengths bands of a Wi-Fi transmitter (left), its primal space topology involving two bridge edges (middle), and the corresponding Poincaré dual (right).

Topologically, the space cell S_1 renders an annulus whose inner hole is filled by both S_2 and S_3. Moreover, S_2 itself is an annulus whose inner hole is occupied by the closed disk S_3. In this example, the inner holes of both S_1 and S_2 obviously have a different semantic meaning than the outer space surrounding S_1 which reflects the area without Wi-Fi signal reception. The resulting space cell complex $\mathcal{C}$ is homeomorphic to $\mathbb{B}^2$. Its minimal CW decomposition $TP(\mathcal{C})$ into three 0-cells, five 1-cells including two bridge edges, and three 2-cells is shown in the middle of figure 66. It is an immediate consequence that $v_{TP}(S_1)$ and $v_{TP}(S_2)$ are linked to themselves in cycles in the corresponding Poincaré dual. Note that neither S_2 nor S_3 can be omitted from $\mathcal{C}$ without violating the connectedness condition of S_{out}.

The discussion of spatial settings of 2-dimensional space cell complexes concludes with two examples dealing with overlaps as given in figure 67.

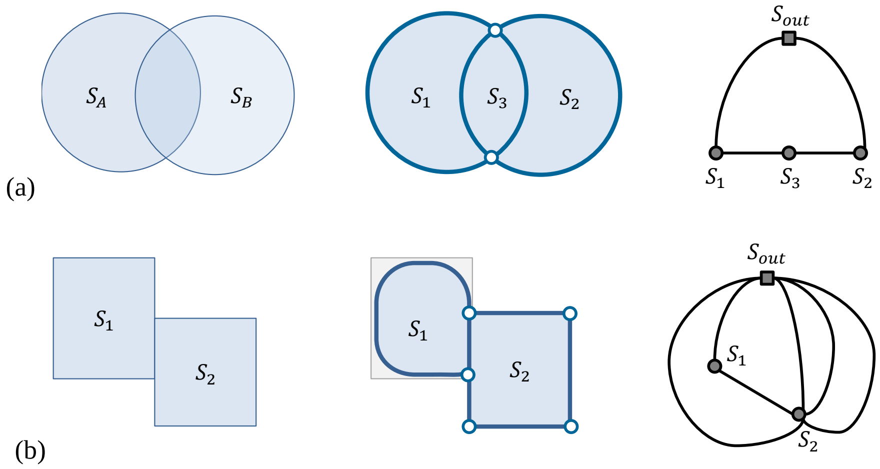


Figure 67: Examples of overlapping space cells (a) and overlapping boundary cells (b).

In figure 67a, two space cells S_A and S_B overlap in primal space which has been excluded for valid space cell complexes per definition 3.7. As shown in chapter 3.1.2.3, this setting has rather to be modelled using three non-overlapping space cells S_1, S_2, and S_3, with $S_1 = S_A \backslash S_B$, $S_2 = S_B \backslash S_A$, and $S_3 = S_A \cap S_B$. First, this adheres to an intuitive semantic understanding as the overlapping region S_3 shares the meaning and properties of both S_A and S_B and hence is distinctly different from both S_1 and S_2. Second, there is also a strong reason from algebraic topology as topological cells in an algebraic cell complex are not allowed to share parts of their interiors (cf. definition A.48). Thus, $Int(TP(S_A))$ and $Int(TP(S_B))$ cannot be translated into overlapping 2-cells in $TP(\mathcal{C})$ of the corresponding space cell complex $\mathcal{C}$. A valid CW decomposition $TP(\mathcal{C}) \cong \overline{\mathbb{B}}^2$ is depicted in the middle of figure 67a in which all space cells S_1, S_2, S_3, and S_{out} are given as regular subcomplexes of $TP(\mathcal{C})$ each of which contains two 0-cells, two 1-cells and one 2-cell and is homeomorphic to $\overline{\mathbb{B}}^2$. In dual space (cf. right of figure 67a), the node $v_{TP}(S_3)$ is the only one not being linked to $v_{TP}(S_{out})$ since S_3 only shares two 0-cells on its common boundary with S_{out}.

The second example illustrated in figure 67b also deals with non-overlapping but 1-dimensional topological cells. It shows two space cells which touch but are displaced so that their common boundary is a subset of a 1-dimensional straight boundary segment in either space cell. In order to correctly capture this configuration by a CW decomposition $TP(\mathcal{C})$ the common boundary has to be represented by a separate 1-cell as shown in the middle of figure 67b. Note that although both space cells represent a rectangular region in $\mathbb{R}^2$, their description in primal topology space differs. Precisely, a minimum possible CW decomposition for this configuration has been applied to the space cell S_1, whereas the topological cells in $TP(S_2)$ more closely follow the geometric description.

3.1.3.2 Three-dimensional Manifold Configurations

The primal space description of a 3-dimensional space cell needs to be homeomorphic to a connected k-shell manifold solid, with k being the number of its boundary components and $k \geq 1$ (cf. condition (ii) of definition 3.2). In the following examples, the primal space geometry is depicted using geometric polyhedra but more complex geometric shapes (e.g., involving freeform surfaces or curves) are allowed. Similar to the illustration of 2-dimensional settings, the following discussion considers the cases of $k = 1$ and $k > 1$ as well as space cell complexes separately.

Space cells with one boundary component. The prototypical example of a 3-dimensional space cell with one boundary component renders a single-shell manifold solid being homeomorphic to the closed 3-ball $\overline{\mathbb{B}}^3$ in primal space. Different geometric realizations of $\overline{\mathbb{B}}^3$ are depicted in the following figure 68.

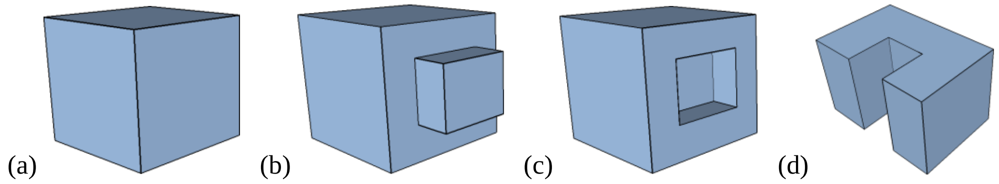
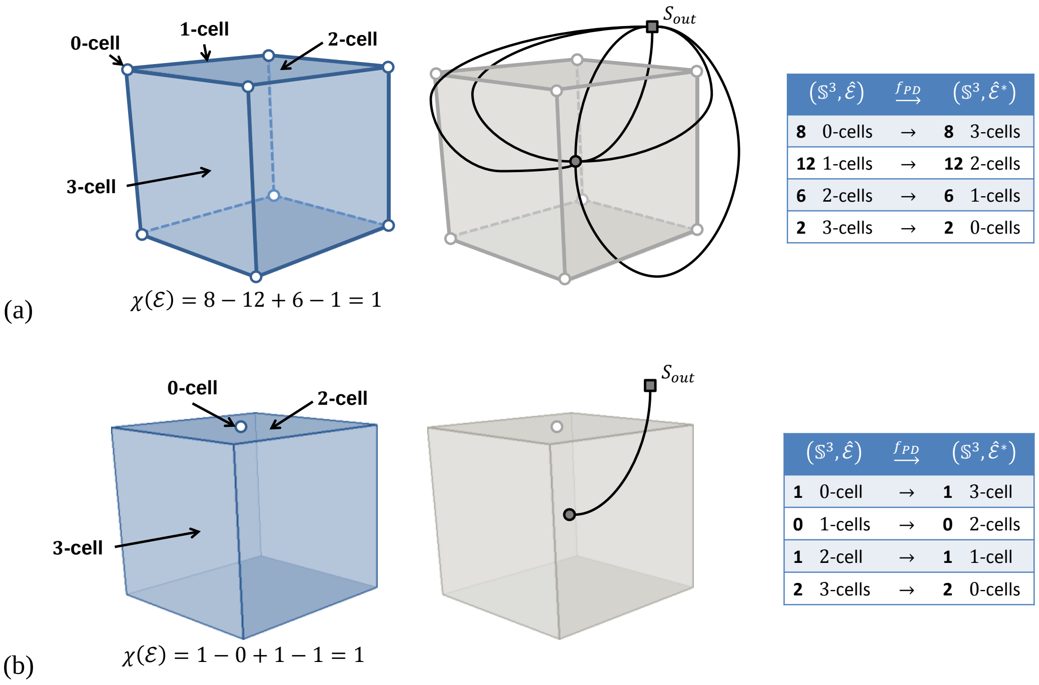


Figure 68: Examples of single-shell manifold solids being homeomorphic to $\overline{\mathbb{B}}^3$.

Similar to the 2-dimensional case, it easily follows that a space cell S whose geometric layout is homeomorphic to $\overline{\mathbb{B}}^3$ can be decomposed by $TP(S)$ into a single 3-cell $e \cong \mathbb{B}^3 = Int(\overline{\mathbb{B}}^3)$ and an arbitrary but finite number of lower dimensional cells denoting the cell boundary ∂e. As elaborated in appendix A.5.3, the Euler characteristic of manifold solids can be computed as the alternating sum of their first three Betti numbers (cf. proposition A.96). Since $\overline{\mathbb{B}}^3$ has one connected component but neither a 2-dimensional hole nor a 3-dimensional interior void, its Euler characteristic is given by $\chi(\overline{\mathbb{B}}^3) = \beta_0(\overline{\mathbb{B}}^3) - \beta_1(\overline{\mathbb{B}}^3) + \beta_2(\overline{\mathbb{B}}^3) = 1 - 0 + 0 = 1$. If S is assumed to be the only space cell in a corresponding space cell complex, then the compactified universal solid captured by $TP(S_{out})$ follows from $\mathbb{S}^3 \backslash Int(TP(S)) \cong \mathbb{S}^3 \backslash Int(\overline{\mathbb{B}}^3)$ and is itself homeomorphic to $\overline{\mathbb{B}}^3$.

In figure 69a, the Poincaré duality transformation is demonstrated for a cuboidal space cell S. The illustrated CW complex $TP(S)$ has Euler characteristic $\chi(TP(S)) = 8 - 12 + 6 - 1 = 1$ and thus agrees with $\chi(\overline{\mathbb{B}}^3)$. In order to retrieve a CW complex $(\mathbb{S}^3, \hat{\mathcal{E}})$ satisfying the Poincaré duality, the 3-cell in $TP(S_{out})$ is attached to the embedding of $TP(S)$ on $\mathbb{S}^3$ which decreases the Euler characteristic by one and $\chi(\hat{\mathcal{E}}) = 0$. This result can be validated against the Euler characteristic of $\mathbb{S}^3$ itself based on its Betti numbers. In contrast to a manifold solid, the fourth Betti number has to be additionally considered in this computation since $\mathbb{S}^3$ is a 3-manifold topologically embedded in Euclidean 4-space $\mathbb{R}^4$. From $\mathbb{S}^3$ being the manifold boundary of $\overline{\mathbb{B}}^4$ it follows that it is a single connected component with one 4-dimensional interior void given by $Int(\overline{\mathbb{B}}^4) = \mathbb{B}^4$. Thus, $\beta_0(\mathbb{S}^3) = \beta_3(\mathbb{S}^3) = 1$ and $\beta_1(\mathbb{S}^3) = \beta_2(\mathbb{S}^3) = 0$ which yields the Euler characteristic $\chi(\mathbb{S}^3) = 1 - 0 + 0 - 1 = 0$. Consequently, $\chi(\hat{\mathcal{E}}) = \chi(\mathbb{S}^3)$. In the dual graph, the six 2-cells on the boundary of the cube are translated into six dual edges linking the dual node of the cube with the dual node of the adjacent outer space cell.

In order to illustrate the possibility of having decoupled geometric and topological descriptions in primal space, figure 69b shows the duality transformation based on the minimal CW complex $(\overline{\mathbb{B}}^3, \delta)$ of polyhedra being homeomorphic to $\overline{\mathbb{B}}^3$ which contains one 0-cell, one 2-cell, and one 3-cell with $\chi(\delta) = 1 - 0 + 1 - 1 = 1 = \chi(\overline{\mathbb{B}}^3)$. As expected, this allows for deriving a dual graph linking $v_{TP}(S)$ and $v_{TP}(S_{out})$ in a single edge.

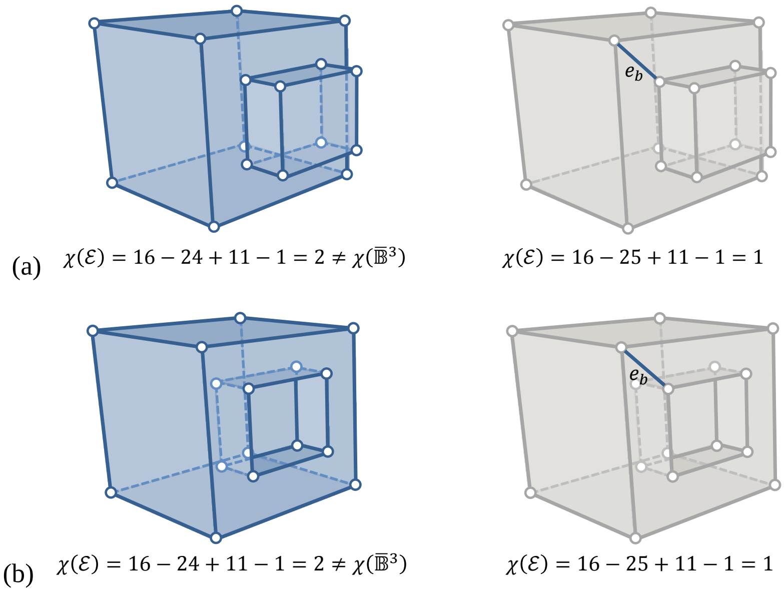

0-cell
1-cell
2-cell
3-cell
S_{out}
$\chi(\mathcal{E}) = 8 - 12 + 6 - 1 = 1$
(a)
0-cell
2-cell
3-cell
S_{out}
$\chi(\mathcal{E}) = 1 - 0 + 1 - 1 = 1$
(b)

$(\mathbb{S}^3, \hat{\mathcal{E}})$	$\xrightarrow{f_{PD}}$	$(\mathbb{S}^3, \hat{\mathcal{E}}^*)$
8 0-cells	→	**8** 3-cells
12 1-cells	→	**12** 2-cells
6 2-cells	→	**6** 1-cells
2 3-cells	→	**2** 0-cells

$(\mathbb{S}^3, \hat{\mathcal{E}})$	$\xrightarrow{f_{PD}}$	$(\mathbb{S}^3, \hat{\mathcal{E}}^*)$
1 0-cell	→	**1** 3-cell
0 1-cells	→	**0** 2-cells
1 2-cell	→	**1** 1-cell
2 3-cells	→	**2** 0-cells

Figure 69: Poincaré duality transformation for a cuboidal space cell (a) and its minimal CW decomposition (b).

The space cell shown in figure 68b needs to be considered more carefully if it is not to be mapped by the minimal CW decomposition. Its underlying polyhedron is constructed from cutting an interior hole into one boundary face of a cuboid and closing this hole by attaching a second smaller cuboid with one of its boundary faces removed. The interior of the resulting polyhedron is topologically connected rendering it homeomorphic to $\overline{\mathbb{B}}^3$. However, the Euler characteristic of the CW decomposition $TP(S)$ depicted in figure 70a is invalid as it results in $\chi(TP(S)) = 16 - 24 + 11 - 1 = 2 \neq \chi(\overline{\mathbb{B}}^3)$. The reason is that the face with the interior hole topologically is an annulus. Accordingly, an additional bridge edge connecting its boundary components has to be introduced. The rectified CW decomposition with Euler characteristic $\chi(TP(S)) = 16 - 25 + 11 - 1 = 1 = \chi(\overline{\mathbb{B}}^3)$ is shown on the right. It is important to note that in three dimensions the bridge edge does not affect the resulting dual graph structure. The Poincaré duality takes the primal 1-dimensional bridge edge to a 2-dimensional counterpart, but only the dual 0-cells and 1-cells are relevant for the dual graph. For the same reasons, also the primal CW decomposition of the polyhedron with non-through hole illustrated in figure 68c involves an additional bridge edge (cf. figure 70b).



(a) $\chi(\mathcal{E}) = 16 - 24 + 11 - 1 = 2 \neq \chi(\overline{\mathbb{B}}^3)$ $\chi(\mathcal{E}) = 16 - 25 + 11 - 1 = 1$

(b) $\chi(\mathcal{E}) = 16 - 24 + 11 - 1 = 2 \neq \chi(\overline{\mathbb{B}}^3)$ $\chi(\mathcal{E}) = 16 - 25 + 11 - 1 = 1$

Figure 70: Examples of bridge edges in the primal space topology of 3*-dimensional space cells.*

The boundaries of the polyhedra in the above examples are homeomorphic to the 2-sphere $\mathbb{S}^2$. However, and per definition A.91, the manifold boundary of a single-shell manifold solid is only limited to belong to the class of closed, orientable surfaces which includes surfaces with genus greater than zero and specifically the n-holed torus $\mathbb{T}^2$, with $n \geq 0$ being the genus of the surface and the 0-holed torus being equivalent to $\mathbb{S}^2$ (cf. appendix A.5.2). A manifold solid enclosed by an n-holed torus is equivalently said to be an n-holed toroid and is denoted as T in the following. Examples for space cells whose primal space geometry realizes an n-holed toroid are depicted below.

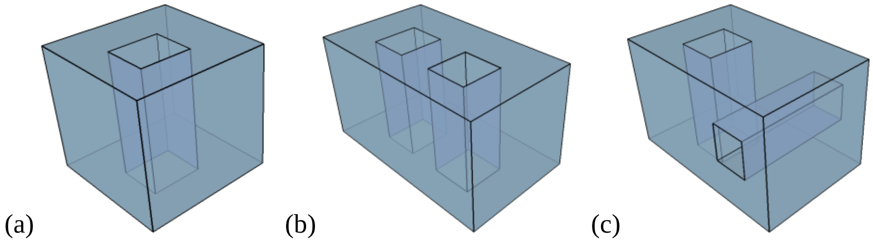
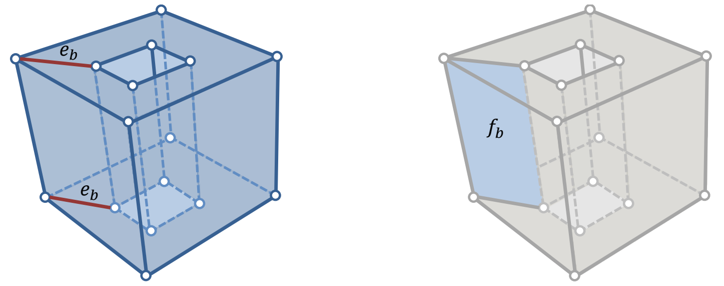


Figure 71: Examples of single-shell manifold solids being homeomorphic to an n-holed toroid.

Assume the 1-holed toroid T as shown in figure 71a. Similar to the 2-dimensional annulus, it may be used, for example, to model a topographic space cell such as a room whose navigable space surrounds a non-navigable region that extends over the entire height of the room (e.g., built obstacles like columns and pillars inside the room). It is to be characterized as a single connected component with one 2-dimensional hole so that $\chi(T) = 1 - 1 + 0 = 0$ according to its Betti numbers. Figure 72a illustrates a CW complex $(X, \mathcal{E})$ which decomposes the toroid into sixteen 0-cells, twenty-six 1-cells including two bridge edges, ten 2-cells and one 3-cell leaving $16 - 26 + 10 - 1 = -1$ for its Euler characteristic and thus $X \ncong T$. Although the bridge edges ensure the connectedness of the 1-skeleton X_1, the interior of T is not homeomorphic to $\mathbb{B}^3$ and can only be described by a single 3-cell e when adding an additional *bridge face* to $\mathcal{E}$ as depicted in figure 72b. Otherwise, the cell boundary ∂e formed by the 2-skeleton X_2 is not a continuous image of $\mathbb{S}^2$ which however is enforced by the characteristic map $\phi_e\colon \overline{\mathbb{B}}^3 \to X$ for e and its restriction $\varphi_e = \phi_e|_{\mathbb{S}^2} : \mathbb{S}^2 \to X_2$ on the boundary (cf. definition A.48). Considering this additional bridge face, the Euler characteristic $\chi(\mathcal{E}) = 16 - 26 + 11 - 1 = 0$ is equal to $\chi(T)$. A toroid is hence a valid primal space representation for a space cell of the structured space model.

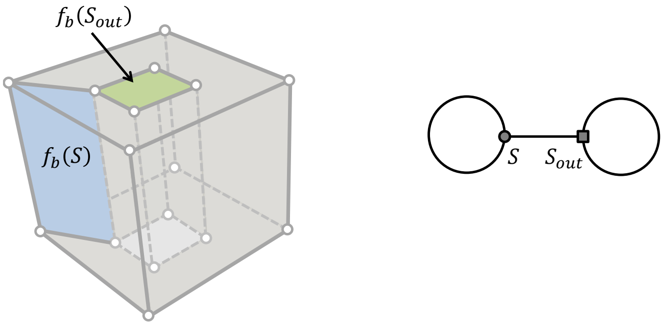


(a) $\chi(\mathcal{E}) = 16 - 26 + 10 - 1 = -1 \neq \chi(T)$ (b) $\chi(\mathcal{E}) = 16 - 26 + 11 - 1 = 0 = \chi(T)$

Figure 72: Incorrect CW decomposition of the toroid (a) and correct decomposition involving a bridge face (b).

Let S be a space cell with $TP(S) \cong T$. The 1-holed toroid T differs from the 2-dimensional annulus in that it only has a single boundary component and hence does not disconnect the compactified universal solid given by $\mathbb{S}^3 \backslash Int(T)$. Nevertheless, it also affects the primal topological representation of S_{out} due to $\mathbb{S}^3 \backslash Int(T)$ again yielding a solid torus T_2. The 3-cell e_2 in the corresponding CW decomposition $TP(S_{out})$ of T_2 can be viewed as the space surrounding T in figure 72. It again requires an additional bridge face in the CW complex of figure 72a to make its 2-skeleton a continuous image of $\mathbb{S}^2$ and a valid boundary for e_2. This additional bridge face in $TP(S_{out})$ is shown on the left of figure 73. The resulting CW complex $(\mathbb{S}^3, \hat{\mathcal{E}})$ satisfying the Poincaré duality can be easily validated through $\chi(\hat{\mathcal{E}}) = 16 - 26 + 12 - 2 = 0 = \chi(\mathbb{S}^3)$.

Each bridge face has an impact on the resulting dual graph as it is mapped onto a dual edge by the duality transformation. Since the bridge face in $TP(S)$ lies on the boundary of the contained 3-cell to itself, its dual counterpart connects $v_{TP}(S)$ in a cycle which equivalently holds for S_{out}. So similar to bridge edges in two dimensions, the dual edge encodes the obvious information that S is adjacent to itself and thus is finally filtered from the dual graph $L(G_{TP})$ of a space layer L. The dual graph for S and S_{out} is illustrated on the right of figure 73. However, a minimal CW decomposition for each toroid into one 0-cell, two 1-cells, two 2-cells, and one 3-cell has been applied in order to retrieve a single dual edge linking $v_{TP}(S)$ with $v_{TP}(S_{out})$.



$\chi(\mathbb{S}^3) = 16 - 26 + 12 - 2 = 0$

Figure 73: Bridge face in $f_b(S_{out})$ in $TP(S_{out})$ for a single toroid (left) and corresponding dual graph assuming a minimal CW decomposition (right).

Although the hole of a toroidal space cell needs not be filled by an additional space cell in order to apply the mathematical formalization of the structured space model, this may be beneficial from a semantic point of view. For example, if the hole represents a navigation obstacle in topographic space then representing the obstacle by its own space cell explicitly differentiates it from the outer space S_{out}. A corresponding space cell would be homeomorphic to $\overline{\mathbb{B}}^3$ in primal space rendering S_{out} homeomorphic to $\mathbb{S}^3 \backslash Int(T \cup \overline{\mathbb{B}}^3) \cong \overline{\mathbb{B}}^3$. Thus, the primal topological description $TP(S_{out})$ does not involve a bridge face in this case.

Space cells with more than one boundary component. In three dimensions, space cells with more than one boundary component are topologically described by connected multi-shell manifold solids in primal space. Intuitively, a connected multi-shell manifold solid can be obtained from a single-shell manifold solid by cutting one or more 3-dimensional holes into its interior. Technically, a hole results from removing the interior of another single-shell manifold solid. Geometric realizations of connected multi-shell manifold solids are presented in figure 74 by polyhedra with one or more interior holes each of which is homeomorphic to $\mathbb{B}^3$.

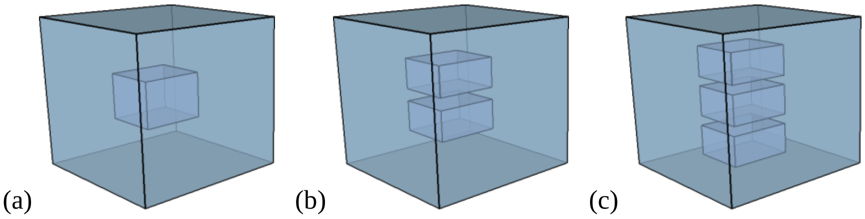


Figure 74: Examples of connected multi-shell manifold solids.

The exemplified space cells may be used, for example, to precisely model the signal coverage area of a sensor in three dimensions with the interior voids denoting lack of signal reception possibly due to specific materials in walls surrounding these areas and absorbing the signal. But also in topographic space the modelling of space cells with interior voids is useful. For example, the *exterior shell* as proposed in the model of (Boguslawski & Gold 2011) for the representation of the outer shell of a building is precisely the exterior shell of a k-shell manifold solid that at least has one more interior boundary component enclosing the spatial configuration of “void” areas within the building such as rooms, corridors, etc. This again illustrates that the notion of the *exterior shell* in (Boguslawski & Gold 2011) is not to be confused with the notion of outer space in the structured space model (see also chapter 3.1.2.4).

The Euler characteristic of the connected multi-shell manifold solids shown in figure 74 is retrieved by adding the number of interior holes to 1 for its single connected component. For example, $\chi(M) = 1 - 0 + 1 = 2$ for the connected 2-shell manifold solid M underlying the polyhedron with one interior hole. It obviously follows from the discussion so far that the CW complex $(X, \mathcal{E})$ as shown on the left of figure 75a is not a valid decomposition of M for two reasons. First, its 1-skeleton is not connected but M is a connected topological space. Second, the 2-skeleton is not a continuous image of $\mathbb{S}^2$ and thus not a valid boundary for a single 3-cell. The comparison of the Euler characteristics of M and $\mathcal{E}$ confirms this: $\chi(\mathcal{E}) = 16 - 24 + 12 - 1 = 3 \neq \chi(M)$. Similar to toroidal space cells, an additional bridge face as presented in the middle of figure 75 is required which gives $\chi(\mathcal{E}) = 16 - 26 + 13 - 1 = 2 = \chi(M)$ for the rectified CW complex.

Due to its disjoint boundary components, M disconnects the compactified universal solid $\mathbb{S}^3 \backslash Int(M)$ into two disjoint connected components each being homeomorphic to $\overline{\mathbb{B}}^3$ which violates the connectedness condition of the outer space cell. Similar to the discussion of the 2-dimensional annulus, it thus requires two separate space cells S_A and S_B with $TP(S_A) \cong M$ and S_B filling the interior void of S_A. Then S_{out} follows from $\mathbb{S}^3 \backslash Int(S_A \cup S_B)$ and is connected. The corresponding dual graph based on a minimal CW decomposition of S_A, S_B, and S_{out} is depicted on the right of figure 75 and is isomorphic to the dual graph for the annulus (cf. figure 63). Again, the bridge face links $v_{TP}(S_A)$ to itself in a cycle.

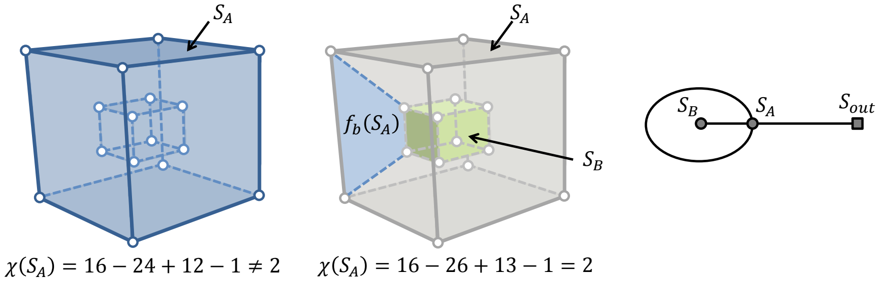

S_A
S_A
$f_b(S_A)$
S_B
S_B
S_A
S_{out}
$\chi(S_A) = 16 - 24 + 12 - 1 \neq 2$
$\chi(S_A) = 16 - 26 + 13 - 1 = 2$

Figure 75: Incorrect CW decomposition of a connected 2*-shell manifold (left), corrected CW composition involving a bridge face (middle), and resulting dual graph assuming a minimal CW decomposition and an additional space cell filling the interior void (right).*

Intuitively, the number of bridge faces in three dimensions is related to the number of 2-dimensional holes and interior cavities of the space cell. This notion is stated in the following proposition for arbitrary 3-dimensional space cells.

Proposition 3.35 (Bridge faces in 3-dimensional space cells). Let S be a 3-dimensional space cell, and let $\beta_1(X)$ and $\beta_2(X)$ be the second respectively third Betti number of the topological space X underlying $GM(S)$. Then it requires precisely $\beta_1(X) + \beta_2(X)$ additional 2-cells (*bridge faces*) as well as finitely many additional lower dimensional cells to make the 2-skeleton of $TP(S)$ a continuous image of $\mathbb{S}^2$. Each bridge face e_2 in $TP(S)$ is carried to a dual 1-cell e_2^* whose boundary $\overline{e_2^*} \setminus e_2^*$ is $v_{TP}(S)$, and thus $\overline{e_2^*} \cong \mathbb{S}^1$. It therefore connects $v_{TP}(S)$ in a cycle.

For example, consider the 3-dimensional space cell sketched in figure 76 which renders a toroid T with an interior void being homeomorphic to $\mathbb{B}^3$. Equivalently, it is a manifold solid M with a single connected component, one 2-dimensional hole due to the 1-holed torus, and one 3-dimensional hole. Its Euler characteristic can thus be computed as $\chi(M) = \beta_0(M) - \beta_1(M) + \beta_2(M) = 1 - 1 + 1 = 1$, and the additional number of bridge faces in $TP(S)$ is given as $\beta_1(M) + \beta_2(M) = 1 + 1 = 2$. The right of figure 76a shows the corresponding CW complex $(M, \mathcal{E})$ involving both bridge faces and with Euler characteristic $\chi(\mathcal{E}) = 28 - 46 + 20 - 1 = 1$ which matches $\chi(M)$. Finally, adding S_{out} gives rise to the minimal dual graph as depicted in figure 76b with S_{out} itself being a toroid having one additional bridge face in $TP(S_{out})$.

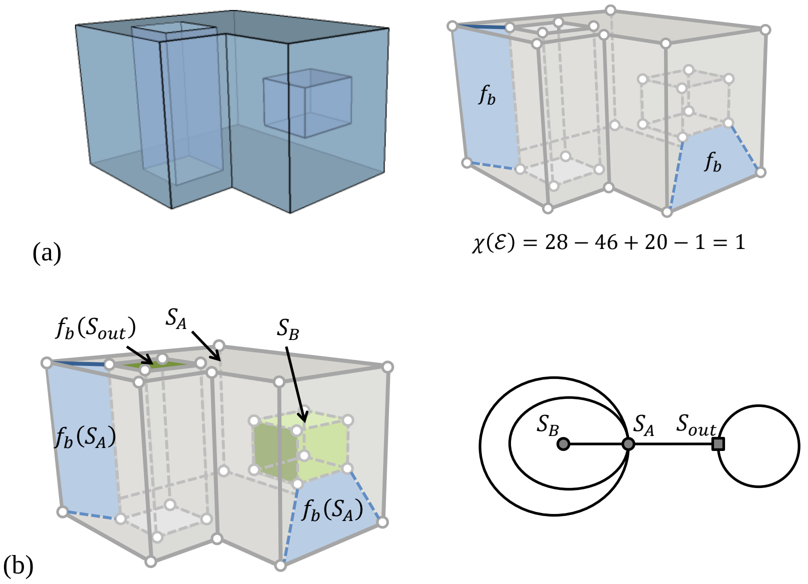


Figure 76: CW decomposition of a toroid with internal void involving two bridge faces (a), and minimal dual graph after adding S_{out} and an additional hole-filling space cell (b).

The necessity of modelling bridge faces in a 3-dimensional topological cell complex of toroidal spaces and of spaces with interior cavities is also presented in (Boguslawski & Gold 2011). However, the impact of such spaces on their ambient space as well as the dual representation of bridge faces in the Poincaré transforms of the cell complex and its ambient space are not illustrated. The duality-based works of (Lee 2001) and (Jensen et al. 2009) as well as the previous publications on the MLSEM lack a discussion of both bridge edges and bridge faces.

Complexes of 3-dimensional space cells. In case the spatial configuration $GM(\mathcal{C})$ of a 3-dimensional space cell complex $\mathcal{C}$ forms a topological 3-manifold, then this 3-manifold can be classified as k-shell manifold solid. Figure 77 exemplifies corresponding space cell complexes each of which is built from a set of simple space cells being homeomorphic to $\overline{\mathbb{B}}^3$. From left to right, the space cell complexes themselves are homeomorphic to $\overline{\mathbb{B}}^3$, the 1-holed toroid T, and a connected 2-shell manifold solid whose shells are 2-spheres. Thus, $TP(\mathcal{C})$ is connected and the topological description $TP(S_{out}(\mathcal{C}))$ of the outer space follows the scenarios presented in the context of single space cells.

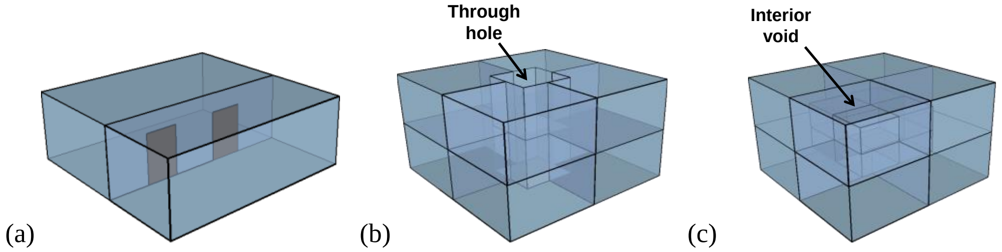


Figure 77: Examples of 3-dimensional space cell complexes.

Figure 77a is the 3-dimensional equivalence of the scenario depicted in figure 64a. It generally exemplifies the modelling of more than one 2-cell on the common boundary of two 3-dimensional space cells. Suppose the space cells represent two neighboured rooms which share two doors d_1 and d_2 enclosed by a paper-thin model of the separating wall w. A possible CW decomposition of the common boundary into twelve 0-cells, fourteen 1-cells, and three 2-cells is shown in figure 78a, with one 2-cell per wall and door. Bridge edges are not introduced in this example since the doors are not modelled as interior holes of the 2-cell representing the wall. This yields 1 as Euler characteristic for the CW decomposition and thus matches the Euler characteristic of a closed 2-disk. A further partitioning of the wall as required in two dimensions (cf. figure 64a and related discussion) is not necessary in this 3-dimensional setting. The resulting dual graph for the space cell complex is sketched in figure 78b. Both doors and the wall are translated into separate dual edges linking the room nodes. And both room nodes are connected to the dual node of the outer space cell through a single dual edge assuming that their common boundary with S_{out} is captured by a single 2-cell. Alternative cell decompositions of the shared room boundary are possible. For example, it could also be described by only two 2-cells for the doors which would consequently prune the dual edge representing the wall from the dual graph.

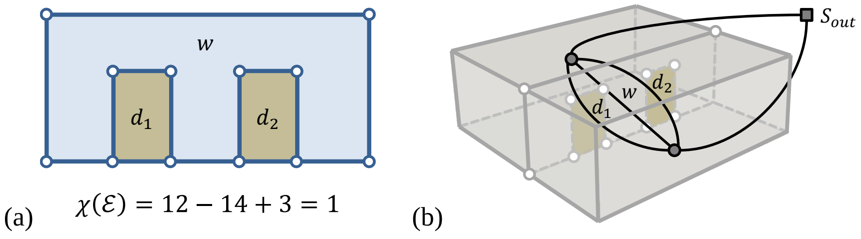


Figure 78: CW decomposition of the shared boundary of two 3-dimensional space cells involving more than one 2-cell (a) and corresponding dual graph (b).

The primal space representation of the space cell complexes in figure 77b and c illustrates the stacking of space cells on top of each other in $\mathbb{R}^3$. For example, figure 77b could be viewed as topographic space layer with the space cells representing rooms on two floors being separated by paper-thin walls and slabs. The through hole of the toroid rendered by their spatial configuration could denote an inner courtyard or possibly an elevator shaft. In either case, the modelling of one or more space cells filling the hole would capture its semantics and make it available in the resulting dual graph. The entire space cell complex could likewise model a single room and its fine-grained 3-dimensional subdivision. For the space cell complex shown in figure 77c, the interior void must be explicitly filled by one or more individual space cells in order to ensure the connectedness of the outer space. Assume the common boundary of two space cells is captured by a single 2-cell in the CW decomposition $TP(\mathcal{C})$ of either space cell complex then the dual graphs for both settings follow as shown in figure 79. Note that the through hole in figure 77b is not filled by additional space cells in this example, and thus the outer space is a toroid in $\mathbb{S}^3$ whose CW decomposition requires an additional bridge face.

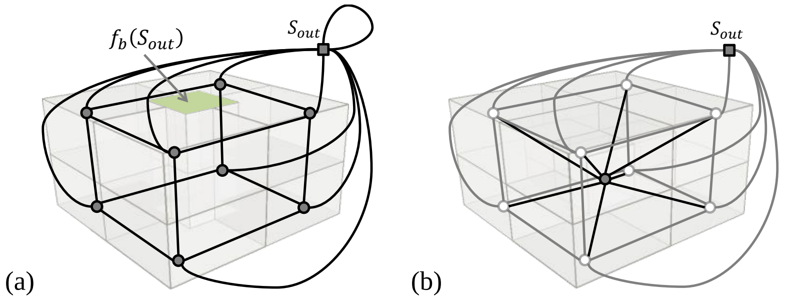


Figure 79: Dual graphs for the examples from figure 78b and c assuming a minimal CW decomposition.

An example for a space cell complex $\mathcal{C}$ whose primal space geometry $GM(\mathcal{C})$ renders a k-shell manifold solid M with more than one connected component is shown in figure 80a. Similar to the 2-dimensional example depicted in figure 65, the illustrated spherical space cells might represent the disjoint coverage areas of two sensor space cells on a sensor space layer. Since the space cells result in two connected components for M, the first three Betti numbers are given as $\beta_0(M) = 2$ and $\beta_1(M) = \beta_2(M) = 0$, and correspondingly $\chi(M) = 2$. A minimal CW decomposition of $TP(\mathcal{C})$ into two 0-cells, two 2-cells, and two 3-cells agrees with this result. The compactified universal solid results from cutting the interior of each space cell from $\mathbb{S}^3$ which yields a connected 2-shell manifold solid, and according to proposition 3.35 one additional bridge face has to be introduced in $TP(S_{out})$. The minimal CW complex $(\mathbb{S}^3, \hat{\mathcal{E}})$ for this setting satisfying the Poincaré duality is given in figure 80b and decomposes $\mathbb{S}^3$ into four 0-cells, four 1-cells, three 2-cells, and three 3-cells with $\chi(\hat{\mathcal{E}}) = 4 - 4 + 3 - 3 = 0$ being equal to $\chi(\mathbb{S}^3)$. The corresponding dual graph in figure 80c is isomorphic to the dual graph resulting for the equivalent 2-dimensional model (cf. figure 65).

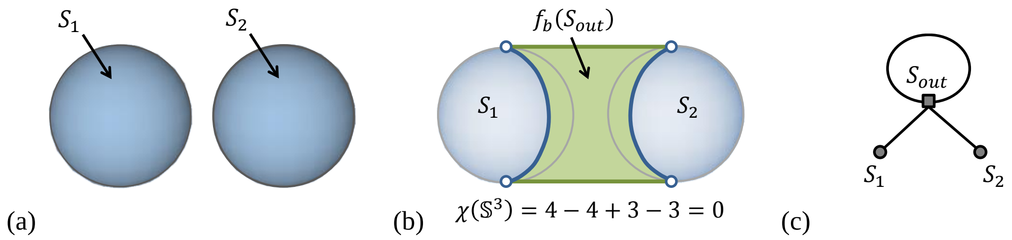


Figure 80: Two disjoint solid space cells (a), additional bridge face $f_b(S_{out})$ for the outer space cell S_{out} (b), and resulting dual graph.

The above example can be generalized to arbitrarily shaped connected components in a 3-dimensional space cell complex. Suppose the primal space configuration of space cells results in two connected components with the first one being homeomorphic to $\overline{\mathbb{B}}^3$ and the second one rendering a 1-holed toroid T. Then removing their interiors from $\mathbb{S}^3$ retrieves a connected 2-shell manifold solid. Since topologically either shell may be exterior or interior, the manifold solid can be viewed as 1-holed toroid with a spherical inner void (cf. figure 76), or vice versa as closed 3-ball with a toroidal inner void. In both cases two additional bridge faces need to be modelled for $TP(S_{out})$ to correctly decompose the compactified outer space.

For the same reasons as discussed in the context of 2-dimensional space cell complexes in chapter 3.1.3.1 (cf. figure 67), overlapping space cells as well as overlapping topological k-cells in CW decompositions of space cells and space cell complexes are neither allowed in three dimensions.

3.1.3.3 Non-manifold Configurations

The duality transformation of an n-dimensional space cell complex $\mathcal{C}$ is independent from whether or not the topological space underlying $GM(\mathcal{C})$ is a topological n-manifold. Although the Poincaré duality theorem presupposes an oriented closed manifold, this requirement is satisfied by the compact space $\mathbb{S}^n$ in which $TP(\mathcal{C})$ is topologically embedded. Therefore, the only additional condition to be met is that the CW complex $(\mathbb{S}^n, \hat{\mathcal{E}})$ which results from attaching $TP(S_{out})$ to the embedding of $TP(\mathcal{C})$ (cf. definition 3.14) be a valid CW decomposition of $\mathbb{S}^n$. The following figure 81 illustrates non-manifold configurations of space cells in two and three dimensions.

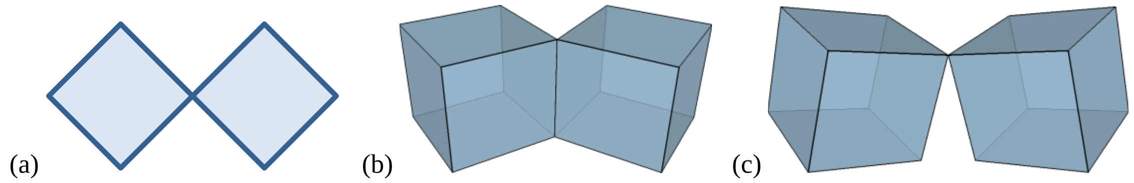
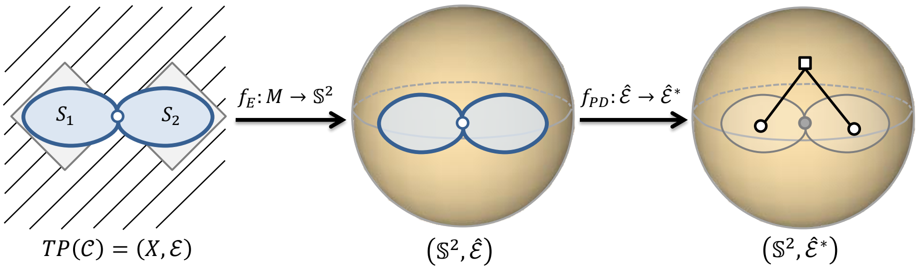


Figure 81: Examples of non-manifold space cell complexes in two (a) and three dimensions (b and c).

All space cell complexes shown violate the definition of a topological n-manifold with boundary since the points where their contained space cells touch neither have an open neighbourhood homeomorphic to an open subset of $\mathbb{R}^n$ nor to an open subset of the half-space $\mathbb{R}^n_+ = \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_n \geq 0\}$ (cf. definition A.40). Nevertheless, assume the 2-dimensional space cell complex $\mathcal{C}$ illustrated on the left of figure 81 and its CW decomposition $TP(\mathcal{C})$ into one 0-cell, two 1-cells, and two 2-cells as depicted on the left of figure 82. Then gluing the 2-cell in $TP(S_{out})$ to $TP(\mathcal{C})$ yields a CW complex $(\mathbb{S}^2, \hat{\mathcal{E}})$ with Euler characteristic $\chi(\hat{\mathcal{E}}) = 1 - 2 + 3 = 2$ which equals the Euler characteristic of $\mathbb{S}^2$. Hence, $\hat{\mathcal{E}}$ is obviously a valid CW decomposition of $\mathbb{S}^2$ and its Poincaré transform is sketched on the right of figure 82.


S_1
S_2
$f_E: M \to \mathbb{S}^2$
$f_{PD}: \hat{\mathcal{E}} \to \hat{\mathcal{E}}^*$
$TP(\mathcal{C}) = (X, \mathcal{E})$
$(\mathbb{S}^2, \hat{\mathcal{E}})$
$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$

$(\mathbb{S}^2, \hat{\mathcal{E}})$		$\xrightarrow{f_{PD}}$	$(\mathbb{S}^2, \hat{\mathcal{E}}^*)$	
1	0-cell	→	**1**	2-cell
2	1-cells	→	**2**	1-cells
3	2-cells	→	**3**	0-cells

Figure 82: Poincaré duality transformation for the non-manifold space cell complex from figure 81a.

In the derived dual graph, the dual nodes of both space cells are linked to $v_{TP}(S_{out})$ but not to each other because in primal space the space cells only share a 0-cell on their common boundary which is carried to a dual 2-cell. This result is consistent with the illustrated examples for manifold space cell complexes in the previous chapters. Suppose the example is augmented with one more space cell as presented in figure 83 which makes the space cell complex a manifold space homeomorphic to $\overline{\mathbb{B}}^2$. Then in the dual graph based on a minimal CW decomposition of the space cells (cf. right of figure 83), the two dual nodes of the original space cells are neither linked to each other. It follows that the non-manifold setting of space cells is mapped onto a valid and consistent dual representation. The same result holds for arbitrary non-manifold space cell complexes in two and three dimensions.

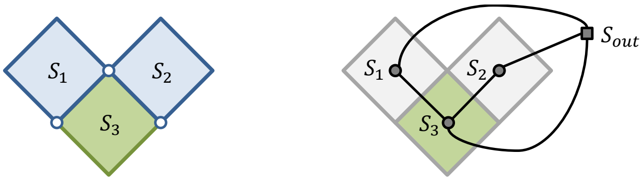


Figure 83: Adding a further space cell to retrieve a manifold space cell complex from figure 81a (left) and corresponding dual graph assuming a minimal CW decomposition (right).

Observe that the dual representation of the non-manifold space cell complex in figure 82 is almost identical to that of the manifold configuration rendered by two disconnected space cells as discussed in chapter 3.1.3.1 (cf. figure 65). The dual graph in that case only differs in its additional dual edge being incident to $v_{TP}(S_{out})$ which results from the bridge edge in $TP(S_{out})$. Although the bridge edge and its dual counterpart do not carry relevant information in the context of navigation, this example nicely demonstrates their relevance from a topological point of view.

The mathematical formalization of a space cell complex as presented in chapter 3.1.2.4 already supports non-manifold configurations of space cells. Precisely, the primal space geometry $GM(\mathcal{C})$ of a space cell complex $\mathcal{C}$ is not restricted to form a topological n-manifold with boundary (cf. definition 3.8), and neither is its primal space topology $TP(\mathcal{C})$ (cf. definition 3.9). However, it is important to note that although a space cell complex may be non-manifold, each of its contained space cells is required to render a topological n-manifold with boundary according to definition 3.2 in order to ensure its one-to-one mapping onto a single node in the dual graph. The outer space cell $S_{out}(\mathcal{C})$ is an exception from this rule. Since $TP(S_{out})$ describes the complement space of the topological embedding of $TP(\mathcal{C})$ in $\mathbb{S}^n$, it necessarily is non-manifold if $TP(\mathcal{C})$ is non-manifold. The mathematical formalization of S_{out} also accounts for this fact as both its primal space geometry $GM(S_{out})$ and its primal space topology $TP(S_{out})$ are defined differently from that of ordinary space cells (cf. definition 3.12 and definition 3.13).

In summary, it follows that the structured space model and its mathematical foundation are flexible enough to capture spatial settings of space cells that render both manifold and non-manifold topological spaces. At least for the dual-half edge structure proposed by (Boguslawski & Gold 2011), non-manifold cases are explicitly excluded. And similar to the discussion about bridge edges and faces, the issue of non-manifold configurations of cells is neither addressed in (Lee 2001) and (Jensen et al. 2009), nor in the previous publications on the MLSEM.

3.2 Combining Different and Multiple Space Representations

The structured space model facilitates the separate and independent modelling of arbitrary notions of indoor space and their corresponding space partitioning on different space layers. Each space layer is useful in its own right for decoupled tasks in indoor navigation. For example, and as illustrated in chapter 3.1.2.3, topographic space layers allow for path planning and visual guidance, and hence suffice to derive navigation or escape routes within a building that may be communicated through a guidance system installed in the building (e.g., display panels for routing humans to the nearest or safest emergency exit) without the need for accompanying sensor or logical space models. Likewise, the dual graph of a sensor space layer already enables tracking and localization of moving persons or objects in relation to the modelled sensor space cells, and thus is not necessarily dependent on a complementary topographic space representation. However, the tracking of persons or objects along a route as well as their localization with reference to topographic regions obviously requires an integrated, context-dependent view and the simultaneous evaluation of multiple space representations. In this chapter it is shown how different and multiple space layers are formally combined to a consistent *multilayered space model* of indoor space based on the common geometric-topological representation schema for space cells and space layers and its mathematical formalization. The abstract conceptualization of indoor space through the structured space model ensures that the multilayered space representation again is applicable to various notions of indoor space. The conceptual idea of modelling multiple representations of indoor space on different space layers is sketched in figure 84. The resulting stack of layers provides a complete view on an indoor space in question with each layer capturing a structuring of space along distinct semantics.

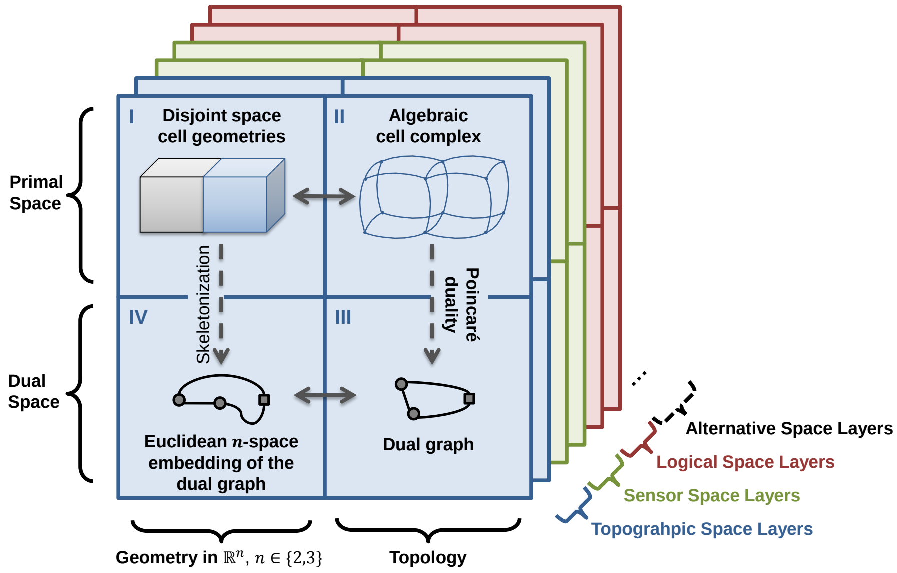


Figure 84: Modelling of indoor space on different and multiple space layers.

The number and types of space layers in this multilayered space model is unbounded and certainly depends on the navigation application to be realized. For example, several topographic space layers may be required to model the interior built environment and its decomposition into navigable and non-navigable spaces for different modes of locomotion. In case pedestrian navigation should be supported only, a single topographic space layer however may already be adequate. Likewise, each positioning technology available in a building or facility is modelled as individual sensor space layer which also holds for different partitionings into virtual space cells on logical or alternative space layers along non-physical criteria. So the number of layers also differs with respect to the available localization infrastructure and the logical facts to be mapped. Moreover, suppose that a logical layer represents disaster areas such as (parts of) rooms and corridors affected by a fire incident which prohibit access or movement, then apparently this space partitioning is only relevant during the emergency situation. Thus, space layers may also be dynamically added or removed from the multilayered space model.

Although the space layers are depicted as thin slices in figure 84, it is important to remember that each space layer may provide a 3-dimensional covering of the entire indoor space. For example, a single topographic space layer is suitable for reflecting the rooms, corridors, doors, stairs, elevators, and further architectural entities on all floors of a building. A 2-dimensional space layer, on the other hand, is in fact typically limited to a single building floor which then accordingly restricts the modelling scope of additional layers in a multilayered representation. It follows that in this case a collection of 2-dimensional multilayered space models is required to describe the entire indoor space, with each multilayered representation covering a separate building floor. From this, a necessary condition to the combination of different and multiple space layers can be deduced that all space layers must share the same dimension.

A collection of space layers is called a *space layer complex* and is defined in the following.

Definition 3.36 (Space layer complex). A *space layer complex* $\mathcal{L} = \{L_\alpha\}_{\alpha \in I_\mathcal{L}}$ is a set of finitely many space layers L_α of the same dimension n, with $2 \leq n \leq 3$ and $|\mathcal{L}| \geq 0$.

A space layer complex is said to have dimension n. It is an easy fact from the conceptualization of space underlying the structured space model that the space partitioning on each space layer $L \in \mathcal{L}$ may substantially vary due to differing numbers, shapes and spatial extents of space cells contained in $\mathcal{C}(L)$. This implies that although the space cells in $\mathcal{C}(L)$ are required to be mutually non-overlapping, this condition does not hold between the space cells on two different space layers in $\mathcal{L}$. However, all space cells in the collection of space layers have in common that they describe the same real world indoor space, and every space layer is a tiling of $\mathbb{R}^n$. Thus, it is also immediate that every space cell $S_\alpha \in \{\mathcal{C}(L_1), S_{out}(L_1)\}$ has a non-empty intersection in primal space with at least one space cell $S_\beta \in \{\mathcal{C}(L_2), S_{out}(L_2)\}$ for two distinct space layers $L_1, L_2 \in \mathcal{L}$. This relationship between the space cells on different space layers provides the conceptual basis for the coupling of space layers in the multilayered space model.

In order to visually illustrate this relationship, the following figure 85 reuses the simple indoor setting as introduced in chapter 3.1.2.3. It shows a space layer complex consisting of two space layers. The first space layer represents the built-up space by three topographic space cells capturing two rooms each of which is connected to a corridor through a door surface, whereas the second space layer models three sensor space cells denoting the overlapping coverage areas of two Wi-Fi transmitters. In figure 85b and c, the top view of the primal geometric extent of the space cells as well as the corresponding Poincaré transform is sketched for either space layer (quadrants II and IV of the geometric-topological representation schema for space layers presented in figure 42 have been omitted for clarity). In both dual graphs, the dual nodes of two adjacent space cells are linked by a single dual edge assuming that the common boundary between the space cells is mapped onto a single 2-cell in primal topology space.[17]

[17] Note that in a 2-dimensional setting the boundary part of the corridor between both doors would have to be modelled as separate 1-cell in primal topology space which would correspondingly yield at least two dual edges linking the dual node of the corridor with the dual node of the outer space (cf. figure 64 and related discussion in chapter 3.1.3.1). The dual graph of the sensor space layer would be identical in two dimensions.

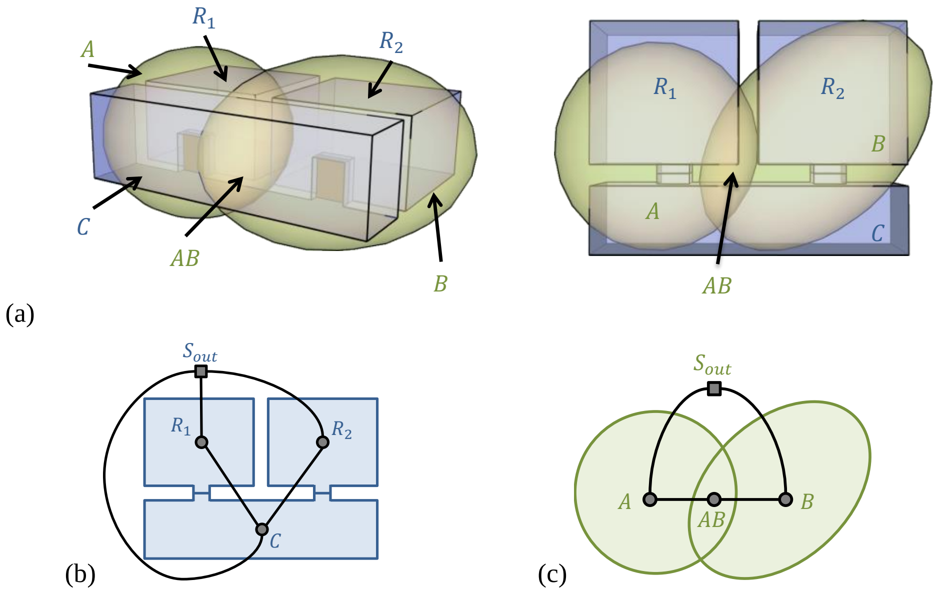


Figure 85: Example 3*-dimensional indoor scene from figure 18 (a), and dual graphs of the corresponding topographic space layer (b) and sensor space layer (c).*

A person or object travelling through indoor space moves through a distinct space cell on each layer at a given point in time (cf. chapter 3.1.1.1). For example, consider a person being located in room R_1 in the above example. Then this person is necessarily simultaneously within the spatial region covered by either sensor space cell A, AB, or the outer space cell of the sensor space layer, but not within space cell B. This knowledge is implicitly available from the geometric layout and the spatial configuration of the space cells between both space layers, and thus can be derived by geometric operations in two or three dimensions which however are computationally expensive. Therefore, a purely combinatorial model encoding the spatial configuration of space cells on different space layers is proposed that realizes the coupling between space layers in the multilayered space model and facilitates computationally efficient reasoning.

The linkage is formally expressed in terms of *inter-layer edges* connecting the dual nodes of space cells from different space representations. Precisely, an inter-layer edge between the dual nodes of two space cells on different space layers is established if and only if the intersection of the interiors of their primal space geometries is non-empty. It follows that the space layers participating in a space layer complex are linked in dual topology space (cf. quadrant III in figure 42) based on the topological relationships between their space cells in primal geometry space (cf. quadrant I in figure 41). Given two space layers L_1 and L_2, the set of inter-layer edges is derived by pairwise intersecting the primal geometric description of every space cell on L_1 with that of every space cell on L_2. This constitutes a *bipartite graph* in dual space whose nodes are divided into the two disjoint sets $V(G_{TP}(L_1))$ and $V(G_{TP}(L_2))$, and whose edges join two nodes from either set. When considering k space layers then the pairwise intersection of space cells over all space layers obviously yields a k*-partite graph* with k disjoint partitions of dual nodes. For a given space layer complex, this k-partite graph is called *inter-layer graph*.

Definition 3.37 (Inter-layer graph, inter-layer edge). The *inter-layer graph* $ILG(\mathcal{L})$ of a space layer complex $\mathcal{L}$ is a k-partite graph $ILG = (V, E)$, with $k = |\mathcal{L}|$, and where

(i) the node set $V(ILG)$ is partitioned into k disjoint subsets $V_1, \dots, V_k$ with $\forall i \in \{1, \dots, k\}: L_i \in \mathcal{L} \wedge V_i = V\big(G_{TP}(L_i)\big)$, and

(ii) the edge set $E(ILG)$ is given by $E(ILG) = \big\{\{v_{TP}(S_\alpha), v_{TP}(S_\beta)\} \,\big|\, Int\big(GM(S_\alpha)\big) \cap Int\big(GM(S_\beta)\big) \neq \emptyset\big\}$ with $\forall i \neq j \in \{1, \dots, k\}: v_{TP}(S_\alpha) \in V_i \wedge v_{TP}\big(S_\beta\big) \in V_j$. An edge $e \in E(ILG)$ is said to be an *inter-layer edge*.

Due to every space layer $L \in \mathcal{L}$ being a complete covering of $\mathbb{R}^n$, each node in a given partition $V_i \in V(ILG)$ is adjacent to at least one node in every other partition $V_j \in V(ILG)$, with $i \neq j$. Thus, each node in $V(ILG)$ is incident to at least $k-1$ inter-layer edges in $E(ILG)$, and therefore $|E(ILG)| \geq \frac{k(k-1)}{2}$. The lower limit $|E(ILG)| = \frac{k(k-1)}{2}$ only occurs when $\mathcal{L}$ contains k copies of the minimal space layer L_{min} (cf. definition 3.32).

The precise topological relationship between two space cells in primal space linked by an inter-layer edge in dual space can be categorized based on the 4-intersection model (cf. Egenhofer & Franzosa 1991 and appendix A.6). The 4-intersection model suffices for this purpose since the primal space geometry $GM(S)$ of every space cell S realizes an n-dimensional object embedded in the ambient Euclidean space $\mathbb{R}^n$ yielding a co-dimension of zero for $GM(S)$, with $2 \leq n \leq 3$ (cf. definition 3.2 and definition 3.12). Moreover, and as discussed in appendix A.6, the 4-intersection model can be applied in both two and three dimensions.

Let $e = \{v_{TP}(S_\alpha), v_{TP}(S_\beta)\} \in E(ILG)$ be an edge of an inter-layer graph ILG. Per condition (ii) of definition 3.37, the intersection $Int(GM(S_\alpha)) \cap Int(GM(S_\beta))$ of the primal space geometries of the space cells S_α and S_β is non-empty. Then, a 4-intersection matrix $\mathfrak{I}_4(S_\alpha, S_\beta)$ capturing the topological relationship between S_α and S_β is defined as

$$\mathfrak{I}_4(S_\alpha, S_\beta) = \begin{pmatrix} \neg\emptyset & Int(GM(S_\alpha)) \cap \partial GM(S_\beta) \\ \partial GM(S_\alpha) \cap Int(GM(S_\beta)) & \partial GM(S_\alpha) \cap \partial GM(S_\beta) \end{pmatrix}. \tag{3.38}$$

This 4-intersection matrix yields six possible topological relationships. In case $\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset \end{pmatrix}$, the space cell S_α *contains* the space cell S_β, whereas S_α is *inside* S_β for $\begin{pmatrix} \neg\emptyset & \emptyset \\ \neg\emptyset & \emptyset \end{pmatrix}$. Likewise, S_α *covers* S_β given the intersection matrix $\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$ and is *coveredBy* S_β for $\begin{pmatrix} \neg\emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset \end{pmatrix}$. Two more possible relationships are identified through $\begin{pmatrix} \neg\emptyset & \emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$ and $\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset \end{pmatrix}$ which denote that S_α and S_β are *equal* respectively that both space cells *overlap*.

In order to make the topological relationship between two space cells explicit in the combinatorial model, it is mapped onto a corresponding *edge label* assigned to the inter-layer edge, which renders the inter-layer graph an edge-labelled graph.

Definition 3.39 (Inter-layer edge labelling). Let $ILG(\mathcal{L})$ be the inter-layer graph of a space layer complex $\mathcal{L}$. For each edge $e = \{v_{TP}(S_\alpha), v_{TP}(S_\beta)\} \in E(ILG(\mathcal{L}))$ there exists a continuous map $l_e: e \rightarrow \mathfrak{I}_4(S_\alpha, S_\beta)$ (called *edge labelling* for e) which associates e with the binary topological relationship of its incident space cells S_α and S_β.

The following figure exemplifies the bipartite inter-layer graph for the indoor setting introduced in figure 84. On the left, the primal space representations of both space layers are shown in a 2-dimensional top view to better illustrate the overlaps between the topographic space cells and the sensor space cells. The right part depicts the resulting inter-layer graph in dual topology space which connects the dual nodes of space cells having a non-empty intersection of their interiors in primal space. For instance, since the coverage area of the Wi-Fi space cell A spatially overlaps with the primal space geometries of the space cells R_1 and C as well as the outer space cell of the topographic space layer, the corresponding dual nodes are linked by inter-layer edges. In this example, all inter-layer edges express *overlap* relationships between incident dual nodes and hence their edge labels are not illustrated separately.

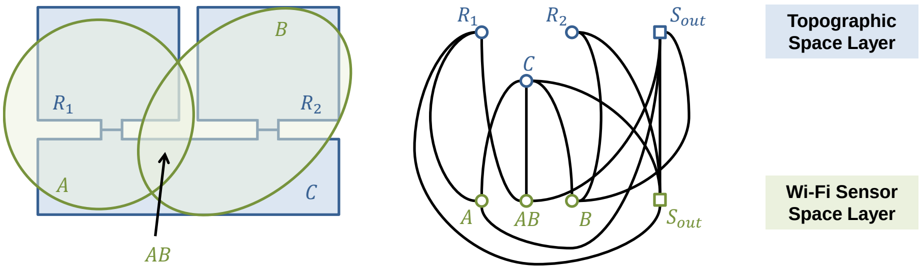


Figure 86: Bipartite inter-layer graph for the indoor setting from figure 84.

The inter-layer graph $ILG(\mathcal{L})$ of a space layer complex $\mathcal{L}$ is complementary to the Poincaré dual graphs on the separate space layers contained in $\mathcal{L}$. First, each Poincaré dual graph is restricted to a single space layer $L \in \mathcal{L}$ and is therefore equivalently referred to as intra-layer graph of L (cf. definition 3.20). Second, whereas the edges of each Poincaré dual graph denote adjacency relationships between mutually non-overlapping space cells, the edges of the inter-layer graph encode mutually overlapping space cells. It intuitively follows that the edges of the inter-layer graph are orthogonally aligned with the edges of each intra-layer graph. And since the node sets of the Poincaré dual graphs in $\mathcal{L}$ are just the disjoint partitions of $V(ILG(\mathcal{L}))$, the separate graph representations can be integrated into a single graph structure. This graph structure is said to be a *multilayered graph* within the MLSEM and it is unique for a given space layer complex $\mathcal{L}$.

Definition 3.40 (Multilayered graph). The *multilayered graph* $MLG(\mathcal{L}) = (V, E)$ of a space layer complex $\mathcal{L}$ is the union of the inter-layer graph $ILG(\mathcal{L})$ associated with $\mathcal{L}$ and every intra-layer graph $G_{TP}(L)$ with $L \in \mathcal{L}$, so that

(i) the node set $V(MLG(\mathcal{L})) = V(ILG(\mathcal{L}))$ is equal to the node set of the intra-layer graph, and

(ii) the edge set $E(MLG(\mathcal{L}))$ is given as disjoint union of the edge set of the inter-layer graph and the edge sets of all intra-layer graphs in $\mathcal{L}$, and thus $E(MLG(\mathcal{L})) = E(ILG(\mathcal{L})) \cup \bigcup_{\alpha \in I_{\mathcal{L}}} E(G_{TP}(L_\alpha))$.

The multilayered graph resulting for the above example is given in figure 87. The intra-layer edges of each Poincaré dual graph when applying a minimum CW decomposition are drawn as continuous lines, whereas the inter-layer edges between the space layers are depicted as dashed lines.

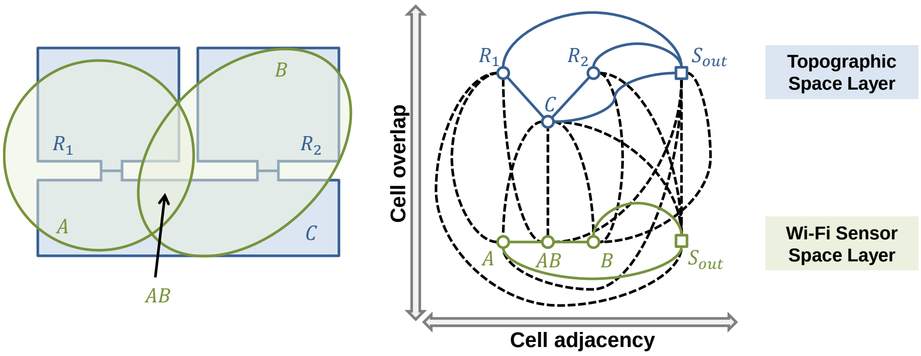


Figure 87: Multilayered graph for the indoor setting from figure 84 which spans a 2-dimensional space along cell overlaps and adjacencies.

The coupling between both space layers in the multilayered graph as shown in figure 87 enables localization and tracking of moving persons and objects within the sketched topographic indoor scene. For example, if a navigation user carries a mobile device that is Wi-Fi-enabled and receives the signal from transmitter B only, then according to the inter-layer edges between the dual node of sensor space cell B and the dual nodes on the topographic space layer, the user must be located in either room R_2 or in the corridor C. If the user moves and, assuming a continuous signal reception, the signal of Wi-Fi transmitter A is received in addition, then the user must be located in C but not in R_1. The latter would require passing sensor space cell A beforehand which spatially reflects the area where solely the signal of transmitter A is received. However, this contradicts with the Poincaré dual graph of the sensor space layer according to which sensor space cell A is not adjacent to B and hence can only be reached from B via space cell AB or the outer space cell reflecting areas without Wi-Fi signal reception. Thus, the tracking of the user movement on the intra-layer graph of the sensor space layer facilitates reasoning about alternative positioning estimates. This result is mathematically formalized in the next chapter 3.3.

In figure 88, the example is enriched with a third space layer mapping two security zones $High$ and Low by two corresponding space cells which together cover all topographic entities. Whereas R_2 and C are spatially enclosed by the space cell Low, R_1 is partly covered by Low and $High$ (suppose a security gate within the room). The primal space representation as well as the corresponding dual graph of the security space layer is shown in figure 88a. The left of figure 88b provides an integrated top view on all three layers and reveals overlapping space cells. The resulting multilayered graph is shown on the right of figure 88b. The topological relationships in primal space between the security space cell Low and the topographic space cells R_2 and C can be classified as *contains* relationships, which is reflected by labels for the corresponding inter-layer edges of the 3-partite inter-layer graph. All further inter-layer edges denote *overlap* relationships and hence are not labelled separately.

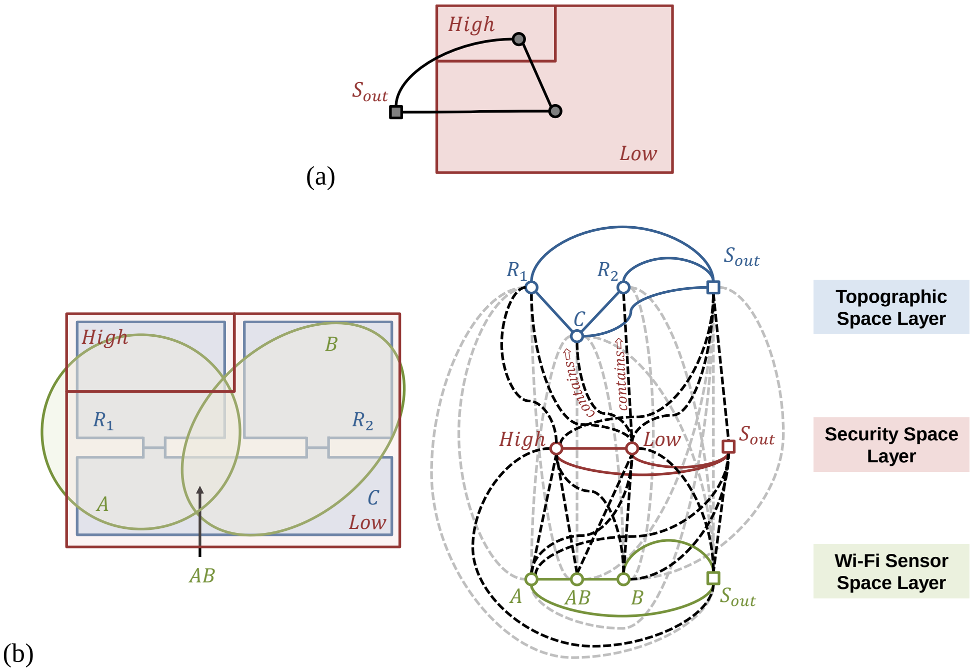


Figure 88: Adding an additional security space layer (a) to the indoor setting from figure 84 (b).

The security zones demonstrate the modelling of a logical space layer. The layout and configuration of the security space cells are not aligned with the architectural constraints and entities of the topographic space layer but follow a separate partitioning schema. This decoupled modelling approach allows for changing the setting and spatial extent of individual security zones without affecting topographic or sensor space cells. The information about the membership of topographic space cells in security zones is combinatorially encoded in the multilayered graph and thus can be efficiently evaluated by graph traversal algorithms.

The following figure 89 presents an alternative illustration of the three space layers and their related multilayered graph structure.

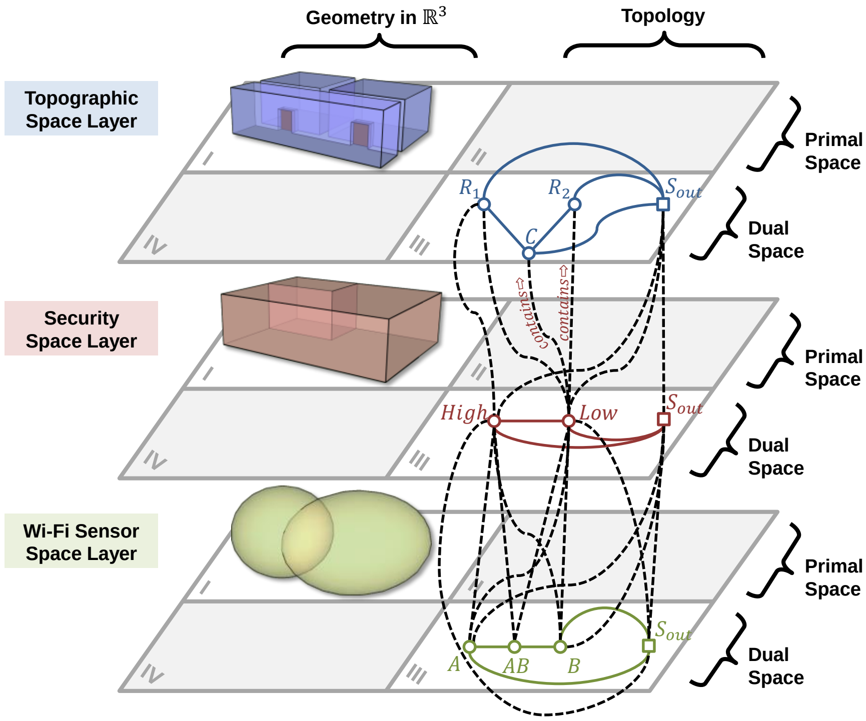


Figure 89: Alternative illustration of the space layers and their multilayered graph (excerpt) from figure 88.

The primal and dual space representations of each space layer in figure 89 are arranged in the four quadrants of the geometric-topological model for space layers (cf. figure 42). The illustration clearly depicts the coupling of the space layers in dual topology space through the edges of the inter-layer graph and their orthogonal alignment with the edges of each intra-layer graph. Note that the contents of quadrants II and IV as well as the inter-layer edges between the topographic space layer and the sensor space layer are not shown in figure 89 for readability reasons. Arbitrary further space layers may be added to this stack of layers in order to capture a comprehensive model of indoor space and to cover the information needs of a specific navigation system.

In summary, the multilayered graph as introduced in this chapter is a conceptual core element of the MLSEM. It formally maps the complex topological relationships between space cells associated with different semantics and partitioning schemas in two or three dimensions onto a single, purely combinatorial structure. Adjacencies or overlaps in primal space can be efficiently deduced from the simple information about how the dual nodes of the space cells are linked to each other in dual space. The multilayered graph can be traversed both within specific space layers by following intra-layer edges and between space layers by following inter-layer edges at the same time. This enables reasoning about all space layers or subsets thereof. Moreover, from every dual node in the multilayered graph the complementary primal and dual space representations of the corresponding space cell as well as its associated semantic and symbolic information can be accessed due to the consistent spatio-semantic model of space cells and its mathematical formalization. The same holds for the dual intra-layer edges and associated boundary cells. Therefore, the multilayered graph not only allows for traversing within and between space layers in dual space, but also for accessing and switching between the geometric-topological descriptions in primal and dual space as well as for their simultaneous evaluation.

3.3 Space-Event Modelling and the Joint State of Navigation

The multilayered graph not only encodes topological relationships between spaces but also is to be seen as a generic *space-event model*. As discussed in chapter 3.1.2.2, the intra-layer graph of a space layer equivalently describes a state diagram. An important prerequisite to this understanding is that the space cells on a space layer are enforced to be mutually non-overlapping and to render a jointly exhaustive tiling of Euclidean n-space. This space representation allows for the dual nodes of the space cells to be interpreted as mutually exclusive and jointly exhaustive states of a user travelling through the space with the dual edges denoting state transitions. Since a moving person or object physically can only be within a single space cell at a given point in time, there is only one *active state* possible at any given time.

In a multilayered setting, there is one such state diagram for each space layer participating in a space layer complex. It follows that at any point in time there is one active state per state diagram. The overall state of the moving person or object is thus given by the joint consideration of all active states on all space layers. This overall state is called the *joint state of navigation* and it is necessarily unique at any given point in time. Since the moving person or object is physically located within all active space cells simultaneously, the joint state of navigation can be spatially interpreted as the intersection of all primal space geometries associated with the active space cells. It hence mutually constrains the possible locations on all space layers. The joint state of navigation changes whenever the movement of the person or object through indoor space triggers a state transition on at least one space layer.

A joint state is a subset of the multilayered graph structure of a space layer complex. Each dual node in the multilayered graph corresponds to a distinct state and hence the node set of the multilayered graph represents the entire state space of the space-event model. A joint state is then given by a clique of dual nodes so that the clique contains exactly one node from each space layer and all nodes in the clique are mutually linked by inter-layer edges. However, not every such clique necessarily denotes a joint state. Since inter-layer edges only reflect pairwise overlaps between space cell in primal space, it is further required that the intersection of all primal space geometries of the space cells contained in the clique be non-empty. Formally, a joint state for a given set of space layers is defined as follows.

Definition 3.41 (Joint state). Let $\mathcal{L}$ be a space layer complex and $ILG(\mathcal{L})$ its inter-layer graph. Then a *joint state* JS is a pair (V, GM), where

(i) $V(JS)$ is a subset of nodes in $V(ILG)$, with $V(JS) \subseteq V(ILG)$ and $|V(JS)| = |\mathcal{L}|$, such that every two nodes in $V(JS)$ are connected by an inter-layer edge in $E(ILG)$,

(ii) $GM(JS)$ is the intersection geometry of the primal space geometries of the space cells whose dual nodes are contained in $V(JS)$ given by $GM(JS) = \bigcap_{\alpha \in I_{V(JS)}} GM(S_\alpha)$, with $v_{TP}(S_\alpha) \in V(JS)$, and

(iii) the interior of $GM(JS)$ is non-empty, and thus $Int(GM(JS)) \neq \emptyset$.

The condition (i) implies that the subgraph of $ILG(\mathcal{L})$ induced by $V(JS)$ is complete. The intersection geometry $GM(JS)$ is an n-dimensional bounded but not necessarily connected subset of $\mathbb{R}^n$ in primal space with n being the dimension of the space layer complex $\mathcal{L}$. Intuitively, $GM(JS)$ denotes the distinct partition of indoor space where a moving person or object must be physically located in when the joint state JS is triggered to be active. Since the absolute position of the person or object within the region covered by $GM(JS)$ cannot be determined more precisely without additional means, $GM(JS)$ is equivalently called *uncertainty region*. The size of $GM(JS)$ hereby provides a measure for the uncertainty of the absolute position of the person or object which has been equivalently stated for the size of the active space cell on each space layer in chapter 3.1.2.2. However, since the intersection geometry $GM(JS)$ can at most be equal to the geometry of the smallest space cell participating in JS, it typically represents a smaller partition of indoor space than each of the active space cells. Therefore, the remaining uncertainty region can be decreased by both smaller space cells on individual space layers and highly overlapping space cells between several space layers. The uncertainty of the absolute position is hence correlated with the granularity of the space partitioning.

The following figure 90 shows an artificial space layer complex $\mathcal{L}$ containing three space layers each of which contains a single space cell. As shown on the left, the space cells pairwise overlap which yields the multilayered graph depicted on the right. Although the dual nodes $v_{TP}(A)$, $v_{TP}(B)$, and $v_{TP}(C)$ are a clique of $ILG(\mathcal{L})$ they do not constitute a valid joint state due to the intersection $GM(A) \cap GM(B) \cap GM(C)$ being the empty set which violates condition (iii) of the above definition. This conforms to the obvious fact that a navigation user cannot be located in the space cells A, B, and C at the same time. All other cliques in $ILG(\mathcal{L})$ render valid joint states, for example, consider $V(JS) = \{v_{TP}(A), v_{TP}(B)v_{TP}(S_{out}^C)\}$ whose uncertainty region $GM(JS)$ is drawn as hatched area on the left of figure 90.

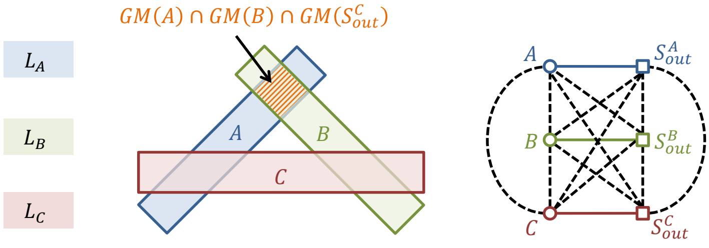


Figure 90: Example for a clique $\{v_{TP}(A), v_{TP}(B), v_{TP}(C)\}$ which does not render a valid joint state.

The set of all joint states for a given space layer complex $\mathcal{L}$ constitutes the *joint space state* of $\mathcal{L}$ and is defined as follows.

Definition 3.42 (Joint state space of a space layer complex). The collection $\mathcal{JS} = \{JS_\alpha\}_{\alpha \in I_{\mathcal{JS}}}$ of all joint states over the inter-layer graph $ILG(\mathcal{L})$ of a space layer complex $\mathcal{L}$ with each joint state JS_α satisfying definition 3.41 is said to be the *joint state space* of $\mathcal{L}$.

It is an immediate consequence of conditions (ii) and (iii) of definition 3.41 that the intersection geometries associated with the joint states in the joint state space $\mathcal{JS}(\mathcal{L})$ are pairwise non-overlapping and at most touch at their boundaries, and that their spatial configuration yields a complete tiling of $\mathbb{R}^n$. Otherwise the joint states contained in $\mathcal{JS}$ would not be mutually exclusive and jointly exhaustive.

In general, the number of joint states in $\mathcal{JS}(\mathcal{L})$ is unbounded but can be expressed exactly if $\mathcal{L}$ contains less than three space layers. Precisely, let $k = |\mathcal{L}|$ be the number of space layers in $\mathcal{L}$. Then obviously $|\mathcal{JS}(\mathcal{L})| = 0$ if $k =$

0. In case $k = 1$, the number of joint states is necessarily equal to the number of dual nodes in $V(ILG(\mathcal{L}))$. If $\mathcal{L}$ contains exactly two space layers, then condition (iii) of definition 3.41 is automatically satisfied for all pairs of dual nodes linked by an inter-layer edge, and the number of joint states is equal to the number of edges in $E(ILG(\mathcal{L}))$. However, for $k > 2$, only a lower limit for the number of joint states can be stated. Due to the complete covering of $\mathbb{R}^n$ on each space layer in $\mathcal{L}$, every dual node in $V(ILG(\mathcal{L}))$ participates in at least one joint state. Let $V_1, \dots, V_k$ be the disjoint partitions of nodes in the k-partite graph $ILG(\mathcal{L})$, then the minimal number of joint states is equal to the number of nodes in the smallest partition. Thus,

$$\begin{aligned} |\mathcal{JS}(\mathcal{L})| &= \begin{cases} 0, & k = 0; \\ \left|V\big(ILG(\mathcal{L})\big)\right|, & k = 1; \\ \left|E\big(ILG(\mathcal{L})\big)\right|, & k = 2; \end{cases} \text{ and} \\ |\mathcal{JS}(\mathcal{L})| &\geq \ |V_i|, \text{with } \forall i \neq j \in \{1, \dots, k\}\colon |V_i| \leq \left|V_j\right| \wedge V_i, V_j \in V\big(ILG(\mathcal{L})\big), \quad k > 2. \end{aligned} \tag{3.43}$$

The lower limit $|\mathcal{JS}(\mathcal{L})| = 1$ only occurs when $\mathcal{L}$ contains k copies of the minimal space layer L_{min} (cf. definition 3.32) and $k > 0$.

If presupposing a continuous movement through space, then, for a given space layer in $\mathcal{L}$, a navigation user either remains within the same space cell or has moved to an adjacent space cell in two consecutive joint states. The set of possible *joint state transitions* therefore results from the joint consideration of the intra-layer edges on each space layer being incident to the dual nodes participating in the joint states. Formally, given two joint states $JS_1, JS_2 \in \mathcal{JS}(\mathcal{L})$, a joint state transition between JS_1 and JS_2 implies that the dual nodes participating in JS_1 are either equal to or adjacent to the dual nodes in JS_2 on the same space layer. Moreover, it requires that the intersection geometries of both JS_1 and JS_2 be topologically adjacent in order to ensure a continuous movement.

Definition 3.44 (Joint state transition). Let $\mathcal{JS}(\mathcal{L})$ be the joint state space of a space layer complex $\mathcal{L}$. Then, a *joint state transition* $JST = \{JS_1, JS_2\}$ is a 2-element subset of $\mathcal{JS}(\mathcal{L})$, with $JST \subseteq \mathcal{JS}(\mathcal{L})$ and $JS_1 \neq JS_2$, for which the following two conditions hold:

(i) Let $V_1(MLG), \dots, V_k(MLG)$ be the disjoint node partitions of the multilayered graph $MLG(\mathcal{L})$ of $\mathcal{L}$. Then, $\forall v_1 \in V(JS_1), v_2 \in V(JS_2)\colon v_1, v_2 \in V_i(MLG) \Rightarrow v_1 = v_2 \vee \{v_1, v_2\} \in E(MLG)$, with $1 \leq i \leq k = |\mathcal{L}|$.

(ii) The intersection of the boundaries of $GM(JS_1)$ and $GM(JS_2)$ is non-empty, and thus $\partial GM(JS_1) \cap \partial GM(JS_2) \neq \emptyset$.

Example 3.45. The conceptual idea of joint states is illustrated in the following along the 2-dimensional indoor setting presented in figure 91.[18] Since the space layer complex $\mathcal{L}$ contains two space layers in this example, the number of overall joint states on $\mathcal{L}$ is given by $\left|E\big(ILG(\mathcal{L})\big)\right| = 13$. Suppose a navigation user is positioned in room R_1 and carries a Wi-Fi-enabled mobile device which only receives the signal of Wi-Fi transmitter A. Then the *active joint state* $JS_0 \in \mathcal{JS}(\mathcal{L})$ of the user can be spatially interpreted as being simultaneously in the topographic space cell R_1 and the sensor space cell A, and is equivalently denoted as (R_1, A) in the following. Formally, the joint state is given by the clique $V(JS_0) = \{v_{TP}(R_1), v_{TP}(A)\}$ and the intersection geometry $GM(JS_0) = GM(R_1) \cap GM(A)$. In figure 91, $GM(JS_0)$ is depicted as hatched area on the left whereas the joint state clique $V(JS_0)$ is highlighted in the multilayered graph on the right.

As stated above, the intersection geometry $GM(JS_0)$ represents the region of indoor space where the navigation user must be physically located in. Since the sensor space cell A only covers a part of room R_1, this region is obviously smaller than the space occupied by R_1 itself. The joint consideration of both space cells thus reduces the uncertainty about the user's absolute position within R_1 to the uncertainty region $GM(JS_0)$. The remaining parts of R_1 are described by the two distinct joint states (R_1, AB) and (R_1, S_{out}^{Wi-Fi}).

[18] Note that in two dimensions it takes at minimum two 1-cells to describe the common boundary of the space cell C with S_{out}^{Topo} on the topographic space layer. Its dual node is hence linked to S_{out}^{Topo} through two dual edges in this example in contrast to the dual graph shown in figure 87 (cf. chapter 3.2) for the equivalent 3-dimensional setting.

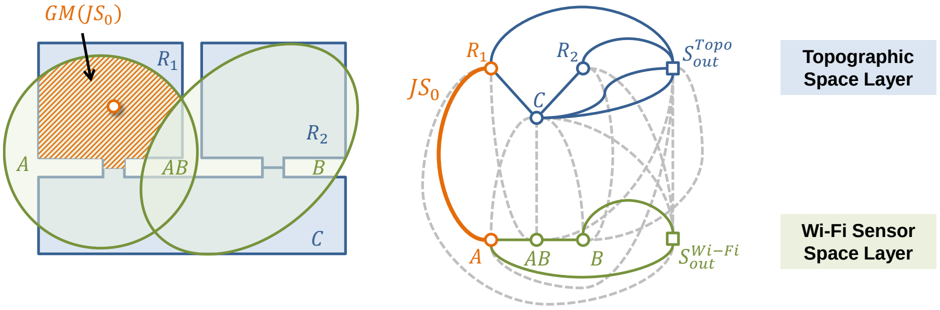
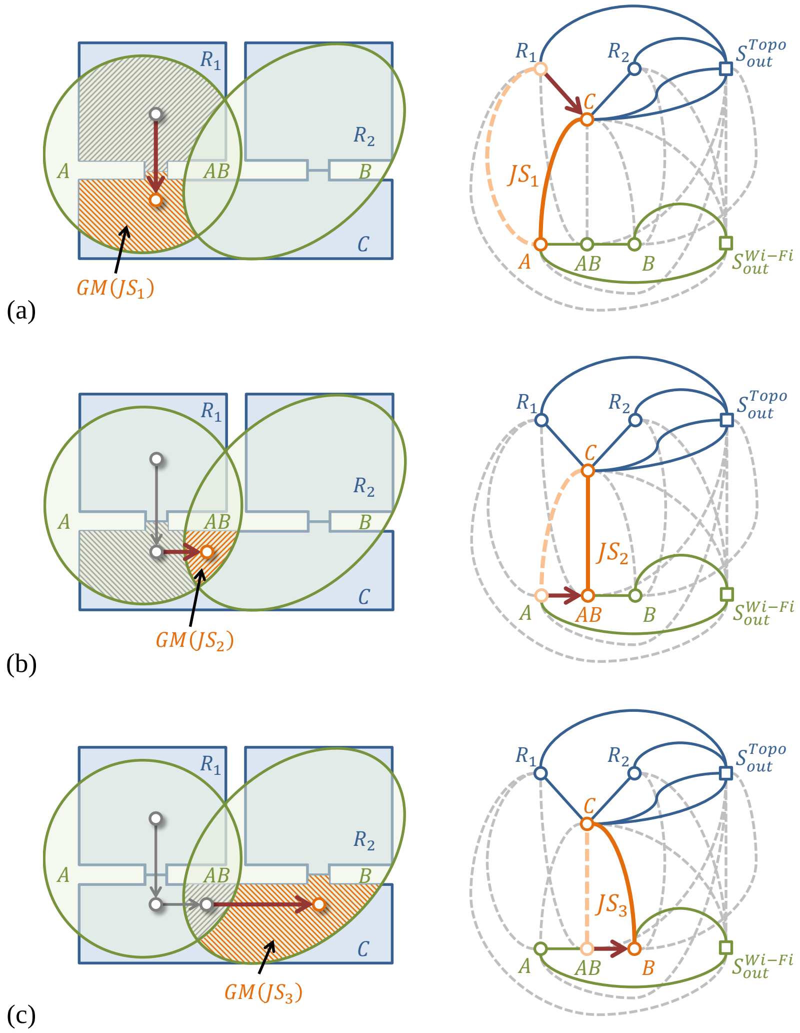


Figure 91: Uncertainty region (left) and graph representation (right) of the joint state (R_1, A).

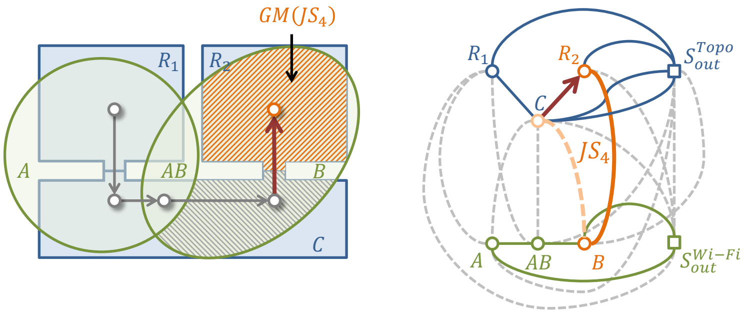
As soon as the navigation user leaves room R_1 and enters the corridor C, this movement triggers a joint state transition $JST_0 = \{JS_0, JS_1\}$ that yields the new active joint state JS_1 given by the pair (C, A). The dual nodes of R_1 and C are linked by a dual edge in the intra-layer graph of the topographic space layer, whereas the user remains within space cell A on the sensor space layer. Moreover, $GM(JS_0)$ and $GM(JS_1)$ share a part of their boundary where the primal space geometries of R_1 and C touch. It follows that both the combinatorial condition (i) and the geometric condition (ii) of definition 3.44 are obeyed and JST_0 is a valid joint state transition. On the left of figure 92a, the movement of the navigation user as well as the primal space geometry $GM(JS_1)$ of the new active joint state are illustrated. The right of figure 92a shows both the previous joint state JS_0 and the active joint state JS_1 as well as the transition between R_1 and the C in the multilayered graph. In figure 92b to d, consecutive joint states and their transitions are sketched that reflect a continuous movement of the navigation user to R_2.



(a)

(b)

(c)



(d)

Figure 92: Subsequent joint states and corresponding uncertainty areas (a to d) as the navigation users travels through indoor space.

The entire movement of the navigation user along the depicted trajectory in the above example can be described as a chain of joint states $(R_1, A) \rightarrow (C, A) \rightarrow (C, AB) \rightarrow (C, B) \rightarrow (R_2, B)$ with each arrow denoting a valid joint state transition between two consecutive joint states. Intuitively, a joint state transition is executed when an *event* on at least one of the space layers occurs that is induced by the movement of the user. Examples for topographic space events for the given chain are *"Entering corridor from Room 1"* or *"Entering Room 2 from the corridor"*, whereas events on the sensor space layer include *"Receiving signal from Wi-Fi transmitters A and B"* or *"Receiving signal from Wi-Fi transmitter A only"*. This event-driven chain of joint states can be mathematically described through a *finite-state machine*.

Definition 3.46 (Finite-joint-state machine). Let $\mathcal{L}$ be a space layer complex. Then a *finite-joint-state machine* $FJSM(\mathcal{L})$ for $\mathcal{L}$ is a quintuple $(\Sigma, \mathcal{JS}(\mathcal{L}), \mathcal{JS}_0, \phi, F)$ where

(i) Σ is a finite, non-empty set of events,

(ii) $\mathcal{JS}(\mathcal{L})$ is the finite, non-empty joint state space of $\mathcal{L}$,

(iii) $\mathcal{JS}_0$ is a finite, non-empty set of initial states of $FJSM(\mathcal{L})$, with $\mathcal{JS}_0 \subseteq \mathcal{JS}(\mathcal{L})$,

(iv) ϕ is a map $\phi: \mathcal{JS}(\mathcal{L}) \times 2^{\Sigma} \rightarrow 2^{\mathcal{JS}(\mathcal{L})}$ (called the *joint-state-transition map*) which returns a set of possible next active joint states for $FJSM(\mathcal{L})$ given its active joint state $p \in \mathcal{JS}(\mathcal{L})$ and a finite subset of events $E \subseteq \Sigma$, so that $\phi(p, E) \subseteq \mathcal{JS}(\mathcal{L})$ and the set $\{p, q_i\}$ is a valid joint state transition satisfying definition 3.44 for each $q_i \in \phi(p, E)$, and

(v) F is a set of final joint states with $F \subseteq \mathcal{JS}(\mathcal{L})$ and $|F| \geq 0$.

Given a finite-joint-state machine $FJSM(\mathcal{L})$ for a space layer complex $\mathcal{L}$, an ordered set of n consecutive joint states $\{JS_0, JS_1, \dots, JS_n\} \in \mathcal{JS}(\mathcal{L})$, with JS_0 being the initial state and JS_n being the final state, is then chained by the joint-state-transition map ϕ as shown in the following diagram:

$$JS_0 \xrightarrow{\phi(JS_0,E_0)} JS_1 \xrightarrow{\phi(JS_1,E_1)} \dots JS_{n-1} \xrightarrow{\phi(JS_{n-1},E_{n-1})} JS_n. \tag{3.47}$$

Every element in this chain represents the active joint state of $FJSM(\mathcal{L})$ at a given point in time t. It follows that every active joint state JS_n at t_n only depends on the previous joint state JS_{n-1} at t_{n-1}as well as the set of events $E_{n-1} \in \Sigma$ that trigger the joint state transition.

$FJSM(\mathcal{L})$ is said to be a *non-deterministic* automaton since the co-domain of the joint-state-transition map ϕ is defined to be the power set $2^{\mathcal{JS}(\mathcal{L})}$ of the joint space state of $\mathcal{L}$ instead of being $\mathcal{JS}(\mathcal{L})$ itself. Thus, ϕ may not only return a single next active joint state for $FJSM(\mathcal{L})$ like in case of a deterministic automaton but rather a set of *candidate states* which includes the empty set. The reason for modelling $FJSM(\mathcal{L})$ non-deterministic is that a deterministic automaton would require full knowledge about all events that simultaneously occur on every space layer in $\mathcal{L}$ and together cause the joint state transition. However, in practice, typically a sparse subset of events is observable or available yielding more than one possible transition. For the same reason, the initial state $\mathcal{JS}_0$ of $FJSM(\mathcal{L})$ is defined to be a non-empty set of candidate states instead of a single and deterministic initial state. In the following, the impact of sparse events is illustrated along a slight modification of the above example 3.45.

Example 3.48. Let $FJSM(\mathcal{L})$ be a finite-joint-state machine for the indoor setting illustrated in example 3.45. Further, let $\Sigma = \{"Receiving\ A\ only", "Receiving\ B\ only", "Receiving\ A\ and\ B", "No\ signal"\}$ denote the set of possible events for $FJSM(\mathcal{L})$ with each event reflecting the currently received signals from both Wi-Fi transmitters A and B. Assume the Wi-Fi-enabled mobile device of a navigation user is within the vicinity of Wi-Fi transmitter A only. If we additionally suppose that we cannot tell whether the user is located in room R_1, then there exist two possible initial joint states $JS_0^\alpha \leftarrow (R_1, A)$ and $JS_0^\beta \leftarrow (C, A)$ due to the dual node of the sensor space cell A being adjacent to the dual nodes of R_1 and C on the topographic space layer in the multilayered graph. Note that the dual node of A is additionally linked to the dual node of the outer topographic space. However, being out of the topographic indoor space can usually be regarded an invalid state in the context of indoor navigation and thus the joint state (S_{out}^{Topo}, A) is not considered in this example. Without additional a priori knowledge or input (e.g., from the navigation user), the candidates JS_0^α and JS_0^β are equally likely with $\mathcal{JS}_0 = \{JS_0^\alpha, JS_0^\beta\}$ denoting the initial state of $FJSM(\mathcal{L})$ (cf. figure 93). The uncertainty region $GM(\mathcal{JS}_0)$ of the navigation user obviously follows as $GM(\mathcal{JS}_0) = GM(JS_0^\alpha) \cup GMJS_0^\beta$ and is depicted as hatched area on the left of figure 93.

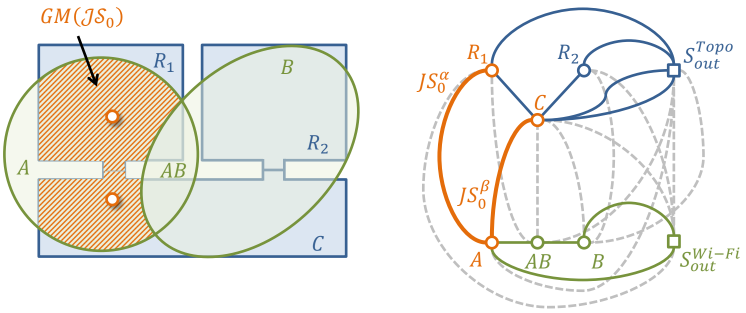


Figure 93: Example of a non-deterministic active joint state of the navigation user.

Suppose the navigation user started in room R_1 and continuously moves along the same trajectory as given in example 3.45. Then the corridor is entered next and the corresponding topographic space event *"Entering corridor from Room 1"* could trigger a joint state transition for $FJSM(\mathcal{L})$. However, this event is not observed by a sensor and is typically neither reported by the user in a real navigation scenario. Thus, it is not contained in Σ in this example, and $FJSM(\mathcal{L})$ remains in $\mathcal{JS}_0$. The next observable event is rather fired by the mobile device on the sensor space layer which receives the signal of Wi-Fi transmitter B in addition to that of A when the user moves along the corridor. The two possible candidates $JS_1^\alpha \leftarrow (R_1, AB)$ and $JS_1^\beta \leftarrow (C, AB)$ for the next active joint state of $FJSM(\mathcal{L})$ can be deduced by simply evaluating the inter-layer edges between sensor space cell AB and the topographic space layer. However, both candidates must additionally satisfy the joint-state-transition map ϕ.

Let $e_0 =$ "$Receiving\ A\ and\ B$" denote the captured event. Then, $\phi(JS_0^\alpha, e_0) = \{JS_1^\alpha\}$ only returns JS_1^α as possible next active joint state due to the alternative joint state transition $\{JS_0^\alpha, JS_1^\beta\}$ violating the geometric condition (ii) of definition 3.44. Thus, if JS_0^α is supposed to be the initial active state of $FJSM(\mathcal{L})$ then the joint state chain $JS_0^\alpha \xrightarrow{\phi(JS_0^\alpha, e_0)} JS_1^\alpha$ is yielded. Likewise, if JS_0^β is taken as initial active state then $\phi(JS_0^\beta, e_0) = \{JS_1^\beta\}$ and $JS_0^\beta \xrightarrow{\phi(JS_0^\beta, e_0)} JS_1^\beta$ since there is no valid transition between JS_0^β and JS_1^α. It obviously follows that a decision between both candidates cannot be made if JS_0^α and JS_0^β remain equally likely. The situation is sketched in the following figure.

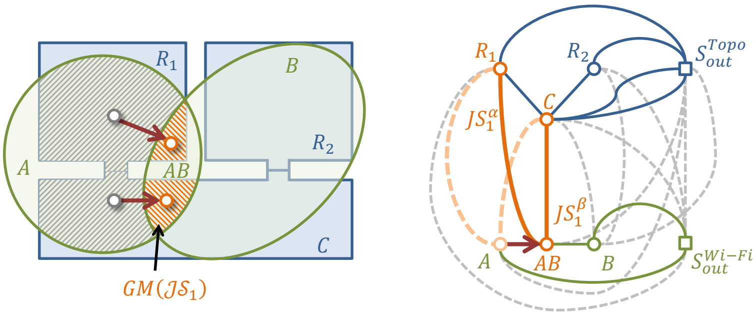


Figure 94: A non-deterministic active joint state following from the non-deterministic predecessor in figure 93.

The active joint state can be deterministically derived if $e_1 =$ "$Receiving\ B\ only$" is considered the next event. Possible candidates for this event taken from the multilayered graph are $JS_2^{\alpha} \leftarrow (R_2, B)$ and $JS_2^{\beta} \leftarrow (C, B)$. However, $\phi(JS_1^{\alpha}, e_1) = \emptyset$ since both the transitions $\{JS_1^{\alpha}, JS_2^{\alpha}\}$ and $\{JS_1^{\alpha}, JS_2^{\beta}\}$ already violate condition (i) of definition 3.44. An empty set resulting from ϕ can be regarded an invalid state and thus can be announced as error by $FJSM(\mathcal{L})$. In contrast, $\phi\left(JS_1^{\beta}, e_1\right) = \{JS_2^{\beta}\}$ yields a valid result which thus becomes the active joint state of $FJSM(\mathcal{L})$ with the corresponding joint state chain $JS_0^{\beta} \xrightarrow{\phi(JS_0^{\beta},e_0)} JS_1^{\beta} \xrightarrow{\phi(JS_1^{\beta},e_1)} JS_2^{\beta}$. Note that this joint state chain does not reflect the initial state of the navigation user whom we assumed to start in $JS_0^{\alpha} \leftarrow (R_1, A)$. However, first, this is irrelevant for considering the next active joint state JS_3 as this only depends on JS_2^{β}. Second, JS_0^{α} is also contained in $\mathcal{JS}_0$ and is a valid predecessor for JS_0^{β}. Thus, the joint state chain can be easily extended to JS_0^{α} $\xrightarrow{\phi(JS_0^{\alpha},\emptyset)} JS_0^{\beta} \xrightarrow{\phi(JS_0^{\beta},e_0)} JS_1^{\beta} \xrightarrow{\phi(JS_1^{\beta},e_1)} JS_2^{\beta}$.

Even if the active joint state JS_n of $FJSM(\mathcal{L})$ is known, the finite-joint-state machine remains non-deterministic which means that $\phi(JS_n, E \in \Sigma)$ may again return a set of candidates for the next active joint state JS_{n+1}. Likewise, and in contrast to the above example, it cannot be guaranteed that for a given sequence of events the active joint state can be determined at any given point in time. For example, assume no overlap between sensor space cells A and B in the above indoor setting as shown in figure 95. This consequently reduces the possible events to the set $\Sigma = \{$"$Receiving\ A\ only$", "$Receiving\ B\ only$", "$No\ signal$"$\}$. Then, for any sequence of events, only candidate sets for the active joint state are derived.

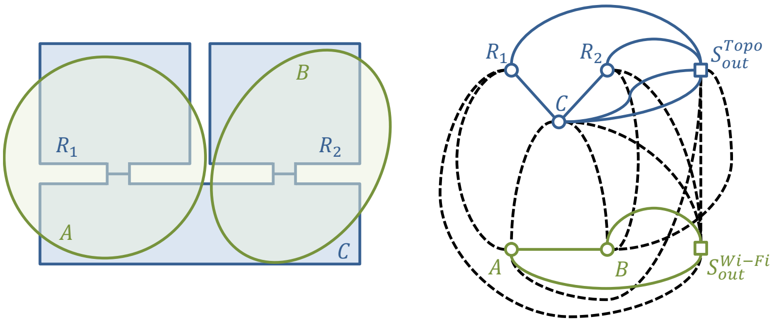


Figure 95: Example for a setting where no sequence of events on the Wi-Fi sensor space layer leads to a deterministic active joint state.

The finite-joint-state machine can be used as simulation tool to analyse and detect non-deterministic active joint states and to minimize them in real navigation systems. Intuitively, the reduction of the number of non-deterministic outputs of ϕ requires further joint states and/or additional events in Σ. The following figure 96 demonstrates this by adding an RFID sensor space layer capturing four low range RFID sensors to the example space model.

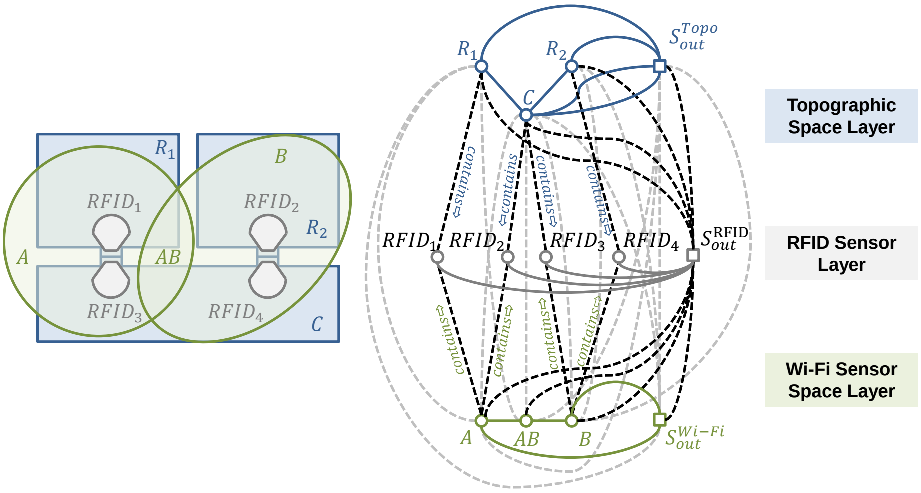


Figure 96: Adding an RFID sensor space layer to the example indoor scene.

The RFID sensors are mounted above each door within both rooms and the corridor. The corresponding sensor space cells reflect the signal reception areas of the RFID sensors, and the five events $E_{RFID} = \{"RFID_1", "RFID_2", "RFID_3", "RFID_4", "No\ RFID\ reception",\}$ are added to the set of events Σ of $FJSM(\mathcal{L})$. Assume that the mobile device of the navigation user is capable of processing both Wi-Fi and RFID signals. If the mobile device only receives the signal from Wi-Fi transmitter A, then again two possible initial joint states $JS_0^\alpha \leftarrow (R_1, A, S_{out}^{RFID})$ and $JS_0^\beta \leftarrow (C, A, S_{out}^{RFID})$ are yielded which are depicted in figure 97a. However, note that the uncertainty regions $GM(JS_0^\alpha)$ and $GM(JS_0^\beta)$ are smaller than in the setting without the additional RFID space layer, and consequently the user cannot be located in front of the door between room R_1 and the corridor as this would mean that an RFID signal is received. If $e_0 = "RFID_1"$ is fired as next event, the active joint state can already be uniquely identified as $JS_1 \leftarrow (R_1, A, RFID_1)$ with JS_0^α being its predecessor. Moreover, the uncertainty about the absolute position of the user is further reduced due to the small spatial extent of $GM(JS_1)$. Although it conforms to an intuitive understanding that the localization precision increases within the reception area of low range RFID sensors, the multi-layered graph as well as the notion of joint states defined on top of this graph provides the means to quantitatively describe this increase based on a mathematically consistent formal model.

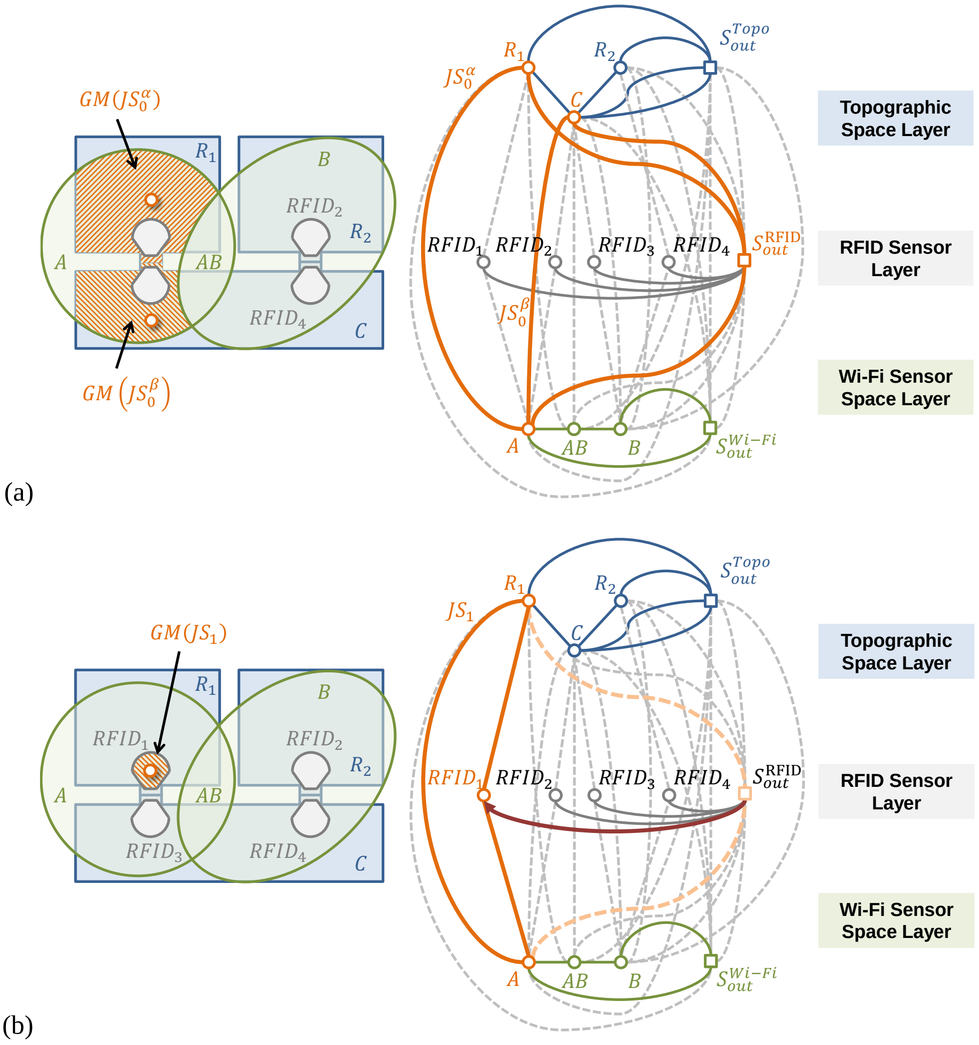


Figure 97: Whereas the initial joint state remains non-deterministic (a), it becomes deterministic in the next step based on the additional consideration of the RFID sensor layer (b).

A similar result can be, for example, achieved by alternatively (or additionally) placing barcodes (QR-codes, etc.) next to each door which could be scanned by the moving person or object using an RGB sensor of the mobile device. In a corresponding sensor space layer, the space cells would then denote that portion of space where the user must be physically located in order to interact with the barcode. According to the conceptual model of (Bhatt et al. 2009) (cf. chapter 2.1.3), this space can be classified as *functional space* of the barcode. Since barcodes are a passive localization method, the interaction of the user is required. A navigation system detecting a sequence or loop of non-deterministic candidates for its active joint state could thus ask the user to scan a nearby barcode in order to be able to pick one of the candidates and to derive a deterministic active joint state.

This example again demonstrates the strengths and benefits of the proposed multilayered approach to indoor space modelling. First, the space layers are modelled independently from each other, and thus adding a new space layer or changing the layout or configuration of space cells on one space layer does not interfere with the other space layers. Second, multiple and different localization methods can be represented in order to enhance the localization precision but also to support different capabilities of the end-user device.

Example 3.49. In addition to non-deterministic joint states also contradictory consecutive joint states are possible in a real navigation scenario. Suppose for the above example 3.48 that two consecutive events $e_0 =$ "$Receiving\ A\ only$" and $e_1 =$ "$Receiving\ B\ only$" occur at subsequent points in time t_0 and t_1 with no event being fired in between. Then there are no two joint states JS_1 and JS_2 in the entire joint state space $\mathcal{JS}(\mathcal{L})$ so that JS_1 contains the dual node of sensor space cell A, JS_2 contains the dual node of sensor space cell B, and $\{JS_1, JS_2\}$ is a valid joint state transition. This simply follows from the fact that the dual nodes of A and B are not linked in the Poincaré dual graph of the sensor space layer and any pair $\{JS_1, JS_2\}$ necessarily disobeys condition (i) of definition 3.44, and thus $\phi(JS_1\ , e_1) = \emptyset$. Figure 98 visually supports this result. However, the sequence of events may nevertheless occur, for example, in case the navigation user temporarily switches off the mobile device whilst moving from room R_1 to room R_2.

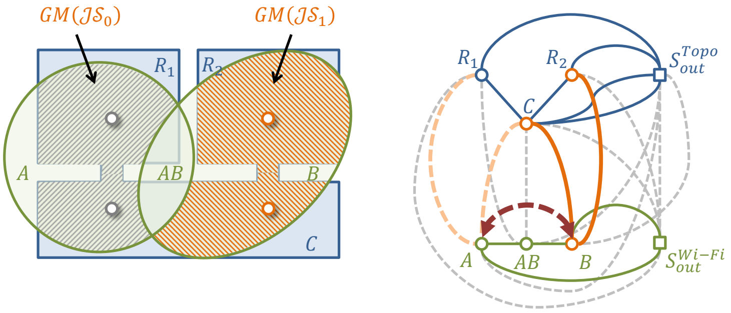


Figure 98: Example of contradictory consecutive joint states.

The result $\phi(JS_1\ , e_1) = \emptyset$ of the joint-state-transition map is consistent. The main assumption underlying the definition of joint state transitions (cf. definition 3.44) and thus implicitly of the finite-joint-state machine (cf. definition 3.46) is a *continuous movement* through indoor space. In topographic space, this refers to the continuous bodily movement of a person or object through the built reality (e.g., a user physically cannot enter R_1 from R_2 without moving along the corridor C beforehand). On a sensor space layer, this requires a continuous mapping of the physical sensor characteristics captured by the sensor space cells. Whereas the latter is violated in the illustrated scenario, invalid consecutive states on a topographic space layer are, for example, discussed as *kidnapped robot problem* (e.g., Engelson & McDermott 1992) in the field of mobile robotics.

For the finite-joint-state machine $FJSM(\mathcal{L})$, an empty set returned by ϕ represents an invalid state because $FJSM(\mathcal{L})$ must be in an active joint state at any given point in time even if this state cannot be uniquely determined through ϕ. To solve this problem for the illustrated scenario, $FJSM(\mathcal{L})$ can simply be re-initialized using the set of joint states containing the dual node of sensor space cell B as new initial joint state $\mathcal{JS}_0$ since e_1 reflects a valid event triggered on the sensor space layer according to which the user must be somewhere in $GM(\mathcal{JS}_0)$. And since the next active joint state only depends on the new initial state $\mathcal{JS}_0$, the lack of signal reception before $\mathcal{JS}_0$ has no further impact.

3.4 Subspacing and Hierarchical Structures

In chapter 2.4, the multi-granular representation of indoor space has been identified as an essential requirement to indoor space modelling in the context of indoor navigation. Whereas the previous chapters on the MLSEM rather focused on the modelling of different notions of space on separate space layers, a multi-granular space representation addresses the same notion of space but different decompositions thereof following from alternative partitioning schemas. Examples are complementary decompositions of topographic space in order to reflect navigable and non-navigable spaces for different modes of locomotion such as pedestrians, wheelchair users, driving or flying autonomous robots, or along additional user-centric or environmental criteria. Likewise, topographic space may be organized hierarchically on multiple levels of detail so that fine-grained entities such as (parts of) rooms and corridors are related to spatial aggregates on coarser levels like building floors and wings. Such hierarchical structures are prerequisites for range queries or multi-level path queries and rely upon *containment* relationships between the entities. As discussed in chapter 2.2, many hybrid indoor space models implicitly employ or at least recognize the need for multi-granular and hierarchical space structures but only few proposals provide formal definitions and algorithms.

The decomposition of a space cell A into smaller, more fine-grained parts in primal space is called *subspacing* within the MLSEM. If a part B, which results from a subspacing of A, itself fulfils the requirements of a space cell as given in chapter 3.1.1.3, then B is called a *subspace cell* of A, and A is said to be the *superspace cell* of B. Intuitively, a subspace cell is spatially contained inside its superspace cell. In order to make this *containment* relationship explicit in the dual space representation, both space cells need to be modelled on different space layers in a space layer complex since the Poincaré dual graph on a single space layer can only express the *connectedness* between space cells in terms of topological adjacency. In fact, the inter-layer edges of the multilayered graph as defined in chapter 3.2 already capture general *overlap* relationships between space cells including spatial containment (cf. definition 3.37). The modelling of subspace cells on a separate space layer is also consistent with the core concept of the structured space model according to which different partitionings of indoor space are reflected by different space layers. A space layer all of whose space cells have superspace cells on a second space layer is equivalently said to be a *subspace layer*. It follows that different subspacings along different contextual criteria like those illustrated above necessarily have to be mapped by separate subspace layers.

The following figure sketches the conceptual idea of a multi-granular space representation within the MLSEM based on the modelling of subspace layers. The different decompositions of indoor space shown in this figure are arranged in a tree diagram with each tree node representing a separate space layer. The space representation becomes more fine-grained towards the leaves of the tree at the bottom of the diagram. The edges between space layers denote subspace relationships based on the spatial containment between space cells, whereas the space cells on space layers on the same hierarchical level of the diagram only exhibit general overlap relationships. The figure illustrates that subspacing is mostly relevant for topographic space but is not restricted to that in the context of the MLSEM. The example subspace hierarchy of the topographic space assumes a decomposition of the interior built environment into building floors and overlapping building wings. Further subdivisions of the building floors then account for the navigable spaces for different modes of locomotion as well as for arbitrary groups of space entities (e.g., functional, organizational, and logical groups). Of course, the modelled space views and their hierarchical structuring depend on the navigation system to be implemented and thus may, of course, differ from this example.

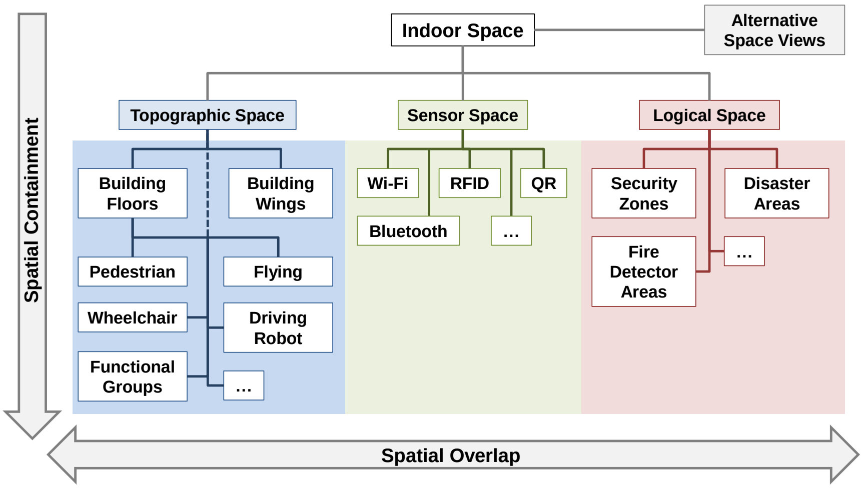


Figure 99: Example of subspace hierarchies based on spatail containment relationships.

Although the MLSEM as introduced so far already supports subspace cells and subspace layers implicitly, both concepts as well as related properties are formally defined and presented in the following.

Definition 3.50 (Subspace cell, superspace cell). If A and B are two n-dimensional space cells with $GM(A) \subseteq GM(B)$, then A is called a *subspace cell* of B, denoted by $A \leq B$. Equivalently, B is said to be a *superspace cell* of A, denoted by $B \geq A$.

The binary relation $\leq$ induces an inclusion relation between A and B which is a consequence of the primal space geometry $GM(A)$ being a subset of $GM(B)$ in Euclidean space $\mathbb{R}^n$.

Definition 3.51 (Subspace layer, superspace layer). Let $K, L \in \mathcal{L}$ be two space layers participating in an n-dimensional space layer complex $\mathcal{L}$. Further, let $\mathcal{C}(K)$ and $\mathcal{C}(L)$ be the space cell complexes of K respectively L. Then K is called a *subspace layer* of L, denoted by $K \leq L$, iff $\forall S_K \in \mathcal{C}(K)\ \exists\, S_L \in \mathcal{C}(L)\colon S_K \leq S_L$. Equivalently, L is said to be a *superspace layer* of K, denoted by $L \geq K$.

Similar to subspace cells, the binary relation $\leq$ denotes an inclusion relation between the space layers K and L which follows from the fact that the primal space geometry of each space cell on K is a subset of the primal space geometry of a space cell on L in $\mathbb{R}^n$. Note that $K \leq L$ implies that the primal space geometry of the space cell complex of K is a subset of that of L, so that $K \leq L \Rightarrow GM(\mathcal{C}(K)) \subseteq GM(\mathcal{C}(L))$. But in contrast, $GM(\mathcal{C}(K)) \subseteq GM(\mathcal{C}(L))$ is not a sufficient condition for $K \leq L$.

Proposition 3.52 (Poset of space cells over different space layers). Let $\mathcal{L}$ be a space layer complex and $\mathfrak{C}(\mathcal{L}) = \bigcup_{\alpha \in I_{\mathcal{L}}} \mathcal{C}(L_\alpha)$ be the set of all space cells contained in $\mathcal{L}$ (excluding the outer space cells). Then the pair $\mathcal{P} = (\mathfrak{C}(\mathcal{L}), \leq)$ renders a *partially ordered set* (also called *poset*) with $\mathfrak{C}(\mathcal{L})$ being its ground set and the inclusion relation $\leq$ imposing a *partial order* on the set of space cells $\mathfrak{C}(\mathcal{L})$ so that for all S_1, S_2, S_3 in $\mathcal{P}$ it holds that

(i) $S_1 \leq S_1$ (reflexivity),
(ii) $S_1 \leq S_2 \wedge S_2 \leq S_1 \Rightarrow S_1 = S_2$ (antisymmetry), and
(iii) $S_1 \leq S_2 \wedge S_2 \leq S_3 \Rightarrow S_1 \leq S_3$ (transitivity).

It follows that the space cells in a set of space layers can be partially ordered by inclusion or, equivalently, by their spatial containment. Two space cells $S_1, S_2 \in \mathcal{P}$ are obviously only comparable, i.e. $S_1 \leq S_2$ or $S_2 \leq S_1$, if they belong to different space layers since the space cells on the same space layer are not allowed to spatially overlap and thus only $S_1 = S_2$ may occur. For two space layers $L_1, L_2 \in \mathcal{L}$ with $L_1 \leq L_2$ it is immediate that all space cells in $\mathcal{C}(L_1)$ are comparable to the space cells in $\mathcal{C}(L_2)$.

Due to the inclusion relation between subspace layers, also a set of space layers can be arranged along the spatial containment of its elements.

Proposition 3.53 (Poset of space layers). Let $\mathcal{L}$ be a space layer complex. Then the pair $\mathcal{Q} = (\mathcal{L}, \leq)$ renders a *partially ordered set* with $\mathcal{L}$ being its ground set and the inclusion relation $\leq$ imposing a *partial order* on the set of space layers $\mathcal{L}$ so that for all L_1, L_2, L_3 in $\mathcal{Q}$ it holds that

(i) $L_1 \leq L_1$ (reflexivity),
(ii) $L_1 \leq L_2 \wedge L_2 \leq L_1 \Rightarrow L_1 = L_2$ (antisymmetry), and
(iii) $L_1 \leq L_2 \wedge L_2 \leq L_3 \Rightarrow L_1 \leq L_3$ (transitivity).

Based on the spatial ordering of space cells and space layers, *hierarchical structures* between them can be defined as follows.

Definition 3.54 (Greatest space cell, hierarchy of a space cell). Let $\mathcal{L}$ be a space layer complex and let $\mathcal{P} = (\mathfrak{C}(\mathcal{L}), \leq)$ be the poset of the space cells in $\mathcal{L}$. Then a space cell S contained in a subset $\mathcal{H} \subseteq \mathcal{P}$ is said to be the *greatest space cell* in $\mathcal{H}$ if any space cell in $\mathcal{H}$ is transitively included in S, and thus $\forall S_i \in \mathcal{H}\colon S_i \leq S$. The subset $\mathcal{H}$ is said to be a *hierarchy* of the space cell S.

Definition 3.55 (Strict hierarchy of a space cell, sub-hierarchy). Let $\mathcal{H}(S)$ be a hierarchy of a space cell S. $\mathcal{H}(S)$ is called a *strict hierarchy* of S if any subset $\mathcal{A} \subseteq \mathcal{H}(S)$ with $S \notin \mathcal{A}$ is also a hierarchy of a space cell $X \in \mathcal{A}$. Then, $\mathcal{A}(X)$ is called a *sub-hierarchy* of S.

Let $\mathcal{K} = \{L_1, L_2, \ldots, L_k\}$ be a set of space layers with $L_1 \geq L_2 \geq \cdots \geq L_k$. Then any hierarchy $\mathcal{H}(S)$ of a space cell $S \in \mathcal{C}(L_1)$ necessarily is a strict hierarchy.

The hierarchical structure of space cells also has an obvious impact on the joint state space of the underlying space layer complex $\mathcal{L}$. Precisely, given a hierarchy $\mathcal{H}(S)$ a space cell S, it follows that the dual node $v_{TP}(S)$ of S participates in any joint state $JS \in \mathcal{JS}(\mathcal{L})$ that contains the dual node $v_{TP}(X)$ of a space cell $X \in \mathcal{H}(S)$. In case of $\mathcal{H}(S)$ being a strict hierarchy the same also holds for any sub-hierarchy $\mathcal{A}(X) \subseteq \mathcal{H}(S)$.

The arrangement of space layers in a hierarchy is defined in the same way as for space cells but using a poset of space layers instead.

Definition 3.56 (Greatest space layer, hierarchy of a space layer). Let $\mathcal{L}$ be a space layer complex and let $Q = (\mathcal{L}, \leq)$ be the poset of the space layers in $\mathcal{L}$. Then a space layer L contained in a subset $\mathcal{K} \subseteq Q$ is said to be the *greatest space layer* in $\mathcal{K}$ if any space layer in $\mathcal{K}$ is transitively included in L, and thus $\forall L_i \in \mathcal{K}: L_i \leq L$. The subset $\mathcal{K}$ is said to be a *hierarchy* of the space layer L.

Definition 3.57 (Strict hierarchy of space layers, sub-hierarchy). Let $\mathcal{K}(L)$ be a hierarchy of a space layer L. $\mathcal{K}(L)$ is called a *strict hierarchy* of L if any subset $\mathcal{A} \subseteq K(L)$ with $L \notin \mathcal{A}$ is also a hierarchy of a space layer $X \in \mathcal{A}$. Then, $\mathcal{A}(X)$ is called a *sub-hierarchy* of L.

Whereas the inclusion relation between space cells in a space layer complex $\mathcal{L}$ has been defined based on their spatial containment in primal geometry space so far, it also induces characteristic graph structures in the dual space representation of $\mathcal{L}$. A space cell S_1 can be identified to be a subspace cell of S_2 in the multilayered graph $MLG(\mathcal{L})$ of $\mathcal{L}$, if the dual node of S_1 is linked by exactly one inter-layer edge to the dual node of S_2 but to no other dual node on the space layer containing S_2. The following proposition formally states this fact.

Proposition 3.58. Let $L_1, L_2 \in \mathcal{L}$ be two space layers participating in a space layer complex $\mathcal{L}$, with $S_1 \in \mathcal{C}(L_1)$ and $S_2 \in \mathcal{C}(L_2)$ being two space cells from either space layer. Then $S_1 \leq S_2$ implies that there exists an inter-layer edge $e = \{v_{TP}(S_1), v_{TP}(S_2)\} \in E(MLG(\mathcal{L}))$ but no further inter-layer edges linking $v_{TP}(S_1)$ to the dual node of any other space cell on L_2, and thus $\{\{v_{TP}(S_1), v_{TP}(S_\propto)\} \mid S_\propto \in \{\mathcal{C}(L_2), S_{out}(L_2)\} \setminus S_2\} \notin E(MLG(\mathcal{L}))$.

This proposition immediately follows from the fact that $GM(S_1) \subseteq GM(S_2)$ in case $S_1 \leq S_2$ per definition 3.50. The topological relationship between S_1 and S_2 according to the 4-intersection matrix $\mathfrak{I}_4(S_1, S_2)$ (cf. formula 3.38) must therefore be classifiable as either *inside*, *coveredBy*, or *equal*. Nevertheless, the above proposition also demonstrates that reasoning about subspace relationships between both space cells and entire space layers can be performed in a purely combinatorial and computationally efficient way as soon as the multilayered graph has been established for a given space layer complex. Based on the single inter-layer edge between a superspace cell and its subspace cell, the graph representation of a space cell hierarchy can also be deduced.

Proposition 3.59 (Star graph structure of a space cell hierarchy). Let $\mathcal{H}(A)$ be a hierarchy of a space cell A contained in a space layer complex $\mathcal{L}$. Then a subgraph $T_k \subseteq MLG(\mathcal{L})$ containing the dual nodes of all space cells in $\mathcal{H}(A)$ as well as all inter-layer edges linking $v_{TP}(A)$ to the dual nodes of all its lower space cells renders a *star structure*. Formally, the star T_k induced by $\mathcal{H}(A)$, with $k = |\mathcal{H}(A)| - 1$, is given by

(i) $V(T_k) = \{v_{TP}(S_\alpha)\}_{\alpha \in I_{\mathcal{H}(A)}}$, with $|V(T_k)| = k + 1$, and

(ii) $E(T_k) = \{\{v_{TP}(A), v_{TP}(S_i)\} \mid S_i \in \mathcal{H}(A) \wedge S_i \neq A\}$, with $|E(T_k)| = k$.

A star T_k is a special case of a *tree structure* and hence of a connected acyclic graph. It can be equivalently classified as complete bipartite graph with the size of one of its two node partitions being one. In case $k > 1$, the dual node $v_{TP}(A)$ of the greatest space cell A in $\mathcal{H}(A)$ is said to be the internal node of the star $T_k(\mathcal{H}(A))$ with k additional leaf nodes, and it is the only node having a degree greater than one. In case $k \leq 1$, $T_k(\mathcal{H}(A))$ is said to have no internal node but rather $k + 1$ leaves.

However, not every star T_k contained in $MLG(\mathcal{L})$ means that the space cells whose dual nodes constitute the node set $V(T_k)$ automatically render a space cell hierarchy for the space cell A when $v_{TP}(A)$ is assumed to be the internal node of T_k. Thus, without knowing a priori whether or not a star $T_k \subseteq MLG(\mathcal{L})$ encodes a space cell hierarchy $\mathcal{H}(A)$ of the space cell A, a purely combinatorially check involves two more conditions. First, the edge set $E(T_k)$ may only contain inter-layer edges so that $E(T_k) \subseteq E(ILG(\mathcal{L})) \subseteq E(MLG(\mathcal{L}))$. Second, let L_A be the space layer containing the space cell A whose dual node is the internal node of T_k, with $A \in \mathcal{C}(L_A)$ and $L_A \in \mathcal{L}$. Then none of the leaf nodes of T_k may be additionally linked to a dual node of another space cell on L_A, and hence

$\{\{v_{TP}(S_\alpha), v\} \mid S_\alpha \in \{\mathcal{C}(L_A), S_{out}(L_A)\} \setminus A \wedge v \in V(T_k)\} \notin E(MLG(\mathcal{L}))$. Obviously, the second condition follows from proposition 3.58. In case $k = 1$, this second condition has to be met by at least one of the two leaf nodes.

The illustrated combinatorial check also allows for identifying subspace layers. Let $L_1, L_2 \in \mathcal{L}$ be two space layers of a space layer complex $\mathcal{L}$, with $L_1 \geq L_2$. Then there exists a space cell hierarchy $\mathcal{H}(S)$ for every space cell $S \in \mathcal{C}(L_1)$ which contains a (possibly empty) subset of space cells in $\mathcal{C}(L_2)$. Correspondingly, the inter-layer graph between L_1 and L_2 contains n stars where n equals the number of space cells in $\mathcal{C}(L_1)$.

Two disjoint space cell hierarchies on the space layer complex $\mathcal{L}$ are said to be *adjacent* if their stars are linked by at least one intra-layer edge in $MLG(\mathcal{L})$. They are additionally said to be *connected* if there exists at least one simple path between both stars in the intra-layer graph on at least one of the involved space layer $L \in \mathcal{L}$. Spatially, this path describes a *passage* between both hierarchies.

Definition 3.60 (Adjacent and connected space cell hierarchies, connector, passage). Let $\mathcal{L}$ be a space layer complex and $\mathcal{A}(X)$, $\mathcal{B}(Y)$ be two space cell hierarchies of the space cell X respectively Y with $\mathcal{A}(X) \cap \mathcal{B}(Y) = \emptyset$. Further, let A and B be two space cells from either hierarchy on the same layer $L \in \mathcal{L}$, with $A \in \mathcal{A}(X)$, $B \in \mathcal{B}(Y)$, and $A, B \in \mathcal{C}(L)$. If there exists a simple path $P \subseteq G_{TP}(L)$ of the intra-layer graph $G_{TP}(L)$ of L so that $v_{TP}(A), v_{TP}(B) \in V(P)$ and $\forall S_i \in \mathcal{A}(X) \cup \mathcal{B}(Y) \setminus \{A, B\}$: $v_{TP}(S_i) \notin V(P)$ as well as $P \cong \bar{\mathbb{B}}^1$ then

(i) $\mathcal{A}(X)$ and $\mathcal{B}(Y)$ are said to be *adjacent* on L,
(ii) $\mathcal{A}(X)$ and $\mathcal{B}(Y)$ are said to be *connected* on L if $|V(P)| > 2$,
(iii) a space cell S is said to be a *connector* between $\mathcal{A}(X)$ and $\mathcal{B}(Y)$ on L if $|V(P)| = 3$ and $v_{TP}(S) \in V(P)$,
(iv) P is said to be a *passage* between $\mathcal{A}(X)$ and $\mathcal{B}(Y)$.

In the following, the modelling of subspace layers and multi-granular space decompositions is exemplified for different scenarios.

Example 3.61. The first example demonstrates the modelling of spatial containment hierarchies between topographic space layers in order to reflect the nested hierarchical structure of a building or site. Consider the 3-dimensional setting as shown in the following figure 100. It consists of three space layers representing the built environment of a two floored building on three different levels of details.

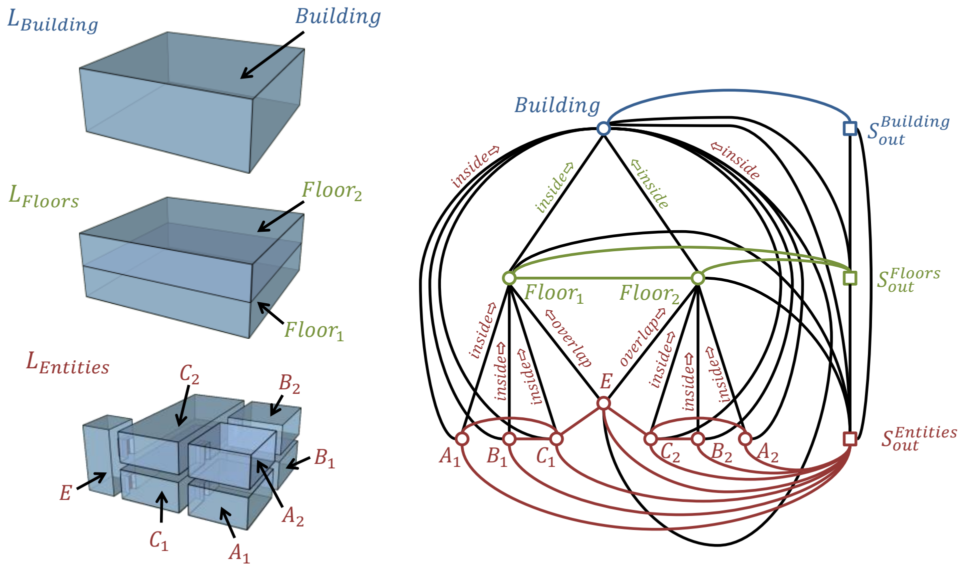


Figure 100: Example of subspace layers capturing the nested hierarchical structure of topographic space.

The first and coarsest level of detail is given by the layer $L_{Building}$ which contains a single space cell $Building$ representing the entire space occupied by the building. In this example, the geometry of the $Building$ space cell reflects the minimum bounding volume of the exterior shell of the building. The second layer L_{Floors} subspaces the building into the space cells $Floor_1$ and $Floor_2$ that represent its two floors and whose primal space geometries are subsets of the $Building$ geometry. The finest topographic space representation is captured by the third layer $L_{Entities}$ which models the architectural entities contained in the building. In order to keep the example simple, the first floor is supposed to only contain two rooms being accessible from a corridor, and this configuration is mapped by the three space cells A_1, B_1, and C_1. The similar setting is given for the second floor and reflected by the space cells A_2, B_2, and C_2. Moreover, the space cell E represents an elevator shaft which connects the corridors C_1 and C_2 on both floors. Note that whereas A_1, B_1, and C_1 are spatially contained in $Floor_1$ which equivalently holds for A_2, B_2, C_2, and $Floor_2$, the primal space geometry of E necessarily overlaps with $Floor_1$ and $Floor_2$ but, of course, is contained inside $Building$. The resulting multilayered graph encoding the hierarchical structure of the building is shown on the right of figure 100.

The multilayered graph can be queried for the inclusion relations between space cells and thus allows for reasoning about the building hierarchy. For example, it easily follows from proposition 3.58 that A_1 is a subspace cell of $Floor_1$ since the dual node of A_1 is only adjacent to that of $Floor_1$ on L_{Floors}, and therefore $A_1 \leq Floor_1$. For the same reason it can be deduced that $B_1 \leq Floor_1$ and $C_1 \leq Floor_1$ as well as $Floor_1 \leq Building$. Due to the transitivity of the inclusion relation (cf. condition (iii) of proposition 3.52) we immediately get $A_1 \leq Floor_1 \wedge Floor_1 \leq Building \Rightarrow A_1 \leq Building$. Similar inclusion relations can be observed between A_2, B_2, C_2 and $Floor_2$ as well as $Building$. However, the situation of the elevator cell E is different as it is linked to both $Floor_1$ and $Floor_2$ in the multilayered graph which violates proposition 3.58 and automatically yields $E \nleq Floor_1 \wedge E \nleq Floor_2$. Only when compared to $Building$ it follows that $E \leq Building$.

The ordering of space cells implies that the poset $\mathcal{H} = \{Building, Floor_1, Floor_2, A_1, B_1, C_1, A_2, B_2, C_2, E\}$ is a hierarchy of $Building$ which, of course, conforms to the intuitive spatial understanding of the indoor setting. As of proposition 3.59, $\mathcal{H}(Building)$ renders a T_9 star being a subgraph of the multilayered graph. This star structure is illustrated separately in figure 101.

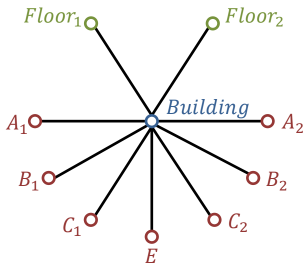


Figure 101: The T_9 star of the hierarchy $\mathcal{H}(Building)$ from figure 100.

The hierarchy $\mathcal{H}(Building)$ is non-strict since the subset $\mathcal{A} \subset \mathcal{H}$ with $\mathcal{A} = \{Floor_1, A_1, B_1, C_1, E\}$ is not a hierarchy of $Floor_1$ due to $E \nleq Floor_1$. However, if we consider the subset $\mathcal{B} = \mathcal{H} \backslash \{E\}$ then $\mathcal{B}$ is a strict hierarchy of $Building$. Moreover, the two subsets $\mathcal{F}_1 = \{Floor_1, A_1, B_1, C_1\}$ and $\mathcal{F}_2 = \{Floor_2, A_2, B_2, C_2\}$ of $\mathcal{B}$ are hierarchies of $Floor_1$ respectively $Floor_2$. Due to $\{v_{TP}(Floor_1), v_{TP}(Floor_2)\} \in G_{TP}(L_{Floors})$, the hierarchies $\mathcal{F}_1(Floor_1)$ and $\mathcal{F}_2(Floor_2)$ are *adjacent* on L_{Floors}. They are even *connected* on $L_{Entities}$ since E is linked to both C_1 and C_2 in the intra-layer graph of $L_{Entities}$. Correspondingly, E is a *connector* for both hierarchies and the path from C_1 to C_2 via E describes a *passage* between them on $L_{Entities}$ (cf. definition 3.60).

The inclusion relations between space cells also induce a partial order on the set of space layers. First, due to $\mathcal{B} = \{Building, Floor_1, Floor_2\}$ being a hierarchy of $Building$ that contains all space cells on $L_{Building}$ and L_{Floors} it follows that L_{Floors} is a subspace layer of $L_{Building}$ with $L_{Floors} \leq L_{Building}$. In a similar way it can be argued that $L_{Entities} \leq L_{Building}$. But the condition for subspace layers postulated in definition 3.51 is not fulfilled between $L_{Entities}$ and L_{Floors} since $E \nleq Floor_1 \wedge E \nleq Floor_2$ and thus $L_{Entities} \nleq L_{Floors}$. This might be contrary to a first spatial intuition because the union of the primal space geometries of all space cells on $L_{Entities}$ is obviously a subset of $GM(Floor_1) \cup GM(Floor_2)$ in $\mathbb{R}^3$. However, and as stated above, the fact that $GM(\mathcal{C}(L_{Entities})) \subseteq GM(\mathcal{C}(L_{Floors}))$ is only a necessary but not a sufficient condition for an inclusion relation between $L_{Entities}$ and L_{Floors}. Finally, the spatial ordering of the space layers lets us deduce that $\mathcal{K} = \{L_{Building}, L_{Floors}, L_{Entities}\}$ is a non-strict hierarchy of $L_{Building}$. Obviously, $\mathcal{K}(L_{Building})$ cannot be made strict by omitting the space cell E as done for the space cell hierarchy $\mathcal{H}(Building)$ simply because $E \notin \mathcal{K}(L_{Building})$.

It is important to note that the illustrated reasoning about the hierarchical structure of the building can be performed purely based on the simple information how the dual nodes of the space cells are linked by inter-layer edges in the multilayered graph. This spatial knowledge enables multi-level path planning. For example, assume a navigation user wants to be routed from room A_1 to room A_2. This path query can be fully answered and presented with fine-grained routing commands by only considering the intra-layer graph on $L_{Entities}$. However, the containment relationships between space cells additionally support the generation of intermediate and coarser routing instructions which reflect the hierarchical organization of the indoor space and thus especially assist human wayfinding (e.g., Richter et al. 2009, Stoffel et al. 2009). For instance, due to $A_1 \leq Floor_1$ and $A_2 \leq Floor_2$, a first routing instruction could simply state that *"A_1 is on the first floor. The target room A_2 is located on the second floor"*. Moreover, since A_1 and A_2 belong to disjoint space cell hierarchies of $Floor_1$ and $Floor_2$ connected by E, the navigation system may presented a subsequent instruction like *"Take the elevator to go to the second floor"*. The hierarchical information also allows for providing partial visualizations of the planned route. In the above example, the way through either floor or, more generally, the way through each space cell hierarchy along the path may be illustrated separately. Finally, suppose the navigation user instead asks for a path between rooms A_1 and B_1, then according to $A_1 \leq Floor_1 \wedge B_1 \leq Floor_1$, this path must be entirely located on the first floor. A path search algorithm may thus discard any space cell X with $X \leq Floor_2$ when planning the path. It follows that space cell hierarchies are also feasible to narrow down the search space for path queries.

Additional hierarchical facts of the indoor space can be simply added to the stack of space layers. For example, if the building in figure 100 can also be structurally subdivided into wings then this vertical decomposition of the $Building$ space cell can be captured by space cells on a separate space layer. The MLSEM is however not restricted to structural hierarchies between topographic space layers but, in contrast, the space cells participating in a space cell hierarchy may belong to different notions of space. Assume for the above example that rooms A_1 and A_2 are spatially enclosed by another space cell L on a logical space layer. Then this spatial containment could express the membership of A_1 and A_2 in an *organizational unit* (e.g., departments in an office building) represented by the space cell hierarchy $\mathcal{H}(L) = \{L, A_1, A_2\}$. A navigation user could hence not only be presented with the information *"You are in room A_1"* but additionally with *"A_1 belongs to department L"*. Likewise, a user could choose the entire department L as target in a path query instead of a specific room. A path search algorithm then simply needs to pick a space cell from $\mathcal{H}(L)$ as suitable target, e.g. the nearest space cell in $\mathcal{H}(L)$ with respect to the current position of the user. Alternatively, the space cell L might represent a hierarchy resulting from the *function* of the contained space cells (e.g., offices in a building, check-in counters at airports, etc.) or denoting associated *access constraints*. For example, suppose the space cell L reflects a security zone or a disaster area. If the navigation user is not granted access to L then the space cells contained in the space cell hierarchy $\mathcal{H}(L)$ need to be avoided by path search algorithms in order to generate a valid route.

From the approaches to indoor space modelling discussed in chapter 2.2, the most formally developed models for the representation of hierarchical structures in built indoor environments are proposed by (Lorenz et al. 2006), (Stoffel et al. 2007), and (Stoffel et al. 2009). In all approaches, the hierarchy is similarly built from the spatial containment of 2-dimensional regions and is mapped onto a tree graph structure. The regions on coarser levels of the hierarchy hereby result from the spatial union of neighboured regions on the next more fine-grained level. Only the finest resolution of the space representation is explicitly represented in these models whereas spatial aggregates on coarser levels are only derived implicitly and thus cannot possess arbitrary shapes. The resulting tree contains a single node per region with one root node representing the entire indoor environment. An edge between two nodes from subsequent hierarchical levels denotes a *containment* relationship. Edges between nodes on the same level are also allowed and reflect the *connectedness* of both regions. It follows that the MLSEM fully covers the expressivity of these models. A slight difference, however, results from the fact that the rooted trees applied in these models only provide containment edges between subsequent levels of the hierarchy. In contrast, the dual nodes of the space cells participating in a space cell hierarchy are linked by inter-layer edges over all space layers involved in the hierarchy (cf. figure 100). Thus, the resulting multilayered graph contains a rooted tree as subgraph but itself can be classified as lattice. Also note that using a rooted tree prohibits modelling of what has been defined as non-strict space cell hierarchies in this chapter. This means that the hierarchical structure encoded by the multilayered graph in the above example cannot be expressed with any of the alternative approaches. Moreover, all models have in common that they are restricted to structural hierarchies in topographic space.

The dual-graph-based approaches proposed by (Lee 2001), (Jensen et al. 2009), and (Boguslawski & Gold 2011) lack the means for expressing the hierarchical organization of the built environment. Only the Node-Relation Structure (NRS) of (Lee 2001) supports a hierarchical relationship between the nodes of the dual graph (combinatorial network model, CNM) and its spatial embedding (geometric network model, GNM). This hierarchy results from the fact that complex architectural entities such as long or concave-shaped corridors are mapped by straight medial axis transforms in the GNM whereas they are represented by a single dual node in the CNM. Thus, the dual node in the CNM (called *master node*) is hierarchically related to a set of graph elements in the GNM. However, the two graph representations in dual space are not underpinned by a corresponding subspacing of the architectural entity in primal space and therefore do not encode a spatial containment hierarchy between different cells.

Example 3.62. The decomposition of complex topographic regions into more fine-grained parts in order to map the region onto several elements in the navigation graph instead of just one single node is not only applied by (Lee 2001) but in fact (implicitly or explicitly) by most cell-based approaches to indoor space modelling (cf. chapter 2.2.2). The resulting graph structure typically follows the layout of the region more naturally and hence is better suited to provide precise geometric trajectories through indoor space, e.g. for humans and autonomous robots. The following examples illustrate how the subspacing model of the MLSEM is used for this purpose and demonstrate that it provides a formal framework which can be used to explain the results from other works.

Figure 102 shows a corridor with three adjacent rooms on either side in a 2-dimensional setting. Every room as well as the corridor itself are modelled as separate space cells on a topographic space layer which yields the weak Poincaré dual graph (cf. definition 3.33) depicted on the right. The dual edges to the dual node of the outer space are omitted in the following for readability.[19]

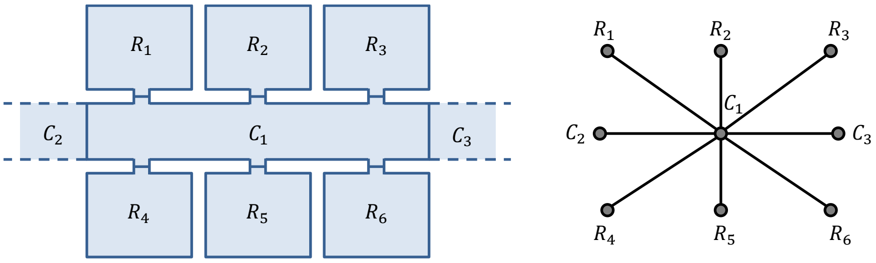


Figure 102: Example of a 2-dimensional corridor with adjacent rooms (left) and the resulting weak dual graph (right).

The intra-layer graph shown above captures the building topology correctly and is suitable for path planning purposes without any further modification. Since it is a purely topological graph structure, it may have an additional isomorphic spatial embedding (cf. quadrants III and IV of figure 42). The following figure 103a illustrates a possible embedding in which each space curve associated with a dual edge is chosen so that it entirely lies in navigable space.

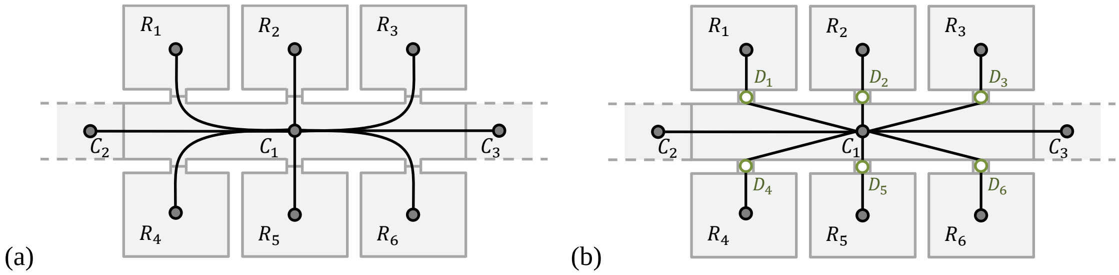


Figure 103: Possible Euclidean space embedding of the dual graph from figure 102 (a) and the resulting weak dual graph when representing doors as individual space cells (b).

[19] Note that if a minimal CW decomposition of the space cell configuration in primal topology space is assumed, then it requires a minimum of eight 1-cells to describe the disconnected boundary parts of the corridor between the rooms, and thus the dual node of the corridor exhibits eight dual edges to the outer node. In three dimensions, the boundary of the corridor could instead be represented by a single 2-cell yielding a single dual edge to the dual node of the outer space (cf. related discussion in chapter 3.1.3).

Remember that a space curve needs not be a straight line segment but only a homeomorphic image of $\overline{\mathbb{B}}^1$ (cf. condition (i) of definition 3.21). The space curves sketched in figure 103 thus already describe geometric space trajectories that could be followed by autonomous robots without the need for subspacing the corridor. A similar result can be obtained when the door spaces are modelled as space cells themselves and hence are represented as individual dual nodes as shown in figure 103b. In this case, straight line segments between the nodes may already suffice to ensure an embedding in navigable space.

In literature, the most frequent argument against mapping large places onto just one node within the navigation graph is that this representation leads to circuitous geometric trajectories (e.g., Stoffel et al. 2007, Yuan & Schneider 2010a, Liu & Zlatanova 2011b, Goetz & Zipf 2011). In fact, following the space curves shown in figure 103a and b from room R_1 to room R_4 or, equivalently, from the adjacent corridor C_2 to R_4 would involve passing the spatial embedding of the dual node of C_1 and thus render a detour. This argument however mostly holds for mobile robots that move along a predefined geometric path. For human navigation, routing instructions such as *"Go to the opposite room"* or *"Take the first door on the right of corridor C_1"* more efficiently support the wayfinding to room R_4 than a precise geometric trajectory in most scenarios since humans typically move freely inside the built environment and are not bound to follow a given space curve. (Lorenz & Ohlbach 2006) and (Lorenz et al. 2006) have shown how to generate such guidance for human wayfinding based on the quantitative information provided by their indoor space model (e.g., the angular distribution of doors around the centre of the corridor, cf. chapter 2.2.2). This information is also available from the primal space geometries of space cells and their spatial configuration in a space cell complex. A second argument against circuitous spatial embeddings is that they yield non-optimal routes and hence impact the derivation of shortest paths between two places. A possible decomposition of the corridor as employed by most approaches in order to derive a place graph that avoids detours is depicted in the following figure. The corridor is split along three subspace cells placed in front of the doors in this example. The MLSEM does however not presuppose a specific choice of subspacing.

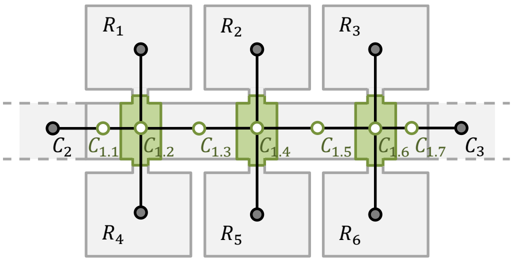
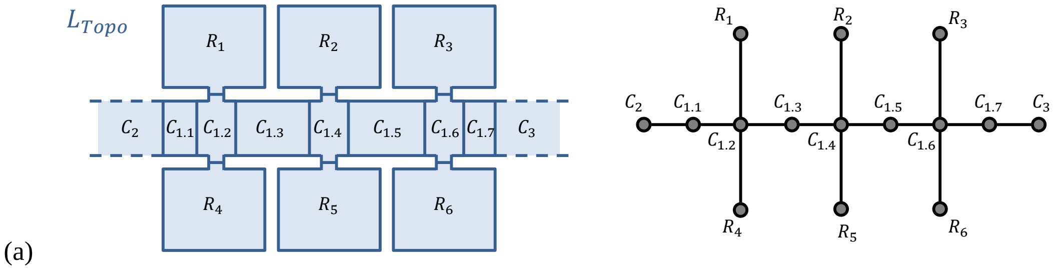


Figure 104: Decomposition of the corridor from figure 102 into seven subspace cells $C_{1.1}$ to $C_{1.7}$.

The following figure 105 shows three alternatives for capturing the above space cell configuration on separate space layers of the MLSEM. In the first alternative, a single topographic space layer is used to reflect the subspacing of the corridor. The resulting weak dual graph shown on the right of figure 105a agrees with most proposals for mapping large corridors in literature (e.g., Gilliéron & Merminod 2003, Meijers et al. 2005, Lorenz & Ohlbach 2006, Stoffel et al. 2007, Goetz & Zipf 2011, Boguslawski & Gold 2011). However, the information that the space cells $C_{1.1}$ to $C_{1.7}$ form a single corridor is lost in this space representation. A routing instruction like *"Take the first door on the left of corridor C_1"* cannot be easily generated any more since the dual edges representing the doors are no longer linked to a single dual node of the corridor (cf. Lorenz & Ohlbach 2006).



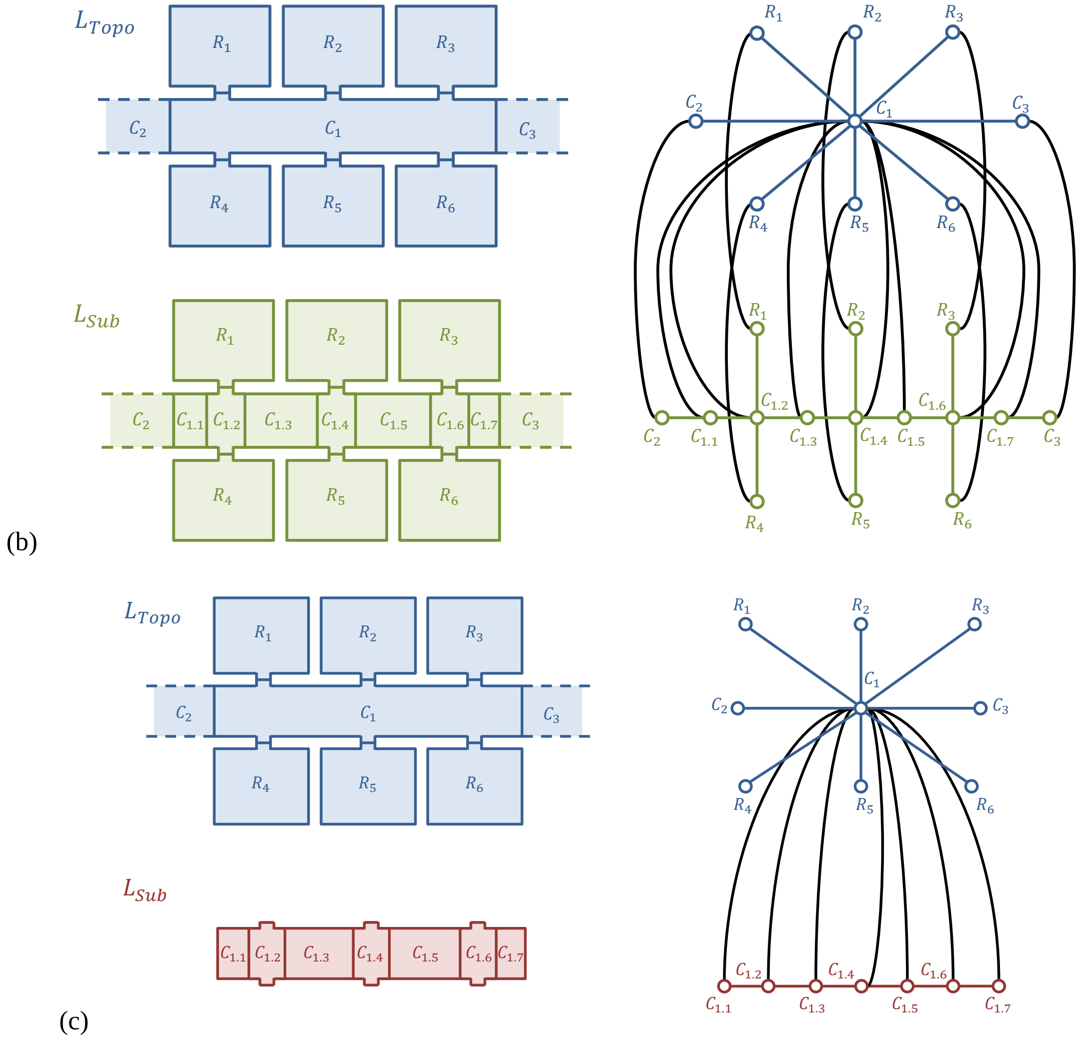


Figure 105: Three possible alternatives for representing subspace cells on different space layers (left) and their corresponding multilayered graph structures (right).

The second alternative presented in figure 105b consists of two separate topographic space layers L_{Topo} and L_{Sub}. Whereas the former layer carries each architectural entity to a single topographic space cell (cf. figure 102), the latter provides the subspacing of the corridor and additionally contains copies of all non-subspaced space cells from L_{Topo}. In contrast to most other approaches to indoor space modelling, the MLSEM is hence capable of providing both the real world shape of an architectural entity and its subdivision into smaller parts within an integrated model. Obviously, $L_{Topo} \geq L_{Sub}$ and $GM(\mathcal{C}(L_{Topo})) = GM(\mathcal{C}(L_{Sub}))$. Moreover, the poset $\mathcal{H} = \{C_1, C_{1.1}, C_{1.2}, \dots, C_{1.7}\}$ is a hierarchy of the corridor space cell C_1 on L_{Topo} which keeps the information that the space cells $C_{1.1}$ to $C_{1.7}$ make up a single corridor. The multilayered graph shown on the right of figure 105b thus facilitates the generation of routing instructions as proposed by (Lorenz & Ohlbach 2006) as well as the derivation of precise space trajectories and the computation of shortest paths without detours. Note that the topological relationship between the space cells on L_{Topo} and their non-subspaced counterparts on L_{Sub} necessarily is *equals*.

The dual node of C_1 in figure 105b conforms to the conceptual idea of a master node in the NRS model of (Lee 2001) since it corresponds to a subgraph on L_{Sub} constituted by the dual nodes of the subspace cells $C_{1.1}$ to $C_{1.7}$ and their intra-layer edges. However, (Lee 2001) introduces this hierarchical relationship between the dual topological graph and its spatial embedding for one and the same primal space representation, whereas in the MLSEM the hierarchy is established between alternative primal space representations and their topological graphs in dual space. The NRS model of (Lee 2001) can thus be explained 1) by using the primal space representation of L_{Topo}, 2) by representing the CNM through the intra-layer graph $G_{TP}(L_{Topo})$ on L_{Topo}, and 3) by choosing the spatial embedding $G_{GM}(L_{Sub})$ of the intra-layer graph on L_{Sub} for representing the GNM. The primal space representation of L_{Sub} then provides the subspacing of primal cells which is not explicitly given in the NRS.

The third alternative configuration of space layers illustrated in figure 105c differs from the second one in that it only contains the subspace cells of the corridor but does not redundantly repeat non-subspaced space cells from L_{Topo}, and thus $L_{Topo} \geq L_{Sub}$ but $GM(\mathcal{C}(L_{Topo})) \supset GM(\mathcal{C}(L_{Sub}))$. The hierarchy $\mathcal{H}(C_1)$ is still observable in the resulting multilayered graph which however does not explicitly encode the relation between the rooms and the corridor parts any more. A more detailed discussion of this space representation as well as its strengths and weaknesses follows in chapter 3.5.

An example for subspacing a concave-shaped topographic region is given in figure 106. For the same reasons as stated above, a mapping of the U-shaped corridor onto a single space cell (cf. figure 106a) is already feasible for path planning and guidance of humans but is rather insufficient for precise space trajectories and shortest path computations. A graph representation that more naturally follows the layout of the corridor again requires subspacing. In figure 106b, an example decomposition of the corridor is shown. It conforms to the results of the 2-dimensional subspacing algorithm introduced by (Stoffel et al. 2007) which utilizes visibility criteria in order to partition concave-shaped regions (cf. chapter 2.2.2). For the mapping of the indoor scene and the decomposition of the corridor onto separate space layers of the MLSEM, the same considerations as discussed above apply.

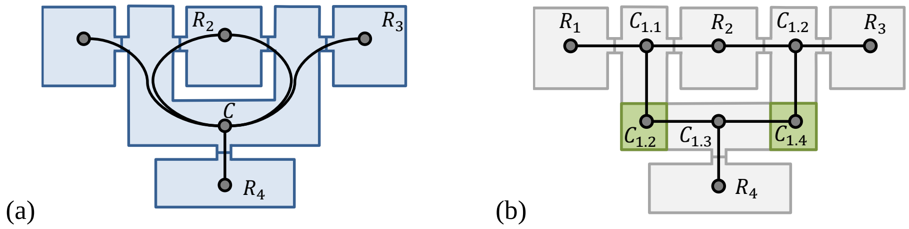


Figure 106: U-shaped corridor (a) and a possible subspacing (b) with overlaid multilayered graphs.

Example 3.63. Another reason for subspacing topographic regions and architectural entities is to explicitly tag navigable and non-navigable sections therein. For example, non-navigable sections may reflect fixed obstacles inside rooms or corridors, or may be related to the mode of locomotion of the navigation user. Although most conceptual-based approaches agree in the need for explicitly representing obstacles, cell-based approaches are commonly limited to the representation of the free indoor space (cf. chapter 2.2.2). In contrast, regular grid-based approaches typically indicate whether or not grid cells are navigable (e.g., Bandi & Thalmann 1998, Yuan & Schneider 2010b, Afyouni et al. 2010).

A simple 3-dimensional example for modelling obstacles within the MLSEM is presented in the following figure. It shows a sequence of three rooms one of which contains a column that spans the entire height of the room. Suppose that each room as well as the space occupied by the column is mapped onto a separate space cell on a topographic space layer.[20] The corresponding weak dual graph for a minimal CW decomposition of this setting is shown on the right.

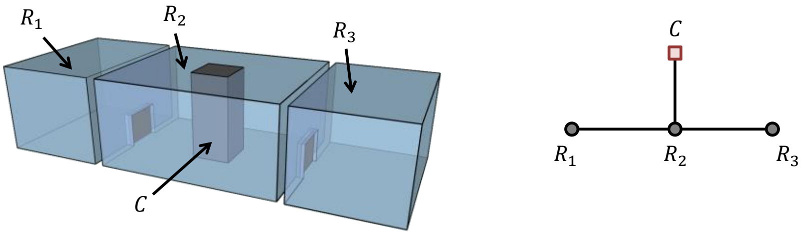


Figure 107: Three rooms with a column (left) and the corresponding weak dual graph (right). The dual node of the column C is depicted as square node in order to indicate that it is non-navigable.

The mathematical model of a space cell S as introduced in chapter 3.1.1.3 formally describes a partition of indoor space but does not assume its navigability. This information rather needs to be modelled as semantic attribute $a \in$

[20] Cutting the column from the room cell leaves a space being homeomorphic to a toroid. The CW decomposition of the room cell therefore involves an additional bridge face (cf. proposition 3.35) which however is not mapped onto a dual edge in the Poincaré dual graph $G_{TP}(L)$ of the space layer L. The column itself is simply mapped by a space cell being homeomorphic to $\bar{\mathbb{B}}^3$ which fills the hole of the toroid.

$A(S)$. According to definition 3.6, an attribute $a = (n, v)$ is given as name-value pair. Assume an attribute whose name n is "$isNavigable$" and whose value v is Boolean so that $v \in \{true, false\}$. This attribute can be used to indicate the navigability of space cells and can be easily evaluated by path search algorithms. In the above example, this attribute is only set to $false$ for the space cell C representing the column. The dual node of C is depicted as square in the dual graph to visually illustrate this fact.

The dual space representation provides a valid view on the building topology. Since the column is contained inside room R_2, it only exhibits a single dual edge to R_2 but neither to R_1 nor R_3. Moreover, R_1, R_2, and R_3 are connected by dual edges indicating a navigable path from R_1 to R_3. However, one cannot tell that the column is surrounded by free space at either side based on this graph structure. A possible subspacing of room R_2 into four subspace cells as shown below enriches the model with this knowledge. The resulting dual graph captures the spatial setting more accurately and hence is feasible, for example, to provide precise space trajectories leading a navigation user around the column.

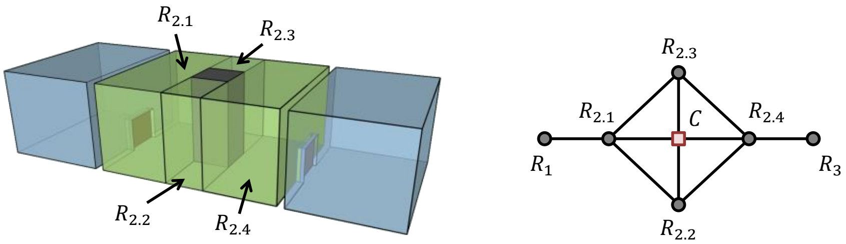


Figure 108: Possible subspacing of the room cell containing the column into four subspace cells.

A valid mapping of the indoor setting onto a topographic space layer and an additional subspace layer is illustrated in figure 109. Note that the hierarchy $\mathcal{H}(R_2)$ given by the poset $\mathcal{H}(R_2) = \{R_2, R_{2.1}, R_{2.2}, R_{2.3}, R_{2.4}\}$ does not contain the space cells C and C_{Sub} representing the column since R_2 and C are modelled as non-overlapping space cells on the same space layer. Consequently, C_{Sub} and R_2 are not linked by an inter-layer edge. In order to make $\mathcal{H}(R_2)$ contain C_{Sub} and hence to encode the spatial containment relationship between the room and the column in the multilayered graph, the column may only be modelled on the subspace layer, and thus the space cell C must be removed on the superspace layer. Also note that alternative configurations of space layers as presented in the previous example 3.62 are again possible.

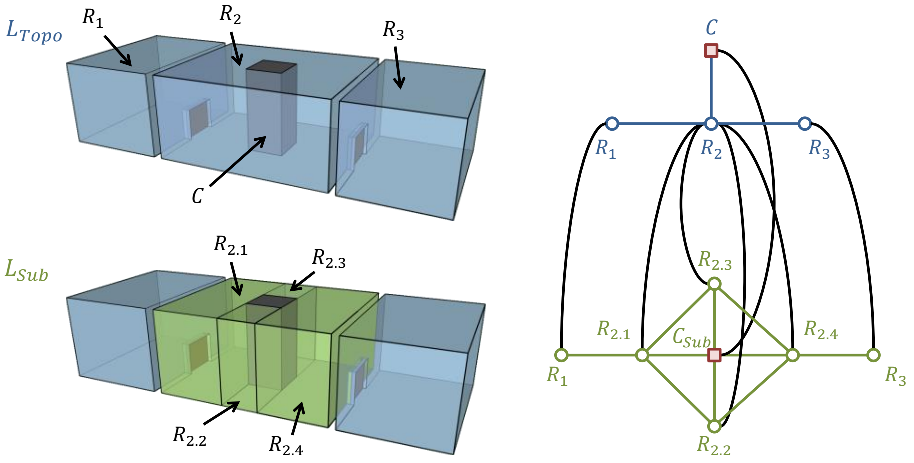


Figure 109: Modelling of the examples from figure 107 and figure 108 as topographic space layer and subspace layer (left) and their resulting multilayered graph (right) (the outer space cells are omitted for readability).

Figure 110 shows a slight modification of the above setting. Suppose a small step instead of a column inside room R_2. The step as well as each room is again mapped by a separate space cell. Applying a minimal CW decomposition in primal topology space yields the weak dual graph as shown on the right which is identical to that in figure 107.

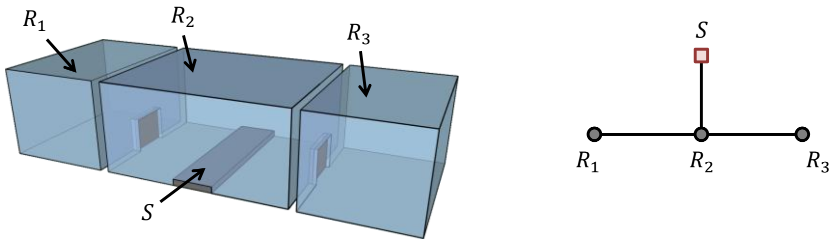


Figure 110: Three rooms with a step (left) and the corresponding weak dual graph (right).

The space occupied by the step itself is obviously non-navigable and cannot be entered by any navigation user. The dual node of the step is thus marked as being non-navigable in the above dual graph. Further, assume that pedestrians can easily pass the step but that it renders a non-passable obstacle for wheelchair users. Thus, wheelchair users cannot enter the free space above the step as well. In order to model this situation, the space above the step needs to be represented as a separate subspace cell and tagged as obstacle. The resulting configuration of space cells is shown below. The weak dual graph on the right does not provide a valid path from $R_{2.1}$ to $R_{2.3}$ any longer since the dual nodes of $R_{2.1}$ and $R_{2.3}$ are only linked via obstacle nodes. However, $R_{2.1}$ and $R_{2.3}$ themselves are still reachable from the neighboured rooms.

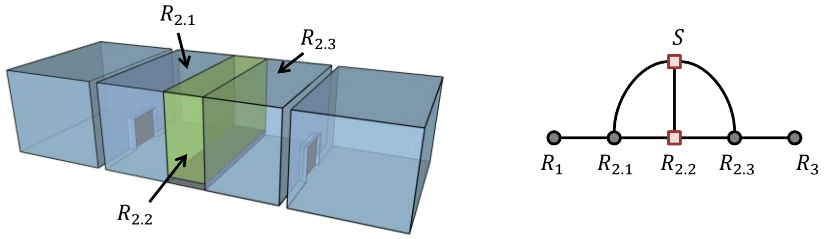


Figure 111: Possible subspacing of the room cell containing the step into three subspace cells.

Alternatively, both obstacle spaces may be entirely cut from the room cell R_2 and thus become a part of the outer topographic space. This modelling approach conforms with the examples in the previous publications on the MLSEM (cf. Nagel et al. 2010). In chapter 3.5, a binary algebraic operation on space layers realizing the subtraction of space cells is formally developed. The corresponding primal and dual space representations are depicted in figure 112. In the dual graph, a path between $R_{2.1}$ and $R_{2.2}$ now necessarily involves the dual node of the outer space. This, however, can be regarded an invalid path on a topographic space layer. Although the space representation requires less graph elements to encode the same spatial fact, the obstacles are only modelled implicitly and hence are not available, for example, for visual guidance purposes. Moreover, as will be shown below, obstacle spaces cannot always be cut from the navigable space without losing information.

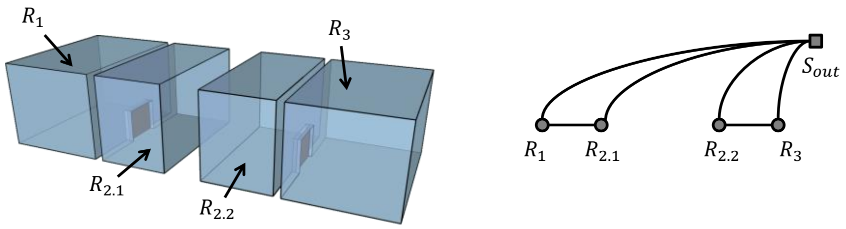


Figure 112: Cutting the step and the free space above it from the topographic space model.

Since the indoor space models for pedestrians and wheelchair users are contradictious, they are placed on two separate space layers $L_{Pedestrian}$ and $L_{WheelChair}$. In figure 113, this is illustrated for the models from figure 110 and figure 111. The space layer for wheelchair users is obviously a subspace layer, and thus $L_{WheelChair} \leq L_{Pedestrian}$. The space cell hierarchy $\mathcal{H}(R_2) = \{R_2, R_{2.1}, R_{2.2}, R_{2.3}\}$ again does not contain the space cells representing the step on either space layer for the same reasons as given in the previous example. However, the step cell could be easily removed from $L_{Pedestrian}$ as it does not render an obstacle on this layer, which would add $S_{WheelChair}$ to $\mathcal{H}(R_2)$.

In this example, the modelling of both space layers $L_{Pedestrian}$ and $L_{WheelChair}$ not only captures a multi-granular space representation but also provides different views on indoor space for different contexts of navigation. It follows that depending on the mode of locomotion of a specific navigation user, a navigation system must either choose $L_{Pedestrian}$ or $L_{WheelChair}$ for path planning and guidance purposes (cf. chapter 3.6). In order to provide a complete space model for wheelchair users, additional (sub)space cells on the layer $L_{WheelChair}$ may be marked non-navigable (e.g., space cells representing stairs or narrow passages).

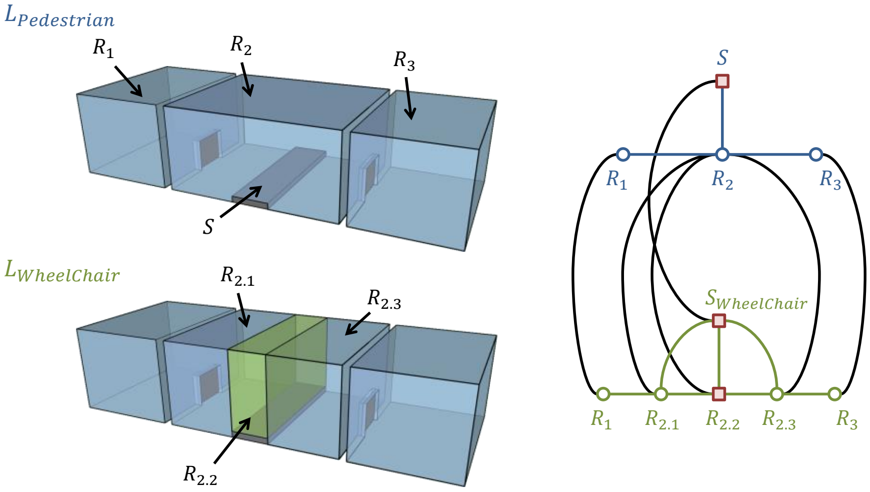


Figure 113: Capturing the examples from figure 110 and figure 111 as separate space layers for pedestrians and wheelchair users (left) and their resulting multilayered graph (right).

The following example shows a landing inside room R_2 in front of the door to room R_1 instead of a step. The weak dual graph for this setting once more agrees with that from figure 110 and figure 107.

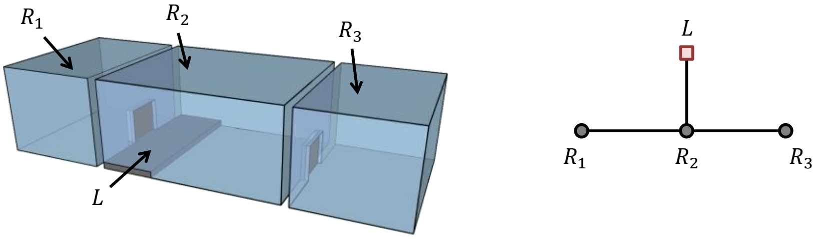


Figure 114: Three rooms with a landing (left) and the corresponding weak dual graph (right).

Assume the landing is too high to be entered by a wheelchair user. It therefore denotes an obstacle for wheelchair users when moving from R_3 to R_1. However, and in contrast to the example of figure 111, the landing may be entered from room R_1, and thus the free space above the landing is not only navigable for pedestrians. In order to model a space layer for wheelchair users, this free space again needs to be captured by a separate subspace cell. But instead of marking the entire space cell as non-navigable, only a part of its cell boundary must be tagged as such. The corresponding space layer is sketched in figure 115. As per definition 3.24, the cell boundary is available through boundary cells. Let B be the boundary cell representing the common boundary between $R_{2.1}$ and $R_{2.2}$ as shown below. Its primal space geometry is given as $GM(B) = GM(R_{2.1}) \cap GM(R_{2.2})$. The dual space topology $e_{TP}(B)$ is exactly the dual edge linking the dual nodes of $R_{2.1}$ and $R_{2.2}$. When associating B with the $isNavigable$ attribute and setting its value to $false$, then this information is accessible from $e_{TP}(B)$ for path search algorithms. The dual edge is drawn as dashed line to visually illustrate that it is non-navigable. The resulting dual graph now encodes the fact that both $R_{2.1}$ and $R_{2.2}$ are navigable spaces for wheelchair users but that there is no obstacle-free path joining them. In this example, cutting both the landing and the space above it from the original room cell R_2 is not an option since the information that the landing can be entered by wheelchair users would be lost.

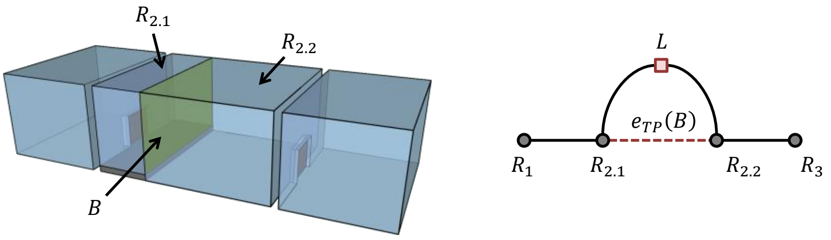


Figure 115: Possible subspacing of the room cell containing the landing into two subspace cells and marking the boundary cell B non-navigable for wheelchair users (depicted as dashed edge in the weak dual graph).

The example illustrated in figure 116 shows the wheelchair space layer when mounting a ramp into room R_2 which enables access to the landing. A possible subspacing of room R_2 into four subspace cells is shown on the left (R_2 is shown in a top view and the space cells for both rooms R_1 and R_3 are omitted for readability). In addition to the two obstacle spaces occupied by the landing and the ramp, also the boundary cell between the subspaces $R_{2.1}$ and $R_{2.3}$ needs to be marked as non-navigable like in the above example. The same holds for the boundary cell separating the subspaces $R_{2.2}$ and $R_{2.3}$ since a path between both would involve falling off the edge of the ramp. The final dual graph is shown on the right, and by adding the ramp and a corresponding subspace model it now provides a valid route for wheelchair users from R_1 to R_3 along the path $R_1 \rightarrow R_{2.1} \rightarrow R_{2.2} \rightarrow R_{2.4} \rightarrow R_3$ and vice versa. All other paths however involve one or more non-navigable dual nodes or non-navigable dual edges.

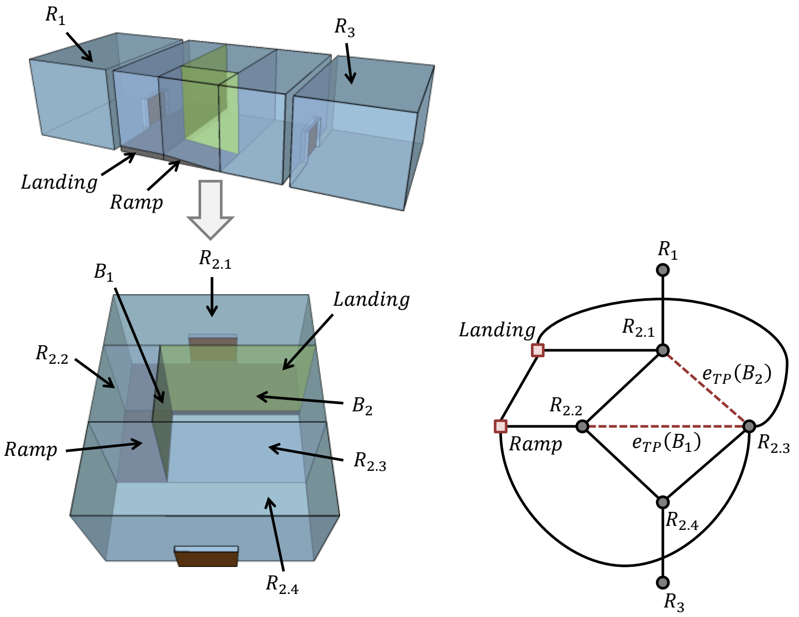


Figure 116: Adding a ramp to the example from figure 115 and corresponding subspacing. The boundary cells B_1 and B_2 are marked non-navigable for wheelchair users.

Finally, obstacles suspended from the ceiling are modelled in the same way as obstacles on the floor. Consider the setting shown in figure 117a. It yields the same weak dual graph as in the above examples of the column, the step, and the landing. In order to make the free spaces below the obstacle and at either side of the obstacle accessible in the dual graph, a subspacing as exemplified in figure 117b is required which results in the depicted weak dual graph. Suppose that an arbitrarily shaped navigation user wants to move from room R_1 to R_3. Whether the free space below the obstacle is passable for that user can be tested by evaluating the geometric shape of the user against the transition surface between the subspace cells $R_{2.1}$ and $R_{2.3}$. Let B be the corresponding boundary cell. Then the transition surface is given by its primal space geometry $GM(B) = GM(R_{2.1}) \cap GM(R_{2.3})$ and denotes the vertical profile of the passage. If the free space below the obstacle is not modelled as cuboidal space cell as in the example but also by an arbitrarily shaped geometric object, then obviously its entire primal space geometry $GM(R_{2.3})$ has to be evaluated instead. Similar geometric tests allow for verifying whether a navigation user can pass the suspended obstacle at either side. Sophisticated geometric algorithms facilitating path planning for arbitrarily shaped objects in a 3-dimensional obstacle space have been presented in (Yuan & Schneider 2010b) and (Yuan & Schneider 2011), and can be transferred in the context of the MLSEM.

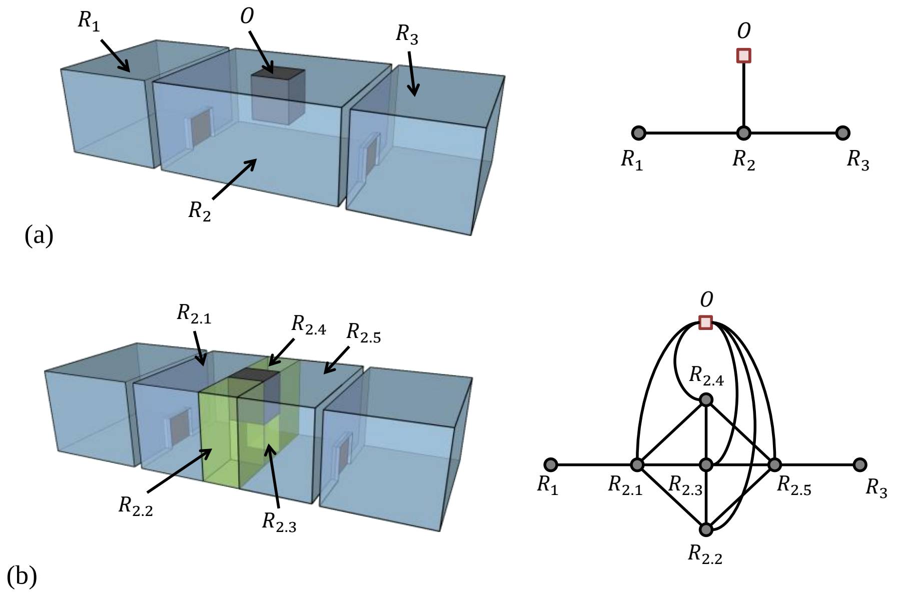


Figure 117: Three rooms with a suspended obstacle (a), possible subspacing of the room cell containing the obstacle into five subspace cells (b) and the corresponding weak dual graphs (right).

In summary, the exemplified scenarios in this chapter illustrate that the MLSEM provides a flexible support for multi-granular and hierarchical space representations which covers the capabilities of comparable grid-based and cell-based approaches. The conceptual separation of different views on indoor space by means of space layers facilitates to preserve the real world shape of an architectural entity while at the same time providing more generalized representations on superspace layers as well as more fine-grained decompositions on subspace layers. The number and type of subspace layers is unbounded and each subspace layer may follow its own partitioning schema. As demonstrated above, subspacing the topographic space is suitable for deriving fine-grained navigation graphs and space trajectories at different levels of detail as well as for denoting navigable and non-navigable subspaces, for example, due to navigation obstacles or the mode of locomotion. Since its mathematical formalization abstracts from a given notion of space, subspacing can also be applied to alternative space representations such as sensor or logical spaces (e.g., to denote a hierarchy of security zones).

The fine-grained representation of large rooms or corridors through complementary visibility graphs as proposed by a variety of research works (e.g., Stoffel et al. 2007, Yuan & Schneider 2010a, Liu & Zlatanova 2011a, Liu & Zlatanova 2011b, Goetz & Zipf 2011) cannot directly be mapped by the MLSEM since the multilayered graph is derived from topological relationships rather than from intervisibility criteria. However, mutual visibility is not feasible to express both connectedness and containment relationships between spatial regions. (Stoffel et al. 2007) and (Liu & Zlatanova 2011b) therefore recognize the need for building visibility graphs on top of an underlying space model providing connectivity information and hierarchical structures, which can be answered by the MLSEM.

3.5 Space Layer Algebra

In this chapter, a formal algebra for space layers facilitating the derivation of a new space layer from two input space layers is introduced. For this purpose, three operations on space layers are developed, namely a *merge operation*, a *difference operation*, and an *intersection operation*.

3.5.1 Merge Operation

In order to motivate the introduction of a *merge operation*, the discussion about two basic alternatives for the modelling of subspace layers as presented in the previous chapter 3.4 (cf. figure 105b and c) is resumed. The following 2-dimensional example recaps both alternatives. It shows a topographic space layer L_{super} containing three space cells that represent two rooms R_1 and R_2 being connected to a corridor C (cf. figure 43). The corridor cell is decomposed into three subspace cells Sub_1, Sub_2, and Sub_3 on a subspace layer L_{sub}. Whereas in the first modelling alternative presented in figure 118a the non-subspaced room cells are mirrored on the subspace layer, the second alternative in figure 118b differs in that the subspace layer only reflects the partitioning of the corridor.

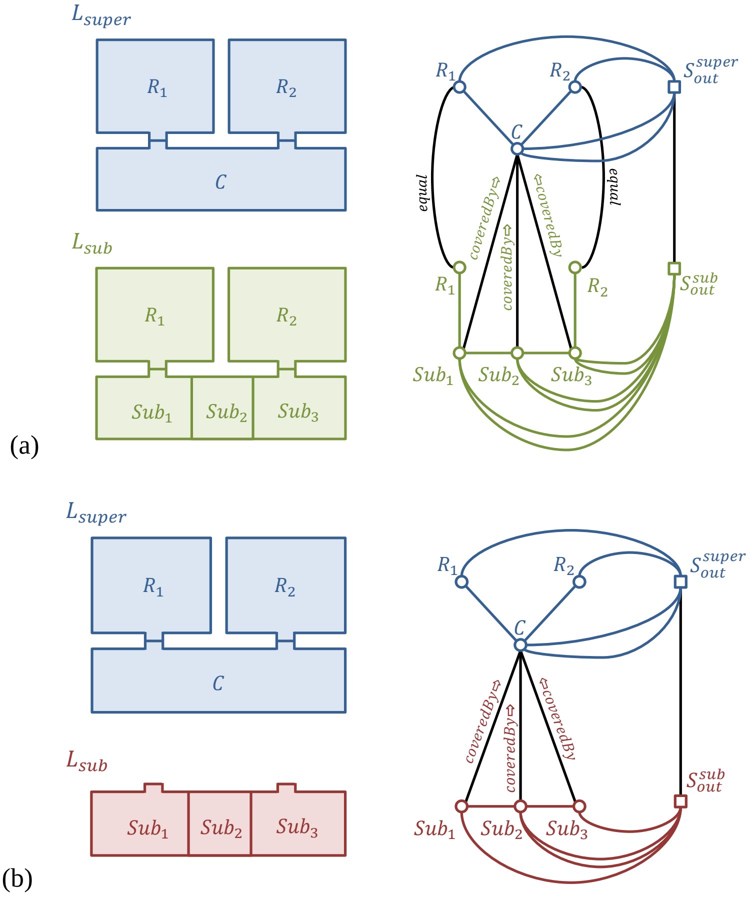


Figure 118: Two different possibilities to capture the subspacing of space cells on a subspace layer.

Both representations show similarities. Precisely, the inclusion relation $L_{super} \geq L_{sub}$ as well as the fact that the poset $\mathcal{H} = \{C, Sub_1, Sub_2, Sub_3\}$ forms a space cell hierarchy of the corridor C on L_{super} holds in either case. However, the multilayered graph resulting for the second alternative is less expressive. Although it encodes the information that the space cells on L_{sub} entirely occupy the space covered by C (since the dual node of C is not linked to S_{out}^{sub}), it cannot be queried whether a navigation user entering the corridor from room R_1 will be located in Sub_1, Sub_2, or Sub_3. The reason is that the inter-layer edges of the multilayered graph only capture overlap and containment relationships between space cells from different layers but not their topological adjacency. Answering this question for the second alternative requires the additional consideration of the joint state space of the space layer complex. For example, consider the four joint states given by $JS_1 \leftarrow (R_1, S_{out}^{sub})$, $JS_2 \leftarrow (C, Sub_1)$, $JS_3 \leftarrow (C, Sub_2)$, and $JS_4 \leftarrow (C, Sub_3)$. Then a valid joint state transition only exists between JS_1 and JS_2 since $GM(JS_1) \cap GM(JS_2) \neq \emptyset$ which is not true for JS_1 and JS_3 as well as for JS_1 and JS_4. In the first modelling alternative, this knowledge is already available for path search algorithms from the purely combinatorial model of the intra-layer graph of L_{sub}.

As a consequence, the planning of paths containing the subspace cells on L_{sub} involves additional geometric tests in the second alternative (which may, of course, be precomputed). While this can be viewed as disadvantage, a strong benefit of the second model is that it avoids redundancy. One motivation for the modelling of indoor space on different space layers is that changes on one layer should not interfere with the space representation on a second layer. However, if the spatial layout or configuration of room R_1 or room R_2 is altered on L_{super} in the first modelling alternative (e.g., a door is installed between both rooms) this change has to be reflected on L_{sub} since it contains copies of both rooms. Otherwise, it alone will not be suitable for path planning any more. Moreover, the redundant representation of both R_1 and R_2 is less efficient with respect to the storage and management of the entire indoor space model. For example, suppose a topographic space layer that contains a large number of rooms and corridors but a subspace layer that provides subdivisions for only a small subset thereof. Then the number of redundant space cells required on the subspace layer clearly outweighs the number of subspace cells whereas only the latter enrich the information about the topographic space and hence add value to the indoor space model.

It can be observed that the subspace layer L_{sub} from the first alternative shown in figure 118a in fact provides an integrated representation of both space layers as modelled in the second alternative. So the question is whether a formal operation can be defined which *merges* two space layers so that this integrated representation is obtained.

A first step towards the development of a *merge operation* for space layers is to define a map that carries two overlapping space cells to a set of mutually non-overlapping space cells covering the same partition of indoor space. Let X and Y be the two overlapping space cells as shown in figure 119a. Then three non-overlapping space cells $S_{X\backslash Y}$, $S_{Y\backslash X}$, and $S_{X\cap Y}$ suffice to describe the same space as shown in figure 119b. Their primal space geometries result from *regularized Boolean set operations* applied to $GM(X)$ and $GM(Y)$. Regularized Boolean set operations ensure that the result of every operation on two n-dimensional subsets of $\mathbb{R}^n$ is again n-dimensional (Farin et al. 2002). Hence, pathological cases such as isolated lower-dimensional elements or open boundaries are excluded which may be produced by an ordinary Boolean set operation but which do not conform to the notion of the primal space geometry of space cells (cf. definition 3.2).

Definition 3.64 (Regularized Boolean set operations). Let P and Q be two n-dimensional point sets in $\mathbb{R}^n$. A *regularized Boolean set operation* denoted by op^* is realized by first taking the interior of the point set obtained by the ordinary Boolean set operation $P\ op\ Q$ and then by taking the closure, and thus $P\ op^*\ Q = \overline{Int(P\ op\ Q)}$ with $op \in \{\cup,\cap,\backslash\}$ (e.g., Gold 2004, Foley 2010).

For example, the regularized difference denoted by $\backslash^*$ between $GM(X)$ and $GM(Y)$ is given as $GM(X) \backslash^* GM(Y) = \overline{Int(GM(X)\backslash GM(Y))}$. Likewise, their regularized intersection $\cap^*$ is expressed through $GM(X) \cap^* GM(Y) = \overline{Int(GM(X) \cap GM(Y))}$.[21]

Remember that per definition 3.2, the primal space geometry of a space cell needs to render a non-empty, connected, and compact n-manifold with boundary. This condition is fulfilled for all three space cells $S_{X\backslash Y}$, $S_{Y\backslash X}$, and $S_{X\cap Y}$ in figure 119b when setting $GM(S_{X\backslash Y}) = GM(X) \backslash^* GM(Y)$, $GM(S_{Y\backslash X}) = GM(Y) \backslash^* GM(X)$, and $GM(S_{X\cap Y}) = GM(X) \cap^* GM(Y)$.

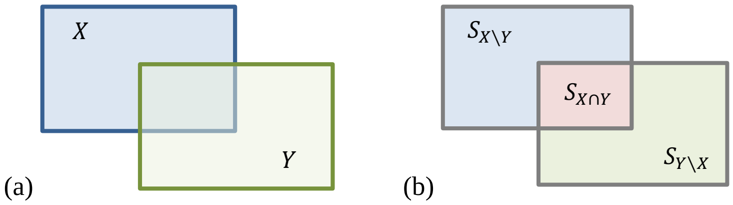


Figure 119: Two overlapping space cells (a) and three non-overlapping space cells covering the same space (b).

Although $GM(X) \backslash^* GM(Y)$, and $GM(Y) \backslash^* GM(X)$, and $GM(X) \cap^* GM(Y)$ are always closed and bounded subsets of $\mathbb{R}^n$ per definition 3.64 and hence compact according to the Heine-Borel theorem (cf. theorem A.30), they not necessarily satisfy the connectedness condition of definition 3.2. Moreover, both $GM(X) \backslash^* GM(Y)$ and $GM(Y) \backslash^* GM(X)$ may be empty sets. Supporting examples are presented in figure 120a to c. For instance, consider $GM(X) \backslash^* GM(Y)$ in figure 120a which results in two non-empty but disconnected subsets of $\mathbb{R}^n$ and thus cannot be captured by a single space cell but rather has to be carried to two space cells instead. In contrast, $GM(X) \backslash^* GM(Y)$ yields the empty set in figure 120c which is not allowed for the primal space geometry of a space cell.

[21] For example, consider two manifold solids $M_1$and M_2 living in $\mathbb{R}^3$ which touch at their boundary. Then the intersection $M_1 \cap M_2$ is of dimension less than three. Thus, the topological interior $Int(M_1 \cap M_2)$ is necessarily the empty set since no point in $M_1 \cap M_2$ has an open neighbourhood homeomorphic to the open unit 3-ball. It follows that $M_1 \cap^* M_2 = \overline{Int(M_1 \cap M_2)}$ is also the empty set. The same consideration holds in case of the regularized difference and union operation. Note that the topological interior may not be confused with the manifold interior which needs not be empty.

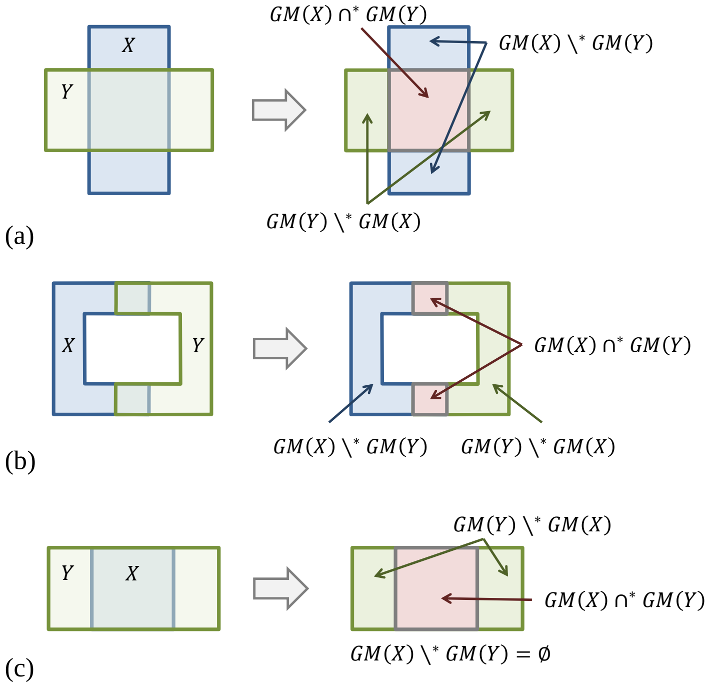

(a)
X
Y
$GM(X) \cap^* GM(Y)$
$GM(X) \setminus^* GM(Y)$
$GM(Y) \setminus^* GM(X)$
(b)
X
Y
$GM(X) \cap^* GM(Y)$
$GM(X) \setminus^* GM(Y)$
$GM(Y) \setminus^* GM(X)$
(c)
Y
X
$GM(Y) \setminus^* GM(X)$
$GM(X) \cap^* GM(Y)$
$GM(X) \setminus^* GM(Y) = \emptyset$

Figure 120: Examples of different intersection results between two space cells.

Even if a regularized Boolean set operation produces a non-empty, connected, and compact subset of $\mathbb{R}^n$ it may still be non-manifold. Precisely, the class of topological manifolds is not closed under regularized Boolean set operations (Farin et al. 2002). This is demonstrated in figure 121a for the operation $GM(X) \setminus^* GM(Y)$. Consequently, $GM(X) \setminus^* GM(Y)$ is also not suitable as primal space geometry of a single space cell. In order to resolve this issue, $GM(X) \setminus^* GM(Y)$ has to be partitioned into finitely many non-overlapping manifold subsets each of which is then described by a separate space cell. A possible resolution following this schema is shown in figure 121b where $GM(X) \setminus^* GM(Y)$ is completely covered by two valid space cells. Algorithms for detecting non-manifold structures as well as for carrying them to manifold representations are, for example, presented by (Mäntylä 1988) or (Farin et al. 2002).

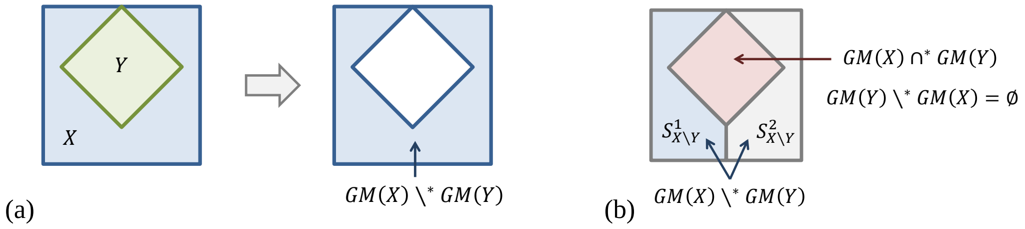

Y
X
(a)
$GM(X) \setminus^* GM(Y)$
(b)
$GM(X) \cap^* GM(Y)$
$GM(Y) \setminus^* GM(X) = \emptyset$
$S^1_{X\setminus Y}$
$S^2_{X\setminus Y}$
$GM(X) \setminus^* GM(Y)$

Figure 121: Mapping a non-manifold intersection result (a) onto a set of space cells (b).

It follows from the examples that in the general case it takes three sets $\mathcal{S}_{X\setminus Y}$, $\mathcal{S}_{Y\setminus X}$, and $\mathcal{S}_{X\cap Y}$ of mutually non-overlapping space cells instead of just three space cells as in figure 119 to describe the space occupied by X and Y, and both $\mathcal{S}_{X\setminus Y}$ and $\mathcal{S}_{Y\setminus X}$ may be the empty set.

Definition 3.65 (Make disjoint map for two space cells). Let X and Y be two n-dimensional overlapping space cells with $Int(GM(X)) \cap Int(GM(Y)) \neq \emptyset$. Then the continuous and surjective map $f_{DISJOINT}: \{X, Y\} \to \mathcal{C}_{XY}$ carries X and Y to a space cell complex $\mathcal{C}_{XY}$ that is the union of three disjoint sets $\mathcal{S}_{X\setminus Y}$, $\mathcal{S}_{Y\setminus X}$, and $\mathcal{S}_{X\cap Y}$ so that

(i) $\forall S_{X\setminus Y} \in \mathcal{S}_{X\setminus Y}: GM\big(S_{X\setminus Y}\big) \subseteq GM(X) \setminus^* GM(Y) \wedge A\big(S_{X\setminus Y}\big) = A(X)$, with $\big|\mathcal{S}_{X\setminus Y}\big| \geq 0$ and $\bigcup_{\alpha \in I_{\mathcal{S}_{X\setminus Y}}} GM(S^{\alpha}_{X\setminus Y}) = GM(X) \setminus^* GM(Y)$,

(ii) $\forall S_{Y\setminus X} \in \mathcal{S}_{Y\setminus X}: GM\big(S_{Y\setminus X}\big) \subseteq GM(Y) \setminus^* GM(X) \wedge A\big(S_{Y\setminus X}\big) = A(Y)$, with $\big|\mathcal{S}_{Y\setminus X}\big| \geq 0$ and $\bigcup_{\beta \in I_{\mathcal{S}_{Y\setminus X}}} GM(S^{\beta}_{Y\setminus X}) = GM(Y) \setminus^* GM(X)$, and

(iii) $\forall S_{X\cap Y} \in \mathcal{S}_{X\cap Y}: GM(S_{X\cap Y}) \subseteq GM(X) \cap^* GM(Y) \wedge A(S_{X\cap Y}) = A(X) \cup A(Y)$, with $|\mathcal{S}_{X\cap Y}| \geq 1$ and $\bigcup_{\gamma \in I_{\mathcal{S}_{X\cap Y}}} GM(S^{\gamma}_{X\cap Y}) = GM(X) \cap^* GM(Y)$.

The operation of applying $f_{DISJOINT}$ to two space cells X and Y is called *make disjoint*. The number of space cells in every subset of $\mathcal{C}_{XY}$ has to be equal to the number of connected components resulting from the regularized Boolean operation given that each component is manifold. This condition can be expressed using the first Betti number β_0 of the corresponding topological spaces so that $|\mathcal{S}_{X\setminus Y}| = \beta_0(GM(X) \setminus^* GM(Y))$, $|\mathcal{S}_{Y\setminus X}| = \beta_0(GM(Y) \setminus^* GM(X))$, and $|\mathcal{S}_{X\cap Y}| = \beta_0(GM(X) \cap^* GM(Y))$ (cf. theorem A.94).

The above definition ensures that the primal space geometry $GM(f_{DISJOINT}(X,Y))$ is equal to $GM(X) \cup^* GM(Y)$ and is described by a set of valid space cells. Additionally, it addresses the transition of symbolic and semantic attributes from X and Y to each space cell in $\mathcal{C}_{XY} = f_{DISJOINT}(X,Y)$. Since every $S_{X\setminus Y} \in \mathcal{S}_{X\setminus Y}$ represents a partition of space occupied merely by X, it is assigned the attribute set $A(X)$. For the similar reason, every $S_{Y\setminus X} \in \mathcal{S}_{Y\setminus X}$ is assigned the attribute set $A(Y)$. The space cells in $\mathcal{S}_{X\cap Y}$ cover both X and Y and hence retrieve the union $A(X) \cup A(Y)$ of both attribute sets.

The map $f_{DISJOINT}$ also has to yield the primal space topologies of every space cell participating in $\mathcal{C}_{XY}$ as well as of $\mathcal{C}_{XY}$ itself in a deterministic way. The following definition utilizes the geometric carriers of the topological $(n-1)$-cells in both $TP(X)$ and $TP(Y)$ for this purpose (cf. definition 3.3 and the related discussion in chapter 3.1.1.3). The approach is to intersect each of these carriers with the boundary $\partial GM(S_i)$ of a space cell $S_i \in \mathcal{C}_{XY}$ in primal geometry space in order to retrieve the set of geometric carriers for the $(n-1)$-cells in the primal topological description $TP(S_i)$. Since the resulting geometric carriers need to be in one-to-one correspondence with the $(n-1)$-cells, the $(n-1)$-skeleton of $TP(S_i)$ can be constructed based on the carriers and their inclusion relations. This approach guarantees that the cell decomposition of the boundary of S_i has at least the same granularity as the cell decomposition of the corresponding boundary part of X and Y, and thus no information is lost in primal topology space when applying $f_{DISJOINT}$ to X and Y. Note that the dual topology space representation of all space cells in $\mathcal{C}_{XY}$ automatically follows when embedding $\mathcal{C}_{XY}$ in a space layer.

Definition 3.66 (Primal space topology of $f_{DISJOINT}(X,Y)$). Let $\mathcal{C}_{XY} = f_{DISJOINT}(X,Y)$ be the space cell complex obtained for two n-dimensional overlapping space cells X and Y. Further, let G_{n-1} be the set of all $(n-1)$-dimensional geometric carriers in bijective correspondence with the $(n-1)$-cells in both $TP(X)$ and $TP(Y)$.

(i) For all $S_i \in \mathcal{C}_{XY}$, $TP(S_i)$ is derived as follows:

a. Let K_{n-1} be the set of $(n-1)$-dimensional geometrics carriers for $TP(S_i)$ so that $\forall g \in G_{n-1}: g \cap^* \partial GM(S_i) \neq \emptyset \Rightarrow g \cap^* \partial GM(S_i) \in K_{n-1}$ and $\forall k, l \in K_{n-1}: Int(k) \cap Int(l) = \emptyset$.

b. $\partial GM(S_i) \setminus^* \bigcup_{\alpha \in I_{K_{n-1}}} k_\alpha \neq \emptyset \Rightarrow \partial GM(S_i) \setminus^* \bigcup_{\alpha \in I_{K_{n-1}}} k_\alpha \in K_{n-1}$ (e.g., required in figure 121).

c. For every $k \in K_{n-1}$ an $(n-1)$-cell e_{n-1} is added to $TP(S_i)$ whose geometric carrier is k. The lower dimensional cells as well as their geometric carriers are induced by the inclusion relations between the carries in K_{n-1}.

d. $TP(S_i)$ is obtained by attaching a single n-cell e_n to the $(n-1)$-skeleton $TP(S_i)_{n-1}$. Depending on the dimension n, additional bridge edges or bridge faces might be necessary (cf. proposition 3.34 and proposition 3.35). The geometric carrier of e_n is $GM(S_i)$.

(ii) $TP(\mathcal{C}_{XY})$ results from identifying the $(n-1)$-cells on the common boundaries of all space cells participating in $\mathcal{C}_{XY}$.

The following figure 122 exemplifies the result of $f_{DISJOINT}$ in primal topology space for two 2-dimensional space cells X and Y. In figure 122a, the space cells as well as their associated CW decompositions provided by $TP(X)$ and $TP(Y)$ are illustrated. Note that both primal space geometries $GM(X)$ and $GM(Y)$ describe rectangular shapes in $\mathbb{R}^2$. Whereas the topological surface underlying $GM(X)$ is decomposed into four 0-cells, four 1-cells and one 2-cell by $TP(X)$, a minimal cell decomposition into one 0-cell, one 1-cell and one 2-cell is assumed for $TP(Y)$. Correspondingly, the 1-dimensional geometric carrier of the 1-cell in $TP(Y)$ is the entire boundary $\partial GM(Y)$ of the primal space geometry of Y being homeomorphic to the 1-sphere $\mathbb{S}^1$. In case of the space cell X, the four straight line segments on the boundary $\partial GM(X)$ can be chosen as 1-dimensional geometric carriers for the 1-cells in $TP(X)$ instead with each being homeomorphic to $\overline{\mathbb{B}}^1$.

Figure 122b shows the overlap of the primal space geometries of X and Y in $\mathbb{R}^2$. When applying $f_{DISJOINT}(X,Y)$ to this configuration, the three space cells $S_{X\setminus Y}$, $S_{Y\setminus X}$, and $S_{X\cap Y}$ are retrieved. They are depicted in an exploded view in figure 122c to better visualize their individual primal space topologies. For example, $TP(S_{X\cap Y})$ consists of three 0-cells, three 1-cells and one 2-cell. Consider the 1-cell $e_1 \in TP(S_{X\cap Y})$. Its geometric carrier follows from the regularized Boolean intersection of the boundary $\partial GM(S_{X\cap Y})$ with the only 1-dimensional geometric carrier $\partial GM(Y)$ associated with $TP(Y)$ (cf. (i) a of definition 3.66). The resulting subset of $\mathbb{R}^2$ is homeomorphic to $\overline{\mathbb{B}}^1$ whose interior $Int(\overline{\mathbb{B}}^1) = \mathbb{B}^1$ is captured by e_1 and whose boundary $\partial\overline{\mathbb{B}}^1$ is carried to two 0-cells (cf. (i) c of definition 3.66). Likewise, the geometric carriers of e_2 and e_3 result from intersecting $\partial GM(S_{X\cap Y})$ with the straight line segments on the boundary $\partial GM(X)$ of the space cell X. The CW decompositions of $TP(S_{X\setminus Y})$ and $TP(S_{Y\setminus X})$ can be explained in a similar way.

Finally, figure 122d illustrates that the CW decomposition of the resulting primal space topology $TP(f_{DISJOINT}(X,Y))$ has finer granularity than both $TP(X)$ and $TP(Y)$.

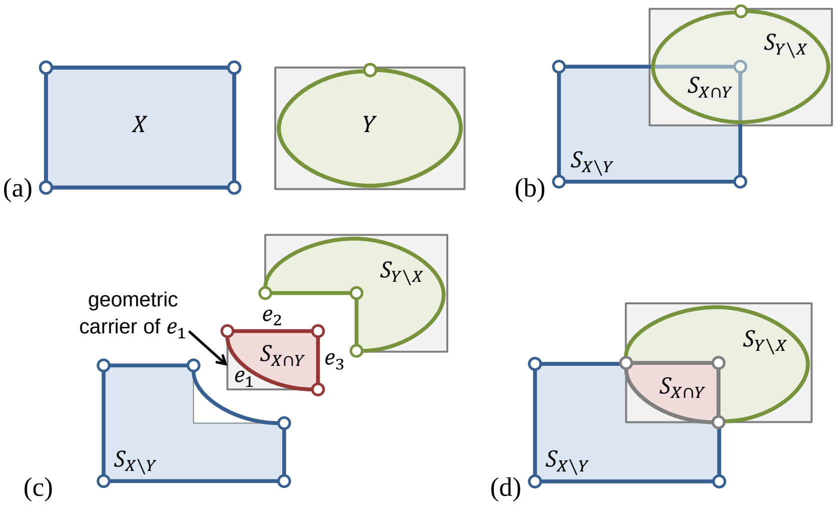


Figure 122: Two rectangular space cells with different representations in primal topology space (a), their overlap in primal space (b), and the result of the $f_{DISJOINT}$ map as exploded (c) and non-exploded view (d).

The formal definition of $f_{DISJOINT}$ allows for realizing a *merge operation* for two space layers sharing the same dimension. The main idea of this operation is to *make* the overlapping space cells on both space layers *disjoint*. The *MergeSpaceLayers* algorithm presented below (cf. algorithm 3.67) illustrates the sequence of steps to implement the merge operation. Its input are two space layers L_1 and L_2, and it yields their merged representation as another space layer L_{merge}. The algorithm works on copies of the space cell complexes $\mathcal{C}(L_1)$ and $\mathcal{C}(L_2)$ which are called $\mathcal{C}^1_{tmp}$ respectively $\mathcal{C}^2_{tmp}$ so that neither L_1 nor L_2 are affected by the algorithm. Moreover, the output space layer L_{merge} is initialized with the minimal space layer L_{min} to guarantee a non-empty result (cf. lines 1-3).

In a main iteration loop (cf. lines 4-19), first every space cell $X \in \mathcal{C}^1_{tmp}$ which does not spatially overlap with any other space cell in $\mathcal{C}^2_{tmp}$ is directly put in the resulting space cell complex $\mathcal{C}(L_{merge})$ of L_{merge}. Since X however is necessarily covered by $S_{out}(L_2)$, it receives the semantic attributes modelled for $S_{out}(L_2)$. The space cell X is then removed from $\mathcal{C}^1_{tmp}$ (cf. lines 5-9). In a second step, equivalent actions are performed on all $Y \in \mathcal{C}^2_{tmp}$ for which there is no overlapping space cell in $\mathcal{C}^1_{tmp}$ (cf. lines 10-14). Third, a pair $\{X, Y\}$ of overlapping space cells, with $X \in \mathcal{C}^1_{tmp}$ and $Y \in \mathcal{C}^2_{tmp}$, is taken as input for the map $f_{DISJOINT}$ which results in $\mathcal{C}_{XY} = f_{DISJOINT}(X,Y)$. The space cells contained in $S_{X\cap Y}(\mathcal{C}_{XY})$ cannot possibly overlap with any other space cell on L_1 or L_2 and thus are added to $\mathcal{C}(L_{merge})$. However, this needs not be true for the space cells in both $S_{X\setminus Y}(\mathcal{C}_{XY})$ and $S_{Y\setminus X}(\mathcal{C}_{XY})$ which are therefore pushed back into $\mathcal{C}^1_{tmp}$ respectively $\mathcal{C}^2_{tmp}$ while at the same time removing X and Y from these sets (cf. lines 15-19). The execution of the main loop continues until both $\mathcal{C}^1_{tmp}$ and $\mathcal{C}^2_{tmp}$ are empty sets.

As soon as the main loop ends, the space cell complex $\mathcal{C}(L_{merge})$ of the output space layer is complete. Note that the primal space representation of $S_{out}(L_{merge})$ implicitly follows from $\mathcal{C}(L_{merge})$ (cf. definition 3.12 and definition 3.13). Likewise, the dual topological space representation of all space cells on L_{merge} as well as of L_{merge} itself is a consequence of applying the Poincaré duality to $\mathcal{C}(L_{merge})$ according to definition 3.16. If the geometric-topological configuration of the space cells in $\mathcal{C}(L_{merge})$ is identical to that of either $\mathcal{C}(L_1)$ or $\mathcal{C}(L_2)$, then the dual geometric representation of the corresponding space layer is taken for $G_{GM}(L_{merge})$ (cf. lines 20-25). Otherwise, a deterministic *GetDefaultEmbedding* algorithm for the derivation of $G_{GM}(L_{merge})$ is assumed (e.g., assigning the centroids of space cells to their dual nodes and embedding the dual edges using straight line segments). In the final part of the algorithm (cf. lines 26-32), symbolic and semantic attributes from boundary cells in both $\mathcal{B}(L_1)$ and $\mathcal{B}(L_2)$ are carried to overlapping ones in $\mathcal{B}(L_{merge})$. Further actions on boundary cells are not necessary since their primal and dual space representations are again derived from the space cell complex $\mathcal{C}(L_{merge})$. The outer space cell $S_{out}(L_{merge})$ is assigned the attributes from both $S_{out}(L_1)$ and $S_{out}(L_2)$, and the resulting space layer L_{merge} itself receives the union of attributes from L_1 and L_2.

Algorithm 3.67. MergeSpaceLayers(L_1,L_2)

```
Input:   $L_1, L_2$, with $dim(L_1) = dim(L_2)$
Output:  $L_{merge}$
 1: $\mathcal{C}^1_{tmp} \leftarrow \mathcal{C}(L_1)$
 2: $\mathcal{C}^2_{tmp} \leftarrow \mathcal{C}(L_2)$
 3: $L_{merge} \leftarrow L_{min}$
 4: while $\mathcal{C}^1_{tmp} \neq \emptyset \wedge \mathcal{C}^2_{tmp} \neq \emptyset$ do
 5:     $Q \leftarrow \{X \in \mathcal{C}^1_{tmp} \mid Int(GM(X)) \cap Int(GM(Y)) = \emptyset, \forall Y \in \mathcal{C}^2_{tmp}\}$
 6:     for each $X \in Q$ do
 7:         $A(X) \leftarrow A(X) \cup S_{out}(L_2)$
 8:         $\mathcal{C}(L_{merge}) \leftarrow \mathcal{C}(L_{merge}) \cup X$
 9:         $\mathcal{C}^1_{tmp} \leftarrow \mathcal{C}^1_{tmp} \backslash X$
10:     $Q \leftarrow \{Y \in \mathcal{C}^2_{tmp} \mid Int(GM(Y)) \cap Int(GM(X)) = \emptyset, \forall X \in \mathcal{C}^1_{tmp}\}$
11:     for each $Y \in Q$ do
12:         $A(Y) \leftarrow A(Y) \cup S_{out}(L_1)$
13:         $\mathcal{C}(L_{merge}) \leftarrow \mathcal{C}(L_{merge}) \cup Y$
14:         $\mathcal{C}^2_{tmp} \leftarrow \mathcal{C}^2_{tmp} \backslash Y$
15:     if $\exists\{X, Y\}: Int(GM(X)) \cap Int(GM(Y)) \neq \emptyset \wedge X \in \mathcal{C}^1_{tmp} \wedge Y \in \mathcal{C}^2_{tmp}$ then
16:         $\mathcal{C}_{XY} \leftarrow f_{DISJOINT}(X, Y)$
17:         $\mathcal{C}(L_{merge}) \leftarrow \mathcal{C}(L_{merge}) \cup \mathcal{S}_{X \cap Y}(\mathcal{C}_{XY})$
18:         $\mathcal{C}^1_{tmp} \leftarrow \mathcal{C}^1_{tmp} \backslash X \cup \mathcal{S}_{X \backslash Y}(\mathcal{C}_{XY})$
19:         $\mathcal{C}^2_{tmp} \leftarrow \mathcal{C}^2_{tmp} \backslash Y \cup \mathcal{S}_{Y \backslash X}(\mathcal{C}_{XY})$
20: if $\mathcal{C}(L_{merge}) = \mathcal{C}(L_1)$ then
21:     $G_{GM}(L_{merge}) \leftarrow G_{GM}(L_1)$
22: else if $\mathcal{C}(L_{merge}) = \mathcal{C}(L_2)$ then
23:     $G_{GM}(L_{merge}) \leftarrow G_{GM}(L_2)$
24: else
25:     $G_{GM}(L_{merge}) \leftarrow GetDefaultEmbedding(G_{TP}(L_{merge}))$
26: for each $B \in \mathcal{B}(L_{merge})$ do
27:     $Q \leftarrow \{C \in \mathcal{B}(L_1) \mid Int(GM(B)) \cap Int(GM(C)) \neq \emptyset\}$
28:     $Q \leftarrow Q \cup \{C \in \mathcal{B}(L_2) \mid Int(GM(B)) \cap Int(GM(C)) \neq \emptyset\}$
29:     for each $C \in Q$ do
30:         $A(B) \leftarrow A(B) \cup A(C)$
31: $A\left(S_{out}(L_{merge})\right) \leftarrow A(S_{out}(L_1)) \cup A(S_{out}(L_2))$
32: $A(L_{merge}) \leftarrow A(L_1) \cup A(L_2)$
```

Based on the *MergeSpaceLayers* algorithm, the merge operation is formally defined as follows.

Definition 3.68 (Merge operation on two space layers). Let $\mathbb{L}^n$ be the non-empty set of all n-dimensional space layers, with $2 \leq n \leq 3$. Then the map $\bigoplus: \mathbb{L}^n \times \mathbb{L}^n \rightarrow \mathbb{L}^n$ uniquely associates each pair of space layers $\{L_1, L_2\}$ in

$\mathbb{L}^n$ with another space layer L_{merge} of $\mathbb{L}^n$ as defined by algorithm 3.67. It thus yields a closed binary operation on $\mathbb{L}^n$ called *merge operation* which is denoted by $L_1 \oplus L_2 = L_{merge}$.

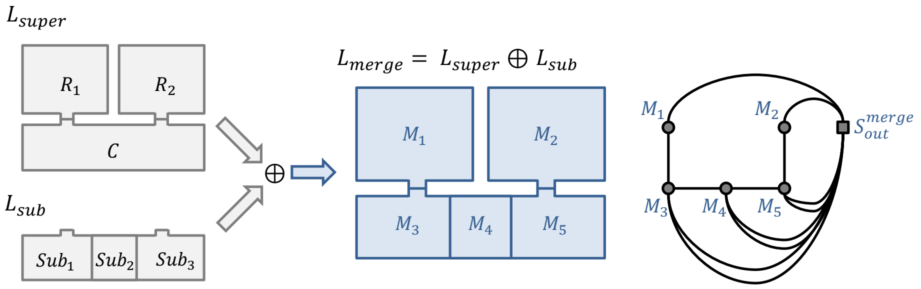
The result of the merge operation does not depend on the ordering of its two operands $L_1, L_2 \in \mathbb{L}^n$. This is due to the fact that for every pair $\{X, Y\}$ of overlapping space cells, with $X \in \mathcal{C}(L_1)$ and $Y \in \mathcal{C}(L_2)$, the resulting primal space representation $\mathcal{C}\left(L_{merge}\right)$ of L_{merge} contains both the intersection and the difference of their primal space geometries (cf. lines 17-19 of algorithm 3.67). The merge operation is therefore *commutative*, and thus $L_1 \oplus L_2 = L_2 \oplus L_1$. For the same reason, $\oplus$ also has the *associative* property and it holds that $(L_1 \oplus L_2) \oplus L_3 = L_1 \oplus (L_2 \oplus L_3)$ for all elements L_1, L_2, and L_3 of $\mathbb{L}^n$. In mathematics, especially in the field of abstract algebra, the pair $(\mathbb{L}^n, \oplus)$ is thus said to be a *commutative* or *Abelian semigroup* (cf. Jacobson 2009).

We can even identify more characteristics of $(\mathbb{L}^n, \oplus)$. First, the merge operation is *idempotent* since when applied to the same space layer $L \in \mathbb{L}^n$, it gives L as result. Precisely, in case $L \oplus L$, the space cells in every pair $\{X, Y\}$ of overlapping space cells necessarily occupy the same indoor space, and thus $f_{DISJOINT}(X, Y) = \{X\} = \{Y\}$. Consequently, $\mathcal{C}\left(L_{merge}\right)$ equals $\mathcal{C}(L)$ and according to line 21 of algorithm 3.67 $G_{GM}\left(L_{merge}\right) = G_{GM}(L)$. And since all semantic attributes of L and its elements are carried to L_{merge} it follows that $L \oplus L = L_{merge} = L$. Second, L_{min} is the *identity element* for $\oplus$. Consider the merge operation $L \oplus L_{min}$. Due to $\mathcal{C}(L_{min}) = \emptyset$ per definition 3.32, all space cells on L are carried to $\mathcal{C}\left(L_{merge}\right)$ in a single iteration of the main loop in algorithm 3.67 (cf. lines 5-9). Thus, again $\mathcal{C}\left(L_{merge}\right) = \mathcal{C}(L)$ and $G_{GM}\left(L_{merge}\right) = G_{GM}(L)$ which yields $L \oplus L_{min} = L$, and through the commutative property $L_{min} \oplus L = L$. The existence of an identity element renders the algebraic structure $(\mathbb{L}^n, \oplus)$ not only an Abelian semigroup but also an *Abelian monoid* (cf. Jacobson 2009).

The space partitioning of the resulting space layer L_{merge} necessarily has equal or finer granularity than that of both operands of the merge operation. This excludes the existence of *inverse elements* in $(\mathbb{L}^n, \oplus)$. The idea of an inverse element is that it *cancels the action* of its original element. Formally, an inverse element $L^{-1} \in \mathbb{L}^n$ for a given space layer $L \in \mathbb{L}^n$ with $L \neq L_{min}$ requires that $L \oplus L^{-1} = L^{-1} \oplus L = L_{min}$. However, since $\mathcal{C}(L) \neq \emptyset$ it follows that $\mathcal{C}(L \oplus L^{-1}) \neq \emptyset$ and thus $\mathcal{C}(L \oplus L^{-1}) \neq \mathcal{C}(L_{min})$ which contradicts $L \oplus L^{-1} = L_{min}$. All space layers in $\mathbb{L}^n$ are hence *non-invertible* regarding the merge operation and do not possess an inverse element which would cancel the merge with another space layer. The only obvious exception from this rule is the identity element L_{min} itself which is thus said to be a *self-inverse element* of the monoid $(\mathbb{L}^n, \oplus)$ with $L_{min} \oplus L_{min} = L_{min}$.

The merge operation is applicable to any number of space layers so that $L_{merge} = L_1 \oplus L_2 \oplus \ldots \oplus L_k$, with $L_1, L_2, \ldots, L_k \in \mathbb{L}^n$. Due to $\oplus$ being commutative, L_{merge} is independent of the order in which the space layers occur in the sequence. Moreover, the merge operation abstracts from the choice of notion of indoor space underlying its input space layers. It can therefore be used to merge arbitrary space layers reflecting topographic space, sensor space, logical space, or any alternative notion of space. The merge operation intuitively resolves the ambiguity between space layers. Before merging, the same partition of indoor space is occupied by k space cells having heterogeneous semantics. After merging, the space cells on L_{merge} are mutually non-overlapping and thus render a homogeneous and unambiguous representation of indoor space.

Example 3.69. The following figure 123 illustrates the merge operation for the space layers L_{super} and L_{sub} from the second alternative of the introductory example (cf. figure 118b).



M_1	$R_1 \cap S_{out}^{sub}$
M_2	$R_2 \cap S_{out}^{sub}$
M_3	$C \cap Sub_1$
M_4	$C \cap Sub_2$
M_5	$C \cap Sub_3$
S_{out}^{merge}	$S_{out}^{super} \cap S_{out}^{sub}$

Figure 123: Applying the merge operation to the space layers from figure 118b. The table on the right documents the input space cells and the applied operations that lead to the space cells on the merge space layer.

The resulting space layer $L_{merge} = L_{super} \oplus L_{sub}$ captures the same space representation as the subspace layer in the first alternative of figure 118a. Thus, the merge operation allows for combining the advantages of both alternatives. On the one hand, every subspace layer only has to provide subspacings of superspace cells which are relevant to its own notion of space and partitioning schema (e.g., non-navigable spaces for wheelchair users, spaces affected by a fire incident, etc.) without the need for redundantly repeating all non-subspaced space cells from the superspace layer. This reduces modelling efforts and storage requirements, and changes to non-subspaced space cells on the superspace layer need not be reflected on the subspace layer. On the other hand, topological adjacency information between subspace cells and superspace cells which are not encoded in the multilayered graph can be derived by merging both layers. This, for example, supports path planning as corresponding algorithms only have to process the intra-layer graph of the merged space layer instead of simultaneously evaluating the intra-layer graphs from both L_{super} and L_{sub} as well as joint state transitions. Typically, many path finding algorithms such as Dijkstra's algorithm (Dijkstra 1959), its more efficient A^* extension (cf. Russell & Norvig 2010), or the Bellman-Ford algorithm (Bellman 1958, Ford 1956) are not designed for multilayered graph structures.

Suppose that the space layer L_{super} in figure 123 represents the navigable topographic space for pedestrians, which is expressed through an attribute $a = (\text{"isNavigable"}, true)$ (cf. example 3.63 in chapter 3.4) modelled for each of its space cells, i.e. $\forall S \in \mathcal{C}(L_{super}): a \in A(S)$. Moreover, assume that L_{sub} provides a more fine-grained decomposition of the corridor for wheelchair users, and let $Sub_2 \in L_{sub}$ represent a non-navigable space with $b = (\text{"isNavigable"}, false)$ and $b \in A(Sub_2)$. Merging both space layers then provides an integrated space representation for wheelchair users. The space cell $M_4 \in L_{merge}$ resulting from $f_{DISJOINT}(C, Sub_2)$ is assigned the attributes from both C and Sub_2 due to $A(M_4) = A(C) \cup A(Sub_2)$ (cf. condition (iii) of definition 3.65) which yields $A(M_4) = \{(\text{"isNavigable"}, true), (\text{"isNavigable"}, false), \ldots\}$. In general, attributes of a merged space cell sharing the same name need to be jointly evaluated since each attribute is a valid thematic characterization of the partition of indoor space covered by the space cell. The joint evaluation depends on the type of the attribute value and the query to be answered. In the given example, answering whether M_4 is navigable for wheelchair users within a path query requires a logical conjunction of the Boolean values of the attributes named "isNavigable" which results in $true \wedge false = false$. The example illustrates that the merge operation preserves and integrates the semantic and symbolic attributes of both space cells and boundary cells from either operands. Note that a more powerful and expressive model for navigation constraints is introduced in chapter 5. However, the presented rules are also applicable to this model (cf. chapter 5.4).

Although it is an immediate consequence of algorithm 3.67 that $GM(\mathcal{C}(L_{super})) \subseteq GM(\mathcal{C}(L_{merge}))$ as well as $GM(\mathcal{C}(L_{sub})) \subseteq GM(\mathcal{C}(L_{merge}))$, figure 123 clearly shows that L_{merge} is only a superspace layer of L_{sub} but not of L_{super}. The inclusion relation $L_{sub} \leq L_{super}$ implies that the space cells on L_{super} are greater than the space cells on L_{sub}. Consequently, the space partitioning on L_{merge} cannot be more fine-grained than that of L_{sub} and it follows that $L_{sub} \leq L_{merge}$. However, in the general case that the operands of the merge operation are not related by inclusion the resulting space layer L_{merge} is not a superspace layer of either operand.

Proposition 3.70. Let $L_1, L_2 \in \mathbb{L}^n$ bet two space layers from the non-empty set $\mathbb{L}^n$ of all n-dimensional space layers. Then $GM(\mathcal{C}(L_1)) \subseteq GM(\mathcal{C}(L_1 \oplus L_2))$ and $GM(\mathcal{C}(L_2)) \subseteq GM(\mathcal{C}(L_1 \oplus L_2))$. Moreover, it holds that

(i) $L_1 \leq L_2 \Rightarrow L_1 \leq L_1 \oplus L_2$,

(ii) $L_2 \leq L_1 \Rightarrow L_2 \leq L_1 \oplus L_2$, and

(iii) $L_1 \nleq L_2 \wedge L_2 \nleq L_1 \Rightarrow L_1 \nleq L_1 \oplus L_2 \wedge L_2 \nleq L_1 \oplus L_2$.

Example 3.71. An example for merging two space layers with different notions of space is depicted in figure 124a. Whereas L_{Topo} represents the same topographic setting as in the previous example, L_{Sec} provides two security zones modelled by two separate space cells Sec_1 and Sec_2 which together cover all space cells on L_{Topo} (cf. figure 124b). Suppose that the security level is *low* for Sec_1 but *high* for Sec_2. Note that both space layers are not related by inclusion in this example, and hence $L_{Topo} \nleq L_{Sec} \wedge L_{Sec} \nleq L_{Topo}$. Moreover, a minimal CW decomposition is assumed for all space cells.

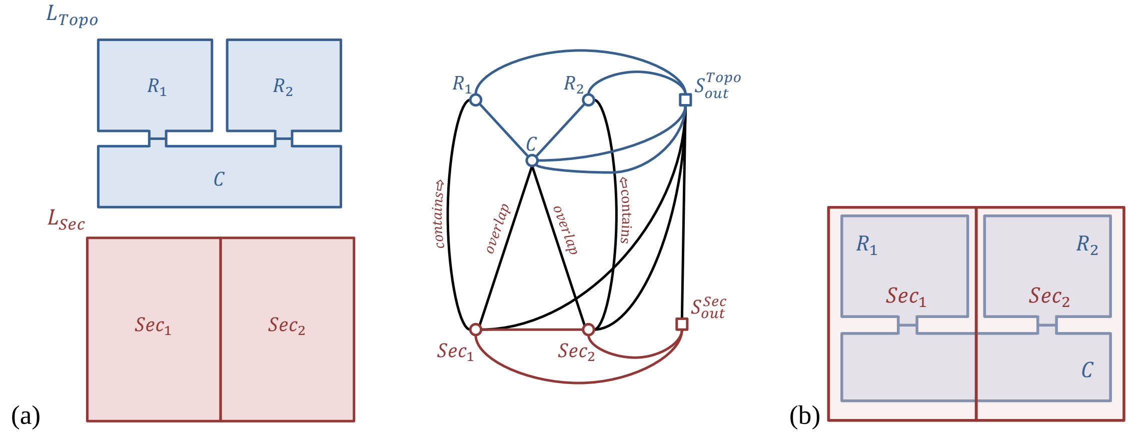
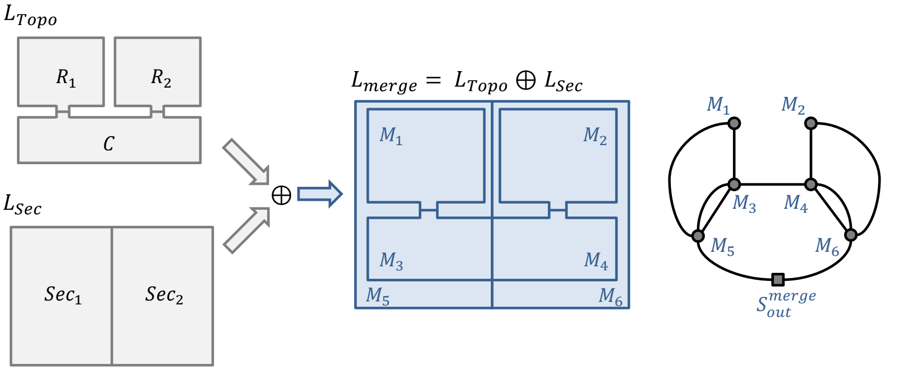


Figure 124: A topographic space layer and a security space layer (left) and their multilayered graph (right).

From the multilayered graph shown on the right of figure 124a the two distinct space cell hierarchies $\mathcal{A}(Sec_1) = \{Sec_1, R_1\}$ and $\mathcal{B}(Sec_2) = \{Sec_2, R_2\}$ can be deduced which reflect the space cells contained in the security zones. However, the corridor space cell C is not participating in either hierarchy since it spatially overlaps with both Sec_1 and Sec_2. Similar to the above example 3.69, it is thus not decidable based on the multilayered graph structure whether a navigation user having permission only for areas with low security level is allowed to enter the corridor C from room R_1. The merge operation $L_{Topo} \oplus L_{Sec}$ helps to resolve this ambiguity as illustrated in the following figure.



M_1	$R_1 \cap Sec_1$
M_2	$R_2 \cap Sec_2$
M_3	$C \cap Sec_1$
M_4	$C \cap Sec_2$
M_5	$Sec_1 \setminus R_1 \setminus C$
M_6	$Sec_2 \setminus R_2 \setminus C$
S_{out}^{merge}	$S_{out}^{Topo} \cap S_{out}^{Sec}$

Figure 125: Applying the merge operation to the space layers from figure 124.

Let $l_{low} = ("SecurityLevel", "low")$ and $l_{high} = ("SecurityLevel", "high")$ be the two semantic attributes capturing the information about the security level associated with the security zones, and thus $l_{low} \in A(Sec_1)$ and $l_{high} \in A(Sec_2)$. Further, let $s_{R_1} = ("PlaceLabel", "Room1") \in A(R_1)$, $s_{R_2} = ("PlaceLabel", "Room2") \in A(R_2)$, and $s_C = ("PlaceLabel", "Corridor") \in A(C)$ be the symbolic place labels of the architectural entities each of which is modelled as semantic attribute of a topographic space cell. After having merged both space layers, the semantics of overlapping space cells are combined and the attributes are available from the resulting space cells on L_{merge}. In order to answer whether a navigation user can move from R_1 to C without leaving the low security zone, we need to consider the space cells $M_1, M_3 \in L_{merge}$ because $s_{R_1}, l_{low} \in A(M_1)$ and $s_C, l_{low} \in A(M_3)$. Since the dual nodes of M_1 and M_3 are directly linked in the intra-layer graph of L_{merge}, there exists a valid path for the navigation user.

One important conclusion of the survey of existing approaches to indoor space modelling presented in chapter 2.2 was that most of the discussed proposals agree in only representing topographic regions and in mapping semantic information such as symbolic place labels or security levels onto attributes associated with the regions or their corresponding elements in the navigation graph (cf. chapter 2.4). The above example demonstrates that the MLSEM is capable of providing an equivalent space representation based on the merge operation. But even more importantly, it shows that the MLSEM is to be seen as a general framework that goes beyond the presented approaches and rather can be used to explain and derive attributively enriched topographic indoor space models which therefore can be understood as being more specific models.

Example 3.72. The merge operation is also feasible to obtain a single space layer for a given notion of space and partitioning schema. This is especially important in case the partitioning schema itself does not generate mutually non-overlapping space cells. In topographic space, non-overlapping space cells inherently follow from decomposing the interior built environment according to the architectural constraints. However, the following example shows a Wi-Fi space layer whose sensor space cells reflect the partitioning of indoor space along the signal coverage areas of two Wi-Fi transmitters (cf. figure 46 in chapter 3.1.2.3). Since Wi-Fi coverage areas are typically meant to overlap in order to ensure full signal coverage within a building or site, the sensor space cells consequently overlap in primal space. When modelling this space partitioning on a single Wi-Fi space layer, the sensor space cells need to be made disjoint beforehand as sketched below (cf. chapter 3.1.3).

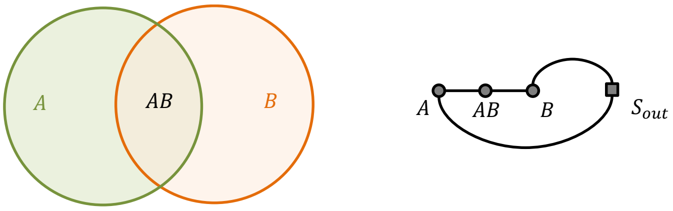
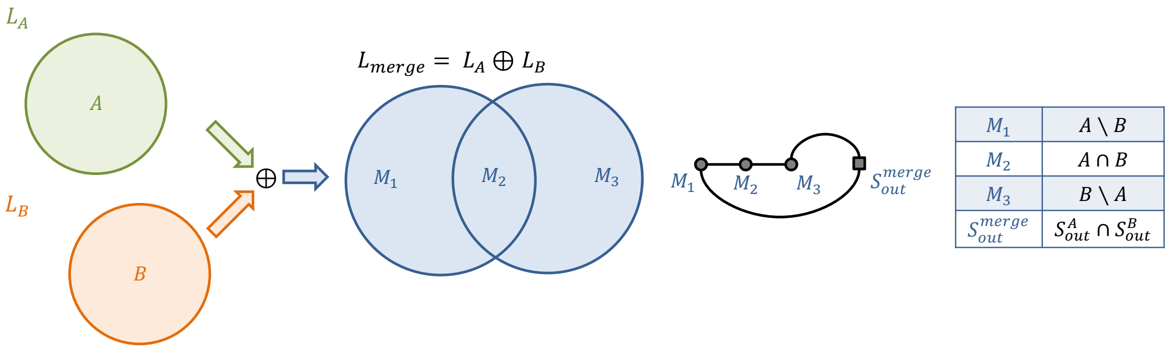


Figure 126: Three non-overlapping Wi-Fi space cells representing the overlapping coverage areas of two Wi-Fi transmitters on a sensor space layer (left) and the weak dual graph (right).

Obviously, the same configuration of space cells can be realized by merging two separate sensor space layers with each layer representing the signal coverage of a single Wi-Fi transmitter (cf. figure 127). This way of modelling has a number of benefits. First, the partitioning of indoor space along the characteristics of individual sensors necessarily results in non-overlapping space cells as shown on the left of figure 127 for both Wi-Fi space layers. Second, suppose the building is equipped with further Wi-Fi transmitters. Then the indoor space model can be simply augmented with additional sensor space layers capturing the signal coverage areas of each new Wi-Fi transmitter without interfering with the space representation on existing Wi-Fi layers. Likewise, if existing Wi-Fi transmitters are dismounted or temporarily break down then only their corresponding space layers need to be removed from the MLSEM. Third, the merge operation as illustrated in figure 127 can be performed on the fly in order to retrieve mutually non-overlapping coverage areas involving all Wi-Fi transmitters. This integrated space representation is advantageous for localization and tracking purposes as it results in smaller space cells and hence reduces the uncertainty about the absolute position of moving persons or objects (cf. chapter 3.3). Since the output of the merge operation is again a space layer, it can be processed and stored like any other space layer within the MLSEM.



M_1	$A \setminus B$
M_2	$A \cap B$
M_3	$B \setminus A$
S_{out}^{merge}	$S_{out}^{A} \cap S_{out}^{B}$

Figure 127: Modelling of the two Wi-Fi space cells from figure 126 on two sensor separate space layers (left) and the result of a corresponding merge operation (right) which yields the space layer from figure 126.

What has been demonstrated for Wi-Fi space cells in the above example is equivalently applicable to any other sensor space representation or notion of space. For example, instead of modelling both the low and high security zones on a single space layer in figure 124, individual logical space layers for each security level may be provided.

3.5.2 Difference Operation

Based on the $f_{DISJOINT}$ map introduced in the previous chapter, a second operation on the non-empty set of n-dimensional space layers $\mathbb{L}^n$ can be defined which returns the *difference* between two input space layers. Intuitively, given two operands $L_1, L_2 \in \mathbb{L}^n$, their difference results from spatially subtracting the space cells on L_2 from those on L_1. The following *SubstractSpaceLayers* algorithm formally solves the *difference operation*.

Algorithm 3.73. SubstractSpaceLayers(L_1,L_2)

```
Input:   $L_1, L_2$, with $dim(L_1) = dim(L_2)$
Output:  $L_{diff}$
 1:  $\mathcal{C}^1_{tmp} \leftarrow \mathcal{C}(L_1)$
 2:  $\mathcal{C}^2_{tmp} \leftarrow \mathcal{C}(L_2)$
 3:  $L_{diff} \leftarrow L_{min}$
 4:  $Q \leftarrow \{X \in \mathcal{C}^1_{tmp} \mid Int(GM(X)) \cap Int(GM(Y)) = \emptyset, \forall Y \in \mathcal{C}^2_{tmp}\}$
 5:  for each $X \in Q$ do
 6:      $\mathcal{C}(L_{diff}) \leftarrow \mathcal{C}(L_{diff}) \cup X$
 7:      $\mathcal{C}^1_{tmp} \leftarrow \mathcal{C}^1_{tmp} \backslash X$
 8:  while $\exists\{X, Y\}: Int(GM(X)) \cap Int(GM(Y)) \neq \emptyset \wedge X \in \mathcal{C}^1_{tmp} \wedge Y \in \mathcal{C}^2_{tmp}$ do
 9:      $\mathcal{C}_{XY} \leftarrow f_{DISJOINT}(X, Y)$
10:      $\mathcal{C}(L_{diff}) \leftarrow \mathcal{C}(L_{diff}) \cup \mathcal{S}_{X\backslash Y}(\mathcal{C}_{XY})$
11:      $\mathcal{C}^2_{tmp} \leftarrow \mathcal{C}^2_{tmp} \backslash Y \cup \mathcal{S}_{Y\backslash X}(\mathcal{C}_{XY})$
12:  if $\beta_{n-1}\left(GM\left(\mathcal{C}(L_{diff})\right)\right) > 0$ then
13:      $\mathcal{C}^2_{tmp} \leftarrow \mathcal{C}(L_2)$
14:      for each $Y \in \mathcal{C}^2_{tmp}$ do
15:          if $Int(GM(Y)) \cap Int(S_{out}(L_1)) = \emptyset$ then
16:              $A(Y) \leftarrow \emptyset$
17:              $\mathcal{C}(L_{diff}) \leftarrow \mathcal{C}(L_{diff}) \cup Y$
18:  if $\mathcal{C}(L_{diff}) = \mathcal{C}(L_1)$ then
19:      $G_{GM}(L_{diff}) \leftarrow G_{GM}(L_1)$
20:      $A\left(S_{out}(L_{diff})\right) \leftarrow A(S_{out}(L_1))$
21:      $A(L_{diff}) \leftarrow A(L_1)$
22:  else
23:      $G_{GM}(L_{diff}) \leftarrow GetDefaultEmbedding(G_{TP}(L_{diff}))$
24:  for each $\mathrm{B} \in \mathcal{B}(\mathrm{L}_{\mathrm{diff}})$ do
25:      $Q \leftarrow \{C \in \mathcal{B}(L_1) \mid Int(GM(B)) \cap Int(GM(C)) \neq \emptyset\}$
26:      for each $C \in Q$ do
27:          $A(B) \leftarrow A(B) \cup A(C)$
```

The *SubstractSpaceLayers* algorithm takes two n-dimensional space layers L_1 and L_2 as input. Similar to *MergeSpaceLayers* (cf. algorithm 3.67), the space cell complexes of L_1 and L_2 are copied to $\mathcal{C}^1_{tmp}$ respectively $\mathcal{C}^2_{tmp}$ and the output space layer L_{diff} is initialized with the minimal space layer L_{min} (cf. lines 1-3).

In a first step, all space cells in $\mathcal{C}^1_{tmp}$ which do not spatially overlap with any space cell in $\mathcal{C}^2_{tmp}$ are removed from $\mathcal{C}^1_{tmp}$ and carried to the resulting space cell complex $\mathcal{C}(L_{diff})$ of L_{diff} without modification (cf. lines 4-7). Second, if there exists a pair $\{X, Y\}$ of space cells so that $X \in \mathcal{C}^1_{tmp} \wedge Y \in \mathcal{C}^2_{tmp}$ and the intersection of the interiors of their primal space geometries is non-empty, then Y has to be subtracted from X. For this purpose, both space cells are passed as arguments to the map $f_{DISJOINT}$ which yields the space cell complex $\mathcal{C}_{XY} = f_{DISJOINT}(X, Y)$. Since the space cells contained in the set $\mathcal{S}_{X\backslash Y}(\mathcal{C}_{XY})$ occupy the space covered by $GM(X)$ but not by $GM(Y)$, they reflect the result of the subtraction and thus are taken to $\mathcal{C}(L_{diff})$. Only the space cells remaining in $\mathcal{S}_{Y\backslash X}(\mathcal{C}_{XY})$ can possibly overlap with further space cells in $\mathcal{C}^1_{tmp}$ and hence are kept in $\mathcal{C}^2_{tmp}$ whereas Y is removed from $\mathcal{C}^2_{tmp}$.

This second step of the algorithm is iteratively executed until there is no more pair of overlapping space cells in $\mathcal{C}^1_{tmp}$ and $\mathcal{C}^2_{tmp}$ (cf. lines 8-11).

In contrast to the merge operation, the space cell complex $\mathcal{C}(L_{diff})$ is not necessarily complete after the main iteration loop is left. The reason is that $GM(\mathcal{C}(L_{diff}))$ may have $(n-1)$-dimensional interior holes after the space cells on L_2 have been subtracted from L_1. These holes need to be explicitly filled by additional space cells in order to ensure that the universal solid respectively the universal face occupied by $S_{out}(L_{diff})$ is a connected space (cf. definition 3.13 and the related discussion in chapter 3.1.3). In line 12 of algorithm 3.73, the $(n-1)$th Betti number β_{n-1} of the topological space rendered by $GM(\mathcal{C}(L_{diff}))$ is used to decide whether there exist $(n-1)$-dimensional holes (cf. theorem A.94). The hole filling is then realized by copying those space cells from L_2 into $\mathcal{C}(L_{diff})$ which are completely enclosed by $GM(\mathcal{C}(L_{diff}))$ or, put differently, whose interiors in primal geometry space do not overlap with the interior of the outer space cell on L_1 (cf. lines 13-17). The attribute sets of these space cells is replaced with the empty set since their meaning changes when being added to $\mathcal{C}(L_{diff})$ (cf. line 16).

Having added the hole-filling space cells from L_2, both the primal space representation of L_{diff} and its intra-layer graph follow in a deterministic way. If the spatial configuration of space cells in $\mathcal{C}(L_{diff})$ is equal to that of $\mathcal{C}(L_1)$ after line 17, then none of the space cells on the input space layer L_1 has been affected by the subtraction. In this case, the spatial embedding of the intra-layer graph as well as the semantic attributes of $S_{out}(L_1)$ and L_1 itself are carried to L_{diff} (cf. lines 19-21). Otherwise, the spatial embedding of the intra-layer graph is determined by the *GetDefaultEmbedding* algorithm and no attributes are adopted for $S_{out}(L_{diff})$ and L_{diff} since both their spatial extent and their semantics differ from $S_{out}(L_1)$ and L_1 (cf. line 23). Finally, the semantic attributes from boundary cells in $\mathcal{B}(L_1)$ are transferred to overlapping ones in $\mathcal{B}(L_{diff})$ (cf. lines 24-27). Note that, unlike the merge operation, there is no transition of semantic attributes from elements of L_2 at any step of the algorithm.

The *SubstractSpaceLayers* algorithm is utilized in the following definition of the *difference operation* on two space layers.

Definition 3.74 (Difference operation on two space layers). Let $\mathbb{L}^n$ be the non-empty set of all n-dimensional space layers, with $2 \leq n \leq 3$. Then the map $\ominus : \mathbb{L}^n \times \mathbb{L}^n \rightarrow \mathbb{L}^n$ uniquely associates each pair of space layers $\{L_1, L_2\}$ in $\mathbb{L}^n$ with another space layer L_{diff} of $\mathbb{L}^n$ as defined by algorithm 3.73. It thus yields a closed binary operation on $\mathbb{L}^n$ called *difference operation* which is denoted by $L_1 \ominus L_2 = L_{diff}$.

The spatial layout of the output space layer L_{diff} obviously depends on whether L_2 is subtracted from L_1 or vice versa. Moreover, $(L_1 \ominus L_2) \ominus L_3 = L_1 \ominus (L_2 \ominus L_3)$ only holds in special cases, e.g. if the space cells in $\mathcal{C}(L_1)$, $\mathcal{C}(L_2)$, and $\mathcal{C}(L_3)$ do not overlap at all. It thus follows that the difference operation is neither commutative nor associative. For this reason, the pair $(\mathbb{L}^n, \ominus)$ only forms a more general type of algebraic structure than the Abelian monoid associated with the merge operation, namely a *magma* (cf. Jacobson 2009).

The minimal space layer L_{min} is a *right identity* element for the difference operation so that $L \ominus L_{min} = L_{diff} = L$ for all $L \in \mathbb{L}^n$. Since $\mathcal{C}(L_{min}) = \emptyset$, the resulting space cell complex $\mathcal{C}(L_{diff})$ contains all space cells from $\mathcal{C}(L)$ after line 7 of algorithm 3.73 and is not changed in any subsequent step. Consequently, $G_{GM}(L_{diff}) = G_{GM}(L)$ and all semantic attributes of L and its elements are preserved in L_{diff} (cf. line 18-21, 24-27). Therefore, $L_{diff} = L$. However, due to the lacking commutative property, L_{min} is not a *left identity* but it rather holds that $L_{min} \ominus L = L_{min}$ which also reflects the fact that space layers cannot model negative indoor space.

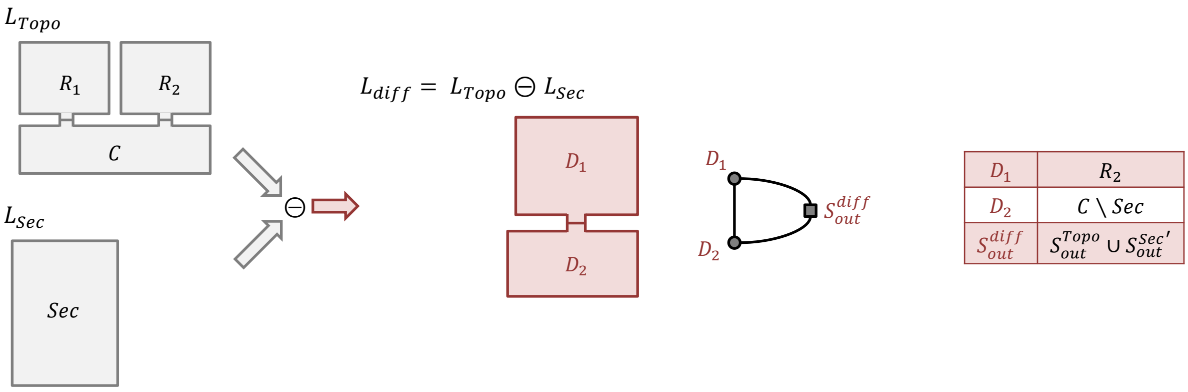
Conforming to an intuitive spatial understanding, subtracting a space layer from itself yields L_{min}. In algorithm 3.73, L_{diff} is first initialized with L_{min} which implies that $\mathcal{C}(L_{diff}) = \emptyset$. If we consider $L \ominus L$ then for every pair $\{X, Y\}$ of overlapping space cells it holds that $f_{DISJOINT}(X, Y) = \{X\} = \{Y\}$. But since $\mathcal{S}_{X \cap Y}(\mathcal{C}_{XY})$ is not considered for the difference operation (cf. lines 8-11), the space cell complex $\mathcal{C}(L_{diff})$ remains the empty set, and thus $L \ominus L = L_{diff} = L_{min}$. It immediately follows that in contrast to the merge operation every space layer $L \in \mathbb{L}^n$ is *self-invertible* in $(\mathbb{L}^n, \ominus)$. The difference action of L can therefore be cancelled by L itself. However, due to the lacking associativity property, this requires that L meets itself face to face in a difference operation. For example, the action of L in the difference $X \ominus L$ can be cancelled through $X \ominus (L \ominus L)$ because $X \ominus (L \ominus L) = X \ominus L_{min} = X$.

Note that this result does not hold for $(L \ominus L) \ominus X$ since the difference operation is not commutative and L_{min} is only a right identity.

A space layer possesses even more inverse elements in the magma $(\mathbb{L}^n, \ominus)$. Let $L_{right}^{-1} \in \mathbb{L}^n$ be a space layer with $GM(\mathcal{C}(L_{right}^{-1})) \supseteq GM(\mathcal{C}(L))$. Then $L \ominus L_{right}^{-1} = L_{min}$ also holds and L_{right}^{-1} is called a *right inverse* of L. Likewise, $L_{left}^{-1} \in \mathbb{L}^n$ solves $L_{left}^{-1} \ominus L = L_{min}$ if $GM(\mathcal{C}(L_{left}^{-1})) \subseteq GM(\mathcal{C}(L))$. Correspondingly, L_{left}^{-1} is a *left inverse* of L, and L is said to be both *left-invertible* and *right-invertible* regarding the difference operation. The expression $X \ominus (L \ominus L) = X$ can therefore be rewritten as $X \ominus \left(L \ominus L_{right}^{-1}\right) = X$ or, equivalently, as $X \ominus \left(L_{left}^{-1} \ominus L\right) = X$.

Similar to the merge operation, the difference operation may also operate on a sequence of space layers given by $L_{diff} = L_1 \ominus L_2 \ominus \dots \ominus L_k$, with $L_1, L_2, \dots, L_k \in \mathbb{L}^n$. Note that again the order of space layers is significant for the output space layer L_{diff}. Let L_i be a space layer in this sequence, with $1 < i < k$. In case $L_{i-1} \ominus L_i = L_{min}$, the computation of the difference can be aborted directly since the result $L_{diff} = L_{min}$ will not change for any space layer succeeding L_i.

Example 3.75. If the merge operation is understood to *integrate* two different conceptual views on indoor space (i.e., two different space layers) then the difference operation can be said to provide a view on indoor space *without* another view. For example, the difference operation may be used to create a view of topographic space without security zones. Consider the following figure 128 which is a slight modification of the indoor setting presented in figure 124 as the security layer L_{Sec} only contains one space cell Sec representing a single security zone.



D_1	R_2
D_2	$C \setminus Sec$
S_{out}^{diff}	$S_{out}^{Topo} \cup S_{out}^{Sec'}$

Figure 128: Applying the difference operation to the space layers from figure 124.

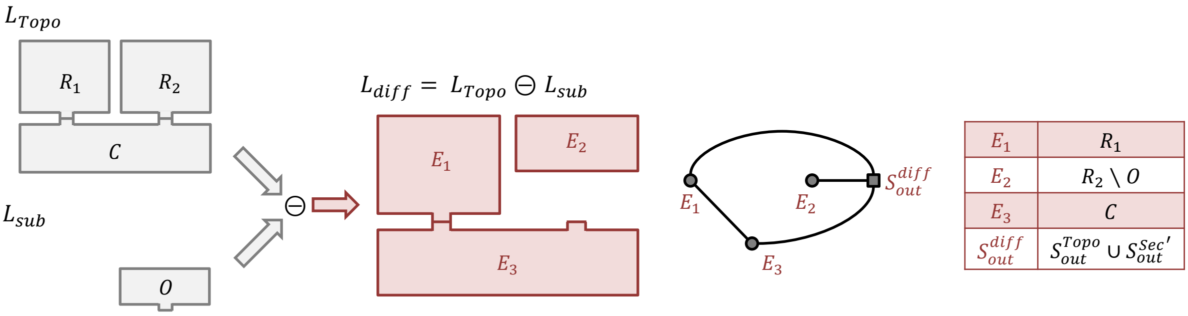
The difference $L_{Topo} \ominus L_{Sec}$ results in the two space cells D_1 and D_2 which represent all those topographic areas not requiring access permission to the security zone. The intra-layer graph of L_{diff} is hence feasible for planning routes for navigation users without corresponding security status. The difference operation thus allows for pruning the intra-layer graph of its first operand according to specific semantics provided by its second operand. But it not only yields a subset of the intra-layer graph, it also curtails the primal space representation accordingly. This distinguishes the difference operation from alternative graph pruning approaches as proposed in several indoor space models for indoor navigation. Moreover, these approaches are mostly restricted to removing graph elements based on specific values of associated attributes but not due to spatial facts (cf. chapter 2.2).

Since the space cell on L_{Sec} is spatially subtracted from the topographic space cells on L_{Topo}, it follows that the space cells D_1 and D_2 on the resulting difference layer L_{diff} are necessarily subspaces of the topographic space cells R_2 respectively C which renders L_{diff} a subspace layer of L_{Topo}. This fact can be generalized for arbitrary space layers as stated in the following proposition.

Proposition 3.76. Let $L_1, L_2 \in \mathbb{L}^n$ bet two space layers from the non-empty set $\mathbb{L}^n$ of all n-dimensional space layers. Then the difference operation $L_1 \ominus L_2$ yields a subspace layer of the first operand L_1, and thus $L_1 \ominus L_2 \leq L_1$.

In case L_1 is a left inverse of L_2, the inclusion relation $L_1 \ominus L_2 \leq L_2$ additionally holds since then $L_1 \ominus L_2 = L_{min}$ and obviously $L_{min} \leq L_2$. In all other cases, the subtraction result is not a subspace layer of the second operand L_2.

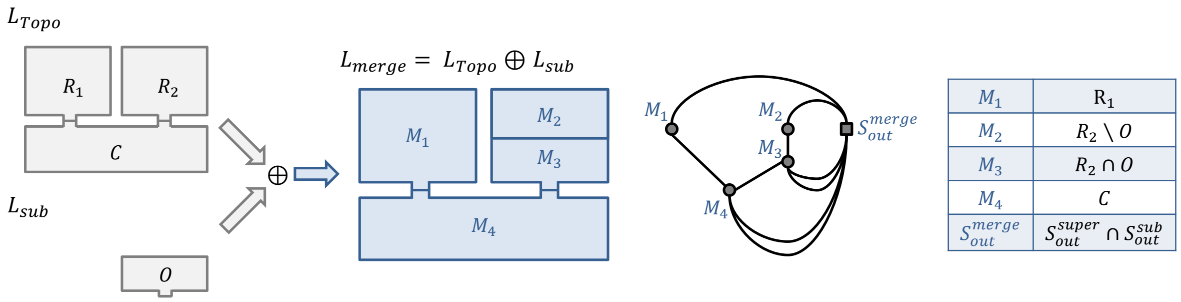
Assume the topographic space view is enriched with a subspace layer L_{sub} which contains a single space cell denoting a non-passable obstacle for wheelchair users within the room cell R_2. The difference operation $L_{Topo} \ominus L_{sub}$ then provides the navigable topographic space for this mode of locomotion as shown in the following figure. Without the obstacle space, the space cells E_2 and E_3 reflecting the navigable space within R_2 respectively C are not connected any more. The resulting intra-layer graph correspondingly lacks a dual edge linking their dual nodes. It is again a subgraph of the intra-layer graph of L_{Topo} and is suitable for planning paths for wheelchair users.[22]



E_1	R_1
E_2	$R_2 \setminus O$
E_3	C
S_{out}^{diff}	$S_{out}^{Topo} \cup S_{out}^{Sec'}$

Figure 129: Subtracting a subspace cell representing an obstacle from a topographic space layer.

In example 3.69, the merge operation has been used to integrate a superspace layer with its subspace layer. This approach is also feasible in this example and the outcome of $L_{Topo} \oplus L_{sub}$ is illustrated below.



M_1	R_1
M_2	$R_2 \setminus O$
M_3	$R_2 \cap O$
M_4	C
S_{out}^{merge}	$S_{out}^{super} \cap S_{out}^{sub}$

Figure 130: Applying the merge operation to the space layers from figure 129.

The space layers resulting from both $L_{Topo} \ominus L_{sub}$ and $L_{Topo} \oplus L_{sub}$ are themselves subspace layers of L_{Topo} (cf. proposition 3.70 and proposition 3.76). Moreover, it is immediate that $L_{Topo} \ominus L_{sub} \leq L_{Topo} \oplus L_{sub}$. The output of the merge operation is therefore spatially richer as it contains more space cells describing the indoor space than the difference of both space layers. And it is also richer regarding its semantics since it incorporates the attributes from either input space layer. The intra-layer graph of the merged space layer depicted on the right of figure 130 also supports path finding for wheelchair users given that the non-navigable obstacle space cell M_3 is tagged as such (e.g., through a corresponding semantic attribute for O as illustrated in example 3.63 and example 3.69). Instead, when applying the difference operation, the dual nodes reflecting non-navigable spaces are removed from the navigation graph. The navigation graph contains lesser elements and a path finding algorithm needs not evaluate additional attributes. Thus, both operations show complementary advantages. The MLSEM is flexible and expressive enough to derive and support both alternative space representations.

Finally, if a wheelchair user is not granted access to the security zone modelled on L_{Sec}, then figure 131 presents the remaining navigable space for this navigation context which is expressed through $L_{Topo} \ominus L_{sub} \ominus L_{Sec}$.

[22] The subtraction of obstacle subspaces has also been demonstrated in example 3.63 (cf. figure 112). Note that subtracting obstacle subspaces is however not always sufficient in order to derive navigable spaces (cf. related discussion in example 3.63).

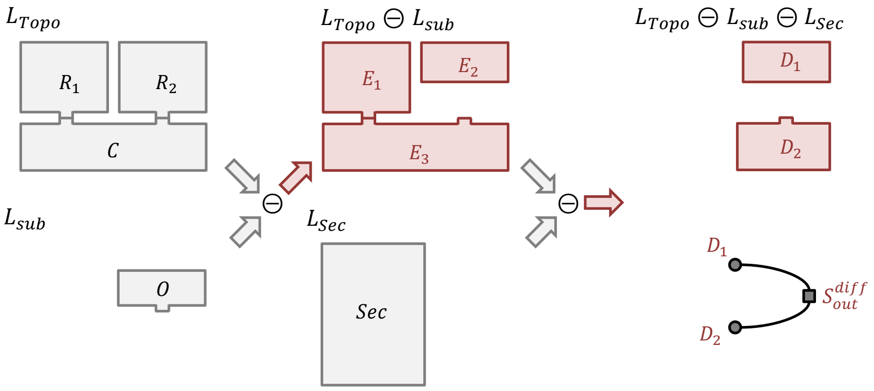


Figure 131: Applying the difference operation subsequently to derive $L_{Topo} \ominus L_{sub} \ominus L_{Sec}$.

3.5.3 Intersection Operation

In addition to the merge and difference of two space layers, a third operation for *intersecting* two space layers is presented in this chapter which complements the space layer algebra of the MLSEM. The result of the *intersection operation* is a space layer whose space cells only cover those partitions of indoor space which are occupied by a space cell from either input space layer. Conceptually, the intersection can be thought of as a *restriction* of two given views on indoor space to their common subspace.

The *IntersectSpaceLayers* algorithm for implementing the intersection operation is presented below. In fact, it is a subset of the *MergeSpaceLayers* algorithm realizing the merge operation (cf. algorithm 3.67). The most important difference between both is that non-overlapping space cells are not carried to the resulting space cell complex $\mathcal{C}(L_{intersect})$ of the output space layer $L_{intersect}$ within the main iteration loop. The corresponding lines 7-8 and 12-13 from algorithm 3.67 have been removed without replacement for the *IntersectSpaceLayers* algorithm. The reader is referred to chapter 3.5.1 for an illustration of the remaining steps.

Algorithm 3.77. IntersectSpaceLayers(L_1,L_2)

```
Input:   $L_1, L_2$, with $dim(L_1) = dim(L_2)$
Output:  $L_{intersect}$
  $\mathcal{C}_{tmp}^1 \leftarrow \mathcal{C}(L_1)$
  $\mathcal{C}_{tmp}^2 \leftarrow \mathcal{C}(L_2)$
  $L_{intersect} \leftarrow L_{min}$
  while $\mathcal{C}_{tmp}^1 \neq \emptyset \wedge \mathcal{C}_{tmp}^2 \neq \emptyset$ do
      $Q \leftarrow \{X \in \mathcal{C}_{tmp}^1 \mid Int(GM(X)) \cap Int(GM(Y)) = \emptyset, \forall Y \in \mathcal{C}_{tmp}^2\}$
      for each $X \in Q$ do
          $\mathcal{C}_{tmp}^1 \leftarrow \mathcal{C}_{tmp}^1 \backslash X$
      $Q \leftarrow \{Y \in \mathcal{C}_{tmp}^2 \mid Int(GM(Y)) \cap Int(GM(X)) = \emptyset, \forall X \in \mathcal{C}_{tmp}^1\}$
      for each $Y \in Q$ do
          $\mathcal{C}_{tmp}^2 \leftarrow \mathcal{C}_{tmp}^2 \backslash Y$
      if $\exists\{X, Y\}: Int(GM(X)) \cap Int(GM(Y)) \neq \emptyset \wedge X \in \mathcal{C}_{tmp}^1 \wedge Y \in \mathcal{C}_{tmp}^2$ then
          $\mathcal{C}_{XY} \leftarrow f_{DISJOINT}(X, Y)$
          $\mathcal{C}(L_{intersect}) \leftarrow \mathcal{C}(L_{intersect}) \cup \mathcal{S}_{X \cap Y}(\mathcal{C}_{XY})$
          $\mathcal{C}_{tmp}^1 \leftarrow \mathcal{C}_{tmp}^1 \backslash X \cup \mathcal{S}_{X \backslash Y}(\mathcal{C}_{XY})$
          $\mathcal{C}_{tmp}^2 \leftarrow \mathcal{C}_{tmp}^2 \backslash Y \cup \mathcal{S}_{Y \backslash X}(\mathcal{C}_{XY})$
  if $\mathcal{C}(L_{intersect}) = \mathcal{C}(L_1)$ then
      $G_{GM}(L_{intersect}) \leftarrow G_{GM}(L_1)$
  else
      $G_{GM}(L_{intersect}) \leftarrow GetDefaultEmbedding(G_{TP}(L_{intersect}))$
  for each $B \in \mathcal{B}(L_{intersect})$ do
      $Q \leftarrow \{C \in \mathcal{B}(L_1) \mid Int(GM(B)) \cap Int(GM(C)) \neq \emptyset\}$
      $Q \leftarrow Q \cup \{C \in \mathcal{B}(L_2) \mid Int(GM(B)) \cap Int(GM(C)) \neq \emptyset\}$
```

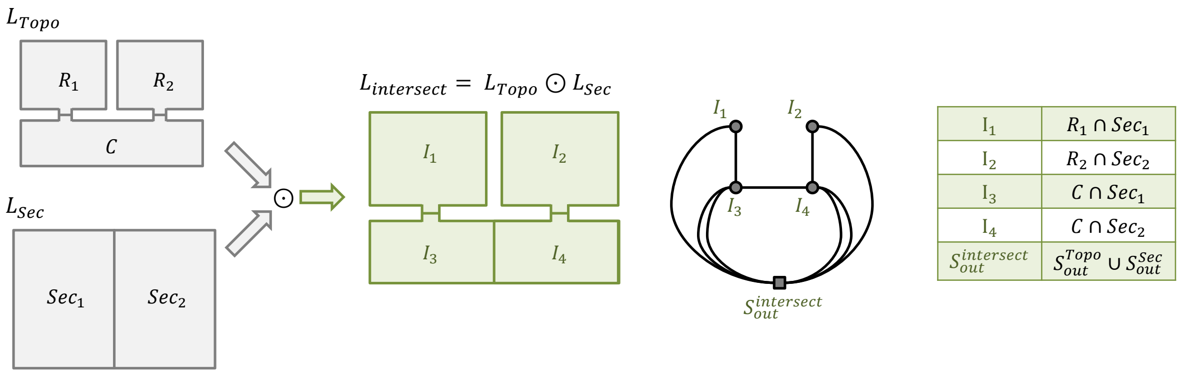
```
23:     for each $C \in Q$ do
24:         $A(B) \leftarrow A(B) \cup A(C)$
25: $A(S_{out}(L_{intersect})) \leftarrow A(S_{out}(L_1)) \cup A(S_{out}(L_2))$
26: $A(L_{intersect}) \leftarrow A(L_1) \cup A(L_2)$
```

Formally, the intersection operation can be stated as follows.

Definition 3.78 (Intersection operation on two space layers). Let $\mathbb{L}^n$ be the non-empty set of all n-dimensional space layers, with $2 \leq n \leq 3$. Then the map $\odot: \mathbb{L}^n \times \mathbb{L}^n \rightarrow \mathbb{L}^n$ uniquely associates each pair of space layers $\{L_1, L_2\}$ in $\mathbb{L}^n$ with another space layer $L_{intersect}$ of $\mathbb{L}^n$ as defined by algorithm 3.77. It thus yields a closed binary operation on $\mathbb{L}^n$ called *intersection operation* which is denoted by $L_1 \odot L_2 = L_{intersect}$.

The order of operands $L_1, L_2 \in \mathbb{L}^n$ for $\odot$ is not significant for $L_{intersect}$ since the resulting primal space representation $\mathcal{C}(L_{intersect})$ reflects the space occupied by both input layers in either case (cf. line 13 of algorithm 3.77). Moreover, $\odot$ is associative so that the order of intersection operations in a sequence of operands does not matter either. For the same reasons as for the merge operation, the intersection operation is idempotent and the minimal space layer L_{min} is both its unique identity element and self-invertible with all other space layers $L \in \mathbb{L}^n$ and $L \neq L_{min}$ being non-invertible regarding $\odot$. Due to the associative property and the existence of an identity element, the algebraic structure $(\mathbb{L}^n, \odot)$ can likewise be classified as an Abelian monoid.

Example 3.79. The difference between the merge and the intersection operation is demonstrated along the two space layers L_{Topo} and L_{Sec} as introduced in figure 124. Whereas example 3.71 discussed the result of merging both space layers (cf. figure 125), their intersection is presented below.



I_1	$R_1 \cap Sec_1$
I_2	$R_2 \cap Sec_2$
I_3	$C \cap Sec_1$
I_4	$C \cap Sec_2$
$S_{out}^{intersect}$	$S_{out}^{Topo} \cup S_{out}^{Sec}$

Figure 132: Applying the intersection operation to the space layers from figure 124.

The example clearly shows that the intersection operation also *integrates* both input space layers in a common view which resolves their ambiguity similar to the merge operation. But at the same time it *restricts* the extent of the resulting space layer to the indoor space inhabited by space cells from both layers. The intra-layer graph depicted on the right of figure 132 is a pruned subgraph of the intra-layer graph yielded by the merge operation since the shared partitions of indoor space involving either S_{out}^{Topo} or S_{out}^{Sec} are not captured by separate space cells. Every space cell on the intersection space layer carries the semantic attributes of its related space cells on L_{Topo} and L_{Sec}. The intra-layer graph thus facilitates path planning in the same way as illustrated for the merge operation in example 3.71 but within the restricted spatial scope of the intersection space layer.

In contrast to the merge operation, there is a clear spatial hierarchy between the operands L_1 and L_2 of the intersection operation and its outcome $L_{intersect}$. Since the primal space geometry of every space cell S on $L_{intersect}$ reflects the regularized Boolean intersection of two input space cells X and Y, with $X \in \mathcal{C}(L_1)$ and $Y \in \mathcal{C}(L_2)$ (cf. line 13 of algorithm 3.77), the space cell S must be spatially contained in both X and Y, and thus $S \leq X \wedge S \leq Y$. The inclusion relation between the space cells induces a spatial ordering of the involved space layers according to definition 3.51 and it follows that $L_{intersect} \leq L_1$ and $L_{intersect} \leq L_2$. This fact can be visually validated in figure 132 and is formally stated in the following proposition. The result of the intersection operation always renders a subspace layer of the result of the merge operation when applied to the same operands, and thus $L_1 \odot L_2 \leq L_1 \oplus L_2$.

Proposition 3.80. Let $L_1, L_2 \in \mathbb{L}^n$ bet two space layers from the non-empty set $\mathbb{L}^n$ of all n-dimensional space layers. Then the intersection operation $L_1 \odot L_2$ yields a subspace layer of both L_1 and L_2, and thus $L_1 \odot L_2 \leq L_1 \wedge L_1 \odot L_2 \leq L_2$.

Example 3.81. When omitting the space cell Sec_2 from the security space layer L_{Sec} in the previous example, the complementary results of the intersection and difference operation become obvious. The following figure 133 captures the result of $L_{Topo} \odot L_{Sec}$ for this modified setting.

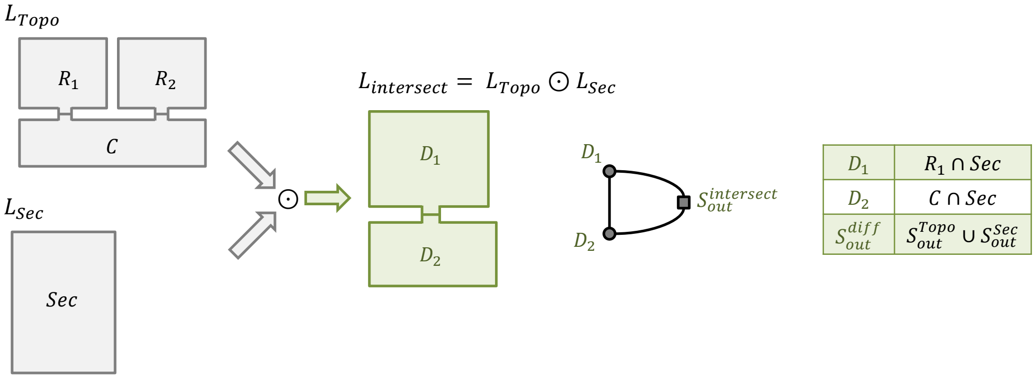


Figure 133: Applying the intersection operation to the space layers from figure 128 which demonstrates the complementary results of the difference and the intersection operations.

Whereas the difference operation $L_{Topo} \ominus L_{Sec}$ reveals the regions in topographic space that are outside the security zone Sec (cf. figure 128 of example 3.75), the intersection of both space layers returns the converse view on topographic space, namely all those regions that are covered by the security zone. The resulting intra-layer graph shown on the right of figure 133 hence enables the planning of paths being entirely inside the security zone without the need for evaluating additional semantic attributes as, for example, required in the previous example.

It follows that the intersection operation $L_1 \odot L_2$ returns the spatial complement of the difference operation $L_1 \ominus L_2$ with respect to L_1. The union of the primal space geometries of $L_1 \odot L_2$ and $L_1 \ominus L_2$ is therefore equal to the primal space geometry of L_1 itself which can be formally expressed through $GM(\mathcal{C}(L_1 \odot L_2)) \cup GM(\mathcal{C}(L_1 \ominus L_2)) = GM(\mathcal{C}(L_1))$. Likewise, due to the commutative property of $\odot$, the intersection result is also the spatial complement of $L_2 \ominus L_1$ with respect to L_2. One could therefore expect that the merge of the complements $L_1 \odot L_2$ and $L_1 \ominus L_2$ again yields L_1. This in fact holds for the primal space geometry. Precisely, let $L_{merge} = (L_1 \odot L_2) \oplus (L_1 \ominus L_2)$ be the result of the merge operation. Then the primal space geometry of the space cell complex of L_{merge} is equal to that of L_1, and thus $GM(\mathcal{C}(L_{merge})) = GM(\mathcal{C}(L_1))$. However, it is important to note that in general $L_{merge} \neq L_1$ given that $L_1 \neq L_2$. First, since $L_1 \odot L_2 \leq L_1$ and $L_1 \ominus L_2 \leq L_1$ (cf. proposition 3.80 and proposition 3.76), the merge operation results in a finer space partitioning than L_1. Second, $L_1 \odot L_2$ integrates the semantics of both L_1 and L_2. Thus, the outcome of $(L_1 \odot L_2) \oplus (L_1 \ominus L_2)$ is also semantically richer than L_1. It can be deduced that $(L_1 \odot L_2) \oplus (L_1 \ominus L_2) = L_1$ only holds in the trivial case $L_1 = L_2$ and additionally if $L_2 = L_{min}$ or if the space cells in $\mathcal{C}(L_2)$ do not spatially overlap with the space cells in $\mathcal{C}(L_1)$ so that $Int(GM(\mathcal{C}(L_2))) \cap Int(GM(\mathcal{C}(L_1))) = \emptyset$.

The fact that $(L_1 \odot L_2) \oplus (L_1 \ominus L_2) = L_1$ is only true in special cases illustrates that the results of the binary operations of the developed space layer algebra not necessarily agree with an intuitive spatial understanding as this understanding relates to the primal geometry space only. All binary operations, however, simultaneously operate on the geometric-topological representation of space cells and space layers in both primal and dual space as well as on their semantic dimension.

Example 3.82. The intersection operation can also be used to ensure that the space cells on two space layers are related by inclusion. Reconsider the indoor setting from figure 100 (cf. example 3.61) which consists of three topographic space layers $L_{Building}$, L_{Floors}, and $L_{Entities}$ reflecting the hierarchical structuring of a building. As discussed in example 3.61, an elevator shaft connecting the building floors is modelled as a single space cell on $L_{Entities}$ which overlaps with the space cells $Floor_1$ and $Floor_2$ on L_{Floors}. Consequently, $L_{Entities}$ is not a subspace layer of L_{Floors} and the hierarchy $\mathcal{H}(L_{Building}) = \{L_{Building}, L_{Floors}, L_{Entities}\}$ is non-strict.

In order to create a subspace relationship between L_{Floors} and $L_{Entities}$, the elevator space cell obviously needs to be cut along the boundaries of the space cells on L_{Floors}. This can be realized through the intersection operation $L_{intersect} = L_{Floors} \odot L_{Entities}$. The result is a subspace layer of both $L_{Entities}$ and L_{Floors} according to proposition 3.80, and the elevator space cell is split into two subspace cells $I_4 \leq Floor_1$ and $I_5 \leq Floor_2$ as can be seen in figure 134. Moreover, since $GM(\mathcal{C}(L_{Entities})) \subseteq GM(\mathcal{C}(L_{Floors}))$ in this example, it consequently follows that $GM(\mathcal{C}(L_{intersect})) = GM(\mathcal{C}(L_{Entities}))$. The space layer $L_{Entities}$ can thus be replaced with $L_{intersect}$ without losing spatial or semantic information. This renders $\mathcal{H}(L_{Building})$ strict and, in contrast to example 3.61, the space cell hierarchies $\mathcal{A}(Floor_1)$ and $\mathcal{B}(Floor_2)$ now include a representation of the elevator. The subspace cells I_4 and I_5 of the elevator form a *passage* between $\mathcal{A}(Floor_1)$ and $\mathcal{B}(Floor_2)$ on $L_{intersect}$ (cf. definition 3.60).

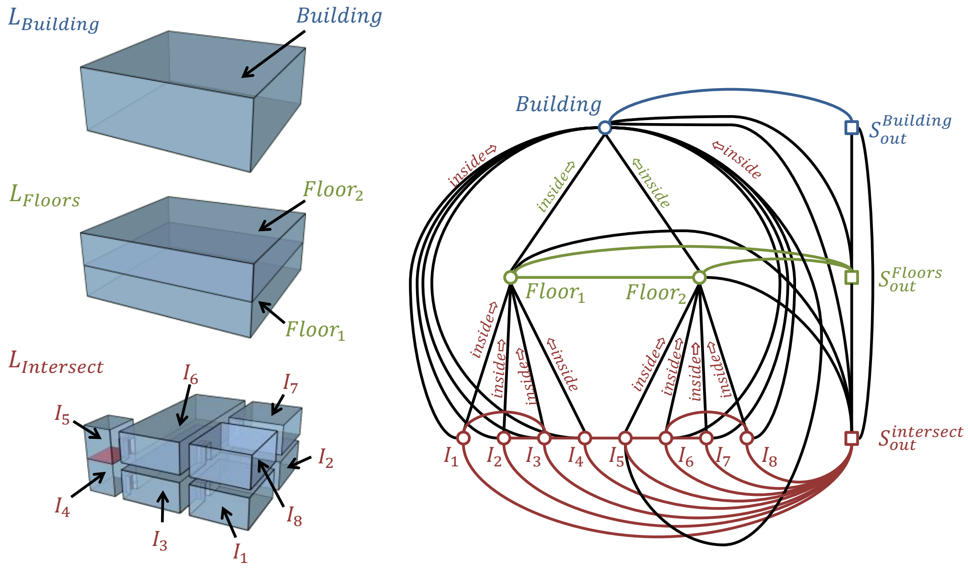


Figure 134: Using the intersection operation to create subspace relationships between the space layers from figure 100.

In the more general case that for two given space layers $L_1, L_2 \in \mathbb{L}^n$ the primal space geometry of L_1 is not a subset of the primal space geometry of L_2 or vice versa, subsequent merge operations are required. Precisely, the two merge space layers $L_{merge}^1 = L_1 \odot L_2 \oplus L_1$ and $L_{merge}^2 = L_1 \odot L_2 \oplus L_2$ need to be derived from the original space layers L_1 and L_2. The intersection $L_1 \odot L_2$ first realizes the inclusion relations between the space cells on both input space layers inhabiting the common subspace. The following merge operation with L_1 respectively L_2 then guarantees that the non-shared partitions of indoor space are not lost. The space cells on the resulting space layers L_{merge}^1 and L_{merge}^2 are therefore comparable but note that this does not imply an inclusion relation between L_{merge}^1 and L_{merge}^2 themselves (cf. proposition 3.70).

3.6 Supporting Different and Multiple Contexts of Navigation

The MLSEM provides the means to describe the indoor space from complementary conceptual views and to combine arbitrary conceptualizations of space within an integrated model. In this chapter, the support for different and multiple contexts of navigation based on this space representation is addressed and illustrated. The navigation context has been defined in chapter 1.2 as the setting and interrelated conditions in which the navigation takes place (cf. Abowd et al. 1999, Mokbel & Levandoski 2009). It can be differentiated into the *user context* (e.g., mode of locomotion, preferences, access rights, capabilities of the mobile device, etc.) which may differ for different navigation users and into the *environmental context* (e.g., physical built-up space, localization technologies, logical navigation constraints such as security zones or temporal access constraints) which is independent from individual users and thus provides a global navigation framework (cf. chapter 1.2).

If the user-centric or environmental contextual information can be expressed spatially (e.g., through assigning a spatial extent or by deriving a spatial partitioning schema), it can be reflected on or related to one or more space layers of the MLSEM. For example, and as demonstrated in the previous chapters, logical spatial facts such as security zones or disaster areas can be mapped onto separate space layers. Likewise, navigable and non-navigable spaces associated with a specific mode of locomotion can be captured by subspace layers of the topographic space, and localization technologies can be described through separate sensor space layers. The individual indoor setting required for navigating a given person or object then results from *selecting* those space layers that match the user context. The user context can hence be understood as *selection criterion* on a given space layer complex which yields a user-specific subset of the modelled space layers (cf. Becker et al. 2009b, Nagel et al. 2010). Moreover, the selection may involve binary operations on space layers in order to *integrate*, *subtract*, or *restrict* specific views on indoor space to the user requirements or a given navigation use case or scenario.

Example 3.83. The general idea of the context-dependent selection process is exemplified in the following along the 2-dimensional space layer complex $\mathcal{L}$ illustrated in figure 135. The built-up space is described through a topographic space layer L_{Topo} and its subspace layer L_{Sub} which provides a single obstacle space being non-navigable for wheelchair users. The two logical space layers L_{Sec_low} and L_{Sec_high} denote the spatial extent of two disjoint security zones associated with a low and a high security level. Finally, the indoor environment is equipped with two positioning systems based on Wi-Fi respectively RFID whose sensor coverage is captured by the sensor space layers L_{Wi-Fi} and L_{RFID}.[23] The separate space layers have already been discussed in examples of previous chapters. Their respective intra-layer graphs are shown on the right of figure 135, whereas the overall multilayered graph structure $MLG(\mathcal{L})$ is omitted for readability.

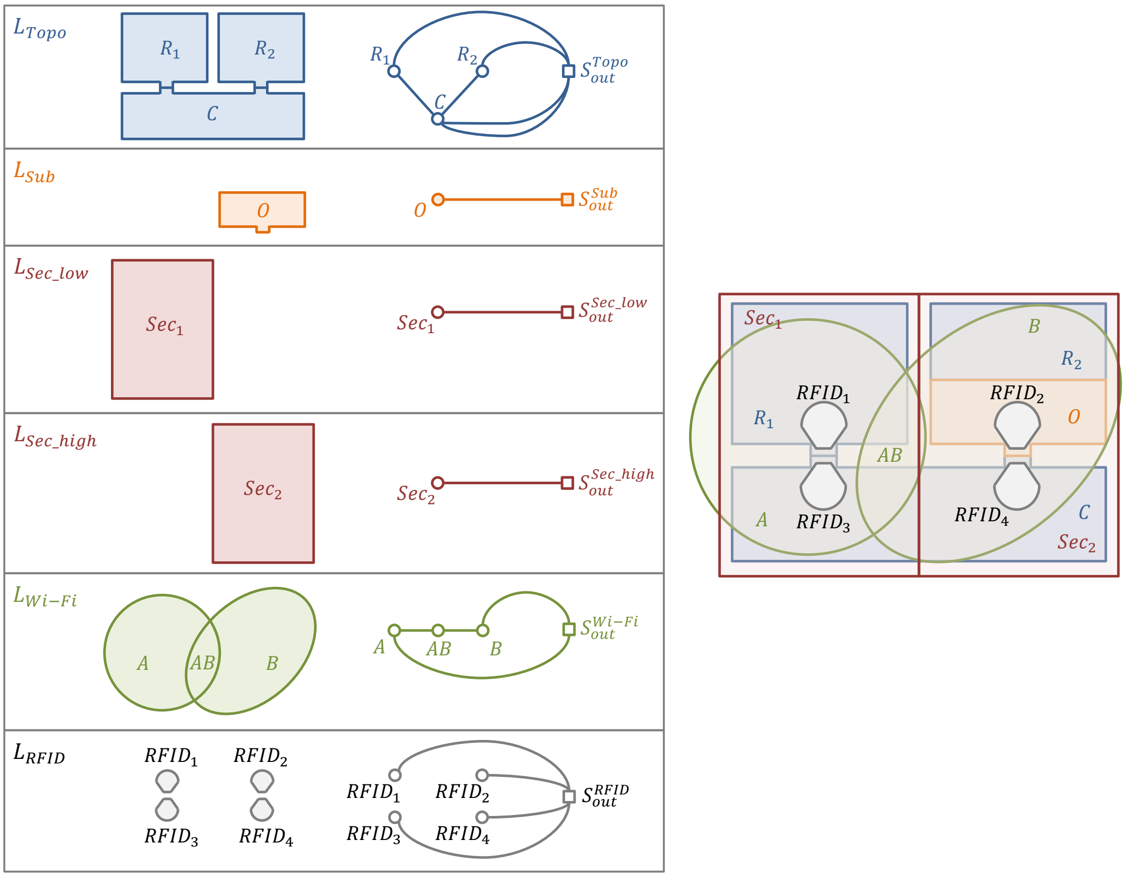


Figure 135: Several 2-dimensional space layers describing different views on indoor space (left) and their spatial overlay (right).

Now, consider a pedestrian navigation user A who has access rights to the low security zone only, and who carries a mobile device that is capable of receiving RFID but not Wi-Fi signals. The context-dependent selection of space layers from $\mathcal{L}$ for navigating A then obviously results in the user-specific space layer complex $\mathcal{L}_A = \{L_{Topo}, L_{Sec\_low}, L_{RFID}\}$ which is a subcomplex of $\mathcal{L}$. The remaining space layers L_{Sub}, L_{Sec_high}, and L_{Wi-Fi} in $\mathcal{L}$ do not match the user context and thus need not be considered. The multilayered graph $MLG(\mathcal{L}_A)$ combining the selected space layers in the subcomplex $\mathcal{L}_A$ renders a subgraph of $MLG(\mathcal{L})$. Since the spatial configuration of space cells on each space layer in $\mathcal{L}_A$ is unaffected from the selection process, their intra-layer graphs remain the same

[23] Note that the sensor space layer L_{Wi-Fi} integrates the coverage areas of the separate Wi-Fi transmitters in a single space view. In contrast, and as demonstrated in example 3.72, the Wi-Fi transmitters could be represented on separate sensor space layers which then might be merged upon selection. The same situation can be observed for the RFID sensors on L_{RFID}.

which also holds for the topological inter-layer relationships between their space cells. Therefore, the node set $V(MLG(\mathcal{L}_A))$ is simply a subset of $V(MLG(\mathcal{L}))$ and only contains the dual nodes from the selected space layers. The edge set $E(MLG(\mathcal{L}_A))$ then results from pruning all those edges in $E(MLG(\mathcal{L}))$ which are incident to dual nodes of space cells on non-selected space layers. The derivation of $MLG(\mathcal{L}_A)$ consequently does not involve any further spatial operations in primal space. The outcome of the selection process for the navigation user A is shown in the following figure.

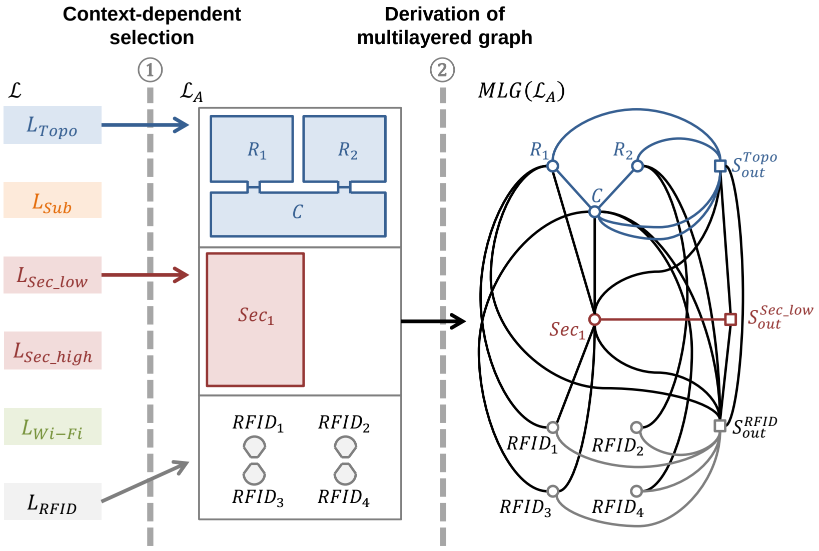
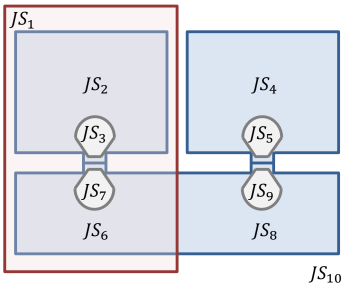


Figure 136: Context-dependent selection of space layers (left) and corresponding multilayered graph (right).

In order to facilitate the localization and tracking of A within the user context, the joint state space $\mathcal{JS}(\mathcal{L}_A)$ for the space layer complex $\mathcal{L}_A$ needs to be established which then can be used to build a user-specific finite-joint-state machine $FJSM(\mathcal{L}_A)$ (cf. definition 3.42 and definition 3.46 in chapter 3.3). Due to $MLG(\mathcal{L}_A) \subset MLG(\mathcal{L})$, the joint states in $\mathcal{JS}(\mathcal{L}_A)$ are not independent from the joint states in $\mathcal{JS}(\mathcal{L})$ but rather can be inferred from the latter in a deterministic way. In a first step, the dual nodes associated with space cells from non-selected space layers in $\mathcal{L}$ are removed from the clique $V(JS)$ of every joint state $JS \in \mathcal{JS}(\mathcal{L})$ yielding a new joint state JS' with $V(JS') \subset V(JS)$ and $GM(JS') = GM(JS)$. Since $GM(JS')$ is obviously non-empty, JS' is a candidate for $\mathcal{JS}(\mathcal{L}_A)$. The removal of dual nodes may however result in two or more joint states $\{JS'_1, \dots, JS'_k\}$ having identical cliques with $\forall i, j \in \{1, \dots, k\}: V(JS'_i) = V(JS'_j)$, which violates the uniqueness condition for joint states within the same joint state space. Therefore, in a second step, duplicate joint state candidates need to be substituted by a single joint state JS'' that is assigned the identical clique and whose primal space geometry $GM(JS'')$ follows from the union of the primal space geometries of the duplicate joint states, so that $V(JS'') = V(JS'_1)$ and $GM(JS'') = \bigcup_{i=1}^{k} GM(JS'_i)$. Due to the union operation, the uncertainty region reflected by $GM(JS'')$ necessarily grows which conforms to the fact that the uncertainty about the absolute position of a navigating person or object increases with lesser overlaps between space cells and thus with lesser space layers (cf. chapter 3.3).

The derivation of the user-specific joint state space is illustrated in the following example. In order to reduce the complexity of the example and to visually support both steps, a subcomplex $\mathcal{L}_B = \{L_{Topo}, L_{RFID}\}$ of the space layer complex $\mathcal{L}_A$ is assumed whose joint state space $\mathcal{JS}(\mathcal{L}_B)$ is deduced from $\mathcal{JS}(\mathcal{L}_A)$. Figure 137 shows the overlay of the space layers L_{Topo}, L_{Sec_low}, and L_{RFID} in $\mathcal{L}_A$ in primal geometry space as well as the spatial extent of the corresponding ten joint states $JS_1, \dots, JS_{10} \in \mathcal{JS}(\mathcal{L}_A)$ which are additionally presented in tabular form on the right of figure 137. Each row of the table corresponds to a separate joint state $JS_i \in \mathcal{JS}(\mathcal{L}_A)$ and enumerates the dual nodes of the space cells participating in $V(JS_i)$ ordered by the space layers contained in $\mathcal{L}_A$. Moreover, the last column of each row depicts the Boolean expression yielding the primal space geometry $GM(JS_i)$.

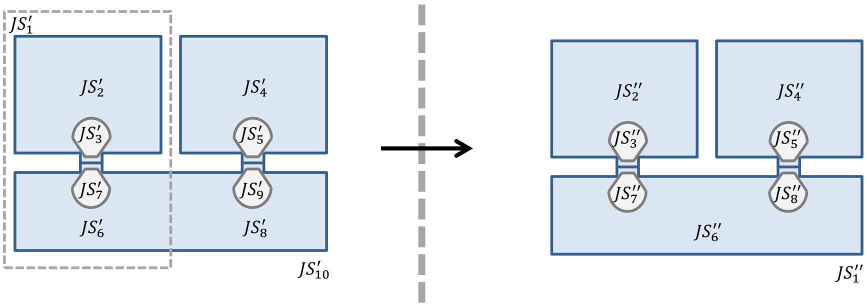


$\mathcal{JS}(\mathcal{L}_A)$

i	$V(JS_i)$			$GM(JS_i)$
	L_{Topo}	L_{Sec_low}	L_{RFID}	
1	$v_{TP}(S_{out}^{Topo})$	$v_{TP}(Sec_1)$	$v_{TP}(S_{out}^{RFID})$	$GM(S_{out}^{Topo}) \cap GM(Sec_1) \cap GM(S_{out}^{RFID})$
2	$v_{TP}(R_1)$	$v_{TP}(Sec_1)$	$v_{TP}(S_{out}^{RFID})$	$GM(R_1) \cap GM(Sec_1) \cap GM(S_{out}^{RFID})$
3	$v_{TP}(R_1)$	$v_{TP}(Sec_1)$	$v_{TP}(RFID_1)$	$GM(R_1) \cap GM(Sec_1) \cap GM(RFID_1)$
4	$v_{TP}(R_2)$	$v_{TP}(S_{out}^{Sec\_low})$	$v_{TP}(S_{out}^{RFID})$	$GM(R_2) \cap GM(S_{out}^{Sec\_low}) \cap GM(S_{out}^{RFID})$
5	$v_{TP}(R_2)$	$v_{TP}(S_{out}^{Sec\_low})$	$v_{TP}(RFID_2)$	$GM(R_2) \cap GM(S_{out}^{Sec\_low}) \cap GM(RFID_2)$
6	$v_{TP}(C)$	$v_{TP}(Sec_1)$	$v_{TP}(S_{out}^{RFID})$	$GM(C\ ) \cap GM(Sec_1) \cap GM(S_{out}^{RFID})$
7	$v_{TP}(C)$	$v_{TP}(Sec_1)$	$v_{TP}(RFID_3)$	$GM(C\ ) \cap GM(Sec_1) \cap GM(RFID_3)$
8	$v_{TP}(C)$	$v_{TP}(S_{out}^{Sec\_low})$	$v_{TP}(S_{out}^{RFID})$	$GM(C\ ) \cap GM(S_{out}^{Sec\_low}) \cap GM(S_{out}^{RFID})$
9	$v_{TP}(C)$	$v_{TP}(S_{out}^{Sec\_low})$	$v_{TP}(RFID_4)$	$GM(C\ ) \cap GM(S_{out}^{Sec\_low}) \cap GM(RFID_4)$
10	$v_{TP}(S_{out}^{Topo})$	$v_{TP}(S_{out}^{Sec\_low})$	$v_{TP}(S_{out}^{RFID})$	$GM(S_{out}^{Topo}) \cap GM(S_{out}^{Sec\_low}) \cap GM(S_{out}^{RFID})$

Figure 137: The spatial extent of the joint states between the space layers L_{Topo}, L_{Sec_low}, and L_{RFID} (left) and their presentation in tabular form (right).

Since the security space layer L_{Sec_low} is not part of the navigation context represented by $\mathcal{L}_B$, the dual nodes of its space cells Sec_1 and $S_{out}^{Sec\_low}$ have to be deleted from the joint states in $\mathcal{JS}(\mathcal{L}_A)$ to obtain the joint state candidates for $\mathcal{JS}(\mathcal{L}_B)$. This can be realized by striking out the respective table column as shown on the left of figure 138. In the resulting projection, the pairs $\{JS'_1, JS'_{10}\}$ and $\{JS'_6, JS'_8\}$ represent identical joint states due to $V(JS'_1) = V(JS'_{10}) = \{v_{TP}(S_{out}^{Topo}), v_{TP}(S_{out}^{RFID})\}$ and $V(JS'_6) = V(JS'_8) = \{v_{TP}(C), v_{TP}(S_{out}^{RFID})\}$. Whereas the remaining candidates are unique and thus can be carried to $\mathcal{JS}(\mathcal{L}_B)$ without changes, the duplicate candidates need to be merged into unique joint states as described above. The final joint state space $\mathcal{JS}(\mathcal{L}_B)$ then contains eight unique joint states which are depicted on the right of in figure 138.



Joint state candidates for $\mathcal{JS}(\mathcal{L}_B)$

i	$V(JS'_i)$		$GM(JS'_i)$
	L_{Topo}	L_{RFID}	
1	$v_{TP}(S_{out}^{Topo})$	$v_{TP}(S_{out}^{RFID})$	$GM(JS_1)$
2	$v_{TP}(R_1)$	$v_{TP}(S_{out}^{RFID})$	$GM(JS_2)$
3	$v_{TP}(R_1)$	$v_{TP}(RFID_1)$	$GM(JS_3)$
4	$v_{TP}(R_2)$	$v_{TP}(S_{out}^{RFID})$	$GM(JS_4)$
5	$v_{TP}(R_2)$	$v_{TP}(RFID_2)$	$GM(JS_5)$
6	$v_{TP}(C)$	$v_{TP}(S_{out}^{RFID})$	$GM(JS_6)$
7	$v_{TP}(C)$	$v_{TP}(RFID_3)$	$GM(JS_7)$
8	$v_{TP}(C)$	$v_{TP}(S_{out}^{RFID})$	$GM(JS_8)$
9	$v_{TP}(C)$	$v_{TP}(RFID_4)$	$GM(JS_9)$
10	$v_{TP}(S_{out}^{Topo})$	$v_{TP}(S_{out}^{RFID})$	$GM(JS_{10})$

$\mathcal{JS}(\mathcal{L}_B)$

i	$V(JS''_i)$		$GM(JS''_i)$
	L_{Topo}	L_{RFID}	
1	$v_{TP}(S_{out}^{Topo})$	$v_{TP}(S_{out}^{RFID})$	$GM(JS'_1) \cup GM(JS'_7)$
2	$v_{TP}(R_1)$	$v_{TP}(S_{out}^{RFID})$	$GM(JS'_2)$
3	$v_{TP}(R_1)$	$v_{TP}(RFID_1)$	$GM(JS'_3)$
4	$v_{TP}(R_2)$	$v_{TP}(S_{out}^{RFID})$	$GM(JS'_4)$
5	$v_{TP}(R_2)$	$v_{TP}(RFID_2)$	$GM(JS'_5)$
6	$v_{TP}(C)$	$v_{TP}(S_{out}^{RFID})$	$GM(JS'_6) \cup GM(JS'_8)$
7	$v_{TP}(C)$	$v_{TP}(RFID_3)$	$GM(JS'_7)$
8	$v_{TP}(C)$	$v_{TP}(RFID_4)$	$GM(JS'_9)$

Figure 138: Derivation of the joint states for a subset of the space layers from figure 137.

Following the same procedure, the joint space state $\mathcal{JS}(\mathcal{L}_A)$ can be derived from $\mathcal{JS}(\mathcal{L})$.

The user-dependent subsetting of a given space layer complex may additionally require binary operations on the selected space layers (e.g., to integrate subspace layers with their superspace layer or to map all space layers affecting the navigable and non-navigable spaces for a given user context to a single space layer and intra-layer graph for path planning). Since the navigation user A of the above example is only allowed to move inside the security zone, the view on topographic space could be restricted accordingly by intersecting L_{Sec_low} and L_{Topo} which yields the navigable spaces for A on a single space layer.[24] The space layer complex $\mathcal{L}_A$ is then given by the set $\mathcal{L}_A = \{L_{Topo} \odot L_{Sec\_low}, L_{RFID}\}$ instead.

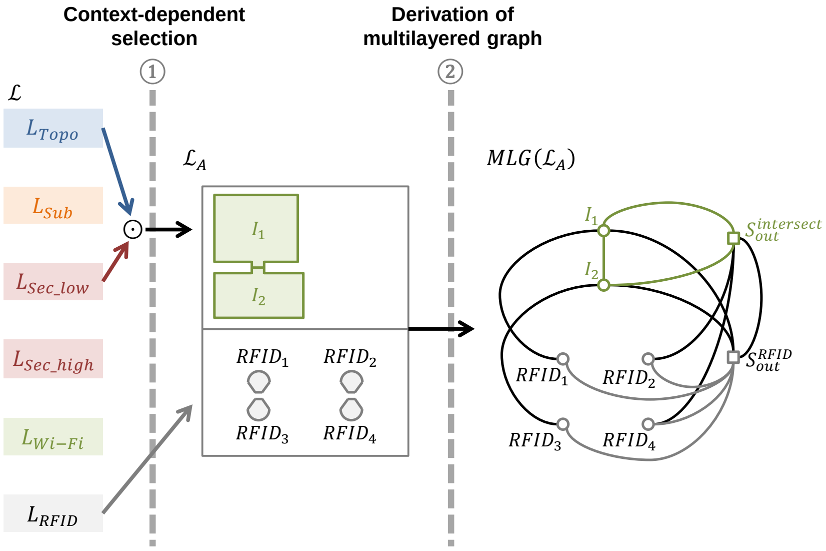


Figure 139: Example of context-dependent selection of space layers involving binary operations on the selected space layers (left) and corresponding multilayered graph (right).

It is immediate that the multilayered graph $MLG(\mathcal{L}_A)$ shown on the right of figure 139 is not a subset of $MLG(\mathcal{L})$ anymore because the space partitioning on $L_{Topo} \odot L_{Sec\_low}$ differs from L_{Topo} as well as L_{Sec_low}. Thus, the inter-layer edges between the dual nodes of $L_{Topo} \odot L_{Sec\_low}$ and L_{RFID} as well as the joint states between both space layers cannot be derived according to the above procedure but have to be computed anew. This generally holds if the user-specific selection contains a space layer that results from binary operations and hence is not identical to the space layers in the original space layer complex. For example, consider a wheelchair user C who may travel between the security zones (e.g., after passing a security gate) and who can be localized using both Wi-Fi and RFID sensors. Then a possible selection of space layers matching the user context of C is given by $\mathcal{L}_C = \{(L_{Topo} \ominus L_{Sub}) \odot (L_{Sec\_low} \oplus L_{Sec\_high}), L_{Wi-Fi}, L_{RFID}\}$.[25] In this scenario, only the intra-layer edges and joint states between the sensor space layers L_{Wi-Fi} and L_{RFID} can be deduced from $MLG(\mathcal{L})$ and $\mathcal{JS}(\mathcal{L})$ since both space layers are carried unchanged from $\mathcal{L}$ to $\mathcal{L}_C$.

[24] The merge operation $L_{Topo} \oplus L_{Sec\_low}$ is also feasible in this example to reflect the navigable spaces for the user A. In fact, due to $GM(\mathcal{C}(L_{Sec\_high})) \cap^* GM(\mathcal{C}(L_{Sec\_low})) = \emptyset$ and $GM(\mathcal{C}(L_{Sec\_low})) \cup GM(\mathcal{C}(L_{Sec\_high})) \supset GM(\mathcal{C}(L_{Topo}))$, even $L_{Topo} \ominus L_{Sec\_high}$ yields the navigable spaces for A.

[25] Observe that the subtraction $L_{Topo} \ominus L_{Sub}$ removes the non-passable obstacles for wheelchair users from the view on topographic space, whereas the subsequent intersection with $L_{Sec\_low} \oplus L_{Sec\_high}$ integrates the information about security zones into this view.

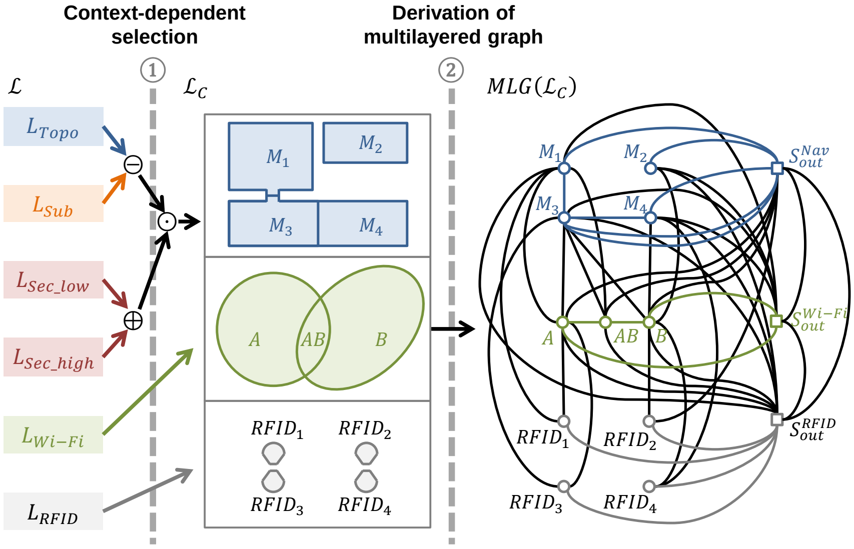

Context-dependent selection
Derivation of multilayered graph
① ②
$\mathcal{L}$ $\mathcal{L}_C$ $MLG(\mathcal{L}_C)$
L_{Topo} L_{Sub} L_{Sec_low} L_{Sec_high} L_{Wi-Fi} L_{RFID}
M_1 M_2 M_3 M_4
A AB B
$RFID_1$ $RFID_2$ $RFID_3$ $RFID_4$
S^{Nav}_{out} S^{Wi-Fi}_{out} S^{RFID}_{out}

Figure 140: Example of a context-dependent selection of space layers involving binary operations on the selected space layers (left) and corresponding multilayered graph (right).

Let L_{Nav} denote the navigable space layer for C resulting from $(L_{Topo} \ominus L_{Sub}) \odot (L_{Sec\_low} \oplus L_{Sec\_high})$. Both the merge and the intersection operation in this expression ensure that semantic attributes modelled for the security space cells Sec_1 and Sec_2 are available for path planning from the space cells on the resulting space layer L_{Nav} (cf. example 3.71 and example 3.81). Alternatively, L_{Sec_low} and L_{Sec_high} may themselves be added to $\mathcal{L}_C$ yielding the user-specific selection $\mathcal{L}_C = \{L_{Nav}, L_{Sec\_low}, L_{Sec\_high}, L_{Wi-Fi}, L_{RFID}\}$. Since, according to proposition 3.80, L_{Nav} is a subspace layer of $L_{Sec\_low} \oplus L_{Sec\_high}$, it follows that the space cells on L_{Nav} are related by inclusion to those on L_{Sec_low} and L_{Sec_high}. The two space cell hierarchies $\mathcal{H}(Sec_1) = \{Sec_1, M_1, M_3\}$ and $\mathcal{H}(Sec_2) = \{Sec_2, M_2, M_4\}$ then equivalently provide the information which space cell on L_{Nav} is associated with which security zone (cf. the related discussion in example 3.61). So, in this case, additional semantic attributes for Sec_1 and Sec_2 are not required, but instead a path search algorithm needs to be capable of evaluating the resulting hierarchical graph structure spanned between L_{Nav}, L_{Sec_low}, and L_{Sec_high} as shown in the following figure 141.

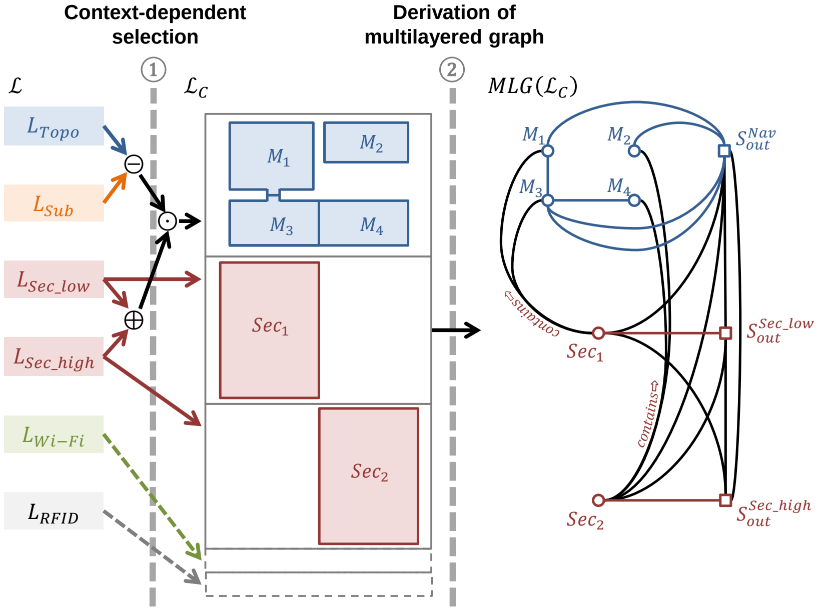

Context-dependent selection
Derivation of multilayered graph
① ②
$\mathcal{L}$ $\mathcal{L}_C$ $MLG(\mathcal{L}_C)$
L_{Topo} L_{Sub} L_{Sec_low} L_{Sec_high} L_{Wi-Fi} L_{RFID}
M_1 M_2 M_3 M_4
Sec_1 Sec_2
S^{Nav}_{out} $S^{Sec\_low}_{out}$ $S^{Sec\_high}_{out}$
⇐contains
contains⇒

Figure 141: Example of a context-dependent selection of space layers involving binary operations on the selected space layers (left) and corresponding multilayered graph (right).

The last example nicely illustrates the 2-dimensional navigation space spanned by each multilayered graph. Whereas the inter-layer edges linking dual nodes from different space layers enable localization, tracking, and reasoning about hierarchical containment relationships, the orthogonally aligned intra-layer edges facilitate path planning. The selection and algebraic combination of space layers for navigating a specific user hence needs to consider both dimensions appropriately.

The number of space layers contained in a space layer complex determines the number of complementary views on a given indoor space. Obviously, the more space layers are provided the richer is the space description and the more navigation contexts can be supported through subsets and algebraic combinations of the space layers as demonstrated in the above examples. The MLSEM thus enables the ad-hoc selection and simultaneous consideration of those spatial and logical facts being relevant for individual navigation users. As discussed in chapter 2.2, most existing proposals to indoor navigation reduce the combinatorial complexity a priori through restricting the indoor space representation to a rigid setting that only supports a single or a limited number of predefined navigation contexts. In contrast, the multilayered modelling approach of the MLSEM allows adding further space layers to a given indoor setting at any time in order to capture even previously unknown navigation configurations without interfering with existing space layers. Since the number of space layers in a space layer complex is unbounded, so is the number of potentially supported navigation contexts. This flexible support for arbitrary navigation contexts has been postulated as essential requirement for indoor navigation systems in chapter 2.4 and is hence answered by the MLSEM.

The possibility to add and also remove space layers from a given space layer complex also facilitates to react on dynamic changes in the indoor environment. Suppose for the simple indoor setting introduced in figure 135 that a fire incident affects the room R_1 and thus makes it non-accessible for all navigation users. A corresponding subspace layer L_{Fire} of the topographic space layer L_{Topo} could be added to the space layer complex $\mathcal{L}$ whose space cells reflect the spatial extent of the disaster areas. Of course, this space layer would have to be updated in short intervals in a real world scenario in order to capture the expansion as well as the stagnation of the fire incident (e.g., the areas could be reported by fire detectors monitoring parts of the topographic space).[26] In order to consider the disaster areas in navigation contexts, it could, for example, be subtracted from the context-dependent view on navigable and non-navigable spaces of a specific navigation user. For instance, reconsider navigation user A with the individual space layer complex $\mathcal{L}_A = \{L_{Topo} \odot L_{Sec\_low}, L_{RFID}\}$ (cf. figure 139). The update for A could then be expressed as $L_{Topo} \odot L_{Sec\_low} \ominus L_{Fire}$ which deletes the dual node representing R_1 from the user-specific navigation graph and thus avoids paths involving R_1. Moreover, if we assume that in case of a fire emergency the movement restrictions enforced by security zones are overruled to allow users to get to the nearest exit, then L_{Sec_low} could be additionally removed from $\mathcal{L}_A$. Thus, $\mathcal{L}_A$ would finally be given by the set $\{L_{Topo} \ominus L_{Fire}, L_{RFID}\}$.

[26] Likewise, the monitored areas themselves could be represented as space cells on a logical space layer similar to the coverage areas on a sensor space layer. Whether a specific area is navigable or not could then be expressed through semantic attributes associated with the space cells.

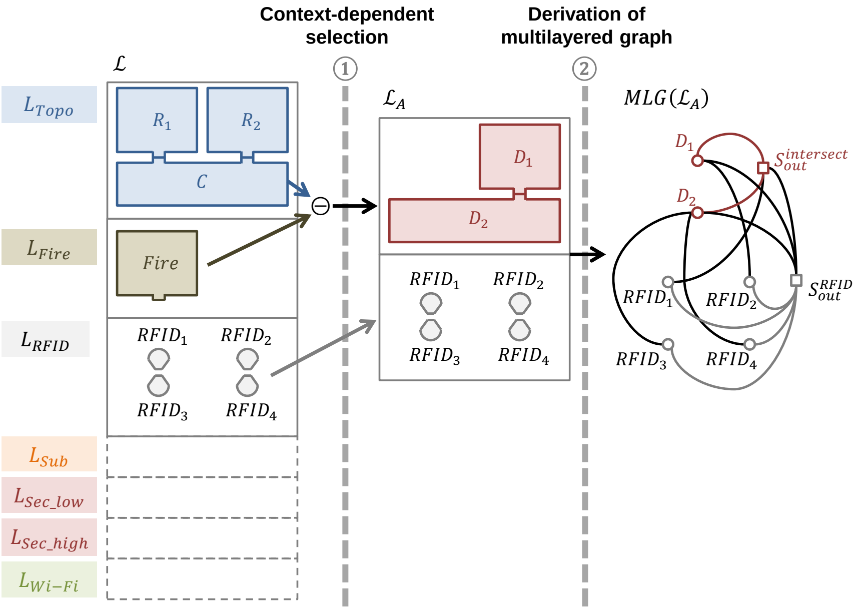


Figure 142: Example of a context-dependent selection of space layers involving binary operations on the selected space layers (left) and corresponding multilayered graph (right).

The space representation for fire fighter forces necessarily involves both L_{Topo} and L_{Fire} to enable finding shortest paths to the fire spots. This can be realized by adding both layers unchanged to the corresponding context-dependent space layer complex, e.g. $\mathcal{L}_F = \{L_{Topo}, L_{Fire}, \dots\}$. In this case, a path search algorithm would have to evaluate the inter-layer edges in $MLG(\mathcal{L}_F)$ between the space cells from either layer in order to identify the topographic regions affected by fire. Alternatively, the merge operation $L_{Topo} \oplus L_{Fire}$ would integrate both views in a single space layer yielding $\mathcal{L}_F = \{L_{Topo} \oplus L_{Fire}, \dots\}$ instead. As soon as the fire has been successfully fought, the space layer L_{Fire} can be removed from $\mathcal{L}$ in order to retrieve the state of the indoor space model before the fire incident.

As can be seen from the examples of this chapter, the MLSEM provides a strong framework for expressing user-related and environmental contextual information on separate space layers and for deriving configurations tailored to a given user context. If the contextual information however needs to be expressed in terms of individual space cells or boundary cells (cf. example 3.63) rather than entire space layers and involves non-spatial aspects (e.g., access restrictions or temporal constraints), then a complementary and more fine-grained modelling approach is advisable which is discussed in detail in chapter 5 of this thesis.

3.7 Spatio-semantic Analyses beyond Indoor Navigation

The MLSEM is not per se restricted to the task of indoor navigation. Due to the abstraction of arbitrary notions of indoor space through space cells and space layers as well as the explicit encoding of topological adjacency and overlap relationships between space cells in the multilayered graph, the MLSEM can rather be viewed as a *generic space-relation model* that facilitates spatio-semantic analyses even beyond navigation purposes. Although the focus of this thesis is put on indoor navigation, the spatio-semantic reasoning capabilities of the MLSEM are briefly illustrated in this chapter along a simple example which at least is implicitly related to the context of navigation.

Example 3.84. Consider the 2-dimensional indoor setting shown in figure 143 which again is based on a topographic space layer L_{Topo} consisting of two rooms R_1 and R_2 being connected to a corridor C, and a Wi-Fi sensor space layer L_{Wi-Fi} capturing the coverage areas of two Wi-Fi transmitters A and B (cf. figure 87 in chapter 3.2). Whereas figure 143a sketches the spatial overlay of both space layers, their individual view on indoor space is depicted in figure 143b.

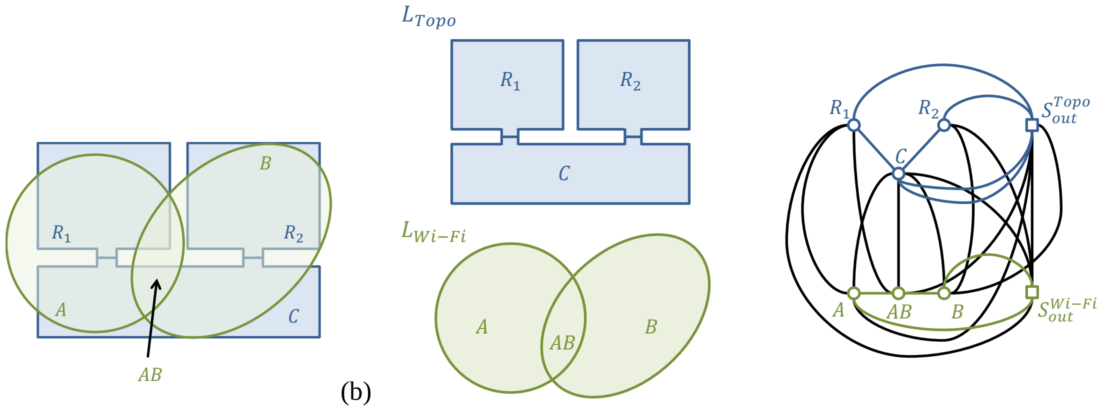


Figure 143: Spatial overlay of two 2*-dimensional space layers (a) and their separate representation as well as the corresponding multilayered graph (b).*

Let MLG be the multilayered graph of the space layer complex $\mathcal{L} = \{L_{Topo}, L_{Wi-Fi}\}$ as shown on the right of figure 143b. From the fact that the dual node of R_1 is linked to the dual node of the outer space cell S_{out}^{Wi-Fi} on L_{Wi-Fi}, which is formally expressed through $\{v_{TP}(R_1), v_{TP}(S_{out}^{Wi-Fi})\} \in E(MLG)$, it immediately follows that the topographic space region described by the space cell R_1 is not fully covered by sensor space cells from L_{Wi-Fi} and thus there is a lack of Wi-Fi signal reception in parts of R_1. The same can be observed for the space cells R_2 and C. This implies that moving persons or objects cannot be localized at every place in this example setting when merely using Wi-Fi as positioning technology. On the contrary, from $\{v_{TP}(R_1), v_{TP}(A)\} \in E(MLG)$ and $\{v_{TP}(R_1), v_{TP}(AB)\} \in E(MLG)$ it can be likewise deduced that parts of the room R_1 are covered by Wi-Fi signals and they can even be associated with the respective Wi-Fi transmitters.

The places in topographic space with and without Wi-Fi signal reception can be made explicit spatially. For instance, the region in R_1 covered by the Wi-Fi space cell A simply follows from the regularized Boolean intersection between their geometries in primal space. Let $GM_A(R_1)$ denote this subset of indoor space then it is given as $GM_A(R_1) = GM(R_1) \cap^* GM(A)$. Likewise, the region $GM_{\neg A}(R_1)$ in R_1 not being covered by A is yielded by the corresponding regularized Boolean difference so that $GM_{\neg A}(R_1) = GM(R_1) \setminus^* GM(A)$. Clearly, $GM(R_1) = GM_A(R_1) \cup^* GM_{\neg A}(R_1)$. What has been demonstrated for the space cells R_1 and A equivalently holds for any other pair of space cells from either space layer. If the dual nodes of two space cells are not linked in the multilayered graph, then it is a necessary consequence that there is no common subset in primal geometry space. For example, $\{v_{TP}(R_1), v_{TP}(B)\} \notin E(MLG) \Leftrightarrow GM_B(R_1) = \emptyset$ (cf. definition 3.37). Since the space cell S_{out}^{Wi-Fi} conceptually represents that partition of indoor space not being covered by Wi-Fi signals, it can be chosen to detect the subset of a topographic space cell with or without Wi-Fi signal reception independent of specific Wi-Fi space cells. Accordingly, $GM_{\neg S_{out}^{Wi-Fi}}(R_1) = GM(R_1) \setminus^* GM(S_{out}^{Wi-Fi})$ reveals those places inside R_1 where moving persons or objects are localizable using Wi-Fi. In the multilayered graph, the dual node of R_1 is only linked to the dual nodes of A and AB in addition to S_{out}^{Wi-Fi}. Thus, it is obvious that $GM_{\neg S_{out}^{Wi-Fi}}(R_1) = GM_A(R_1) \cup^* GM_{AB}(R_1)$.

In general, let $L_1, L_2 \in \mathcal{L}$ be two n-dimensional space layers participating in a space layer complex $\mathcal{L}$. Further, let S_1 and S_2 be two space cells from either space layer, with $S_1 \in \{\mathcal{C}(L_1), S_{out}(L_1)\}$ and $S_2 \in \{\mathcal{C}(L_2), S_{out}(L_2)\}$. Then the n-dimensional subsets $GM_{S_2}(S_1)$ and $GM_{\neg S_2}(S_1)$ of $\mathbb{R}^n$ denoting that part of the primal space geometry of S_1 covered by S_2 respectively not covered by S_2 are given as

$$GM_{S_2}(S_1) = GM(S_1) \cap^* GM(S_2) \text{ and } GM_{\neg S_2}(S_1) = GM(S_1) \setminus^* GM(S_2). \tag{3.85}$$

Let GM be an n-dimensional subset of $\mathbb{R}^n$ and $V(GM)$ its volume respectively area (depending on the dimension n), with $2 \leq n \leq 3$. Then the ratio $r_{S_2}(S_1)$ to which S_1 is covered by S_2 as well as the non-coverage ratio $r_{\neg S_2}(S_1)$ are expressed through the fractions

$$r_{S_2}(S_1) = \frac{V(GM(S_1) \cap^* GM(S_2))}{V(GM(S_1))} \text{ and } r_{\neg S_2}(S_1) = \frac{V(GM(S_1) \setminus^* GM(S_2))}{V(GM(S_1))}. \tag{3.86}$$

Both ratios provide a quantitative measure for the coverage between two space cells. Note that the primal space geometry $GM(S_1)$ per definition 3.2 is non-empty which gives $V(GM(S_1)) > 0$. In order to acquire the spatial coverage on the level of the two space layers L_1 and L_2, the primal space geometries of the respective space cell complexes need to be considered so that

$$GM_{L_2}(L_1) = GM\big(\mathcal{C}(L_1)\big) \cap^* GM\big(\mathcal{C}(L_2)\big) \text{ and } GM_{\neg L_2}(L_1) = GM\big(\mathcal{C}(L_1)\big) \setminus^* GM\big(\mathcal{C}(L_2)\big). \tag{3.87}$$

Likewise, the corresponding coverage ratios can be easily derived as

$$r_{L_2}(L_1) = \frac{GM(\mathcal{C}(L_1)) \cap^* GM(\mathcal{C}(L_2))}{GM(\mathcal{C}(L_1))} \text{ and } r_{\neg L_2}(L_1) = \frac{GM(\mathcal{C}(L_1)) \setminus^* GM(\mathcal{C}(L_2))}{GM(\mathcal{C}(L_1))}. \tag{3.88}$$

The analysis of common spatial regions between two space layers is further supported by the difference and the intersection operation. The following figure illustrates the results of $L_{Topo} \ominus L_{Wi-Fi}$ as well as $L_{Topo} \odot L_{Wi-Fi}$ for the above example.

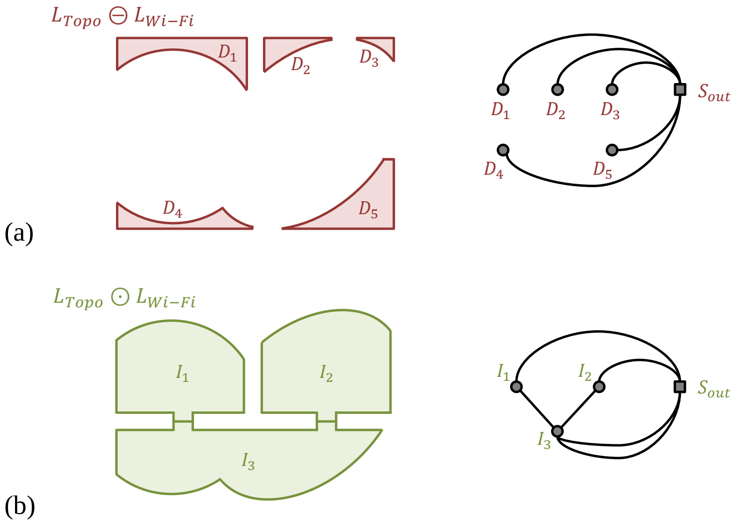


Figure 144: Result of the difference (a) and intersection (b) of the space layers from figure 143.

Let $L_{diff} = L_{Topo} \ominus L_{Wi-Fi}$ be the space layer that results from subtracting L_{Wi-Fi} from L_{Topo} as shown in figure 144a. It is obvious that the primal space geometry $GM(\mathcal{C}(L_{diff}))$ is equal to the subset yielded by $GM_{\neg L_{Wi-Fi}}(L_{Topo})$. However, the space description of L_{diff} is more expressive since it also provides a dual space representation as well as a semantic dimension. For example, the intra-layer graph of L_{diff} facilitates additional reasoning about the Wi-Fi signal coverage area (cf. right of figure 144a). Precisely, since the dual nodes are not linked by dual edges in the intra-layer graph of $L_{Topo} \ominus L_{Wi-Fi}$, it can be deduced that a navigation user travelling from one topographic space cell to another must cross a Wi-Fi sensor space cell on the way. This fact can be visually validated in figure 143a but is also confirmed by the intra-layer graph of the space layer $L_{Topo} \odot L_{Wi-Fi}$ as shown on the right of figure 144b since it is isomorphic to the intra-layer graph of L_{Topo} itself (cf. figure 143b) and the space cells on both layers are in one-to-one correspondence.

It follows that the MLSEM helps to spatially analyse the as-is state of the sensor coverage for a given indoor environment and localization technology in detail. The retrieved information can be used, for example, to support the planning of new positions for additional sensors and transmitters so that regions without sensor coverage are minimized or avoided. The presented formulas for computing the spatial coverage and ratios between two space layers clearly hold for arbitrary notions of indoor space. For example, they can be used to detect whether there are topographic regions not being covered by a security zone which might be an important query in security sensitive environments such as airports. Likewise, if in case of an emergency situation disaster areas are provided on an additional temporary space layer, then the coverage ratio between topographic space and the disaster areas can be used to understand the degree of demolition of a building or site even on the level of single space cells or their hierarchical aggregations. Although such queries are not directly related to navigation tasks they nevertheless impact indoor navigation.

Chapter 4
Conceptual Data Model

Almost all approaches to the modelling of indoor space for indoor navigation as discussed in chapter 2.2 stop at the level of the conceptual definition of indoor space and its spatial and semantic aspects or (if considered at all) at the mathematical formalization of these concepts. A mapping of the conceptual entities and their mathematical embedding onto a complete and consistent computer representation is seldom addressed. However, a computer representation is a natural prerequisite for the computer-based processing and storage of indoor space models as well as their exchange between computer systems, and thus renders a key issue in establishing an indoor navigation system or an LBS computing environment. This chapter is therefore dedicated to the derivation of a computer representation for the MLSEM which provides a complete mapping for all its conceptual entities as well as their spatio-semantic aspects in conformance with the sound mathematical framework developed in chapter 3.

The chapter is structured into three main parts. Since the geometric-topological modelling of space cells in both primal and dual space is at the core of the MLSEM, the first part reviews important spatial representation schemes for the computer modelling of spatial objects as well as geometric-topological data models proposed in the field of GIS. The review clarifies mathematical foundations and discusses existing data models against their suitability to fully express both the semantic and the geometric-topological aspects of the MLSEM as defined in chapter 3. Motivated by the goal to set the MLSEM in a standardized framework, focus is put on the ISO 19100 series of international standards for the modelling of geographic information issued by ISO/TC 211. The general concepts for the modelling of geographic features and their spatial representation as defined by the *General Feature Model* (ISO 19109) and the *Spatial Schema* (ISO 19107) are presented in detail and their feasibility in the context of the MLSEM is demonstrated.

In the second part, a conceptual data model for the MLSEM is designed and formally expressed in UML in conformance with the ISO standards. The complete and consistent mapping of the spatio-semantic representation of indoor space as developed in the previous chapter is argued in detail against the UML model. Moreover, further data requirements in the context of indoor navigation are identified and corresponding elements are added to the conceptual data model of the MLSEM. In a subsequent engineering step, a data encoding and exchange format is derived. The third part of the chapter then elaborates on the relation of the developed conceptual data model to existing international standards in the field of location-based service architectures.

4.1 Spatial Representation Schemes

The computer modelling of low dimensional geometric objects has been a field of intensive research over the last decades. In the early 1970’s, the term *solid modelling* was coined to refer to the emerging body of theory dealing with the *informationally complete* representation of *rigid solids* in order to enable and advance computer-aided design, manufacturing, construction, and architecture systems (Rossignac & Requicha 1999, Farin et al. 2002). Solid modelling is conceived as discipline in its own right nowadays and encompasses the theoretical foundations of geometric-topological modelling of solid objects and lower dimensional geometric shapes as well as the consistent set of associated techniques and computerized systems (also called solid modellers). The concept of informational completeness postulates that the computer representation of any physically existing or envisioned object shall (at least in principle) permit the automatic calculation of any well-defined geometric or topological property of the object (Requicha & Voelcker 1982), which differentiates solid modelling from other branches in computational geometry and computer graphics. On the one hand, this requires a common understanding of what a solid means mathematically. On the other hand, a relation between mathematical solids and their computer-based representation has to be defined that ideally captures the object in an unambiguous and unique way. A representation is unambiguous (or complete) if it corresponds to one and only one solid. It is unique if a given solid can only be encoded in a single way (Foley 2010). A formal mapping between an element of the mathematical model space and its computer representation is called a *representation scheme* (Requicha 1980).

Various representation schemes and hybrid combinations thereof have evolved in many different domains over time and underlie a range of complementary data structures, algorithms, and solid modellers. The following figure 145 illustrates the general modelling paradigm which shapes the notion of solid modelling by defining the relation between physical objects, mathematical models of solidity, and related classes of representations schemes (cf. Requicha 1980, Farin et al. 2002).

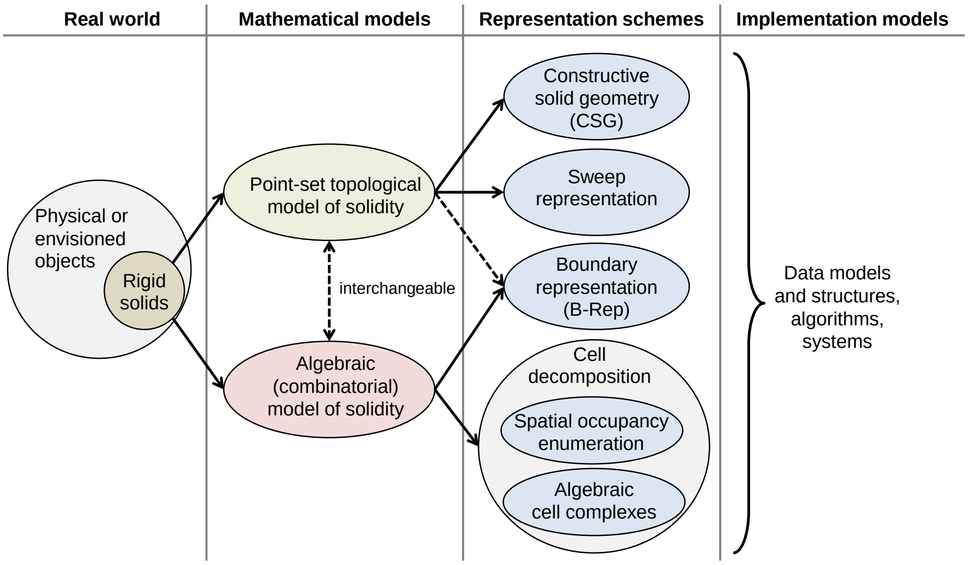

Real world
Mathematical models
Representation schemes
Implementation models
Physical or envisioned objects
Rigid solids
Point-set topological model of solidity
interchangeable
Algebraic (combinatorial) model of solidity
Constructive solid geometry (CSG)
Sweep representation
Boundary representation (B-Rep)
Cell decomposition
Spatial occupancy enumeration
Algebraic cell complexes
Data models and structures, algorithms, systems

Figure 145: Solid modelling paradigm after (Requicha 1980) and (Farin et al. 2002).

Intuitively speaking, a solid is a closed volume containing its boundary. As shown in the above figure, the field of solid modelling is typically restricted to the domain of *rigid solids* (e.g., Gröger & Plümer 2011a). The rigidity condition refers to the invariant configuration or shape of the solid which needs to be independent of the solid's location and orientation. Moreover, it is commonly requested that rigid solids be homogenous in dimension, compact subsets of $\mathbb{R}^3$, closed under rigid motions and certain Boolean set operations, and that their boundary clearly determines what is the inside and the exterior of the solid (cf. Requicha 1980). The two main mathematical models of solidity discussed in literature and shown in shown in figure 145 have their roots in point-set respectively algebraic topology. Both models are summarized briefly in the following as they can be associated with complementary classes of computer representations.

Point-set topological model of solidity. The first model utilizes the class of semi-analytical sets to define the notion of physical solidity by a continuum of points in $\mathbb{R}^3$. A set is semi-analytic if it can be represented through a finite Boolean combination of half-spaces each defined as set of the form $\{(x, y, z) \mid f_i(x, y, z) \leq 0\}$, where every function f_i is analytic (i.e., for every point p in the domain of f_i there is an open neighbourhood of p in which f_i can be equivalently expressed as Taylor series about p, e.g., in particular polynomials) (Requicha 1980). The homogeneity of dimension is ensured by enforcing the semi-analytical set describing the solid to be regular. Regularity implies that a semi-analytical set X is equal to the closure of its interior, and thus $X = \overline{Int(X)}$. If the set X is regular in $\mathbb{R}^3$ then each of its interior points has an open neighbourhood homeomorphic to the open unit 3-ball $\mathbb{B}^3$. Therefore, interior points having lower dimensional neighbourhoods are excluded which prevents physical artefacts of the solid such as dangling faces or edges as well as open boundaries. In order to meet the compactness condition of solids, the semi-analytical set additionally needs to be bounded in $\mathbb{R}^3$. Since regularity involves closedness, the boundedness property makes the set compact according to the Heine-Borel theorem (cf. theorem A.30).

In summary, rigid solids are given by bounded regular semi-analytical subsets of $\mathbb{R}^3$ which are also referred to as *r-sets* (Requicha 1980). Since this definition of solidity relies upon fundamental concepts of point-set topology (e.g., the interior, exterior, closure, and boundary of sets as well as the concept of neighbourhoods, cf. definition A.3 and definition A.5), the first mathematical model is equivalently called *point-set topological model of solidity*. It can be shown that the class of r-sets is closed under regularized Boolean set operations (cf. definition 3.64). R-sets are not restricted to the modelling of 3-dimensional solid objects but can also be applied to model surfaces and curves. However, the dimension of the ambient Euclidean or reference space needs to be adapted correspondingly in order to have the same meaning of the topological properties. For example, consider an r-set describing a surface being homeomorphic to the closed 2-disk $\overline{D}^2$. If this surface lives in $\mathbb{R}^2$ then the topological boundary of the r-set is a circle and its interior is an open 2-disk as expected. If, however, the surface is embedded in $\mathbb{R}^3$ then the topological boundary is the entire disk itself whereas its interior is the empty set (cf. appendix A.2) which breaks the presented formalism.[27]

The domain of *manifold solids* as comprehensively developed in appendix A.5 renders a subclass of rigid solids as defined above. Precisely, r-sets allow for representing non-manifold solid spaces which is illustrated in figure 146. It shows two L-shaped solids on the left as well as the result of their regularized Boolean union on the right. Whereas each L-shaped solid is homeomorphic to $\overline{\mathbb{B}}^3$ and thus is a topological 3-manifold, the union is non-manifold due to the self-tangency of the boundary but it can be legally represented as r-set.

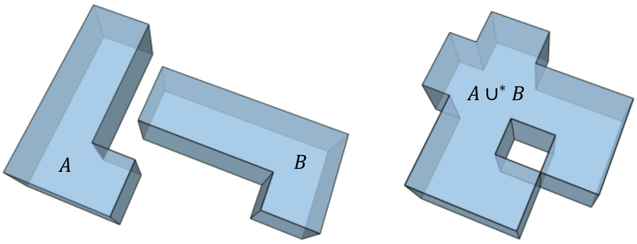


Figure 146: Two L-shaped manifold solids (left) and their non-manifold union (right) (Rossignac & Requicha 1999).

It is an easy consequence that manifold solids are not closed under regularized Boolean operations and are limited in expressivity which is seen as their main disadvantage when compared to rigid solids. This has also been demonstrated and discussed in chapter 3.5 in the context of the *make disjoint* map $f_{DISJOINT}$ (cf. definition 3.65) which underpins the developed space layer algebra. However, manifold solids also face advantages (cf. Farin et al. 2002). First, solids described by non-manifold r-sets are often considered non-physical since 3-dimensional real world objects do rather not touch at exactly a curve (cf. figure 146) or a point. Second, and more importantly, the 2-dimensional closed surface bounding a manifold solid can be classified up to homeomorphism based on its Euler-Poincaré characteristic (cf. theorem A.85). Likewise, k-shell manifold solids can be characterized by the alternating sum of their first three Betti numbers as illustrated in appendix A.5.3 (cf. proposition A.96). Both facts provide a strong but simple validity check for manifold solids (see also examples in chapter 3.1.3). Moreover, the set of Euler operators (cf. Mäntylä 1988) is closed on the domain of manifold solids. Thus, every topologically valid manifold solid can be constructed from an initial manifold solid by a finite sequence of Euler operators (cf. Mäntylä 1988). Finally, the Jordan-Brouwer Separation theorem (cf. theorem A.90) ensures the boundary determinism of manifold solids. Mostly due to these nice properties, many data models and structures for solid objects proposed in literature are based upon the more restricted manifold model of solidity (Foley 2010). Also note that non-manifold r-sets can be decomposed into and thus described by a finite set of manifold spaces which is utilized in the definition of $f_{DISJOINT}$ but also is the basis for the second mathematical model of solidity.

Algebraic (combinatorial) model of solidity. The *algebraic* or *combinatorial model of solidity* describes a solid object as 3-dimensional subset S of $\mathbb{R}^3$ that is decomposed into disjoint pieces each of which is a topological manifold (not necessarily of the same dimension) and whose union is S. The partition of the subspace S into a collection of disjoint submanifolds is called a *stratification*, whereas each of the submanifolds contained in the partition is said to be a *stratum*. A stratification thus provides a structure for point sets which are not necessarily manifold themselves. In literature, different types of stratifications with different topological properties are discussed. An often cited example is the *Whitney stratification* (Whitney 1965), and it has been shown that any r-set set admits a Whitney stratification (Thom 1969). Commonly, the way strata are combined and fit together is restricted to satisfy the *frontier condition* according to which a stratum M_1 that intersects the manifold boundary of a stratum M_2 has to lie entirely in the manifold boundary of M_2. Strata can be viewed to provide *“finite ‘spatial*

[27] Note that this consistently follows from the notion of interior, exterior, closure, and boundary of subsets of topological spaces as defined in appendix A. In literature, also different and non-equivalent definitions exist according to which the dimension of the ambient space not necessarily has to be adapted. However, in the context of the MLSEM, the differences are harmless because n-dimensional space cells are embedded in Euclidean n-space per definition 3.2. Thus, the dimension of the spatial object always agrees with the dimension of its ambient space leaving a co-dimension of zero.

addresses' for points in an otherwise innumerable continuum" (Farin et al. 2002, p. 478). The idea of the combinatorial model of solidity is to offer a computationally efficient way for deriving geometric and topological properties of the solid object from (simpler) properties of the strata and the information how they are combined.[28] Similar to r-sets, the combinatorial model of solidity is general enough to also include the modelling of lower-dimensional objects such as surfaces and curves.

It obviously follows that CW complexes as elaborated on in appendix A.3 are topological spaces which are stratified into open n-cells of various dimensions each of which being homeomorphic to the open unit n-ball $\mathbb{B}^n$ (cf. definition A.46 and definition A.51).[29] Strata differ from open n-cells in that they are not required to be connected, bounded, or globally homeomorphic to $\mathbb{B}^n$. For example, the unit 2-sphere $\mathbb{S}^2$ admits a Whitney stratification of a single stratum, whereas its minimal CW decomposition requires two cells, namely a 2-cell and a 0-cell. Similar to the general notion of stratification, also CW complexes are not required to form topological manifolds. A strong advantage of CW complexes over general stratifications is that the Euler characteristic is applicable to finite CW complexes and in fact is a topological invariant (cf. proposition A.83). Hence, finite CW complexes can be used to ensure and check the topological validity of the underlying solid object (in case the solid adheres to the notion of manifold solidity). This again explains why many data structures employing the algebraic model of solidity rely upon finite CW complexes or restrictions thereof.

The models of point-set topological solidity and algebraic solidity are consistent and interchangeable (cf. Farin et al. 2002), which is illustrated for the case of manifold solids and CW complexes in appendix A.5. This fact however does not imply that representation schemes based on either model are also interchangeable, i.e. that solid objects represented by a specific scheme can be translated into another scheme. In the following, important representation schemes in solid modelling are discussed and evaluated against their capability to express the geometric-topological description of both 2-dimensional and 3-dimensional space cells, boundary cells, and space cell complexes as defined in chapter 3. In his landmark paper, (Requicha 1980) has classified six distinct families of unambiguous representation schemes which are called *Pure Primitive Instancing*, *Spatial Occupancy Enumeration*, *Cell Decomposition*, *Constructive Solid Geometry* (CSG), *Sweep Representation*, and *Boundary Representation* (B-Rep). This classification is still commonly accepted in literature although both new and hybrid schemes have been introduced since then. An interesting alternative classification is proposed by (Farin et al. 2002) who distinguish *implicit and constructive* schemes from *enumerative and combinatorial* schemes which more closely follows the two mathematical models of solidity and also underlies the illustration in figure 145.

Constructive Solid Geometry (CSG). For the family of representation schemes referred to as *Constructive Solid Geometry* (CSG), a rigid solid object is built from a given set of *standard primitive objects* by applying regularized Boolean set operations and transformations. The modelling primitives are parameterized solids given by r-sets which typically reflect simple shapes such as blocks, cylinders, cones, triangular prisms, toroids, and balls. These primitives are instantiated using appropriate shape parameters controlling their spatial extent, position, and orientation (e.g., a block can be simply obtained by specifying its edge lengths). The placement in space depends on convention but typically each primitive is associated with its own local coordinate reference frame all of which have to be related to a common global reference frame. The CSG representation is expressed as ordered binary tree whose leaf nodes are the primitive instances and whose internal nodes either represent a regularized union, intersection, or difference, or a transformation which includes rigid body motions (rotation, translation) and scaling. It follows that each internal node produces an intermediate solid object with the root node denoting the resulting CSG solid. The following figure illustrates a simple CSG tree as well as the final solid object.

[28] Note that, in general, a purely geometric representation in addition to computational geometry algorithms suffices to derive properties such as containment respectively inclusion, adjacency, or boundary (e.g., from intersection operations or point-in-polygon checks). However, a combinatorial topological structure helps to accelerate computational geometry (cf. ISO 19107:2003) and to ensure the consistency and validity of complex geometric configurations (e.g., Gröger & Plümer 2011a).

[29] Likewise, a (Euclidean) simplicial complex is a stratification of its underlying polyhedron (cf. definition A.68).

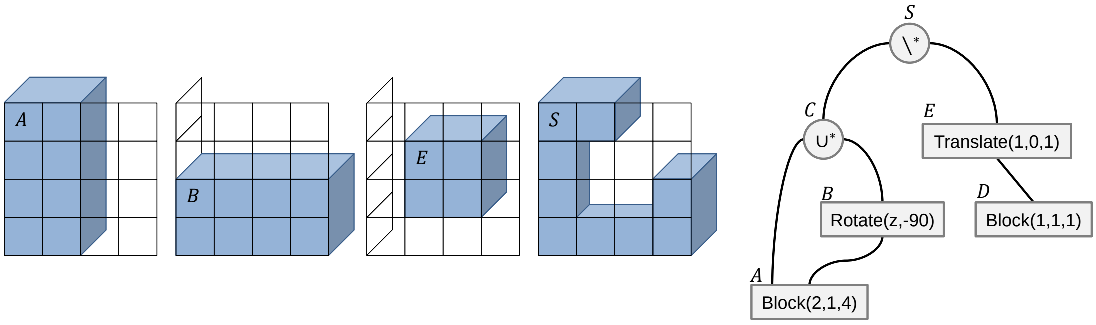


Figure 147: Example of a simple CSG tree (Rossignac & Requicha 1999).

The CSG scheme is the prototypical representative of the class of implicit and constructive schemes since the rigid solid is not given explicitly but only through a hierarchical sequence of construction operations which have to be satisfied by the final point set. It hence naturally corresponds to the point-set topological view of solidity. Due to the fact that the primitives are r-sets which are closed under regularized Boolean set operations, any CSG tree is a valid and unambiguous solid representation and itself evaluates to an r-set. CSG representations are concise and efficient with respect to storage space, but are non-unique because different sets of operations may yield the same object. A range of extensions to the presented notion of CSG have been presented in literature including the modelling of nonstandard primitives and unbounded CSG solids by relaxing the requirement for primitives to render r-sets as well as additional operations such as bending, twisting, Minkowski sums, and free-form deformations (see Rossignac & Requicha 1999 for references).

The CSG scheme is obviously suitable for describing the primal space geometry of arbitrary 2-dimensional and 3-dimensional space cells in conformance with definition 3.2. Since non-manifold and even unbounded solids are supported, it also facilitates the geometric description of space cell complexes as well as the corresponding outer space cell in primal space. The nice properties of CSG representations are however confronted with their limited expressivity regarding topological properties of (intermediate) CSG solids such as the connectedness of r-sets or the neighbourhood of points contained in the r-sets. As a direct consequence, CSG trees do not carry any information or representation of the solid's interior and boundary, and also lack the means to parameterize them. Thus, changing the spatial properties of the resulting CSG solid requires changing the parametric description of the leaf primitives and the re-computation of all intermediate representations. Moreover, since the configuration of the CSG solid is not known a priori, subsets of the represented solid point set are not spatially addressable and thus cannot be consistently referenced (cf. Farin et al. 2002). This precludes the explicit modelling of space cell boundaries which however are an important conceptual element of the MLSEM and a necessary prerequisite for the primal topological space representation of space cells according to definition 3.3. Computations of topological relationships between CSG solids are also non-trivial and commonly have to be solved through purely geometric checks or involve translations of the CSG solid into complementary schemes such as the Boundary Representation.

Sweep Representation. A second implicit and constructive representation scheme is the *Sweep Representation*. The basic idea of sweeping is to represent a solid object through a base set and a predefined space curve. The solid volume then is swept by the base set moving through space along the trajectory given by the space curve. Although this idea is straightforward and concise, *general sweep representations* typically fail to guarantee the solidity of the swept volume, and algorithms for computing geometric and topological properties of the represented solids are rarely available and non-trivial. Therefore, restrictions are often applied to the base set and/or the space trajectory. For example, *translational sweeping* enforces the base set to render a planar 2-dimensional r-set and defines the trajectory as straight line segment perpendicular to the base set, which results in simple extrusion solids. *Rotational sweeping* schemes define the base set similarly but provide a rotation axis instead. Both translational and rotational sweeping schemes produce rigid solids in an unambiguous and non-unique manner, but are obviously limited to express objects with translational or rotational symmetry. In fact, elements of the interior built environment such as rooms, corridors, or walls can mostly be idealized through translational sweeps which hence suffice to describe the primal space geometry of 3-dimensional topographic space cell. However, space cells on sensor or logical space layers may exhibit more complex shapes and hence require general sweeping methods. Moreover, sweep representations face the same drawbacks regarding the representation of the solid's interior and boundary as well as the derivation of topological relationships between solids as discussed for the family of CSG schemes.

Cell Decomposition. Enumerative and combinatorial representation schemes describe the space occupied by a solid object through a collection of smaller spatial chunks commonly called *cells* and a set of rules that generate the points belonging to the solid (cf. Farin et al. 2002). They hence correspond to the algebraic view on solidity and represent the solid point set explicitly. Two main families of *Cell Decomposition* schemes can be distinguished. The first family is called *Spatial Occupancy Enumeration* whose underlying principle is that the solid object is assembled from primitive rigid solids of uniform size and shape which admit a simple implicit or parametric description such as cubes (also called voxels in this context) or tetrahedrons. The primitives are enforced to be disjoint and to touch at most at faces, edges, or vertices which leads to their arrangement in rigid structures called spatial arrays. This assembly is guaranteed to be a valid, unambiguous, and unique solid representation and facilitates simple and efficient algorithms since every primitive can be spatially addressed. Spatial Occupancy Enumeration schemes are applicable to both 2-dimensional and 3-dimensional solid objects and underlie many geometric indoor space models (cf. chapters 2.1.2 and 2.2.1). Their main deficiencies are that complex shapes can only be approximated and that spatial inclusion relations between the primitive cells are not available explicitly. Whereas the former can be healed by allowing primitives of irregular shape and size (called *Irregular Cell Decomposition* schemes, cf. chapter 2.1.2), the latter hinders the immediate derivation of the solid's boundary and the referencing of subsets thereof which however is a prerequisite in the context of the MLSEM.

The second family of cell decomposition methods addresses the latter drawback by using *Cell Complexes* as representation scheme whose primitives are solid cells of different dimensions and which implement the algebraic model of solidity directly. The prototypical examples of this family are *Euclidean simplicial complexes* (cf. definition A.68) which are presented in detail in appendix A.4. Due to the facial structure of simplices inducing their spatial ordering and the strict rules for gluing n-simplices to an n-dimensional simplicial complex, all topological properties of the resulting complex such as its interior and boundary are available explicitly, exactly, and in a combinatorial way. Note that whereas simplicial complexes are discussed in appendix A.4 as purely topological structure on topological spaces, they also need to provide the geometry of the represented solid object in the context of solid modelling. Instead of defining the geometry separately for each simplex, often only the 0-simplices are associated with point coordinates. The geometric carriers of higher-dimensional simplices then result from (usually) linear interpolations based on inclusion relations between the simplices. Alternatively, only higher-dimensional simplices receive a geometric representation (e.g., 2-simplicies are assigned planar polygons) which allows for deriving the geometry of lower-dimensional simplices through intersection operations. Either way, the topological and geometric descriptions are not independent and thus need to be kept consistent (e.g., it must be ensured that 0-simplices are assigned distinct points in $\mathbb{R}^n$, that curves associated with 1-simplices do not intersect, etc.). This renders Cell Complex schemes verbose and maintaining the validity and consistency of the solid object a non-trivial task.

Similar to enumeration schemes, the triangulation of a solid object based on simplicial complexes is typically limited to an approximation of the real object shape. Therefore, CW complexes are commonly utilized to relax the strict restrictions of simplicial complexes (cf. appendix A.4) and to admit arbitrary shapes for cells which allow for spatially exact representations (e.g., the geometric description of a 2-cell may involve curved and freeform surfaces as long as the geometric configuration remains homeomorphic to a 2-cell). Since the facial structure of simplicial complexes is not inherent to general CW decompositions, the CW complexes are often enforced to be regular or proper (cf. definition A.57) and to render manifold solid objects. Like with simplicial complexes, the consistency between topological cells and geometric carriers needs to be carefully ensured.

Cell Complex schemes are unambiguous but non-unique solid representations. At least for schemes based on CW complexes it is an easy consequence that they are suitable for describing arbitrary 2-dimensional and 3-dimensional space cells in both primal geometry and topology space. Since the space cell boundaries are explicitly available in these schemes, topological relationships between space cells can be efficiently computed. Moreover, non-manifold and unbounded solids can be described which facilitates the representation of space cell complexes and the outer space.

Boundary Representation. The Boundary Representation (B-Rep) is one of the earliest methods for describing solid objects in computer graphics and solid modelling. The term Boundary Representation refers to a family of representation schemes that describe shapes merely by their boundaries. The theoretical foundation of B-Rep is the Jordan-Brouwer separation theorem according to which a closed $(n-1)$-dimensional manifold M separates $\mathbb{R}^n$ into a bounded and an unbounded component whose common topological boundary is M. If the bounded component is viewed as the solid object to be modelled then the representation of the boundary M already suffices to describe the solid in an unambiguous way. This original notion of a B-Rep solid is closely linked to the manifold view of solidity (cf. Mäntylä 1988). Although, in general, there is no assumption on the representation of the manifold boundary which may thus involve parametric, constructive, or combinatorial methods, most approaches to B-Rep modelling implement its cellular decomposition. Further restrictions are commonly applied to the manifold solid itself. For example, the solid object is often required to admit a polyhedral embedding in $\mathbb{R}^n$ which rules out pathological cases such as the Alexander horned sphere (cf. appendix A.5). However, it is important to note that the B-Rep scheme is also feasible to describe non-manifold solids adhering to the more general point-set topological model of solidity (cf. Farin et al. 2002).

B-Reps show characteristics of both implicit and enumerative representation schemes. They share with implicit schemes such as CSG that the interior point set of the resulting solid object is only given implicitly. However, and similar to enumerative cell decompositions, the solid's boundary is represented explicitly and provided with spatial addresses. The topological primitives of *combinatorial B-Rep models* are commonly called *nodes* (0-dimensional), *edges* (1-dimensional), and *faces* (2-dimensional) which are combined to form the closed boundary surface of the solid object. The primitives are typically not restricted to render triangulations such as with simplicial complexes. However, there are also subtle differences between combinatorial B-Rep models and algebraic cell complexes when it comes to the representation of interior holes within faces and voids within the solid volume. As shown in chapter 3.1.3, the cell decomposition of compact surfaces and manifold solids with more than one boundary component involves additional 1-cells (*bridge edges*) respectively 2-cells (*bridge faces*) in algebraic cell complexes. In contrast, most B-Rep schemes avoid the modelling of additional edges and faces by introducing 1-dimensional *loops* as well as 2-dimensional *shells* as further primitives (cf. Mäntylä 1988).[30] Whereas loops are commonly defined to be homeomorphic to the 1-sphere $\mathbb{S}^1$, a shell is mostly understood to be a closed manifold surface being homeomorphic to the n-holed torus $\mathbb{T}^2$.[31] The boundary of each face is then given by loops which themselves are cycles of edges that only intersect in their bounding nodes. Likewise, the boundary of the solid volume is made up by shells being sets of connected faces that only intersect in common edges or nodes. In figure 148, the differences between B-Rep models and algebraic cell complexes are illustrated for a surface with hole respectively a solid with cavity.

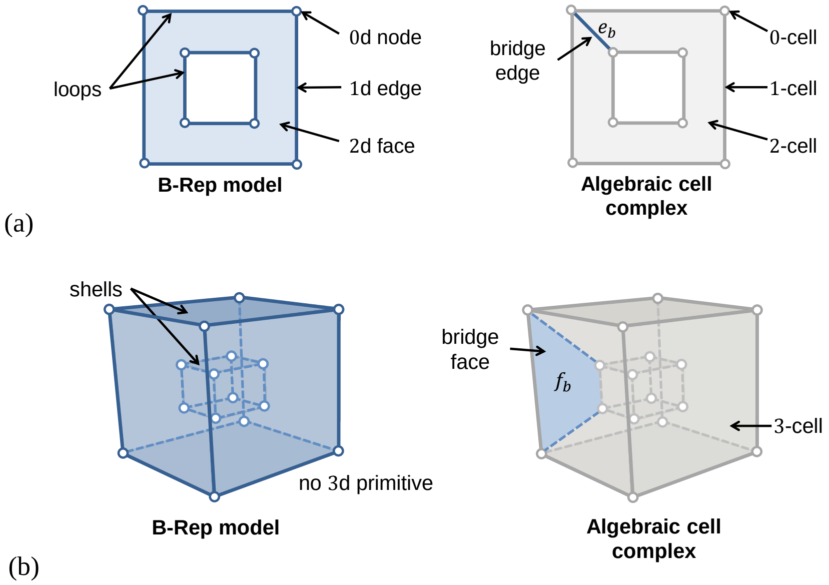


Figure 148: Differences between B-Rep models and algebraic cell complexes in two dimensions (a) and three dimensions (b).

The B-Rep face shown on the left of figure 148a has two bounding loops each of which is a topological circle decomposed into four edges and four nodes. Note that the face itself is not a closed 2-cell since its interior is not homeomorphic to the open unit 2-ball $\mathbb{B}^2$. From this it immediately follows that the cellular structure of the face

[30] The names of the additional primitives differ for different B-Rep schemes. For example, loops are likewise called *rings*, whereas shells are often referred to as *bodies* or *volumes*.

[31] More restrictive B-Rep schemes limit shells to be homeomorphic to the 2-sphere $\mathbb{S}^2$.

violates condition (ii) of definition A.48 and therefore is not a valid algebraic cell complex. Topologically, the face is homeomorphic to the annulus for which a valid CW decomposition consisting of eight 0-cells, nine 1-cells, and one 2-cell is shown on the right of figure 148b (cf. discussion in chapter 3.1.3.1). This result can also be validated using the Euler characteristic. For the annulus, it is given by $\chi = 2 - 2\mathrm{g} - \mathrm{h} = 2 - 0 - 2 = 0$. The same Euler characteristic follows for the CW decomposition due to $\chi = 8 - 9 + 1 = 0$, but not for the alternating sum of the number of primitives in the B-Rep model. A similar observation can be made for the 3-dimensional solid with cavity shown in figure 148b.

Consequently, the Euler characteristic as defined for finite CW complexes in proposition A.83 is not applicable for combinatorial B-Rep models which include loops and shells. It has therefore been extended in literature to account for these primitives. Precisely, let N, E, F, L, and S denote the number of nodes, edges, faces, loops, respectively shells in a B-Rep model and let G be the sum of each shell's genus. Then the so-called *extended Euler-Poincaré formula* is given by

$$N - E + F - (L - F) - 2(S - G) = 0 \;\; \text{(Mäntylä 1988).} \tag{4.1}$$

Applied to the above B-Rep face we get $8 - 8 + 1 - (2 - 1) - 0 = 0$ which satisfies the formula. The same holds for the B-Rep solid in figure 148b due to $16 - 24 + 12 - (12 - 12) - 2(2 - 0) = 0$. The extended Euler-Poincaré formula thus is suitable for checking the validity of the cellular structure of the boundary representation.

Like with general cell decomposition schemes, B-Reps have to ensure the consistency of the topological primitives and their geometric carriers which is supported by the inclusion relations between the primitives and their induced spatial order. Current B-Rep schemes often admit curved and freeform surfaces as geometric embeddings for faces and hence allow for spatially exact solid representations. They share the drawback with cell decompositions that they are non-trivial to construct and to maintain in a valid way (e.g., Gröger & Plümer 2012a), but are usually less verbose due to the reduction of the modelling dimension. Regarding the geometric-topological representation of 2-dimensional and 3-dimensional space cells of the MLSEM in primal space, B-Reps are an alternative to general Cell Complex schemes. Although B-Reps are not necessarily valid CW complexes as shown above, the missing bridge edges and faces can be easily derived (cf. proposition 3.34 and proposition 3.35) in order to satisfy the mathematical model of the primal space topology of space cells. Moreover, they provide the necessary combinatorial information to deduce topological properties and relationships between space cells and generally support both unbounded and non-manifold structures. A further advantage of the B-Rep scheme is that solid objects represented in one of the complementary schemes discussed in this chapter can be unambiguously translated into the B-Rep scheme which is not true vice versa for all combinations (cf. Requicha 1980).

4.2 Geometric-Topological Data Models in GIS

Based on the presented classification of computer-based representation schemes for 3-dimensional and lower dimensional spatial objects, this chapter reviews geometric-topological data models and data structures that have been proposed for the modelling, storage, and exchange of spatial objects in the field of GIS, and which rely upon the discussed representation schemes and their theoretical foundations.

Early approaches to the description of the spatial characteristics of geographic real world objects in up to three dimensions have been presented more than two decades ago. Most of the data models that have emerged since then rely upon Cell Decomposition schemes and in particular favour the Boundary Representation approach (Gröger & Plümer 2011a). Besides the advantages offered by this family of representation schemes as illustrated in the previous chapter, this can also be explained by the way spatial data is acquired in the GIS domain. Typically, spatial information about geographic objects is derived from measurements and surface observations based on (sensor-specific) extraction methods from fields such as photogrammetry or engineering surveying (Nagel et al. 2009). Geographic objects are hence spatially described by their observable boundary surfaces applying an *accumulative modelling principle* which is naturally supported by the B-Rep model (Kolbe & Plümer 2004). Nevertheless, existing GIS data models vary in the primitives they use and the rules for combining them, in their geometric and topological expressivity, and the restrictions they apply to the represented solid objects (e.g., manifold vs. non-manifold, bounded vs. unbounded, etc.). A further distinction can be made as to which extent conceptual and semantic data modelling (cf. 2.1.3) is supported in addition to the spatial representation of the geographic objects. A comprehensive survey of prominent 3-dimensional topological models in GIS is given in (Zlatanova 2000) and (Zlatanova et al. 2004).

The *Formal Data Structure* (FDS) proposed by (Molenaar 1990) is often cited as being the first 3-dimensional geometric-topological data model in the context of GIS (e.g., Zlatanova et al. 2004, Gröger & Plümer 2011a). As shown in figure 149, it introduces *nodes*, *arcs/edges*, and *faces* as the topological primitives with faces being bounded by edges that are decomposed into arcs having exactly two nodes as limit. Based on these primitives, 0-dimensional to 3-dimensional geometric objects, namely *points*, *lines*, *surfaces*, and *bodies*, can be realized. Bodies are restricted to render connected, compact manifold solids which are described by their bounding faces. Both interior holes in surfaces as well as cavities in bodies are supported. The general notion of Cell Decomposition schemes is relaxed in such that singularities can be modelled through nodes or arcs that are lying inside faces or bodies. Only the nodes are associated with point coordinates. The geometry of arcs then results from linear interpolations whereas faces are enforced to be planar. Bodies are consequently given by geometric polyhedra. The geometric information is hereby strictly coupled with the topological primitives. For example, although it is possible to represent a complex non-planar surface object as composition of several face patches, a single face primitive cannot be used to topologically describe a non-planar surface. Thus, the same geometric object does not admit different decompositions into separate configurations of topological primitives. This flexibility however has been introduced as strength of the MLSEM (cf. chapter 3.1.3) which would be lost when applying the FDS as data model for the representation of space cells. An advantage of the FDS is the support for semantic *classes* which can be associated with the geometric objects and which allow for modelling semantic aspects of the real world object in addition to their spatial characteristics.

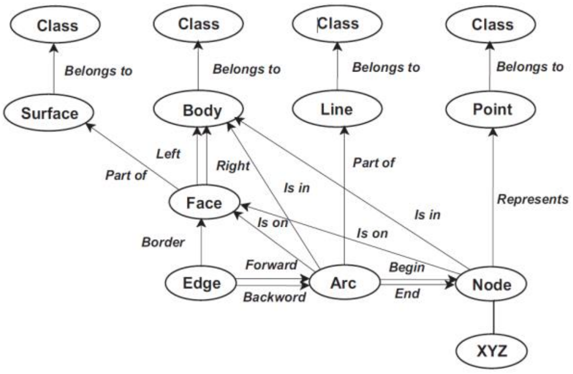


Figure 149: The Formal Data Structure (FDS) from (Molenaar 1990) (taken from Zlatanova et al. 2004).

The FDS has been used and modified in several subsequent research works. Whereas (Flick 1999) adds a dedicated 3-dimensional topological primitive, the *Simplified Spatial Model* (SSM) proposed by (Zlatanova 2000) even decreases the number of topological primitives by omitting arcs and edges from the data model. This makes the SSM more concise and optimizes storage space. Faces are defined to have planar and convex geometric embeddings and are given by an ordered set of nodes carrying the point coordinates, with each pair of subsequent nodes implicitly representing an arc. Similar to the FDS, the 3-dimensional body objects render geometric polyhedra and singularities in faces and bodies are allowed (cf. figure 150a). The *Urban Data Model* (UDM) of (Coors 2003) follows a similar approach and also neglects 1-dimensional topological primitives (cf. figure 150b). However, the UDM is more restrictive than the SSM since faces need to be triangular and singularities are excluded mainly in order to facilitate efficient visualizations. Both the SSM and the UDM establish a strong coupling between topological and geometric entities. Moreover, the lack of 1-dimensional primitives hinders the modelling of boundary cells in case of 2-dimensional space cells which however is a necessary prerequisite of the MLSEM. The *Tetrahedral Network* (TEN) structure presented by (Pilouk 1996) realizes a data model and database schema for representing spatial objects by means of simplicial complexes. It introduces the topological primitives *node*, *arc*, *triangle*, and *tetrahedron*, which are combined to form complex geometric shapes following the strict rules established by the mathematical notion of simplicial complexes. The TEN structure consequently provides a topological primitive for each dimension (cf. figure 150c). Likewise, the object-oriented data model for complex 3-dimensional geographic objects (OO3D) developed by (Shi et al. 2003) relies upon a decomposition of spatial objects into simplicial complexes and facilitates the attributive modelling of semantic information (cf. figure 150d). Both the TEN and the OO3D model require triangulations respectively tetrahedralizations of complex shapes. Although corresponding algorithms are available, this is to be seen disadvantageous in the context of the MLSEM since arbitrarily-shaped space cells need to be artificially subdivided which impacts the coherence between spatial and semantic entities.

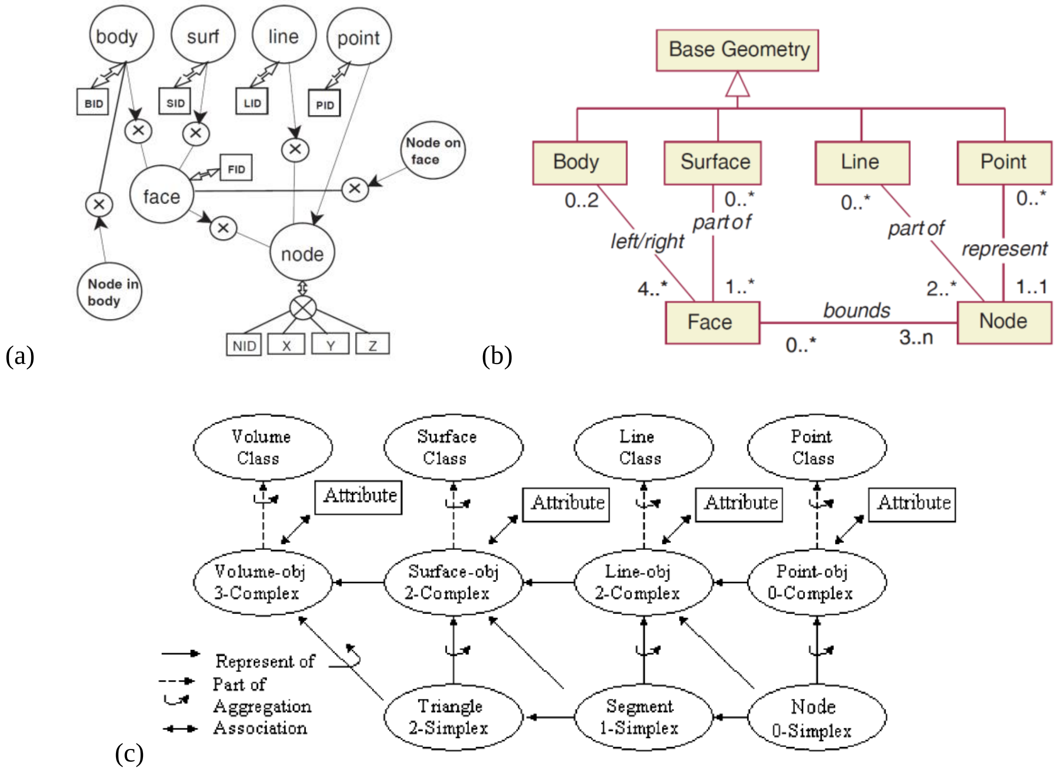


Figure 150: SSM (a), UDM (b), and OO3D (c) (taken from Zlatanova et al. 2004).

In contrast to the above data models, (Kim et al. 2009) present a geometric-topological data model called the *Prism Model* which uses Sweeping in addition to B-Rep. The primitives of the Prism Model are *points*, *curves*, *surfaces*, and *solids*. Whereas these primitives can be constructed according to the cellular decomposition rules of the B-Rep scheme (e.g., surfaces are bounded by curves which are bounded by points), they can also be used as base sets to express 1-dimensional up to 3-dimensional *extrusive geometries*, namely *extrusive points*, *extrusive curves*, and *extrusive surfaces* as shown in figure 151.

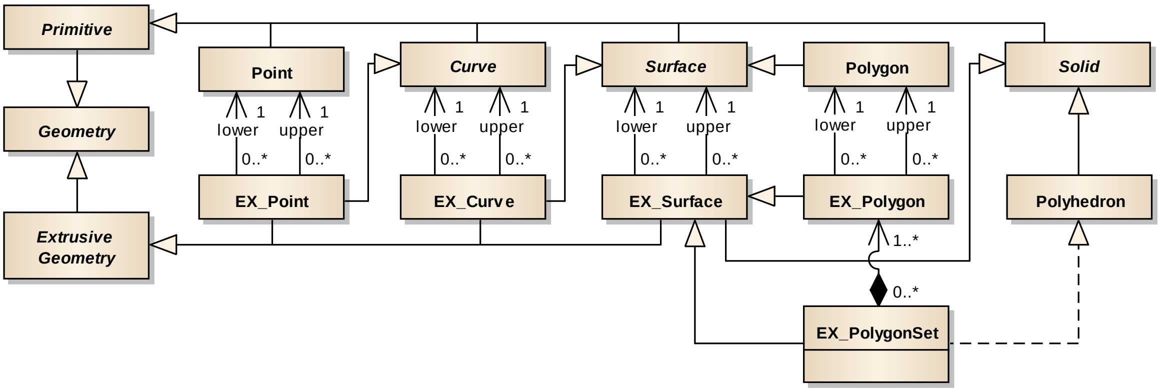
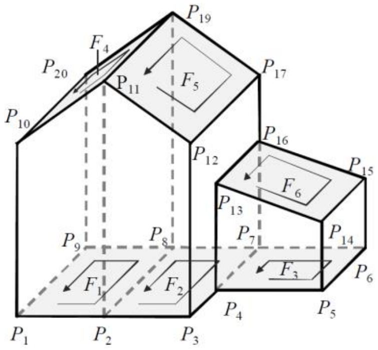


Figure 151: The Prism Model (after Kim et al. 2009).

Each extrusive geometry entity is bounded by a *lower* and an *upper* primitive of the same dimension. The point set of the extrusive geometry is then swept out by moving the lower primitive along a straight space curve to the upper primitive, which basically realizes a translational sweeping. For example, an extrusive point is given by a lower and an upper point primitive and the straight line segment in between, and thus renders a curve primitive. Likewise, an extrusive surface is given by two surface primitives that sweep out a solid primitive (polygonal prism). Since the resulting swept solid is necessarily bounded by six surfaces, the *polyhedron* primitive (being a subtype of solid) allows for describing more complex solid configurations. A polyhedron is constructed from a set of *extrusive polygons*, with both the upper and lower geometry given by a collection of *polygons* that must be isomorphic to a single surface. However, both the solid and the polyhedron primitive are restricted to be homeomorphic to a connected, compact 1-shell manifold solid. Obviously, neither extrusive surfaces nor extrusive solids hence support interior voids.

The Prism Model is more restrictive than general translational sweeping. First, the sweeping direction is enforced to be parallel to the z-axis (encoding height information). Second, the lower and upper primitives need to have the same footprint when projected onto the x-y-plane, and each of the vertices of the upper geometry must have higher z-values than its corresponding vertex in the lower geometry (Kim et al. 2009). These restrictions ensure that the resulting extrusive geometry is always valid and even allow that the upper geometry can have a different shape than the lower geometry, which provides more flexibility than general translational sweeping. An example of an extrusive polygon set describing a saddle-roof building and an adjacent garage is shown in the following figure.



$Solid_1 = \{EPS_1\}$
$EPS_1 = \{EP_1, EP_2, EP_3\}$
$EP_1 = \{F_1, F_4\}$
$F_1 = \{P_1, P_2, P_8, P_9\}$
$F_4 = \{P_{10}, P_{11}, P_{19}, P_{20}\}$
$EP_2 = \{F_2, F_5\}$
$F_2 = \{P_2, P_3, P_7, P_8\}$
$F_5 = \{P_{11}, P_{12}, P_{17}, P_{19}\}$
$EP_3 = \{F_3, F_6\}$
$F_3 = \{P_4, P_5, P_6, P_7\}$
$F_6 = \{P_{13}, P_{14}, P_{15}, P_{16}\}$

Figure 152: A saddle-roof building described using the Prism Model (Kim et al. 2009).

The Prism Model facilitates a compact representation of spatial objects which is less verbose than comparable B-Rep descriptions. Although the geometric expressivity of the applied translational sweeping is limited, it suffices, for example, to describe the regular shapes of entities of the interior built environment such as rooms, corridors, or elevator shafts. Based on the strict representation and construction rules for extrusive geometries, (Kim et al. 2009) develop a formal model and corresponding algorithms for efficiently computing the topological relationships *upperDisjoint*, *upperMeet*, *equal*, *overlap*, *contains*, *inside*, *covers*, *coveredBy*, *lowerDisjoint*, and *lowerMeet* between two extruded objects. Thus, the topological expressivity of the Prism Model is stronger than that of usual implicit and constructive representation schemes. The Prism Model is hence feasible to model the primal space geometry of topographic space cells in two and three dimensions, and to derive intra-layer and inter-layer relationships between them. However, due to the implicit representation of parts of the boundary of extruded objects, the explicit modelling of boundary cells is hindered. Moreover, arbitrarily-shaped space cells (e.g., sensor space cells) can only be coarsely approximated based on extrusive geometries.

In a series of paper, (Gröger & Plümer 2011a), (Gröger & Plümer 2011b), and (Gröger & Plümer 2012b) introduce a formally well-developed combinatorial B-Rep model whose topological primitives are called *vertex*, *edge*, *face*, and *solid* (cf. figure 153). The vertices, edges, and faces correspond to closed n-cells of the corresponding dimension and are enforced to constitute a proper CW complex, i.e. the non-empty intersection of any two primitives must be a lower dimensional primitive which is part of the boundaries of both primitives. From faces being closed 2-cells it follows that interior holes are not supported. Solids are given implicitly by the enumeration of their bounding faces and are further distinguished into *bounded* and *partially bounded* solids. A bounded solid is defined to be topologically equivalent to a connected, compact 1-shell manifold solid, and thus admits a boundary being homeomorphic to an n-holed torus (cf. definition A.93). However, this also implies that interior cavities are ruled out. The geometric representation follows from point coordinates associated with the vertices with the restriction that edges need to be straight line segments and faces have to be planar. On top of the solid primitives, *features* are defined which allow for specifying the semantics of the spatial objects. The data model supports hierarchical aggregations of features to form *complex features* as long as the spatial aggregate associated with a complex feature remains homeomorphic to a bounded solid.

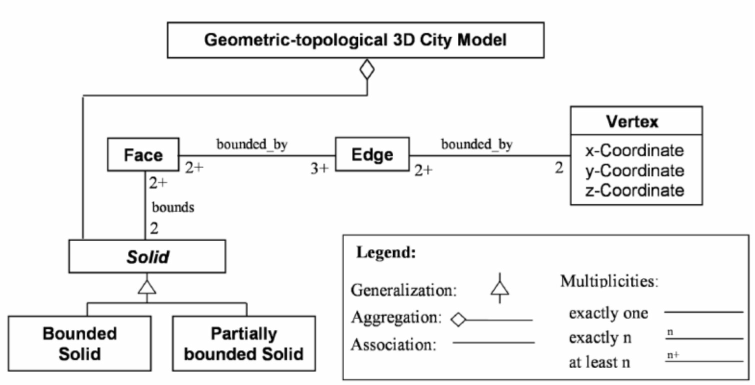


Figure 153: The geometric-topological model of (Gröger & Plümer 2011a).

The main motivation for the development of the data model is the representation of 3-dimensional city models. A city model is hereby understood as collection of non-overlapping solid features (e.g., buildings, vegetation and transportation objects, etc.). Moreover, (Gröger & Plümer 2011a) assume a full tessellation of 3-dimensional Euclidean space $\mathbb{R}^3$ and therefore foresee the possibility to model non-compact manifold solids with boundary which are referred to as partially bounded solids. Two partially bounded solids are introduced: an *air solid* representing the free space above terrain (air space), and an *earth solid* modelling the earth's mass. The partial boundary of the air solid is defined to be the visible surface when looking onto the city model from above, and thus includes the terrain surface as well as the those boundary faces of bounded solids being adjacent to the air space. Likewise, the partial boundary of the earth solid is the surface seen from below. Note that in both cases the partial boundary is itself an open 2-manifold (i.e., non-compact and without boundary), and is given by a 2.8-dimensional map which has been proposed by (Gröger & Plümer 2005) as mathematically sound data structure for the representation of open 2-manifolds based on a proper CW decomposition. A 2.8-dimensional map includes a single face called *Out* which is topologically unbounded and hence non-compact. The entire tessellation of $\mathbb{R}^3$ is then given by a connected, proper CW complex which contains the 2.8-dimensional map as subcomplex. In figure 154a, a simple setting of two buildings modelled as aggregation of bounded solids as well as the partial boundary of the air and the earth solid is depicted, whereas figure 154b shows a cross profile.

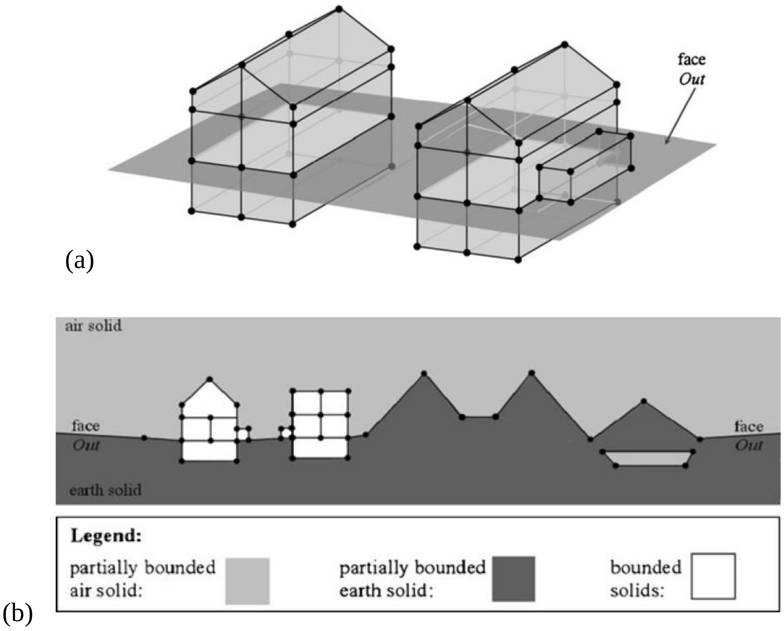


Figure 154: Two buildings described by six respectively seven bounded solids (a) and sketch of a cross profile showing the partially bounded air and earth solid (b) (Gröger & Plümer 2012b).

(Gröger & Plümer 2011a) also put a strong focus on the geometric-topological consistency and validity of data which is structured according to their data model. Therefore, *axioms* are introduced which on the one hand are proven to be equivalent to the underlying mathematical model as described above, but on the other hand allow for computationally effective and efficient implementations of routines for detecting errors such as pairwise penetrations or intersections of solids. Although consistency is a global property, global consistency checks are expensive and thus the axioms and algorithms are restricted locally to faces and edges in the boundary of a solid. In order to maintain the consistency of the data throughout changes and dynamic updates, a complete set of *transaction rules* for operations such as splitting and merging of faces or solids is presented in (Gröger & Plümer 2012b). The rules exceed commonly applied Euler operators in such that they explicitly address 3-dimensional objects and guarantee both their geometric and topological consistency.

The data model presented by (Gröger & Plümer 2011a) is generally feasible to represent both 2-dimensional and 3-dimensional space cells of the MLSEM in both primal geometry and topology space. Although the model is intended to describe entire city models, the modelling of the building interior along disjoint solid primitives is not excluded (cf. Gröger & Plümer 2012b). The aggregation of non-overlapping solids conforms to the notion of a space cell complex (cf. definition 3.7) and facilitates its consistent representation which is not supported by the models presented above. Moreover, the full tessellation of space agrees with the equivalent condition expressed for the primal space representation of space layers, and the notion of the air and earth solid corresponds to the concept of the *outer space cell* which has already been introduced in early publications on the MLSEM (cf. Becker et al. 2009a). However, the data model of (Gröger & Plümer 2011a) also lacks expressivity compared to the mathematical model underlying the MLSEM. First, interior voids are allowed for space cells but are neither supported by face nor by solid primitives. Second, due to the restriction to proper CW decompositions the number of cells required to represent a topological space is higher than for general CW decompositions. Thus, the dual graph representation will consequently contain a higher number of nodes and edges. Third, both bounded and partially bounded solids are restricted to manifold spaces which excludes non-manifold configurations of space cell complexes as discussed in chapter 3.1.3.3. Fourth, the notion of outer in the MLSEM is more general and actually can be viewed as the space occupied by both the air and the earth solid, and thus makes the explicit modelling of unbounded 2.8-dimensional maps obsolete. And finally, the geometry is strongly coupled with the topological primitives and disallows curved or freeform surfaces, which requires the approximation of complex shapes.

Another B-Rep model called *cell-tuple structure* has been proposed by (Brisson 1989) and further refined by (Pigot 1996). It extends classical data structures for B-Reps from computational geometry such as the *quad-edge data structure* of (Guibas & Stolfi 1985) or the *facet-edge data structure* of (Dobkin & Laszlo 1987) by utilizing n-dimensional *subdivided manifolds* to describe the boundary of an $(n+1)$-dimensional spatial object. A subdivided manifold is simply given by an n-dimensional, finite, regular CW complex whose underlying space is a closed orientable n-manifold. The modelling primitives of the cell-tuple structure are hence closed k-cells, with $0 \leq k \leq n$. In contrast to the data model of (Gröger & Plümer 2011a), the CW complex is not restricted to meet the intersection property. Based on the k-cells, a *cell-tuple* is defined as $(n+1)$-tuple containing a k-cell from each dimension which need to be related by inclusion. A *switch* operator is introduced that allows for navigating between adjacent cell tuples and thus to iteratively visit all cells of the subdivided manifold. Similar to the model of (Gröger & Plümer 2011a), the geometric embedding of 1-cells has to be a straight line segment whereas 2-cells need to be planar. An interesting aspect of the cell-tuple structure is that it supports the simultaneous representation of the corresponding dual CW complex, and the authors describe the need of an *outside* space in case of representing manifolds with boundary. However, a precise definition of this outside space is not provided. In general, the cell-tuple structure and especially their underlying subdivided manifolds are suited to model both the primal and dual geometric-topological representation of 2-dimensional and 3-dimensional space cell complexes. A major drawback is the lack of semantic modelling elements (like with almost all B-Rep structures from computational geometry). Moreover, internal voids as well as non-manifold structures are not supported, and neither is a decoupled modelling of geometry and topology.

The *dual-half edge* (DHE) structure of (Boguslawski & Gold 2009) is obviously a further geometric-topological B-Rep data model and important related work in the context of the MLSEM. Since it has already been introduced in chapter 2.2.3 and has been comprehensively discussed and compared to the MLSEM in the course of the previous chapters, a redundant presentation is omitted here.

4.3 ISO 19100 Standards Family

The ISO 19100 standards family is a series of international standards for the definition, description, and management of *geographic information* developed and published by the Technical Committee 211 (ISO/TC 211) of the International Organization for Standardization (ISO). Geographic information is broadly understood to comprise any knowledge and information about real world objects and phenomena that can be directly or indirectly given a location relative to the Earth's surface. The ISO 19100 standards series aims at enabling a common notion of geographic information as well as at establishing a common language for describing and structuring geospatial objects along their spatial and non-spatial aspects. A main goal of the standardization activity is to facilitate *interoperability* between heterogeneous geographic information systems. On the one hand, this addresses *syntactic interoperability* which refers to the capability of two or more (distributed) computing systems to communicate and exchange geographic information and data independent of software components, system borders, and data providers based on standardized data formats, interfaces, and protocols. On the other hand, *semantic interoperability* shall be accomplished in order to allow a computing system to automatically interpret the meaning and the logical structures of the exchanged content according to a shared and unambiguous reference information model. In order to reach this goal, the ISO 19100 series includes the specification of standardized methods, tools, and services that support the acquisition, analysis, access, presentation, and transfer of geographic data between users, systems, and locations (cf. ISO 19101:2002). In this sense, the ISO 19100 standards family can be viewed as a collection of independent abstract standards which cover different aspects of geographic information (e.g., spatial, semantic, temporal, or qualitative aspects) from different viewpoints (e.g., data capturing, modelling, processing, providing), but which do not intend to provide a complete model for the whole universe of geographic information (Kresse & Fadaie 2004).

In the context of spatio-semantic data modelling, two core ISO specifications are of relevance and are discussed in more detail in the following subchapters. First, the ISO 19107:2003 *"Geographic information – Spatial schema"* provides a general conceptual schema for describing the geometric-topological characteristics of spatial objects in up to three dimensions, and defines standard spatial operations on spatial objects. Second, the ISO 19109:2005 *"Geographic information – Rules for application schemas"* defines the concept of a *feature* as the basic unit for geoinformation modelling and processing, and introduces rules and principles for the conceptual modelling of features and their properties (amongst them spatial ones) in a given application context. The abstract concepts presented in both standards form the theoretical foundation for concrete implementations and support the semantic interoperability between geographic information systems. Further standards in the context of geographic information modelling deal with related topics such as spatial referencing based on coordinates (ISO 19111:2007) or geographic identifiers (ISO 19112:2003), the description of temporal characteristics (ISO 19108:2005) and quality aspects (ISO 19113:2002) of features as well as the modelling and provision of corresponding metadata (ISO 19115:2003). A detailed presentation of these topics is however outside the scope of this thesis, and the reader is referred to corresponding literature (e.g., Kresse & Fadaie 2004).

In addition to ISO/TC 211, the Open Geospatial Consortium (OGC) is one of the most important bodies to develop and release standards in the field of GIS at an international level. Whereas ISO is to be seen as an umbrella organization composed of representatives of national standardization bodies from its various member countries, the OGC is a non-profit industry consortium with members from industry, academia, and government agencies. OGC standards are commonly said to be *de-facto* standards as they are enforced through the broad acceptance of the GIS community rather than through some formal or legal act as with *de-jure* standards issued by standards setting bodies such as ISO. Similar to the ISO 19100 volumes, the OGC differentiates between *Abstract Specifications* (organized into *Topics*) that provide the conceptual foundation and general architecture for most of the OGC standardization activities, and platform-specific *Implementation Standards* that built upon the abstract concepts and focus on open interfaces, data encodings, and application schemas in order to enable both syntactic and semantic interoperability between different kinds of spatial processing systems mostly over the internet. Since 1998, both organizations closely cooperate in harmonizing the conceptual basis for geographic information and in the joint development and mutual approval of geospatial standards. For example, the Spatial Schema of ISO 19107:2003 has replaced the *OGC Abstract Specification Topic 1 – Feature Geometry*, and the notion of a geographic feature as well as the conceptual modelling of features (*Topics 5*, *8*, and *10*) has been aligned with the definitions, rules and principles from ISO 19109:2005.[32] Hence, both ISO 19107:2003 and ISO 19109:2005 provide a standardized and commonly accepted framework for spatio-semantic data modelling in the field of GIS.

[32] Likewise, OGC standards such as the *OGC Web Map Service* (WMS), the *OGC Web Feature Service* (WFS), or the *OGC Geography Markup Language* (GML) have been incorporated in the ISO 19100 standards family (cf. ISO 19128:2005, ISO 19142:2010, and ISO 19136:2007).

4.3.1 ISO 19107 – Spatial Data Modelling

The ISO 19107 Spatial Schema organizes the spatial characteristics of geographic features into two separate conceptual models, namely a *geometry model* and a *topology model*. The geometry model offers 0-dimensional up to 3-dimensional geometric objects that facilitate the quantitative description of the spatial characteristics of features including their shape, size, and location by means of coordinates and mathematical functions. The topology model, on the other hand, defines corresponding topological primitives which allow for constructing combinatorial structures that encode qualitative and invariant spatial characteristics such as the connectivity of geometric primitives and the topological relationships between features. The Spatial Schema employs the B-Rep scheme for the representation of geometric and topological objects. The Unified Modelling Language (UML) is consistently being used as conceptual modelling language to describe the spatial concepts in each model as well as their properties, operations, and associations in object-oriented terms.

The following figure 155 illustrates the most important conceptual entities of the ISO 19107 *geometry model* arranged in UML class hierarchy.

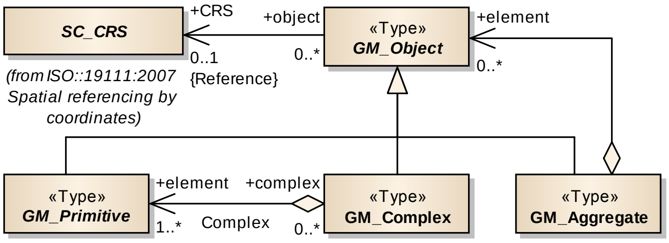


Figure 155: The most important conceptual entities of the ISO 19107:2003 geometry model.

GM_Object[33] is the abstract root class of the geometric taxonomy and represents a geographically referenced geometry object. It is defined to be the combination of a coordinate geometry and a coordinate reference system with the latter being represented by an instance of the class *SC_CRS* specified in ISO 19111:2007. A geometry object can either be a primitive geometry (*GM_Primitive*), a structured complex of primitives (*GM_Complex*) or an unstructured collection of geometry objects (*GM_Aggregate*). Any geometric object that is used to describe a geographic feature is hence a single primitive or a (structured) collection thereof which makes the primitives the basic building blocks of the geometry model.

Geometric primitives are distinguished into *points* (*GM_Point*), *curves* (*GM_Curve*), *surfaces* (*GM_Surface*), and *volumetric objects* (*GM_Solid*) (cf. figure 156). Conceptually, a geometric primitive is a single, connected, homogeneous element of a metric space that is not further decomposed into other primitives and that presents information about a geometric configuration.[34] The ambient space is usually assumed to be 2-dimensional or 3-dimensional Euclidean space. The 0-dimensional point primitives are simply given by a coordinate tuple. Curves are continuous images of the open unit 1-ball $\mathbb{B}^1$ and are composed of one or more 1-dimensional continuous *curve segments* each of which results from a set of control points and an associated interpolation method. The control points need not be contained in the point set covered by the curve and are not described through *GM_Point* objects. Supported interpolation methods include linear, geodesic, circular, clothoid, and splines, which facilitates the description of straight lines up to freeform curves. Surfaces are the basis for 2-dimensional geometry, and are defined to locally represent a continuous image of $\mathbb{B}^2$. Similar to curves, they are composed of one or more connected *surface patches* which may differ in the applied interpolation and definition methods. Besides planar and curved surface patches, again freeform patches are supported. Finally, the 3-dimensional solid primitives are given by continuous images of the interior of an n-holed toroid and are enforced to live in three dimensions.

[33] By convention, the names of classes belonging to the geometry model are prefixed with “*GM_*”, whereas classes from the topology model receive the prefix “*TP_*”.

[34] Note that in ISO 19107 the concept of a geometric primitive does not imply that it is not decomposable. A geometric primitive is rather a behavioural label. It is such by its use and behaviour, not by its structure. Whereas “*not decomposable*” is a structural imperative, primitives are simply “*not decomposed*” which represents a choice. In fact, most geometric primitives are decomposable infinitely many times (ISO 19107:2003).

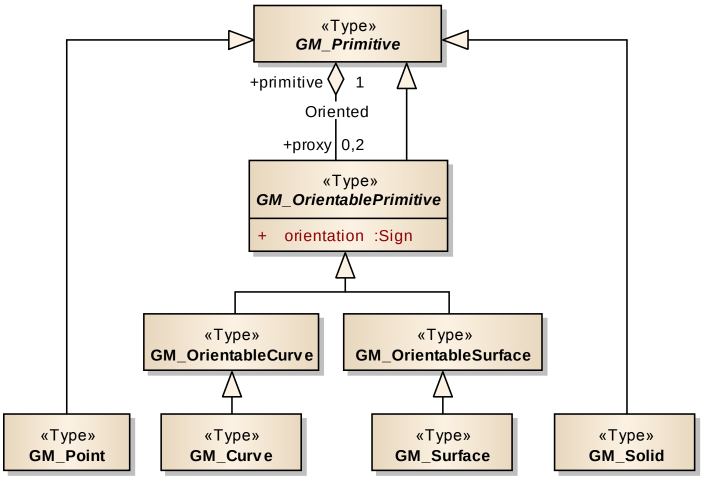


Figure 156: The geometric primitives of ISO 19107:2003.

Curves and surfaces are *oriented geometries*, i.e. they have an orientation. For curves, the orientation reflects the direction in which the curve is traversed. The positive orientation is defined to be the direction following from the curve parameterization (i.e., from its start to its end), and the negative orientation denotes the reverse direction. Surfaces are orientable if two opposite sides (a “top” side and a “back” side) can be distinguished, with the positive orientation in the direction of its upward normal. In order to give a primitive an orientation, the abstract class *GM_OrientablePrimitive* provides an *orientation* flag. *GM_OrientablePrimitive* instances reference an orientable base primitive and denote through a “+” or “-” sign whether the oriented primitive agrees or disagrees with the orientation of the referenced object. In the latter case, the orientation of the referenced primitive has to be reversed. The classes *GM_Curve* and *GM_Surface* are transitive subclasses of *GM_OrientablePrimitive* and per definition reflect the positive orientation (e.g., a *GM_Curve* “is a” positive oriented *GM_OrientableCurve*). A direct consequence of oriented primitives is that non-orientable surfaces such as the Möbius strip or the Klein bottle (cf. appendix A.5.2) are not covered by ISO 19107.

Every instance of *GM_Object* can be *simple* or *non-simple*. Being simple implies that the geometric object does not have self-intersections or self-tangencies. Formally, each interior point of the geometric object must have an open neighbourhood whose intersection with the object is homeomorphic to the open unit n-ball $\mathbb{B}^n$, with n being the dimension of the object.[35] A *GM_Object* is said to be a *cycle* if its boundary is the empty set. It follows that the geometric primitives of the Spatial Schema are more expressive compared to the geometric-topological models discussed in chapter 4.2. For example, non-manifold geometric configurations such as self-intersecting and self-tangent (and thus non-simple) curves are supported. Likewise, closed curves (i.e., circles) as well as closed surfaces (i.e., n-holed tori) can be represented through a single instance of *GM_Curve* and *GM_Surface* being cycles.

The boundaries of geometric objects are represented through subtypes of the abstract class *GM_Boundary* as illustrated in the following figure 157. Since a boundary may consist of more than one primitive, *GM_Boundary* itself is a subtype of *GM_Complex*. Boundary objects are necessarily cycles. Whilst point primitives have no boundary, instances of *GM_CurveBoundary*, *GM_SurfaceBoundary*, and *GM_SolidBoundary* provide the boundary of the respective primitive. It is important to note that, per definition, the point set described by a geometric primitive does not contain its boundary points. In contrast, each subtype of *GM_Primitive* provides a *boundary()* operator that returns the boundary through a corresponding subtype of *GM_Boundary* (cf. figure 158).

[35] Note that the ISO 19107 specification uses the notion of an n-sphere in the definition of simple geometric objects instead (cf. 19107:2003, chapter 6.2.2.6, p. 27), however without properly stating what a sphere topologically means. In order to be consistent with the definition of an n-sphere as boundary of a closed $(n + 1)$-ball as given in appendix A.2, the n-sphere has been replaced with the open n-ball here.

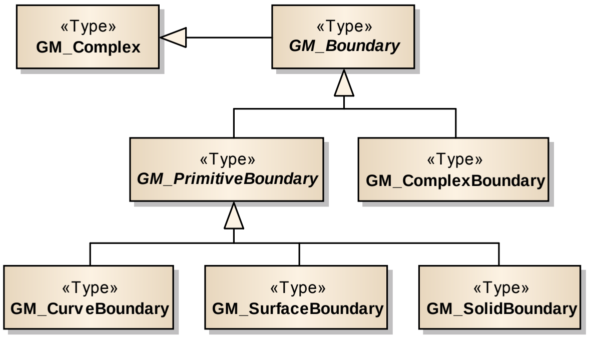


Figure 157: The hierarchy of the boundary classes.

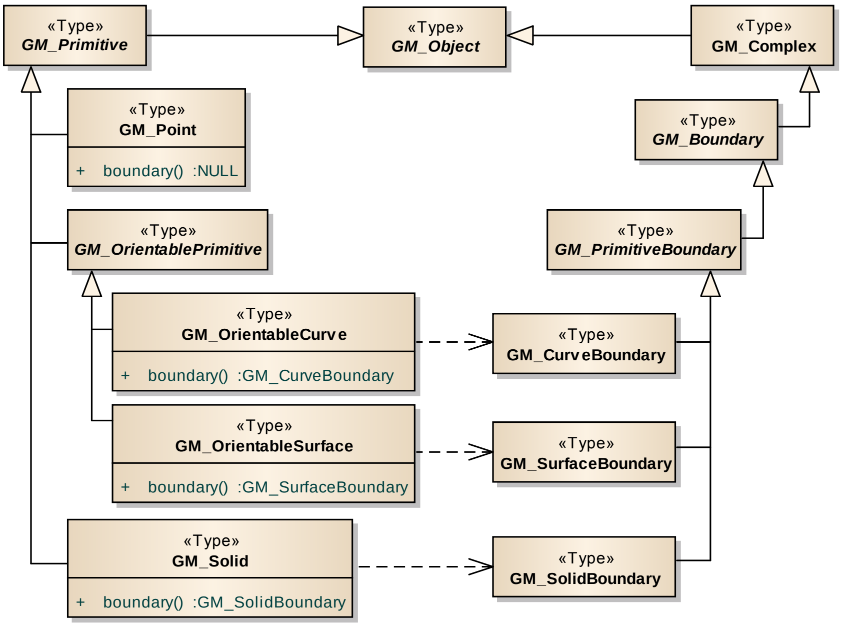


Figure 158: The boundary() operator of the geometric primitives returns an instance of a specific boundary class.

The boundary of a curve consists of two *GM_Point* primitives that denote the start and the end point of the curve and that may be coincident. For the description of the boundary of surfaces and solid objects, two more geometric primitives are introduced, namely *GM_Ring* and *GM_Shell* (cf. figure 159 and chapter 4.1). A *GM_Ring* is defined to be a sequence of consistently oriented curve primitives connected in a cycle, and each ring needs to be simple. The boundary of a surface may have various components each of which is reflected by a separate ring. Typically, one of these rings can be identified to be the exterior boundary of the surface (i.e., the closed manifold that separates the bounded surface from the unbounded component according to the Jordan-Brouwer separation theorem). In this case, further interior boundary rings separate the surface from interior holes. In case no exterior boundary can be identified, all boundaries are listed as interior boundaries. Examples for this are the 2-dimensional cylinder surface or the universal face which thus is supported by the Spatial Schema. The orientation of the curve primitives contained in a ring needs to be consistently chosen so that the interior of the surface is on the left of the curve with respect to its direction. Thus, when looking from the direction of its positive orientation, the exterior boundary of a *GM_Surface* appears counterclockwise and interior boundaries necessarily clockwise. Although each *GM_Ring* is required to be simple, the entire boundary of the surface represented by *GM_SurfaceBoundary* needs not be simple (e.g., boundary rings are allowed to be tangent).

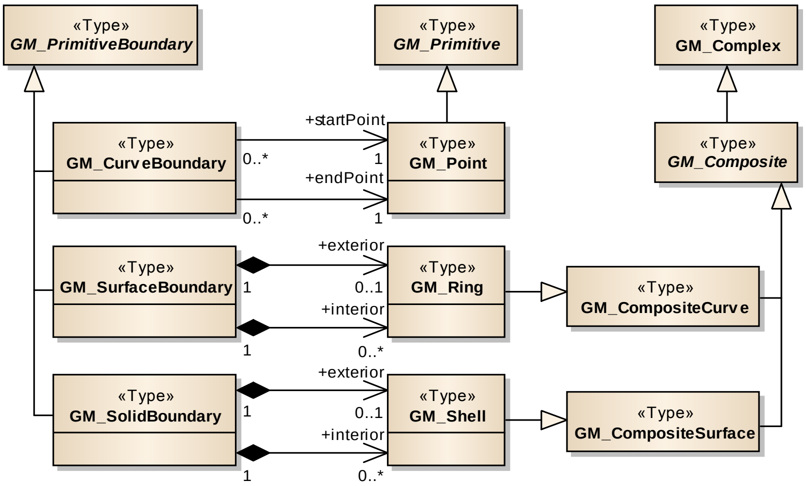


Figure 159: The description of the boundaries of the geometric primitives.

Similar to the boundary of surfaces, the *GM_SolidBoundary* of a solid object consists of one or more *GM_Shell* instances with each *GM_Shell* being a set of consistently oriented surface primitives connected in a cycle. The orientation of each surface contained in a shell needs to be chosen so that its upward normal points away from the solid's interior (i.e., the solid volume is below the surface). In case the solid is bounded, one *GM_Shell* can be identified to be the exterior boundary with zero or more interior boundaries enclosing interior voids of the solid. The Spatial Schema also supports the representation of the unbounded universal solid which only has interior shells. Like with rings, each boundary shell needs to be simple. This however does not imply that the *GM_SolidBoundary* itself is simple. Moreover, note that the modelling of exterior and interior shells respectively rings supersedes the need to have bridge faces and edges.

Geometric objects can be gathered in finite sets represented through *GM_Aggregate* and *GM_Complex* (cf. figure 155). A *GM_Aggregate* is an unstructured set of *GM_Object* instances which does not impose any constraints on the geometric configuration. Thus, the contained geometric objects may be of any dimension and may geometrically intersect, overlap, or even be equal to each other. Aggregate geometries hence allow the modelling of singularities. The subtype *GM_MultiPrimitive* and its specializations further restrict the elements of the aggregate to be geometric primitives of the same dimension as shown in the following figure.

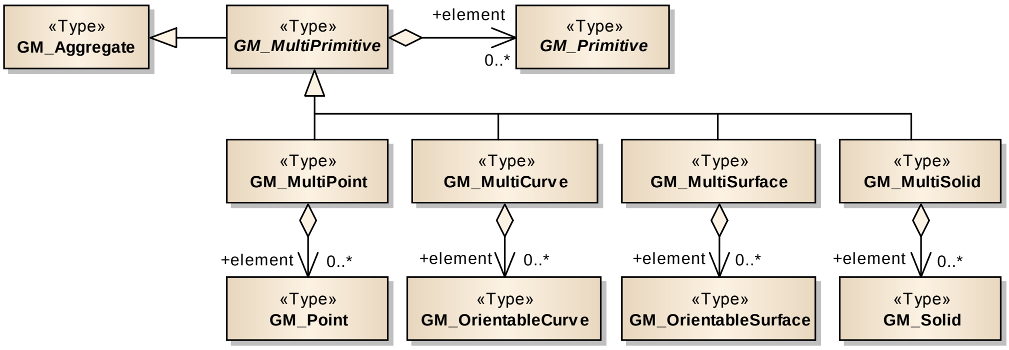


Figure 160: The subtypes of GM_Aggregate and their relation to the geometric primitives.

A *GM_Complex* is a structured set of geometric primitives (sharing the same coordinate reference system) that are enforced to be simple and geometrically disjoint. Moreover, a *GM_Complex* must also contain the elements forming the boundary of every primitive in the geometric complex.[36] The simplest n-dimensional complex contains a single *GM_Primitive* of dimension n as well as all lower dimensional primitives in its boundary. Thus, there is a subtle but important difference in the semantics of a *GM_Primitive* and a *GM_Complex*. For example, whereas a

[36] The boundary of a primitive in a geometric complex is said to be a subcomplex of that complex.

GM_Curve primitive has to reference its boundary points but does not contain them, a curve represented as 1-dimensional *GM_Complex* also references its boundary points but additionally contains them.[37]

A more restrictive subtype of *GM_Complex* is given by the abstract class *GM_Composite*. A geometric composite is a complex whose underlying core geometry must be isomorphic to a single geometric primitive. It is generated from a homogeneous collection of primitives having the same dimension and includes their boundaries. The Spatial Schema introduces an instantiable subtype of *GM_Composite* for each dimension. Since composite geometries are built from primitives but at the same time act as a single primitive, these subtypes are also derived from the corresponding geometric primitive classes as depicted in figure 161. This multiple inheritance increases the semantic ambiguity. The primary semantics of a *GM_Composite* is that of a geometric complex and hence includes the boundary. When used in interfaces inherited from *GM_Primitive*, it may also be used as a primitive object which shall not contain its boundary. It is obvious that both representations are topologically different.

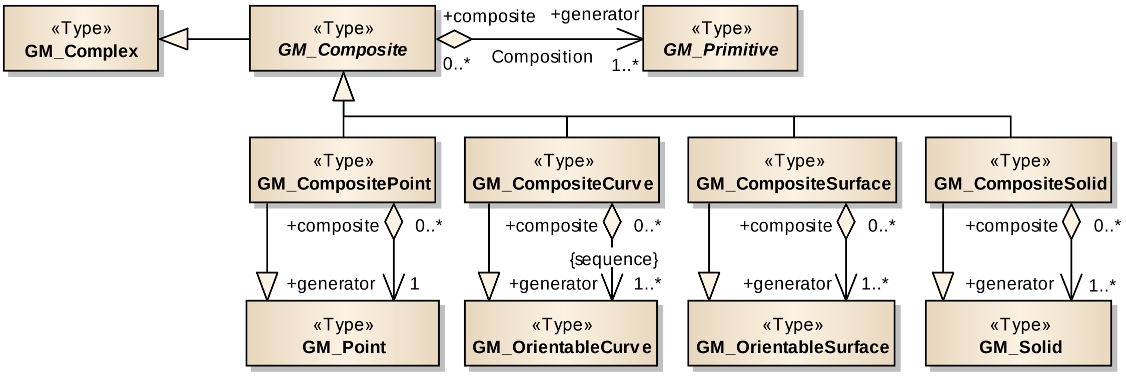


Figure 161: The subtypes of GM_Complex and their relation to the geometric primitives.

The *topology model* of ISO 19107 parallels the presented geometry model. It facilitates the representation of topological complexes as purely combinatorial structures built from topological primitives. The topological primitives are called *TP_Node*, *TP_Edge*, *TP_Face*, and *TP_Solid* and are equivalent to the geometric primitives of the respective dimension. They are subtypes of *TP_Primitive* and represent the non-decomposed building blocks of the topology model. *TP_Primitive* itself is derived from the abstract class *TP_Object* being the root of the topological taxonomy and whose only other subtype is *TP_Complex*. A *TP_Complex* is a structured set of primitives and corresponds to a geometric complex.[38]

The relation between the topology and the geometry model is sketched in the following figure 162. Normally, a topological complex is constructed from and geometrically *realized* through a geometric complex (cf. *Realization* association). In this case, each topological primitive contained in the topological complex is in a dimension-preserving, one-to-one correspondence with a geometric primitive of the same geometric complex, and thus is likewise geometrically *realized* through this geometric primitive. The geometric primitives including their boundary thus fulfil the requirements of *geometric carriers* for the topological primitives as introduced in chapter 3.1.1.3 (cf. definition 3.3 and the related discussion). And similarly, the geometric complex is to be seen as the geometric carrier of the topological complex. However, the Spatial Schema also allows a topological complex to be modelled independently from a geometric realization and to be assigned to a geographic feature as its spatial representation. Then this topological complex may ignore geometric constraints but relate features independently of their coordinate geometry. Conversely, a geometric configuration need not be equipped with a topological structure which then, however, requires purely computational geometry algorithms to answer topological queries (e.g., to retrieve

[37] According to ISO 19107, complexes are to be used if the sharing of geometry is important (cf. ISO 19107:2003, chapter 6.6.2.1, p. 93). Since primitives only return their boundary upon request through their *boundary()* operator, the returned *GM_Boundary* is always geometrically equivalent, but not necessarily contains the identical instances of *GM_Primitive* from a computational point of view. Thus, persistently referencing the boundary (e.g., if it is shared by two primitives) is not possible. This is different for geometric complexes since the boundaries are explicitly contained in the complex.

[38] Both logically and structurally, topological and geometric objects could even share the same subclass structure (cf. ISO 19107:2003, p. 101). It is handled differently in ISO 19107 in order to be able to define different properties and operators for both type hierarchies.

the boundary or topological relationships). This modelling flexibility of the Spatial Schema exceeds the expressivity of all geometric-topological models presented in chapter 4.2.

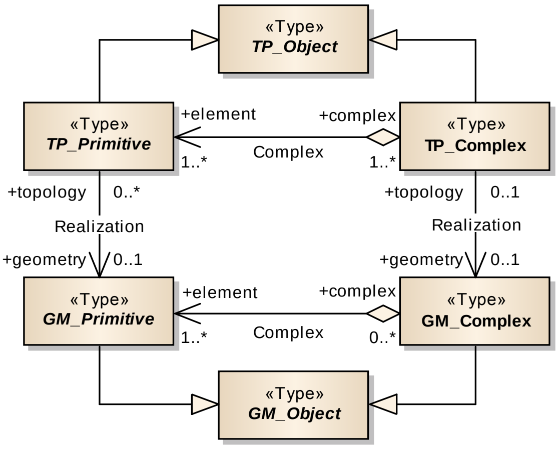


Figure 162: The relation between geometric and topological objects.

As shown in the above UML diagram, each *TP_Primitive* must belong to at least one *TP_Complex*.[39] This differs from the geometry model where a *GM_Primitive* may more loosely exist without a *GM_Complex*. Thus, the topology model of ISO 19107 always requires the modelling of topological complexes because a *TP_Complex* not only contains the topological primitive in question but also the elements on its boundary. A topological complex may only be realized geometrically by exactly one geometric complex. If it is realized, then all of its primitives must be associated with precisely one *GM_Primitive* from the complex geometry. If a single topological primitive shall not be realized, then the entire *TP_Complex* it belongs to as well as all of its contained primitives may not have a geometric realization. In contrast, since a geometric primitive may belong to different and decoupled geometric complexes, it may be the realization of different topological primitives. This allows for defining multiple and different topological structures on the same geometric configuration.

Equivalent to the semantics of *GM_OrientablePrimitive*, every *TP_Primitive* has a positive and a negative orientation. The abstract class *TP_DirectedTopo* and its subtypes are used to model directed primitives by denoting their orientation attributively through a “+” respectively a “-” sign. The following figure shows the relation between the topological primitives and their counterparts derived from *TP_DirectedTopo*. Note that every primitive is associated to two directed proxies representing its positive and negative orientation. Moreover, and similar to the geometry model, it is at the same time a subtype of the corresponding directed topological primitive and semantically represents the positive orientation. For example, a *TP_Face* is a subtype of *TP_DirectedFace* and equivalent to a positive directed *TP_DirectedFace*. The orientation of a *TP_Primitive* is further required to be consistent with the orientation of the *GM_Primitive* that realizes it.

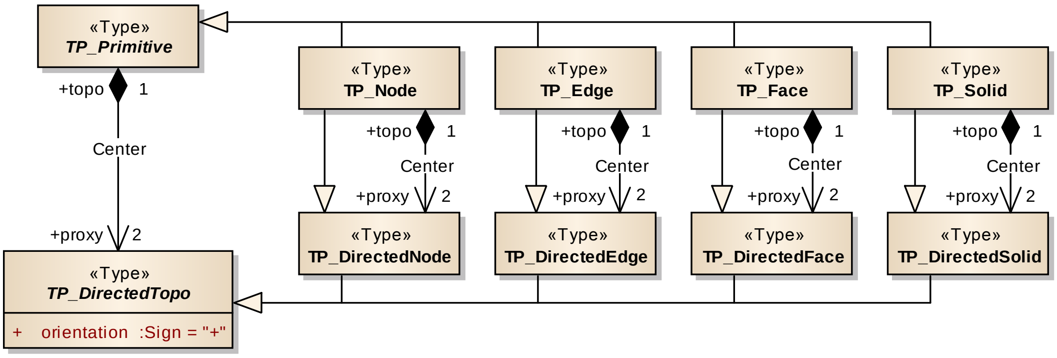


Figure 163: The topological primitives and their directed counterparts.

[39] In fact, it may only belong to precisely one *maximal* topological complex. A complex is said to be maximal if it is not a subcomplex of a larger complex.

The orientation is relevant for properly describing the boundary and coboundary of the topological primitives. The *boundary()* operator of each primitive returns an instance of *TP_EdgeBoundary*, *TP_FaceBoundary*, or *TP_SolidBoundary*, respectively an empty set for *TP_Node*. The definition of the boundaries as well as the orientation of the contained topological primitives follows the same semantics as in the geometry model. For example, *TP_Ring* and *TP_Shell* are used to represent the exterior and interior boundary components of faces and solids. In addition, the *coBoundary()* operator returns a set of those *TP_DirectedTopo* instances which have the primitive on their boundary. For example, the coboundary of a *TP_Face* returns a set of directed solids representing the solids whose boundary contains the face. Whereas the boundary of a topological primitive has to be of dimension one less than that of the primitive, the dimension of the coboundary needs to be one dimension higher (but is undefined for solids). The coboundary information is especially useful to evaluate topological adjacency relationships between primitives. Figure 164 presents the corresponding UML representation.

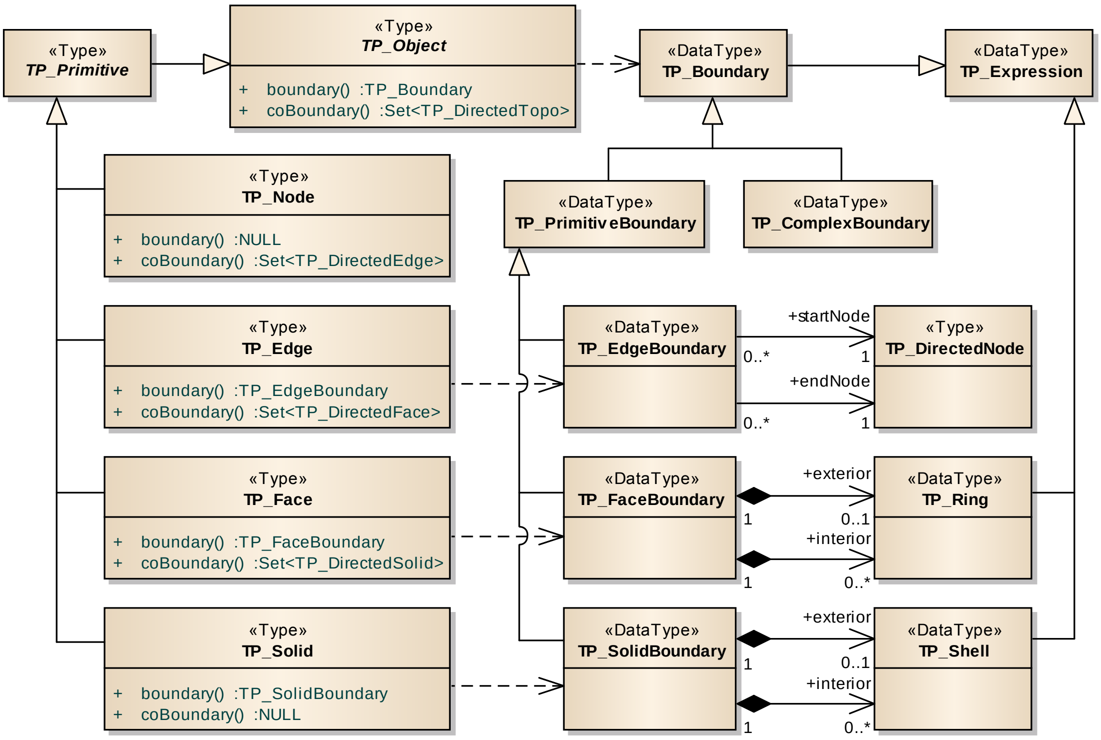


Figure 164: The boundary() operator of the topological primitives returns an instance of a specific boundary class. The coBoundary() operator returns an unstructured set of directed topological primitives.

A *topological complex* in the sense of the ISO 19107 Spatial Schema may not be confused with a *cell complex* as defined in algebraic topology (cf. definition A.48). Most importantly, the n-cells participating in an algebraic cell decomposition are required to be homeomorphic to the open unit n-ball $\mathbb{B}^n$ (cf. definition A.46) whereas the topological primitives in ISO 19107 may also be homeomorphic to multiply-punctured open n-balls without the need for additional bridge edges or faces, and even may be non-compact. While these aspects exceed the expressivity of algebraic cell complexes, the fact that an n-dimensional topological primitive has to go in $(n-1)$ dimensions on its boundary is a restriction of the mathematical notion which implies one consequence with respect to the MLSEM. Precisely, consider a single 3-dimensional space cell being homeomorphic to the closed 3-ball $\overline{\mathbb{B}}^3$. Then the minimal cell decomposition of the boundary 2-sphere is given by one 2-cell and one 0-cell (cf. figure 69b in chapter 3.1.3). However, in ISO 19107, it takes a minimum of two *TP_Face* instances (hemispheres) which meet at a common *TP_Edge* whose boundary is a single *TP_Node* in order to describe the 2-sphere. Note that this restriction also holds for the alternative approaches to geometric-topological data modelling presented in chapter 4.2 most of which are even further limited to regular cell decompositions.

There is also another important difference between the Spatial Schema and the alternative data models which strongly supports the ideas of the MLSEM. Remember that a single *TP_Primitive* is geometrically realized through a single *GM_Primitive*. Since composite geometries are universally subtyped (transitively) from *GM_Primitive*, every *TP_Primitive* may likewise be realized through a *GM_Composite*. For example, a single *TP_Face* suffices to represent a structured set of geometric surfaces if this set renders a *GM_CompositeSurface*. Put more generally, a connected sequence of $(n-1)$-dimensional geometric primitives on the boundary of an n-dimensional spatial object may be topologically captured by a single $(n-1)$-dimensional *TP_Primitive* object. Since, within the MLSEM, the number of $(n-1)$-dimensional boundary elements is correlated with the number of dual edges in the resulting Poincaré dual graph, this flexibility of the Spatial Schema allows for reducing the number of dual edges without affecting the geometric configuration of the spatial object (cf. related discussion in chapter 3.1.3). None of the presented alternative geometric-topological data models provides this expressivity.

Based on the discussions in this chapter, it follows that the ISO 19107 Spatial Schema facilitates the modelling of the primal space geometry of 2-dimensional and 3-dimensional space cells through simple *GM_Surface* respectively *GM_Solid* primitives by restricting them to render connected, compact manifolds with boundary as required by conditions (i) and (ii) of definition 3.2. In order to describe the space cell complex associated with a space layer, the geometric primitives can be combined in a *GM_Complex* structure which also supports non-manifold geometric configurations of space cells. The definition of a *GM_Complex* hereby naturally conforms to definition 3.7 according to which the space cells participating in a space cell complex must be mutually non-overlapping. Moreover, the Spatial Schema is strong enough to also represent the non-compact geometry of the outer space cell complementing a space cell complex through a single geometric primitive by simply listing the boundary components of the space cell complex as interior *GM_Ring* respectively *GM_Shell* objects in the boundary of the outer space cell. Since only single *GM_Ring* and *GM_Shell* objects need to be simple but not the boundary of the primitive itself, non-manifold spaces underlying the outer space cell can also be described. The primal space topology of the space cells is then provided in a decoupled way using topological primitives (*TP_Face* respectively *TP_Solid*) being arranged in a possibly non-manifold *TP_Complex* that topologically represents the space cell complex on a given space layer. Hence, this *TP_Complex* is geometrically realized through the corresponding *GM_Complex* that also provides the geometric carriers for each individual topological primitive.

When restricting the dimension of geometric and topological complexes to one, the Spatial Schema can likewise be used to represent graph structures in geometry and topology space. Thus, a 1-dimensional *TP_Complex* is feasible to model the dual space topology of a space layer as given in definition 3.20. The topological primitives in the primal and dual *TP_Complex* of a space layer are then in a dimension-reversing, one-to-one correspondence reflecting the Poincaré duality. The *TP_Node* primitives contained in the dual complex hence represent the dual space topology of the single space cells, whereas each *TP_Edge* reflects a space cell boundary. Note that the missing bridge edges and faces in primal space do not impact this bijective pairing since their dual counterparts are excluded from the dual space representation according to condition (ii) of definition 3.20. The *TP_Complex* capturing the intra-layer graph of a space layer can then be geometrically embedded through a decoupled 1-dimensional *GM_Complex* which satisfies the requirements for the dual geometry space representation of a space layer as imposed by definition 3.21. Each *TP_Edge* primitive in the dual *TP_Complex* may again be realized through a composite collection of space curves being isomorphic to a single curve primitive. This facilitates the modelling of fine-grained and precise geometric space trajectories and travelling paths in dual geometry space based on *GM_CompositeCurve* objects (cf. 3.1.2.3) whilst the purely topological adjacency information is captured by a single *TP_Edge*.

In summary, the spatial data model introduced in ISO 19107 provides the necessary means to model the geometric-topological representation of space cells and space layers in both primal and dual space. Moreover, its spatial expressiveness covers the whole range of geometric-topological configurations of space cells and space cell complexes presented in chapter 3.1.3. In the following figure, the four quadrants of the conceptual geometric-topological space representation of space cells and space layers as introduced in figure 41 and figure 42 are augmented with corresponding spatial model elements from the ISO 19107 Spatial Schema.

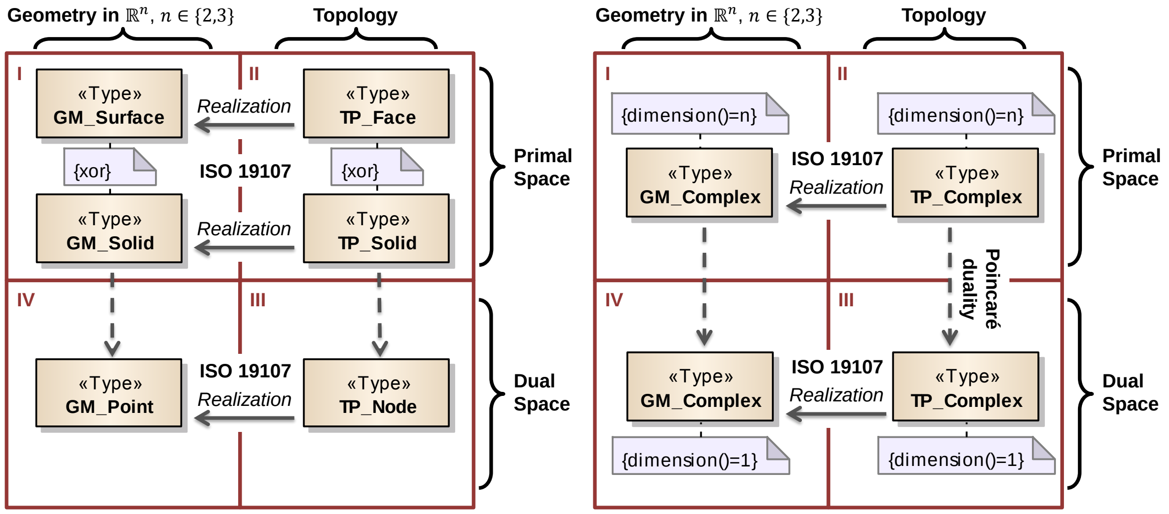


Figure 165: Using ISO 19107 concepts to model the four quadrants of the geometric-topological representation schema of space cells (left) and space layers (right).

4.3.2 ISO 19109 – Semantic Data Modelling

Within the ISO 19100 standards family, the fundamental unit of geographic information are *features*. The ISO 19101:2002 specification defines a *feature* as abstraction of a real world phenomenon. A feature is said to be a *geographic feature* if it is associated with a location relative to the Earth. Features are *conceptual entities* that represent a real world object together with its semantics and spatial characteristics, its relationships to other features, and its behaviour. This notion is in contrast to earlier approaches to GIS data modelling that simplified geographic information through purely geometric objects being equipped with a set of thematic attributes. The ISO 19100 series thus represents a paradigm shift towards the conceptual modelling (cf. chapter 2.1.3) of geo-data by semantically classifying a view of the real world through well-defined objects (i.e., features) that may or may not be expressed spatially.

The rules and principles for defining and organizing features in a conceptual data model are provided by the *General Feature Model* (GFM) which is specified in ISO 19109:2005. The GFM is to be seen as a *metamodel* that sets the standard framework for the description and modelling of features as derived from a universe of discourse but which itself does not provide any classification of concrete real word objects. In contrast, any view of the real world is always understood to be a partial abstraction that depends on and satisfies a given application field but which does not have universal scope. A feature and its properties thus describe how a real world phenomenon is perceived in the context of a specific geographic application, but this perception may differ for different applications. A conceptual data model that adheres to the rules and principles of the GFM is therefore said to be an *application schema*.[40]

Figure 166 shows the main conceptual elements of the GFM for defining features in application schemas. The ISO 19109 uses UML as conceptual schema language for specifying the GFM itself.

[40] Besides the definition of the GFM, ISO 19109 provides standard rules regarding the derivation of application schemas. These rules address topics such as the step-wise refinement of feature definitions according to the application needs, the transition from the GFM concepts to an appropriate representation in application schemas using a standardized conceptual schema language such as UML or EXPRESS (ISO 10303-11:1994), or the integration of further standardized ISO 19100 schemas with the application schema. The detailed presentation and discussion of these rules is however outside the scope of this thesis. The reader is referred to ISO 19109:2005.

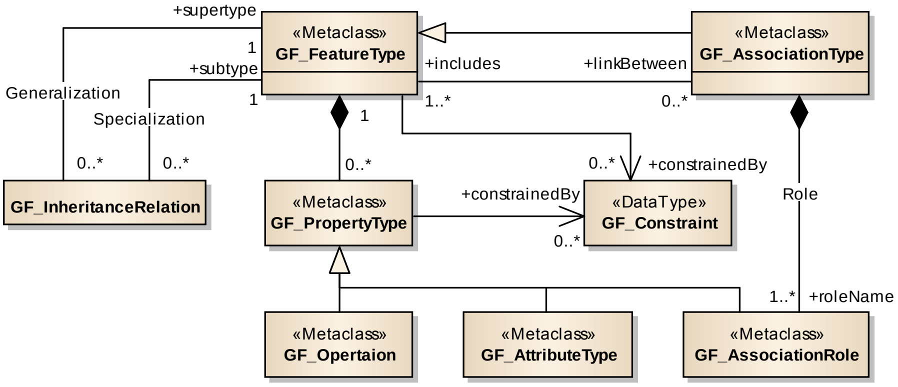


Figure 166: The main concepts of the General Feature Model as defined in ISO 19109:2005.

The central concept of the GFM is called *GF_FeatureType*.[41] A *GF_FeatureType* is a metaclass whose instances represent semantically homogeneous groups of real world phenomena. The GFM is based on object-oriented modelling principles according to which an individual *feature type* (i.e., an instance of *GF_FeatureType*) is to be seen equivalent to the concept of a *class*. The *objects* of feature classes are said to be *feature instances* with each feature instance representing a concrete real world object being a member of its feature type. Note that in ISO 19109 the term *"feature"* is used interchangeably to either reference an abstraction of a real world phenomenon independent of types and instances (meta meta level), the abstract GFM concept of a feature type (meta level), an individual feature type (application level), or a concrete feature instance (data level). Besides grouping features into types, the GFM addresses the description of attributes associated with each type (*GF_AttributeType*), the relationships among the types (*GF_InheritanceRelation* and *GF_AssociationType*), and the behaviour of features modelled as operations (*GF_Operation*). In order to ensure the integrity of data being exchanged according to an application schema, both feature types (including associations) and the properties of feature types may be subject to constraints (*GF_Constraint*) which can be expressed in terms of a constraint language such as the Object Constraint Language (OCL).

The attributes of a feature carry the relevant information about its characteristics. The *GF_AttributeType* class is a metaclass for defining a feature attribute through specifying its name, its value type or domain as well as its cardinality. The GFM distinguishes five kinds of attribute types being subtypes of *GF_AttributeType* as shown in figure 167. Temporal attributes (*GF_TemporalAttributeType*) are used to provide a time reference for a feature as either point in time or time interval according to the temporal schema specified in ISO 19108:2005. The spatial characteristics of features are described through spatial attributes (*GF_SpatialAttributeType*) whose value domain is restricted to *GM_Object* and *TP_Object* from ISO 19107:2003. Non-geometric location information about features (e.g., identifiers such as names or addresses) can be expressed through location attributes (*GF_LocationAttributeType*) being specified in ISO 19112:2003. The provision of metadata about features or their attributes is facilitated through metadata attributes (*GF_MetadataAttributeType*) in accordance with ISO 19115:2003. Finally, any other descriptive characteristic information about features is called a thematic attribute (*GF_ThematicAttributeType*) and can be defined according to user or application needs independent of an existing specification from the ISO 19000 standards series.

[41] The prefix "*GF_*" is consistently being used within ISO 19109 to mark the classes from the General Feature Model.

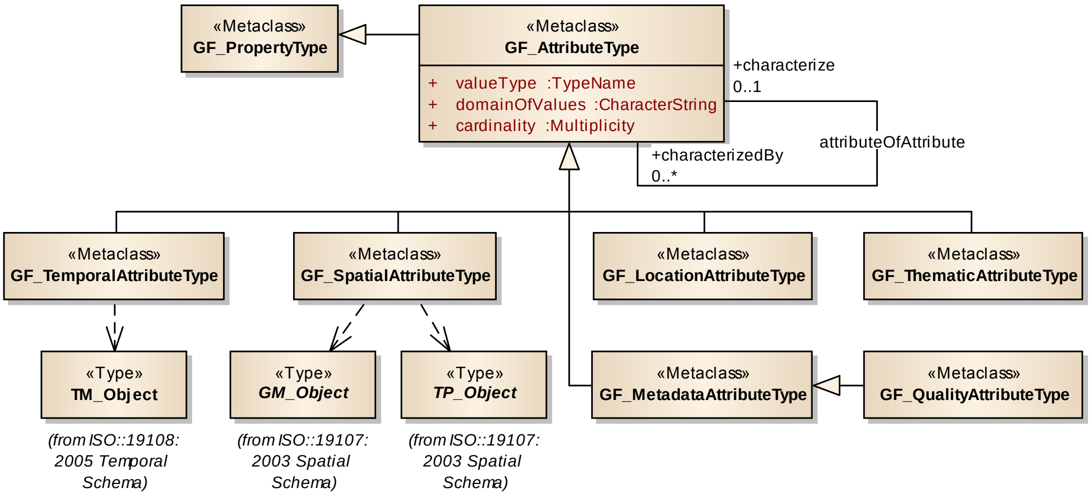


Figure 167: The subtypes of GF_AttributeType (excerpt).

Relationships between feature types include semantic classification hierarchies (*GF_InheritanceRelation*) that render taxonomies by structuring feature types into supertypes and subtypes. General semantic associations between feature types such as *has-a* and *part-of* relationships are modelled through instances of the metaclass *GF_AssociationType* which itself is a subtype of *GF_FeatureType* and hence may carry additional attributes. The role a feature takes in an association is specified as feature property by means of the metaclass *GF_AssociationRole*. Associations allow for modelling semantic decomposition hierarchies between features. Thus, the whole feature (e.g., a building) but also its semantic parts (e.g., storeys, floors, room, corridors, etc.) can be expressed as individual feature types and put in appropriate associations. The resulting semantic structure can even be paralleled by a corresponding spatial structuring of the features. This dual structure is inherently implied by ISO 19109. For example, a building feature type can be assigned a *GM_Complex* (or *TP_Complex*) as spatial representation, whereas its (nested) semantic parts are spatially described through (nested) subcomplexes of that *GM_Complex* (or *TP_Complex*). In literature, the correspondence between the semantic and the spatial decomposition of features and spatial objects is discussed under the term *spatio-semantic coherence* (cf. Stadler & Kolbe 2007, Kolbe 2009). Often, existing data models in GIS focus on a sound geometric-topological representation of spatial objects but neglect the coherent representation of semantics. This is especially true when applying spatial representation schemes such as the spatial occupancy enumeration or triangulations respectively tetrahedralizations of geometric objects. In such cases, a semantic feature is spatially decomposed into smaller parts which however do not reflect its structuring into individual semantic parts. However, and as discussed by (Stadler & Kolbe 2007), a coherent structuring is beneficial in terms of knowledge inference and reasoning because every semantic entity “knows” its spatial location and extent, whereas every spatial object “knows” its semantic meaning.

The GFM offers a comprehensive framework for the spatio-semantic description of real world entities which exceeds the semantic modelling capabilities of the alternative data models discussed in chapter 4.2. In the context of the MLSEM, space cells, boundary cells, and space layers have been introduced as the core conceptual entities for the representation of interior spaces, and their semantic dimension is an essential aspect of their mathematical definitions (cf. definition 3.1, definition 3.24, and definition 3.19). Whereas the spatial characteristics of the MLSEM are fully expressible through the Spatial Schema as illustrated in the previous chapter, the concepts introduced by the GFM provide the necessary means to also capture and model its semantic dimension. The ISO 19100 standards family is hence a natural choice for defining a conceptual data model for the MLSEM.

4.4 Conceptual Data Model of the Multilayered Space-Event Model

The ISO 19100 compliant realization of a conceptual data model for the MLSEM is presented in the following sections. The data model proposed here is based on a first draft presented by (Becker et al. 2009b) and (Nagel et al. 2010) (cf. appendix D). In the course of this thesis, this draft has been substantially reworked and augmented by additional model elements in order to fully cover the spatio-semantic expressivity of the MLSEM. For example, the initial data model did not foresee the geometric-topological representation of space cells and space layers in both two and three dimensions but is rather restricted to the latter case. This restriction not only limits the possible applications of the MLSEM but, more importantly, is not justified from its mathematical definition as developed in this thesis and thus has been resolved. Moreover, missing conceptual entities (e.g., regarding the modelling of space layer complexes as well as joint states and their transitions) have been added as well as the possibility to link instances of the MLSEM to their original data sources (e.g., entities in IFC or CityGML models) and to represent individual navigation routes enriched with guidance information. A further fundamental extension addressing the modelling of navigation constraints is discussed and presented in its own chapter 5.

4.4.1 MLSEM Application Schema

The conceptual data model of the MLSEM is specified as *application schema* in the sense of ISO 19109 and is formally described using UML as conceptual modelling language. Figure 168 shows the organization of the MLSEM schema in terms of UML packages. The overall data model is split into seven leaf packages which cover separate conceptual aspects of the MLSEM schema and which are called *Space Representation*, *Joint States*, *Source Object*, *External Reference*, *Groups and Sequences*, *Route*, and *Model Linkage*. The package *Space Representation* renders a dependency for the other leaf packages of the MLSEM schema (except *External Reference*). A package is said to be dependent on another package if it uses structures and definitions from that package. In the UML diagram in figure 168, dependency relations are expressed as dashed arrows classified with the stereotype *«uses»* and with the arrowhead denoting the direction of the dependency (i.e., from the user to the provider). Additionally, the contents of the MLSEM schema reference elements from the ISO 19107:2003 Spatial Schema and the ISO 19108:2005 Temporal Schema. The integration with both standard schemas from the ISO 19100 series thus results in additional schema dependencies. The discussion and presentation of the MLSEM schema in the subsequent sections follows the illustrated package structure.

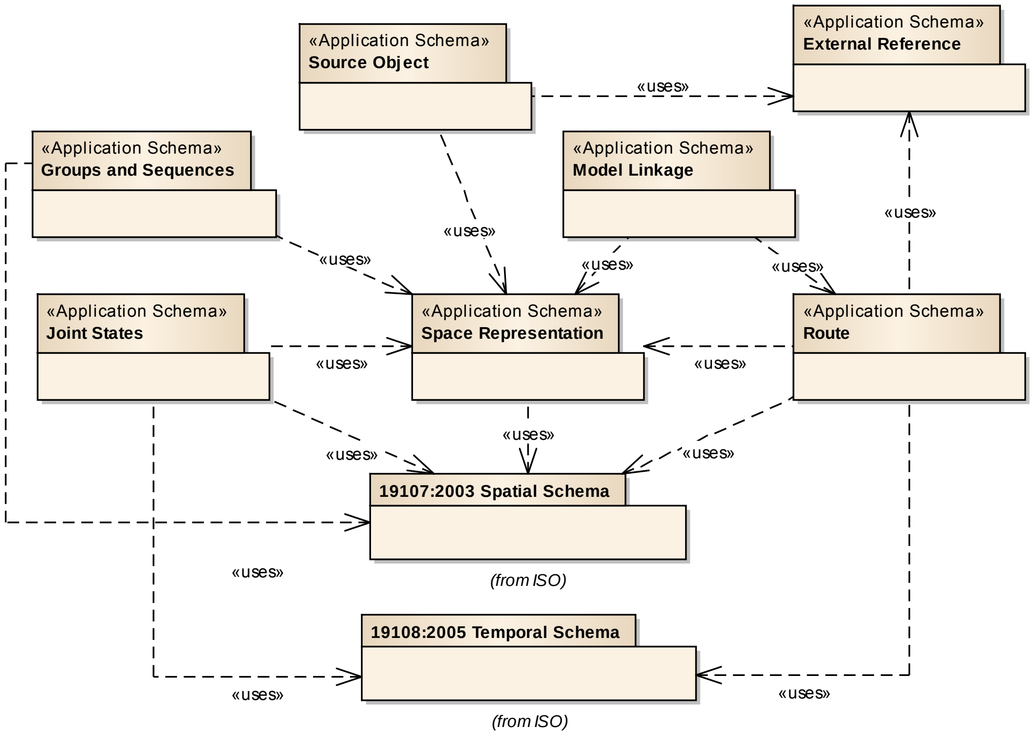


Figure 168: The packages of the MLSEM application schema and their dependencies.

4.4.1.1 Space Representation Package

The *Space Representation* package defines the basic classes for the spatio-semantic description of indoor space on different space layers in both primal and dual space. The semantic concepts of this package as well as their relationships are depicted in the UML class diagram shown in figure 169.

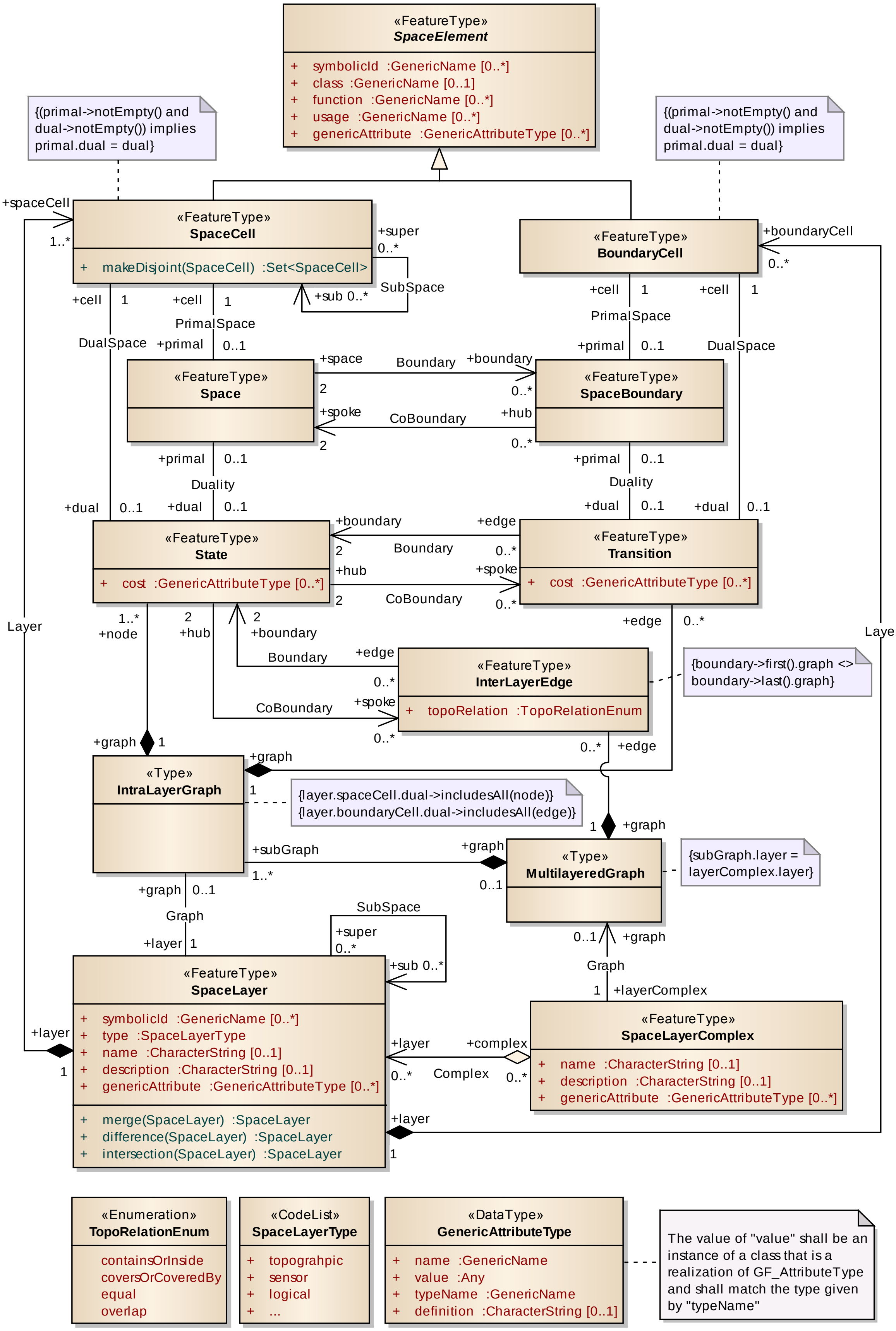

«FeatureType»
SpaceElement
+ symbolicId :GenericName [0..*]
+ class :GenericName [0..1]
+ function :GenericName [0..*]
+ usage :GenericName [0..*]
+ genericAttribute :GenericAttributeType [0..*]
{(primal->notEmpty() and dual->notEmpty()) implies primal.dual = dual}
{(primal->notEmpty() and dual->notEmpty()) implies primal.dual = dual}
+spaceCell
1..*
«FeatureType»
SpaceCell
+ makeDisjoint(SpaceCell) :Set<SpaceCell>
+super
0..*
SubSpace
+sub 0..*
«FeatureType»
BoundaryCell
+boundaryCell
0..*
+cell
1
+cell
1
PrimalSpace
DualSpace
+primal
0..1
+cell
1
+cell
1
PrimalSpace
+primal
0..1
DualSpace
«FeatureType»
Space
+space
2
Boundary
+boundary
0..*
«FeatureType»
SpaceBoundary
+spoke
2
CoBoundary
+hub
0..*
+primal
0..1
Duality
+dual
0..1
+dual
0..1
+primal
0..1
Duality
+dual
0..1
+dual
0..1
«FeatureType»
State
+ cost :GenericAttributeType [0..*]
+boundary
2
Boundary
+edge
0..*
«FeatureType»
Transition
+ cost :GenericAttributeType [0..*]
+hub
2
CoBoundary
+spoke
0..*
Layer
1..*
+node
2
+hub
2
+boundary
Boundary
+edge
0..*
«FeatureType»
InterLayerEdge
+ topoRelation :TopoRelationEnum
{boundary->first().graph <> boundary->last().graph}
CoBoundary
+spoke
0..*
+edge
0..*
Layer
+graph
1
«Type»
IntraLayerGraph
+graph
1
{layer.spaceCell.dual->includesAll(node)}
{layer.boundaryCell.dual->includesAll(edge)}
0..*
+edge
1
+graph
+subGraph
1..*
+graph
0..1
«Type»
MultilayeredGraph
{subGraph.layer = layerComplex.layer}
+graph
0..1
Graph
+layer
1
SubSpace
+super
0..*
+sub 0..*
0..1
+graph
Graph
1
+layerComplex
«FeatureType»
SpaceLayer
+ symbolicId :GenericName [0..*]
+ type :SpaceLayerType
+ name :CharacterString [0..1]
+ description :CharacterString [0..1]
+ genericAttribute :GenericAttributeType [0..*]
+ merge(SpaceLayer) :SpaceLayer
+ difference(SpaceLayer) :SpaceLayer
+ intersection(SpaceLayer) :SpaceLayer
+layer
1
+layer
0..*
Complex
+complex
0..*
«FeatureType»
SpaceLayerComplex
+ name :CharacterString [0..1]
+ description :CharacterString [0..1]
+ genericAttribute :GenericAttributeType [0..*]
+layer
1
«Enumeration»
TopoRelationEnum
containsOrInside
coversOrCoveredBy
equal
overlap
«CodeList»
SpaceLayerType
+ topograhpic
+ sensor
+ logical
+ ...
«DataType»
GenericAttributeType
+ name :GenericName
+ value :Any
+ typeName :GenericName
+ definition :CharacterString [0..1]
The value of "value" shall be an instance of a class that is a realization of GF_AttributeType and shall match the type given by "typeName"

Figure 169: The semantic concepts of the Space Representation package.

Semantic concepts. The MLSEM application schema structures the semantic view on indoor space along the two main conceptual entities *SpaceCell* and *BoundaryCell*. Both classes are realizations of the GFM metaclass *GF_FeatureType* which is consistently indicated through the stereotype *«FeatureType»* in all leaf packages of the

MLSEM schema. The *SpaceCell* feature type carries the notions of a *space cell* (cf. definition 3.1) and the *outer space cell* (cf. definition 3.11) from the formal mathematical model of the MLSM into the conceptual data model. Likewise, the *BoundaryCell* feature type is the equivalent of the formal notion of a *boundary cell* (cf. definition 3.24) within the data model. Both *SpaceCell* and *BoundaryCell* are subtypes of the abstract feature type *SpaceElement* which provides their common abstraction.

The *SpaceCell* entity is related to the feature types *Space* and *State* through the *PrimalSpace* and *DualSpace* associations. Whereas a *Space* feature represents the 2-dimensional or 3-dimensional partition of indoor space occupied by the space cell in primal space, the *State* feature type conceptually captures the state of physically being within the space cell and thus corresponds to its dual node representation. Both associations realize a one-to-one correspondence between the *SpaceCell* and its space representations. *Space* and *State* are themselves linked by the *Duality* association which allows for navigating between the primal and dual space description of a space cell and thus translates the formulas 3.17 and 3.18 from the mathematical model. The conceptualization of boundary cells within the data model follows the same approach. The 2-dimensional respectively 1-dimensional subset of primal space inhabited by a boundary cell is reflected through an instance of the feature type *SpaceBoundary* which at the same time semantically classifies a part of the boundary of a *Space*. The *Transition* feature type, on the other hand, captures the dual edge representation of a boundary cell and therefore expresses a topological adjacency relationship as well as the transition between two *State* features which is triggered by the movement from one space cell to another. Both space representations are similarly linked to the *BoundaryCell* via the *PrimalSpace* and *DualSpace* association, whereas the *Duality* relation between *SpaceBoundary* and *Transition* realizes their formal pairing according to the formulas 3.27 and 3.28. Hence, it likewise can be viewed as switch between primal and dual space.

The conceptual data model of the MLSEM differs from its mathematical formalization in that it introduces a *loose coupling* between the presented model elements. Precisely, a *SpaceCell* feature may exist with or without associated instances of *Space* and *State* as specified by the multiplicities on the *PrimalSpace*, *DualSpace*, and *Duality* relations. It follows that a *SpaceCell* may be represented in both primal and dual space but may likewise possess only one or even none of both space representations. The same holds for the representation of a *BoundaryCell* through *SpaceBoundary* and *Transition* features. In contrast, the mathematical notion of space cells and boundary cells builds upon the bijective pairing of both primal and dual space according to the Poincaré duality and thus renders a strict coupling. The rationale for introducing a loose coupling between the space representations is to allow the decoupled acquisition, storage, and exchange of data from either aspect of space. For example, the primal space representation of the interior built environment may be acquired from initial measurements and capturing methods or from existing building data. Either way, the dual space representation is most likely not addressed at the same time but has to be derived in a consecutive step which itself is possibly preceded by further data qualification and refinement steps (e.g., in order to ensure the geometric-topological consistency of the primal space data). The loose coupling of the model elements hence facilitates to first represent, store, and subsequently refine the primal space data using the MLSEM data model, and to add the dual space data afterwards (or vice versa). Moreover, for a given navigation application it might be sufficient to only populate the dual space dimension of space cells and boundary cells while completely neglecting an accompanying primal space model (or vice versa). And even if both space representations are fully populated, it might be beneficial to only exchange data from one or the other. For example, a mobile device may only be fed with (a subset of) the dual graph-based representation in order to meet storage and computation limits. Due to the loose coupling approach, the conceptual data model of the MLSEM offers the flexibility to support these scenarios. A navigation application applying the MLSEM schema may however be more restrictive in that it imposes a strict coupling.

The *SpaceCell* feature type inherits the thematic attributes *symbolicId*, *class*, *function*, *usage*, and *genericAttribute* from *SpaceElement* to characterize concrete feature instances.[42] For example, a topographic space region represented as *SpaceCell* entity can be classified as room, corridor, stair, elevator, or obstacle using its *class* attribute. The intended purpose of the *SpaceCell* may additionally be denoted through its *function* attribute (e.g., the function of a room might be office, laboratory, or storage), whereas a deviating real usage can be captured by the *usage*

[42] The value domain of the thematic attributes is *GenericName* which is a type defined in ISO/TS 19103:2005 that represents a name associated with a globally unique namespace. This allows for providing predefined lists of possible values for each attribute which can be referenced but also distinguished through their namespace value.

attribute.[43] The *symbolicId* property then allows for attaching one or more human-friendly descriptive location names or identifiers (e.g., the room number) and hence supports the realization of a symbolic addressing or georeferencing schema suitable for navigation queries. Since various further characteristics might be required in a navigation application, a *SpaceCell* can be augmented with an unbounded number of additional *generic attributes*. The value domain of generic attributes is specified by *GenericAttributeType* which essentially defines a name-value pair. The set of generic attributes thus directly realizes the set $A(S)$ of semantic attributes associated with a space cell S according to definition 3.6. The type of a generic attribute is constrained to be a realization of a *GF_AttributeType* and therefore comprises spatial, temporal, and thematic characteristics as well as metadata (cf. figure 167). The name of the type needs to be consistently referenced through the *typeName* attribute which enables querying generic data types without having to evaluate their class definition. Optionally, a human-readable *definition* can be provided for generic attributes. The same set of thematic attributes is also inherited by the *BoundaryCell* feature type and can be used in this context, for instance, to classify the tangible boundaries of rooms into walls, floors, and ceilings, or to mark parts of the boundary as being intangible or traversable (e.g., doors or virtual boundaries). In addition to thematic properties, the *SubSpace* association of *SpaceCell* links a space cell to its zero or more subspace cells and hence allows for mapping (transitive) inclusion relationships (cf. definition 3.50). Moreover, the *SpaceCell* feature type defines a *makeDisjoint* operation which implements the $f_{DISJOINT}$ map as developed in chapter 3.5.1 (cf. definition 3.65).

It is important to note that *SpaceCell* and *BoundaryCell* are not restricted to the description of topographic space but are used to model space partitionings following from arbitrary notions of space (e.g., sensor space or logical space). Possible classifications of *SpaceCell* and *BoundaryCell* features as well as reasonable thematic attributes obviously vary for each notion of space. For this reason, the conceptual data model of the MLSEM as presented in this thesis neither provides in-depth ontologies for space cells and boundary cells, nor predefined lists of attribute values. Although the proposed data model thus remains on a rather abstract level, it nevertheless offers the expressivity to describe arbitrary types of space cells and boundary cells (through *symbolicId*, *class*, *function*, and *usage*) as well as their semantic properties (through *generic attributes*). If required, the introduced concepts may be further subtyped according to the information needs of navigation applications. For example, (Brown et al. 2012) present an ontology for topographic space which classifies the interior built environment against indoor navigation criteria. The authors differentiate topographic regions into *IndoorSpace* (navigable end-spaces and connector spaces such as rooms and corridors), *IndoorObstacleSpace* (fixed or movable obstacles restricting movement), and *TransitionSpace* (openings providing passage between *IndoorSpace* entities such as doors, windows, or virtual entities), and define further subtypes and relationships.[44] This ontology builds upon the previous data model of the MLSEM (Becker et al. 2009b, Nagel et al. 2010) and thus can be easily hooked under the concepts of *SpaceCell* and *BoundaryCell*. On the contrary, the high degree of abstraction also allows for connecting the entities of the MLSEM data model to conceptual elements from existing information models (e.g., navigational and non-navigational indoor space models as presented in chapter 2.2) while at the same time keeping the semantic impedance low. For example, chapter 6 demonstrates how *SpaceCell* and *BoundaryCell* can be associated with entities from existing building models in order to populate topographic space data.

The *SpaceCell* and *BoundaryCell* feature types are implicitly related on the level of their primal respectively dual space representations. In primal space, this relation is expressed through the *Boundary* and *CoBoundary* associations between *Space* and *SpaceBoundary*. The *Boundary* relation enforces every *SpaceBoundary* to be on the boundary of precisely two *Space* entities. On the one hand, this constraint ensures the complete tiling of Euclidean space as formally required by the definition of space layers (cf. definition 3.19). On the other hand, it means that a *BoundaryCell* cannot exist in primal space without at least two corresponding *SpaceCell* instances. The boundary of a *Space* may be decomposed into zero or more *SpaceBoundary* instances. Typically, a *Space* has at least one boundary component. The only case for modelling a *Space* without boundary component is the representation of the unbounded and boundaryless outer space cell S_{out} on the minimal space layer L_{min} (cf. definition 3.32). The *CoBoundary* association allows for querying the *Space* instances sharing a given *SpaceBoundary*, and hence for evaluating topological adjacency relationships between *Space* entities (and thus implicitly between *SpaceCell* features). In dual space, *State* and *Transition* are also linked by *Boundary* and *CoBoundary* associations. However,

[43] The attribute triplet *class*, *function*, and *usage* is inspired from the conceptual data model of CityGML (cf. Gröger et al. 2012) and is specified identically to ease data mappings.

[44] This work was carried out in a Master's thesis conducted at the Institute for Geodesy and Geoinformation Science, Technische Universität, Berlin, under the supervision of the author.

and conformant to the Poincaré duality, the directions of the associations are reversed. Every *Transition* needs to be bounded be exactly two *State* instances and expresses the topological adjacency of the associated *SpaceCell* features. The *CoBoundary* relation enumerates the *Transition* features being incident to a single *State*. Consistent with the associations between the primal space feature types, the modelling of a *BoundaryCell* requires two *SpaceCell* features, and a *State* with empty coboundary necessarily represents S_{out} on L_{min}.

In order to ensure data consistency, OCL constraints are associated with *SpaceCell* and *BoundaryCell*. Precisely, in the context of *SpaceCell*, the following OCL invariant needs to be fulfilled.[45]

```
context SpaceCell inv:
    (primal->notEmpty() and dual->notEmpty()) implies primal.dual = dual
```

The OCL expression states that if a *SpaceCell* has both *Space* and *State* representations, then the *State* instance accessible through primal.dual (i.e., following the *PrimalSpace* and *Duality* association) equals the *State* instance linked by the *DualSpace* association. Since the constraint is marked as invariant, the expression must be true for every *SpaceCell* to be valid. The identical OCL constraint is expressed on the *BoundaryCell* feature type. The MLSEM data model hence guarantees the consistent navigation between primal and dual space instances as well as their associated *SpaceCell* and *BoundaryCell* features that carry the semantic information.

The *SpaceLayer* feature type realizes the notion of a space layer as given in definition 3.19 and thus aggregates *SpaceCell* and *BoundaryCell* features. According to the minimal possible representation L_{min} of a space layer (cf. definition 3.32), every *SpaceLayer* needs to contain at least one *SpaceCell* representing the outer space. Each *SpaceLayer* instance can be characterized through a *symbolicId* as well as a *name* and a *description*. The mandatory *type* attribute denotes the notion of space reflected by the space layer. Its value is taken from the extensible code list *SpaceLayerType* which predefines the values *topographic*, *sensor*, and *logical*. Further semantic information can be provided as *generic attributes* which hence correspond to the set of semantic attributes $A(L)$ of a space layer L as given in definition 3.19. Similar to the *SpaceCell* feature type, (transitive) inclusion relationships between *SpaceLayer* features (cf. definition 3.51) can be expressed through the *SubSpace* association. The binary operations of the space layer algebra developed in chapter 3.5 are implemented by the *merge*, *difference*, and *intersection* methods which take a *SpaceLayer* feature as input and return a *SpaceLayer* as result.

The intra-layer graph of a space layer is semantically represented by the *IntraLayerGraph* type. Its node set results from the aggregation of one or more *State* features, whereas its edge set is a collection of corresponding *Transition* features. The following OCL constraint on *IntraLayerGraph* ensures that the *State* and *Transition* instances participating in an *IntraLayerGraph* are dual space representations of *SpaceCell* and *BoundaryCell* features contained in the *SpaceLayer* to which the *IntraLayerGraph* is associated through the *Graph* association.

```
context IntraLayerGraph inv:
    layer.spaceCell.dual->includesAll(node) and
    layer.boundaryCell.dual->includesAll(edge)
```

The *IntraLayerGraph* constitutes an undirected graph due to the *Boundary* and *CoBoundary* associations between *State* and *Transition* being unordered. This agrees with the mathematical model of the MLSEM according to which the edges of the intra-layer graph are 1-cells in the dual space topology of the space layer and thus are undirected per definition. It also conforms to an intuitive spatial understanding since the direction of movement is generally much less restricted in indoor spaces than, for example, in road networks. The *cost* attributes modelled for both *State* and *Transition* allow for applying measures to single graph elements rendering the *IntraLayerGraph* a weighted graph. The cost values can be fed, for example, to cost functions for finding the shortest or best paths. The data type of *cost* is *GenericAttributeType* which admits arbitrary units of costs including distance, monetary, and time-based measures. For the most common navigation algorithms, the cost value is typically restricted to be non-negative. This requirement may be imposed by applications but is not enforced by the MLSEM data model itself. The normal mechanism of cost functions is to define the total cost as the sum of the costs applied to the graph elements. However, the MLSEM schema does not presuppose a choice of cost function but is kept generic

[45] Note that in the UML diagrams only the expression part of the OCL constraint is depicted. All expressions are assumed to be invariants. The contextual information (i.e., to which type the constraint applies) is implicitly provided through attaching the UML note in which the OCL expression is embedded to a UML model element.

so that different types of cost functions (e.g., static, predictive, or dynamic functions) that rely on different types and values of cost measures (possibly including negative values expressing penalties on graph elements) can be supported.

SpaceLayer features are assembled to form a space layer complex (cf. definition 3.36) through the feature type *SpaceLayerComplex*. A *SpaceLayerComplex* can be given a *name* and a *description* and consists of zero or more *SpaceLayer* instances. A given *SpaceLayer* may likewise participate in zero or more *SpaceLayerComplex* features each of which can be used, for example, to represent a different context of navigation from a base set of space layers (cf. chapter 3.6). Every *SpaceLayerComplex* can be assigned an instance of *MultilayeredGraph* that carries the notion of the multilayered graph into the MLSEM data model. In conformance with definition 3.40, a *MultilayeredGraph* is required to contain the set of *IntraLayerGraph* subgraphs associated with the *SpaceLayer* features participating in the *SpaceLayerComplex*. The subgraphs are linked through *InterLayerEdge* features so that every *InterLayerEdge* connects two *State* instances from different *IntraLayerGraph* entities. The feature type *InterLayerEdge* hence corresponds to the notion of inter-layer edges as given in definition 3.37 (ii). Both consistency requirements are ensured by the following OCL invariants.

```
context MultilayeredGraph inv:
    subGraph.layer = layerComplex.layer

context InterLayerEdge inv:
    boundary->first().graph <> boundary->last().graph
```

The *Boundary* and *CoBoundary* associations between *State* and *InterLayerEdge* are defined in the same way as between *State* and *Transition*. Thus, every *InterLayerEdge* is bounded by precisely two dual nodes and cannot be instantiated without its boundary. Similar to the mathematical model, an *InterLayerEdge* feature is labelled with the binary topological relationship of its incident space cells in primal space (cf. definition 3.39). For this purpose, the *InterLayerEdge* feature type provides the attribute *topoRelation*. Allowed values for *topoRelation* are specified in *TopoRelationEnum* which enumerates the possible topological relationships between space cells according to the 4-intersection matrix given in formula 3.38. Since *InterLayerEdge* instances are undirected the relationships *contains* and *inside* as well as *covers* and *coveredBy* are mapped onto a joint representation, namely *containsOrInside* respectively *coversOrCoveredBy*. Note that the inter-layer graph of a space layer complex as defined in the mathematical model (cf. definition 3.37) is not represented by a separate semantic model element of the MLSEM schema since it is contained in and can be easily derived from the *MultilayeredGraph*.

Spatial characteristics of the semantic concepts. In the UML diagram in figure 170, the geometric-topological description of the introduced semantic concepts is presented. The specification of the spatial characteristics employs the ISO 19107 Spatial Schema.

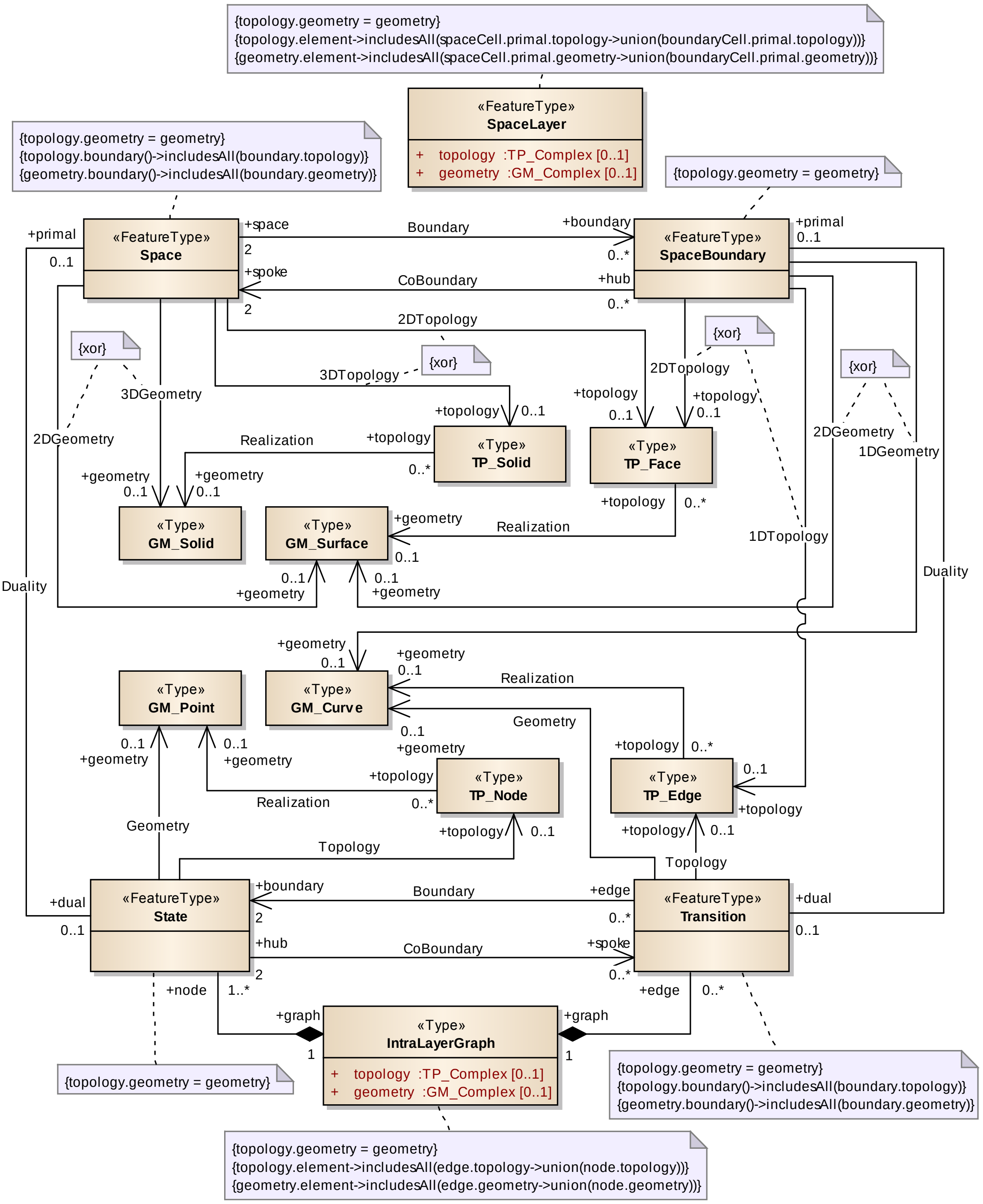


Figure 170: The geometric-topological model of the Space Representation package.

The primal space geometry $GM(S)$ of a space cell S is translated into the two associations *2DGeometry* and *3DGeometry* linking the *Space* feature type to the geometric primitive *GM_Surface* respectively *GM_Solid* from the Spatial Schema. Since per definition 3.2 a space cell is either represented in two or three dimensions, a Boolean xor constraint is applied to both associations. This exclusive disjunction postulates that *Space* instances may participate in at most one of the associations at a time, and thus may be spatially described through an instance of either *GM_Surface* or *GM_Solid* but not both. The geometric primitives are restricted to meet conditions (i) and (ii) of definition 3.2 when used in the context of bounded space cells, but are likewise used to model the non-compact and possibly non-manifold space occupied by the outer space cell (cf. definition 3.12) as discussed in chapter 4.3.1. Thus, and similar to the semantic view, the MLSEM schema does not introduce separate model elements for the spatial description of the outer space cell. Following the same modelling approach, the primal space topology $TP(S)$ of a space cell is mapped onto the two associations *2DTopology* and *3DTopology* which relate *Space* to the topological primitives *TP_Face* and *TP_Solid*. The Boolean xor constraint on both associations again guarantees that every instance of *Space* is topologically represented exclusively in either two or three dimensions. Both the primal space geometry $GM(B)$ (cf. definition 3.26) and topology $TP(B)$ (cf. definition 3.25) of a boundary cell B are realized correspondingly through constrained pairs of associations between *SpaceBoundary* and geometric respectively topological primitives in two and one dimensions.

The spatial data model is more flexible than the mathematical definitions in that both *Space* and *SpaceBoundary* may have neither a geometric nor a topological representation. Thus, the spatial description of primal space may be given purely geometrically, purely topologically, or in a geometric-topologically consistent manner. This relaxation of the mathematical model is mostly introduced for practical reasons since many software systems (including GIS applications and spatial databases) still nowadays are limited to the modelling and storage of geometric aspects of features whereas topology is often only supported in terms of topological relationships between features which are derived (possibly on the fly) from the geometric configuration using computational geometry algorithms. Due to the flexibility in the spatial representation of space cells and boundary cells, the MLSEM data model can thus be used to feed such systems and to carry their output. Moreover, neglecting either the geometric or the topological dimension also reduces the storage space required for indoor settings modelled according to the MLSEM schema. However, it is important to remember that the number of geometric elements describing the primal space not necessarily equals the number of topological entities as discussed in chapter 3.1.3. Therefore, omitting a given topological decomposition of primal space may lead to information loss since it might not be automatically recoverable in the same way from the geometric configuration (and vice versa).

Based on the *Realization* association between *TP_Primitive* and *GM_Primitive*, the following OCL constraint makes sure that the geometric primitive associated with a *Space* feature actually realizes the topological primitive. It implicitly follows from this constraint that the dimensions of both primitives agree. The identical OCL constraint is also modelled for *SpaceBoundary*.

```
context Space inv:
    topology.geometry = geometry
```

Since the *Boundary* and *CoBoundary* associations between *Space* and *SpaceBoundary* only address the semantic level, further OCL constraints are required in order to consistently map these relations into geometry and topology space. Precisely, if a *SpaceBoundary* is on the boundary of a *Space* then its geometric respectively topological primitive needs to be on the boundary of the corresponding primitive associated with the *Space*.[46] The OCL expression uses of the *boundary()* operation defined for *GM_Object* and *TP_Object* for this purpose.

```
context Space inv:
    topology.boundary()->includesAll(boundary.topology) and
    geometry.boundary()->includesAll(boundary.geometry)
```

As shown in chapter 4.3.1, a *TP_Primitive* cannot exist without a *TP_Complex*. Since a *Space* (*SpaceBoundary*) is uniquely associated to a *SpaceCell* (*BoundaryCell*) which itself is uniquely associated to a *SpaceLayer*, the *TP_Complex* is modelled as spatial attribute *topology* of the *SpaceLayer* feature type. Similarly, *SpaceLayer* provides a *GM_Complex* through its *geometry* attribute which captures the geometric configuration of the space cells in primal space. The below OCL constraint is attached to *SpaceLayer* in order to assert that the geometric complex is the realization of the topological complex and that all topological and geometric primitives modelled for the *SpaceCell* and *BoundaryCell* instances contained in a *SpaceLayer* feature participate in the *TP_Complex* respectively *GM_Complex*. Moreover, from the definition of *TP_Complex* and *GM_Complex* it implicitly follows that the spatial primitives are required to be non-overlapping which satisfies the identical requirement imposed for the primal space representation of space layers (cf. definition 3.7 and chapter 3.1.2.1).

```
context SpaceLayer inv:
    topology.geometry = geometry and
```

[46] The boundary and coboundary relations between both geometric primitives and topological primitives as defined in ISO 19107:2003 have been omitted from the UML diagram in figure 170 for readability (see chapter 4.3.1 instead). Note that the *Boundary* and *CoBoundary* associations on the semantic level are more restrictive than the ones put in place by the Spatial Schema due to the fact that the MLSEM assumes a complete tiling of Euclidean space.

```
topology.element->includesAll(spaceCell.primal.topology->union(boundaryCell.primal.topology)) and
geometry.element->includesAll(spaceCell.primal.geometry->union(boundaryCell.primal.geometry))
```

The spatial representation of the semantic entities in dual space follows a similar approach. The dual space geometry $v_{GM}(S)$ of a space cell S (cf. definition 3.5) is given by a *GM_Point* that is associated with the *State* feature type. This *GM_Point* is the geometric carrier for a *TP_Node* that represents the dual space topology $v_{TP}(S)$ of S (cf. definition 3.4) and hence is also related to *State*. Likewise, the dual space geometry $e_{GM}(B)$ and topology $e_{TP}(B)$ of a boundary cell B (cf. definition 3.24) are mapped onto a *GM_Curve* realizing a *TP_Edge* both of which are spatial characteristics of the *Transition* feature type. The geometric and topological primitives form complexes which are modelled as spatial attributes on the level of the *IntraLayerGraph* feature type. Thus, the *topology* attribute of type *TP_Complex* realizes the dual graph $G_{TP}(L)$ of a space layer L (cf. definition 3.20) whose spatial embedding $G_{GM}(L)$ (cf. definition 3.21) is available as *GM_Complex* through the *geometry* attribute. The consistency of the spatio-semantic description in dual space is enforced by OCL constraints which are identically specified as discussed above for the primal space representation. It is important to note that the strict mathematical definition is again relaxed in order to facilitate purely geometric, purely topological, as well as geometric-topological descriptions of dual space.

According to ISO 19107, all geometric objects participating in the same *GM_Complex* must be associated with the same coordinate reference system. Thus, the space cells and boundary cells on a given *SpaceLayer* feature have to share the same reference system for their primal space geometry. The same holds true for their dual space geometries which are elements of the *GM_Complex* modelled for the *IntraLayerGraph* feature type. However, since the primal and dual space geometries are aggregated in separate *GM_Complex* objects, they may consequently be associated with the same or a different reference system. Likewise, two space layers (e.g., a topographic space layer and a sensor space layer) not necessarily have to share the same reference system. The possibility to support multiple and different coordinate reference systems has been identified as an important requirement to indoor space modelling in chapter 1.2 and thus is met by the MLSEM and the underlying Spatial Schema.

The Spatial Schema also supports the derivation of a default Euclidean space embedding of the dual graph. Precisely, each *GM_Object* provides a *representativePoint()* operation which is defined to return a point value that is guaranteed to be inside this *GM_Object* (which may be the centroid of the geometric object).[47] The representative point of the geometric primitive of a *Space* feature can thus be used to instantiate the *GM_Point* object representing the *geometry* of the dual *State* feature. Using OCL notation, this can be expressed for a *State* instance as follows based on the *Duality* association linking the *State* and the *Space* feature types.

```
State::geometry : GM_Point = GM_Point(State::primal.geometry.representativePoint())
```

Then, the curve geometry of *Transition* features can be defined as straight line segment (*GM_LineString*) connecting the two incident points in the *Boundary* association.

```
Transition::geometry : GM_Curve = GM_Curve(GM_LineString(Transition::boundary.first().position,
    Transition::boundary.second().position))
```

The same way, the *GetDefaultEmbedding* algorithm as used in the definition of the operations of the space layer algebra in chapter 3.5 can be implemented.

4.4.1.2 Joint States Package

The *Joint States* package augments the space representation of the MLSEM schema with a conceptual data model for joint states and joint state transitions as introduced in chapter 3.3. The UML class diagram defining the contents of the *Joint States* package is shown in the following figure 171.

[47] Per definition 3.5, for every space cell S it is mandated that $v_{GM}(S) \in Int(GM(S))$.

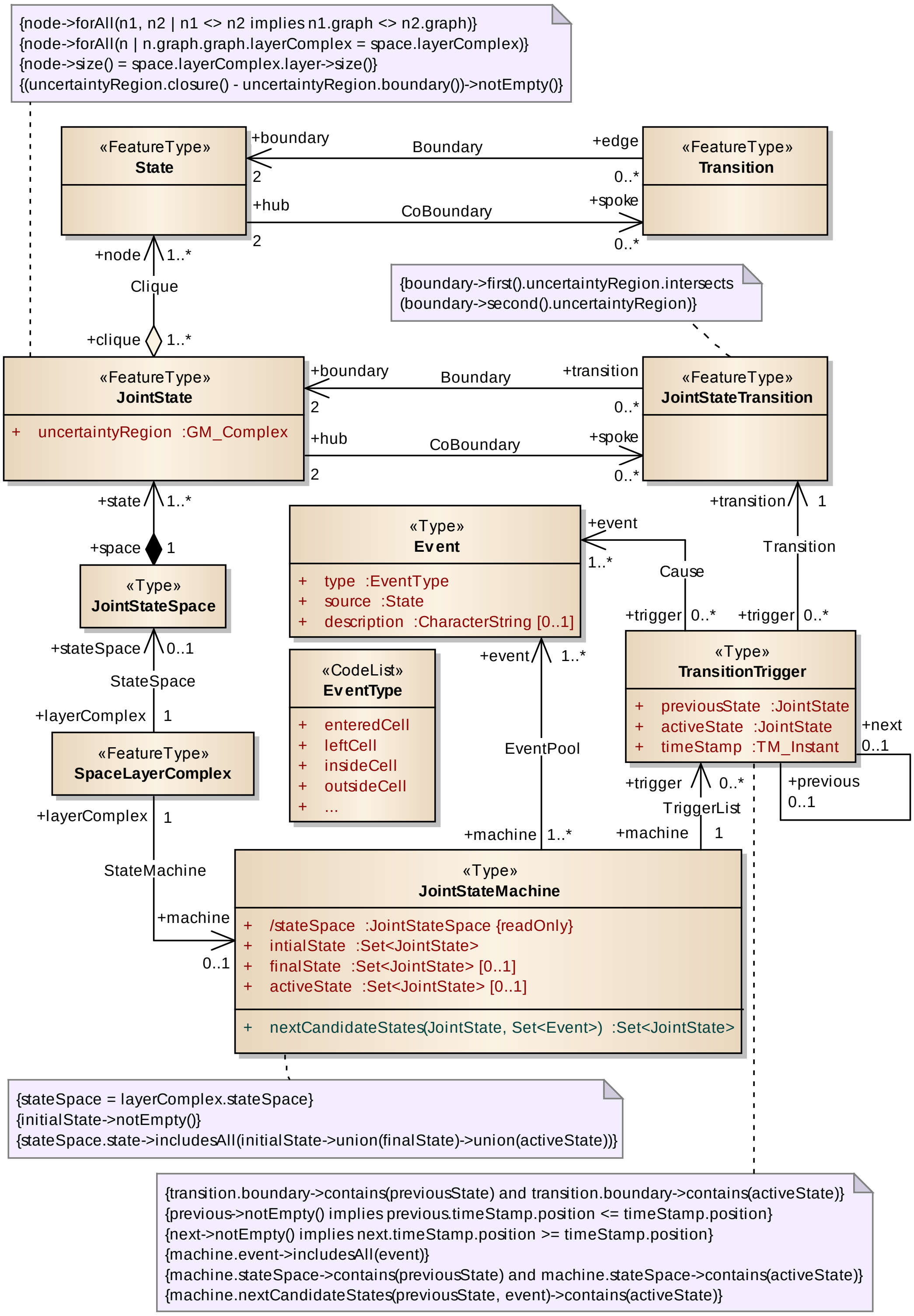

{node->forAll(n1, n2 | n1 <> n2 implies n1.graph <> n2.graph)}
{node->forAll(n | n.graph.graph.layerComplex = space.layerComplex)}
{node->size() = space.layerComplex.layer->size()}
{(uncertaintyRegion.closure() - uncertaintyRegion.boundary())->notEmpty()}
«FeatureType»
State
+boundary
Boundary
+edge
2
0..*
+hub
CoBoundary
+spoke
2
0..*
«FeatureType»
Transition
+node
1..*
Clique
+clique
1..*
{boundary->first().uncertaintyRegion.intersects
(boundary->second().uncertaintyRegion)}
«FeatureType»
JointState
+ uncertaintyRegion :GM_Complex
+boundary
Boundary
+transition
2
0..*
+hub
CoBoundary
+spoke
2
0..*
«FeatureType»
JointStateTransition
+state
1..*
+space
1
«Type»
JointStateSpace
+stateSpace
0..1
StateSpace
+layerComplex
1
«FeatureType»
SpaceLayerComplex
+layerComplex
1
StateMachine
+machine
0..1
«Type»
Event
+ type :EventType
+ source :State
+ description :CharacterString [0..1]
+event
1..*
Cause
«CodeList»
EventType
+ enteredCell
+ leftCell
+ insideCell
+ outsideCell
+ ...
+event
1..*
EventPool
+machine
1..*
+transition
1
Transition
+trigger
0..*
+trigger
0..*
«Type»
TransitionTrigger
+ previousState :JointState
+ activeState :JointState
+ timeStamp :TM_Instant
+next
0..1
+previous
0..1
+trigger
0..*
TriggerList
+machine
1
«Type»
JointStateMachine
+ /stateSpace :JointStateSpace {readOnly}
+ intialState :Set<JointState>
+ finalState :Set<JointState> [0..1]
+ activeState :Set<JointState> [0..1]
+ nextCandidateStates(JointState, Set<Event>) :Set<JointState>
{stateSpace = layerComplex.stateSpace}
{initialState->notEmpty()}
{stateSpace.state->includesAll(initialState->union(finalState)->union(activeState))}
{transition.boundary->contains(previousState) and transition.boundary->contains(activeState)}
{previous->notEmpty() implies previous.timeStamp.position <= timeStamp.position}
{next->notEmpty() implies next.timeStamp.position >= timeStamp.position}
{machine.event->includesAll(event)}
{machine.stateSpace->contains(previousState) and machine.stateSpace->contains(activeState)}
{machine.nextCandidateStates(previousState, event)->contains(activeState)}

Figure 171: The contents of the Joint State package.

A joint state as given in definition 3.41 is translated into the *JointState* feature type which consists of a set of *State* features that are related through the *Clique* association. In accordance with condition (i) of definition 3.41, the *State* instances need to belong to different *IntraLayerGraph* entities which themselves have to be part of the same *MultilayeredGraph*. Every *JointState* participates in precisely one *JointStateSpace* which maps the notion of a joint state space from the mathematical model (cf. definition 3.42) and which hence is linked to exactly one *SpaceLayerComplex* via its *StateSpace* association. The *MultilayeredGraph* containing the *State* features of all *JointState* features within a *JointStateSpace* needs to be associated with the same *SpaceLayerComplex* as the *JointStateSpace* itself. Moreover, the number of *State* features participating in the *Clique* of a *JointState* has to be equal to the number of *SpaceLayer* features aggregated by this *SpaceLayerComplex*. Both requirements are also a consequence of condition (i) of definition 3.41. The following OCL constraint on *JointState* captures these consistency invariants.

```
context JointState inv:
    node->forAll(n1, n2 | n1 <> n2 implies n1.graph <> n2.graph) and
    node->forAll(n | n.graph.graph.layerComplex = space.layerComplex) and
    node->size() = space.layerComplex.layer->size()
```

In addition to the clique of dual nodes, a joint state JS is also characterized by its intersection geometry $GM(JS)$ that results from the intersection of the primal space geometries of the corresponding space cells (cf. condition (ii) of definition 3.41). Since this intersection geometry equivalently reflects the region of indoor space in which a moving person or object must be located when the joint state is triggered to be active, it is modelled as spatial attribute named *uncertaintyRegion* of the *JointState* feature type. The type of *uncertaintyRegion* is chosen to be *GM_Complex* as the intersection geometry cannot be guaranteed to be connected. According to condition (iii) of definition 3.41, the interior of this *GM_Complex* may not be the empty set which is ensured by an additional OCL constraint.

```
context JointState inv:
    (uncertaintyRegion.closure() - uncertaintyRegion.boundary())->notEmpty()
```

Transitions between joint states (cf. definition 3.44) are modelled as instances of the *JointStateTransition* feature type. *JointState* and *JointStateTransition* are linked by *Boundary* and *CoBoundary* associations which are identically defined as the equivalent associations between *State* and *Transition* as introduced in the previous chapter. It hence follows that every *JointStateTransition* has precisely two *JointState* features on its boundary, whereas the coboundary of a *JointState* may be the empty set. An empty coboundary however only occurs if a *SpaceLayerComplex* merely contains copies of the minimal space layer L_{min} (cf. formula 3.43). In order to satisfy condition (ii) of definition 3.44, the *uncertaintyRegion* geometries of the *JointState* instances on the boundary of a *JointStateTransition* need to have a non-empty intersection which is expressed as OCL invariant on *JointStateTransition* using the *intersects(GM_Object)* operation of *GM_Object*.

```
context JointStateTransition inv:
    boundary->first().uncertaintyRegion.intersects(boundary->second().uncertaintyRegion)
```

The *JointStateMachine* type realizes the notion of a finite-joint-state machine as developed in definition 3.46. An instance of *JointStateMachine* is linked to a single *SpaceLayerComplex* via the *StateMachine* association. Its finite, non-empty state space is accessible through the thematic attribute *stateSpace* of type *JointStateSpace*. This attribute is marked as derived and read-only since its value necessarily must match the *JointStateSpace* of the related *SpaceLayerComplex* feature. The *stateSpace* thus conforms to condition (ii) of definition 3.46, and its derivation rule is expressed by the following OCL constraint.

```
context JointStateMachine inv:
    stateSpace = layerComplex.stateSpace
```

The attributes *initialState* and *finalState* likewise correspond to conditions (iii) and (v) of definition 3.46. Whereas *initialState* denotes the non-empty set of candidate *JointState* features describing competing initial states of the automaton, the value of the *finalState* attribute provides its (possibly empty) set of final states. The *joint state of navigation*, i.e. the currently active state of the automaton, is described through the *activeState* attribute. Its value is also given by a set of *JointState* features since the finite-joint-state machine per definition is non-deterministic. The OCL constraint below guarantees that *initialState*, *finalState*, as well as *activeState* are subsets of the *stateSpace* of the *JointStateMachine*, and that the *initialState* is non-empty.

```
context JointStateMachine inv:
    initialState->notEmpty() and
    stateSpace.state->includesAll(initialState->union(finalState)->union(activeState))
```

A further essential component of the *JointStateMachine* is its pool of events that may trigger a joint state transition when dispatched (cf. condition (i) of definition 3.46). Events are conceptualized by the *Event* class which defines the mandatory attributes *type* and *source*. The event *type* classifies the cause of the event and is taken from the extensible code list *EventType* which predefines the values *enteredCell*, *leftCell*, *insideCell*, and *outsideCell*. Each of these values is to be interpreted in the context of a given space cell which thus is referenced through its dual *State* representation using the *source* attribute. The dispatching of an event is typically induced by the movement of a navigation user. For example, assume a space cell S that represents the signal reception area of a Wi-Fi transmitter. The event of receiving the previously non-available signal of S can be expressed as instance of *Event* with type = enteredCell and source = v(S). The event of continuously receiving the signal of S (i.e., the user keeps travelling inside $GM(S)$) is modelled as separate *Event* with type = insideCell. Likewise, the event of leaving the signal reception area $GM(S)$ is captured by an *Event* with type = leftCell. Every *Event* instance can be assigned an optional *description* (e.g., to attach a human-readable message). The event pool of the *JointStateMachine* is realized through the *EventPool* association.

The *nextCandidateStates* operation of *JointStateMachine* implements the joint-state-transition map ϕ as defined in condition (iv) of definition 3.46. Similar to ϕ, it takes a *JointState* feature reflecting the (assumed) active state of the automaton as well as a set of received events as input. Its output is a (possibly empty) set of *JointState* instances which represent the non-deterministic next active state of the machine and thus can be assigned to its *activeState* attribute. Without further knowledge or assumptions, the candidates are equally likely (cf. example 3.48 in chapter 3.3). Note that both the *nextCandidateStates* operation as well as the *JointStateMachine* itself are only provided as interfaces which have to be implemented by a navigation application. Such implementations may introduce, for example, additional dynamic aspects and probabilistic methods for the derivation of the next active state (e.g., a maximum speed constraint to reduce the uncertainty region as proposed in the cell-based positioning approach of Jensen et al. 2010).

In order to track location sequences of a navigation user, the *TransitionTrigger* type has been added to the data model. An instance of *TransitionTrigger* represents a series of events at a given point in time which triggered the transition between two consecutive active joint states of the navigation user. A list of transition triggers ordered by time positions hence captures the trajectory through indoor space travelled by the user on all space layers contained in the *SpaceLayerComplex*. Thus, for a navigation system implementing the MLSEM schema, the *JointStateMachine* supports the task of localization, whereas the concept of the *TransitionTrigger* addresses the tracking of users. A *TransitionTrigger* has a mandatory *previousState* of type *JointState* which represents the active joint state of the user at a previous point in time. The set of events linked through the *Cause* relation changes the *previousState* to the new *activeState* following the associated *JointStateTransition*. The *timeStamp* attribute captures the point in time when the dispatched events triggered the transition. Its value domain is given by the *TM_Instant* type which is defined in the ISO 19108:2005 Temporal Schema for the modelling of time positions. It obviously follows that the *JointState* features denoted by *previousState* and *activeState* have to render the boundary of the triggered *JointStateTransition*. A *TransitionTrigger* may further point to its predecessor respectively successor in time in order to establish a time-ordered sequence of locations. Therefore, the *previous* (*next*) trigger needs to be recorded at a previous (later) point in time. The OCL constraint shown below enforces both conditions. The time-based expressions rely upon the partial order of time positions as defined in ISO 19108:2005 for the *position* attribute of *TM_Instant*.

```
context TransitionTrigger inv:
   transition.boundary->contains(previousState) and transition.boundary->contains(activeState) and
   (previous->notEmpty() implies previous.timeStamp.position <= timeStamp.position) and
   (next->notEmpty() implies next.timeStamp.position >= timeStamp.position)
```

Each *TransitionTrigger* is uniquely related to a *JointStateMachine* through the *TriggerList* association. When feeding both the *previousState* and the set of events of a *TransitionTrigger* to the *nextCandidateStates* operation of the *JointStateMachine*, then the *activeState* shall be contained in the resulting output. Since the joint-state machine is non-deterministic, zero or more *TransitionTrigger* instances may exist for the same point in time capturing competing active joint states. Also contradictory *TransitionTrigger* instances at consecutive points in times are possible and allowed as discussed in example 3.49 (cf. chapter 3.3). They may be filtered online (i.e., in real time while the user is travelling through indoor space) based on additional input or methods (see above), or offline using a posteriori knowledge since both the *JointStateMachine* and the list of *TransitionTrigger* can be stored based on the MLSEM data model and hence be evaluated at a later point in time.

In order to apply the *nextCandidateStates* operation to the information carried by a *TransitionTrigger*, it must be guaranteed that the set of events as well as the *previousState* and *activeState* are subsets of the event pool respectively the joint state space of the associated *JointStateMachine*.

```
context TransitionTrigger inv:
    machine.event->includesAll(event) and
    machine.stateSpace->contains(previousState) and machine.stateSpace->contains(activeState) and
    machine.nextCandidateStates(previousState, event)->contains(activeState)
```

4.4.1.3 Source Object and External Reference Packages

The data for populating single *SpaceCell* and *BoundaryCell* features as well as entire *SpaceLayer* instances may be derived from existing data sets. For example, especially in case of topographic space, existing 2-dimensional or 3-dimensional building models are a rich data source (e.g., models in formats such as IFC, CityGML, X3D, KML, COLLADA, ESRI BISDM, or legacy CAD). The *Source Object* package allows for providing a reference to original data sources and even for storing an entire source object inline the MLSEM data. This is especially useful since external datasets may provide information beyond the scope and expressivity of the MLSEM schema, and thus this information remains accessible. The following UML diagram depicts the contents of the *Source Object* package.

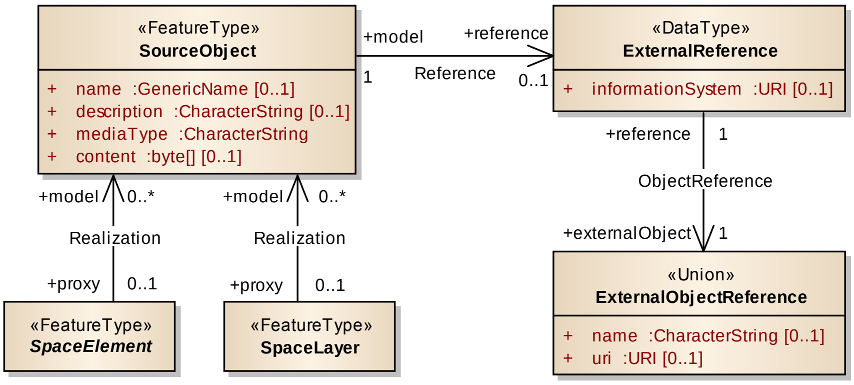


Figure 172: The contents of the Source Object and External Reference packages.

The central class of the package is *SourceObject*. An instance of *SourceObject* represents an identifiable object from an external data source that corresponds to either a single *SpaceElement* (i.e., a *SpaceCell* or *BoundaryCell)* or *SpaceLayer*, each of which takes the role of a *proxy* for the source object according to the *Realization* associations. Every *SpaceElement* and *SpaceLayer* feature may on the contrary be linked to zero or arbitrarily many data sources. A *SourceObject* can be given a *name* and a *description*, but at least has to provide the data media type (formerly referred to as MIME type) in which the source object is encoded through the *mediaType* attribute. A digital copy of the source object itself may then be provided as binary array using the *content* attribute. The binary data explicitly includes text-based representations such as XML data documents. The source object may alternatively be given by a reference to an external data source via the *Reference* association to *ExternalReference*.

Both *ExternalReference* and *ExternalObjectReference* are classes defined in the *External Reference* package of the MLSEM schema.[48] An *ExternalReference* provides a link pointing to the source object within an external information system. The information system is given by a URI, whereas the object itself is identified through an instance of *ExternalObjectReference* that may either carry the object's *name* or another URI as unique identifier. If the *informationSystem* attribute of *ExternalReference* is not set, then the *ExternalObjectReference* must be given as URI. An *ExternalReference* may point to arbitrary data resources including databases and file-based datasets, and even allows for encoding preformatted requests to web services that deliver the source object (e.g., standardized web services such as the OGC Web Feature Service).

[48] The concepts of *ExternalReference* and *ExternalObjectReference* are inspired from the conceptual data model of CityGML (cf. Gröger et al. 2012) and are specified identically to ease data mappings.

4.4.1.4 *Groups and Sequences Package*

SpaceCell and *BoundaryCell* features may be aggregated in groups and sequences based on application-specific and non-spatial criteria. The grouping concept of the MLSEM schema is provided by the *Groups and Sequences* package and is depicted in the UML diagram in figure 173.

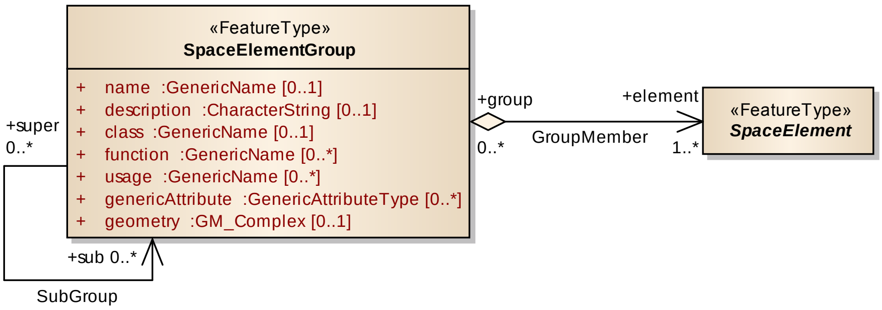

«FeatureType»
SpaceElementGroup
+ name :GenericName [0..1]
+ description :CharacterString [0..1]
+ class :GenericName [0..1]
+ function :GenericName [0..*]
+ usage :GenericName [0..*]
+ genericAttribute :GenericAttributeType [0..*]
+ geometry :GM_Complex [0..1]
+super
0..*
+sub 0..*
SubGroup
+group
0..*
GroupMember
+element
1..*
«FeatureType»
SpaceElement

Figure 173: The SpaceElementGroup feature type.

Space element groups. A group is realized by an instance of the feature type *SpaceElementGroup* and may contain an arbitrary number of *SpaceElement* features as group members. It may optionally be characterized through a *name* and a *description*, and can be further classified using the attribute triplet *class*, *function*, and *usage* which have the same semantics as in the context of *SpaceElement* (cf. chapter 4.4.1.1). Further semantic information about the group itself are modelled as *generic attributes*. Groups can be recursively contained in other groups based on the *SubGroup* association linking *SpaceElementGroup* to itself. This subgrouping mechanism facilitates arbitrarily nested structures and is not restricted to tree-like hierarchies. Moreover, each group can be assigned a spatial representation expressed as *GM_Complex* which, for example, may follow from the geometric descriptions of its group members.

The grouping concept of the MLSEM schema is to be seen complementary to the multi-granular representation of indoor space based on subspacings of space cells and space layers and the induced spatial hierarchies. Subspacing, as comprehensively developed in chapter 3.4, builds upon spatial containment relationships which impose a partial ordering of space cells and space layers. Whereas spatial containment is especially feasible to represent the nested hierarchical structure of the interior built environment (cf. example 3.61), it may not always be suitable to express groups of entities that rather follow from semantic criteria. Examples are groupings of rooms inside a building according to functional, organizational, or logical aspects as proposed by (Richter et al. 2009) (cf. chapter 2.1.1). A *SpaceElementGroup* is intended to represent such aggregations and semantic hierarchies of *SpaceCell* and *BoundaryCell* features which are not primarily based on spatial containment. There are no restrictions on the group members which may even belong to different *SpaceLayer* features and thus may spatially overlap. Members of subgroups may likewise spatially overlap with members of their parent groups and may also occupy a larger region of space (in contrast to subspaces). For this reason, *SpaceElementGroup* features obviously cannot participate in the spatial description of space layers and hence have no impact on the dual graph representation of indoor space event if they carry their own geometry.

Nevertheless, groups are useful in path queries as they allow for addressing aggregates of space cells. For example, suppose a navigation user wants to be routed to a given department within an office building but has no further information about a specific room number. Then a *SpaceElementGroup* embracing all entities belonging to the department could be used in answering the query. A group could also cover the restrooms within a facility (e.g., at airports, museums, etc.) or similar shops in a large shopping malls (e.g., bookshops, food stores, etc.) in order to quickly respond to corresponding path queries. Likewise, a group might contain all emergency exists within a building or all walls made of a specific material that can be easily torn down by rescue personnel in evacuation scenarios. Another related purpose of groups is to represent the results of (frequent) semantic or spatial queries to an instance of the MLSEM in order to store the results and make them easily accessible at later points in time.

Space element sequences. In addition the aggregation of *SpaceElement* features in groups, the MLSEM schema also supports the definition of sequences of *SpaceElement* features by means of the *SpaceElementSequence* feature type. The corresponding UML diagram is presented below.

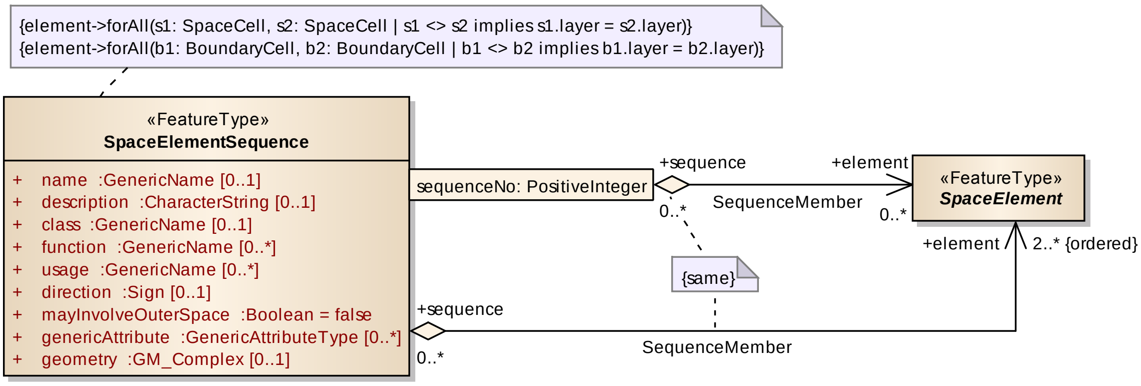

{element->forAll(s1: SpaceCell, s2: SpaceCell | s1 <> s2 implies s1.layer = s2.layer)}
{element->forAll(b1: BoundaryCell, b2: BoundaryCell | b1 <> b2 implies b1.layer = b2.layer)}
«FeatureType»
SpaceElementSequence
+ name :GenericName [0..1]
+ description :CharacterString [0..1]
+ class :GenericName [0..1]
+ function :GenericName [0..*]
+ usage :GenericName [0..*]
+ direction :Sign [0..1]
+ mayInvolveOuterSpace :Boolean = false
+ genericAttribute :GenericAttributeType [0..*]
+ geometry :GM_Complex [0..1]
sequenceNo: PositiveInteger
+sequence
0..*
SequenceMember
+element
0..*
«FeatureType»
SpaceElement
{same}
+element
2..* {ordered}
+sequence
0..*
SequenceMember

Figure 174: The SpaceElementSequence feature type.

A *SpaceElementSequence* is defined as an ordered set of space cells and boundary cells that can be linked in a path from the first *SpaceElement* feature in the set to the last so that all intermediate *SpaceElement* features are visited in the given order. In figure 174, the aggregation association between a *SpaceElementSequence* and its participating *SpaceElement* features is modelled as both an unqualified and a qualified association. The unqualified association specifies that in total there are at least two *SpaceElement* features in every sequence. The qualified association is used to distinguish the separate *SpaceElement* members participating in a specific *SpaceElementSequence* instance by a positive integer based on the *sequenceNo* qualifier. Since this qualified association is in fact the same association as the unqualified one, both are linked by the *same* constraint and share the identical association and role names. The *sequenceNo* qualifier provides a unique index that determines the ascending order of the sequence members. Thus, the sequence starts with that element having the lowest value for *sequenceNo*.

All elements participating in a *SpaceElementSequence* are restricted to be associated with the same *SpaceLayer* feature. Thus, and in contrast to groups, sequences cannot be expressed between space cells and boundary cells from different space layers. This constraint is enforced by the following OCL invariant.

```
context SpaceElementSequence inv:
    element->forAll(s1: SpaceCell, s2: SpaceCell | s1 <> s2 implies s1.layer = s2.layer) and
    element->forAll(b1: BoundaryCell, b2: BoundaryCell | b1 <> b2 implies b1.layer = b2.layer)
```

The simplest example for a sequence is an alternating series of *SpaceCell* and *BoundaryCell* features with every member being incident to its successor (predecessor) when traversing along the *Boundary* and *CoBoundary* associations between their primal respectively dual space representations. A *SpaceElementSequence* may however also be generated from a sparse set of elements, in which case the validation of the sequence involves corresponding path finding algorithms. For example, consider a sequence given by a set of three non-adjacent space cells $\{S_1, S_2, S_3\}$. For this sequence to be valid requires that there exists a path linking S_1 with S_3 via S_2. In case of multiple and different paths fulfilling this requirement, the sequence is to be seen as a proxy representing all of them. The problem of finding a path between the elements can be performed on the intra-layer graph of that space layer the sequence members belong to. A sequence may contain duplicate *SpaceElement* features in order to express cyclic paths.

Note that according to the qualified *SequenceMember* association each unique value of the *sequenceNo* qualifier may designate zero or more *SpaceElement* features. Thus, the ordered set of associated instances is partitioned into subsets each of which may contain more than one *SpaceElement* instance being linked by a given qualifier value. This modelling approach allows, for example, for specifying a set of space cells or boundary cells as possible start or end points of a sequence by simply assigning the same *sequenceNo* qualifier. Likewise, a set of space cells or boundary cells can be used in the midst of a sequence to denote alternative intermediate elements along the path. Validating such a sequence requires that there exists a path from the predecessor to the successor in the *SequenceMember* association which involves at least one *SpaceElement* from the intermediate set. Or, put differently, a path satisfying a sequence must contain at least one *SpaceElement* feature from the set of features associated with each unique sequence number in that order. The sequence then represents all possible paths satisfying this requirement. For example, assume a sequence is given by a single space cell S_1 with the sequence number 1, a set $\{S_2, S_3, S_4\}$ of space cells sharing the sequence number 2, and a final space cell S_5 with the sequence number 3. Assume there exists a path from S_1 to S_5 involving S_2, then this sequence is valid. If there is another path from S_1 to S_5 via both S_3 and S_4, then this path is also captured by the sequence. Since subsequent members in the *SequenceMember* association can be given as set of *SpaceElement* features, finding a path between such sets may render a complex task.

The *SpaceElementSequence* feature type carries the same semantic attributes as *SpaceElementGroup*. Moreover, every sequence may optionally denote its *direction* using a '+' or a '–' sign as value. The positive direction is from the first to the last member in the ordered set, whereas the '–' sign specifies the reverse direction. If the *direction* attribute is not present, the sequence is undirected. The Boolean attribute *mayInvolveOuterSpace* indicates whether or not a path validation for a sparse set may consider paths involving the outer space cell of the space layer (set to *false* per default). If the path validation fails for a given set of *SpaceElement* features then the sequence is deemed invalid.

Sequences can be used, for example, to express movement patterns in indoor space that can be given a direction in contrast to the undirected intra-layer graph of a *SpaceLayer*. In chapter 5 it is shown how navigation constraints such as prohibited and restricted maneuvers (e.g., security gates or doors which may only be passed in one direction) that have to be excluded from routes for navigation users are modelled based on sequences. Note that since sequences are defined for *SpaceElement* features, the modelling of movement patterns and maneuvers does not presuppose a dual space representation of the involved space cells and boundary cells, and hence is independent from an instance of *IntraLayerGraph*.

Relation to the developed space layer algebra. In the following, the impact of the operations of the developed space layer algebra (cf. chapter 3.5) on groups and sequences is discussed.

Example 4.2. Consider the 2-dimensional indoor scene depicted in figure 175. It shows a topographic space layer L_{Topo} which contains two room cells R_1 and R_2 being connected to a corridor. A second space layer L_{Sec} describes a single security zone captured by the space cell Sec that partly affects the room R_1. In order to represent the fact that the two rooms belong to the same organizational unit (e.g., a research department), a *SpaceElementGroup* is modelled containing the space cells R_1 and R_2 as group members.

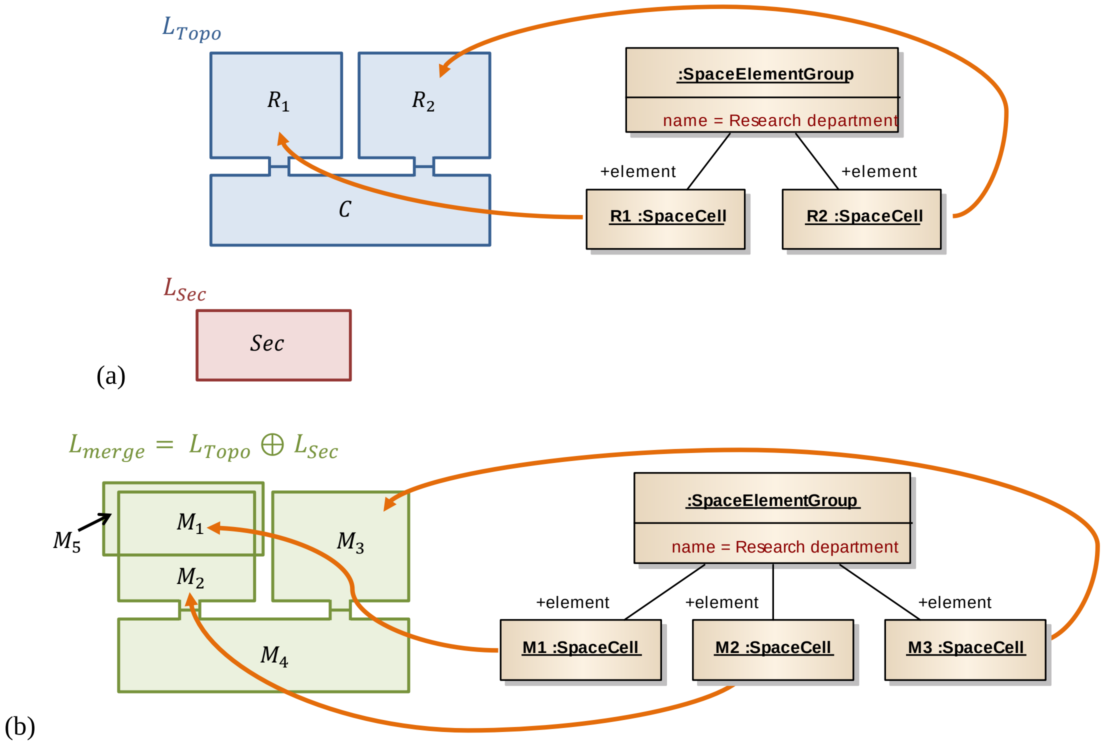


Figure 175: Impact of the merge operation on a SpaceElementGroup feature.

Suppose that both space layers need to be selected for the context of a given navigation user and are input to the merge operation $L_{Topo} \oplus L_{Sec}$ to retrieve an integrated view. According to its definition provided in chapter 3.5, the merge operation results in the space layer L_{merge} shown in figure 175b which contains the space cells M_1 to M_5. In order to not lose the information about the grouping of the rooms on L_{merge}, a new instance of *SpaceElementGroup* has to be created that carries the same semantic information and comprises all those space cells on L_{merge} which are spatially contained in or covered by the space cells R_1 and R_2 on L_{Topo}. Thus, the space cells M_1, M_2, and M_3 need to belong to this group. For example, path queries involving the organizational unit as target can then directly be answered on L_{merge} based on this new group. Since the information which space cells on L_{merge} result from which group members is available during the merge operation (cf. algorithm 3.67), populating a new instance of *SpaceElementGroup* can be realized fully automatically. The same mechanism also applies to the creation of new groups on space layers resulting from difference or intersection operations. Note that a *SpaceElement* feature may only be mapped onto *SpaceElement* features of the same type (i.e., *SpaceCell* or *BoundaryCell*) in the newly created group.

It can be seen in the above example that subspace cells of a space cell being contained in a *SpaceElementGroup* implicitly belong to that same group as well. This also conforms to an intuitive spatial understanding. For example, assume the room cell R_1 is decomposed into four subspace cells on a separate subspace layer. Then these subspace cells obviously belong to the same organizational unit. This already follows from the inclusion relations between R_1 and its subspace cells which can be queried from the multilayered graph. A further explicit modelling of the subspace cells as group members using the *GroupMember* association is redundant and can be omitted. The creation of a new *SpaceElementGroup* as illustrated above is hence only required if the space layers are combined using the space layer algebra and the resulting layer is not participating in the same space layer complex.

Similar considerations apply to sequences. In figure 176, the same topographic space layer L_{Topo} is depicted again. Further space cells are added on either side of the corridor to indicate subsequent corridor segments. Assume a directed *SpaceElementSequence* is defined on the corridor cells using the ordered set $\{C_1, C_2, C_3\}$ with each element receiving its index in the ordered set as sequence number. This sequence could be used, for example, to denote that the corridor may only be traversed in the direction from C_1 to C_3 in emergency situations (cf. chapter 5 for details on how to express such a movement restriction). An additional subspace layer L_{Sub} provides three subspace cells Sub_1 to Sub_3 which decompose the corridor cell C_2 intro three parts.

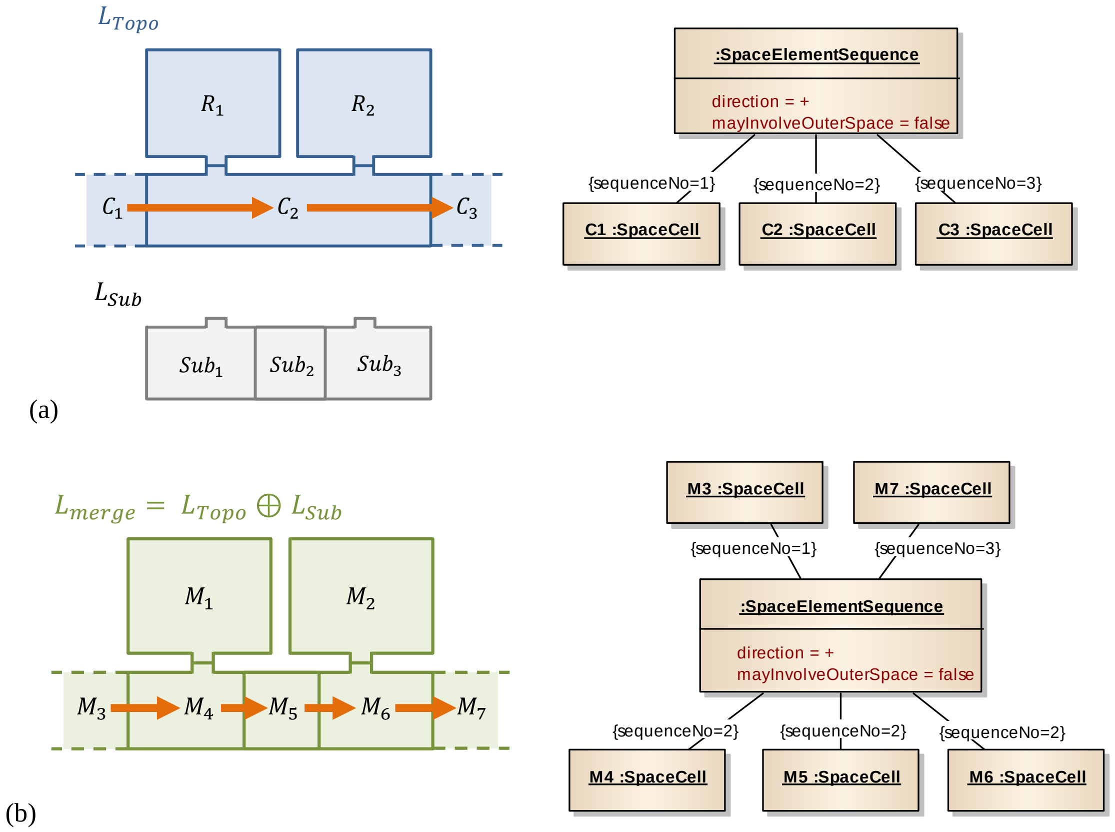


Figure 176: Impact of the merge operation on a SpaceElementSequence feature.

Let us again suppose that L_{Topo} and L_{Sub} shall be integrated into a single space layer using a merge operation whose result is sketched in figure 176b. If the merge space layer L_{merge} is to be used in path searches, then the information about the directed sequence must be maintained on this layer. Like with groups this requires the instantiation of a new *SpaceElementSequence* that first carries the same semantic information. Second, every *SpaceElement* feature participating in the original sequence has to be replaced for the newly created sequence with all those *SpaceElement* features on L_{merge} that are of the same type and that are spatially inside or covered by the original sequence member. The corresponding *SpaceElement* features also need to receive the same sequence number as the original sequence member. In the above example, the new *SpaceElementSequence* is therefore given by the ordered set $\{M_3, M_4, M_5, M_6, M_7\}$. Since the space cells M_4 to M_6 are spatially contained in C_2 they are assigned the sequence number 2 which is identical to the sequence number of C_2 in the original sequence. For the same reason, M_3 is assigned the sequence number 1 and M_7 carries the sequence number 3. The resulting sequence is valid since there exists a path from M_3 to M_7 which involves a *SpaceElement* feature from every unique sequence number. In fact, in this simple example, there exits only one path which visits all space cells contained in the new sequence.

The illustrated mechanism for creating a new *SpaceElementSequence* can again be realized fully automatically. The information about sequences is hence not lost when applying the operations of the space layer algebra. However, the fact that a sequence is valid on a given space layer does not imply that the sequence remains valid under those operations. This is exemplified along the modified setting shown in figure 177. Assume the subspace layer L_{Sub} only contains Sub_2 which denotes an obstacle for wheelchair users (e.g., a step). Consequently, L_{Sub} can be subtracted from L_{Topo} using the difference operation in order to retrieve a view on the topographic space without the obstacle which is suitable for a wheelchair user. The resulting space layer L_{diff} is shown below.

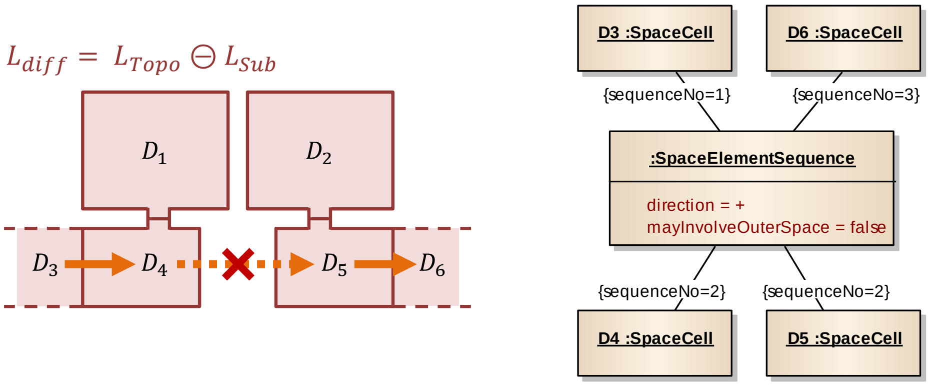


Figure 177: Invalid SpaceElementSequence feature after difference operation.

According to the illustrated procedure, the sequence $\{C_1, C_2, C_3\}$ on L_{Topo} can be translated into a new sequence $\{D_3, D_4, D_5, D_6\}$ on L_{diff} with D_4 and D_5 sharing the same sequence number 2. Obviously, there is a path from D_3 to D_4 and from D_5 to D_6. However, D_4 and D_5 are not topologically adjacent and thus are not connected directly. A path between both therefore involves the outer space cell in this example. If we assume that traversing the outer space cell is to be avoided (and thus the attribute *mayInvolveOuterSpace* is set to false on the sequence), then there is no path from D_3 to D_6 via D_4 or D_5 and the sequence on L_{diff} is invalid.

If a *SpaceElement* feature of the original sequence is not carried to the resulting space layer at all, then its associated sequence number is not linked to a *SpaceElement* feature in the newly created sequence. For instance, suppose that the subspace layer L_{Sub} provides a single space cell having the same spatial extent as C_2 itself. Then after a difference operation there are no space cells on L_{diff} which are contained in or covered by C_2. Hence, the new sequence will not contain a space cell for the sequence number 2.

4.4.1.5 Route Package

The *Route* package supplies classes and types to describe routes and supporting guidance data based on the conceptual entities introduced in the *Spatial Representation* package of the MLSEM schema. A route represents a possible path for a navigation user to travel from a start to a destination location along some portion of the graph-based conceptualization of indoor space, and commonly renders the response to a user-triggered path query. From the approaches to indoor space modelling discussed in chapter 2.2, most works solely focus on the spatial and conceptual description of the interior built environment and its mapping onto a navigation graph structure which captures the entire indoor space and hence is meant to support the process of path planning. Only few approaches additionally address the conceptual modelling of routes and guidance information as well as the exchange of route data decoupled from the entire indoor space model. However, the route presentation as well as the provision of route instructions as well as visual or textual aids and descriptors has been identified a major challenge to indoor navigation (cf. chapter 1.2). Moreover, the decoupled exchange of route data is especially important in case the tasks of path planning and route presentation are to be distributed on different systems, e.g. on a routing service and a mobile end-user device. Whereas traditional car navigation systems are mostly self-contained, distributed systems architectures are typically assumed in indoor navigation scenarios.

Conceptual entities for the modelling of user-specific routes or paths are mainly discussed in conceptual-based approaches to indoor space modelling (cf. chapter 2.2.4) such as OntoNav (Anagnostopoulos et al. 2005, Tsetsos et al. 2006), ONALIN (Dudas et al. 2009), or the unified ontology for navigation in indoor and outdoor space proposed by (Yang & Worboys 2011). An abstract model for route and route guidance is also presented in (Gilliéron & Merminod 2003). The *Route* package of the MLSEM has been designed to comprise the concepts introduced by these approaches. The UML data model is depicted in figure 178.

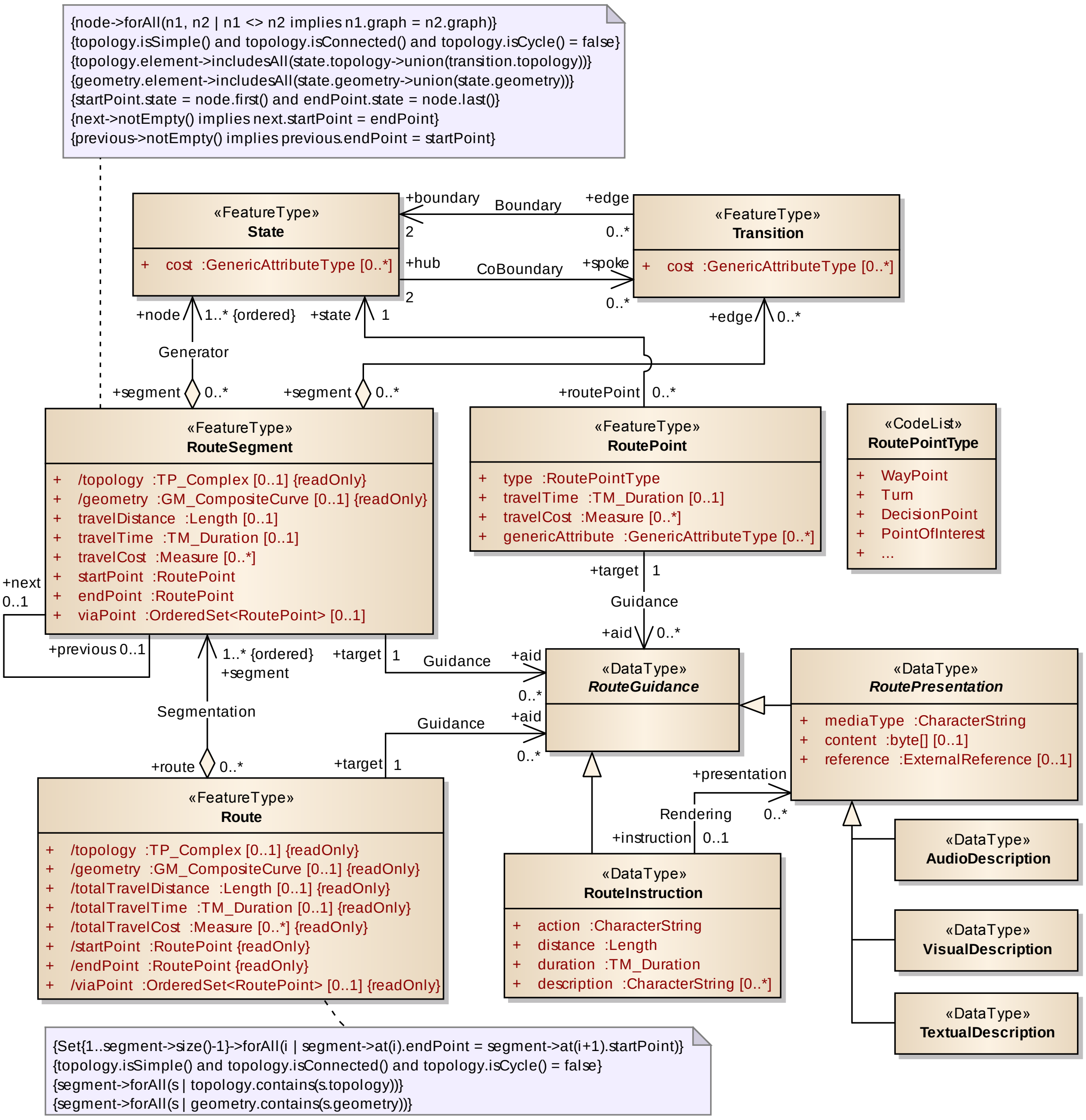

{node->forAll(n1, n2 | n1 <> n2 implies n1.graph = n2.graph)}
{topology.isSimple() and topology.isConnected() and topology.isCycle() = false}
{topology.element->includesAll(state.topology->union(transition.topology))}
{geometry.element->includesAll(state.geometry->union(state.geometry))}
{startPoint.state = node.first() and endPoint.state = node.last()}
{next->notEmpty() implies next.startPoint = endPoint}
{previous->notEmpty() implies previous.endPoint = startPoint}
«FeatureType»
State
+ cost :GenericAttributeType [0..*]
+boundary
Boundary
+edge
2
0..*
+hub
CoBoundary
+spoke
2
0..*
«FeatureType»
Transition
+ cost :GenericAttributeType [0..*]
+node
1..* {ordered}
+state
1
+edge
0..*
Generator
+segment
0..*
+segment
0..*
+routePoint
0..*
«FeatureType»
RouteSegment
+ /topology :TP_Complex [0..1] {readOnly}
+ /geometry :GM_CompositeCurve [0..1] {readOnly}
+ travelDistance :Length [0..1]
+ travelTime :TM_Duration [0..1]
+ travelCost :Measure [0..*]
+ startPoint :RoutePoint
+ endPoint :RoutePoint
+ viaPoint :OrderedSet<RoutePoint> [0..1]
«FeatureType»
RoutePoint
+ type :RoutePointType
+ travelTime :TM_Duration [0..1]
+ travelCost :Measure [0..*]
+ genericAttribute :GenericAttributeType [0..*]
«CodeList»
RoutePointType
+ WayPoint
+ Turn
+ DecisionPoint
+ PointOfInterest
+ ...
+next
0..1
+previous 0..1
1..* {ordered}
+segment
+target
1
Guidance
+aid
0..*
+target
1
Guidance
+aid
0..*
Segmentation
Guidance
+aid
0..*
+route
0..*
+target
1
«DataType»
RouteGuidance
«DataType»
RoutePresentation
+ mediaType :CharacterString
+ content :byte[] [0..1]
+ reference :ExternalReference [0..1]
+presentation
Rendering
0..*
+instruction
0..1
«FeatureType»
Route
+ /topology :TP_Complex [0..1] {readOnly}
+ /geometry :GM_CompositeCurve [0..1] {readOnly}
+ /totalTravelDistance :Length [0..1] {readOnly}
+ /totalTravelTime :TM_Duration [0..1] {readOnly}
+ /totalTravelCost :Measure [0..*] {readOnly}
+ /startPoint :RoutePoint {readOnly}
+ /endPoint :RoutePoint {readOnly}
+ /viaPoint :OrderedSet<RoutePoint> [0..1] {readOnly}
«DataType»
RouteInstruction
+ action :CharacterString
+ distance :Length
+ duration :TM_Duration
+ description :CharacterString [0..*]
«DataType»
AudioDescription
«DataType»
VisualDescription
«DataType»
TextualDescription
{Set{1..segment->size()-1}->forAll(i | segment->at(i).endPoint = segment->at(i+1).startPoint)}
{topology.isSimple() and topology.isConnected() and topology.isCycle() = false}
{segment->forAll(s | topology.contains(s.topology))}
{segment->forAll(s | geometry.contains(s.geometry))}

Figure 178: The contents of the Route package.

In the context of the MLSEM schema, a *Route* is expressed in terms of one or more *RouteSegment* features as well as a number of *RoutePoint* features. A *RoutePoint* represents an important point along the route such as a waypoint that has to be traversed, a turn or decision point which involves an action of the user, or a point of interest that may also serve as landmark supporting self-localization. The *RoutePoint* feature type is realized as wrapper for the *State* feature type and decorates it with additional thematic properties. Thus, a route point is associated with precisely one dual node in the *IntraLayerGraph* of a given *SpaceLayer* and by this provides access to the corresponding *SpaceCell*, its thematic attributes, and its primal space representation. The *type* of a *RoutePoint* feature is given by a value from the extensible code list *RoutePointType* which predefines *WayPoint*, *Turn*, *DecisionPoint*, and *PointOfInterest*. Every *RoutePoint* can provide a *travelTime* which denotes the time required for a navigation user to pass this part of the route. The value domain of *travelTime* is *TM_Duration* from the Temporal Schema which allows describing a length or distance in the temporal dimension. Additional travel costs for a route point including time-base, distance, or monetary costs can be provided through the *travelCost* attribute. Its value is meant to be the output of a cost function that takes one or more *cost* values (taken from the wrapped *State* feature, cf. chapter 4.4.1.1) as input. For each *travelCost*, the *Measure* data type from ISO/TS 19103:2005 facilitates to provide a reference to the definition of the cost unit. Moreover, further semantic characteristics can be attached to a *RoutePoint* feature using *generic attributes*.

A *RouteSegment* is a simply connected, non-cyclic, directed subset of an *IntraLayerGraph* consisting of an ordered set of one or more *State* features being linked in sequence by *Transition* features. The *State* features are said to be the *Generator* of the *RouteSegment* from which the *Transition* instances implicitly follow. In contrast to the more general notion of a *SpaceElementSequence*, a *RouteSegment* is hence expressed as path on the intra-layer graph and must contain all graph elements constituting the path. The spatial description of a *RouteSegment* is realized through the spatial attributes *topology* and *geometry* whose values are instances of *TP_Complex* and *GM_CompositeCurve* (acting as complex geometry) that contain the topological respectively geometric primitives of the associated *State* and *Transition* features and thus are necessarily subcomplexes of the spatial representation of the entire *IntraLayerGraph*. For this reason, both *topology* and *geometry* are marked as derived and read-only. Moreover, either attribute is optional like with the spatial characteristics of the *IntraLayerGraph*. The described constraints for *RouteSegment* are mapped onto the following OCL invariant.

```
context RouteSegment inv:
    node->forAll(n1, n2 | n1 <> n2 implies n1.graph = n2.graph) and
    topology.isSimple() and topology.isConnected() and topology.isCycle() = false and
    topology.element->includesAll(state.topology->union(transition.topology)) and
    geometry.element->includesAll(state.geometry->union(state.geometry))
```

As for semantic information, the *RouteSegment* feature types defines the optional attributes *travelDistance, travelTime*, and *travelCost*. The *travelDistance* captures the distance or, more precisely, the length of the route segment which, for example, can be derived as sum of the lengths of the *GM_Curve* primitives associated with the contained *Transition* features. The *travelDistance* may have zero length in case the *RouteSegment* only contains a single *State* instance. Both the *travelTime* and *travelCost* attributes are identically defined as for the *RoutePoint* feature type. In addition to these cost-based values, each *RouteSegment* has to announce its start and end point as instance of *RoutePoint* using the mandatory *startPoint* and *endPoint* attributes. Precisely, the start point has to wrap the first *node* within the ordered *Generator* association, whereas the end point corresponds to the last *node*. If the *RouteSegment* only consists of a single *State* feature, then both points are obviously identical. The provision of an start and an end point induces a natural direction of the route segment. Optionally, arbitrarily many *State* features in-between the start and end point can be classified as further route points and provided as ordered set of *RoutePoint* instances through the *viaPoint* attribute. *RouteSegment* features can be linked to each other in order to point to the *next* or the *previous* segment in a sequence. Additional constraints are applied to these relations in order to enforce that the *endPoint* of a given route segment is the *startPoint* of its successor (and vice versa).

```
context RouteSegment inv:
    startPoint.state = node.first() and endPoint.state = node.last() and
    (next->notEmpty() implies next.startPoint = endPoint) and
    (previous->notEmpty() implies previous.endPoint = startPoint)
```

Finally, a *Route* is built from a set of *RouteSegment* features ordered by their start and end points so that the *endPoint* of a given segment in the set agrees with the *startPoint* of its consecutive segment and the entire *Route* itself renders a simply connected, non-cyclic, directed subset of an *IntraLayerGraph*. Similar to *RouteSegment*, the *topology* and *geometry* of a *Route* feature are described by a topological respectively geometric complex (in the form of a *GM_CompositeCurve*) both of which are derived from the spatial union of the route segments. The OCL constraint shown below expresses the corresponding consistency requirements.

```
context Route inv:
    Set{1..segment->size()-1}->forAll(i | segment->at(i).endPoint = segment->at(i+1).startPoint) and
    topology.isSimple() and topology.isConnected() and topology.isCycle() = false and
    segment->forAll(s | topology.contains(s.topology)) and
    segment->forAll(s | geometry.contains(s.geometry))
```

The *totalTravelDistance*, *totalTravelTime*, and *totalTravelCost* attributes of *Route* represent the total values of the travel distance, time, and costs associated with the individual route segments. The basic assumption is that the total route values can be computed as the sum of the values of the segments or by applying a further cost function. In either case, the attributes can be marked as derived and read-only. The same holds for the attributes *startPoint*, *endPoint*, and *viaPoint*. Whereas the start point (end point) of the route obviously is the start point (end point) of the first (last) route segment, the ordered *viaPoint* set contains all route points in-between without duplicates.

Every element of a route, i.e. every individual *RoutePoint* and *RouteSegment* feature as well as the *Route* itself, can be enriched with guidance information by assigning one or more instances of the abstract data type *RouteGuidance* to the feature instance. Guidance information is further distinguished into route instructions and presentations of the route modelled as the subtypes *RouteInstruction* and *RoutePresentation*.

A *RouteInstruction* provides information needed while travelling along a route that allows that route to be traversed. Route instructions should be given so that the route can be navigated by executing a list of consecutive instructions. The *action* attribute of *RouteInstruction* describes the actual action to be taken by the navigation user and thus represents a navigation command. For example, at a single *RoutePoint*, an action might be to *"turn left"* or to *"take the elevator to the fifth floor"*. For a *RouteSegment*, the action might cover multiple commands such as *"turn left, follow the corridor, and turn right at its end"*, or it might represent a high-level command such as *"enter the building and take the main elevator to the fifth floor"* while more detailed instructions (e.g., on how to find the main elevator) are provided as actions on individual route points. Likewise, a *Route* might be assigned individual high-level instructions, whereas a detailed instruction list along the entire route simply follows from the instructions assigned to its ordered set of route segments and route points. Each *RouteInstruction* can additionally provide information about the *distance* and *duration* required to execute the action. Moreover, the instruction can be further augmented with one or more textual *descriptions* providing advisory that might be useful in understanding the action to be taken.

A *RoutePresentation* reflects the rendering of a (part of a) route as *audio description* (e.g., spoken commands), *visual description* (e.g., 2-dimensional maps or images as well as 3-dimensional visualization models), or *textual description* (e.g., textual commands provided as web page or in printed form). The appropriate media type depends on the physical and perceptual capabilities of the navigation user as well as the presentation capabilities of the end-user device. A route presentation may support one or more route instructions but may also be provided independently. Moreover, multiple presentations of different media types may be associated with a single (part of a) route. The possibility to assign descriptions to single *RoutePoint* features, *RouteSegment* features, and the *Route* feature itself again yields a high flexibility in providing route presentations with different scope and detail level. For example, for a single *RoutePoint*, a 3-dimensional graphics model including the route geometry, navigation symbols, and further visual aids may be rendered that lets the user virtually explore the ambient indoor space at the spot. On the level of a *RouteSegment*, a more generalized view on the path embedded in a 2-dimensional floor plan might suffice. For the entire *Route*, a video might be attached showing a fly-through of the indoor environment along the route. The geometric information for visual presentations may hereby be taken from the primal space representations of the *SpaceCell* instances associated with the route elements and possibly from associated *SourceObject* data (cf. chapter 4.4.1.3). The abstract *RoutePresentation* class is subtyped into *AudioDescription*, *VisualDescription*, and *TextualDescription*. The encoding of the presentation data has to be specified through the *mediaType* attribute. Similar to the *SourceObject* feature, the actual data can then be given either inline as binary array or by pointing to an external data source using an *ExternalReference*. The latter allows for preformatted requests to web-based portrayal services such as the OGC Web Map Service (WMS), Web 3D Service (W3DS), or Web View Service (WVS).

It is important to note that the *Route* package of the MLSEM schema only provides an interface that addresses the description, storage, and exchange of route and guidance information and that needs to be implemented by a navigation system. Therefore, algorithms and mechanisms for the (automatic) generation of route instructions as well as presentations are not included but are outside the scope of this thesis. In chapter 2.2.2, the approach of (Lorenz et al. 2006) for the automatic derivation of route commands from a cell-based indoor space model has been presented which can be adopted for the domain of the MLSEM. The proper presentation of routes is still being researched (e.g., Giudice et al. 2010, Lorenz et al. 2010). The MLSEM data model does not anticipate a choice of presentation but rather is flexible enough to link a route with any type of guidance information available.

4.4.1.6 Model Linkage Package

The *Model Linkage* package defines a general mechanism for connecting an instance of the MLSEM with a navigation network from another MLSEM model or even from different sources. This is required in cases where the spatial scope and extent of a single MLSEM instance is limited to a subset of the entire navigation space. For example, assume a navigation system shall support the movement between the indoor spaces of two adjacent buildings on the same site. If the indoor space of each building is mapped by its own 3-dimensional space layer complex, then the corresponding multilayered graphs are disconnected and hence are not suitable for finding paths between two places from either building. Second, in two dimensions, a single space layer complex is even restricted to the mapping of a single floor within the same building. If, in contrast, each floor of that facility is carried to a topographic space layer within the same 2-dimensional space layer complex, then the interior spaces on the separate space layers would necessarily overlap and thus be connected by inter-layer edges. Since most of the spatial entities on different building floors however have no spatial overlaps in the real world, this graph representation would obviously provide a false conceptualization of the indoor space. It easily follows that a further conceptual entity is required to express the linkage between the separate building floors. A third example for the *Model Linkage* package is to connect an MLSEM representation to an existing navigation structure that follows a different modelling and space representation paradigm. For instance, the indoor space of a building given as instance of the MLSEM may be linked to a representation of the outdoor transportation network (e.g., the road network given according to a standard such as GDF) in order to connect the indoor space with the outdoor world and to enable the determination of outdoor routes to and from the building.

A summary of the *Model Linkage* package is presented as UML diagram in figure 179. The central concept for linking two navigation networks is *TransferTransition*. A *TransferTransition* denotes a transition between two elements from either network structure which themselves are wrapped by the feature type *TransferState*. When applied to an instance of the MLSEM, *TransferState* simply acts as a proxy for a *State* feature and thus for a dual node in the corresponding multilayered graph. In all other cases, a *TransferState* shall reference a node within the navigation graph to be linked through its *externalNode* property using an *ExternalReference*. A *TransferTransition* then establishes an edge between the two *TransferState* nodes in order to connect the separate graph representations. Put differently, a *TransferState* marks an interface within a navigation graph that can be connected to a *TransferState* interface within another navigation graph through a *TransferTransition*. It follows that, in terms of the MLSEM, the linkage between two navigation models is realized in dual space. In case *TransferState* wraps an instance of the *State* feature type, then this *State* feature provides access to both the primal space representation and the semantic properties of the associated *SpaceCell* feature. Otherwise, it depends on the modelling approach being applied to the navigation model to be linked whether this additional information is available. Note that *TransferTransition* semantically coincides with the *Transition* feature type, and thus is meant to expresses a topological adjacency relationship between the spaces represented by the *TransferState* feature instances.

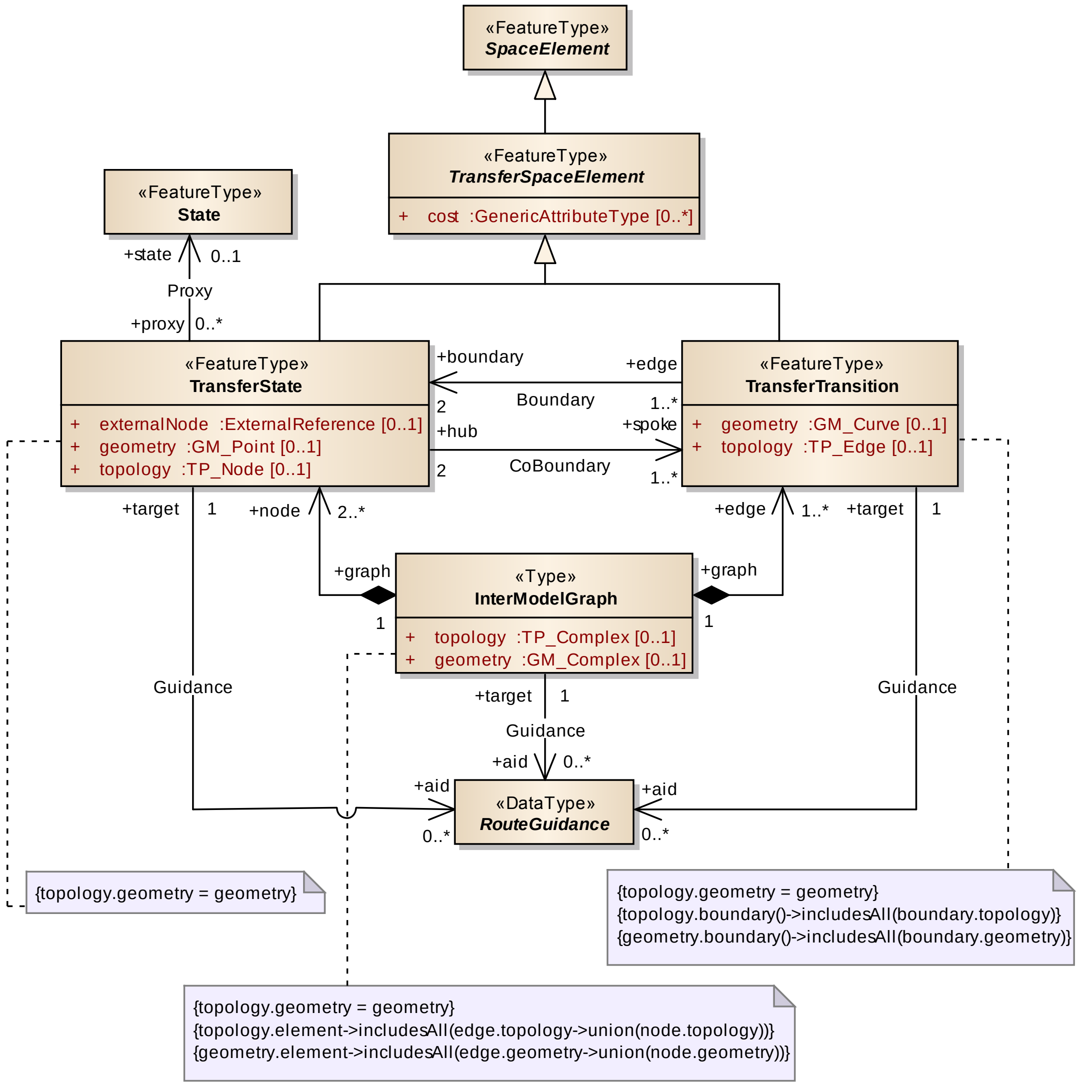

«FeatureType»
SpaceElement
«FeatureType»
TransferSpaceElement
+ cost :GenericAttributeType [0..*]
«FeatureType»
State
+state
0..1
Proxy
+proxy
0..*
«FeatureType»
TransferState
+ externalNode :ExternalReference [0..1]
+ geometry :GM_Point [0..1]
+ topology :TP_Node [0..1]
+boundary
+edge
2
Boundary
1..*
+hub
+spoke
2
CoBoundary
1..*
«FeatureType»
TransferTransition
+ geometry :GM_Curve [0..1]
+ topology :TP_Edge [0..1]
+target
1
+node
2..*
+edge
1..*
+target
1
+graph
1
«Type»
InterModelGraph
+ topology :TP_Complex [0..1]
+ geometry :GM_Complex [0..1]
+graph
1
Guidance
Guidance
+target
1
Guidance
+aid
0..*
+aid
0..*
«DataType»
RouteGuidance
+aid
0..*
{topology.geometry = geometry}
{topology.geometry = geometry}
{topology.boundary()->includesAll(boundary.topology)}
{geometry.boundary()->includesAll(boundary.geometry)}
{topology.geometry = geometry}
{topology.element->includesAll(edge.topology->union(node.topology))}
{geometry.element->includesAll(edge.geometry->union(node.geometry))}

Figure 179: The contents of the Model Linkage package.

TransferState and *TransferTransition* are derived by specialization from the feature type *TransferSpaceElement* which is implemented as subtype of *SpaceElement*. They hence inherit the properties of *SpaceElement* that allow for semantically classifying the involved transfer spaces and their transitions. *TransferSpaceElement* further adds the possibility to model the *cost* for travelling between the connected navigation structures. The value domain of this *cost* value is given as *GenericAttributeType* in order to support arbitrary units of cost such as distance, monetary, or time-based measures. Similar to *State* and *Transition*, also *TransferState* and *TransferTransition* are related by *Boundary* and *CoBoundary* associations that enforce every *TransferTransition* to be bounded by exactly two *TransferState* features, and every *TransferState* to have a non-empty coboundary. Likewise, both *TransferState* and *TransferTransition* may have a geometric-topological representation based on 0-dimensional respectively 1-dimensional primitives from the Spatial Schema. The spatial description of a *TransferState* feature may hereby reference or repeat the spatial description of the related entity from the original navigation structure. It obviously follows that the *TransferState* and *TransferTransition* features constitute an undirected, weighted graph which is conceptually mapped by the *InterModelGraph* type. This inter-model graph needs to be evaluated in path searches in order to find routes between the linked navigation structures. An instance of *InterModelGraph* at minimum consists of a single *TransferTransition* feature and its boundary nodes. Its spatial characteristics are available through the attributes *geometry* and *topology* both of which realize a complex spatial object (*GM_Complex* respectively *TP_Complex*) being formed by the spatial primitives of the graph elements. The spatio-semantic consistency of the inter-model graph is ensured by means of OCL constraints that are identically defined as for the *IntraLayerGraph* feature type and its components (cf. chapter 4.4.1.1 for the discussion of the OCL constraints). Finally, every *TransferState*, *TransferTransition*, and *InterModelGraph* feature can be assigned *RouteGuidance* information in order to support navigation users in traveling between the separate navigation networks.

The usage of the *Model Linkage* package is exemplified in figure 180. It shows two 2-dimensional instances of the MLSEM capturing subsequent floors of the same building. In order to keep the example simple, the space layer complex of each MLSEM only contains a single topographic space layer. The floors are connected by a staircase which is represented by the space cell S on both space layers. On the ground floor shown on the left of figure 180, the staircase is adjacent to a corridor C which is linked through doors to two rooms R_1 and R_2. The first floor on the right of figure 180 differs in that the corridor gives passage to a single room only. At the bottom of figure 180, the corresponding multilayered graph representations of the MLSEM instances are depicted.[49] The dual nodes of the stair cells in either multilayered graph are referenced by two instances TS_1 and TS_2 of *TransferState* which are linked by a *TransferTransition*. The resulting *InterModelGraph* hence creates a linkage between both MLSEM instances and encodes the fact that the building floors can be reached from each other using the stairs.[50]

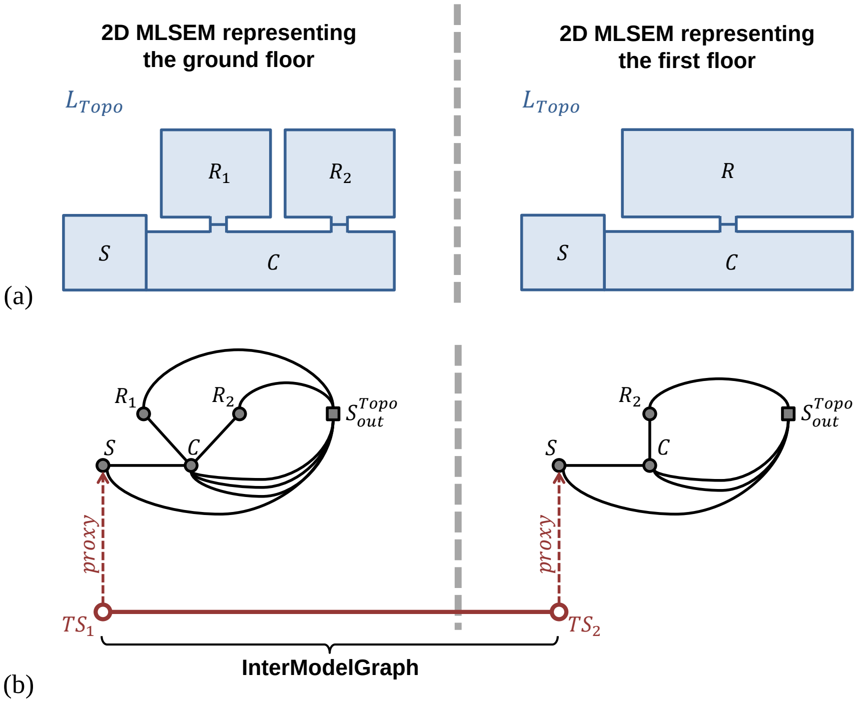


Figure 180: Two 2*-dimensional instances of the MLSEM linked by an InterModelGraph.*

A second example is provided in the following figure 181. On the left of figure 181, the same ground floor as above is shown and realized as 2-dimensional instance of the MLSEM. It however is extended by an additional space cell E representing the entrance hall of the building. The multilayered graph in dual space is depicted at the bottom of figure 181. Assume the road network around the building is additionally available as graph-based conceptualization in a model other than the MLSEM (cf. right of figure 181). Two instances of *TransferState* can then be used to identify the interfaces between both models, i.e. the dual node of the entrance hall E and a node entity N within the road network. A *TransferTransition* connecting the transfer states then links the multilayered graph of the MLSEM instance with the road network and denotes that navigation users will enter the road network at N when leaving the building via E. Note that the fact that the MLSEM instance is given in two dimensions does not imply that the navigation structure linked by a *TransferTransition* also has to be modelled in two dimensions (or vice versa).

[49] Note that in either multilayered graph the dual node of the corridor is linked to the dual node of the outer space by more than one dual edge. The reason for this is that, in two dimensions, the boundary of the corridor space cell has to be described by more than one 1-dimensional boundary cell. See chapter 3.1.3.1 for a comprehensive discussion.

[50] Obviously, if the indoor space was represented by a 3-dimensional MLSEM instance, then the staircase could either be represented by one or two space cells on the same topographic space layer without additional *TransferTransition* edges.

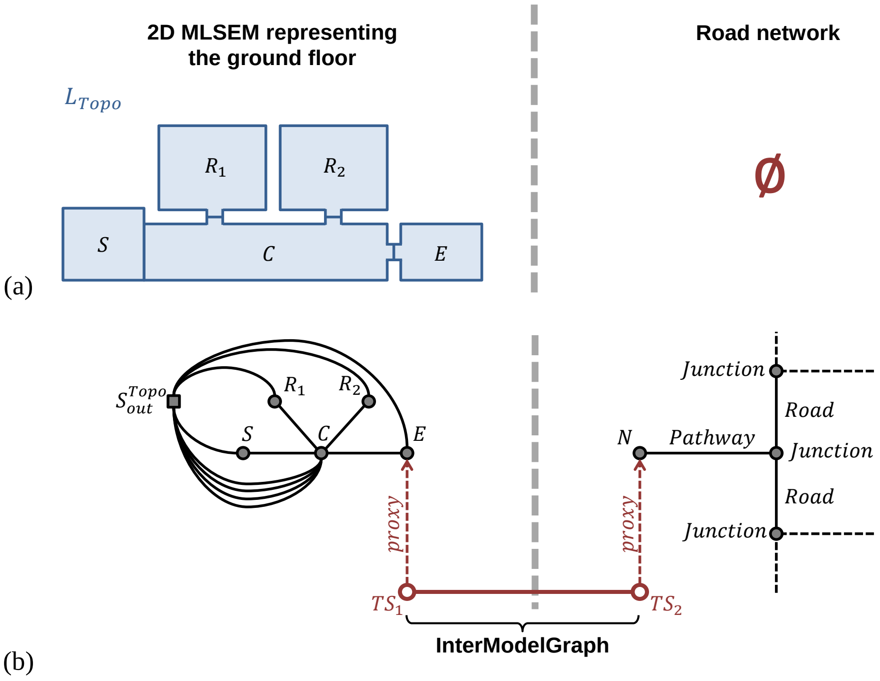


Figure 181: A 2*-dimensional instance of the MLSEM linked to a road network using the InterModelGraph.*

4.4.2 MLSEM Data Exchange Format

In order to facilitate the processing, transport, and storage of indoor space information according to the MLSEM schema, the conceptual data model has to be transformed into a platform-specific model which realizes the conceptual and thus platform-independent model elements for a given technological platform or technology stack such as specific programming languages, database systems, or file-based encodings.

In the context of the ISO 19100 family of international standards, the *Geography Markup Language* (GML) provides the means to express an ISO 19109 conformant application schema in XML Schema and, as such, to realize an XML encoding for data that implements the logical structure of the geographic features as well as both their spatial and non-spatial properties as described in the ISO 19109 application schema. GML is an ISO 19118:2011 compliant XML grammar written in XML Schema that, on the one hand, specifies XML encodings for several of the conceptual classes defined in the ISO 19100 series including the conceptual models from ISO/TS 19103, ISO 19107, ISO 19108, and ISO 19109. On the other hand, GML itself serves as modelling language for geographic systems and thus introduces its own concepts and a rich set of modelling primitives for the description of open and vendor-neutral geospatial application schemas as well as for the transport and storage of geographic information in XML, and moreover defines the corresponding XML Schema syntax, mechanisms, and conventions. The GML concepts are mostly drawn from and identically defined as the concepts specified in the ISO 19100 series (e.g., the notion of a geographic feature and its geometric and topological description) but also comprise conceptual entities not yet being captured by ISO 19100 specifications (e.g., the notion of feature collections and dynamic features). Following ISO 19109, also GML refers to an application schema as set of feature types being essential in the context of an application or application domain. An application schema specified in XML Schema based on the modelling language and rules defined by GML is said to be a *GML application schema*. It follows that GML application schemas are platform-specific encoding models with XML Schema providing the technology stack. As with other XML based grammars, two parts of the grammar can be distinguished: the GML application schema that describes the GML document, and the GML document (or instance document) that contains the actual data.

GML was initially developed within the Open Geospatial Consortium and was first published as OGC Recommendation Paper in May 2000.[51] In 2001, the OGC membership adopted GML version 2.0 as official OGC standard. With this and its later follow-up versions, GML became increasingly accepted and implemented in the GIS community. The early versions of GML were completely based on the OGC Abstract Specification suite. However,

[51] GML version 1.0 was presented in form of three profiles, two of which were implemented using *Document Type Definitions* (DTD) whereas the third profile was based on the *Resource Description Framework* (RDF) as XML technology. The approach of using static schemas specified in XML Schema was first introduced in GML 2.0.

the significant intersection of the work of OGC and ISO/TC 211 soon became apparent but only few encodings in GML were conformant with the abstract specifications of the ISO 19100 series. Hence, both OGC and ISO/TC 211 joined forces in the further alignment of their standards work and in the development of GML. With the release of GML 3.0 in 2002 and its minor update to GML 3.1.1 in 2004 (Cox et al. 2012), the OGC achieved major steps towards the consistency of GML with ISO specifications. In 2007, the revision of GML to version 3.2.1 (Portele 2007) was mutually approved by both organisations and thus has been incorporated into the ISO 19100 family as ISO 19136:2007 standard. The latest release of GML 3.3 has been issued by the OGC in 2012 (Portele 2012) and will lead to a future revision of ISO 19136:2007.

The GML specification recognizes two alternative approaches for exposing geographic data with GML and hence for constructing a GML application schema. First, the GML application schema is directly specified in XML Schema by adhering to the modelling principles and rules provided by GML. An underlying conceptual and platform-independent model is not presupposed for this alternative. Second, an ISO 19109 conformant application schema described in UML is mapped onto a corresponding GML application schema. GML specifies normative rules for this UML-to-GML mapping which need to be obeyed by conformant systems and implementations. This second approach more closely follows the idea of transforming a platform-independent model into a platform-specific representation as envisioned by the Model-Driven Architecture (MDA) approach in software and systems engineering.[52] In fact, both commercial and free software tools are available that enable the (semi-)automatic conversion from UML to GML.[53]

In (Nagel et al. 2010), a first mapping of the former data model of the MLSEM to GML 3.1.1 has been presented under the name *IndoorGML*. In the context of this thesis, this GML mapping has been comprehensively reworked in order to capture the redesign of the MLSEM application schema and its various extensions as developed in the previous chapter 4.4.1. A major goal of the work undertaken in this thesis was both to ensure the conformance of the conceptual MLSEM data model with the ISO abstract specifications and to realize its rule-based UML-to-GML mapping in accordance with GML/ISO 19136. The translation of the conceptual MLSEM classes and their relations into corresponding XML Schema components was hereby carried out by applying the normative GML mapping rules in a straightforward manner. A detailed description of these normative rules goes beyond the scope of this thesis but is provided as part of the GML 3.2.1 specification (cf. annex E in Portele 2007). The reader is thus referred to this document. The resulting GML 3.3 application schema is presented in appendix B and an example instance document is presented in chapter 7.1. It finally realizes a data encoding format that allows for structuring, storing and transporting indoor space and navigation data between multiple and different computer systems according to the mathematical and conceptual framework of the MLSEM as well as in compliance with international standards. Moreover, and due to the GML based implementation, the suite of standardized OGC Web Services for geographic information is applicable to the MLSEM schema and corresponding instance documents. For example, clients and servers with interfaces that implement the OGC Web Feature Service can be used to read and write MLSEM data over the internet and to even execute transactions on the data. Likewise a range of portrayal services is available for data presentation and styling both in two and three dimensions, or for the distributed processing of data.

4.5 Integration with Existing GIS Standards for Location-Based Services

In this chapter, the relation of the MLSEM application schema to existing standards for the description of location-based services (LBS) in the field of GIS is illustrated. Location-based services are distributed services meant to support applications for which the location and/or time data of a user equipped with a mobile device render control features for the behaviour of the application. Location-based services exploit the position of the mobile user in order to offer added value information. Typical examples include simple *"Where am I?"* services, more complex discovery and directory services providing information about the location context of the user (e.g., in order to find nearby places, products, or services), and (reverse) geocoding services for translating the location into an addressing schema and vice versa. If the location information shall support the determination of an optimal path to a destination location then the service is commonly called *navigation service*. Services that render geographic information for display on a mobile device are said to be *presentation* or *rendering services*. Location-based information

[52] MDA was first introduced by the Object Management Group (OMG). See http://www.omg.org/mda/ for more information.

[53] *ShapeChange* is an example for a freely available mapping tool. See http://shapechange.net/ for more information.

can also be collected about the mobile user. For example, a service providing data about location changes is usually said to be a *tracking service*.

The MLSEM is a spatio-semantic information model for describing indoor spaces for indoor navigation. It comprises feature and data types, as well as attributes, relationships, and operations associated with those types, which allow for realizing rich indoor space models that underpin location-based services. However, the MLSEM schema is not intended to address the definition and implementation of the services themselves which includes, amongst others, the description of service capabilities and interfaces, communication and messaging protocols between servers and mobile clients, or the distributed service architecture. These complementary topics are covered by two existing international GIS standards, namely the ISO 19133:2005 specification *"Geographic information – Location-based services – Tracking and navigation"* from the ISO 19100 standards series as well as the OGC standard *OpenGIS Location Services* (OpenLS) (Mabrouk 2008). Both standards and the integration of the MLSEM are presented in the following.

4.5.1 ISO 19133 – Location-based services

The scope of ISO 19133 is illustrated along its proposed conceptual architecture for location-based services as depicted in the following figure 182. It defines four core LBSs that are called *Navigation Service*, *Tracking Service*, *Location Transformation Service*, and *Navigation Rendering Service*. The basic assumption of ISO 19133 is that these services are made available on the web to be accessed by mobile devices. ISO 19133 does not presuppose any specific requirements on the underlying network platform or protocols. Moreover, services not specifically marked as being part of ISO 19133 are assumed to provide required functionality but are not specified in detail.

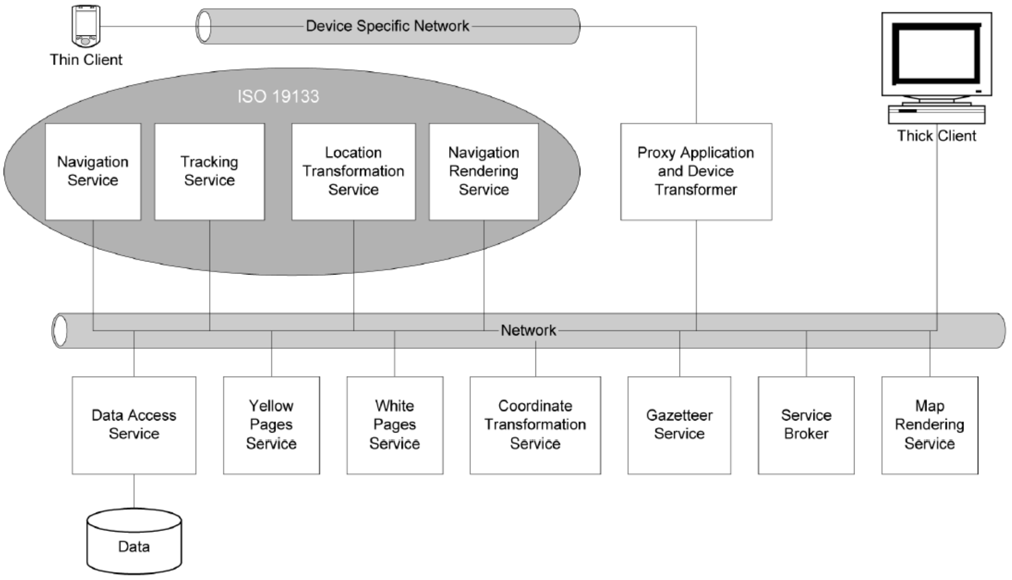


Figure 182: Conceptual architecture for location-based services as specified by ISO 19133.

The LBSs provide external interfaces to a navigation application in order to trigger the tasks of path planning, localization, tracking, and presentation, and to return the result of corresponding queries to the mobile device (e.g., a specific route, location change, or route presentation). According to ISO 19133, the services may implement the functionality needed to perform these tasks based on the underlying *Data* repository, or may delegate them to appropriate internal services. Hence, ISO 19133 does not specify algorithms or techniques for answering navigation queries (e.g. for finding or rendering a route), but leaves these topics to a concrete implementation. However, it defines the data types for delivering the content in responses to navigation queries which results in an overlap of the *Navigation (Rendering) Service* with the MLSEM, and in particular with its *Route* package as defined in chapter 4.4.1.5.

The conceptual data model of the ISO 19133 *Navigation Service* is shown in figure 183. *NS_NavigationService*[54] is the main interface for the communication between a mobile client and a *Navigation Service*. The functionality

[54] By convention, the names of classes from the *Navigation Service* are prefixed with "*NS_*", whereas classes from the network model of ISO 19133 receive the prefix "*NT_*".

of the service is offered by the *route* operation which takes an *NS_RouteRequest* as input and returns an *NS_RouteResponse* that contains the user-specific route. Route requests are differentiated into the types *basic*, *predictive*, *dynamic*, and *complex* as defined by the *NS_RouteRequestType* code list. A navigation service needs to announce the supported request types through its *capabilities()* operation and has to properly handle requests by using appropriate path finding algorithms.

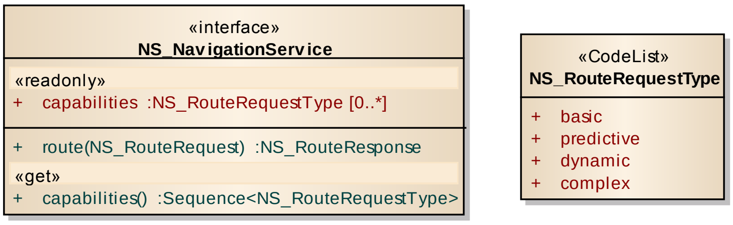

«interface»
NS_NavigationService
«readonly»
+ capabilities :NS_RouteRequestType [0..*]
+ route(NS_RouteRequest) :NS_RouteResponse
«get»
+ capabilities() :Sequence<NS_RouteRequestType>
«CodeList»
NS_RouteRequestType
+ basic
+ predictive
+ dynamic
+ complex

Figure 183: NS_NavigationService interface of ISO 19133:2005.

An excerpt of the UML diagram for *NS_RouteRequest* is shown below. It represents the actual request of a navigation user and specifies the start and end location of the requested route as well as additional waypoints that shall be traversed or avoided. Further attributes depend on the type of the route request and include the planned departure and arrival times, a cost function that chooses amongst route candidates, the route preferences of the user, or requested route renderings. Note that again only interfaces but no implementations are provided, for example, for specifying a cost function that evaluates distance, monetary, or time-based cost values using the *NS_CostFunction* type.[55] The resulting route is then encapsulated in an *NS_RouteResponse* and given by the type *NT_Route*.

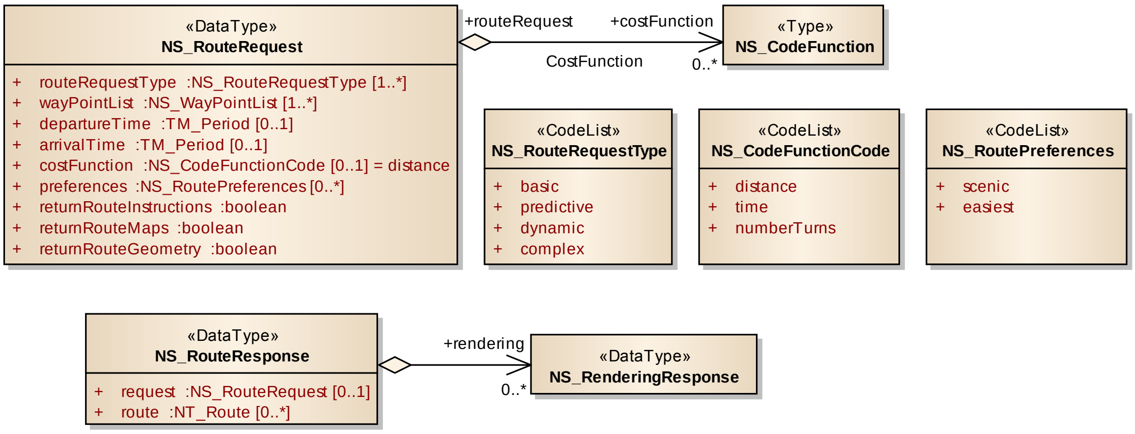

«DataType»
NS_RouteRequest
+ routeRequestType :NS_RouteRequestType [1..*]
+ wayPointList :NS_WayPointList [1..*]
+ departureTime :TM_Period [0..1]
+ arrivalTime :TM_Period [0..1]
+ costFunction :NS_CodeFunctionCode [0..1] = distance
+ preferences :NS_RoutePreferences [0..*]
+ returnRouteInstructions :boolean
+ returnRouteMaps :boolean
+ returnRouteGeometry :boolean
+routeRequest
+costFunction
CostFunction
0..*
«Type»
NS_CodeFunction
«CodeList»
NS_RouteRequestType
+ basic
+ predictive
+ dynamic
+ complex
«CodeList»
NS_CodeFunctionCode
+ distance
+ time
+ numberTurns
«CodeList»
NS_RoutePreferences
+ scenic
+ easiest
«DataType»
NS_RouteResponse
+ request :NS_RouteRequest [0..1]
+ route :NT_Route [0..*]
+rendering
0..*
«DataType»
NS_RenderingResponse

Figure 184: NS_RouteRequest and NS_RouteResponse.

ISO 19133 defines its own simple network model that builds on the topology model of the ISO 19107 Spatial Schema and that shall be used for spatially expressing the route (cf. figure 185). This network model is also seen as universe of discourse for the navigation problem and hence may be used to encode the navigation space data within the *Data* repository. The entire *NT_Network* is defined as subtype of *TP_Complex*. It consists of *NT_Junction* elements being subtypes of *TP_Node* as well as *NT_Link* elements being subtypes of *TP_DirectedEgde*. The boundary and coboundary relationships between *NT_Junction* and *NT_Link* are hence inherited from the Spatial Schema (cf. figure 164) and induce a directed graph. Likewise, the network structure may be embedded geometrically through the *Realization* association of the topological objects. Nevertheless, and similar to the MLSEM data model, the ISO 19133 network model also defines connectivity from a semantic point of view. Precisely, an *NT_Junction* representing a single node of the navigation graph is described by one or more *NT_Turn* instances which denote their own entry and exit links. The reason for this microtopology is that ISO 19133 mainly (but not exclusively) addresses road networks for car navigation. For example, if an *NT_Junction* represents a cross-road with several incoming links, then *NT_Turn* elements can be used to model possible traffic patterns. Like with *State*

[55] The full presentation of the modelling of cost functions based *NS_CostFunction* is outside the scope of this thesis. The reader is referred to the ISO 19133:2005 specification document instead.

and *Transition* features of the MLSEM, the nodes and edges in the network can carry cost values which are accessible through a *cost* operation that takes a cost function and a time position as input. A single route is modelled as subtype of *NT_Network* and is further presented in figure 186.

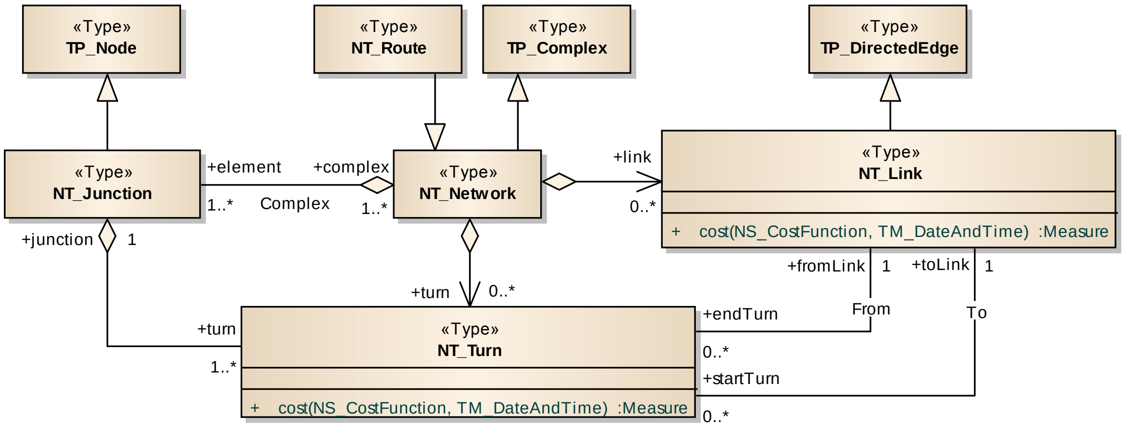


Figure 185: NT_Network type from ISO 19133:2005 for modelling the navigation space.

Similar to the MLSEM *Route* feature, an *NT_Route* is a simply connected, non-cyclic, directed subset of an *NT_Network* and hence must be geometrically realizable through a *GM_CompositeCurve* primitive. Further meta information about the route are provided as *NT_RouteSummary* which comprises the time and distance of the route as well as its start point, end point, and possible intermediate stops. Parts of the route can be classified as *NT_Maneuver* which represents a legal sequence of navigation actions expressed as ordered set of turns and corresponding links. The *NT_Maneuver* type is comparable to a *RouteSegment* of the MLSEM schema with the difference that a *Route* is entirely (and not just partly) made of a *RouteSegment* features. Every *NT_Maneuver* can be assigned free text using *NT_Advisory* elements in order to provide useful information for executing the maneuver. Cost values for the maneuver as well as its start and end points respectively links are accessible by invoking operations of *NT_Maneuver*.

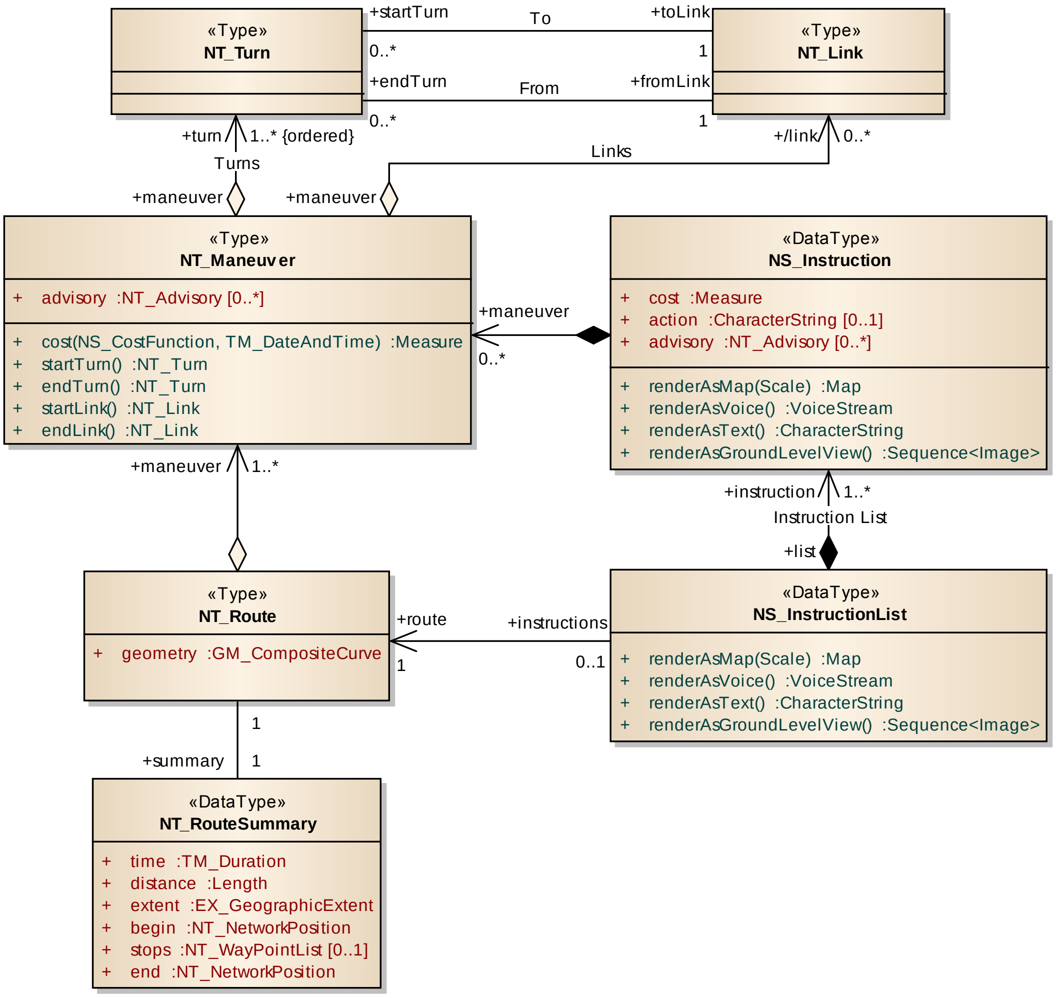

«Type»
NT_Turn
+startTurn
To
+toLink
0..*
1
«Type»
NT_Link
+endTurn
From
+fromLink
0..*
1
+turn
1..* {ordered}
Turns
+/link
0..*
Links
+maneuver
+maneuver
«Type»
NT_Maneuver
+ advisory :NT_Advisory [0..*]
+ cost(NS_CostFunction, TM_DateAndTime) :Measure
+ startTurn() :NT_Turn
+ endTurn() :NT_Turn
+ startLink() :NT_Link
+ endLink() :NT_Link
+maneuver
0..*
«DataType»
NS_Instruction
+ cost :Measure
+ action :CharacterString [0..1]
+ advisory :NT_Advisory [0..*]
+ renderAsMap(Scale) :Map
+ renderAsVoice() :VoiceStream
+ renderAsText() :CharacterString
+ renderAsGroundLevelView() :Sequence<Image>
+maneuver
1..*
+instruction
1..*
Instruction List
+list
«Type»
NT_Route
+ geometry :GM_CompositeCurve
+route
1
+instructions
0..1
«DataType»
NS_InstructionList
+ renderAsMap(Scale) :Map
+ renderAsVoice() :VoiceStream
+ renderAsText() :CharacterString
+ renderAsGroundLevelView() :Sequence<Image>
1
+summary
1
«DataType»
NT_RouteSummary
+ time :TM_Duration
+ distance :Length
+ extent :EX_GeographicExtent
+ begin :NT_NetworkPosition
+ stops :NT_WayPointList [0..1]
+ end :NT_NetworkPosition

Figure 186: NT_Route type for describing a route and answering an NS_RouteRequest.

Route guidance is provided through *NS_Instruction* which captures a single route instruction associated with an *NT_Maneuver*. The *NS_Instruction* data type defines the attributes *cost*, *action*, and *advisory*. Whereas *action* textually describes the action to be taken to navigate through the maneuver, *cost* represents its associated and pre-calculated cost value. Further *advisory* may be provided as *NT_Advisory* descriptions. An ordered sequence of *NS_Instruction* entities can be assembled to an instruction list represented as *NS_InstructionList* that provides guidance for the entire route. Every route instruction as well as instruction list may be rendered in different media types. ISO 19133 supports portrayals as 2-dimensional map (typically as raster image), voice stream, textual description, and as sequence of raster images providing ground level views. The rendering is available through corresponding operations which may be internally delegated to a rendering service. Similar to the MLSEM, the data types *Map*, *VoiceStream*, and *Image* are simply defined as binary arrays.

The ISO 19133 *Navigation Rendering Service* specifies a portrayal service for calculated routes which can be invoked from mobile clients through the external interface *NS_RenderingService* (cf. figure 187). It offers a *render* operation in order to translate routes into a visualization form suitable for the end-user device. The *NS_RenderingRequest* expects an instance of *NT_Route* as well as one or more types of target portrayals as specified by the *NS_RenderingType* code list, and is forwarded as input to the *render* operation. The resulting portrayal is then provided as *NS_RenderingResponse* which contains the corresponding binary data. Note that if a route is rendered as instruction list, then the list and its contained instructions may in turn provide their own rendering as described above.

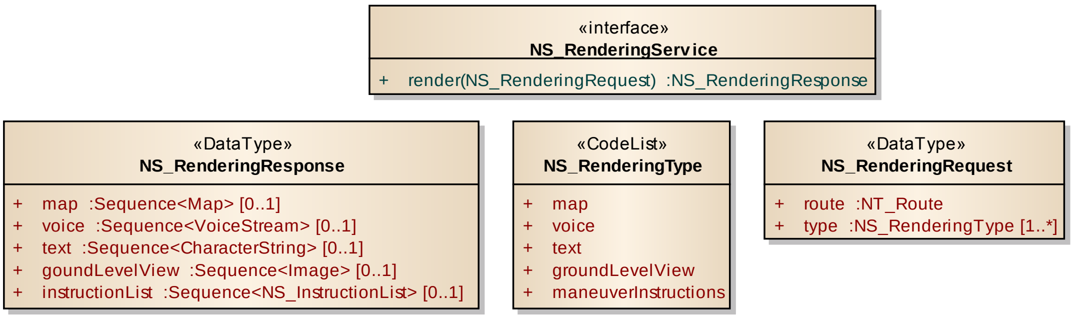

«interface»
NS_RenderingService
+ render(NS_RenderingRequest) :NS_RenderingResponse
«DataType»
NS_RenderingResponse
+ map :Sequence<Map> [0..1]
+ voice :Sequence<VoiceStream> [0..1]
+ text :Sequence<CharacterString> [0..1]
+ goundLevelView :Sequence<Image> [0..1]
+ instructionList :Sequence<NS_InstructionList> [0..1]
«CodeList»
NS_RenderingType
+ map
+ voice
+ text
+ groundLevelView
+ maneuverInstructions
«DataType»
NS_RenderingRequest
+ route :NT_Route
+ type :NS_RenderingType [1..*]

Figure 187: NS_RenderingService interface for requesting route portrayals.

The conceptual data model of the MLSEM *Route* package substantially agrees with the data types defined for describing routes as well as guidance information and portrayals within the ISO 19133 navigation services, but there are also subtle but important differences. First, and most importantly, ISO 19133 is primarily designed for outdoor navigation and especially for enabling car navigation applications. This follows from the semantics associated with the core components such as the *NT_Network* model. For example, an *NT_Link* may be classified, amongst others, as *road*, *ferry*, *rail*, *tunnel*, or *bridge*. Likewise, predefined types for *NT_Junction* include *intersection*, *roundabout*, *enclosedTrafficArea*, or *station*. Classifications for entities of the interior built environment are not provided, not to mention alternative space concepts. Second, the graph structure realized by *NT_Network* is neither expressive enough to capture the multilayered and hierarchical approach of the MLSEM nor the simultaneous description of indoor space entities in both primal and dual space. For these reasons, the *Route* feature type of the MLSEM schema is not modelled as subtype of *NT_Route* (and hence transitively as subtype of *TP_Complex*) but is defined on the spatio-semantic graph representation of the MLSEM. Third, route guidance within the MLSEM explicitly includes 3-dimensional visualization models which are not comprised by the predefined portrayal types of ISO 19133.

Besides those differences, the *Route* package of the MLSEM schema has been designed so that its contents can be mapped onto the conceptual data model of ISO 19133, which enables different scenarios for the integration of the MLSEM in the ISO 19133 service architecture. At first, an MLSEM indoor space model takes the role of a *Data* repository within this architecture (cf. figure 182), and thus provides the backing data structure underneath the services that allows answering service requests based on its rich spatio-semantic information model. This includes support for hierarchical and context-dependent path planning as well as localization and tracking based on the multilayered graph structure, but also comprises the derivation of route instructions and route presentations (e.g., using standardized OGC web portrayal services). The results can then be directly encoded as ISO 19133 compliant data content in order to be transported to the mobile clients via the service interfaces. Additionally, the results can also be expressed and stored as MLSEM *Route* feature in the data repository. By this means, the data repository not only serves the indoor space model itself but also provides routing information including route guidance and portrayals. This enables speeding up service responses as typical routes inside a facility can be pre-calculated and directly answered from the repository. Likewise, complex routes may be cached inside the repository and only be transported in smaller chunks to the mobile client in order to meet storage or computation restrictions of the client. The advantage of using the MLSEM *Route* feature instead of *NT_Route* for this purpose is that it is consistently embedded in the MLSEM schema and thus provides access to the entire indoor space model which, for example, is important in case of dynamic rerouting. Moreover, an MLSEM *Route* captures all the information required to construct an *NT_Route* and thus can be translated upon request into an ISO 19133 compliant representation to be delivered to the navigation application.

However, and as indicated above, the LBS framework of ISO 19133 requires extensions in order to be suitable for indoor navigation. Besides missing classifications for indoor entities, also the addressing schema for the identification and naming of locations as specified by ISO 19133 is not appropriate for indoor environments. It can only be used to represent postal street addresses with additional large-scale information about countries, cities, and administrative areas. A (symbolic) referencing schema for locations and places inside a facility is not yet supported. Moreover, 3-dimensional route presentations need to be included to support the generation of effective indoor route visualizations comprising multiple generalization levels (cf. Hagedorn et al. 2009).

4.5.2 OpenGIS Location Services (OpenLS)

Similar to ISO 19133, the OGC OpenLS specification also has the primary objective to define access to standardized services and content required for establishing an open location services platform on which web-based LBS applications targeting at mobile devices can be built. The top-level service architecture as specified by the current version 1.2 of OpenLS (Mabrouk 2008) is depicted in the following figure.

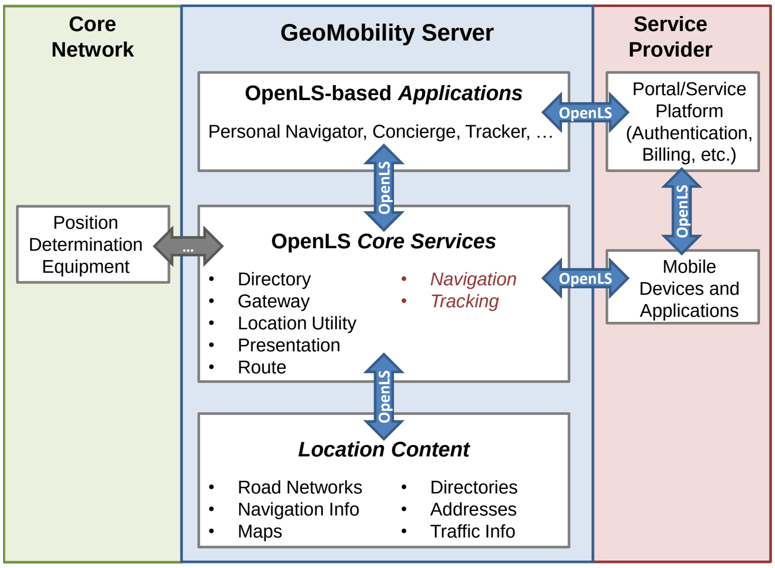


Figure 188: Conceptual architecture for location-based services as specified by OpenLS (Mabrouk 2008).

The core component of this architecture is called *GeoMobility Server* which is structured as a set of five *Core Services*. OpenLS defines both the interfaces to communicate with these services as well as an information model (*Abstract Data Types*, ADT) to represent the data payload of service requests and responses. Further content such as local LBS-aware applications accessing the services or data such as maps, points of interests, or routes being used and delivered by the services may be additionally contained in the *GeoMobility Server* or be accessed over the internet. The OpenLS specification makes no assumptions regarding the underlying network platform and protocols.

The OpenLS core services are called *Directory Service*, *Gateway Service*, *Location Utility Service*, *Presentation Service*, and *Route Service*. Whereas the directory service allows for finding nearby places, products, or services, the gateway service is meant to fetch the current location of a mobile device from the underlying mobile network structure. The location utility service facilitates to translate this location into the OpenLS addressing schema for locations and vice versa based on geocoding and reverse geocoding mechanisms. Both the route and the presentation service are illustrated in more detail in the following. Two extensions of the OpenLS service stack are described under separate cover in accompanying official OGC standards. First, a *Navigation Service* as extension of the route service is introduced in (Fuchs et al. 2008) which however is strongly tailored to car navigation applications and thus not further discussed in this thesis. Second, (Smyth 2008) defines a *Tracking Service* interface for recording location changes of a mobile client.

The OpenLS services are specified in XML Schema in order to realize a physical XML messaging format intended for network data transfer, e.g. based on protocols such HTTP or by embedding OpenLS messages into standards from the field of service-oriented architectures (SOA) such as SOAP. In figure 189, the main interface of the route service for requesting a route for a given navigation user is illustrated as UML diagram.[56]

[56] The OpenLS XML Schema components have been translated into UML model elements for this thesis. The resulting UML diagrams are informal as they are not included in the normative part of the OpenLS specification.

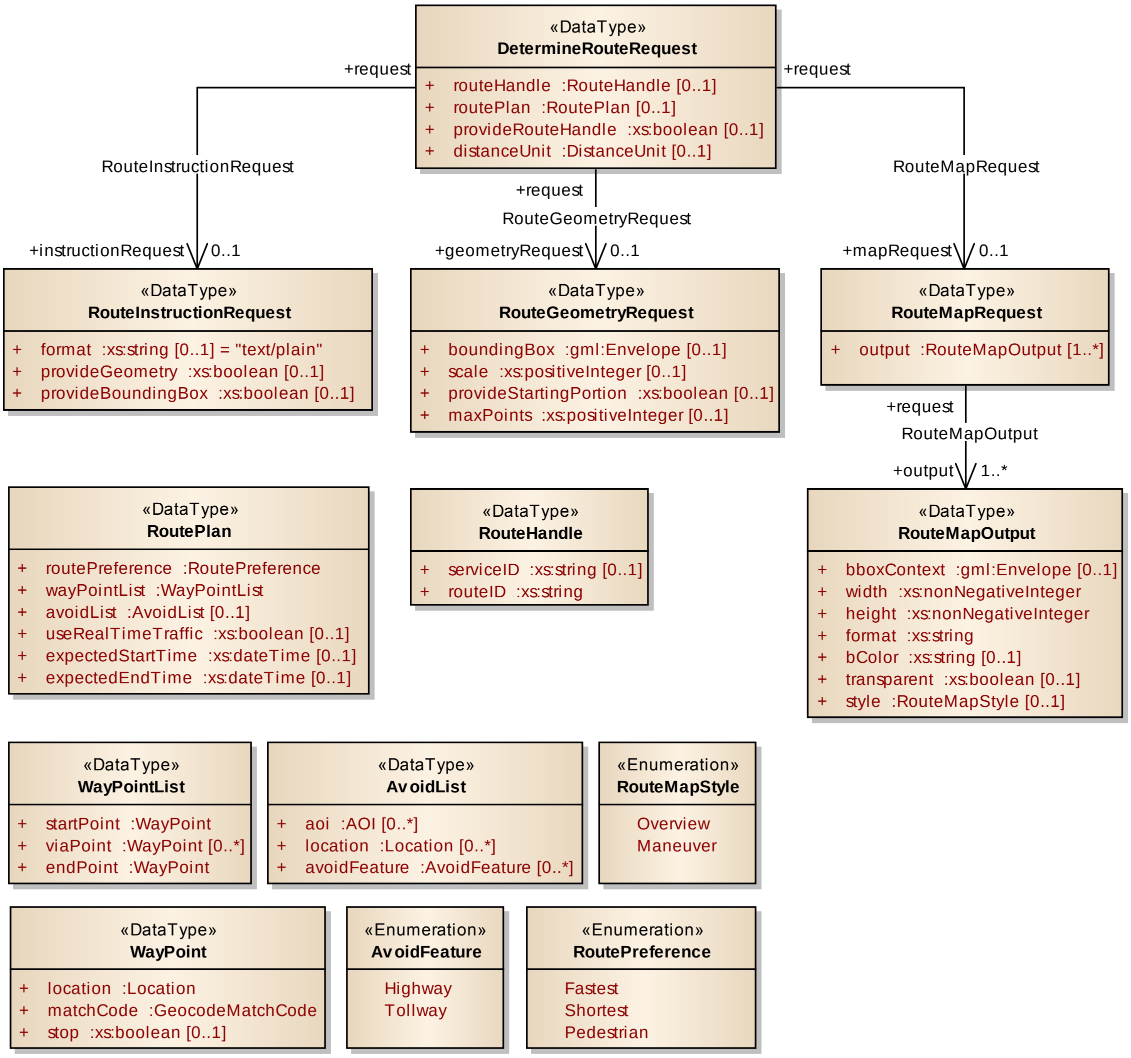

«DataType»
DetermineRouteRequest
+ routeHandle :RouteHandle [0..1]
+ routePlan :RoutePlan [0..1]
+ provideRouteHandle :xs:boolean [0..1]
+ distanceUnit :DistanceUnit [0..1]
+request
+request
+request
RouteInstructionRequest
RouteGeometryRequest
RouteMapRequest
+instructionRequest
0..1
+geometryRequest
0..1
+mapRequest
0..1
«DataType»
RouteInstructionRequest
+ format :xs:string [0..1] = "text/plain"
+ provideGeometry :xs:boolean [0..1]
+ provideBoundingBox :xs:boolean [0..1]
«DataType»
RouteGeometryRequest
+ boundingBox :gml:Envelope [0..1]
+ scale :xs:positiveInteger [0..1]
+ provideStartingPortion :xs:boolean [0..1]
+ maxPoints :xs:positiveInteger [0..1]
«DataType»
RouteMapRequest
+ output :RouteMapOutput [1..*]
+request
RouteMapOutput
+output
1..*
«DataType»
RoutePlan
+ routePreference :RoutePreference
+ wayPointList :WayPointList
+ avoidList :AvoidList [0..1]
+ useRealTimeTraffic :xs:boolean [0..1]
+ expectedStartTime :xs:dateTime [0..1]
+ expectedEndTime :xs:dateTime [0..1]
«DataType»
RouteHandle
+ serviceID :xs:string [0..1]
+ routeID :xs:string
«DataType»
RouteMapOutput
+ bboxContext :gml:Envelope [0..1]
+ width :xs:nonNegativeInteger
+ height :xs:nonNegativeInteger
+ format :xs:string
+ bColor :xs:string [0..1]
+ transparent :xs:boolean [0..1]
+ style :RouteMapStyle [0..1]
«DataType»
WayPointList
+ startPoint :WayPoint
+ viaPoint :WayPoint [0..*]
+ endPoint :WayPoint
«DataType»
AvoidList
+ aoi :AOI [0..*]
+ location :Location [0..*]
+ avoidFeature :AvoidFeature [0..*]
«Enumeration»
RouteMapStyle
Overview
Maneuver
«DataType»
WayPoint
+ location :Location
+ matchCode :GeocodeMatchCode
+ stop :xs:boolean [0..1]
«Enumeration»
AvoidFeature
Highway
Tollway
«Enumeration»
RoutePreference
Fastest
Shortest
Pedestrian

Figure 189: DetermineRouteRequest type for querying routes. The namespace "gml" is associated with elements from OGC GML whereas "xs" is used for elements specified by the XML Schema Definition Language.

The *DetermineRouteRequest* lets a navigation application trigger the determination of an appropriate route for a given *RoutePlan*. The route plan comprises at least the start and end location of the route as well as optionally further waypoints along the route (provided as *WayPointList*) or locations that shall be avoided (denoted as *AvoidList*). A *Location* in the sense of OpenLS can be given as address, as points or areas of interest (POI, AOI) that are possibly acquired through the directory service, or simply as position based on a coordinate tuple (in both two and three dimensions) equipped with a spatial reference system. Further attributes of a *RoutePlan* include the route preference (fastest, shortest, or pedestrian routes) as well as optionally the expected start and end time of the travel. OpenLS also supports storing a route for a mobile subscriber and hence offers the means to reference a stored route in a route request through a *RouteHandle*. A stored route is identified through a *serviceID* that points to the service which holds the route (typically by means of a URI) and a *routeID* that denotes the route store at the service.

The route has to be requested in one or more of the following three forms. First, as preformatted list of route instructions (*RouteInstructionRequest*) which are per default returned as textual descriptions and may possibly include the route geometry. Second, as a purely geometric representation (*RouteGeometryRequest*) which may be restricted to a given geographic bounding box (encoded as GML *Envelope*) or to a maximum number of contained points. Or third, as map view (*RouteMapRequest*) with different outputs as further described by the *RouteMapOutput* type. The map can be restricted to a given bounding box context. It is assumed to be returned as raster image of a given *width* and *height*, and encoded in the media type (MIME type) requested by the *format* attribute. The map may show the entire route in an *Overview* or just a single route *Maneuver* (*RouteMapStyle*).

The response of the route service is encoded as *DetermineRouteResponse* whose UML diagram is presented below. A route response comprises a summary of the route as well as the requested route forms for display on the mobile device.

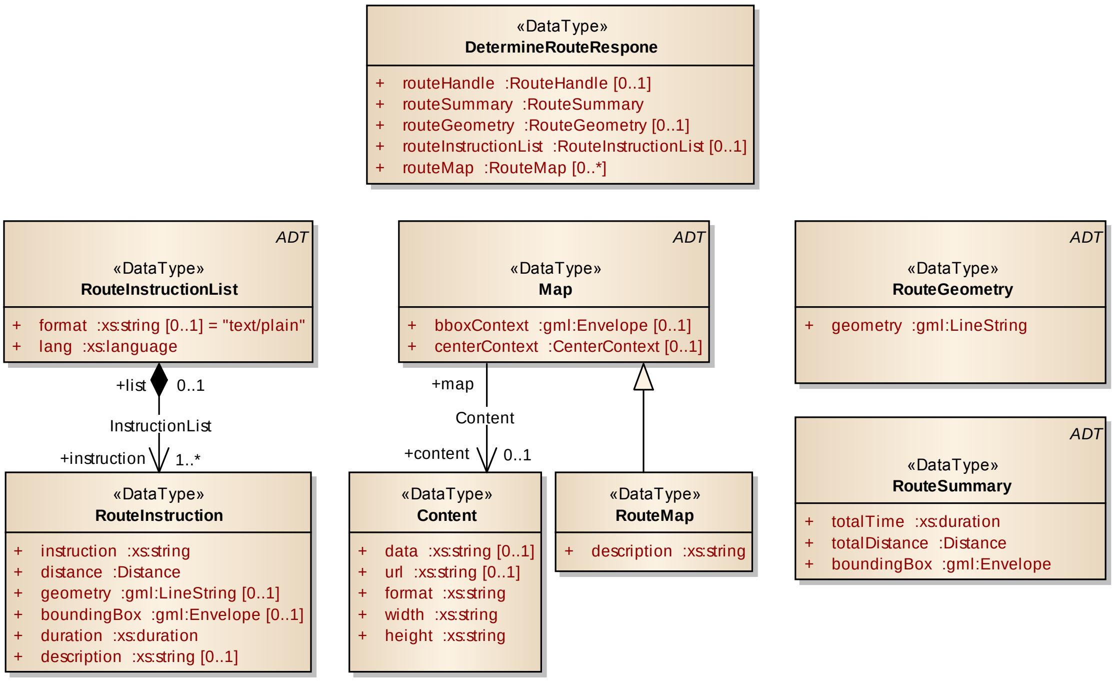


Figure 190: DetermineRouteResponse type for answering a route request.

As described above, the content of a *DetermineRouteResponse* message is specified in OpenLS in terms of *Abstract Data Types*. The *RouteSummary* ADT provides meta information about the route which is limited to the description of the total time and distance as well as the geographic region. The route summary is thus less expressive than both the *NT_RouteSummary* entity of ISO 19133 and the MLSEM *Route* feature. The *RouteGeometry* ADT is a geometric description of the route using a *LineString* primitive from the GML schema. A GML *LineString* is an XML encoding of a *GM_LineString* from ISO 19107 and thus represents a 2-dimensional or 3-dimensional sequence of straight line segments. Unlike ISO 19133 or the MLSEM schema, the route topology is not expressed explicitly but has to be derived from the geometry. Moreover, the geometric description using a *GM_CompositeCurve* as applied in both ISO 19133 and the MLSEM schema is more generic as it also comprises freeform curves. The *RouteInstructionList* ADT represents an ordered list of textual turn-by-turn instructions and travel advisories associated with single route segments or maneuvers, which is preformatted for presentation on the mobile device in the media type specified by the *format* attribute. Each *RouteInstruction* consists of the actual *instruction* to be taken as well as an optional *description*. Moreover, the *distance* and *duration* for executing the instruction can be stated as cost values, and both the route geometry and bounding box associated with the instruction can be provided. Each *RouteInstructionList* may denote the language in which it is given through the *lang* attribute. Finally, the route response may contain a set of *RouteMap* views. The *RouteMap* data type is a subtype of the *Map* ADT. The *Content* object associated with a *Map* describes the map data which may either be embedded in the XML message (as binary encoded string through the *data* attribute) or given as URL pointer to a remote service where the data can be found (e.g., as preformatted request to an OGC Web Map Service). The *description* attribute of *RouteMap* allows for connecting a route instruction with the map provided that the *description* of the instruction matches.

As for presentation tasks, the OpenLS framework only offers map portrayals via its presentation service. The response to a corresponding *PotrayMapRequest* is hence a set of *Map* ADT objects. The request itself contains one or more *Output* descriptions for the maps whose parameters are similarly defined as in *RouteMapOutput*. Again, the maps may be requested to be encoded inline the response or to be given as pointers to another map service. Moreover, different map layers can be specified that shall be included in or excluded from the returned base map.[57] In addition, map overlays can be defined which includes the rendering of points of interests, route geometries (as returned from a route service), coordinate positions, or further maps on top of the base map. For each map layer and overlay, a corresponding presentation style can be announced which provides user-defined symbolization and colouring of geographic feature. This style may be, for example, encoded using the OGC *Symbology Encoding* (SE) language or the OGC *Styled Layer Descriptor* (SLD) profile of the WMS. The following figure contains the UML representation of the OpenLS presentation service.

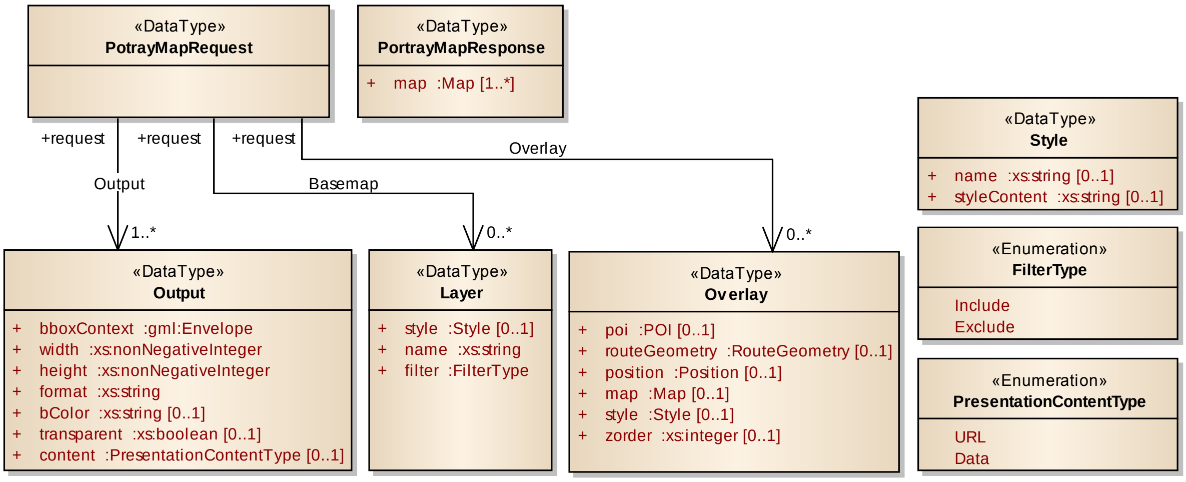


Figure 191: OpenLS Presentation Service based on the PotrayMapRequest type.

Like with the ISO 19133 service architecture, the role of the MLSEM within the OpenLS framework is primarily that of providing the rich base data in the implementation underneath the core services. Hence, an instance of the MLSEM serves the required input for answering requests from the discovery, location utility, route, presentation, or tracking service, which may be realized inside or outside the *GeoMobility Server* using standardized interfaces for accessing the MLSEM data such as the OGC WFS. In contrast to ISO 19133, the OpenLS platform much more implies such a backing data structure since the ADTs are rather to be understood as high-level abstractions of the route object. For example, OpenLS neither defines nor assumes a geometric-topological network structure such as *NT_Network* for the description of routes or even of the entire navigation data.[58] The geometric, topological, and semantic view on the navigation problem has instead to be provided by the base data. This not only increases the flexibility of the OpenLS platform but also the requirements on the underlying information model which are however fully met by the MLSEM. OpenLS also implements the idea of a distributed systems architecture more consequently since, for example, both routes and maps can be delivered to the mobile client in form of handles to remote services. This strongly supports the idea of using the MLSEM *Route* feature for the pre-calculation and caching of routes, and to translate the stored routes on demand into corresponding ADTs. It follows from the presentation of the selected OpenLS ADTs (cf. figure 190) that they can be easily populated from the information kept with a *Route* feature, which was also an important design goal of the MLSEM *Route* package. A client being capable of consuming MLSEM data may even be fed with handles to the *Route* feature itself (or likewise to its associated *RouteInstruction* and *RoutePresentation* information) via the OpenLS interfaces.

However, there are also restrictions of OpenLS especially in the context of indoor navigation. First, map portrayals are assumed to be 2-dimensional representations delivered as raster images. The display of the route in its 3-dimensional context using arbitrary visualization models (or even further media types such as videos) is not foreseen by the route and presentation services, although proposals for corresponding portrayal services and styling descriptors exist in the OGC Web Services suite (e.g., W3DS, WVS, and 3D-SLD). Moreover, additional presentations of the route such as audio descriptions cannot be delivered which however may be required for navigating visually impaired users. Second, and similar to ISO 19133, OpenLS uses an addressing schema for locations that

[57] As with any OGC Web Service, a presentation service has to provide a capabilities document which, amongst others, lists the available layers.

[58] In fact, version 0.5.0 of the OpenLS *Navigation Service* (cf. Bychowski et al. 2003) still contained the specification a network structure for expressing manoeuvres which showed considerable overlap with the *NT_Network* of ISO 19133. However, with the release of version 1.0.0 of the *Navigation Service*, this specification was omitted without substitution in favour of a more flexible OpenLS platform.

only supports postal street addresses but that is not generic enough to also name locations and places in indoor environments based on symbolic identifiers. Thus, the identification of indoor locations is only possible through 3-dimensional coordinate positions based on the current OpenLS specification.

Since its first release in 2003, OpenLS has received much attention not only from software vendors but also in scientific research. Several works have proposed extensions to the core services in order to bring OpenLS to the third dimension (e.g., Verbree & Zlatanova 2007, Papataxiarhis et al. 2007, Neis & Zipf 2008, Ogawa et al. 2011). Most of the work done focuses on the determination and presentation of routes in the context of 3-dimensional city models in order to enable outdoor navigation applications. In (Schilling & Goetz 2010), strategies and extensions for seamless indoor-outdoor applications are discussed and prototypically implemented on top of OpenLS as part of the OGC OWS Phase 6 testbed.[59] According to (Goetz & Zipf 2010), both a proper 3-dimensional visualization as well as a rich spatio-semantic indoor space model providing the underlying base data for routing and presentation still remain the major open issues of LBSs in the context of indoor navigation. The MLSEM provides the information model for describing this rich base data which can be integrated with ISO 19133 respectively OpenLS as shown in this chapter in order to overcome these issues.

[59] OGC Web Services (OWS) testbeds are part of the OGC interoperability initiative and are global, collaborative prototyping programs conducted annually with partners from industry, public administration, and academia. One major goal is to test and validate candidate specifications in order to deliver them into the OGC specification process.

Chapter 5
Consideration of Navigation Constraints

This chapter proposes a formal conceptual model for the representation of *navigation constraints* that captures the knowledge about environmental and user-dependent movement restrictions in an indoor environment and facilitates reasoning and inference about this knowledge when navigating a user through the environment. Everyday examples of navigation constraints are movement limitations enforced on road networks such as prohibited or restricted turns at junctions, directional restrictions such as one-way roads, or limitations addressing speed, time, or allowed vehicle types. Navigation constraints also play an essential role in indoor navigation and many of the constraints applied to road networks have similar counterparts in the indoor domain (Worboys 2011). Examples for movement restrictions in indoor space include corridors or gates that may only be passed in one direction, restricted movement patterns, and places that are inaccessible due to physical limitations of the user, lack of permission, or time restrictions.

In the course of the previous chapters, a Boolean attribute $isNavigable$ has been introduced as example for modelling a movement restriction on space cells and boundary cells (cf. example 3.63), which however is obviously limited in expressivity. The scope of this chapter is therefore to develop a conceptual model and taxonomy for navigation constraints that goes beyond simple and static Boolean flags and that is expressive enough to formulate a multitude of possible movement restrictions in indoor environments. Moreover, this model has to be accompanied by a rule base governing the evaluation of navigation constraints in order to algorithmically (and dynamically) answer whether a specific portion of the indoor environment is traversable for a given navigation user. In a first step, a classification scheme for navigation constraints is proposed and related work is reviewed against their support for the identified categories. Based on this, a navigation constraint model is developed which addresses the different types of movement restrictions as well as their temporal, physical, and logical preconditions. Since movement restrictions are strongly related to the elements of the indoor space model, the constraint model is defined as integral part of the MLSEM application schema as presented in chapter 4. By this means, navigation constraints can directly be stored and exchanged with the indoor space data. The feasibility of the developed model is then demonstrated along typical use cases for navigation constraints in indoor environments. The topic of navigation constraints has not been included in former publications of the MLSEM (cf. Becker et al. 2009a, Becker et al. 2009b, Nagel et al. 2010) and thus renders a novel extension to this previous work.

5.1 Classification of Navigation Constraints

In the context of this thesis, navigation constraints are understood to account for the *traversability* of the space entities in a navigation space by a navigation user. Traversability can hereby be viewed as a function of physical, logical, or temporal conditions on a space entity that characterizes the entity to be navigable by the user in case it evaluates to true, and non-navigable otherwise. Navigation constraints hence determine under which conditions movement is physically possible or admitted in the environment and enable corresponding reasoning processes (Stoffel et al. 2007).[60]

According to (Hendricks et al. 2003), the different types of movement constraints in a spatio-temporal navigation space can be classified into four categories that are described using the modal verbs *can*, *may*, *must*, and *should*. A *can* constraint refers to the physical ability of the navigation user to navigate a space entity. For example, pedestrians typically *can* take the stairs whereas wheelchair users, mobility impaired persons, or driving robots *cannot*. Likewise, wheelchair users and mobile robots *cannot* travel through doors being too narrow or along ramps whose slope is too steep. The category of *may* constraints, on the other hand, is associated with a permission following from rules or laws. Typical examples include access rights that require authorization of the user (e.g., through keys, biometric scans, screening), a certain user state (e.g., holding a boarding pass), or the membership

[60] Navigation constraints in this sense are not to be confused with cost values provided for the elements of a navigation graph. Whereas cost values determine the (physical, semantic, or temporal) cost for traversing a space entity and hence are typically used in finding the shortest, fastest, or best path, navigation constraints denote whether the space entity is traversable at all.

to a group (e.g., access allowed for staff members only). Access restrictions may also be related to time (e.g., opening hours) or to the state of an object (e.g., high-security laboratory). *May* constraints obviously can overrule the physical ability to move. For instance, although a navigation user *can* walk through a door, this movement *may not* be allowed. On the contrary, if the navigation user physically *cannot* pass through the door, then a positive permission will not change this compelling physical restriction. Movement obligations (e.g., predefined routes for visitors) are classified as *must* constraints, whereas *should* denotes a weaker form of must that implies a choice of the navigation user (e.g., the preference to take the stairs instead of elevators).

When considering the number of spatial entities involved in the movement restriction, a complementary classification to that of (Hendricks et al. 2003) can be made into *simple* and *complex* navigation constraints. Whereas a *simple constraint* is associated with a single space entity such as a door or a room, a *complex constraint* is applied to a set of entities that may follow from semantic or spatial criteria. For example, a complex constraint can be used to denote that the set of all elevators within a facility may not be used in case of an emergency situation or that all rooms belonging to a given organizational unit may only be entered by staff members of that unit. Further important representatives of complex constraints are prohibited maneuvers that are expressed on an ordered sequence of spatially connected (but possibly semantically different) entities. Prohibited maneuvers are typically *may* constraints that disallow physically possible movements such as specific turns or movement directions when travelling along the architectural entities within a building. If a prohibited maneuver is enforced physically (e.g., two places are connected through a one-way door) then it renders a complex *can* constraint.

A navigation constraint is said to be a *combined constraint* if it demands multiple and different physical, logical, or temporal conditions to be fulfilled either partially or completely in order for the constraint to apply. For example, assume that access to a specific room such as an office or a shop depends on both a temporal condition (e.g., office or opening hours) and an additional user authorization (e.g., holding a key or having a certain age). A combined constraint is also required to express that windows on the ground floor of a facility may serve as exits given an emergency situation (logical condition) and provided that the navigation user physically can step on the window ledge (physical condition). Obviously, a combined constraint can be both simple and complex. Moreover, it may be the result of expressing more than one navigation constraint on a single space entity. For instance, a door can be associated with two separate navigation constraints providing access rules for weekdays and the weekend. Depending on the time of navigation, only one or the other has to be satisfied. If the door further restricts the movement in one direction, it may also participate in a complex prohibited maneuver that has to be obeyed in addition to the access rule. The examples demonstrate that the combination of navigation constraints causes a high degree of combinatorial complexity. A conceptual model for navigation constraints must hence be expressive enough to support arbitrary combinations of constraints and has to establish the rules for their evaluation (e.g., in case of competing combinations).

Navigation constraints mostly affect the task of path finding. Candidate routes for a given navigation user are only those routes that satisfy all constraints applied to the space entities making up the route. A route is hence the result of considering and evaluating navigation constraints but itself represents an unconstrained path that can be travelled by the user. But there are also navigation constraints which are non-effective in path searches in the sense that they do not exclude space entities from the route and which thus cannot be enforced by the navigation system. Examples from the domain of car navigation are speed limits. Although a specific road is assigned a speed limit (and thus a legal permission constraint), this road is not a priori excluded from path searches even if the navigation user violates the constraint. However, a speed limit constraint may become effective in path searches, for example, if it expresses a minimum speed (e.g., on motorways) which has to be met by the vehicle of the navigation user or if the user prefers to avoid roads with specific speed limits. An example navigation constraint in indoor environments may disallow certain movement types such as running and jumping within laboratories being equipped with sensitive measuring devices. Again, this constraint will typically not prevent the navigation system from routing a user through the laboratory, and it is left at the discretion of the user to obey the movement type restriction. This may be different in cases where a navigation user physically cannot observe the constraint such as a mobile robot causing too much vibration when driving through the laboratory. A navigation constraint along a route that has not been considered in the underlying path finding process should at least be communicated to the user as route advisory. For example, modern car navigation systems typically display a speed limit constraint on top of the digital road map.

It follows from the discussed types of navigation constraints and the illustrated examples that in most cases navigation constraints need to be evaluated against the context of a specific navigation user since obviously both capabilities and permissions differ for different users. It thus commonly involves environmental as well as user-related contextual information in order to answer whether or not a movement restriction applies to a given user. From this a further distinction into *global* and *user-dependent* navigation constraints can be deduced. Constraints having *global* scope are independent of the user context and thus equally limit the movement of all navigation users. Examples in this category include architectural constraints (e.g., walls or fixed obstacles) as well as temporal constraints that are not combined with further user-related conditions. In contrast, the evaluation of *user-dependent* navigation constraints requires access to the user profile which thus needs to capture information about the physical and perceptual capabilities of the user, permissions and access rights, memberships to user groups, etc. (cf. chapter 1.2). For example, assume a navigation constraint announces that a corridor can only be passed by users weighing less than a maximum value and being capable of navigating a specific floor material (e.g., a glass surface or a floor grating). Whether this corridor is traversable for a given navigation user requires knowledge about the user's actual weight and motor capabilities (possibly including capabilities of assistive devices such as wheelchairs or crutches). Moreover, rules or methods need to be defined for handling incomplete knowledge about the user in reasoning processes (e.g., falling back on default assumptions or using predefined values from user group profiles).

5.2 Navigation Constraints in Related Work

Limited work has been undertaken in literature to formalize a conceptual model that, on the one hand, supports the various types of constraints as discussed in the previous section and, on the other hand, facilitates their rule-based evaluation in reasoning processes. The explicit representation of navigation constraints as essential part of the indoor space model is acknowledged and addressed in the works of (Meijers et al. 2005), (Tsetsos et al. 2006), (Stoffel et al. 2007), (Slingsby & Raper 2008), (Yuan & Schneider 2010a), (Goetz & Zipf 2011), and (Worboys 2011) (cf. chapter 2.2). In (Meijers et al. 2005), navigation constraints are mapped onto surface classifications. For example, *non-granting* polygons are used to represent a physical *cannot* constraint, whereas *granting* surfaces denote access restrictions such as doors requiring authorization or emergency exists. Also directional movement constraints are represented by separate surface types. However, additional logical or temporal movement restrictions as well as combined or complex constraints cannot be expressed, and rules for the evaluation of constraints are not presented. (Slingsby & Raper 2008) discuss extensions to the model of (Meijers et al. 2005) in order to model temporal constraints and the user context but do not present a conceptual or formalized model. Both (Yuan & Schneider 2010a) and (Goetz & Zipf 2011) propose to control the accessibility of space entities by means of semantic attributes associated with the elements of the navigation graph. However, also their approaches lack a formal model for expressing knowledge about navigation constraints and for reasoning about this knowledge.

(Stoffel et al. 2007) propose the encoding of *can* and *may* constraints as Boolean expressions of the form $\bigwedge_{*}(attr = value \vee attr \in valuePartition)$ on boundary nodes as well as regions graphs of their indoor space model. Each constraint refers to a semantic attribute $attr$ associated with a graph element and enforces its value to either match a predefined $value$ or to be within an allowed set of values denoted by $valuePartition$. The constraints are said to be *hard constraints* since they act as a filter that only admits those graph elements to be considered in path searches for which the Boolean expression evaluates to true. Combined constraints are supported in that more than one semantic attribute of an individual graph element may be restricted. A combined constraint is evaluated as Boolean intersection of all attribute restrictions so that the violation of a single restriction already suffices for the constraint to apply. (Stoffel et al. 2007) remain on an abstract level and do not present a conceptual model for physical, logical, or temporal constraints. Moreover, neither complex constraints nor the evaluation of constraints against the user context are discussed.

Within the OntoNav framework (cf. chapter 2.2.4), (Tsetsos et al. 2006) suggest the use of first-order logic for explicitly expressing and reasoning about user-dependent navigation constraints. Precisely, constraints are given as conditional statements that denote both a hypothesis and a conclusion in terms of predicates over arguments. For example, the formula $UNO{:}HandicappedUser(u) \wedge INO{:}Stairway(s) \rightarrow INO{:}isExcludedFor(s,u)$ asserts that if the object u is a handicapped user and s is a stair then s may not be a path element for u. The concepts $HandicappedUser$ and $Stairway$ are given semantic meaning in the underlying user respectively indoor navigation ontology (UNO, INO) of OntoNav. Likewise, the conclusion $isExcludedFor$ is a predefined binary relation between the ontological concepts and marks a path element to be excluded from routes for a given navigation user.

(Tsetsos et al. 2006) emphasize the role of the user context in navigation constraints and propose a user model for capturing user-related information (cf. figure 25). However, the authors merely focus on the physical traversability by the user while neglecting movement limitations following from logical or temporal facts such as access restrictions. Furthermore, rules for evaluating combined constraints are not presented, and complex constraints are unaddressed.

Navigation constraints in ISO 19133. A conceptual data model for navigation constraints is specified within the location-based services framework of ISO 19133:2005 (cf. chapter 4.5.1). The UML notation for this constraint model is shown in figure 192.

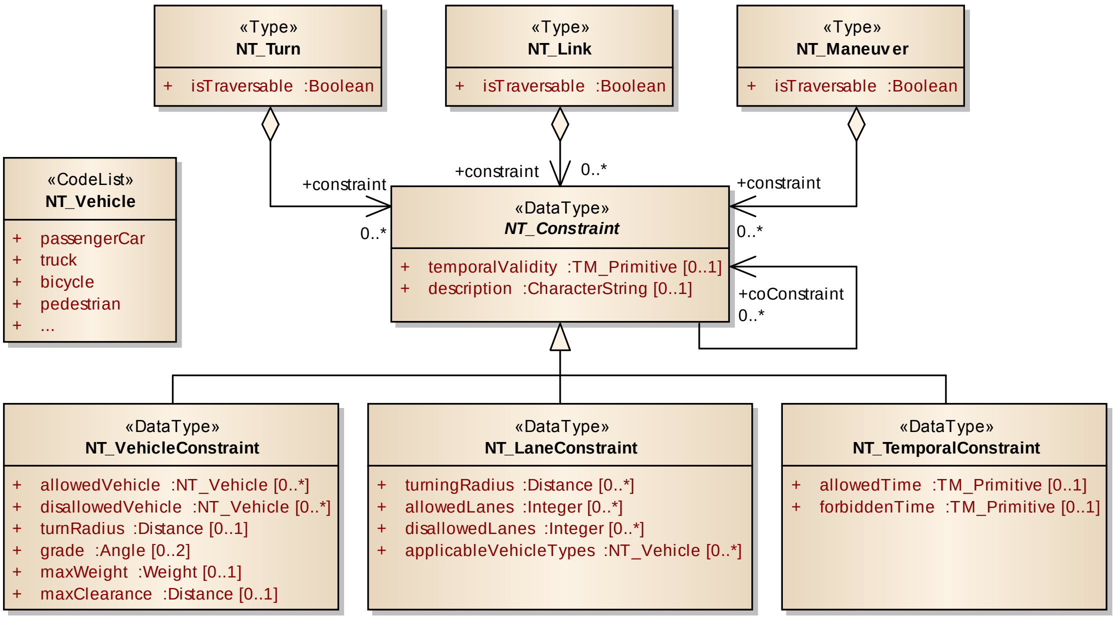

«Type»
NT_Turn
+ isTraversable :Boolean
«Type»
NT_Link
+ isTraversable :Boolean
«Type»
NT_Maneuver
+ isTraversable :Boolean
«CodeList»
NT_Vehicle
+ passengerCar
+ truck
+ bicycle
+ pedestrian
+ ...
+constraint
0..*
+constraint
0..*
+constraint
0..*
«DataType»
NT_Constraint
+ temporalValidity :TM_Primitive [0..1]
+ description :CharacterString [0..1]
+coConstraint
0..*
«DataType»
NT_VehicleConstraint
+ allowedVehicle :NT_Vehicle [0..*]
+ disallowedVehicle :NT_Vehicle [0..*]
+ turnRadius :Distance [0..1]
+ grade :Angle [0..2]
+ maxWeight :Weight [0..1]
+ maxClearance :Distance [0..1]
«DataType»
NT_LaneConstraint
+ turningRadius :Distance [0..*]
+ allowedLanes :Integer [0..*]
+ disallowedLanes :Integer [0..*]
+ applicableVehicleTypes :NT_Vehicle [0..*]
«DataType»
NT_TemporalConstraint
+ allowedTime :TM_Primitive [0..1]
+ forbiddenTime :TM_Primitive [0..1]

Figure 192: The conceptual navigation constraint model of ISO 19133:2005.

In ISO 19133:2005, navigation constraints are given as subtypes of the abstract data type *NT_Constraint* which is used to denote turn and link restrictions on the elements of a navigation graph. *NT_Constraint* is specialized into the predefined types *NT_VehicleConstraint*, *NT_LaneConstraint*, and *NT_TemporalConstraint*. The class names clearly indicate that the constraint model of ISO 19133 primarily targets the domain of car navigation. Each navigation constraint carries a *temporalValidity* attribute to specify the time at which the constraint is valid and thus has to be obeyed by a navigation user. Textual guidance on the constraint may be provided to the user using the *description* attribute.

An *NT_VehicleConstraint* restricts the types of vehicles that are allowed to navigate a turn or link. It thus renders a *may* constraint. Since both the allowed and disallowed vehicle types (including pedestrians) can be explicitly listed through corresponding attributes, the constraint facilitates to express both permission and denial. If the value of *allowedVehicle* is left empty, the default assumption is that all vehicle types except those provided by *disallowedVehicle* are allowed. The converse holds if *disallowedVehicle* is blank. In case both attributes are not provided, all vehicle types are allowed. The representation and evaluation of both *may* and *may not* conditions is a general pattern that is also applied to *NT_LaneConstraint* and *NT_TemporalConstraint*. The vehicle constraint is further characterized by the attributes *turnRadius*, *grade*, *maxWeight*, and *maxClearance*. Whereas *turnRadius* specifies the minimum radius of curvature of the restricted network entity, *grade* denotes its maximum slope. A vehicle *cannot* traverse the entity if is incapable of negotiating the turn radius or slope. Moreover, *maxWeight* and *maxClearance* define the maximum weight and height of an allowed vehicle. It easily follows that answering these conditions presupposes a model of the vehicle that holds the corresponding information. However, ISO 19133 specifies neither this model nor methods for its evaluation. *NT_LaneConstraint* enumerates the allowed and disallowed lanes from which a turn can be made legally. An optional turn radius can be provided which follows the same semantics as for *NT_VehicleConstraint*. Per default, the lane constraint applies to all vehicle types, which may be restricted to specific vehicles through the attribute *applicableVehicleTypes*. Finally, an *NT_TemporalConstraint* is used to denote the time when a network entity is open or closed to traffic.

An instance of *NT_Constraint* is only considered if all its conditions are met. Moreover, every constraint can be combined with various other constraints through the *coConstraint* association in order to further specify its scope.[61] For example, a temporal constraint may be limited to specific vehicle types listed in a vehicle constraint. ISO 19133 defines a Boolean conjunction for the evaluation of co-constraints, i.e. all conditions need to be satisfied for the combined constraint to hold. Navigation constraints are expressed on *NT_Turn*, *NT_Link*, and *NT_Maneuver* entities (cf. figure 185 and figure 186). Each entity may receive zero or more constraints. In contrast to co-constraints, the interpretation of multiple constraints assigned to a single entity is a Boolean disjunction of their separate conditions. Since *NT_Maneuver* is a directed sequence of turns and links and thus involves more than one entity, ISO 19133 also supports the modelling of complex constraints such as prohibited maneuvers.

The ISO 19133 constraint model is only used to mark those network entities which are non-traversable under some conditions but not in others. Whether an entity physically cannot be traversed in any circumstances is instead modelled through the Boolean attribute *isTraversable* on *NT_Turn*, *NT_Link*, and *NT_Maneuver*. Similar to both (Stoffel et al. 2007) and (Tsetsos et al. 2006), the constraints are meant to be evaluated in path searches and thus affect the route of a navigation user. Navigation constraints not influencing the route such as speed limits are hence not covered by separate constraint types. In contrast, speed limits are only regarded as possible input to cost functions in order to derive the expected travel time for a route, and may additionally be reflected in route guidance information.

The expressivity of the ISO 19133 constraint model partially exceeds the proposals of (Stoffel et al. 2007) and (Tsetsos et al. 2006) in that both combined and complex constraints are supported. The predefined constraint types as well as their physical and temporal conditions are however tailored to vehicle navigation on road networks. Although the model could be used in similar contexts in indoor environments (e.g., automative mobile robots used in production lines or for the transportation of goods), its support for expressing movement limitations for additional modes of locomotion or logical constraint conditions such as access permissions is limited. Unlike ISO 19133, the OGC OpenLS framework for location-based services does not cover navigation constraints. This can be explained from the fact that OpenLS is focused on the definition of service interfaces and therefore does not include a separate model of the navigation graph and its entities (cf. chapter 4.5.2).

Navigation constraints in the *Geographic Data Files* (GDF) standard. The international ISO standard *Geographic Data Files* (GDF, ISO/DIS 14825:2011) issued by the Technical Committee ISO/TC 204 is used to model, describe and interchange digital road map data. The rationale of the GDF development is to establish interoperability for exchanging digital road networks between map manufacturers and navigation system integrators in the Intelligent Transport Systems (ITS) and in-vehicle navigation market. Since its first release in 1996, major road map vendors and producers of automotive navigation systems such as NAVTEQ and TomTom have quickly adopted GDF. GDF has nowadays evolved to a mature outdoor navigation standard covering a wide range of ITS applications and services such as car (and portable) navigation systems or road traffic analysis and management. The GDF standard specifies a conceptual data model for describing road information based on the representation of real world objects (referred to as *features*), the characteristics of features, and the topological and non-topological interrelations between features.[62] Navigation constraints are understood as an integral part of road information and hence are addressed and supported by GDF.

In GDF, navigation constraints on features such as roads or junctions are modelled as attributes of the features or as attributes of the relationships between two or more features. Figure 193 shows an excerpt of the GDF attribute and relationship model that is part of the overall conceptual data model of GDF.

[61] Note that the *NT_LaneConstraint* is conceptually inconsistent with this modelling principle. Instead of using the attribute *applicableVehicleTypes*, the allowed or disallowed vehicle types should rather be denoted through co-constraints of type *NT_VehicleConstraint*.

[62] Note that, although road map data essentially captures geographic information, GDF does not employ the *General Feature Model* for the modelling of geographic features and their spatial and non-spatial characteristics as defined in the ISO 19100 standards family (cf. chapter 4.3). During the revision of GDF to its current version 5.0, major activities towards the harmonization with ISO/TC 211 standards were undertaken. An in-depth discussion of the GDF feature model and its differences to the ISO 19100 standards family is however out of scope of this thesis.

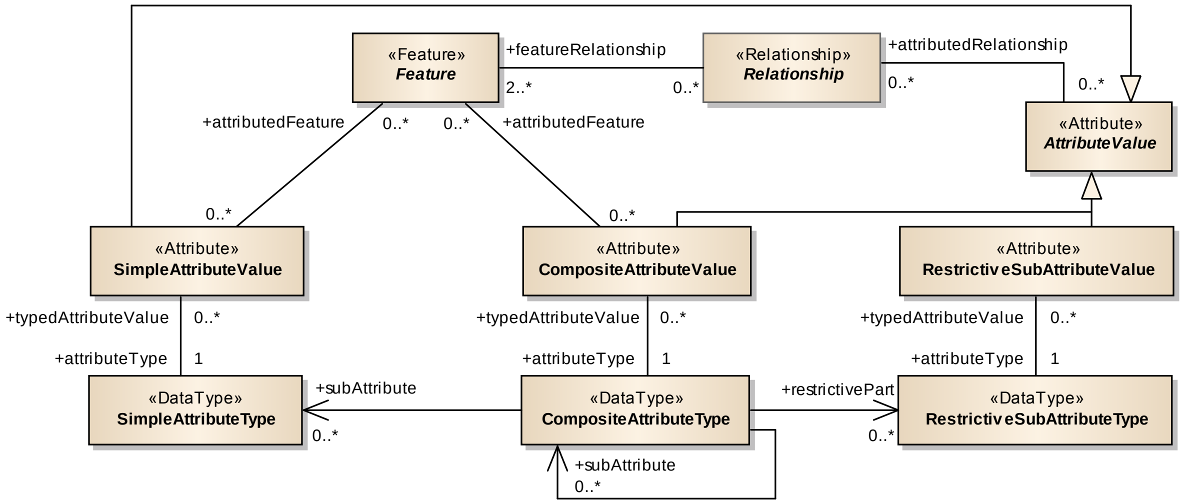


Figure 193: Excerpt of the GDF attribute and relationship model (after ISO/DIS 14825:2011).

The GDF attribute model distinguishes between *simple* and *composite attribute types*. In contrast to a simple attribute, a composite attribute is an aggregation of a number of attributes called sub-attributes. The value of a composite attribute is hence composed of the values of its sub-attributes which themselves only represent a part of the complete attribute information. A sub-attribute may in turn be simple or composite allowing for recursively building a hierarchical tree of sub-attribute sets. Attributes (simple or composite) are used to represent thematic information about a feature as well as navigation constraints associated with the feature. For instance, a simple attribute can model the *name* of the road or its *number of lanes* (thematic information) but can also denote the *minimum speed* or the *maximum height allowed* for vehicles that may use the road (legal *may* constraints). By attaching more than one attribute to the same feature, combined constraints can be expressed. In this case, all individual attributes have to be obeyed in order for the feature to be traversable (Boolean conjunction).

The validity of attributes can be further restricted to the value of a so-called restrictive sub-attribute. As opposed to normal (simple or composite) sub-attributes, the value specified in a restrictive sub-attribute may thus not be interpreted independently. If attached to a feature, restrictive sub-attributes always appear in combination with the attribute they restrict. However, they may appear alone on relationships and in these cases restrict the validity of the relationship. The GDF standard predefines the following restrictive sub-attributes: *lane dependent validity*, *linear segmentation*, *pedestrian type*, *scope*, *side of line*, *validity direction*, *validity period*, and *vehicle type*. For example, the *speed limit* attribute of a road element can be combined with a *vehicle type* sub-attribute. The resulting composite attribute limits the specific vehicle types for which the speed limit applies. If multiple restrictive sub-attributes are related to the same simple, composite or sub-attribute of a composite attribute, then they simultaneously restrict the attribute value and are interpreted in a Boolean conjunction. If multiple restrictive sub-attributes are related to different levels (logical groups) of a hierarchical attribute tree, then they are applied sequentially to the different logical groups which comes down to an intersection in set theory terms.

In addition to simple or combined constraints, GDF also supports complex constraints involving more than one feature instance. Conceptually, combined constraints are specialized relationship classes. Figure 194 illustrates the model of a *prohibited manoeuvre* constraint at road junctions. A prohibited manoeuvre legally restricts a physically possible movement at a junction from an incoming (*firstRoadElement*) to an outgoing (*secondRoadElement*) road element. A corresponding traffic sign as well as subsequent road elements and route guidance information (attribute *multimediaFileAttachment*) can be provided in addition. Note that due to its definition, a prohibited manoeuvre relationship is always uni-directional with the reverse manoeuvre direction not being prohibited (unless stated by a separate relationship instance). Relationships can carry attributes as well as individual restrictive sub-attributes (cf. figure 193). For example, a prohibited manoeuvre may be assigned a *validity period* to define that it is only valid during specific hours of the day (e.g., *9am-5pm*). Besides *manoeuvre*, GDF predefines further relationship classes (e.g., *fork*, *toll route*, *divided junction*, *through route*, *give way regulation*) which can be used to express complex navigation constraints.

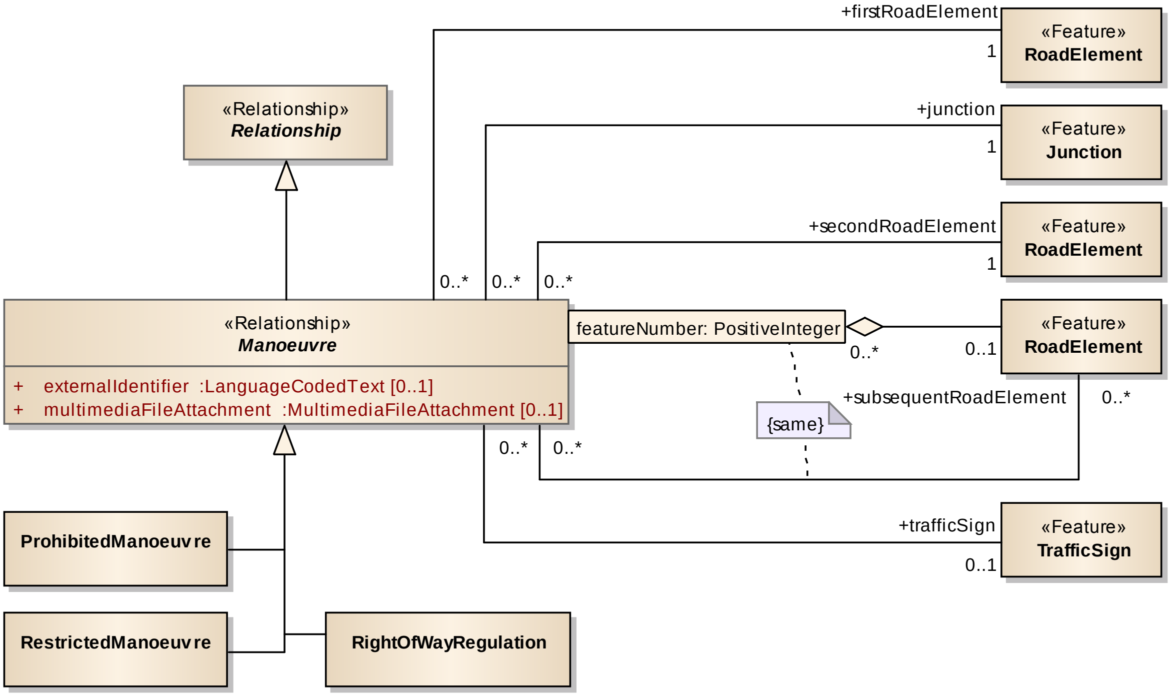


Figure 194: The GDF Manoeuvre relationship class allowing for the modelling of complex navigation constraints.

Similar to ISO 19133, GDF provides the possibility to define navigation constraints on road elements in a comprehensive way that exceeds simple constraints. However, and in contrast to ISO 19133, navigation constraints are not specified in a separate conceptual model but are expressed as normal attributes and relationships. Consequently, there is no categorization or differentiation into physical, logical, and temporal conditions or global and user-dependent constraints. Moreover, whether an attribute carries thematic information or denotes a navigation constraint cannot be answered from the data model and thus is not machine-understandable. The meaning of an attribute or relationship rather follows from the attribute and relationship catalogues as predefined by the GDF standard. Although many of the constraints are relevant in indoor environments as well (e.g., prohibited maneuvers), both catalogues target the domain of outdoor car navigation and hence lack important constraint types for indoor navigation (e.g., mode of locomotion or logical constraints such as access permissions).

5.3 Conceptual Data Model for Navigation Constraints

In the following, a conceptual model for expressing and evaluating navigation constraints as part of the MLSEM is proposed which addresses the various types of constraints as identified in section 5.1 and aims at overcoming the limitations of existing approaches as discussed in the previous section. The work undertaken in this thesis continues and comprehensively refines first ideas for a conceptualization of navigation constraints for indoor navigation in the context of the MLSEM as presented in (Brown et al. 2012)[63] (cf. appendix E).

Figure 195 shows the central concepts of the developed constraint model and their relations to the elements of the MLSEM schema. The model employs the rules and principles for defining application schemas as specified in ISO 19109 and is formally described using UML as conceptual modelling language.

[63] This work was carried out in a Master's thesis conducted at the Institute for Geodesy and Geoinformation Science, Technische Universität, Berlin, under the supervision of the author. A UML diagram representing the initial classification of navigation constraints as proposed in this former work is depicted in appendix E.

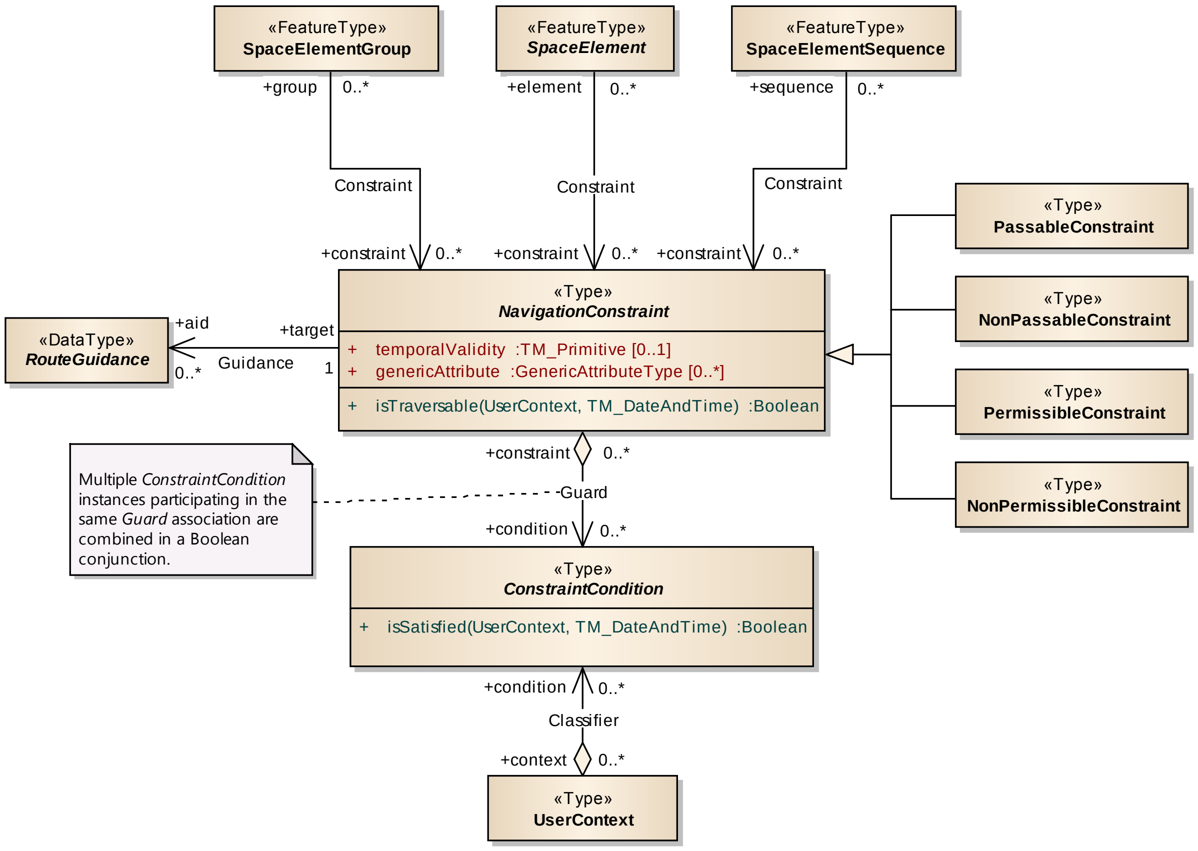


Figure 195: The MLSEM navigation constraint model.

5.3.1 Navigation Constraints

Navigation constraints are conceptually mapped to the abstract type *NavigationConstraint* which is further specialized into separate types for modelling *can* and *may* constraints according to the classification of (Hendricks et al. 2003). The category of physical *can* constraints is translated into the two concepts *PassableConstraint* and *NonPassableConstraint*. Whereas a *PassableConstraint* is used to define the conditions under which movement is physically possible for a navigation user, a *NonPassableConstraint* reflects the contrary concept and hence is used to mark space entities that are physically non-navigable. The physical ability to traverse a space entity can thus be expressed either in a positive (*can*) or in a negative (*cannot*) way. The similar approach also underlies the modelling of logical and legal movement limitations which can either be represented as permission (*may*) through the *PermissibleConstraint* concept or as denial (*may not*) using a *NonPermissibleConstraint*. A denial intuitively renders a navigation constraint since it defines the conditions under which movement is disallowed for a user. But also a permission can be viewed as a constraint in the sense that it restricts movement in case the conditions of the permission are not met by the user. In practice, it is often simpler to express a permission that implies a denial (e.g., "access for staff members only") than explicitly formulating all conditions for that denial (e.g., naming all user groups other than staff members for which access is disallowed) or vice versa. For this reason, the proposed constraint model contains both positive and negative concepts for physical and legal movement restrictions although one or the other would already be sufficient from a theoretical point of view. *Must* constraints are not reflected by separate concepts since they can also be represented using either a *PermissibleConstraint* or a *NonPermissibleConstraint*. However, user preferences (i.e., *should* constraints) are not covered by the MLSEM constraint model but are rather seen as input to cost functions that need to be considered in path finding algorithms. This conforms to the understanding of user preferences in other approaches to the modelling of navigation constraints (e.g., Tsetsos et al. 2006, Stoffel et al. 2007, ISO 19133:2005).

A *NavigationConstraint* can be applied to one or more *SpaceElement* features, *SpaceElementGroup* features, or instances of *SpaceElementSequence* (cf. chapter 4.4.1) through their *Constraint* associations. This flexibility allows for modelling *simple* constraints on single space cells and boundary cells as well as *complex* constraints on groups and sequences thereof. Attaching a navigation constraint to a *SpaceElementSequence* explicitly includes the possibility to express directed and undirected prohibited maneuvers. If a single *SpaceElement* is the target of multiple navigation constraints (either alone or as member of a group or sequence) then this set of constrains obviously renders a *combined* constraint on that *SpaceElement* feature. Note that since the feature types *TransferTransition* and *TransferState* of the MLSEM *Model Linkage* package (cf. chapter 4.4.1.6) are also defined as (transitive) subtypes of *SpaceElement*, navigation constraints can likewise be modelled for the linkage between two separate navigation networks (e.g., between an instance of the MLSEM and a road network).

Every instance of *NavigationConstraint* can be further characterized based on optional semantic attributes. Similar to ISO 19133, the attribute *temporalValidity* is used to specify a time frame within which the constraint is deemed valid and thus has to be considered by a navigation system when routing a user. The temporal validity is not to be confused with a temporal access constraint such as opening hours of a shop or office, but rather defines when these opening hours are effective. For example, a temporal constraint may denote *"Monday-Friday, 9am-5pm"* as opening hours, whereas the temporal validity *"July-August"* restricts these opening hours to a specific season. The temporal validity attribute also allows for providing dynamic constraints that modify a navigation system within a predefined time frame in order to react on a current situation or scenario (e.g., adding prohibited maneuvers to control the flow of navigation which are only valid in heavy traffic times or within evacuation scenarios). Further semantic information about the navigation constraint can be provided by means of *generic attributes* (cf. chapter 4.4.1.1). Moreover, each navigation constraint can be enriched with *RouteGuidance* information as defined in the MLSEM *Route* package (cf. chapter 4.4.1.5). This is especially important in case of movement restrictions that do not affect the route itself but nevertheless need to be communicated to the navigation user. In contrast to ISO 19133, the guidance is not restricted to textual descriptions but also comprises visual and audio presentations.

5.3.2 Constraint Conditions

Whereas the instances of *NavigationConstraint* and their relations to the space entities of the MLSEM schema classify the type of the navigation constraint, the preconditions under which the constraint applies are modelled through the abstract concept *ConstraintCondition*. A *ConstraintCondition* is thus to be understood as *guard* which prevents the application of the navigation constraint in case the precondition is not satisfied. Each *NavigationConstraint* may be controlled by zero or more guards as denoted by the *Guard* association. Obviously, the relevant conditions for different movement restrictions will substantially vary for different indoor environments, types of navigation users, and navigation scenarios. The aim of the MLSEM constraint model is hence to provide a common abstraction that is expressive enough to represent typical constraint conditions that are required in the context of indoor navigation. The UML diagram in the following figure 196 illustrates the proposed conceptualization and taxonomy for constraint conditions.

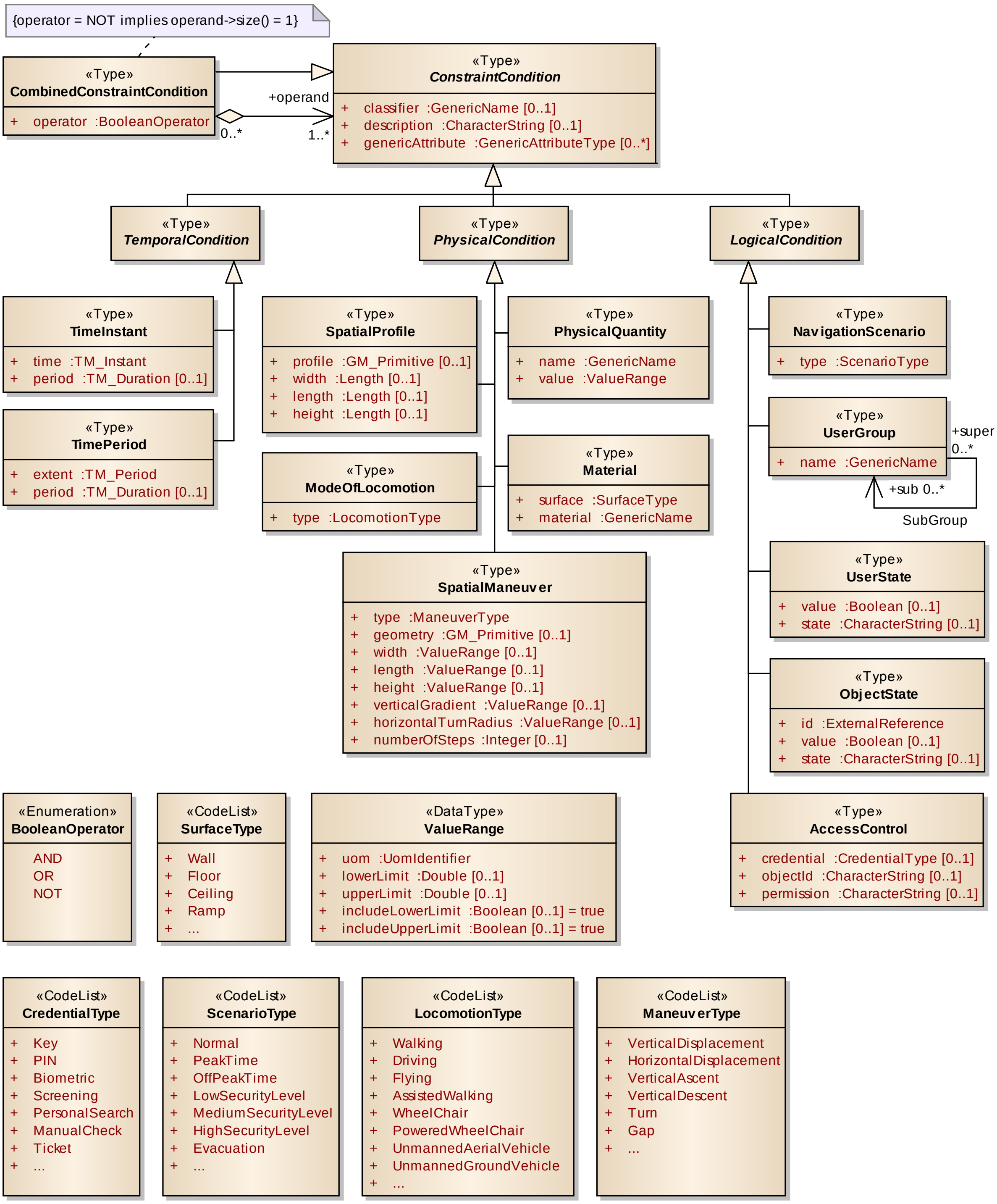


Figure 196: Conceptualization of constraint conditions guarding the application of navigation constraints.

Following the terminology of the previous sections, constraint conditions are classified into three categories, namely *temporal*, *physical*, and *logical* conditions, which are represented by the corresponding conceptual entities *TemporalCondition*, *PhysicalCondition*, and *LogicalCondition*. Each condition type inherits the attribute *classifier* from the root class *ConstraintCondition* which optionally can be used to define the meaning and function of the constraint condition. The *classifier* shall be taken from a predefined list of unique values that are maintained and given semantic meaning by either the navigation application or a global authority. Two or more instances of *ConstraintCondition* that share the same *classifier* must be of the same type in order to be comparable for a navigation system. Examples for the usage of the *classifier* attribute are given in the further course of this chapter. In addition, constraint conditions can carry a textual *description* that may be presented to the navigation user (e.g., as route guidance). Semantic information required by a navigation application but not foreseen by the generic MLSEM constraint model can be added as *generic attributes*.

The category of temporal conditions is populated by the two concrete types *TimeInstant* and *TimePeriod*. A *TimeInstant* is used to specify a point in time at which the navigation constraint holds. In accordance with the ISO 19108:2005 Temporal Schema, this specific point in time is provided as *time* attribute whose value domain is *TM_Instant*. The optional attribute *period* allows for specifying whether the time instant recurs on a regular basis and identifies the duration of the interval between two successive occurrences. The *TimePeriod* condition denotes an *extent* in time using *TM_Period* (i.e., a duration in the temporal dimension defined by a beginning and an end point given as *TM_Instant*) during which the constraint applies and may likewise be augmented with a periodic time. Both temporal conditions suffice to express diverse time values such as precise calendar dates (e.g., *"Wednesday, September 26, 2012"*) or abstract time intervals (e.g., *"every first Monday of the month"*, *"Monday-Friday, 9am-5pm"*).[64]

Physical conditions are modelled through the data types *SpatialProfile*, *SpatialManeuver*, *PhysicalQuantity*, *Material*, and *ModeOfLocomotion*. A *SpatialProfile* specifies a profile the navigation user must be able to physically fit through in order to traverse the space entity. The description of the profile can either be provided as geometric object in up to three dimensions or as parametric representation using one or more of the attributes *width*, *length*, and *height* whose value domain *Length* (from ISO/TS 19103:2005) facilitates to define a unit for each parameter. For example, a *SpatialProfile* can be used to express that a door has a width of $90cm$ and thus renders a physical obstruction to movement if the navigation user exceeds this width restriction. If the profile is additionally enriched with a *height* value then the navigation user must also be able to observe that height. A parametric description of the spatial profile can obviously be easily checked by a navigation system. However, more complex shapes of the spatial profile are also supported and are given as instance of *GM_Primitive*. For instance, the free space for passing through a door can be modelled as 2-dimensional vertical surface or even as 3-dimensional volumetric geometry. Likewise, a geometric object may describe a narrowing corridor whose width changes along its length. The geometry of the spatial profile needs not be identical to the primal space representation of the space entity associated with the navigation constraint but may, for example, be a generalized representation that simplifies geometric checks. If the application needs more precise information, then the primal space geometry of the space entity should be queried directly.

Often travelling along space entities or overcoming obstacles requires the physical ability to perform a specific movement or maneuver. For example, a navigation user must be able to manage steps of a given height in order to climb a stair or to enter the ledge of a window that serves as emergency exit. Similarly, ramps can only be navigated if the user is able to cope with the slope and acute angles in corridors might require a minimum turn radius especially for driving users such as wheelchair users or mobile robots. The *SpatialManeuver* condition allows for defining the spatial characteristics of such movements that need to be met by the navigation user in order to traverse the space entity. The type of the maneuver has to be specified through the mandatory *type* attribute whose values are taken from the extensible code list *ManeuverType* that contains predefined values such as *VerticalDisplacement*, *Turn*, and *Gap*. The parameters *width*, *length*, and *height* are used for spatially describing the movement. Their data type *ValueRange* allows for defining intervals of values associated with a unit of measure. The endpoints of the interval are encoded through the attributes *lowerLimit* and *upperLimit*. If one of both attributes is not set, then an infinite endpoint is assumed rendering the interval half-bounded. The Boolean attributes *includeLowerLimit* and *includeUpperLimit* denote whether the respective endpoint should be included in the interval and hence allow for modelling (semi-)open intervals when set to false. The unit of measure for the amount is encoded through the *uom* attribute (e.g., as conventional identifier[65] such as *"m"* or *"kg"*, or as reference to a definition). The slope of the required movement can be characterized using the attribute *verticalGradient*, whereas the radius of curvature to be met by the user can be given as *horizontalTurnRadius*. Both attributes are also of type *ValueRange*. If the movement has to be performed inside a specific spatial area, then this area can be expressed as either 2-dimensional or 3-dimensional *geometry* using a *GM_Primitive*. Finally, the *numberOfSteps* attribute facilitates to announce the number of steps involved in the movement. For example, a *SpatialManeuver* condition of type *VerticalDisplacement* participating in a navigation constraint associated with a stair can be used to express that the navigation user must be able to cope with a vertical *height* difference of at least $20cm$ in order to take the steps. Additionally, the minimum requirements to negotiate the step treads can be announced using the *width* and *length* attributes or by

[64] The proper provision of time values using the conceptual classes *TM_Instant*, *TM_Period*, and *TM_Duration* is specified in ISO 19108:2005 and thus out of scope of this thesis. Corresponding XML encodings are defined by GML (ISO 19136:2007).

[65] For example, according to the *Unified Code of Units of Measure* (UCUM) specification which is freely available at http://aurora.regenstrief.org/~ucum/ucum.html.

providing a 2-dimensional horizontal geometry. In order for the navigation application to correctly interpret the provided parameters, the *SpatialManeuver* should be identified through a unique *classifier* that is accompanied by an unambiguous definition.

The *PhysicalQuantity* condition is used to model a measure of some quantifiable aspect of a space entity that needs to be observed when navigating the entity. The *name* attribute identifies the physical phenomenon to be modelled (e.g., *"Weight"* or *"Speed"*), whereas the *value* and its unit of measure are given as instance of the *ValueRange* data type. The semantic meaning of the physical quantity is specified by the *classifier* attribute. For example, a *PhysicalQuantity* condition might announce a maximum weight of $80kg$. A possible classifier *"UserWeight"* then indicates that only navigation user not exceeding this weight limit are allowed to traverse the space entity, whereas a classifier *"SupportedLoad"* might refer to the load supported by the space entity itself (e.g., the maximum load of a glass floor surface or an elevator).

Since specific materials of surfaces cannot be crossed by certain navigation users (e.g., by wheelchair users, visually impaired persons, or mobile robots), the *Material* condition enables to describe the surface material in a corresponding navigation constraint. The *surface* in question is identified through a value from the extensible code list *SurfaceType*. Its *material* is encoded as character string whose possible values should be predefined for a navigation system. Finally, the physical condition type *ModeOfLocomotion* restricts a navigation constraint to a specific mode of locomotion of the navigation user. Predefined locomotion types include *Walking*, *Driving*, *Flying*, *AssistedWalking*, and *WheelChair*. If necessary, this list can be extended to also represent more specific modes of locomotion such as *WalkingWithCrutches*.

The subtypes of *LogicalCondition* provide the possibility to model legal or logical constraint conditions. At first, the *NavigationScenario* type defines the scenario for which the navigation constraint is valid. For example, permission to use an emergency door may be restricted in normal navigation situations but granted in case of an evacuation. The *type* of the *NavigationScenario* is taken from the extensible code list *ScenarioType*. In order to assign a navigation constraint to a specific group of navigation users, a *UserGroup* condition can be modelled. The user group is textually described through the *name* attribute (e.g., *"staff member"*, *"visitors"*, *"visually impaired persons"*, *"transportation robots"*, etc.) whose values may come from a predefined code list. User groups may be hierarchically organized based on the *SubGroup* association linking *UserGroup* with itself (e.g., *"staff member"* can be refined into *"facility staff"*, *"scientific staff"*, etc.). Navigation constraints modelled for a parent group also hold for all its subgroups, whereas each subgroup may receive more fine-grained permissions or restrictions than its (transitive) parent groups. The *UserState* condition expresses an actual state of the navigation user that must be met for the constraint to apply. At airports, for instance, users should only be routed to the security gates after having checked in their luggage and if they carry a boarding pass. In most cases, the user state can be given as Boolean *value* with the *classifier* attribute providing an identifier for the user state in question. For the given example, possible classifiers are *"hasCheckedInLuggage"* or *"hasBoardingPass"* which can be marked for every user as being true or false. If a Boolean value is not sufficient then the user state can be encoded as character string using the *state* attribute (e.g., expressed in natural or formal language). Similarly, the *ObjectState* condition requires an object of the indoor environment to be in a certain state. It follows the same semantics as *UserState* but carries an additional *id* attribute of type *ExternalReference* that specifies the object itself. Obviously, a navigation system must be capable of setting, querying, and tracking user and object states when using these conditions.

The last logical condition *AccessControl* is used to represent the various types of access restrictions on space entities that require some sort of authorization of the navigation user in order to be allowed to traverse the entity. The type of credential that enables access is announced through the *credential* attribute whose values are taken from the extensible code list *CredentialType*. A credential may be a physical object (e.g., a mechanical key or a key card), something the user knows (e.g., a password or a PIN), or a biometric feature of the user (e.g., a fingerprint or iris recognition). Likewise, access may require a screening or a personal search of the user (e.g., at security gates) or may depend on payment (e.g., fares checked by a ticket controller). The accompanying *objectId* attribute optionally identifies the actual physical item or virtual credential to be used in the authorization process. For example, a possible value for *objectId* is the unique number of the mechanical key or key card that unlocks a specific door. Alternative or complementary permission rights required for accessing the space entity can be denoted using the *permission* attribute.

Combined Constraint Conditions. In many use cases, a navigation constraint involves multiple preconditions. For example, access to a room may be restricted to predefined opening hours for specific user groups, whereas holders of a corresponding key might be allowed to enter the room at any time except weekends. Expressing this access constraint obviously requires a combination of *TimePeriod*, *UserGroup*, and *AccessControl* conditions. The *CombinedConstraintCondition* type facilitates the modelling of such combinations. Precisely, a combined constraint condition is a Boolean operation on a set of one or more instances of *ConstraintCondition* being the operands. Each combined condition has to announce the Boolean *operator* that is applied to its operands and which is taken from the *BooleanOperator* enumeration. The supported operations are logical conjunction (*AND*), disjunction (*OR*), and negation (*NOT*). While both *AND* and *OR* can take more than one operand, *NOT* is an unary operator and thus is restricted to take a single *ConstraintCondition*. This restriction is enforced by the following OCL invariant on *CombinedConstraintCondition*.

```
context CombinedConstraintCondition inv:
    operator = NOT implies operand->size() = 1
```

CombinedConstraintCondition itself is a subtype of *ConstraintCondition*. Therefore, each combined condition may take the form of an operand within another combined condition. By this means, arbitrarily complex and nested Boolean expressions can be formulated, which also allows for deriving further Boolean operations such as implication from the predefined *AND*, *OR*, and *NOT* operations by composition. As stated above and shown in figure 195, a *NavigationConstraint* can be linked with multiple constraint conditions. The *ConstraintCondition* instances participating in the corresponding *Guard* association are combined in a conjunction per definition. According to the classification of navigation constraints elaborated in chapter 5.1, the combination of several conditions renders the constraint itself a *combined constraint*.

5.3.3 User Context

Since user-dependent navigation constraints need to be evaluated against the capabilities of the user, the MLSEM constraint model foresees the possibility to provide user-related contextual information through the data type *UserContext*. As shown in the UML diagram in figure 195, a *UserContext* essentially is an aggregation of constraint conditions. Thus, the same conceptual entities used for describing the preconditions for navigation constraints are also used for classifying the physical, logical, or temporal preconditions of a navigation user.

This mechanism allows for answering constraint conditions by querying the *UserContext* for a matching instance of the same condition type. The *classifier* attribute can hereby be used to explicitly relate constraint conditions with corresponding information from the user context. For example, assume a *NonPassableConstraint* denotes that a space entity cannot be traversed if the user weighs more than $80kg$ using a *PhysicalQuantity* condition that carries the classifier *"UserWeight"* (see example above). Further suppose that the *UserContext* of a navigation user also contains an instance of *PhysicalQuantity* being classified as *"UserWeight"* which provides the actual weight of that user. Then testing whether the constraint condition is satisfied simply requires comparing both measures. A *UserContext* may provide more than one instance of the same condition type such as several groups the user belongs to (*UserGroup*), a set of access credentials such as keys the user possesses (*AccessControl*), or multiple user states (*UserState*). The mode of locomotion of the user can be expressed using an instance of *ModeOfLocomotion*, and a *SpatialProfile* facilitates to provide a parametric or geometric description of the user's spatial extent. More than one spatial profile is useful in order to check a physical navigation constraint against different possible extents of the user (e.g., whether the user fits through a door when walking upright, when crouching, or when walking sideways). Maneuver capabilities of the user can be expressed through instances of *SpatialManeuver* (e.g., the maximum height difference for taking steps, the maximum turn radius, or the maximum slope supported by the user or the assistive device). Also temporal conditions might be part of the *UserContext*. For instance, assume a user plans a future trip. A temporal condition could then denote the time frame (*TimePeriod*) within which the user wants to travel. This time frame could be tested against temporal navigation constraints such as opening hours. Likewise, the user context may contain a *NavigationScenario* in order to plan or simulate a route for a specific scenario.

The examples demonstrate the expressivity of the proposed constraint model. However, it is important to note that the presented *UserContext* is intentionally tailored to the information needed for evaluating navigation constraints. The task of navigation typically requires more information about the navigation user such as the capabilities of the mobile device in order to choose proper sensor space layers for localization and tracking (cf. chapter 1.2). A conceptual model for such a comprehensive user profile is however out of scope of this thesis. If existent, navigation constraints may likewise be checked against that profile rather than relying on *UserContext*. Alternatively, the profile may be used to populate the *UserContext* instance for a given navigation user. The MLSEM constraint model is flexible enough to support either alternative.

5.3.4 Evaluation of navigation constraints

The conceptual constraint model discussed so far allows for representing knowledge about various types of navigation constraints. In the following, the fundamental rules for reasoning and inference about this knowledge are presented.

The default assumption of the MLSEM constraint model is that any space cell or boundary cell is traversable. If, however, a space cell or boundary cell is the target of at least one navigation constraint (may it be simple or complex), then the opposite holds true and the space cell or boundary cell is assumed to be non-traversable. In this case, traversability needs to be answered from the evaluation of the associated constraints.

The operation *isTraversable* inherited from *NavigationConstraint* is used to decide whether a given navigation constraint restricts the movement of a specific navigation user along the associated space entity. It expects both a *UserContext* and the time of navigation (e.g., the current time or a future time) encoded as *TM_DateAndTime* from ISO 19108 as input, and returns a Boolean value as output. Obviously, if *isTraversable* evaluates to true then the navigation constraint does not impede the movement of the user and vice versa. The outcome of *isTraversable* depends on 1) the temporal validity of the navigation constraint, 2) the evaluation of the guarding constraint conditions, and 3) the type of the navigation constraint.

The temporal validity is checked against the provided time of navigation. Only in case a temporal validity period is modelled for the navigation constraint and the time of navigation is outside that period, the navigation constraint can be considered ineffective and *isTraversable* returns true without further checks. In all other cases, the constraint conditions determine the application of the navigation constraint. Whether or not a constraint condition is met is denoted through its *isSatisfied* operation inherited from the root class *ConstraintCondition*. Similar to *isTraversable*, the *UserContext* and the time of navigation have to be passed to this operation. A condition is deemed to be satisfied if it can be answered from the user context. For example, if a *SpatialProfile* denotes a door width of $90cm$ and the user has a width of $80cm$, then the *isSatisfied* operation shall return true. Likewise, if the user group *“staff members”* is expected but the navigation user does not belong to this group, then the *isSatisfied* operation for this *UserGroup* condition has to yield false. User-independent conditions are examined in the same way. For instance, assume the time of navigation is outside the opening hours given by an instance of *TimePeriod*, then obviously *isSatisfied* renders false for this condition. If the navigation system lacks the information to evaluate the constraint condition (e.g., the user profile does not contain the user width or group), the default assumption is that the condition is not satisfied.[66] In case of a *CombinedConstraintCondition*, the output of *isSatisfied* is the result of applying the modelled Boolean operation to the outcomes of the *isSatisfied* operations of the operands.

The application of a *NavigationConstraint* requires that all of its guards (i.e., all conditions participating in the *Guard* association, cf. figure 195) are fulfilled and hence their *isSatisfied* operations yield true. Consequently, a single guard not being met already suffices to prevent the application of the constraint. If the navigation constraint is not controlled by guards, then it always applies. As stated above, the result of the *isTraversable* operation finally depends on the type of the constraint. For the group of positively formulated navigation constraints (i.e., *PassableConstraint* and *PermissibleConstraint*), *isTraversable* must return true if the constraint applies, and false otherwise. The reverse holds for the negative counterparts *NonPassableConstraint* and *NonPermissibleConstraint*. Thus, if a negatively formulated navigation constraint applies, then the *isTraversable* operation has to return false.

In order to illustrate the separate steps involved in the evaluation of a navigation constraint, the *isTraversable* operation of the *NavigationConstraint* class is translated into the algorithm 5.1 as shown below. The algorithm takes a single instance c of one of the subtypes of *NavigationConstraint*, a representation $User$ of the user context, as well as the time t of navigation as input parameters. The output is a Boolean flag $isTraversable$. The algorithm

[66] Of course, the navigation system may first query the navigation user for lacking information or make use of predefined group profiles before applying the default assumption.

presupposes an implementation of the *isSatisfied* operation of *ConstraintCondition* (cf. line 5). Since this implementation depends on the representation of the user context which may vary for different navigation systems, it is not further specified here but subject to future research[67]. Likewise, functions for checking the temporal validity of c (cf. line 2), for getting all guards associated with c (cf. line 3), and for checking the type of c (cf. line 8) need to be provided by the navigation system.

Algorithm 5.1. IsTraversable($c, User, t$)

```
Input:   $c, User, t$
Output:  $isTraversable \in \{true, false\}$
1:  $isTraversable \leftarrow true$
2:  if $IsWithinTemporalValidity(c, t) = true$ then
3:      $G \leftarrow GetGuards(c)$
4:      for each $g \in G$ do
5:          if $IsSatisfied(g, User, t) = false$ then
6:              $isTraversable \leftarrow false$
7:              break
8:  if $IsNonPassableConstraint(c) = true \lor IsNonPermissibleConstraint(c) = true$ then
9:      $isTraversable \leftarrow \neg isTraversable$
```

A space entity can only be traversed by a given navigation user if the *isTraversable* operations of all navigation constraints associated with that space entity evaluate to true. Thus, multiple navigation constraints on a single space entity are combined in a logical conjunction, and a single constraint returning false for its *isTraversable* operation renders the space entity non-traversable. This rule conforms to an intuitive understanding for the group of *can* constraints since a physical movement restriction cannot be overruled by a complementary *can* or *may* constraint. However, one (or a subset) of a set of *may* constraints being satisfied might already suffice for a navigation user to be allowed to navigate the space entity (e.g., suppose an office may be entered during office hours or by staff members holding a key). Such alternative legal or logical permissions cannot be expressed in terms of a logical conjunction and hence are not to be mapped onto separate *may* constraints on the space entity, but rather need to be captured by a single *PermissibleConstraint* or *NonPermissibleConstraint* that has separate guards for the alternative permissions combined in a logical disjunction (using a *CombinedConstraintCondition*).

Based on algorithm 5.1, the following simple $IsTraversable$ algorithm determines the traversability of a space entity S (i.e., a space cell or a boundary cell) according to the presented rule base. It also expects a user context $User$ and the time of navigation t as input and returns a Boolean flag $isTraversable$. The $GetConstraints$ function in line 2 of the algorithm is assumed to return the set of (simple or complex) navigation constraints that address the space entity S.

Algorithm 5.2. IsTraversable($S, User, t$)

```
Input:   $S, User, t$
Output:  $isTraversable \in \{true, false\}$
1:  $isTraversable \leftarrow true$
2:  $C \leftarrow GetConstraints(S)$
3:  for each $c \in C$ do
4:      if $IsTraversable(c, User, t) = false$ then
5:          $isTraversable \leftarrow false$
6:          break
```

Space cells and boundary cells, which are non-traversable according to algorithm 5.2, shall not be part of the route of the navigation user. First, this means that constraint conditions that shall not affect the route of the user such as speed limits or movement type restrictions must always return true for their *isSatisfied* operation. This has to be

[67] At the time of writing this thesis, methods for defining and evaluating the user context based on the proposed conceptual model for navigation constraints are worked on in a Master's thesis conducted at the Institute for Geodesy and Geoinformation Science, Technische Universität, Berlin. Results will be published as soon as the work will have been completed.

ensured by the navigation system (e.g., such conditions could by identified through a unique value of their *classifier* attribute). Second, there might be situations in which a navigation system has to deviate from this rule. For example, assume a navigation user is actually localized within an area for which she lacks access permission. Then adhering to *may* constraints associated with the surrounding space entities may prevent the system from finding a path that guides the user outside the prohibited area. Thus, certain *may* constraints could be temporarily suspended for this user. On the contrary, the navigation system could likewise trigger an alarm in this situation and refuse to route the user. Since a multitude of possible behaviours is possible, such exceptional situations are out of scope of the MLSEM constraint model but have to be solved for a specific navigation system.

5.3.5 Integration with the MLSEM Application Schema

The conceptual integration of the proposed constraint model with the MLSEM schema presented in chapter 4.4.1 is illustrated by means of a UML package diagram in the following figure 197. The constraint model is captured by the package *Constraints* whose dependencies to further MLSEM packages as well as schemas from the ISO 19100 standards family is depicted through dashed arrows. The mapping of the constraint model to a GML application schema is documented in appendix B.7.

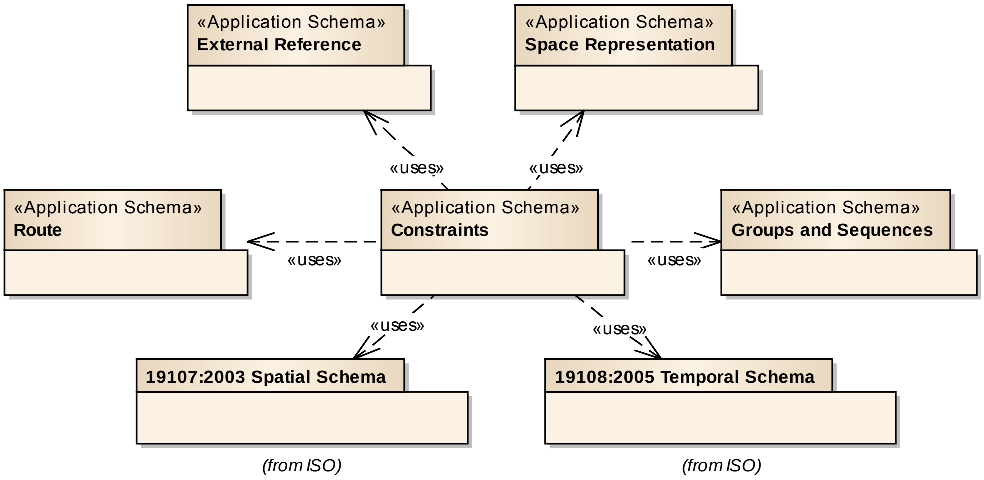


Figure 197: The integration of the Constraints package with the MLSEM application schema.

5.4 Example Usage of the MLSEM Constraint Model

The usage of the MLSEM constraint model is illustrated in this section with selected examples of navigation constraints in indoor environments. The examples are meant to demonstrate the modelling of various types of constraints based on the conceptual entities introduced in the previous chapter. For this reason, UML object diagrams are presented for each example that capture the view on the instances as well as their attributes and associations that are required to express the individual constraints.

Example 5.3. The first example shows how to explicitly mark space cells and boundary cells representing obstacles or tangible boundaries as being physically non-passable. In figure 198a, a simple setting of two 3-dimensional space cells modelling a room and a contained column is sketched. One boundary surface of the room cell is assumed to be a wall. Both the *SpaceCell* feature representing the column and the *BoundaryCell* capturing the wall are associated with a separate instance of *NonPassableConstraint*. The constraints have no further conditions and thus render global constraints that restrict movement for all navigation users. Figure 198b depicts the scene from a top view and includes an excerpt of the corresponding intra-layer graph in dual space. Note that the same instances of *NonPassableConstraint* are also available from to the dual node and dual edge representing the column respectively the wall due to the one-to-one correspondence between the primal and dual space representation of space cells and boundary cells.

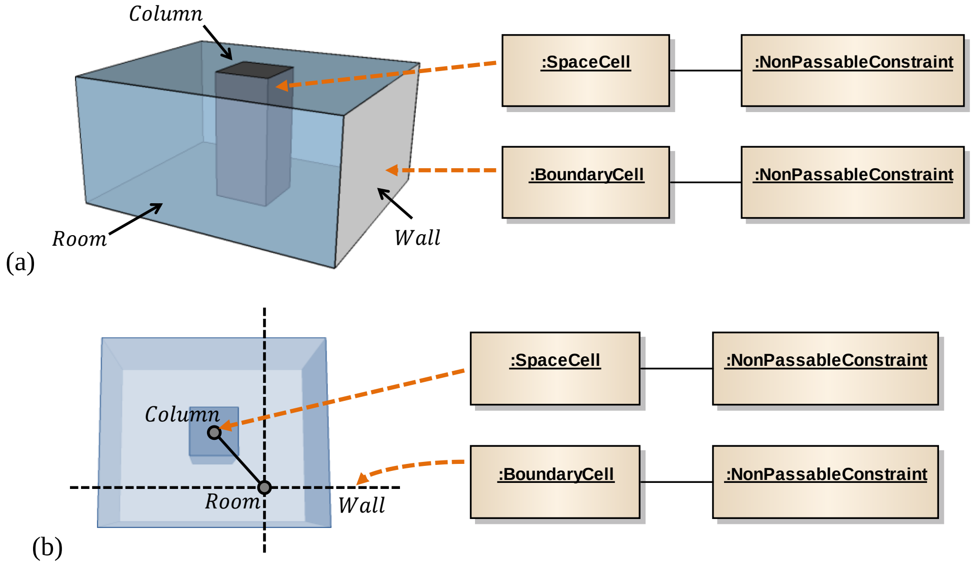


Figure 198: Modelling of non-passable constraints and their availability in primal space (a) and dual space (b).

Using a navigation constraint without further preconditions is equivalent to the modelling of a simple and static Boolean flag $isNavigable$ as discussed in example 3.63 (cf. chapter 3.4). In fact, example 3.63 can be easily realized without information loss by using a *NonPassableConstraint* instead.

The *NonPassableConstraint* instances in figure 198 are simple constraints since they target a single space entity. Alternatively, the column cell and the wall cell could be modelled as members of a *SpaceElementGroup* feature that receives the navigation constraint. Likewise, separate groups for different types of obstacles (e.g., one group for all walls contained in an indoor space model, one group for all columns, etc.) can be defined and associated with individual navigation constraints. The MLSEM constraint model supports this flexibility. Although all alternatives have the same effect (i.e., the space entities are regarded non-passable), one or the other may be preferred from a modelling or data management point of view.

Example 5.4. Figure 199 illustrates the usage of a *SpatialProfile* condition to denote that a navigation user has to observe a maximum width of $70cm$ and a maximum height of $2.3m$ in order to fit through a door connecting two rooms. The scene is depicted in two dimensions with each room represented as space cell. The door itself is modelled as 1-dimensional boundary cell shared by both rooms.[68] The navigation constraint is given as instance of *PassableConstraint* on the door cell. This positive constraint implies that the door physically obstructs movement for all navigation users who exceed the width and height restriction. Note that whereas the width of the door could also be implicitly deduced from the primal space geometry of the door cell, the height parameter is obviously not available from the geometry.

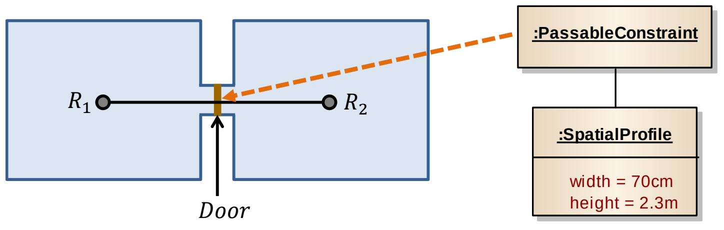


Figure 199: Using a SpatialProfile condition to constrain the movement through a door modelled as boundary cell.

Assume two navigation users *Alice* and *Bob* whose navigation contexts are depicted as instances of *UserContext* in figure 200. Both contexts contain a *SpatialProfile* which can be used to check whether the door restrictions are met. Whereas *Alice* obviously fits through the door based on her provided *width* and *height*, *Bob* is assumed to be a wheelchair user whose wheelchair is wider than the door. Thus, the *SpatialProfile* condition of the door is not

[68] The door may also be represented as 2-dimensional space cell instead. Although this would affect the layout of the intra-layer graph, it has no impact on the modelling of the navigation constraint.

met by *Bob*, and the *isTraversable* operation of *PassableConstraint* will yield false. The navigation system therefore has to search for another path for *Bob*.

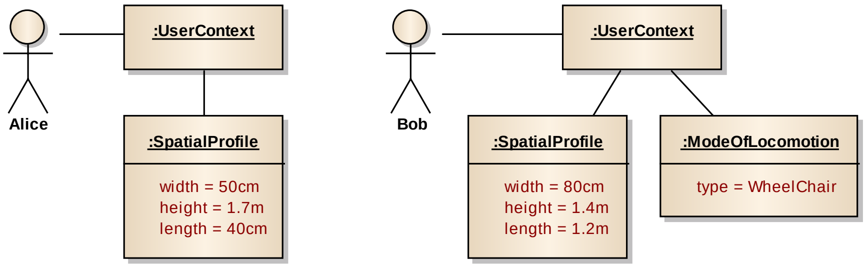


Figure 200: Two possible user contexts for evaluating the navigation constraint from figure 199.

Example 5.5. The scene in figure 201 shows four space cells mapping a staircase with two connected places at the bottom and at the top of the stair. The space cell representing the free space above the stair body is associated with a *NonPassableConstraint* stating that it is physically non-traversable for wheelchair users and for users who can at most negotiate steps of a height of 20cm. Whereas the former condition is expressed using an instance of *ModeOfLocomotion*, the latter condition is given by a *SpatialManeuver* entity of type *VerticalDisplacement*. Note that the *SpatialManeuver* does not model the steps themselves but rather imposes a requirement on the physical capability of the user. It additionally announces the number of steps of the stair. Although a navigation user might generally be capable of taking a single step of that height, the stair might nevertheless render an obstacle if it involves too many steps (e.g., for elderly people). Both constraint conditions participate in a *CombinedConstraintCondition*. The stair body itself renders an obstacle for all navigation users and is hence assigned an unconditioned *NonPassableConstraint*.

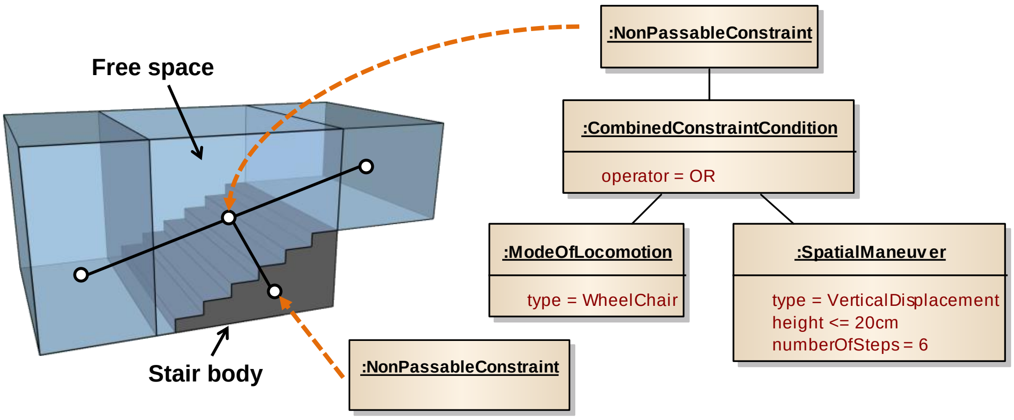


Figure 201: Constraining a stair for wheelchair users and users not capable of negotiating a given step height.

In the above example, a single space cell is used to capture the free space above the stair body. In contrast, the free space above each individual step may be modelled through distinct (sub)space cells. This fine-grained representation then even allows for associating different navigation constraints with each step. A similar step-wise representation of stairs is proposed in the grid-based LEGO approach of (Yuan & Schneider 2010b) (cf. figure 2c).

Another example for *SpatialManeuver* is illustrated in figure 202. It shows a room containing a landing in front of a door and a ramp that allows entering the landing. This setting recaps the same configuration as discussed in example 3.63 of chapter 3.4 (cf. figure 116). The free space inside the room is decomposed into four space cells S_1 to S_4. Additional space cells for representing the ramp and the landing themselves are omitted for readability. The free space S_2 above the ramp is associated with a *PassableConstraint* whose *SpatialManeuver* requires a user to be capable of coping with a slope of at least $10\ degrees$ over a length of $2.5m$ in order to navigate S_2. In addition, a *SpatialProfile* expresses that the user must meet a maximum width of $2m$ to use the ramp. The boundary cell shared by S_2 and S_3 is assigned an unconditioned *NonPassableConstraint* in order to avoid routing a user off the edge of the ramp. A final *NonPassableConstraint* marks the boundary cell between S_3 and the free space S_4 above the landing non-navigable if the navigation user cannot negotiate a vertical height difference of $30cm$.

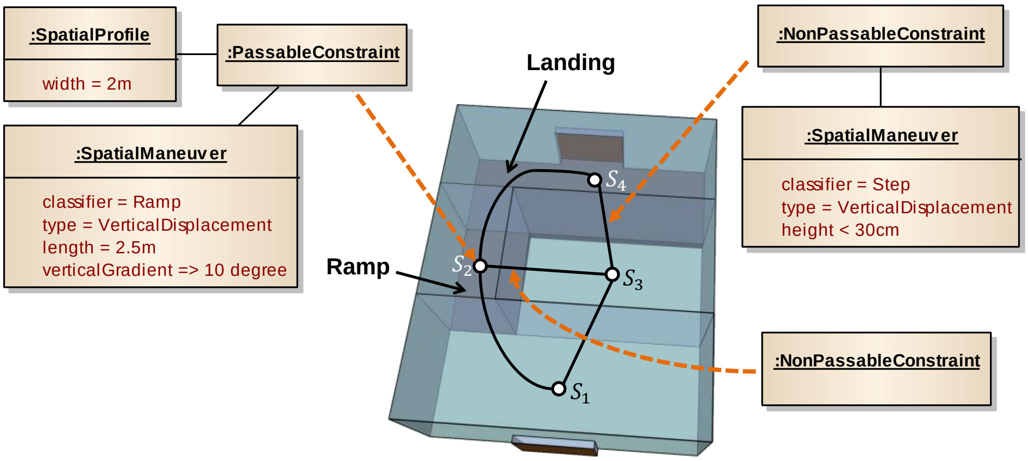


Figure 202: Fine-grained representation of navigation constraints for the scenario from figure 116.

In figure 203, a T-shaped section of two corridors is shown in two dimensions. The corridors are carried to four subspace cells C_1 to C_4. An undirected *SpaceElementSequence* generated from the space cells C_4, C_2, and C_3 denotes a possible turn at this section. The value of the *sequenceNo* qualifier (cf. chapter 4.4.1.4) for each element associated with the sequence is depicted in curly brackets in the UML object diagram. The sequence is the target of a *PassableConstraint* that marks the turn to be passable only if the navigation user (or its assistive device) has a minimum turn radius of less than 90 $degrees$. The corresponding *SpatialManeuver* condition further denotes the 2-dimensional area in which the turn has to be executed (depicted as dashed area in figure 203). This geometry is generally not required to coincide with the spatial extent of a space cell involved in the constraint. A similar turn restriction could be expressed for the undirected sequence $\{C_4, C_2, C_1\}$. Note that a candidate path must contain a subpath which entirely (and not just partly) satisfies the *SpaceElementSequence* in order for the associated navigation constraint to be considered. Thus, the *PassableConstraint* in the above example is only evaluated if a candidate path for a navigation user contains the dual nodes of C_4, C_2, and C_3 one after another (or vice versa). In contrast, the constraint is not applied if the navigation user is only routed from C_4 to C_2 but not to C_3.

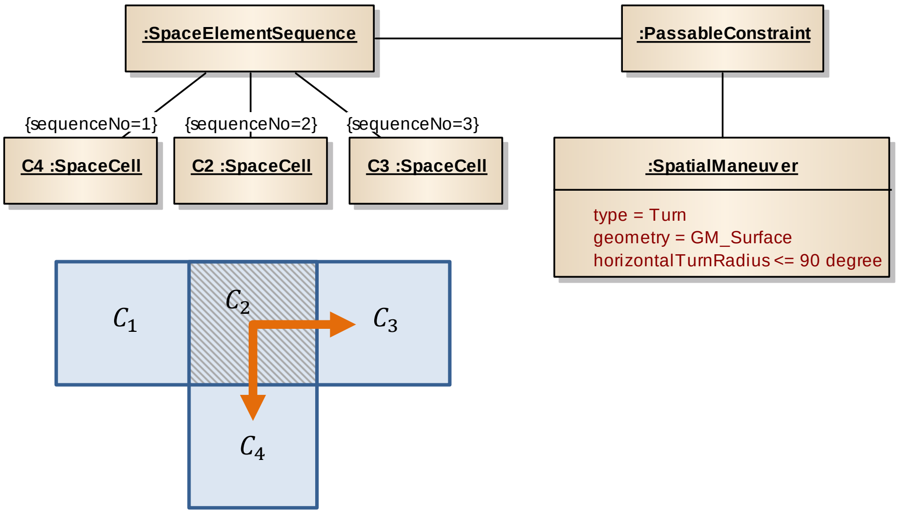


Figure 203: A spatial maneuver restriction associated with a T-shaped corridor.

Example 5.6. Assume the floor surface of a corridor is made of glass. Then only navigation users being capable of travelling along glass surfaces can pass the corridor. Further assume that a weight restriction is imposed on users entering the corridor. Both movement restrictions are captured by separate *PassableConstraint* instances in the following figure 204.

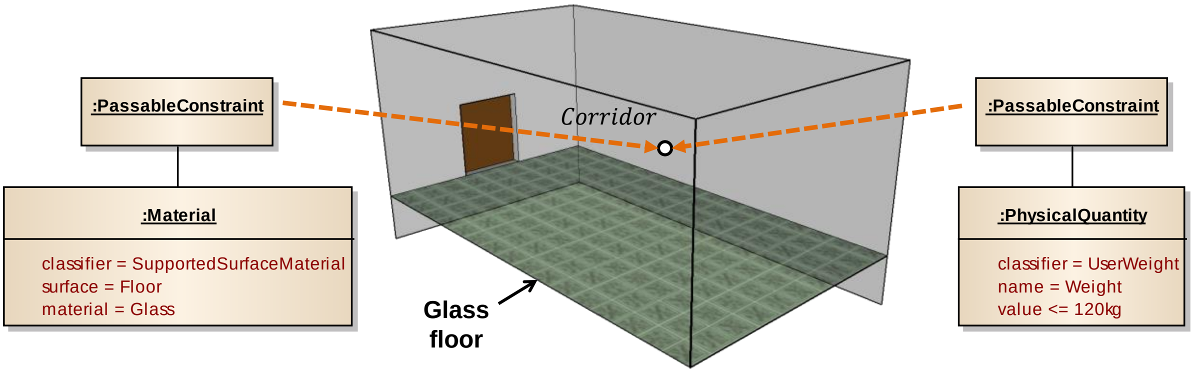


Figure 204: Example of a Material and a PhysicalQuantity condition denoting a glass floor surface of a corridor with a maximum user weight constraint.

The first *PassableConstraint* carries a *Material* condition that denotes the material of the floor surface. The condition is further classified as *"SupportedSurfaceMaterial"* in order to express that it is only satisfied in case the navigation user is capable of moving on the given surface material.[69] The weight restriction is modelled as *PhysicalQuantity* condition that enforces an upper limit of $120kg$ for the *"UserWeight"*. If either of the two constraints is not met, then the corridor is non-passable. Instead of having two instances of *PassableConstraint*, both conditions may also be linked to a single *PassableConstraint* since the guards of navigation constraints are *AND* combined per default.

Example 5.7. An example of a legal movement restriction is presented in figure 205. It shows an emergency door which may only be used in case of an evacuation or by security staff members who have the key to open the door. This *may* constraint is modelled as *PermissibleConstraint* with a nested *CombinedConstraintCondition* as shown in figure 205. The scenario restriction is stated as instance of *NavigationScenario* of type *Evacuation*. Either this condition is satisfied or the navigation user both belongs to the *UserGroup "SecurityStaff"* and meets the *AccessControl* condition in order to be allowed to pass the door. The latter requires the user to carry a key with the identifier *"Door_001"* as credential. The *PermissibleConstraint* implies that the door may not be traversed if the combined condition is not satisfied (e.g., in normal navigation conditions).

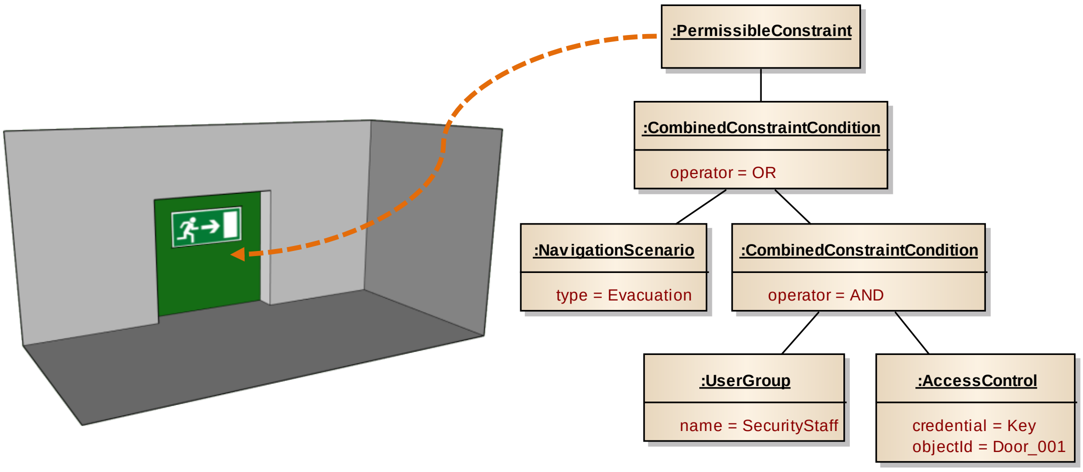


Figure 205: Combined legal constraint associated with an emergency door modelled as boundary cell.

The scenario condition is also feasible to restrict the usage of elevators in case of an evacuation situation. For instance, a *NonPermissibleConstraint* controlled by a *NavigationScenario* of type *Evacuation* could be modelled for a *SpaceElementGroup* that contains all space cells representing affected elevators.

Also the object state may be used as trigger to control access to space cells. Consider the example in figure 206 which shows two rooms being connected to a corridor. Assume that both rooms are equipped with fire detectors

[69] As stated in chapter 5.3.2, both the meaning and interpretation of values of the *classifier* attribute have to be predefined by the navigation system or a global authority.

whose states can be accessed online. In case a fire detector is active, only fire fighter forces shall be allowed to enter the room. This fact is modelled as *NonPermissibleConstraint*. The *ObjectState* condition is classified as *"isActive"* and provides the unique *id* of the object in question. Obviously, it must be possible for the navigation system to query the current state using this *id* value (e.g., from a database). The condition is satisfied if *"isActive"* evaluates to true as required by the *value* attribute. Moreover, the navigation user may not belong to the *UserGroup* *"FireFighter"* in order for the *NonPermissibleConstraint* to apply and thus to restrict movement for that user.

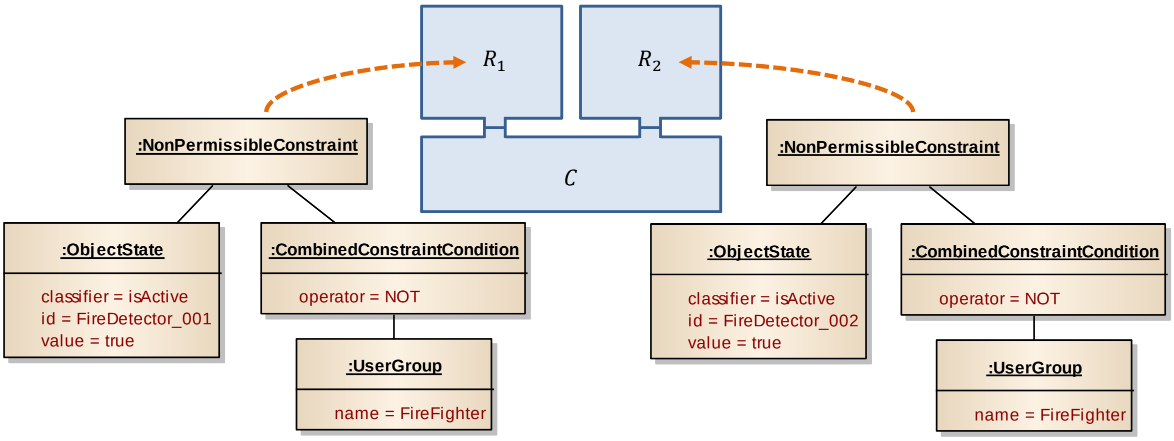


Figure 206: Example of restricting access to a room based on the state of an object.

The combination of *can* and *may* constraints is illustrated in the figure below for a window serving as emergency exit. The window *can* only be traversed if the navigation user is able to perform vertical steps with a minimum height difference of 80*cm* in order to enter the window ledge, and if the user fits through the window frame. However, the window *may* only be traversed in an evacuation scenario. According to algorithm 5.2, this user-independent legal condition overrules the physical capability of a navigation user to use the window.

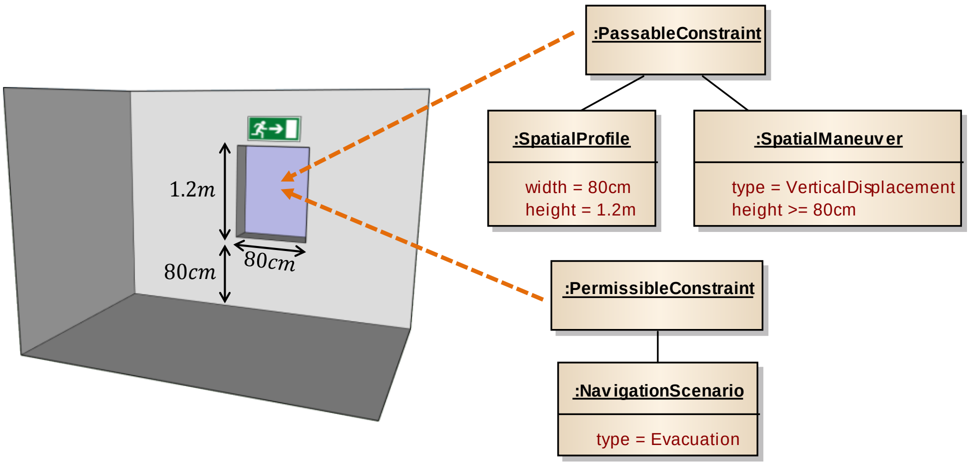


Figure 207: Physical and legal navigation constraints associated with an emergency window.

A possible user context of a navigation user *Alice* is shown in figure 208. *Alice* is a pedestrian and provides two spatial profiles. One profile denotes the spatial extent of *Alice* when walking upright, while the second profile captures the spatial extent in case *Alice* crouches. The different meaning of either profile is indicated through their *classifier* attributes. Obviously, the *SpatialProfile* condition on the window can only be satisfied if *Alice* crouches through the window. The user context of *Alice* also contains a *SpatialManeuver* stating that *Alice* can manage vertical displacements up to 1m height difference. Therefore, *Alice* physically can use the window in case of an emergency situation.

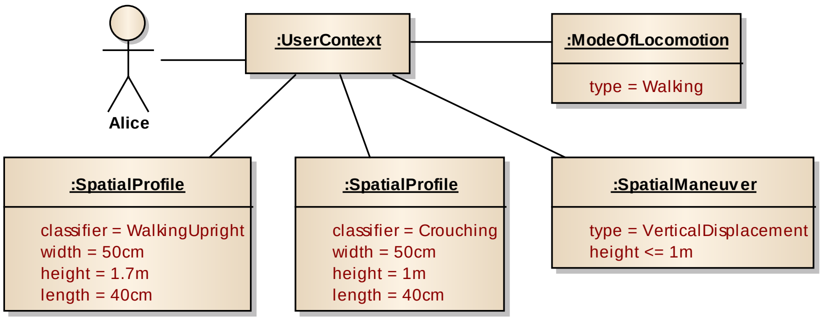


Figure 208: Example of a user context providing separate spatial profiles for walking upright and for crouching.

Example 5.8. The indoor scene sketched in figure 209 is based on an example for navigation constraints discussed in (Worboys 2011)).[70] Assume a room R is connected through doors to opposite sides of a U-shaped corridor. The corridor itself is mapped onto five space cells C_1, to C_5. The shortest path from C_1 to C_5 obviously leads through the room R. However, if we suppose R to be a non-public office, then navigation users without access permission should rather be routed via $C_2 - C_3 - C_4$. Thus, the path $C_1 - R - C_5$ renders a prohibited maneuver for these users. In order to model this constraint, the ordered set $\{C_1, R, C_5\}$ can be captured by a *SpaceElementSequence* being associated with a permission constraint. A *PermissibleConstraint* is used to mark the sequence as being traversable for staff members only. Note that the sequence is undirected in order to prohibit the movement from either side of the corridor.

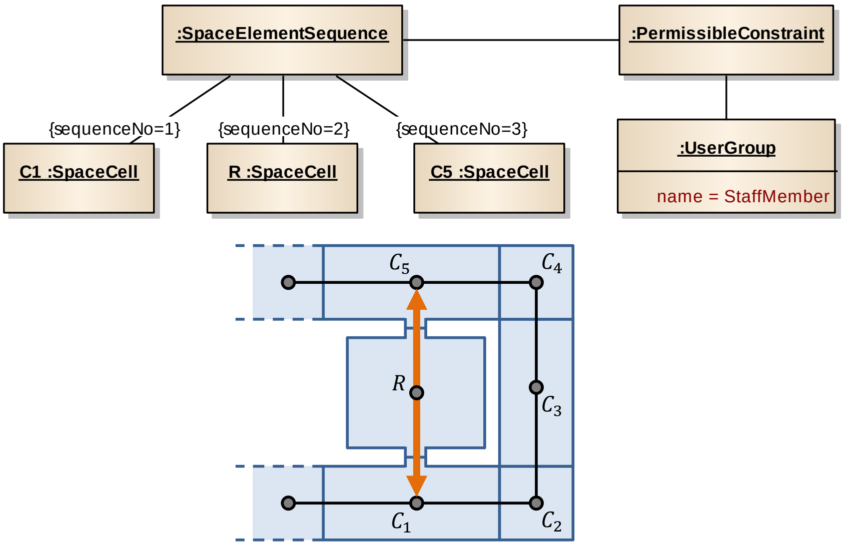


Figure 209: Example of a prohibited maneuver (after Worboys 2011).

Suppose that access to room R is further controlled by keypads at either door that require entering a password for the door to open. To capture this knowledge, two additional instances of *PermissibleConstraint* guarded by *AccessControl* conditions are associated with the doors or, more precisely, with the boundary cells shared by C_1 and R as well as C_5 and R as shown in the following figure 210. Note that in this simple example, the access constraint on each door applies when traversing the door in either direction. If the password is only required in one direction (e.g., when entering the room from the corridor), then the *PersmissibleConstraint* has to be related to a directed *SpaceElementSequence* instead.

[70] A conceptual model for the representation and evaluation of navigation constraints is however not presented by Worboys 2011.

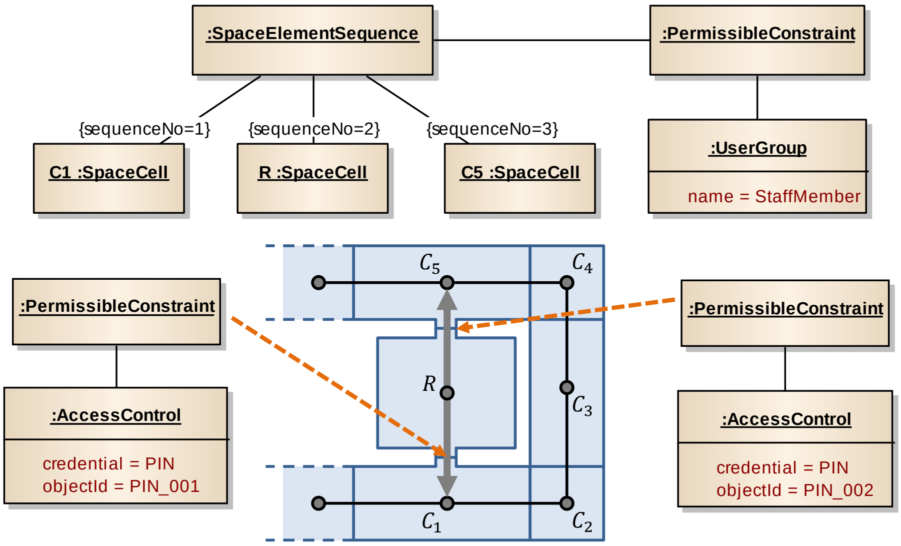


Figure 210: Augmenting the example from figure 209 with access restrictions on the doors.

In the above example, the values *"PIN_001"* and *"PIN_002"* of *objectId* are not meant to encode the required password itself but are rather used as identifier for the credential. For example, suppose the two navigation users *Alice* and *Bob* as shown below. Both *Alice* and *Bob* are staff members and thus are allowed to traverse room R. However, the user context of *Bob* does not contain instances of *AccessControl*. A navigation system thus cannot answer from the user context whether *Bob* knows the password to open the doors. Per default (cf. chapter 5.3.4), a constrained condition is assumed to be unsatisfied in case of incomplete knowledge and hence Bob may not be routed along room R. Alternatively, the system could ask *Bob* (e.g., via his end-user device) for this information. In contrast, the user context of *Alice* contains an *AccessControl* entity to denote that she knows the password credential identified by *"PIN_001"*. Thus, *Alice* at least can open one door of the room, whereas also her context lacks information about the second door.

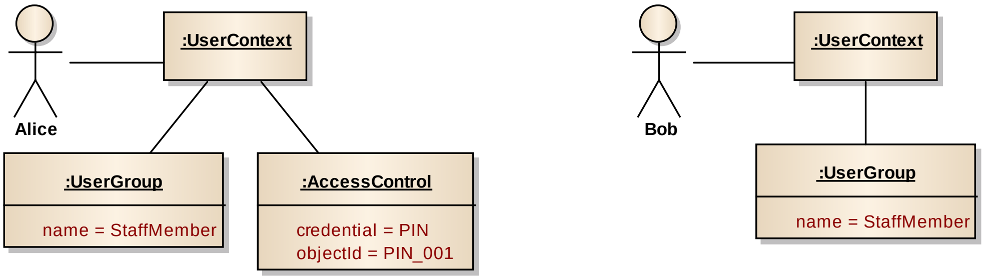


Figure 211: Possible user contexts to evaluate the access restrictions in figure 210.

Another example for a prohibited maneuver is presented in figure 212. The 2-dimensional setting consists of two space cells C_1 and C_2 representing two adjacent corridors, four room cells R_1 to R_4 being connect to C_2, and a staircase captured by the space cell S. In case of an evacuation, all persons shall be routed to the stairs but not to the opposite direction in order to control the flow of navigation. For this purpose, a *NonPermissibleConstraint* is expressed on the directed sequence $\{C_2, C_1\}$ and is associated with a *NavigationScenario* condition of type *Evacuation*. Due to the '+' sign used as value of the *direction* attribute of the *SpaceElementSequence*, the direction of the sequence is positive and hence from C_2 to C_1.

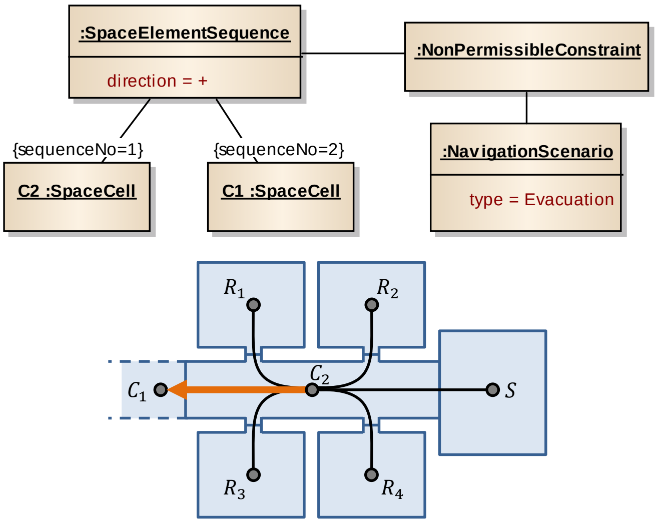


Figure 212: A prohibited maneuver denoting a one-way corridor.

Suppose the *NavigationScenario* condition is fulfilled. Then the prohibited maneuver restricts a route from R_2 via C_2 to C_1. In contrast, the movement from R_2 via C_2 to S is not affected by the *NonPermissibleConstraint* and hence allowed.

A directed maneuver may also be enforced physically. For example, assume a one-way ticket control gate inside a station. This gate can only be passed in one direction and hence physically obstructs movement into the other direction. A simplified scene is depicted in figure 213. The ticket gate T as well as the two places S_1 and S_2 separated by T are modelled as individual space cells. If the ticket gate only enables access from S_1 to S_2, then obviously the directed sequence $\{S_2, T, S_1\}$ needs to be marked as non-passable. Moreover, suppose that access to S_2 also requires the user to have a ticket. This can be modelled as *PermissibleConstraint* on T with an associated *AccessControl* condition.

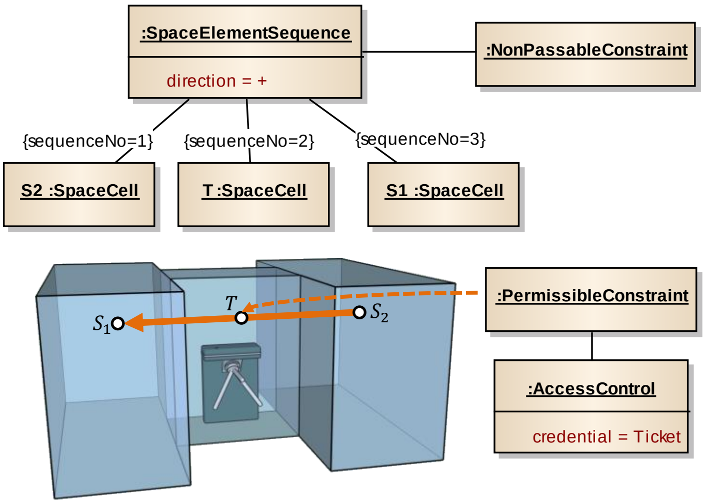


Figure 213: Example of a physically enforced directed maneuver at a one-way ticket control gate.

Example 5.9. Temporal conditions are used in the same way as physical and logical conditions. For example, assume a shop within a shopping mall that is open every weekday between $10am$ and $5pm$. Staff members are further allowed to enter the shop between $8am$ and $7pm$. Cleaning personnel may additionally enter the shop on Saturdays, whereas no time restrictions apply to users having the main key. In figure 214, the shop is represented by a single space cell and the illustrated restrictions are applied to the space cell representing the main entry door of the shop.

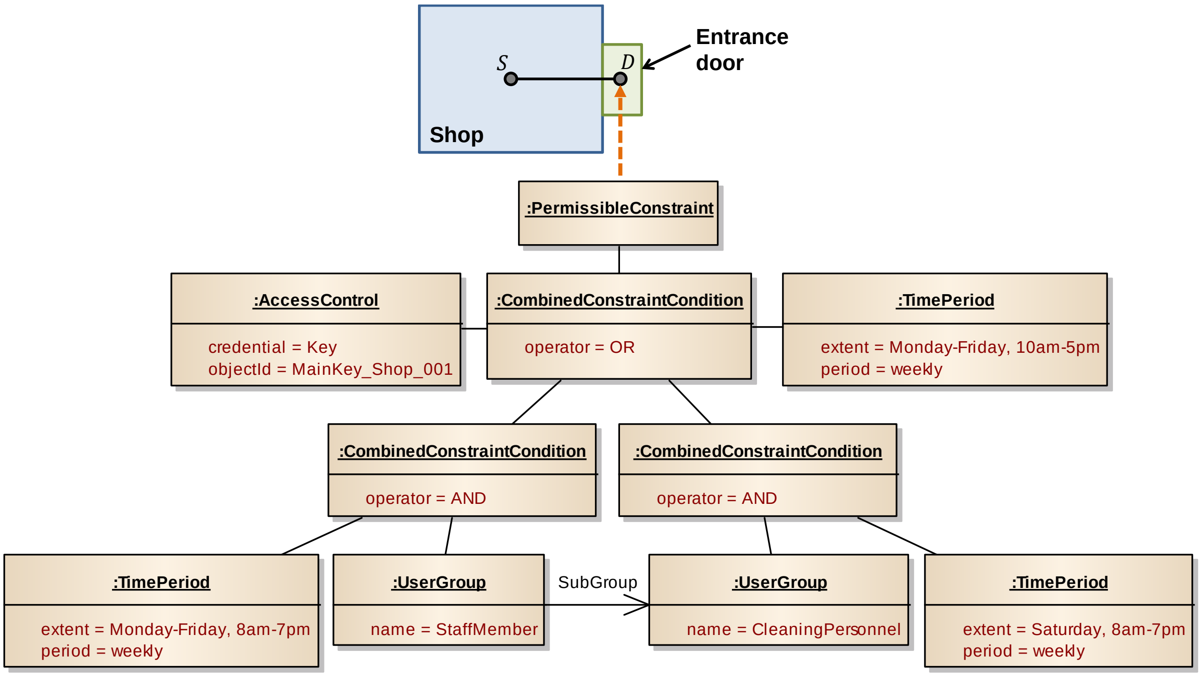


Figure 214: Combined legal and temporal conditions governing access to a shop.

The illustrated access permissions are mapped onto separate conditions of a single *PermissibleConstraint* which are combined by an *OR* operator. Thus, only one of them needs to be satisfied for a navigation user to be allowed to traverse the door. Note that an alternative representation using a separate *PermissibleConstraint* instance for each access permission would not yield the same result since multiple navigation constraints associated with a single space entity are combined in a logical conjunction per definition (cf. algorithm 5.2). Also note that the *UserGroup* named *"CleaningPersonnel"* is given as subgroup of the *UserGroup "StaffMember"*. Thus, the *TimePeriod* condition being effective for staff members also holds for members of the cleaning personnel, and only the additional access permission on Saturdays is explicitly stated for the latter user group.

Example 5.10. Figure 215 shows a simplified view of a security checkpoint at an airport. Passengers are required to move from space cell S_1 to S_3, whereas the reverse direction from S_3 to S_1 is prohibited. The space cell S_2 represents a metal detector at which the passengers are screened. A personal search may follow at space cell S_3. Moreover, a passenger shall only be routed to S_1 between $4am$ and $23pm$ and if the passenger carries a boarding pass and has checked in the luggage. Members of the security staff or the border control forces are allowed to traverse the space cells S_1, S_2, and S_3 in either direction and may additionally access space cell S_4 in order to control the security check. The set of navigation constraints expressing this scenario is depicted below.

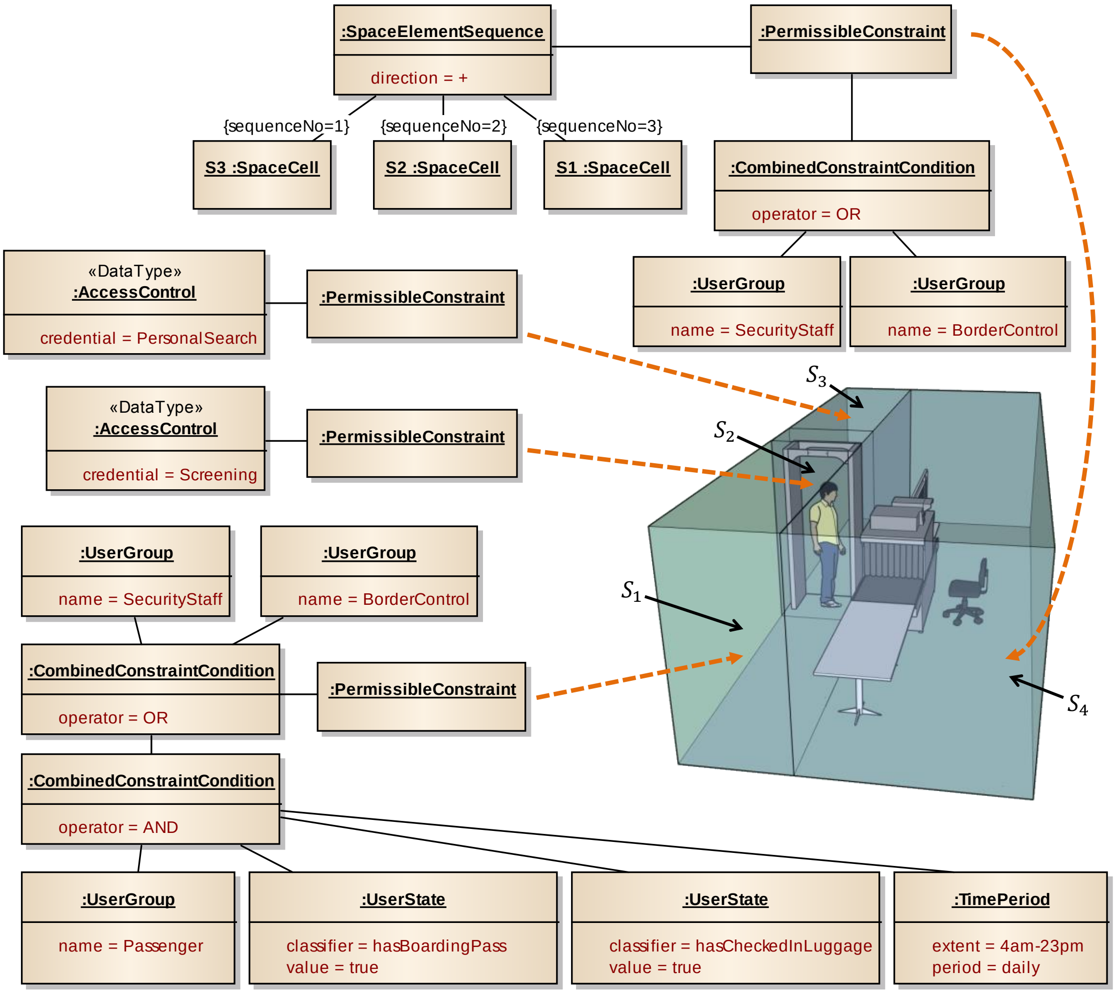


Figure 215: Possible navigation constraints for a security checkpoint at an airport.

The *PermissibleConstraint* associated with the directed sequence $\{S_3, S_2, S_1\}$ allows this path only for members of the *UserGroup "SecurityStaff"* or *"BorderControl"*, and thus implicitly renders a prohibited maneuver for all other navigation users. Likewise, the space cell S_4 is only accessible for members of either group. *UserState* conditions are used to control the access to space cell S_1 for passengers. The classifiers *"hasBoardingPass"* and *"hasCheckedInLuggage"* need to be understood by the navigation system, and both conditions have to be true for a given navigation user in order for S_1 to be included in a route. The required information about the user state may, for example, be queried from the passenger. Note that the space cells S_2 and S_3 are target of a *PermissibleConstraint* being associated with an *AccessControl* condition that models the screening and the personal search at the checkpoint. Whether a passenger is rejected can however only be decided at the time she is screened or searched. Thus, both constraints shall not prevent the navigation system from routing the passenger to the checkpoint but instead need to be non-effective in path searches. As discussed above, the *isSatisfied* operation of both *AccessControl* conditions should therefore return true per default. Nevertheless, a navigation system can use the represented knowledge about the *AccessControl* in order to generate appropriate route guidance information.

Navigation constraints on separate space layers. The following two examples demonstrate the modelling and evaluation of both simple and complex navigation constraints on separate space layers. In general, the explicit representation of navigation constraints is to be seen complementary to the modelling of environmental and user-related contextual information by means of (sub)space layers as discussed in the previous chapters of this thesis. Whereas space layers decompose the indoor space according to multiple and different notions of space and partitioning schemas, navigation constraints provide meta information about the possibility to move along the space cells and boundary cells on a given space layer.

Example 5.11. The first scenario in figure 216 shows a 2-dimensional space layer complex that contains a topographic space layer L_{Topo} and a security space layer L_{Sec}. The corresponding multilayered graph is depicted on the right of figure 216. The security space layer captures a single security zone Sec whose spatial extent is not aligned with the architectural layout of the built-up space. Assume that only navigation users having a high security status are allowed to traverse the security zone. This fact is expressed as *UserState* condition being associated with Sec through a *PermissibleConstraint*. A second constraint is modelled for the room R_1 on the topographic space layer in order to restrict the access to given office hours.

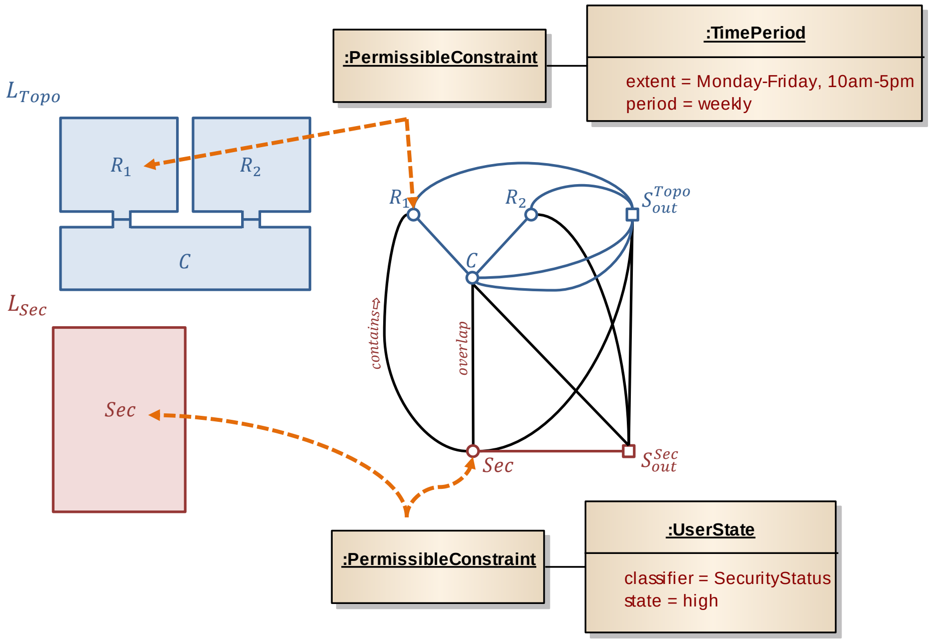


Figure 216: Navigation constraints associated with space cells on different space layers.

The security zone obviously affects the movement of navigation users through the topographic space. As can be deduced from the multilayered graph, the space cell Sec spatially contains R_1. Thus, the space described by R_1 is in fact addressed by both constraints. According to the rule base governing the evaluation of multiple navigation constraints associated with a single space entity (cf. algorithm 5.2), the *isTraversable* operation of either *PermissibleConstraint* instance therefore needs to evaluate to true in order for a navigation user to be allowed to enter R_1. This also conforms to an intuitive understanding since the security zone (and thus all spaces inside this zone) may only be accessed by authorized users. If both constraints would be *OR* combined instead, then the simple fact that the user travels within the office hours would grant access to R_1 and thus overrule the security constraint. It immediately follows from this example that a navigation constraint modelled for a given space cell also holds for all its subspace cells on further space layers. Put differently, a path finding algorithm not only has to evaluate the navigation constraints related to the space cell itself but also those constraints being associated with its (transitive) superspace cells. The superspace cells can be easily queried from the multilayered graph, and the $GetConstraints$ function used in algorithm 5.2 has to be implemented in this sense.

Regarding the corridor cell C, the multilayered graph only contains the information that C is partially within the security zone. Due to $C \nleq Sec$, the security constraint obviously does not involve the entire corridor according to the above rule. In order to precisely reveal which parts of C are affected by the constraint, the views on indoor space provided by both L_{Topo} and L_{Sec} need to be integrated using the merge operation $L_{Topo} \oplus L_{Sec}$ as defined in chapter 3.5. In terms of the mathematical model of the MLSEM, *simple* navigation constraints are considered as semantic attributes of a space cell and thus participate in the set $A(S)$ of a space cell S (cf. definition 3.6). The same holds true for navigation constraints modelled for boundary cells. This notion is important since it ensures the consideration of navigation constraints within the operations of the developed space layer algebra. Precisely, in the context of the merge operation, all semantic attributes are carried from the input space layers to the merged space layer as union of the attribute sets of overlapping space cells (cf. algorithm 3.67). The result of $L_{Topo} \oplus L_{Sec}$ is presented in figure 217.

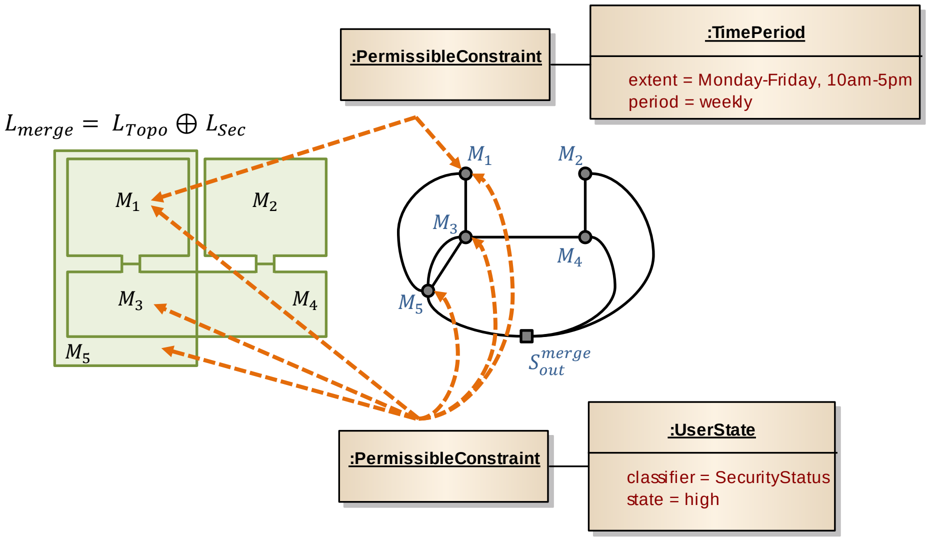


Figure 217: Applying the merge operation $L_{Topo} \oplus L_{Sec}$ to the example from figure 216.

For example, consider the space cell M_1 in the resulting space layer. Its attribute set $A(M_1)$ results from the union $A(R_1) \cup A(Sec)$ and thus contains both *PermissibleConstraint* instances. Again, and conformant with the result discussed above, both constraints need to be evaluated in a logical conjunction since they render a combined constraint on M_1 (cf. algorithm 5.2). Thus, M_1 can only be traversed within the specified office hours and if the user has the required security status. The space cells M_3 and M_5 only receive the security access restriction from the space cell Sec, with M_3 consequently denoting the part of the corridor C affected by the security zone. It is important to note that the presented consideration of simple navigation constraints within the operations of the space layer algebra is consistent with the usage of the static Boolean flag $isNavigable$ in the examples of the previous chapters of this thesis (e.g., example 3.69 in chapter 3.5.1). Consequently, in all those examples, this flag can be replaced with a corresponding instance of *NavigationConstraint*.

Example 5.12. Similar considerations apply to *complex* navigation constraints. Since a complex constraint affects a set of space cells which themselves may be decomposed into subspace cells on separate space layers, the constraint also has to be evaluated in the context of those subspace cells. With respect to the operations of the space layer algebra, a difference to simple navigation constraints results from the fact that a complex constraint is not to be seen as semantic attribute of a single space cell or boundary cell. In contrast, complex navigation constraints can be understood as semantic information about their associated *SpaceElementGroup* or *SpaceElementSequence*. In chapter 4.4.1.4 it has been demonstrated how groups and sequences are to be mapped onto new instances of *SpaceElementGroup* respectively *SpaceElementSequence* that address the space cells on the space layer resulting from a merge, difference, or intersection operation. A complex navigation constraint simply has to be carried to this new group or sequence in order to be available on the resulting space layer.

For example, reconsider the scenario illustrated in figure 212. Assume the topographic space layer L_{Topo} is accompanied by a subspace layer L_{Sub} which decomposes the corridor cell C_2 into three subspace cells Sub_1 to Sub_3 as shown below. In addition to the prohibited maneuver that restricts the movement from C_2 to C_1 on L_{Topo}, the space cell Sub_2 is assumed to represent a step within the corridor that can only be traversed in case the navigation user can negotiate a minimum height difference of $30cm$.

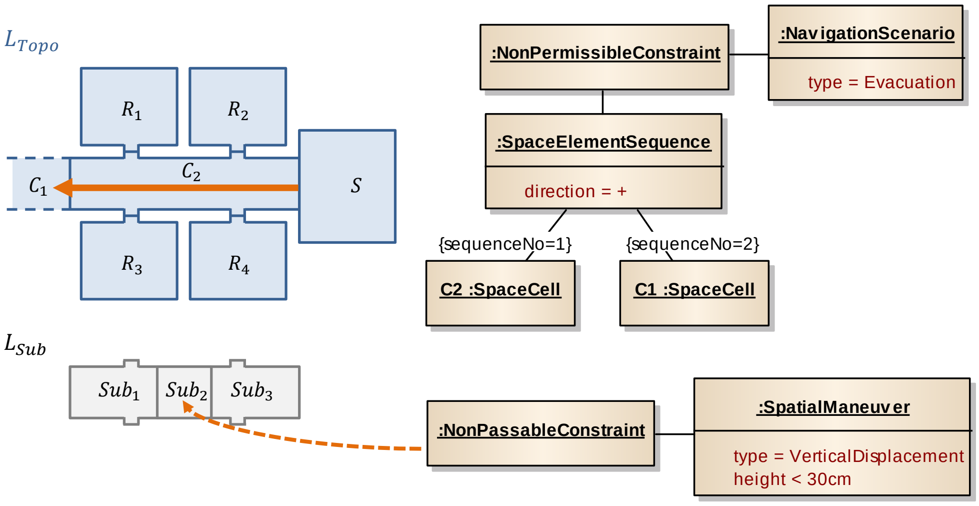


Figure 218: Denoting a prohibited maneuver on a topographic space layer and a spatial maneuver restriction on a subspace layer.

In order to route a wheelchair user through this indoor setting, both space layers need to be selected and might be integrated into a single view using the merge operation $L_{Topo} \oplus L_{Sub}$. The resulting space layer L_{merge} is depicted in figure 219. The directed sequence $\{C_2, C_1\}$ on L_{Topo} is translated into a new instance of *SpaceElementSequence* on L_{merge} generated by the ordered set $\{M_8, M_7, M_6, M_5\}$. Note that the space cells M_6 to M_8 share the same value for their *sequenceNo* qualifier since all of them are spatially contained in C_2. The complex *NonPermissibleConstraint* on the original sequence is transferred to this new sequence in order to not lose the information about the prohibited maneuver constraint. Likewise, the *NonPassableConstraint* associated with the space cell Sub_2 is carried to the equivalent space cell M_7 on L_{merge} according to the rules for simple navigation constraints presented above. Thus, all navigation constraints are preserved under the merge operation.

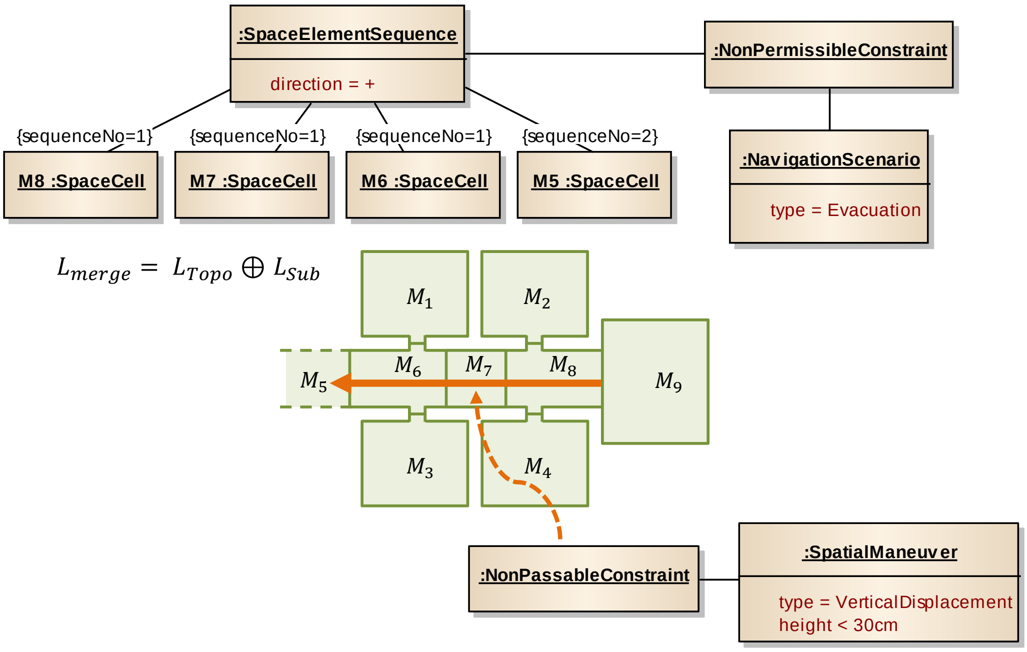


Figure 219: Applying the merge operation $L_{Topo} \oplus L_{Sub}$ to the example from figure 218.

An obvious advantage of the modelling of navigation constraints on separate space layers is that the constraints on a space layer that is not part of the user-dependent layer selection need not be considered in path queries. For example, assume that the space layer L_{Sub} from figure 218 is not part of the selection for a pedestrian user. Then the constraint on Sub_2 has not to be evaluated when routing this user, which might speed up path searches. However, this consequence also must be considered when modelling the indoor space and its navigation constraints.

5.5 Implicit Knowledge about Navigation Constraints

The proposed MLSEM constraint model aims at an *explicit* representation of knowledge about navigation constraints. Therefore, constraints are represented by their own conceptual entities which are associated with the elements of the indoor space in order to provide additional information about movement restrictions.

However, movement restrictions are often also available *implicitly* from the underlying indoor space model. This especially holds for geometric constraints that affect the physical traversability of indoor spaces. For example, whether a navigation user fits through a door or can travel along a narrow passage can generally be answered from the geometric shape and extent of topographic space entities on the one hand, and a suitable geometric description of the user's extent on the other hand. In the field of robot motion and path planning, the notion of configuration space (C-space) has been proposed to simplify the evaluation of physical traversability in 2-dimensional and 3-dimensional settings (cf. Lozano-Perez 1983). The general idea of C-space is to reduce the robot to a point representation while at the same time growing obstacles in the environment such as walls or furniture by the generalized shape of the robot, its orientation, and its possible configurations (e.g., translational or rotational motions of the robot's body and end effectors). Physical traversability can then be determined by continuously moving the point-based abstraction of the robot (or more generally of a navigation user) through the resulting free C-space. Thus, this free C-space implicitly denotes the physically unconstrained navigation space. An additional explicit representation of navigation constraints (e.g., maneuver capabilities of the navigation user) is thus not required.

The calculation of the free C-space is computationally expensive and especially demanding for high-detailed, 3-dimensional space models. Moreover, the C-space is necessarily tailored to a specific (type of) navigation user and thus, for example, cannot account for multiple and different types of locomotion simultaneously. Most grid-based approaches to indoor space modelling as discussed in chapter 2.2.1 address the computational efforts by discretizing the space into regular grid cells and evaluating their physical accessibility with respect to a parametric geometric description of the navigation user (e.g., Bandi & Thalmann 1998) or a rigid volumetric abstraction (e.g., Yuan & Schneider 2011). In addition to spatial facts, also semantic information available from the indoor space model can be used to implicitly deduce simple navigation constraints. For instance, if space entities are identified as stairs then a navigation system can conclude from common knowledge and experience that these entities are typically non-traversable for wheelchair users and driving mobile robots. Likewise, entities classified as walls can be assumed to render obstacles. All hybrid modelling approaches presented in chapter 2.2 semantically classify the spatial and structural entities of indoor space and hence enable implicit knowledge about navigation constraints to be retrieved. In most cases, this will however not suffice to derive complex or combined constraints that also include logical or temporal conditions.

The scope of the MLSEM constraint model is neither to replace existing approaches to the implicit derivation of navigation constraints nor to render them unnecessary. In contrast, the MLSEM fully supports the representation of implicit knowledge about navigation constraints by means of (sub)space layers and semantic information. For example, and as discussed in chapter 3.4, consider a subspace layer of the topographic space that only contains areas which are navigable for a given mode of locomotion. Then the subspace cells implicitly represent the physical traversability of the topographic space for that mode of locomotion. An additional explicit modelling of *can* constraints can thus be omitted since the subspace layer itself denotes the physically unconstrained space for a corresponding navigation user. Likewise, the subspace layer may reflect those spaces which are non-navigable. In this case, each subspace cell implicitly renders a physical movement restriction (i.e., a *cannot* constraint). Again, there is no need for additional explicit physical navigation constraints since the navigable space can be deduced from the topographic superspace layer using appropriate operations of the space layer algebra. In both cases, the navigable or non-navigable areas may be identified using existing implicit approaches such as C-space methods.

In a recent work that parallels this thesis, (Khan & Kolbe 2012) present an approach for the derivation of the physically unconstrained navigation space which uses the subspacing concept of the MLSEM and thus nicely integrates with the MLSEM framework. The authors define and formalize the physical movement requirements for different types of locomotion (walking, driving, and flying) by identifying common physical constraints such as the average width, length, and height of a wheelchair. The approach goes beyond existing methods in that further restrictions such as topological constraints (e.g., pedestrians typically walk on the floor surface whereas flying objects should always be above the floor surface), weight restrictions, maneuver restrictions (e.g., the maximum height of steps that can be taken), or maximum speed restrictions are considered. Similar to the idea of C-space in the field of robotics, these requirements are then used to compute those areas in topographic space in which a navigation user of that locomotion type can freely move. The result of this analysis is represented as a separate topographic subspace layer. Put differently, the physical traversability of a given mode of locomotion is used as partitioning schema in order to derive a separate view on indoor space. When routing a navigation user through the environment, only the appropriate subspace layer matching the user's mode of locomotion has to be selected while neglecting the topographic superspace layer. Any path on that subspace layer can then be assumed to be traversable. This conforms to the default assumption of the proposed MLSEM constraint model according to which space cells and boundary cells not being explicitly addressed by navigation constraints are deemed navigable (cf. chapter 5.3.4). The work of (Khan & Kolbe 2012) however merely addresses physical navigation constraints, whereas logical and temporal movement restrictions as well as complex or combined constraints are not addressed by the authors.

The implicit representation of navigation constraints simplifies path findings as the traversability of spaces is determined a priori and thus directly available from the space representation. A main drawback is rendered by the fact that this precomputation is expensive and always requires a 2-dimensional or 3-dimensional indoor space model. Although most works focus on physical constraints, this restriction is not compulsory. For example, separate subspace layers could likewise be introduced for denoting those topographic areas that are only accessible by specific user groups such as staff members or within a given time frame. However, this modelling approach may quickly result in a large number of space layers depending on the types and combinations of navigation constraints to be expressed. In contrast, a benefit of the explicit modelling of navigation constraints is that it allows for expressing knowledge about constraints independent of a specific representation of the indoor space. For example, a physical navigation constraint such as that a door can only be traversed if the width of the user is less than $1m$ can obviously be expressed on the door entity and evaluated against a user profile without the need for an accompanying 2-dimensional or 3-dimensional geometric description of the door. Hence, a simple graph-based representation of the indoor environment enriched with a set of explicit navigation constraints applied to the graph elements is already sufficient to realize constraint-aware path searches for different navigation users. If a geometric description of the indoor space is available though, it can be additionally used to evaluate the navigation constraints possibly based on existing methods and algorithms from implicit approaches. Moreover, it helps to populate the constraint model, for example, by extracting the door width from the geometry in order to formulate a corresponding navigation constraint. Another advantage of the explicit representation of navigation constraints is the flexibility in supporting multiple and different user contexts. For instance, the work of (Khan & Kolbe 2012) is based on generalized and static assumptions about the physical movement requirements of different types of locomotion such as that pedestrians typically can negotiate a vertical height difference of $1m$. However, a given pedestrian may deviate from these assumptions (e.g., elderly pedestrians). In this case it is preferable to model the requirements for performing a spatial maneuver directly on the space entity as proposed in this chapter, and to dynamically evaluate this maneuver restriction against a given user context during path searches. Since both alternatives are valid and fully supported by the MLSEM, a navigation system has to choose a proper balance between the implicit and explicit representation of navigation constraints.

Chapter 6
Relation to Existing Building Modelling Standards

Geospatial information about the interior built environment becomes increasingly available in the context of urban and building information modelling. Although building models from these fields aim at representing the building topography as designed or observed and hence are not primarily intended to meet the challenges and requirements to indoor space models for indoor navigation, it has been shown in chapter 2.3 that they provide a rich and valuable source of spatio-semantic information about the built-up space and the architectural entities therein. Most approaches to the modelling of indoor space for indoor navigation as discussed in chapter 2.2 implicitly presuppose the availability of such building data in two or three dimensions. However, the relation to building models from existing standards such as CityGML and IFC is seldom discussed.

The MLSEM framework has deliberately been defined in the previous chapters to be independent of the different approaches to building modelling. A topographic space layer can thus be populated without the need for existing building data. The spatial and semantic information required to describe the space cells making up the topographic indoor space may instead be acquired by initial measurements and observations. On the other hand, the MLSEM is also meant to be complementary to existing building modelling standards, and in this neither aims at replacing these standards nor at duplicating their concepts for the representation of the interior built environment. In contrast, if building data is available according to one of these standardized models, it may serve as input to populate a topographic space view within the MLSEM framework.

In this chapter, the mapping from selected building modelling standards to the MLSEM is investigated. Focus is put on the differences in the geometric-topological description of the architectural entities within these standards and on how these differences affect the resulting primal and dual space representations of the resulting topographic space layer. The different spatial modelling paradigms mainly arise from different model scopes and the way the spatial characteristics of features are acquired in different domains such as GIS and BIM (cf. Kolbe & Plümer 2004, Nagel et al. 2009). As illustrated in chapter 2.3, the geometry of features in the field of GIS is typically derived from surveying and photogrammetric extraction methods. As a consequence, features are spatially represented by their visible respectively observable surfaces which are accumulated to form more complex structures such as closed volumes. BIM models, on the other hand, aim at representing architectural designs and constructions, and thus apply a generative modelling approach in order to describe how facilities are composed of structural building elements that are commonly represented as volumetric primitives. From this, a general classification of building models into *surface-based models* and *volumetric elements models* can be drawn. In academic approaches, the model complexity is often further reduced by employing surface-based models which reflect the entire building as *paper model*, i.e. representations of walls are paper-thin and thus neglect the wall thickness. Both surface-based and volumetric elements models are understood to be inherently 3-dimensional in the context of this chapter. A further important class of building models are 2-dimensional *floor plans* providing a projected view on the architectural and physical building features from above on a per floor basis.

The discussion in this chapter is structured along these four categories of building models. In addition to the spatial characteristics, the conceptual elements from the building modelling standards are identified with respective elements from the conceptual data model of the MLSEM. On the one hand, this allows for the transition of semantic information in the mapping process. On the other hand, it demonstrates that the developed indoor space model is to be seen complementary to existing building models and that it provides generic conceptual interfaces to connect such models. The mapping results for different building modelling standards are compared and proposals for minimizing their differences are presented.

6.1 Surface-Based Models

The mapping of *surface-based models* onto the MLSEM is exemplified along the international OGC standard CityGML which has been comprehensively introduced in chapter 2.3.2. In figure 220, a 3-dimensional CityGML LOD4 representation of the simple indoor scene introduced in chapter 3.1.2.3 is shown. In addition to the two rooms being connected to a corridor through doors, a column and the exterior building shell have been added to the model. Moreover, a window and a door exemplify openings connecting the interior environment to the surrounding outer space. The spatial entities are labelled with the names of the conceptual entities from the CityGML building model (cf. UML diagram in figure 32) which are used in representing this scene. In figure 220b, a top view of the model is presented.

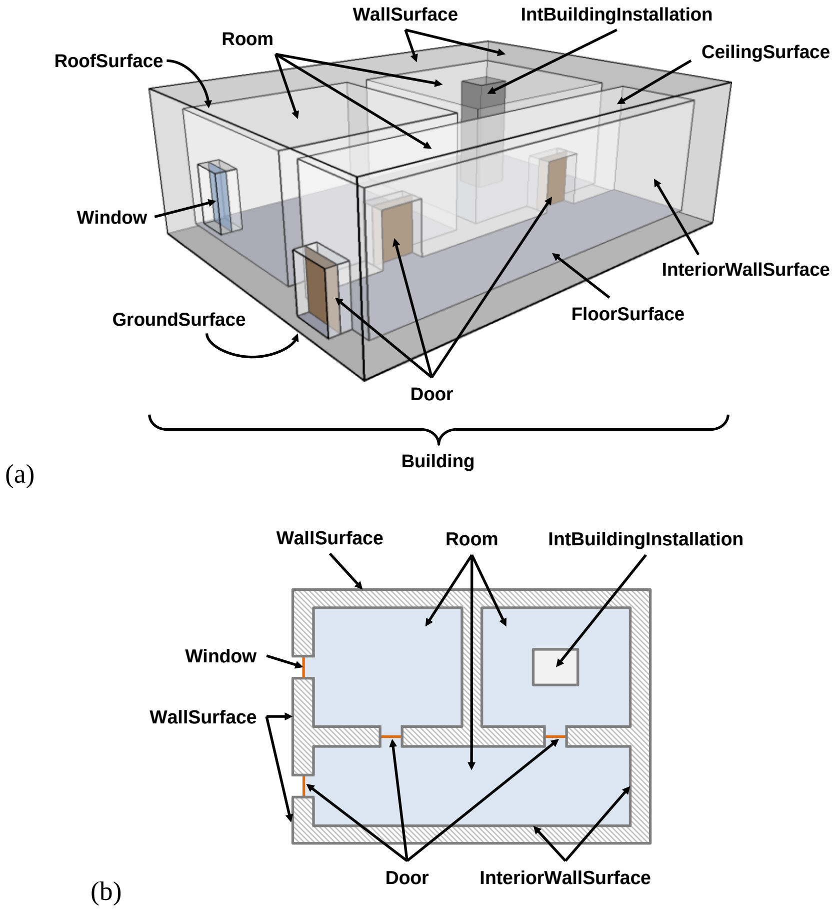


Figure 220: Example indoor scene as 3*-dimensional CityGML LOD4 model (a) and as* 2*-dimensional top view (b).*

In CityGML, the rooms and the corridor are semantically captured by instances of the conceptual feature type *Room* which represents a portion of free space inside a facility and spatially is described as bounded volume. According to the surface-based modelling approach, the *Room* volume is enclosed by the surfaces which are observable from inside the *Room* and which hence represent parts of the embracing structural elements (e.g., walls and slabs). The boundary surfaces are semantically classified into *InteriorWallSurface*, *CeilingSurface*, and *FloorSurface*. Note that the thickness of walls is maintained in the CityGML model. Therefore, the *Room* geometries only touch at those surfaces which represent the doors between the rooms and the corridors. In a similar way, the volume of the *Building* feature itself is bounded by the exterior shell which is observable from outside the building. The surfaces of the exterior shell are represented by the feature types *WallSurface*, *RoofSurface*, and *GroundSurface*. Again, the exterior shell touches the interior *Room* geometries only at surface-based openings such as windows and doors. Finally, the column is modelled as *IntBuildingInstallation* feature whose tangible surfaces are mapped onto *InteriorWallSurface* instances.

The space enclosed by the exterior and interior boundary surfaces of the building represents the non-navigable space occupied by walls and slabs (depicted as hatched area in figure 220b). This *wall space* is typically non-observable and hence is not further partitioned along the wall and slab elements. It follows that neither the wall space nor the structural building components occupying it are explicitly modelled in CityGML. For example, consider the wall separating the two rooms. It is only implicitly represented through the two *InteriorWallSurface* instances on the boundary of each room. However, the fact that both *InteriorWallSurface* features belong to the same wall element is not available from the semantic CityGML entities, but rather has to be derived from their spatial configuration. Moreover, a single *InteriorWallSurface* feature may even be spatially described by multiple geometric surfaces belonging to different wall elements (e.g., all wall surfaces of a single room may be semantically captured by a single instance of *InteriorWallSurface*). Thus, CityGML does not enforce a one-to-one mapping between the semantic surface-based features and the structural components of the building skeleton (cf. Nagel et al. 2009).

The mapping of the CityGML model to a topographic space layer of the MLSEM is shown below. The resulting space representation mainly coincides with the original example in chapter 3.1.2.3. The *Room* entities are simply mapped onto *SpaceCell* features, and the boundary surfaces of the rooms and the corridor are translated into *BoundaryCell* instances. The boundary surfaces are thus explicitly represented as dual edges in the intra-layer graph and their semantics (e.g., material information) is available for path searches. Alternatively, surfaces being topologically connected can be captured by a single *BoundaryCell* in order to reduce the number of dual edges. The wall space can likewise be described by a single *SpaceCell*, and the boundary surfaces of the exterior building shell are also mapped onto *BoundaryCell* features. The *Door* and *Window* surfaces within the model are carried to separate instances of *BoundaryCell*. A final *SpaceCell* is used to translate the *IntBuildingInstallation* representing the column. The resulting intra-layer graph depicted on the right of figure 221 assumes a minimum number of boundary cells. Since the space cell representing the wall space cannot be traversed, it might receive an unconditioned *NonPassableConstraint* automatically. The same holds for the space cell resulting from the *IntBuildingInstallation*.

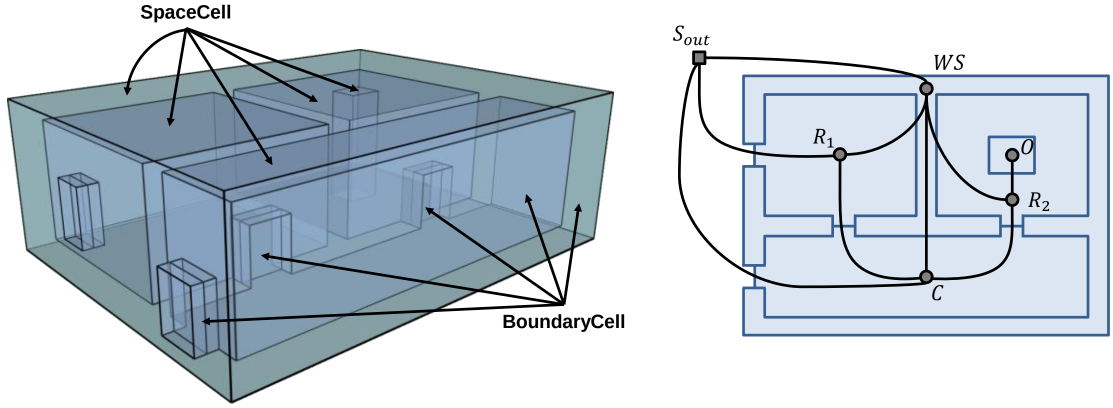


Figure 221: Corresponding topographic space layer of the MLSEM for the CityGML model from figure 220. The intra-layer graph on the right assumes a minimal CW decomposition.

In general, the primal space geometry of *SpaceCell* and *BoundaryCell* features can be directly taken or derived from the corresponding CityGML features since both CityGML and the MLSEM employ the Boundary Representation (B-Rep) scheme and are based on the ISO 19107:2003 Spatial Schema. However, there are also exceptions from this assumption which result from the fact that space cells on the same topographic space layer are enforced to be mutually non-overlapping, whereas volumetric shapes in a CityGML model are allowed to spatially overlap and to permeate each other. For example, consider the room and the column in the above example. In CityGML, the geometry of the room is given as single-shell manifold solid being a homeomorphic image of $\bar{\mathbb{B}}^3$ and hence covers the space occupied by the column.[71] In contrast, for the MLSEM, the primal space geometry of the room cell has to be given as 1-holed toroid whose trough hole is filled by the column cell being homeomorphic to $\bar{\mathbb{B}}^3$ (cf. chapter 3.1.3.2). Put differently, a regularized Boolean subtraction of the volume of the column from the room geometry in CityGML is required in order to retrieve the primal space geometry of the room cell for the MLSEM. The same holds for the building volume enclosed by the exterior building shell. The spatial representation of the CityGML *Building* feature is also homeomorphic to $\bar{\mathbb{B}}^3$ in this example. However, the room geometries are not modelled as interior voids of this volume in CityGML but rather as separate solids permeating the building volume. In the context of the MLSEM, the solid geometries of the rooms therefore also need to be subtracted from the building volume, which results in the wall space as shown in figure 221.

[71] Note that the boundary surfaces of a *Room* feature are not enforced to render a closed surface (*GM_CompositeSurface*). In contrast, they may also be given as a *GM_MultiSurface* which is an aggregate geometry and thus not necessarily ensures a bounded volume. This flexibility in the spatial representation is also applied to all other CityGML feature types. It therefore might require additional algorithms from computational geometry in order to retrieve a solid geometry.

An alternative mapping of the CityGML model is depicted in figure 222. In this example, the building volume is carried to a space cell on a separate topographic space layer L_{super}, whereas the rooms, the corridor, and the column are translated to space cells on the subspace layer L_{sub} as illustrated above. The resulting multilayered graph is shown on the right of figure 222. In contrast to the first alternative, the space cell on L_{super} now shares the same geometry as the *Building* feature in the CityGML model. Based on the spatial inclusion relationships between the space cells on L_{super} and L_{sub}, the multilayered graph encodes the nested hierarchical structure of the built environment (cf. example 3.61).[72] The non-navigable wall space is not explicitly mapped by a space cell anymore. If required, it can be deduced, for example, from applying the merge operation $L_{super} \oplus L_{sub}$. Note that the dual nodes of the room cell R_1 and the corridor cell C are linked to the dual node of the outer space S_{out}^{sub} via two dual edges since the window and the door are again represented by individual boundary cells.

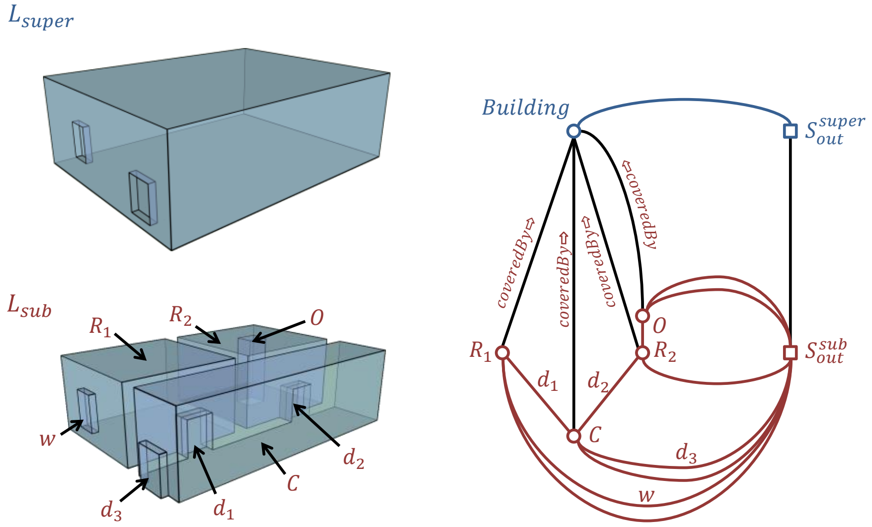


Figure 222: Alternative mapping of the CityGML model capturing the hierarchical structure of the built space.

CityGML does not impose the spatial representation of *Door* and *Window* features as single surfaces. In contrast, doors and windows may likewise be described by solid objects. Figure 223 shows the impact of this modelling approach on the space layer L_{sub}. The *Door* and *Window* features are mapped onto instances of *SpaceCell* rather than *BoundaryCell*, and thus are available as separate dual nodes in the intra-layer graph.

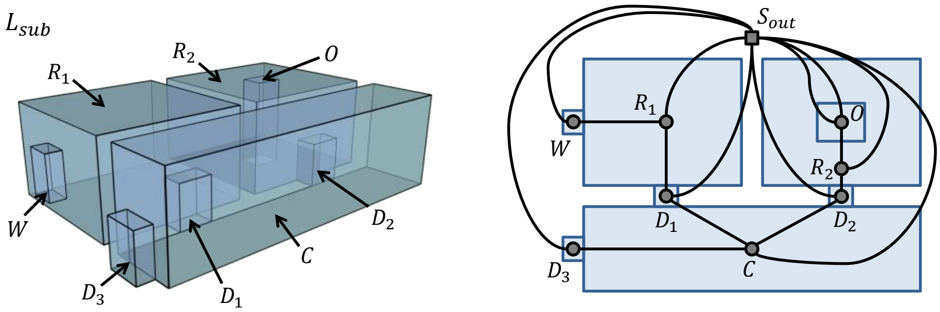


Figure 223: Impact of modelling CityGML doors and windows as solid objects.

The linkage between the conceptual model elements of CityGML and the MLSEM is summarized in figure 224 based on excerpts of their UML diagrams. The identification of semantic concepts in either model also allows for carrying thematic attributes from the CityGML features to their counterparts in the MLSEM. For example, symbolic information of *Room* features such as room names or numbers can be kept for the equivalent *SpaceCell* instances in the MLSEM (e.g., using generic attributes). Likewise, material information assigned to an *InteriorWallSurface* feature can be carried to the corresponding *BoundaryCell*. The rich semantics about the interior built

[72] Further hierarchical subspace layers (e.g., for representing the separate building floors) may be introduced in the same way. In general, CityGML lacks a specific concept for modeling floors. In order to overcome this issue, the current specification recommends the usage of *CityObjectGroup* features for aggregating all features on the same floor (cf. chapter 2.3.2). A *CityObjectGroup* can be assigned its own solid geometry and thus can be mapped onto a *SpaceCell* representation.

environment provided by a CityGML model is thus not lost in a model translation. Moreover, and due to the primal and dual space representation of space cells and boundary cells, the semantic information is directly available from the navigation graph (e.g., for path finding algorithms).

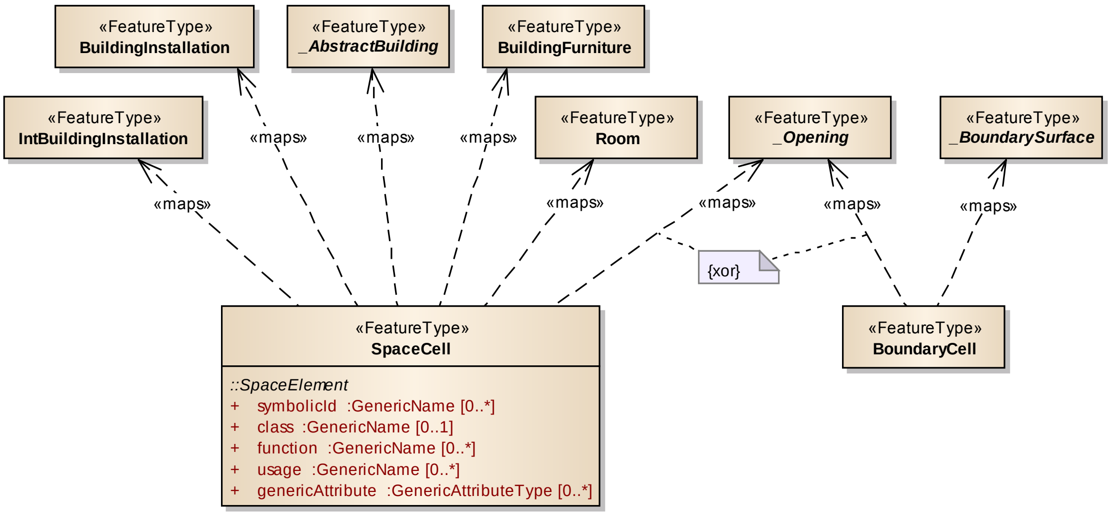

«FeatureType» BuildingInstallation
«FeatureType» _AbstractBuilding
«FeatureType» BuildingFurniture
«FeatureType» IntBuildingInstallation
«FeatureType» Room
«FeatureType» _Opening
«FeatureType» _BoundarySurface
«maps»
{xor}
«FeatureType» SpaceCell
::SpaceElement
+ symbolicId :GenericName [0..*]
+ class :GenericName [0..1]
+ function :GenericName [0..*]
+ usage :GenericName [0..*]
+ genericAttribute :GenericAttributeType [0..*]
«FeatureType» BoundaryCell

Figure 224: Linkage between the conceptual UML model elements of CityGML and the MLSEM.

6.2 Volumetric Elements Models

The derivation of a topographic space layer from *volumetric elements models* is illustrated in the following along the international standard IFC from the field of BIM (cf. chapter 2.3.1). The IFC representation of the example indoor scene used in the previous section is sketched in figure 225. The conceptual IFC entities for describing the elements of the built structure have been introduced in the UML diagram presented in figure 30.

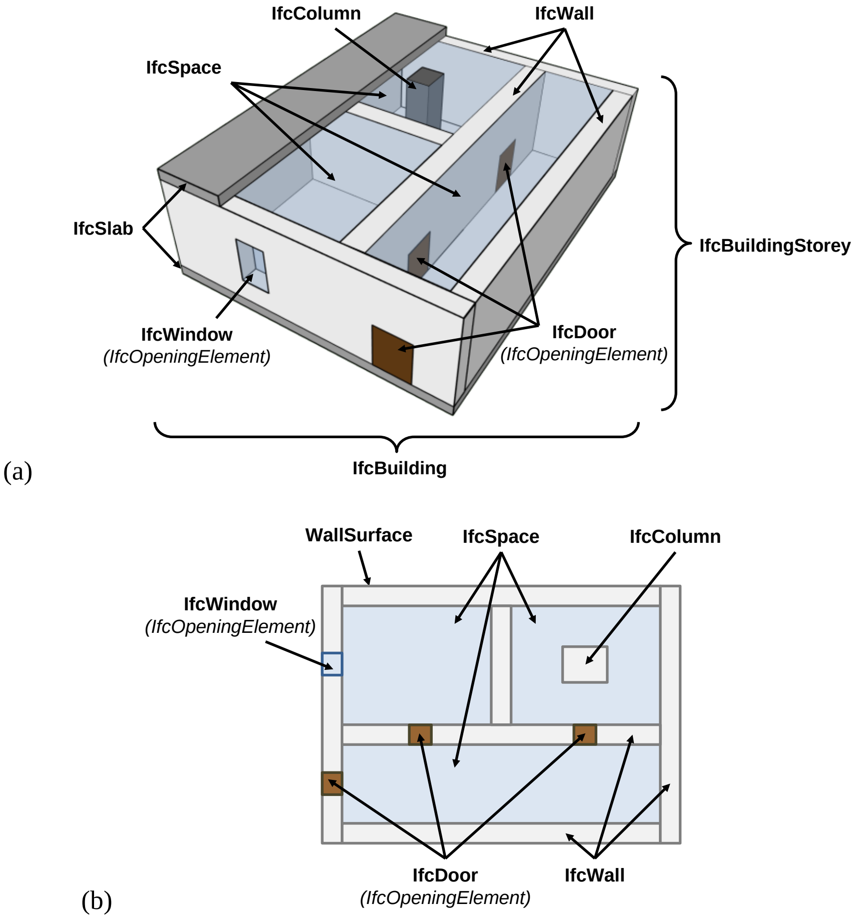


Figure 225: Example indoor scene as 3-dimensional IFC building model (a) and as 2-dimensional top view (b).

A substantial difference between the IFC model and the surface-based description of the indoor setting is the explicit representation of the main structural components that constitute the building carcass. Precisely, the IFC model contains those elements that occupy the wall space (e.g., walls, slabs, columns, beams) and that represent the building structure *as designed* but not necessarily *as observed* (cf. Nagel et al. 2009). The wall elements are modelled as instances of *IfcWall*. As can be seen from figure 225, an *IfcWall* may contribute to either or both the exterior building shell and the boundary of rooms and corridors. *IfcSlab* elements enclose the interior spaces horizontally and thus describe their lower support (floor) as well as their upper construction (ceiling, roof). Both walls and slabs may have voids which are represented as conceptual entities in their own right through the class *IfcOpeningElement*. Doors (*IfcDoor*) and windows (*IfcWindow*) are normally inserted into these openings in order to fill the void. A further vertical member of the building structure in the above example is the column inside one of the rooms which is modelled as *IfcColumn*. Similar to CityGML, the free spaces inside a facility such as rooms and corridors are described as bounded volume and captured by *IfcSpace* entities.

In IFC, the spatial representation of building elements is typically given as 3-dimensional solid object (Benner et al. 2005). In most cases, the geometry of a single building element hereby renders a single-shell manifold solid. The IFC elements can thus be mapped to individual *SpaceCell* features when translating an IFC model to the MLSEM. A possible topographic space layer following from the above setting is shown in figure 226. The intra-layer graph resulting from applying the mathematical model of the MLSEM consequently contains a dual node for every building element. Due to the wall space being partitioned into several space cells, the intra-layer graph contains more graph elements and hence is more complex compared to the surface-based representation shown in figure 221. This might be seen as disadvantageous since the dual nodes of building elements are typically non-traversable and hence need to be neglected in path searches anyways. However, it also allows for attaching valuable information such as the material of the building elements to the corresponding space cells which makes this information accessible from the dual nodes. Especially in emergency situations the material might become essential since fire fighter forces or rescue personnel equipped with appropriate tools might be able to tear down specific walls (e.g., a combined navigation constraint can be used to express this fact). In the mapping of a surface-based model, the entire wall space is captured by a single space cell which obviously does not suffice to denote the different materials of different structural elements. In contrast, the material information has to be kept with the *BoundaryCell* features instead. This representation also faces drawbacks. First, material information needs to be stored redundantly for all boundary cells belonging to the same building element. Second, the information that two or more boundary cells represent parts of the same building element is only available implicitly (cf. discussion in chapter 6.1). From this perspective, the greater number of dual nodes resulting from an IFC model might also be seen as advantage.

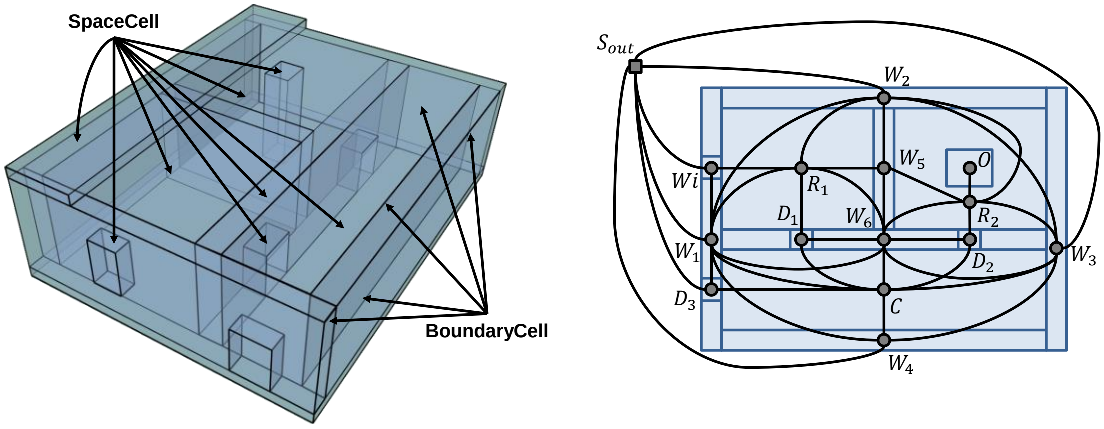


Figure 226: Corresponding topographic space layer of the MLSEM for the IFC model from figure 225. The intra-layer graph on the right assumes a minimal CW decomposition.

IFC supports various spatial representation schemes for describing the volumetric shape of building elements and free spaces. Amongst others, this comprises CSG, B-Rep, translational and rotational sweep representations, and parametric descriptions. It immediately follows that representations other than B-Rep need to be translated into the B-Rep scheme in order to serve as primal space representation of *SpaceCell* features in the context of the MLSEM. This translation involves additional algorithms from computational geometry but can be realized in an unambiguous way (cf. chapter 4.1). In contrast to surface-based models, the boundary of the volumetric elements is not represented by separate conceptual entities that can be mapped to *BoundaryCell* features. Nevertheless, IFC foresees the possibility to denote the 2-dimensional surface geometry where two adjacent building entities touch, which can be used to populate the primal space representation of boundary cells. Topological adjacency is generally expressed on a logical level through the relationship class *IfcRelConnects* (cf. chapter 2.3.1). The subtype *IfcRelConnectsElements* allows for additionally capturing the physical connection geometry (*IfcConnectionGeometry*) between two elements of the building structure that are given as subtypes of *IfcElement* (e.g., between walls, slabs, windows, doors). These boundary surfaces are typically non-observable and thus are not represented in surface-based models. Since *IfcSpace* is not derived from *IfcElement*, the relationship class *IfcRelSpaceBoundary* is provided for the representation of the surfaces delimiting a free space. *IfcRelSpaceBoundary* is similarly subtyped from *IfcRelConnects* and defines the relationship of a space to its surrounding building elements. It may have a connection geometry attached which can be classified to be intangible or tangible. Intangible space boundaries occur at adjacent physical *IfcElement* entities, whereas tangible boundaries can be used, for example, to express a virtual and hence navigable connection between two *IfcSpace* instances. It follows that a connection geometry of an *IfcRelSpaceBoundary* coincides with an observable (or virtual) interior surface in surface-based models. Further semantic information about the space boundary can be drawn from the building element associated with the *IfcRelSpaceBoundary*. Space boundaries being topologically connected can be carried to a single *BoundaryCell* feature. If relationship classes or connection geometries are not provided by an IFC model, then the primal space representation of boundary cells has to be derived from the intersection of the volumetric building elements.

Similar to CityGML, also IFC allows spatial overlaps and permeations between the spatial entities of the built environment. For example, openings and their filling elements at least partly occupy the same portion of space per definition. Consequently, they cannot be mapped to separate space cells on the same topographic space layer. In figure 226, the spatial description of the *IfcOpeningElement* has been used for the space cells representing the doors and windows. This might be sufficient in many cases since the more detailed real world geometry of the *IfcDoor* and *IfcWindow* elements can still be queried on demand when being related to the *SpaceCell* using an instance of *SourceObject* (e.g., for visualization purposes or for evaluating navigation constraints). Likewise, the solid manifold describing the *IfcColumn* is spatially contained inside the *IfcSpace* representing the room. The primal space geometry of the room cell for the MLSEM hence has to be derived from a regularized Boolean subtraction as illustrated in the context of CityGML.

When neglecting the physical wall and slab elements by simply omitting the instances of the corresponding conceptual entities from the IFC model, then the topographic space layer as depicted in the following figure 227 can be derived from the remaining *IfcSpace* entities, their space boundaries, as well as additional elements such as doors and windows. Both the primal and dual space representation agrees with the space layer derived from the CityGML model as shown in figure 223. It thus follows that, despite their different modelling paradigms, IFC and CityGML building models can lead to similar MLSEM representations. Note that in this alternative, the transition of semantic attributes from the omitted building elements to the *BoundaryCell* features representing the corresponding space boundaries of *IfcSpace* entities is essential in order to not loose information.

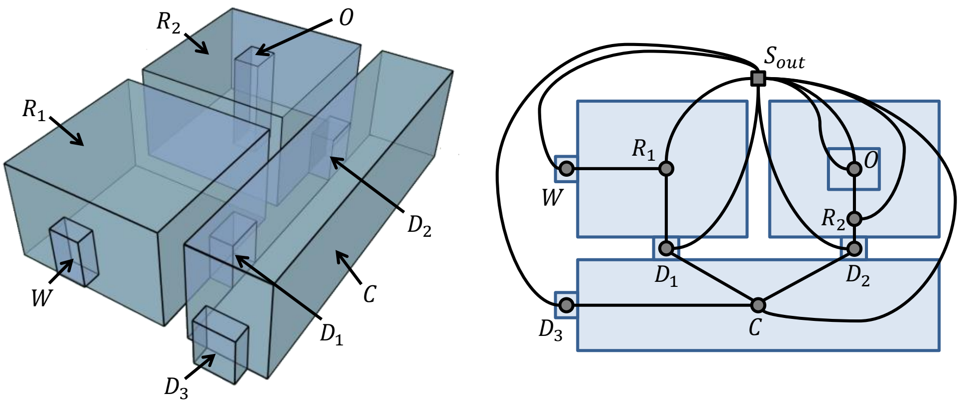


Figure 227: Omitting the physical wall and slab elements from the IFC model from figure 225.

IFC models inherently represent the hierarchical structure of facilities (cf. chapter 2.3.1). An *IfcBuilding* is decomposed into building floors which are defined as *IfcBuildingStorey*. Moreover, *IfcSpace* entities may span over several connected spaces which can be used, for example, to model sub-hierarchies on the floor level such as wings or organizational units. Since each of these hierarchical structure elements may also carry a geometric representation, topographic subspace layers capturing the spatial containment relationships can be easily derived.

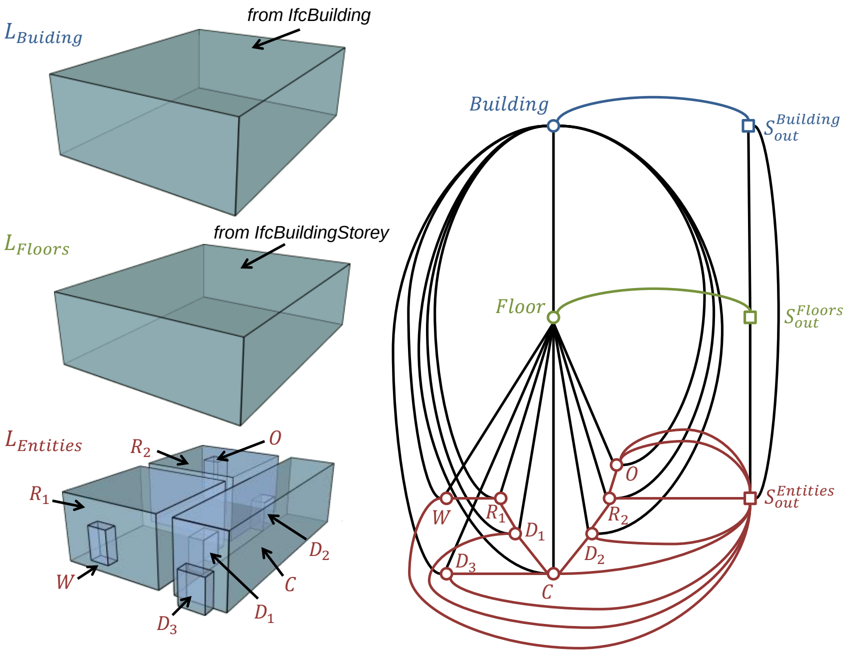


Figure 228: Subspace layer hierarchy derived from the IFC model from figure 225.

Finally, the conceptual mapping between IFC and the MLSEM is shown along excerpts of their UML models in figure 229. All IFC model elements are subsumed under the notion of the *SpaceCell* feature type and there is no direct conceptual counterpart for *BoundaryCell*. However, the primal space geometry of *BoundaryCell* features may be populated from *IfcConnectionGeometry* elements as discussed above.

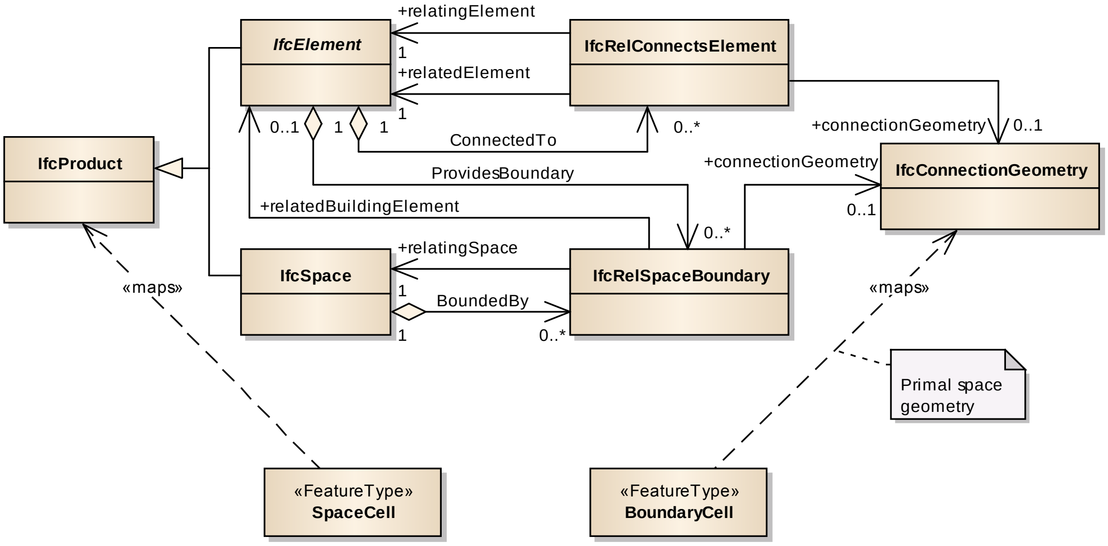


Figure 229: Linkage between the conceptual UML model elements of IFC and the MLSEM.

6.3 Paper Models

Paper models of buildings can be viewed as subtypes of surface-based building models which are characterized by the fact that the volumetric shape of the structural building elements such as walls and slabs is abstracted by 2-dimensional surfaces. This abstraction results in a skeleton representation of the wall space whose inherent dimensional reduction decreases the overall model complexity. Various methods for deriving the skeleton of solid objects are discussed in literature, amongst which medial axis transforms are one of the most widely studied. Skeletons are commonly requested to be homotopy equivalent to the solid object, and thus the number of holes and interior voids has to remain the same. Likewise, relative dimensions of the object's shape as well as topological relationships to other objects need to be preserved in the skeletal abstraction.

A paper model of the example indoor scene that observes these rules is sketched in figure 230. The walls and slabs are represented through their mid surfaces. The rooms and the corridor are enclosed by these mid surfaces and thus render bounded volumes. Since the wall thickness is abstracted, it follows that the modelled free space is actually greater than the available free space. This fact needs to be considered when using paper models in the context of indoor navigation. Similar to the CityGML representation, the column is described by its tangible boundary surfaces and not idealized by a single mid surface. This ensures that the topological information about the column rendering a through hole of the room space is kept within the model. The doors and windows are given as 2-dimensional surfaces on the boundary of the rooms and the corridor, which again corresponds to the CityGML model in figure 220.

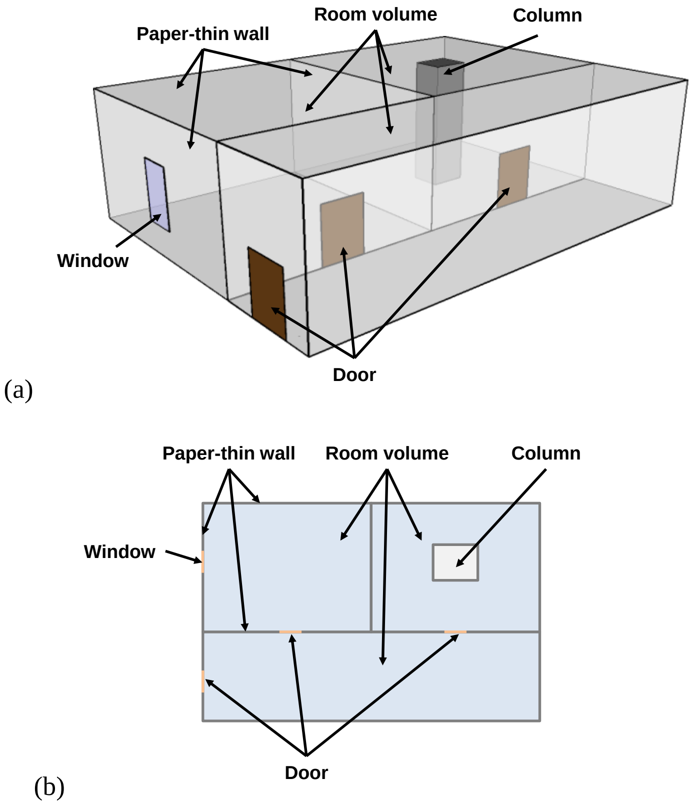


Figure 230: Example indoor scene as 3-dimensional paper model (a) and as 2-dimensional top view (b).

The identification of space cells and boundary cells in the presented paper model is shown below. The resulting dual graph assumes a minimum number of boundary cells and is similar to the graph derived from the surface-based model in figure 221. Nevertheless, there are also distinct differences between both. First, the wall space is not mapped to a separate dual node anymore. Therefore, the dual nodes of the rooms R_1 and R_2 are directly connected through a dual edge. Second, two dual edges link R_1 with the corridor C since the door and its surrounding wall render separate 2-cells on the common boundary of both space cells in primal topology space (cf. figure 77a and the related discussion). For the same reason, there are also two dual edges connecting R_2 and C as well as both R_1 and C with S_{out}. In order to avoid paths involving the dual edges of the wall surfaces, the *BoundaryCell* features may be assigned *NonPassableConstraint* entities.

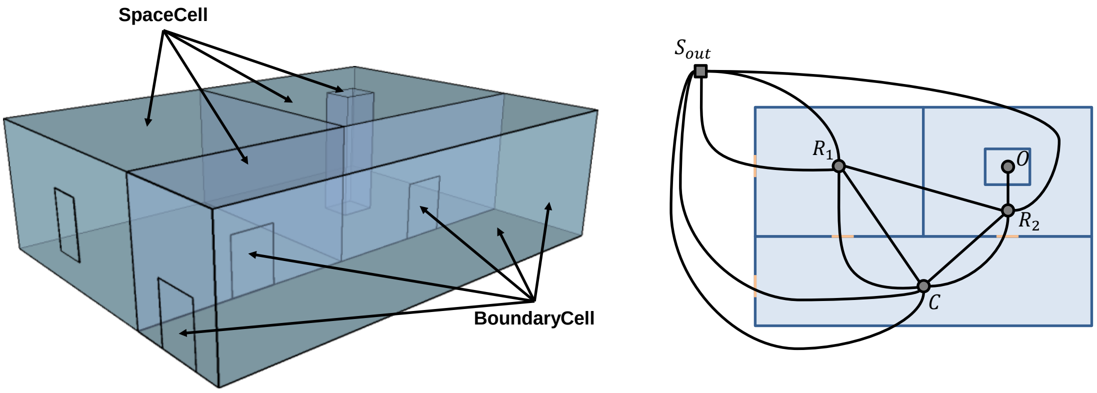


Figure 231: Corresponding topographic space layer of the MLSEM for the paper model from figure 230. The intra-layer graph on the right assumes a minimal CW decomposition.

A further level of abstraction is often applied in paper models by omitting the representations of windows and doors in wall surfaces. The impact of this modelling approach on the dual space representation is illustrated in figure 232. Obviously, the number of dual edges is reduced since the rooms are connected to the corridor through a single dual edge. Although this graph representation is closer to the surface-based model in figure 221, a differentiation between the wall and its enclosed openings is not possible which hinders the modelling of semantic information about the separate space entities (e.g., material information about the wall and access control associated with the doors). Moreover, assume there are two or more openings within the same wall surface. Then this fact cannot be reflected in the dual graph. Nevertheless, the MLSEM is flexible enough to also express this modelling approach.

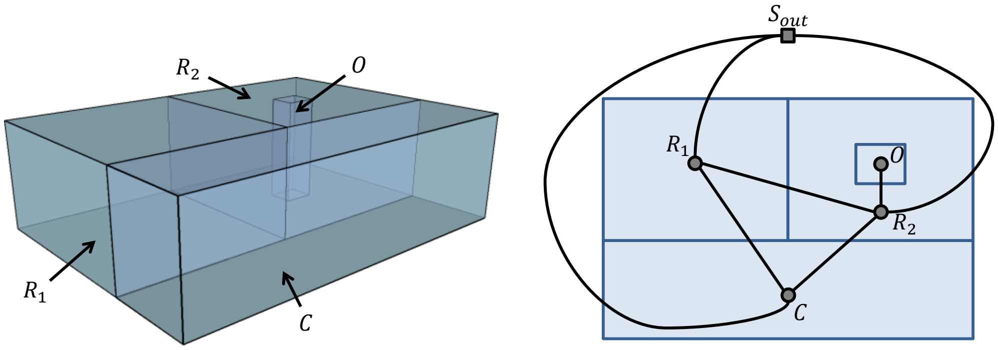


Figure 232: Further abstraction model of the paper model from figure 230.

From the approaches to indoor space modelling for indoor navigation as presented in chapter 2.2, the proposal of (Boguslawski & Gold 2009) utilizes a paper-based representation of the interior built environment. A conceptual model of the indoor entities that could be related with the *SpaceCell* and *BoundaryCell* concepts of the MLSEM is however not presented by the authors. Alternatively, also CityGML is capable of capturing paper-thin representations of the interior built environment, in which case the translation of semantics to the entities of the MLSEM follows the rules presented in chapter 6.1.

6.4 Floor Plan Models

The group of 2-dimensional *floor plan models* is discussed along the ESRI Building Interior Space Data Model (BISDM) (cf. chapter 2.3.3). A possible BISDM representation of the example indoor setting is presented in the following figure 233. The rooms and the corridor are mapped onto instances of *InteriorSpace* and are spatially described by 2-dimensional polygons. The 1-dimensional *FloorplanLine* geometries representing the boundaries of walls are chosen so that they reflect the projected shape of the separate wall components and thus also include non-visible boundaries. The boundary lines of the doors and the window are given in a similar way and hence preserve the thickness of the corresponding spatial entities.

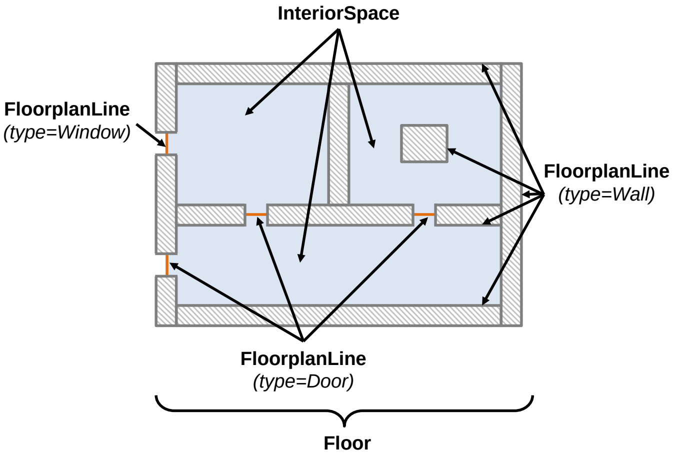


Figure 233: Example indoor scene as 2*-dimensional ESRI BISDM model.*

The figure below depicts the mapping of the BISDM model to a 2-dimensional topographic space layer of the MLSEM. It is an easy consequence that *InteriorSpace* entities are translated into *SpaceCell* features, whereas the *FloorplanLine* instances are carried to *BoundaryCell* features. Additional space cells are required to capture those spaces occupied by wall, door, and window elements. Since these spaces are not available from the BISDM model as 2-dimensional objects directly, they need to be derived from the spatial configuration of the boundary lines. The resulting intra-layer graph in figure 234 assumes a minimum number of boundary cells. It agrees to a great extent to the intra-layer graph derived from the IFC model of the indoor scene as shown in figure 226 which is due to the fact that the structural building elements are at least described implicitly in the above BISDM model. Since in two dimensions, openings necessarily disconnect a wall into spatially disjoint components which have to be mapped to individual space cells, the intra-layer graph in figure 234 even contains a greater number of graph elements.

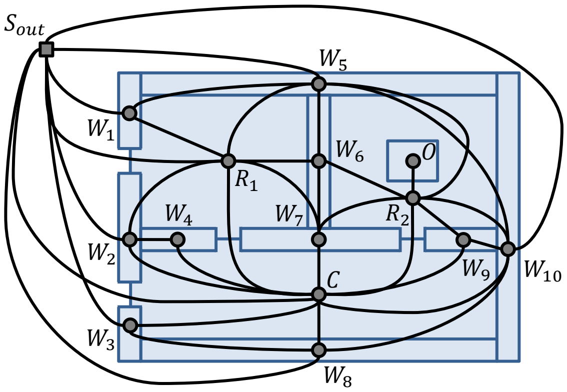


Figure 234: Corresponding topographic space layer of the MLSEM for the ESRI BISDM model from figure 233. The intra-layer graph assumes a minimal CW decomposition.

The ESRI feature model generally employs the B-Rep scheme for the description of spatial characteristics. The geometric representation of the BISDM floor plan elements can thus be taken for the primal space geometry of the respective *SpaceCell* and *BoundaryCell* features. However, ESRI polygons and polylines are allowed to have disconnected parts (so-called multipart features), and polylines may branch. In contrast, both space cells and boundary cells are required to form connected manifold spaces (cf. definition 3.2 and definition 3.26). Therefore, it may involve multiple space cells respectively boundary cells to represent the space covered by a single ESRI polygon or polyline. Moreover, and similar to CityGML and IFC, the floor plan elements are not constrained to be spatially non-overlapping. *FloorplanLine* instances, for example, may rather lie within the geometry of *Floor* and *InteriorSpace* entities (e.g., the border lines of the column in figure 233). Likewise, the geometries of interior spaces are obviously inside the geometry of floors and floor sections. The mapping to the primal space geometry of MLSEM features may thus require regularized Boolean operations. If a BISDM model only contains *Floor* and *InteriorSpace* features but no *FloorplanLine* instances, then the primal space representation of boundary cells has to be derived from the intersection of the non-overlapping polygon geometries.

Figure 235 illustrates alternative BISDM representations of the indoor scene. In figure 235a, only the visible boundaries of the building elements are captured by *FloorplanLine* objects, whereas the wall openings are represented through a centreline. When translated to a topographic space layer of the MLSEM following the above rules (cf. right of figure 235a), the resulting intra-layer graph is mostly consistent with the one derived from the CityGML model in figure 221. Note that topologically connected *FloorplanLine* entities have been carried to a single boundary cell in order to reduce the number of dual edges. Similar to the above example, the differences in the dual topology space compared to figure 221 again arise from the reduction of the dimension. For example, in two dimensions, the wall space is disconnected and hence needs to be described by two space cells. The BISDM model in figure 235b uses centrelines for all element boundaries. The dual graph of the MLSEM mapping on the right of figure 235b therefore corresponds to the 3-dimensional representation derived from a paper model as depicted in figure 231. Again, observe the greater number of graph elements (e.g., each room is linked to the corridor through three dual edges in two dimensions).

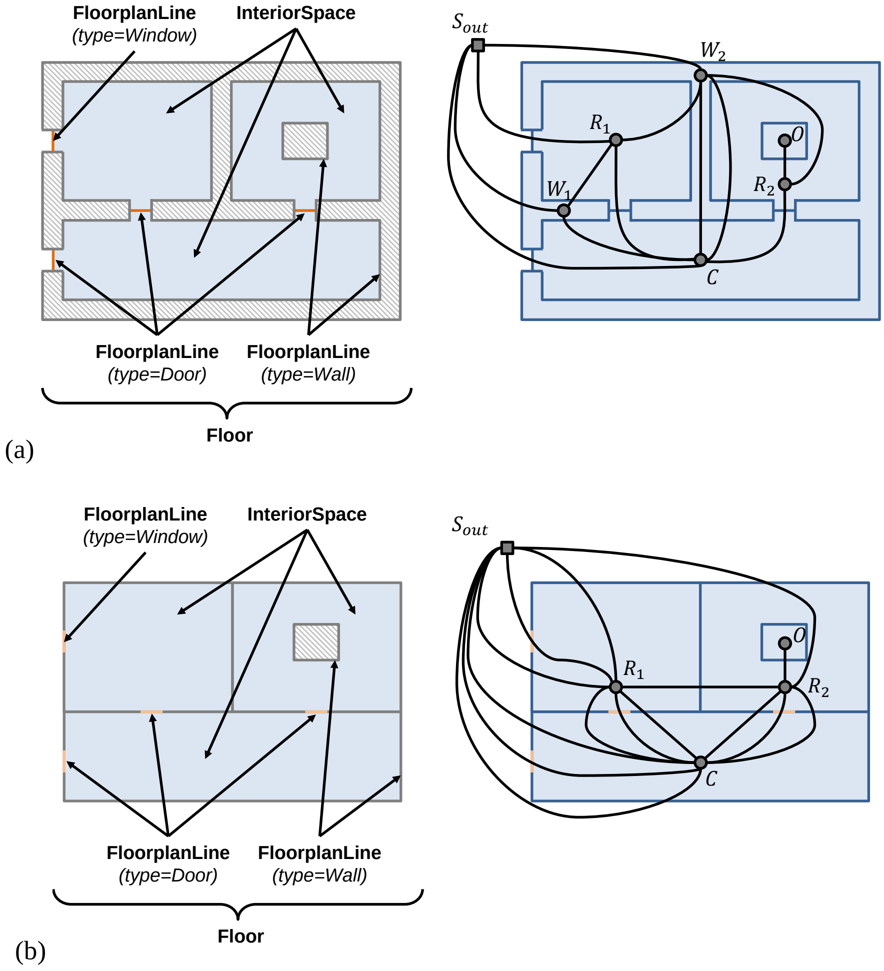


Figure 235: Alternative ESRI BISDM models of the example indoor scene based on the visible boundaries of building elements (a) and their centreline representation (b).

Since the BISDM foresees spatial representations for the footprint of the *Building* as well as for the extent of the separate *Floor* entities and *FloorSection* parts therein, this information can be used to populate a corresponding hierarchy of subspace layers for the topographic space in order to make the nested hierarchical structure of the facility as well as the spatial containment relationships between the interior entities available in the MLSEM representation.

Similar to the discussion of CityGML and IFC in the previous sections, the mapping between the conceptual model elements of the BISDM and the MLSEM is illustrated based on their UML diagrams in figure 236.

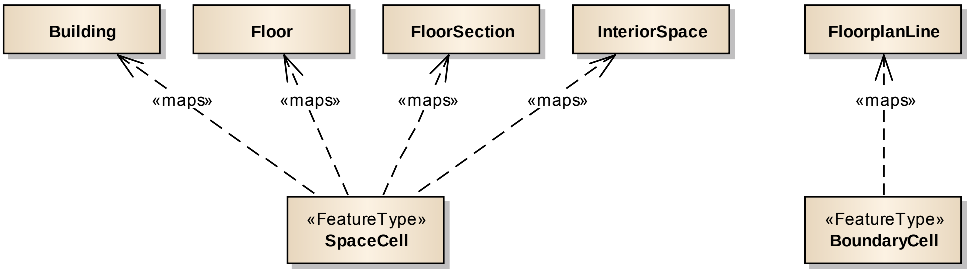


Figure 236: Linkage between the conceptual UML model elements of the ESRI BISDM and the MLSEM.

Although the conceptual model of the BISDM allows a *Building* object to aggregate a collection of *Floor* entities that are differentiated by their *baseElevation* property, it is important to note that each of these *Floor* entities needs to be captured by a separate 2-dimensional space layer complex when mapping the BISDM representation to the MLSEM (cf. discussion in chapter 4.4.1.6). The concept of *TransferTransition* as introduced in chapter 4.4.1.6 then allows for linking the separate multilayered graphs representing the different floor plans. Precisely, a *TransferTransition* allows for establishing a link between the dual nodes of space cells representing the same vertical transition space on different floors (e.g., an elevator, staircase, or escalator). This information can be evaluated by path search algorithms in order to determine routes between the different floors. Note that the notion of *FloorTransition* as defined for the graph-based conceptualization of indoor space within the BISDM (cf. chapter 2.3.3) directly corresponds to *TransferTransition* and thus facilitates the automatic creation of corresponding instances. Likewise, the centreline representations of corridors and hallways modelled by BISDM *FloorLines* are feasible to populate the dual space geometry of a topographic space layer. Navigation constraints provided for the elements of the BISDM navigation graph can also be translated to the MLSEM without information loss. For example, Boolean flags denoting the navigability of graph edges can be carried to instances of *NonPassableConstraint* associated with the corresponding MLSEM features. Moreover, restricted turns as well as time-based costs can be mapped to prohibited maneuver constraints (cf. chapter 5) respectively cost values associated with the dual edges of the intra-layer graph (cf. chapter 4.4.1.1).

Summary. It follows from the discussion in this chapter that the feature types *SpaceCell* and *BoundaryCell* from the MLSEM application schema provide suitable *conceptual interfaces* which connect the MLSEM to semantic building modelling standards such as CityGML, IFC, and ESRI BISDM without the need for duplicating or replacing these standards. The generic notion of space and the high abstraction level of the conceptual interfaces allow for keeping the semantic impedance low in this mapping. Semantic information can then be translated along identified elements from either conceptual model. Likewise, it has been shown that the primal space geometry can be populated from the connected elements of the building modelling standards even if they follow different spatial modelling paradigms. Purely geometric or graphical building models (cf. chapter 2.3.4) have not been explicitly considered in this chapter. Nevertheless, such models at least suffice to derive the primal space geometries of space cells and boundary cells on a topographic space layer (from which the dual space representations follow in a deterministic manner). However, and discussed in chapter 2.3.4, visualization models rather focus on the geometry and appearance of those surfaces which are required for rendering an indoor scene. Whether these surfaces are geometric-topologically consistent or form a solid object is commonly of small importance. Thus, populating the primal space geometry of space cells and boundary requires the reconstruction of well-behaved spatial objects in the sense of definition 3.2. Moreover, since geometric and graphical building models lack semantics, further object recognition and interpretation procedures might become necessary to semantically classify the derived spatial objects (cf., Baltsavias 2004, Brenner 2005, Kolbe et al. 2009).

Chapter 7

Example Applications of the MLSEM

This chapter provides two “proof of concept” demonstrations for the applicability and feasibility of the MLSEM and the concepts developed in this thesis. First, the XML representation of an indoor scene according to the GML application schema discussed in chapter 4.4.2 and provided in appendix B is illustrated. The example thus nicely concludes a main thread and goal of this thesis: from the definition of a spatio-semantic indoor space model for indoor navigation at a conceptual and mathematical level (cf. chapter 3) via its complete and consistent mapping onto a conceptual data model (cf. chapter 4 and chapter 5) to the actual encoding of indoor space data according to the MLSEM framework in a standardized way. The XML structures for the most important concepts of the MLSEM are discussed in detail in order to demonstrate the possibility for the lossless exchange of MLSEM indoor space models between computer systems.

The second example is about the acquisition of a 3-dimensional MLSEM indoor space model for a real world building, namely the *Main Building* of the Technische Universität Berlin. The example presents the results of a Master students’ project conducted at the Institute for Geodesy and Geoinformation Science of the Technische Universität Berlin which was co-supervised and intensely supported by the author. The students were expected to model a topographic space layer mapping the interior built environment of the *TU Main Building* in both primal and dual space and to augment this space view with a sensor space layer capturing the signal reception areas of Wi-Fi transmitters equipped in the building. In addition, the multilayered graph structure linking both space layers in dual space had to be established. A second deliverable of the students’ project was the specification of a database schema allowing for the storage and management of MLSEM indoor space models within a spatial database in accordance with the conceptual data model developed in chapter 4.

Note that the realization of a (prototypical) indoor navigation system is out of scope of this thesis since this necessarily involves implementations for the tasks of localization, path planning, as well as guidance and tracking. The focus of this thesis is however on the spatio-semantic modelling of indoor environments as mandatory prerequisite for these navigation tasks but not on their practical implementation.

7.1 XML Encoding of an MLSEM Indoor Space Model

The MLSEM conformant XML encoding of indoor space data is demonstrated for the indoor scene shown in figure 237. It recaps the example from chapter 3.1.2.3 which has been discussed at several places throughout this thesis. The scene consists of two space layers, namely a topographic space layer and a Wi-Fi space layer, whose spatial configuration and overlap in primal space is sketched on the left of figure 237. The resulting multilayered graph integrating the corresponding dual space representations is depicted on the right. In order to reduce the XML encoding of the spatial representations of space cells and boundary cells in XML excerpts presented in the subsequent sections, the indoor space is described in two dimensions only.

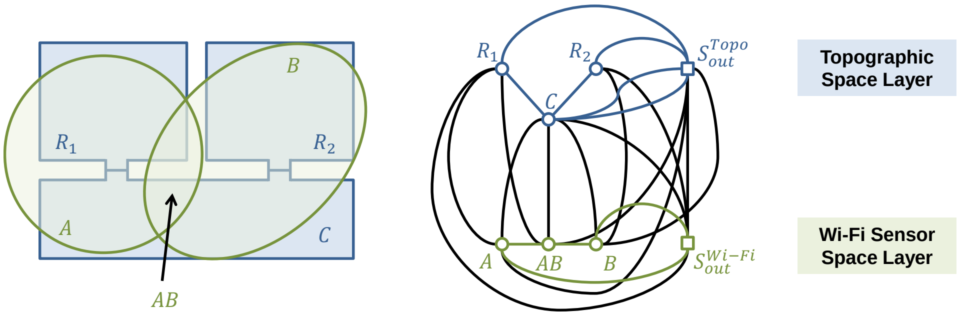


Figure 237: Example 2*-dimensional space layer complex (left) and corresponding multilayered graph (right).*

The general XML skeleton of any MLSEM instance document is presented in the listing 1 shown below. The root element <SpaceLayerComplex> encodes the *SpaceLayerComplex* feature type of the conceptual data model. It hence realizes a container for the *SpaceLayer* features describing the separate views on indoor space each of which is realized as nested <SpaceLayer> element. The *SpaceCell* and *BoundaryCell* features on a space layer as well as the resulting *IntraLayerGraph* representation are captured inside the <SpaceLayer> element. Further subelements of <SpaceLayerComplex> facilitate the encoding of both the multilayered graph and the joint states associated with the space layer complex itself.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SpaceLayerComplex xmlns="http://www.tu-berlin.de/igg/mlsem/1.0"
 xmlns:gml="http://www.opengis.net/gml/3.2"
 xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.tu-berlin.de/igg/mlsem/1.0 mlsem.xsd">

 <!-- space layers -->
 <layer>
    <SpaceLayer>
       <spaceCell> … </spaceCell>
       <boundaryCell> … </boundaryCell>
       <graph> … </graph>
    </SpaceLayer>
 </layer>

 <!-- multilayered graph -->
 <graph>
    <MultilayeredGraph>
       <subGraph> … </subGraph>
       <edge> … </edge>
    </MultilayeredGraph>
 </graph>

 <!-- joint states -->
 <stateSpace>
    <JointStateSpace>
       <state> … </state>
       <transition> … </transition>
    </JointStateSpace>
 </stateSpace>
</SpaceLayerComplex>
```

Listing 1: General XML skeleton of an MLSEM instance document.

XML encoding of space layers, space cells, and boundary cells. The XML encoding of a *SpaceLayer* feature is discussed for the topographic space view in the following. Figure 238 shows the two rooms R_1 and R_2 as well as the corridor C inhabiting the indoor space along with their dimensions in metres. In order to describe the geometry of the corresponding space cells in both primal and dual space, a local 2-dimensional Cartesian coordinate system is assumed for this example whose origin is defined to be located at the lower left corner of the corridor. In general, GML supports both local (engineering) and global spatial reference systems to be associated with the geometries in an instance document. Whereas the latter are typically predefined by an authority organization such as the European Petroleum Survey Group (EPSG) and hence can be given by a reference to this definition, a spatial reference system can also be self-defined within the same or an external GML document (cf. Portele 2012). The GML encoding of the local coordinate system used in this example is however omitted for brevity. In accordance with ISO 19107, GML provides the flexibility to apply multiple and different reference systems within the same MLSEM instance document which has been identified as essential requirement for indoor space models in chapter 1.2.

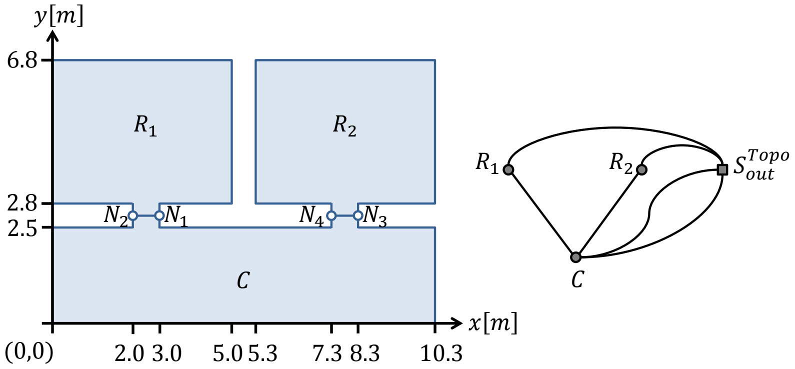


Figure 238: Local engineering coordinate system applied to the topographic space layer.

A <SpaceLayer> is characterized by its mandatory <type> property. In listing 2, this property receives the value *topographic* which is taken from a predefined but extensible code list. The nested <SpaceCell> element maps the space cell representing the room R_1 from the above figure. The room number *5126* is encoded as symbolic information through the <symbolicId> property, whereas the <class> property is used to classify the space cell as *office*. Further semantic information about the office room such as its usage and function as well as a name, description, or arbitrary further generic attributes may be added but have been omitted in this example for brevity. The same holds for the semantic attributes of the <SpaceLayer> element.

```xml
<SpaceLayer gml:id="TopographicSpace">
 <type>topographic</type>
 <spaceCell>
   <SpaceCell gml:id="R1">
     <symbolicId codeSpace="roomNo">5126</symbolicId>
     <class>office</class>
     <primalSpace>
       <Space gml:id="R1_PS">
         <geometry>
           <gml:Polygon gml:id="R1_PS_Geometry" srsDimension="2">
             <gml:exterior>
               <gml:Ring>
                 <gml:curveMember>
                   <gml:LineString gml:id="R1_PS_Geometry_Wall">
                     <gml:posList>3 2.65 3 2.8 5 2.8 5 6.8 0 6.8 0 2.8 2 2.8 2 2.65</gml:posList>
                   </gml:LineString>
                 </gml:curveMember>
                 <gml:curveMember>
                   <gml:LineString gml:id="R1_PS_Geometry_Door">
                     <gml:posList>2 2.65 3 2.65</gml:posList>
                   </gml:LineString>
                 </gml:curveMember>
               </gml:Ring>
             </gml:exterior>
           </gml:Polygon>
         </geometry>
         <topology>
           <gml:Face gml:id="R1_PS_Topology">
             <gml:directedEdge>
               <gml:Edge gml:id="R1_PS_Topology_Wall">
                 <gml:directedNode>
                   <gml:Node gml:id="R1_PS_Topology_N1"/> <!-- cf. N1 in figure 238-->
                 </gml:directedNode>
                 <gml:directedNode>
                   <gml:Node gml:id="R1_PS_Topology_N2"/> <!-- cf. N2 in figure 238-->
                 </gml:directedNode>
                 <gml:curveProperty xlink:href="#R1_PS_Geometry_Wall"/>
               </gml:Edge>
             </gml:directedEdge>
             <gml:directedEdge>
               <gml:Edge gml:id="R1_PS_Topology_Door">
                 <gml:directedNode xlink:href="#R1_PS_Topology_N2"/> <!-- cf. N2 in figure 238-->
```

```
                  <gml:directedNode xlink:href="#R1_PS_Topology_N1"/> <!-- cf. N1 in figure 238-->
                  <gml:curveProperty xlink:href="#R1_PS_Geometry_Door"/>
                </gml:Edge>
              </gml:directedEdge>
              <gml:surfaceProperty xlink:href="#R1_PS_Geometry"/>
            </gml:Face>
          </topology>
        </Space>
      </primalSpace>
    </SpaceCell>
  </spaceCell>
  …
</SpaceLayer>
```

Listing 2: XML encoding of the primal space topology and geometry of the topographic space cells.

The primal space geometry and topology of a *SpaceCell* feature is mapped by a <Space> element inside the corresponding <SpaceCell> node. XML constructs from GML (identified by the namespace prefix *gml*) are used to specify both space representations. In listing 2, the primal space <geometry> of the room cell R_1 is given by a 2-dimensional <gml:Polygon> whose exterior boundary component is described by two separate 1-dimensional curve primitives. The first boundary curve denotes the walls enclosing the room and separating it from the outer space, whereas the second curve represents the door between R_1 and C. The curve geometries are implemented as <gml:LineString> which enforces a linear interpolation between the provided coordinate tuples. Together both curve geometries form a closed ring being homeomorphic to $\mathbb{S}^1$. The primal space <topology> is then provided as <gml:Face> which realizes the *TP_Face* primitive from ISO 19107. Similar to the geometric representation, its boundary is also defined by two <gml:Edge> elements. The first edge uses the node primitives N_1 and N_2 (cf. figure 238) as its start respectively end node. Its geometric realization is specified by referencing the curve geometry representing the walls from the primal space geometry using the XLink mechanism of GML. This reference not only ensures data consistency between both space representations but also compacts the XML encoding. The second edge reuses both <gml:Node> instances in a reversed order to express the topological 1-cell representing the door, and thus references the corresponding curve geometry for its Euclidean space embedding. The entire 2-dimensional face primitive itself is finally realized by the <gml:Polygon> geometry object.

The <Space> element is deliberately chosen so that it demonstrates a fully populated geometric-topological description of R_1 in primal space. Although this well reflects the mathematical notion of a space cell, either of both representations might already be sufficient for a navigation system to be stored and exchanged in an MLSEM instance document. For this reason, both the conceptual data model as well as the derived GML application schema of the MLSEM foresees the possibility to merely encode the primal space geometry, topology, or even neither of them (cf. related discussion in chapter 4.4.1.1). For example, in listing 2, the <topology> property could be deleted from the <Space> element without invalidating the instance document.

The fact that the boundary of R_1 is topologically described by two 1-cells in the above example implies that the dual node of R_1 is incident to two dual edges according to the Poincaré duality theorem. Note that the illustrated example reflects a minimum decomposition of the boundary of R_1 into 1-cells. An alternative cell decomposition could capture each wall component by a separate 1-cell in order to make the walls available as individual dual edges in the intra-layer graph. The encoding of the dual space representation of R_1 is depicted in the following XML excerpt.

```
<SpaceCell gml:id="R1">
  <symbolicId codeSpace="roomNo">5126</symbolicId>
  <class>office</class>
  <primalSpace> <!-- cf. listing 2 --> </primalSpace>
  <dualSpace>
    <State gml:id="R1_DS">
      <geometry>
        <gml:Point gml:id="R1_DS_Geometry" srsDimension="2">
          <gml:pos>2.5 4.8</gml:pos>
        </gml:Point>
      </geometry>
      <topology>
        <gml:Node gml:id="R1_DS_Topology">
```

```
          <gml:pointProperty xlink:href="#R1_DS_Geometry"/>
        </gml:Node>
      </topology>
    </State>
  </dualSpace>
</SpaceCell>
```

Listing 3: XML encoding of the dual space representation of the room R_1.

The <State> element allows for the geometric-topological description of a space cell in dual space. In the above listing 3, the dual <topology> node of R_1 (cf. right of figure 238) is given by a <gml:Node> element that implements a 0-dimensional *TP_Node* primitive. Its geometric realization references the additional dual space <geometry> implemented as <gml:Point> object. The spatial encoding in dual space hence follows the layout of the encoding of the primal space representation as shown in listing 2. In this example, the coordinate tuple of the <gml:Point> geometry denotes the centre point of the polygon that describes the shape of R_1.

The <SpaceCell> encoding of the room cell R_2 can be realized in the same way as illustrated for R_1. The following listing 4 therefore shows the XML representation of the corridor cell C. The primal <Space> occupied by C is also geometrically expressed by a <gml:Polygon>. Note that the exterior boundary of the polygon is described by four curve geometries two of which reuse those <gml:LineString> objects from the room geometries that spatially describe the door entities. Since the orientation of those curve segments needs to be reversed in the context of C, the corresponding XLink references are embedded in <gml:OrientableCurve> elements. Two more curve primitives are (at minimum) required to model the remaining disconnected walls bounding the corridor against the outer space. Only these additional curves require explicit coordinates. The primal space <topology> of the corridor cell is then defined by a 2-dimensional <gml:Face> primitive having four 1-cells on its boundary. Their start and end points obviously coincide with the node primitives N_1 to N_4 (cf. figure 238) being used in the topological description of the room cells. For this reason, the nodes are not defined anew but referenced per XLink. Since the 1-cells representing the door entities are on the common boundary of R_1 and C respectively R_2 and C, the corresponding <gml:Edge> elements are also reused. Again, their orientation has to be flipped in the context of C. Finally, the <State> description of C in dual space is defined similar to the dual space representations of the room cells (cf. listing 3).

```
<SpaceCell gml:id="C">
 <class>corridor</class>
 <primalSpace>
   <Space gml:id="C_PS">
     <geometry>
       <gml:Polygon gml:id="C_PS_Geometry" srsDimension="2">
         <gml:exterior>
           <gml:Ring>
             <gml:curveMember>
               <gml:LineString gml:id="C_PS_Geometry_Wall1">
                 <gml:posList>2 2.65 2 2.5 0 2.5 0 0 10.3 0 10.3 2.5 8.3 2.5 8.3 2.65</gml:posList>
               </gml:LineString>
             </gml:curveMember>
             <gml:curveMember>
               <gml:OrientableCurve gml:id="C_PS_Geometry_Door_R2" orientation="-">
                 <gml:baseCurve xlink:href="#R2_PS_Geometry_Door"/>
               </gml:OrientableCurve>
             </gml:curveMember>
             <gml:curveMember>
               <gml:LineString gml:id="C_PS_Geometry_Wall2">
                 <gml:posList>7.3 2.65 7.3 2.5 3 2.5 3 2.65</gml:posList>
               </gml:LineString>
             </gml:curveMember>
             <gml:curveMember>
               <gml:OrientableCurve gml:id="C_PS_Geometry_Door_R1" orientation="-">
                 <gml:baseCurve xlink:href="#R1_PS_Geometry_Door"/>
               </gml:OrientableCurve>
             </gml:curveMember>
           </gml:Ring>
         </gml:exterior>
       </gml:Polygon>
```

```
      </geometry>
      <topology>
        <gml:Face gml:id="C_PS_Topology">
          <gml:directedEdge>
            <gml:Edge gml:id="C_PS_Topology_Wall1">
              <gml:directedNode xlink:href="#R1_PS_Topology_N2"/> <!-- cf. N2 in figure 238-->
              <gml:directedNode xlink:href="#R2_PS_Topology_N3"/> <!-- cf. N3 in figure 238-->
              <gml:curveProperty xlink:href="#C_PS_Geometry_Wall1"/>
            </gml:Edge>
          </gml:directedEdge>
          <gml:directedEdge xlink:href="#R2_PS_Topology_Door" orientation="-"/>
          <gml:directedEdge>
            <gml:Edge gml:id="C_PS_Topology_Wall2">
              <gml:directedNode xlink:href="#R2_PS_Topology_N4"/> <!-- cf. N4 in figure 238-->
              <gml:directedNode xlink:href="#R1_PS_Topology_N1"/> <!-- cf. N1 in figure 238-->
              <gml:curveProperty xlink:href="#C_PS_Geometry_Wall2"/>
            </gml:Edge>
          </gml:directedEdge>
          <gml:directedEdge xlink:href="#R1_PS_Topology_Door" orientation="-"/>
          <gml:surfaceProperty xlink:href="#C_PS_Geometry"/>
        </gml:Face>
      </topology>
    </Space>
  </primalSpace>
  <dualSpace>
    <State gml:id="C_DS">
      <geometry>
        <gml:Point gml:id="C_DS_Geometry" srsDimension="2">
          <gml:pos>5.15 1.25</gml:pos>
        </gml:Point>
      </geometry>
      <topology>
        <gml:Node gml:id="C_DS_Topology">
          <gml:pointProperty xlink:href="#C_DS_Geometry"/>
        </gml:Node>
      </topology>
    </State>
  </dualSpace>
</SpaceCell>
```

Listing 4: XML encoding of the corridor space cell in primal and dual space.

The outer space cell on the topographic space layer is captured by a further <SpaceCell> whose XML mapping is presented in listing 5. In contrast to the space cells discussed above, the spatial encoding has to account for the fact that the *outer* space is topologically unbounded and hence non-compact. Conceptually, the ISO 19107 Spatial Schema supports the modelling of non-compact manifolds in both two and three dimensions by listing their boundary components as interior rings respectively interior shells of a corresponding primitive geometry object (cf. chapter 4.3.1). This mechanism also applies to the GML encoding of the conceptual classes from ISO 19107. On the topographic space layer, the boundary of the outer space cell corresponds to the boundary of the space cell complex being constituted by the room and the corridor cells. The <gml:Polygon> describing its primal space geometry therefore has a single interior <gml:Ring> which contains orientation-reversing references to the corresponding curve elements on the boundary of the primal space geometries of R_1, R_2, and C. Since exterior and interior boundary components are not distinguished in topology, the <gml:Face> representing the primal space topology of the outer space cell is characterized as being the *universal* and hence unbounded face by setting its XML attribute *universal* to *true* (this attribute is also available for a 3-dimensional <gml:TopoSolid>). Moreover, the <gml:Face> receives the unbounded <gml:Polygon> as geometric realization. Observe that the <gml:Face> likewise uses XLink references to the 1-cells on the common boundary with R_1, R_2, and C, and changes the sense of their orientation.

The dual <State> element of the outer space cell is populated by a <gml:Node> which implements the dual node S_{out}^{Topo} as shown on the right of figure 238. In contrast to the dual nodes of R_1, R_2, and C, the <gml:Node> lacks a geometric embedding. The dual space representation of the outer space cell is hence only expressed in topology space in this example. On the one hand, this encoding has been chosen to illustrate the flexibility of the MLSEM application schema. On the other hand, the outer space cell on a topographic space layer will likely represent non-navigable space in many scenarios and therefore not necessarily has to receive a Euclidean space embedding in such cases.

```
<SpaceCell gml:id="Topo_Outer">
 <gml:description>Outer space on topographic space layer</gml:description>
 <class>outer</class>
 <primalSpace>
   <Space gml:id="Topo_Outer_PS">
     <geometry>
       <gml:Polygon gml:id="Topo_Outer_PS_Geometry" srsDimension="2">
         <gml:interior>
           <gml:Ring>
             <gml:curveMember>
               <gml:OrientableCurve gml:id="Topo_Outer_PS_Geometry_Wall_R1" orientation="-">
                 <gml:baseCurve xlink:href="#R1_PS_Geometry_Wall"/>
               </gml:OrientableCurve>
             </gml:curveMember>
             <gml:curveMember>
               <gml:OrientableCurve gml:id="Topo_Outer_PS_Geometry_Wall2_C" orientation="-">
                 <gml:baseCurve xlink:href="#C_PS_Geometry_Wall2"/>
               </gml:OrientableCurve>
             </gml:curveMember>
             <gml:curveMember>
               <gml:OrientableCurve gml:id="Topo_Outer_PS_Geometry_Wall_R2" orientation="-">
                 <gml:baseCurve xlink:href="#R2_PS_Geometry_Wall"/>
               </gml:OrientableCurve>
             </gml:curveMember>
             <gml:curveMember>
               <gml:OrientableCurve gml:id="Topo_Outer_PS_Geometry_Wall1_C" orientation="-">
                 <gml:baseCurve xlink:href="#C_PS_Geometry_Wall1"/>
               </gml:OrientableCurve>
             </gml:curveMember>
           </gml:Ring>
         </gml:interior>
       </gml:Polygon>
     </geometry>
     <topology>
       <gml:Face gml:id="Topo_Outer_PS_Topology" universal="true">
         <gml:directedEdge xlink:href="#R1_PS_Topology_Wall" orientation="-"/>
         <gml:directedEdge xlink:href="#C_PS_Topology_Wall2" orientation="-"/>
         <gml:directedEdge xlink:href="#R2_PS_Topology_Wall" orientation="-"/>
         <gml:directedEdge xlink:href="#C_PS_Topology_Wall1" orientation="-"/>
         <gml:surfaceProperty xlink:href="#Topo_Outer_PS_Geometry"/>
       </gml:Face>
     </topology>
   </Space>
 </primalSpace>
 <dualSpace>
   <State gml:id="Topo_Outer_DS">
     <topology>
       <gml:Node gml:id="Topo_Outer_DS_Topology"/>
     </topology>
   </State>
 </dualSpace>
</SpaceCell>
```

Listing 5: XML encoding of the outer space cell on the topographic space layer in primal and dual space.

In addition to *SpaceCell* features, a <SpaceLayer> element also aggregates the *BoundaryCell* features on the space layer which are mapped onto <BoundaryCell> elements within an MLSEM instance document. Since the spatial description of each space cell already involves the representation of its boundary elements as illustrated in the previous listings, a <BoundaryCell> element simply has to provide XLink references to these descriptions for the encoding of its primal and dual space representation. The following listing 6 illustrates this for the boundary cell between the room cell R_1 and the outer space cell.

```
<SpaceLayer gml:id="TopographicSpace">
 <type>topographic</type>
 <spaceCell> <!-- cf. previous listings --> </spaceCell>
 …
 <boundaryCell>
    <BoundaryCell gml:id="R1-Topo_Outer">
       <gml:description>Wall between R1 and Outer</gml:description>
       <class>wall</class>
       <primalSpace>
          <SpaceBoundary gml:id="R1-Topo_Outer_PS">
             <geometry xlink:href="#R1_PS_Geometry_Wall"/>
             <topology xlink:href="#R1_PS_Topology_Wall"/>
          </SpaceBoundary>
       </primalSpace>
       <dualSpace>
          <Transition gml:id="R1-Topo_Outer_DS">
             <topology>
                <gml:Edge gml:id="R1-Topo_Outer_DS_Topology">
                   <gml:directedNode xlink:href="#R1_DS_Topology"/>
                   <gml:directedNode xlink:href="#Topo_Outer_DS_Topology"/>
                </gml:Edge>
             </topology>
          </Transition>
       </dualSpace>
    </BoundaryCell>
 </boundaryCell>
 …
</SpaceLayer>
```

Listing 6: XML encoding of the primal space topology and geometry of a topographic boundary cell.

The <BoundaryCell> element in listing 6 is semantically classified as *wall* entity through its <class> property. The primal space representation is encoded using a <SpaceBoundary> whose nested <geometry> and <topology> elements reuse the corresponding curve geometry respectively 1-cell defined in the context of R_1 (cf. listing 2). In dual topology space, the <Transition> element then connects the dual nodes of R_1 and S_{out}^{Topo} through a <gml:Edge>. An additional Euclidean space embedding is not provided for this dual edge. However, assume the dual edges between R_1 and C as well as between R_2 and C shall be associated with space curves in order to create a geometric network which, for example, facilitates shortest path searches or allows for providing precise traveling paths for mobile robots. A possible geometric embedding of both dual edges is illustrated in figure 239 (cf. chapter 3.1.2.3). Listing 7 illustrates the corresponding XML encoding. Note that the <gml:LineString> primitives consistently reference the geometric embedding of the dual nodes being incident to the dual edges.

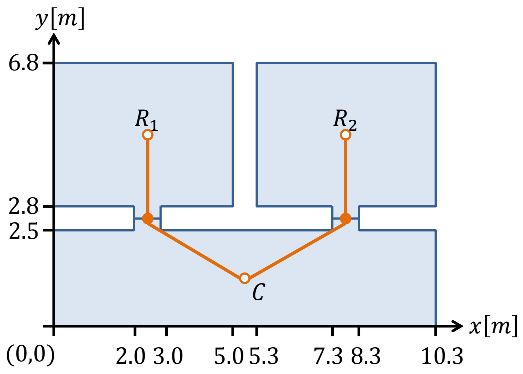


Figure 239: Possible Euclidean space embedding of dual edges between R_1 and C as well as between R_2 and C.

```
<BoundaryCell gml:id="R1-C">
 <gml:description>Door between R1 and C</gml:description>
 <class>door</class>
 <primalSpace> … </primalSpace>
 <dualSpace>
    <Transition gml:id="R1-C_DS">
       <geometry>
```

```
        <gml:LineString gml:id="R1-C_DS_Geometry" srsDimension="2">
          <gml:pointProperty xlink:href="#R1_DS_Geometry"/>
          <gml:pos>2.5 2.65</gml:pos>
          <gml:pointProperty xlink:href="#C_DS_Geometry"/>
        </gml:LineString>
      </geometry>
      <topology>
        <gml:Edge gml:id="R1-C_DS_Topology">
          <gml:directedNode xlink:href="#R1_DS_Topology"/>
          <gml:directedNode xlink:href="#C_DS_Topology"/>
          <gml:curveProperty xlink:href="#R1-C_DS_Geometry"/>
        </gml:Edge>
      </topology>
    </Transition>
 </dualSpace>
</BoundaryCell>
…
<BoundaryCell gml:id="R2-C">
 <gml:description>Door between R2 and C</gml:description>
 <class>door</class>
 <primalSpace> … </primalSpace>
 <dualSpace>
    <Transition gml:id="R2-C_DS">
      <geometry>
        <gml:LineString gml:id="R2-C_DS_Geometry" srsDimension="2">
          <gml:pointProperty xlink:href="R2_DS_Geometry"/>
          <gml:pos>7.8 2.65</gml:pos>
          <gml:pointProperty xlink:href="#C_DS_Geometry"/>
        </gml:LineString>
      </geometry>
      <topology>
        <gml:Edge gml:id="R2-C_DS_Topology">
          <gml:directedNode xlink:href="#R2_DS_Topology"/>
          <gml:directedNode xlink:href="#C_DS_Topology"/>
          <gml:curveProperty xlink:href="#R2-C_DS_Geometry"/>
        </gml:Edge>
      </topology>
    </Transition>
 </dualSpace>
</BoundaryCell>
```

Listing 7: XML encoding of the dual space representation of two boundary cells.

In the above listings, the relationship between a *SpaceCell* and its *BoundaryCell* features can be deduced from the geometric-topological descriptions of their associated *Space* and *SpaceBoundary* entities in primal space. This relationship can however also be expressed explicitly on the level of the semantic entities using the <boundary> and <coBoundary> properties of <Space> respectively <SpaceBoundary>. The following listing 8 demonstrates this for the space cell R_1 and the boundary cell between R_1 and C. Note that the *boundary* and *coBoundary* associations can even be used independently of an accompanying spatial representation (cf. chapter 4.4.1.1). An MLSEM instance document may therefore only contain semantic links between space cells and boundary cells (e.g., in case the spatial characteristics of the indoor features are not available or have not been acquired yet). In dual space, the <State> and <Transition> elements of space cells and boundary cells can equally denote their topological relationships using the semantic <boundary> and <coBoundary> properties.

```
<SpaceCell gml:id="R1">
 <primalSpace>
    <Space gml:id="R1_PS">
      <geometry> <!-- cf. listing 2 --> </geometry>
      <topology> <!-- cf. listing 2 -->  </topology>
      <boundary xlink:href="#R1-Topo_Outer_PS"/>
      <boundary xlink:href="#R1-C_PS"/>
    </Space>
 </primalSpace>
 <dualSpace>
    <State gml:id="R1_DS">
      <geometry> <!-- cf. listing 3--> </geometry>
```

```
     <topology> <!-- cf. listing 3--> </topology>
     <coBoundary xlink:href="#R1-C_DS"/>
     <coBoundary xlink:href="#R1-Topo_Outer_DS"/>
   </State>
 </dualSpace>
</SpaceCell>
…
<BoundaryCell gml:id="R1-C">
 <primalSpace>
   <SpaceBoundary gml:id="R1-C_PS">
     <geometry> <!-- cf. listing 7 --> </geometry>
     <topology> <!-- cf. listing 7 --> </topology>
     <coBoundary xlink:href="#R1_PS"/>
     <coBoundary xlink:href="#C_PS"/>
   </SpaceBoundary>
 </primalSpace>
 <dualSpace>
   <Transition gml:id="R1-C_DS">
     <geometry> <!-- cf. listing 7 --> </geometry>
     <topology> <!-- cf. listing 7 --> </topology>
     <boundary xlink:href="#R1_DS"/>
     <boundary xlink:href="#C_DS"/>
   </Transition>
 </dualSpace>
</BoundaryCell>
```

Listing 8: XML encoding of boundary and coboundary relations.

XML encoding of the intra-layer graph. The intra-layer graph of a space layer is encoded using the nested element <IntraLayerGraph>. For the topographic space layer elaborated in the above sections, it suffices to simply put XLink pointers to the <State> and <Transition> elements defined for its space cells and boundary cells in order to realize the graph structure presented in figure 238 and figure 239. The 1-dimensional geometric respectively topological complex constituted by the intra-layer graph can either be implicitly queried from the dual space representations of the *State* and *Transition* features being referenced through the <node> and <edge> properties or explicitly provided using the <geometry> and <topology> properties. The latter is exemplified in listing 9 for the dual space topology. The *TP_Complex* of ISO 19107 is translated into the GML construct <gml:TopoComplex>. Since, in this example, the topological primitives in dual space have already been defined in the context of the <State> and <Transition> elements, the members of the <gml:TopoComplex> are again denoted using XLink references. In an alternative structuring of an MLSEM instance document, the topological nodes and edges may be encoded within the <gml:TopoComplex> of the intra-layer graph instead and then be referenced from within the <State> and <Transition> elements.[73]

```
<SpaceLayer gml:id="TopographicSpace">
 <type>topographic</type>
 <spaceCell> <!-- cf. previous listings --> </spaceCell>
 …
 <boundaryCell> <!-- cf. previous listings --> </boundaryCell>
 …
 <graph>
   <IntraLayerGraph gml:id="ILG_Topo">
     <topology>
       <gml:TopoComplex gml:id="ILG_Topo_Topology">
         <gml:maximalComplex xlink:href="#ILG_Topo_Topology"/>
         <gml:topoPrimitiveMember xlink:href="#R1_DS_Topology"/>
         <gml:topoPrimitiveMember xlink:href="#R2_DS_Topology"/>
         <gml:topoPrimitiveMember xlink:href="#C_DS_Topology"/>
         <gml:topoPrimitiveMember xlink:href="#Topo_Outer_DS_Topology"/>
```

[73] Remember from the conceptual data model of the MLSEM that the *SpaceLayer* feature type also provides the spatial properties *geometry* and *topology* in order to capture the geometric respectively topological configuration of primitives in primal space (cf. figure 170 in chapter 4.4.1.1). The encoding of the corresponding *GM_Complex* and *TP_Complex* objects for the example topographic space layer has however been omitted in the provided XML listings for brevity. It nevertheless follows the same principles as discussed for the dual space complexes of the *IntraLayerGraph* feature. Thus, either the <gml:GeometricComplex> and <gml:TopoComplex> populating the <geometry> and <topology> properties of a <SpaceLayer> provide XLink references to the primal space descriptions of the <Space> and <SpaceBoundary> elements or vice versa.

```
          <gml:topoPrimitiveMember xlink:href="#R1-Topo_Outer_DS_Topology"/>
          <gml:topoPrimitiveMember xlink:href="#R2-Topo_Outer_DS_Topology"/>
          <gml:topoPrimitiveMember xlink:href="#C-Topo_Outer1_DS_Topology"/>
          <gml:topoPrimitiveMember xlink:href="#C-Topo_Outer2_DS_Topology"/>
          <gml:topoPrimitiveMember xlink:href="#R1-C_DS_Topology"/>
          <gml:topoPrimitiveMember xlink:href="#R2-C_DS_Topology"/>
        </gml:TopoComplex>
      </topology>
      <node xlink:href="#R1_DS"/>
      <node xlink:href="#R2_DS"/>
      <node xlink:href="#C_DS"/>
      <node xlink:href="#Topo_Outer_DS"/>
      <edge xlink:href="#R1-Topo_Outer_DS"/>
      <edge xlink:href="#R2-Topo_Outer_DS"/>
      <edge xlink:href="#C-Topo_Outer1_DS"/>
      <edge xlink:href="#C-Topo_Outer2_DS"/>
      <edge xlink:href="#R1-C_DS"/>
      <edge xlink:href="#R2-C_DS"/>
    </IntraLayerGraph>
  </graph>
</SpaceLayer>
```

Listing 9: XML encoding of the intra-layer graph of the topographic space layer. The <gml:TopoComplex> references itself as maximal complex.

The XML representation of the Wi-Fi sensor space layer (cf. figure 237) applies the same schema as discussed for the topographic space view and hence is not presented in detail in the following. The listing 10 only shows an excerpt containing the encoding of the Wi-Fi space cell A, its boundary cell separating A from the outer space, and the resulting intra-layer graph. Note that the extent of the Wi-Fi space cell is approximated by a polygon geometry with straight line segments and that the geometry is encoded in the same local coordinate system as the topographic space cells in this example (cf. figure 238). As discussed above, this is however not mandatory but a different spatial reference system could be used instead. In this case, coordinate transformations between the involved reference systems are required (e.g., to derive inter-layer edges).

```
<SpaceLayer gml:id="WiFi-SensorSpace">
  <type>sensor</type>
  <spaceCell>
    <SpaceCell gml:id="A">
      <symbolicId>WiFi access point A</symbolicId>
      <symbolicId codeSpace="MAC">00-24-E8-33-61-F1</symbolicId>
      <class>WiFi signal coverage area</class>
      <primalSpace>
        <Space gml:id="A_PS">
          <geometry>
            <gml:Polygon gml:id="A_PS_Geometry" srsDimension="2">
              <gml:exterior>
                <gml:Ring>
                  <gml:curveMember>
                    <gml:LineString gml:id="A_PS_Geometry_1"> <!-- common boundary with the outer space -->
                      <gml:posList>4.90604614654405 4.46202627848359 4.86284695553131 4.62324785418903
                        4.18844815527895 5.42696504723101 3.27983310965741 5.95155418841035
                        2.24659420289855 6.13374208518965 1.2133552961397 5.95155418841035
                        0.304740250518156 5.42696504723101 -0.369658549734208 4.62324785418903
                        -0.774394259102692 3.11275362318841 -0.369658549734206 1.60225939218778
                        0.304740250518158 0.798542199145804 1.21335529613969 0.273953057966467
                        2.24659420289855 0.091765161187163 3.75708843389917 0.496500870555649
                        4.24371757974206 0.904831207352533</gml:posList>
                    </gml:LineString>
                  </gml:curveMember>
                  <gml:curveMember>
                    <gml:LineString gml:id="A_PS_Geometry_2"> <!-- common boundary with the space cell AB -->
                      <gml:posList>4.24371757974206 0.904831207352533 3.92318130156336
                        2.16384990991809 4.38026406197645 3.76369410784373 4.90604614654405
                        4.46202627848359</gml:posList>
                    </gml:LineString>
                  </gml:curveMember>
```

```
                     </gml:Ring>
                  </gml:exterior>
               </gml:Polygon>
            </geometry>
            <topology>
               <gml:Face gml:id="A_PS_Topology">
                  <gml:directedEdge>
                     <gml:Edge gml:id="A_PS_Topology_1">
                        <gml:directedNode>
                           <gml:Node gml:id="A_PS_Topology_N1"/>
                        </gml:directedNode>
                        <gml:directedNode>
                           <gml:Node gml:id="A_PS_Topology_N2"/>
                        </gml:directedNode>
                        <gml:curveProperty xlink:href="#A_PS_Geometry_2"/>
                     </gml:Edge>
                  </gml:directedEdge>
                  <gml:directedEdge>
                     <gml:Edge gml:id="A_PS_Topology_2">
                        <gml:directedNode xlink:href="#A_PS_Topology_N2"/>
                        <gml:directedNode xlink:href="#A_PS_Topology_N1"/>
                        <gml:curveProperty xlink:href="#A_PS_Geometry_1"/>
                     </gml:Edge>
                  </gml:directedEdge>
                  <gml:surfaceProperty xlink:href="#A_PS_Geometry"/>
               </gml:Face>
            </topology>
            <boundary xlink:href="#A-WiFi_Outer_PS"/>
            <boundary xlink:href="#A-AB_PS"/>
         </Space>
      </primalSpace>
      <dualSpace>
         <State gml:id="A_DS">
            <topology>
               <gml:Node gml:id="A_DS_Topology"/>
            </topology>
         </State>
      </dualSpace>
   </SpaceCell>
</spaceCell>
…
<boundaryCell>
   <BoundaryCell gml:id="A-WiFi_Outer">
      <primalSpace>
         <SpaceBoundary gml:id="A-WiFi_Outer_PS">
            <geometry xlink:href="#A_PS_Geometry_1"/>
            <topology xlink:href="#A_PS_Topology_2"/>
         </SpaceBoundary>
      </primalSpace>
      <dualSpace>
         <Transition gml:id="A-WiFi_Outer_DS">
            <topology>
               <gml:Edge gml:id="A-WiFi_Outer_DS_Topology">
                  <gml:directedNode xlink:href="#A_DS_Topology"/>
                  <gml:directedNode xlink:href="#WiFi_Outer_DS_Topology"/>
               </gml:Edge>
            </topology>
            <boundary xlink:href="#A_DS"/>
            <boundary xlink:href="#WiFi_Outer_DS"/>
         </Transition>
      </dualSpace>
   </BoundaryCell>
</boundaryCell>
…
<graph>
   <IntraLayerGraph gml:id="ILG_WiFi">
      <topology> … </topology>
      <node xlink:href="#A_DS"/>
      <node xlink:href="#B_DS"/>
      <node xlink:href="#AB_DS"/>
      <node xlink:href="#WiFi_Outer_DS"/>
      <edge xlink:href="#A-WiFi_Outer_DS"/>
      <edge xlink:href="#B-WiFi_Outer_DS"/>
      <edge xlink:href="#A-AB_DS"/>
      <edge xlink:href="#B-AB_DS"/>
    </IntraLayerGraph>
  </graph>
</SpaceLayer>
```

Listing 10: Excerpt of the XML encoding of the Wi-Fi sensor space layer.

XML encoding of the multilayered graph. The multilayered graph of a space layer complex is captured by the <MultilayeredGraph> subelement of <SpaceLayerComplex> within an MLSEM instance document. Extracts of its encoding for the above example (cf. right of figure 237) are presented in listing 11. Note that the <MultilayeredGraph> embeds the intra-layer graphs of the space layers participating in the space layer complex through the <subGraph> property by using XLink references. The inter-layer edges between two dual nodes from different space layers are then listed as <edge> properties of the <MultilayeredGraph> element and are encoded through <InterLayerEdge> elements. A mandatory attribute of every inter-layer edge is <topoRelation> which encodes the topological relationship between the primal space representations of the linked space cells. The <boundary> property then provides XLink references to the <State> elements implementing the dual space representation of the space cells being connected through the inter-layer edge.

Listing 11 exemplifies the XML encoding of the inter-layer edges between the space cells R_1 and A, C and A as well as S_{out}^{Topo} and S_{out}^{Wi-Fi}. In all cases, the topological relationship between the space cells from topographic and sensor space is *overlap*.

```
<MultilayeredGraph gml:id="MLG">
  <subGraph xlink:href="ILG_Topo"/>
  <subGraph xlink:href="ILG_WiFi"/>
  <edge>
    <InterLayerEdge gml:id="ILE_R1-A">
      <topoRelation>overlap</topoRelation>
      <boundary xlink:href="#R1_DS"/>
      <boundary xlink:href="#A_DS"/>
    </InterLayerEdge>
  </edge>
  <edge>
    <InterLayerEdge gml:id="ILE_C-A">
      <topoRelation>overlap</topoRelation>
      <boundary xlink:href="#C_DS"/>
      <boundary xlink:href="#A_DS"/>
    </InterLayerEdge>
  </edge>
  …
  <edge>
    <InterLayerEdge gml:id="ILE_WiFi_Outer-Topo_Outer">
      <topoRelation>overlap</topoRelation>
      <boundary xlink:href="#WiFi_Outer_DS"/>
      <boundary xlink:href="#Topo_Outer_DS"/>
    </InterLayerEdge>
  </edge>
</MultilayeredGraph>
```

Listing 11: XML encoding of the resulting multilayered graph structure.

XML encoding of joint states and joint state transitions. For localization purposes, the joint states within a space layer complex as well as their transitions have to be reflected in the MLSEM instance document. As illustrated in listing 1, the <stateSpace> property of a <SpaceLayerComplex> is used to associate a space layer complex with a <JointStateSpace> element that carries this information. The joint states themselves are hereby mapped by <JointState> subelements. The clique of dual nodes from the multilayered graph which forms a single joint state is provided through <node> properties of <JointState> with each <node> pointing to a <State> element from a different space layer participating in the space layer complex. The non-empty intersection of the primal space geometries of the associated space cells is denoted using the mandatory <uncertaintyRegion> property of <JointState>. Since the intersection geometry is not guaranteed to be a connected space, it is encoded as geometric complex. The following figure 240 illustrates two joint states given by the pairs (R_1, A) and (C, A) as hatched areas within the example indoor setting. The corresponding XML encoding is presented in listing 12.

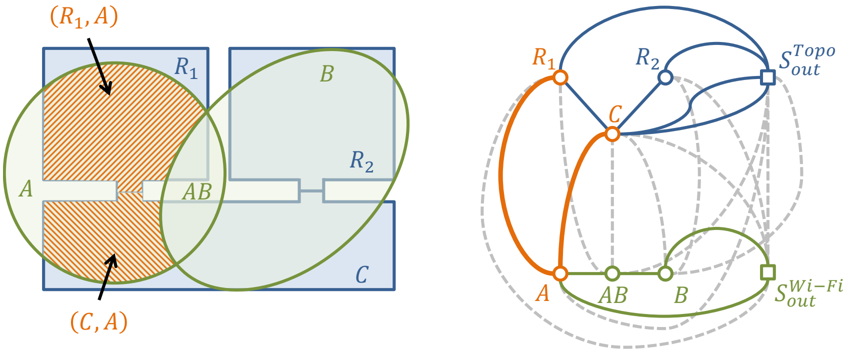


Figure 240: Two example joint states (R_1, A) and (C, A).

```
<SpaceLayerComplex>
 <layer> <!-- cf. previous listings --> </layer>
 …
 <graph> <!-- cf. listing 11 --> </graph>
 …
 <stateSpace>
   <JointStateSpace gml:id="JointStateSpace">
     <state>
       <JointState gml:id="JS_R1-A">
         <node xlink:href="#R1_DS"/>
         <node xlink:href="#A_DS"/>
         <uncertaintyRegion>
           <gml:GeometricComplex gml:id="JS_R1-A_Geometry" srsDimension="2">
             <gml:element>
               <gml:Polygon gml:id="JS_R1-A_Geometry_1">
                 <gml:exterior>
                   <gml:LinearRing>
                     <gml:posList>0 5.06378975886794 0.304740250518156 5.42696504723101 1.2133552961397
                       5.95155418841035 2.24659420289855 6.13374208518965 3.27983310965741
                       5.95155418841035 4.18844815527895 5.42696504723101 4.86284695553131
                       4.62324785418903 4.90604614654405 4.46202627848359 4.38026406197645
                       3.76369410784373 4.10493227430726 2.8 3 2.8 3 2.65 2 2.65 2 2.8 0 2.8 0
                       5.06378975886794</gml:posList>
                   </gml:LinearRing>
                 </gml:exterior>
               </gml:Polygon>
             </gml:element>
           </gml:GeometricComplex>
         </uncertaintyRegion>
       </JointState>
     </state>
     <state>
       <JointState gml:id="JS_C-A">
         <node xlink:href="#C_DS"/>
         <node xlink:href="#A_DS"/>
         <uncertaintyRegion>
           <gml:GeometricComplex gml:id="JS_C-A_Geometry" srsDimension="2">
             <gml:element>
               <gml:Polygon gml:id="JS_C-A_Geometry_1">
                 <gml:exterior>
                   <gml:LinearRing>
                     <gml:posList>3 2.65 3 2.5 4.01922091047463 2.5 3.92318130156336 2.16384990991809
                       4.24371757974206 0.904831207352533 3.75708843389917 0.496500870555649
                       2.24659420289855 0.091765161187163 1.21335529613969 0.273953057966467
                       0.304740250518158 0.798542199145804 0 1.16171748750888 0 2.5 2 2.5 2 2.65 3 2.65
                     </gml:posList>
                   </gml:LinearRing>
                 </gml:exterior>
               </gml:Polygon>
             </gml:element>
           </gml:GeometricComplex>
         </uncertaintyRegion>
       </JointState>
     </state>
   </JointStateSpace>
 </stateSpace>
</SpaceLayerComplex>
```

Listing 12: XML encoding of the two joint states from figure 240 including their uncertainty region.

Since the intersection geometries of the illustrated joint states (R_1, A) and (C, A) are topologically adjacent, there exists a joint state transition between both. This transition is represented as <JointStateTransition> element which is attached to a <JointStateSpace> through the <transition> property. A <JointStateTransition> merely denotes the pair of joint states on its <boundary> and utilizes XLink references for this purpose. The XML encoding of the transition between (R_1, A) and (C, A) is shown below. Note that the modelling of a finite-joint-state machine is optional. Although the GML application schema of the MLSEM supports the storage and exchange of finite-joint-state machines, the corresponding XML representation is not further discussed in this example.

```
<JointStateSpace gml:id="JointStateSpace">
 <state> <!-- cf. listing 12 --> </state>
 …
 <transition>
   <JointStateTransition gml:id="JST_1">
     <boundary xlink:href="#JS_R1-A"/>
     <boundary xlink:href="#JS_C-A"/>
   </JointStateTransition>
 </transition>
</JointStateSpace>
```

Listing 13: XML encoding of the joint state transition between (R_1, A) and (C, A).

XML encoding of navigation constraints. Assume the topographic space entities of the example indoor setting shall be enriched with navigation constraints. For example, the *BoundaryCell* features representing the doors between R_1 and C as well as R_2 and C may be assigned a *PassableConstraint* with a *SpatialProfile* condition providing a geometric profile the navigation user must be able to physically fit through in order to traverse the doors. All remaining boundary cells could receive an unconditioned *NonPassableConstraint* in order to rule out paths along their dual edges. Whereas the UML instance diagrams expressing these simple navigation constraints are shown in figure 241, the XML mapping is exemplified in listing 14 for the constraints on the wall separating R_1 from the outer space and on the door between R_1 and C.

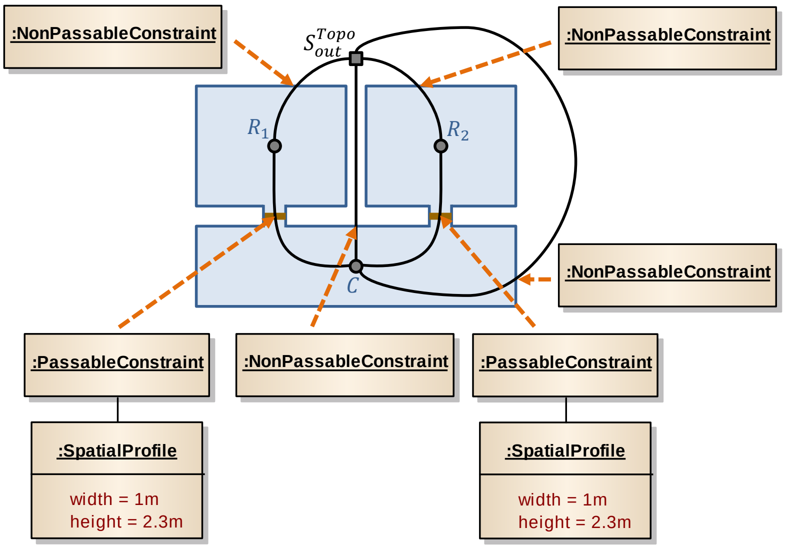


Figure 241: Passable and non-passable navigation constraints associated with the walls and doors.

```
<BoundaryCell gml:id="R1-Topo_Outer">
 <gml:description>Wall between R1 and Outer</gml:description>
 <class>wall</class>
 <constraint>
    <NonPassableConstraint gml:id="R1-Topo_Outer_NonPassableConstraint"/>
 </constraint>
 <primalSpace> <!-- cf. listing 6 --> </primalSpace>
 <dualSpace> <!-- cf. listing 6 --> </dualSpace>
</BoundaryCell>
…
<BoundaryCell gml:id="R1-C">
 <gml:description>Door between R1 and C</gml:description>
 <class>door</class>
 <constraint>
    <PassableConstraint gml:id="R1-C_PassableConstraint">
       <condition>
          <SpatialProfile gml:id="R1-C_SpatialProfile">
             <width uom="m">1</width>
             <height uom="m">2.3</height>
          </SpatialProfile>
       </condition>
    </PassableConstraint>
 </constraint>
 <primalSpace> <!-- cf. listing 7 --> </primalSpace>
 <dualSpace> <!-- cf. listing 7 --> </dualSpace>
</BoundaryCell>
```

Listing 14: XML encoding of the simple navigation constraints from figure 241.

As can be seen from the above listing, the navigation constraints are given as subelements of <BoundaryCell>. The unconditioned <NonPassableConstraint> is simply realized as empty XML element. The <SpatialProfile> condition of the <PassableConstraint> is provided as nested element through the <condition> property. In this example, the <SpatialProfile> denotes the <width> and <height> of the door using metres as unit of measure. If a navigation user exceeds these restrictions then the door renders a movement obstruction for this user.

In order to illustrate a combined navigation constraint, suppose that the room R_1 shall only be accessible by staff members or in case of an evacuation (cf. figure 242). In the following XML excerpt of the <SpaceCell> representing R_1, the required *UserGroup* and *NavigationScenario* conditions are captured by corresponding XML elements and are given as <operand>s of a <CombinedConstraintCondition> which expresses the logical disjunction through its <operator> property. The <PermissibleConstraint> embracing the combined constraint condition is necessarily provided as subelement of the <SpaceCell>.

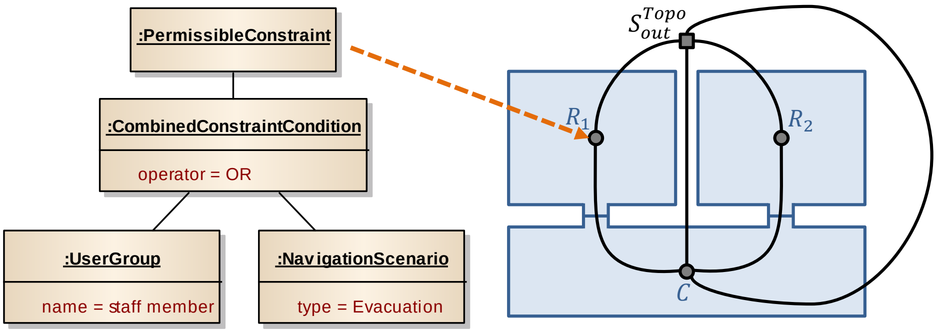


Figure 242: Example of a combined navigation constraint associated with the room cell R_1.

```
<SpaceCell gml:id="R1">
 <symbolicId codeSpace="roomNo">5126</symbolicId>
 <class>office</class>
 <constraint>
    <PermissibleConstraint gml:id="R1_PermissibleConstraint">
       <condition>
          <CombinedConstraintCondition gml:id="R1_StaffMember_Or_Evacuation">
             <operator>OR</operator>
          <operand>
            <UserGroup gml:id="R1_UserGroup">
              <name>staff member</name>
            </UserGroup>
          </operand>
          <operand>
            <NavigationScenario gml:id="R1_Scenario">
              <type>Evacuation</type>
            </NavigationScenario>
          </operand>
        </CombinedConstraintCondition>
      </condition>
    </PermissibleConstraint>
  </constraint>
  <primalSpace> <!-- cf. listing 2 --> </primalSpace>
  <dualSpace> <!-- cf. listing 3 --> </dualSpace>
</SpaceCell>
```

Listing 15: XML encoding of the combined navigation constraint from figure 242.

A final example for the encoding of navigation constraints is shown in listing 16 and demonstrates the XML representation of a *SpatialManeuver* restriction associated with a *SpaceElementSequence* (cf. figure 243). A *PassableConstraint* is used to state that only those navigation users can traverse between the rooms R_1 and R_2 who are capable of performing a horizontal 180 degree turn on the corridor C within the spatial area between both doors (e.g., driving users such as wheelchair user or mobile robots). This *PassableConstraint* is attached to the undirected sequence $\{R_1, C, R_2\}$. Note that this set is sparse in that it only contains the space cells but not the intermediate boundary cells.

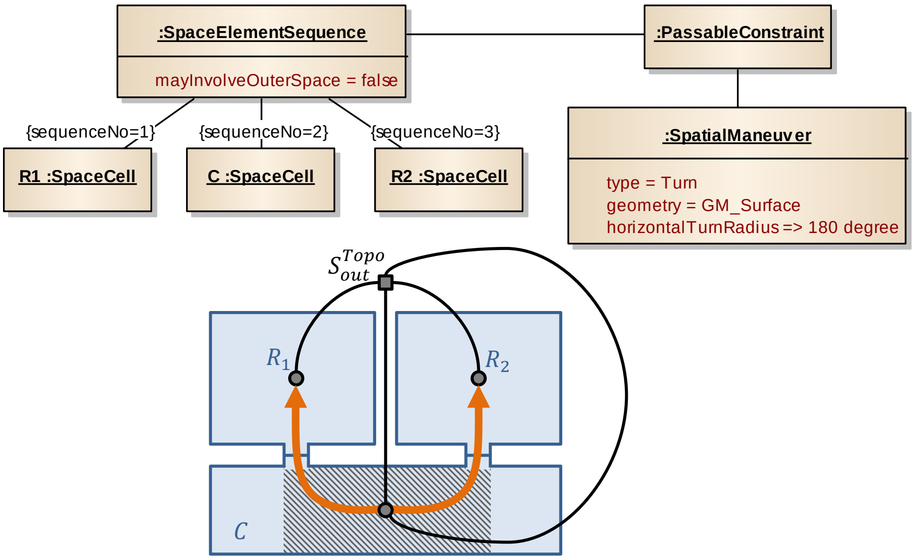


Figure 243: Example spatial maneuver restriction associated with a sequence of topographic space cells.

```
<SpaceLayer gml:id="TopographicSpace">
  <type>topographic</type>
  <spaceCell> <!-- cf. previous listings --> </spaceCell>
  …
  <boundaryCell> <!-- cf. previous listings --> </boundaryCell>
  …
  <graph> <!-- cf. listing 9 --> </graph>
  …
  <sequence>
    <SpaceElementSequence gml:id="Seq_R1-C-R2">
      <mayInvolveOuterSpace>false</mayInvolveOuterSpace>
      <element sequenceNo="1" xlink:href="#R1"/>
      <element sequenceNo="2" xlink:href="#C"/>
      <element sequenceNo="3" xlink:href="#R2"/>
      <constraint>
```

```
       <PassableConstraint gml:id="Seq_R1-C-R2_PassableConstraint">
         <condition>
           <SpatialManeuver gml:id="Seq_R1-C-R2_Maneuver">
             <type>Turn</type>
             <geometry>
               <gml:Polygon gml:id="eq_R1-C-R2_Maneuver_Geometry">
                 <gml:exterior>
                   <gml:LinearRing>
                     <gml:posList>2 2.5 2 0 8.3 0 8.3 2.5 2 2.5</gml:posList>
                   </gml:LinearRing>
                 </gml:exterior>
               </gml:Polygon>
             </geometry>
             <horizontalTurnRadius>
               <uom>degree</uom>
               <lowerLimit>180</lowerLimit>
               <includeLowerLimit>true</includeLowerLimit>
             </horizontalTurnRadius>
           </SpatialManeuver>
         </condition>
       </PassableConstraint>
     </constraint>
   </SpaceElementSequence>
 </sequence>
 …
</SpaceLayer>
```

Listing 16: XML encoding of the spatial maneuver restriction from figure 243.

Since every *SpaceElementSequence* is restricted to contain *SpaceElement* features from the same space layer (cf. chapter 4.4.1.4), its XML counterpart <SpaceElementSequence> is modelled inline the corresponding <SpaceLayer> (in contrast, <SpaceElementGroup>s are encoded as child elements of a <SpaceLayerComplex>). The elements of the set generating the sequence are provided through the <element> property. In the above example, XLink references to the <SpaceCell> encodings of R_1, C, and R_2 are used for this purpose. Moreover, each <element> has a mandatory *sequenceNo* attribute to denote the order of the elements within the sequence. Note that paths involving the outer space cell do not satisfy this sequence due to the <mayInvolveOuterSpace> property being set to *false*. Similar to the above examples, the navigation constraint is then attached to the sequence via the <constraint> property. The <SpatialManeuver> condition of the <PassableConstraint> provides the <geometry> within which the movement has to be executed as 2-dimensional <gml:Polygon>. The <horizontalTurnRadius> then encodes the minimum turn radius required for traversing the sequence.

XML encoding of source objects. The final example in this chapter illustrates the encoding of references from <SpaceCell> and <BoundaryCell> elements to their equivalences in an external building model (e.g., given in CityGML or IFC) using a *SourceObject* entity. Suppose there exists a 3-dimensional CityGML representation of the example interior built environment stored in the CityGML instance document *CityGML_Building.gml*. The following listing 17 shows a reference for the room R_1 to a corresponding CityGML feature in this document. For this purpose, a <SourceObject> element is associated with the <SpaceCell> of R_1 which contains an <ExternalReference> that points to a CityGML *Room*. The *Room* is identified through its <gml:id> of value *Room_5126* which must be available in the CityGML dataset. A software consuming the MLSEM instance document can hence switch between the 2-dimensional MLSEM representation and the 3-dimensional CityGML representation of R_1.

```
<SpaceCell gml:id="R1">
 <symbolicId codeSpace="roomNo">5126</symbolicId>
 <class>office</class>
 <model>
   <SourceObject gml:id="R1_Source">
     <reference>
       <ExternalReference>
         <externalObject>
           <ExternalObjectReference>
             <uri>file://CityGML_Building.gml#Room_5126</uri>
           </ExternalObjectReference>
         </externalObject>
```

```
        </ExternalReference>
      </reference>
    </SourceObject>
  </model>
  <constraint> <!-- cf. listing 15 --> </constraint>
  <primalSpace> <!-- cf. listing 2 --> </primalSpace>
  <dualSpace> <!-- cf. listing 3 --> </dualSpace>
</SpaceCell>
```

Listing 17: XML encoding of references to external source objects.

7.2 Modelling and Database Storage of an MLSEM Indoor Space Model

The following sections are dedicated to the presentation of the separate process steps involved in the creation of an indoor space model for the *TU Main Building* (cf. figure 244) that were carried out within the Master students' project mentioned in the introduction to this chapter. The process was entirely realized using standard software tools from the GIS domain (e.g., ArcGIS 10.0 and FME 2012). However, the focus of the subsequent discussion is put on the description of the input data and its transformation into an MLSEM conformant representation rather than on the realization of the process steps in software.[74] Note that the depicted process is not meant as a best practice recommendation or guideline for the realization of an MLSEM indoor space model for use in a real navigation system but is to be understood as proof of concept for the applicability of the spatio-semantic modelling concepts of the MLSEM framework. In chapter 7.2.3, the database schema for the storage and management of MLSEM indoor space models within a spatial database is presented in detail.

Figure 244: Main building of the Technische Universität Berlin (front) shown in a 3-dimensional scene of the official city model of the City of Berlin.

7.2.1 Topographic Space Layer

The starting point for the derivation of a 3-dimensional topographic space layer of the interior built-up space of the *TU Main Building* were official 2-dimensional CAD drawings of the separate building floors. The CAD drawings only contained geometric descriptions of the architectural entities. Symbolic and semantic information such as room numbers or the classification of spaces were merely provided as text labels. Figure 245 depicts the CAD model of a single building floor to illustrate the input data.

[74] In fact, after completion of the students' project and whilst writing this thesis, GIS software tools such as ArcGIS and FME have significantly enhanced their capabilities for processing 3-dimensional geo-data. For example, both the latest releases of ArcGIS and FME (ArcGIS 10.1 respectively FME 2013) now offer sophisticated geometric operations as well as topological analyses in up to three dimensions. Although this functionality strongly supports the creation of a 3-dimensional indoor space model that conforms to the MLSEM framework, it could not be considered within the students' project.

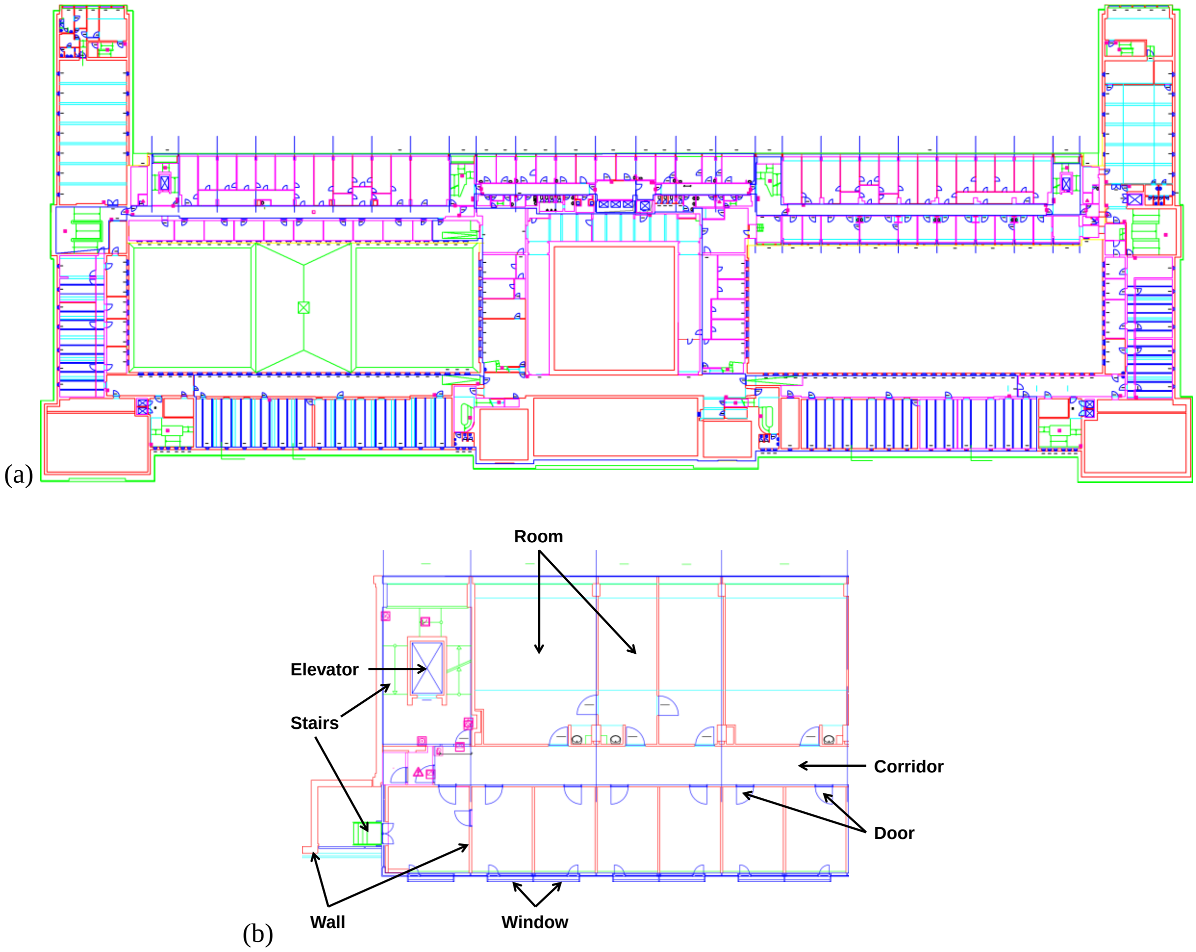
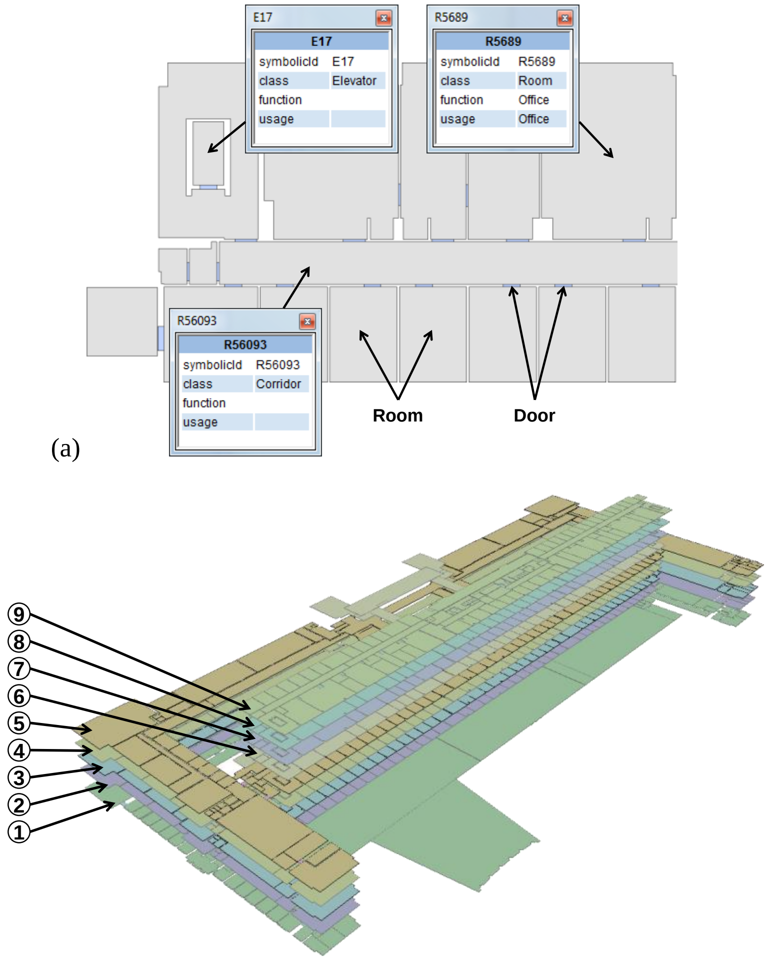


Figure 245: A single CAD floor plan of the TU Main Building (a) and an excerpt of that floor plan illustrating the modelled spatial entities (b).

In a first step, a set of simplified 2-dimensional floor plans was extracted from the CAD input models. Each floor plan was reduced to a representation of the free spaces on the building floor (e.g., rooms, corridors, staircases, and elevator shafts) as well as the doors connecting these spaces. Particular focus was put on the spatial description of the entities in order to ensure geometric shapes to be mutually non-overlapping and doors to touch adjacent rooms at a common 1-dimensional boundary segment. The text labels were translated from the CAD models into semantic attributes of the respective spatial entities of the floor plan. The result of this extraction procedure for the portion of the building floor shown in figure 245b is presented below. Observe that the entities representing the free topographic spaces are disjoint and displaced by the thickness of the separating wall elements. The non-navigable wall space constituting the building skeleton is hence only captured implicitly on the simplified floor plans. The windows were likewise omitted from the topographic space view. Afterwards, the resulting floor plans of the *TU Main Building* were assigned proper height values and stacked on top of each other as depicted in figure 246b.



E17

E17	
symbolicId	E17
class	Elevator
function	
usage	

R5689

R5689	
symbolicId	R5689
class	Room
function	Office
usage	Office

R56093

R56093	
symbolicId	R56093
class	Corridor
function	
usage	

Figure 246: Simplified version of the floor plan excerpt from figure 245b with thematic attributes (a) and all nine simplified floor plans of the TU Main Building stacked on top of each other.

The entities on the floor plans were extruded in a subsequent step according to their real world dimensions which were taken from the CAD input models. This produced 3-dimensional solid geometries for both the free spaces and the doors as sketched in figure 247.[75]

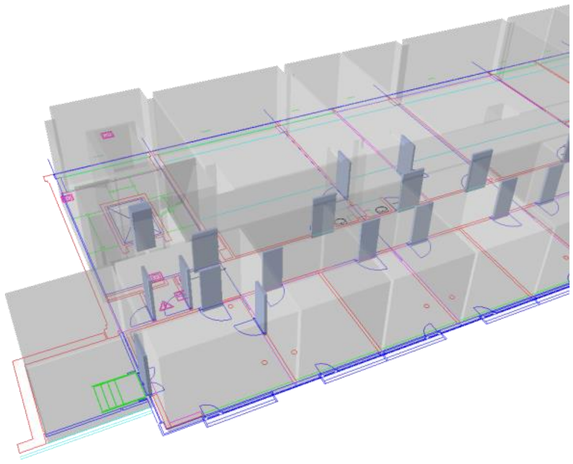


Figure 247: Extrusion of the free spaces and doors on the simplified floor plan from figure 246a.

A CityGML LOD4 model of the *TU Main Building* was created from the extruded floor plans as an intermediate result by mapping the free spaces and doors onto CityGML *Room* respectively *Door* features. Moreover, the 3-

[75] It is immediate that the resulting extrusion geometries are only idealized generalizations of the real world shape of the spatial entities. For the course of the students' project this limitation was however acceptable.

dimensional boundary surfaces of the *Room* features were classified into *InteriorWallSurface*, *CeilingSurface*, *FloorSurface*, and *ClosureSurface* based on simple heuristics. Symbolic and semantic properties of the spatial entities were again preserved in this model transformation and carried to either predefined properties of the CityGML features or to generic CityGML attributes. On the left of figure 248, the resulting LOD4 representation is shown for a small excerpt of a single floor, whereas the entire CityGML building model is presented on the right.

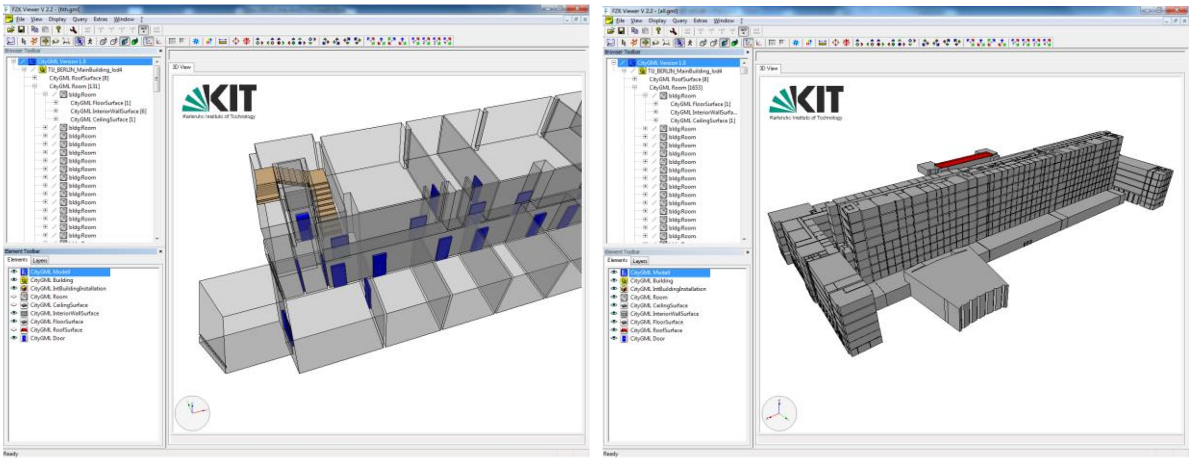


Figure 248: CityGML LOD4 model created from the extruded simplified floor plans.

By choosing CityGML as an intermediate representation of the topographic indoor space, the subsequent steps could be realized using its standardized information model as well-defined interface. This allows the process to be decoupled from the input data of the *TU Main Building*. In contrast, any CityGML LOD4 building model may serve as input for the further steps instead. Moreover, the rules for mapping building models given in CityGML onto a corresponding MLSEM representation as elaborated in chapter 6.1 become applicable.[76] According to these mapping rules, the *Room* and *Door* features were carried to 3-dimensional space cells in order to populate the topographic space layer. Every boundary surface shared by two topologically adjacent space cells (e.g., the surface shared by a room and a door) was correspondingly captured by a 2-dimensional boundary cell. In order to achieve a minimum total number of boundary cells, the remaining boundary part of each space cell against the outer space was described by a single boundary cell.[77] In an alternative mapping, the boundary cells could, for example, also be aligned with the *InteriorWallSurface*, *CeilingSurface*, *FloorSurface*, and *ClosureSurface* features within the CityGML model (cf. chapter 6.1).

As a result, the topographic space layer for the *TU Main Building* was finally inhabited by 3528 space cells (1657 space cells representing free spaces, 1870 door cells, and one outer space cell) with their boundaries being decomposed into 5671 boundary cells. The space layer was stored in a spatial database according to the database schema developed in the students' project (cf. chapter 7.2.3). By this means, the space cells and boundary cells could be easily accessed and further processed using various GIS software tools. Figure 249 shows the resulting topographic space layer and its space cells in primal space. Moreover, space cells representing the doors (figure 249b), staircases and elevator shafts (153 space cells, figure 249c), and offices (892 space cells, figure 249d), are highlighted in separate parts of this figure in order to illustrate different classifications of topographic space cells that were included in the indoor space model as semantic attributes.

[76] Obviously, also IFC is a suitable candidate for specifying this well-defined interface. In case the topographic space layer is to be described in two dimensions only, the ESRI BISDM model renders a further alternative.

[77] In the context of the students' project, every boundary surface shared by two adjacent topographic space cells could be guaranteed to be homeomorphic to $\overline{\mathbb{B}}^2$ in primal space due to the applied extrusion operation. From this it follows that the remaining boundary part of a space cell shared with the outer space could be described by precisely one boundary cell (cf. chapter 3.1.3 for a comprehensive discussion). In a more general case, it has to be evaluated whether the intersection geometry of two adjacent space cells in primal space admits a mapping onto a single boundary cell.

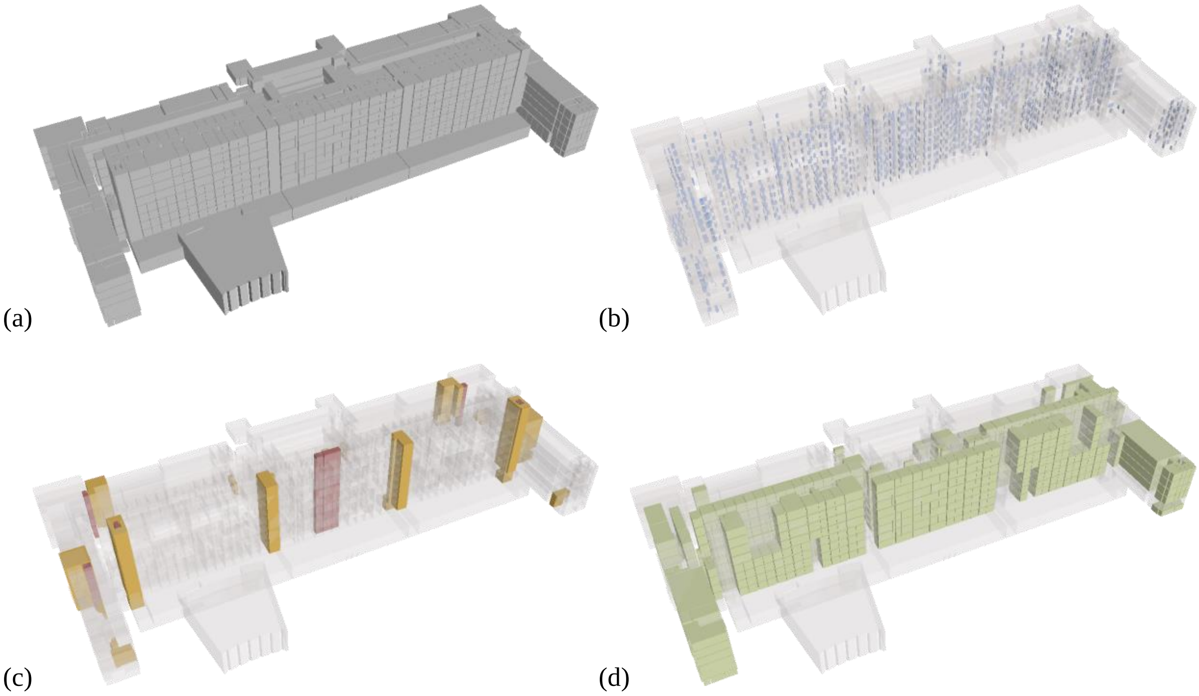


Figure 249: 3-dimensional topographic space layer of the TU Main Building (a), space cells representing doors (b), space cells representing stairs and elevator shafts (c), and space cells representing offices (d).

Since the building skeleton was only implicitly represented on the derived topographic space layer as discussed above, the space cells representing rooms or corridors are not only displaced by the thickness of walls on a single building floor but also by the thickness of slab elements on subsequent floors. In contrast, vertical transition spaces (e.g., staircases and elevator shafts) may obviously not be disconnected between floors. For example, figure 250 shows a part of a staircase joining the nine floors of the *TU Main Building*. Within the students' project, the vertical transition spaces were decomposed into a separate space cell on each floor. The space cells on neighbouring floors were hence ensured to touch at a common (but virtual) boundary. Alternatively, and as discussed in example 3.61, the entire transition space could also be captured by a single space cell.

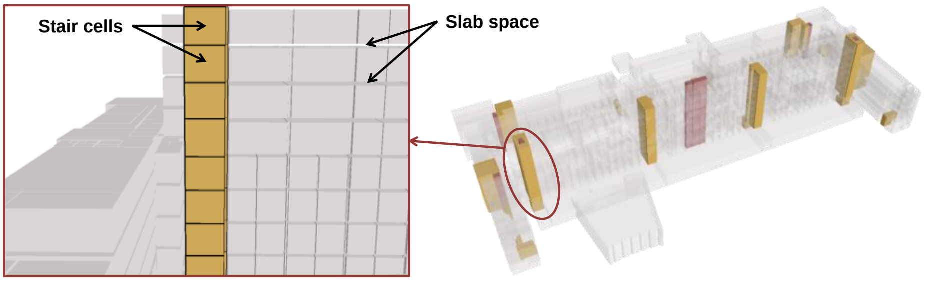


Figure 250: Space cells representing stairs and elevators were ensured to touch over subsequent building floors.

The intra-layer graph structure of the topographic space layer simply followed from pairing the primal space representation of every space cell and boundary cell with a node respectively edge in dual space as defined by the mathematical model of the MLSEM. The Euclidean space embedding of the intra-layer graph was realized by connecting the centroids of the primal space geometries of space cells through straight line segments. The weak intra-layer graph neglecting the dual node of the outer space cell and its incident dual edges is depicted in figure 251. Due to the one-to-one correspondence between the primal and dual space representations, the intra-layer graph was constituted by 3528 dual nodes and 5671 dual edges, whereas its weak form shown below only contained 3527 dual nodes and 4014 dual edges.

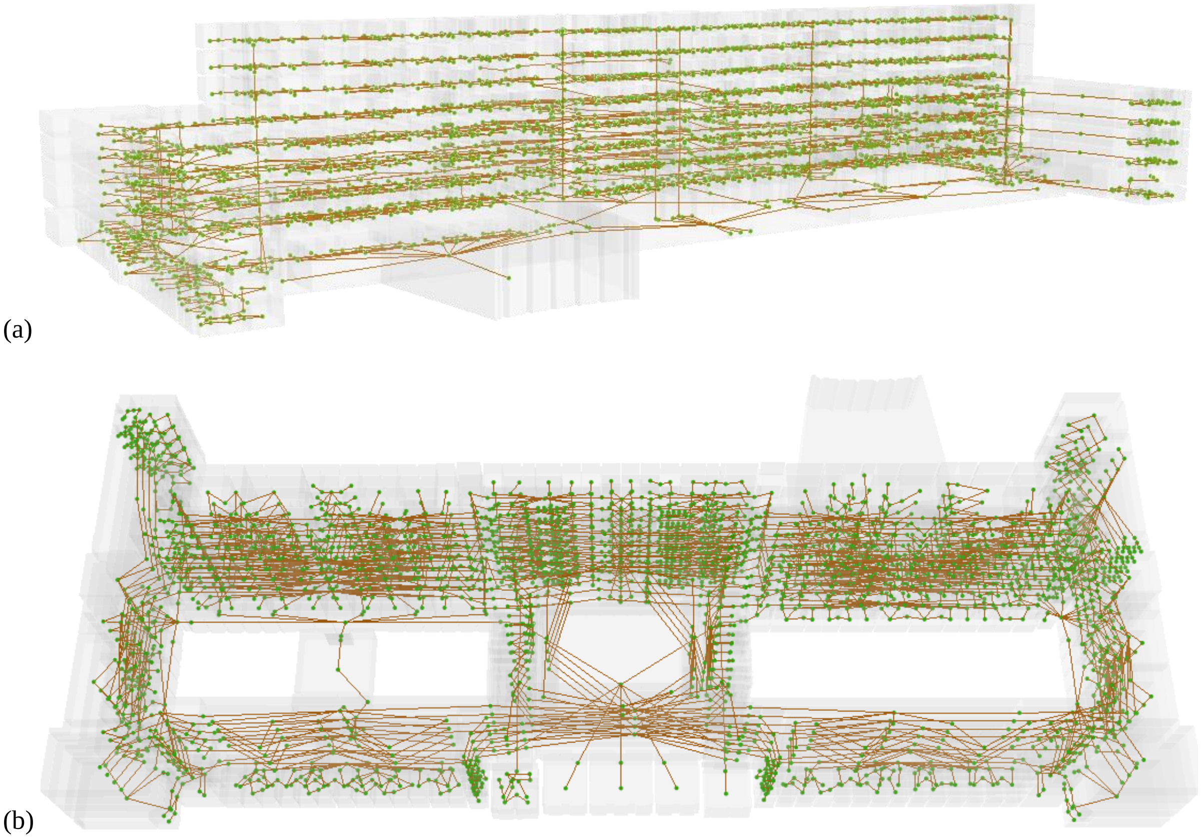


Figure 251: Resulting weak intra-layer graph shown from a front view (a) and a rear view (b).

In figure 252, an excerpt of the weak intra-layer graph for a single building floor is sketched in a 3-dimensional scene and a 2-dimensional top view. As can be seen on the right, the simple approach followed in the students' project results in circuitous spatial embeddings of the intra-layer graph at long corridors and large rooms. Although the topological connectivity information between the space cells is correctly represented, the resulting detours may negatively impact the derivation of shortest paths between two places. In order to derive a geometric network without circuitous embeddings and detours, the space cells have to be subspaced in an appropriate way (cf. example 3.62). This, however, was beyond the scope of the students' project.

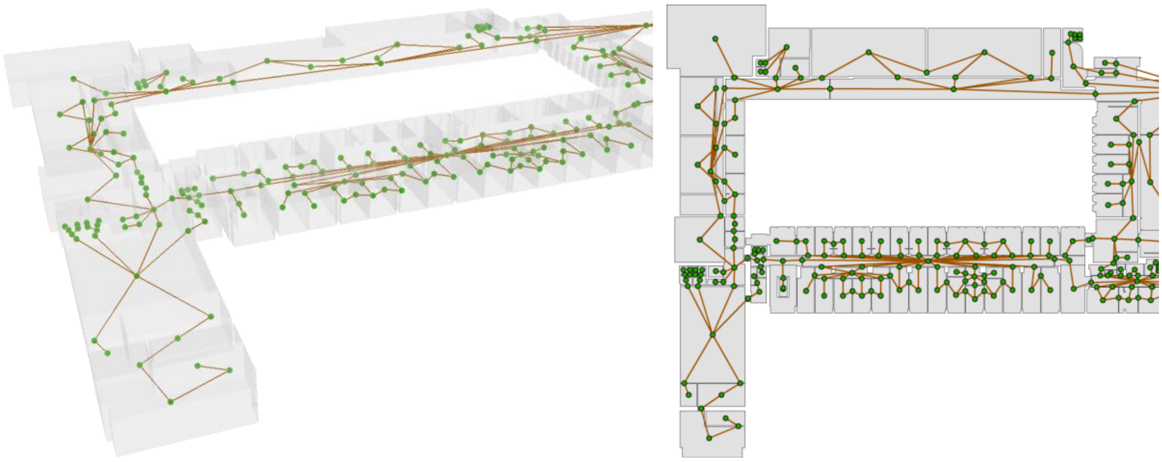


Figure 252: 3-dimensional (left) and 2-dimensional (right) view on an excerpt of the intra-layer graph.

The representation of the vertical transition spaces in dual space is illustrated figure 253. Note that the dual nodes of the transition space cells are vertically linked by dual edges. These edges hence join the horizontally aligned dual subgraphs of the separate building floors.

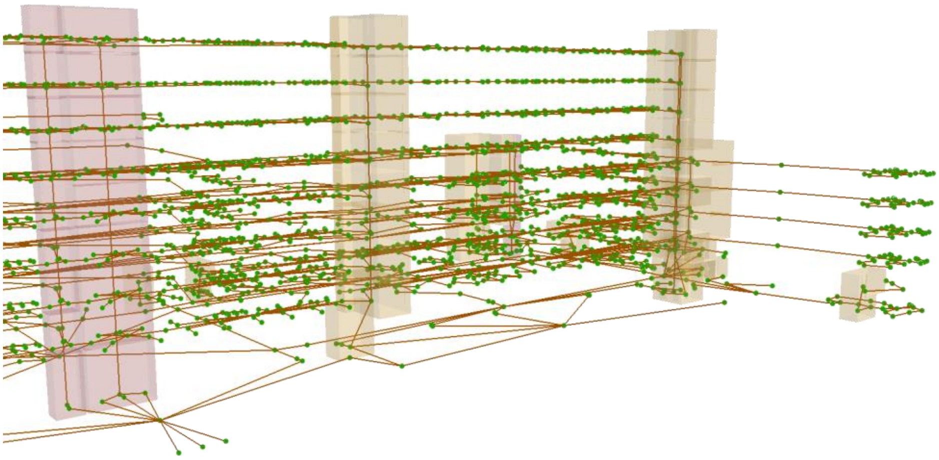


Figure 253: Vertical transition spaces and the linkage of their dual nodes in the intra-layer graph.

Finally, figure 254 shows the result of a simple path search executed on the intra-layer graph in order to find a route from the main entrance of the *TU Main Building* to the lecture room of the Institute for Geodesy and Geoinformation Science.

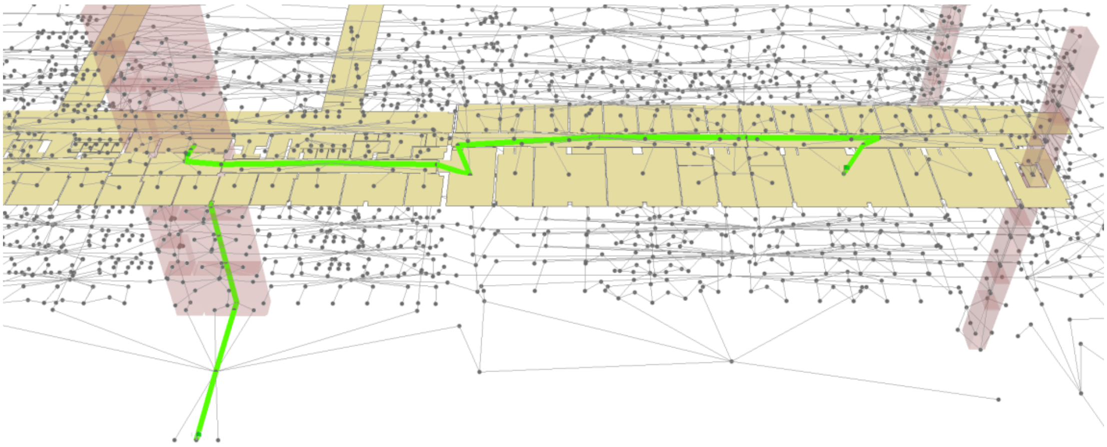
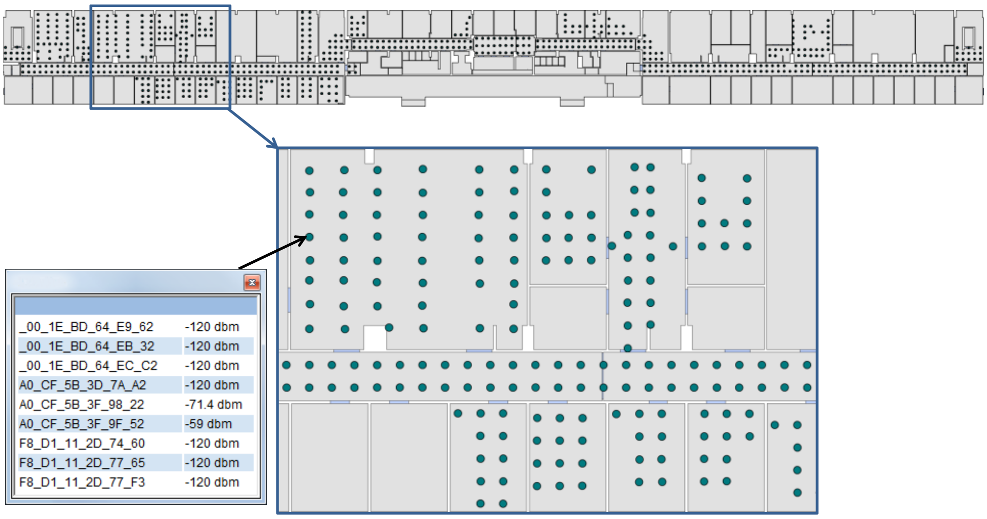


Figure 254: Result of a simple path search on the topographic space layer using ArcGIS 10.1.

7.2.2 Sensor Space Layer

A first step in modelling a sensor space layer mapping the Wi-Fi signal reception areas within the *TU Main Building* was to create a radio map based on *Received Signal Strength Indication* (RSSI) readings from the Wi-Fi access points installed in the building. RSSI provides a measurement of the amount of radio energy present in a radio signal that is received by the wireless networking interface of a Wi-Fi-enabled device. The power level of the signal being received is typically measured in dBm or mW. Starting from a list of all publicly available Wi-Fi access points within the *TU Main Building*, the received signal strengths of these access points were measured at discrete locations on the separate building floors. The resulting RSSI values were identified with both the 3-dimensional coordinates of the point where the measurement took place and the unique MAC address identifier associated with the Wi-Fi access point sending the signal. The set of RSSI values measured at the same point in space with each RSSI value being associated with a different Wi-Fi transmitter is commonly said to be a Wi-Fi *fingerprint*. The resulting fingerprint measurements then served as input data for building a separate radio map for each building floor capturing the spatial distribution of the radio reception conditions. In literature, radio maps like the one realized within the students' project are commonly discussed in the context of Wi-Fi based indoor localization using fingerprinting methods (for an overview see Kolodziej & Hjelm 2006).

In order to specify the locations for the fingerprint measurements, a $1 \times 1\ m$ grid was applied to each floor plan. However, the grid had to be pruned afterwards by removing those locations which were inaccessible for the students due to physical (e.g., furniture) or legal (e.g., lacking access rights) restrictions. For each remaining grid point fingerprint data was acquired. A laptop equipped with a standard network adapter[78] and running a wireless network monitoring tool[79] was used as measuring device and delivered the RSSI values of the Wi-Fi access points in dBm. The measurements were recorded in a fixed height of $1.2m$ which was assumed to be the average height for holding a mobile end-user device. The correct height was ensured by placing the laptop onto a tripod. At every grid point, ten measurements were taken with a time interval of $3s$, and the mean RSSI value for each MAC address was used for the fingerprint at this location. In figure 255, an excerpt of the resulting radio map for the sixth floor of the *TU Main Building* containing fingerprints from nine Wi-Fi access points at 715 sample points is shown as example.



_00_1E_BD_64_E9_62	-120 dbm
_00_1E_BD_64_EB_32	-120 dbm
_00_1E_BD_64_EC_C2	-120 dbm
A0_CF_5B_3D_7A_A2	-120 dbm
A0_CF_5B_3F_98_22	-71.4 dbm
A0_CF_5B_3F_9F_52	-59 dbm
F8_D1_11_2D_74_60	-120 dbm
F8_D1_11_2D_77_65	-120 dbm
F8_D1_11_2D_77_F3	-120 dbm

Figure 255: Grid points for the sixth floor (top) and example fingerprint measured at a single grid point (bottom).

In a second step, sensor space cells were obtained from the raw data of the radio map. Since the fingerprint samples were only available at discrete locations, a spatially continuous surface predicting the Wi-Fi reception conditions for the unsampled locations on the basis of information from the nearest available measured points had to be produced using spatial interpolation techniques. This process was carried out for each MAC address identifier separately in order to estimate the signal reception areas of the individual Wi-Fi access points. *Inverse distance weighting* (IDW) was used as interpolation method. IDW relies on the assumption that sample points that are closer to an unsampled point are more similar to it and hence have a greater influence on its predicted value than those further away (Li & Heap 2008). An unsampled location receives a distance-weighted average of the RSSI values from its surrounding samples with the weight attached to each sample being an inverse function of its distance from the unsampled location. Each measured point therefore has a local influence that diminishes with distance. IDW is an exact interpolator since prediction values for sample points are equal to their measured value which implies that minima and maxima in the interpolated surface can only occur at sample points.

The resulting interpolated surfaces were partitioned into reception areas using three signal strength bands as partitioning criteria. Whereas the best signal strength received for the entire radio map was $-30\ dBm$, the lowest power level reported was $-120\ dBm$. However, the signal reception already became unreliable starting from $-90\ dBm$. Therefore, three signal strength bands were chosen to cover the ranges from 1) -30 to $-60\ dBm$ indicating a very good signal reception, 2) -60 to $-90\ dBm$ indicating a good or fair signal reception, and 3) -90 to $-120\ dBm$ indicating no (reliable) signal reception. An example interpolation map for the Wi-Fi transmitter associated with the MAC address *00:1E:BD:64:E9:62* is shown below.

[78] An ASUS notebook K52JK/K52JB series equipped with an Atheros AR9285 wireless network adapter was used.

[79] The commercial product *WirelessMon 4.0* was used. See http://www.passmark.com/products/wirelessmonitor.htm for more information.

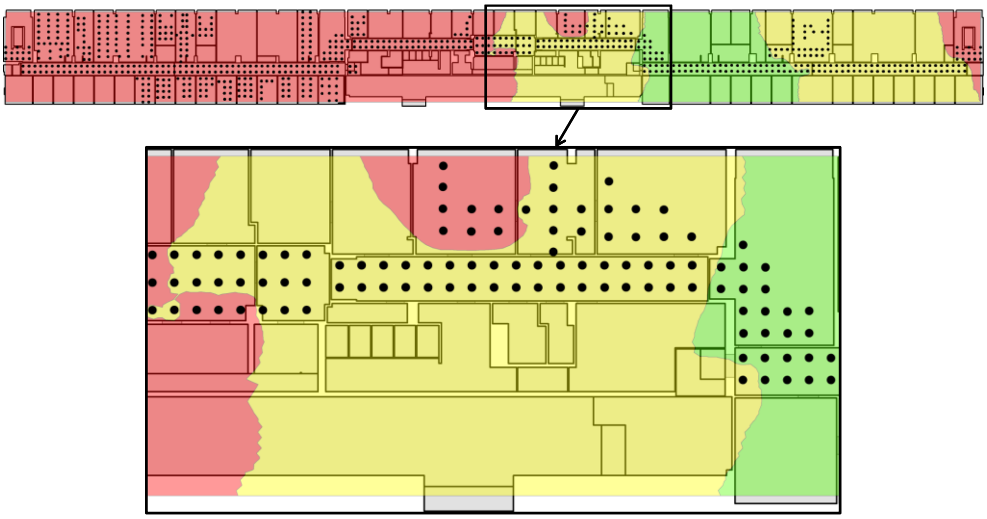


Figure 256: The resulting interpolation surfaces for the signal strength bands -30 *to* -60 dBm *(green),* -60 *to* -90 dBm *(yellow), and* -90 *to* -120 dBm *(red) derived for the MAC address 00:1E:BD:64:E9:62.*

Another interpolation map denoting the signal reception areas for the MAC address *A0:CF:5B:3F:9F:52* is presented in figure 257. Note that the corresponding Wi-Fi access point is not installed on the sixth floor of the *TU Main Building* but on the fifth floor which explains the islands with good signal strength within areas of no signal reception. Moreover, the map nicely demonstrates the impact of walls on the signal propagation. Whereas the signal can be received in some rooms in the upper left part of the map, it is not available within the adjacent corridor. In general, radio waves are subject to different propagation anomalies such as multipath fading or time delay as well as diffraction, reflection, scattering, and absorption at physical objects of the built environment. The IDW interpolation as applied in the students' project does not account for these anomalies when predicting the signal strength values for unsampled locations. In both figure 256 and figure 257 it can therefore be observed that the borders of the derived reception areas are less influenced by the built-up space at locations where no measurements took place. The interpolation maps would hence gain in precision by increasing the number of sample points and fingerprint measurements. As shown by (Muttitanon et al. 2007), IDW can even produce better results for Wi-Fi maps than, for example, geostatistical interpolation methods such as *Kriging* in case the sample points are dense enough to capture the local extent of the interpolation surface. However, the approach followed in the students' project and presented here is only meant as example. The MLSEM framework itself is independent of any method or technique for deriving the spatial extent of sensor space cells.

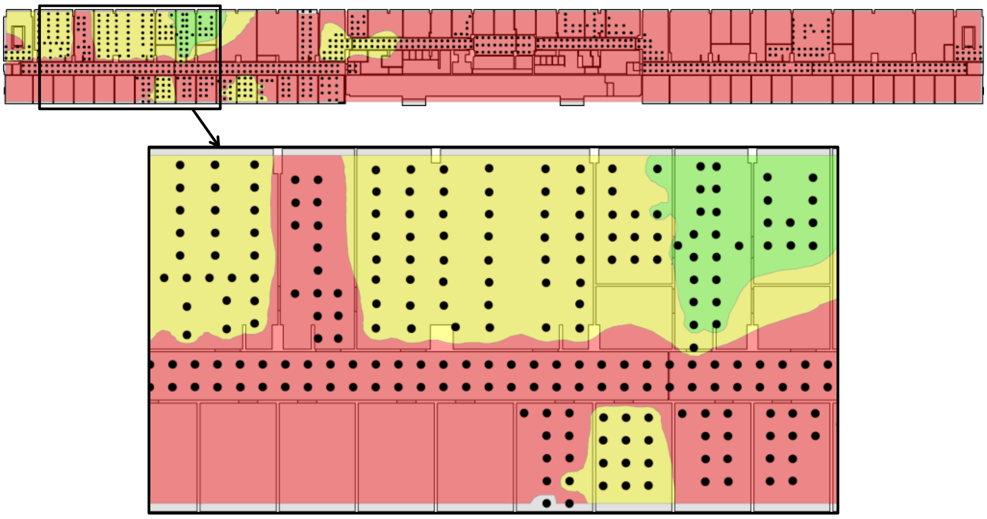


Figure 257: The resulting interpolation surfaces for the signal strength bands -30 *to* -60 dBm *(green),* -60 *to* -90 dBm *(yellow), and* -90 *to* -120 dBm *(red) derived for the MAC address A0:CF:5B:3F:9F:52.*

The signal reception areas of a given interpolation map were finally extruded by the maximum room height on the respective building floor in order to receive 3-dimensional Wi-Fi sensor space cells and their separating 2-dimensional boundary cells. A separate Wi-Fi sensor space layer was populated for each MAC address identifier. The following figure shows the 3-dimensional representations of the interpolation maps from figure 256 and figure 257. The separate sensor space layers were then loaded into the MLSEM database (cf. chapter 7.2.3) in addition to the topographic space layer.

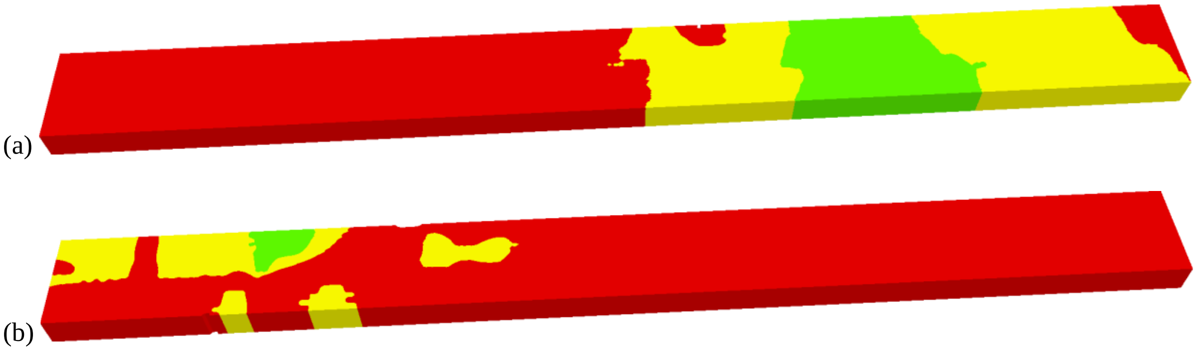


Figure 258: 3-dimensional sensor space layers derived based on the interpolation results from figure 256 (a) and figure 257 (b).

The intra-layer graphs were constituted in the similar way as for the topographic space layer. An additional outer space cell was introduced on each layer whose dual node representation was linked to the dual node of every sensor space cell through one or more dual edges. For the sensor space layer associated with the MAC address *00:1E:BD:64:E9:62* the resulting intra-layer graph consisted of seven dual nodes and eleven dual edges. Its weak form is sketched in figure 259. Note that the geometric embedding again was realized using centroids and straight line segments.

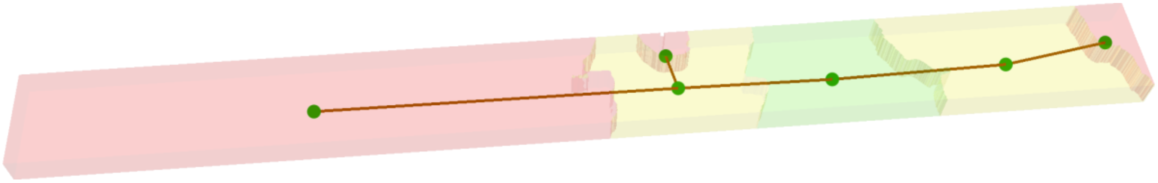


Figure 259: Resulting weak intra-layer graph for the MAC address 00:1E:BD:64:E9:62 (cf. figure 258a).

Based on the description of the topographic space and the Wi-Fi sensor spaces, the linkage between both space views in dual space could be established in a subsequent step. Precisely, inter-layer edges connecting the dual nodes of two space cells from either space representation were introduced if the intersection of the primal space geometries was non-empty. The primal space overlaps between the space cells from the topographic space layer and the Wi-Fi sensor space layer shown in figure 259 is presented below. Note that only the rooms and doors located on the sixth floor of the *TU Main Building* are visualized due to the fact that the Wi-Fi signal reception areas were acquired separately for each building floor. Thus, the topographic space cells on the remaining building floors are necessarily covered by the outer space cell of the depicted Wi-Fi sensor space layer. The corresponding bipartite inter-layer graph linking the dual nodes from either space layer is shown in figure 260b. The outer space cells as well as their incident dual edges have been omitted for readability. Also note that both space layers are visualized on top of each other in order to better illustrate the inter-layer edges. The entire inter-layer graph for both space layers was built from 3535 dual nodes and 3566 inter-layer edges. The excerpt shown below contains 250 dual nodes (244 from the topographic space and 6 from the Wi-Fi sensor space) as well as 276 inter-layer edges.

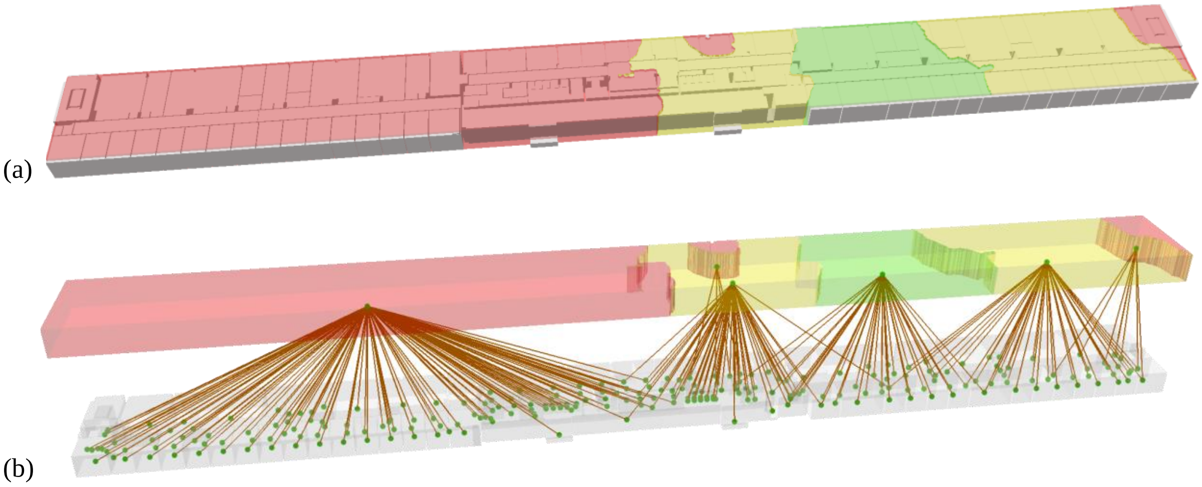
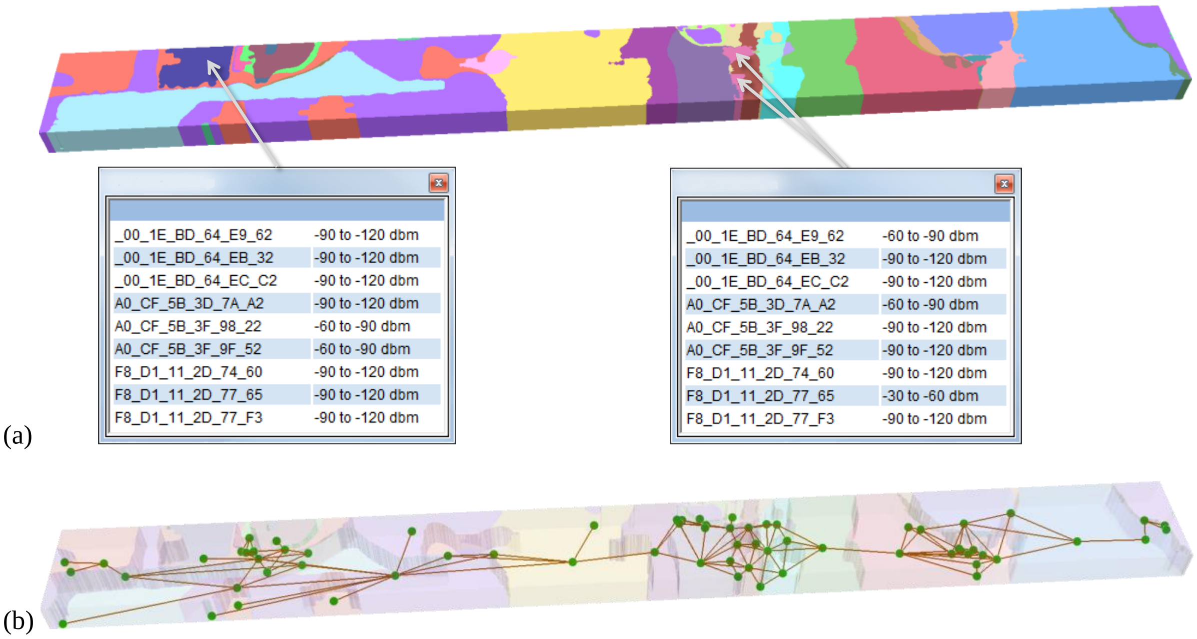


Figure 260: Primal space overlaps between the space cells from the topographic space layer and the Wi-Fi sensor space layer from figure 259 (a) and the corresponding inter-layer edges (b).

In a final step, the signal reception areas of the separate Wi-Fi transmitters were integrated in a common space view by applying the *merge operation* to the set of derived Wi-Fi sensor space layers. For the sixth floor of the *TU Main Building*, the *merge operation* yielded the space layer as presented in figure 261a. The sensor space cells on the merged space layer represent the non-overlapping intersection of the signal reception areas of all nine Wi-Fi access points available on the sixth floor. Merged space cells capturing the same combination of input space cells have been assigned the same colour. The corresponding weak intra-layer graph depicted in figure 261b consists of 68 dual nodes and 144 dual edges.



_00_1E_BD_64_E9_62	-90 to -120 dbm
_00_1E_BD_64_EB_32	-90 to -120 dbm
_00_1E_BD_64_EC_C2	-90 to -120 dbm
A0_CF_5B_3D_7A_A2	-90 to -120 dbm
A0_CF_5B_3F_98_22	-60 to -90 dbm
A0_CF_5B_3F_9F_52	-60 to -90 dbm
F8_D1_11_2D_74_60	-90 to -120 dbm
F8_D1_11_2D_77_65	-90 to -120 dbm
F8_D1_11_2D_77_F3	-90 to -120 dbm

_00_1E_BD_64_E9_62	-60 to -90 dbm
_00_1E_BD_64_EB_32	-90 to -120 dbm
_00_1E_BD_64_EC_C2	-90 to -120 dbm
A0_CF_5B_3D_7A_A2	-60 to -90 dbm
A0_CF_5B_3F_98_22	-90 to -120 dbm
A0_CF_5B_3F_9F_52	-90 to -120 dbm
F8_D1_11_2D_74_60	-90 to -120 dbm
F8_D1_11_2D_77_65	-30 to -60 dbm
F8_D1_11_2D_77_F3	-90 to -120 dbm

Figure 261: Single Wi-Fi sensor space layer resulting from a merge operation applied to the sensor space layers associated with the separate MAC addresses (a) and corresponding weak intra-layer graph (b).

The overlaps in primal space between the space cells on the merged Wi-Fi sensor space layer and those on the topographic space layer are captured in figure 262a. Similar to the above example, only the topographic space cells located on the sixth floor are presented for readability. The excerpt of the inter-layer graph linking both space representations is made of 308 dual nodes (244 from the topographic space and 68 from the merged Wi-Fi sensor space) and 473 inter-layer edges. The corresponding portion of the multilayered graph is finally retrieved by simply adding the intra-layer edges from either space layer (261 from the topographic space and 144 from the merged Wi-Fi sensor space) to the inter-layer graph and is depicted in figure 262b. The figure nicely illustrates that intra-layer edges and inter-layer edges are orthogonally aligned in the multilayered graph. Moreover, each Wi-Fi space cell on the merged space layer is associated with less room cells compared to figure 260b which decreases the uncertainty areas of joint states and hence increases the localization accuracy.

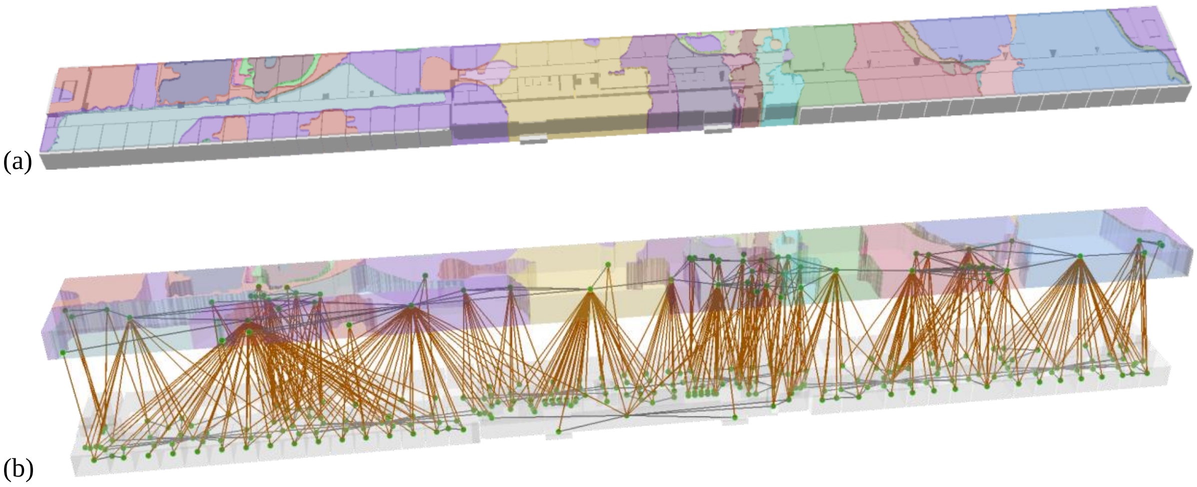


Figure 262: Primal space overlaps between the space cells from the topographic space layer and the Wi-Fi sensor space layer from figure 261 (a) and the corresponding excerpt of the multilayered graph (b).

7.2.3 MLSEM Database Schema

As part of the students' project, the conceptual and platform-independent data model of the MLSEM (cf. chapter 4.4) has been transformed to a database schema which facilitates the storage and management of indoor space representations by means of a spatial database according to the conceptual framework of the MLSEM. Whereas the GML application schema presented in chapter 4.4.2 and appendix B realizes a complete mapping of the conceptual data model onto an XML based implementation model, the engineering step carried out during the students' project aimed at providing a simple but conformant translation of the most important core concepts into relational database tables as well as corresponding fields and relationships. The *Oracle Database 11g Release 2 Enterprise Edition* and its *Oracle Spatial and Graph* extension have been chosen as target implementation platform due to the native and rich support for 3-dimensional spatial data types and functions.[80] The database schema was successfully deployed and tested within the students' project in order to manage the 3-dimensional indoor space model of the *TU Main Building* as described in the previous sections.

A graphical depiction of the database structure imposed by the developed MLSEM database schema is presented in figure 263. The diagram uses an informal UML extension for the modelling of relational database structures. Database tables are represented as UML classes with the table name given as class name associated with a *«table»* stereotype being displayed as table icon in the top right corner. Database fields are modelled as attributes of the table class and are assigned the stereotype *«column»*. Primary key columns carry a `PK` flag, whereas `FK` indicates a foreign key. A field which renders both a primary and a foreign key is labelled with `pfK`. The asterisk symbol ('*') is used on fields that cannot be `NULL`. The behaviour associated with the database fields (e.g., indexes, constraints, triggers, procedures) is modelled as stereotyped operations of the table class. For example, the stereotypes *«PK»* and *«FK»* are applied to primary respectively foreign key constraints. In addition, foreign key/primary key dependencies between two tables are represented as UML associations stereotyped as *«FK»*. This relationship implies a parent and a child table with the parent defining the primary key and the child implementing a referencing foreign key. The arrowhead of the association points from the child to the parent. The fields involved in the relationship as well as their dependency are given in parentheses as association label. Note that the formal text-based definition of the database schema is expressed in SQL and provided in appendix C. It was automatically generated from the UML model.

[80] The developed database schema could also be easily adapted for alternative spatial databases such as *PostgreSQL/PostGIS*. This however involves specifying extra table structures for managing 3-dimensional spatial objects unless natively supported by the database system in question.

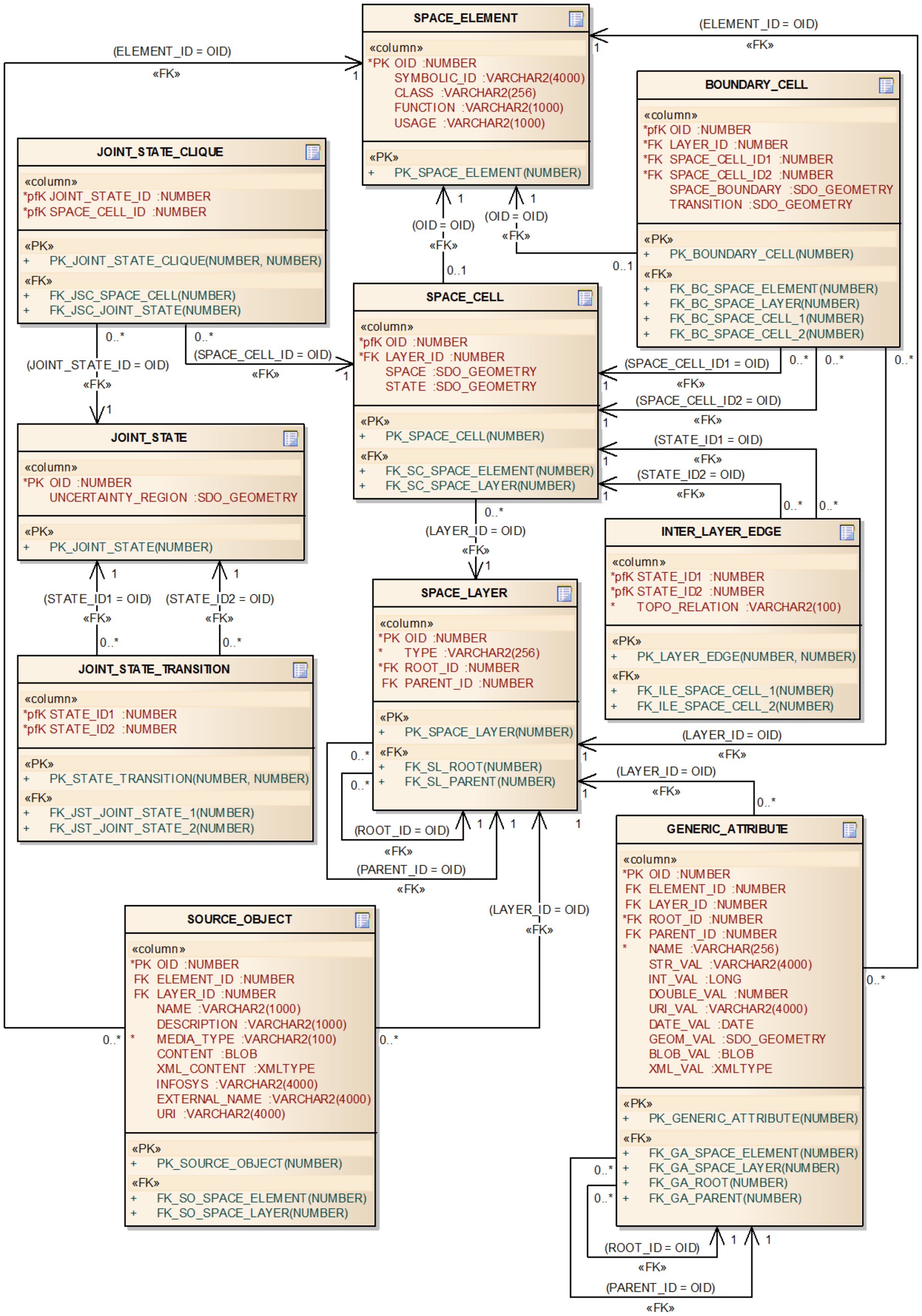


Figure 263: Graphical UML notation of the MLSEM database schema.

The central tables of the database schema are `SPACE_CELL` and `BOUNDARY_CELL` which map the feature types *SpaceCell* respectively *BoundaryCell* from the conceptual data model. The `OID` field of `SPACE_CELL` serves as primary key for space cell tuples. Whereas the primal and dual space representation of a *SpaceCell* feature are captured by the feature types *Space* and *State* in the conceptual schema, the database schema reduces both representations to the fields SPACE and STATE which only hold the geometric characteristics of the space cell in either space. Hence, SPACE stores the primal space geometry of a space cell in two or three dimensions and STATE denotes the geometric embedding of the dual node. The object-relational data type SDO_GEOMETRY is used for this purpose. Both SPACE and STATE can be NULL if a corresponding geometric representation is not available.

The SDO_GEOMETRY data type as defined by Oracle follows the ISO 19107 Spatial Schema specification and supports most of the conceptual entities defined therein as well as their counterparts from GML/ISO 19136. However, SDO_GEOMETRY is per definition restricted to the representation of topologically compact geometric objects in Euclidean 2-space or 3-space. Thus, and in contrast to both ISO 19107 and GML/ISO 19136, the non-compact primal space geometries of *outer* space cells cannot be expressed. In this, the database schema also differs from the mapping of the conceptual data model of the MLSEM to a GML application schema (cf. the XML encoding of an outer space cell as exemplified in chapter 7.1). As a workaround, the following two-step procedure is proposed to derive a feasible substitute geometry for the outer space cell on a given space layer: 1) determine the buffered bounding box respectively bounding volume of all space cells on the space layer in question and 2) subtract the primal space geometries of these space cells from the intermediate geometry using a regularized Boolean subtraction operation.[81] The buffer applied to the bounding box/volume in the first step should be chosen so that the primal space geometry of the outer space cell encloses the space cells on every other space layer participating in the indoor space model. Although this is not a mandatory requirement, it nevertheless simplifies the detection of inter-layer edges in subsequent steps.

Further thematic attributes of SPACE_CELL tuples are stored in fields of the table SPACE_ELEMENT which translates the abstract supertype *SpaceElement* of *SpaceCell*. The specialization relationship between *SpaceCell* and *SpaceElement* is modelled as foreign key relationship between both tables. The columns SYMBOLIC_ID, CLASS, FUNCTION, and USAGE defined for SPACE_ELEMENT hereby correspond to the thematic properties of the *SpaceElement* feature type.

The database realization of the *BoundaryCell* feature type follows a similar approach. The 2-dimensional respectively 1-dimensional primal space geometry of a boundary cell is stored in the SPACE_BOUNDARY field of the BOUNDARY_CELL table, whereas the geometric embedding of its dual edge representation is captured by the column TRANSITION. Again, the entities *SpaceBoundary* and *Transition* from the conceptual data model are simplified by this mapping. Two compulsory foreign key associations between BOUNDARY_CELL and SPACE_CELL model the topological relationships between space cells and boundary cells. If interpreted in primal topology space, then the relationships denote the two space cells on the coboundary of a boundary cell. In dual topology space, the two dual nodes on the boundary of a dual edge are referenced. Note that both notions are in one-to-one correspondence due to the applied Poincaré duality. Since *BoundaryCell* is also derived from the supertype *SpaceElement*, a further foreign key relationship is established to the SPACE_ELEMENT table and allows for providing additional thematic properties on the related tuple of SPACE_ELEMENT.

Both SPACE_CELL and BOUNDARY_CELL provide foreign key references to the table SPACE_LAYER which translates the *SpaceLayer* feature type. Based on the topological relationships between those SPACE_CELL and BOUNDARY_CELL tuples being associated with the same entry in SPACE_LAYER, the intra-layer graph of this space layer can be built. In Oracle, a separate data model called *Network Data Model* (NDM) is available for the storage and analysis of network structures. It has not been chosen as essential part of the MLSEM database schema since it can be easily populated automatically and on-demand from the foreign key relationships between SPACE_CELL and BOUNDARY_CELL as well as their STATE and TRANSITION fields. Since Oracle 11g, advanced analysis and modelling features such as path arithmetic support and the consideration of user and application attributes are available for NDM structures which thus strongly support the task of path planning. A comprehensive discussion of the NDM is however beyond the scope of this thesis. SPACE_LAYER has a mandatory TYPE field which classifies a space layer entity and hence realizes the *type* property of *SpaceLayer*. The ROOT_ID and PARENT_ID columns are foreign keys both of which reference another tuple of SPACE_LAYER. They are used to model arbitrarily nested subspace layer hierarchies in a tree-like graph structure. Whereas PARENT_ID points

[81] Both steps are supported in both two and three dimensions by current GIS software tools such as ArcGIS 10.1 or FME 2013.

to the direct parent (if any) of a SPACE_LAYER tuple within a given tree, ROOT_ID always has to reference the root node of this tree. The latter is a self-reference in case the space layer is the root node. In order to retrieve a whole subspace layer hierarchy, simply all tuples sharing the same ROOT_ID have to be selected.[82]

The inter-layer edges linking space cells from different space layers in dual space are stored in the database table INTER_LAYER_EDGE which thus agrees with the conceptual feature type *InterLayerEdge*. Like with the BOUNDARY_CELL table, INTER_LAYER_EDGE is related to SPACE_CELL by two foreign key associations in order to reference the dual nodes on the boundary of every inter-layer edge. The topological relationship between the primal space geometries of the involved space cells has to be provided through the mandatory TOPO_RELATION field. In contrast to the conceptual data model, a separate table for the resulting multilayered graph is not provided since it can be implicitly derived from the tables SPACE_CELL, BOUNDARY_CELL, and INTER_LAYER_EDGE as well as their relationships.

In addition to this, the proposed MLSEM database schema also supports the explicit storage of joint states and their transitions. A single joint state is represented as tuple of the table JOINT_STATE. Similar to its conceptual counterpart *JointState*, the uncertainty region associated with the joint state can be provided as SDO_GEOMETRY object. Note that disconnected geometric objects can be represented by a single instance of SDO_GEOMETRY which is a mandatory prerequisite to the spatial modelling of uncertainty regions (cf. definition 3.41). The clique of dual nodes constituting a joint state is provided through the JOINT_STATE_CLIQUE table which simply realizes an *n:m* relationship between JOINT_STATE and SPACE_CELL by a foreign key association to either table. Finally, the tuples in the JOINT_STATE_TRANSITION table express the transitions between pairs of joint states. Since every joint state transition has precisely two joint states on its boundary (cf. the definition of the *JointStateTransition* feature type), two foreign key associations with the JOINT_STATE table have been modelled for JOINT_STATE_TRANSITION. The *JointStateSpace* associated with a set of space layers simply follows from all tuples in JOINT_STATE and thus is not addressed by a separate database table.

The SOURCE_OBJECT table is a direct realization of the *SourceObject* feature type and thus allows for storing or linking representations of indoor space entities from external data sources in different data formats. The table offers fields for the NAME, the DESCRIPTION, and the MEDIA_TYPE of the source object with the latter being mandatory. The external object itself can be stored explicitly as binary data (BLOB) or as XML fragment (XMLTYPE) using the columns CONTENT respectively XML_CONTENT. Alternatively, a reference pointing to the source object in the external data source can be provided. For this purpose, the thematic attributes of the feature types *ExternalReference* and *ExternalObjectReference* from the conceptual data model have been added to SOURCE_OBJECT and are to be used in the same way as defined for these conceptual entities. Each tuple of SOURCE_OBJECT can be either related to a SPACE_ELEMENT or a SPACE_LAYER entry.

The final table GENERIC_ATTRIBUTE of the MLSEM database schema maps the *GenericAttributeType* data type and thus enables the modelling of generic attributes for both SPACE_ELEMENT and SPACE_LAYER tuples. Besides a mandatory NAME, each generic attribute has to express its value through precisely one of the fields STR_VAL, INT_VAL, DOUBLE_VAL, URI_VAL, DATE_VAL, GEOM_VAL, BLOB_VAL, or XML_VAL. In contrast to the conceptual entity *GenericAttributeType*, the database schema hence predefines a list of simple data types for the value of the generic attribute. The flexibility of the conceptual data model is nevertheless maintained, on the one hand, through the possibility of providing arbitrary binary or XML data. On the other hand, the GENERIC_ATTRIBUTE table supports attributes of attributes in order to also capture complex data types. This is realized by allowing generic attributes to be arranged in a tree-like graph structure based on the fields ROOT_ID and PARENT_ID (cf. table SPACE_LAYER). For example, assume a space cell shall be assigned a generic attribute capturing its height of 1.7 metres. Then, a first generic attribute of name *height* could provide a value of 1.7 using the DOUBLE_VAL field. A second generic attribute of name *uom* referencing the first as its parent could denote the unit for this measure (e.g., as STR_VAL representation or as URI pointer to some formal definition using URI_VAL). Moreover, this modelling approach also allows for providing a named collection of generic

[82] Oracle also supports hierarchical queries for traversing tree-like structures using the CONNECT BY clause to be used in SELECT statements whose condition can be expressed based on the ROOT_ID and PARENT_ID fields. Alternative spatial databases (e.g., PostgreSQL/PostGIS) typically also provide the means to formulate hierarchical queries.

attributes (e.g., a collection of simple generic attributes referencing the same root attribute through ROOT_ID which only provides the NAME for this collection). Named collections are useful, for example, in mapping the concepts of *GenericAttributeSet* and *IfcPropertySet* as defined in the standards CityGML respectively IFC.

The conceptual entities from the packages *Groups and Sequences*, *Route*, *Model Linkage*, and *Constraints* from the MLSEM application schema have not been considered in this first draft of the MLSEM database schema. Their mapping has to be addressed in future work.

Chapter 8
Conclusions and Outlook

This chapter presents and evaluates the main results of the thesis and summarizes methods, strategies, and approaches applied during the research. Moreover, contributions to the field of indoor navigation as well as related fields are discussed and future research is identified and outlined.

8.1 Review and Evaluation

At the outset of this thesis, a fundamental issue of existing indoor space models for indoor navigation has been identified, namely the lack of a consistent semantic and mathematical definition of indoor space and of an accompanying standardized computer representation. Taking this as motivation, the definition of an indoor space model at the level of 1) a *conceptual model*, 2) a *mathematical model*, and 3) a *computer representation* that conforms to international standards from the geoinformation community has been formulated as goal for this thesis. The review of results is structured along these three levels.

Conceptual model of indoor space. A comprehensive study of related work in the field of indoor space modelling has provided knowledge about existing concepts for the semantic, spatial, and mathematical representation of indoor space. Relevant approaches were evaluated against the identified challenges to indoor navigation, and their strengths and weaknesses in answering these challenges were argued. From these findings, mandatory requirements were deduced that served as basis for the design of a conceptual model for the spatio-semantic description of indoor space.

An important objective of the research was to define a generic notion of space that would allow for modelling the interior built environment in which the bodily movement takes place but also for describing complementary physical or logical decompositions of indoor space. This is realized through the concepts *space cell*, *boundary cell*, and *space layer*. A *space cell* is defined as partition of indoor space whose semantic meaning follows from the applied partitioning schema. It enables the modelling of navigable and non-navigable topographic spaces (e.g., rooms, corridors, walls), sensor spaces (e.g., signal coverage areas), or logical spaces (e.g., security zones) and in that provides a common abstraction to indoor space that goes beyond alternative approaches. A salient aspect of space cells is their geometric-topological representation in both primal and dual space. Based on this *dual representation* schema, a one-to-one pairing between a 2-dimensional respectively 3-dimensional partition of indoor space in primal space and its dual node representation within a graph-based conceptualization is realized. The spatial and semantic scope of space cells is limited by *boundary cells* which are defined to describe both tangible and virtual boundaries of space cells. Boundary cells follow the similar generic representation schema but are of one dimension less in primal space and are mapped onto an edge in dual space.

The aggregation of mutually non-overlapping space cells to a larger view on indoor space is then captured by a *space layer*. The *dual representation* schema is consistently extended to space layers. Whereas the primal space representation captures the spatial configuration of space cells and boundary cells in either two or three dimensions, the dual space representation provides a deterministic mapping onto a corresponding navigation graph. In contrast to many alternative approaches which (if at all) define a graph-based conceptualization of indoor space purely at the level of semantic entities, the dual graph structure as proposed in this thesis directly follows from the spatial configuration of space cells and their qualitative *connectedness* relationships. It can hence be derived for arbitrary space representations and notions of space. In order to add quantitative and metric information, the dual graph can be embedded into Euclidean space.

It follows that the concept of space layers allows for the modelling of multiple and different indoor space representations including navigable and non-navigable topographic spaces for different modes of locomotion, sensor spaces for different localization technologies and methods, and multiple logical spaces. It thus answers both the research **question 1.5** and **question 1.6** (cf. chapter 1.3). The notions of space cell and space layer were further extended to *subspace cells* and *subspace layers* in order to express hierarchical and nested structures as well as *containment* relationships within the built environment or alternative notions of space (cf. **question 1.7**).

An important advantage of the developed modelling approach over related work is that the space views are independent in that changes to the configuration of space cells on one space layer do not affect the structure of other space layers. Their joint consideration within the separate tasks of navigation is enabled through a *multilayered graph* which renders a core element of the conceptual model. The multilayered graph integrates the dual graphs of the participating space layers and links their dual nodes based on topological overlap relationships between the space cells in primal space. It hence spans a 2-dimensional space along the *connectedness* of space cells on each space layer and their *overlap/containment* between different space layers, and thus can be said to be a *generic space-relation model*. The resulting graph structure enables hierarchical and multi-level path searches (cf. **question 1.4**). Due to the bijective pairing between space cells (boundary cells) and the dual nodes (dual edges) of the graph, the primal space geometry of space cells (boundary cells) as well as their semantic and symbolic properties are accessible from the graph elements which allows for spatio-semantic path searches beyond the fastest or the shortest path. Moreover, this information can be used, for example, to derive (hierarchical) route instructions and visual aids for route guidance, or to dynamically evaluate accessibility constraints (e.g., whether a user physically fits through an opening or narrow passage) (cf. **question 1.4**). Corresponding methods have been proposed in literature (e.g., Lorenz et al. 2006, Yuan & Schneider 2011) and can be transferred in the context of the developed indoor space model.

The dual nodes of the multilayered graph further define a *state space* of possible navigation states of a person or object travelling through indoor space. At every point in time, the person or object can only be located within the spatial scope of precisely one space cell per space layer whose dual node thus denotes the active navigation state on that space layer. The joint consideration of the active joint states over all space layers then renders the *joint state of navigation* of the person or object, and the primal space geometries of the involved space cells mutually constraint the absolute position. The multilayered graph is therefore also to be seen as *generic space-event model* and it has been shown in this thesis how this notion facilitates the localization and tracking of navigation users (cf. **question 1.4**).

The indoor space model supports an unbounded number of space layers. Thus, arbitrary views and aspects of indoor space can be (dynamically) added or removed. For navigating a given user through the indoor space, only those space layers have to be considered that match the *navigation context* of the user (e.g., a topographic space layer matching the mode of locomotion and one or more sensor space layers reflecting the localization capabilities of the mobile device). Supporting multiple and different navigation contexts is hence realized by the ad-hoc selection of a corresponding subset of space layers and by establishing a user-dependent multilayered graph on that subset (cf. **question 1.8**).

The conceptual model comprises the representation of environmental and user-dependent *navigation constraints* (cf. **question 1.9**). A classification of navigation constraints into 1) physical movement obstructions (*can* constraints) and 2) legal access restrictions (*may* constraints) is proposed. Moreover, *constraint conditions* are defined which act as *guards* and thus determine the physical, temporal, or logical conditions under which a navigation constraint applies. Navigation constraints can be associated with a single space cell or boundary cell (*simple* constraint) as well as with groups or (un)directed sequences thereof (*complex* constraint such as a prohibited maneuver). Moreover, constraint conditions can be combined in logical expressions (*combined* constraint). This renders a flexible framework which goes beyond existing approaches in literature. The same concepts can also be used to model the user context (e.g., user-related movement restrictions). Rules and algorithms for evaluating the navigation constraints against the user context in path searches are developed and presented in this thesis. The expressivity of the constraint model was demonstrated along typical scenarios and uses cases for navigation constraints in indoor environments.

Mathematical model of indoor space. A correct, consistent, and complete mathematical formalization was developed for the elements of the conceptual model. The formalization work mainly drew from fields such as topology, manifold theory, and graph theory, and was impeded by the fact that presentations of these fields in literature often differ with respect to terms, symbols, and definitions. This was taken as motivation to provide a comprehensive and consistent overview of the applied theory which is presented in a supplementary appendix of this thesis.

A main focus of the mathematical model was on the sound mathematical embedding of the *dual representation* of space cells, boundary cells, and space layers. The primal space geometry of a space cell is defined as bounded regular semi-analytic subset of Euclidean space and is required to render a topological manifold with boundary to ensure its well-behavedness. In contrast to alternative approaches to indoor space modelling, this foundational definition for a partition of indoor space is valid in both two and three dimensions and explicitly includes free-form surfaces and curves. The primal space topology then captures a CW decomposition of the manifold. This implies a consistent relation between the geometric and topological description in primal space. The CW complex allows for the reasoning about key topological properties of the space cell in a computationally efficient manner. Moreover, and based on methods from algebraic topology, the CW complexes of separate space cells can be glued together along their boundaries in order to express their mutually non-overlapping configuration on a space layer. This again results in a CW complex that forms the primal space topology of the space layer, whereas the primal space geometry simply follows from the union of the space cell geometries. The spatial configuration of space cells needs not be manifold which is especially important with respect to the modelling of sensor spaces or logical spaces and again renders the proposed modelling approach more flexible than those of related work.

The one-to-one correspondence between the primal and dual space representation is formalized based on the *Poincaré duality theorem* from algebraic topology. This theorem is also utilized in alternative approaches in literature. However, problems in these approaches are identified and elaborated in depth in this thesis. The sound application of the Poincaré duality within the context of indoor space modelling has therefore been a main research task. As a result, the concept of an *outer space cell* was introduced and defined mathematically. The outer space cell completes the tiling of Euclidean space on a given space layer in primal geometry space. Since it is topologically unbounded and possibly non-manifold, it distinctively differs from the notion of a space cell. Moreover, the CW complex of a space layer has to be embedded in a closed manifold in order to apply the Poincaré duality, which was realized through a *one-point compactification* of its ambient space. Both aspects are not sufficiently addressed in related work. The Poincaré dual of the CW complex then allows for consistently defining both the dual adjacency graph in dual topology space and its isomorphic embedding in dual geometry space. Moreover, the deterministic one-to-one pairing between the primal space representation of space cells (boundary cells) and elements of the dual graph has been elaborated based on these results. Finally, it is shown that the dual-graph-based conceptualizations of alternative approaches can be explained through a *weak dual graph* and its additional pruning based on semantic criteria.

The sound mathematical definition of a *multilayered graph* structure which consists of the Poincaré dual graphs from multiple space layers and additional inter-layer edges expressing topological overlap and containment relationships also substantially exceeds the results of previous work applying the Poincaré duality. Based on this structure, a *non-deterministic finite-state machine* is developed which supports the tasks of localization and tracking. For this purpose, the concept of a state machine was extended to consider *joint state*s, i.e., multiple active states on separate space layers, as well as *joint state transitions*. In contrast to alternative proposals, and due to the generic notion of space underlying the proposed indoor space model, arbitrary representations and combinations of topographic, sensor, and logical spaces can be fed to the state machine. Although a continuous movement through indoor space is assumed on all space layers, the active joint state at a point in time t_n only depends on its successor at t_{n-1} and the set of *events* that fired the transition. This renders the state machine robust and simplifies implementations.

In order to formalize the notion of *subspace cells* and *subspace layers*, a *partial order* has been defined for space cells and space layers based on their geometric inclusion. Hierarchical structures are then expressed as *partially ordered sets* of space cells or space layers. Whereas hierarchies are mostly restricted to render tree-like graph structures in alternative approaches, they additionally result in *star structures* within the proposed multilayered graph which makes the approach followed in this thesis more flexible and more expressive. Moreover, hierarchical structures are not limited to the interior built environment but may also be applied to further notions of space and between different notions of space.

A further novel and salient aspect of the mathematical model developed in this thesis is the definition of a *space layer algebra* that allows for algebraic expressions to be built up from space layers and three well-defined binary operations, namely the *merge*, *intersection*, and *difference* operation. The operations simultaneously address the separate geometric and topological description of space layers in both primal and dual space as well as their semantic dimension, and hence substantially go beyond purely geometric set operations. The space layer algebra can be used to *integrate* two different space layers (e.g., different partitionings of the same or different notions of space) in a common view or to *subtract* one space layer from another. It is shown in detail how the space layer algebra supports the (dynamic) selection and combination of space layers according to a given context of navigation. Moreover, it allows for the modelling of subspace layers without having to redundantly copy non-subspaced space cells from the superspace layer, which reduces modelling efforts and storage requirements. A concept similar to the developed space layer algebra has not been presented in any of the investigated approaches to indoor space modelling.

Computer representation. A review of geometric-topological data models proposed in the field of GIS has provided knowledge about existing approaches to the computer representation of spatial objects and revealed their limitations in expressing the spatial aspects of space cells, boundary cells, and space layers in both primal and dual space. Focus was then put on the ISO 19100 standards series for the modelling of geographic features and their spatio-semantic characteristics. The feasibility of mapping the conceptual and mathematical model based on this standards series was argued in depth. Following on from these results, UML was used as formal language to design an ISO-conformant, complete computer representation of the developed indoor space model. In order to ensure consistency, invariants reflecting conceptual or mathematical constraints were introduced into this data model using OCL as formal declarative language. By applying the formal rule set provided in ISO 19136, an XML-based data exchange format was derived from the data model. Moreover, the mapping to a relational spatial database schema was demonstrated for its most important elements.

The resulting data model is more flexible than the conceptual model and its mathematical formalization. For example, it allows the dual space representation of a space layer to be populated without an accompanying primal space representation (or vice versa) in case the latter is not available or required by the navigation application. Likewise, it facilitates to represent, store, and exchange intermediate steps in the acquisition and refinement of indoor data (e.g., starting from a pure geometric description of primal space to a full geometric-topological description in both primal and dual space). Even a purely semantic and symbolic description of indoor space is already sufficient. This flexibility thus increases the practical applicability of the indoor space model without violating its conceptual or mathematical consistency. In contrast, the one-to-one correspondence between the primal and dual space representations of space cells, boundary cells, and space layers is still ensured by the data model even if it is not fully populated.

The data model defines the following additional concepts which address data needs in indoor navigation:

- *Cost values* for graph elements (e.g., distance, monetary, and time-based measures);
- *Groupings* of space cells and boundary cells to express functional or logical aggregations of spatial entities (e.g., all rooms belonging to the same department or all toilets in a public building) which may be addressed in path searches;
- *Sequences* of (sparse sets of) space cells and boundary cells to express (un)directed paths on the undirected Poincaré dual graph of a space layer (e.g., to model prohibited maneuvers);
- *Embedding or referencing* of complementary representations of space cells, boundary cells, or space layers in different formats or external repositories (e.g., a CityGML representation of a room cell);
- *Representation and exchange of routes* that result from path searches including waypoints, cost values (e.g., travel time, travel distance), and route guidance matching the needs of the navigation user (e.g., textual, visual, or audio); and
- *Linkage* of elements of the multilayered graph with elements from another navigation graph (e.g., outdoor road network) or from another multilayered graph (e.g., between 2-dimensional representations of subsequent building floors) to enable seamless transitions.

A discussion against existing standards from the field of urban and building information modelling revealed that geoinformation about the interior built environment provided in one of these standards can be used to populate a topographic space layer of the indoor space model. It was shown that the semantic entities *SpaceCell* and *BoundaryCell* of the developed data model serve as *conceptual interfaces* to link concepts from existing semantic building models (IFC, CityGML, and ESRI BISDM) without duplicating them. The generic notion of space helps to keep the *semantic impedance* low in this mapping. Moreover, it was argued that purely geometric building models already suffice to derive a valid space layer (cf. **question 1.11**). The impact of the different spatial modelling paradigms applied in the field of GIS and BIM on the resulting primal and dual space representation were discussed and proposals for minimizing their differences were presented.

The relation of the conceptual data model to existing international standards on location-based services and architectures (OGC OpenLS and ISO 19133) was investigated. On the one hand, it was demonstrated that the expressivity of the developed concepts exceeds these standards regarding the representation of the navigation space, navigation constraints, or route information and guidance. On the other hand, these standards also address the design of system interfaces and operations for an LBS computing environment, which was out of scope of this thesis. However, it was shown that indoor space and route information modelled according to the developed conceptual data model can be mapped onto and exchanged via these interfaces (cf. **question 1.12**).

Finally, the feasibility and applicability of the developed conceptual data model was illustrated along two “proof of concept” demonstrations.

Conclusions. Based on the research for this thesis and the presented findings and results, conclusions are drawn in the following with respect to the research hypotheses stated in chapter 1.4.

This thesis builds upon the previous work on the *Multilayered Space-Event Model* (MLSEM) as presented in (Becker et al. 2009a), (Becker et al. 2009b), and (Nagel et al. 2010). **Hypothesis 1.1** posits that the conceptual model of indoor space as defined in these publications is already complete. However, results show that this is false. In contrast, the MLSEM has been substantially further developed in this thesis to address the needs and challenges to indoor space modelling in the context of indoor navigation. The most salient extensions to the MLSEM as proposed in this thesis are listed below:

- Sound definition of *space cells* and *boundary cells* as smallest building blocks for structuring the indoor space;
- Explicit modelling of *non-navigable spaces* (e.g., obstacles) in addition to free spaces;
- Consistent extension of the 3-dimensional geometric-topological representation schema for space cells, boundary cells, and space layers to *two dimensions* (e.g., to model floor plans);
- Sound definition of the *outer space cell* and its spatio-semantic aspects; proof of its relevance for the application of the Poincaré duality and for the notion of a complete state-event model;
- Sound definition of *joint states, joint state transitions*, and a *finite-joint-state machine* supporting localization and tracking;
- Sound definition of *subspace cells* and *subspace layers* as well as hierarchical space structures;
- Design of a *space layer algebra* to combine several space layers in a single view on indoor space and to derive a single graph-based conceptualization (e.g., for path planning);
- Modelling and evaluation of *navigation constraints*;
- Sound definition of *additional concepts* as listed in the above section on the computer representation;
- Correct, consistent, and complete *mathematical formalization*; and
- Design of an ISO-conformant and complete *conceptual data model* and *data exchange format*.

The last two points on this list also provide a positive answer to **hypothesis 1.2** which claims that the MLSEM can be formalized in a mathematically sound manner and can be mapped onto a data model that conforms to existing modelling standards for geographic information.

Based on the presented results and their evaluation against research **question 1.4** to **question 1.9** as well as the comprehensive review of related work it can be deduced that the concepts of the MLSEM are feasible to explain and realize most of the indoor space models for indoor navigation from alternative approaches independent of whether they utilize a regular grid-based or irregular cell-based structuring of the interior environment. The MLSEM hence renders a *generic framework* and *conceptual superset* of these approaches (cf. **hypothesis 1.3**). Since the multilayered graph structure of the MLSEM relies on topological adjacency and overlap relationships, it allows for explaining *place graphs* from other approaches that encode the connectivity or accessibility between places or their hierarchical structures. However, *visibility graphs* encoding the mutual visibility between places are in fact beyond the expressivity of the MLSEM. Approaches applying visibility graphs often recognize the need for building visibility graphs on top of an underlying space model providing connectivity information and hierarchical structures (e.g., Stoffel et al. 2007, Liu & Zlatanova 2011a, Goetz & Zipf 2011). This need can therefore be answered by the MLSEM.

Hypothesis 1.10 asserts that the MLSEM is valid in both two and three dimensions and can be populated from existing (and even purely geometric) building models. Both aspects can be affirmed from the research results.

It can be concluded that the MLSEM as presented in this thesis answers the requirements for indoor space models for the purpose of indoor navigation which have been postulated in chapter 2.4 in order to address the challenges to indoor navigation as identified in chapter 1.2. The main goal of this thesis to design a *spatio-semantic model of indoor space* that meets the multiple challenges and thus can serve as solid foundation for the implementation of indoor navigation systems (cf. **hypothesis 1.3**) has therefore been reached.

8.2 Contributions

The research of this thesis mainly contributes to the field of indoor navigation. The application of the developed concepts can further be envisioned in the field of building information modelling.

Indoor navigation. The principal contribution of this thesis is a mathematical sound approach to the spatio-semantic modelling of indoor space that allows for the integration and joint consideration of 1) different and multiple representations of the physical built environment that reveal its hierarchical and nested structure as well as navigable and non-navigable spaces for various modes of locomotion, 2) different and multiple sensor spaces that reflect diverse indoor localization technologies and methods, 3) different and multiple logical and thematic spaces (e.g., security zones) that are not necessarily aligned with the physical built structure or subject to physical observations, and 4) different and multiple environmental and user-contextual navigation constraints. The indoor space model moreover facilitates the ad-hoc selection and combination of available localization technologies supported by the mobile end-user device and of appropriate representations of navigable, non-navigable, and logical spaces according to the context of individual navigation users. Based on this context-dependent space representation, the complementary tasks of localization, path planning, tracking, and guidance can be realized.

The thesis thus develops a generic, flexible, and context-aware framework for indoor navigation which, to the best knowledge of the author, exceeds the expressivity of alternative approaches to indoor space modelling in literature. The mathematical formalization of the MLSEM defines indoor space at a foundational level and also provides a deterministic graph-based conceptualization. In contrast to most alternative approaches, a complete, consistent, and standardized computer representation of the MLSEM is presented. The MLSEM hence provides a solution to the main problems of existing indoor space models in the field of indoor navigation which have been identified by (Kolodziej & Hjelm 2006) and which have motivated this research.

Building Information Modelling. The modelling of buildings and their interior built reality is subject to the field of building information modelling. Although modelling standards as well as spatial representation paradigms are well established, the MLSEM can contribute to this field by providing a generic space-relation model that facilitates spatio-semantic analyses and simulations beyond indoor navigation. This is illustrated along two examples in the following.

First, logical spaces and their relations to topographic spaces of the interior built environment provide important information about buildings. The MLSEM provides a framework that allows for integrating both views on indoor space in a common model and for reasoning about their spatial relationships based on the multilayered graph and the operations of the space layer algebra (cf. chapter 3.7). For example, the design and spatial configuration of security zones is relevant for security sensitive environments such as airports. Based on the MLSEM, places can be evaluated to which extent they are covered by security zones or, alternatively, the layout of security zones can be planned such that vulnerable places are covered. This can obviously be transferred to arbitrary notions of space. Thus, the MLSEM similarly allows for planning the sensor deployment inside a building to ensure full signal coverage. Likewise, the modelling of dynamically changing disaster areas may help disaster managers to understand the demolition status of a building and to plan evacuation scenarios.

Second, the MLSEM can also be applied to understand the structure of the building itself. For example, the dual graph can be input to network analyses in order to identify the relative importance or influence of a particular space within a building (e.g., by means of degree or closeness centrality). Due to the one-to-one correspondence between dual and primal space, the corresponding primal space geometry of important places can be additionally evaluated to reveal possible bottlenecks, for example, in emergency situations. When applied in the planning phase of a building, the MLSEM can thus help in avoiding such bottlenecks. Moreover, since the layout of the dual graph only depends on the spatial configuration of the space cells in primal space (but not on further semantic aspects), it can be viewed as being a *fingerprint* for a given spatial configuration. Using methods from subgraph matching, the fingerprint of a spatial configuration (e.g., of a building floor or a corridor and its adjacent rooms) can thus be used to identify the same or similar configurations within the same or a different building which then allows for transferring knowledge about spatial settings.

Existing building modelling standards such as IFC or CityGML fail to support these examples since they cannot account for logical spaces and lack a graph-based space representation. Whereas this thesis has demonstrated that the MLSEM can be populated with building data given in such standards, the MLSEM may additionally serve as pattern for extending the standards with a sound graph-based conceptualization of space.

8.3 Outlook and Future Research

The last section of this chapter outlines open issues and recommends directions for future research.

Extension of the semantic conceptualization of indoor space. The MLSEM offers a rich spatial description for space cells and boundary cells in both primal and dual space as well as for their (hierarchical) configuration on one or more space layers. As for the semantic description of the spatial entities inhabiting indoor space, the two feature types *SpaceCell* and *BoundaryCell* are proposed. Although both feature types provide thematic attributes for refining the semantic scope of space cells and boundary cells (e.g., using the *class*, *usage*, and *function* attributes), a further classification of navigation spaces and their semantics may be desirable in order to extend the semantic expressivity of the MLSEM. For example, in topographic space, a general distinction into non-navigable *obstacle spaces* and *navigable spaces* could be introduced. A path search algorithm could then simply identify obstacles on the semantic level in order to filter them in candidate routes. Navigable spaces could be further classified into *vertical* or *horizontal transition spaces* (e.g. elevators, corridors), *end spaces* (e.g., rooms), or *openings* (e.g., doors, windows). From this classification scheme, semantic validation rules could be derived. For example, a transition space could be enforced to connect at least two end spaces on the same or subsequent building floors, whereas end spaces could be required to offer at least one opening in order to be valid. The conceptual data model of the MLSEM as proposed in this thesis lacks a similar semantic expressivity.

Proposals for semantically rich indoor space models have been presented in chapter 2.2.4. They could hence serve as starting point for extending the semantic conceptualization of indoor space within the MLSEM in future research. In (Brown et al. 2012)[83] and (Liu & Zlatanova 2012), two separate semantic models of the indoor navigation space are proposed both of which are already based on the concepts of the MLSEM and thus can be viewed as first steps to extend its semantic expressivity. Whereas (Brown et al. 2012) mainly put focus on the taxonomy of navigation spaces, (Liu & Zlatanova 2012) additionally model relationships between semantic entities and present formal validation rules. However, all of the mentioned approaches merely address the topographic space and neglect the ontology of complementary notions of space. Moreover, a detailed semantic topographic space model may impede the interoperability with existing standards from the field of building information modelling. In contrast, and as shown in chapter 6, the semantic impedance between the MLSEM as proposed in this thesis and standards such as CityGML or IFC is rather low not at least because the concepts *SpaceCell* and *BoundaryCell* remain at a high abstraction level and thus provide suitable conceptual interfaces to these standards.

Implementation of a (prototypical) navigation system. The feasibility of the MLSEM has been illustrated in a large number of examples in this thesis. The two “proof of concept” demonstrations in chapter 7 further have shown that the MLSEM can be instantiated for a real world building and that a corresponding indoor space model can be stored and exchanged between computer systems in a consistent and lossless manner. However, the implementation of a (prototypical) navigation system based on the MLSEM that simultaneously addresses the tasks of localization, path planning, tracking, and guidance was out of scope of this thesis not at least because of the limited time frame. This open challenge therefore needs to be addressed as next step. For this reason, the Institute for Geodesy and Geoinformation Science of the Technische Universität Berlin has defined a research and development

[83] The author has been involved in this research work which is currently refined within a Master’s thesis conducted at the Institute for Geodesy and Geoinformation Science, Technische Universität, Berlin, under the supervision of the author.

project called mobi.*free* which aims at realizing a navigation system for guiding passengers inside connecting stations of the public transportation system. The project scope explicitly includes support for different groups of navigation users and types of locomotion (especially elderly and disabled people) as well as multi-sensor localization. The project thus is meant to serve as lighthouse project for the research work of this thesis but still needs to complete the approval phase.

The practical implementation of concepts developed in this thesis is also addressed in complementary, ongoing research works. For example, a dissertation that parallels this research investigates and implements geometric-topological operations for automatically subspacing the topographic space according to different types of locomotion. First results have been presented in (Khan & Kolbe 2012) (cf. chapter 5.5). Two Master's theses conducted at the Institute for Geodesy and Geoinformation Science are dedicated to the planning of context-dependent best paths through the interior environment based on the MLSEM with focus on 1) multi-level paths and the derivation of human-friendly, hierarchical route instructions using fuzzy methods (cf. Lorenz et al. 2006, chapter 2.2.2) and 2) the evaluation of navigation constraints against various user profiles. However, at the time of writing this thesis, results have not been available yet. Additional work has to be carried out to practically realize further conceptual and mathematical aspects of the MLSEM such as the proposed space layer algebra or the finite-joint-state machine.

Cell-based and non-cell-based localization. The MLSEM follows a cell-based hybrid approach to indoor space modelling. The description of indoor space hence requires knowledge about the shape, size, and location of the space cells on a space layer. As shown in this thesis, the cell-based approach is feasible to model topographic spaces and subspaces thereof but also to represent logical and sensor spaces due to the generic notion of space underlying the MLSEM. Regarding sensor spaces, the partitioning of indoor space according to, for example, the reception areas or signal strength bands of sensors or transmitters can be used to obtain suitable space cells. Likewise, space cells can denote functional spaces such as the partition of space where the user must be physically located in order to interact with an object/sensor (e.g., when using QR-codes for localization) (cf. chapter 3.3). Although this yields a high flexibility, not all existing localization techniques and methods admit a cell-based mapping. For example, techniques such as *Time of Arrival* (TOA), *Time Difference of Arrival* (TDOA), or *Angle of Arrival* (AOA) rather rely on measuring the propagation time of signals or the direction of propagation (e.g., using technologies such as UWB or Wi-Fi). Based on obtained distances and angles, often geometric functions (e.g., triangulation, trilateration) are then applied to derive absolute or relative position estimates. Likewise, techniques such as dead reckoning or photogrammetric localization can neither be expressed in terms of space cells. Since sophisticated non-cell-based localization approaches have been, and are being, proposed in literature, an important field of future research is therefore their integration with the MLSEM and the proposed cell-based localization. A minimum requirement for all localization techniques is a rich navigation space model for the evaluation and communication of position estimates, which is at least answered by the MLSEM.

A challenge to the cell-based mapping of signal propagation characteristics are the hard boundaries of space cells. For example, Wi-Fi radio wave propagation is subject to signal attenuation not only when passing through materials of the built structure (e.g., wood, reinforced concrete) but also severely when passing through people. The latter will cause the signal reception area of a Wi-Fi transmitter to dynamically shrink with an increasing number of people in the vicinity of the transmitter. However, the boundaries of space cells are defined to be fixed within the MLSEM which hinders the support for dynamically shrinking or growing space cells. It thus is recommended that the maximum measurable reception area is chosen as shape for a corresponding sensor space cell since the size of the space cell correlates to the uncertainty about the absolute position of a navigation user within the cell (cf. chapter 3.3). Future research should explore the modelling of uncertain and hence soft space cell boundaries. For example, the difference between the maximum and minimum measureable reception area could conceptually be modelled as uncertainty band and mapped onto one or more space cells associated with differing likelihoods of sensor observations. It is important to note that the MLSEM makes no assumption on the determination of the reception area (and thus the shape of the sensor space cell). Possible methods include fingerprinting (cf. chapter 7.2.2) or radio propagation models which face separate advantages and disadvantages. For example, radio propagation models have to account for further signal propagation characteristics such as noise, interference, or multipath effects which affect the shape and size of the space cell.

The precision of the absolute position determination of navigation users based on the finite-joint-state machine presented in chapter 3.3 is limited by the uncertainty area of the active joint state of navigation. Besides adding further sensor space layers in order to yield highly overlapping space cells that reduce the spatial extent of the uncertainty area, further methods for increasing the precision were out of scope of this thesis but render a task for future research. This work can draw from already existing proposals in literature. For example, (Jensen et al. 2009) and (Jensen et al. 2010) (cf. chapter 2.2.3) present a cell-based localization method using RFID sensors which applies the maximum speed of the moving person or object to constrain the absolute location at a finer granularity than that of RFID cells. Moreover, the authors use the same criterion to predict the possible location of a navigation user who cannot be associated with an RFID cell in a deterministic manner. This approach can be directly utilized for the MLSEM which also supports deterministic and non-deterministic active states of a navigation user. In (Liao et al. 2003), Bayesian inference is used to increase the precision of localization based on noisy and sparse sensor information (e.g., RFID or Bluetooth). A general particle filter is applied to produce position estimates based on 1) likelihoods of sensor observations that are averaged over mutually non-overlapping location patches and 2) a Gaussian motion model of the user dynamics taking into account the velocity of the user as well as transition probabilities on the edges of the navigation graph which are derived from motion pattern learning. A salient aspect of the approach is that the position estimates are projected onto the navigation graph which is realized as Voronoi diagram of the environment (cf. chapter 2.1.2) in order to increase the efficiency and robustness of the approach (e.g., in case sensors frequently fail to detect the navigation user). The Voronoi tracking approach of (Liao et al. 2003) can also be translated into the domain of the MLSEM. Since the MLSEM does not apply Voronoi diagrams, the Poincaré dual graph of topographic space has to be spatially embedded in free space in order to create equal conditions. The location patches for averaging the likelihoods of sensor observations can then be modelled as sensor space cells. Whereas the approach of (Liao et al. 2003) uses a more powerful motion model than the approach of (Jensen et al. 2010), the latter better accounts for the unconstrained movement possibilities in indoor spaces. Nevertheless, both approaches are strong starting points for future research and rely on a precise model of the navigation space which can answered by the MLSEM.

Outdoor navigation. The scope of this thesis was intentionally limited to the modelling of indoor space for the purpose of indoor navigation. Nevertheless, supporting the seamless transition between indoor and outdoor in a common framework is a field of intensive research (e.g., see Lee & Zlatanova 2008, Ruppel et al. 2008, Giudice et al. 2010, Yang & Worboys 2011, Vanclooster & Maeyer 2012) and also renders a future research topic in the context of the MLSEM. The conceptual data model proposed in this thesis already foresees the modelling of transfers between the elements of two navigation graphs in a separate *InterModelGraph* (cf. chapter 4.4.1.6). This graph structure hence represents the linkage between two navigation spaces and can be used to define hand-over points between corresponding navigation systems. The *InterModelGraph* is feasible to connect two instances of the MLSEM but likewise can be used to link the MLSEM with an outdoor navigation graph. One aspect of future work is therefore the evaluation of the *InterModelGraph* concept in an indoor/outdoor context.

The MLSEM is a generic space-relation model. The notion of space cells and space layers is thus not restricted to the interior environment but also allows for structuring the outdoor space. (Boguslawski & Gold 2011) have already shown how their dual-graph-based approach can be used to model the surroundings of buildings based on thin 3-dimensional cells in order to derive an integrated representation of indoor and outdoor spaces (cf. chapter 2.2.3). Since the model of (Boguslawski & Gold 2011) can be explained based on the research of this thesis (cf. chapter 3), a similar approach can be applied in the context of the MLSEM. The MLSEM is even more expressive because it facilitates both a 2-dimensional and 3-dimensional space representation. For example, it can be used to model a 2-dimensional digital road map with space cells representing the roads and junctions. Based on the spatial configuration of the space cells and the formalism developed in this thesis, a corresponding road network can be derived in a deterministic manner. Likewise, roads and junctions can be represented as 3-dimensional space cells which not necessarily have to be thin but rather can be extruded to the (actual or virtual) vertical clearance above the road which again results in a deterministic road network. Further concepts of the MLSEM are also applicable. For example, roads can be subspaced into separate lanes and toll sections may be denoted as logical spaces. Moreover, constraints such as speed limits or prohibited maneuvers can be expressed using the proposed navigation constraint model (possibly involving additional constraint conditions). Due to the generic notion of space, space cells representing pedestrian pathways or indoor spaces can be integrated into the same MLSEM instance which would result in a single navigation graph seamlessly connecting the indoor and outdoor world. Likewise, and in contrast to (Boguslawski & Gold 2011), indoor and outdoor spaces can be modelled in separate MLSEM instances (possibly with different dimensions) and then be linked through the *InterModelGraph* concept. A second aspect of future research in this context is therefore the evaluation of the feasibility of modelling outdoor spaces with the MLSEM.

A third important aspect is the relation of the MLSEM to existing standards for outdoor navigation such as the *Geographic Data Files* (GDF, ISO/DIS 14825:2011) and their representation of outdoor spaces. According to (Rüetschi 2007), two fundamental types of spaces can be generally distinguished, namely *network space* and *scene space*. Whereas environments that exhibit a clear network structure are subsumed under the notion of network space, environments without an obvious network structure are called scene spaces. Models for outdoor navigation mostly follow the idea of network space. For example, GDF differs between point-like and line-like features in transportation systems and builds the navigation network upon their connectivity. Roads are identified as linear spatial features and mapped onto graph edges, whereas junctions are seen as point objects and carried to graph nodes. This differentiation is only applied on a conceptual level and corresponds to an intuitive human perception and understanding of road spaces. In contrast, the space representation of the MLSEM (and of most of the presented indoor space models) falls into the category of scene space since it describes regions of space (called *scenes* by Rüetschi 2007) which may be hierarchically grouped and connected with one another. The concept of scene space is therefore closer to the human cognitive model of indoor space. When applying the MLSEM to road spaces, both roads and junctions would consequently be carried to graph nodes with the edges denoting their transitions but not linear spatial features. The fundamentally different notions of space underlying GDF and the MLSEM hence also lead to different graph representations of the road space. Although both graph structures are obviously suitable to perform path planning, a comprehensive understanding of the implications of these differences requires more investigations in future research.

International standard for indoor navigation. At the time of writing this thesis, the OGC is in the process of developing an international standard for indoor navigation. A corresponding *Standards Working Group* (SWG) called *IndoorGML SWG*[84] was formed within OGC in March 2012 whose main goal, as stated in the charter document of this SWG, is cited in the following:

> *"The purpose of this IndoorGML Standard Working Group is to develop an application schema of OGC GML and progress the document to the state of an adopted OGC standard. The goal of this candidate standard is to establish a common schema framework for indoor navigation applications. This SWG will start from the discussion paper (OGC 10-191r1, Requirements and Space-Event Modeling for Indoor Navigation* [that is, (Nagel et al. 2010)]*), which summarizes the requirements and basic idea of a standard for indoor navigation."* (OGC Doc. No. 12-005, p. 2)[85].

This excerpt of the SWG charter nicely documents the relevance and impact of the previous research on the MLSEM. Since standardization has been motivated in chapter 1 as a fundamental prerequisite for the success of the field of indoor navigation, the author would like to conclude this thesis with the hope that the substantial further development of the MLSEM as carried out in this thesis as well as the presented findings and results will be beneficial in and contribute to this standardization process.

[84] See http://www.opengeospatial.org/projects/groups/indoorgmlswg for more information. The term *IndoorGML* is adopted from Becker et al. 2009b where it is used as name for the initial GML application schema of the MLSEM.

[85] See https://portal.opengeospatial.org/files/?artifact_id=47562.

Appendix A
Basic Notions and Concepts from Topology

This appendix recaps basic notions and concepts from the mathematical field of *topology* as applied in the context of this thesis and fixes important terms, symbols, and definitions. The definitions and results presented here are mostly adapted from (Munkres 1984), (Hatcher 2008), (Lee 2011), (Jänich 2012), and (Morris 2012).

A.1 Point-set Topology

Point-set topology is closely linked to set theory and deals with the definition of the abstract nature and characteristics of *topological spaces*. The notion of topological space is defined in the following by means of rather simple and abstract axioms. On the one hand, this abstract notion provides a useful "umbrella" concept which serves as foundation for vastly different branches of topology. On the other hand, because of its simplicity and elasticity, the set-theoretic notion of topological space is often seen to be generalized far beyond an intuitive meaning. Thus, additional restrictions will be imposed on topological spaces in the course of this appendix.

Definition A.1 (Topological space, open set). A *topological space* is a pair $(X, \mathcal{T})$ where X is a set and $\mathcal{T}$ is a family of subsets of X, called *open sets*, such that

(i) X and the empty set, $\emptyset$, belong to $\mathcal{T}$,
(ii) the union of any (finite or infinite) collection of sets in $\mathcal{T}$ belongs to $\mathcal{T}$, and
(iii) the intersection of any finite collection of sets in $\mathcal{T}$ belongs to $\mathcal{T}$.

$\mathcal{T}$ is called the *topology* on X. If the topology on X is understood, it is said that X is a topological space and $\mathcal{T}$ is conventionally omitted from the notation. The elements of a topological space are called *points* regardless of their actual nature. The members of $\mathcal{T}$ are called *open sets*. The complements to the open sets are said to be *closed sets* and are defined as follows.

Definition A.2 (Closed set). A subset $A \subset X$ of a topological space $(X, \mathcal{T})$ is *closed* if its complement $X \setminus A$ is open in $(X, \mathcal{T})$.

This definition implies that the union of a finite collection of closed sets and intersections of an arbitrary number of closed sets are closed. In contrast, both the intersection of infinitely many open sets and the union of infinitely many closed sets may be open or closed.

A fundamental concept of topological spaces is the notion of *neighbourhood* which expresses a *closeness* relation between a point and a set without the need for a quantitative measure such as a distance function.

Definition A.3 (Neighbourhood, open neighbourhood). A subset N of X is defined as *neighbourhood* of a point x in N if there exists an open set $O \in \mathcal{T}$such that $x \in O \subseteq N$.

If $N \in \mathcal{T}$, then N is called an *open neighbourhood* of every point x in N which is more restricted in such that every neighbourhood contains an open neighbourhood. The open neighbourhoods determine the family of open sets O, and hence the topology $\mathcal{T}$.

Definition A.4 (Limit point). A point $x \in X$ is said to be a *limit point* of a subset A of X, if every open neighbourhood of x contains a point of A different from x.

A limit point of the subset A is said to be arbitrarily close to A. The concepts of neighbourhood and limit point allow for rephrasing the conditions for open and closed sets. A subset $A \subset X$ is *open*, iff (if and only if) it contains an open neighbourhood of every point $x \in A$. The subset $A \subset X$ is said to be *closed* iff it contains the set $L(A)$ of all its limit points, $A \cup L(A)$. If A is both open and closed in $(X, \mathcal{T})$, then it is said to be *clopen*. In every topological space, both X and the empty set, $\emptyset$, are clopen.

Definition A.5 (Closure, interior, exterior, boundary). Based on the notions of neighbourhood as well as open and closed sets, the *closure*, *interior*, *exterior* and *boundary* of any subset $A \subset X$ can be given as follows:

(i) The *closure* $\bar{A}$ is the set $\bar{A} = \bigcap\{C | A \subset C$ and C is closed in $X\}$ (the smallest closed set containing A). Equivalently, $\bar{A} = A \cup L(A)$. From this it follows that all points in the closure are said to be arbitrarily close to A.

(ii) The *interior* $Int(A)$ is the set $Int(A) = \bigcup\{O | O \subset A$ and O is open in $X\}$ (the union of all open sets which lie entirely in A).

(iii) The *exterior* $Ext(A)$ is the set $Ext(A) = X \setminus \bar{A}$ (the complement of the closure of A in X).

(iv) The *boundary* ∂A is the set $\partial A = \bar{A} \setminus Int(A)$ (the closure of A without the interior of A). It follows that a closed set contains its boundary.

It obviously follows from definition A.5 that the closure of a subset $A \subset X$ is always a closed subset, and both the interior and the exterior of A are always open. The boundary can equivalently be defined as the intersection of the closure of A with the closure of its complement in X, i.e., $\partial A = \bar{A} \cap \overline{(X \setminus A)}$. Since the intersection of two closed sets is closed itself, the boundary is also closed.

The interior, exterior, and boundary of the set A are disjoint sets. Thus the set A divides X into three disjoint partitions, and equivalently $X = Int(A) \cup Ext(A) \cup \partial A$ where all unions are disjoint. The interior points are always contained in A, whereas exterior points never lie in A.

Instead of enumerating all contained open sets, the topology $\mathcal{T}$ can also be defined by describing a smaller collection $\mathfrak{B}$ of open subsets which is said to generate the topology.

Definition A.6 (Basis, neighbourhood basis). A collection $\mathfrak{B}$ of open subsets of X is called a *basis* for the topology $\mathcal{T}$ if every open set $O \in \mathcal{T}$ is a union of members of $\mathfrak{B}$. A *neighbourhood basis* $\mathcal{U}_x$ for $x \in X$ is a set of open neighbourhoods such that every open neighbourhood of x contains a neighbourhood from $\mathcal{U}_x$. A neighbourhood basis is also said to be a *local basis*.

The countability properties of topological spaces provide a notion for the limitation on size of the space. In a sense, they restrict the number of open sets the space can have. A topological space can be *separable*, *first-countable* or *second-countable*. The first property of a topological space being separable is based on the definition of a *dense subset* which is given first.

Definition A.7 (Dense subset). A subset A of a topological space $(X, \mathcal{T})$ is said to be *dense* in X or *everywhere dense* in X if every point $x \in X$ is a point or a limit point of A, and thus $X = \bar{A}$.

Definition A.8 (Separable, first-countable, second-countable). A topological space $(X, \mathcal{T})$ is called

(i) *separable* if it has a countable dense subset,

(ii) *first-countable* if at each point $x \in X$ there is a *countable* local basis, i.e., the number of sets in the neighbourhood basis $\mathcal{U}_x$ is *countable*, and

(iii) *second-countable* if it has a *countable* basis $\mathfrak{B}$.

All second-countable spaces are both first-countable and separable. But a separable space need not be even first-countable (Steen & Seebach 1995).

Definition A.9 (Subspace, subspace topology). If Y is a non-empty subset of a topological space $(X, \mathcal{T})$, then the topological space $(Y, \mathcal{T}_Y)$ is said to be a *subspace* of $(X, \mathcal{T})$ with the *subspace topology* $\mathcal{T}_Y$ whose open sets are of the form $\mathcal{T}_Y = \{O \cap Y : O \in \mathcal{T}\}$. The subspace topology is also called the *induced topology*.

If a property of a topological space is also possessed by all its subspaces, then this property is said to be *hereditary*. For example, a subspace of a first-countable (second-countable) space is first-countable (second-countable).

Two topological spaces are topologically *equivalent* if there exists a *homeomorphism* of one onto the other. Homeomorphism is the most fundamental relation in topology and builds upon the notion of *continuity* of topological spaces which is expressed in terms of open sets.

Definition A.10 (Continuous map). A function $f: X \rightarrow Y$ between two topological spaces $(X, \mathcal{T})$ and $(Y, \mathcal{T}_1)$ is continuous iff for each open set U in Y, the inverse image $f^{-1}(U)$ is open in X, and thus $f^{-1}(U) \in \mathcal{T}$. Continuous functions are also called *maps* or *mappings*.

Definition A.11 (Homeomorphism). A function $f: X \rightarrow Y$ between two topological spaces $(X, \mathcal{T})$ and $(Y, \mathcal{T}_1)$ is called *homeomorphism* if f is a bijection and both f and f^{-1} are continuous. If such a map f exists, the topological spaces are said to be *homeomorphic* which is denoted $(X, \mathcal{T}) \cong (Y, \mathcal{T}_1)$.

A property of a topological space which is preserved under homeomorphisms is called *topological property* or *topological invariant*. Invariant properties are, e.g., neighbourhoods, connectedness, separation, compactness, and embedding.

Definition A.12 (Connectedness, path-connectedness). A topological space $(X, \mathcal{T})$ is said to be *disconnected* iff there are non-empty disjoint open subsets $A, B \subset X$ whose union represents X, i.e., $A \cap B = \emptyset$ and $X = A \cup B$. It is called *connected* iff it cannot be decomposed as the union of two such sets. Equivalently, it is connected if the only clopen subsets of X are X and $\emptyset$ (Morris 2012).

A topological space is called *path-connected* if for any two points $x_0, x_1 \in X$ there exists a path joining x_0 to x_1, i.e., a continuous map $f: [0,1] \rightarrow X$ with $f(0) = x_0$ and $f(1) = x_1$. The notion of path-connectedness is stronger in such that every path-connected topological space is also connected.

Separation properties (also called separation axioms) provide a measure of how rich a given topology is by classifying topological spaces according to the degree to which distinct points or closed sets may be *separated* by open sets. Whereas the countability properties restrict the number of open sets, the separation properties ensure that a topological space has enough open subsets, for example, to ensure that closeness expressed through neighbourhoods does have the same intuitive meaning as provided by a quantitative measure. The following list only provides a subset of common separation properties ordered by decreasing generality.

Definition A.13 (Separation properties). A topological space $(X, \mathcal{T})$ is said to be

(i) a T_1 or *Frechét* space if any two distinct points $x, y \in X$ are separated, i.e. there exist open sets $U_x, V_y \in \mathcal{T}$ containing x, y respectively such that $x \notin V_y$, and $y \notin U_x$,

(ii) a T_2 or *Hausdorff* space if any two distinct points $x, y \in X$ are separated by open neighbourhoods, i.e. there exist disjoint open sets $U_x, V_y \in \mathcal{T}$ containing x, y respectively,

(iii) a T_4 or *normal* space if it is *Hausdorff* and any two closed disjoint subsets possess disjoint open neighbourhoods.

Definition A.14 (Open covering, compactness). A family of open sets $\{O_i\}, i \in I$ in a topological space $(X, \mathcal{T})$ is called an *open covering* of X if $X = \bigcup_{i \in I} O_i$. For a finite subfamily $O_{i_1}, O_{i_2}, \dots, O_{i_n}$, of $\{O_i\}, i \in I$ is called a *finite subcovering* if $X = O_{i_1} \cup O_{i_2} \cup \dots \cup O_{i_n}$. The topological space $(X, \mathcal{T})$ is called *compact* if every open covering of X has a finite subcovering.

Compactness can be thought of as a topological generalization of finiteness (Morris 2012). For example, spaces with finitely many points or whose topology only has finitely many open sets are obviously compact according to the above definition. A counterexample for an infinitely large space is the half-open interval $[0, \infty)$ which correspondingly is non-compact. However, the property of being compact is only defined in terms of open sets, and thus only depends on the topology on a space.

The main theorem on compactness says that the image of a continuous map of a compact space is compact (for a proof see Lee 2011).

Theorem A.15 (Main theorem on compactness). Let $(X, \mathcal{T})$ and $(Y, \mathcal{T}_Y)$ be topological spaces, and let $f: X \rightarrow Y$ be a continuous map. If X is compact, then $f(X)$ is compact (Lee 2011).

A subset A of the topological space $(X, \mathcal{T})$ is said to be compact if every open covering of A by open subsets of X has a finite subcovering. If A is considered as subspace of $(X, \mathcal{T})$ with the induced topology $\mathcal{T}_A$, then the topological space $(A, \mathcal{T}_A)$ is compact.

Although many topological spaces are not compact as whole space, they may look like a compact space at a small enough scale. Spaces satisfying this property are called *locally compact*.

Definition A.16 (Locally compact). A topological space $(X, \mathcal{T})$ is said to be *locally compact* if for every point $x \in X$ there exists a compact subset of X containing a neighbourhood of x.

An important set of topological spaces discussed in literature is the class of *locally compact Hausdorff spaces*. Prominent instances of locally compact Hausdorff spaces are *Euclidean spaces* which are introduced in the further course of this chapter and for which the property of being locally compact will be revisited.

Definition A.17 (Embedding). A both injective and continuous map $f: X \to Y$ is called an *embedding* of the topological space $(X, \mathcal{T})$ into $(Y, \mathcal{T}_Y)$ if it induces a homeomorphism between X and the subspace $f(X) \subseteq Y$. Since the image $f(X)$ is contained in Y, the embedding $f: X \to Y$ allows X to be treated as subspace of Y with the topology induced by Y.

Metric spaces. An important subclass of topological spaces is the class of *metric spaces* which were first introduced in 1906 by Maurice Fréchet and further elaborated by Felix Hausdorff in 1914 (Morris 2012). In general topological spaces, the notion of closeness allows to express whether a point $x \in X$ is infinitely near to a set $A \subset X$ (i.e., in the closure of A), or whether the set A is a smaller neighbourhood for x than the set B. The comparison of neighbourhoods of different points as well as the notion of closeness between points requires an additional structure on the topological space. This additional structure is called *uniform structure* and a topological space having a uniform structure is said to be a *uniform space*. A natural way to describe the closeness between points is in terms of a distance function. The notion of *distance* is inherent to metric spaces.

Definition A.18 (Metric, metric space). A *metric* on a set X is a function $d: X \times X \to \mathbb{R}$ such that for $x, y \in X$

(i) $d(x, y) \geq 0$ and $d(x, y) = 0$ iff $x = y$ (*positivity*),
(ii) $d(x, y) = d(y, x)$ (*symmetry*), and
(iii) $d(x, y) \leq d(x, z) + d(z, y)$ for all $x, y, z \in X$ (*triangle inequality*).

If d is a metric on X, then (X, d) is called a *metric space* and $d(x, y)$ is said to be the *distance* between the points x and y.

Definition A.19 (Open and closed ball). If (X, d) is a metric space and r any positive real number, then the set $B(x, r) = \{y \in X \mid d(x, y) < r\}$ is called the *open ball* centred at x of radius r. The set $\bar{B}(x, r) = \{y \in X \mid d(x, y) \leq r\}$ is said to be the *closed ball* about x.

A subset $O \subset X$ is called *open* iff for every $x \in O$ there exists an $r > 0$ such that an open ball $B(x, r)$ is contained in O. Thus, each open ball in a metric space (X, d) is an open set, and conversely, each closed ball is a closed set.

From this and definition A.3 immediately follows that a neighbourhood of a point $x \in X$ is any subset N of X that contains an open ball about x as subset. Any open ball $B(x, r)$ about x is an open neighbourhood of x. Since the open sets of a metric space can be defined using the notion of open balls, the following proposition holds.

Proposition A.20. The collection of open balls $B(x, r)$ in (X, d) for $r > 0$ and $x \in X$ forms a basis for a topology $\mathcal{T}$ on X.

The topology $\mathcal{T}$ is said to be the *topology induced by the metric d*, or alternatively the *metric topology*. Different metrics on the same set X which induce the same topology are called *equivalent*.

In general, the approach of point-set topology is to avoid the notion of distance and to only focus on the open sets, and thus the topology $\mathcal{T}$ itself, in order to describe properties of a topological space. However, due to this additional structure, metric spaces have certain nice properties which are important in this thesis and discussed in the following.

Proposition A.21. Every metric space is first countable. Every separable metric space is second-countable. Thus, for metric spaces the properties of being separable and second-countable are equivalent.

Proposition A.22. Every metric space is both Hausdorff (T_2) and normal (T_4).

Proposition A.23. Let (X, d) be any metric space and $\mathcal{T}$ the topology induced on X by d. For any subset Y of X, a metric d_1 on Y can be obtained by restricting d, i.e., $d_1(x, y) = d(x, y)$ for all $x, y \in Y$. Then, the topology $\mathcal{T}_1$ induced on Y by the metric d_1 is the same as the subspace topology $\mathcal{T}_2$ on Y (which is induced by $\mathcal{T}$ on X), i.e., $\mathcal{T}_1 = \mathcal{T}_2$.

Definition A.24 (Bounded and totally bounded subset). A subset A of a metric space (X, d) is said to be *bounded* if A can be covered by a single open ball such that $A \subseteq B(x, r)$ for $x \in X$ and $r > 0$. The subset A is called *totally bounded* if A can be covered by a finite number of open balls, that is $A \subseteq \bigcup_{i=1}^{n} B(x_i, r)$ for $x_1, \dots, x_n \in X$ and $r > 0$.

Every totally bounded metric space is also bounded. Every subspace of a totally bounded metric space is totally bounded which makes it a hereditary property. But the property of being totally bounded is not a topological property, and thus not preserved under homeomorphisms.

The *boundedness* of a subset A of a metric space (X, d) denotes a measure for how far the subset extends in (X, d). Boundedness depends on the metric and is not to be confused with finiteness. The compactness of a subset, which was introduced as topological counterpart of finiteness, is only expressed on the topology and hence on the open subsets of a topological space (cf. definition A.14). Finiteness implies boundedness, but not conversely. This relation is expressed in the following proposition.

Proposition A.25. Let A be a compact subset of a metric space (X, d). Then A is closed and bounded (Morris 2012).

Omitting the metric structure, this implies more generally that a compact subset of a Hausdorff topological space is closed. Furthermore, a closed subset of a compact space is compact in the subspace topology. For *Euclidean space* with its richer topological structure, the *Heine-Borel theorem* allows for rephrasing the proposition A.25 to determine compact subsets (cf. theorem A.30).

Euclidean space and Euclidean topology. An essential set of metric spaces is the family of *normed vector spaces*.

Definition A.26 (Norm, normed vector space). For a vector space V over the field of real numbers, a *norm* is a map $p: V \to \mathbb{R}$ such that for all $x, y \in V$ and k in the field

(i) $p(x) \geq 0$ and $p(x) = 0$ iff $x = 0$ (*positivity*),
(ii) $p(x + y) \leq p(x) + p(y)$ (*triangle inequality*), and
(iii) $p(\mathrm{k}x) = |\mathrm{k}|p(x)$ (*positive scalability*) with $|\cdot|$ being the absolute value of a member in the field.

The norm is usually denoted $\|\cdot\|$. A *normed vector space* is a pair $(V, \|\cdot\|)$ where V is a vector space with a norm $\|\cdot\|$. It easily follows from definition A.18 that for each normed vector space, the norm induces a corresponding metric d on the set V which is given by $d(x, y) = \|x - y\|$, for $x, y \in V$ (Morris 2012).

There are several natural ways to introduce a norm on the n-dimensional real vector space $\mathbb{R}^n$. For example, for any real number $p \geq 1$, the *p-norm* of a vector $x \in \mathbb{R}^n$ is given by

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}.$$

For $p = 1$, the resulting 1-norm is called *Manhattan norm* and yields the *Manhattan distance*. The *usual norm* on $\mathbb{R}^n$ is the 2-norm or *Euclidean norm* which is based on the standard *inner product* of a vector $x \in \mathbb{R}^n$ with itself, also called the (Euclidean) *length* of the vector, and defined as

$$\|x\|_2 = \sqrt{\langle x, x \rangle} = \sqrt{\sum_{i=1}^{n} (x_i)^2}.$$

The metric induced by the Euclidean norm is called the *Euclidean metric* which defines a distance function for any two vectors $x, y \in \mathbb{R}^n$ as

$$d(x,y) = \|x-y\|_2 = \sqrt{(x_1-y_1)^2 + \ \ldots + (x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}.$$

The n-dimensional real vector space $\mathbb{R}^n$ becomes a metric space by applying this Euclidean structure. This thesis is concerned only with the Euclidean structure on $\mathbb{R}^n$ which is used in the following to define Euclidean n-space.

Definition A.27 (Euclidean n-space). The n-dimensional real vector space $\mathbb{R}^n$ with the Euclidean structure is called *Euclidean n-space* for all non-negative integers n.

In literature, Euclidean n-space is sometimes denoted $\mathbb{E}^n$. This thesis follows the common notation to use the symbol $\mathbb{R}^n$ itself to refer to Euclidean n-space with the Euclidean structure being understood. By convention, $\mathbb{R}^0$ is the singleton set $\{0\}$.

Based on the Euclidean norm and definition A.19, the n-dimensional open balls of the Euclidean n-space can be defined as follows.

Definition A.28 (Open n-ball, open unit n-ball). An *open n-ball* centred at a point $x \in \mathbb{R}^n$ is the set of points $B^n(x,r) = \{y \in \mathbb{R}^n \mid \|y-x\| < r\}$ for some $r \in \mathbb{R}^+$. The *open unit n-ball* about the origin is the set $\mathbb{B}^n = \{x \in \mathbb{R}^n \mid \|x\| < 1\}$.

The closed n-ball $\bar{B}^n(x,r) \subset \mathbb{R}^n$ is given by $\bar{B}^n(x,r) = \{y \in \mathbb{R}^n \mid \|y-x\| \leq r\}$ and the closed unit n-ball is defined as $\bar{\mathbb{B}}^n = \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$. Intuitively, the concept of the open and closed n-ball is analogous to the open and closed interval in $\mathbb{R}^n$ respectively.

The metric topology on $\mathbb{R}^n$ which is induced by the Euclidean metric and whose basis is given by the set of open n-balls (cf. proposition A.20) is called the *Euclidean topology*. The Euclidean topology is also referred to as the *standard topology* or *usual topology* on the Euclidean space. Since Euclidean space is a normed vector space and hence a metric space, it is also a topological space. Thus, it fulfils the axioms provided by definition A.1 and possesses all properties of metric spaces. Due to its rich topological structure, Euclidean space is even a stronger form of a metric space.

The usual topology on $\mathbb{R}^n$ induces a topology on every subset as well which becomes a subspace in its own right (cf. proposition A.23). Unless explicitly stated otherwise, in this thesis the usual topology is assumed for subsets of $\mathbb{R}^n$.

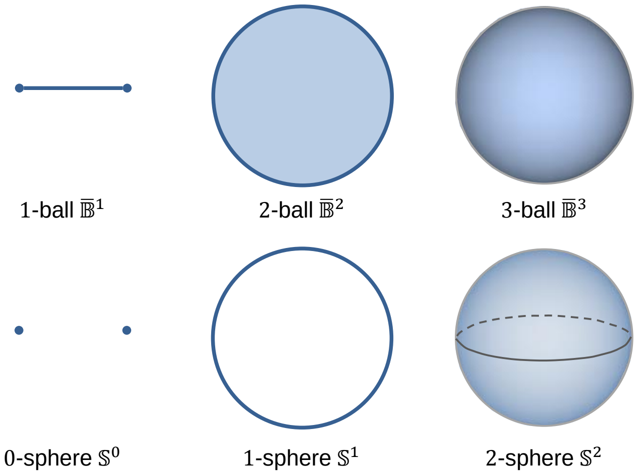

1-ball $\bar{\mathbb{B}}^1$
2-ball $\bar{\mathbb{B}}^2$
3-ball $\bar{\mathbb{B}}^3$
0-sphere $\mathbb{S}^0$
1-sphere $\mathbb{S}^1$
2-sphere $\mathbb{S}^2$

Figure 264: The unit 1-ball $\bar{\mathbb{B}}^1$, the unit 2-ball $\bar{\mathbb{B}}^2$ (also called closed disk), and the unit 3-ball $\bar{\mathbb{B}}^3$ (top). In $\mathbb{R}^n$, the boundary of a unit n-ball is an $(n-1)$-dimensional sphere (bottom). Note that balls may have different shapes in different metric spaces.

Any open n-ball in $\mathbb{R}^n$ is homeomorphic to any other open n-ball. Analogously, all closed n-balls in $\mathbb{R}^n$ are homeomorphic. Moreover, an open n-ball with the usual topology is homeomorphic to $\mathbb{R}^n$ itself.

Proposition A.29. Euclidean n-space $\mathbb{R}^n$ with the standard topology is second-countable.

This proposition is easily verified by restricting the uncountable basis of all open n-balls to the set of open n-balls of rational centre and radius. The rational numbers form a countable dense subset of $\mathbb{R}$, and so does the set of vectors $\{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_i \in \mathbb{Q}, i = 1, .., n\}$ for $\mathbb{R}^n$. Thus, Euclidean n-space is separable. Since the restricted set of open n-balls is countable and still forms a basis, Euclidean n-space is second-countable, which also follows from proposition A.21.

An important theorem for determining compact subsets of $\mathbb{R}^n$ is the *Heine-Borel theorem*.

Theorem A.30 (Heine-Borel theorem). A subspace $A \subset \mathbb{R}^n$ is compact iff it is closed and bounded.

All closed intervals $[a, b]$, for $a, b \in \mathbb{R}$ with $a < b$ are closed and bounded subsets of $\mathbb{R}$ and, thus, compact spaces according to the Heine-Borel theorem (Morris 2012). This illustrates that the topological notion of compactness is more general than finiteness as $[a, b]$ are infinite sets.

Any open or closed n-ball in $\mathbb{R}^n$ is contained inside an open n-ball $B^n(x, r)$ with $x \in \mathbb{R}^n$ and finite radius $r < \infty$ (also follows from proposition A.22) and hence is a bounded subset of $\mathbb{R}^n$. Each closed n-ball is also a closed subset of $\mathbb{R}^n$, and it follows from the Heine-Borel theorem that all closed n-balls in $\mathbb{R}^n$ are compact spaces with the induced usual topology but the open n-balls are not. Any closed subset of a closed n-ball in $\mathbb{R}^n$ is compact in the usual topology too (cf. discussion of proposition A.25). For example, the $(n - 1)$-sphere as boundary of a closed n-ball is compact in $\mathbb{R}^n$.

As the open unit n-ball $\mathbb{B}^n$ is non-compact and $\mathbb{R}^n \cong \mathbb{B}^n$, it follows that $\mathbb{R}^n$ is non-compact too because compactness is invariant under homeomorphism. Non-compact topological spaces can be made compact by embedding them in compact spaces. Such a *compactification* of a non-compact space is often useful to open up the space for the additional properties and structures of compact spaces.

Definition A.31 (Compactification). An embedding $f: X \to Y$ of the topological space $(X, \mathcal{T})$ into the compact space $(Y, \mathcal{T}_Y)$ is called a *compactification* of X iff $f(X) \subset Y$ is a dense subset of Y.

There are different sorts of compactifications of non-compact topological spaces. The smallest compactification is the *Alexandroff* or *one-point compactification* on the class of locally compact Hausdorff spaces. The idea of the *Alexandroff compactification* is to adjoin a single point to the non-compact space such that the resulting space is compact. This extra single point is often called *point at infinity* and denoted $\{\infty\}$.

Definition A.32 (Alexandroff compactification). The *Alexandroff* or *one-point compactification* of a locally compact, non-compact Hausdorff space $(X, \mathcal{T})$ is given by $\hat{X} = (X \cup \{\infty\}, \mathcal{T}_1)$, where $\infty \notin X$, and $\mathcal{T}_1 = \mathcal{T} \cup \{(X \setminus C) \cup \{\infty\} \mid C \subset X \text{ is compact}\}$.

The topology $\mathcal{T}_1$ on the set $X \cup \{\infty\}$ results from the union of the open sets $\mathcal{T}$ in X and the sets of the form $(X \setminus C) \cup \{\infty\}$. Since X is required to be Hausdorff, it easily follows from proposition A.25 that any compact subset $C \subset X$ is closed in X and hence $X \setminus C$ is an open subset in X. For any open covering of $\hat{X}$, one set contains $\{\infty\}$ and such is of the form $(X \setminus C) \cup \{\infty\}$. In order to show that $\hat{X}$ is compact, only finitely many more open sets are needed to cover C in order to have a finite subcover of $\hat{X}$. Since C is compact by definition, such a finite subfamily exists. Thus the Hausdorff property of X ensures that the one-point compactification $(\hat{X}, \mathcal{T}_1)$ is a compact space.

Moreover, the compactification $(\hat{X}, \mathcal{T}_1)$ remains Hausdorff due to the locally compact condition of $(X, \mathcal{T})$. First, since X is Hausdorff, all $x, y \in \hat{X}$ possess disjoint open neighbourhoods if $x, y \in X$. Thus, $\hat{X}$ being Hausdorff only requires that any point $x \in X$ can be separated from ∞. Since X is locally compact, it follows from definition A.16 that there exists on open neighbourhood K around x such that the closure $\overline{K}$ in X is compact. Then $(X \setminus \overline{K}) \cup \{\infty\}$ is an open neighbourhood of ∞ in $\hat{X}$ which is disjoint from K.

The Alexandroff compactification can be applied to Euclidean n-space $\mathbb{R}^n$ and to non-compact subsets thereof (cf. discussion of definition A.16). For example, the real line $(-\infty, +\infty) = \mathbb{R}^1$ with the usual topology can be compactified by bending its opposite ends stretching to negative and positive infinity towards each other and adding a point at infinity (can be intuitively thought of as unsigned infinity) which results in a compact circle.

The similar procedure can be applied to any open subset (a, b), with $a, b \in \mathbb{R}^1$ and $a < b$. In either case, the resulting compactified space is homeomorphic to the unit 1-sphere $\mathbb{S}^1$ which is compact as a closed and bounded subset of $\mathbb{R}^2$. This can be generalized as follows.

Proposition A.33. The one-point compactification of Euclidean n-space $\mathbb{R}^n$ is homeomorphic to the n-sphere $\mathbb{S}^n$.

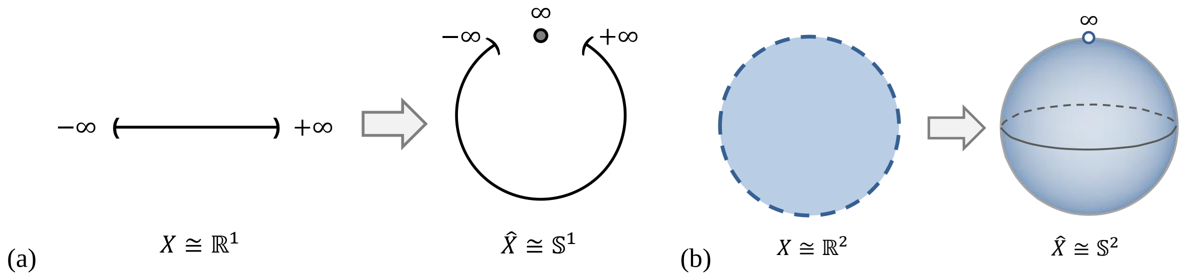


Figure 265: One-point compactification of $\mathbb{R}^1$ *(left) and* $\mathbb{R}^2$ *(right).*

Proposition A.34. Euclidean n-space $\mathbb{R}^n$ is path-connected.

The open and closed n-balls are path-connected as well. The unit n-sphere $\mathbb{S}^n$ is path-connected for $n > 1$. The 0-sphere $\mathbb{S}^0$ is not connected and thus not path-connected, whereas $\mathbb{S}^{-1} = \emptyset$ is connected (Lee 2011). More generally, a subspace $A \subset \mathbb{R}^n$ is path-connected if it is *convex* in the sense that every point on the line segment connecting two points $x, y \in A$ also lies in A.

A basic foundation of Euclidean n-space is the *invariance of dimension* which was first proved by L. E. J. Brouwer in 1911 based on his theorem on the invariance of domain.

Theorem A.35 (Invariance of dimension). If non-empty open sets $U \subset \mathbb{R}^m$ and $V \subset \mathbb{R}^n$ are homeomorphic, then $m = n$.

A consequence of this theorem is that $\mathbb{R}^m \ncong \mathbb{R}^n$ if $m \neq n$ (Hatcher 2008).

A.2 Topological Manifolds

The basic structure and nice properties of Euclidean n-space $\mathbb{R}^n$ are intuitively understood and restrict the simplicity and elasticity of the set-theoretic notion of topological space as given by definition A.1. The most important objects in the field of *algebraic topology* are therefore spaces which locally look like Euclidean space. Such spaces are called *topological manifolds* and are elaborated in the following.

Lemma A.36. A topological space $(X, \mathcal{T})$ is *locally Euclidean of dimension* n iff there exists a non-negative integer n such that each point $x \in X$ has an open neighbourhood homeomorphic to the open unit n-ball in $\mathbb{R}^n$.

The property of being locally Euclidean requires the topological space to resemble Euclidean n-space at a small enough scale. However, the global structure of the space may be more complex and, thus, need not be homeomorphic to $\mathbb{R}^n$.

Definition A.37 (Topological manifold). An n*-dimensional topological manifold* M is a second-countable Hausdorff space that is locally Euclidean of dimension n.

In literature, the term *topological manifold* is used to differentiate manifolds satisfying the above definition from manifolds having an additional structure such as *differentiable*, *smooth*, or *complex manifolds*. The latter types of manifolds are outside the scope of this thesis, and thus the terms n*-manifold* or just *manifold* (in case the dimension is understood or irrelevant) are equivalently used throughout this thesis to refer to a topological manifold.

Obviously, the prototypical example of an n-manifold is $\mathbb{R}^n$ itself with the usual topology. In fact, any open subset of $\mathbb{R}^n$ with the induced usual topology also satisfies definition A.37 because the properties of being Hausdorff and second-countable are hereditary. This can even be generalized to open subsets of an n-manifold (cf. Lee 2011).

Proposition A.38. Every open subset of an n-manifold is an n-manifold.

A manifold is a topological space in its own right, irrespective of whether or not it is a subset of another space. Examples for topological n-manifolds according to definition A.37 are the open line and the circle (closed loop) which are both 1-manifolds. The plane and any open disk are 2-manifolds. The interior of a solid ball is a 3-manifold whereas its boundary, the 2-sphere, is a 2-manifold as is the open cylinder surface.

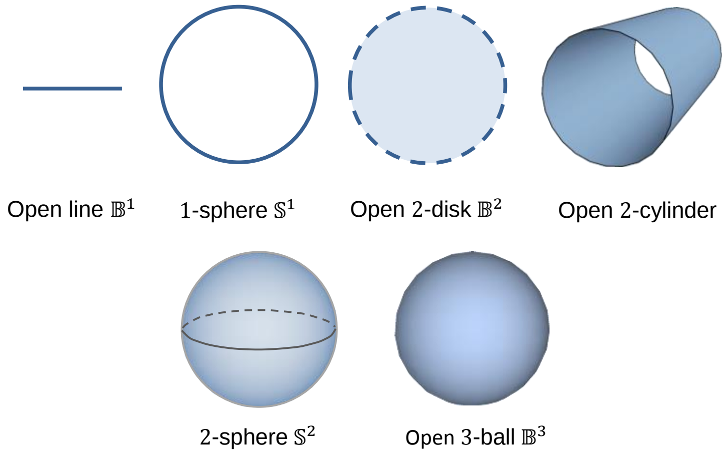


Figure 266: Examples of topological manifolds of different dimensions.

The definition A.37 given above allows a topological manifold to be either a connected or a disconnected topological space. In the connected case, the dimension n of the manifold is constant and an intrinsic property of the manifold. For a disconnected manifold the components are required to be of the same dimension in the context of this thesis, although more general definitions can be found in literature (cf. Morris 2012 with further references). This allows imposing the invariance of dimension theorem (cf. theorem A.35) on topological manifolds which makes their dimension a topological property.

Proposition A.39. If $m \neq n$, a non-empty topological space cannot be an m-manifold and an n-manifold (Lee 2011).

An example for a disconnected 2-manifold in $\mathbb{R}^3$ is the 2-sphere with a contained circle removed.

Counterexamples for non-manifold spaces are the union of the x-axis and the y-axis in $\mathbb{R}^2$, the conical surface in $\mathbb{R}^3$, and the space described by two cuboids sharing a common edge in $\mathbb{R}^3$, all with their Euclidean topology.

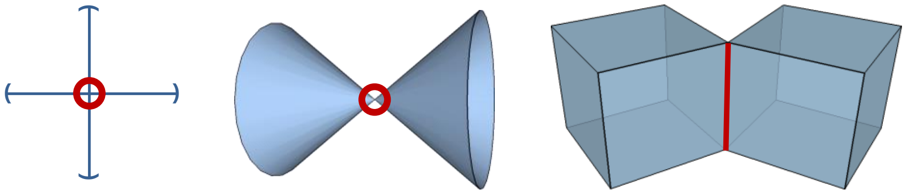


Figure 267: Examples of non-manifold spaces which are not locally Euclidean at every point (depicted in red).

It follows from definition A.37 that closed n-balls in $\mathbb{R}^n$ with the usual topology do not satisfy the requirements of a topological n-manifold as given above, because they are not locally Euclidean at their boundary points. However, closed n-balls as well as subsets of $\mathbb{R}^n$ homeomorphic to a closed n-ball are important objects in geometric and solid modelling. Thus, a more generalized definition of a topological manifold is given in the following.

Definition A.40 (Topological manifold with boundary). An *n-dimensional topological manifold M with boundary* is a second-countable Hausdorff space in which each point has an open neighbourhood homeomorphic to either an open subset of $\mathbb{R}^n$ or to an open subset of the half-space $\mathbb{R}^n_+ = \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_n \geq 0\}$, for some non-negative integer n.

This definition of an n-manifold with boundary is broader than definition A.37 because the boundary can be the empty set $\emptyset$, and thus a topological manifold in the sense of definition A.37 can be described as *n-manifold with* *empty boundary* or *without boundary*. This thesis employs the generalized meaning of topological manifolds as given in definition A.40.

The simplest example for an n-manifold with boundary is $\mathbb{R}_{+}^{n}$ itself with the usual topology. Further examples are the closed disk (circle plus interior) in $\mathbb{R}^2$ and a solid 3-ball (2-sphere plus interior) in $\mathbb{R}^3$.

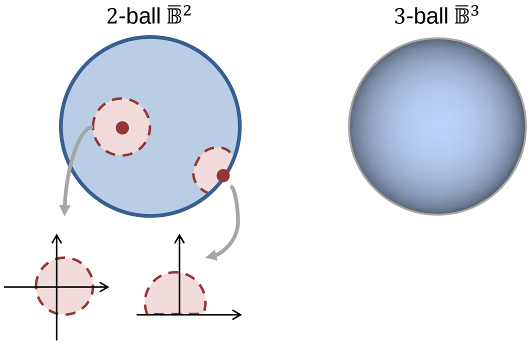

2-ball $\overline{\mathbb{B}}^2$
3-ball $\overline{\mathbb{B}}^3$

Figure 268: Examples of topological manifolds with boundary in two and three dimensions. On the left, the 2*-dimensional neighbourhood of an interior point and a boundary point are depicted in red.*

Definition A.41 (Interior and boundary of an n-manifold with boundary). An n-dimensional topological manifold M with boundary decomposes into two disjoint subsets called the *interior* and the *boundary* of M which are defined as follows:

(i) The interior points are exactly those points in M which have an open neighbourhood homeomorphic to an open subset of $\mathbb{R}^n$. The set of interior points is said to be the *interior* of M and is denoted *Int(*M*)*.

(ii) The *boundary* of M is the set of points contained in the complement of *Int(*M*)* in M and is denoted ∂M.

Int(M*)* is an open subset of M, and thus $\partial M = M \setminus Int(M)$ is closed in M. Since *Int(*M*)* and ∂M are disjoint subsets whose union is M, the following theorem for the invariance of the boundary can be deduced (Lee 2011).

Theorem A.42 (Invariance of the boundary). Let M be an n-manifold with boundary, then a point $x \in M$ is either an interior point or a boundary point, but neither both nor none.

From this and the invariance of dimension (cf. proposition A.39) it follows that a homeomorphism between two manifolds with boundary carries boundary points to boundary points.

Both the set of boundary points and interior points describe manifold spaces themselves. For example, the closed unit n-ball $\overline{\mathbb{B}}^n$ is an n-manifold with boundary for which $\partial\overline{\mathbb{B}}^n = \mathbb{S}^{n-1}$ and $\mathbb{S}^{n-1}$ is an $(n-1)$-manifold without boundary. The interior $Int(\overline{\mathbb{B}}^n)$ is the open unit n-ball $\mathbb{B}^n$, which itself is an n-manifold without boundary. This correlation generally holds for any n-manifold.

Proposition A.43. Let M be an n-manifold with boundary, then *Int(*M*)* is an n-dimensional manifold without boundary, and ∂M is an $(n-1)$-dimensional manifold without boundary.

If M is a 0-dimensional manifold with boundary, then it is a discrete space which is made up of one or more isolated points. In this case, each point in M has an open neighbourhood homeomorphic to $\mathbb{R}^0$, and thus $Int(M) = M$. Correspondingly, $\partial M = \emptyset$, as it should be for a (-1)-dimensional manifold.

The meaning of *interior* and *boundary* of a topological manifold is distinctively different from the meaning of both terms in general point-set topology as introduced in definition A.5 which refers to subsets of topological spaces. In order to distinguish both meanings, in literature the terms *topological interior* and *topological boundary* are sometimes used in reference to definition A.5, whereas *manifold interior* and *manifold boundary* represent the meaning as given in definition A.41.

The topological boundary of a manifold must not necessarily coincide with its manifold boundary. For example, the closed disk $\overline{D}^2$ is a 2-manifold with boundary whose manifold boundary $\delta\overline{D}^2 = S^1$ is the 1-dimensional circle.

If $\bar{D}^2$ is regarded as topological space on its own, then its topological boundary is empty. In contrast, the topological boundary is also the circle if $\bar{D}^2$ is considered as subset of $\mathbb{R}^2$. As subset of $\mathbb{R}^3$, the topological boundary is even the entire disk itself (cf. Lee 2011).

Definition A.44 (Open and closed manifold). A manifold without boundary is called *closed* if it is compact, and *open* if it is non-compact.

Again, both terms must not be confused with the notion of open and closed subsets of a topological space as defined in general point-set topology and introduced in definition A.1 and definition A.2.

The following example easily follows from the machinery developed so far: The closed unit n-ball $\bar{\mathbb{B}}^n$ as subspace of $\mathbb{R}^n$ with the induced usual topology is compact according to the Heine-Borel theorem (cf. theorem A.30). It is also a compact n-manifold in $\mathbb{R}^n$ but not a *closed manifold* because it has the unit $(n-1)$-sphere $\mathbb{S}^{n-1}$ as (manifold) boundary. For $n > 1$, $\mathbb{S}^{n-1}$ itself is a compact and boundaryless manifold in $\mathbb{R}^n$, and thus a prototypical example of a *closed manifold*. The open unit n-ball $\mathbb{B}^n$ as (manifold) interior of $\bar{\mathbb{B}}^n$ is an *open manifold* because it is boundaryless but non-compact in $\mathbb{R}^n$.

Proposition A.45. Every topological manifold with or without boundary is locally compact (Lee 2011).

From this proposition and definition A.40 it follows that a topological manifold is a locally compact Hausdorff space and hence the one-point compactification can be applied to non-compact topological manifolds.

Some more general definitions of topological manifolds in literature do not require the manifold to be Hausdorff or second-countable as stated in definition A.40. However, such manifold spaces are often considered pathological in the sense that they not conform to an intuitive spatial understanding, and thus are ruled out for the course of this thesis. Since Euclidean n-space is both Hausdorff and second-countable, an m-manifold not satisfying one or both requirements would not be realizable as locally Euclidean subset in $\mathbb{R}^n$. In fact, according to the Whitney embedding theorem both properties ensure that any m-manifold embeds in some finite-dimensional Euclidean n-space with $m \leq n$ (Skopenkov 2008), and it can be shown that every topological m-manifold following the definition A.40 is homeomorphic to some subset of a Euclidean n-space (e.g., Lee 2011, Munkres 1984).

This thesis is concerned only with m-dimensional manifolds M which are themselves subsets of Euclidean n-space $\mathbb{R}^n$ for some finite n and m, with $0 \leq m \leq n$, and which have the topology induced by the usual topology on $\mathbb{R}^n$. Although the manifold definition given in this chapter aims at describing a manifold as a topological space on its own independent of its surrounding space, the focus of this thesis is restricted to m-manifolds living in $\mathbb{R}^n$. If $M \subset \mathbb{R}^n$, then $\mathbb{R}^n$ is said to be the *ambient space* of M. Since the properties of being Hausdorff and second-countable are hereditary as shown above, both properties can be taken for granted for subsets of Euclidean space, and thus for any topological m-manifold of interest in this thesis.

A.3 Cell Complexes and CW Complexes

Many topological spaces can be constructed by attaching or “gluing” together subsets, each with a simple topology, “nicely” along their boundaries. Such subsets are commonly called *cells* and topological spaces which result from an aggregation of cells and hence have a cellular topology are said to be *cell complexes.* The decomposition of the topological space into cells results in a combinatorial structure which makes such spaces especially interesting because important invariant properties classifying the topological space itself can be deduced from the simple information about how the cells are attached to each other. Thus, the combinatorial structure also helps to accelerate the algorithmic computation of topological properties (Lee 2011). Cell complexes are a rich field of study and a main tool in *algebraic topology* (Hatcher 2008), and many authors have proposed cell complexes for the spatial modelling of shapes and solid objects.

The primary building blocks of cellular spaces are *cells* of dimension n. A *cell* is a topological space in its own right but it is also a subspace of the larger topological space which is built up inductively from a collection of cells.

Definition A.46 (Open and closed n-cell). An *open n-cell* is any topological space homeomorphic to the open unit n-ball $\mathbb{B}^n$. A *closed n-cell* is any space homeomorphic to the closed unit n-ball $\bar{\mathbb{B}}^n$. The non-negative integer n is called the dimension of the n-cell.

More concisely, an open n-cell is the image of the open unit n-ball $\mathbb{B}^n$ under a homeomorphism of its closure into $\mathbb{R}^n$. This emphasises the fact that an n-cell itself not necessarily is a subspace of Euclidean space in which the open unit n-ball $\mathbb{B}^n$ has been defined (cf. definition A.28). In literature, sometimes the notion of an open (closed) *topological n-ball* as subset of some topological space X (whose topology need not be induced by a metric) is introduced which is defined to be homeomorphic to an open (closed) Euclidean n-ball. In this general sense, an open (closed) n-cell is an open (closed) *topological n-ball*.

Similar to the terminology used for topological manifolds, the terms *n-dimensional cell* or just *cell* (in case the dimension is understood or irrelevant) are equivalently used for n-cells throughout this thesis.

From the discussion of topological n-manifolds, the notion of the *interior* and *boundary* of a closed n-cell as topological space on its own can be deduced.

Definition A.47 (Interior and boundary of a closed n-cell). Let e be a closed n-cell. Since per definition $e \cong \overline{\mathbb{B}}^n$, and $\overline{\mathbb{B}}^n$ is a compact n-manifold with boundary, so is e. From definition A.41 and proposition A.43 it follows that $Int(\overline{\mathbb{B}}^n) = \mathbb{B}^n$ and $\partial\overline{\mathbb{B}}^n = \mathbb{S}^{n-1}$. Under some homeomorphism $f: \overline{\mathbb{B}}^n \to e$, the image of $\mathbb{B}^n$ is the interior of e denoted $Int(e)$, and equivalently the image of $\mathbb{S}^{n-1}$ is the boundary of e denoted ∂e.

It easily follows that $\partial e \cong \mathbb{S}^{n-1}$, and that $Int(e) \cong \mathbb{B}^n$ is an open n-cell. Due to the invariance of boundary (cf. theorem A.42), the boundary and interior are well defined and disjoint subsets of closed n-cells (Lee 2011). Although n-cells are topological n-manifolds with or without boundary, the converse obviously need not be true. The definition of an n-cell is much stricter in such the entire topological space described by the cell has to be homeomorphic to an open or closed Euclidean n-ball whereas topological manifolds are only required to have open neighbourhoods for each point that are homeomorphic to the open or half-closed n-ball.

In literature, cell complexes are often defined by describing the procedure of how to inductively form a topological space X from cells of increasing dimensions (e.g., Hatcher 2008). Starting with the empty set $\emptyset$, a discrete space X_0 is built from attaching disjoint 0-cells to the empty set which are basically singleton sets. X_1 results from unioning 1-cells whose boundaries are identified to lie on these points. Attaching 2-cells forms a new space X_2 by identifying the boundary of each 2-cell into the collection of cells of lower dimension, and so on. A more technical definition of a *cell complex* is given as follows (Lee 2011).

Definition A.48 (Cell complex). A *cell complex* is a pair $(X, \mathcal{E})$ where X is a topological space together with a partition $\mathcal{E}$ of X into subspaces that are open cells of various dimensions and whose disjoint union is X (called *cell decomposition* of X), such that the following conditions are satisfied:

(i) X is Hausdorff,

(ii) for each open n-cell $e \in \mathcal{E}$ and $n \geq 0$, there exists a continuous map $\phi_e: \overline{\mathbb{B}}^n \to X$ (called the *characteristic map* for e) that restricts to a homeomorphism from $Int(\overline{\mathbb{B}}^n)$ onto e, and maps $\partial\overline{\mathbb{B}}^n$ into the union of all cells of $\mathcal{E}$ of dimension strictly less than n.

The space X is said to be the *underlying space* of the cell complex. The open cells contained in $\mathcal{E}$ are often simply called the *"cells of X"* (Lee 2011). Although X is partitioned by open cells, this does not imply that each open cell $e \in \mathcal{E}$ is also an open subset of X in the sense of general point-set topology.

The cell complex $(X, \mathcal{E})$ is said to be *finite-dimensional* if there is a non-negative integer n such that all cells in X are at most of dimension n. In case of a cell complex being finite-dimensional, the largest n is called the *dimension* of the cell complex.

Definition A.49 (n-skeleton X_n). The *n-skeleton* X_n of a cell complex $(X, \mathcal{E})$ is the subspace $X_n \subseteq X$ consisting of the union of all cells in $\mathcal{E}$ of dimension less or equal to n.

The Hausdorff property is imposed on cell complexes in order to rule out pathological cases which are not conformant with the inductive construction of cell complexes. Due to the main theorem on compactness (cf. theorem A.15), ϕ_e takes the compact set $\overline{\mathbb{B}}^n$ to a compact set in X. Since X is Hausdorff, the image $\phi_e(\overline{\mathbb{B}}^n)$ is closed in X (cf. generalization of proposition A.25). Thus, the image of a characteristic map for e is equal to the closure of e in X, such that $\phi_e(\overline{\mathbb{B}}^n) = \bar{e}$ (cf. Lee 2011).

The restriction of the characteristic map to its boundary $\partial\bar{\mathbb{B}}^n$ $(= \mathbb{S}^{n-1})$ is called the *attaching map* and is denoted $\varphi_e = \phi_e|_{\mathbb{S}^{n-1}} : \mathbb{S}^{n-1} \to X_{n-1}$. Intuitively, the attaching map defines how an open n-cell is glued to the open cells contained in the $(n-1)$-skeleton. From condition (ii) of definition A.48 it follows that $\phi_e(Int(\bar{\mathbb{B}}^n)) = e$ and $\varphi_e(\mathbb{S}^{n-1}) \cap e = \emptyset$. Thus, the image of φ_e for e can be given as $\varphi_e(\mathbb{S}^{n-1}) = \bar{e} \setminus e$. The image $\bar{e} \setminus e$ of the attaching map is said to be the *cell boundary* of e and it is a closed subset which lies in X^{n-1} (Jänich 2012).

It is important to note that the terminology, although well established in literature, can be confusing. The *cell boundary* of an open cell $e \in \mathcal{E}$ as defined above need not be equal to the *topological boundary* of e as subset of the topological space $(X, \mathcal{E})$. This follows from the fact that e is not necessarily an open subset in X. Moreover, ϕ_e is not required to be injective on the cell boundary per definition. This means that the cell boundary is not required to be a homeomorphic image of $\mathbb{S}^{n-1}$ but just a continuous image, and thus the closure $\bar{e}$ need not be a closed cell (Lee 2011). Hence the notion of the *cell boundary* of an open cell is distinct from the *boundary of a closed cell* as introduced in definition A.47.

For example, assume the attaching map φ_e for an open 1-cell e maps $\partial\bar{\mathbb{B}}^1$ $(= \mathbb{S}^0)$ onto a single 0-cell which intuitively results in a closed circle. Then obviously the cell boundary of e is not homeomorphic to $\mathbb{S}^0$, and the closure $\bar{e}$ is not a closed cell. This also immediately follows from the definition of closed cells (cf. definition A.46) because $\bar{e} \cong \mathbb{S}^1 \ncong \bar{\mathbb{B}}^1$.

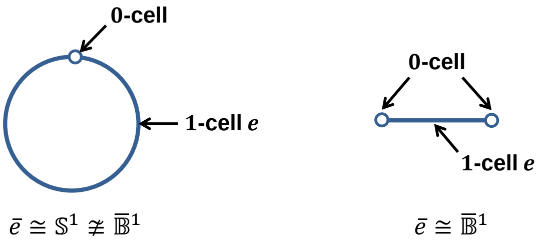

0-cell
1-cell e
$\bar{e} \cong \mathbb{S}^1 \ncong \bar{\mathbb{B}}^1$
0-cell
1-cell e
$\bar{e} \cong \bar{\mathbb{B}}^1$

Figure 269: Two attaching maps for an open 1*-cell* e*. On the left,* φ_e *carries* $\partial\bar{\mathbb{B}}^1$ *onto a single* 0*-cell and thus* $\bar{e} \cong \mathbb{S}^1$*. On the right,* φ_e *carries* $\partial\bar{\mathbb{B}}^1$ *onto a homeomorphic image of* $\mathbb{S}^0$ *and thus the closure* $\bar{e}$ *is a closed cell.*

Definition A.50 (Finite and locally finite cell complex). A cell complex $(X, \mathcal{E})$ is said to be *finite* if its cell decomposition $\mathcal{E}$ only contains finitely many open cells. It is said to be *locally finite* if each point $x \in X$ is contained in a finite number of cell closures.

In order to ensure that infinite cell complexes are well-behaved, two more restrictions called *closure finiteness* and *weak topology* are introduced. A cell complex satisfying both properties is said to be a *CW complex*. CW complexes were first introduced by J. H. C. Whitehead in 1949 (Whitehead 1949) and can be defined as follows based on the notion of cell complexes.

Definition A.51 (CW complex). A CW complex is a cell complex $(X, \mathcal{E})$ which additionally satisfies the following properties:

(i) (Closure finiteness). The closure $\bar{e}$ of each cell $e \in \mathcal{E}$ is contained in a union of *finitely* many cells.

(ii) (Weak topology). A subset $A \subset X$ is closed iff every intersection $A \cap \bar{e}$ is closed in X for every $e \in \mathcal{E}$.

Condition (i) can be equivalently rephrased by stating that the cell boundary $\bar{e} \setminus e$ of every open cell $e \in \mathcal{E}$ is contained in the union of finitely many cells of lower dimension. For every locally finite and hence finite cell complex, both properties are automatically satisfied (Jänich 2012).

For an n-dimensional CW complex the following proposition on the correlation between open cells and open subsets holds.

Proposition A.52. Let X be an n-dimensional CW complex. Then every n-cell of X is an open subset of X.

In any CW complex, subspaces can be easily derived based on the cellular structure of the entire space. Such subspaces are called *subcomplexes* and are defined as follows.

Definition A.53 (Subcomplex). A *subcomplex* of a CW complex X is a subspace $A \subseteq X$ which is a union of cells of X, such that the closure of each cell in A is contained in A.

For each cell in A, the image of its attaching map is also contained in the subcomplex A. It follows that a subcomplex is a union of closed sets of X and hence is closed in X itself. Together with the induced subspace topology and the cell decomposition inherited from X, the subcomplex A is itself a CW complex (Lee 2011, Hatcher 2008). Both the union and the intersection of any collection of subcomplexes result in subcomplexes.

Due to the closure finiteness property of CW complexes, the closure $\bar{e}$ of each open cell is contained in a finite subcomplex. The n-skeleta of a CW complex as given by definition A.49 are prototypical examples of subcomplexes. It follows from proposition A.52, that the open n-cells contained in the n-skeleton are open subsets in X_n.

A map between two CW complexes X and Y identifying cells in either n-skeleton is said to be *cellular* and defined as follows.

Definition A.54 (Cellular map). Let X and Y be two CW complexes. Then the continuous map $f: X \to Y$ is said to be *cellular* iff it takes the n-skeleton of X to the n-skeleton of Y, i.e. $f(X_n) \subseteq Y_n$, for all non-negative integers n.

Proposition A.55. Let X be a CW complex. Then X is compact iff it is finite. A subset of $A \subseteq X$ is compact iff it is closed and contained in a finite subcomplex.

The proposition implies that compactness is preserved under attaching finitely many cells. Every finite subcomplex of a CW complex is obviously compact because it is built from the union of finitely many compact sets of the form $\bar{e}$ (Lee 2011).

Instead of studying whether a given topological space X together with some cell decomposition $\mathcal{E}$ is a CW complex, the following theorem provides a formalisation for an inductive approach to build up CW complexes based on the machinery developed so far (cf. Lee 2011).

Theorem A.56 (Construction theorem for CW complexes). Let $X_0 \subseteq X_1 \subseteq \cdots \subseteq X_{n-1} \subseteq X_n \subseteq \cdots$ be a sequence of topological spaces. Then $X = \bigcup_n X_n$ is a CW complex whose n-skeleton is X_n for each n, iff the following conditions hold:

(i) X_0 is a non-empty discrete space (thus containing at least one 0-cell).
(ii) For each $n \geq 1$, X_n is obtained from X_{n-1} by attaching a (possibly empty) collection of n-cells.

The way cells are glued together in a CW complex can be further restricted by requiring the characteristic map ϕ_e of each open n-cell $e \in \mathcal{E}$ to be a homeomorphism onto its image $\bar{e}$ (and thus an embedding). Since the characteristic map homeomorphically maps the interior of $\bar{\mathbb{B}}^n$ onto e per definition (cf. definition A.48), this especially requires the image of the attaching map φ_e to be homeomorphic to the $(n-1)$-sphere, and thus $\varphi_e(\mathbb{S}^{n-1}) = (\bar{e} \setminus e) \cong \mathbb{S}^{n-1}$. In this case, the closure $\bar{e}$ is homeomorphic to the closed n-ball $\bar{\mathbb{B}}^n$ which makes it a closed cell. An open cell which admits such an attaching map is called a *regular cell*.

Definition A.57 (Regular CW complex). A CW complex is said to be a *regular CW complex* if each of its open cells e is regular, and the closure of each open cell $\bar{e}$ is a finite subcomplex.

In a regular CW complex $(X, \mathcal{E})$, the image of the characteristic map ϕ_e of each open cell $e \in \mathcal{E}$ is equal to a closed cell and each open cell meets every 0-cell in $\mathcal{E}$ at most once. A regular CW complex is said to have the *intersection property* iff the non-empty intersection of any two closed cells in X is also a closed cell. Such CW complexes are also called *proper*.

The example below shows different CW decompositions for the 2-sphere.

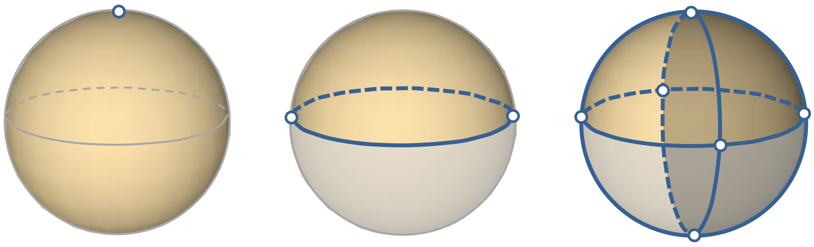


Figure 270: Different CW decompositions of the 2*-sphere* $\mathbb{S}^2$ *(from left to right: minimal CW complex with one* 0*-cell and one* 2*-cell, regular CW complex with two cells per dimension, and proper CW complex containing six* 0*-cells, twelve* 1*-cells, and eight* 2*-cells).*

Proposition A.58. Let X be a CW complex. Then the following are equivalent.

(i) The 1-skeleton is connected.

(ii) X is connected.

(iii) X is path-connected.

As stated at the beginning of this chapter, the closed and open n-cells can be characterized as topological n-manifolds with or without boundary. However, the topological space derived by a CW decomposition need not be manifold itself. This fact is independent from whether the CW complex itself is regular or even proper. The following proposition restricts a CW complex to have a manifold as underlying space.

Proposition A.59. A finite, locally Euclidean CW complex is a topological manifold.

Every CW complex is Hausdorff by definition. If a CW complex is finite, then it only has countably many open cells and thus is separable. Moreover, due to the finiteness property there are only finitely many open sets, and this implies that the CW complex is second-countable. Given that the CW complex is locally Euclidean, it satisfies the requirements of a topological manifold as provided by definition A.37. It follows from the theorem on invariance of dimension (cf. theorem A.35), that the dimension of the underlying topological manifold is equal to the dimension of the CW complex.

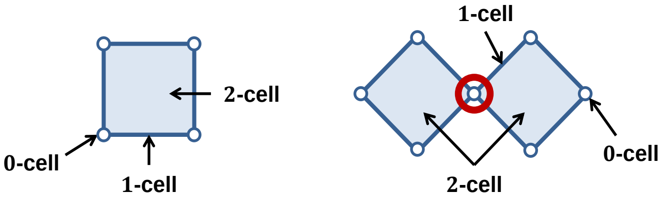


Figure 271: Manifold CW complex homeomorphic to $\overline{\mathbb{B}}^2$ *(left), and non-manifold CW complex (right).*

Proposition A.59 provides a check whether a CW complex is also a topological manifold. However, the converse question whether a topological n-manifold (with or without boundary) admits a cell decomposition is more difficult. The next chapter introduces the notion of *simplicial complexes* which have been used in literature to answer this question.

A.4 Simplicial Complexes

Simplicial complexes are a specialized class of complexes and were first introduced by Henri Poincaré in 1899 (Poincaré 1899). A simplicial complex is a structure on a topological space which is constructed by gluing n-dimensional *simplices* together along their boundaries. This follows the general idea of cell complexes to inductively form a topological space from a collection of disjoint building blocks. However, simplicial complexes postulate more restrictive requirements than general cell complexes on the building blocks and the way the building blocks are attached to each other. This ensures that all topological information is encoded in terms of purely combinatorial data which is to be seen the main advantage of simplicial complexes (Lee 2011). From this viewpoint, simplicial complexes are special types of CW complexes although they preceded the work on CW complexes by several decades.

The primary building block of a simplicial complex is the n-dimensional *simplex* or *n-simplex*. Intuitively, a 0-simplex corresponds to a single point, a 1-simplex to a straight line segment, a 2-simplex to a filled triangle, and a 3-simplex to a solid tetrahedron (Mäntylä 1988). An n-simplex is the generalized analog of these examples in any dimension.

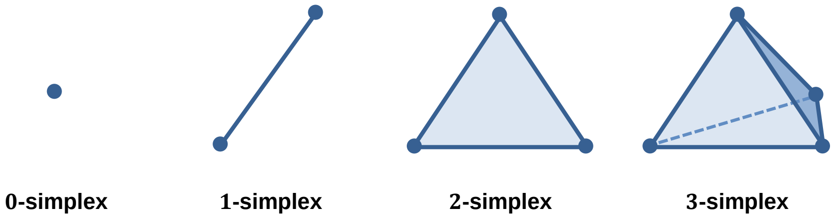


Figure 272: 0*- to* 3*-dimensional simplices.*

Definition A.60 (n-simplex). For any non-negative integer n, let $\{v_0, \dots, v_n\}$ be a set of $n+1$ points in Euclidean space $\mathbb{R}^m$, $m \geq n$, whose difference vectors $v_1 - v_0, \dots, v_n - v_0$ are linearly independent. Then the smallest convex set in $\mathbb{R}^m$ containing the points $v_0, \dots, v_n$ is called n-dimensional *simplex* or *n-simplex*, and is given by the set $[v_0, \dots, v_n] = \{\lambda_0 v_0 + \dots + \lambda_n v_n \in \mathbb{R}^m \mid \sum_i \lambda_i = 1 \text{ and } \lambda_i \geq 0 \text{ for all } i\}$.

Each of the points v_i is called a *vertex* of the n-simplex. The integer n denotes the dimension of the simplex and is always one less than the number of its vertices. The coefficients λ_i are the *barycentric coordinates* of a point $x = \sum_i \lambda_i v_i$ in $[v_0, \dots, v_n]$ (Hatcher 2008). The empty set $\emptyset$ is defined as simplex of dimension -1 having 0 vertices. As subset of the ambient space $\mathbb{R}^m$, any n-simplex is equipped with the induced usual topology.

An important example for an n-simplex is the *standard n-simplex* Δ^n which is defined as follows.

Definition A.61 (Standard n-simplex Δ^n). The *standard n-simplex* $\Delta^n = \{(\lambda_0, \dots, \lambda_n) \in \mathbb{R}^{n+1} \mid \sum_i \lambda_i = 1 \text{ and } \lambda_i \geq 0 \text{ for all } i\}$ is the convex hull of the positive unit coordinate vectors in $\mathbb{R}^{n+1}$.

Any n-simplex can be realized as the image of Δ^n under a homeomorphism, and, consequently, any two n-simplices are homeomorphic. Similar to the cells in a cell complex, every n-simplex is a topological space in its own right. It immediately follows from the above definition that the standard n-simplex as topological space is homeomorphic to the closed unit n-ball, i.e. $\Delta^n \cong \overline{\mathbb{B}}^n$. Consequently, an n-simplex is a compact n-manifold with boundary and the following proposition on the relation between n-simplices and n-cells holds.

Proposition A.62. Every n-simplex is a closed n-cell (Lee 2011).

It is important to note that the n-simplex as building block of a simplicial complex distinctively differs from the building blocks of general cell decompositions which are open cells per definition A.48. n-simplices are always understood to be closed simplices respectively closed cells (Lee 2011). Moreover, an inherent property of an n-simplex is its facial structure which is defined in the following and which is irrelevant for general cell decompositions.

Definition A.63 (Face, proper face). Each non-empty subset of the $n+1$ vertices of Δ^n spans a k-simplex Δ^k with $0 \leq k \leq n$ which is called a *face* of Δ^n. Any face not equal to Δ^n itself is called a *proper face* of Δ^n.

The 0-dimensional faces of Δ^n are just the set of its vertices. The 1-dimensional faces are called the *edges* of Δ^n, whereas the $(n-1)$-dimensional faces of Δ^n are said to be its *boundary faces*. Based on the notion of faces, the *boundary* and *interior* of Δ^n as well as an *open n-simplex* can be specified.

Definition A.64 (Boundary and interior of Δ^n, open n-simplex). The union of all boundary faces of Δ^n (equivalently, the union of all its proper faces) is called the *boundary* of Δ^n and is denoted $\partial\, \Delta^n$. The *interior* $Int(\Delta^n)$ is the set $\Delta^n - \partial\, \Delta^n$. An *open n-simplex* is exactly the interior of an n-simplex.

The boundary of Δ^n as defined above is equal to its manifold boundary. For example, the boundary of a 1-simplex consists of its two vertices, and thus an open 1-simplex is a line segment minus its vertices. Since the boundary of a 0-simplex is the empty set $\emptyset$, the open 0-simplex is the same as the (closed) 0-simplex.

Analogous to the discussion of cells, the introduced terms and concepts may not be confused. Unless $n = m$, an open n-simplex is not an open subset of its ambient topological space $\mathbb{R}^m$, neither are the boundary and the interior of an n-simplex equal to its topological boundary and interior as subset of $\mathbb{R}^m$ in the sense of general point-set topology (Lee 2011).

Definition A.65 (Orientation of an n-simplex, natural orientation). Two linear orderings of the vertex set $\{v_0, \dots, v_n\}$ of an n-simplex are called *equivalent*, if they differ by an even permutation. The equivalence classes are said to be the *orientations* of the n-simplex (Hatcher 2008).

Obviously, for $n > 0$ exactly two orientations are constituted on every n-simplex by the ordering of its vertices. The *natural orientation* of an n-simplex given by $v_0 < \cdots < v_n$ is denoted $[v_0, \dots, v_n]$, whereas a minus sign is commonly used as prefix to denote the reversed orientation, i.e. $-[v_0, \dots, v_n]$. For example, for a 2-simplex this means that $[v_1, v_2, v_0] = [v_0, v_1, v_2]$ but $[v_0, v_1, v_2] = -[v_0, v_2, v_1]$. So, generally, swapping two vertices as well as applying an on odd number of two-vertex swaps reverses the orientation.

The natural orientation of an n-simplex also induces an orientation on all of its boundary faces.

Definition A.66 (Induced orientation). Given an oriented n-simplex $[v_0, \dots, v_n]$, the *induced orientation* on the i-th boundary face τ_i spanned by the vertex set $\{v_0, \dots, \hat{v}_i, \dots, v_n\}$ (where the hat sign "ˆ" marks the missing vertex v_i) is defined to be $(-1)^i[v_0, \dots, \hat{v}_i, \dots, v_n]$.

When applied to an oriented 2-simplex $[v_0, v_1, v_2]$, the induced orientations on its edges are $[v_0, v_1]$, $[v_1, v_2]$, and $-[v_0, v_2] = [v_2, v_0]$. Also note that an oriented 1-simplex $[v_0, v_1]$ induces the orientation $-[v_0]$ onto its first vertex and $[v_1]$ onto its terminal vertex, and, thus, also a 0-simplex receives two distinct induced orientations. The following figure sketches both examples.

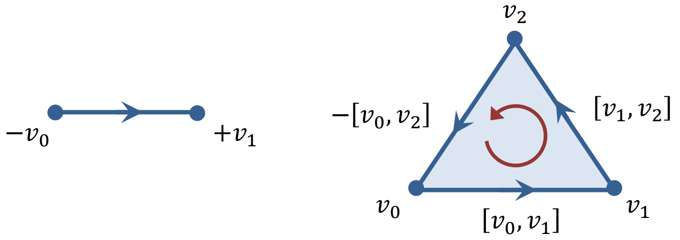


Figure 273: Induced orientations for an oriented 1*-simplex (left) and an oriented* 2*-simplex (right).*

Definition A.67 (Oriented boundary). Let σ^n be an oriented n-simplex. Then the *oriented boundary* $\partial\sigma^n$ is the set of all boundary faces with the induced orientation, given by $\partial\sigma^n = \sum_{i=0}^{n}(-1)^i[v_0, \dots, \hat{v}_i, \dots, v_n]$.

Finally, a simplicial complex is defined as collection of disjoint n-simplices.

Definition A.68 (Euclidean simplicial complex). A *(Euclidean) simplicial complex* is a collection $\mathcal{K}$ of simplices in some Euclidean space $\mathbb{R}^m$, with $m \geq 0$, such that

(i) for any face $\sigma \in \mathcal{K}$ also all faces $\tau \subseteq \sigma$ are in $\mathcal{K}$,
(ii) for any two faces $\sigma, \sigma' \in \mathcal{K}$ the non-empty intersection $\sigma \cap \sigma'$ is a face of σ and σ' (*intersection property*), and
(iii) $\mathcal{K}$ is a locally finite collection.

A simplicial complex containing only finitely many simplices is called *finite simplicial complex* and condition (iii) is redundant in such cases. The *dimension* of $\mathcal{K}$ is the largest dimension of a simplex in $\mathcal{K}$ and obviously is no greater than m. A *subcomplex* $\mathcal{K}' \subseteq \mathcal{K}$ is defined as non-empty subset of $\mathcal{K}$ for which (i) holds and, thus, which is a simplicial complex on its own. The *n-skeleton* of $\mathcal{K}$ is the subcomplex which consists of all simplices of $\mathcal{K}$ of dimension at most n, with $n \leq m$.

To give an n-dimensional simplicial complex $\mathcal{K}$ an orientation requires two n-simplices $\sigma, \sigma' \in \mathcal{K}$ sharing a common boundary face τ to induce opposite orientations on τ, i.e. to be *consistently oriented*.

Definition A.69 (Orientation, orientable). Let $\mathcal{K}$ be an n-dimensional Euclidean simplicial complex. An *orientation* of $\mathcal{K}$ is a consistent choice of orientation on each n-simplex. If $\mathcal{K}$ admits an orientation, it is said to be *orientable*.

Each simplicial complex $\mathcal{K}$ is completely determined up to homeomorphism by its *face poset* (partially ordered set) which is the set of all faces of $\mathcal{K}$ ordered by inclusion. Thus, simplicial complexes can be described by the purely combinatorial data of how n-simplices are inductively attached to $(n-1)$-simplices starting from a discrete set of vertices.

Definition A.70 (Polyhedron of a simplicial complex). Let $\mathcal{K}$ be a Euclidean simplicial complex in $\mathbb{R}^m$. Then the union of all simplices in $\mathcal{K}$ together with the subspace topology induced from $\mathbb{R}^m$ is a topological space called the *polyhedron* of $\mathcal{K}$ and denoted by $|\mathcal{K}|$.

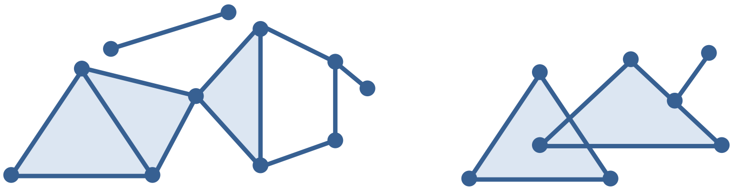


Figure 274: 2-dimensional simplicial complex in $\mathbb{R}^2$ with non-manifold polyhedron (left), and no simplicial complex (right).

It can be deduced from the above definitions that each simplicial complex can be realized as a regular CW decomposition and, thus, each polyhedron $|\mathcal{K}|$ is homeomorphic to a proper CW complex. This correlation is formalized by the following proposition.

Proposition A.71. Let $\mathcal{K}$ be a Euclidean simplicial complex. Then the collection of the interiors of the simplicies in $\mathcal{K}$ is a regular CW decomposition of $|\mathcal{K}|$ (Lee 2011).

There is a subtle but important difference in the terminology of simplicial complexes and CW complexes. The term “CW complex” is used to reference a topological space together with its particular CW decomposition (cf. definition A.48). In contrast, the term “simplicial complex” only refers to the collection of disjoint simplices (which are its cells) whereas the underlying topological space is said to be its “polyhedron”. In literature, the polyhedron is often equivalently called the *carrier* or *underlying space* of a simplicial complex.

Proposition A.71 allows for applying all properties of CW complexes to the polyhedron of a simplicial complex. For example:

Lemma A.72. Let $\mathcal{K}$ be a Euclidean simplicial complex. Then its polyhedron $|\mathcal{K}|$ is

(i) Hausdorff,
(ii) compact iff $\mathcal{K}$ is finite,
(iii) connected iff the polyhedron of its 1-skeleton is path-connected.

The Hausdorff property easily follows from the fact that being Hausdorff is hereditary in metric spaces (cf. proposition A.22). If $\mathcal{K}$ is finite then $|\mathcal{K}|$ is the finite union of n-simplices being closed subsets in $\mathbb{R}^m$, and, thus, $|\mathcal{K}|$ is itself closed. Since in this case $|\mathcal{K}|$ is also bounded in $\mathbb{R}^m$, it follows from the Heine-Borel theorem (cf. theorem A.30) that $|\mathcal{K}|$ is compact.

Definition A.73 (Triangulation, trianguable). A topological space X is said to be *trianguable* if it is homeomorphic to the polyhedron of a simplicial complex $\mathcal{K}$, i.e. $X \cong |\mathcal{K}|$. Such a homeomorphism is called a *triangulation* of X.

For example, both the closed unit ball $\overline{\mathbb{B}}^n$ and the unit sphere $\mathbb{S}^{n-1}$ are trianguable. A triangulation for the non-compact space $\mathbb{R}^1$ can be realized by constructing an infinite, locally finite simplicial complex in $\mathbb{R}^1$ whose 1-simplicies are the closed intervals $[n, n+1], n \in \mathbb{Z}$ and whose vertex set is the set of integers. This approach can be generalized to $\mathbb{R}^n$.

An obvious consequence based on the theory developed is that every regular CW complex (with or without intersection property) is trianguable. Similar to simplicial complexes, a regular CW decomposition provides a partial order which is determined by its closed cells and their inclusion relations. Thus, also regular CW complexes are determined up to homeomorphism by their face poset which is purely combinatorial (Hatcher 2008). The main difference between a regular CW decomposition on a space X and a triangulation of X is that the first typically requires much fewer cells and that the cells have a flexible number of faces and need not to be straight (Agoston 2005).

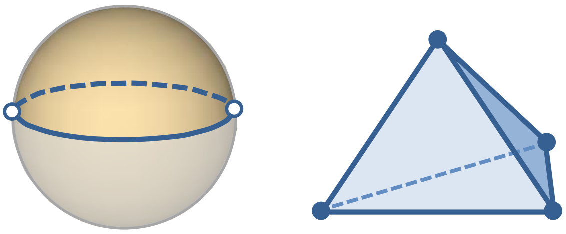


Figure 275: Regular CW decomposition of the 2*-sphere* $\mathbb{S}^2$ *with two cells per dimension (cf. figure 270) (left) and a 2-dimensional simplicial complex consisting of three* 2*-simplices, six* 1*-simplices, and four* 0*-simplices whose underlying polyhedron is homeomorphic to* $\mathbb{S}^2$ *(right).*

However, not every non-regular CW complex admits a triangulation (Agoston 2005). The facial structure of simplicial complexes is not inherent to general CW decompositions. In fact, CW complexes are a strong generalization of polyhedra because of removing the requirement that all attaching maps are embeddings.

In literature, different levels of generality ranging between simplicial complexes and CW complexes are discussed. The reason is that a Euclidean simplicial complex as given in definition A.68 has some strict restrictions. First, it is required to live in Euclidean space $\mathbb{R}^m$ and, consequently, its polyhedron is a subspace of $\mathbb{R}^m$. Second, each n-simplex is required to have $n+1$ distinct vertices and, third, each set of $n+1$ vertices may only define at most one n-simplex (Edelsbrunner & Harer 2010).

A natural generalization of simplicial complexes admits more general shapes for cells such as cubes or convex polytops from which *cube complexes* or *polytopal complexes* (also called *polyhedral* complexes) can be obtained (e.g., Poincaré 1895). By allowing structures built from homeomorphic images of Δ^n, the requirement of an ambient Euclidean space for the underlying topological space is removed and systems of loops of simplices become possible. Such structures were introduced by Eilenberg and Ziller (Eilenberg & Zilber 1950) under the name *semisimplicial complex* but are nowadays discussed as Δ*-complex* (Hatcher 2008) which are an important tool in modern simplicial homology theory in algebraic topology.

This thesis is concerned only with triangulable (but not necessarily regular) CW complexes and their polyhedral triangulations. A more detailed discussion of the sketched types of generalized simplicial complexes is therefore beyond the scope of this thesis, and the reader is directed to the provided literature.

A.5 Topological Classification of Low-Dimensional Manifolds

The developed theory of CW complexes and simplicial complexes is used in this section to present a classification of topological manifolds in up to three dimensions. Classification aims at providing a list of standard n-dimensional manifolds together with a theorem stating that every n-manifold is homeomorphic to exactly one standard n-manifold on that list. Technically speaking, classification lists the topological equivalence classes on the set of all n-manifolds induced by homeomorphism.

In 1908, Ernst Steinitz and Heinrich Tietze originally formulated what became known as the *Hauptvermutung* (German for *main conjecture*) in combinatorial and nowadays algebraic topology. The Hauptvermutung states that if two simplicial complexes have homeomorphic polyhedra then they are *combinatorially equivalent*, i.e. there is a single triangulation that is a subdivision of both of them. The combinatorial topology of a simplicial complex $\mathcal{K}$ is thus be determined by the topology of the polyhedron $|\mathcal{K}|$. One hope associated with this conjecture was thus to apply the combinatorial model of simplicial complexes to the domain of topological manifolds which would allow reducing topological questions about manifolds to both purely combinatorial and computationally efficient ones about simplicial complexes.

Obviously, this requires topological manifolds to be trianguable spaces. It can be deduced from the previous sections that manifolds generally are a stronger form of polyhedra and their generalized representation as regular CW complexes. However, this does not imply that every manifold in any dimension admits a triangulation or, equivalently, can be realized as regular CW complex. Although simplices as well as cells in CW decompositions are defined to be topological manifolds in their own right, the space constructed from gluing a collection of cells together need not be a manifold space itself (cf. proposition A.59).

Triangulation is to be seen a necessary but not sufficient condition for the Hauptvermutung. In 1961, John Milnor (Milnor 1961) presented high-dimensional pairs of compact non-manifold polyhedra with combinatorially inequivalent triangulations which disproved the Hauptvermutung in general. It was further obstructed for the manifold version by examples of 4-manifolds that are not trianguable (Jänich 2012). In high-dimensional topology with dimension $n > 4$ the question whether there are non-trianguable manifolds is still open in literature and a topic of current research. However, the Hauptvermutung holds for topological manifolds in the low dimensions $n \leq 3$. It was proved in the 1920s for all manifolds of dimension $n \leq 2$ by Christos Papakyriakopoulos and Tibor Radó (cf. Radó 1925, Ranicki 1996) and by Edwin Moise (Moise 1977) for 3-dimensional manifolds in the early 1950s. The proofs are highly technical and beyond the scope of this thesis. A complete discussion is presented in (Ranicki 1996).

An important result of the work on the Hauptvermutung is the following triangulation theorem for topological manifolds of dimension strictly less than 4.

Theorem A.74 (Triangulation theorem for manifolds). Every compact topological manifold of dimension $n \leq 3$ is trianguable (Ranicki 1996).

The theorem implies that every compact topological manifold of dimension $n \leq 3$ admits a regular CW decomposition. The Hauptvermutung has been rephrased for low-dimensional topological manifolds in the following weaker manner.

Proposition A.75 (Manifold Hauptvermutung). Triangulations of homeomorphic topological manifolds of dimension $n \leq 3$ are combinatorially equivalent, i.e. they become isomorphic after subdivision (Ranicki 1996).

The weak manifold version of the Hauptvermutung and the combinatorial triangulation theorem provide the necessary and sufficient algebraic topology to constitute classification theorems for low-dimensional topological manifolds based on homeomorphic cell decompositions. However, the hope to use simplicial complexes as tool for the topological analysis of manifolds in arbitrary dimensions could not be realized. The following sections elaborate on the classification of 0-dimensional up to 3-dimensional manifolds insofar as required for this thesis.

A.5.1 Zero-dimensional and One-dimensional Manifolds

The classification of 0-manifolds is trivial but nevertheless given for the sake of completeness. A connected 0-manifold is simply a discrete space and is commonly called *point*. Disconnected 0-manifolds are discrete sets of points classified by their cardinality.

In literature, 1-manifolds are often synonymously called *curves*. This term conforms to an intuitive spatial understanding of 1-manifolds, although in a technical sense it is less specific than the term 1-manifold. A complete classification of connected 1-manifolds with or without boundary is presented in (Lee 2011). The two main theorems resulting from this work are given in the following. Their proofs build upon the elementary theory developed in the previous sections and can also be reviewed in (Lee 2011).

Theorem A.76 (Classification of 1-manifolds without boundary). Every non-empty connected 1-manifold is homeomorphic to $\mathbb{S}^1$ if it is compact, and to $\mathbb{R}^1$ if it is not (Lee 2011).

Theorem A.77 (Classification of 1-manifolds with boundary). A connected 1-manifold with non-empty boundary is homeomorphic to $[0,1]$ if it is compact, and to $[0, \infty)$ if it is not (Lee 2011).

The classification of connected 1-manifolds is hence based on the two topological invariants of compactness and presence of boundary. Non-connected 1-manifolds result from the disjoint union of connected components each of which is again a 1-manifold that can be characterized according to the above theorems.

This thesis is mainly concerned with compact 1-manifolds. The only type of a closed 1-manifold (i.e. connected, compact, and without boundary; cf. definition A.44) is the *circle* which is an homeomorphic image of the unit 1-sphere $\mathbb{S}^1$. A connected compact 1-manifold with boundary is said to be a *line segment* and is homeomorphic to the closed unit 1-ball $\overline{\mathbb{B}}^1$.

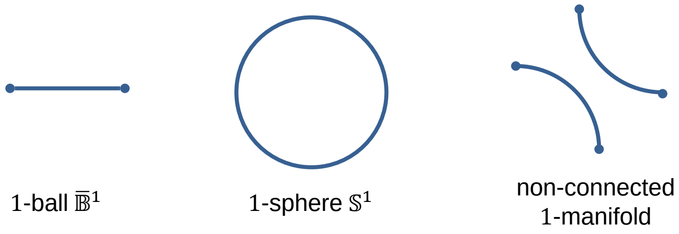


Figure 276: Examples of 1*-manifolds (from left to right: a line segment, a circle, and a non-connected* 1*-manifold).*

Proposition A.78. Every 1-manifold is orientable.

This proposition follows from the fact that every 1-manifold admits a triangulation and, thus, is homeomorphic to the polyhedron of a 1-dimensional simplicial complex. It is easily conceivable that for all four types of 1-manifolds characterized by theorem A.76 and theorem A.77 a consistent choice of orientation on each 1-simplex which participates in the corresponding simplicial complex can be made.

A famous theorem based on closed 1-manifolds is the *Jordan curve theorem* named after Camille Jordan who was the first to prove it in 1887. The Jordan curve theorem is an important tool used in complex analysis and low-dimensional topology.

Theorem A.79 (Jordan curve theorem). Let C be a closed 1-manifold topologically embedded in $\mathbb{R}^2$. Then C is called *Jordan Curve* and separates $\mathbb{R}^2$ into two connected components.

Precisely, the complement $\mathbb{R}^2 \backslash C$ consists of exactly two connected components, i.e. $\mathbb{R}^2 \backslash C = \mathcal{D}_1 \cup \mathcal{D}_2$ with $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$. One of the components is bounded (commonly called the *interior*) whereas the other is unbounded (commonly called the *exterior*). Both components share C as their topological boundary.

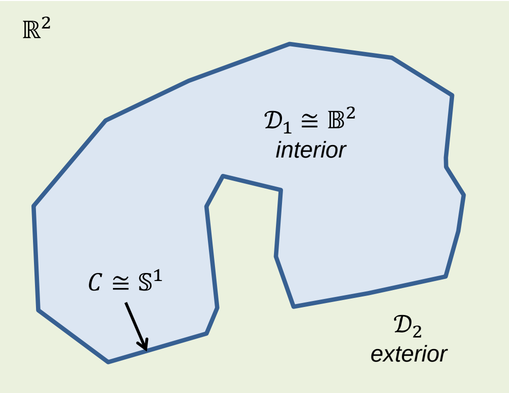


Figure 277: Example of a Jordan curve C in $\mathbb{R}^2$ and the resulting interior component $\mathcal{D}_1$ (blue) and exterior component $\mathcal{D}_2$ (green).

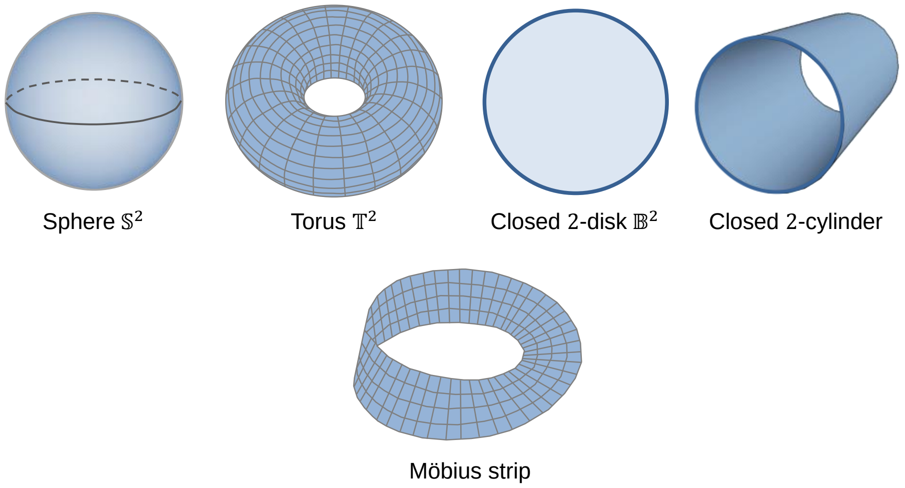
The *Jordan-Schoenflies theorem* is an addition to the Jordan curve theorem and asserts that under the conditions of theorem A.79 there is a homeomorphism $f: \mathbb{R}^2 \to \mathbb{R}^2$ of $\mathbb{R}^2$ onto itself such that $f(C)$ is the unit circle in the plane. Then the interior component with bounded closure is homeomorphic to the closed unit disk (i.e., the closed unit 2-ball) and thus $\mathbb{R}^2 \backslash C = \mathcal{D}_1 \cup \mathcal{D}_2$ with $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$ and $\overline{\mathcal{D}}_1 \cong \overline{\mathbb{B}}^2$.

Both $\mathcal{D}_1$ and $\mathcal{D}_2$ as well as $\mathbb{R}^2$ are 2-manifolds and subject to classification in the next section.

A.5.2 Two-dimensional Manifolds

The classification of topological 2-manifolds has been a field of detailed study for a long time and is to be seen an important driver of modern topology. As a result, 2-manifolds are nowadays well understood mathematically and characterized up to homeomorphism. The term *surface* is commonly used in literature as synonym to reference a 2-manifold.

Examples of non-compact 2-manifolds without boundary are $\mathbb{R}^2$ itself as well as open subsets thereof such as open disks homeomorphic to the open unit 2-ball. The unbounded component which results from embedding a Jordan Curve in $\mathbb{R}^2$ is an example of a non-compact 2-manifold with boundary. Important compact surfaces are illustrated in the following figure. The sphere $\mathbb{S}^2$ and the torus $\mathbb{T}^2$ $(= \mathbb{S}^1 \times \mathbb{S}^1)$ are compact 2-manifolds without boundary. Examples of compact 2-manifolds with boundary are the closed disk, the cylinder surface including both boundary circles, as well as the Möbius strip.



Sphere $\mathbb{S}^2$ Torus $\mathbb{T}^2$ Closed 2-disk $\mathbb{B}^2$ Closed 2-cylinder

Möbius strip

Figure 278: Examples of compact surfaces.

Similar to the 1-manifold case, the topological properties of compactness and number of boundary components are two determining factors for the characterization of 2-manifolds. But they obviously do not suffice for a complete classification up to topological equivalence. For example, although both the sphere $\mathbb{S}^2$ and the torus $\mathbb{T}^2$ are compact 2-manifolds without boundary there is no continuous deformation between them. And even though both the closed disk $\overline{D}^2$ and the Möbius strip are compact and have a single circle as manifold boundary there is no homeomorphism taking one into the other.

Closed surfaces. In the following, the classification of 2-manifolds is first developed for the class of compact, connected 2-manifolds without boundary which are equivalently called *closed surfaces*[86]. The most fundamental examples for closed surfaces are the sphere $\mathbb{S}^2$, the torus $\mathbb{T}^2$, and the projective plane $\mathbb{R}P^2$. From these, further closed surfaces can be constructed using the *connected sum construction*. The connected sum of two closed surfaces is obtained by removing an open disk from each and gluing the surfaces along the boundaries of the resulting holes. Formally, the connected sum is not restricted to closed surfaces but is generally defined for n-dimensional manifolds as follows.

Definition A.80 (Connected sum). Let M_1 and M_2 be two connected n-manifolds. The subspaces $M'_i = M_i \setminus B_i{}^n$ which are formed by deleting open n-balls $B_i{}^n \subseteq M_i$ are n-manifolds with boundary whose boundaries are homeomorphic to $\mathbb{S}^{n-1}$ (cf. proposition A.43). If $f: \partial M'_2 \to \partial M'_1$ is any homeomorphism, the adjunction space $M'_1 \cup_f M'_2$ is called a *connected sum* of M_1 and M_2 and is denoted by $M_1 \# M_2$ (Lee 2011).

[86] The reader's attention is again drawn to the fact that the meaning of the term *closed* in the context of manifold theory differs from its meaning in general point-set topology. For example, in literature the term *closed disk* is used to name a *closed subset* of $\mathbb{R}^2$ which is homeomorphic to the closed unit 2-ball. But a *closed disk* is not a *closed surface* because it has a manifold boundary.

The connected sum $M_1 \# M_2$ is itself a connected n-manifold. It can be shown that the connected sums of a given pair $M_1 \# M_2$ can at most result in two non-homeomorphic manifolds for one of which f is orientation-preserving and for the other orientation-reversing (Lee 2011).

When removing an open n-ball from the n-sphere $\mathbb{S}^n$, the resulting manifold space is homeomorphic to a closed n-ball. Thus, forming the connected sum $M \# \mathbb{S}^n$ with the n-sphere is equivalent to cutting out an open n-ball from M and gluing back a closed n-ball along the resulting boundary sphere. This does not change the manifold M and correspondingly $\mathbb{S}^n$ is an identity element of the connected sum construction.

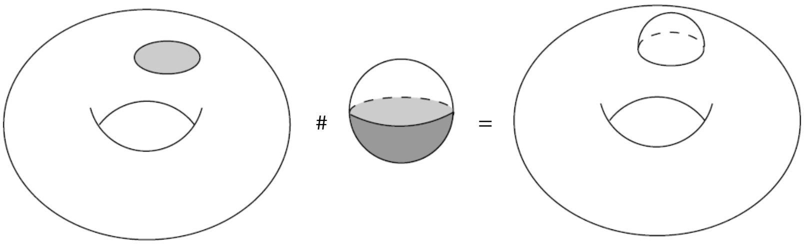


Figure 279: The connected sum $\mathbb{T}^2 \# \mathbb{S}^n$ which results in $\mathbb{T}^2$ (taken from Lee 2011).

The connected sum $M \# \mathbb{T}^2$ of a 2-dimensional manifold M with the torus $\mathbb{T}^2$ is equivalent to removing two open disks from M and attaching a cylinder at both boundary circles. As illustrated in the following figure, this operation is often called "attaching a *handle*" to M, and the torus is sometimes referred to as a sphere with one handle.

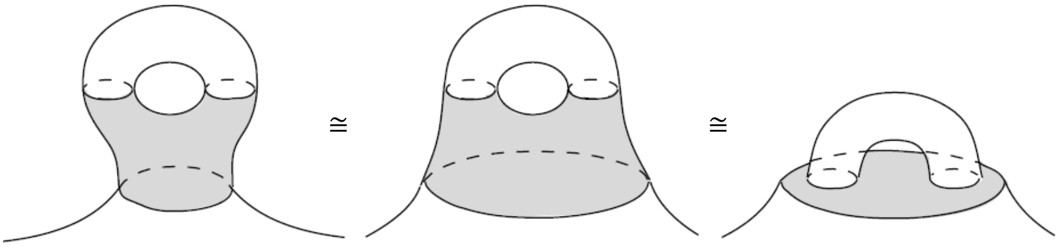


Figure 280: Example of attaching a handle by the operation $M \# \mathbb{T}^2$ (taken from Lee 2011).

In literature, the connected sum of m 2-dimensional tori $\mathbb{T}^2 \# \mathbb{T}^2 \# \ldots \# \mathbb{T}^2$ is commonly said to be an *m-holed torus* or, equivalently, a sphere with m handles. It easily follows that the 0-holed torus is exactly the 2-dimensional sphere $\mathbb{S}^2$ and that the 1-holed torus is simply $\mathbb{T}^2$ itself. When relating to its dimensionality, the term *n-dimensional torus* or just *n-torus* is used and denoted by $\mathbb{T}^n$. This terminology is consistently applied throughout this thesis, although other sources interchangeably use the term n-torus to reference an n-holed torus or to denote its dimension n.

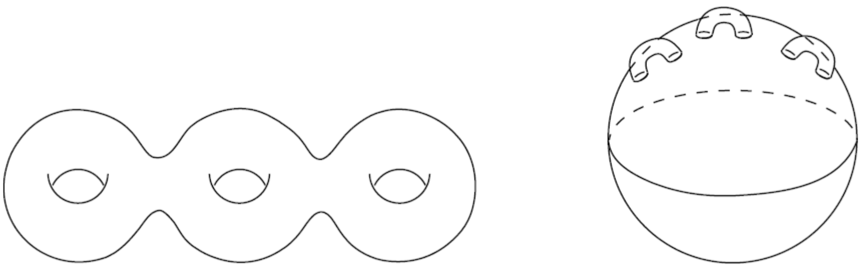


Figure 281: A 3-holed torus (left) or, equivalently, a 2-sphere with three handles (taken from Lee 2011).

The connected sum of two projective planes $\mathbb{R}P^2 \# \mathbb{R}P^2$ yields the famous *Klein bottle*. It is important to note that both the projective plane and the Klein bottle cannot be embedded in $\mathbb{R}^3$ without self-intersections. This is illustrated in the following figure for the Klein bottle.

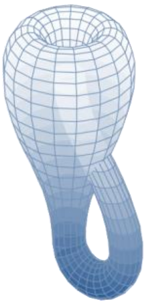


Figure 282: The Klein bottle.

The following lemma on the connected sum of the torus and the projective plane completes the enumeration of connected sums of the fundamental closed surfaces.

Lemma A.81. The connected sum $\mathbb{T}^2\#\mathbb{R}P^2$ is homeomorphic to $\mathbb{R}P^2\#\mathbb{R}P^2\#\mathbb{R}P^2$ (Lee 2011)

It follows from theorem A.74 that every compact surface admits a triangulation. However, in contrast to 1-manifolds, not every compact surface is orientable. The best-known example of a non-orientable compact surface with boundary is the Möbius strip. It can be obtained by identifying the opposite sides of the unit square $[0,1] \times [0,1]$ after a half-twist, i.e. via $(0, t) \sim (1,1 - t)$. The Möbius strip does not allow a consistent choice of orientation on each of its 2-simplices after triangulation. Put differently, a small, oriented circle put inside the Möbius strip will have its orientation reversed after moving it once around the strip even though the orientation is not altered while moving. The path on which the circle has been moved along the Möbius strip is called an *orientation-reversing* closed curve. If the circle follows the path twice its orientation will be flipped once more into its original state and such a path is called an *orientation-preserving* closed curve. If for a 2-manifold all closed curves are orientation-preserving then the 2-manifold is orientable, otherwise it is not (Edelsbrunner & Harer 2010).

If a 2-manifold is orientable then all its triangulations are orientable. Orientability is hence independent of a particular choice of triangulation. It can be shown that the converse also holds: if a 2-manifold admits an orientable triangulation then the 2-manifold is itself orientable (Mäntylä 1988). Each connected orientable 2-manifold has exactly two different consistent choices of orientation which conform to both orientations that are constituted on every 2-simplex by the ordering of its vertices (cf. definition A.65). For example, for a connected oriented surface it is possible to distinguish two sides that are commonly called front-side and back-side. This is not possible for the Möbius strip. It is important to point out that this notion of orientability can be generalized to manifolds in arbitrary dimensions. As shown by proposition A.78, the 1-dimensional case is trivial because all 1-manifolds are orientable.

The boundary of the Möbius strip is a single circle. After removing an open disk from a closed surface a homeomorphism identifying its boundary circle with the boundary of the Möbius strip can be chosen. This operation yields a new closed surface and is also known as "adding a *cross-cap*" (Edelsbrunner & Harer 2010). Adding a cross-cap to the sphere $\mathbb{S}^2$ results in the projective plane $\mathbb{R}P^2$, which is therefore also said to be a sphere with one cross-cap. The Klein bottle results from adding two cross-caps to the sphere which conforms to its construction as connected sum of two projective planes.

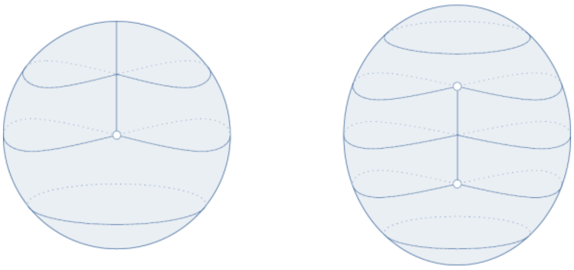


Figure 283: The projective plane $\mathbb{R}P^2$ (left) and the Klein bottle (right) (taken from Edelsbrunner & Harer 2010).

It can be shown that the sphere together with the connected sum of one or more tori exhaust the family of orientable closed surfaces, whereas the connected sum of one or more projective planes exhausts the family of non-orientable closed surfaces. Based on these results the main theorem for the classification of closed surfaces can be given.

Theorem A.82 (Classification of closed surfaces). Every non-empty, compact, connected 2-manifold without boundary is homeomorphic to some member of one of the following three families:

(i) The sphere $\mathbb{S}^2$,
(ii) the connected sum of one or more copies of the torus $\mathbb{T}^2$, or
(iii) the connected sum of one or more copies of the projective plane $\mathbb{R}P^2$.

Any two closed surfaces in the list are not homeomorphic.

One of the earliest results in surface theory is the *Euler's formula* named after Leonhard Euler. In its classical form, the Euler's formula relates the numbers of vertices V, edges E, and faces F of a compact polyhedral surface to their alternating sum $\chi = V - E + F$ where χ is called the *Euler characteristic* of the surface. In case of any simply connected convex polyhedron the Euler's formula states that $V - E + F = 2$. It is important to mention that the classical Euler's formula refers to a "polyhedron" as an elementary 3-dimensional geometric solid. This may not be confused with the notion of "polyhedron" in algebraic topology where the term refers to the underlying space of a simplicial complex (cf. definition A.70). However, the fact that the formula $V - E + F = 2$ holds for any surface bounding a convex geometric polyhedron independent of its triangulation (i.e., the number of vertices, edges, and triangles) indicates that the Euler characteristic is truly a topological property and not just a geometric one.

Correspondingly, the Euler characteristic has been generalized in literature to the much wider class of arbitrary n-dimensional finite CW complexes for which it is defined as the alternating sum of the numbers of cells in each dimension (Lee 2011).

Proposition A.83 (Euler characteristic of a finite CW complex). Let X be a finite CW complex of dimension n. Then the *Euler characteristic of* X is defined by $\chi(X) = \sum_{k=0}^{n}(-1)^k c_k$, where c_k denotes the number of k-cells in the CW decomposition of X.

For example, the minimal CW decomposition of $\mathbb{S}^2$ consists of one 0-cell and one 2-cell (cf. figure 270). The Euler characteristic of the corresponding CW complex X is $\chi(X) = 1 \cdot 1 + (-1) \cdot 0 + 1 \cdot 1 = 2$ which conforms to the result of the classical Euler's formula. An important theorem in topology based on the above proposition states that the Euler characteristic is in fact a topological invariant for finite CW complexes, and thus is preserved under homeomorphism. The proof can be found in (Lee 2011) but requires techniques from homology theory which are beyond the scope of this thesis.

Theorem A.84 (Invariance of the Euler characteristic). Let X and Y be two finite CW complexes whose underlying topological spaces are homeomorphic. Then $\chi(X) = \chi(Y)$.

The above can be rephrased such that X and Y are topologically non-equivalent if their Euler characteristic differs. This allows a purely combinatorial and hence efficient check.

The Euler characteristic can be related to the number of handles and the number of cross-caps of a closed surface respectively. The fundamental theory enabling this relation was given by Poincaré, and thus it is commonly denoted as *Euler-Poincaré characteristic.*

Theorem A.85 (Euler-Poincaré characteristic of a closed surface). For any closed surface M and a positive integer g, the Euler characteristic $\chi(M)$ is equal to

(i) 2 for the sphere,
(ii) $2 - 2g$ for the connected sum of g tori, or
(iii) $2 - g$ for the connected sum of g projective planes.

The integer g is called the *genus* of the closed surface. The genus is equal to the number of handles for the family of orientable closed surfaces. The connected sum of g tori is therefore synonymously called *orientable closed surface of genus* g. Equivalently, the connected sum of g projective planes is said to be the *non-orientable closed* *surface of genus g*, and in this case the genus is equal to the number of cross-caps. The sphere itself is the *closed surface of genus* 0 (Lee 2011). It easily follows that attaching a handle to a closed surface decreases the Euler characteristic by 2 and that adding a cross-cap decreases it by 1.

Although the Euler characteristic is an important topological invariant, it is not sufficient by itself to completely classify closed surfaces. For example, both the torus and the Klein bottle have Euler characteristic 0 but obviously are not homeomorphic. This example can be generalized to a connected sum of n tori which cannot be distinguished from the connected sum of $2n$ projective planes based on their Euler characteristic. The additional topological property required for distinguishing both is their orientability. As an intermediate result it can be deduced that any closed surface is completely classified up to homeomorphism by its Euler characteristic and its orientability.

In the context of closed surfaces, the Jordan-Schoenflies theorem has a natural extension by replacing the plane $\mathbb{R}^2$ with the sphere $\mathbb{S}^2$ such that a simple closed curve C on the sphere $\mathbb{S}^2$ separates the sphere into two connected components each of which has a closure homeomorphic to the closed disk. This extension does not hold for any closed surface though. For example, it takes two disjoint simple closed curves two disconnect the torus $\mathbb{T}^2$. When related to the genus of a closed surface, the extension can be appropriately rephrased for arbitrary closed surfaces.

Theorem A.86 (Jordan curve theorem for closed surfaces). The maximum number of disjoint simple closed curves which can be cut from a closed surface of genus g without disconnecting it is equal to

(i) g for an orientable surface,
(ii) $g + 1$ for a non-orientable surface.

Compact surfaces with boundary, non-compact surfaces. A compact 2-manifold with boundary can be obtained from a closed surface by cutting one or more *holes* into the latter. Examples for compact surfaces with boundary are the closed disk (sphere with one hole), the Möbius strip (projective plane with one hole), the cylinder (sphere with two holes), and the annulus (sphere with two holes) which is homeomorphic to the cylinder.

Technically, a hole results from removing an open disk from a closed surface which yields a circle as boundary component. Removing n disjoint open disks correspondingly yields n disjoint circles as boundary components. This conforms with proposition A.43 which states that the manifold boundary of a 2-manifold with boundary is a 1-manifold without boundary. According to the classification of 1-manifolds, the only type of a connected, compact 1-manifold without boundary is the circle. In case of more than one hole, the manifold boundary is a non-connected, compact 1-manifold without boundary that is a union of disjoint connected components all of which are circles. Since compact surfaces with boundary can be constructed from closed surfaces, their classification is related to theorem A.82 for closed surfaces.

Theorem A.87 (Classification of compact surfaces with boundary). Every compact surface with boundary is completely classified up to homeomorphism by its number of boundary components and the class of the corresponding closed surface.

The Euler characteristic of a closed surface decreases by 1 for each removed open disk. In general, a compact surface of genus g and with h holes has Euler characteristic $\chi(M) = 2 - 2g - h$ for the family of orientable surfaces and $\chi(M) = 2 - g - h$ in the non-orientable case. Thus, each compact surface with boundary is fully determined by its genus g, its number of holes h, and its orientability.

Most literature sources credit the first profound proof of the classification theorem for compact surfaces to Dehn and Heegaard in 1907 (Dehn & Heegaard 1907) who applied the assumption that every compact surface admits a triangulation. This triangulation assumption however was first proved almost two decades later by Radó in 1925 (Radó 1925) (cf. section A.5), whose results thus complete the proof of Dehn and Heegaard. It requires much more theory though to classify non-compact surfaces and, surprisingly, it took a century from the work of Dehn and Heegaard to a complete classification of 2-manifolds including the non-compact case in 2007 (Prishlyak & Mischenko 2007).

A general classification theorem for non-compact surfaces with or without boundary is beyond the scope of this thesis and the reader is redirected to (Prishlyak & Mischenko 2007) for a detailed overview. The only type of non-compact 2-manifold that plays an important role in the course of this thesis is $\mathbb{R}^2$ itself from which zero or more open disks have been removed. In literature, this surface is commonly called *universal face* and represents unbounded Euclidean 2-space. The universal face is exactly the unbounded exterior component which results from embedding zero or more Jordan Curves in $\mathbb{R}^2$ whose disjoint union forms the topological boundary of the universal face.

A.5.3 Three-dimensional Manifolds

A high hope associated with the complete classification of compact 2-manifolds at the beginning of the 20th century was to carry the results and facts to the third dimension. However, this hope has not been realized until now and the classification of 3-manifolds remains an on-going work in current research. In fact, 3-manifolds pose much more complicated problems than 2-manifolds.[87] The objective of this section is therefore to provide a classification of a *manifold solid* as the topological foundation for the spatial modelling of 3-dimensional physical objects.

As a preliminary definition, a manifold solid is characterized as connected, compact, orientable 3-manifold with boundary that lives in an ambient space $\mathbb{R}^3$ and whose boundary is a compact surface. The prototypical example of a manifold solid is the closed unit 3-ball $\overline{\mathbb{B}}^3$ having the unit 2-sphere $\mathbb{S}^2$ as boundary.

The orientability property is again best viewed in terms of triangulation. Since per theorem A.74 all compact 3-manifolds are homeomorphic to the polyhedron of some simplicial complex, orientability of a 3-manifold requires the simplicial complex to admit a consistent choice of orientation on all its 3-simplices. If such an orientation exists, it also induces an orientation on all boundary faces of each 3-simplex (cf. definition A.67). It hence follows that the orientability of a manifold solid has a direct impact on the class of its boundary surface. Precisely, if a connected compact 3-manifold is oriented then its boundary surface is oriented and thus must be a member of the family of orientable closed surfaces.

Theorem A.88. Let M be a connected 3-manifold with boundary embedded in $\mathbb{R}^3$. Then the boundary of M is an embedded orientable closed surface (Mäntylä 1988).

Non-orientable closed surfaces are consequently ruled out as the boundary of a manifold solid which implies the Euler characteristic of the boundary surface to be $2 - 2g$. This intuitively conforms to our spatial understanding because non-orientable closed surfaces cannot be immersed in Euclidean 3-space without self-intersections. The following proposition records this fact which follows from the Whitney embedding theorem.

Proposition A.89. Every closed surface with a penetration-free embedding in $\mathbb{R}^3$ is orientable (Gröger & Plümer 2011a).

It is however not sufficient to only examine the relation of a manifold solid to its boundary surface in order to avoid pathological cases. There are also 3-manifolds which are homeomorphic to a closed 3-ball but which nevertheless yield unreasonable physical objects when embedded in $\mathbb{R}^3$. Examples are 3-manifolds having a fractal-like surface. Thus, the embedding of the 3-manifold in its ambient 3-dimensional Euclidean space has to be additionally considered in order to obtain well-behaved manifold solids (Mäntylä 1988).

In two dimensions, the Jordan curve theorem constitutes the relation between $\mathbb{R}^2$ and an embedded closed 1-manifold. Its generalization to higher dimensions was presented in 1912 by L. E. J. Brouwer who showed that Euclidean space of dimension n separates into two connected domains by the insertion of a closed $(n-1)$-manifold. This fact is known as the *Jordan-Brouwer separation theorem*.

Theorem A.90 (Jordan-Brouwer separation theorem). Let M be a closed $(n-1)$-dimensional manifold topologically embedded in $\mathbb{R}^n$. Then the complement $\mathbb{R}^n \backslash M$ consists of exactly two connected components whose common topological boundary is M.

Equivalently to the Jordan curve theorem, the bounded component is called the *interior* whereas the unbounded one is called the *exterior*. However, the Schoenflies addition in its literal form was disproved for dimension $n \geq$

[87] One of the most famous problems became known as the *Poincaré conjecture*. In 1904, Poincaré asked whether every simply connected closed 3-manifold is homeomorphic to the 3-sphere but not enough machinery was developed to prove an answer. It became one of the greatest challenges in topology for nearly one century before Grigori Perelman finally presented a proof in 2003. For dimensions $n \geq 4$, Markov showed in 1958 that there is no general algorithm for distinguishing two arbitrary topological manifolds (Lee 2011).

3. In 1924, J. W. Alexander (Alexander 1924) presented a counterexample named the *Alexander horned sphere* which is a particular embedding of the 2-sphere in $\mathbb{R}^3$ that results in an infinitely convoluted, intertwined surface. Although the horned sphere together with its interior is homeomorphic to a closed 3-ball, there is no homeomorphism of $\mathbb{R}^3$ onto itself that takes the horned sphere to the unit 2-sphere $\mathbb{S}^2$. As a result, the exterior component of the horned sphere is not simply connected and thus not topologically equivalent to the complementary domain of $\mathbb{S}^2$ in $\mathbb{R}^3$.

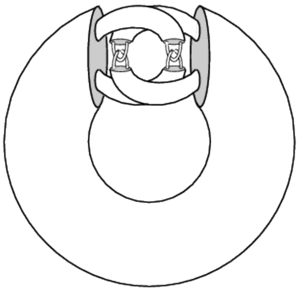


Figure 284: The Alexander horned sphere.

Embeddings in $\mathbb{R}^3$ like the one of the horned sphere are called *wild embeddings* and result in non-well-behaved manifold solids. For dimension $n = 3$ it can be shown that the Schoenflies addition holds if M is embedded polyhedrally in $\mathbb{R}^3$, i.e. the embedding is homeomorphic to the polyhedron of a finite simplicial complex (J. W. Alexander 1924). Polyhedral embeddings are also called *tame embeddings*. For example, if a 2-sphere S is tamely embedded in $\mathbb{R}^3$, then the unbounded exterior component is simply connected and the closure of the bounded component is homeomorphic to the closed unit 3-ball, i.e. $\mathbb{R}^3 \backslash S = B_1 \cup B_2$ with $B_1 \cap B_2 = \emptyset$ and $\bar{B}_1 \cong \bar{\mathbb{B}}^3$. Then the union $S \cup B_1 = \bar{B}_1$ is a well-behaved manifold solid.

For the sake of completeness, it is pointed out that the Jordan-Brouwer separation theorem also has a natural extension by replacing $\mathbb{R}^n$ with the n-dimensional sphere $\mathbb{S}^n$. It easily follows that a closed 2-manifold M^2 tamely embedded in $\mathbb{S}^3$ separates $\mathbb{S}^3$ into two connected components both of which are topologically equivalent to $\bar{\mathbb{B}}^3$ and thus well-behaved manifold solids. In case of a wild embedding, the union of M^2 with the non-simply connected component results in a *crumpled cube* which is topologically different from a closed 3-ball. Crumpled cubes are further examples of non-well-behaved manifold solids.

Finally, this leads to the definition of a *manifold solid* as stated in the following.

Definition A.91 (Manifold solid). A *manifold solid* is a non-empty, connected, compact 3-manifold with boundary embedded in $\mathbb{R}^3$ whose boundary is a closed, oriented 2-manifold tamely embedded in $\mathbb{R}^3$.

A manifold solid is also called *single-shell manifold solid* in literature (Mäntylä 1988). This corresponds to the fact that the manifold solid is embraced by a connected 2-manifold which is also said to be the shell of the solid. Relaxing the connectedness property of the shell gives rise to *multi-shell manifold solids*.

Definition A.92 (k-shell manifold solid). A *k-shell manifold solid* is a non-empty compact 3-manifold embedded in $\mathbb{R}^3$ with $k \geq 1$ boundary components whose disjoint union is a compact, oriented 2-manifold without boundary tamely embedded in $\mathbb{R}^3$.

A single-shell manifold solid obviously is the special case of a k-shell manifold solid for $k = 1$. In case $k > 1$, the solid object is equivalently said to be a *multi-shell manifold solid*. The k connected components of the boundary are the shells of the manifold solid and are required to share the same dimension 2 to ensure the invariance of dimension of the boundary (as postulated for disconnected manifolds in general in section A.2). Although the boundary components of a multi-shell manifold solid M are disconnected, its interior $Int(M)$ need not be disconnected. Suppose $Int(M)$ is a connected topological space then M intuitively can be understood to have $k - 1$ internal voids or cavities (Mäntylä 1988).

Definition A.93 (Connected k-shell manifold solid). A connected *k-shell manifold solid M* is a *k-shell manifold solid* whose interior $Int(M)$ is restricted to be connected. In case $k > 1$, M is said to have $k - 1$ internal voids.

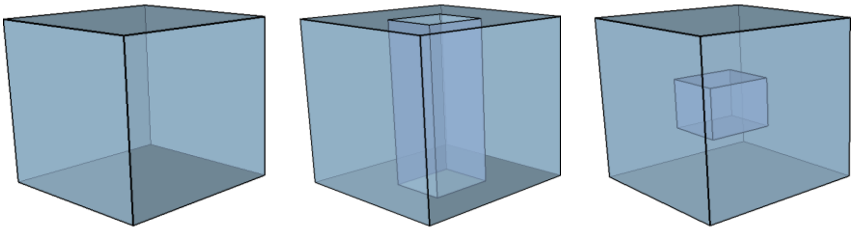


Figure 285: Examples of manifold solids (from left to right: a 1*-shell manifold solid being homeomorphic to* $\overline{\mathbb{B}}^3$*, a* 1*-shell manifold solid with through hole, and a connected* 2*-shell manifold solid with one internal void).*

Analogous to the concept of a universal face in two dimensions, a 3-dimensional *universal solid* can be defined as the exterior component which results from tamely embedding zero or more closed 2-manifolds into $\mathbb{R}^3$ following the Jordan-Brouwer separation theorem. The universal solid itself is unbounded and has the disjoint union of all tamely embedded closed 2-manifolds as topological boundary. Equivalently, it can be obtained from $\mathbb{R}^3$ by removing the set of interior points of one or more manifold solids. The universal solid is a non-compact 3-manifold with or without boundary and is used in the course of this thesis for the representation of unbounded Euclidean 3-space.

As introduced in the previous chapter, an important topological property for the classification of the closed, oriented surface S bounding a connected component of a manifold solid is its Euler characteristic $\chi(S) = 2 - 2g$ (cf. theorem A.85) which can be equivalently computed as alternating sum of the numbers of cells in a CW decomposition of S. Obviously, the Euler characteristic can also be computed for a finite CW complex of the manifold solid itself. For example, the minimal CW decomposition of $\overline{\mathbb{B}}^3$ is achieved by attaching a single 3-cell to the minimal CW decomposition of its boundary $\mathbb{S}^2$ consisting of one 0-cell and one 2-cell. The Euler characteristic $\chi(X)$ of the corresponding CW complex X thus is $\chi(X) = 1 - 0 + 1 - 1 = 1$. Since the Euler characteristic is invariant under homeomorphism, the same result is obtained for any single-shell manifold solid being homeomorphic to $\overline{\mathbb{B}}^3$. For example, the solid 3-cube can be decomposed into eight 0-cells, twelve 1-cells, six 2-cells, and one 3-cell yielding $\chi(X) = 8 - 12 + 6 - 1 = 1$.

In order to completely characterize n-manifolds beyond 2-dimensional surfaces, Henri Poincaré proved the existence as well as the topological invariance of a set of positive integers $P_1, P_2, \dots, P_{n-1}$ representing global properties of a topological space which he derived from certain suitably chosen matrices describing the n-manifold (cf. Veblen & Alexander 1912). Poincaré called these integers *Betti numbers* after the Italian mathematician Enrico Betti. Nowadays, the Betti numbers are formally stated in terms of homology theory.

Theorem A.94 (Betti numbers of a topological space). The integer $\beta_k(X) = \operatorname{rank} H_k(X)$ denoting the rank of the kth homology group $H_k(X)$ of a topological space X is called the *kth Betti number* of X (Lee 2011).

As homology theory is beyond the scope of this thesis, we follow the more general understanding that the Betti numbers allow for counting the number of holes of different dimensions in an orientable, compact n-manifold M. This allows for an informal but intuitive definition of the first three Betti numbers so that

(i) $\beta_0(M)$ is the number of connected components,

(ii) $\beta_1(M)$ is the number of 2-dimensional holes, and

(iii) $\beta_2(M)$ is the number of 3-dimensional holes (equivalently, voids or cavities).

For example, for a closed, orientable surface S of genus g its Betti numbers take the values $\beta_0(S) = 1$, $\beta_1(S) = 2g$, and $\beta_2(S) = 1$. The alternating sum $\beta_0(S) - \beta_1(S) + \beta_2(S)$ equals the Euler characteristic $\chi(S) = 2 - 2g$ of the surface. In fact, the definition of the Euler characteristic can be rephrased and generalized to arbitrary topological spaces based on Betti numbers.

Theorem A.95 (Euler characteristic of a topological space). The *Euler characteristic* $\chi(X)$ of a topological space X is defined as alternating sum $\chi(X) = \sum_k (-1)^k \beta_k(X)$ where $\beta_k(X)$ is the kth Betti number of X provided that each $\beta_k(X)$ is finite and $\beta_k(X) = 0$ for sufficiently large k (Lee 2011).

For a compact n-manifold M it holds that all Betti numbers are finite and that $\beta_k(M) = 0$ as soon as k exceeds the dimension n (Early 1999). This allows for restricting theorem A.95 so that the Euler characteristic $\chi(M)$ is given by $\sum_{k=0}^{n}(-1)^k\beta_k(M)$ for the class of compact n-manifolds. Moreover, if M is limited to be a k-shell manifold solid whose ambient space is $\mathbb{R}^3$ per definition, then it obviously follows that $\beta_3(M) = 0$ in this case. The sequence of the first three Betti numbers therefore suffices to characterize a k-shell manifold solid. A formal proof of this fact along manifold solids being homeomorphic to the 3-dimensional closed ball with h solid handles is provided by (Aleksandrov 1998).

Proposition A.96 (Euler characteristic of k-shell manifold solid). For any k-shell manifold solid M, the Euler characteristic $\chi(M)$ is equal to the alternating sum $\beta_0(M) - \beta_1(M) + \beta_2(M)$ of its first three Betti numbers.

For example, since the single-shell manifold solid $\overline{\mathbb{B}}^3$ has one connected component but neither a circular hole nor an internal void, its Euler characteristic is $\chi(\overline{\mathbb{B}}^3) = 1 - 0 + 0 = 1$ which agrees with the Euler characteristic of the CW decompositions of $\overline{\mathbb{B}}^3$ and the solid 3-cube as presented above.

A.6 Topological Relationships between Spatial Objects

Especially in the field of geographic information science, the practical use of topology is twofold (Zlatanova 2000). First, topology provides the tool to define the *intrinsic properties of spatial objects* (e.g., points, curves, surfaces, solids) that form the primitives for describing the spatial aspects of real-world objects and phenomena. Second, topology is used as formalism for categorizing the invariant *topological relationships* between two such objects which are induced by their spatial configuration.

An early work in the categorization of the topological relationships is the well-known 4-*intersection model* (4IM) developed by (Egenhofer & Franzosa 1991). It is defined in terms of four intersections of the topological boundaries $(\partial A, \partial B)$ and interiors $(Int(A), Int(B))$ of two objects A and B embedded in Euclidean space. The topological relationship is classified based on the content of each intersection (i.e., emptiness and non-emptiness) and hence renders a topological invariant. The model is formally expressed using the following 2×2 operation matrix:

$$\mathfrak{I}_4(A,B) = \begin{pmatrix} Int(A) \cap Int(B) & Int(A) \cap \partial B \\ \partial A \cap Int(B) & \partial A \cap \partial B \end{pmatrix}. \tag{A.97}$$

The classification scheme of the 4IM was initially defined for the topological relationships of 2-dimensional manifolds with a single boundary component embedded in $\mathbb{R}^2$. However, it can be generalized to connected n-dimensional manifolds with one or more boundary components in $\mathbb{R}^n$ (Egenhofer & Sharma 1993, Egenhofer et al. 1994, Zlatanova 2000). For such objects, only eight of the $2^4 = 16$ relationship classes that are expressible with $\mathfrak{I}_4(A,B)$ can be spatially realized and are commonly named *disjoint*, *contains*, *inside*, *equal*, *meet*, *covers*, *coveredBy*, and *overlap*. The following figure 286 illustrates the 2-dimensional case.

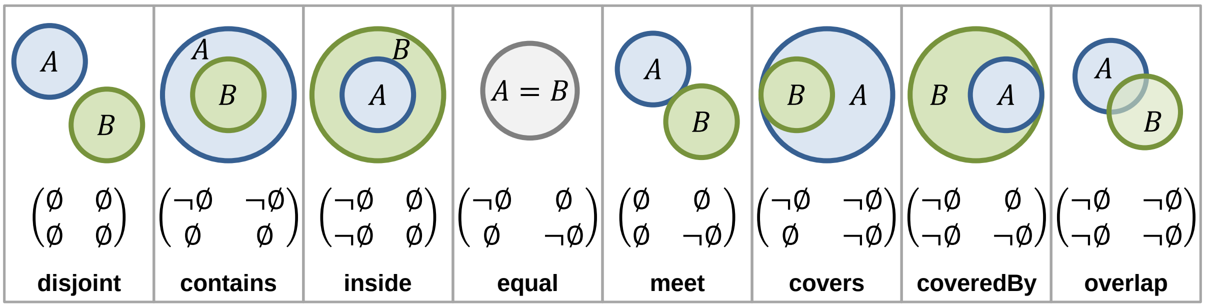

A
B
$\begin{pmatrix} \emptyset & \emptyset \\ \emptyset & \emptyset \end{pmatrix}$
disjoint
A
B
$\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset \end{pmatrix}$
contains
B
A
$\begin{pmatrix} \neg\emptyset & \emptyset \\ \neg\emptyset & \emptyset \end{pmatrix}$
inside
A = B
$\begin{pmatrix} \neg\emptyset & \emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$
equal
A
B
$\begin{pmatrix} \emptyset & \emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$
meet
B
A
$\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \emptyset & \neg\emptyset \end{pmatrix}$
covers
B
A
$\begin{pmatrix} \neg\emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset \end{pmatrix}$
coveredBy
A
B
$\begin{pmatrix} \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset \end{pmatrix}$
overlap

Figure 286: The eight topological relationships according to the 4IM (Egenhofer & Sharma 1993).

Two important extensions of the 4IM have been proposed in literature. First, the 9-*intersection model* (9IM, Egenhofer & Herring 1991) additionally considers the location of the interiors and boundaries of both objects with respect to their exteriors $(Ext(A), Ext(B))$ which yields a 3×3 operation matrix with $2^9 = 512$ possible relationship classes (not all of them can be spatially realized though). Second, the *dimensionally extended* 9-*intersection model* (DE-9IM, Eliseo Clementini & Paolino Di Felice 1995) also observes the dimension of the intersection content in addition to its emptiness. Both extensions obviously allow for a more fine-grained differentiation of the spatial configurations and the corresponding topological relationships between two objects. The additional expressivity of the 9IM is however only required if the co-dimension of one or both objects with respect to its ambient space is greater than zero. Put differently, for connected n-dimensional manifolds with one or more boundary components embedded in $\mathbb{R}^n$ (and thus with co-dimension of zero) both the 4IM and the 9IM yield identical results (Egenhofer & Sharma 1993). Note that the notion of interior, boundary, and exterior again might be confusing especially when considering spatial objects with co-dimension greater than zero. For example, consider a surface being homeomorphic to the closed disk as the ones shown in figure 286 but assume that it is embedded in Euclidean 3-space $\mathbb{R}^3$ with co-dimension one. Then the *topological boundary* of the disk according to definition A.5 is the entire disk itself (cf. appendix A.2 and Lee 2011). As a consequence, the 4IM as illustrated in figure 286 (as well as the 9IM) would yield different results when choosing $\mathbb{R}^3$ as ambient space. The 4IM as well as the 9IM therefore implicitly apply the notion of *manifold interior* and *boundary* although this fact is not explicitly mentioned in the papers of (Egenhofer & Franzosa 1991) and (Egenhofer & Herring 1991).

This thesis is only concerned with the classification of topological relationships between spatial objects having co-dimension zero. Although the DE-9IM generally identifies more detailed relationships than both the 4IM and the 9IM in this case, it is argued in chapter 3.2 that only a subset of all possible relationships is actually required for this thesis which can be expressed based on the 4IM. For this reason, a more detailed discussion of the (DE-)9IM or alternative extensions is omitted here. The reader is referred to (Zlatanova 2000) for a comprehensive overview and further references.

Appendix B

GML Application Schema of the MLSEM

This appendix documents the GML application schema of the Multilayered Space-Event Model that has been derived from the ISO 19109 conformant conceptual data model as defined in chapter 4 of this thesis. Each UML package of the conceptual data model has been translated to a separate XML Schema definition according to the normative UML-to-GML mapping rules provided as part of the GML 3.2.1 specification (cf. annex E in Portele 2007 and chapter 4.4.2). The results of this mapping process are presented in the following sections.

The XML Schema definition shown below is the root schema document of the generated GML application schema. It merely includes all further MLSEM schema files but does not specify any further content. An XML instance document shall validate against this root schema in order to be deemed MLSEM conformant.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0" elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="spaceRepresentation.xsd"/>
 <xs:include schemaLocation="externalReference.xsd"/>
 <xs:include schemaLocation="jointStates.xsd"/>
 <xs:include schemaLocation="sourceObject.xsd"/>
 <xs:include schemaLocation="groupsAndSequences.xsd"/>
 <xs:include schemaLocation="route.xsd"/>
 <xs:include schemaLocation="modelLinkage.xsd"/>
 <xs:include schemaLocation="constraints.xsd"/>
</xs:schema>
```

Listing 18: Root XML Schema document of the MLSEM application schema.

B.1 Space Representation Package

The following XML Schema definition has been generated from the MLSEM *Space Representation* package (cf. chapter 4.4.1.1).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0"
 elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="mlsem.xsd"/>
 <xs:include schemaLocation="jointStates.xsd"/>
 <xs:include schemaLocation="sourceObject.xsd"/>
 <xs:include schemaLocation="groupsAndSequences.xsd"/>
 <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../external/gml/3.2.1/gml.xsd"/>

 <!-- AbstractSpaceElement ================================================================ -->
 <xs:complexType name="AbstractSpaceElementType" abstract="true">
   <xs:complexContent>
     <xs:extension base="gml:AbstractFeatureType">
       <xs:sequence>
         <xs:element ref="symbolicId" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="class" minOccurs="0"/>
         <xs:element ref="function" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="usage" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="genericAttribute" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="model" type="SourceObjectMemberType" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="constraint" type="AbstractNavigationConstraintPropertyType" minOccurs="0"
                    maxOccurs="unbounded"/>
       </xs:sequence>
     </xs:extension>
   </xs:complexContent>
```

```
</xs:complexType>
<xs:element name="AbstractSpaceElement" type="AbstractSpaceElementType" abstract="true"
          substitutionGroup="gml:AbstractFeature"/>

<!-- SpaceLayerComplex ============================================================================ -->
<xs:complexType name="SpaceLayerComplexType">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element ref="genericAttribute" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="layer" type="SpaceLayerMemberType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="graph" type="MultilayeredGraphPropertyType" minOccurs="0"/>
        <xs:element name="stateSpace" type="JointStateSpacePropertyType" minOccurs="0"/>

        <xs:element name="machine" type="JointStateMachinePropertyType" minOccurs="0"/>
        <xs:element name="model" type="SourceObjectMemberType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="group" type="SpaceElementGroupMemberType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="interModelGraph" type="InterModelGraphPropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="SpaceLayerComplex" type="SpaceLayerComplexType" substitutionGroup="gml:AbstractFeature"/>

<!-- SpaceLayer ==================================================================================== -->
<xs:complexType name="SpaceLayerType">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element ref="symbolicId" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="type" type="SpaceLayerTypeType"/>
        <xs:element ref="genericAttribute" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="geometry" type="gml:GeometricComplexPropertyType" minOccurs="0"/>
        <xs:element name="topology" type="gml:TopoComplexPropertyType" minOccurs="0"/>
        <xs:element name="spaceCell" type="SpaceCellMemberType" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element name="boundaryCell" type="BoundaryCellMemberType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="graph" type="IntraLayerGraphPropertyType" minOccurs="0"/>
        <xs:element name="subSpace" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="sequence" type="SpaceElementSequenceMemberType" minOccurs="0"
                  maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="SpaceLayer" type="SpaceLayerType" substitutionGroup="gml:AbstractFeature"/>

<xs:complexType name="SpaceLayerMemberType">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureMemberType">
      <xs:sequence minOccurs="0">
        <xs:element ref="SpaceLayer"/>
      </xs:sequence>
      <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:simpleType name="SpaceLayerTypeType">
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="topographic"/>
        <xs:enumeration value="sensor"/>
        <xs:enumeration value="logical"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:string">
```

```
            <xs:pattern value="other: \w{2,}"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:union>
</xs:simpleType>

<!-- SpaceCell ============================================================================== -->
<xs:complexType name="SpaceCellType">
   <xs:complexContent>
      <xs:extension base="AbstractSpaceElementType">
         <xs:sequence>
            <xs:element name="primalSpace" type="SpacePropertyType" minOccurs="0"/>
            <xs:element name="dualSpace" type="StatePropertyType" minOccurs="0"/>
            <xs:element name="subSpace" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="SpaceCell" type="SpaceCellType" substitutionGroup="AbstractSpaceElement"/>

<xs:complexType name="SpaceCellMemberType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureMemberType">
         <xs:sequence minOccurs="0">
            <xs:element ref="SpaceCell"/>
         </xs:sequence>
         <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<!-- BoundaryCell =========================================================================== -->
<xs:complexType name="BoundaryCellType">
   <xs:complexContent>
      <xs:extension base="AbstractSpaceElementType">
         <xs:sequence>
            <xs:element name="primalSpace" type="SpaceBoundaryPropertyType" minOccurs="0"/>
            <xs:element name="dualSpace" type="TransitionPropertyType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="BoundaryCell" type="BoundaryCellType" substitutionGroup="AbstractSpaceElement"/>

<xs:complexType name="BoundaryCellMemberType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureMemberType">
         <xs:sequence minOccurs="0">
            <xs:element ref="BoundaryCell"/>
         </xs:sequence>
         <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<!-- Space ================================================================================== -->
<xs:complexType name="SpaceType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
         <xs:sequence>
            <xs:element name="geometry" type="SolidOrSurfacePropertyType" minOccurs="0"/>
            <xs:element name="topology" type="DirectedTopoSolidOrFacePropertyType" minOccurs="0"/>
            <xs:element name="cell" type="gml:ReferenceType" minOccurs="0"/>
            <xs:element name="boundary" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="dual" type="gml:ReferenceType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

```
<xs:element name="Space" type="SpaceType" substitutionGroup="gml:AbstractFeature"/>

<xs:complexType name="SpacePropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="Space"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<xs:complexType name="SolidOrSurfacePropertyType">
   <xs:sequence minOccurs="0">
      <xs:choice>
         <xs:element ref="gml:AbstractSolid"/>
         <xs:element ref="gml:AbstractSurface"/>
      </xs:choice>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<xs:complexType name="DirectedTopoSolidOrFacePropertyType">
   <xs:sequence minOccurs="0">
      <xs:choice>
         <xs:element ref="gml:TopoSolid"/>
         <xs:element ref="gml:Face"/>
      </xs:choice>
   </xs:sequence>
   <xs:attribute name="orientation" type="gml:SignType" default="+"/>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- SpaceBoundary ============================================================================== -->
<xs:complexType name="SpaceBoundaryType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
         <xs:sequence>
            <xs:element name="geometry" type="SurfaceOrCurvePropertyType" minOccurs="0"/>
            <xs:element name="topology" type="DirectedFaceOrEdgePropertyType" minOccurs="0"/>
            <xs:element name="cell" type="gml:ReferenceType" minOccurs="0"/>
            <xs:element name="coBoundary" type="gml:ReferenceType" minOccurs="0" maxOccurs="2"/>
            <xs:element name="dual" type="gml:ReferenceType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="SpaceBoundary" type="SpaceBoundaryType" substitutionGroup="gml:AbstractFeature"/>

<xs:complexType name="SpaceBoundaryPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="SpaceBoundary"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<xs:complexType name="SurfaceOrCurvePropertyType">
   <xs:sequence minOccurs="0">
      <xs:choice>
         <xs:element ref="gml:AbstractSurface"/>
         <xs:element ref="gml:AbstractCurve"/>
      </xs:choice>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<xs:complexType name="DirectedFaceOrEdgePropertyType">
```

```
   <xs:sequence minOccurs="0">
      <xs:choice>
         <xs:element ref="gml:Face"/>
         <xs:element ref="gml:Edge"/>
      </xs:choice>
   </xs:sequence>
   <xs:attribute name="orientation" type="gml:SignType" default="+"/>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- State ==================================================================================== -->
<xs:complexType name="StateType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
         <xs:sequence>
            <xs:element name="cost" type="GenericAttributeType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="geometry" type="gml:PointPropertyType" minOccurs="0"/>
            <xs:element name="topology" type="gml:DirectedNodePropertyType" minOccurs="0"/>
            <xs:element name="cell" type="gml:ReferenceType" minOccurs="0"/>
            <xs:element name="coBoundary" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="interLayerCoBoundary" type="gml:ReferenceType" minOccurs="0"
                            maxOccurs="unbounded"/>
            <xs:element name="primal" type="gml:ReferenceType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="State" type="StateType" substitutionGroup="gml:AbstractFeature"/>

<xs:complexType name="StatePropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="State"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- Transition =============================================================================== -->
<xs:complexType name="TransitionType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
         <xs:sequence>
            <xs:element name="cost" type="GenericAttributeType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="geometry" type="gml:CurvePropertyType" minOccurs="0"/>
            <xs:element name="topology" type="gml:DirectedEdgePropertyType" minOccurs="0"/>
            <xs:element name="cell" type="gml:ReferenceType" minOccurs="0"/>
            <xs:element name="boundary" type="gml:ReferenceType" minOccurs="0" maxOccurs="2"/>
            <xs:element name="primal" type="gml:ReferenceType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="Transition" type="TransitionType" substitutionGroup="gml:AbstractFeature"/>

<xs:complexType name="TransitionPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="Transition"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- IntraLayerGraph ========================================================================== -->
<xs:complexType name="IntraLayerGraphType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractGMLType">
         <xs:sequence>
            <xs:element name="geometry" type="gml:GeometricComplexPropertyType" minOccurs="0"/>
            <xs:element name="topology" type="gml:TopoComplexPropertyType" minOccurs="0"/>
            <xs:element name="node" type="gml:ReferenceType" minOccurs="1" maxOccurs="unbounded"/>
            <xs:element name="edge" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="IntraLayerGraph" type="IntraLayerGraphType" substitutionGroup="gml:AbstractGML"/>

<xs:complexType name="IntraLayerGraphPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="IntraLayerGraph"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- InterLayerEdge ============================================================================== -->
<xs:complexType name="InterLayerEdgeType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
         <xs:sequence>
            <xs:element name="topoRelation" type="TopoRelationEnum"/>
            <xs:element name="boundary" type="gml:ReferenceType" minOccurs="0" maxOccurs="2"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="InterLayerEdge" type="InterLayerEdgeType" substitutionGroup="gml:AbstractFeature"/>

<xs:complexType name="InterLayerEdgePropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="InterLayerEdge"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<xs:simpleType name="TopoRelationEnum">
   <xs:restriction base="xs:string">
      <xs:enumeration value="containsOrInside"/>
      <xs:enumeration value="coversOrCoveredBy"/>
      <xs:enumeration value="equal"/>
      <xs:enumeration value="overlap"/>
   </xs:restriction>
</xs:simpleType>

<!-- MultilayeredGraph ============================================================================ -->
<xs:complexType name="MultilayeredGraphType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractGMLType">
         <xs:sequence>
            <xs:element name="subGraph" type="gml:ReferenceType" minOccurs="1" maxOccurs="unbounded"/>
            <xs:element name="edge" type="InterLayerEdgePropertyType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="MultilayeredGraph" type="MultilayeredGraphType" substitutionGroup="gml:AbstractGML"/>

<xs:complexType name="MultilayeredGraphPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="MultilayeredGraph"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- common attributes ============================================================================ -->
```

```
  <xs:element name="symbolicId" type="gml:CodeType"/>
  <xs:element name="class" type="gml:CodeType"/>
  <xs:element name="function" type="gml:CodeType"/>
  <xs:element name="usage" type="gml:CodeType"/>
  <xs:element name="name" type="gml:CodeType"/>
  <xs:element name="description" type="xs:string"/>
  <xs:element name="genericAttribute" type="GenericAttributeType"/>

  <!-- GenericAttributeType =========================================================================== -->
  <xs:complexType name="GenericAttributeType">
    <xs:sequence>
      <xs:element ref="name"/>
      <xs:element name="value" type="AnyPropertyType"/>
      <xs:element name="typeName" type="gml:CodeType"/>
      <xs:element name="definition" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AnyPropertyType">
    <xs:sequence minOccurs="0">
      <xs:any processContents="lax"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
  </xs:complexType>
</xs:schema>
```

Listing 19: XML Schema definition of the MLSEM Space Representation package.

B.2 External Reference Package

The following XML Schema definition has been generated from the MLSEM *External Reference* package (cf. chapter 4.4.1.3).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0"
 elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="mlsem.xsd"/>
 <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../external/gml/3.2.1/gml.xsd"/>

 <!-- ExternalReference ===================================================================== -->
 <xs:complexType name="ExternalReferenceType">
    <xs:sequence>
       <xs:element name="informationSystem" type="xs:anyURI" minOccurs="0"/>
       <xs:element name="externalObject" type="ExternalObjectReferencePropertyType" minOccurs="0"/>
    </xs:sequence>
 </xs:complexType>
 <xs:element name="ExternalReference" type="ExternalReferenceType"/>

 <xs:complexType name="ExternalReferencePropertyType">
    <xs:sequence minOccurs="0">
       <xs:element ref="ExternalReference"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 </xs:complexType>

 <!-- ExternalObjectReference =============================================================== -->
 <xs:complexType name="ExternalObjectReferenceType">
    <xs:sequence>
       <xs:element name="name" type="xs:string" minOccurs="0"/>
       <xs:element name="uri" type="xs:anyURI" minOccurs="0"/>
    </xs:sequence>
 </xs:complexType>
 <xs:element name="ExternalObjectReference" type="ExternalObjectReferenceType"/>

 <xs:complexType name="ExternalObjectReferencePropertyType">
    <xs:sequence minOccurs="0">
       <xs:element ref="ExternalObjectReference"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 </xs:complexType>
</xs:schema>
```

Listing 20: XML Schema definition of the MLSEM External Reference package.

B.3 Joint States Package

The following XML Schema definition has been generated from the MLSEM *Joint States* package (cf. chapter 4.4.1.2).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0"
 elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="mlsem.xsd"/>
 <xs:include schemaLocation="spaceRepresentation.xsd"/>
 <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../external/gml/3.2.1/gml.xsd"/>

 <!-- JointStateSpace ======================================================================= -->
 <xs:complexType name="JointStateSpaceType">
    <xs:complexContent>
       <xs:extension base="gml:AbstractGMLType">
          <xs:sequence>
             <xs:element name="state" type="JointStatePropertyType" maxOccurs="unbounded"/>
             <xs:element name="transition" type="JointStateTransitionPropertyType" minOccurs="0"
                         maxOccurs="unbounded"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="JointStateSpace" type="JointStateSpaceType" substitutionGroup="gml:AbstractGML"/>

 <xs:complexType name="JointStateSpacePropertyType">
    <xs:sequence minOccurs="0">
       <xs:element ref="JointStateSpace"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 </xs:complexType>

 <!-- JointState ============================================================================ -->
 <xs:complexType name="JointStateType">
    <xs:complexContent>
       <xs:extension base="gml:AbstractFeatureType">
          <xs:sequence>
             <xs:element name="node" type="gml:ReferenceType" maxOccurs="unbounded"/>
             <xs:element name="uncertaintyRegion" type="gml:GeometricComplexPropertyType"/>
             <xs:element name="coBoundary" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="JointState" type="JointStateType" substitutionGroup="gml:AbstractFeature"/>

 <xs:complexType name="JointStatePropertyType">
    <xs:sequence minOccurs="0">
       <xs:element ref="JointState"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 </xs:complexType>

 <!-- JointStateTransition ==================================================================== -->
 <xs:complexType name="JointStateTransitionType">
    <xs:complexContent>
       <xs:extension base="gml:AbstractFeatureType">
          <xs:sequence>
             <xs:element name="boundary" type="gml:ReferenceType" minOccurs="0" maxOccurs="2"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="JointStateTransition" type="JointStateTransitionType" substitutionGroup="gml:AbstractFeature"/>
```

```
<xs:complexType name="JointStateTransitionPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="JointStateTransition"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- Event ======================================================================== -->
<xs:complexType name="EventType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractGMLType">
         <xs:sequence>
            <xs:element name="type" type="EventTypeType"/>
            <xs:element name="source" type="gml:ReferenceType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="Event" type="EventType" substitutionGroup="gml:AbstractGML"/>

<xs:complexType name="EventPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="Event"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<xs:simpleType name="EventTypeType">
   <xs:union>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="enteredCell"/>
            <xs:enumeration value="leftCell"/>
            <xs:enumeration value="insideCell"/>
            <xs:enumeration value="outsideCell"/>
         </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:pattern value="other: \w{2,}"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:union>
</xs:simpleType>

<!-- TransitionTrigger =================================================================== -->
<xs:complexType name="TransitionTriggerType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractGMLType">
         <xs:sequence>
            <xs:element name="previousState" type="gml:ReferenceType"/>
            <xs:element name="activeState" type="gml:ReferenceType"/>
            <xs:element name="timeStamp" type="gml:TimeInstantPropertyType"/>
            <xs:element name="transition" type="gml:ReferenceType"/>
            <xs:element name="event" type="gml:ReferenceType" maxOccurs="unbounded"/>
            <xs:element name="previousTransitionTrigger" type="gml:ReferenceType" minOccurs="0"/>
            <xs:element name="nextTransitionTrigger" type="gml:ReferenceType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="TransitionTrigger" type="TransitionTriggerType" substitutionGroup="gml:AbstractGML"/>

<xs:complexType name="TransitionTriggerPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="TransitionTrigger"/>
   </xs:sequence>
```

```
      <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
      <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
   </xs:complexType>

   <!-- JointStateMachine =========================================================================== -->
   <xs:complexType name="JointStateMachineType">
      <xs:complexContent>
         <xs:extension base="gml:AbstractGMLType">
            <xs:sequence>
               <xs:element name="stateSpace" type="gml:ReferenceType"/>
               <xs:element name="event" type="EventPropertyType" maxOccurs="unbounded"/>
               <xs:element name="initialState" type="JointStateSetPropertyType"/>
               <xs:element name="finalState" type="JointStateSetPropertyType" minOccurs="0"/>
               <xs:element name="activeState" type="JointStateSetPropertyType" minOccurs="0"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:element name="JointStateMachine" type="JointStateMachineType" substitutionGroup="gml:AbstractGML"/>

   <xs:complexType name="JointStateMachinePropertyType">
      <xs:sequence minOccurs="0">
         <xs:element ref="JointStateMachine"/>
      </xs:sequence>
      <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
      <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
   </xs:complexType>

   <!-- JointStateSet =============================================================================== -->
   <xs:complexType name="JointStateSetType">
      <xs:sequence>
         <xs:element name="state" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>

      </xs:sequence>
   </xs:complexType>
   <xs:element name="JointStateSet" type="JointStateSetType"/>

   <xs:complexType name="JointStateSetPropertyType">
      <xs:sequence minOccurs="0">
         <xs:element ref="JointStateSet"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

Listing 21: XML Schema definition of the MLSEM Joint States package.

B.4 Source Object Package

The following XML Schema definition has been generated from the MLSEM *Source Object* package (cf. chapter 4.4.1.3).

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0"
 elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="mlsem.xsd"/>
 <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../external/gml/3.2.1/gml.xsd"/>

 <!-- SourceObject ============================================================================ -->
 <xs:complexType name="SourceObjectType">
    <xs:complexContent>
       <xs:extension base="gml:AbstractFeatureType">
          <xs:sequence>
             <xs:element name="mediaType" type="xs:anyURI" minOccurs="0"/>
             <xs:choice>
                <xs:element name="content" type="xs:base64Binary"/>
                <xs:element name="reference" type="ExternalReferencePropertyType"/>
             </xs:choice>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="SourceObject" type="SourceObjectType" substitutionGroup="gml:AbstractFeature"/>

 <xs:complexType name="SourceObjectMemberType">
    <xs:complexContent>
       <xs:extension base="gml:AbstractFeatureMemberType">
          <xs:sequence minOccurs="0">
             <xs:element ref="SourceObject"/>
          </xs:sequence>
          <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
</xs:schema>
```

Listing 22: XML Schema definition of the MLSEM Source Object package.

B.5 Groups and Sequences Package

The following XML Schema definition has been generated from the MLSEM *Groups and Sequences* package (cf. chapter 4.4.1.4).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0"
 elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="mlsem.xsd"/>
 <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../external/gml/3.2.1/gml.xsd"/>

 <!-- SpaceElementGroup ================================================================ -->
 <xs:complexType name="SpaceElementGroupType">
    <xs:complexContent>
       <xs:extension base="gml:AbstractFeatureType">
          <xs:sequence>
             <xs:element ref="class" minOccurs="0"/>
             <xs:element ref="function" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element ref="usage" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element ref="genericAttribute" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="element" type="gml:ReferenceType" maxOccurs="unbounded"/>
             <xs:element name="subGroup" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="geometry" type="gml:GeometricComplexPropertyType" minOccurs="0"/>
             <xs:element name="constraint" type="AbstractNavigationConstraintPropertyType" minOccurs="0"
                             maxOccurs="unbounded"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="SpaceElementGroup" type="SpaceElementGroupType" substitutionGroup="gml:AbstractFeature"/>

 <xs:complexType name="SpaceElementGroupMemberType">
    <xs:complexContent>
       <xs:extension base="gml:AbstractFeatureMemberType">
          <xs:sequence minOccurs="0">
             <xs:element ref="SpaceElementGroup"/>
          </xs:sequence>
          <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>

 <!-- SpaceElementSequence ============================================================= -->
 <xs:complexType name="SpaceElementSequenceType">
    <xs:complexContent>
       <xs:extension base="gml:AbstractFeatureType">
          <xs:sequence>
             <xs:element ref="class" minOccurs="0"/>
             <xs:element ref="function" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element ref="usage" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="direction" type="gml:SignType" minOccurs="0"/>
             <xs:element name="mayInvolveOuterSpace" type="xs:boolean" default="true" minOccurs="0"/>
             <xs:element ref="genericAttribute" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="element" type="SequenceElementReferenceType" minOccurs="2"
                             maxOccurs="unbounded"/>
             <xs:element name="geometry" type="gml:GeometricComplexPropertyType" minOccurs="0"/>
             <xs:element name="constraint" type="AbstractNavigationConstraintPropertyType" minOccurs="0"
                             maxOccurs="unbounded"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="SpaceElementSequence" type="SpaceElementSequenceType"
                 substitutionGroup="gml:AbstractFeature"/>

 <xs:complexType name="SequenceElementReferenceType">
    <xs:complexContent>
       <xs:extension base="gml:ReferenceType">
```

```
        <xs:attribute name="sequenceNo" type="xs:positiveInteger" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SpaceElementSequenceMemberType">
    <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureMemberType">
        <xs:sequence minOccurs="0">
          <xs:element ref="SpaceElementSequence"/>
        </xs:sequence>
        <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

Listing 23: XML Schema definition of the MLSEM Groups and Sequences package.

B.6 Route Package

The following XML Schema definition has been generated from the MLSEM *Route* package (cf. chapter 4.4.1.5).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0"
 elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="mlsem.xsd"/>
 <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../external/gml/3.2.1/gml.xsd"/>

 <!-- Route ==================================================================== -->
 <xs:complexType name="RouteType">
   <xs:complexContent>
     <xs:extension base="gml:AbstractFeatureType">
       <xs:sequence>
         <xs:element name="totalTravelDistance" type="gml:LengthType" minOccurs="0"/>
         <xs:element name="totalTravelTime" type="xs:duration" minOccurs="0"/>
         <xs:element name="totalTravelCost" type="gml:MeasureType" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="geometry" type="gml:CurvePropertyType" minOccurs="0"/>
         <xs:element name="topology" type="gml:TopoComplexPropertyType" minOccurs="0"/>
         <xs:element name="segment" type="RouteSegmentMemberType" maxOccurs="unbounded"/>
         <xs:element name="startPoint" type="gml:ReferenceType"/>
         <xs:element name="endPoint" type="gml:ReferenceType"/>
         <xs:element name="viaPoint" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="guidance" type="RouteGuidancePropertyType" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
     </xs:extension>
   </xs:complexContent>
 </xs:complexType>
 <xs:element name="Route" type="RouteType" substitutionGroup="gml:AbstractFeature"/>

 <xs:complexType name="RoutePropertyType">
   <xs:sequence minOccurs="0">
     <xs:element ref="Route"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 </xs:complexType>

 <!-- RoutePoint =============================================================== -->
 <xs:complexType name="RoutePointType">
   <xs:complexContent>
     <xs:extension base="gml:AbstractFeatureType">
       <xs:sequence>
         <xs:element name="type" type="RoutePointTypeType"/>
         <xs:element name="state" type="gml:ReferenceType"/>
         <xs:element name="travelTime" type="xs:duration" minOccurs="0"/>
         <xs:element name="travelCost" type="gml:MeasureType" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="genericAttribute" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="guidance" type="RouteGuidancePropertyType" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
     </xs:extension>
   </xs:complexContent>
 </xs:complexType>
 <xs:element name="RoutePoint" type="RoutePointType" substitutionGroup="gml:AbstractFeature"/>

 <xs:simpleType name="RoutePointTypeType">
   <xs:union>
     <xs:simpleType>
       <xs:restriction base="xs:string">
         <xs:enumeration value="WayPoint"/>
         <xs:enumeration value="Turn"/>
         <xs:enumeration value="DecisionPoint"/>
         <xs:enumeration value="PointOfInterest"/>
       </xs:restriction>
     </xs:simpleType>
     <xs:simpleType>
       <xs:restriction base="xs:string">
```

```
            <xs:pattern value="other: \w{2,}"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:union>
</xs:simpleType>
<xs:complexType name="RoutePointPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="RoutePoint"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- RouteSegment ============================================================================ -->
<xs:complexType name="RouteSegmentType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
         <xs:sequence>
            <xs:element name="travelDistance" type="gml:LengthType" minOccurs="0"/>
            <xs:element name="travelTime" type="xs:duration" minOccurs="0"/>
            <xs:element name="travelCost" type="gml:MeasureType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="geometry" type="gml:CurvePropertyType" minOccurs="0"/>
            <xs:element name="topology" type="gml:TopoComplexPropertyType" minOccurs="0"/>
            <xs:element name="node" type="gml:ReferenceType" maxOccurs="unbounded"/>
            <xs:element name="edge" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="startPoint" type="RoutePointPropertyType"/>
            <xs:element name="endPoint" type="RoutePointPropertyType"/>
            <xs:element name="viaPoint" type="RoutePointPropertyType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="previousRouteSegment" type="gml:ReferenceType" minOccurs="0"/>
            <xs:element name="nextRouteSegment" type="gml:ReferenceType" minOccurs="0"/>
            <xs:element name="guidance" type="RouteGuidancePropertyType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="RouteSegment" type="RouteSegmentType" substitutionGroup="gml:AbstractFeature"/>

<xs:complexType name="RouteSegmentMemberType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureMemberType">
         <xs:sequence minOccurs="0">
            <xs:element ref="RouteSegment"/>
         </xs:sequence>
         <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<!-- AbstractRouteGuidance ==================================================================== -->
<xs:complexType name="AbstractRouteGuidanceType" abstract="true">
   <xs:sequence/>
</xs:complexType>
<xs:element name="AbstractRouteGuidance" type="AbstractRouteGuidanceType" abstract="true"/>
<xs:complexType name="RouteGuidancePropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="AbstractRouteGuidance"/>
   </xs:sequence>
</xs:complexType>

<!-- AbstractRoutePresentation ================================================================== -->
<xs:complexType name="AbstractRoutePresentationType" abstract="true">
   <xs:complexContent>
      <xs:extension base="AbstractRouteGuidanceType">
         <xs:sequence>
            <xs:element name="mediaType" type="xs:string"/>
            <xs:choice>
               <xs:element name="content" type="xs:base64Binary"/>
               <xs:element name="reference" type="ExternalReferencePropertyType"/>
            </xs:choice>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="AbstractRoutePresentation" type="AbstractRoutePresentationType" abstract="true"
                substitutionGroup="AbstractRouteGuidance"/>

<xs:complexType name="RoutePresentationPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="AbstractRoutePresentation"/>
   </xs:sequence>
</xs:complexType>

<!-- AudioDescription ================================================================= -->
<xs:complexType name="AudioDescriptionType">
   <xs:complexContent>
      <xs:extension base="AbstractRoutePresentationType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="AudioDescription" type="AudioDescriptionType" substitutionGroup="AbstractRoutePresentation"/>

<!-- VisualDescription ================================================================= -->
<xs:complexType name="VisualDescriptionType">
   <xs:complexContent>
      <xs:extension base="AbstractRoutePresentationType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="VisualDescription" type="VisualDescriptionType" substitutionGroup="AbstractRoutePresentation"/>

<!-- TextualDescription ================================================================ -->
<xs:complexType name="TextualDescriptionType">
   <xs:complexContent>
      <xs:extension base="AbstractRoutePresentationType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="TextualDescription" type="TextualDescriptionType" substitutionGroup="AbstractRoutePresentation"/>

<!-- RouteInstruction ================================================================== -->
<xs:complexType name="RouteInstructionType">
   <xs:complexContent>
      <xs:extension base="AbstractRouteGuidanceType">
         <xs:sequence>
            <xs:element name="action" type="xs:string"/>
            <xs:element name="distance" type="gml:LengthType"/>
            <xs:element name="duration" type="xs:duration"/>
            <xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="presentation" type="RoutePresentationPropertyType" minOccurs="0"
                            maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="RouteInstruction" type="RouteInstructionType" substitutionGroup="AbstractRouteGuidance"/>

<xs:complexType name="RouteInstructionPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="RouteInstruction"/>
   </xs:sequence>
</xs:complexType>
</xs:schema>
```

Listing 24: XML Schema definition of the MLSEM Route package.

B.7 Model Linkage Package

The following XML Schema definition has been generated from the MLSEM *Model Linkage* package (cf. chapter 4.4.1.6).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0"
 elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="mlsem.xsd"/>
 <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../external/gml/3.2.1/gml.xsd"/>

 <!-- AbstractTransferSpaceElement ================================================================================== -->
 <xs:complexType name="AbstractTransferSpaceElementType" abstract="true">
    <xs:complexContent>
       <xs:extension base="AbstractSpaceElementType">
          <xs:sequence>
             <xs:element name="cost" type="GenericAttributeType" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="AbstractTransferSpaceElement" type="AbstractTransferSpaceElementType" abstract="true"
                substitutionGroup="AbstractSpaceElement"/>

 <!-- TransferState ======================================================================================================= -->
 <xs:complexType name="TransferStateType">
    <xs:complexContent>
       <xs:extension base="AbstractTransferSpaceElementType">
          <xs:sequence>
             <xs:element name="geometry" type="gml:PointPropertyType" minOccurs="0"/>
             <xs:element name="topology" type="gml:DirectedNodePropertyType" minOccurs="0"/>
             <xs:choice>
                <xs:element name="state" type="gml:ReferenceType"/>
                <xs:element name="externalNode" type="ExternalReferenceType"/>
             </xs:choice>
             <xs:element name="coBoundary" type="gml:ReferenceType" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="guidance" type="RouteGuidancePropertyType" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="TransferState" type="TransferStateType" substitutionGroup="AbstractTransferSpaceElement"/>

 <xs:complexType name="TransferStatePropertyType">
    <xs:sequence minOccurs="0">
       <xs:element ref="TransferState"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 </xs:complexType>

 <!-- TransferTransition ================================================================================================== -->
 <xs:complexType name="TransferTransitionType">
    <xs:complexContent>
       <xs:extension base="AbstractTransferSpaceElementType">
          <xs:sequence>
             <xs:element name="geometry" type="gml:CurvePropertyType" minOccurs="0"/>
             <xs:element name="topology" type="gml:DirectedEdgePropertyType" minOccurs="0"/>
             <xs:element name="boundary" type="gml:ReferenceType" minOccurs="2" maxOccurs="2"/>
             <xs:element name="guidance" type="RouteGuidancePropertyType" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="TransferTransition" type="TransferTransitionType"
                substitutionGroup="AbstractTransferSpaceElement"/>

 <xs:complexType name="TransferTransitionPropertyType">
```

```
      <xs:sequence minOccurs="0">
        <xs:element ref="TransferTransition"/>
      </xs:sequence>
      <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
      <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </xs:complexType>

    <!-- InterModelGraph ================================================================= -->
    <xs:complexType name="InterModelGraphType">
      <xs:complexContent>
        <xs:extension base="gml:AbstractGMLType">
          <xs:sequence>
            <xs:element name="geometry" type="gml:GeometricComplexPropertyType" minOccurs="0"/>
            <xs:element name="topology" type="gml:TopoComplexPropertyType" minOccurs="0"/>
            <xs:element name="node" type="TransferStatePropertyType" minOccurs="2" maxOccurs="unbounded"/>
            <xs:element name="edge" type="TransferTransitionPropertyType" minOccurs="1" maxOccurs="unbounded"/>
            <xs:element name="guidance" type="RouteGuidancePropertyType" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:element name="InterModelGraph" type="InterModelGraphType" substitutionGroup="gml:AbstractGML"/>

    <xs:complexType name="InterModelGraphPropertyType">
      <xs:sequence minOccurs="0">
        <xs:element ref="InterModelGraph"/>
      </xs:sequence>
      <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
      <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </xs:complexType>
  </xs:schema>
```

Listing 25: XML Schema definition of the MLSEM Model Linkage package.

B.8 Constraints Package

The following XML Schema definition has been generated from the MLSEM *Constraints* package (cf. chapter 5.3).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.tu-berlin.de/igg/mlsem/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.tu-berlin.de/igg/mlsem/1.0"
 elementFormDefault="qualified" version="1.0.0">
 <xs:include schemaLocation="mlsem.xsd"/>
 <xs:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="../external/gml/3.2.1/gml.xsd"/>

 <!-- AbstractNavigationConstraint ================================================================== -->
 <xs:complexType name="AbstractNavigationConstraintType" abstract="true">
    <xs:complexContent>
       <xs:extension base="gml:AbstractGMLType">
          <xs:sequence>
             <xs:element name="temporalValidity" type="gml:TimePrimitivePropertyType" minOccurs="0"/>
             <xs:element ref="genericAttribute" minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="condition" type="AbstractConstraintConditionPropertyType" minOccurs="0"
                         maxOccurs="unbounded"/>
             <xs:element name="guidance" type="RouteGuidancePropertyType" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="AbstractNavigationConstraint" type="AbstractNavigationConstraintType" abstract="true"
                substitutionGroup="gml:AbstractGML"/>

 <xs:complexType name="AbstractNavigationConstraintPropertyType">
    <xs:sequence minOccurs="0">
       <xs:element ref="AbstractNavigationConstraint"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 </xs:complexType>

 <!-- PassableConstraint ============================================================================ -->
 <xs:complexType name="PassableConstraintType">
    <xs:complexContent>
       <xs:extension base="AbstractNavigationConstraintType">
          <xs:sequence/>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="PassableConstraint" type="PassableConstraintType"
                substitutionGroup="AbstractNavigationConstraint"/>

 <!-- NonPassableConstraint ======================================================================== -->
 <xs:complexType name="NonPassableConstraintType">
    <xs:complexContent>
       <xs:extension base="AbstractNavigationConstraintType">
          <xs:sequence/>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="NonPassableConstraint" type="NonPassableConstraintType"
                substitutionGroup="AbstractNavigationConstraint"/>

 <!-- PermissibleConstraint ========================================================================== -->
 <xs:complexType name="PermissibleConstraintType">
    <xs:complexContent>
       <xs:extension base="AbstractNavigationConstraintType">
          <xs:sequence/>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
 <xs:element name="PermissibleConstraint" type="PermissibleConstraintType"
                substitutionGroup="AbstractNavigationConstraint"/>
```

```
<!-- NonPermissibleConstraint ====================================================================== -->
<xs:complexType name="NonPermissibleConstraintType">
   <xs:complexContent>
      <xs:extension base="AbstractNavigationConstraintType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="NonPermissibleConstraint" type="NonPermissibleConstraintType"
                substitutionGroup="AbstractNavigationConstraint"/>

<!-- AbstractConstraintCondition ==================================================================== -->
<xs:complexType name="AbstractConstraintConditionType" abstract="true">
   <xs:complexContent>
      <xs:extension base="gml:AbstractGMLType">
         <xs:sequence>
            <xs:element name="classifier" type="xs:string" minOccurs="0"/>
            <xs:element ref="description" minOccurs="0"/>
            <xs:element ref="genericAttribute" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="AbstractConstraintCondition" type="AbstractConstraintConditionType" abstract="true"
                substitutionGroup="gml:AbstractGML"/>

<xs:complexType name="AbstractConstraintConditionPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="AbstractConstraintCondition"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- CombinedConstraintCondition =================================================================== -->
<xs:complexType name="CombinedConstraintConditionType">
   <xs:complexContent>
      <xs:extension base="AbstractConstraintConditionType">
         <xs:sequence>
            <xs:element name="operator" type="BooleanOperator"/>
            <xs:element name="operand" type="AbstractConstraintConditionPropertyType" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="CombinedConstraintCondition" type="CombinedConstraintConditionType"
                substitutionGroup="AbstractConstraintCondition"/>

<xs:simpleType name="BooleanOperator">
   <xs:restriction base="xs:string">
      <xs:enumeration value="AND"/>
      <xs:enumeration value="OR"/>
      <xs:enumeration value="NOT"/>
   </xs:restriction>
</xs:simpleType>

<!-- AbstractTemporalCondition ===================================================================== -->
<xs:complexType name="AbstractTemporalConditionType" abstract="true">
   <xs:complexContent>
      <xs:extension base="AbstractConstraintConditionType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="AbstractTemporalCondition" type="AbstractConstraintConditionType" abstract="true"
                substitutionGroup="AbstractConstraintCondition"/>

<!-- TimeInstant ================================================================================ -->
```

```
<xs:complexType name="TimeInstantType">
   <xs:complexContent>
      <xs:extension base="AbstractTemporalConditionType">
         <xs:sequence>
            <xs:element name="time" type="gml:TimeInstantPropertyType"/>
            <xs:element name="period" type="xs:duration" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="TimeInstant" type="TimeInstantType" substitutionGroup="AbstractTemporalCondition"/>

<!-- TimePeriod =============================================================== -->
<xs:complexType name="TimePeriodType">
   <xs:complexContent>
      <xs:extension base="AbstractTemporalConditionType">
         <xs:sequence>
            <xs:element name="extent" type="gml:TimePeriodPropertyType"/>
            <xs:element name="period" type="xs:duration" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="TimePeriod" type="TimePeriodType" substitutionGroup="AbstractTemporalCondition"/>

<!-- AbstractPhysicalCondition ===================================================== -->
<xs:complexType name="AbstractPhysicalConditionType" abstract="true">
   <xs:complexContent>
      <xs:extension base="AbstractConstraintConditionType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="AbstractPhysicalCondition" type="AbstractPhysicalConditionType" abstract="true"
            substitutionGroup="AbstractConstraintCondition"/>

<!-- SpatialProfile ============================================================= -->
<xs:complexType name="SpatialProfileType">
   <xs:complexContent>
      <xs:extension base="AbstractPhysicalConditionType">
         <xs:sequence>
            <xs:element name="profile" type="gml:GeometricPrimitivePropertyType" minOccurs="0"/>
            <xs:element name="width" type="gml:LengthType" minOccurs="0"/>
            <xs:element name="length" type="gml:LengthType" minOccurs="0"/>
            <xs:element name="height" type="gml:LengthType" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="SpatialProfile" type="SpatialProfileType" substitutionGroup="AbstractPhysicalCondition"/>

<!-- PhysicalQuantity ============================================================ -->
<xs:complexType name="PhysicalQuantityType">
   <xs:complexContent>
      <xs:extension base="AbstractPhysicalConditionType">
         <xs:sequence>
            <xs:element ref="name"/>
            <xs:element name="value" type="ValueRangeType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="PhysicalQuantity" type="PhysicalQuantityType" substitutionGroup="AbstractPhysicalCondition"/>

<!-- ModeOfLocomotion ========================================================= -->
<xs:complexType name="ModeOfLocomotionType">
   <xs:complexContent>
      <xs:extension base="AbstractPhysicalConditionType">
         <xs:sequence>
```

```
            <xs:element name="type" type="LocomotionTypeType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="ModeOfLocomotion" type="ModeOfLocomotionType" substitutionGroup="AbstractPhysicalCondition"/>

<xs:simpleType name="LocomotionTypeType">
   <xs:union>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="Walking"/>
            <xs:enumeration value="Driving"/>
            <xs:enumeration value="Flying"/>
            <xs:enumeration value="AssistedWalking"/>
            <xs:enumeration value="WheelChair"/>
            <xs:enumeration value="PoweredWheelChair"/>
            <xs:enumeration value="UnmannedAerialVehicle"/>
            <xs:enumeration value="UnmannedGroundVehicle"/>
         </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:pattern value="other: \w{2,}"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:union>
</xs:simpleType>

<!-- Material ================================================================================== -->
<xs:complexType name="MaterialType">
   <xs:complexContent>
      <xs:extension base="AbstractPhysicalConditionType">
         <xs:sequence>
            <xs:element name="surface" type="SurfaceTypeType"/>
            <xs:element name="material" type="gml:CodeType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="Material" type="MaterialType" substitutionGroup="AbstractPhysicalCondition"/>

<xs:simpleType name="SurfaceTypeType">
   <xs:union>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="Wall"/>
            <xs:enumeration value="Floor"/>
            <xs:enumeration value="Ceiling"/>
            <xs:enumeration value="Ramp"/>
         </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:pattern value="other: \w{2,}"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:union>
</xs:simpleType>

<!-- SpatialManeuver ============================================================================ -->
<xs:complexType name="SpatialManeuverType">
   <xs:complexContent>
      <xs:extension base="AbstractPhysicalConditionType">
         <xs:sequence>
            <xs:element name="type" type="ManeuverTypeType"/>
            <xs:element name="geometry" type="gml:GeometricPrimitivePropertyType" minOccurs="0"/>
            <xs:element name="width" type="ValueRangeType" minOccurs="0"/>
            <xs:element name="length" type="ValueRangeType" minOccurs="0"/>
            <xs:element name="height" type="ValueRangeType" minOccurs="0"/>
            <xs:element name="verticalGradient" type="ValueRangeType" minOccurs="0"/>
            <xs:element name="horizontalTurnRadius" type="ValueRangeType" minOccurs="0"/>
            <xs:element name="numberOfSteps" type="xs:positiveInteger" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="SpatialManeuver" type="SpatialManeuverType" substitutionGroup="AbstractPhysicalCondition"/>

<xs:simpleType name="ManeuverTypeType">
   <xs:union>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:enumeration value="VerticalDisplacement"/>
            <xs:enumeration value="HorizontalDisplacement"/>
            <xs:enumeration value="VerticalAscent"/>
            <xs:enumeration value="VerticalDescent"/>
            <xs:enumeration value="Turn"/>
            <xs:enumeration value="Gap"/>
         </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:pattern value="other: \w{2,}"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:union>
</xs:simpleType>

<!-- ValueRange ==================================================================================== -->
<xs:complexType name="ValueRangeType">
   <xs:sequence>
      <xs:element name="uom" type="gml:UomIdentifier"/>
      <xs:element name="lowerLimit" type="xs:double" minOccurs="0"/>
      <xs:element name="upperLimit" type="xs:double" minOccurs="0"/>
      <xs:element name="includeLowerLimit" type="xs:boolean" default="true" minOccurs="0"/>
      <xs:element name="includeUpperLimit" type="xs:boolean" default="true" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
<xs:element name="valueRange" type="ValueRangeType"/>

<!-- AbstractLogicalCondition ========================================================================== -->
<xs:complexType name="AbstractLogicalConditionType" abstract="true">
   <xs:complexContent>
      <xs:extension base="AbstractConstraintConditionType">
         <xs:sequence/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="AbstractLogicalCondition" type="AbstractLogicalConditionType" abstract="true"
                substitutionGroup="AbstractConstraintCondition"/>

<!-- NavigationScenario ============================================================================= -->
<xs:complexType name="NavigationScenarioType">
   <xs:complexContent>
      <xs:extension base="AbstractLogicalConditionType">
         <xs:sequence>
            <xs:element name="type" type="ScenarioTypeType"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="NavigationScenario" type="NavigationScenarioType" substitutionGroup="AbstractLogicalCondition"/>

<xs:simpleType name="ScenarioTypeType">
   <xs:union>
      <xs:simpleType>
         <xs:restriction base="xs:string">
```

```
            <xs:enumeration value="Normal"/>
            <xs:enumeration value="PeakTime"/>
            <xs:enumeration value="OffPeakTime"/>
            <xs:enumeration value="LowSecurityLevel"/>
            <xs:enumeration value="MediumSecurityLevel"/>
            <xs:enumeration value="HighSecurityLevel"/>
            <xs:enumeration value="Evacuation"/>
         </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:pattern value="other: \w{2,}"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:union>
</xs:simpleType>

<!-- UserGroup ============================================================================ -->
<xs:complexType name="UserGroupType">
   <xs:complexContent>
      <xs:extension base="AbstractLogicalConditionType">
         <xs:sequence>
            <xs:element ref="name"/>
            <xs:element name="subGroup" type="UserGroupPropertyType" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="UserGroup" type="UserGroupType" substitutionGroup="AbstractLogicalCondition"/>

<xs:complexType name="UserGroupPropertyType">
   <xs:sequence minOccurs="0">
      <xs:element ref="UserGroup"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
   <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
</xs:complexType>

<!-- UserState ============================================================================== -->
<xs:complexType name="UserStateType">
   <xs:complexContent>
      <xs:extension base="AbstractLogicalConditionType">
         <xs:sequence>
            <xs:element name="value" type="xs:boolean" minOccurs="0"/>
            <xs:element name="state" type="xs:string" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="UserState" type="UserStateType" substitutionGroup="AbstractLogicalCondition"/>

<!-- ObjectState =========================================================================== -->
<xs:complexType name="ObjectStateType">
   <xs:complexContent>
      <xs:extension base="AbstractLogicalConditionType">
         <xs:sequence>
            <xs:element name="id" type="ExternalReferenceType"/>
            <xs:element name="value" type="xs:boolean" minOccurs="0"/>
            <xs:element name="state" type="xs:string" minOccurs="0"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<xs:element name="ObjectState" type="ObjectStateType" substitutionGroup="AbstractLogicalCondition"/>

<!-- AccessControl ========================================================================= -->
<xs:complexType name="AccessControlType">
   <xs:complexContent>
      <xs:extension base="AbstractLogicalConditionType">
```

```
        <xs:sequence>
          <xs:element name="credential" type="CredentialTypeType" minOccurs="0"/>
          <xs:element name="objectId" type="xs:string" minOccurs="0"/>
          <xs:element name="permission" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="AccessControl" type="AccessControlType" substitutionGroup="AbstractLogicalCondition"/>

  <xs:simpleType name="CredentialTypeType">
    <xs:union>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Key"/>
          <xs:enumeration value="PIN"/>
          <xs:enumeration value="Biometric"/>
          <xs:enumeration value="Screening"/>
          <xs:enumeration value="PersonalSearch"/>
          <xs:enumeration value="ManualCheck"/>
          <xs:enumeration value="Ticket"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="other: \w{2,}"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
</xs:schema>
```

Listing 26: XML Schema definition of the MLSEM Constraints package.

Appendix C

SQL Definition of the MLSEM Database Schema

This appendix provides the formal definition of the developed MLSEM database schema expressed in statements of the SQL *Data Definition Language* (DDL). The DDL statements are presented in the following listing 27 and have been automatically derived from the UML diagram of the database schema (cf. figure 263 in chapter 7.2.3) using the UML software tool *Enterprise Architect*.[88] The SQL script can be directly executed (and has been tested) on an *Oracle Database 11g Release 2* with *Oracle Spatial and Graph* extension in order to create an MLSEM database instance. A second SQL script is provided in listing 28 which allows for the creation of 3-dimensional indexes on the spatial table columns. Most of the functions and operations on spatial objects offered by the Oracle Spatial database require spatial indexes to be enabled. This second script can also be easily adapted for 2-dimensional spatial indexes in case a 2-dimensional instance of the MLSEM shall be stored and managed.

```
--   --------------------------------------------------
--   Generated by Enterprise Architect Version 9.3.934
--   Created On : Monday, 04 March, 2013
--   DBMS       : Oracle
--   --------------------------------------------------

--  Drop Tables, Stored Procedures and Views
DROP TABLE BOUNDARY_CELL CASCADE CONSTRAINTS
;
DROP TRIGGER TRG_GENERIC_ATTRIBUTE_OID
;
DROP SEQUENCE SEQ_GENERIC_ATTRIBUTE_OID
;
DROP TABLE GENERIC_ATTRIBUTE CASCADE CONSTRAINTS
;
DROP TABLE INTER_LAYER_EDGE CASCADE CONSTRAINTS
;
DROP TRIGGER TRG_JOINT_STATE_OID
;
DROP SEQUENCE SEQ_JOINT_STATE_OID
;
DROP TABLE JOINT_STATE CASCADE CONSTRAINTS
;
DROP TABLE JOINT_STATE_CLIQUE CASCADE CONSTRAINTS
;
DROP TABLE JOINT_STATE_TRANSITION CASCADE CONSTRAINTS
;
DROP TRIGGER TRG_SOURCE_OBJECT_OID
;
DROP SEQUENCE SEQ_SOURCE_OBJECT_OID
;
DROP TABLE SOURCE_OBJECT CASCADE CONSTRAINTS
;
DROP TABLE SPACE_CELL CASCADE CONSTRAINTS
;
DROP TRIGGER TRG_SPACE_ELEMENT_OID
;
DROP SEQUENCE SEQ_SPACE_ELEMENT_OID
;
DROP TABLE SPACE_ELEMENT CASCADE CONSTRAINTS
;
DROP TRIGGER TRG_SPACE_LAYER_OID
;
DROP SEQUENCE SEQ_SPACE_LAYER_OID
```

[88] *Enterprise Architect* is a commercial UML modelling tool developed by the company Sparx Systems Pty Ltd. See http://www.sparxsystems.com.au/ for more information.

```
;
DROP TABLE SPACE_LAYER CASCADE CONSTRAINTS
;

--  Create Tables
CREATE TABLE BOUNDARY_CELL
(
 OID             NUMBER NOT NULL,
 LAYER_ID        NUMBER NOT NULL,
 SPACE_CELL_ID1  NUMBER NOT NULL,
 SPACE_CELL_ID2  NUMBER NOT NULL,
 SPACE_BOUNDARY  SDO_GEOMETRY,
 TRANSITION      SDO_GEOMETRY
)
;

CREATE TABLE GENERIC_ATTRIBUTE
(
 OID         NUMBER NOT NULL,
 ELEMENT_ID  NUMBER,
 LAYER_ID    NUMBER,
 ROOT_ID     NUMBER NOT NULL,
 PARENT_ID   NUMBER,
 NAME        VARCHAR(256) NOT NULL,
 STR_VAL     VARCHAR2(4000),
 INT_VAL     LONG,
 DOUBLE_VAL  NUMBER,
 URI_VAL     VARCHAR2(4000),
 DATE_VAL    DATE,
 GEOM_VAL    SDO_GEOMETRY,
 BLOB_VAL    BLOB,
 XML_VAL     XMLTYPE
)
;

CREATE TABLE INTER_LAYER_EDGE
(
 STATE_ID1      NUMBER NOT NULL,
 STATE_ID2      NUMBER NOT NULL,
 TOPO_RELATION  VARCHAR2(100) NOT NULL
)
;

CREATE TABLE JOINT_STATE
(
 OID                 NUMBER NOT NULL,
 UNCERTAINTY_REGION  SDO_GEOMETRY
)
;

CREATE TABLE JOINT_STATE_CLIQUE
(
 JOINT_STATE_ID  NUMBER NOT NULL,
 SPACE_CELL_ID   NUMBER NOT NULL
)
;

CREATE TABLE JOINT_STATE_TRANSITION
(
 STATE_ID1  NUMBER NOT NULL,
 STATE_ID2  NUMBER NOT NULL
)
;

CREATE TABLE SOURCE_OBJECT
(
 OID            NUMBER NOT NULL,
 ELEMENT_ID     NUMBER,
 LAYER_ID       NUMBER,
 NAME           VARCHAR2(1000),
 DESCRIPTION    VARCHAR2(1000),
 MEDIA_TYPE     VARCHAR2(100) NOT NULL,
 CONTENT        BLOB,
 XML_CONTENT    XMLTYPE,
 INFOSYS        VARCHAR2(4000),
 EXTERNAL_NAME  VARCHAR2(4000),
 URI            VARCHAR2(4000)
)
;

CREATE TABLE SPACE_CELL
(
 OID       NUMBER NOT NULL,
 LAYER_ID  NUMBER NOT NULL,
 SPACE     SDO_GEOMETRY,
 STATE     SDO_GEOMETRY
)
;

CREATE TABLE SPACE_ELEMENT
(
 OID          NUMBER NOT NULL,
 SYMBOLIC_ID  VARCHAR2(4000),
 CLASS        VARCHAR2(256),
 FUNCTION     VARCHAR2(1000),
 USAGE        VARCHAR2(1000)
)
;

CREATE TABLE SPACE_LAYER
(
 OID        NUMBER NOT NULL,
 TYPE       VARCHAR2(256) NOT NULL,
 ROOT_ID    NUMBER NOT NULL,
 PARENT_ID  NUMBER
)
;

--  Create Primary Key Constraints
ALTER TABLE BOUNDARY_CELL ADD CONSTRAINT PK_BOUNDARY_CELL
 PRIMARY KEY (OID)
;

ALTER TABLE GENERIC_ATTRIBUTE ADD CONSTRAINT PK_GENERIC_ATTRIBUTE
 PRIMARY KEY (OID)
;

ALTER TABLE INTER_LAYER_EDGE ADD CONSTRAINT PK_LAYER_EDGE
 PRIMARY KEY (STATE_ID1, STATE_ID2)
;

ALTER TABLE JOINT_STATE ADD CONSTRAINT PK_JOINT_STATE
 PRIMARY KEY (OID)
;

ALTER TABLE JOINT_STATE_CLIQUE ADD CONSTRAINT PK_JOINT_STATE_CLIQUE
 PRIMARY KEY (JOINT_STATE_ID, SPACE_CELL_ID)
;

ALTER TABLE JOINT_STATE_TRANSITION ADD CONSTRAINT PK_STATE_TRANSITION
 PRIMARY KEY (STATE_ID1, STATE_ID2)
;

ALTER TABLE SOURCE_OBJECT ADD CONSTRAINT PK_SOURCE_OBJECT
 PRIMARY KEY (OID)
;

ALTER TABLE SPACE_CELL ADD CONSTRAINT PK_SPACE_CELL
 PRIMARY KEY (OID)
;
```

```
ALTER TABLE SPACE_ELEMENT ADD CONSTRAINT PK_SPACE_ELEMENT
 PRIMARY KEY (OID)
;

ALTER TABLE SPACE_LAYER ADD CONSTRAINT PK_SPACE_LAYER
 PRIMARY KEY (OID)
;

--  Create Foreign Key Constraints
ALTER TABLE BOUNDARY_CELL ADD CONSTRAINT FK_BC_SPACE_ELEMENT
 FOREIGN KEY (OID) REFERENCES SPACE_ELEMENT (OID)
;

ALTER TABLE BOUNDARY_CELL ADD CONSTRAINT FK_BC_SPACE_LAYER
 FOREIGN KEY (LAYER_ID) REFERENCES SPACE_LAYER (OID)
;

ALTER TABLE BOUNDARY_CELL ADD CONSTRAINT FK_BC_SPACE_CELL_1
 FOREIGN KEY (SPACE_CELL_ID1) REFERENCES SPACE_CELL (OID)
;

ALTER TABLE BOUNDARY_CELL ADD CONSTRAINT FK_BC_SPACE_CELL_2
 FOREIGN KEY (SPACE_CELL_ID2) REFERENCES SPACE_CELL (OID)
;

ALTER TABLE GENERIC_ATTRIBUTE ADD CONSTRAINT FK_GA_SPACE_ELEMENT
 FOREIGN KEY (ELEMENT_ID) REFERENCES SPACE_ELEMENT (OID)
;

ALTER TABLE GENERIC_ATTRIBUTE ADD CONSTRAINT FK_GA_SPACE_LAYER
 FOREIGN KEY (LAYER_ID) REFERENCES SPACE_LAYER (OID)
;

ALTER TABLE GENERIC_ATTRIBUTE ADD CONSTRAINT FK_GA_ROOT
 FOREIGN KEY (ROOT_ID) REFERENCES GENERIC_ATTRIBUTE (OID)
;

ALTER TABLE GENERIC_ATTRIBUTE ADD CONSTRAINT FK_GA_PARENT
 FOREIGN KEY (PARENT_ID) REFERENCES GENERIC_ATTRIBUTE (OID)
;

ALTER TABLE INTER_LAYER_EDGE ADD CONSTRAINT FK_ILE_SPACE_CELL_1
 FOREIGN KEY (STATE_ID1) REFERENCES SPACE_CELL (OID)
;

ALTER TABLE INTER_LAYER_EDGE ADD CONSTRAINT FK_ILE_SPACE_CELL_2
 FOREIGN KEY (STATE_ID2) REFERENCES SPACE_CELL (OID)
;

ALTER TABLE JOINT_STATE_CLIQUE ADD CONSTRAINT FK_JSC_SPACE_CELL
 FOREIGN KEY (SPACE_CELL_ID) REFERENCES SPACE_CELL (OID)
;

ALTER TABLE JOINT_STATE_CLIQUE ADD CONSTRAINT FK_JSC_JOINT_STATE
 FOREIGN KEY (JOINT_STATE_ID) REFERENCES JOINT_STATE (OID)
;

ALTER TABLE JOINT_STATE_TRANSITION ADD CONSTRAINT FK_JST_JOINT_STATE_1
 FOREIGN KEY (STATE_ID1) REFERENCES JOINT_STATE (OID)
;

ALTER TABLE JOINT_STATE_TRANSITION ADD CONSTRAINT FK_JST_JOINT_STATE_2
 FOREIGN KEY (STATE_ID2) REFERENCES JOINT_STATE (OID)
;

ALTER TABLE SOURCE_OBJECT ADD CONSTRAINT FK_SO_SPACE_ELEMENT
 FOREIGN KEY (ELEMENT_ID) REFERENCES SPACE_ELEMENT (OID)
;
```

```
ALTER TABLE SOURCE_OBJECT ADD CONSTRAINT FK_SO_SPACE_LAYER
 FOREIGN KEY (LAYER_ID) REFERENCES SPACE_LAYER (OID)
;

ALTER TABLE SPACE_CELL ADD CONSTRAINT FK_SC_SPACE_ELEMENT
 FOREIGN KEY (OID) REFERENCES SPACE_ELEMENT (OID)
;

ALTER TABLE SPACE_CELL ADD CONSTRAINT FK_SC_SPACE_LAYER
 FOREIGN KEY (LAYER_ID) REFERENCES SPACE_LAYER (OID)
;

ALTER TABLE SPACE_LAYER ADD CONSTRAINT FK_SL_ROOT
 FOREIGN KEY (ROOT_ID) REFERENCES SPACE_LAYER (OID)
;

ALTER TABLE SPACE_LAYER ADD CONSTRAINT FK_SL_PARENT
 FOREIGN KEY (PARENT_ID) REFERENCES SPACE_LAYER (OID)
;

--  Create Triggers
CREATE SEQUENCE SEQ_GENERIC_ATTRIBUTE_OID
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 MINVALUE 1
 NOCYCLE
 NOCACHE
 NOORDER
;

CREATE OR REPLACE TRIGGER TRG_GENERIC_ATTRIBUTE_OID
 BEFORE INSERT
 ON GENERIC_ATTRIBUTE
 FOR EACH ROW
 BEGIN
   SELECT SEQ_GENERIC_ATTRIBUTE_OID.NEXTVAL
   INTO :NEW.OID
   FROM DUAL;
 END;
/

CREATE SEQUENCE SEQ_JOINT_STATE_OID
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 MINVALUE 1
 NOCYCLE
 NOCACHE
 NOORDER
;

CREATE OR REPLACE TRIGGER TRG_JOINT_STATE_OID
 BEFORE INSERT
 ON JOINT_STATE
 FOR EACH ROW
 BEGIN
   SELECT SEQ_JOINT_STATE_OID.NEXTVAL
   INTO :NEW.OID
   FROM DUAL;
 END;
/

CREATE SEQUENCE SEQ_SOURCE_OBJECT_OID
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 MINVALUE 1
 NOCYCLE
 NOCACHE
 NOORDER
;

CREATE OR REPLACE TRIGGER TRG_SOURCE_OBJECT_OID
 BEFORE INSERT
 ON SOURCE_OBJECT
 FOR EACH ROW
 BEGIN
   SELECT SEQ_SOURCE_OBJECT_OID.NEXTVAL
   INTO :NEW.OID
   FROM DUAL;
 END;
/

CREATE SEQUENCE SEQ_SPACE_ELEMENT_OID
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 MINVALUE 1
 NOCYCLE
 NOCACHE
 NOORDER
;

CREATE OR REPLACE TRIGGER TRG_SPACE_ELEMENT_OID
 BEFORE INSERT
 ON SPACE_ELEMENT
 FOR EACH ROW
 BEGIN
   SELECT SEQ_SPACE_ELEMENT_OID.NEXTVAL
   INTO :NEW.OID
   FROM DUAL;
 END;
/

CREATE SEQUENCE SEQ_SPACE_LAYER_OID
 INCREMENT BY 1
 START WITH 1
 NOMAXVALUE
 MINVALUE 1
 NOCYCLE
 NOCACHE
 NOORDER
;

CREATE OR REPLACE TRIGGER TRG_SPACE_LAYER_OID
 BEFORE INSERT
 ON SPACE_LAYER
 FOR EACH ROW
 BEGIN
   SELECT SEQ_SPACE_LAYER_OID.NEXTVAL
   INTO :NEW.OID
   FROM DUAL;
 END;
/
```

Listing 27: The MLSEM database schema formally expressed in SQL.

As mentioned above, the following SQL script creates 3-dimensional indexes on the spatial columns of the MLSEM database tables. In Oracle Spatial, this first requires metadata about the spatial objects to be stored. For each spatial column, the metadata has to be entered in the table `USER_SDO_GEOM_METADATA` and includes information about the lower and upper boundary and the tolerance for each dimension (axis) as well as the spatial reference system associated with the spatial objects (given as `SRID` identifier). In the below listing, default bounds and tolerances are used which need to be adapted for a specific database instance. Likewise, the `SRID` is set to `NULL` which has to be changed if a specific reference system shall be used. The spatial indexes are then created with the parameter `'SDO_INDX_DIMS=3'` which is mandatory in order to be able to apply the 3-dimensional spatial functions and operations offered by Oracle Spatial to the spatial objects. The script can also be used for the creation of 2-dimensional spatial indexes by simply omitting the third dimension from the spatial metadata and by deleting the parameter from the `CREATE INDEX` statements.

```
--  Delete Spatial Metadata and drop Spatial Indexes
DELETE FROM USER_SDO_GEOM_METADATA WHERE TABLE_NAME='BOUNDARY_CELL' AND
COLUMN_NAME='SPACE_BOUNDARY'
;
DELETE FROM USER_SDO_GEOM_METADATA WHERE TABLE_NAME='BOUNDARY_CELL' AND
COLUMN_NAME='TRANSITION'
;
DELETE FROM USER_SDO_GEOM_METADATA WHERE TABLE_NAME='GENERIC_ATTRIBUTE' AND
COLUMN_NAME='GEOM_VAL'
;
DELETE FROM USER_SDO_GEOM_METADATA WHERE TABLE_NAME='JOINT_STATE' AND
COLUMN_NAME='UNCERTAINTY_REGION'
;
DELETE FROM USER_SDO_GEOM_METADATA WHERE TABLE_NAME='SPACE_CELL' AND COLUMN_NAME='SPACE'
;
DELETE FROM USER_SDO_GEOM_METADATA WHERE TABLE_NAME='SPACE_CELL' AND COLUMN_NAME='STATE'
;
DROP INDEX BC_SPACE_BOUNDARY_SPX
;
DROP INDEX BC_TRANSITION_SPX
;
DROP INDEX GA_GEOM_VAL_SPX
;
DROP INDEX JS_UNCERTAINTY_REGION_SPX
;
DROP INDEX SC_SPACE_SPX
;
DROP INDEX SC_STATE_SPX
;

--  Create Spatial Metadata
INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
 VALUES ('BOUNDARY_CELL', 'SPACE_BOUNDARY',
  MDSYS.SDO_DIM_ARRAY
  (
    MDSYS.SDO_DIM_ELEMENT('X', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Y', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Z', -1000, 10000, 0.0005)
  ),
  NULL -- SRID
 )
;

INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
 VALUES ('BOUNDARY_CELL', 'TRANSITION',
  MDSYS.SDO_DIM_ARRAY
  (
    MDSYS.SDO_DIM_ELEMENT('X', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Y', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Z', -1000, 10000, 0.0005)
  ),
  NULL -- SRID
 )
;

INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
 VALUES ('GENERIC_ATTRIBUTE', 'GEOM_VAL',
  MDSYS.SDO_DIM_ARRAY
  (
    MDSYS.SDO_DIM_ELEMENT('X', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Y', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Z', -1000, 10000, 0.0005)
  ),
  NULL -- SRID
 )
;
```

```
INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
 VALUES ('JOINT_STATE', 'UNCERTAINTY_REGION',
  MDSYS.SDO_DIM_ARRAY
  (
    MDSYS.SDO_DIM_ELEMENT('X', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Y', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Z', -1000, 10000, 0.0005)
  ),
  NULL -- SRID
 )
;

INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
 VALUES ('SPACE_CELL', 'SPACE',
  MDSYS.SDO_DIM_ARRAY
  (
    MDSYS.SDO_DIM_ELEMENT('X', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Y', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Z', -1000, 10000, 0.0005)
  ),
  NULL -- SRID
 )
;

INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
 VALUES ('SPACE_CELL', 'STATE',
  MDSYS.SDO_DIM_ARRAY
  (
    MDSYS.SDO_DIM_ELEMENT('X', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Y', 0.000, 10000000.000, 0.0005),
    MDSYS.SDO_DIM_ELEMENT('Z', -1000, 10000, 0.0005)
  ),
  NULL -- SRID
 )
;

--  Create Spatial Indexes
CREATE INDEX BC_SPACE_BOUNDARY_SPX ON BOUNDARY_CELL (SPACE_BOUNDARY)
 INDEXTYPE IS MDSYS.SPATIAL_INDEX
 PARAMETERS ('sdo_indx_dims=3') -- for 3D capabilities
;

CREATE INDEX BC_TRANSITION_SPX ON BOUNDARY_CELL (TRANSITION)
 INDEXTYPE IS MDSYS.SPATIAL_INDEX
 PARAMETERS ('sdo_indx_dims=3') -- for 3D capabilities
;

CREATE INDEX GA_GEOM_VAL_SPX ON GENERIC_ATTRIBUTE (GEOM_VAL)
 INDEXTYPE IS MDSYS.SPATIAL_INDEX
 PARAMETERS ('sdo_indx_dims=3') -- for 3D capabilities
;

CREATE INDEX JS_UNCERTAINTY_REGION_SPX ON JOINT_STATE (UNCERTAINTY_REGION)
 INDEXTYPE IS MDSYS.SPATIAL_INDEX
 PARAMETERS ('sdo_indx_dims=3') -- for 3D capabilities
;

CREATE INDEX SC_SPACE_SPX ON SPACE_CELL (SPACE)
 INDEXTYPE IS MDSYS.SPATIAL_INDEX
 PARAMETERS ('sdo_indx_dims=3') -- for 3D capabilities
;

CREATE INDEX SC_STATE_SPX ON SPACE_CELL (STATE)
 INDEXTYPE IS MDSYS.SPATIAL_INDEX
 PARAMETERS ('sdo_indx_dims=3') -- for 3D capabilities
;
```

Listing 28: SQL statements for the creation of 3-dimensional indexes on spatial table columns.

Appendix D

Previous Conceptual Data Model of the MLSEM

Figure 287 presents the conceptual UML data model of the MLSEM as proposed in the previous publications on the MLSEM (cf. Becker et al. 2009b and Nagel et al. 2010).

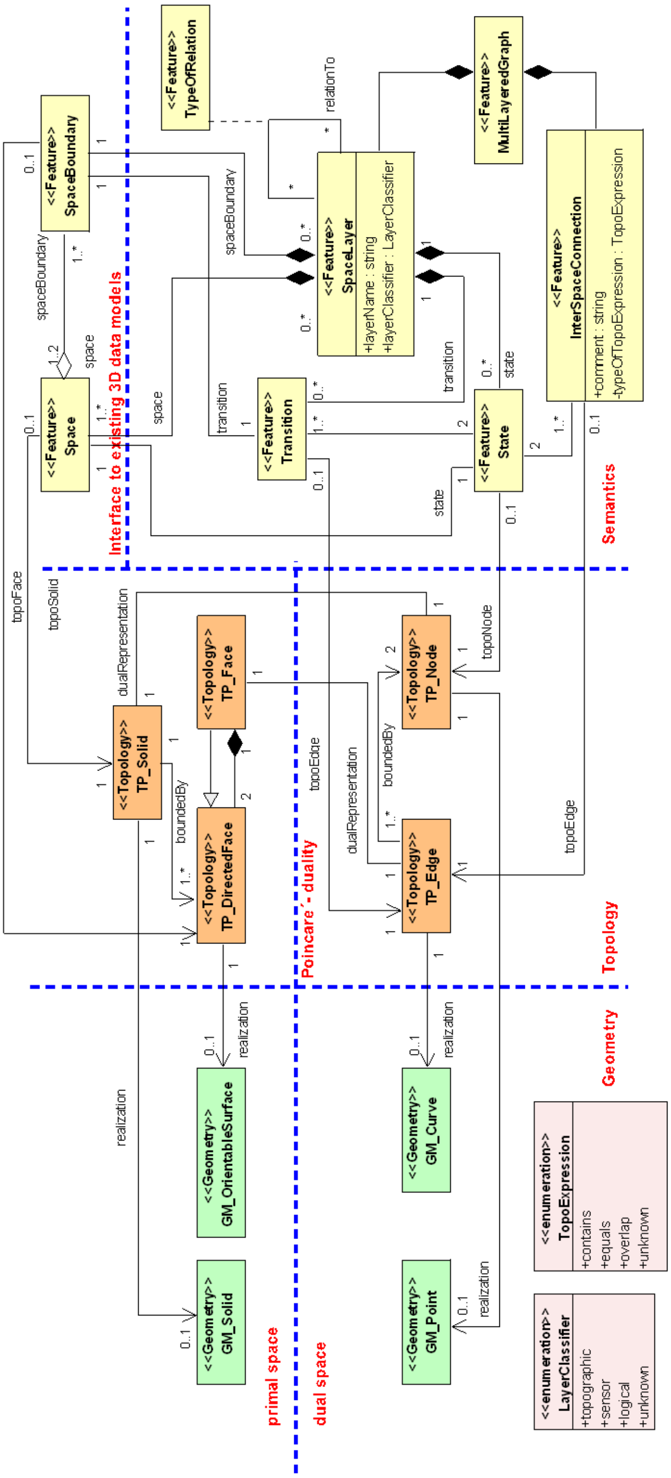


Figure 287: The conceptual UML data model of the former MLSEM as proposed in (Becker et al. 2009b) and (Nagel et al. 2010).

A discussion of the conceptual model elements in figure 287 is beyond the scope of this thesis and the reader is referred to (Becker et al. 2009b) and (Nagel et al. 2010). Nevertheless, the data model nicely documents the substantial rework and extension of the MLSEM data model as carried out in this thesis (cf. chapter 4.4). For example, and as mentioned in chapter 4.4, the previous data model does not foresee the geometric-topological representation of space cells, boundary cells, and space layers in both two and three dimensions but is rather restricted to the latter case. Moreover, the MLSEM has been augmented in this thesis, amongst others, with essential conceptual entities for the modelling of space layer complexes, joint states and their transitions, groups and sequences of space elements, links to original data sources such as IFC or CityGML models, navigation routes and guidance information, navigation constraints, and the connection to complementary navigation graphs.

Appendix E

Initial Conceptual Data Model for Navigation Constraints

Figure 288 shows the draft conceptualization of navigation constraints as proposed by (Brown et al. 2012) in the context of the MLSEM. The reader is referred to this publication for a presentation and discussion of the model and its conceptual entities.

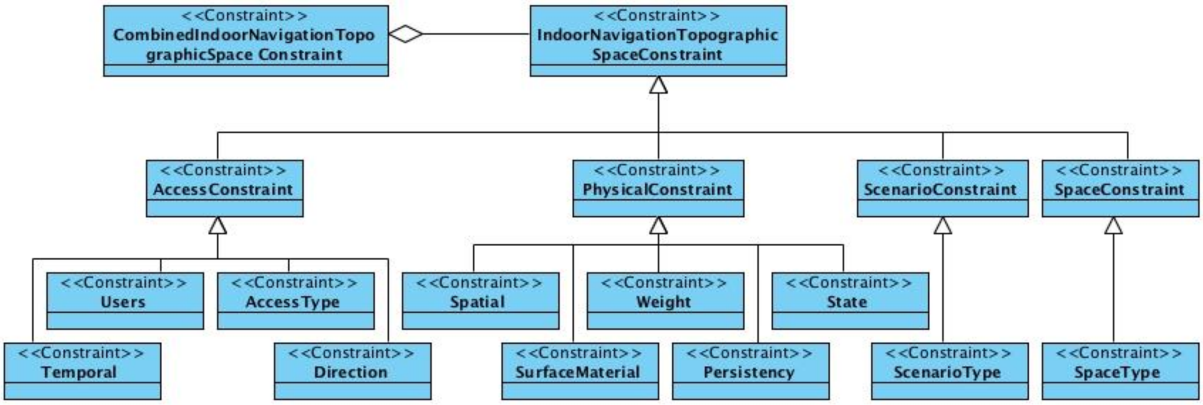


Figure 288: Initial draft for a navigation constraint model for the MLSEM as proposed in (Brown et al. 2012).

The navigation constraint model proposed by (Brown et al. 2012) has been substantially reworked and refined in this thesis (cf. chapter 5). For example, the initial draft does not distinguish between *navigation constraints* (as well as between different types thereof such as *can* and *may* constraints) and *constraint conditions* that guard the application of navigation constraints. In fact, the data model depicted in figure 288 only deals with the definition and taxonomy of constraint conditions but neglects conceptual elements for describing navigation constraints. Moreover, thematic and spatial characteristics of the constraint conditions are neither presented and the proposed model lacks rules and algorithms for the evaluation of navigation constraints in the context of a given navigation user.

Bibliography

Abowd, G. D.; Dey, A. K.; Brown, P. J.; Davies, N.; Smith, M.; Steggles, P. (1999): Towards a Better Understanding of Context and Context-Awareness. In Gellersen, H.-W. (Ed.): Handheld and ubiquitous computing. Proceedings. Berlin: Springer (Lecture Notes in Computer Science, 1707), pp. 304-307.Available online at http://dl.acm.org/citation.cfm?id=647985.743843.

Afyouni, I.; Ray, C.; Claramunt, C. (2010): A fine-grained context-dependent model for indoor spaces. In Winter, S., Jensen, C. S., Li, K.-J. (Eds.): Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. New York, NY, USA: ACM, pp. 33-38.

Afyouni, I.; Ray, C.; Claramunt, C. (2012): Spatial Models for Indoor & Context-Aware Navigation Systems: A Survey. In Duckham, M., Sack, J.-R., Worboys, M. (Eds.): Journal of spatial information science, vol. 4, pp. 85–123. Available online at http://josis.org/index.php/josis/article/viewArticle/73.

Agoston, M. K. (2005): Computer Graphics and Geometric Modeling. Mathematics. London: Springer-Verlag London Ltd. Available online at http://dx.doi.org/10.1007/b138899.

Aleksandrov, P. S. (1998): Combinatorial topology. Mineola, N.Y: Dover Publications.

Alexander, J. W. (1924): An Example of a Simply Connected Surface Bounding a Region which is not Simply Connected. In *Proceedings of the National Academy of Sciences of the United States of America* 10 (1), pp. 8–10.

Ali, A.; Abidi, M. (1988): A 2-D and 3-D robot path planning algorithm based on quadtree and octree representation of workspace. In: Southeastcon ’88., IEEE Conference Proceedings, pp. 391–396.

Anagnostopoulos, C.; Tsetsos, V.; Kikiras, P.; Hadjiefthymiades, S. P. (2005): OntoNav: A Semantic Indoor Navigation System. In: 1st Workshop on Semantics in Mobile Environments (SME05), Ayia.

Anguelov, D.; Koller, D.; Parker, E.; Thrun, S. (2004): Detecting and modeling doors with mobile robots. In: Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004 IEEE International Conference on Detecting and modeling doors with mobile robots, vol. 4, pp. 3777 - 3784 Vol.4.

Aouad, G.; Lee, A.; Wu, S. (2007): Constructing the future. ND modelling. London, New York: Taylor & Francis.

Atkinson, C.; Gutheil, M.; Kiko, K. (2006): On the Relationship of Ontologies and Models. In Brockmanns, S., Jung, J., Sure, Y. (Eds.): Meta-Modelling and Ontologies. Proceedings of the 2nd Workshop on Meta-Modelling and Ontologies (WoMM06), vol. 2006: Bonner Köllen Verlag (Lecture Notes in Informatics (LNI), P-96), pp. 47–60.

Baltsavias, E. P. (2004): Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. In *ISPRS Journal of Photogrammetry and Remote Sensing* 58 (3–4), pp. 129–151. Available online at http://www.sciencedirect.com/science/article/pii/S0924271603000546.

Bandi, S.; Thalmann, D. (1998): Space Discretization for Efficient Human Navigation. In *Computer Graphics Forum* 17 (3), pp. 195–206.

Baras, K.; Moreira, A.; Meneses, F. (2010): Navigation based on symbolic space models. In Mautz, R. (Ed.): International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2010. Zurich, Switzerland, 15 - 17 Sept. 2010. Institute of Electrical and Electronics Engineers; International Conference on Indoor Positioning and Indoor Navigation; IPIN. Piscataway, NJ: IEEE, pp. 1–5.

Barnes, M.; Finch, E. L. (2008): COLLADA – Digital Asset Schema Release 1.5.0. The Khronos Group Inc., Sony Computer Entertainment Inc. Available online at http://www.khronos.org/files/collada_spec_1_5.pdf.

Becker, C.; Dürr, F. (2005): On location models for ubiquitous computing. In *Personal and Ubiquitous Computing* 9 (1), pp. 20–31.

Becker, T.; Nagel, C.; Kolbe, T. H. (2009a): A Multilayered Space-Event Model for Navigation in Indoor Spaces. In Lee, J., Zlatanova, S. (Eds.): 3D geo-information sciences. Berlin: Springer (Lecture Notes in Geoinformation and Cartography), pp. 61–77.

Becker, T.; Nagel, C.; Kolbe, T. H. (2009b): Supporting Contexts for Indoor Navigation Using a Multilayered Space Model. In : Tenth International Conference on Mobile Data Management: systems, services and middleware, 2009. MDM '09 ; 18 - 20 May 2009, Taipei, Taiwan. Institute of Electrical and Electronics Engineers. Piscataway, NJ: IEEE, pp. 680–685.

Bellman, R. (1958): On a Routing Problem. In *Quarterly of Applied Mathematics* 16, pp. 87-90.

Benner, J.; Geiger, A.; Häfele, K.-H. (2010): Concept for Building Licensing based on Standardized 3D Geo Information. In Kolbe, T. H., König, G., Nagel, C. (Eds.): Proceedings of the 5th International Conference on 3D GeoInformation (3DGeoInfo). International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-4/W15. Berlin, Germany, 3-4 November. International Society for Photogrammetry and Remote Sensing (ISPRS), pp. 9–12.

Benner, J.; Geiger, A.; Leinemann, K. (2005): Flexible Generation of Semantic 3D Building Models. In Gröger, G., Kolbe, T. H. (Eds.): Proceedings of the 1st International Workshop on Next Generation 3D City Models. Bonn, Germany, 21-22 June. EuroSDR (Publication #49), pp. 17–22.

Bhatt, M.; Dylla, F.; Hois, J. (2009): Spatio-terminological Inference for the Design of Ambient Environments. In Stewart Hornsby, K., Claramunt, C., Denis, M., Ligozat, G. (Eds.): Spatial information theory. 9th international conference, COSIT 2009, Aber Wrac'h, France, September 21 - 25, 2009 ; proceedings. Berlin: Springer (Lecture Notes in Computer Science, 5756), pp. 371–391.

Biswas, R.; Limketkai, B.; Sanner, S.; Thrun, S. (2002): Towards object mapping in non-stationary environments with mobile robots. In: International Conference on Intelligent Robots and Systems, 2002. IEEE/RSJ, vol. 1, pp. 1014 - 1019 vol.1.

Bittner, T. (2001): The Qualitative Structure of Built Environments. In *Fundamenta Informaticae* 46 (1-2), pp. 97-128. Available online at http://dl.acm.org/citation.cfm?id=1219982.1219989.

Boguslawski, P.; Gold, C. (2009): Construction Operators for Modelling 3D Objects and Dual Navigation Structures. In Lee, J., Zlatanova, S. (Eds.): 3D geo-information sciences. Berlin: Springer (Lecture Notes in Geoinformation and Cartography), pp. 47–59.

Boguslawski, P.; Gold, C. (2010): Euler Operators and Navigation of Multi-shell Building Models. In Neutens, T., Maeyer, P. (Eds.): Developments in 3D Geo-Information Sciences. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg (Lecture Notes in Geoinformation and Cartography), pp. 1–16.

Boguslawski, P.; Gold, C. (2011): Rapid Modelling of Complex Building Interiors. In Kolbe, T. H., König, G., Nagel, C. (Eds.): Advances in 3D Geo-Information Sciences. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg (Lecture Notes in Geoinformation and Cartography), pp. 43–56.

Booch, G. (2007): Object-oriented analysis and design with applications. 3. ed., 1. printing. Upper Saddle River, NJ: Addison-Wesley (The Addison-Wesley object technology series). Available online at http://proquest.safaribooksonline.com/9780201895513.

Booch, G.; Rumbaugh, J.; Jacobson, I. (1999): The unified modeling language user guide. Reading Mass: Addison-Wesley.

Brasche, S.; Bischof, W. (2005): Daily time spent indoors in German homes – Baseline data for the assessment of indoor exposure of German occupants. In *International Journal of Hygiene and Environmental Health* 208 (4), pp. 247–253. Available online at http://www.sciencedirect.com/science/article/pii/S1438463905000635.

Brenner, C. (2005): Building reconstruction from images and laser scanning. In *International Journal of Applied Earth Observation and Geoinformation* 6 (3–4), pp. 187–198. Available online at http://www.sciencedirect.com/science/article/pii/S030324340400087X.

Brisson, E. (1989): Representing geometric structures in *d* dimensions: topology and order. In: Proceedings of the Fifth Annual Symposium on Computational geometry. New York, NY, USA: ACM Press (SCG ’89), pp. 218-227.

Brown, G.; Nagel, C.; Zlatanova, S.; Kolbe, T. H. (2012): Modelling 3D Topographic Space Against Indoor Navigation Requirements. In Pouliot, J., Daniel, S., Hubert, F., Zamyadi, A. (Eds.): Progress and New Trends in 3D Geoinformation Sciences. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg (Lecture Notes in Geoinformation and Cartography).

Brumitt, B.; Shafer, S. (2001): Topological World Modeling Using Semantic Spaces. In: Proceedings of Location Modeling Workshop at UbiComp.

Bychowski, T.; Mabrouk, M.; Niedzwiadek, H.; Herring, J.; Gaillet, J.-F. (2003): OpenGIS - Location Services (OpenLS): Part 6-Navigation Service, version 0.5.0. Open Geospatial Consortium (OGC Doc. No. 03-007r1). Available online at http://www.opengeospatial.org/standards/ols.

Chen, G.; Kotz, D. (2000): A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381. Dartmouth College Hanover, NH, USA.

Chen, P. P. (2002): Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned. In Broy, M., Denert, E. (Eds.): Software pioneers. Contributions to software engineering. Berlin, New York: Springer, pp. 297-310.

Choset, H.; Walker, S.; Eiamsa-Ard, K.; Burdick, J. (2000): Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph. In *The International Journal of Robotics Research* 19 (2), pp. 126–148.

Congli, W.; Tsuzuki, M. d. S. (2004): Representation of Curves and Surfaces in B-Rep Solid Modelers. In Adamowski, J. C., Tamai, E. H., Villani, E., Miyagi, P. E. (Eds.): ABCM Symposium Series in Mechatronics, vol. 1, pp. 498–507.

Coors, V. (2003): 3D-GIS in networking environments. In *Computers, Environment and Urban Systems* 27 (4), pp. 345–357.

Cox, S.; Daisey, P.; Lake, R.; Portele, C.; Whiteside, A. (2012): OpenGIS® Geography Markup Language (GML) Implementation Specification, version 3.1.1. Open Geospatial Consortium (OGC Doc. No. 03-105r1). Available online at http://www.opengeospatial.org/standards/gml.

Dehn, M.; Heegaard, P. (1907): Analysis situs. In: Enzyklopädie der Mathematischen Wissenschaften, III A B 3, pp. 153–220.

Delling, D.; Sanders, P.; Schultes, D.; Wagner, D. (2009): Engineering Route Planning Algorithms. In Lerner, J. (Ed.): Algorithmics of Large and Complex Networks. Design, analysis, and simulation. Berlin, Heidelberg, New York: Springer Berlin Heidelberg (Lecture Notes in Computer Science, 5515), pp. 117-139.

Demyen, D.; Buro, M. (2006): Efficient triangulation-based pathfinding. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06). Innovative Applications of Artificial Intelligence Conference. Menlo Park, Calif: AAAI Press, pp. 942-947. Available online at http://dl.acm.org/citation.cfm?id=1597538.1597687.

Desyllas, J.; Duxbury, E. (2001): Axial maps and visibility graph analysis. In: Proceedings of the 3rd International Symposium on Space Syntax. Georgia Institute of Technology, Atlanta, Georgia, pp. 27.1 – 27.13.

Dijkstra, E. W. (1959): A note on two problems in connexion with graphs. In *Numerische Mathematik* 1 (1), pp. 269–271.

Dobkin, D. P.; Laszlo, M. J. (1987): Primitives for the manipulation of three-dimensional subdivisions. In: Proceedings of the Third Annual Symposium on Computational geometry. New York, NY, USA: ACM Press (SCG ’87), pp. 86-99.

Dudas, P.; Ghafourian, M.; Karimi, H. (2009): ONALIN: Ontology and Algorithm for Indoor Routing. In: Tenth International Conference on Mobile Data Management: systems, services and middleware, 2009. MDM '09 ; 18 - 20 May 2009, Taipei, Taiwan. Institute of Electrical and Electronics Engineers. Piscataway, NJ: IEEE, pp. 720–725.

Early, E. (1999): On the Euler Characteristic. In *The MIT Undergraduate Journal of Mathematics* (1), pp. 39–40.

Eastman, C. M. (2008): BIM handbook. A guide to building information modeling for owners, managers, designers, engineers, and contractors. Hoboken, N.J: Wiley.

Edelsbrunner, H.; Harer, J. L. (2010): Computational topology. An introduction. Providence, RI: American Math. Soc (Applied mathematics).

Egenhofer, M. J.; Clementini, E.; Di Felice, P. (1994): Topological Relations Between Regions With Holes. In *Int. Journal of Geographical Information Systems* 8, pp. 129–142.

Egenhofer, M.; Franzosa, R. (1991): Point-set topological spatial relations. In *International journal of geographical information systems* 5 (2), pp. 161–174.

Egenhofer, M.; Herring, J. (1991): Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases. Technical Report. Orono, ME: Department of Surveying Engineering, University of Maine.

Egenhofer, M.; Sharma, J. (1993): A critical comparison of the 4-intersection and 9-intersection models for spatial relations: formal analysis. In McMaster, R., Armstrong, M. (Eds.): Proceedings of AutoCarto, vol. 11. Minneapolis.

Eilenberg, S.; Zilber, J. A. (1950): Semi-Simplicial Complexes and Singular Homology. In *Annals of Mathematics* 51 (3), pp. 499–513.

Eliseo Clementini; Paolino Di Felice (1995): A comparison of methods for representing topological relationships. In *Information Sciences - Applications* 3 (3), pp. 149–178. Available online at http://www.sciencedirect.com/science/article/pii/106901159400033X.

Engelson, S.; McDermott, D. (1992): Error correction in mobile robot map learning. In: ICRA 1992. Proceedings of the IEEE International Conference on Robotics and Automation. Los Alamitos, Calif: IEEE Computer Society Press, pp. 2555–2560.

Eppstein, D.; Goodrich, M. T.; Sun, J. Z. (2005): The skip quadtree: a simple dynamic data structure for multidimensional data. In: Proceedings of the twenty-first annual symposium on Computational geometry. New York, NY, USA: ACM (SCG ’05), pp. 296-305.

Farin, G. E.; Hoschek, J.; Kim, M.-S. (2002): Handbook of computer aided geometric design. Amsterdam, Boston, Mass: Elsevier.
Available online at http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10190088.

Flick, S. (1999): Konzeption eines adaptiven Frameworks für 3D-Geo-Informationssysteme. Stuttgart, Darmstadt: Fraunhofer IRB.

Fod, A.; Howard, A.; Mataric, M. J. (2002): Laser-Based People Tracking. In: Proceedings of the IEEE International Conference on Robotics & Automation (ICRA). IEEE Computer Society, pp. 3024-3029.

Foley, J. D. (2010): Computer graphics. Principles and practice. 2. ed. in C, reprinted with corrections. Boston: Addison-Wesley (The systems programming series).

Ford, L. R. (1956): Network Flow Theory. Santa Monica, CA: RAND Corporation (RAND Paper). Available online at http://www.rand.org/pubs/papers/P923.

Franz, G.; Mallot, H. A.; Wiener, J. M. (2005): Graph-based models of space in architecture and cognitive science - a comparative analysis. In Leong, Y.-T. (Ed.): Architecture, engineering and construction of built environments. Proceedings of 17th International Conference on Systems Research, Informatics and Cybernetics, August 1 - 7, 2005, Baden-Baden, Germany.

Windsor, Ontario: International Institute for Advanced Studies in Systems Research and Cybernetics (/IIAS, 181), pp. 30-38. Available online at http://eprints.bournemouth.ac.uk/13803/.

Fuchs, G.; Mabrouk, M.; Wilbrink, W.; Smyth, S.; Bychowski, T. (2008): OpenGIS - Location Services (OpenLS): Part 6-Navigation Service, version 1.0.0. Open Geospatial Consortium (OGC Doc. No. 08-028r7). Available online at http://www.opengeospatial.org/standards/ols.

Gibson, J. J. (1986): The ecological approach to visual perception. Hillsdale NJ u.a: Lawrence Erlbaum.

Gilliéron, P.-Y.; Merminod, B. (2003): Personal Navigation System for Indoor Applications. In: Proceedings of the 11th International Association of Institutes of Navigation (IAIN) World Congress. Berlin, Germany, 21-24 October.

Giudice, N. A.; Walton, L. A.; Worboys, M. (2010): The informatics of indoor and outdoor space: a research agenda. In Winter, S., Jensen, C. S., Li, K.-J. (Eds.): Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. New York, NY, USA: ACM, pp. 47–53.

Goetz, M.; Zipf, A. (2010): Open issues in Bringing 3D to Location Base Services (LBS) - A Review Focusing on 3D Data Streaming and 3D Indoor Navigation. In Kolbe, T. H., König, G., Nagel, C. (Eds.): Proceedings of the 5th International Conference on 3D GeoInformation (3DGeoInfo). International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-4/W15. Berlin, Germany, 3-4 November. International Society for Photogrammetry and Remote Sensing (ISPRS), pp. 121–124.

Goetz, M.; Zipf, A. (2011): Formal definition of a user-adaptive and length-optimal routing graph for complex indoor environments. In *Geo-spatial Information Science* 14 (2), pp. 119–128.

Gold, C. M. (2004): Solid modeling. In Goodman, J. E., O'Rourke, J. (Eds.): Handbook of discrete and computational geometry. 2nd ed. Boca Raton: Chapman & Hall/CRC (Discrete mathematics and its applications), pp. 1257–1278.

Gotts, N. M.; Gooday, J. M.; Cohn, A. G. (1996): A connection based approach to commonsense topological description and reasoning. In *The Monist* 79 (1), pp. 51–75.

Gröger, G.; Kolbe, T. H.; Nagel, C.; Häfele, K.-H. (2012): OGC City Geography Markup Language (CityGML) Encoding Standard, version 2.0. Open Geospatial Consortium (OGC Doc. No. 12-019). Available online at http://www.opengeospatial.org/standards/citygml.

Gröger, G.; Plümer, L. (2005): How to Get 3-D for the Price of 2-D - Topology and Consistency of 3-D Urban GIS. In *Geoinformatica* 9 (2), pp. 139-158.

Gröger, G.; Plümer, L. (2010): Derivation of 3D Indoor Models by Grammars for Route Planning. In *Photogrammetrie - Fernerkundung - Geoinformation* 2010 (3), pp. 191–206.

Gröger, G.; Plümer, L. (2011a): How to achieve consistency for 3D city models. In *Geoinformatica* 15 (1), pp. 137–165.

Gröger, G.; Plümer, L. (2011b): Topology of surfaces modelling bridges and tunnels in 3D-GIS. In *Computers, Environment and Urban Systems* 35 (3), pp. 208–216.

Gröger, G.; Plümer, L. (2012a): CityGML – Interoperable semantic 3D city models. In *ISPRS Journal of Photogrammetry and Remote Sensing* (71), pp. 12–23.

Gröger, G.; Plümer, L. (2012b): Provably correct and complete transaction rules for updating 3D city models. In *Geoinformatica* 16 (1), pp. 131–164.

Guibas, L.; Stolfi, J. (1985): Primitives for the manipulation of general subdivisions and the computation of Voronoi. In *ACM Transactions on Graphics (TOG)* 4 (2), pp. 74-123.

Hagedorn, B.; Trapp, M.; Glander, T.; Dollner, J. (2009): Towards an Indoor Level-of-Detail Model for Route Visualization. In: Tenth International Conference on Mobile Data Management: systems, services and middleware, 2009. MDM '09 ; 18 - 20 May 2009, Taipei, Taiwan. Institute of Electrical and Electronics Engineers. Piscataway, NJ: IEEE, pp. 692-697.

Hamilton, A.; Wang, H.; Tanyer, A. M.; Arayici, Y.; Zhang, X.; Song, Y. H. (2005): Urban information model for city planning. In *Journal of Information Technology in Construction (ITcon), Special Issue From 3D to nD modelling* (10), pp. 55–67.

Hatcher, A. (2008): Algebraic topology. 10 print. Cambridge: Cambridge University Press.

Hendricks, M. D.; Egenhofer, M. J.; Hornsby, K. (2003): Structuring a wayfinder's dynamic space-time environment. In Kuhn, W. (Ed.): Proceedings of the International Conference on Spatial Information Theory (COSIT 2003). Berlin: Springer (Lecture Notes in Computer Science, 2825), pp. 75–92.

Hightower, J.; Brumitt, B.; Borriello, G. (2002): The Location Stack: A Layered Model for Location in Ubiquitous Computing. In: Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002). 20 - 21 June 2002, Callicoon, New York. Los Alamitos, California: IEEE Computer Society, pp. 22-28. Available online at http://dl.acm.org/citation.cfm?id=832315.837552.

Hillier, B.; Hanson, J.; Peponis, J.; Hudson, J.; Burdett R. (1983): Space syntax. In *Architects Journal* 178 (48), pp. 67 – 75.

Hillier, B.; Hanson, J. (1984): The Social Logic of Space. Cambridge: Cambridge University Press.

Hoppenot, P.; Pradel, G.; Caleanu, C.; Perrin, N.; Sommeilly, V. (2003): Towards a symbolic representation of an indoor environment. In: IEEE CESA - Conference on Computational Engineering in Systems Applications.

Howell, I.; Batcheler, B.: Building Information Modeling Two Years Later – Huge Potential, Some Success and Several Limitations. Edited by Jerry Laiserin. The Laiserin Letter, Issue 24, 2005. Available online at http://www.laiserin.com/features/bim/newforma_bim.pdf.

Hu, H.; Lee, D.-L. (2004): Semantic location modeling for location navigation in mobile environment. In: Proceedings of the IEEE International Conference on Mobile Data Management. 19 - 22 January 2004, Berkeley, California. Los Alamitos, California: IEEE Computer Soc, pp. 52–61.

Jacobson, N. (2009): Basic algebra. Mineola, NY: Dover Publications (Dover books on mathematics).

Jänich, K. (2012): Topology: Springer New York (Undergraduate texts in mathematics).

Jenkins, P. L.; Phillips, T. J.; Mulberg, E. J.; Hui, S. P. (1992): Activity patterns of Californians: Use of and proximity to indoor pollutant sources. In *Atmospheric Environment. Part A. General Topics* 26 (12), pp. 2141–2148. Available online at http://www.sciencedirect.com/science/article/pii/0960168692904027.

Jensen, C. S.; Lu, H.; Yang, B. (2009): Graph Model Based Indoor Tracking. In: Tenth International Conference on Mobile Data Management: systems, services and middleware, 2009. MDM '09 ; 18 - 20 May 2009, Taipei, Taiwan. Institute of Electrical and Electronics Engineers. Piscataway, NJ: IEEE, pp. 122-131.

Jensen, C. S.; Lu, H.; Yang, B. (2010): Indoor - A New Data Management Frontier. In *IEEE Data Eng. Bull* 33 (2), pp. 12–17.

Johnson, M. (1987): The body in the mind. The bodily basis of meaning, imagination, and reason. Chicago: Univ. of Chicago Press.

Khan, A. A.; Kolbe, T. H. (2012): Constraints and their role in subspacing for the locomotion types in indoor navigation. In: Proceedings of the 3rd International Conference on Indoor Positioning and Indoor Navigation (IPIN) (to be published online in the IEEE Xplore database). New South Wales, Sydney, Australia, November 13-15.

Kim, J.-S.; Kang, H.-Y.; Lee, T.-H.; Li, K.-J. (2009): Topology of the Prism Model for 3D Indoor Spatial Objects. In: Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware. Washington, DC, USA: IEEE Computer Society (MDM ’09), pp. 698-703.

Kolbe, T. H. (2009): Representing and Exchanging 3D City Models with CityGML. In Lee, J., Zlatanova, S. (Eds.): 3D geoinformation sciences. Berlin: Springer (Lecture Notes in Geoinformation and Cartography), pp. 15–31.

Kolbe, T. H.; Nagel, C.; Stadler, A. (2009): CityGML - OGC Standard for Photogrammetry. In Fritsch, D. (Ed.): Photogrammetric Week '09. Heidelberg: Wichmann, pp. 265-277.

Kolbe, T. H.; Plümer, L. (2004): Bridging the Gap between GIS and CAAD. In *GIM International* 18 (7), pp. 12–15.

Kolodziej, K. W.; Hjelm, J. (2006): Local positioning systems. LBS applications and services. Boca Raton, FL: CRC/Taylor & Francis. Available online at http://www.lob.de/cgi-bin/work/suche2?titnr=262264201&flag=citavi.

Kolomvatsos, K.; Papataxiarhis, V.; Tsetsos, V. (2009): Semantic Location Based Services for Smart Spaces. In Sicilia, M.-A., Lytras, M. D. (Eds.): Metadata and Semantics. Boston, MA: Springer US, pp. 515–525. Available online at http://dblp.uni-trier.de/db/conf/mtsr/mtsr2007.html#KolomvatsosPT07.

Kresse, W.; Danko, D. M. (2012): Springer Handbook of Geographic Information. Berlin, Heidelberg: Springer Berlin Heidelberg.

Kresse, W.; Fadaie, K. (2004): ISO standards for geographic information. Berlin: Springer. Available online at http://www.loc.gov/catdir/enhancements/fy0817/2003065240-d.html.

Kruse, E.; Wahl, F. (1998): Camera-based monitoring system for mobile robot guidance. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 1248–1253.

Lacroix, Tristan E. (2013): Indoor Location & Mapping Services Report v5.8. The Convergence of GPS, WiFi, Bluetooth, and MEMS. Market Report. Edited by IndoorLBS, LLC. Available online at http://www.indoorlbs.com/p/market-report.html.

Lamarche, F.; Donikian, S. (2004): Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. In *Computer Graphics Forum* 23, pp. 509-518.

Ledoux, H.; Gold, C. M. (2007): Simultaneous storage of primal and dual three-dimensional subdivisions. In *Computers, Environment and Urban Systems* 31 (4), pp. 393–408.

Lee, J. (2001): 3D Data Model for Representing Topological Relations of Urban Features. In *Proceedings of the 21st Annual ESRI International User Conference, San Diego, CA, USA.*

Lee, J. (2004a): 3D GIS for Geo-coding Human Activity in Micro-scale Urban Environments. In Egenhofer, M. J., Freksa, C., Miller, H. J. (Eds.): Geographic Information Science, vol. 3234: Springer Berlin / Heidelberg (Lecture Notes in Computer Science), pp. 162–178.

Lee, J. (2004b): A Spatial Access Oriented Implementation of a Topological Data Model for 3D Urban Entities. In *Geoinformatica* 8 (3), pp. 235–262.

Lee, J. (2009): GIS-based geocoding methods for area-based addresses and 3D addresses in urban areas. In *Environment and Planning B: Planning and Design* 36 (1), pp. 86–106.

Lee, J. M. (2011): Introduction to Topological Manifolds. 2nd ed. New York, NY: Springer Science and Business Media LLC (Graduate texts in mathematics, 202).

Lee, J.; Kwan, M. (2005): A combinatorial data model for representing topological relations among 3D geographical features in micro-spatial environments. In *International Journal of Geographical Information Science* 19 (10), pp. 1039–1056.

Lee, J.; Zlatanova, S. (2008): A 3D data model and topological analyses for emergency response in urban areas. In Zlatanova, S., Li, J. (Eds.): Geospatial information technology for emergency response. London: Taylor & Francis (International Society for Photogrammetry and Remote Sensing book series, 6), pp. 143–168.

Leonhardt, U. (1998): Supporting Location-Awareness in Open Distributed Systems. PhD Thesis. University of London, London. Department of Computing, Imperial College of Science, Technology and Medicine. Available online at citeseer.ist.psu.edu/article/leonhardt98supporting.html.

Liao, L.; Dieter, F.; Jeffrey, H.; Henry, K.; Dirk, S. (2003): Voronoi Tracking: Location Estimation Using Sparse and Noisy Sensor Data. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Li, D.; Lee, D. L. (2009a): A Lattice-Based Semantic Location Model for Indoor Navigation. In: Tenth International Conference on Mobile Data Management: systems, services and middleware, 2009. MDM '09 ; 18 - 20 May 2009, Taipei, Taiwan. Institute of Electrical and Electronics Engineers. Piscataway, NJ: IEEE, pp. 17–24.

Li, D.; Lee, D. L. (2009b): A topology-based semantic location model for indoor applications. In: 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems 2008 (ACM GIS 2008). Irvine, California, USA, 5 - 7 November. Association for Computing Machinery. Red Hook, NY: Curran, pp. 6:1-6:10.

Liebich, T.; Adachi, Y.; Forester, J.; Hyvarinen, J.; Karstila, K.; Reed, K. et al. (2007): Industry Foundation Classes- IFC2x3 - Technical Corrigendum 1. buildingSMART. Available online at http://buildingsmart-tech.org/.

Li, J.; Heap, A. D. (2008): A review of spatial interpolation methods for environmental scientists. Canberra: Geoscience Australia (Record 2008/23).

Li, K.-J.; Lee, J. (2010): Indoor Spatial Awareness Initiative and Standard for Indoor Spatial Data. In: Proceedings of IROS 2010. IEEE International Workshop on Standardization for Service Robot. Taiwan, October 18.

Liu, L.; Zlatanova, S. (2011a): A 'door-to-door' path-finding approach for indoor navigation. In Altan, Backhause, Boccardo, Zlatanova (Eds.): Proceedings of the Gi4DM 2011 - GeoInformation For Disaster Management. International Archives ISPRS XXXVIII. 7th Gi4DM. Anltalya, Turkey, 3-7 May. International Society for Photogrammetry and Remote Sensing (ISPRS).

Liu, L.; Zlatanova, S. (2011b): Towards a 3D network model for indoor navigation. In Zlatanova, S., Ledoux, H., Fendel, E. M., Rumor, M. (Eds.): Urban and Regional Data Management: UDMS Annual: 2011. UDMS Annual: Taylor & Francis Ltd.

Liu, L.; Zlatanova, S. (2012): A Semantic Data Model for Indoor Navigation. In Claramunt, C., Li, K.-J. (Eds.): ISA ’12: Proceedings of the Fourth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. New York, NY, USA: ACM, pp. 1–8.

Li, X.; Claramunt, C.; Ray, C. (2010): A grid graph-based model for the analysis of 2D indoor spaces. In *Computers, Environment and Urban Systems* 34 (6), pp. 532–540.

Li, Y.; He, Z. (2008): 3D Indoor Navigation: a Framework of Combining BIM with 3D GIS. In: Proceedings of the 44th ISOCARP Congress.

Lorenz, A.; Thierbach, C.; Kolbe, T. H.; Baur, N.; Kohlhofer, G.; Franzen, M. (2010): Untersuchung der Effizienz und Akzeptanz von 2D- und 3D-Kartenvarianten für die Innenraumnavigation. (German only). In *Vorträge Dreiländertagung OVG, DGPF und SGPF - 30. Wissenschaftlich-Technische Jahrestagung der DGPF* (Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation), pp. 342–355.

Lorenz, B.; Ohlbach, H. (2006): A Hybrid Spatial Model for Indoor Spatial Reasoning. In *Proceedings of the 1st Int. Workshop on Mobile Geospatial Augmented Reality*, pp. 2–7.

Lorenz, B.; Ohlbach, H.; Stoffel, E.-P. (2006): A Hybrid Spatial Model for Representing Indoor Environments. In Carswell, J. D., Tezuka, T. (Eds.): Web and wireless geographical information systems. 6th international symposium, W2GIS 2006, Hong Kong, China, December 4 - 5, 2006 ; proceedings, vol. 4295. Berlin: Springer (Lecture Notes in Computer Science, 4295), pp. 102–112.

Lozano-Perez, T. (1983): Spatial Planning: A Configuration Space Approach. In *IEEE Transactions on Computers* 32 (2), pp. 108–120.

Lyardet, F.; Grimmer, J.; Muhlhauser, M. (2006): CoINS: Context Sensitive Indoor Navigation System. In: Eighth IEEE International Symposium on Multimedia, ISM’06, pp. 209–218.

Mabrouk, M. (2008): OpenGIS - Location Services (OpenLS): Core Services, version 1.2. Part 1-Directory Service, Part 2-Gateway Service, Part 3-Location Utility Service (Geocoder/Reverse Geocoder), Part 4-Presentation Service, Part 5-Route Service. Open Geospatial Consortium (OGC Doc. No. 07-074). Available online at http://www.opengeospatial.org/standards/ols.

Mäntylä, M. (1988): An introduction to solid modeling. Rockville, Md: Computer Science Press (Principles of computer science series, 13).

Meijers, M.; Zlatanova, S.; Pfeifer, N. (2005): 3D Geo-Information Indoors: Structuring for Evacuation. In Gröger, G., Kolbe, T. H. (Eds.): Proceedings of the 1st International Workshop on Next Generation 3D City Models. Bonn, Germany, 21-22 June. EuroSDR (Publication #49).

Milner, R. (2008): Bigraphs and Their Algebra. In *Electronic Notes in Theoretical Computer Science* 209, pp. 5–19.

Milnor, J. W. (1961): Two complexes which are homeomorphic but combinatorially distinct. In *Annals of Mathematics* 74 (2), pp. 575-;590.

Moise, E. E. (1977): Geometric topology in dimensions 2 and 3. New York: Springer (Graduate texts in mathematics, 47).

Mokbel, M. F.; Levandoski, J. J. (2009): Toward context and preference-aware location-based services. In Kotidis, Y., Marrón, P. J. (Eds.): Proceedings of the eighth International Workshop on Data Engineering for Wireless and Mobile Access. Providence, Rhode Island, June 29. ACM SIGMOD International Conference on Management of Data. New York, N.Y: ACM Press, pp. 25-32.

Molenaar, M. (1990): A formal data structure for 3d vector maps. In: Proceedings of EGIS'90, vol. 2. Amsterdam, The Netherlands, pp. 770–781.

Monastyrsky, M. I. (2008): Riemann, Topology, and Physics. 2nd ed. Cambridge, MA: Birkhäuser Boston (Modern Birkhäuser Classics).

Moravec, H.; Elfes, A. (1985): High resolution maps from wide angle sonar. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, pp. 116–121.

Morris, S. A. (2012): Topology Without Tears. Available online at http://uob-community.ballarat.edu.au/~smorris/topology.htm.

Müller, P.; Wonka, P.; Haegler, S.; Ulmer, A.; van Gool, L. (2006): Procedural modeling of buildings. In Finnegan, J. (Ed.): Proceedings of ACM SIGGRAPH 2006. New York, NY: ACM, pp. 614-623.

Munkres, J. R. (1984): Elements of algebraic topology. Menlo Park, CA: Addison-Wesley.

Muttitanon, W.; Tripathi, N. K.; Souris, M. (2007): An Indoor Positioning System (IPS) using Grid Model. In *Journal of Computer Science* 3 (17), pp. 907–913.

Nagel, C.; Becker, T.; Kaden, R.; Li, K.-J.; Lee, J.; Kolbe, T. H. (2010): Requirements and Space-Event Modeling for Indoor Navigation: Open Geospatial Consortium (OGC Discussion Paper, Doc. No. 10-191r1). Available online at http://portal.opengeospatial.org/files/?artifact_id=41727.

Nagel, C.; Stadler, A.; Kolbe, T. H. (2009): Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models. In Kolbe, T. H., Zhang, H., Zlatanova, S. (Eds.): International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-3-4/C3. GeoWeb 2009 Academic Track - Cityscapes. Vancouver, BC, Canada, 27-31 July. International Society for Photogrammetry and Remote Sensing (ISPRS), pp. 46–53.

Neis, P.; Zipf, A. (2008): Extending the OGC OpenLS Route Service to 3D for an interoperable realisation of 3D focus maps with landmarks. In *International Journal of Location Based Services (JLBS)* 2 (2), pp. 153–174.

Ogawa, K.; Verbree, E.; Zlatanova, S.; Kohtake, N.; Ohkami, Y. (2011): Toward seamless indoor-outdoor applications: Developing stakeholder-oriented location-based services. In *Geo-spatial Information Science* 14 (2), pp. 109–118.

OMG (2011): Unified Modelling Language (UML), version 2.4.1. Object Management Group. Available online at http://www.omg.org/spec/UML/2.4.1/.

OMG (2012): Object Constraint Language (OCL), version 2.3.1. Object Management Group. Available online at http://www.omg.org/spec/OCL/2.3.1.

Papataxiarhis, V.; Riga, V.; Nomikos, V.; Sekkas, O.; Kolomvatsos, K.; Tsetsos, V. et al. (2007): MNISIKLIS: Indoor Location Based Services for All. In Gartner, G., Cartwright, W., Peterson, M. (Eds.): Location Based Services and TeleCartography: Springer Berlin Heidelberg (Lecture Notes in Geoinformation and Cartography), pp. 263–282.

Park, I.; Lee, J. (2008): Defining 3D Spatial Neighborhoods for Topological Analyses using a 3D Network-Based Topological Data Model - CA-Based Building Evacuation Simulation -. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the ISPRS Congress Beijing 2008. Commission II, Volume XXXVII, Part B2. Beijing, China. International Society for Photogrammetry and Remote Sensing (ISPRS), pp. 305–310.

Pigot, S. (1996): A Topological Model for a 3-Dimensional Spatial Information System. Ph.D. Dissertation. University of Tasmania, Australia. Surveying and Spatial Information Science.

Pilouk, M. (1996): Integrated modelling for 3D GIS. Ph.D. Dissertation. Wageningen Agricultural University, The Netherlands. Department of Geographic Information Processing and Remote Sensing.

Poincaré, H. (1895): Analysis situs. In *Journal de l'École Polytechnique* 1 (2), pp. 1–121.

Poincaré, H. (1899): Complément à l'analysis situs. In *Rendiconti del Circulo Matematico di Palermo* 13, pp. 285–343.

Portele, C. (2007): OpenGIS® Geography Markup Language (GML) Encoding Standard, version 3.2.1. Open Geospatial Consortium (OGC Doc. No. 07-036). Available online at http://www.opengeospatial.org/standards/gml.

Portele, C. (2012): OGC® Geography Markup Language (GML) — Extended schemas and encoding rules, version 3.3.0. Open Geospatial Consortium (OGC Doc. No. 10-129r1). Available online at http://www.opengeospatial.org/standards/gml.

Prishlyak, A. O.; Mischenko, K. I. (2007): Classification of noncompact surfaces with boundary. In *Methods of Functional Analysis and Topology* 13 (1), pp. 62–66.

Radó, T. (1925): Über den Begriff der Riemannschen Fläche. In *Acta Szeged* (2), pp. 101–121.

Randell, D. A.; Cui, Z.; Cohn, A. G. (1992): A Spatial Logic based on Regions and Connection. In: Proceedings 3rd International Conference on Knowledge Representation and Reasoning. Available online at http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=D6292D87952631A877B320442F286BA5?doi=10.1.1.35.7809&rep=rep1&type=pdf.

Ranicki, Andrew (Ed.) (1996): The Hauptvermutung book. A collection of papers of the topology of manifolds. Dordrecht: Kluwer Acad. Publ (K-monographs in mathematics, 1).

Raubal, M.; Worboys, M.; Freksa, C.; Mark, D. (1999): A Formal Model of the Process of Wayfinding in Built Environments. In *Knowledge Creation Diffusion Utilization* 1661, pp. 381-399.

Requicha, A. A. G. (1980): Representations for Rigid Solids: Theory, Methods, and Systems. In *ACM Computing Surveys* 12 (4), pp. 437–464.

Requicha; Voelcker (1982): Solid Modeling: A Historical Summary and Contemporary Assessment. In *IEEE Computer Graphics and Applications* 2 (2), pp. 9–24.

Retscher, G. (2007): Ubiquitous positioning solutions for pedestrian navigation. In *Proceedings of the 5th International Symposium on Mobile Mapping Technology*.

Richter, K.-F.; Winter, S.; Ruetschi, U.-J. (2009): Constructing Hierarchical Representations of Indoor Spaces. In: Tenth International Conference on Mobile Data Management: systems, services and middleware, 2009. MDM '09 ; 18 - 20 May 2009, Taipei, Taiwan. Institute of Electrical and Electronics Engineers. Piscataway, NJ: IEEE, pp. 686–691.

Rossignac, J. R.; Requicha, A. A. G. (1999): Solid modeling. In Webster, J. G. (Ed.): Encyclopedia of electrical and electronics engineering. New York: J. Wiley & Sons.

Rüetschi, U. J. (2007): Wayfinding in scene space: modelling transfers in public transport. PhD Thesis. University of Zurich. Faculty of Science. Available online at http://dx.doi.org/10.5167/uzh-20208.

Ruppel, P.; Klein, C.; Linnhoff-Popien, C. (2008): Indooria - A Platform for Proactive Indoor Location-based Services. In: IEEE Globecom 2008 Design and Developers Forum. New Orleans, LA, USA.

Russell, S. J.; Norvig, P. (2010): Artificial intelligence. A modern approach. 3rd ed. Upper Saddle River, NJ: Prentice-Hall (Prentice-Hall series in artificial intelligence).

Schilling, A.; Goetz, M. (2010): Decision Support Systems using 3D OGC Services and Indoor Routing - Example Scenario from the OWS-6 Testbed. In Kolbe, T. H., König, G., Nagel, C. (Eds.): Proceedings of the 5th International Conference on 3D GeoInformation (3DGeoInfo). International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-4/W15. Berlin, Germany, 3-4 November. International Society for Photogrammetry and Remote Sensing (ISPRS), pp. 159–162.

Schmidt, A.; Aidoo, K. A.; Takaluoma, A.; Tuomela, U.; Laerhoven, K.; Velde, W. (1999): Advanced Interaction in Context. In Gellersen, H.-W. (Ed.): Handheld and ubiquitous computing. Proceedings. Berlin: Springer (Lecture Notes in Computer Science, 1707), pp. 89–101.

Shi, W.; Yang, B.; Li, Q. (2003): An object-oriented data model for complex objects in three-dimensional geographical information systems. In *International Journal of Geographical Information Science* 17 (5), pp. 411–430.

Skopenkov, A. (2008): Embedding and knotting of manifolds in Euclidean spaces. In *London Mathematical Society Lecture Notes* 347, pp. 248–342.

Slingsby, A.; Raper, J. (2008): Navigable Space in 3D City Models for Pedestrians. In Oosterom, P., Zlatanova, S., Penninga, F., Fendel, E. (Eds.): Advances in 3D Geo Information Systems. 1st ed. Berlin, Heidelberg: Springer-Verlag (Lecture Notes in Geoinformation and Cartography).

Smyth, C. S. (2008): OGC Location Services: Tracking Service Interface Standard. Open Geospatial Consortium (OGC Doc. No. 06-024r4). Available online at http://www.opengeospatial.org/standards/ols.

Stachniss, C.; Rottmann, A.; Burgard, W. (2007): Using AdaBoost for Place Labeling and Topological Map Building. In *Robotics Research* 28, pp. 453-472.

Stadler, A.; Kolbe, T. H. (2007): Spatio-semantic Coherence in the Integration of 3D City Models. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Proceedings of the 5th International Symposium on Spatial Data Quality. Volume XXXVI-2/C43. Enschede, The Netherlands. International Society for Photogrammetry and Remote Sensing (ISPRS).

Steen, L. A.; Seebach, J. A. (1995): Counterexamples in topology. New York: Dover Publ. Available online at http://www.loc.gov/catdir/description/dover032/95012763.html.

Stoffel, E.-P.; Lorenz, B.; Ohlbach, H. J. (2007): Towards a semantic spatial model for pedestrian indoor navigation. In Hainaut, J.-L., Rundensteiner, E. (Eds.): Advances in conceptual modeling. ER 2007 Workshop Proceedings, Auckland, New Zealand, November 5-9, 2007. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg (4802), pp. 328-337. Available online at http://dl.acm.org/citation.cfm?id=1784542.1784596.

Stoffel, E.-P.; Schoder, K.; Ohlbach, H. J. (2009): Applying hierarchical graphs to pedestrian indoor navigation. In Aref, W. G., Mokbel, M. F., Schneider, M. (Eds.): Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems 2008 (ACM GIS 2008). Irvine, California, USA, 5 - 7 November 2008. Red Hook, NY: Curran, pp. 54:1-54:4.

Stoter, J.; Reuvers, M.; Vosselmann, G.; Goos, J.; van Berlo, L.; Zlatanova, S. et al. (2010): Towards a 3D Geo-Information Standard in the Netherlands. In Kolbe, T. H., König, G., Nagel, C. (Eds.): Proceedings of the 5th International Conference on 3D GeoInformation (3DGeoInfo). International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-4/W15. Berlin, Germany, 3-4 November. International Society for Photogrammetry and Remote Sensing (ISPRS), pp. 63–67.

Stuckenschmidt, H.; Parent, C.; Spaccapietra, S. (2009): Modular Ontologies. Concepts, theories and techniques for knowledge modularization. New York: Springer.

Thom, R. (1969): Ensembles et morphismes stratifiés. In *Bulletin of the American Mathematical Society* 75, pp. 240–284.

Thrun, S.; Bü, A. (1996): Integrating grid-based and topological maps for mobile robot navigation. In: Proceedings of the thirteenth national conference on Artificial intelligence - Volume 2: AAAI Press (AAAI'96), pp. 944-950.

Tsetsos, V.; Anagnostopoulos, C.; Kikiras, P.; Hadjiefthymiades, S. (2006): Semantically enriched navigation for indoor environments. In *International Journal of Web and Grid Services* 2 (4), pp. 453-478. Available online at http://dl.acm.org/citation.cfm?id=1358499.1358504.

Turner, A.; Penn, A.; Hillier, B. (2005): An algorithmic definition of the axial map. In *Environment and Planning B: Planning and Design* 32 (3), pp. 425 – 444.

van den Brink, L.; Stoter, J.; Zlatanova, S. (2012): Establishing a national standard for 3D topographic data compliant to CityGML. In *International Journal of Geographical Information Science*, pp. 1–22.

van Oosterom, P.; Stoter, J. (2010): 5D data modelling: full integration of 2D/3D space, time and scale dimensions. In Fabrikant, S. I., Reichenbacher, T., van Kreveld, M., Schlieder, C. (Eds.): Geographic information science. Proceedings of the 6th International Conference on Geographic Information Science. Berlin, Heidelberg, New York, NY: Springer (Lecture Notes in Computer Science (LNCS), 6292), pp. 310-324.

Vanclooster, A.; Maeyer, P. (2012): Combining Indoor and Outdoor Navigation: The Current Approach of Route Planners. In Gartner, G., Ortag, F. (Eds.): Advances in Location-Based Services. 8th International Symposium on Location-Based Services, Vienna 2011. 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 283–303.

Veblen, O.; Alexander, J. W. (1912): Manifolds of N Dimensions. In *The Annals of Mathematics* 14 (1/4), pp. p 163-178. Available online at http://www.jstor.org/stable/1967611.

Verbree, E.; Zlatanova, S. (2007): Positioning LBS to the third dimension. In Gartner, G., Cartwright, W., Peterson, M. (Eds.): Location Based Services and TeleCartography: Springer Berlin Heidelberg (Lecture Notes in Geoinformation and Cartography), pp. 107–118.

W3C (2004): Web Ontology Language (OWL). W3C Recommendation. World Wide Web Consortium. Available online at http://www.w3.org/TR/owl-features/.

Wallgrün, J. O. (2010): Hierarchical Voronoi Graphs. Spatial Representation and Reasoning for Mobile Robots. 1st ed. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Walton, L.; Worboys, M. (2009): An algebraic approach to image schemas for geographic space. In Stewart Hornsby, K., Claramunt, C., Denis, M., Ligozat, G. (Eds.): Spatial information theory. 9th international conference, COSIT 2009, Aber Wrac'h, France, September 21 - 25, 2009 ; proceedings. Berlin: Springer (Lecture Notes in Computer Science, 5756), pp. 357-370.

Whitehead, J. H. C. (1949): Combinatorial homotopy. In *Bulletin of the American Mathematical Society* 55 (5), pp. 453–496.

Whitney, H. (1965): Local properties of analytic varieties. In Cairns, S. S. (Ed.): Differential and Combinatorial Topology. A Symposium in Honor of Marston Morse: Princeton University Press, pp. 205–244.

Wilson, T. (2008): OGC KML. Open Geospatial Consortium (OGC Doc. No. 07-147r2). Available online at http://www.opengeospatial.org/standards/kml.

Wonka, P.; Wimmer, M.; Sillion, F.; Ribarsky, W. (2003): Instant architecture. In *ACM Transactions on Graphics (TOG)* 22 (3), pp. 669-677.

Worboys, M. (2011): Modeling indoor space. In Kulik, L., Güting, R. H., Lu, H. (Eds.): Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. New York, NY, USA: ACM (ISA '11), pp. 1-6.

Yang, L.; Worboys, M. (2011): A navigation ontology for outdoor-indoor space. In Kulik, L., Güting, R. H., Lu, H. (Eds.): Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. New York, NY, USA: ACM (ISA '11), pp. 31-34.

Ye, J.; Coyle, L.; Dobson, S.; Nixon, P. (2007): A Unified Semantics Space Model. In Hightower, J., Schiele, B., Strang, T. (Eds.): Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg (4718), pp. 103–120.

Yuan, W.; Schneider, M. (2010a): iNav: An Indoor Navigation Model Supporting Length-Dependent Optimal Routing. In Painho, M., Santos, M. Y., Pundt, H. (Eds.): Geospatial Thinking, vol. 0: Springer Berlin Heidelberg (Lecture Notes in Geoinformation and Cartography), pp. 299–313.

Yuan, W.; Schneider, M. (2010b): Supporting 3D route planning in indoor space based on the LEGO representation. In Winter, S., Jensen, C. S., Li, K.-J. (Eds.): Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness. New York, NY, USA: ACM, pp. 16-23.

Yuan, W.; Schneider, M. (2010c): Supporting Continuous Range Queries in Indoor Space. In: Proceedings of the Eleventh International Conference on Mobile Data Management (MDM), 2010. 23 - 26 May 2010, Kansas City, Missouri, USA. Kansas City, MO, USA, May 23-26. Institute of Electrical and Electronics Engineers. Piscataway, NJ: IEEE, pp. 209–214.

Yuan, W.; Schneider, M. (2011): 3D Indoor Route Planning for Arbitrary-Shape Objects. In Xu, J., Yu, G., Zhou, S., Unland, R. (Eds.): Database Systems for Adanced Applications, vol. 6637: Springer Berlin / Heidelberg (Lecture Notes in Computer Science), pp. 120–131.

Zlatanova, S. (2000): 3D GIS for urban development. PhD Thesis. Graz University of Technology, Graz, Austria. International Institute for Aerospace Survey and Earth Sciences (ITC); Institute for Computer Graphics and Vision (ICGV).

Zlatanova, S.; Rahman, A. A.; Shi, W. (2004): Topological models and frameworks for 3D spatial objects. In *Computers & Geosciences* 30 (4), pp. 419–428.
Available online at http://www.sciencedirect.com/science/article/pii/S0098300404000202.

All URLs in this thesis were checked on March 15, 2013.

Table of Figures

Table of Listings