

**Edda Steinmann**

**Räumlich explizite Funktionen  
als integrierter Bestandteil der Unternehmens-IT**

**München 2008**

---

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





**DGK** Deutsche Geodätische Kommission  
bei der Bayerischen Akademie der Wissenschaften

---

Reihe C

Dissertationen

Heft Nr. 624

**Räumlich explizite Funktionen  
als integrierter Bestandteil der Unternehmens-IT**

Zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften (Dr.-Ing.)  
vom Fachbereich Bauingenieurwesen  
der Universität Siegen  
genehmigte Dissertation

von

**Dipl.-Ing. Edda Steinmann, MSc**

**München 2008**

---

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

Adresse der Deutschen Geodätischen Kommission:



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>

Referent: Univ.-Prof. Dr.-Ing. Monika Jarosch  
1. Korreferent: Ao. Univ.-Prof. Mag. Dr. Josef Strobl  
2. Korreferent: Prof. Dr.rer.pol. Frank Schultmann

Tag der Einreichung: 23.11.2007

Tag der PRüfung: 12.06.2008

---

© 2008 Deutsche Geodätische Kommission, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet,  
die Veröffentlichung oder Teile daraus zu vervielfältigen

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>13</b>
1.1	Hintergrund	13
1.2	Forschungsfragen und Ziele	13
1.3	Aufbau der Arbeit	14
<b>2</b>	<b>Grundlagen Informationstechnologie von Unternehmen</b>	<b>16</b>
2.1	Geschäftsprozesse	16
2.1.1	Grundstruktur von Geschäftsprozessen	16
2.1.2	Bestandteile von Geschäftsprozessen	17
2.1.3	Arten von Geschäftsprozessen	18
2.2	Organisation und Aufgaben eines Energieversorgers	19
2.2.1	Strukturierung und Arten von Energieversorgungsunternehmen	20
2.2.2	Geschäftsprozesse bei Energieversorgungsunternehmen	21
2.2.3	Raumbezogene Geschäftsprozesse	22
2.3	Unternehmens-IT	23
2.3.1	Allgemeine Softwarelandschaft	23
2.3.2	ERP-Systeme	24
2.3.3	Geoinformationssysteme	27
<b>3</b>	<b>Grundlagen (Geo)Informatik</b>	<b>32</b>
3.1	Standards	32
3.1.1	Standardisierungsgremien	32
3.1.1.1	World Wide Web Consortium (W <sup>3</sup> C)	33
3.1.1.2	International Organization for Standardisation (ISO)	35
3.1.1.3	OpenGeospatial Consortium (OGC)	36
3.1.2	Auszeichnungssprachen	37
3.1.2.1	HyperText Markup Language (HTML)	38
3.1.2.2	eXtensible Markup Language (XML)	39
3.1.2.3	Geography Markup Language (GML)	44
3.2	Raumbezogene Datenhaltung	49
3.2.1	Oracle	49
3.2.2	Oracle Spatial	49
3.2.2.1	Geometrietypen	50
3.2.2.2	Logisches Datenmodell	51
3.2.2.3	Physisches Datenmodell	51
3.2.2.4	Räumliche Indizierung	52
3.2.3	Abfragemodell und -umgebung	52
3.3	Systemarchitekturen	56
3.3.1	Model Driven Architecture (MDA)	56
3.3.2	Service-oriented Architecture (SOA)	57
3.3.3	Bewertung	60
3.4	Web Services	62
3.4.1	Simple Object Access Protocol (SOAP)	64
3.4.2	Web Service Description Language (WSDL)	69
3.4.3	Universal Description, Discovery and Integration (UDDI)	73
3.4.4	Web Service Business Process Execution Language (WS-BPEL)	75
3.4.5	Ausblick von Web Services	80
3.5	Geo Web Services	81
3.5.1	Web Map Server (WMS)	82
3.5.2	Web Feature Server (WFS)	87
<b>4</b>	<b>Ansätze zur Nutzung des Raumbezugs</b>	<b>89</b>
4.1	Motivation zur Nutzung des Raumbezugs	89
4.2	Voraussetzungen zur Einbindung von GIS in Geschäftsprozesse	91
4.2.1	Daten und Datenstrukturen	91
4.2.2	Funktionalität	92

4.2.2.1	Netzverfolgung.....	93
4.2.2.2	Plotten .....	93
4.2.2.3	Sachdaten erfassen / editieren.....	94
4.2.2.4	Schnittstelle .....	95
4.2.2.5	Tourenplanung.....	95
4.2.2.6	Visualisierung .....	96
4.3	Ansätze zur Nutzung raumbezogener Informationen .....	97
4.3.1	Variante 1 - Schnittstelle.....	97
4.3.2	Variante 2 - Kopplung.....	98
4.3.3	Variante 3 - Integration .....	99
4.3.4	Variante 4 - dienstbasierter Ansatz.....	100
4.3.5	Bewertung der Varianten.....	100
4.4	DVGW-Referenzmodell für GIS-gestützte Geschäftsprozesse.....	101
4.4.1	Beschreibung des Prozessmodells Störfallmanagement .....	103
4.4.2	Beurteilung des Prozessmodells .....	111
4.5	Aktuelle Marktsituation für Störfallmanagementlösungen.....	112
4.6	Bewertung.....	113
<b>5</b>	<b>Umsetzung des gewählten Ansatzes .....</b>	<b>116</b>
5.1	Modifikation des Referenzmodells für Störfallmanagement.....	116
5.2	Unterteilung des Gesamtprozesses in Web Services .....	119
5.2.1	Unternehmensspezifische Services.....	123
5.2.2	Prozessspezifische Services .....	123
5.2.3	Human Task.....	124
5.3	Verwendete Software-Komponenten.....	124
5.3.1	Oracle Database 10g .....	125
5.3.2	Oracle SOA Suite 10g .....	126
5.3.3	Oracle Application Server 10g .....	127
5.3.4	Oracle Application Server 10g MapViewer .....	128
5.3.5	Oracle BPEL Process Manager.....	130
5.3.6	Oracle JDeveloper 10g.....	132
5.4	Verwendete Daten .....	133
5.4.1	Hauskoordinaten .....	134
5.4.2	Katasterinformationen .....	136
5.4.3	Leistungsdaten.....	136
5.5	Ablaufbeschreibung des Störungslokalisierungs-Service .....	137
5.5.1	Web Services .....	138
5.5.2	BPEL-Prozess.....	138
5.6	Erstellung der Web Services.....	138
5.6.1	Erstellung eines allgemeinen Web Service.....	139
5.6.2	Beschreibung des Adressen-Service.....	142
5.6.3	Beschreibung des WMS-Service .....	145
5.7	Orchestrierung der Web Services mit BPEL.....	146
5.7.1	Erstellung eines allgemeinen BPEL-Prozesses .....	147
5.7.2	Störungslokalisierung mit BPEL .....	151
5.8	Bewertung.....	155
<b>6</b>	<b>Evaluierung der Zielsetzungen .....</b>	<b>157</b>
6.1	Prototypische Realisierung eines Geschäftsprozesses mit Geodatennutzung .....	157
6.2	Wiederverwendbarkeit als Beitrag zur künftigen Nutzung .....	157
6.3	Tauglichkeit der verwendeten Standards .....	158
6.4	Prozessvorgabe durch übergeordnete Institutionen .....	160
6.5	Bewertung.....	160
<b>7</b>	<b>Abschlussbewertung und Ausblick .....</b>	<b>162</b>
<b>8</b>	<b>Literaturverzeichnis.....</b>	<b>164</b>
<b>Anhang A:</b>	<b>Funktionalitäten von GIS.....</b>	<b>169</b>
<b>Anhang B:</b>	<b>Beschreibung der existierenden Lösungen.....</b>	<b>173</b>
<b>Anhang C:</b>	<b>Code der Java-Klasse Adresse.java.....</b>	<b>185</b>

---

<b>Anhang D: Code der Java-Klasse WMS.java .....</b>	<b>190</b>
<b>Anhang E: Code der WSDL-Datei des Adressen-Service .....</b>	<b>191</b>
<b>Anhang F: Code der WSDL-Datei des WMS-Service.....</b>	<b>195</b>
<b>Anhang G: Code der WSDL-Datei des BPEL-Prozesses .....</b>	<b>199</b>
<b>Anhang H: Code der Schemadatei XSD zum BPEL-Prozess.....</b>	<b>200</b>
<b>Anhang I: Code des BPEL-Prozesses.....</b>	<b>201</b>
<b>Anhang J: UML-Diagramm des GAWANIS-Modells (Auszug) .....</b>	<b>206</b>

## Abbildungsverzeichnis

Abbildung 2.1: Struktur eines Geschäftsprozesses (Schwickert, Fischer, 1996).....	17
Abbildung 2.2: Paradigmenwechsel bei Energieversorgern (Schrenner, 2005).....	20
Abbildung 2.3: ERP- und ERP II-Komponenten (Theling, Loos, Sommerrock, 2005) .....	26
Abbildung 2.4: Unterschiede ERP und ERP II (Theling, Loos, Sommerrock, 2005) .....	27
Abbildung 2.5: GIS-Datenformen (GISZH, 2007).....	28
Abbildung 2.6: Übersicht und Detailansicht im GIS (Mettenmeier, 2007).....	29
Abbildung 2.7: GIS und CAD Modellvorstellung (GIUP, 2007).....	30
Abbildung 2.8: Darstellung von New Orleans in Google Earth (Soutschek, 2007) .....	31
Abbildung 3.1: Übersicht der Normierungsgremien .....	33
Abbildung 3.2: Der W <sup>3</sup> C Technology Stack (W <sup>3</sup> C, 2007) .....	34
Abbildung 3.3: Zusammenarbeit der Normierungsgremien (Kleber, 2006).....	37
Abbildung 3.4: Die Entwicklung der Auszeichnungssprachen (Mintert, 1998).....	38
Abbildung 3.5: Aufbau von HTML-Dokumenten.....	39
Abbildung 3.6: Übersicht der XML-Komponenten (Näf, 2007) .....	41
Abbildung 3.7: Beispiel einer XML-Datei.....	41
Abbildung 3.8: Beispiel einer DTD-Datei.....	42
Abbildung 3.9: Beispiel einer XSL-Datei .....	42
Abbildung 3.10: Darstellung einer XML-Datei in einem Web Browser.....	42
Abbildung 3.11: XML und verwandte Technologien nach (Noack, 2001) .....	43
Abbildung 3.12: Die GML-Komponenten im Zusammenhang .....	46
Abbildung 3.13: Graphische Darstellung eines Flurstücks .....	48
Abbildung 3.14: Repräsentation eines Flurstücks in GML.....	48
Abbildung 3.15: Geometrietypen in Oracle Spatial (Oracle, 2003).....	50
Abbildung 3.16: Logische Datenmodell von Oracle Spatial.....	51
Abbildung 3.17: Minimum Bounding Rectangle (MBR) (Oracle, 2003).....	52
Abbildung 3.18: Hierarchischer Index auf Basis von MBRs (Oracle, 2003).....	52
Abbildung 3.19: Räumliche Abfragemodell nach (Oracle, 2003).....	53
Abbildung 3.20: Räumliche Abfrage in GeoMedia Professional .....	54
Abbildung 3.21: Räumliche Abfrage in iSQL*PLUS .....	55
Abbildung 3.22: Ergebnissatz einer räumlichen Abfrage in iSQL*PLUS .....	55
Abbildung 3.23: Modellebenen der MDA nach (OMG, 2003) .....	56
Abbildung 3.24: SOA-Tempel (Dostal, et al., 2005) .....	59
Abbildung 3.25: Das SOA-Dreieck nach (Dostal, et al., 2005) .....	59
Abbildung 3.26: Entwicklung vom Mainframe zur SOA (Manhart, 2006).....	60
Abbildung 3.27: Service und Anwendung in UML nach (George, Westermann 2006)) .....	61
Abbildung 3.28: Ablauf des Generierungsprozesses nach (Richter, et al., 2005).....	62
Abbildung 3.29: Das Web-Service-Dreieck nach (Dostal, et al., 2005) .....	63
Abbildung 3.30: Web-Services fungieren als Abstraktionsschicht (Kiehle, 2006) .....	64
Abbildung 3.31: Web Service Architektur Stapel (Ferdinand, 2005).....	64
Abbildung 3.32: Verschiedene SOAP-Nachrichtenarten nach (Hauser, Löwer, 2004) .....	66
Abbildung 3.33: Aufbau einer SOAP-Nachricht (Stümpert, 2003) .....	67
Abbildung 3.34: SOAP-Nachricht als Anfrage für eine Reisereservierung (W <sup>3</sup> C, 2003) .....	68
Abbildung 3.35: SOAP-Nachricht als Antwort des Buchungssystems (W <sup>3</sup> C, 2003).....	68
Abbildung 3.36: Überblick über die WSDL-Komponenten (Dostal, et al., 2005).....	70
Abbildung 3.37: WSDL 2.0-Komponenten mit Hierarchie (W <sup>3</sup> C, 2006) .....	71
Abbildung 3.38: WSDL.2.0 Beispiel für einen Reservierungsservice (W <sup>3</sup> C, 2006) .....	72
Abbildung 3.39: WSDL-Komponenten im Vergleich (Bauer, 2006) .....	73
Abbildung 3.40: UDDI-Datenmodell (Schmietendorf, 2005) .....	74
Abbildung 3.41: Entwicklung der Geschäftsprozessmodellierung nach (Dostal, et al., 2005).....	76
Abbildung 3.42: Prozess-Orchestrierung (Juric, et al., 2006).....	77
Abbildung 3.43: Prozess-Choreographie (Juric, et al., 2006).....	77
Abbildung 3.44: Struktur eines BPEL Dokumentes nach (Ferdinand, 2005) .....	79
Abbildung 3.45: Definition eines BPEL-Dokumentes nach (Hansen, Neumann, 2005) .....	80

Abbildung 3.46: Prinzip der Kartenerstellung im Internet .....	82
Abbildung 3.47: XML-Antwort auf eine GetCapabilities Anfrage .....	84
Abbildung 3.48: Ergebnis einer GetMap Anfrage.....	86
Abbildung 3.49: Ergebnis einer GetFeatureInfo Anfrage.....	87
Abbildung 3.50: Funktionsweise des WFS.....	88
Abbildung 4.1: Funktion „Netzverfolgung“ nach (DVGW, 2003) .....	93
Abbildung 4.2: Funktion „Plotten“ nach (DVGW, 2003).....	94
Abbildung 4.3: Funktion „Sachdaten ändern/ editieren“ nach (DVGW, 2003) .....	94
Abbildung 4.4: Funktion „Schnittstelle“ nach (DVGW, 2003).....	95
Abbildung 4.5: Funktion „Tourenplanung“ nach (DVGW, 2003) .....	96
Abbildung 4.6: Funktion „Visualisierung“ nach (DVGW, 2003).....	97
Abbildung 4.7: Integrationstopologien (Bator, 2007) .....	99
Abbildung 4.8: Zuordnung von Haupt- und Teil- zu Kernprozessen nach (DVGW, 2003) ...	103
Abbildung 4.9: Darstellung des Ablaufes Störfallmanagement nach (DVGW, 2003).....	105
Abbildung 4.10: Schnittstellen vom/zum Störfallmanagement (DVGW, 2003) .....	109
Abbildung 4.11: Geschäftsprozess Störungs-Management (Hauffe, 2004).....	114
Abbildung 5.1: Ablauf des modifizierten Störfallprozesses.....	117
Abbildung 5.2: Zusammenhänge bei der Abarbeitung im Störfallmanagement.....	118
Abbildung 5.3: Ablauf modifiziertes Störungsmanagement mit Services.....	120
Abbildung 5.4: Tabellenansicht (Hauskoordinaten) in der Oracle-Datenbank .....	126
Abbildung 5.5: Architektur der Oracle SOA Suite (Oracle, 2006b) .....	127
Abbildung 5.6: Oracle Application Server mit laufenden Anwendungen.....	128
Abbildung 5.7: MapViewer Architektur (Oracle, 2005a) .....	129
Abbildung 5.8: Screenshot Map Builder-Tool.....	130
Abbildung 5.9: Oracle BPEL Process Manager (Oracle 2006c) .....	131
Abbildung 5.10: BPEL-Prozess mit Aufruf eines asynchr. Web Services (Oracle, 2006d) ..	132
Abbildung 5.11: BPEL-Projekt im JDeveloper.....	133
Abbildung 5.12: Visualisierung der erzeugten GIS-Daten in Geomedia .....	134
Abbildung 5.13: Tabellenansicht Hauskoordinaten in ORACLE .....	135
Abbildung 5.14: Oracle Map Builder mit visualisierten Hauskoordinaten .....	136
Abbildung 5.15: UML-Diagramm der Leitungsdaten (Auszug) .....	137
Abbildung 5.16: Komponenten im Zusammenspiel.....	139
Abbildung 5.17: Code-Beispiel für den „Hello World“-Web Service.....	139
Abbildung 5.18: Projektstruktur des "HelloService"-Web Service.....	140
Abbildung 5.19: Source-Dateien des "HelloService" -Web Service.....	140
Abbildung 5.20: Test und Ergebnis des "Hello World"-Web Service .....	141
Abbildung 5.21: Java-Klasse "Adresse" mit der "getHausnummer"-Methode.....	142
Abbildung 5.22: Java-Klasse "Adresse" mit der "getRechtswert"-Methode .....	143
Abbildung 5.23: Test der "getHausnummerzusatz"-Methode des Adressen-Service .....	144
Abbildung 5.24: Testergebnis der "getHausnummerzusatz"-Methode .....	144
Abbildung 5.25: Java-Klasse WMS mit Methoden zur Berechnung der Boundingbox.....	145
Abbildung 5.26: Java-Klasse WMS mit Methode "getMVURL" .....	145
Abbildung 5.27: Ergebnis des WMS-Service im InternetExplorer .....	146
Abbildung 5.28: Design des "HelloBPEL"-Prozesses.....	148
Abbildung 5.29: Code des "HelloBPEL"-Prozesses .....	148
Abbildung 5.30: Synchroner Beispielprozess "HelloBPEL" .....	149
Abbildung 5.31: Test des "HelloBPEL"-Prozesses.....	150
Abbildung 5.32: Ergebnis des "HelloBPEL"-Prozesses.....	150
Abbildung 5.33: Auditing des "HelloBPEL"-Prozesses.....	151
Abbildung 5.34: Ablauf der Lokalisierung in BPEL.....	152
Abbildung 5.35: Aufruf der Testinstanz zum Lokalisierungs-Prozess.....	153
Abbildung 5.36: SOAP-Nachricht mit URL als Ergebnis des Tests .....	154
Abbildung 5.37: Ergebnis-URL als Graphik im Browser.....	155

## Tabellenverzeichnis

Tabelle 2.1: (Geschäfts-)Prozessarten nach (Schwickert, Fischer, 1996) .....	19
Tabelle 2.2: Kundenbezogene Prozesse nach (Schrenner, 2005) .....	21
Tabelle 2.3: Betriebsmittelbezogene Prozesse (Schrenner, 2005).....	22
Tabelle 2.4: Anwendungssoftware bei EVU (Schrenner, 2005) .....	24
Tabelle 3.1: Stufenübersicht einer Empfehlung beim W <sup>3</sup> C (Kleber, 2006) .....	35
Tabelle 3.2: Stufenübersicht einer Normierung bei ISO .....	35
Tabelle 3.3: Bezug von Informatikstandards zu GML .....	44
Tabelle 3.4: Entwicklungsstufen der GML.....	45
Tabelle 3.5: Erläuterung der Grundbegriffe von GML nach (Pritschet, 2005).....	46
Tabelle 3.6: Zuordnung der Daten auf die Datenstruktur bei UDDI (Ferdinand, 2005) .....	75
Tabelle 3.7: Zeichen einer WMS-Anfrage nach (OGC, 2002) .....	84
Tabelle 3.8: Parameter der Operation GetCapabilities nach (OGC, 2002).....	84
Tabelle 3.9: Parameter der Operation GetMap nach (OGC, 2002) .....	85
Tabelle 3.10: Parameter der Operation GetFeatureInfo nach (OGC, 2002) .....	86
Tabelle 4.1: Nutzenbetrachtung durch Datenkonsistenz nach (Schrenner, 2005).....	90
Tabelle 4.2: Nutzenbetrachtung durch Datenmehrwert nach (Schrenner, 2005).....	90
Tabelle 4.3: Nutzenbetrachtung bei durchgängigen Workflows nach (Schrenner, 2005) .....	91
Tabelle 4.4: Beschreibung des Ablaufes Störfallmanagement nach (DVGW, 2003) .....	108
Tabelle 4.5: Erläuterung der Schnittstellen nach DVGW (2003) .....	110
Tabelle 5.1: Beschreibung der Komponenten im modifizierten Störfallmanagement.....	122
Tabelle 5.2: Struktur eines BPEL-Projektes im BPEL-Designer .....	147

## Abkürzungsverzeichnis

ALK	Automatisierte Liegenschaftskarte
ANSI	American National Standards Institute
API	Application Programmer Interface
AVA	Ausschreibung, Vergabe, Abrechnung
BEEP	Blocks eXtensible Exchange Protocol
BPEL	Business Process Execution Language
BPQL	Business Process Query Language
BTP	Business Transaction Protocol
CAD	Computer Aided Design
CASE	Computer Aided Software Engineering
CD	Committee Draft
CEN	European Committee for Standardisation
CENELEC	Europ. Komitee für elektrotechn. Normung
CIM	Computing Independent Model
CORBA	Common Object Request Broker Architecture
CR	Candidate Recommendation
CRM	Customer Relationship Management
CSW	Web Catalog Service
DBMS	Datenbankmanagementsystem
DCOM	Distributed Component Object Model
DIN	Deutsches Institut für Normung e.V.
DIS	Draft International Standard
DMS	Dokumentenenmanagementsoftware
DOM	Document Object Model
DTD	Document Type Definition
DVGW	Deutscher Verein des Gas- und Wasserfaches e.V.
DXF	Drawing Exchange Format
EAI	Enterprise Application Integration
EDBS	Einheitliche Datenbankschnittstelle
EJB	Enterprise Java Beans
ENWG	Energiewirtschaftsgesetz
EPSG	European Petroleum Survey Group Geodesy
ESB	Enterprise Service Bus
ERP	Enterprise Resource Planning
EVU	Energieversorgungsunternehmen
FCIT	Fichtner Consulting und IT AG
FDIS	Final Draft International Standard
FOI	Feature of Interest
FTP	File Transfer Protocol
GAWANIS	GA- und Wasser Netz- Informations System
GDI	Geodateninfrastrukturen
GIF	Graphics Interchange Format
GIS	Geographisches Informationssystem
GML	Geography Markup Language
GUI	Graphical User Interface
GDV	Graphische Datenverarbeitung
HTTP	Hypertext Transport Protocol
HTML	Hypertext Markup Language
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IPC	Inter Process Communication
IRI	Internationalized Resource Identifier

IS	International Standard
ISE	Integrated Service Invironment
ISO	International Organization for Standardisation
IT	Informationstechnologie
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
JPEG	Joint Photographics Expert Group
MBR	Minimum Bounding Rectangle
MDA	Modell Driven Architecture
MIT	Massachusets Institute of Technology
MOM	Message Oriented Middleware
NABau	Normenausschuss Bauwesen
NIS	Netzinformationssystem
NLS/PLS	Netz- und Prozessleittechnik
NWIP	New Work Item Proposal
OC4J	Oracle Container for Java
ODBC	Open Database Connectivity
OGC	Open Geospatial Consortium
OMG	Object Management Group
ON	Österreichisches Normungsinstitut
ÖPNV	Öffentlicher Personennahverkehr
OWS	OGC Web Services
PIM	Platform Independent Model
PNG	Portable Network Graphics
PR	Proposed Recomendation
PSM	Platform Spezific Model
P/L	Procedural Language oder Processing Language
RDF	Resource Description Framework
REC	Recommendation
REST	Representational State Transfer
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RFC	Request for Comment
RFC	Remote Function Call
SC	Subkomitee
SCADA	Supervisory Control and Data Acquisition
SDO	Spatial Data Object
SGML	Standard Generalized Markup Language
SLA	Service-Level-Agreement
SMTP	Simple Mail Transfer Protocol
SNV	Schweizerische Normen-Vereinigung
SOA	Service-orientierte Architektur
SOAP	Simple Object Access Protocoll (ab V1.2 kein Akronym mehr)
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TC	Technisches Komitee
TCP	Transmission Control Protocol
THP	Tentative Hold Protocol
TIFF	Tagged Image File Format
UDDI	Universal Description, Discovery and Integration Protocol
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W <sup>3</sup> C	World Wide Web Consortium
WCS	Web Coverage Service

---

WCTS	Web Coordinate Transformation Service
WD	Working Draft
WFS	Web Feature Service
WFS-G	Web Gazetteer Service
WMS	Web Map Service
WS	Web Service
WS-BPEL	siehe BPEL
WS-I	Web Service Interoperability Organisation
WSDL	Web Service Description Language
WTS	Web Terrain Service
WWW	World Wide Web
XHTML	eXtensible Hypertext Markup Language
XLink	eXtensible Linking Language
XML	eXtensible Markup Language
XPath	eXtensible Path Language
XPointer	eXtensible Pointer Language
XQL	eXtensible Query Language
XSD	XML Schema Definition
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformation

## Zusammenfassung

Schwerpunkt im Bereich der Geoinformatik war für Energieversorger bis vor kurzem nahezu ausschließlich die reine Bestandsdatenerfassung und Auswertungen der Daten durch beispielsweise eine topologische Abfrage im Geographischen Informationssystem (GIS). Bisher wurde, rein technologisch betrachtet, die Geoinformatik von der allgemeinen Informatik, die in diesem Zusammenhang für sämtliche Informationstechnologie (IT) ohne Raumbezug steht, getrennt gehandhabt. So führen GIS oft noch ein Inseldasein, bzw. werden über unflexible Schnittstellenlösungen mit anderen Systemen gekoppelt. Die bisherigen Kopplungen sind von mehreren Abhängigkeiten hinsichtlich Technologie, Software, Prozesse oder unternehmensspezifischen Aspekten geprägt. Aufgrund des Paradigmenwechsel von isolierten Einzelsystemen zu geschäftsprozessorientierten Architekturen ist es erforderlich, die geographischen Informationen in die aktuelle Softwarearchitektur der IT zu integrieren. Da die beiden Richtungen der Informatik bisher nur wenige Gemeinsamkeiten hinsichtlich definierter Standards haben, stellt sich die Frage, wie eine technische Anbindung von Geodaten in Geschäftsprozessen aussehen kann. Zu einer Realisierung einer solchen Anbindung werden Vorschläge erarbeitet. Die Web Service-Technologie in einer Service-orientierten Architektur (SOA) wird aufgrund ihrer hohen Flexibilität und Wiederverwendbarkeit sowie der Verwendung von Kernstandards als ideale Systemumgebung herausgestellt.

Der Deutsche Verein des Gas- und Wasserfaches e.V. (DVGW) hat eine Empfehlung zur Nutzung von Geodaten in Geschäftsprozessen ausgesprochen und diese mit einem Referenzmodell sowie dem Technischen Hinweis zu DV-gestütztem Störfallmanagement und Schadenstatistik unter Einbindung von GIS untermauert. Der Hauptprozess Störungsmanagement aus dem Kernprozess Betreiben und Instandhalten bildet die fachliche Basis für weitere Untersuchungen zum Prozessablauf. Es erfolgt eine prototypische Umsetzung der Teilprozesse mit Geobezug des Gesamtprozesses Störfallmanagement in der herausgearbeiteten Systemumgebung.

Anhand des Prototyps lassen sich Auswertungen erstellen, inwieweit die auf beiden Seiten existierenden Standards die gewünschten Anforderungen erfüllen. Obwohl die teilweise doch recht jungen Standards noch reifen müssen, ist der untersuchte Prozess derzeit realisierbar. Des Weiteren ist die Web Service-Technologie in hohem Maße von Wiederverwendbarkeit geprägt, die im konkreten Zusammenhang mit den für das Störungsmanagement entwickelten Web Services analysiert wird.

Es wird ebenso herausgearbeitet, inwieweit eine Prozessablaufvorgabe durch eine übergeordnete Institution, wie beispielsweise den DVGW, erfolgen kann. So finden sich zwar die Prozesse bei allen Energieversorgungsunternehmen (EVU) wieder, jedoch gibt es zu viele unternehmensspezifische Details, die nur eine oberflächliche Gesamtvorgabe durch Verbände sinnvoll erscheinen lassen.

Diese Arbeit zeigt anhand der prototypischen Umsetzung eines Teilprozesses mit Raumbezug des Gesamtprozesses Störungsmanagement, dass sich durch die Abbildung raumbezogener Geschäftsprozesse in Web Services und deren Einbindung in eine Service-orientierte Architektur ein Mehrwert hinsichtlich Flexibilität und einer Annäherung beider Informatik-Disziplinen erzielen lässt.

# 1 Einleitung

## 1.1 Hintergrund

Vor einigen Jahren stand für Energieversorgungsunternehmen im Bereich der Verarbeitung raumbezogener Informationen nahezu ausschließlich die digitale Erfassung von Bestandsdaten im Fokus, wenn auch diese oftmals nur eine Umstellung von einer analogen zur digitalen Datenführung bedeutete. Diese Datenerfassung für die Bestandsdokumentation ist bei den meisten Unternehmen mittlerweile abgeschlossen, und es können Nutzungspotentiale der erfassten Geodaten erschlossen werden. Hierin begründet sich der Paradigmenwechsel von isolierten Einzelsystemen zu Geschäftsprozess-orientierten Architekturen mit dem Trend, die erfassten Geodaten in die Prozesse des Energieversorgers einzubinden und damit einen weiteren Mehrwert zu erzeugen.

So wird auch von Verbänden wie beispielsweise dem DVGW (Deutscher Verein des Gas- und Wasserfaches e.V.) die Geschäftsprozessunterstützung bei Energieversorgungsunternehmen durch Geographische Informationssysteme empfohlen. Sowohl Softwarehersteller als auch die Unternehmen haben entsprechende Konzepte und Lösungen entwickelt, um diese Anforderungen umzusetzen. Die Ergebnisse zeigen starre Lösungen, die entweder an bestimmte Unternehmen, Software, Prozesse oder Technologien gebunden sind. Dieser Zustand ist von Abhängigkeiten gekennzeichnet und wirkt sich nicht positiv auf die Durchführung von raumbezogenen Geschäftsprozessen aus. Durch die benannten Abhängigkeiten gehen oftmals hohe finanzielle, zeitliche und personelle Aufwände hinsichtlich der Erstellung und Wartung der Lösungen einher. Diese Aufwände stehen wiederum im Gegensatz zu den Unternehmenszielen, die durch die optimale Abwicklung von Geschäftsprozessen erreicht werden sollen.

## 1.2 Forschungsfragen und Ziele

Vor dem geschilderten Hintergrund ergeben sich verschiedene Forschungsfragen. Die daraus resultierenden Ziele werden am Ende des Kapitels zusammengefasst.

Für den beispielhaften Geschäftsprozess Störungsmanagement in einem Energieversorgungsunternehmen bestehen derzeit keinerlei Erfahrungen bezüglich der praktischen Umsetzung innerhalb einer Softwarearchitektur, die die Abarbeitung von Geschäftsprozessen in den Vordergrund stellt. An dieser Stelle kann die Forschung einen Beitrag leisten, inwiefern sich theoretisch existierende Konzepte für Integrationen von verschiedenen Daten und Programmen in aktuellen Softwarearchitekturen umsetzen lassen.

***Forschungsfrage: Wie lassen sich theoretisch existierende Konzepte für Integrationen von verschiedenen Daten und Programmen in aktuellen Softwarearchitekturen umsetzen?***

Mit dem angesprochenen Paradigmenwechsel wurden mehr und mehr Kernstandards für die Softwaretechnologie entwickelt. Diese Entwicklung wurde jedoch von getrennten Gremien begleitet, die einerseits die allgemeine Informatik und andererseits die Geoinformatik als Schwerpunkt hatten. So stoßen schließlich zwei inkompatible Technologien im Rahmen der Geschäftsprozessorientierung aufeinander. Eine Aufgabe dieser Arbeit ist die Untersuchung, wie eine ausschließlich standardbasierte Softwarearchitektur, die eine Nutzung von räumlich expliziten Funktionen ermöglichen soll, gestaltet sein kann und inwiefern sie in der Lage ist, die geschilderte Situation deutlich zu verbessern. Des Weiteren ist es in diesem Zusammenhang notwendig, eine Aussage zur Nutzung vorhandener Standards hinsichtlich ihrer Möglichkeiten und Grenzen zur Abbildung von Geschäftsprozessen zu formulieren.

***Forschungsfragen: Wie kann eine ausschließlich standardbasierte Softwarearchitektur, die eine Nutzung von Geodaten ermöglichen soll, gestaltet sein? Kann eine Geschäftsprozessoptimierung durch diese Softwarearchitektur erfolgen? Ist es möglich, einen Geschäftsprozess rein über die Nutzung vorhandener Standards abzubilden oder gibt es Grenzen?***

Da die bislang existierenden Lösungen für Geschäftsprozesse bei Energieversorgungsunternehmen von verschiedenen Faktoren abhängen und somit eine 1:1-Implementierung von Lösung A bei Energieversorger B keinesfalls möglich ist, liegt es nahe, eine Umsetzungsvariante zu finden, die ein großes Maß an Wiederverwend-

barkeit und Flexibilität aufweist. Die Web Service-Technologie, die aktuell zu diesem Thema in der Fachwelt diskutiert wird, bringt grundsätzlich die geforderten Eigenschaften mit. Es gilt zu beurteilen, wie sich Flexibilität und Wiederverwendbarkeit von definierten Prozessen mit Geodatenbezug innerhalb sowie außerhalb des Unternehmens verhalten. Diese Eigenschaften gilt es sowohl in in sich abgeschlossenen Gesamtprozessen sowie einzelnen Teilprozessen zu untersuchen. Auch zu diesen Fragestellungen kann die Forschung einen Beitrag leisten, da es in dieser Hinsicht noch keinerlei Erfahrungen von Prozessen mit Geodaten in Zusammenhang mit der Wiederverwendbarkeit von Web Services gibt.

***Forschungsfragen: Wie verhält sich die Web Service-Technologie hinsichtlich Flexibilität und Wiederverwendbarkeit von definierten Prozessen mit Geodatenbezug? Gilt dieses Verhalten sowohl für Teil- als auch für Gesamtprozesse?***

Da die Geschäftsprozessunterstützung durch Geographische Informationssysteme bei Energieversorgungsunternehmen beispielsweise vom DVGW empfohlen wird, gilt es zu hinterfragen, inwieweit in sich geschlossene Abläufe oder gar unternehmensweite Prozesse durch übergeordnete Institutionen zur Nutzung vorgegeben werden können. Eine einheitliche Nutzung von Geodaten in Geschäftsprozessen ist ebenfalls ein Gebiet, in dem die Wissenschaft Ergebnisse beisteuern kann.

***Forschungsfrage: Kann eine einheitliche Nutzung von Geodaten in Prozessen durch eine übergeordnete Institution vorgegeben werden?***

Die aus den Forschungsfragen abgeleiteten Ziele der Arbeit lassen sich wie folgt zusammenfassen:

1. Entwicklung und prototypische Realisierung eines Geschäftsprozesses mit Geodatennutzung mittels existierender Standards in einer aktuellen Softwarearchitektur
2. Beitrag zur künftigen Nutzung von räumlich expliziten Funktionen in Geschäftsprozessen
3. Evaluierung der Tauglichkeit von verfügbaren Standards und Normen für die Geodatennutzung in Geschäftsprozessen
4. Entwicklung eines übergeordneten, technischen Konzeptes für die Nutzung von Geodaten in ausgewählten Geschäftsprozessen

### **1.3 Aufbau der Arbeit**

Die vorliegende Arbeit gibt zunächst einen Einblick in die Denkweise und den Aufbau von Geschäftsprozessen im Allgemeinen. In Kapitel 2 wird ein Überblick über die Aufgaben, die von Energieversorgungsunternehmen bewältigt werden müssen und welche Softwarekomponenten zur Bearbeitung zur Verfügung stehen, gegeben. Das darauf folgende Kapitel 3 dient der Grundlagenbetrachtung, die für die weiteren Untersuchungen notwendig ist. Neben der Vorstellung der relevanten Basisstandards (HTML, XML und GML) sowie die dafür zuständigen Gremien (W<sup>3</sup>C, ISO, OGC) wird die raumbezogene Datenhaltung erläutert. Ein Überblick über die gesamte IT-Landschaft liefert die Beschreibung von zwei Softwarearchitekturen, der modellgetriebenen Architektur (MDA) und der Service-orientierten Architektur (SOA). Die Web Service Technologie in einer Service-orientierten Architektur, die derzeit aufgrund ihrer Plattform- und Programmiersprachenunabhängigkeit als Optimum einer IT-Landschaft diskutiert wird, wird mit ihren Kernstandards (SOAP, WSDL, UDDI) und der Prozesssteuerungssprache WS-BPEL (abgekürzt BPEL) vorgestellt. Ebenso werden die Geo Web Service anhand der WMS- und WFS-Spezifikationen betrachtet. Diese Grundlagen dienen der technischen Betrachtung einer GIS-Anbindung in die Unternehmens-IT. In Kapitel 4 werden die Ansätze zur Nutzung des Raumbezugs in Geschäftsprozessen analysiert. Zu Beginn werden die Voraussetzungen für die Einbindung von geographischen Informationen in Geschäftsprozesse definiert. Daraufhin wird untersucht, welche generellen Möglichkeiten zur Integration von GIS-Systemen und Geodaten in die Unternehmens-IT bestehen. Zur fachlichen Diskussion dient das Referenz-

modell für GIS-gestützte Geschäftsprozesse des DVGW, welches das einzige seiner Art ist. Als weitere inhaltliche Grundlage bietet sich der Technische Hinweis GW 133 zum DV-gestützten Störfallmanagement und Schadensstatistik unter Einbindung von GIS an, welcher auf zuerst genanntem Referenzmodell beruht. Zum Zeitpunkt der Arbeit erhältliche Lösungen für den Bereich Störfallmanagement werden aufgeführt und in die vorhergehende Analyse eingeordnet. Es wird ein Vorschlag zur Realisierung erarbeitet. Das Kapitel 5 beschäftigt sich mit der prototypischen Umsetzung des Hauptprozesses Störfallmanagement des Kernprozesses Betreiben und Instandhalten, wobei zunächst eine Modifikation des Modells vorgenommen und diese dann in das gewählte Konzept eingearbeitet wird. Es folgt eine Umsetzung in einer ORACLE-Software-Umgebung. Die für die Abbildung des Störfallmanagements notwendigen Daten werden erfasst und dienen als Grundlage. Die Funktionalität für die Störungslokalisierung wird sowohl in geographischen als auch nichtgeographischen Web Services abgebildet, die über die Prozesssteuerungssprache BPEL orchestriert werden. Die Ergebnisse werden hinsichtlich der eingangs formulierten Ziele zur Wiederverwendbarkeit und zum Einsatz von Standards untersucht. Die Gesamtbewertung und ein Ausblick schließen diese Arbeit ab.

## 2 Grundlagen Informationstechnologie von Unternehmen

Die steigende Produktkomplexität, die zunehmende Marktdynamik und der wachsende Einfluss des Kunden auf die Gestaltung der Produkte sind nach (Scheer, Werth, 2005) nur einige neue Einflussfaktoren, die Unternehmen vor die Herausforderung stellen, schnell auf Markt- und Nachfrageänderungen reagieren zu können. Betriebswirtschaftliche Anforderungen aus den Märkten haben zu der Entwicklung flexibler Infrastrukturen beigetragen. Insbesondere sind Software-Architekturen zu nennen, die erstmals dem Geschäftsprozess-Paradigma folgen. Im Folgenden wird die Informationstechnologie von Unternehmen (Unternehmens-IT) insbesondere von Energieversorgungsunternehmen samt deren Arbeitsabläufen näher betrachtet. Das Kapitel beginnt mit grundlegenden Denkweisen über Geschäftsprozesse in der Theorie, betrachtet deren Umsetzung inklusive Raumbezug bei Energieversorgungsunternehmen und zeigt die zugehörige EDV-Landschaft eines mittelständischen Unternehmens.

### 2.1 Geschäftsprozesse

Während sich die Unternehmen in vergangenen Jahren mit der effizienten Ausführung und lokalen Optimierung von einzelnen Funktionsbereichen beschäftigt haben, ist der Gesamtzusammenhang der in der IT abgebildeten und betriebenen Systeme in den Hintergrund gerückt. Es entstanden einzelne „Silo-Applikationen“, die für einen steigenden Aufwand und höhere Kosten zur Abstimmung und Koordination zwischen Fachabteilungen und IT-Systemen verantwortlich waren. Zur Überwindung dieses Dilemmas wurden die Geschäftsprozesse in den Mittelpunkt gestellt. So sind seit Mitte der 90er Jahre Geschäftsprozessmanagement oder Business Process Management sowohl für die Geschäftsführung als auch für die Fachbereiche und die IT gleichermaßen wichtige Themen. Denn faktisch können IT und die Prozesse nicht mehr voneinander getrennt werden.

Dieses Unterkapitel setzt sich explizit mit der Thematik der Geschäftsprozesse auseinander. Es beinhaltet die Definition von Geschäftsprozessen, ihren Aufbau und erläutert die Prozesse bei Energieversorgungsunternehmen und dort im Speziellen die Darstellung des Raumbezugs.

#### 2.1.1 Grundstruktur von Geschäftsprozessen

Da der Begriff des Geschäftsprozesses im Laufe der Zeit stark an Bedeutung gewonnen hat, ist es notwendig, ihn genauer zu definieren. In der Literatur (Schwickert, Fischer, 1996) gibt es eine Vielzahl von zum Teil auch sehr unterschiedlichen Definitionen für den Geschäftsprozess, in denen jeweils andere Komponenten hervorgehoben werden. Es gibt jedoch die folgenden Gemeinsamkeiten.

Eine strukturierte und gesteuerte Folge von Aktivitäten durch Menschen oder Maschinen ausgeführt, die angibt, wie aus einem vorgegebenen Input ein bestimmtes Ergebnis erzeugt werden soll, wird als Prozess bezeichnet. Um ablaufen zu können, benötigen Prozesse immer eine bestimmte Infrastruktur (IT-Systeme, Fahrzeuge zum Transport, Maschinen zur Verarbeitung,...). Im Detail heißt dies, dass der Ablauf des Geschäftsprozesses durch vorgegebene Regeln, die die Vorgehensweise eingrenzen oder gar bestimmen, festgelegt wird. Die einzelnen Teilprozesse, Tätigkeiten bzw. Funktionen, die im Rahmen des Geschäftsprozesses ausgeführt werden sollen, werden durch diese Regeln definiert. Der Geschäftsprozess ergibt also eine logisch zusammenhängende Kette, in Form einer Kombination von materiellen oder immateriellen Einsatzgütern (Input), die nach oben genannten Regeln zu Arbeitsergebnissen (Leistung/Output) bearbeitet werden. (Schwickert, Fischer, 1996) In der Abbildung 2.1: Struktur eines Geschäftsprozesses (Schwickert, Fischer, 1996) Abbildung 2.1 wird die Grundstruktur des Geschäftsprozesses aufgezeigt.

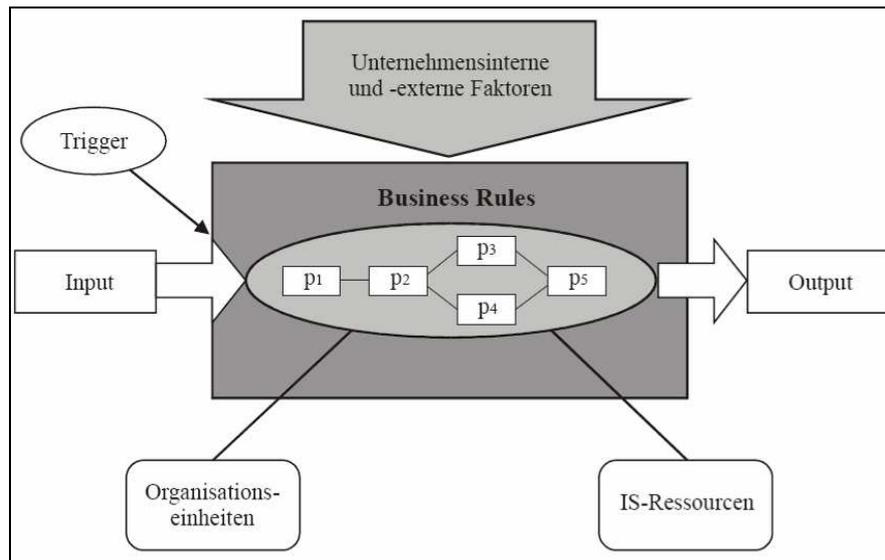


Abbildung 2.1: Struktur eines Geschäftsprozesses (Schwickert, Fischer, 1996)

Sollen reale Unternehmen in Prozesse gliedert werden, so wird deutlich, dass es tendenziell sehr viele „Prozesse“ im allgemeinen Sinne geben wird. Daher werden Prozesse in geeigneten Schemata kategorisiert. Bewährt hat sich die Gliederung in kleinere Prozesse, so genannte Unterprozesse oder Subprozesse. Folglich ist nachfolgende Gliederung sinnvoll:

### Kernprozess – Hauptprozess – Teilprozess – Aktivität

Aus Prozesssicht ist die Aktivität die kleinste und der Kernprozess die größte Einheit in der genannten Gliederung. Selbst für jeden Teilprozess oder jede Aktivität müssen alle vordefinierten Rahmenbedingungen, die einen direkten Einfluss auf den Ablauf haben, beachtet werden, damit ein Geschäftsprozess ablaufen kann. Dies sind zum einen unternehmensinterne Faktoren. Da sich jeder Geschäftsprozess entlang der Wertschöpfungskette und den Unternehmenszielen orientiert, wird er nach seinem Beitrag zur Erreichung dieser Ziele beurteilt. Zum anderen werden unternehmensexterne Faktoren gleichermaßen als Rahmenbedingungen berücksichtigt, da Einflüsse und Bedingungen, die von außen auf das Unternehmen einwirken wie beispielsweise Gesetze, ebenso auf den Ablauf des Prozesses einwirken.

## 2.1.2 Bestandteile von Geschäftsprozessen

Im Folgenden werden die einzelnen Komponenten eines Geschäftsprozesses dargestellt. Es handelt sich um eine weitgreifende Definition der Bestandteile, um alle Charakteristika eines geschäftlichen Ablaufes zu erfassen. Nach (Schwickert, Fischer, 1996) sind die nachfolgend aufgeführten Fragmente die wesentlichen Bestimmungselemente eines Geschäftsprozesses.

- Teilprozesse oder Aktivitäten,
- Subjekte oder Aufgabenträger,
- das Objekt,
- Input,
- Arbeitsmittel,
- logische, zeitliche und räumliche Beziehungen.

Teilprozesse sowie Aktivitäten, die den kleinsten Teilprozess im Gesamtablauf bilden, stellen die Teile dar, die zusammengesetzt den gesamten Prozess ergeben und somit auch zueinander in Beziehung stehen.

Ausführer der Aufgaben (Aufgabenträger oder Subjekte) sind die Ausführungsorgane der einzelnen Aktivitäten. Traditionell handelt es sich um Menschen, in der heutigen Zeit werden jedoch vermehrt Arbeiten von Maschinen wie beispielsweise Computer übernommen.

Beim Objekt des Geschäftsprozesses handelt es sich um einen materiellen oder immateriellen Gegenstand, der eben besagten Geschäftsprozess durchläuft.

Der Input für einen Geschäftsprozess kann ebenso wie das Objekt aus materiellen oder Immateriellen sowie aus einer Kombination von beiden bestehen. Beispiel für einen immateriellen Input sind Informationen, die für die Abwicklung von derart entscheidender Bedeutung sind, dass der Prozess sonst nicht in definierter Weise ablaufen kann. Selbstverständlich gilt diese Aussage auch für den materiellen Input wie beispielsweise Rohstoffe.

Arbeitsmittel haben in diesem Zusammenhang eine zweifache Bedeutung. Zum einen dienen sie dem Menschen dazu, seine Arbeit auszuführen. Hier kann es sich um Informationstechnologien handeln, die als Arbeits- oder Entscheidungshilfe eingesetzt werden. Zum anderen können sie gleichzeitig auch Aufgabenträger sein.

In Geschäftsprozessen existieren verschiedene Arten von Beziehungen, logischer, zeitlicher und räumlicher Art. Die Ablauflogik erfasst die genaue Reihenfolge der Teilprozesse im Gesamtprozess. Dadurch wird deutlich, welche Aktivitäten parallel, nebenläufig oder sequentiell ausgeführt werden können. Die Zeit bezeichnet die Durchlaufzeit, d.h. die Dauer vom Start bis zum Ende des Prozesses. Sie kann weiterhin in Bearbeitungszeit, Transfer- oder Transportzeit sowie Liegezeit unterschieden werden. Im Zusammenhang mit der Aufbauorganisation stehen die räumlichen Beziehungen des Geschäftsprozesses. Aufgabenträger, die innerhalb des Prozesses mit Arbeitsmitteln tätig werden, ordnet man im Unternehmen auch bestimmten Abteilungen zu, auf die im Prozessablauf Rücksicht genommen werden muss.

Nach der Erläuterung der Basiskomponenten lässt sich abschließende, verfeinerte Definition für den Begriff „Geschäftsprozess“ nach (Schwickert, Fischer, 1996) zitieren.

*„Der Prozess ist eine logische zusammenhängende Kette von Teilprozessen, die auf das Erreichen eines bestimmten Zieles ausgerichtet sind. Ausgelöst durch ein definiertes Ereignis wird ein Input durch den Einsatz materieller und immaterieller Güter unter Beachtung bestimmter Regeln und der verschiedenen unternehmensinternen und –externen Faktoren zu einem Output transformiert. Der Prozess ist in ein System von umliegenden Prozessen eingegliedert, kann jedoch als selbstständige, von anderen Prozessen isolierte Einheit, die unabhängig von Abteilungs- und Funktionsgrenzen ist, betrachtet werden.“*

### 2.1.3 Arten von Geschäftsprozessen

Nach der Definition des Prozessbegriffs wird in diesem Abschnitt der Frage nachgegangen, inwieweit sich Geschäftsprozesse in verschiedene Arten unterscheiden lassen. Es wird eine allgemeine Einteilung von Prozessen vorgenommen, die im folgenden Kapitel für Energieversorgungsunternehmen weiter spezifiziert wird.

Zunächst wird nach unterschiedlichen Organisationseinheiten, die in einem Geschäftsprozess mitwirken, untersucht. Es ergibt sich nach (Schwickert, Fischer, 1996) die Einteilung in folgende drei Bereiche, die nach der Beschreibung tabellarisch aufgelistet werden:

- Unternehmensübergreifende Geschäftsprozesse, die zwischen zwei oder mehreren Unternehmen ablaufen, so dass der Output die Unternehmensgrenze überschreitet.
- Funktionsübergreifende Geschäftsprozesse, die nur unternehmensintern statt- und Verwendung finden.

- Stellenübergreifende Geschäftsprozesse, die innerhalb einer Abteilung oder eines Unternehmens ablaufen, aber keinem höheren Prozess zuzuordnen sind.

Eine weitere Unterscheidung nach (Schwickert, Fischer, 1996) kann in der Art des Gestaltungsobjekts, das durch den Geschäftsprozess transformiert wird, erfolgen. Es werden hier zwei Bereiche unterschieden, die in der realen Welt jedoch nicht eindeutig zugeordnet werden können, da jeder materielle Prozess von Informationen begleitet wird oder selbst als solcher interpretiert werden kann.

- Durch einen materiellen Geschäftsprozess wird ein physisches Objekt erschaffen oder verändert. Hierzu zählen alle raum-zeitlichen Beziehungen.
- Bei informationellen Geschäftsprozessen werden Informationen erschaffen oder manipuliert. So stehen mentale Handlungen im Mittelpunkt von in Raum und Zeit fortschreitenden Abläufen. Der Ausgangspunkt solcher Prozesse wird von der Erfüllung von Informationsverarbeitungsaufgaben gebildet.

Die in diesem Kapitel letzte Einteilung von Geschäftsprozessen wird anhand der Art der Tätigkeiten, die innerhalb des Prozesses ausgeübt werden, vorgenommen. (Schwickert, Fischer, 1996)

- Wenn die Durchführung der Prozesse direkt mit der Erreichung des Unternehmenszieles verbunden ist, wird dieser als operativer oder primärer Geschäftsprozess bezeichnet.
- Ein Steuerungsprozess, auch Sekundärprozess genannt, hingegen, der nur die Ressourcen zur Durchführung der operativen Prozesse koordiniert, ist somit nur indirekt an der Zielerfüllung beteiligt.

In der folgenden Tabelle sind die erläuterten Prozesstypen jeweils mit einem Beispiel aufgelistet. (Schwickert, Fischer, 1996)

PROZESSDIMENSION/-TYP		BEISPIEL
<b>Organisationseinheiten</b>	Unternehmensübergreifend	Einkaufsabwicklung
	Funktionsübergreifend	Produktentwicklung
	Stellenübergreifend	Personalentwicklung
<b>Gestaltungsobjekt des Prozesses</b>	materiell	Produktfertigung
	informationell	Strategieentwicklung
<b>Tätigkeiten</b>	Operativ	Auftragsbearbeitung
	Steuerungsprozeß	Budgetierung

Tabelle 2.1: (Geschäfts-)Prozessarten nach (Schwickert, Fischer, 1996)

## 2.2 Organisation und Aufgaben eines Energieversorgers

Das aktuelle Marktumfeld der Energieversorgungsunternehmen ist bestimmt von den Wirkungen der europäischen Liberalisierung. Ziel der politischen Bestrebungen ist dabei vor allem die Öffnung des Strom- und Gasmarktes. Die erste Stufe der Liberalisierung (Öffnung des Strommarktes) wurde mit dem am 29.4.1998 in Deutschland in Kraft getretenen Energiewirtschaftsgesetz (EnWG) umgesetzt. Dieses Gesetz wurde auf Basis

der Binnenmarktrichtlinie „Elektrizität“ der Europäischen Union von den Gesetzgebungsgremien der Bundesrepublik Deutschland verabschiedet. Das gesetzlich anerkannte Monopol wurde damit den Energieversorgungsunternehmen in ihren Versorgungsgebieten entzogen und ein allgemeiner Netzzugang zu den Stromnetzen geschaffen. Dieser Hintergrund hat innerhalb der Unternehmen zu einem Paradigmenwechsel geführt, der durch die Abbildung 2.2 verdeutlicht wird (Schrenner, 2005).

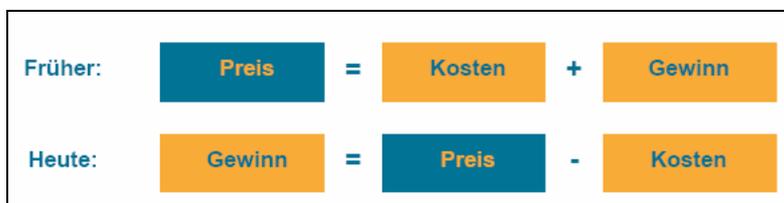


Abbildung 2.2: Paradigmenwechsel bei Energieversorgern (Schrenner, 2005)

Die zweite Stufe der Liberalisierung des Strom- und Gasmarktes ist durch das Unbundling geprägt. Durch Unbundling sollen Diskriminierungen, Quersubventionen und Wettbewerbsverzerrungen im liberalisierten Energiemarkt vermieden werden. Unbundling besteht aus der gesellschaftsrechtlichen und buchhalterischen Entflechtung der Funktionen „Erzeugung – Übertragung – Verteilung“ von Energie. Bei nahezu allen derzeit am Markt tätigen deutschen EVU sind sowohl Netz als auch Vertrieb in einem Unternehmen zusammengefasst. Dies ist nach dieser gesetzlichen Forderung nicht mehr zulässig und bringt wiederum weit reichende Veränderungen in Unternehmensstruktur und Organisation mit sich (Schrenner, 2005).

Das genannte Marktumfeld der Energieversorger ist in der beschriebenen Form zwar nur für den Strom- und Gasmarkt gültig, jedoch wirken sich die dadurch hervorgerufenen Veränderungen samt Denk- und Handlungsweisen auch auf andere Versorgungsmedien aus. Insbesondere ergeben sich diese Wirkungen zwangsläufig in den Querverbundunternehmen mit mehreren verschiedenen Versorgungssparten.

### 2.2.1 Strukturierung und Arten von Energieversorgungsunternehmen

Grundsätzlich lassen sich Energieversorgungsunternehmen durch drei Merkmale klassifizieren. Es erfolgt eine Unterscheidung durch Gesellschafterstruktur, betriebene Sparten und Ausdehnung bzw. Struktur des Versorgungsgebiets. Diese Kategorisierung wird nachfolgend näher beschrieben. Es sei angemerkt, dass Stadtwerke in Städten mit mehr als 500.000 Einwohnern als eine Mischform dieser Grundtypen bezeichnet werden können.

Die Unterscheidung nach der Gesellschafterstruktur erfolgt in zwei Typen. Stadtwerke, die zusätzlich zu Energieversorgungsdienstleistungen auch hoheitliche Aufgaben einer Kommune abdecken (z.B. Abwasserentsorgung) sowie Infrastrukturdienstleistungen, wie den öffentlichen Personennahverkehr (ÖPNV), anbieten, sind in der Regel vollständig kommunale Energieversorger. Teil- oder mehrheitlich privatisierte Unternehmen stehen im Gegensatz dazu. Hier sind in der Regel Großkonzerne der Energiebranche mehrheitlich beteiligt. Die Konzernstrategie leitet die Unternehmensentscheidungen.

Struktur und Ausdehnung des Versorgungsgebiets bietet gleichermaßen eine Zweiteilung an. Unternehmen, die in der Regel in kleineren Städten und Überlandgebieten tätig sind, werden als Flächenversorger bezeichnet und betreuen oft nur ein Medium mit jedoch einem hohen Spezialisierungsgrad. Die Kundenbetreuung erfolgt meist über lokale Kundencenter und Werkhöfe. Diese Dezentralisierung stellt an die IT-Strukturen von Informationsquellen der Unternehmen besondere Anforderungen. Stadtwerke, die in der Regel mehrere Versorgungsmedien in einem Stadtgebiet betreiben und verwalten, deren Versorgungsgebiet überschaubar ist, Betriebshof und Kundencenter zentral liegen, sind eine weitere Klasse von Energieversorgern. Die Mitarbeiter verfügen über eine ausgezeichnete Ortskenntnis und arbeiten im gesamten Versorgungsgebiet.

Das Portfolio an Dienstleistungen von Energieversorgungsunternehmen lässt sich in drei Hauptgruppen unterteilen. Ein definiertes Netz von Betriebsmitteln liegt allen zugrunde.

- Versorgung (Strom, Gas, Wasser, Fernwärme),
- Entsorgung (Abfall, Abwasser) und
- Infrastruktur (Telekommunikation, Kabel-TV, ÖPNV).

## 2.2.2 Geschäftsprozesse bei Energieversorgungsunternehmen

Das Energiewirtschaftsgesetz (EnWG) fordert von den Energieversorgungsunternehmen „eine möglichst sichere, preisgünstige und umweltverträgliche leitungsgebundene Versorgung mit Elektrizität und Gas im Interesse der Allgemeinheit“ (EnWG, 1998).

Das Gesetz bezieht sich zwar in erster Linie nur auf die Medien Strom und Gas, das daraus resultierende Handeln ist jedoch in der Regel für alle Sparten generalisiert. Dieses neue Handeln hat Auswirkungen im technischen Betrieb, angefangen von der Erzeugung bis hin zur Verteilung von Energie. Innerhalb eines Energieversorgungsunternehmens gibt es eine Vielzahl von Prozessen, auf die diese Gesetzgebung Einfluss hat (Schrenner, 2005). Man spricht hier von betriebsmittelbezogenen und kundenbezogenen Geschäftsprozessen, die im Folgenden weiter dargestellt werden.

In der Regel laufen alle Kundenkontakte über einen zentralen Ansprechpartner wie beispielsweise das Kundenbüro. Dieses nimmt hinsichtlich kundenbezogener Geschäftsprozesse eine zentrale Stellung ein. Diese Betreuung wird auch als „One Face to the Customer“-Prinzip bezeichnet.

Die Bereiche zeigen die logische Abfolge der Beziehung zwischen Kunde und EVU auf. Beginn des Kundenkontakts erfolgt im Marketing und Vertrieb. Danach folgt im günstigsten Fall die Realisierung eines oder mehrerer Hausanschlüsse. Die Abrechnung des Verbrauchs und weitere Informationen oder Mitteilungen des Kunden werden im Rahmen der Kundenbetreuung abgehandelt. Die nachfolgende Übersicht erläutert tabellarisch die wichtigsten Aufgabenfelder im Rahmen der Kundenbeziehung.

<b>Marketing und Vertrieb</b>	Ziel dieses Bereichs ist es, mit minimalem Einsatz möglichst viele Kunden sowohl für Verlegung eines Hausanschlusses als auch für Energiedienstleistungen zu gewinnen.
<b>Hausanschlusswesen</b>	Die zentrale Aufgabe innerhalb eines EVU ist das Hausanschlusswesen. Vom ersten Kundenkontakt über die Erstellung eines Kostenvoranschlags bis zur Realisierung und Inbetriebnahme wird alles angeboten bzw. abgedeckt.
<b>Kundenbetreuung</b>	Kundenbetreuung ist ein allgemeiner Begriff, der sich in mehreren spezialisierten Vorgängen wieder findet, beispielsweise die Abwicklung im Kontakt mit dem Kunden (z.B. Ummeldung, Kontenänderung, Störmeldeannahme, Mahnwesen, etc.).

Tabelle 2.2: Kundenbezogene Prozesse nach (Schrenner, 2005)

Die aufgezeigten Aufgaben eines Energieversorgungsunternehmens stellen den Kreislauf von betriebsmittelbezogenen Geschäftsprozessen dar und bilden damit den Lebenszyklus eines technischen Objekts ab. Es werden die Phasen Planung, Bauausführung, Inbetriebnahme, Wartung und Ausbau/Stilllegung durchlaufen. Als Besonderheit wird die Beauskunftung über die Leitungssituation an Dritte hervorgehoben, da dieser Geschäftsprozess eine Mischung aus kunden- und betriebsmittelbezogenen Geschäftsprozessen darstellt und während des gesamten Zyklus geschieht. Die nachfolgende tabellarische Übersicht zeigt die wichtigsten Phasen hinsichtlich der Betriebsmittel:

<b>Planung</b>	Aufgrund eines ermittelten Bedarfs (z.B. Erneuerung, Neubau) erfolgt die Planung einer Anlage.
<b>Bauausführung</b>	Nach der Planung beginnt die Bauausführung. Diese kann von unternehmenseigenen Personal oder Fremdfirmen realisiert werden.
<b>Inbetriebnahme</b>	Nach Abschluss der Bauphase wird die Anlage in Betrieb genommen.
<b>Wartung</b>	Wird eine Anlage betrieben, so ist diese während des Betriebs nach entsprechenden Vorschriften zu warten und instand zu halten.
<b>Ausbau / Stilllegung</b>	Wird eine Anlage nicht mehr benötigt, kann sie ausgebaut oder stillgelegt werden.
<b>Beauskunftung</b>	Während des gesamten Zyklus einer Anlage ist diese zu dokumentieren. Jeder, der Schachtarbeiten im Einzugsbereich des EVU durchführen möchte, bekommt Informationen über den vorhandenen Leitungsbestand.

Tabelle 2.3: Betriebsmittelbezogene Prozesse (Schrenner, 2005)

### 2.2.3 Raumbezogene Geschäftsprozesse

Da etwa 80% aller Informationen einen Raumbezug besitzen (IMAGI, 2006), fließen diese räumlichen oder Geo-Informationen auch in die Geschäftsprozesse mit ein. In erster Linie handelt es sich um technische bzw. betriebsmittelbezogene Geschäftsprozesse, in der diese Informationen einen Mehrwert erzeugen. Das verdeutlicht das nachfolgende Beispiel:

Die grafische Lokalisierung eines Kunden, der eine Störung meldet oder einen Hausanschluss beantragen möchte, liefert im Gegensatz zu einem Kunden, der seine Rechnung zum ersten Mal nicht bezahlt hat, einen wesentlichen Mehrwert. Bei letzterem werden lediglich Standortinformationen übertragen, während im ersten Beispiel weitere Informationen aus der Darstellung gewonnen werden können. So kann im Störfall die Umgebung samt den Netzen angezeigt und bei Antragstellung gleich die Beurteilung über eine Versorgungsmöglichkeit des neuen Kunden erfasst werden.

Des Weiteren wird unterschieden, ob die räumlichen Informationen nur abgerufen, im Rahmen des Prozesses verändert bzw. neu erzeugt werden oder ob sie genutzt werden, um weitere Informationen zu generieren.

Zunächst steht die generelle Verfügbarkeit der Geoinformation zur Diskussion. Die Möglichkeit, ein Objekt (Störung, Trafostation, Kunde,...) im Kontext auf einer Karte darzustellen, erleichtert die Abarbeitung technischer Geschäftsprozesse. Nach (Hauffe, 2004) ist die Fehler- und Kundenlokalisierung ein entscheidender Vorteil für die Abarbeitung von entsprechenden Geschäftsprozessen.

Ein weiterer Grad der Nutzung von Geodaten ist ihre Veränderung oder Erstellung während eines Arbeitsablaufs. Hierunter fällt die Korrektur der Daten nach der Ersterfassung im Rahmen eines Geschäftsprozesses. So kann bei einer Instandhaltungstätigkeit festgestellt werden, dass Attribute oder sogar die Lage eines Objekts nicht korrekt im Datenbestand aufgenommen worden ist. Die Folge ist eine Änderung der Attribute bzw. der geografischen Lage des GIS-Objektes. Eine Erstellung von Objekten kann beispielsweise im Rahmen des Störfallmanagements durch die Erfassung der genauen Störungslokalisierung erfolgen. Sie kann zur Priorisierung von weiteren Störungen dienen und nach der Erfassung auf einer Übersichtskarte, die alle aktuellen Störungen im Versorgungsgebiet anzeigt, visualisiert werden.

Als dritte Möglichkeit zur Nutzung der GIS-Daten in einem Geschäftsprozess steht die Informationserzeugung durch den Raumbezug. Wird im Rahmen einer Instandhaltungsmaßnahme eine Trafostation kurzzeitig vom Netz genommen, können im GIS die folgenden Informationen generiert werden: Gibt es eine Alternativversorgung für die Kunden? Falls nein, welche Kunden müssen über den kurzzeitigen Stromausfall benachrichtigt werden?

In den realen Prozessen der Energieversorgungsunternehmen wird sich die Unterscheidung in der Nutzung der Geoinformationen in Arbeitsprozessen nicht eindeutig herausstellen lassen. So werden die räumlichen Informationen in nahezu jedem Prozess in unterschiedlicher Art und Weise benötigt. Es entstehen Mischformen. Als Beispiel sei das Störfallmanagement genannt. Zu Beginn des Prozesses wird eine Lokalisierung der Schadenstelle, sofern bekannt, bzw. des Meldenden benötigt. Im weiteren Verlauf wird beispielsweise ein GIS-Objekt „Störung“ erzeugt und auf einem Leitungsabschnitt als Punktobjekt platziert. Zugehörige Attributdaten werden ebenfalls angelegt. Je nach Reparaturanforderung muss ein Leitungsabschnitt gesperrt werden. Im GIS erfolgt die Sperrstreckenermittlung und daraus resultierend eine Liste der unversorgten Kunden. Die Geoinformationen fließen demnach an verschiedenen Stellen mit in den Arbeitsprozess ein.

## 2.3 Unternehmens-IT

### 2.3.1 Allgemeine Softwarelandschaft

Faktisch sämtliche Verwaltungs- und Geschäftsprozesse von Energieversorgern werden durch EDV-Anwendungen unterstützt oder sogar komplett abgewickelt. Das Portfolio der verwendeten Anwendungssoftware ist dabei vielfältig. Es lassen sich nach (Schrenner, 2005) folgende Hauptgruppen unterscheiden. Office-Anwendungen, Kommunikationssoftware, Dokumentenmanagementsoftware (DMS), Enterprise Resource Planning (ERP), Netzdokumentationssoftware (GIS/CAD), Netzberechnungsprogramme und Netzleittechnik (SCADA-Systeme). Die weitere Beschreibung der Gruppen folgt in der Tabelle. (Schrenner, 2005):

<b>Office-Anwendungen</b>	Standardanwendungen zur Erstellung von Schriftstücken und Kalkulationen, z.B. MS Word, MS Excel.
<b>Kommunikationssoftware</b>	Software zur Abwicklung der Kommunikation zwischen Mitarbeitern (Fax- und E-Mailverkehr) sowie der Koordination von Mitarbeiterressourcen und deren Aufgaben, z.B. MS Outlook, Groupwise.
<b>Dokumentenmanagementsoftware (DMS)</b>	Programm zur Verwaltung von Dateien und Dokumenten innerhalb des Unternehmens. Hauptanwendungsgebiet ist meist die hausinterne Post. Die verwendeten Systeme sind unterschiedlich, so dass in vielen Unternehmen auch ERP- oder GIS-Systeme zur Dokumentenverwaltung genutzt werden.
<b>Enterprise Resource Planning (ERP-Systeme)</b>	Softwaresysteme wie z.B. SAP, Schleupen CS, Navision, die den gesamten betriebswirtschaftlichen Ablauf des Unternehmens unterstützen, steuern und auswerten.
<b>Netzdokumentationssoftware (GIS / CAD)</b>	Software zur grafischen und alphanumerischen Abbildung der Netze aus technischer Sicht. Die Systeme werden sowohl zur Planung als auch zur Auskunft genutzt.
<b>Netzberechnungsprogramme</b>	Programme zur Berechnung und Bewertung von Netzen. Ein Informationsaustausch zur Netzdokumentation ist meist vorhanden.

### Netzleittechnik (SCADA-Systeme)

"Supervisory Control And Data Acquisition", kurz SCADA genannt, ein Netzleitsystem, welches in Schaltwarten für Versorgungsnetze zum Einsatz kommt. Die Systeme werden meist völlig autark vom restlichen Netz betrieben, da für diese Systeme eine sehr hohe Verfügbarkeit gewährleistet sein muss.

Tabelle 2.4: Anwendungssoftware bei EVU (Schrenner, 2005)

Grundsätzlich gelten für alle Systeme, dass sie effektiv und wirtschaftlich eingesetzt und betrieben werden müssen. Die Ausrichtung des Softwareeinsatzes fußt deshalb zunehmend auf Standardisierung.

Im Folgenden werden die Kategorien der ERP- und GIS-Systeme näher betrachtet.

### 2.3.2 ERP-Systeme

ERP steht für Enterprise Resource Planning. Diese Bezeichnung verwendet man für die unternehmerische Aufgabe, alle dort befindlichen Ressourcen, worunter wiederum alle Betriebsmittel wie Personal, Kapital, Produktion verstanden werden, so effizient und effektiv wie möglich einzuplanen, um den eigentlichen Geschäftsprozess möglichst optimal zu gestalten und auszunutzen. Typische Funktionsbereiche einer ERP-Software sind laut (Wikipedia, 2006a):

- Materialwirtschaft (Beschaffung, Lagerhaltung, Disposition, Bewertung),
- Produktion,
- Finanz- und Rechnungswesen,
- Controlling,
- Personalwirtschaft,
- Forschung und Entwicklung,
- Verkauf und Marketing,
- Stammdatenverwaltung.

(Hansen, Neumann, 2005) definiert ein ERP-System wie folgt:

*„... versteht man ein aus mehreren Komponenten bestehendes integriertes Anwendungspaket, das die operativen Prozesse in allen wesentlichen betrieblichen Funktionsbereichen unterstützt (Finanz- und Rechnungswesen, Personalwirtschaft, Materialwirtschaft, Produktion, Vertrieb). Die Integration wird dabei von einer zentralen Datenbank getragen, wodurch Datenredundanzen vermieden und integrierte Geschäftsprozesse ermöglicht werden.“*

Laut (Wikipedia, 2006a) wird ein großer Teil des weltweiten ERP-Marktes zwischen sechs Anbietern aufgeteilt. Es handelt sich um die Firmen SAP (R/3, mySAP), Oracle (E-Business Suite), Sage (in Deutschland Office Line und Classic Line), und Microsoft (Microsoft Dynamics AX (ehemals Axapta) und Dynamics NAV (ehemals Navision)). Die restlichen ca. 50 % des Marktes teilen sich verschiedene kleinere ERP/PPS-Anbieter. Alle großen Anbieter bieten zu den bereits aufgeführten Funktionsbereichen (z. B. Finanzwesen und Materialwirtschaft) Module an, die von den Kunden auf ihre individuellen Bedürfnisse durch so genanntes „Customizing“ angepasst

werden können (Wikipedia, 2006a). Selbstverständlich wird in diesem Marktbereich auch OpenSource-Software angeboten.

ERP-Systeme decken die betriebswirtschaftlichen Anwendungen eines Unternehmens sowohl mengen- als auch wertmäßig in ihrer Gesamtheit ab und erlauben eine integrierte Abwicklung unternehmensweiter Geschäftsprozesse. Hierin inbegriffen sind laut (Scheer, Werth, 2005) die Funktionen des logistischen Teilbereichs (Einkauf, Verkauf, Produktionsplanung, Materialwirtschaft), des Rechnungswesens sowie des Personalmanagements. Dabei erfolgt im geeigneten Kontext eine weit reichende Vernetzung der individuellen Funktionen. So bewirkt beispielsweise ein Warenausgang eine Korrektur der Dispositionsplanung und eine entsprechende Wertberichtigung der Lagerbewertung. Diese Referenzprozesse erlauben jedoch nur eine sehr enge, vom Hersteller vorgegebene Anpassung, die in der Regel bei der Einführung des Systems einmalig vorgenommen wird und dann unverändert bleibt. Dies erweist sich als eines der größten Hemmnisse einer Flexibilisierungsstrategie in einem Unternehmen.

Eben diese starren Strukturen sowie die monolithische Architektur der ERP-Systeme lässt laut (Scheer, Werth, 2005) eine weitergehende Agilität nicht zu. Unternehmen erkennen zunehmend, dass sich ERP-Systeme für die Geschäftsanforderungen als zu unflexibel erweisen und ferner die bereitgestellte Funktionalität nicht ausreicht, um laut (Theling, Loos, Sommerrock, 2005) die notwendigen oder gewünschten Geschäftsprozesse umzusetzen. Basisanforderung bleibt die flexible Unterstützung komplexer Aktivitäten durch Unternehmenssoftware. Dabei geht es nicht ausschließlich um die effiziente organisatorisch-technische Reorganisation existierender Prozesse, sondern auch um die Möglichkeit zur schnellen Neukonstruktion von Prozessen.

Die zunehmende Bedeutung zwischenbetrieblicher Kooperationen haben die bisher beschriebenen ERP-Systeme vor neue Herausforderungen gestellt. Die unternehmensweiten Standardsoftwarelösungen waren den anstehenden Geschäftsanforderungen nicht mehr in vollem Umfang gewachsen. Durch den Einsatz von Internettechnologien ist die Einbindung von Lieferanten und Kunden in das unternehmenseigene Informationssystem stark begünstigt worden. Die Erweiterung von Standard-ERP-Systemen um neue überbetriebliche Funktionalitäten wird nach (Theling, Loos, Sommerrock, 2005) in der Literatur als ERP II- bzw. Extended ERP-System bezeichnet. Analog zum ERP-Begriff ist der Terminus ERP II gebräuchlich, es hat jedoch bis dato keine Vereinheitlichung zu ERP II stattgefunden, wie (Theling, Loos, Sommerrock, 2005) belegt.

ERP II gibt der Integration eine neue Qualität. Diese umfasst den sprach- und ortsunabhängigen Austausch von Echtzeitinformationen zur unternehmensübergreifenden Optimierung der Planung und Steuerung von Aktivitäten. Die Virtualität der Geschäftsbeziehungen nimmt zu und wird auch als Collaborative Commerce (C-Commerce) bezeichnet. Es hat sich in der internationalen Literatur die Meinung durchgesetzt, dass ERP II in jedem Fall Komponenten des Supply Chain Management und Customer Relationship Management enthalten sollte. Im gleichen Zusammenhang werden auch Data Warehouse-Anwendungen und Verfahren der Business Intelligence angeführt. Des Weiteren können ERP II-Systeme branchenspezifische Funktionen und Voreinstellungen enthalten. Es werden sonstige E-Business-Applikationen grundsätzlich als ERP II-Bausteine angesehen, wie Abbildung 2.3 verdeutlicht.

Die Implementierung von ERP II-Modulen erfordert erfahrungsgemäß die Umgestaltung bestehender Geschäftsprozesse (Business Process Reengineering). Die folgende Abbildung illustriert ein herkömmliches ERP-System erweitert mit ERP II-Komponenten, das auch als hybrid bezeichnet wird.

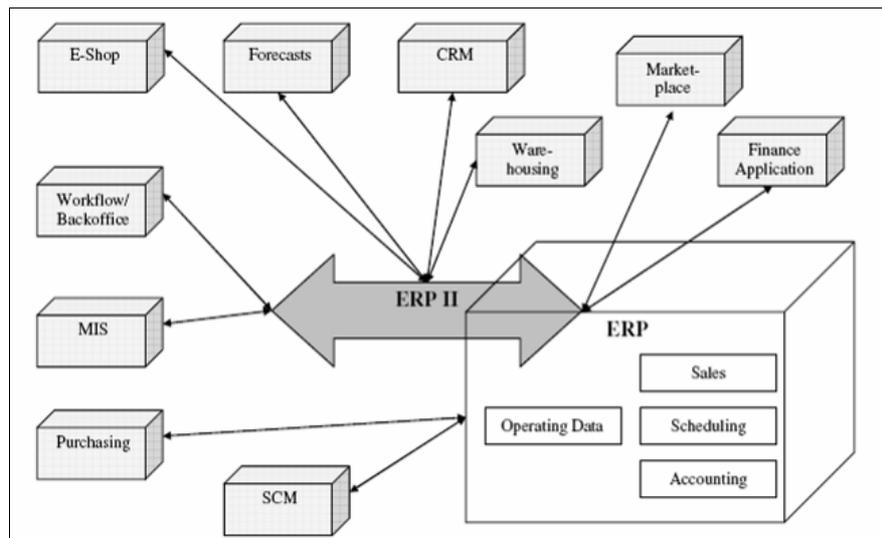


Abbildung 2.3: ERP- und ERP II-Komponenten (Theling, Loos, Sommerrock, 2005)

Erweiterte Applikationen können laut (Theling, Loos, Sommerrock, 2005) entweder von den ERP-Herstellern selbst entwickelt oder durch Zukauf von spezialisierten Teilsystemen anderer Anbieter in bestehende ERP-Systeme eingefügt werden. Das Unternehmen SAP hat beispielsweise den Weg der Entwicklung beschritten, wohingegen Oracle, J.D. Edwards, Peoplesoft und Baan sich für die Verschmelzung von externen Softwarelösungen entschieden haben. Die Realisierung einer ERP II-Lösung aus Anwendungen verschiedener Hersteller kann nur über eine herstellerunabhängige Kommunikation zwischen den Applikationen realisiert werden. Dies ist die Basis für alle computergestützten E-Business-Aktivitäten. Stammen die erweiterten Anwendungen von den führenden Herstellern aus den einzelnen Gebieten, spricht man von einer Best-of-Breed-Lösung.

Die Gartner Group identifiziert laut (Theling, Loos, Sommerrock, 2005) sechs charakteristische Unterschiede zwischen ERP und ERP II, welche in Abbildung 2.4 in kompakter Form dargestellt werden. ERP-Systeme zielen auf die Optimierung der innerbetrieblichen Aktivitäten ab und können daher als "Backbone" des Unternehmens bezeichnet werden. Bei ERP II-Systemen dienen diese Funktionen eher als Plattform für den zwischenbetrieblichen Informationsaustausch in der Wertschöpfungskette. Dementsprechend verschiebt sich der Fokus von den internen auf die unternehmensübergreifenden Prozesse. Waren ERP-Systeme noch monolithisch und der Umwelt gegenüber verschlossen, erfordert die Interoperabilität der ERP II-Systeme eine Web- und Komponentenbasierte Lösung. Im Vergleich zu ERP-Systemen lässt diese Technologie eine schnellere, unkomplizierte und kostengünstige Implementierung von einzelnen Modulen zu. Im Gegensatz zu ERP-Systemen, die ursprünglich bei Fertigungs- und Vertriebsunternehmen eingesetzt wurden, berücksichtigen ERP II-Systeme auch die Besonderheiten unterschiedlicher Wirtschaftszweige mittels branchenspezifischer Lösungen. Diese "präkonfigurierte" Standardsoftware, die auch als Branchensoftware bezeichnet wird, soll den Aufwand des Customizing verringern.

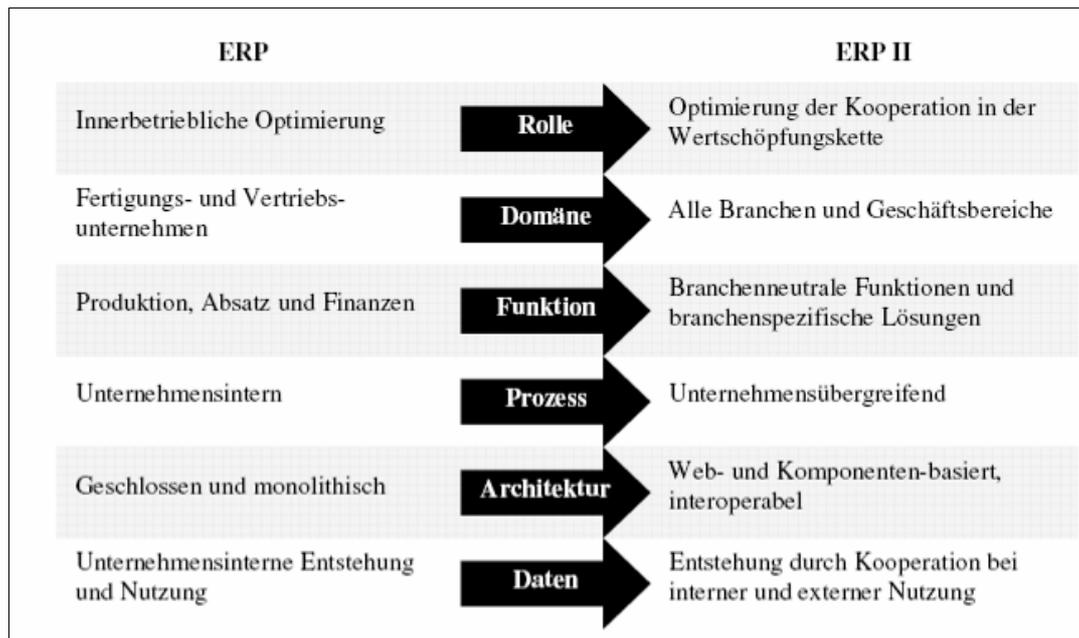


Abbildung 2.4: Unterschiede ERP und ERP II (Theling, Loos, Sommerrock, 2005)

### 2.3.3 Geoinformationssysteme

Eine wichtige Voraussetzung für einen effizienten Netzbetrieb in einem Energieversorgungsunternehmen sind die Informationen und daraus resultierende Kenntnisse über das eigene Netz. Zu diesem Zweck werden mit zum Teil erheblichem Aufwand die Netze dokumentiert. Da der größte Teil des Anlagevermögens im Erdreich verborgen und eine optische Wahrnehmung der Anlagen nicht möglich ist, bildet die Netzdokumentation de facto die genaueste verfügbare Abbildung des Anlagevermögens. Diese Netzdokumentation wird inzwischen bei den meisten deutschen Energieversorgern nicht mehr analog (Pläne) sondern in digitaler Form durchgeführt. In der Regel bildet ein GIS die Basis für die digitale Netzdokumentation. Die Dokumentation der Netze wird hier als eine grundsätzliche Aufgabe von GIS-Systemen bei einem Energieversorgungsunternehmen identifiziert. Dabei werden zusätzlich zur geographischen Lage auch technische Eigenschaften der Anlagen erfasst (Schrenner, 2005). Es ist jedoch zu beachten, dass die Begriffsbedeutung „digitale Dokumentation“ nur ein funktionaler Aspekt eines Geographischen Informationssystems darstellt. Ein geographisches Informationssystem (auch: Geoinformationssystem, Kurzform GIS) wird nach (Bill, Zehner, 2001) wie folgt definiert: Ein GIS ist ein *„rechnergestütztes Informationssystem, das aus Hardware, Software, Daten und den Anwendungen besteht. Mit ihm können raumbezogene Daten digital erfasst und redigiert, gespeichert und reorganisiert, modelliert und analysiert sowie alphanumerisch und grafisch präsentiert werden.“*

Nach dieser doch recht abstrakten Definition sollen die Eigenschaften von einem geographischen Informationssystem näher erläutert werden. Der Begriff GIS wird in weiterem Zusammenhang sowohl für GIS-Projekte, als auch für GIS-Software verwendet. Unter einem GIS-Projekt versteht man alles, was für digitales raumbezogenes Arbeiten nötig ist (Hardware, Software, Daten und Organisationsformen). Die GIS-Software ist dabei als Werkzeug anzusehen, mit dem die Daten im Sinne der oben angesprochenen Definition erfasst, verwaltet, analysiert, fortgeführt und präsentiert bzw. ausgegeben werden. Durch ein GIS werden Objekte aus der realen Welt als Geodaten in einem IT-System abgebildet. Diese Abbildung besteht aus Geometrie (z.B. Koordinaten) und zugehörige, beschreibende Sachdaten (z.B. Name, Adresse, Ort). Die Ablage der Daten erfolgt in Dateiform oder in einem Datenbanksystem. Die Abbildung 2.5 zeigt die GIS-Datenformen und ihre Beziehung zueinander.

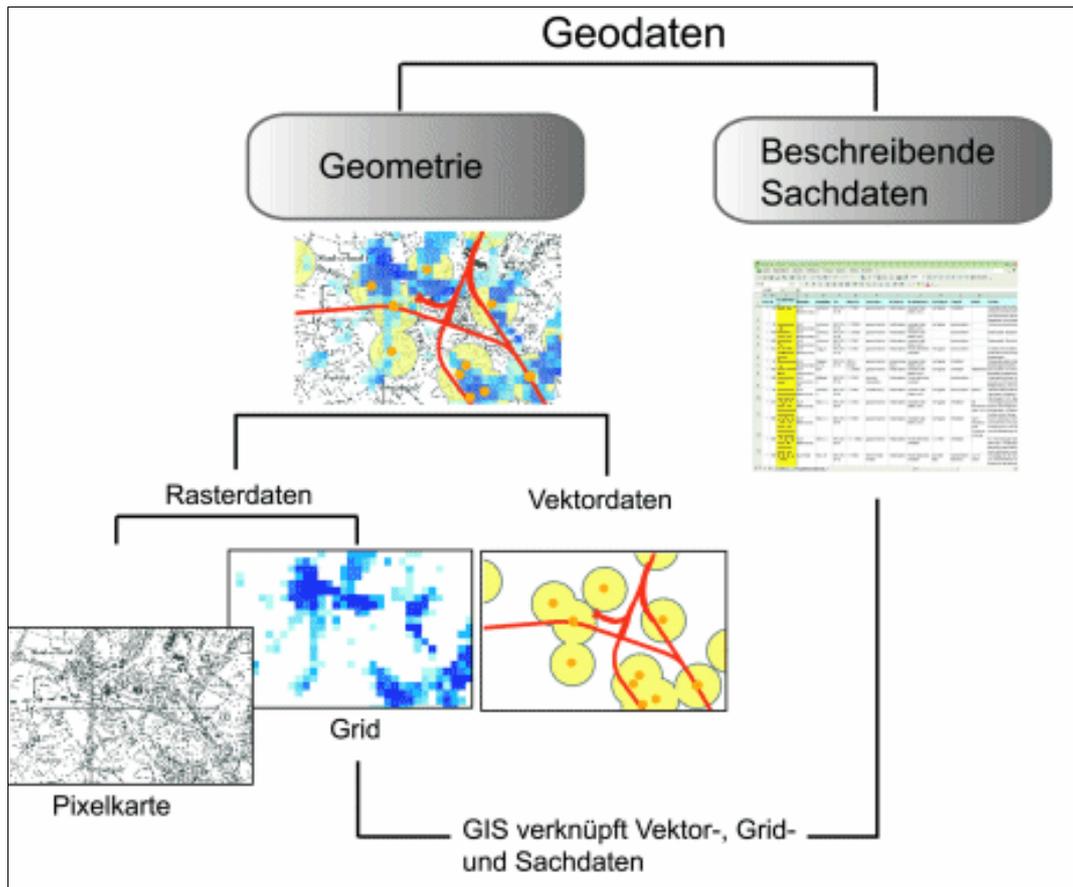


Abbildung 2.5: GIS-Datenformen (GISZH, 2007)

Die Darstellung in Kartenform kann durch Legenden als so genannte Darstellungsfiler variiert werden. Gleiche Objekte können dadurch unterschiedlich visualisiert werden und man erhält thematische Karten. Des Weiteren konzentriert sich ein GIS auf sämtliche Informationen, die zu einem Objekt gespeichert worden sind. In einigen Systemen lassen sich sogar mehrere Geometrien für ein Objekt anlegen, um eine andere Darstellung des Objektes in Karten mit unterschiedlichen Maßstäben zu erreichen. Beispielsweise hat ein Wasserschieber in einem Detailplan mit Maßstab 1:500 ein anderes Symbol als in einem Übersichtplan mit dem Maßstab 1:2000. Diese Unterschiede verdeutlicht Abbildung 2.6.

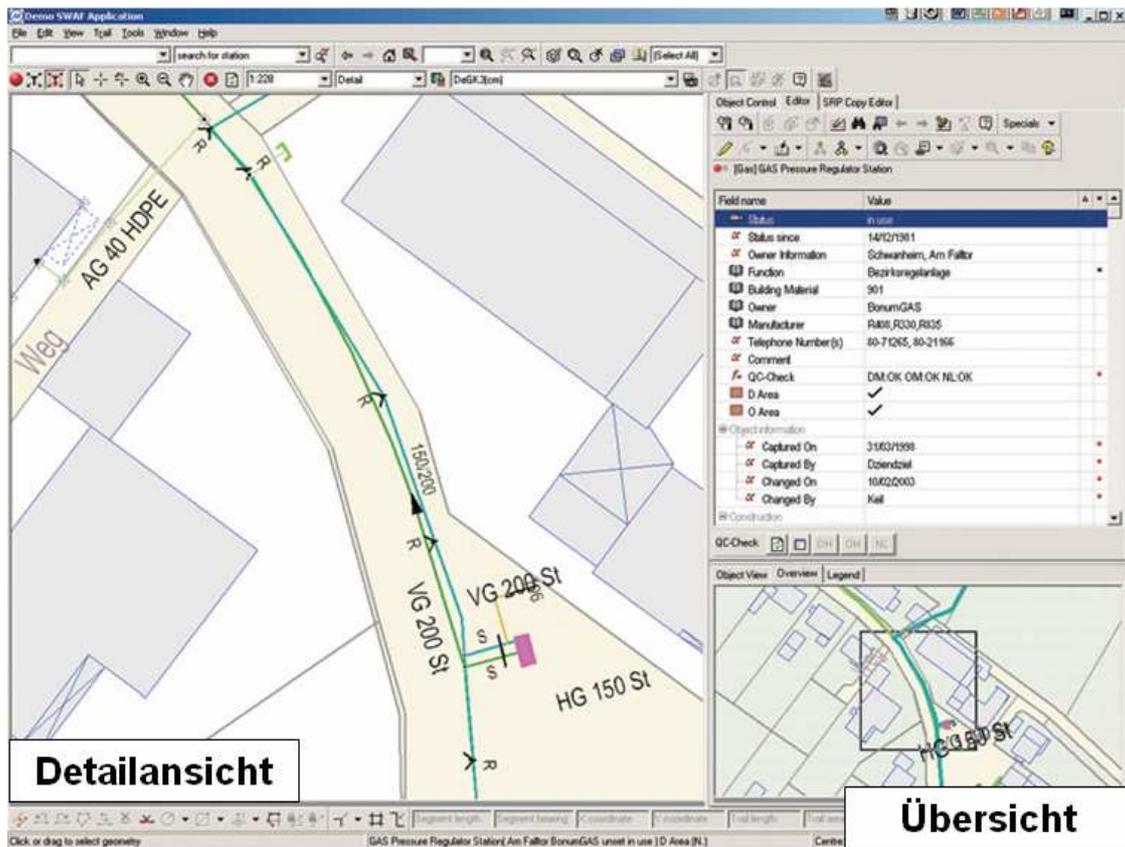


Abbildung 2.6: Übersicht und Detailansicht im GIS (Mettenmeier, 2007)

Im Gegensatz zu verwandten Produkten, wie dem CAD-System (Computer Aided Design) verfügt das GIS über die Möglichkeit, Geometrie- und Sachdaten sämtlicher Zusammenhänge (komplex, logisch-inhaltlich, räumlich) zu erfassen und zu verwalten. Laut (Wikipedia, 2006b) erweitert ein GIS die Nutzungsmöglichkeiten der klassischen Landkarte. Neben der Darstellung spielen Geoperatoren zur Analyse der Geodaten eine wichtige Rolle. Auf Basis eines guten und vollständigen geometrischen und attributiven Datenbestandes ermöglicht ein GIS die Anwendung von zahlreichen Methoden, durch die neue Informationen generiert werden können. Dies wird in der Abbildung 2.7 der GIS-CAD-Modellvorstellung verdeutlicht. Exemplarisch ist hier eine Anzahl von Methoden aufgelistet. Für eine vollständige Auflistung siehe (Wikipedia, 2006b).

- Abfragen und Selektionen,
- Abstandszonen,
- Verschneidung,
- Topologie- bzw. Netzwerkanalysen,
- Georeferenzierung,
- Generalisierung und
- Visualisierung und Präsentation.

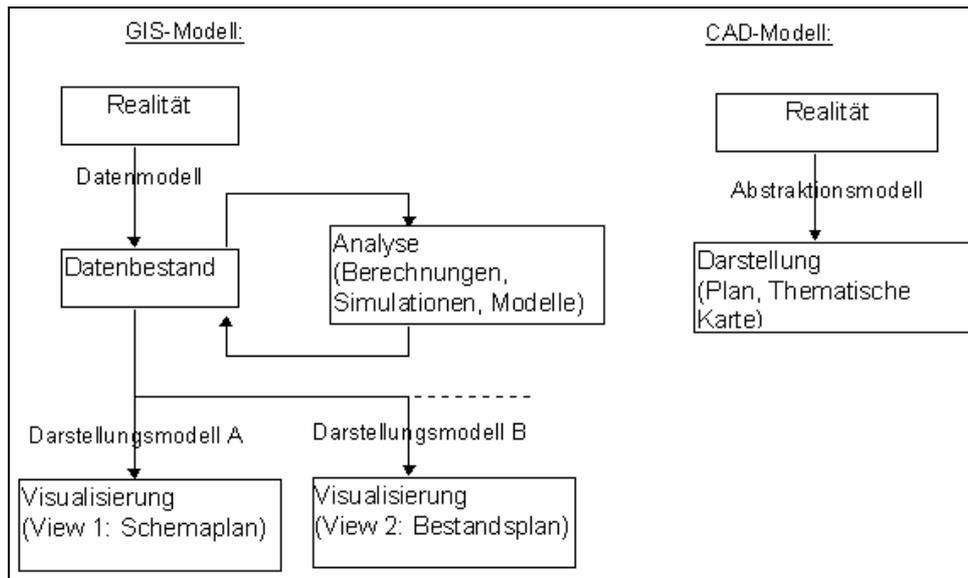


Abbildung 2.7: GIS und CAD Modellvorstellung (GIUP, 2007)

Wie bei den ERP-Systemen gibt es auch hier die Wahl zwischen proprietärer Software (als Beispiele können Smallworld GIS, ArcGIS, MapInfo, GeoMedia genannt werden) und solcher, die als OpenSource Software entwickelt und lizenziert wird. Die Software GRASS GIS ist in diesem Bereich die bekannteste. Die Einsatzgebiete von GIS werden durch die wachsende Verbreitung von verteilten, dienstbasierten Architekturen erheblich erweitert. Die Informationen und Funktionalitäten traditioneller "Desktop" GIS werden durch die Eingliederung in Geodateninfrastrukturen (GDI) mit browserbasierten Arbeitsplätzen einer erheblich breiteren Anwendergemeinschaft zur Verfügung gestellt.

Schon Ende der 50er Jahre gab es Entwicklungen, aus denen später das GIS entstanden ist. Die Ideen für rechnergestützte räumliche Präsentationen, Überlagerungen und erste Anwendungen von Vektorgraphiken und Drahtmodellen waren die Basis. Das erste Geländemodell wurde am Massachusetts Institute of Technology (MIT) entwickelt. Anwendungen der Rastertechnik als digitale Bildverarbeitung wurde in den 60er Jahren eingesetzt und es entstand nahezu parallel in dem Harvard Laboratory for Computer Graphics and Spatial Analysis und der ETH Zürich die Idee der unabhängigen Datenebenen. Das war die Geburtsstunde des GIS. Das erste GIS-Projekt gab es in Kanada (CGIS). Die Bezeichnung GIS entstand jedoch erst auf dem ersten Symposium, das Roger Tomlinson 1970 organisierte. Die Entwicklungen samt Forschern aus Harvard gingen unter anderem in der Firma ESRI und der GIS-Lösung Arc/Info auf, während in der Schweiz die Idee für das GIS in der Firma Digital und später Adasys überging. Für den klein- und mittelmaßstäbigen Bereich kann ESRI (Environmental Systems Research Institute) als GIS-Marktführer weltweit angesehen werden. Adasys hingegen hat bisher nur in der Schweiz einen Namen erlangt (GIUP, 2007).

Nach (Bartelme, 2005) kann die Historie von GIS in fünf teilweise überlappende Phasen untergliedert werden:

### 1. 1955 - 1975: Zeit der Pioniere;

individuelle, isolierte Lösungswege der Entwickler.

### 2. 1970 - 1985: Zeit der Behörden

Entwicklung von Konzepten (z.B. ALK - Automatisierte Liegenschaftskarte) und beginnende Umstellung von Basisdaten in die digitale Form, GIS wird als Erfassungswerkzeug genutzt.

### 3. 1979 - 1990: Zeit der Firmen

Es entsteht ein GIS-Markt, der die Hardware leistungsfähiger werden lässt und eine Umstellung von Großrechnern auf Workstation ermöglicht. Es entstehen Unternehmen, die heute einen Großteil des

Marktes bedienen: ESRI mit dem Produkt Arc/Info, SIEMENS mit SICAD, oder INTERGRAPH, um nur einige wenige zu nennen.

#### 4. 1988 - 1998: Zeit der Nutzer

GIS entwickelten sich hin zu Systemen, die - modular aufgebaut - einen Werkzeugkasten darstellen, der an Benutzerwünsche angepasst, zu so genannten Fachschalen zusammengestellt werden kann.

#### 5. Ab ca. 1995: Zeit des offenen Marktes

Angebot und Nachfrage bestimmen den Markt sowohl für GIS-Software als auch für Geodaten.

Ab 2005 wurde die Thematik der Geoinformation durch Programme wie Google Earth, NASA Worldwind oder Microsoft Virtual Earth allen Bürgern zugänglich gemacht. Mit einem Earth Viewer ist nun jeder mit PC und Internetanbindung in der Lage, Luftbilder und geographische Informationen der ganzen Welt auf seinem PC zu visualisieren. Die große Panik, dass diese neue Technik, die bestehenden GIS-Softwarelösungen vertreiben würde, hat sich nicht bestätigt. Für die Erfassung und Analyse von Geodaten ist nach wie vor ein Vollarbeitsplatz notwendig. Jedoch hat diese Art der Präsentation von Geodaten die Welt für die Nutzung und sogar für die Notwendigkeit von Geodaten sensibilisiert. So zeigt der nachfolgende Screenshot New Orleans nach Hurrikan Katrina mit Notunterkünften des Roten Kreuzes.

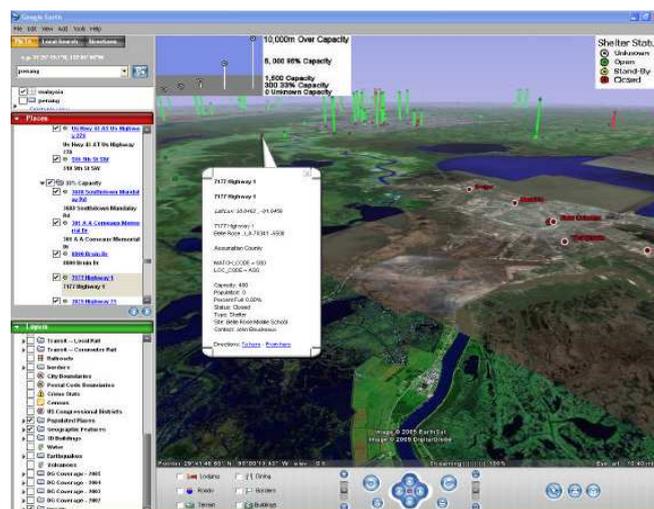


Abbildung 2.8: Darstellung von New Orleans in Google Earth (Soutschek, 2007)

## 3 Grundlagen (Geo)Informatik

### 3.1 Standards

Standards und Normen sind Begleiter des täglichen Lebens geworden. Wenngleich diese Begleiter nicht immer wahrgenommen werden. Beispielsweise passt ein elektrischer Haartrockner durch den genormten Netzstecker in die Steckdose, in der die benötigte Netzspannung von 220 Volt anliegt. Auch können wir relativ sicher sein, dass uns durch genormte Sicherheitsstandards beim Einstecken des Steckers in die Steckdose nichts passiert. Doch muss nicht der in einem Land A erworbene Haartrockner auch zwangsweise in anderen Ländern in die Steckdose passen. Daher gibt es Adapter, also eine Art von Schnittstellen, über die auf die Ressourcen fremdländischer Stromnetze zugegriffen werden kann. Hinzu kommt allerdings, dass nicht unbedingt die benötigte Spannung von 220 Volt anliegt, sondern vielleicht nur 110 Volt. Es wird deutlich, dass Standards und Normen auf nationale Bereiche eingeschränkt sein können.

Dieses Beispiel zeigt, dass alle Komponenten (Netzstecker, Steckdose, Spannung, etc.) eines technischen Systems aufeinander abgestimmt sein müssen. Daher ist es wichtig, dass Standards und Normen nicht nur in Form von Dokumenten beschrieben sind, sondern auch dass sie technisch umgesetzt werden. Zur Sicherung von praxisnahen Entwicklungen ist daher die Industrie und Anwender sowie die Wissenschaft in Standardisierungsprozesse einzubeziehen.

Die global ausgerichtete Informationstechnologie macht es erforderlich, dass national ausgerichtete Standardisierungen in den Hintergrund rücken. In diesem Bereich ist es wichtig, internationale Basisnormen zu definieren, um eine globale Kommunikation sicher zu stellen. Dennoch können internationale Standards die Grundlage für nationale technische Spezifikationen bilden, durch die nationale Besonderheiten berücksichtigt werden. Ein konkretes Beispiel dazu stellt die zukünftige Normbasierte Austauschschnittstelle des Liegenschaftskatasters dar, bei der auf der Grundlage von internationalen Standards und Normen ein nationaler Austauschmechanismus definiert wurde.

Die rasante Entwicklung des Internets in den 90er Jahren verlieh dem Geoinformationswesen einen enormen Innovationsschub. War es früher üblich, durch die getrennte Verwaltung von Geodaten diese bilateral auszutauschen und mehrfach abzulegen, ist es durch die Internettechnologie möglich, Geodaten unterschiedlicher Herkunft den Anwendern zur Verfügung zu stellen. Dazu werden komplexe Infrastrukturen für Geodaten auf regionaler, nationaler und internationaler Ebene diskutiert und aufgebaut. Technisch gesehen spielt es dabei kaum eine Rolle, ob ein Datenaustausch nur innerhalb einer einzelnen Institution, also in einem Intranet oder innerhalb des gesamten Internets stattfinden soll. Grundlage sind so genannte Dienste (Services), die Informationen zwischen Clients und Servern übertragen. Damit der Informationsaustausch in einem globalen Netzwerk sichergestellt werden kann, bedarf es an standardisierten Festlegungen, wie Daten zur Verfügung gestellt und übertragen werden. Dazu sind Standards und Normen von entscheidender Bedeutung.

#### 3.1.1 Standardisierungsgremien

Die wichtigsten Ziele bei der Standardisierung und Normierung sind den Datenaustausch zu vereinheitlichen und die Erhaltung des Informationsgehaltes beim Datentransfer zu gewährleisten. Darüber hinaus soll die Integration von Geodaten in fremde Anwendungen, die Interoperabilität, sichergestellt werden. Die Begriffe Standard und Norm werden im angelsächsischen Sprachraum nicht unterschieden, wodurch es häufig bei Übersetzungen zu Verwechslungen kommt. Die deutsche Sprache hingegen kennt zwischen diesen Bezeichnungen einen deutlichen Unterschied. Eine Norm ist ein durch ein autorisiertes Gremium erstelltes Dokument, welches die Vereinheitlichung von materiellen und immateriellen Gegenständen zum Nutzen der Allgemeinheit regelt (DIN 820 Teil 1). Dieses autorisierte Gremium ist seit 1975 das „Deutsches Institut für Normung (DIN)“, welches laut Vertrag mit der BRD die zuständige Normierungsorganisation ist. Das DIN vertritt die BRD auch bei internationalen Organisationen zum Zwecke grenzübergreifender Normierungen. Die auf dieser Ebene erstellten Normen werden auch *de jure Standards* genannt. Andere Spezifikationen, die von nicht hoheitlichen Gremien erstellt werden, bezeichnet man als *de facto Standards*. Ein Standard bezeichnet also eine Spezifikation, die sich in

bestimmten Bereichen etabliert hat. Im GIS-Bereich hat sich beispielsweise u.a. das Drawing Exchange Format (dxf) als Standard für den Datenaustausch bewährt. Standards können durch eine Institution, bzw. ein autorisiertes Gremium in eine Norm übernommen werden.

Daher stellt sich die Frage, wer normt und wer Standards setzt? Die folgende Abbildung gibt einen Überblick über die verschiedenen Normierungsgremien auf nationaler und internationaler Ebene.

<b>Nationale Normierungsgremien</b>  DIN Deutsches Institut für Normung e.V. ON Österreichisches Normungsinstitut SNV Schweizerische Normen-Vereinigung ANSI American National Standards Institute	<b>Europäische Normierungsgremien</b>  CEN European Committee for Standardisation CENELEC Europ. Komitee für elektrotechn. Normung
	<b>Internationale Normierungsgremien</b>  ISO International Organization for Standardisation IEC International Electrotechnical Commission

Abbildung 3.1: Übersicht der Normierungsgremien

Das DIN organisiert sich in zurzeit 76 Normenausschüsse mit darunter angeordneten Fachbereichen. Diesen Fachbereichen sind jeweils Projektbereiche zugeordnet. Das Geoinformationswesen ist dem Normenausschuss Bauwesen (NABau) und dort dem Fachbereich Vermessungswesen / Geoinformation zugeordnet. In diesem Fachbereich befindet sich der Projektbereich Kartographie und Geoinformation.

### 3.1.1.1 World Wide Web Consortium (W<sup>3</sup>C)

Damit Informationen auf Grundlage der Internettechnologie ausgetauscht werden können, bedarf es Standards, auf die sich sowohl Anbieter von Informationen als auch die Nutzer berufen können. Die Entwicklung solcher Grundlagen wird dem World Wide Web Consortium (W<sup>3</sup>C) zugeschrieben, welches 1994, fünf Jahre nach der Gründung des WWW ebenfalls durch Tim Berners-Lee, ins Leben gerufen wurde (Eberhardt, Fischer, 2003). Seitdem ist er auch Vorsitzender des Gremiums zur Standardisierung des World Wide Web betreffender Techniken (Wikipedia, 2006c). Das Konsortium beschäftigt sich sowohl mit der Weiterentwicklung von bestehenden als auch mit der Neuentwicklung von Standards. Bezüglich letzterer hat sich das W<sup>3</sup>C verpflichtet, ausschließlich Technologien zu verwenden, die frei von Patentgebühren sind (W<sup>3</sup>C, 2007). Hauptziel des W<sup>3</sup>C ist es, durch die Entwicklung von Protokollen und Richtlinien einen industrieweiten Konsens über die Web-Technologie herbeizuführen und das World Wide Web zur Nutzung seiner vollen Kapazität zu führen. Zu den etwa 430 Mitgliedern (Stand: Nov. 2006) gehören Industrieunternehmen, Universitäten, Regierungen, Standardisierungsgremien oder auch allgemeine Internetnutzer. Zur Leitung der technischen Aktivitäten und der Arbeitsabläufe hat das W<sup>3</sup>C ein Team eingerichtet, welches aus über 60 Forschern und Ingenieuren besteht. Die meisten Team-Mitglieder arbeiten an den drei gastgebenden Institutionen MIT/CSAIL (USA), ERCIM (Frankreich) und Keio Universität (Japan). Dieses Team beobachtet Marktentwicklungen, neue Technologien und die Tätigkeiten der verwandten Standardisierungsgremien und gibt dadurch dem W<sup>3</sup>C eine Orientierung vor. Weiterhin werden über das Team die Ergebnisse des Konsortiums kommuniziert und der Öffentlichkeit präsentiert.

Die Abbildung 3.2 zeigt den W<sup>3</sup>C Technology Stack, der als Modell mit 2 Ebenen (Layers) die Standards des W<sup>3</sup>C aufzeigt. Dabei wird das WWW als „One Web“ bezeichnet und als Anwendung verstanden, die auf der Internettechnologie aufsetzt. Diese Illustration zeigt eine Sichtweise der Web Infrastruktur mit einem besonderen Fokus auf die Arbeiten des W<sup>3</sup>C. Die Web Architektur ist in verschiedene Ebenen eingeteilt, die jeweils aufeinander aufbauen. Von unten nach oben enthalten diese Ebenen:

- URI/IRI, HTTP
- Web Architectural Principles

- XML Infosets; RDF(S) Graphs
- XML, Namespaces, Schemas, XQuery/XPath, XSLT, DOM, XML Base, XPointer, RDF/XML, SPARQL

Über diesen Layern befinden sich 6 Bereiche, die die W<sup>3</sup>C-Aktivitäten in Gruppen einteilen: Web Applications, Mobile, Voice, Web Services, Semantic Web, und Privacy.

Das orangene Banner, welches die horizontale Koordination darstellt, verknüpft die 4 Bereiche: Web Accessibility, Internationalization, Mobile Access, Device Independence, und Quality Assurance.

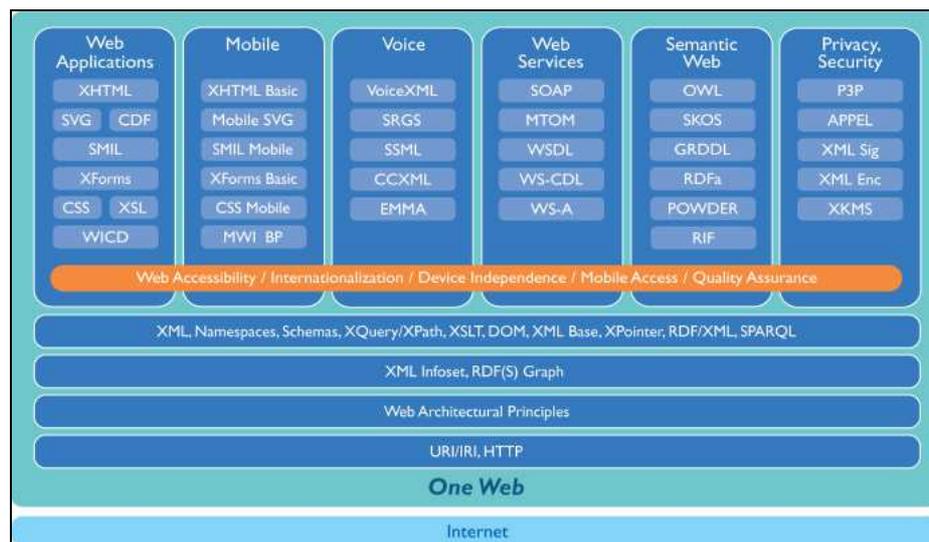


Abbildung 3.2: Der W<sup>3</sup>C Technology Stack (W<sup>3</sup>C, 2007)

Die Ziele des W<sup>3</sup>C lassen sich laut (Donaubauer, 2004) in sieben Punkten zusammenfassen:

- universelle Zugangsmöglichkeiten (stationär und mobil sowie über alle Arten von Endgeräten).
- das semantische Web (Computer sollen die Bedeutung von Daten verstehen können).
- Vertraulichkeit (Aufbau eines "Web of Trust", das es den Benutzern ermöglicht, Verantwortung oder Rechenschaft für ihre bereitgestellten Ressourcen wie Daten und Dienste, zu übernehmen, bzw. dies für die Ressourcen anderer Nutzer einzufordern).
- Interoperabilität (das W<sup>3</sup>C, als eine herstellerunabhängige Organisation, fördert die Interoperabilität, indem es offene, nicht proprietäre Computersprachen und Protokolle entwirft).
- Entwicklungsfähigkeit (bei der Entwicklung von Standards für das WWW werden die Prinzipien der Einfachheit, der Modularität und der Erweiterbarkeit angestrebt).
- Dezentralisierung (zur Reduzierung der Anfälligkeit des WWW als Ganzes, Fehlertoleranz, Vermeidung von Engpässen).
- attraktivere Multimedia (mehr Interaktivität, Berücksichtigung der Wünsche des Endbenutzers nach mehr multimedialem Inhalt, u.a. durch Sprachen wie Scalable Vector Graphics (SVG)).

Das W<sup>3</sup>C hat nicht denselben Status wie beispielsweise das DIN und kann daher keine Normen erstellen und veröffentlichen. Aus diesem Grund werden die de facto Standards des W<sup>3</sup>C auch *Recommendations* (Empfehlungen) genannt. Zur Verabschiedung können Standardisierungsvorschläge von außen beim W<sup>3</sup>C eingereicht

werden oder sie werden in einer eigenen Arbeitsgruppe (Working Group) entwickelt. Bevor eine Empfehlung veröffentlicht wird, sind verschiedene Vorstufen erforderlich. Die folgende Übersicht stellt die verschiedenen Stufen einer Empfehlung dar.

Stufe	Beschreibung
Note	Mitteilung einer Arbeitsgruppe des W3C oder eingereichte Vorschläge.
Working Draft (WD)	Zur Diskussion freigegebener Arbeitsentwurf.
Candidate Recommendation (CR)	Mehrere WD führen zu einer CR, wobei durch Implementierungen die Praxistauglichkeit nachzuweisen ist.
Proposed Recommendation (PR)	Überprüfung der Spezifikation durch alle Mitglieder und der Direktion und Abschlussbeurteilung.
Recommendation (Rec)	Eine verabschiedete Spezifikation, die nicht mehr verändert wird. Änderungen führen zu einer neuen Version.

Tabelle 3.1: Stufenübersicht einer Empfehlung beim W<sup>3</sup>C (Kleber, 2006)

### 3.1.1.2 International Organization for Standardisation (ISO)

Die International Organization for Standardisation (ISO) wurde 1946 in Genf gegründet und hat heute mehr als 140 Mitglieder. Die Mitglieder dieser Organisation sind nicht Einzelpersonen oder Firmen, sondern die nationalen Normierungsinstitute. Für die BRD ist beispielsweise das DIN Mitglied bei der ISO. Die Mitglieder entsenden Fachleute in die Normierungsgremien der ISO. Um Doppelarbeit auf nationaler und internationaler Ebene zu verhindern, werden nationale „Spiegelgremien“ gebildet, die den jeweiligen internationalen Komitees zuarbeiten. Die ISO strukturiert sich in Technische Komitees, Subkomitees und Arbeitsgruppen. Die Technischen Komitees, wie etwa das TC 211 Geographic Information / Geomatics, sind aus einer Vielzahl von Arbeitsgruppen aufgebaut.

Die Entwicklung einer Norm bei ISO erfolgt durch die Technischen Komitees (TC) und Subkomitees (SC) in sechs Schritten:

Stufe	Beschreibung
New Work Item Proposal (NWIP)	Vorschlag für eine Norm (Proposal stage)
Working Draft (WD)	Vorbereitung der Norm (Preparatory stage)
Committee Draft (CD) Draft International Standard (DIS)	Komiteebearbeitung (Committee stage)
Final Draft International Standard (FDIS)	Einholung von Erkundigungen (Enquiry stage)
International Standard (IS)	Genehmigung der Norm (Approval stage)
	Publikation der Norm (Publication stage)

Tabelle 3.2: Stufenübersicht einer Normierung bei ISO

Das ISO/TC211 wurde 1994 gegründet. Die bis dahin bei CEN/TC 287 auf europäischer Ebene erarbeiteten Normen wurden abgeschlossen und als Vornormen veröffentlicht. Dadurch flossen die Ergebnisse des CEN in die internationalen Normungsaktivitäten ein.

### 3.1.1.3 OpenGeospatial Consortium (OGC)

1994 ist das OpenGIS Consortium gegründet worden, welches im September 2005 in das OpenGeospatial Consortium (OGC) umbenannt worden ist. Oberstes Ziel dieser Organisation ist es, Schnittstellenspezifikationen für interoperable Geoinformationssysteme zu entwickeln. Das folgende Zitat von Carl Reed, Executive Director (Specification Program), zeigt die grundlegende Intention des OGC (Reed, 2004):

***“A world in which everyone benefits from geographic information and services made available across any network, application, or platform”.***

Die derzeit über 330 Mitglieder des OGC setzen sich zusammen aus Behörden, Firmen, Organisationen und Universitäten, die im Bereich Geoinformatik arbeiten. Verschiedene Mitgliedschaftsstufen spiegeln den Grad der Mitsprachemöglichkeit wieder. Strategische Mitglieder, wie die NASA oder das amerikanische Großunternehmen Lockheed Martin, sind an der Gestaltung des Konsortiums beteiligt, während prinzipielle Mitglieder, wie ESRI oder Intergraph, sowohl bei Planungen als auch bei technischen Inhalten mitbestimmen können. Ein assoziiertes Mitglied wirkt ohne Stimmberechtigung bei technischen Inhalten mit. In Abhängigkeit von der Mitgliedschaft gehören die Mitglieder entweder dem Planungs- (PC) oder dem technischen (TC) Komitee an. Letzteres gliedert sich in verschiedene Arbeitsgruppen (WG) und speziellen Interessengruppen (SIG). In den Arbeitsgruppen werden abstrakte Spezifikationen erstellt. Zu diesen werden Ausschreibungen formuliert, nach denen konkrete Vorschläge zur Implementierung der Spezifikationen eingereicht werden können. Dazu begleitend erstellt das Implementierungsprogramm (IP) gesponserte Testbeds, bei denen OGC-Mitglieder durch konkrete Anwendungsfälle interoperable Prototypen entwickeln und die Implementierungsspezifikationen überprüfen und verbessern.

Die Verabschiedung einer Spezifikation erfolgt durch Abstimmung der Mitglieder. Dabei wird ein möglichst großer Konsens angestrebt, um eine möglichst große Akzeptanz einer Spezifikation zu erreichen. Die Mitgliedsfirmen des OGC versichern mit der Verabschiedung einer Spezifikation, diese in ihre kommerziellen Produkte zu implementieren.

Um Doppelarbeit zwischen OGC und ISO zu vermeiden, wurde 1998 ein Abkommen geschlossen, nach dem das OGC sukzessive die ISO-Normen als abstrakte Spezifikationen übernehmen und daraus Implementierungsspezifikationen entwickeln wird. Diese wiederum können bei ISO eingereicht und nach einem Abstimmungsprozess zu einer internationalen Norm werden. Ebenso gilt es, Doppelarbeit auf nationaler und internationaler Ebene zu verhindern und dafür Sorge zu tragen, dass sich verabschiedete Normen und Standards von verschiedenen Organisationen nicht widersprechen, sondern ergänzen. Auch dazu ist eine enge Zusammenarbeit der jeweiligen Organisationen notwendig. Die folgende Darstellung macht dies deutlich.

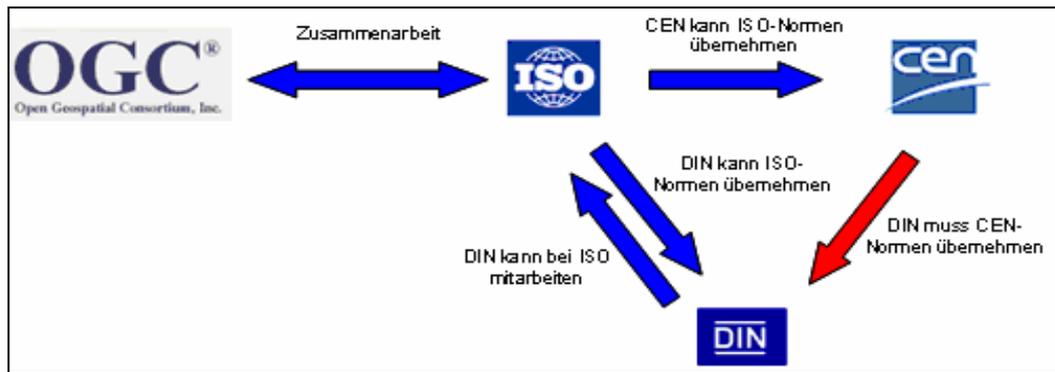


Abbildung 3.3: Zusammenarbeit der Normierungsgremien (Kleber, 2006)

Die enge Zusammenarbeit zwischen dem OGC und der ISO zeigt sich dadurch, dass viele Mitarbeiter in entscheidenden Positionen beider Gremien vertreten sind. Praktisch sieht die Zusammenarbeit so aus, dass das OGC abstrakte Spezifikationen durch die neu erstellten ISO-Normen ersetzt hat und anders herum das OGC Implementierungsspezifikationen bei ISO einreicht, um den Status einer de jure Norm zu erlangen. Die Zusammenarbeit wird durch die gremienübergreifende Gruppe TC211 – OGC coordination group (TOCG) organisiert.

### 3.1.2 Auszeichnungssprachen

Durch den Trend zu offenen, interoperablen Systemen in der Informatik, werden Auszeichnungssprachen, auch Markup-Sprachen genannt, für die Kodierung von Daten immer interessanter. Ihr Vorteil ist die einheitliche Basis-Syntax, die system- und anwendungsunabhängig ist und die Möglichkeit zum selbstbeschreibenden Datenaustausch bietet. Solche Sprachen stellen beispielsweise die Hypertext Markup Language (HTML), die eXtensible Markup Language (XML) und die Geography Markup Language (GML) dar (Donaubauer, 2004). Wird ein Dokument mit einer Dokumentenauszeichnungssprache erstellt, so werden die verschiedenen Bestandteile des Dokumentes durch eine bestimmte Notation gekennzeichnet. In HTML sind das beispielsweise die spitzen Klammern (<>). Dokumente, die mit einer Auszeichnungssprache erstellt worden sind, bestehen also aus Auszeichnungen und Text. Da Auszeichnungen die Definition zur Interpretation des Textes bilden, kann eine Anwendung, die das Dokument liest, diesen Text darstellen. Mutter der Metasprachen HTML und XML ist die Standard Generalized Markup Language (SGML). Dabei handelt es sich um eine international normierte, herstellerunabhängige Dokumentenbeschreibungssprache für den Inhalt und die logische Struktur von Dokumenten. SGML geht aus der Entwicklung der Generalized Markup Language (GML) hervor. Diese wurde von IBM als Sprache zur Auszeichnung von Dokumenten, die auf mehreren Informationssystemen benutzt werden konnten, entwickelt. Zu Beginn der 80er Jahre wurde mit SGML ein Industriestandard geschaffen, der 1986 durch die ISO als Standard (ISO 8879) angenommen wurde. SGML wurde von Charles Goldfarb mit dem Ziel entwickelt, eine logische Struktur von Textdokumenten zu schaffen, wobei die Darstellung vom Inhalt strikt getrennt sein sollte. So wurden nur Regeln definiert, nach denen elektronische Dokumente aufgebaut werden. Um SGML zu nutzen, wurden Definitionen von Strukturen und Inhalten erforderlich. Solche Definitionen werden als „Document Type Definition“ (DTD) bezeichnet.

Anfang der 90er Jahre kam der öffentliche Durchbruch für die generische Codierung von Dokumenten mit der Entwicklung der HyperText Markup Language durch das Europäische Forschungsinstitut für Teilchenphysik (CERN). Mit HTML wurde ein SGML-Dokumenttyp für Dokumente geschaffen, der kompakt, aber dennoch effizient war. Es bereitete nunmehr keine Probleme, Software für diese einfache Markup-Sprache zu entwickeln oder Dokumente zu kodieren. HTML war ein großer Schritt für das Web und die Auszeichnungssprachen, da das öffentliche Interesse an elektronischen Dokumenten dadurch wuchs, dass HTML aus einem festen überschaubaren Satz von Tags bestand und einfach zu erlernen war.

Trotz des enormen Schubs für das Web und der breiten Akzeptanz weltweit hatte HTML einige Nachteile. HTML bietet keine Möglichkeit, eigene Befehle zu definieren. Auch sind umfangreichere Formatierungen mit dieser Metasprache nicht möglich. Daher ist es als generelles Repository-Format nicht umfassend genug, vor allem da es keine Definition eigener DTDs zulässt.

Die Komplexität von SGML und die Unzufriedenheit mit den existierenden Standards war der erste Ansatzpunkt des W<sup>3</sup>C Mitte der 1990er Jahre für der Entwicklung einer erweiterbaren (Auszeichnungs-)Sprache, die die Flexibilität von SGML sowie die Einfachheit und Akzeptanz von HTML in sich vereinen sollte. Der zweite Ausgangspunkt zur Entwicklung einer neuen Sprache waren neue Anforderungen an Auszeichnungssprachen und Grenzen von HTML. Die zunehmende Übermittlung von Daten zur Anzeige und Verarbeitung von Informationen sowie der Wunsch nach einer Möglichkeit, Daten problemspezifisch beliebig zu strukturieren, machten eine neue, flexiblere und vor allem erweiterbare Sprache unumgänglich. Die Bemühungen zur Schaffung einer solchen Sprache endeten schließlich in der „eXtensible“, also erweiterbaren Markup-Language.

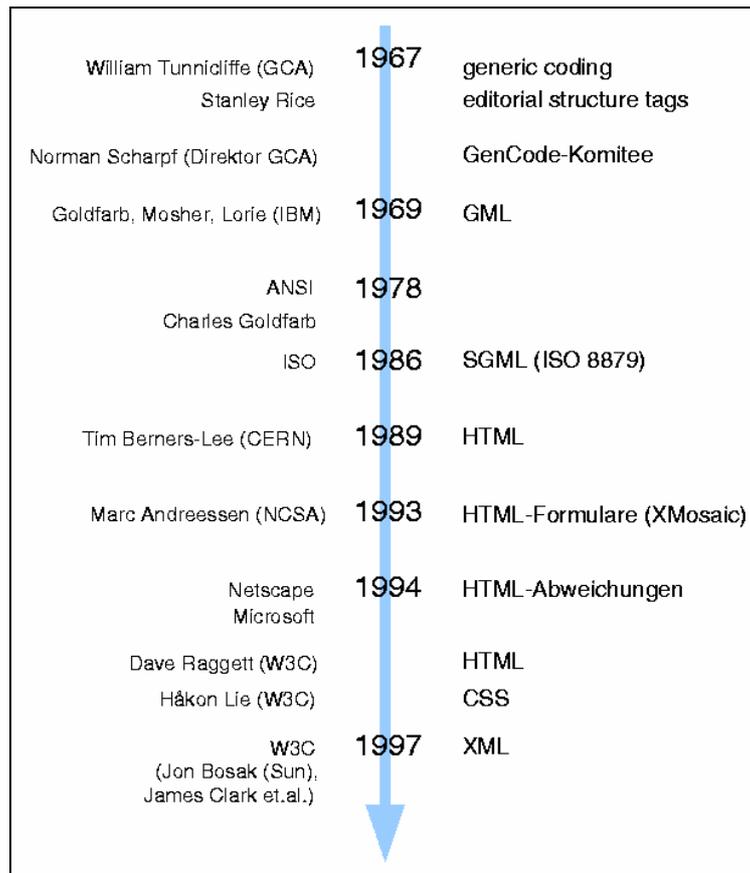


Abbildung 3.4: Die Entwicklung der Auszeichnungssprachen (Mintert, 1998)

### 3.1.2.1 HyperText Markup Language (HTML)

Die Hypertext Markup Language (HTML) wird oftmals auch als Hypertext bezeichnet und ist eine textbasierte Auszeichnungssprache. Die bedeutet zum einen, dass sie mit einem gewöhnlichen Texteditor wie beispielsweise dem Notepad unter Windows erstellt werden kann und zum anderen können Inhalte wie Texte, Bilder und Hyperlinks zu anderen Quellen in Dokumenten dargestellt werden. Solche HTML-Dokumente werden von einem Webbrowser dargestellt, wobei ein Dokument neben den darzustellenden Inhalten weitere Metainformationen enthalten kann. Diese zusätzlichen Informationen geben beispielsweise Auskunft über den Autor der Seite, die im Text verwendete Sprache oder Suchbegriffe, anhand derer das Dokument im WWW gefunden werden kann.

Jedes HTML-Dokument hat einen bestimmten Aufbau und folgt einer festen Syntax, wodurch deutlich wird, dass eine gewisse Unflexibilität vorhanden ist. Der Vorteil ist allerdings, dass diese Syntax von jedem Webbrowser ohne zusätzliche Informationen lesbar und interpretierbar ist. Ein HTML-Dokument beginnt immer mit dem Tag <html> und endet mit </html>. Dazwischen ist zunächst der Dateiheder mit <head> und </head> angeordnet. Zudem gehört der so genannte Body mit dem Tag <body> ebenfalls zum festen Bestandteil eines

HTML-Dokumentes. Innerhalb des Bodys stehen die eigentlichen Inhalte des Dokumentes, die nach außen dargestellt werden sollen. Die folgende Abbildung zeigt den typischen Aufbau eines HTML-Dokumentes.

```
<html>
  <head>
    <Title>Hier steht der Titel des Dokumentes</Title>
    ... (weitere Metainformationen)
  </head>
  <body>
    <h1>Das ist eine Überschrift</h1>
    ... (weitere darzustellende Elemente)
  </body>
</html>
```

Abbildung 3.5: Aufbau von HTML-Dokumenten

Neben diesem sehr einfachen Aufbau können sich HTML-Dokumente auch Programmiersprachen bedienen, um flexiblere Inhalte darzustellen. Durch den beispielhaften Tag `<script language="JavaScript" type="text/javascript"> ... </script>` innerhalb des Bodys wird auf ein Programm in der Sprache JavaScript verwiesen. Auf diese Weise kann der Quellcode einer Scriptsprache in ein HTML-Dokument eingebettet werden.

### 3.1.2.2 eXtensible Markup Language (XML)

XML wurde seit 1996 entwickelt und beruht in weiten Teilen auf dem Gedankengut der Standard Generalized Markup Language (SGML). Bei der Entwicklung von XML bezog man sich stark auf die SGML-Spezifikation und konzentrierte sich darauf, das funktionierende Konzept von SGML zu vereinfachen. Mit SGML war es bereits möglich, eigene Auszeichnungssprachen zu entwickeln, jedoch war diese Sprache viel zu komplex. Die neue Sprache sollte leichter anzuwenden sein als SGML und musste daher einfacher gestaltet werden. Das Erstellen und Gestalten von Dokumenten sollte mit einfachen und verbreiteten Werkzeugen möglich sein und auch die maschinelle Verarbeitung und Transformation von Dokumenten sollte vereinfacht werden, da diese Sprache für die breite Masse und nicht nur für große Unternehmen gedacht war. In XML treffen sich die Grundlagen und Möglichkeiten von SGML mit der interneterprobten Technologie von HTML. XML blieb somit auch aufwärts kompatibel, d.h. Unternehmen, die schon länger oder immer noch mit SGML-Umgebungen und Werkzeugen arbeiteten, konnten auch neuere XML-Dokumente verarbeiten. 1998 erklärte das W<sup>3</sup>C XML in der Version 1.0 zu einer Empfehlung. Folgende Ziele standen nach (W<sup>3</sup>C, 2007) beim Entwurf von XML im Fokus:

- XML soll sich im Internet auf einfache Weise nutzen lassen.
- XML soll ein breites Spektrum von Anwendungen unterstützen.
- XML soll zu SGML kompatibel sein.
- Es soll einfach sein, Programme zu schreiben, die XML-Dokumente verarbeiten.
- Die Zahl optionaler Merkmale in XML soll minimal sein, bestenfalls Null.

- XML-Dokumente sollten für Menschen lesbar und angemessen verständlich sein.
- Der Entwurf von XML sollte zügig abgefasst sein.
- Der Entwurf von XML sollte formal und präzise sein.
- XML-Dokumente sollen leicht zu erstellen sein.
- Knappheit von XML-Markup ist von minimaler Bedeutung.

In ihrem Aufbau sehen sich HTML und XML auf den ersten Blick sehr ähnlich. Dennoch gibt es gravierende Unterschiede. So ist HTML eine Auszeichnungssprache, die Web-Seiten beschreibt. Jeder Web-Browser kann ein HTML-Dokument darstellen und somit auch interpretieren. Basis dafür ist, dass der Browser alle Bestandteile von HTML samt ihrer Bedeutung kennt. Soll eine Zeichenkette in fetter Schrift dargestellt werden, so findet sich folgender Syntax im Dokument wieder: `<b>...</b>`. Es ist also lediglich bekannt, wie die Information dargestellt werden soll, die Beschreibung dieser geht allerdings verloren. Hier kommt der Unterschied zu XML zum Tragen. Bei XML lassen sich Attribute und Tags beliebig benennen und einsetzen. Das oben erwähnte `<b>` kann in XML für „bold“, „Bemerkung“ oder auch „Buchnummer“ stehen, wobei diese Bedeutung nichts mit dem Buchstaben „B“ gemein haben muss. Es existieren also keine Tags mit fester Bedeutung. XML verwendet Tags zur Abgrenzung von Daten, deren Interpretation von der Anwendung vorgenommen wird, die die Daten verarbeitet. Auf diese Weise können die Strukturen beliebig komplex sein und Tag- und Attributnamen nach Belieben erstellt werden. Um diesen Aufbau bzw. diese Struktur festzulegen, gibt es Document Type Definitions (DTD). Die Dateidefinitionen werden in einer separaten Datei mit der Endung \*.dtd abgelegt (Noack, 2001). So wird die Struktur einer XML-Datei verbindlich und zentral festgelegt. Anhand der DTD können XML-Dokumente des gleichen Typs bezüglich ihrer Struktur geprüft werden. Diese Prüfung übernehmen viele Editoren und XML-verarbeitende Tools auf Wunsch automatisch. Die Verwendung von DTDs ist jedoch trotz der beschriebenen Vorteile nicht obligatorisch (Näf, 2007).

Durch die Anlehnung des DTD-Konzeptes an SGML ist die Definition feingliedriger Strukturen für XML-Dokumente nicht möglich. XML-Syntax wurde ebenfalls nicht verwendet, so dass ein Zugriff über das XML-„Document Object Model“ (DOM), das als sprach- und plattformunabhängige Schnittstelle gilt, nicht möglich ist. Der zweite, etwas neuere, Standard zur Dokumentenmodellierung wird als XML Schema Definition (XSD) oder kurz XML-Schema bezeichnet. Ein XML-Schema beschreibt das Format eines XML-Dokumentes. Hierbei werden XML-Fragmente, so genannte Vorlagen (Templates), benutzt, um die Struktur eines Dokumentes festzulegen. Der Vorteil von XML-Schemata gegenüber DTD's liegt darin, dass es sich bei den Schemata selbst um XML handelt und diese somit mit den gleichen Editoren bearbeitet werden können, wie die XML-Dokumente. In diesem Zusammenhang wird der Begriff der Validierung von XML-Dokumenten eingeführt. Ein Dokument ist also gültig hinsichtlich eines Schemas (schema valid), wenn es zu diesem Dokument ein Schema gibt und das Dokument entsprechend aufgebaut ist (Kleber 2005).

Ein XML-Dokument kann ähnlich wie ein HTML-Dokument von einem Web Browser gelesen und dargestellt werden. Voraussetzung hierfür ist die Verwendung einer Bibliothek, die ihre Dienste durch eine „Application Programmer Interface“ (API) Anwendungen zur Verfügung stellt. In einer Microsoft Systemumgebung bildet die Datei „MSXML.dll“ diese Bibliothek und stellt den XML-Parser zur Verfügung. Es handelt sich hier um einen validierenden Parser, der eine Validierung und Überprüfung der Wohlgeformtheit des XML-Dokumentes durchführt. Wohlgeformtheit wird erfüllt, wenn gewisse syntaktische Regeln innerhalb des Dokumentes eingehalten werden.

Zur Darstellung eines XML-Dokumentes eines bestimmten Typs, wird die Vorschrift benötigt, wie die Informationen im XML-Dokument darzustellen sind. Eine solche Vorschrift wird mit der „eXtensible Stylesheet Language“ (XSL) bzw. „eXtensible Stylesheet Language Transformation“ (XSLT) aufgestellt. Im Allgemeinen bezeichnet man eine solche Vorschrift als XSL Stylesheet. Stylesheet steht im Englischen für "Druckformatvorlage", "Layout-Datei" oder "Stilvorlage" (Eberhardt, Fischer, 2003). Des Weiteren bedarf es eines Instrumentes, welches das XML-Dokument nach der Vorschrift darstellen oder in ein anderes Dokument umwandeln kann.

Dazu gibt es XSL-Prozessoren, die als Werkzeug das XML-Dokument anhand einer XSL(T)-Vorschrift in ein anderes Format umwandeln können.

Die Abbildung 3.5 zeigt den Zusammenhang der beschriebenen Komponenten und Formate.

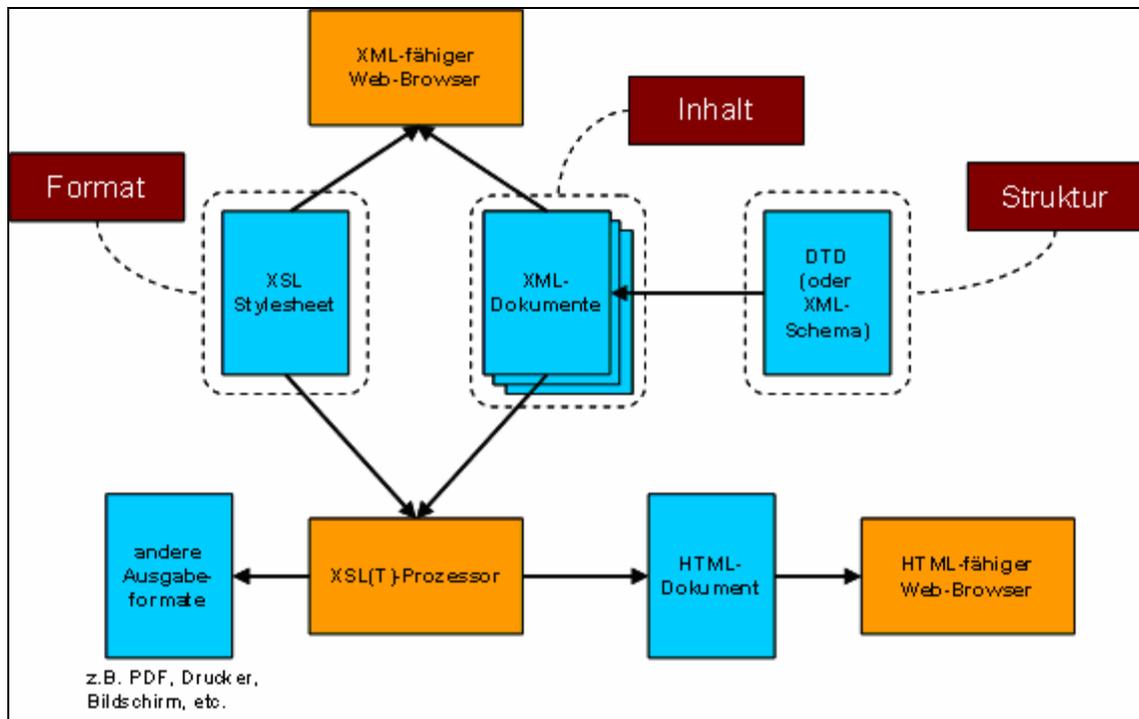


Abbildung 3.6: Übersicht der XML-Komponenten (Näf, 2007)

Zur Verdeutlichung wird abschließend ein Beispiel einer XML-Datei mit einer Typdefinition in Form einer DTD-Datei und einer XSL-Datei dargestellt. Die letzte Abbildung zeigt die Darstellung des XML-Dokumentes in Abhängigkeit von dem Stylesheet in einem Web Browser.

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="beispiel.xsl"?>
<!DOCTYPE ADRESSENLISTE SYSTEM "Beispiel.dtd">
<ADRESSENLISTE>
  <KONTAKT>
    <NAME>Peter Mueller</NAME>
    <STRASSE>Hauptstrasse 47</STRASSE>
    <PLZ>08150</PLZ>
    <ORT>MUSTERHEIM</ORT>
    <TELEFON type="Mobil">0049 1234567</TELEFON>
    <TELEFON type="Privat">0049 9876543</TELEFON>
    <EMAIL>Peter.Mueller@Musterheim.de</EMAIL>
  </KONTAKT>
</ADRESSENLISTE>

```

Abbildung 3.7: Beispiel einer XML-Datei

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ADRESSENLISTE (KONTAKT)*>
<!ELEMENT KONTAKT (NAME, STRASSE, PLZ, ORT, TELEFON*, EMAIL)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT STRASSE (#PCDATA)>
<!ELEMENT PLZ (#PCDATA)>
<!ELEMENT ORT (#PCDATA)>
<!ELEMENT TELEFON (#PCDATA)>
<!ATTLIST TELEFON
  type (Mobil | Privat) #REQUIRED>
<!ELEMENT EMAIL (#PCDATA)>

```

Abbildung 3.8: Beispiel einer DTD-Datei

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/XSL" >
  <xsl:template match="/">
    <html>
      <head>
        <title>Adressenliste</title>
      </head>
      <body>
        <table cellpadding="30" cellspacing="30">
          <xsl:for-each select="ADRESSENLISTE/KONTAKT">
            <tr>
              <td bgcolor="orange"><b>
                <xsl:value-of select="NAME"/></b><br/>
                <xsl:value-of select="STRASSE"/><br/>
                <xsl:value-of select="PLZ"/>
                <xsl:value-of select="ORT"/><br/>
                <xsl:value-of select="EMAIL"/><br/>
                <xsl:for-each select="TELEFON">
                  <xsl:value-of select="@type"/>:
                  <xsl:value-of select="."/>
                </xsl:for-each>
              </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>

```

Abbildung 3.9: Beispiel einer XSL-Datei

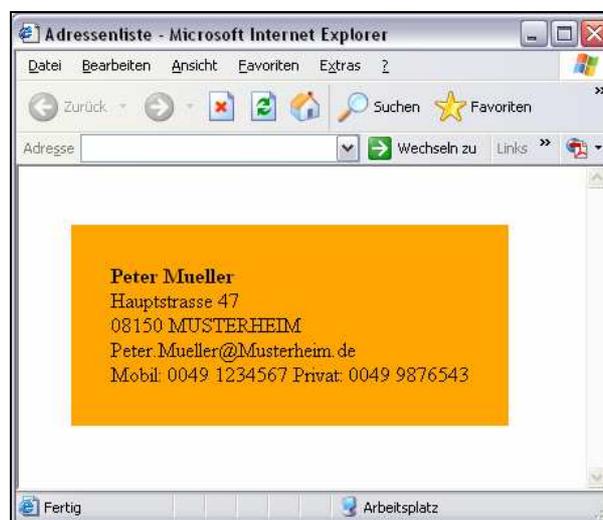


Abbildung 3.10: Darstellung einer XML-Datei in einem Web Browser

Die XML-Technologie bietet zusätzlich zu den bereits genannten Bestandteilen weitere Anwendungen und Methoden zur Bearbeitung eines entsprechenden Dokumentes. Die folgende Liste zeigt die wichtigsten XML-verwandten Technologien (Noack, 2001), (Eberhardt, Fischer, 2003).

- Namespaces (Namensräume) bieten eine einfache Möglichkeit, Element- und Attributnamen in XML eindeutig zu benennen.
- DTD (Document Type Definition) ist eine Methode mit der man Elementtypen, Attribute und Entities innerhalb eines XML-Dokumentes definieren kann.
- XHTML (Extensible Hypertext Markup Language) beinhaltet die Neuformulierung von HTML 4.0 in XML 1.0 und ist eine Sprache zum Erstellen von Web-Seiten.
- XSL (Extensible Stylesheet Language) ermöglicht das Erstellen von Stylesheets.
- XSLT (Extensible Stylesheet Transformation Language) erlaubt mit Hilfe eines XSL-Stylesheets ein XML-Dokument in ein anderes Format zu transformieren.
- XLink (Extensible Linking Language) ist eine Spezifikation für Hyperlinks. Der Adressierungsstandard XPointer ist inbegriffen.
- XPointer (Extensible Pointer Language) gibt an, wie Adressen in XLink-Ausdrücken verwendet werden. Mit ihm wird festgelegt, wie das Format des Teils eines Hyperlinks, der auf eine Sprungmarke eines XML-Dokuments verweist, auszusehen hat.
- XQL (Extensible Query Language) ist eine Abfragesprache, die XML als Datenmodell verwendet.
- XPath (Extensible Path Language) ist eine Sprache, die eine Ausdrucksweise festlegt, mit der bestimmte Elemente in einem Dokument selektiert werden können. Dies ist die Grundlage von XSL und XPointer.

Die folgende Abbildung zeigt die Zusammenhänge der XML-verwandten Technologien.

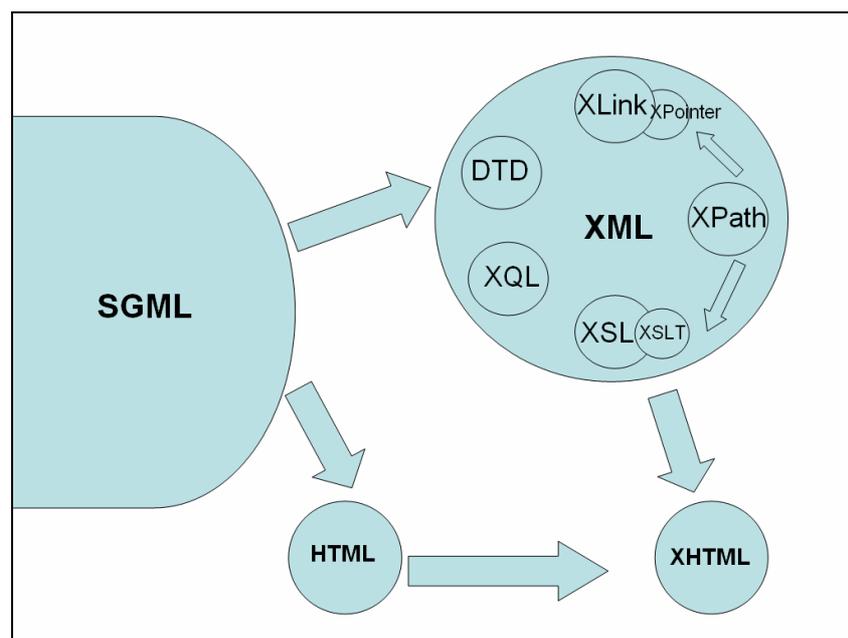


Abbildung 3.11: XML und verwandte Technologien nach (Noack, 2001)

Abschließend lassen sich die folgenden Leistungsmerkmale von XML, die zu seiner Verbreitung beigetragen haben, zusammenfassen:

- Mit XML können beliebige Informationen gespeichert und dargestellt werden, und das in einer Form, die auf die jeweilige Anwendungssituation angepasst ist.
- XML ist ein offener Standard und herstellerunabhängig.
- Durch die Verwendung von Unicode als Standard-Zeichensatz unterstützt XML eine Vielzahl von Schriften.
- XML ist durch seine einfache und klare Syntax und seine eindeutige Struktur einfach von Menschen und Programmen zu lesen und analysieren.
- Durch Trennung der Formatinformationen vom Dokumentinhalt wird die Informationsstruktur nicht beeinflusst.
- XML-Dokumente sind weitestgehend selbst dokumentierend, welches die Lesbarkeit enorm erhöht.
- Mit XML können Daten plattformunabhängig ausgetauscht werden.

### 3.1.2.3 Geography Markup Language (GML)

Die Geography Markup Language (GML) ist eine XML-basierende Kodierungssprache, um raumbezogene und nicht-raumbezogene Informationen zu transportieren und zu speichern. GML wurde entwickelt, um Interoperabilität beim Transfer von Geodaten zwischen verteilten Softwarekomponenten zu erreichen. Der praktische Einsatz von GML beschränkt sich demnach nicht nur auf den Austausch statischer Informationen zwischen System AB und System XY, sondern bildet auch die Basis für den serviceorientierten Datenaustausch über das Internet (Kleber, 2005).

Der Bezug zwischen den Standards der Informatik und GML wird in der folgenden Tabelle dargestellt (Donaubauer, 2004).

Standard der Informatik	Rolle in der GML-Spezifikation
XML	Grundlage für XML-Dokumente, GML-Basis- und Anwendungsschemata
XML-Schema	Grundlage für GML-Basis- und GML-Anwendungsschemata
XLink	Repräsentation von Referenzen zwischen GML Features und deren Geometrie als GML Subfeatures
URI	u.a. Struktur und Syntax der Bezeichner für Raumbezugssysteme
XPointer	Verweis auf Teile eines GML Anwendungsschemas

Tabelle 3.3: Bezug von Informatikstandards zu GML

Da sich GML in der Praxis bereits bewährt hat, wurde von TC211 ein neues Projektteam eingesetzt, das die Norm 19136 ausarbeitet. Diese definiert die modellbasierte Anbindung von GML an das übrige Normenwerk und legt fest, wie das GML-Format der Transferdatei einer bestimmten Anwendung aus dem Datenmodell dieser Anwendung hergeleitet werden kann. Die folgende Darstellung zeigt die zeitliche Entwicklung von GML mit

seinen verschiedenen Versionen. Daraus ist ersichtlich, dass der Bedarf einer solchen Kodierungssprache einerseits enorm ist und andererseits in den letzten Jahren sprunghaft angestiegen ist. Eng verzahnt mit der Entwicklung der GML ist die Betrachtung der Arbeiten der Normierungsgremien. Dies hat dazu geführt, dass derzeit bei ISO an der Norm 19136 Geography Markup Language gearbeitet wird. Wenn diese Entwicklungen abgeschlossen sind, wurde eine OGC-Spezifikation zu einer ISO-Norm (Portele, 2005).

Datum	Version	Anmerkung
05/2000	GML 1.0	Basiert auf DTD
02/2001	GML 2.0	Basiert auf XML-Schemas
01/2002	GML 2.1.1	Revisionen
08/2002	GML 2.1.2 (66 Seiten Spezifikation)	Revisionen
01/2003	GML 3.0 (528 Seiten Spezifikation)	Zusätzlich Topologie, Dimension, Zeit, Maßeinheiten, etc.
04/2004	GML 3.1	Einreichung bei ISO TC211/WG4 ISO/CD 19136
12/2006	GML 3.2.1	Revisionen GML 3.2.1. wird ISO 19136

Tabelle 3.4: Entwicklungsstufen der GML

Bei der Abbildung von geometrischen Objekten verwendet GML die so genannten „Simple Features“-Definitionen des OGC. Durch die Erzeugung „einfacher Objekte“ wurde eine Grundlage für die gemeinsame Objektbildung geschaffen. Das abstrakte Objektmodell der „Simple Features“ wird in den „Implementation Specifications“ des OGC benutzt. Nach diesen Definitionen erzeugt das Objekt ein abstraktes Element der realen Welt, wobei ein Objekt sich über Eigenschaften definiert, die durch die Benennung von Name, Typ und Wert beschrieben werden. Jedes Objekt kann sowohl graphische als auch nichtgraphische Eigenschaften besitzen. Man spricht von einem geometrischen Objekt, wenn das Objekt eine graphische Eigenschaft besitzt. Daraus wird deutlich, dass die Geometrie eines Objekts nichts weiter ist, als eine Eigenschaft eines Objekts. Die Geometrie-eigenschaft wird ihrerseits gesondert unter der Klasse „Geometry“ modelliert. Durch das Zusammenfassen mehrerer Objekte entstehen so genannte „Feature Collections“, welche ihrerseits auch wieder Eigenschaften aufweisen können. Die folgende Tabelle und Abbildung zeigen die Zusammenhänge hinsichtlich der Begrifflichkeiten auf.

Begriff	Beschreibung	Beispiel
<b>Feature</b> (Objekt)	Repräsentation eines Objekts der realen Welt	Repräsentationen für Straßen, Gebäude oder Adressen

<b>Complex Feature</b> (Komplexes Objekt)	Aus anderen Objekten zusammengesetztes Objekt	Repräsentationen von Leitungsnetzen von Stromversorgern; aus einzelnen Leitungsabschnitten zusammengesetzt
<b>Feature Type</b> (Objekt-Typ)	Gleichartige Objekte werden zu Typen zusammengefasst	Zusammenfassung aller Fluss-Repräsentationen (Rhein, Main, ...) ergibt den Objekt-Typ „Fluss“
<b>Properties</b> (Eigenschaften)	Informationen, die zu einem Objekt angegeben werden (im GIS als „Attribute“ bezeichnet); jede Eigenschaft hat einen Namen, einen Datentyp und einen Wert; Folge aller Eigenschaften ist der Inhalt eines Elements	Ein Gebäude hat die beiden Eigenschaften „hausnummer“ und „geometrie“; Der Datentyp von „hausnummer“ könnte „zahl“ sein, ein möglicher Wert „27“
<b>Feature Collection</b> (Objekt-Kollektion)	Gruppierung von beliebigen Objekten; die Aufgabe einer Objekt-Kollektion ist, einen Container für mehrere Objekte zu bilden;  Die Assoziation zwischen einer Feature Collection und den Features, die sie beinhaltet, drückt die Eigenschaft feature-Member aus.	„Stadt“ ist eine Feature Collection, welche die featureMember „Häuser“, „Straßen“ usw. enthalten kann

Tabelle 3.5: Erläuterung der Grundbegriffe von GML nach (Pritschet, 2005)



Abbildung 3.12: Die GML-Komponenten im Zusammenhang

Bei der Erläuterung von XML wurde auf den Zweck der DTD-Dateien eingegangen. Diese Beschreibungsart von Datenstrukturen wurde lediglich in der GML-Version 1.0 genutzt. Alle folgenden Versionen verwenden die wesentlich flexibleren XML-Schemadefinitionen. Da sich über GML nicht nur Geometrieobjekte sondern auch Objekte ohne Geometrie, Objektverbände oder auch Objekte, die nur eine geometrische Ausprägung haben, abbilden lassen, sind ursprünglich drei Basisschemas definiert worden.

- geometry.xsd  
dient zur Definition abstrakter, repräsentierbarer und konkreter Geometrielemente (Punkt, Linie, Linienring) und auch komplexe Typdefinitionen.

- feature.xsd  
erweitert das Geometrieschema um die Möglichkeit zur Zusammenfassung von Geometrieelementen („Feature Collections“). Des Weiteren kann dieses Schema allgemeine Objekteigenschaften, wie beispielsweise einen Identifikator oder Namen einbringen.
- xlink.xsd  
erlaubt das Verweisen auf interne und externe Quellen. Über den Import-Tag der geometry.xsd wird der gesamte Namensraum von xlink.xsd zur Verfügung gestellt. Mit dieser Datei ist der bestehende XML-Standard übernommen worden.

Mit Erscheinen der Version GML 3.0 sind wesentliche Erweiterungen eingebracht worden, so dass die drei vorgestellten Schemadefinitionen nicht mehr ausreichend waren. Obgleich die Schemadatei feature.xsd bestehen blieb, ist das Geometrieschema in fünf Dateien aufgeteilt worden.

- geometryBasic0d1d.xsd
- geometryBasic2d.xsd
- geometryPrimitives.xsd
- geometryAggregates.xsd
- geometryComplexes.xsd

Diese Dateien kommen je nach Anwendungsfall zum Einsatz. Die Schemadatei geometryBasic0d1d.xsd beispielsweise wird lediglich verwendet, wenn ein- oder zweidimensionale Geometrien (Punkte und Linien) vorliegen. Aufgrund der umfangreichen Neuerungen sind mit der Version GML 3.0 zusätzliche Schemadateien definiert worden. So werden topologische Informationen in der Schemadatei „topology.xsd“ definiert, zeitliche Aspekte in der Schemadatei temporal.xsd.

Die folgend Abbildung zeigt einen Ausschnitt aus einer Flurkarte, auf der ein bestimmtes Flurstück markiert ist. Anschließend wird dieses Flurstück mithilfe der GML-Syntax nach GML 3.2 dargestellt.

Das Element <featureMember> hat verschiedene Eigenschaften wie <est:ROWID> oder <est:LAND>. Eine weitere Eigenschaft stellt der Tag <est:FLÄCHE> dar, wodurch die Geometrie des Flurstücks ausgedrückt wird. Hinter dem Kürzel EPSG4326 versteckt sich die Bezeichnung für das Koordinatensystem nach den Definitionen der European Petroleum Survey Group Geodesy (EPSG). In diesem Fall handelt es sich um ein geographisches Koordinatensystem mit dem Datum WGS84. Die Geometrie selber wird als Umring mit einer Folge von Koordinaten festgelegt. Diese Koordinaten werden in dem Tag <gml:posList> eingeschlossen.



Abbildung 3.13: Graphische Darstellung eines Flurstücks

```

<?xml version="1.0" encoding="UTF-8" ?>
<est:FeatureCollection xmlns="http://steinmann-siegen.de/GML/3.2"
xmlns:gml="http://www.opengis.net/gml" xmlns:est="http://steinmann-
siegen.de/GML/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://steinmann-siegen.de/GML/3.2
d:\warehouses\GMLData.xsd">
  <gml:boundedBy>
    <gml:Null>unknown</gml:Null>
  </gml:boundedBy>
  <gml:featureMember>
    <est:Flurstueck gml:id="Flurstueck.1">
      <est:ROW_ID>1</est:ROW_ID>
      <est:OBJNUM>S005PFP</est:OBJNUM>
      <est:GEMARKUNG>1644</est:GEMARKUNG>
      <est:FLURNR>12</est:FLURNR>
      <est:LAND>05</est:LAND>
      <est:FSTNR>143</est:FSTNR>
      <est:NENNER />
      <est:Fläche>
        <gml:Polygon srsName="EPSG:4326">
          <gml:exterior>
            <gml:LinearRing>
              <gml:posList srsDimension="2">50.8832631380446 8.14550508114865
                50.8832652600398 8.14569673762893 50.88306944375
                8.14565584121007 50.8830696429186 8.1455233992652
                50.8830769660372 8.14534492829339 50.8832659209031
                8.14533266078138 50.8832631380446 8.14550508114865</gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </est:Fläche>
    </est:Flurstueck>
  </gml:featureMember>
</est:FeatureCollection>

```

Abbildung 3.14: Repräsentation eines Flurstücks in GML

## 3.2 Raumbezogene Datenhaltung

Die datenbankbasierte Speicherung von Informationen ist seit Jahrzehnten einem Entwicklungsprozess unterworfen, der sich stets an neue Erkenntnisse und Anforderungen der Zeit angepasst hat. Zum einen haben sich die Modelle, nach denen Informationen beschrieben und gespeichert werden, weiter entwickelt und zum anderen haben sich die Informationen selbst hinsichtlich Art und Menge verändert. Bei den Betrachtungen in diesem Kapitel kommt die Besonderheit hinzu, dass es sich bei den zu verarbeitenden Informationen um solche handelt, die einen Raumbezug haben. Diese Betrachtungen setzen voraus, dass Realweltobjekte in einem Datenmodell als Objekte mit Eigenschaften und Methoden definiert werden. Nach den Grundlagen der objektorientierten Datenmodellierung und den Vorzügen der relationalen Ansätze hat sich in der geographischen Informationsverarbeitung so genannte objektrationale Modellierung von Realweltobjekten als besonders vorteilhaft erwiesen. Es gilt also, die geometrischen Ausprägungen von Objekte der realen Welt zu modellieren und in Datenbanken zu speichern. An dieser Stelle empfiehlt es sich, drei Fachbegriffe zu definieren. Der Begriff Datenbank bezeichnet zunächst lediglich einen wie auch immer strukturierten Datenbestand. Mehrere Datenbanken können in einem System verwaltet werden, man spricht von einem Datenbanksystem. Die Software, die eine oder mehrere Datenbanken verwaltet, nennt man Datenbankmanagementsystem (DBMS).

Auf dem Markt gibt es neben den kommerziellen Produkten, wie beispielsweise SQL Server, DB 2 und Oracle, die eine räumliche Erweiterung anbieten, auch freie Software, die z.B. PostGIS verkörpert. Im Folgenden wird mit der Firma Oracle und dem gleichnamigen Produkt eine Lösung vorgestellt, die einerseits marktführend und gleichzeitig auch als OGC compliant zertifiziert ist. Mit Oracle lässt sich eine Datengrundlage schaffen, auf deren Basis sich Ansätze zur Integration raumbezogener Informationen in die Unternehmens-IT diskutieren lassen.

### 3.2.1 Oracle

Neben der Firma Microsoft ist die Oracle Corporation der zweitgrößte Hersteller von Softwareprodukten. Diese Produkte beziehen sich auf das Management von Informationen. Das Unternehmen erwirtschaftete im Jahr 2006 mit über 56.000 Mitarbeitern in 145 Ländern weltweit einen Umsatz von über 14 Milliarden US-Dollar. Im Jahr 1977 gründete Lawrence J. Ellison die Firma Software Development Laboratories und entwickelte für die CIA ein System namens Oracle. Im Jahr 1983 wurde das Unternehmen selbst in Oracle umbenannt. Das Unternehmen hat seinen Hauptsitz in Redwood Shores, Kalifornien, und bietet neben Datenbanken, Unternehmensanwendungen und Entwicklungswerkzeugen auch Beratung und Ausbildung an (Oracle, 2007a).

Zur Verwaltung von raumbezogenen und nichtraumbezogenen Informationen stellt die Firma Oracle ein gleichnamiges DBMS her, welches in verschiedenen Editionen erhältlich ist. In Abhängigkeit vom Umfang der Funktionalitäten, kann der Anwender in der seit 2004 aktuellen Version 10g zwischen der Enterprise Edition, der Standard Edition, der Personal Edition und der Lite Edition wählen. Das „g“ in der Bezeichnung der Produktversion steht für „Grid“. Unter Grid-Computing versteht man die Aufteilung von Rechenleistungen auf verschiedene Computer, was durch das Produkt Oracle unterstützt wird. Durch die Zusammenfassung von Servern zu Gruppen werden so genannte Cluster gebildet, wodurch vorhandene Ressourcen zwischen Anwendungen dynamisch verteilt werden können (Oracle, 2007a).

### 3.2.2 Oracle Spatial

Wenn in einer Datenbank räumliche Daten gespeichert und analysiert werden sollen, sind in der Regel die Standardfunktionalitäten, die ein DBMS bietet, nicht ausreichend. Dies gilt insbesondere dann, wenn die Modellierung der raumbezogenen Daten objektorientiert, bzw. objektrational aufgebaut ist. Daher ist es nicht nur bei Oracle üblich, das DBMS um räumliche Komponenten zu erweitern. Eine solche räumliche Erweiterung hat Oracle bereits in der 1996 erschienenen Version 7.3.3 implementiert und nannte diese Erweiterung „Spatial Data Options“. Dem wachsenden Bedarf an diesen Funktionalitäten wurde Oracle dadurch gerecht, dass die räumlichen Komponenten in Zukunft in eine kostenpflichtige Erweiterung mündeten. Zu dem Zeitpunkt, als die Version 8.0.4 des DBMS anstand, führte Oracle die so genannten Cartridges ein. Dabei handelte es sich um Pakete, in denen verschiedenste Funktionalitäten für unterschiedlichste Anwendergruppen zusammengefasst wurden. Für

die räumlichen Funktionalitäten wurde die Spatial Cartridge definiert. Die Anwendung dieser Erweiterung setzte allerdings grundsätzlich die 8i Enterprise Edition voraus. Von dem Konzept der kostenpflichtigen Erweiterungen trennte man sich allerdings im Jahr 2001 wieder, als die Version 9i auf den Markt kam. Hierbei implementierte man eine Vielzahl an räumlichen Funktionalitäten in die Standard und Enterprise Edition, so dass sie nicht mehr gesondert erworben und bezahlt werden mussten. Die implementierten Funktionalitäten und Dienste wurden unter dem Begriff „Spatial Locator“ zusammengefasst. Anwender, für die der Spatial Locator nicht ausreichend ist, müssen das kostenpflichtige Modul „Spatial“ erwerben, welches die erweiterten Funktionalitäten bietet. Dieses Konzept hat sich bis zur aktuellen Version 10g durchgesetzt.

Der Spatial Locator enthält Datentypen, Operationen, räumliche Indizierungsfunktionen und eingeschränkte Funktionen und Prozeduren der Spatial Option. Folgende Komponenten sind im Spatial Locator enthalten (Oracle, 2003).

- Das Datenbankschema MDSYS, welches die Speicherung, die Syntax und die Semantik der unterstützten geometrischen Datentypen vorschreibt
- Ein Mechanismus zur räumlichen Indizierung
- Eine Vielzahl an Operatoren und Funktionen, um räumliche Abfragen zu erstellen sowie weitere räumliche Analyseoperatoren
- Administrative Werkzeuge

Im Folgenden wird auf die wichtigsten Kernpunkte von Oracle Spatial eingegangen.

### 3.2.2.1 Geometriertypen

Eine Geometrie ist eine geordnete Folge von Stützpunkten die mit geraden Linien oder Kreisbögen verbunden sind. Die Semantik einer Geometrie wird durch ihren Geometriertyp bestimmt. Oracle Spatial unterstützt verschiedene geometrische Primitive sowie Kollektionen, die sich aus diesen Primitiven zusammensetzen. Die folgende Abbildung zeigt die verschiedenen Geometriertypen.

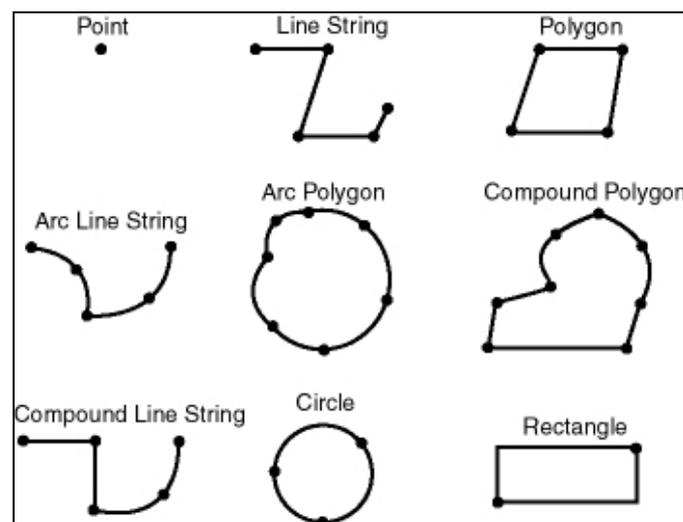


Abbildung 3.15: Geometriertypen in Oracle Spatial (Oracle, 2003)

### 3.2.2.2 Logisches Datenmodell

Das logische Datenmodell von Oracle Spatial besteht aus hierarchisch angeordneten Elementen, Geometrien und Layern. Die Elemente bestehen aus den geometrischen Primitiven wie Punkte, Linien, Polygone oder zusammengesetzte Formen. Durch diese Elemente werden die Geometrien gebildet, welche durch zusätzliche Attribute ergänzt werden können. Geometrien werden einem räumlichen Layer zugeordnet, welcher einen abgegrenzten Raum und ein räumliches Referenzsystem hat. Die folgende Abbildung zeigt das logische Datenmodell.

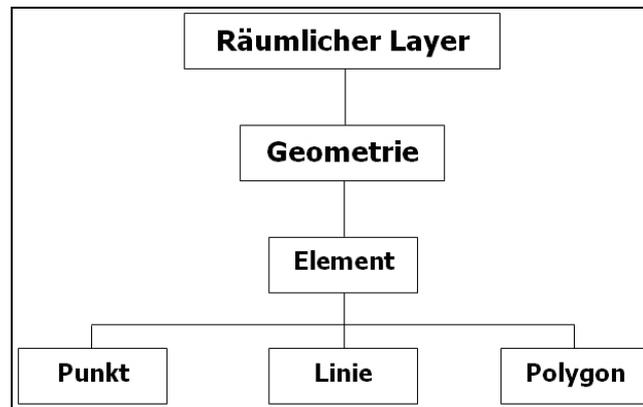


Abbildung 3.16: Logische Datenmodell von Oracle Spatial

### 3.2.2.3 Physisches Datenmodell

Seit der Version 9.2 von Oracle wird zur Speicherung von räumlichen Daten nur noch das objektrelationale Datenmodell unterstützt. Die Unterstützung des relativen Datenmodells wurde damit endgültig aufgegeben. Das objektrelationale Datenmodell zeichnet sich dadurch aus, dass die Geometrie in einer einzigen Tabellenspalte abgespeichert wird, wobei pro Geometrieinstanz ein Datensatz angelegt wird. Der Datentyp dieser Geometriespalte ist im Schema MDSYS definiert und nennt sich SDO\_GEOMETRY. Dieser Datentyp wiederum wird anhand mehrerer Attribute definiert, die im Folgenden aufgezeigt sind.

```

CREATE TYPE sdo_geometry AS OBJECT (
    SDO_GTYPE          NUMBER,
    SDO_SRID           NUMBER,
    SDO_POINT          SDO_POINT_TYPE,
    SDO_ELEM_INFO     SDO_ELEM_INFO_ARRAY,
    SDO_ORDINATES     SDO_ORDINATE_ARRAY);
  
```

- **SDO\_GTYPE:**  
beschreibt den Geometriotyp.
- **SDO\_SRID:**  
beschreibt das Koordinatensystem durch einen Schlüssel.
- **SDO\_POINT:**  
beschreibt ein Objekt vom Objekttyp MDSYS.SDO\_POINT\_TYPE mit den Attributen X, Y und Z vom Typ Number.
- **SDO\_ELEM\_INFO:**  
beschreibt das Listenfeld SDO\_ELEM\_INFO\_ARRAY mit dem Objekttyp SDO\_ELEM\_INFO\_TYPE.
- **SDO\_ORDINATES:**  
beschreibt die Koordinaten des Objekts mit dem Listenfeld des Typs SDO\_ORDINATE\_ARRAY.

### 3.2.2.4 Räumliche Indizierung

Eine Schlüsselfunktionalität von Oracle Spatial ist die Indizierung räumlicher Daten. Ein räumlicher Index bietet ebenso wie jeder andere Index auch einen Mechanismus, um die Suche innerhalb des Datenbestandes zu optimieren. In diesem Fall jedoch basiert der Mechanismus auf räumlichen Kriterien wie beispielsweise Verschneidungen. Ein räumlicher Index wird beispielsweise benötigt, wenn Objekte innerhalb eines indizierten Datenbestandes gefunden werden sollen, die in einer räumlichen Beziehung zu einem bestimmten Punkt oder einem Gebiet liegen. Oracle Spatial unterstützt sowohl den R-tree- als auch den Quadtree-Index, wobei ausdrücklich darauf verwiesen wird, dass der R-tree zu bevorzugen ist. Diese Methode ist signifikant leistungsfähiger in der Version 10g von Oracle (Oracle, 2003).

Der R-tree Index umschließt jede Geometrie mit dem kleinsten umgebenen Rechteck. Dieses Rechteck wird mit Minimum Bounding Rectangle (MBR) bezeichnet. Die folgende Abbildung verdeutlicht dieses Rechteck.



Abbildung 3.17: Minimum Bounding Rectangle (MBR) (Oracle, 2003)

Für eine ganze Ebene an Geometrien legt der R-tree Index einen hierarchischen Index auf der Basis der MBRs der Geometrien an. Dieses Prinzip wird in der folgenden Abbildung verdeutlicht.

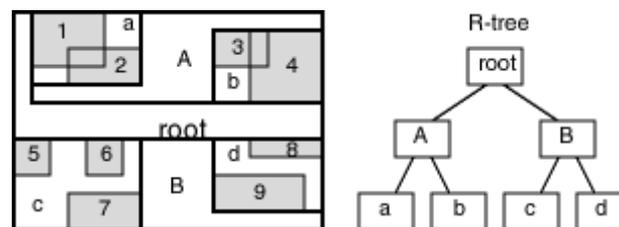


Abbildung 3.18: Hierarchischer Index auf Basis von MBRs (Oracle, 2003)

Die grauen Vierecke 1 bis 9 sind die Geometrien eines Layers. Die Felder a bis d sind die Blattknoten des R-trees. Sie beinhalten minimale Umringe von Geometrien und besitzen so genannte Pointer zu den eingeschlossenen Geometrien. Diese Pointer sind als Referenz zu verstehen, damit eine Zuordnung zwischen den Umringen und den Geometrien stattfinden kann. Diese Umringe werden anhand eines weiteren MBRs zu größeren Einheiten zusammengefasst. In Abbildung 3.18 beinhaltet die Fläche A die Teilflächen a und b. Auch hierbei wird die Zuordnung über Pointer geregelt. Das Root-Element beinhaltet den kleinsten Umring um alle im Layer vorkommenden Geometrien. Durch diese Vorgehensweise ist es möglich, Geometrien anhand ihres Index ausfindig zu machen und somit wesentlich effizienter zu Verarbeiten.

Für die Speicherung des R-tree Index sieht das Datenbankschema MDSYS die Tabelle SDO\_INDEX\_TABLE vor. Darüber hinaus unterstützt der Index eine Sequenz mit der Bezeichnung SDO\_RTREE\_SEQ\_NAME um sicherzustellen, dass gleichzeitige Updates des Index durch Netzwerknutzer gemacht werden können.

### 3.2.3 Abfragemodell und -umgebung

Oracle Spatial nutzt ein Abfragemodell, welches auf zwei Ebenen basiert, um räumliche Daten abzufragen und zu kombinieren. Innerhalb der zwei Ebenen werden zwei eigenständige Operationen durchgeführt, die miteinander kombiniert werden und schließlich zum exakten Abfrageergebnis führen. Diese zwei Operationen werden als primärer und sekundärer Filter bezeichnet. Der Primärfilter umfasst eine erste Auswahl von Kandidaten und leitet diese an den Sekundärfilter weiter. Er vergleicht Geometrieapproximationen, um den Rechenaufwand zu

verringern. Da dieser Filter nur eine Näherung auf Grundlage der Geometrien erstellt, ist die Ausgabemenge ein wesentlich größerer Datensatz als das tatsächliche Ergebnis. Der Sekundärfilter berechnet aus dem Ausgangsdatsatz des Primärfilters den Ergebnissatz. Der Sekundärfilter ist zwar wesentlich umfangreicher als der Primärfilter, aber der Rechenumfang ist auch nur auf eine kleine Teilmenge des Gesamtdatensatzes anzuwenden, da dieser durch den Primärfilter reduziert wurde. Die folgende Abbildung zeigt das Prinzip des Abfragemodells.

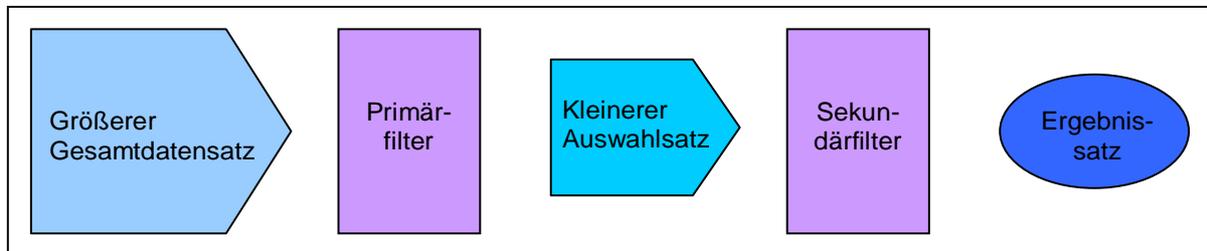


Abbildung 3.19: Räumliche Abfragemodell nach (Oracle, 2003)

Für die Anwendung des Primärfilters ist es notwendig, dass die Daten mit einem räumlichen Index versehen wurden. Oracle Spatial verwendet nicht grundsätzlich beide Filter. Vielmehr ist in vielen Fällen auch nur der Primärfilter ausreichend. Wenn beispielsweise die Zoom-Funktion in einer GIS-Anwendung verwendet wird, werden die Daten lediglich anhand eines Rechtecks dargestellt. Die Ermittlung des Rechtecks geschieht nur mit dem Primärfilter. Die GIS-Anwendung kann daraufhin den Ausschnitt des sichtbaren Bereichs ermitteln.

Beim Aufbau einer Datenbasis als Grundlage für eine GIS-Anwendung sind neben den theoretischen Überlegungen zum Datenmodell verschiedene Schritte in Bezug auf die Datenhaltung notwendig. Zunächst müssen Datenbanktabellen erstellt werden, in denen die Daten später abgelegt werden können. Diese Tabellen haben unter Umständen bestimmte Beziehungen untereinander, die es festzulegen gilt. Wenn das Datenmodell in konkrete Tabellen und Bedingungen umgesetzt ist, müssen diese Tabellen mit Informationen gefüllt werden. Im Anschluss daran können die Daten durch Abfrage- und Analysemechanismen verwendet werden. Für diese Aufgaben, die allesamt in Oracle gelöst werden müssen, stehen Konstrukte zur Verfügung, die auf der Structured Query Language (SQL) basieren. SQL ist eine auf dem ANSI-Standard basierende Sprache, mit der Tabellen einer Datenbank erstellt und verändert werden können, Daten einer Tabelle eingefügt, manipuliert und abgefragt werden können und Tabellen miteinander verknüpft werden können. Um mit einer Datenbank via SQL zu kommunizieren, stehen in Oracle verschiedene Werkzeuge zur Verfügung. Ein solches Werkzeug stellt SQL\*Plus dar, mit dem man SQL-Befehle erstellen und an die Datenbank zur Ausführung weiterleiten kann. Die Datenbank verarbeitet den Befehl und leitet das Ergebnis wieder an die Benutzeroberfläche von SQL\*Plus zurück.

Zur Abfrage von Daten aus einer Datenbank wird ein SQL-Befehl verwendet, der mindestens aus den Komponenten SELECT und FROM besteht. Zudem kann eine weitere Bedingung mit WHERE eingesetzt werden. Das folgende Beispiel verdeutlicht eine SQL-Abfrage. Dargestellt ist der Ausschnitt einer Tabelle, in der Informationen zu Flurstücken abgelegt sind. Aus dieser Tabelle soll eine Untermenge ermittelt werden, die bestimmten Kriterien entspricht.

Kreis	Gemeinde	Gemarkung	Flur	Zähler	Nenner
Musterkreis	Musterstadt	Musterheim	12	78	2
Musterkreis	Musterstadt	Musterheim	12	79	3
Musterkreis	Musterstadt	Musterheim	13	43	2

Abfrage:       SELECT a.Flur, a.Zaehler, a.Nenner  
                   FROM Flurstuecke a  
                   WHERE a.Flur = 12

Ergebnis:

Flur	Zaehler	Nenner
12	78	2
12	79	3

Der Ergebnissatz enthält nur die Flurstücke, die in der Flur 12 vorkommen.

Die beispielhafte Abfrage bezieht sich nur auf Attributspalten einer einzigen Tabelle, in der zunächst keine räumlichen Daten erkennbar sind. Wie bereits beschrieben, speichert Oracle Spatial Geometrien in einer Tabellenspalte vom Datentyp SDO\_GEOMETRY. Die Geometrien, die mithilfe dieses Datentyps gespeichert sind, lassen sich ebenso mit SQL abfragen, wie die im Beispiel dargestellten attributiven Informationen. Dazu erweitert Oracle SQL um spezifische räumliche Abfragemechanismen und Befehle. Im Folgenden wird ein Beispiel vorgestellt, in dem alle Städte des Bundesstaats Colorado (USA) ermittelt werden sollen. Dazu stehen in einer Oracle Datenbank die beiden Tabellen „Cities“ und „States“ zur Verfügung. Beide Tabellen enthalten neben attributiven Informationen die Geometrie einer Stadt (Punktgeometrie) bzw. eines Staates (Flächengeometrie). Die folgende Abbildung zeigt zunächst eine Abfrage in Form einer räumlichen Verschneidung mithilfe des GIS-Systems GeoMedia Professional der Firma Intergraph, mit dem auf den Oracle Datenbestand zugegriffen wird. Die Ergebnismenge ist in Form der blauen Punkte innerhalb der Staatsgrenzen von Colorado dargestellt.

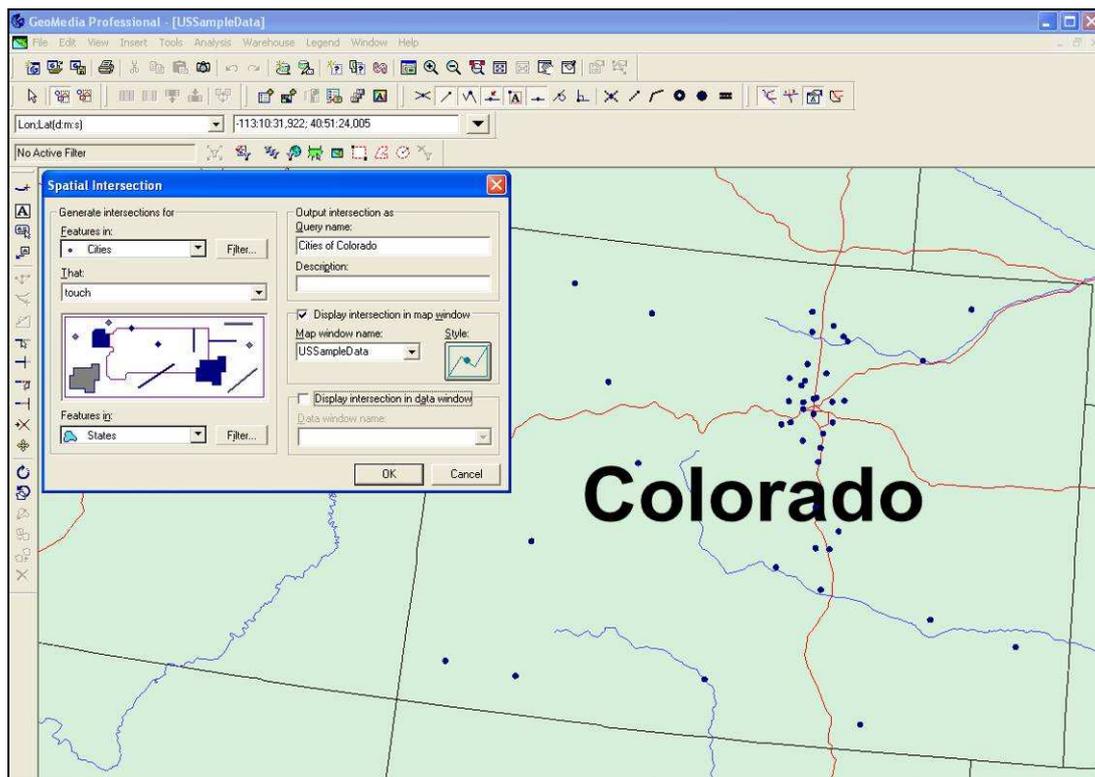


Abbildung 3.20: Räumliche Abfrage in GeoMedia Professional

Auf selbigen Datenbestand lässt sich ebenfalls mit den Werkzeugen von Oracle zugreifen und diese Abfrage „manuell“ durchführen. Dazu ist die folgende Abfrage an die Datenbank weiterzuleiten.

```
SELECT A.STATE_NAME, B.CITY_NAME
FROM USA.STATES A, USA.CITIES B
WHERE SDO_RELATE(B.GEOMETRY,A.GEOMETRY,'MASK=INSIDE QUERYTYPE=
JOIN')='TRUE' AND A.STATE_NAME = 'Colorado';
```

Der Unterschied zur vorherigen Abfrage ist der, dass hier zwei Tabellen in die Abfrage aufgenommen worden sind (States und Cities) und diese Tabellen durch einen Join miteinander verknüpft werden. Die eigentliche Bedingung, dass nur die Städte ausgewählt werden, die innerhalb des Staats Colorado liegen, wird mit SDO\_RELATE und dem räumlichen Operator MASK=INSIDE ausgedrückt. Diese Abfrage wird mit dem Werkzeug iSQL\*PLUS erstellt und an die Datenbank weitergeleitet. Dieses Werkzeug unterscheidet sich von SQL\*PLUS nur insofern, als dass es mit einem Web-Browser zu öffnen ist.

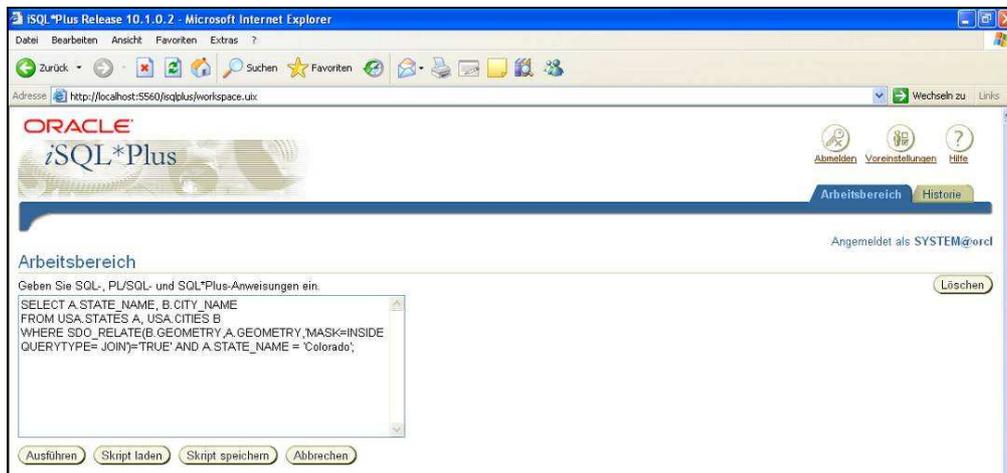


Abbildung 3.21: Räumliche Abfrage in iSQL\*PLUS

Die Datenbank antwortet mit dem Ergebnissatz, der in Tabellenform in der Oberfläche von iSQL\*PLUS dargestellt wird.

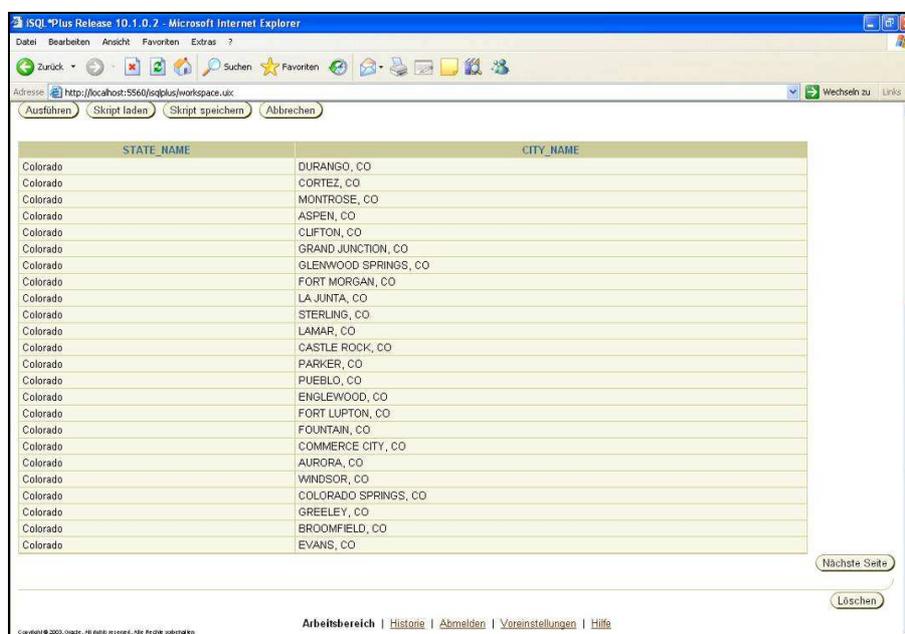


Abbildung 3.22: Ergebnissatz einer räumlichen Abfrage in iSQL\*PLUS

SQL ist keine Programmiersprache, sondern eine Abfragesprache. Das bedeutet, dass man mit SQL keine typischen Programmieraufgaben lösen kann wie beispielsweise die Bildung von Schleifen. Zu diesem Zweck bietet Oracle eine Erweiterung von SQL um prozedurale Mechanismen. Diese Technologie wird unter der Bezeichnung P/L-SQL zusammengefasst. P/L steht dabei für Procedural Language oder auch Processing Language.

Damit ist es möglich, umfangreiche Programme zu erstellen und damit einen gesamten Block an Anweisungen an die Datenbank zu leiten und verarbeiten zu lassen.

### 3.3 Systemarchitekturen

Dieses Kapitel beschreibt zwei Systemarchitekturen, die derzeit im Zusammenhang mit der Integration verschiedenartiger Softwarebestandteile und Unternehmensabläufe diskutiert werden. Dabei handelt es sich um die Model Driven Architecture (MDA) und die Service-oriented Architecture (SOA). Diese beiden Architekturen sind vom Ansatz her nicht miteinander zu vergleichen oder gegenüber zu stellen, sondern vielmehr getrennt zu betrachten. Im Anschluss einer Betrachtung beider Architekturen sind Rückschlüsse zu ziehen, inwiefern sich MDA und SOA eventuell ergänzen können.

#### 3.3.1 Model Driven Architecture (MDA)

Bei der Model Driven Architecture handelt es sich um einen Standard der Object Management Group (OMG). Diese 1989 gegründete Organisation, der heute über 800 Mitglieder angehören, hat sich zum Ziel gesetzt, die Standardisierung von offenen, verteilten, objektorientierten Systemen auf der Basis von Middlewarekonzepten voranzutreiben. Weitere Beispiele dafür sind die Common Object Request Broker Architecture (CORBA) oder die Unified Modeling Language (UML). Daran wird deutlich, dass die fachliche Ausrichtung der OMG etwas differiert zur der des OGC oder der ISO. Das OMG standardisiert vielmehr, wie ein System aufgebaut und beschrieben wird, während die Arbeiten des OGC eher auf die Entwicklung von Spezifikationen und deren Implementierung abzielen.

Basierend auf den Computer Aided Software Engineering (CASE)-Ideen der 80er und 90er Jahre greift der MDA-Ansatz den Gedanken der modellgetriebenen Softwareentwicklung auf und generiert aus einem beschreibenden Modell den Quellcode für die Softwareanwendung. Die MDA geht von dem Prinzip aus, verschiedenen Stufen des Softwareentwicklungsprozesses eine Modellebene zuzuordnen. Solche Modelle entstehen bei der herkömmlichen Softwareentwicklung oftmals nur als Randprodukt zu Dokumentationszwecken. Anwendungen werden auf der Grundlage von Quellcode erstellt, wobei eine Änderung in der Anwendung für gewöhnlich durch Anpassung des Quellcodes durchgeführt wird. Solche Änderungen werden dann nicht zwingend auch in einem Modell nachgeführt. Die MDA hingegen betrachtet eine Softwarekomponente losgelöst von dessen Implementierung, also unabhängig vom Quellcode. Die folgende Abbildung zeigt verschiedene Modellebenen der MDA und damit ihr Prinzip.

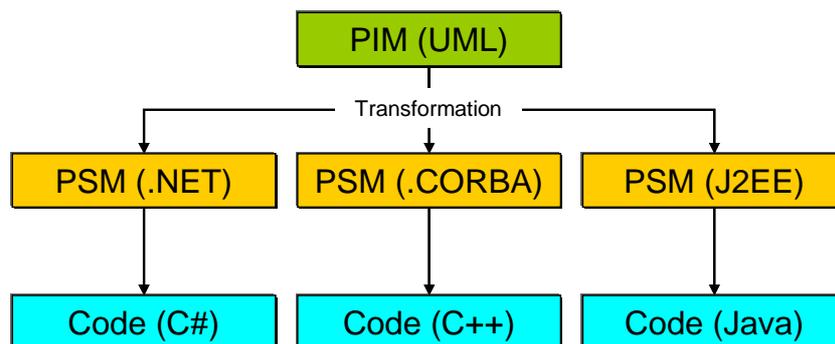


Abbildung 3.23: Modellebenen der MDA nach (OMG, 2003)

Am Anfang einer Softwareentwicklung steht eine Idee, eine Vision darüber, wie ein fertiges (Teil)produkt später aussehen und funktionieren könnte. Diese Idee ist noch vollkommen losgelöst von möglichen Implementierungen auf festen Systemplattformen. Diese Idee kann in den Köpfen von Produktentwicklern oder in Form von Manuskripten existieren. Daher bezeichnet man diese computerunabhängige Modellebene auch Computing Independent Modell (CIM). Diese Idee wird konkretisiert, indem mithilfe einer standardisierten Beschreibungssprache ein Modell entwickelt wird, welches sich durch eine Allgemeingültigkeit auszeichnet und unabhängig

von einer Systemplattform ist. Dieses Modell wird daher als Platform Independent Modell (PIM) bezeichnet und wird typischerweise mit der Unified Modeling Language (UML) beschrieben. Die Modellierungssprache UML besitzt eine Notation mit der es möglich ist, Softwarebestandteile und deren Beziehungen untereinander zu beschreiben. Der UML-Standard sieht so genannte UML-Profile vor, die es dem Designer erlauben, die Modellierungssprache um neue semantische Konstrukte zu erweitern, um somit eigene Bedürfnisse zu berücksichtigen. (OMG, 2003)

Die nächste Modellebene verlässt den fachlichen Aspekt einer Anwendung und geht über zu einer technischen Sichtweise. Sie ist abhängig von der Plattform, auf der die Anwendung später implementiert werden soll. Mögliche Plattformen wären hier eine Microsoft Windows Umgebung oder auch eine Java-Plattform. Diese Modellebene wird daher als Platform Specific Modell (PSM) bezeichnet. Abhängig von dieser Modellebene wird der zugehörige Quellcode erzeugt, der je nach Plattform beispielsweise in C-Sharp (C#) oder auch als Java-Code darzustellen ist. Durch die Trennung der Fachlogik von der Implementierungstechnologie wird eine gewisse Wiederverwendbarkeit der fachlichen Modellierung sichergestellt, was beispielsweise bei einer Migration auf eine andere Plattform von Bedeutung ist.

Die Überführung eines Modells auf die nächste Modellebene wird Transformation genannt. Anhand von zu definierenden Regeln wird eine Transformation voll oder teilweise automatisiert durchgeführt. Inwiefern ein Automatismus stattfinden kann, ist davon abhängig, ob sich eine Transformation ganz, teilweise oder überhaupt nicht formal beschreiben lässt.

Die Vorteile, die sich aus dem Prinzip der MDA ergeben, liegen auf der Hand. Änderungen in der Anwendung müssen nicht mehr an verschiedenen Stellen im Quellcode vorgenommen werden, sondern sind lediglich auf Modellebene zu definieren. Der neue Quellcode ergibt sich wieder durch erneute Transformation der Modelle. Dadurch wird der Automatisierungsgrad der Entwicklung wesentlich erhöht und somit Fehlerquellen minimiert. Weiterhin kann einem technologischen Wechsel schneller begegnet werden, indem aus dem PIM durch neue Transformationsregeln ein der neuen Technologie entsprechendes PSM abgeleitet wird. Durch die Verwendung standardisierter Beschreibungssprachen lässt sich ein Modell leichter mit den Werkzeugen unterschiedlicher Hersteller integrieren. (OMG, 2003)

Diesen Vorteilen stehen aber durchaus auch Nachteile der MDA gegenüber. Es stellt sich bei automatisch generiertem Quellcode grundsätzlich die Frage nach der Qualität bezüglich der Lesbarkeit, der Strukturierung und Wartbarkeit. Ein Quellcode, der nicht vollständig automatisiert generiert werden kann, muss manuell nachbearbeitet werden. Für Entwickler ist es generell schwierig, sich in Quellcode einzuarbeiten, der nicht selbst geschrieben wurde. Zudem müssen gerade die manuell durchgeführten Änderungen exakt dokumentiert werden, da diese im Modell und in den Transformationsregeln ansonsten nicht enthalten sind. Eine erneute Ableitung des Quellcodes aus dem (geänderten) Modell führt in der Regel auch wieder zu einer manuellen Anpassung des Quellcodes.

### 3.3.2 Service-oriented Architecture (SOA)

Betrachtet man die Entwicklung des Internets rückwirkend, so ist nach (Dostal, et al., 2005) zu beobachten, dass der Fokus immer hin zum Computer und den Applikationen und mehr und mehr weg vom Menschen verschoben wird. Zu Beginn gab es die reine Mensch-zu-Mensch-Kommunikation, wie sie heute noch im Email-Verkehr vorzufinden ist. Ab 1990 startete die Verbreitung des Internets. Der Mensch steht zwar immer noch im Mittelpunkt, allerdings nun in einer Mensch-Maschine-Kommunikation. Es handelt sich um eine synchrone, vom Menschen initiierte Unterhaltung. Es wird eine Anfrage an einen Rechner, in der Regel einen Web Server, gestellt und eine Antwort erwartet. Die jüngste Entwicklung zeigt, dass der Mensch nicht mehr im Fokus steht, sondern meist Anwendungen oder Computer. Das gilt besonders für die Kommunikation. Der Mensch hat lediglich nicht mehr die Kontrolle über die Kommunikation. Die Unterhaltung findet zwischen den Applikationen statt, beispielsweise die Kommunikation zwischen Browser und Web Server. Der Browser agiert als verlängerter Arm des Benutzers, jedoch nur ereignisgetrieben.

Bei der Maschine-zu-Maschine-Kommunikation geht es jedoch nicht um eine neue Form von „Remote Procedure Call“ (RPC). Anstelle der Verwendung von Funktionsaufrufen, die meist hart kodiert und schon bei der Pro-

grammierung der Anwendung bedacht werden müssen, geht es bei der neuen Denkweise eher um Möglichkeiten, die benötigten Funktionalitäten oder Dienste dynamisch zur Laufzeit aufzurufen und einzubinden. Dieses Szenario wird als „Service Oriented Architecture“ (Service-orientierte Architektur) bezeichnet (Dostal, et al., 2005). Die Service-orientierte Architektur ist ein Ansatz für die Wiederverwendung bestehender Funktionalitäten oder Unternehmensressourcen durch ihre Kapselung in Services. Unterschiedliche Services können einfach miteinander kombiniert werden und durch die Verwendung von bestehenden Funktionalitäten können die Entwicklungskosten von neuen Anwendungen minimiert werden (Bien, 2004).

Hintergrund für die Einführung einer SOA ist die in vielen Unternehmen oft heterogene Anwendungslandschaft. Die Wartung, Entwicklung und Pflege der Schnittstellen zwischen den verschiedenen Systemen bedeutet einen hohen finanziellen Aufwand. Mit der Einführung einer SOA können die Unternehmen ihre Systemlandschaften flexibler und übersichtlicher gestalten (Ferdinand, 2005).

Bien (2004) betont die Stärke von SOA bei der Wiederverwendung bestehender Funktionalitäten und die Möglichkeit, die funktionalen Bestandteile einer SOA beliebig miteinander zu kombinieren. Darüber hinaus ermöglicht SOA die Entwicklung technologieneutraler Komponenten. Der Vorteil von SOA liegt insbesondere in der Erweiterbarkeit von Systemen, da alle Komponenten miteinander kombinierbar sind; ein Vorteil, der durch die konsequente Trennung von der Beschreibung der Services und deren Funktionalitäten erreicht wird (Kiehle, 2006).

Im Folgenden werden die wichtigsten Aspekte dieser Abstraktion aufgeführt, um eine Definition der Service-orientierten Architektur ableiten zu können. Die wichtigsten Merkmale einer SOA sind nach (Dostal, et al., 2005):

- Lose Kopplung der Dienste,
- Dynamisches Binden,
- Verzeichnisdienst (Repository),
- Verwendung von Standards,
- Einfachheit und
- Sicherheit.

Definition von SOA nach (Dostal, et al., 2005):

***“Unter einer SOA versteht man eine Systemarchitektur, die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht.”***

Die folgende Abbildung zeigt diese Definition umgesetzt in eine Grafik als so genannter SOA-Tempel. Offene Standards, Sicherheit und Zuverlässigkeit bilden das Fundament. Die verteilten Dienste, die lose Kopplung, die Plattformunabhängigkeit und die Prozessorientierung stellen die tragenden Säulen einer SOA dar.

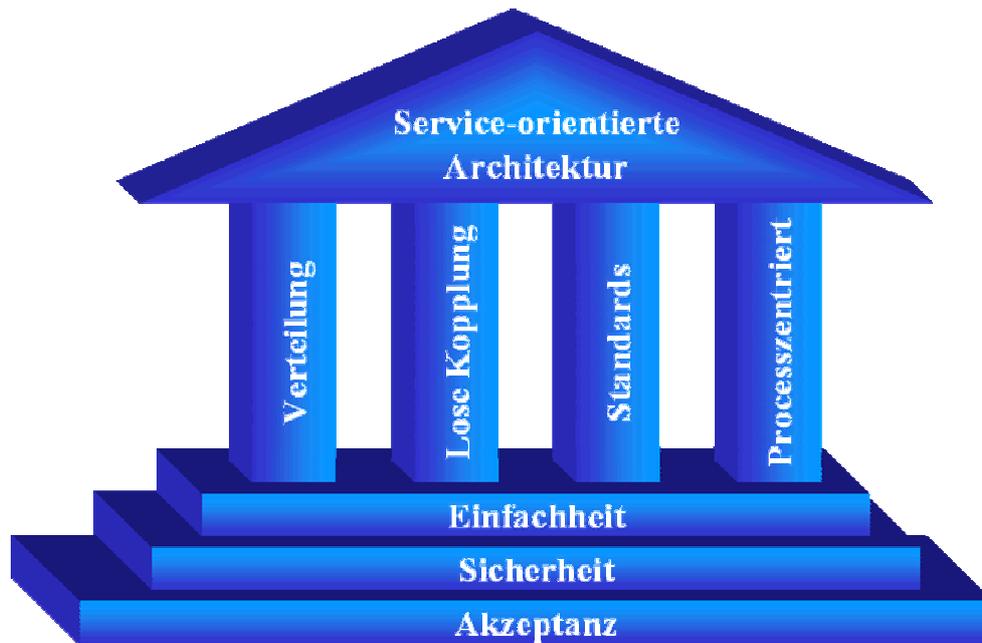


Abbildung 3.24: SOA-Tempel (Dostal, et al., 2005)

Die Grundlage zu diesem Prinzip bilden drei Beteiligte, der Service-Konsument, der Service-Anbieter und das Service-Verzeichnis. Die Interaktionen zwischen den Beteiligten werden häufig mit publish (Publizieren), find (Suchen, Ergebnis) und bind (Frage, Antwort) bezeichnet. (Hauser, Löwer, 2004) erklärt die folgende Abbildung wie folgt. Ein Service-Anbieter stellt einen Service zur Verfügung. Er veröffentlicht ihn in einem Service-Verzeichnis, um den Service bekannt zu machen. Der Service-Nutzer sucht und findet den Service im Verzeichnis (Registry). Mit der Adresse (URL= Uniform Resource Locator) kann der Konsument den Service anfragen und erhält entsprechende Antwort. Jede weitere Kommunikation findet gleichermaßen auf Basis von Frage und Antwort statt.

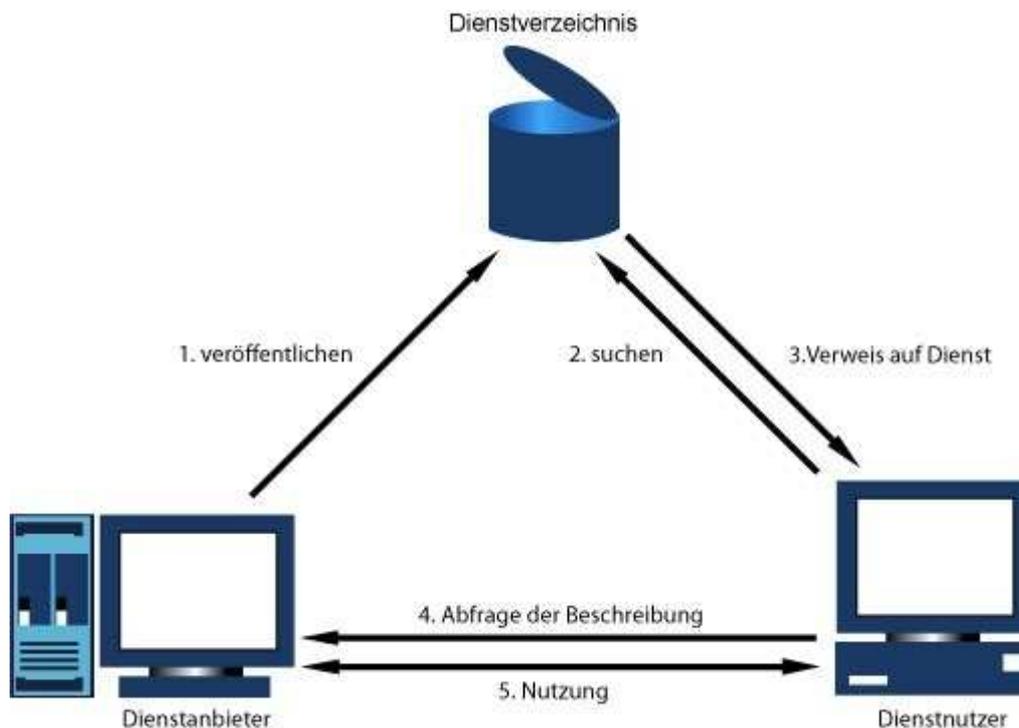


Abbildung 3.25: Das SOA-Dreieck nach (Dostal, et al., 2005)

Nach (Dostal, et al., 2005) wird in zwei Programmiererebenen unterschieden. Zum einen das "Programmieren im Kleinen", welches mit nahezu jeder Programmiersprache möglich ist. Auf Basis von exakten Beschreibungen werden kleine Komponenten erstellt. Eine Vielzahl solcher Komponenten kann unabhängig voneinander und damit auch parallel erzeugt werden. Zum anderen benötigt man eine Sprache, mit der im Großen programmiert werden kann. Mit dieser Sprache wird beschrieben, wie die kleinen Komponenten miteinander agieren. Die eigentliche Leistung ist die Erstellung einer solchen gehobenen Architektur, die eine verständliche Beschreibung des kompletten Systems ermöglicht. Nach dieser Beschreibung ist es dann möglich, einzelne Bestandteile voneinander unabhängig oder sogar an verschiedenen Orten zu erstellen. Später soll es ebenfalls möglich sein, bei Bedarf Komponenten zu ersetzen. Im Zusammenhang mit SOA spricht man von diesen Komponenten als Dienste oder Services, die im Rahmen von Geschäftsabläufen oder auch Prozessen angesprochen, ablaufen oder verarbeitet werden.

SOA ist das derzeit letzte Glied in einer Reihe von Programmierkonzepten. Nach dem Spaghetti-Code aus den Anfängen der Programmierentwicklung wuchs die prozedurale Programmierung. Dadurch war es möglich, einzelne Funktionalitäten zu kapseln und danach an beliebigen Stellen zu nutzen. Das gleichzeitig entwickelte Modulkonzept, das die Verteilung eines Programms auf mehrere Dateien ermöglichte, erlaubte eine erste übersichtliche Strukturierung der Programme und förderte die Erstellung von Bibliotheken. Dieses Konzept konnte die weiter steigende Komplexität der Programme nicht mehr bewältigen, so dass Mitte der 80er Jahre die objektorientierte Programmierung ins Leben gerufen wurde. Mit ihr gab es eine noch bessere Strukturierung und erstmals eine brauchbare Wiederverwendbarkeit. Objekte sind jedoch in den meisten Fällen zu feingranular und schränken zum Teil sehr stark auf die verwendete Sprache und Umgebung ein. Dies führte schließlich zur komponentenbasierten Entwicklung und Strukturierung, die eine verständliche Beschreibung umfangreicher Architekturen ermöglichte. In dieser Kette, die durch die folgende Abbildung veranschaulicht wird, folgen die Serviceorientierten Architekturen als das logisch nächste Glied (Dostal, et al., 2005).

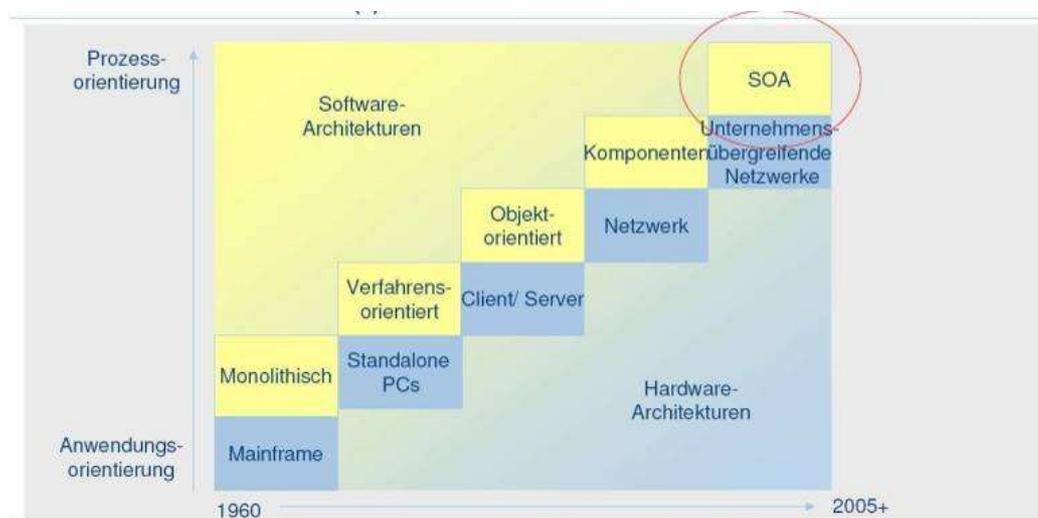


Abbildung 3.26: Entwicklung vom Mainframe zur SOA (Manhart, 2006)

Das Ziel einer SOA ist es, die Unternehmensziele in Einklang mit den verwendeten Softwaresystemen zu bringen. Durch die Definition von Services und Schnittstellen soll dies zu Flexibilität und Kosteneinsparungen führen. Die SOA ist nicht an eine bestimmte Technologie gebunden. Die Realisierung kann beispielsweise mit Common Object Request Broker Architecture (CORBA), Distributed Component Object Model (DCOM), Remote Method Invocation (RMI) oder Web-Services erfolgen (Ferdinand, 2005).

### 3.3.3 Bewertung

Die Darstellung der beiden Modellarchitekturen macht deutlich, dass sie sich von ihrer Ausrichtung her nicht miteinander vergleichen lassen. Beide Architekturen adressieren das Thema Modellierung auf unterschiedliche

Weise. SOA konzentriert sich auf die stereotypischen Rollen von Modellen, die auf der Trennung von Zuständigkeiten basieren. MDA konzentriert sich auf Abstraktionsebenen, die die Rolle von Modellen innerhalb eines Prozesses definieren. Werden diese beiden Architekturen miteinander kombiniert, entsteht einerseits die konsequente Kapselung von Funktionalitäten und andererseits deren Wiederverwendbarkeit in Form von Diensten. Im Ergebnis ist eine schnelle Anpassbarkeit von Systemen an sich ändernde Geschäftsprozesse gegeben. Der MDA-Ansatz kann innerhalb einer SOA die Komplexität einer Anwendungsumgebung zu beherrschen helfen und die Konsistenz, Übersicht und Wartbarkeit sicherstellen. Doch wie lässt sich der MDA-Ansatz beim Aufbau einer SOA konkret nutzen?

In einer Service-orientierten Anwendungsumgebung sind Services für die fachlichen Aufgaben zu definieren und die Schnittstellen zu den Service-verarbeitenden Anwendungen zu erstellen. Betrachtet man diese Aufgaben im Zusammenhang mit MDA, ergibt sich eine typische Vorgehensweise. Zunächst werden die Services einer Anwendung durch Signatur und Semantik spezifiziert und die Schnittstellen mit den Anwendungen abgestimmt. Danach ist anhand der Modellinformationen der Schnittstellen-Code zu generieren. Das folgende UML-Diagramm zeigt den Zusammenhang zwischen einem Service und einer Anwendung.

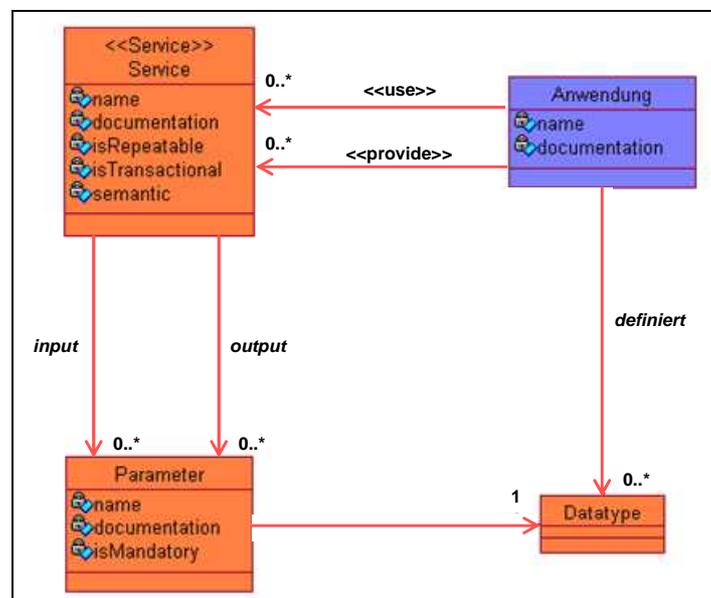


Abbildung 3.27: Service und Anwendung in UML nach (George, Westermann 2006)

Eine Anwendung kann Services bereitstellen oder sie verwenden. Ein Service hat festgelegte Ein- und Ausgabeparameter, deren Datentyp durch die Anwendung festgelegt wird.

Nach den Ansätzen der MDA und den Grundsätzen und Vorschlägen der Norm ISO 19118 Encoding kann aus den Definitionen des UML-Anwendungsschemas plattformunabhängiges XML erzeugt werden. Ebenso können zu Dokumentationszwecken Anwendungsbeschreibungen oder Objektkataloge erstellt werden. XML eignet sich sozusagen als Transportformat und wird in einem weiteren Schritt um plattformspezifischen Quellcode angereichert. Für diesen Schritt sind so genannte Codegeneratoren zu spezifizieren. Solche Generatoren sind oftmals mit Standardfunktionalitäten innerhalb der Modellierungssoftware enthalten. Diese müssen in der Regel auf die speziellen Anforderungen angepasst werden. Die folgende Abbildung zeigt beispielhaft den Generierungsprozess.

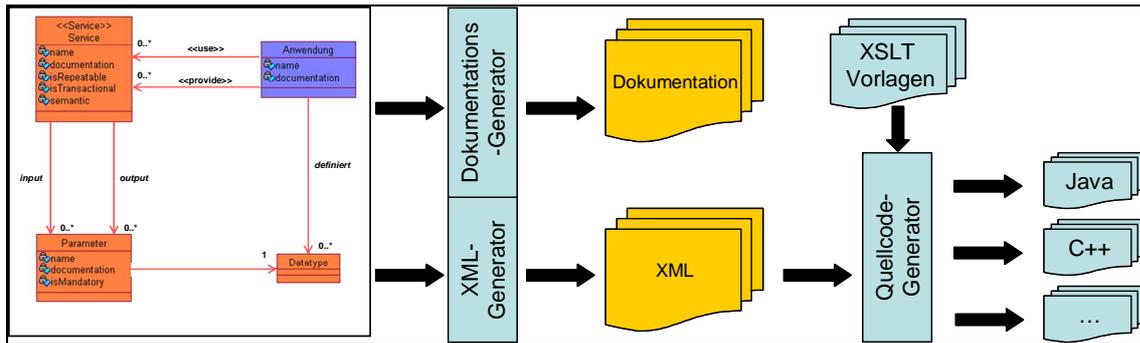


Abbildung 3.28: Ablauf des Generierungsprozesses nach (Richter, et al., 2005)

Dieses Beispiel zeigt die grundsätzliche Möglichkeit einer Nutzung der MDA-Ansätze innerhalb einer Service-orientierten Systemarchitektur. Allerdings muss deutlich herausgestellt werden, dass hierbei Kosten und Nutzen gegeneinander aufzuwiegen sind. Für eine kleinere Integrationsaufgabe ist es sicherlich nicht lohnenswert, den dargestellten Aufwand zu betreiben. Die Definition des Anwendungsschemas mit UML und die Erstellung oder Anpassung der Generatoren nimmt einen hohen Kostenanteil in Anspruch, der sich erst bei größeren Integrationsvorhaben amortisieren kann.

### 3.4 Web Services

Web Services sind eine mögliche Implementierung einer Service-orientierten Architektur. Ähnlich wie bei der SOA gibt es auch für Web Services eine Vielzahl an Definitionen. Eine Sammlung von existierenden Definitionen hat (Dostal, et al., 2005) aufgeführt. Diesen Definitionsansätzen ist gemein, dass ein Web-Service als Technologie anzusehen ist, die es ermöglicht, auf verteilten Systemen Anwendungen zu implementieren.

Der Begriff Web Service ist relativ neu, jedoch nicht die Idee, welche die Grundlage bildet. Bei der Verwendung von Web Services werden erprobte und weit verbreitete Internet-Standards wie z.B. das Hypertext Transfer Protocol (HTTP) und die Extensible Markup Language (XML) genutzt. Durch die Verwendung dieser Standards entsteht eine offene und flexible Architektur, die unabhängig von der verwendeten Plattform und Programmiersprache ist. Ein Web Service ist folglich keine Technik, sondern eine Menge von Spezifikationen und Standards, die zusammen eine Technologie bilden. Vor diesem Hintergrund ist die Definition des W<sup>3</sup>C verständlicher (W<sup>3</sup>C, 2007):

*„A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.“*

Das grundlegende Konzept einer Service-orientierten Architektur bilden

- Kommunikation,
- Dienstbeschreibung und
- Verzeichnisdienst.

In der Umsetzung dieser Architektur mit Web Services werden die Kernkomponenten mit folgenden Standards beschrieben:

- SOAP (Simple Object Access Protocol) beschreibt das XML-basierte Nachrichtenformat der Kommunikation und dessen Eingliederung in ein Transportprotokoll.
- WSDL (Web Service Description Language) ist eine XML-basierte Beschreibungssprache, um Dienste (Web-Services) zu beschreiben.
- UDDI (Universal Description, Discovery and Integration Protocol) beschreibt einen Verzeichnisdienst für Web-Services. Es wird eine standardisierte Verzeichnisstruktur für die Metadatenverwaltung der Web-Services spezifiziert. Dazu zählen allgemeine Anforderungen, Eigenschaften oder benötigte Informationen zum Auffinden von Web Services.

Das Zusammenspiel der Standards, die sich bereits in diesem Bereich durchgesetzt haben, wird im Web-Services-Dreieck in der folgenden Abbildung deutlich.

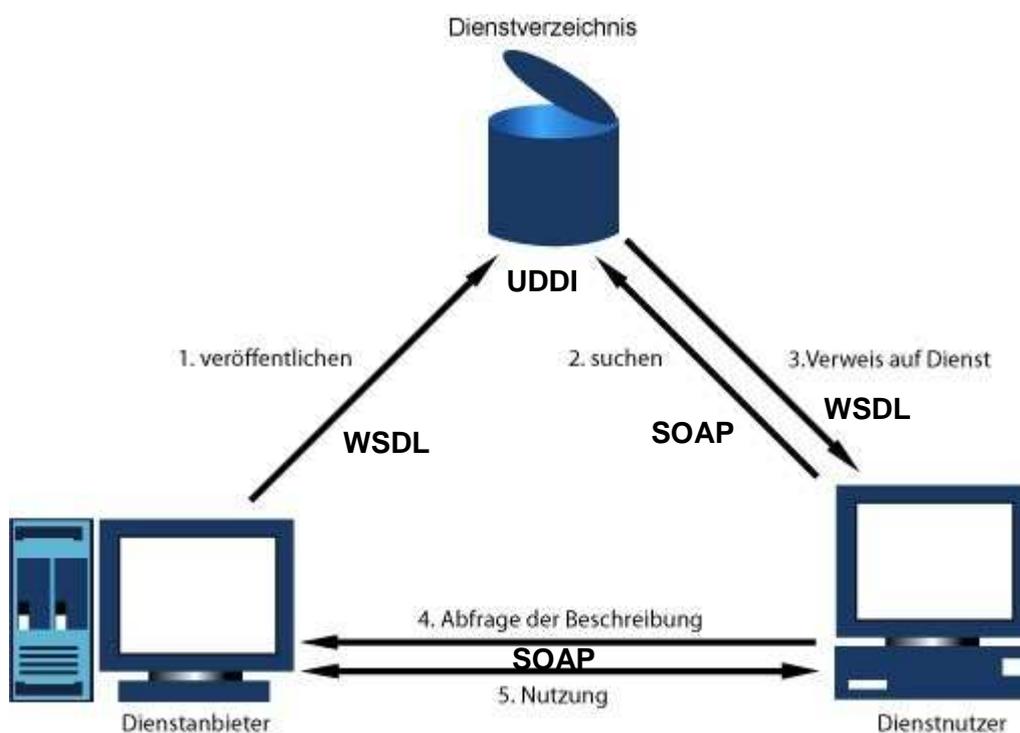


Abbildung 3.29: Das Web-Service-Dreieck nach (Dostal, et al., 2005)

Der Begriff der Web-Service-Architektur beschreibt ein Entwicklungsmodell für Internetanwendungen, welches vor allem von der just-in-time-Integration von Diensten abhängt. Just-in-time-Integration meint eine Einbindung von benötigten Komponenten zur Laufzeit. Die Web-Service-Architektur manifestiert sich im Publish-Find-Bind-Paradigma. Die Metadaten eines Web Services werden von einem Service-Anbieter mithilfe einer Service-Registrierung veröffentlicht (*publish*). Ein Service-Konsument hat die Möglichkeit eine Service-Registrierung nach einem geeigneten Service zu durchsuchen. Wird ein geeigneter Service gefunden (*find*), so kann dieser direkt vom Service-Konsumenten konsumiert (*bind*) werden (Kiehle, 2006).

Ein Web Service ist nicht für menschliche Benutzer gedacht, sondern für Softwaresysteme, die automatisiert Daten austauschen und Funktionen auf entfernten Rechnern aufrufen. Es handelt sich demnach um eine Kommunikation ausschließlich zwischen Maschinen. Dies unterscheidet die Web-Service-Architektur vom Client-Server-Ansatz des Internets. Hierbei greift der Client auf die Daten und Funktionalitäten des Servers zu und stellt die Inhalte, die er vom Server geliefert bekommt, im Browser dar. Der Web-Service-Ansatz dagegen beinhaltet den Austausch von Daten zwischen verschiedenen Anwendungen (Hauser, Löwer, 2004). Laut (Bien, 2004) lassen sich Web-Services wie folgt charakterisieren:

- Die Funktionalität eines Services ist für sich abgeschlossen.
- Ein Service sollte von anderen Services unabhängig sein.
- Ein Service sollte technologieneutral entworfen werden.
- Die Dokumentation eines Service sollte für seine Verwendung ausreichen.
- Ein Service sollte über (auch formal definierte) Metadaten verfügen, um dadurch einen automatisierten Zugriff auf die Funktionalität einer Komponente zu erleichtern.

Durch die Eigenschaft der Kapselung dienen Web Services als Abstraktionsschicht zwischen einer spezifischen Anwendung und einem Nutzer. Dadurch ist die Implementierung austauschbar. Dies verdeutlicht die folgende Abbildung.

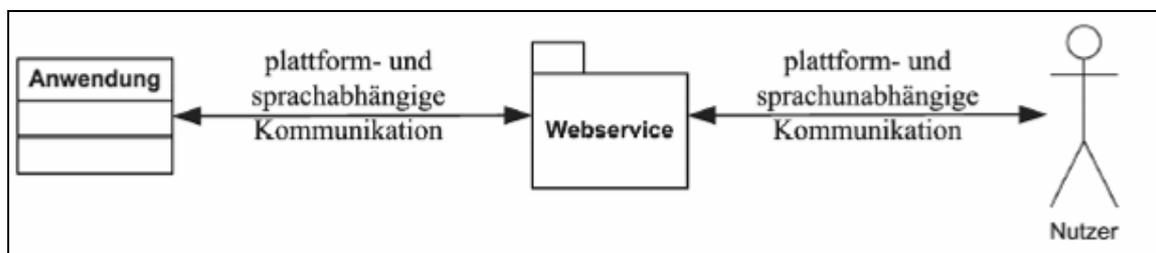


Abbildung 3.30: Web-Services fungieren als Abstraktionsschicht (Kiehle, 2006)

Auf Grund der Verwendung von allgemein anerkannten Standards wie (Hypertext Transport Protocol (HTTP), Simple Mail Transfer Protocol (SMTP) oder Extensible Markup Language (XML) erreichen Web Services eine echte Unabhängigkeit von Plattformen und Implementierungen. Die folgende Abbildung zeigt eine Zusammenfassung der Web-Service-Standards nach dem W<sup>3</sup>C.

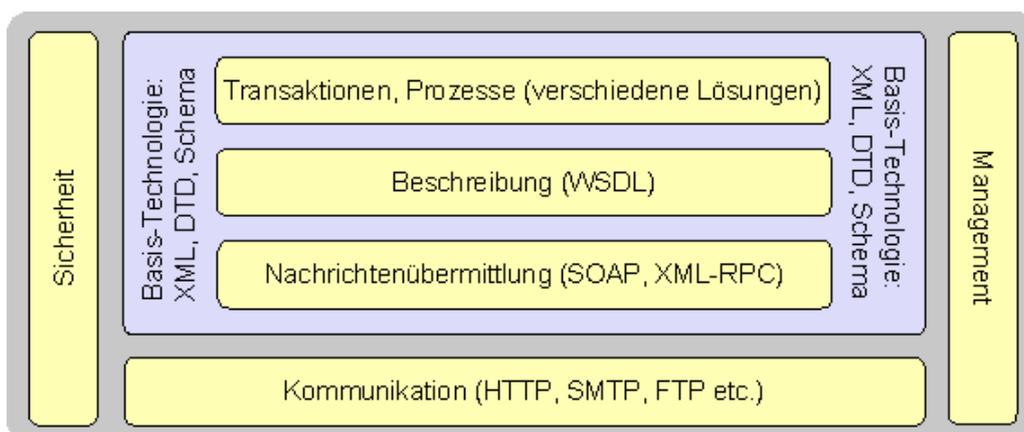


Abbildung 3.31: Web Service Architektur Stapel (Ferdinand, 2005)

Im Folgenden werden die wesentlichen Basiskomponenten von Web Services näher beschrieben.

### 3.4.1 Simple Object Access Protocol (SOAP)

SOAP stellt die Kommunikationskomponente von Web-Services dar, wobei es grundsätzlich um den Austausch von XML-Nachrichten geht. SOAP basiert selbst auf XML und ist unabhängig von der Programmiersprache und der Plattform auf der die Applikation läuft.

Die Spezifikation legt lediglich fest, wie eine Nachricht aufgebaut sein muss, um als SOAP-Nachricht gelten zu können. Die Umsetzung dieser Nachricht in die für den Benutzer optimale Laufzeitumgebung bleibt ihm selbst überlassen. SOAP wird vom W<sup>3</sup>C verwaltet und liegt aktuell in der Version 1.2 vor. Bis zu dieser Version war SOAP eine Abkürzung für Simple Object Access Protocol. In der aktuellen Version wurde der Name SOAP zwar beibehalten, aber auf das Akronym wurde verzichtet.

SOAP ist nicht an ein bestimmtes Transportprotokoll gebunden. Es wurde aber im Hinblick auf HTTP entworfen, so dass SOAP zum Datenaustausch zwischen Programmen in allen gängigen Netzwerkumgebungen eingesetzt werden kann. Die Kommunikation über SOAP wird nicht von Firewalls beeinträchtigt, so dass sich das Protokoll für die ungehinderte Übertragung über das Internet eignet. Die SOAP-Spezifikation in der Version 1.2 umfasst mehrere Dokumente:

- **SOAP Version 1.2 Part 0 (Primer)**

Dieser Teil stellt eine Einführung ohne normativen Charakter dar.

- **SOAP Version 1.2 Part 1 (Messaging Framework)**

Teil 1 der Spezifikation legt normativ den Rahmen der Nachricht fest, indem die Elemente beschrieben werden. Auch die Beförderung der Nachricht über ein Transportprotokoll ist Gegenstand dieses Spezifikationsteils.

- **SOAP Version 1.2 Part 2 (Adjuncts)**

Im Teil 2 der Spezifikation wird speziell für HTTP beschrieben, wie eine Anbindung an ein Transportprotokoll aussehen kann. Des Weiteren wird ein Datenmodell für SOAP und ein Codierungsschema für entfernte Methodenaufrufe (Remote Procedure Call, RPC) definiert.

In der Praxis wird SOAP bevorzugt in Verbindung mit HTTP eingesetzt. SOAP und das verwendete Transportprotokoll gehen eine Bindung ein. Hieraus hat sich der Begriff „binding“ gebildet. Wird SOAP in Kombination mit HTTP eingesetzt, nutzt es eine HTTP-Verbindung, um RPCs durchzuführen. Hierbei werden die HTTP-Methoden GET und POST genutzt. Die Nachrichtenübermittlung mit HTTP und der Nachrichtenpfad mit SOAP sind beide statuslos. Dies bedeutet, dass auf dem Weg der aktuelle Status der Nachricht nicht bekannt ist (Hauser, Löwer 2004). SOAP-Nachrichten können als einfache Nachrichten in nur eine Richtung jedoch auch an mehrere Empfänger versendet werden. Die folgende Abbildung zeigt die verschiedenen Nachrichtenarten, die zwischen Sendern und Empfängern auftreten können.

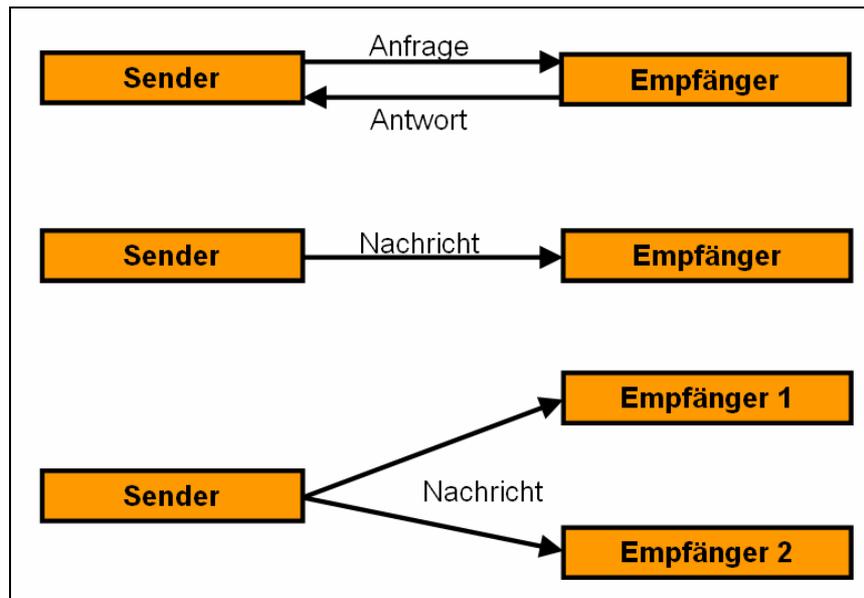


Abbildung 3.32: Verschiedene SOAP-Nachrichtenarten nach (Hauser, Löwer, 2004)

Neben den bekannten Sendern und Empfängern von Nachrichten kann es auch so genannte Vermittler oder Intermediäre (SOAP intermediaries) geben, die die Nachricht weiterleiten und unter Umständen auch Teile daraus selbst nutzen. Intermediäre werden über ihre URI identifiziert und ihr Verhalten wird in Rollen (SOAP roles) festgelegt. Alle drei im Zuge der Übermittlung partizipierenden Parteien werden Knoten (nodes) genannt.

Eine SOAP-Nachricht ist ein in drei Teile aufgeteiltes XML-Dokument. Diese Teile sind der SOAP-Envelope, der SOAP-Header und der SOAP-Body. Während der SOAP-Header optional ist, müssen die beiden Elemente Envelope und Body in jeder SOAP-Nachricht vorhanden sein. Das XML-Dokument ist wie folgt aufgebaut:

- Der **Envelope** ist das oberste Element des XML-Dokuments und umschließt die gesamte Nachricht ähnlich einem Umschlag.
- Der **Header** ist ein generischer Mechanismus, der eine SOAP-Nachricht um Funktionalität erweitert, die vorher von den kommunizierenden Anwendungen nicht abgestimmt worden ist. Dazu werden Attribute definiert, die angeben, wer diese Aufgaben umsetzen soll und ob es sich um optionale oder verpflichtende Aufgaben handelt.
- Der **Body** enthält Informationen für den eigentlichen Empfänger der Nachricht. Hier ist ein Element vordefiniert, das zum Melden von eventuell aufgetretenen Fehlern benötigt wird (Fault-Element).

Der SOAP-Header darf hierbei nicht mit dem HTTP-Header verwechselt werden. Wird eine SOAP-Nachricht über HTTP verschickt, kommt der HTTP-Header noch hinzu. Die folgende Abbildung zeigt die allgemeine Struktur einer SOAP-Nachricht.

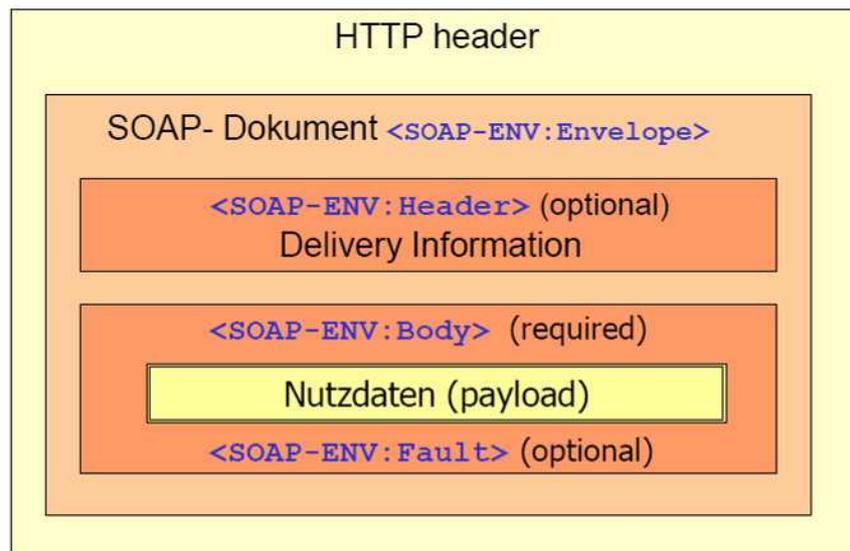


Abbildung 3.33: Aufbau einer SOAP-Nachricht (Stümpert, 2003)

Innerhalb des Body-Teils sind die eigentlichen (Nutz-) Daten enthalten. Generell wird unterschieden zwischen einer Anfrage (Request), einer Antwort (Response) und einer Fehlermeldung (Fault). Im Folgenden soll anhand eines Codebeispiels der Aufbau einer SOAP-Nachricht erläutert werden. Im dargestellten Anwendungsfall handelt es sich um ein Reservierungsprogramm, das eine geplante Dienstreise für Firmenangestellte über ein Buchungssystem ausführt. Der Nachrichtenaustausch erfolgt über SOAP-Nachrichten. Das Beispiel ist dem Primer der SOAP Version 1.2 (Part 0) entnommen (W3C, 2003). Das Codebeispiel 1 zeigt die Reservierungsanfrage als SOAP-Nachricht.

```
<?xml version='1.0' ?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reservation
      xmlns:m="http://travelcompany.example.org/reservation"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:reference>
      <m:dateAndTime>2001-11-29T13:20:00.000-05:00</m:dateAndTime>
    </m:reservation>
    <n:passenger
      xmlns:n="http://mycompany.example.com/employees"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <n:name>Áke Jógvan Øyvind</n:name>
    </n:passenger>
  </env:Header>
  <env:Body>
    <p:itinerary
      xmlns:p="http://travelcompany.example.org/reservation/travel">
      <p:departure>
        <p:departing>New York</p:departing>
        <p:arriving>Los Angeles</p:arriving>
        <p:departureDate>2001-12-14</p:departureDate>
        <p:departureTime>late afternoon</p:departureTime>
        <p:seatPreference>aisle</p:seatPreference>
      </p:departure>
      <p:return>
        <p:departing>Los Angeles</p:departing>
        <p:arriving>New York</p:arriving>
        <p:departureDate>2001-12-20</p:departureDate>
        <p:departureTime>mid-morning</p:departureTime>
        <p:seatPreference/>
      </p:return>
    </p:itinerary>
  </env:Body>
</env:Envelope>
```

```

    </p:return>
  </p:itinerary>
  <q:lodging
    xmlns:q="http://travelcompany.example.org/reservation/hotels">
    <q:preference>none</q:preference>
  </q:lodging>
</env:Body>
</env:Envelope>

```

Abbildung 3.34: SOAP-Nachricht als Anfrage für eine Reisereservierung (W<sup>3</sup>C, 2003)

In einer SOAP-Nachricht wird zuerst die XML-Version und eventuell der verwendete Zeichensatz angegeben. Diese Angaben sind optional. Im Beispiel findet sich die XML-Version wieder. Der Header beinhaltet zwei Header-Blöcke (*reservation* und *passenger*), die beide über einen eigenen XML-Namensraum definiert werden. Der Header-Block *reservation* enthält Referenz und Zeitstempel und die Identität des Reisenden wird im Header-Block *passenger* dargestellt. Die beiden Header-Blöcke müssen von den nächsten innerhalb des Nachrichtenpfades auftretenden SOAP-Zwischenstationen verarbeitet werden oder, wenn es keinen Intermediary gibt, von dem definierten Empfänger der Nachricht. Über das Attribut *env:role* mit dem Wert "http://www.w3.org/2002/12/soap-envelope/role/next" wird angezeigt, dass die Nachricht an den nächsten SOAP-Knoten der Route gerichtet ist. Alle SOAP-Knoten müssen diese Rolle (*role*) verstehen. Das Attribut *env:mustUnderstand* mit dem Wert "true" zeigt, dass die oder der Knoten die Header-Blöcke vollständig auf eine eigens für den Header-Block spezifizierten Weise verarbeitet werden muss, andernfalls soll die Verarbeitung verworfen und ein Fehler (*fault*) generiert werden. Der Body-Block ist in zwei Sub-Elemente unterteilt. Der Block *p:itinerary* enthält die Informationen zum gewünschten Hin- und Rückflug. Der Block *q:lodging* gibt Präferenzen für die Übernachtung an. Das Codebeispiel 2 zeigt die SOAP-Nachricht als Antwort auf die Anfrage von Beispiel 1.

```

<?xml version='1.0' ?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reservation
      xmlns:m="http://travelcompany.example.org/reservation"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:reference>
      <m:dateAndTime>2001-11-29T13:35:00.000-05:00</m:dateAndTime>
    </m:reservation>
    <n:passenger
      xmlns:n="http://mycompany.example.com/employees"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <n:name>Åke Jógvan Øyvind</n:name>
    </n:passenger>
  </env:Header>
  <env:Body>
    <p:itineraryClarification
      xmlns:p="http://travelcompany.example.org/reservation/travel">
      <p:departure>
        <p:departing>
          <p:airportChoices>JFK LGA EWR</p:airportChoices>
        </p:departing>
      </p:departure>
      <p:return>
        <p:arriving>
          <p:airportChoices>JFK LGA EWR</p:airportChoices>
        </p:arriving>
      </p:return>
    </p:itineraryClarification>
  </env:Body>
</env:Envelope>

```

Abbildung 3.35: SOAP-Nachricht als Antwort des Buchungssystems (W<sup>3</sup>C, 2003)

In der SOAP-Nachricht als Antwort beinhaltet der Body wiederum den Inhalt der Nachricht. In diesem Fall werden jeweils drei Flugplätze zur Auswahl angeboten. Der Body ist an das Schema des Namensraumes „http://travelcompany.example.org/reservation/travel“ angelehnt: Die Header-Blöcke werden mit ihren Sub-Elementen innerhalb der Antwort returniert, so dass die Nachrichten auf SOAP-Ebene in Beziehung gesetzt werden können. Die Header können jedoch auch einen anderen anwendungsspezifischen Nutzen besitzen (W<sup>3</sup>C, 2003).

Wie bereits erwähnt, benötigt SOAP ein Transportprotokoll, das die Nachricht aufnimmt. In der Praxis wird meistens HTTP als Transportprotokoll eingesetzt. SOAP kann jedoch noch mit weiteren Transportprotokollen eine Verbindung eingehen. Als Alternative zu HTTP zur Übertragung von Nachrichten gilt das Simple Mail Transfer Protocol (SMTP). Das Mail-Format SMTP erlaubt lediglich einen asynchronen Nachrichtenaustausch. Für RPCs ist SMTP also nicht direkt geeignet. Das File Transfer Protocol (FTP) dient im Wesentlichen zur Übertragung von Daten. Der Versand von Nachrichten geschieht in der Praxis eher selten. Das Blocks Extensible Exchange Protocol (BEEP) ist ein noch relativ unbekanntes Transportprotokoll, welches derzeit noch näher spezifiziert wird. BEEP setzt direkt auf dem Transmission Control Protocol (TCP) auf und bietet Authentifizierungs- und Sicherheitsmechanismen sowie ein Fehlermanagement an. Das Protokoll ist leicht erweiterbar und es lassen sich neue Protokolle direkt darauf aufsetzen. BEEP hat gegenüber HTTP den Vorteil, dass es mit weniger Overhead auskommt und auch eine asynchrone Kommunikation zulässt. BEEP wird bisher nur vereinzelt eingesetzt (Hauser, Löwer, 2004).

SOAP, WSDL und UDDI bilden die drei grundlegenden Standards für Web-Services. SOAP ist für die Kommunikation zwischen dem Service-Anbieter und dem Service-Konsumenten zuständig. Die beiden anderen Standards werden im Folgenden beschrieben.

### 3.4.2 Web Service Description Language (WSDL)

Die Beschreibung von Schnittstellen ist ein wesentlicher Baustein für ein verteiltes System wie die Service-orientierte Architektur. Eine neutrale und vollständige Schnittstellenbeschreibung ermöglicht es, dass voneinander unabhängige Parteien miteinander kommunizieren. So können Systeme flexibel aus unterschiedlichen Komponenten zusammengesetzt werden. Die Schnittstellenbeschreibungssprache Web-Service Description Language (WSDL) stellt ein XML-basierendes Vokabular zur Beschreibung von Schnittstellen sowie deren Operationen und Dienste dar (Dostal, et al., 2005). WSDL liegt in der Version 2.0 derzeit als Candidate Recommendation beim W<sup>3</sup>C, so dass sie kurz vor der Verabschiedung zum W<sup>3</sup>C -Standard steht. Ebenso wie bei der Definition von SOAP gibt es drei Teile von WSDL:

- **WSDL Version 2.0 Part 0 (Primer)**

Dieser Teil stellt eine Einführung ohne normativen Charakter dar.

- **WSDL Version 2.0 Part 1 (Core Language)**

Teil 1 der Spezifikation legt normativ die Basissprache (Core Language) fest, in der Web Services beschrieben werden. Sie basiert auf einem abstrakten Modell der Funktionalitäten, die der Web Service bereitstellt. Kriterien für die Konformität zu WSDL sind ebenfalls Gegenstand dieser Spezifikation.

- **WSDL Version 2.0 Part 2 (Adjuncts)**

Im Teil 2 werden die vordefinierten Erweiterungen aus Teil 1 näher beschrieben. Es handelt sich um Message exchange patterns, Operation styles und Binding Extensions.

WSDL ist gleichzeitig ein Modell als auch eine XML-Sprache. Ein Web Services wird aus zwei Blickrichtungen beleuchtet: auf der Ebene der Funktionalität (abstrakt) und auf der Ebene der technischen Details (konkret). Eine semantische Beschreibung des Dienstes findet nicht statt. Die Beschreibung des Web Services basiert auf einem abstrakten Modell dessen, was der Service anbietet. Dazu wird die Beschreibung der Funktionalität von den

technischen Details über den Ort und die Art und Weise, wie ein Dienst angeboten wird, getrennt. Die Konstrukte von WSDL ermöglichen die Wiederverwendung an anderer Stelle. Die in der nachfolgenden Abbildung aufgeführten Komponenten stehen zur Beschreibung eines Dienstes zur Verfügung.

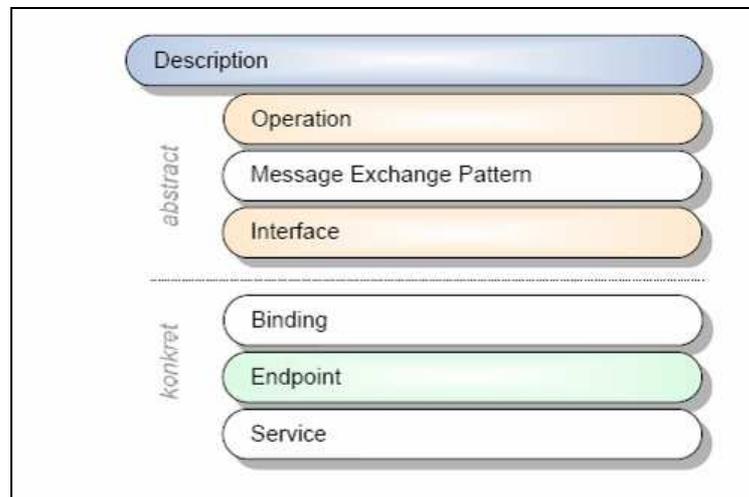


Abbildung 3.36: Überblick über die WSDL-Komponenten (Dostal, et al., 2005)

Die Dokumentstruktur eines XML-Dokuments besteht aus mehreren verschachtelten Elementen. Diese Elemente werden im nachfolgend beschrieben. Das Grundgerüst wird von den folgenden Elementen gebildet (nach Hauser, Löwer, 2004):

- *definitions*: als Wurzelement des Dokumentes, welches den Namen des Service, Namespaces für den Service und für verwendete Standards enthält. Des Weiteren fungiert dieses Element als Umschlag für die WSDL-Komponenten (*interfaces*, *bindings*, *services*) und *type system*-Komponenten (Elementdeklarationen und Typdefinitionen).
- *types*: definiert die Datentypen, die standardmäßig aus der XML-Schema-Spezifikation verwendet werden.
- *message*: definiert die Struktur der Nachrichten, die bei der Kommunikation mit dem Servicenutzer ausgetauscht werden. Sind mehrere Nachrichten, beispielsweise Anfrage und Antwort enthalten, gibt es mehrere *message*-Elemente.
- *binding*: gibt die Art des Protokolls und Datenformats an, die von in einem *interface* definierten Operationen benötigt werden.
- *Interface* ist eine Kapsel für *operations*, die ihrerseits Nachrichten (*messages*) enthalten, welche zu oder von einem Service gesendet werden.
- *service*: definiert die Endpunkte eines Dienstes. Jeder Endpunkt steht mit einer URI im untergeordneten *endpoint*-Element. Das *service*-Element enthält die Netzwerkadressen der einzelnen Schnittstellen zum System.
- *documentation*: enthält eine textliche Beschreibung des Dienstes. Hier können weitere Angaben zur Nutzung des Dienstes und seinen Schnittstellen gemacht werden.

Die folgende Abbildung zeigt den hierarchischen Aufbau der WSDL-Komponenten.

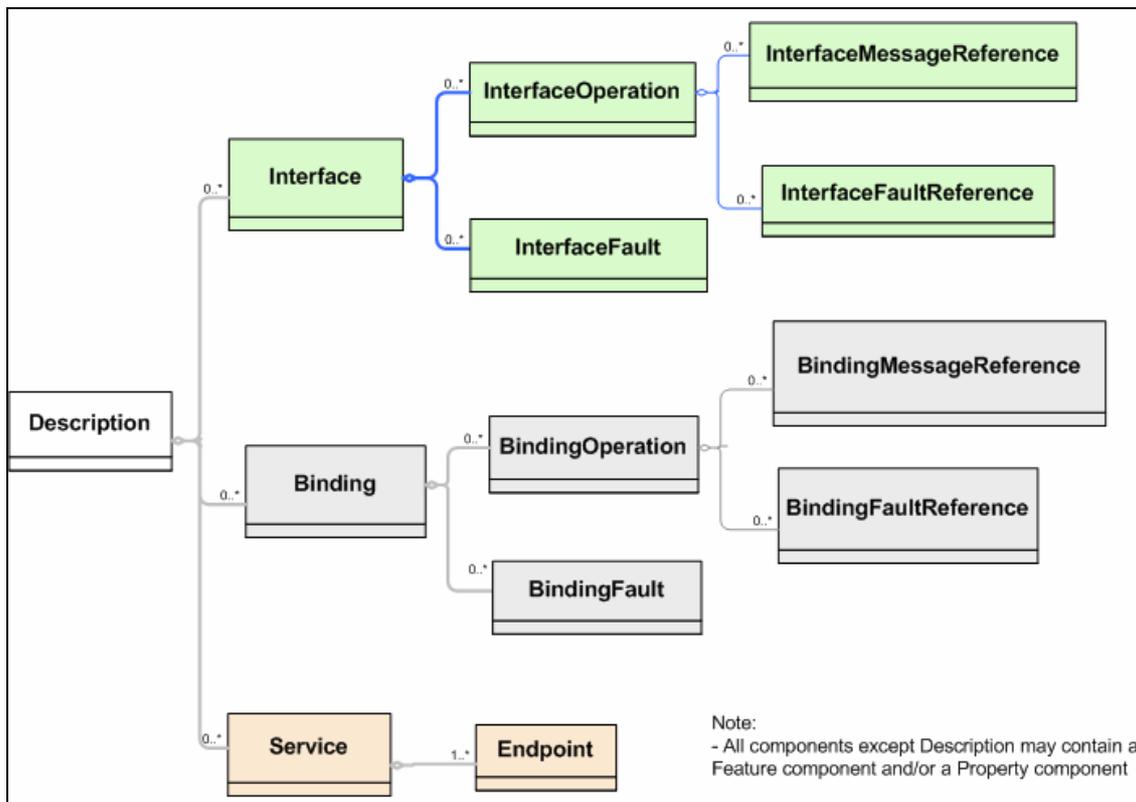


Abbildung 3.37: WSDL 2.0-Komponenten mit Hierarchie (W<sup>3</sup>C, 2006)

Die nachfolgende Abbildung zeigt ein WSDL 2.0 Dokument aus dem Primer der Web Services Description Language (WSDL) Version 2.0 (Part 0). Das Beispielszenario lautet wie folgt: Ein fiktives Hotel „GreatH“ stellt einen Reservierungsservice über einen Web Service bereit. Es handelt sich um einen Verfügbarkeitservice (CheckAvailability) für ein Hotelzimmer. Um die Verfügbarkeit abzufragen, muss der Client ein Datum für den Check-In und Check-Out sowie eine Zimmerkategorie angeben. Als Antwort retourniert der Web Service einen Zimmerpreis, wenn die angefragte Kategorie verfügbar ist. Falls nicht, wird ein Preis von Null zurückgegeben. Wenn Eingangsdaten ungültig sind, gibt der Web Services eine Fehlermeldung zurück. Der Service akzeptiert eine `checkAvailability` message und sendet entweder eine `checkAvailabilityResponse` oder eine `invalidDataFault` message zurück.

```

<?xml version="1.0" encoding="utf-8" ?>
<description
  xmlns="http://www.w3.org/2006/01/wsdl"
  targetNamespace="http://greath.example.com/2004/wsdl/resSvc"
  xmlns:tns="http://greath.example.com/2004/wsdl/resSvc"
  xmlns:ghns="http://greath.example.com/2004/schemas/resSvc"
  xmlns:wsoap="http://www.w3.org/2006/01/wsdl/soap"
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsdlix="http://www.w3.org/2006/01/wsdl-extensions">

  <documentation>
    This document describes the GreatH Web service. Additional
    application-level requirements for use of this service --
    beyond what WSDL 2.0 is able to describe -- are available
    at http://greath.example.com/2004/reservation-documentation.html
  </documentation>

```

```

<types>
  <xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://greath.example.com/2004/schemas/resSvc"
    xmlns="http://greath.example.com/2004/schemas/resSvc">

    <xs:element name="checkAvailability" type="tCheckAvailability"/>
    <xs:complexType name="tCheckAvailability">
      <xs:sequence>
        <xs:element name="checkInDate" type="xs:date"/>
        <xs:element name="checkOutDate" type="xs:date"/>
        <xs:element name="roomType" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>

    <xs:element name="checkAvailabilityResponse" type="xs:double"/>

    <xs:element name="invalidDataError" type="xs:string"/>

  </xs:schema>
</types>

<interface name = "reservationInterface" >

  <fault name = "invalidDataFault"
    element = "ghns:invalidDataError"/>

  <operation name="opCheckAvailability"
    pattern="http://www.w3.org/2006/01/wsdl/in-out"
    style="http://www.w3.org/2006/01/wsdl/style/iri"
    wsdlx:safe = "true">
    <input messageLabel="In"
      element="ghns:checkAvailability" />
    <output messageLabel="Out"
      element="ghns:checkAvailabilityResponse" />
    <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
  </operation>
</interface>

<binding name="reservationSOAPBinding"
  interface="tns:reservationInterface"
  type="http://www.w3.org/2006/01/wsdl/soap"
  wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP">

  <fault ref="tns:invalidDataFault"
    wsoap:code="soap:Sender"/>

  <operation ref="tns:opCheckAvailability"
    wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>
</binding>

<service name="reservationService"
  interface="tns:reservationInterface">

  <endpoint name="reservationEndpoint"
    binding="tns:reservationSOAPBinding"
    address = "http://greath.example.com/2004/reservation"/>

</service>
</description>

```

Abbildung 3.38: WSDL.2.0 Beispiel für einen Reservierungsservice (W<sup>3</sup>C, 2006)

Die Unterschiede von der WSDL-Version 1.1 zur aktuellen WSDL-Version 2.0 bestehen im Wesentlichen aus folgenden Punkten, die in (Dhesiaseelan, 2004) aufgeführt werden.

- Erweiterung der Beschreibungssprache um weitere Semantik. Damit wird der targetNamespace ein Pflichtattribut im Definitionselement.
- Entfernung der Message-Konstrukte, die über die XML Schematypen im Element „types“ spezifiziert werden.
- PortTypes wurden in interfaces umbenannt. Vererbung für die interfaces wird über erweiterte Attribute im selben Element erreicht.
- Ports wurden in endpoints umbenannt.

Die folgende Abbildung zeigt die Komponenten von WSDL mit den Änderungen in den Bezeichnungen von Schnittstelle und Endpunkt des zu beschreibenden Web Services.

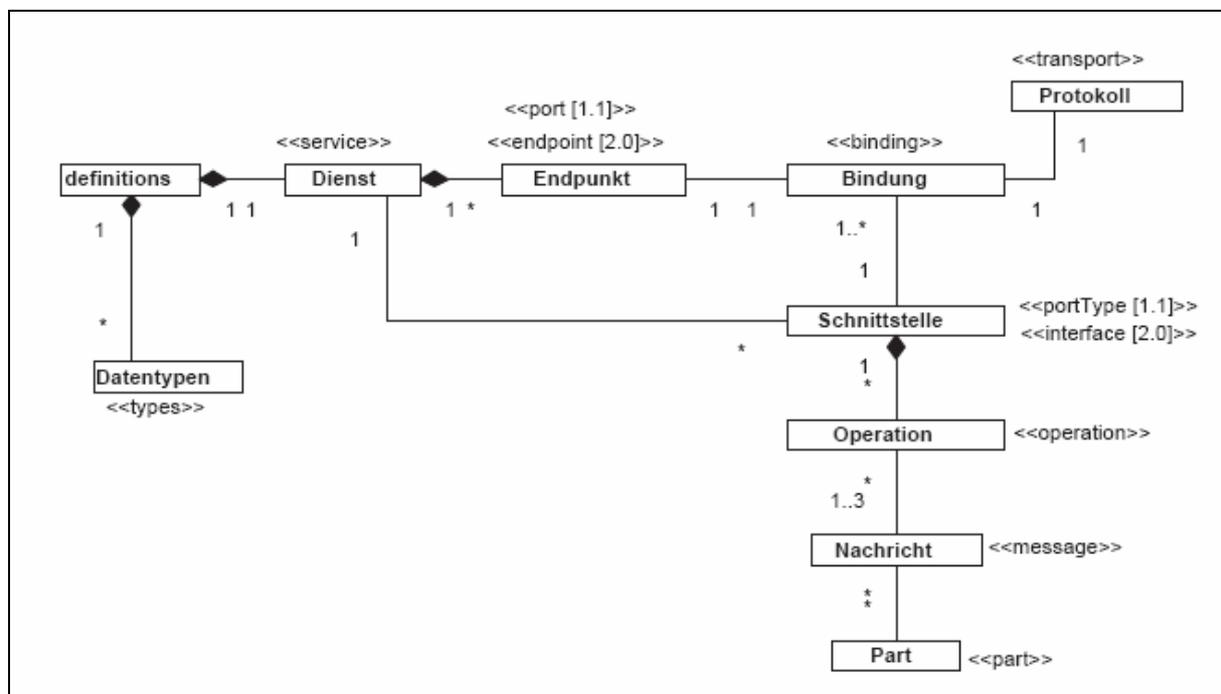


Abbildung 3.39: WSDL-Komponenten im Vergleich (Bauer, 2006)

### 3.4.3 Universal Description, Discovery and Integration (UDDI)

Der Nutzen eines Web-Services ist gering, wenn er vom Service-Konsumenten nicht aufgefunden werden kann. Verzeichnisdienste bieten eine durchsuchbare Übersicht über Ressourcen in einem Netzwerk und machen diese den Benutzern und Applikationen zugänglich. Wegen ihrer Übersichtsfunktion sind sie ein unverzichtbarer Bestandteil von Service-orientierten Architekturen. Universal Description, Discovery and Integration of Web Services (UDDI) hat sich in diesem Bereich als Standard durchgesetzt. Im Gegensatz zu SOAP und WSDL wird die Entwicklung von UDDI nicht vom W<sup>3</sup>C, sondern von OASIS vorangetrieben und liegt aktuell in der Version 3 vor.

UDDI bietet Funktionen zum Klassifizieren, Katalogisieren und Verwalten von Daten und Metadaten über Web Services an, so dass diese Dienste gefunden und verwendet werden können. UDDI selbst besteht aus zwei Komponenten (nach Dostal, et al., 2005):

- Das UDDI-XML-Schema definiert das Datenmodell und stellt die folgenden Datenstrukturen bereit: *businessEntity*, *businessService*, *bindingTemplate*, *tModel* und *publisherAssertion*.
- Das UDDI-API besteht aus einer Anzahl von Anwendungsschnittstellen, die selbst Web Services sind und somit das SOAP-Protokoll des W<sup>3</sup>C verwenden. Diese Schnittstellen dienen der Veröffentlichung, Verwaltung und Suche von Daten im Verzeichnis. Nutzer senden Nachrichten im SOAP-Format an einen UDDI-Server, der mit Ergebnissen antwortet, die in XML formuliert sind.

Das UDDI-XML-Schema wurde entwickelt, um die Hauptfunktionen von UDDI optimal zu unterstützen. Die Modellierung in UDDI stellt nicht nur die technische Beschreibung der Web Services dar, sie erlaubt darüber hinaus auch die Erfassung von Informationen über die Anbieter und Dienste. Die folgende Abbildung zeigt das UDDI-Datenmodell.

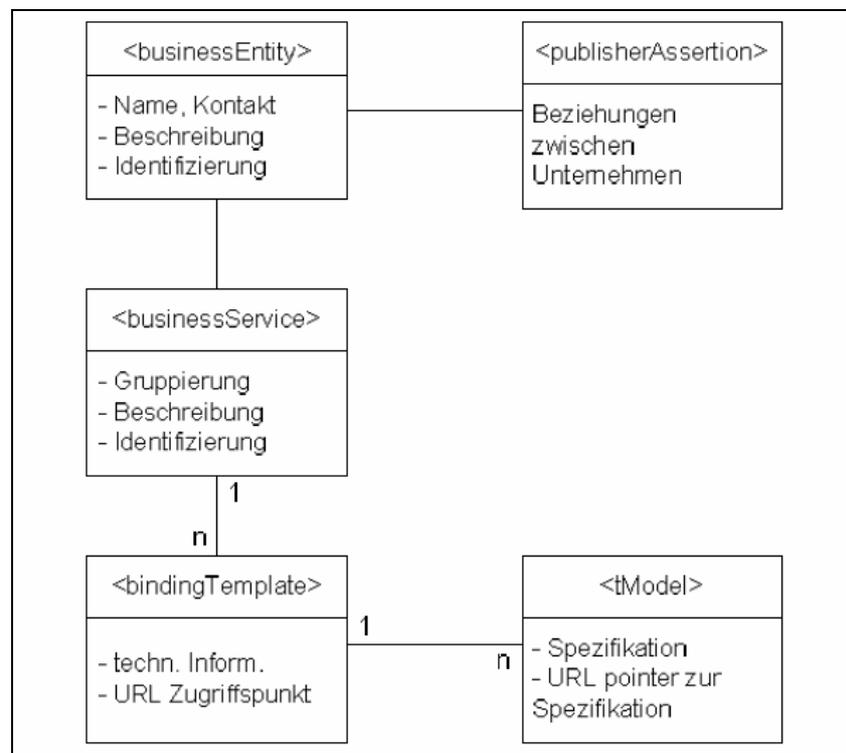


Abbildung 3.40: UDDI-Datenmodell (Schmietendorf, 2005)

Das Element `businessEntity` ist das Wurzelobjekt, von dem weitere Informationen verkettet sind. Es enthält allgemeine Angaben zum Service-Anbieter, beispielsweise Kontaktinformationen, und kann einer Kategorie zugeordnet werden. Des Weiteren beinhaltet es eine URL zum Finden weiterer Informationen (Schmietendorf, 2005). Das in dem Element `businessEntity` enthaltene Element `businessService` beinhaltet Informationen zu den vom Unternehmen angebotenen Web Services. Hier erfolgt gleichermaßen eine Einteilung in Kategorien samt Verweisen zu weiterführenden Informationen. Zugehörige technische Dienstbeschreibungen werden in dem `bindingTemplate` abgelegt. `BindingTemplates` sind Unter-elemente der `BusinessService` Elemente, zu denen sie eine n:1-Beziehung haben, d.h. zu einem `BusinessService` Element können mehrere `BindingTemplates` gehören. Es werden der technische Zugangspunkt (`accessPoint`) sowie technische Charakteristika, Parameter und Einstellungen, die dem Dienst zugeordnet werden, beschrieben. Die `BindingTemplates` verweisen auf die vierte Datenstruktur von UDDI, die `tModels`, die den Servicetyp darstellen und Informationen wie eine Kategorienliste, Schnittstellendefinition sowie Verweise auf zusätzliche Informationen enthalten. Meist enthält das `tModel` nicht selbst die technische Beschreibung des

Dienstes, sondern nur einen Link dorthin. Die meisten `tModels` Links verweisen auf ein WSDL-Dokument. Ein Servicetyp kann von unterschiedlichen Unternehmen angeboten werden. Der Nutzer prüft den Web Service anhand des `tModels` auf Kompatibilität und sucht daraufhin den passenden Dienst aus. Der Elementtyp `publisherAssertion` enthält Informationen über Beziehungen von Unternehmen wie z.B. Allianzen, Partnerschaften, Mutter/Tochtergesellschaften, um einen Missbrauch zu verhindern. Sobald ein Unternehmen ein `publishAssertion` registriert, bleibt dieses solange für andere unsichtbar, bis der beteiligte Partner ebenfalls dasselbe `publishAssertion` registriert (Schmietendorf, 2005).

Das Informationsangebot von UDDI-Verzeichnissen wird auch in drei Kategorien eingeteilt, in die so genannten Weißen Seiten (White Pages), Gelbe Seiten (Yellow Pages) und Grüne Seiten (Green Pages). Die Weißen Seiten enthalten Kontakt- und Geschäftsrelevante Informationen. Die Gelben Seiten entsprechen dem gleichnamig bekannten Branchenverzeichnis. Die technischen Informationen zu den Web Services werden in den Grünen Seiten abgelegt. Die folgende Auflistung zeigt die Zuordnung der Informationen zur Datenstruktur bei UDDI (Ferdinand, 2005).

Information Struktur	White Pages	Yellow Pages	Green Pages
<b>businessEntity</b>	+	+	-
<b>businessService</b>	-	+	-
<b>bindingTemplate</b>	-	-	
<b>tModel</b>	-	-	+

Tabelle 3.6: Zuordnung der Daten auf die Datenstruktur bei UDDI (Ferdinand, 2005)

Ein UDDI-Verzeichnis bietet zwei grundlegende Funktionen an: Das Eintragen neuer Web-Services in das Verzeichnis (Publishing) und das Abrufen (Inquiry) eingetragener Web-Services. Der Nutzer muss sich zunächst im UDDI-Verzeichnis bzw. auf der entsprechenden Webseite registrieren, um die Funktionen von UDDI vollständig nutzen zu können. Erst dann kann er das Verzeichnis durchsuchen oder einen neuen Web Service mithilfe der Web-Schnittstelle einfügen. Die eigenen UDDI-Einträge mithilfe der Web-Schnittstelle der verschiedenen UDDI-Verzeichnisse zu verwalten, widerspricht dem Web-Service-Ansatz, der für die Kommunikation zwischen Maschinen entwickelt wurde. Deshalb stellen die UDDI-Verzeichnisse Application Programming Interfaces (APIs) zur Verfügung, die Methoden enthalten, mit denen die UDDI-Einträge verwaltet werden können. Hier unterscheidet man zwischen Publishig-API und Inquiry-API (Hauser, Löwer, 2004).

Mit UDDI, WSDL und SOAP sind die drei Basisstandards für Web Services vorgestellt worden. Die folgenden Kapitel bieten einen Überblick zum angehenden Standard der Geschäftsprozesssteuerung und einen zusammenfassenden Ausblick über weitere Spezifikationen in Bereich der Web Services.

### 3.4.4 Web Service Business Process Execution Language (WS-BPEL)

Web Services sind in sich abgeschlossene Dienste, die mittels SOAP, WSDL und UDDI in einer Service-orientierten Architektur bereitgestellt werden können. Für komplexere Interaktionen können eine beliebige Anzahl von Diensten in einer definierten Reihenfolge genutzt werden. Die Prozessorientierung ist somit die vierte tragende Säule in dem bereits vorgestellten SOA-Tempel. Unter Prozessorientierung versteht man hier die Möglichkeit, einzelne Dienste so zusammenzustellen („komponieren“), dass aus ihrem Zusammenspiel ein geschäftlicher Mehrwert entsteht. Der Ablauf mehrerer Arbeitsschritte wird auch als Geschäftsprozess bezeichnet. Die Web Service Business Process Execution Language, kurz WS-BPEL, ist eine XML-basierte Sprache zur Beschreibung von Geschäftsprozessen, deren einzelne Aktivitäten durch Web Services implementiert sind.

WS-BPEL entstand auf Basis der beiden Spezifikationen WSFL von IBM und XLANG von Microsoft. WSFL und XLANG dienen zur Koordination von Prozessen auf Basis von Web Services. Bei einem Vergleich der beiden Spezifikationen stellten IBM und Microsoft fest, dass WSFL und XLANG ähnliche Ziele verfolgen und

gut zueinander passen. Aus diesem Grund entwickelten IBM und Microsoft mit den Unternehmen BEA, SAP und Siebel die Version 1.1 unter dem Namen Business Process Execution Language for Web Services (BPEL4WS). Diese Version wurde 2003 an das Standardisierungsgremium OASIS übergeben. Die Spezifikation läuft bei OASIS unter dem Namen Web Service Business Process Execution Language (WS-BPEL), um in Einklang mit den anderen WS-Standards wie WSDL, WS-Security usw. zu gelangen. Die Entwicklung von WS-BPEL wird in der folgenden Abbildung aufgezeigt.

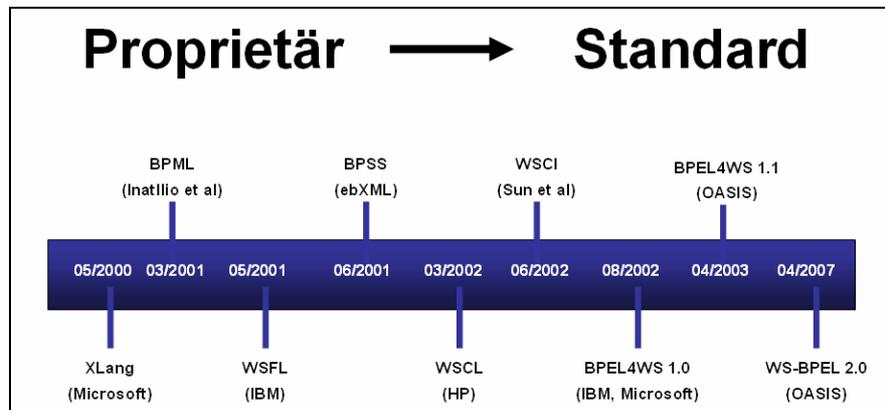


Abbildung 3.41: Entwicklung der Geschäftsprozessmodellierung nach (Dostal, et al., 2005)

Im Folgenden wird diese Spezifikation lediglich als WS-BPEL und abgekürzt BPEL bezeichnet. BPEL liegt seit Januar 2007 beim Standardisierungsgremium OASIS in der Version 2.0 als Committee Specification vor und ist im April 2007 als OASIS-Standard verabschiedet worden. Die Aktuelle Spezifikation kann unter (OASIS, 2007) abgerufen werden. Da BPEL 2.0 ein recht junger Standard und noch nicht in den erhältlichen Softwarekomponenten abgebildet wurde, wird im Weiteren mit der Version 1.1 von BPEL gearbeitet. Erwähnenswert ist, dass die neue Version Detail-Verbesserungen beinhaltet, jedoch nicht kompatibel zu den Vorgängerversionen ist.

Das Zusammenführen von Web Services zu einem Geschäftsprozess beinhaltet zwei Aspekte, die sich in der Sichtweise voneinander unterscheiden. Nach (Hansen, Neumann, 2005) wird zwischen einer externen oder auch öffentlichen Sicht und einer internen Sicht auf Geschäftsprozesse getrennt. Bei letzterer werden definiert, welche Funktionen wie zusammengefügt werden, welche Daten genutzt und welche Parteien in einem Geschäftsprozess interagieren. Die externe Sicht auf den Geschäftsprozess definiert die Reihenfolge der Nachrichten, die mit einem Geschäftsprozess ausgetauscht werden. Diese beschriebenen Aspekte werden auch als Orchestrierung oder Choreographie bezeichnet, die im Folgenden aus (Hansen, Neumann, 2006) als Definition zitiert sowie in Abbildung 3.42 und Abbildung 3.43 skizziert werden.

**„Die Orchestrierung definiert, welche Funktionen in welcher Reihenfolge ausgeführt werden, an welche Prozesse Teilaufgaben delegiert werden können, wie die Zustandsdaten eines Prozesses gespeichert oder Daten zwischen mehreren Formaten konvertiert werden müssen.“**

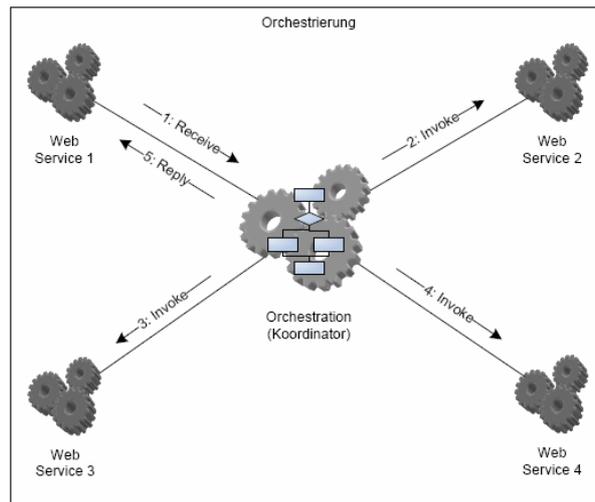


Abbildung 3.42: Prozess-Orchestrierung (Juric, et al., 2006)

*„Die Choreographie definiert das Zusammenspiel und die Interaktion mehrerer beteiligter Parteien. Dazu gehört insbesondere die Modellierung des Verhaltens von Prozessen, das heißt, welche Nachrichten sie senden oder empfangen können, und welche Zustände sie einnehmen können“* (Hansen, Neumann, 2006).

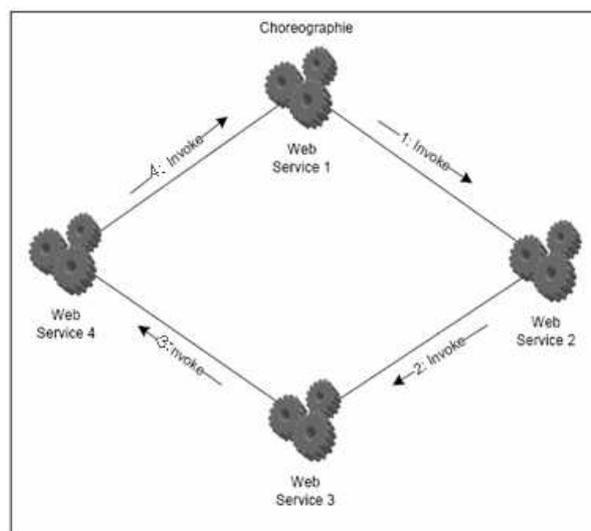


Abbildung 3.43: Prozess-Choreographie (Juric, et al., 2006)

Die nächsten Abschnitte stellen das Konzept der Orchestrierung vor und erläutern die Sprachspezifikation zum Beschreiben von Prozessen. Die wesentlichen Elemente der Geschäftsprozesssteuerungssprache BPEL sind nach (Hauser, Löwer, 2004 und Dostal, et al., 2005) die Folgenden:

- Partner (Teilnehmer an einem Geschäftsprozess),
- Messages (Nachrichten, die unter den Partnern ausgetauscht werden),
- Data Handling (Funktionen zum Aktualisieren und Austauschen von Zustandsvariablen der Prozesse),
- Activities (Einzelschritte, die einen Prozess bilden) und
- Scopes (Kontext um verknüpfte Aktivitäten).

Einzelne Schritte in einem BPEL-Prozess werden Aktivitäten genannt, die in Basisaktivitäten und strukturierte Aktivitäten unterschieden werden. Die wichtigsten Basisaktivitäten sind im Folgenden aufgezählt:

- Invoke (Aufruf eines Web Service),
- Receive (Blockierendes Warten auf Nachrichten),
- Reply (Generierung einer Antwort auf eine empfangene Nachricht),
- Assign (Kopieren von Daten von einem Platz zu einem anderen),
- Wait (Für eine bestimmte Dauer den Ablauf anhalten),
- Throw (Ausgeben von Fehlermeldungen),
- Empty (Keine Aktivität durchführen) und
- Terminate (Beenden der kompletten Prozessinstanz).

Strukturierte Aktivitäten können beliebig ineinander verschachtelt werden, so dass auch sehr komplexe Prozesse abgebildet werden können. Scopes bilden hierbei den Kontext für Aktivitäten und enthalten speziell den gemeinsamen Umgang mit zugrunde liegenden Daten und die Fehlerbehandlung durch Fault Handler und Compensation Handler. Zu den strukturierten BPEL-Aktivitäten zählen die folgenden Elemente (Hauser, Löwer, 2004 und Dostal, et al., 2005):

- Sequence (Reihenfolge von Schritten),
- Switch (Definition von Verzweigungen auf Basis von Fallunterscheidungen),
- While (Definition von Aktivitätsschleifen),
- Pick (Durchführen einer Aktivität nach einem erwarteten Ereignis),
- Flow (Definition parallel abzuarbeitender Prozessschritten) und
- Scope (Aktivitäten mit eigenem Scope).

Die einzelnen Schritte fügen sich wie folgt zu einer Gesamtstruktur eines WS-BPEL-Dokuments zusammen:



Abbildung 3.44: Struktur eines BPEL Dokumentes nach (Ferdinand, 2005)

Die in der abgebildeten Struktur eines BPEL-Dokumentes dargestellten Elemente sind in der folgenden Abbildung, die einen BPEL-Beispielcode zeigt, wieder zu finden.

#### Dokumentstart

```
- <process name="Einkaufsprozess" targetNamespace="http://eier.at/order"
  xmlns="https://schemas.xmlsoap.org/ws/2003/03/business-process/" xmlns:Ins="http://manufacturing.org/wsd/purchase">
```

#### Rollenverteilung

```
- <partnerLinks>
  <partnerLink name="einkaufen" partnerLinkType="Ins:einkaufenLT" myRole="Verkaeufer" />
  <partnerLink name="versenden" partnerLinkType="Ins:versendenLT" myRole="Versender" partnerRole="Empfaenger" />
</partnerLinks>
```

#### Variablen

```
- <variables>
  <variable name="Bestellung" messageType="Ins:EKMessage" />
  <variable name="Rechnung" messageType="Ins:REMessage" />
  <variable name="Einkauf-Fehler" messageType="Ins:BestellungFaultType" />
  <variable name="Lieferanfrage" messageType="Ins:versandRequestMessage" />
  <variable name="Lieferinfo" messageType="Ins:versandInfoMessage" />
  <variable name="Versandtermin" messageType="Ins:terminMessage" />
</variables>
```

#### Fehlerbehandlung

```
- <faultHandlers>
- <catch faultName="Ins:kannbestellungnichtkomplettieren" faultVariable="Einkauf-Fehler">
  <reply partnerLink="einkaufen" portType="Ins:BestellungPT" operation="versandBestellung" variable="Einkauf-Fehler"
    faultName="kannbestellungnichtkomplettieren" />
</catch>
</faultHandlers>
```

### Kontrollfluss

```

- <sequence>
  <receive partnerLink="einkaufen" portType="Ins:bestellungPT" operation="versandBestellung" variable="Bestellung" />
  <invoke partnerLink="versenden" portType="Ins:versendenPT" operation="anfrageVersand" inputVariable="Lieferanfrage"
    outputVariable="Lieferinfo" />
  <receive partnerLink="versenden" portType="Ins:versandCallbackPT" operation="sendeVersandtermin"
    variable="Versandtermin" />
  <receive partnerLink="einkaufen" portType="Ins:bestellungPT" operation="versandBestellung" variable="Rechnung" />
</sequence>
</process>

```

Abbildung 3.45: Definition eines BPEL-Dokumentes nach (Hansen, Neumann, 2005)

### 3.4.5 Ausblick von Web Services

An der Standardisierung von Web Services wird intensiv gearbeitet. Derzeit sind die Standards die bereits vorgestellten SOAP, WSDL, UDDI und BPEL. Es gibt bereits Vorschläge für erweiterte Kernspezifikationen oder zusätzliche Spezifikationen, die die bestehenden in vielen Bereichen ergänzen oder auch in einigen ersetzen könnten. Die wichtigsten Themenbereiche werden von (Hansen, Neumann, 2005) aufgeführt und erläutert. Es handelt sich um die Thematiken Sicherheit, Transaktionen, Dienstgüte und Verrechnung, die im Folgenden kurz dargestellt werden.

Sicherheitsaspekte haben gerade beim Nachrichtenverkehr über ein offenes Netzwerk wie das Internet eine immense Bedeutung. Die Web-Service-Security-Spezifikation bietet speziell für die Web Services ein Sicherheitsrahmenwerk, das durch ergänzende Felder im SOAP-Header ermöglicht, SOAP-Meldungen zu verschlüsseln oder sie mit digitalen Signaturen zu versehen. Auch gibt es bereits Ansätze, die den Schutz der Privatsphäre bei der Nutzung von Web Services vorsehen (Hansen, Neumann, 2005).

Der Bereich der Transaktionen von Geschäftsprozessen, die sich über Wochen oder auch Monate hinziehen können, ist im Handling komplizierter als einfache Transaktionen, bei denen meist das Sperren von Datensätzen in Datenbanken ausreicht. Ein alternatives Konzept bilden die Transaktionen mit Kompensationsprozessen, wobei bei einem Fehler ein Prozess mit kompensierenden Aktivitäten gestartet wird. Dieser macht die Auswirkungen der fehlgeschlagenen Transaktion rückgängig. Spezifikationen wie das Tentative Hold Protocol (THP) des W<sup>3</sup>C, das Business Transaction Protocol (BTP) von OASIS sowie Web Service Transaction und Web Service Business Activity von IBM in Zusammenarbeit mit weiteren Firmen der Industrie können in diesem Zusammenhang genannt werden (Hansen, Neumann, 2005).

Schlagwort für den Bereich der Dienstgüte ist das Service-Level-Agreement (SLA) zwischen Dienstanbieter und Dienstanutzer, bei dem sich der Dienstleister verpflichtet, den Dienst in einer vordefinierten Güte und in einem bestimmten Umfang zu liefern. Verfügbarkeit, Antwortzeit oder auch Telefonunterstützung sind beispielhafte Güteattribute eines Dienstes. Protokolle von bereits gelaufenen Operationen dienen hierbei als Basis für Analysen, um die Dienstgüte messbar zu machen. Eine ebenso große Bedeutung hat die innere Optimierung von Abläufen. Als erster Ansatz für diese Thematik ermöglicht die Business Process Query Language (BPQL) die dynamische Auswertung der Effizienz. Es liegen jedoch noch keine Vorschläge zur konkreten Gestaltung vor (Hansen, Neumann, 2005).

Die Verrechnung spielt bei kommerziellen Web Services eine entscheidende Rolle. Für die unterschiedlichen Verrechnungsvarianten, wie beispielsweise Bezahlung nach Benutzungshäufigkeit oder Pauschalgebühr, liegen bisher noch keine standardisierten Ansätze für Web Services vor (Hansen, Neumann, 2005).

Für die weitere Betrachtung ist eine Erweiterung der Kernspezifikation BPEL mit zu berücksichtigen. Da BPEL ausschließlich für die Kommunikation zwischen Maschinen entwickelt worden ist, aber die Benutzerinteraktion, die nahezu für jeden Prozess als Initiator gilt, notwendig ist, wurde BPEL4people als Initiative von den Firmen SAP und IBM ins Leben gerufen. BPEL4people soll als Erweiterungsschicht zur Kernsprache WS-BPEL standardisierte Konzepte zur Benutzerinteraktion zur Verfügung stellen und liegt seit 2005 als White Paper bei OASIS zur Standardisierung vor. Oracle hat in seinem BPEL Process Manager bereits die ersten Ansätze als „Human task“ von BPEL4people integriert.

## 3.5 Geo Web Services

In diesem Abschnitt wird der Begriff Geo Web Services eingeführt, der die technologischen Aspekte der zuvor beschriebenen Web-Services um raumbezogene Elemente erweitert. Dabei geht es darum, Geoinformationen, die am Ort ihrer Erfassung und Bearbeitung vorgehalten werden, einer breiten Anwendergemeinschaft in Form von Dienstleistungen zur Verfügung zu stellen.

Rechnerlesbare Geoinformationen sind unabhängig von ihrer fachlichen Aussage wie Planungszweck, Bodennutzung oder Umweltqualität und verfügen über einen Raumbezug. Der Raumbezug solcher Daten kann entweder direkt über die Angabe von Koordinaten oder indirekt beispielsweise über eine Adressangabe oder eine Straßenkilometrierung festgelegt werden. Mithilfe von Geoinformationssystemen und speziell ausgerichteten Geodatenbanken werden die Geoinformationen erfasst, verwaltet und analysiert. Dadurch ist im Gegensatz zu einer früheren Führung der Geoinformation in Form von Amtsblättern und Tabellen eine wesentlich effizientere Integration dieser Daten in Verwaltungs- und Unternehmensabläufe möglich. Auf der Grundlage von Geodaten entstehen also umfangreiche Analyse- und Visualisierungsmöglichkeiten, die sich durch thematisch ausgerichtete Sachkarten ausdrücken lassen. Die rasante Entwicklung des Internets verbunden mit neuen Technologien seit den 1990er Jahren hat sich zweifelsfrei auch auf das Geoinformationswesen übertragen. Dadurch können Geodaten, die getrennt verwaltet werden, unabhängig von ihrem Speicherort schnell und unkompliziert miteinander kombiniert werden. Sie müssen nicht mehr bilateral ausgetauscht und mehrfach gespeichert werden, sondern stehen dauerhaft und jederzeit dem Fachanwender oder auch dem Bürger „online“ zur Verfügung. Insellösungen weichen damit einer vernetzten Umgebung von raumbezogenen Informationen. In diesem Zusammenhang spricht man von einer Geodateninfrastruktur (GDI), die sowohl auf kommunaler Ebene, als auch auf Bundesland-, bundesweiter und europäischer Ebene thematisiert und aufgebaut wird. Beispiele für den Aufbau solcher GDI sind die Projekte Geobasisportal NRW auf Bundeslandebene (NRW, 2007) oder INSPIRE auf Europaebene (EC-GIS, 2007). Die grundsätzlichen Bestandteile einer GDI sind die Geodaten, die Transportnetze, Geodienste und Web- bzw. Geostandards (IMAGI, 2006). Die folgenden Web-basierten Geodienste spielen speziell beim Aufbau einer GDI und allgemein bei der Verteilung von Geoinformationen eine wichtige Rolle.

- **Web Map Service (WMS)**

Der WMS dient der Übertragung und Visualisierung von Geodaten durch deren Präsentation in einem Rasterbildformat wie PNG, JPG oder GIF.

- **Web Feature Service (WFS)**

Der WFS ist ähnlich dem WMS, nur das hierbei Vektordaten übertragen und dargestellt werden können. Es besteht also ein direkter Onlinezugriff auf die zugrunde liegenden Daten.

- **Web Coverage Service (WCS)**

Ein Coverage ist ein Rasterlayer, in dem einzelne Punkte mit Attributen versehen werden können, die Eigenschaften von geographischen Standorten beschreiben. Der WCS liefert Vektordaten und greift auf gerasterte Datenbestände zu.

- **Web Gazetteer Service (WFS-G)**

Der WFS-G stellt den Zugang zu raumbezogenen Daten über geographische Namensverzeichnisse her. Zu einem geographischen Namen werden die Koordinaten oder ein Kartenausschnitt geliefert.

- **Web Terrain Service (WTS)**

Durch einen WTS werden Geodaten 3-dimensional dargestellt auf der Grundlage von digitalen Geländemodellen.

- **Web Coordinate Transformation Service (WCTS)**

Auf der Grundlage von Transformationsparametern werden unterschiedliche Datenquellen zur Laufzeit der Anfrage in ein gemeinsames Referenzsystem transformiert.

- **Web Catalog Service (CSW)**

Der CWS dient der elektronischen Veröffentlichung und der Abfrage von Metainformationen über Geodienste und Geodaten.

Im Folgenden werden die beiden Geo Web Services WMS und WFS vorgestellt, die für die Erstellung und Verbreitung von Karten auf Grundlage der Internettechnologie von elementarer Bedeutung sind.

### 3.5.1 Web Map Server (WMS)

Bevor näher auf die Funktionsweise des WMS und später des WFS eingegangen wird, werden die Grundlagen zur Verteilung von Geodaten vorgestellt. Einfach ausgedrückt geht es darum, dass ein Nutzer A am Ort A auf die Geodaten des Rechners des Nutzers B an Ort B zugreift. Als Transportmedium zwischen Ort A und Ort B steht das Internet zur Verfügung. Der Nutzer A benötigt zur Erstellung der Anfrage einen Client, der im Allgemeinen durch einen Web Browser dargestellt wird, mit dem eine Anfrage gestellt wird. Die Anfrage beinhaltet das Gebiet, von dem Geodaten bezogen werden sollen, und den fachlichen Inhalt der Geodaten. Zudem könnte beispielsweise noch das geforderte Format, in dem die Karte erstellt werden soll, angegeben werden. Diese Anfrage wird an den Rechner des Nutzers B weitergeleitet. Dieser verarbeitet die Anfrage, indem er in Verzeichnissen oder Katalogen nach den angefragten Daten sucht. Eventuell werden für die Zusammenstellung des Ergebnisses weitere Server angebunden. Ist die Datenquelle ausfindig gemacht, navigiert der Server im Datenbestand und bereitet die zur Anfrage passenden Daten auf und schickt sie an den Client zurück. Die aufbereiteten Daten können in Form von Rasterbildern oder auch Vektordaten übermittelt werden. Die folgende Abbildung macht das Prinzip der Kartenerstellung deutlich.

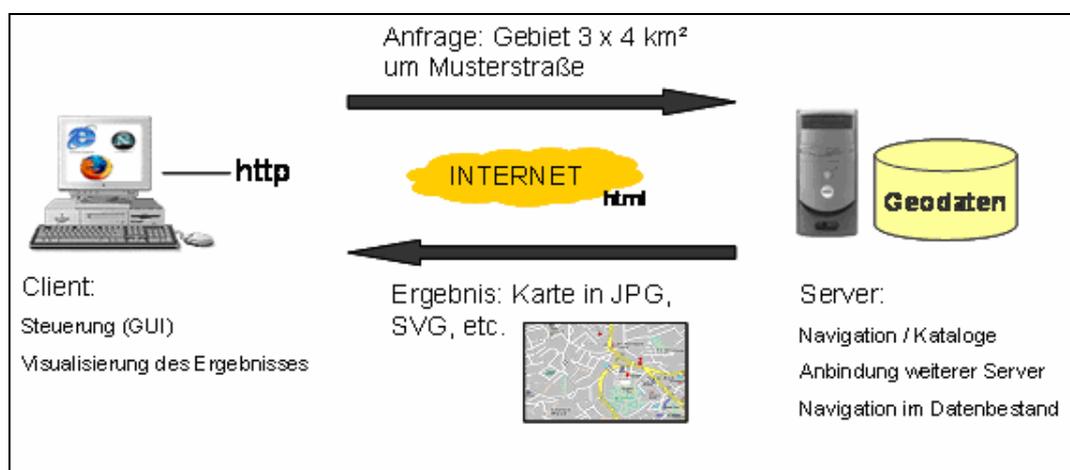


Abbildung 3.46: Prinzip der Kartenerstellung im Internet

Die Web Map Service Spezifikation wurde in der Version 1.0 bereits im Jahr 2000 veröffentlicht. Diese wurde weiter entwickelt und führte 2002 zur Version 1.1.1, die derzeit in der Praxis am meisten implementiert ist. Nach Angaben des OGC wurde die Version WMS 1.1.1 zurzeit in 163 Produkten implementiert (Stand: Februar 2007). Die im August 2004 erschienene Version 1.3 wurde zum selben Zeitpunkt erst in 5 Produkten implementiert (OGC, 2007).

Die WMS Spezifikation definiert die Syntax der Anfrage nach einem Kartenbild sowie das Format und die Eigenschaften des Ergebnisses. Dazu sind die drei Operationen

- GetCapabilities
- GetMap und
- GetFeatureInfo

definiert. Die Implementierung der ersten beiden Operationen ist verpflichtend, während die Operation GetFeatureInfo optional ist. Sind die ersten beiden Operationen implementiert, spricht man von einem Static, bzw. Basic WMS. Ist die dritte Operation zusätzlich implementiert, spricht man von einem Dynamic, bzw. Queryable WMS. Im Folgenden werden die drei Operationen näher betrachtet.

### GetCapabilities

Durch die Operation GetCapabilities fragt der Client nach dem Leistungsumfang des Dienstes. Hier wird nach den Eigenschaften und den Dateninhalten gefragt. Die Antwort erfolgt durch ein XML-Dokument, welches in festgelegter Form die spezifischen Metadaten zu den angebotenen Geodaten beinhaltet. Die Metadaten enthalten Angaben über den Dienst, mögliche Ausgabeformate, verfügbare Datenlayer und über das Koordinatensystem.

Die Anfrage eines Clients an einen Server kann mit einem gewöhnlichen Web Browser erfolgen. Dazu werden die Anfrageparameter in die Adressleiste des Browsers eingegeben und über http übertragen. Es sind daher keine besonderen Anforderungen an den Client vorhanden. Eine WMS-Anfrage hat den folgenden allgemeinen Aufbau.

```
http://host[:port]/path?{name[=value]&}
```

Die Angaben in den eckigen und den geschweiften Klammern sind optionale Bestandteile, wobei der Inhalt der geschweiften Klammern mehrfach vorkommen kann. Der WMS verwendet in der Anfragesyntax sechs der Zeichen, die im Request For Comment 2396 (RFC) durch die Internet Engineering Task Force (IETF) für den allgemeinen Aufbau einer URI festgelegt worden sind. Die folgende Tabelle zeigt diese sechs Zeichen und ihre Bedeutung.

Zeichen	Bedeutung
?	Zeichen für den Beginn der Anfrage
&	Zeichen zwischen den Parametern einer Anfrage
=	Zeichen zwischen Namen und Wert eines Parameters

/	Zeichen zwischen MIME type und Subtype im Format-Parameter
:	Zeichen zwischen Namespace und Bezeichnung im SRS-Parameter
,	Zeichen zwischen Werten einer Parameterliste

Tabelle 3.7: Zeichen einer WMS-Anfrage nach (OGC, 2002)

Zudem sind spezielle Schlüsselbegriffe (Parameter) für eine Anfrage festgelegt. Die folgende Tabelle zeigt diese Parameter für die Operation GetCapabilities.

Parameter	Opt./nicht Opt.	Beschreibung
VERSION	optional	WMS-Version
SERVICE	nicht optional	Typ des Services
REQUEST	nicht optional	Typ der Anfrage
UPDATESEQUENCE	optional	Sequenznummer

Tabelle 3.8: Parameter der Operation GetCapabilities nach (OGC, 2002)

Das folgende Beispiel zeigt eine GetCapabilities Anfrage.

```
http://localhost:8888/mapviewer/wms?REQUEST=GetCapabilities&SERVICE=WMS&VERSION=1.1.1
```

Die Antwort auf diese Anfrage mit den angegebenen Parametern ist ein XML-Dokument, welches folgendermaßen aufgebaut sein könnte.

```
<?xml version="1.0" encoding="UTF-8" ?>
<DOCTYPE WMT_MS_Capabilities [View Source for full doctype...]>
<WMT_MS_Capabilities version="1.1.1">
  <Service>
    <Name>OGC:WMS</Name>
    <Title>WMS 1.1.1 for Oracle Application Server 10g MapViewer</Title>
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="http://Server:80/mapviewer/wms" />
    <Fees>none</Fees>
    <AccessConstraints>none</AccessConstraints>
  </Service>
  <Capability>
    <Request>
      <GetCapabilities>
        <GetMap>
          <Format>image/png</Format>
          <Format>image/gif</Format>
          <Format>image/jpeg</Format>
          <Format>image/svg+xml</Format>
          <Format>image/png8</Format>
          <DCPType>
            <GetMap>
              <GetFeatureInfo>
                <Format>text/xml</Format>
              </GetFeatureInfo>
            </DCPType>
          </DCPType>
        </Request>
        <Exception>
          <LayerQueryable="0" opaque="0" noSubsets="0">
            <Name>WMS</Name>
            <Title>Oracle WebMapServer-Layer nach Datenquelle.</Title>
            <SRS>EPSG:4326</SRS>
            <LatLonBoundingBox minx="180.0" miny="-90.0" maxx="180.0" maxy="90.0" />
            <LayerQueryable="0" opaque="0" noSubsets="0">
              <Name>wms</Name>
              <Title>Datasource wms</Title>
              <LayerQueryable="0" opaque="0" noSubsets="0">
                <Name>WMS_BASEMAP</Name>
                <Title>Basemap WMS_BASEMAP</Title>
                <SRS>SDD:82027</SRS>
                <LatLonBoundingBox minx="8.30794121294173" miny="49.7246542744933" maxx="8.86191986941254" maxy="49.996362597612" />
                <LayerQueryable="1" opaque="0" noSubsets="0">
                  <Name>GEBALUDE</Name>
                  <Title>GEBALUDE</Title>
                  <SRS>SDD:82027</SRS>
                  <BoundingBox SRS="SDD:82027" minx="3450000.0" miny="5510000.0" maxx="3490000.0" maxy="5540000.0" resx="50000.0" resy="50000.0" />
                  <ScaleHint min="1000000" max="0" />
                </Layer>
              </Layer>
            </LayerQueryable>
          </Layer>
        </Exception>
      </GetCapabilities>
    </Request>
  </Capability>
</WMT_MS_Capabilities>
```

Abbildung 3.47: XML-Antwort auf eine GetCapabilities Anfrage

Aus diesem Beispiel sind die Metadaten ersichtlich, welche die GetCapabilities Anfrage an den Client zurückgibt. Es werden die möglichen Ausgabeformate (PNG, GIF, JPEG, SVG+XML, PNG8), die zur Verfügung stehenden Layer (Gebäude und Flurstuecke), das Koordinatensystem (EPSG:4326) sowie die räumliche Abgrenzung des Gebietes beschrieben. Aus diesen Angaben kann die Operation GetMap abgeleitet werden, die im Folgenden beschrieben wird.

### GetMap

Durch die Anfrage GetMap wird eine georeferenzierte Karte vom Nutzer angefordert. Durch verschiedene Parameter werden in der Anfrage beispielsweise das Ausgabeformat, die Kartenlayer, die Kartengröße oder das Koordinatensystem festgelegt. Die folgende Tabelle zeigt die verschiedenen Parameter der Operation GetMap.

Parameter	Opt./nicht Opt.	Beschreibung
VERSION	nicht optional	WMS-Version
REQUEST	nicht optional	Typ der Anfrage
LAYERS	nicht optional	darzustellende Kartenlayer
STYLES	nicht optional	Liste der Styles
SRS	nicht optional	Koordinatensystem
BBOX	nicht optional	Kleinster Umring um das Gebiet
WIDTH	nicht optional	Kartenbreite in Pixel
HEIGHT	nicht optional	Kartenhöhe in Pixel
FORMAT	nicht optional	Ausgabeformat der Karte
TRANSPARENT	optional	Angabe zur Transparenz
BCOLOR	optional	Hintergrundfarbe

Tabelle 3.9: Parameter der Operation GetMap nach (OGC, 2002)

Der Parameter SRS, der für die Angabe des Koordinatensystems steht, wird als Zahlencode des EPSG angegeben. Innerhalb einer Oracle-Umgebung kann dieser Parameter auch mit SDO und einem zugehörigen Schlüssel angegeben werden. Beispielsweise steht SDO:82027 für das Koordinatensystem Gauß-Krüger der Zone 3, welches bei EPSG unter EPSG:31467 geführt wird. Mit SDO wird innerhalb der Geometriespeicherung direkt das Koordinatensystem der zugrunde liegenden Objekte angesprochen.

Das folgende Beispiel zeigt eine Kartenanfrage GetMap und das Ergebnis dieser Anfrage in einem Web Browser.

```
http://localhost:8888/mapviewer/wms?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&BBOX=3464666,5516491,3483169,5535899&SRS=EPSG:31467&HEIGHT=700&WIDTH=700&FORMAT=image/jpeg&LAYERS=Flurstuecke&EXCEPTIONS=application/vnd.ogc.se_inimage
```

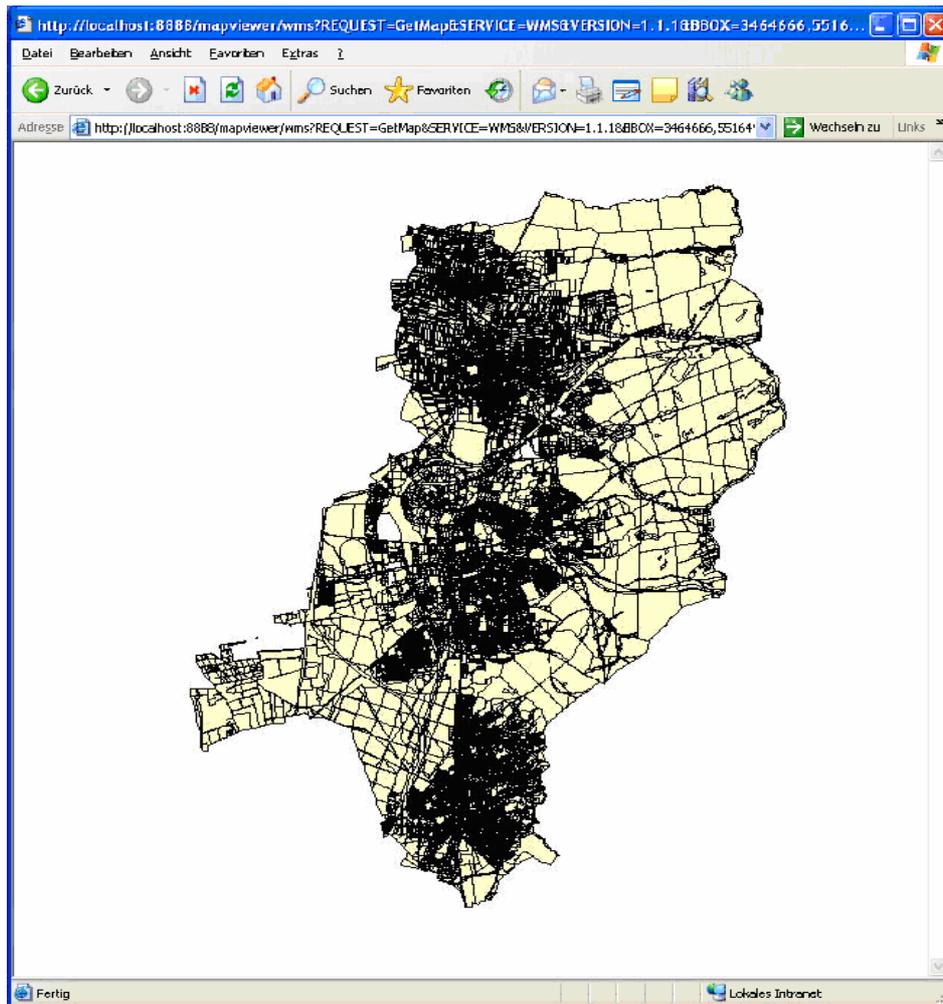


Abbildung 3.48: Ergebnis einer GetMap Anfrage

Das Ergebnis ist eine Rastergrafik im JPEG-Format, die von einem Web Browser ohne zusätzliche Anforderungen dargestellt werden kann.

### GetFeatureInfo

Durch die optionale Eigenschaft GetFeatureInfo können zuvor festgelegte Informationen zu einzelnen Objekten abgefragt werden. Auch für diese Operation sind Parameter definiert, von denen manchen optional sind. Die folgende Tabelle zeigt die Parameter einer GetFeatureInfo Anfrage.

Parameter	Opt./nicht Opt.	Beschreibung
VERSION	nicht optional	WMS-Version
REQUEST	nicht optional	Typ der Anfrage
QUERY_LAYERS	nicht optional	darzustellende Kartenlayer
INFO_FORMAT	optional	Liste der Styles
FEATURE_COUNT	optional	Koordinatensystem
X	nicht optional	Kleinster Umring um das Gebiet
Y	nicht optional	Kartenbreite in Pixel

Tabelle 3.10: Parameter der Operation GetFeatureInfo nach (OGC, 2002)

Der Aufbau einer GetFeatureInfo Anfrage entspricht etwa dem der GetMap Anfrage, nur dass die zusätzlichen Parameter QUERY\_LAYERS, INFO\_FORMAT, X und Y verwendet werden. Diese Anfrage zielt darauf ab, zu

einem vorgegebenen Punkt auf der Karte die Informationen zu den Objekten zu bekommen, die unter diesem Punkt zu finden sind. Dazu wird festgelegt, welche Kartenlayer abgefragt werden sollen (QUERY\_LAYER), in welchem Format das Ergebnis dargestellt werden soll (INFO\_FORMAT) und welcher Punkt abgefragt werden soll (X und Y). Die Koordinaten werden hierbei in Bildpunkten der Karte angegeben. Das folgende Beispiel zeigt eine GetFeatureInfo Anfrage. Es sollen die Informationen zu dem Gebäude im XML-Format ausgegeben werden, das unter dem Punkt 410/280 zu finden ist. Das darauf folgende Ergebnis ist eine XML-Datei, die in einem Web Browser dargestellt werden kann.

```
http://localhost:8888/mapviewer/wms?REQUEST=GetFeatureInfo&VERSION=1.1
.1&BBOX=3464666,5516491,3483169,5535899&SRS=EPSG:4326&HEIGHT=700&WIDTH
=700&INFO_FORMAT=text/xml&QUERY_LAYERS=GEBAEUDE&X=410&Y=280&EXCEPTIONS
=application/vnd.ogc.se_xml
```

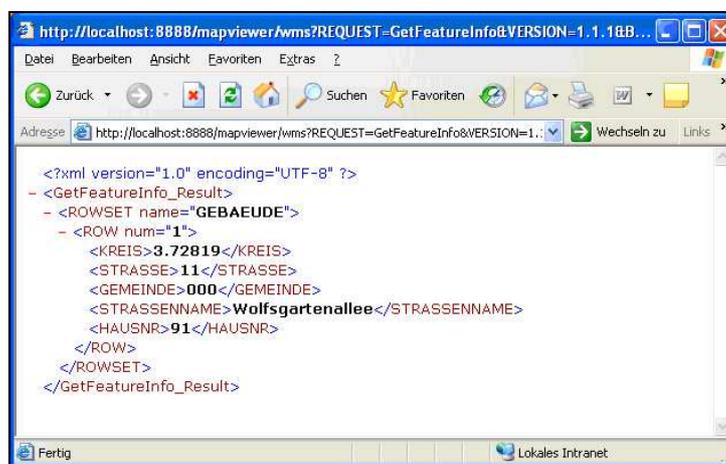


Abbildung 3.49: Ergebnis einer GetFeatureInfo Anfrage

Durch den WMS können verschiedene Karten mit unterschiedlicher fachlicher Ausrichtung miteinander kombiniert werden. Diese Karten können auf verschiedenen Servern weltweit vorgehalten werden und haben alle denselben Raumbezug. Die Vorteile dieser Technologie liegen auf der Hand. Ein Client kann sich ohne teure Softwarelizenzen zu besitzen, zur Laufzeit die Informationen zusammenstellen, die er für seine momentane Aufgabe benötigt. Ein solcher Client kann im einfachsten Fall ein Web Browser sein, wobei der Nutzer sich hierbei die URL der jeweiligen Anfragen selbst zusammenstellen muss. Ein Client kann auch eine vorhandene GIS-Software sein, da die meisten Hersteller bereits eine WMS Schnittstelle in ihre Produkte implementiert haben. Der Nutzer stellt sich in einfachen Oberflächen seine Informationen zusammen und die Software generiert daraus die notwendige Syntax für die WMS Anfragen. Zudem gibt es frei verfügbare Web Clients, mit denen es möglich ist, von jedem Ort der Erde beliebige WMS Server anzubinden und Informationen zusammen zu führen. Ein Beispiel dafür ist der WMS Viewer der Firma Intergraph, der unter der Internetadresse <http://www.wmsviewer.com> zu finden ist. Der Quelltext dieser Anwendung ist offen gelegt und wird zur Verfügung gestellt, so dass sich jeder Nutzer die Anwendung lokal installieren und anpassen kann.

### 3.5.2 Web Feature Server (WFS)

Bei allen Vorteilen, die ein WMS bietet, sind die Funktionalitäten in der Praxis oftmals nicht ausreichend. In dem Moment, in dem die Visualisierung der Geodaten durch ein Bild als Ausgabeformat nicht mehr genügt, stößt der WMS an seine Grenzen. Der WMS kann nicht auf die zugrunde liegenden Daten selbst zugreifen und die Analyse und Interpretation der Geodaten obliegt dem Betrachter. An dieser Stelle setzt der Web Feature Service (WFS) an, der ausschließlich Vektordaten an den Client weiterleitet. Diese Daten kann der Client dann visualisieren, analysieren oder in anderer Form weiterverarbeiten.

Die Version 1.0 der WFS-Spezifikation wurde vom OGC 2002 verabschiedet und bisher in 90 Produkte implementiert (Stand Februar 2007). Die Folgeversion 1.1 liegt seit Mai 2005 vor und wurde bislang in noch keinem Produkt implementiert. Zwischen diesen beiden Versionen liegen einige signifikante Unterschiede vor, die im weiteren Verlauf an den notwendigen Stellen erläutert werden. Die folgende Abbildung illustriert zunächst das Funktionsprinzip eines WFS.

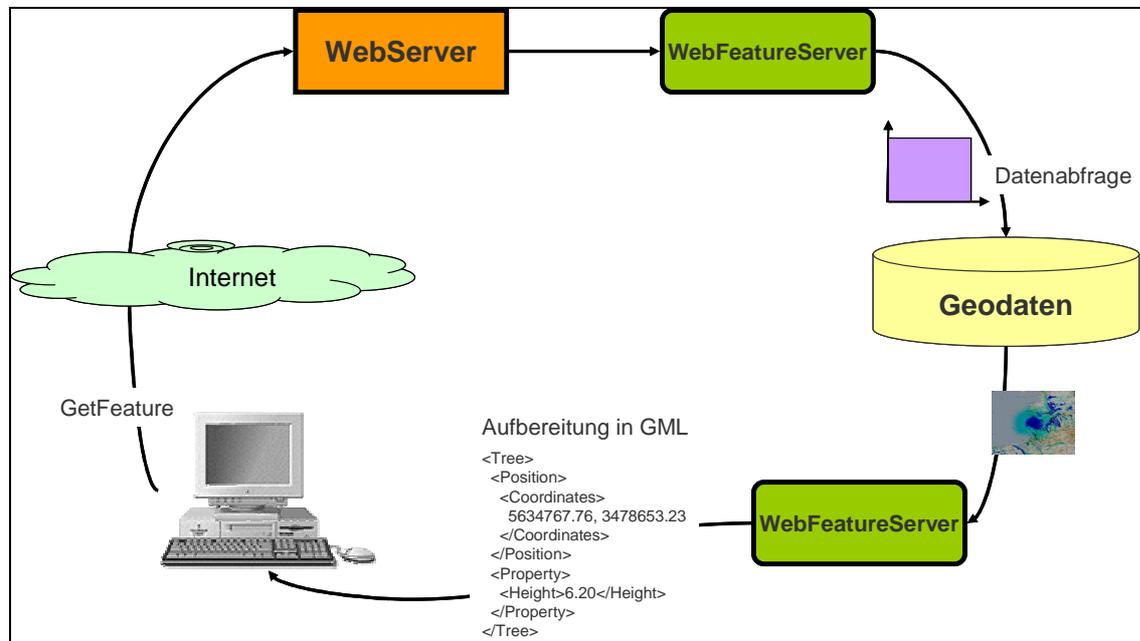


Abbildung 3.50: Funktionsweise des WFS

Ein Client stellt eine Anfrage über das Internet und einen Web Server an den Web Feature Server. Diese Anfrage nach Vektordaten wird mit der Operation GetFeature gestellt und kann in Form einer URL oder als XML-Dokument erfolgen. Mit der Version 1.1 kann erstmals SOAP zur Nachrichtenübermittlung eingesetzt werden. Der WFS bereitet diese Anfrage auf und ermittelt die notwendigen Daten. Die Datenhaltung antwortet mit den gesuchten Daten und leitet sie an den WFS zurück. Dieser bereitet die Daten in GML auf und schickt sie an den Client zurück. Es wird deutlich, dass es dem Client obliegt, in welcher Form er die Daten visualisiert bekommen möchte, da der WFS in jedem Fall ein GML-Dokument ausgibt. Entweder ist der Client mit einer Software ausgestattet, die GML in der Art interpretieren kann, dass sie daraus eine Karte erzeugt, oder es ist ein MapServer vorgeschaltet, der aus GML eine Karte erzeugt. Diese Karte kann sowohl in einem Rasterformat aufbereitet werden oder auch in einem Vektorformat wie beispielsweise Scalable Vector Graphics (SVG). Die Version 1.0 der WFS-Spezifikation basiert noch auf der GML-Version 2.1, während die Version 1.1 auf GML Version 3 basiert.

Der WFS-Standard 1.0 definiert fünf Operationen, mit denen raumbezogene Objekte eingefügt, aktualisiert, gelöscht oder abgefragt werden können. Ähnlich dem WMS gibt es auch beim WFS verpflichtende und optionale Operationen. Werden die drei Standardoperationen GetCapabilities, DescribeFeatureType und GetFeature implementiert, spricht man von einem Basic WFS. Dieser bietet dem Anwender einen lesenden Zugriff auf die Geodaten. Wird zusätzlich die Operation Transaction implementiert, spricht man von einem Transactional WFS. Dadurch kann der Anwender schreibend auf die Geodaten zugreifen. Zudem können die Operationen LockFeature, bzw. GetFeatureWithLock verwendet werden, um einzelne Objektinstanzen mit Veränderungssperren auszustatten. Mit der Version 1.1 der WFS-Spezifikation ist die Operation GetGMLObjekt hinzugekommen, wodurch ein WFS Elemente zurückgeben kann, die durch die XML-Technologie XLinks beschrieben worden sind. Ist diese Operation neben den Basisoperationen implementiert, spricht man von einem XLink-WFS.

## 4 Ansätze zur Nutzung des Raumbezugs

Dieses Kapitel ist in einen technischen und einen fachlichen Teil gegliedert. Im technischen Bereich, welcher zu Beginn des gesamten Kapitels aufgeführt ist, werden zunächst die Motivation zur Nutzung des Raumbezugs erläutert und allgemeine Voraussetzungen für die Einbindung von GIS in Geschäftsprozesse aufgezeigt. Im Anschluss daran diskutiert Kapitel 0 Möglichkeiten zur technischen Umsetzung und bewertet die einzelnen Varianten. Es folgt der fachliche Teil, welcher mit der Untersuchung des Referenzmodells des Deutschen Vereins des Gas- und Wasserfaches e.V. (DVGW) für die Einbindung von GIS in Geschäftsprozesse am Beispiel Störfallmanagement einen Prozess exemplarisch herausstellt. Kapitel 4.5 listet vorhandene Lösungen auf und stellt sie hinsichtlich der Zielsetzung dieser Arbeit zur Diskussion. Abschließend werden die technischen und fachlichen Lösungsansätze bewertet und der Realisierungsbedarf konkretisiert.

### 4.1 Motivation zur Nutzung des Raumbezugs

Die Motivation der Energieversorgungsunternehmen zur digitalen Dokumentation ihrer Betriebsmittel durch den Einsatz geografischer Informationssysteme ist laut (Stockwald, 2000) sowohl wirtschaftlich als auch operationell und strategisch begründet. Anders jedoch sieht es mit der Verwendung des Raumbezuges im Zusammenhang mit einer Software-Integration aus. Zur Diskussion steht der tatsächliche Nutzen, der durch eine GIS-Integration entsteht. Es existieren laut (Schrenner, 2005) und (Behr, 2002) drei wesentliche Ziele einer Integration mit einem Geographischen Informationssystem:

- Konsistente Datenbestände,
- Datenmehrwert und
- Durchgängige Arbeitsabläufe.

Jedem der drei Ziele können nach (Behr, 2002) diese vier Nutzungskategorien zugeordnet werden:

- Nutzen durch erhöhte Produktivität,
- operationeller Nutzen,
- strategischer Nutzen und
- externer Nutzen.

Im Folgenden werden die Ziele der Integration hinsichtlich der Kategorien auf ihre Nutzungspotentiale untersucht.

Teilweise werden die gleichen Realweltverhältnisse von mehreren Systemen, beispielsweise von GIS und ERP-Teilsystemen, abgebildet. Ziel einer Integration ist es somit, eine redundante Datenhaltung so weit wie möglich zu reduzieren und die Datenbestände automatisiert inhaltlich korrekt zu halten. Die anschließende tabellarische Auflistung führt die Nutzungspotentiale detailliert auf.

Kategorie	Nutzungspotential
Nutzen durch erhöhte Produktivität	Vermeidung von Doppeleingaben
operationeller Nutzen	Erhöhung der Datenqualität hinsichtlich Vollständigkeit, Lage- und Attributgenauigkeit
	Verbesserung der Datenaktualität
strategischer Nutzen	Vereinigung und Vereinheitlichung von Datenbeständen und Software-Systemen
	Auskunftssicherheit gegenüber Externen

Tabelle 4.1: Nutzenbetrachtung durch Datenkonsistenz nach (Schrenner, 2005)

Ausschließlich durch den konsistenten Datenbestand wird kein Datenmehrwert erzeugt. Durch die Integration wird es ermöglicht, rein alphanumerische Daten in einen geographischen Bezug zu setzen. Ebenso können die geographischen Daten mit zusätzlichen alphanumerischen Daten bereichert werden. So entsteht eine interdisziplinäre Sicht auf Sachfragen. Das folgende Beispiel belegt dies anhand von Fragestellungen im Störungsmanagement eines Energieversorgungsunternehmens (Schrenner, 2005).

**Fragestellung ERP:** Welche Anlagen wurden instand gesetzt? Weshalb waren Instandsetzungen erforderlich? Welche Ursachen und Schadensbilder traten auf?

**Erweiterung durch GIS:** Gibt es räumliche Anhäufungen? Sind Leitungen eines bestimmten Materials besonders anfällig? Bestehen Zusammenhänge mit topographischen Begebenheiten (Vegetation bei Wurzeinwuchs)?

Die Nutzenbetrachtung durch den Datenmehrwert hinsichtlich der bereits beschriebenen Kategorien folgt in der anschließenden Tabelle.

Kategorie	Nutzungspotential
operationeller Nutzen	Interdisziplinäre Sicht auf Daten
	Schaffung neuer Entscheidungsgrundlagen
	Erhöhung der Qualität von Managemententscheidungen
strategischer Nutzen	Eröffnung neuer Geschäftsfelder, z.B. durch Geodatenverarbeitung

Tabelle 4.2: Nutzenbetrachtung durch Datenmehrwert nach (Schrenner, 2005)

Das dritte Ziel der Integration ist die durchgängige Abarbeitung von Arbeitsaufgaben, welche bisher in mehreren Systemen geleistet und aufeinander abgestimmt wurden. Eine durchgängige Bearbeitung von Geschäftsprozessen ist das wohl größte Nutzungspotential der Integration von Softwaresystemen. Insbesondere in den Prozessen, welche durch viele verschiedene Zuständigkeiten und viele Medienbrüche gekennzeichnet sind, erschließt die

Integration erhebliche Wertschöpfungspotentiale hinsichtlich Schnelligkeit, Zuverlässigkeit, Qualitätssicherung und gerichts-feste Dokumentation. Nach (Behr, 2002 und Schrenner, 2005) sind folgende Potentiale erkennbar.

<b>Kategorie</b>	<b>Nutzungspotential</b>
Nutzen durch erhöhte Produktivität	Reduzierung des Arbeitsumfangs
	Beschleunigung von Arbeitsabläufen
operationeller Nutzen	Rechtzeitige Bereitstellung von Informationen
	Verringerung von Medienbrüchen
	Bedienerfreundlichkeit
	Dokumentation von Abläufen und Bearbeitungsständen
strategischer Nutzen	Optimierung von Geschäftsprozessen
	Schaffung qualifizierterer Arbeitsplätze
	Festigung der Kundenbindung (durch Qualität, Lieferbereitschaft, Beratung)
Externer Nutzen	z.B. durch Beschleunigung der Bearbeitung der Kundenanträge

Tabelle 4.3: Nutzenbetrachtung bei durchgängigen Workflows nach (Schrenner, 2005)

Die andiskutierten drei Ziele samt ihren Nutzungspotentialen lassen sich unterschiedlich bewerten. Die monetäre Erfassung der Wertschöpfung lässt sich oft nicht konkret darstellen.

Die Datenkonsistenz schafft direkten Nutzen durch die Vermeidung der doppelten Dateneingabe. Ebenso ist sie die Basis für die weiteren Potentiale. Der Datenmehrwert bietet nur dann auch einen solchen, wenn neue Datensichten entstehen, welche zu anderen, kosteneinsparenden Managemententscheidungen führen. Die Prozessoptimierung ist jedoch der Bereich, in welchem durch exakte Planung erhebliche Einsparpotentiale zu finden sind. Da die drei Ziele ineinandergreifen bzw. aufeinander aufbauen, kann die Bewertung des Wertschöpfungspotentials nur als Gesamtheit betrachtet und nicht auf einzelne Zielstellungen bezogen werden.

## **4.2 Voraussetzungen zur Einbindung von GIS in Geschäftsprozesse**

Alle Unternehmen, die prozessorientiert handeln, benötigen prozessorientierte Werkzeuge. Wesentliche Voraussetzung dafür ist die Integrationsfähigkeit der Einzelkomponenten hinsichtlich Daten und Funktionalitäten zu einem informationsverarbeitenden Gesamtsystem. Im Folgenden werden die Rahmenbedingungen diskutiert, die ein Geographisches Informationssystem samt seinen Daten erfüllen sollte, um einer Integration zu genügen.

### **4.2.1 Daten und Datenstrukturen**

Die jeweiligen Nutzungsanforderungen bestimmen die Detaillierungstiefe und den geographischen Bezug der erfassten Bestands-, Zustands- und Vorgangsdaten. Homogenität und Konsistenz zwischen geographischen und alphanumerischen Daten gelten auch über die Grenzen des Einzelsystems hinaus. (DVGW, 2002) führt in die-

sem Zusammenhang die Adresse auf, die meist inhomogen in verschiedenen Einzelsystemen geführt wird. Die nachfolgende Auflistung zeigt verschiedene Verwendungsmöglichkeiten einer Adresse.

- Straßename in Plänen (GIS),
- Anlagebezeichnung (Instandhaltungssoftware),
- Bezeichnung Abnahmestelle/Zählerstandort (Zählerwesen) und
- Kundenanschrift (Kundendatenbank).

Daraus folgt, dass Datenmodelle und -Strukturen offen zugänglich, lesbar und im Rahmen fester Regeln auch erweiterbar sein müssen. Ein erster Schritt in die richtige Richtung sind beispielsweise die GAWANIS-Datenmodelle des DVGW. Sie beschreiben die Strukturen und Dateninhalte, die für den Aufbau und Betrieb eines Netzinformationssystems erforderlich sind. Sie sind systemunabhängig und bleiben somit offen für neue oder veränderte Anwendungen. Fertig gestellt und verfügbar sind zum Zeitpunkt der Arbeit folgende Datenmodelle, auf die der DVGW eine Überprüfung von angebotenen Systemlösungen auf GAWANIS-Konformität anbietet:

- Gas- und Wasserrohrnetz und
- Kathodischer Korrosionsschutz.

Des Weiteren müssen eventuell vorhandene Schnittmengen sowie die Abgrenzung zu anderen Systemen bekannt sein. Ist eine redundanzfreie Datenhaltung nicht möglich, so ist eine kontrollierte Redundanz anzustreben, die beispielsweise einmal erfasste Daten in einem weiteren System zur Verfügung stellt, ohne sie dort zur Bearbeitung freizugeben. Die Erfassung und Fortführung von Daten darf jedoch nur in einem System erfolgen.

Bei Versorgungsunternehmen lassen sich prozessrelevante Geodaten laut (DVGW, 2003) in drei Klassen zusammenfassen:

- Daten der Grund- und Katasterkarte:  
In diesen Bereich gehören alle Geodaten, die nicht Anlagen oder Leitungen betreffen (z.B. Flurstücke, Gebäude, Straßenverzeichnis).
- Leitungsdaten:  
Hierzu zählen die technischen Betriebsmitteldaten (Leitungen, Anlagen etc.).
- Daten im ERP-/CRM-System:  
Alle betriebswirtschaftlichen Daten ohne unmittelbaren Raumbezug fallen in diese Gruppe (z.B. Kundendaten, Anlagen)

Neben den Daten mit entsprechender Struktur sind zur Eingliederung eines Geographischen Informationssystems auch bestimmte Funktionalitäten erforderlich, welche im Folgenden Kapitel näher beschrieben werden.

## 4.2.2 Funktionalität

Reine Erfassungs- und Fortführungsfunktionalität für die Bestandsdokumentation des Leitungsnetzes, wie sie von vielen Unternehmen bislang als Hauptfunktionsbereiche eines GIS genutzt worden sind, reichen aufgrund erhöhter Anforderungen nun nicht mehr aus. Themenpläne und Generalisierungsfunktionen für Schema- und Übersichtspläne sind nur ein Beispiel für den Bedarf, der über die reine Bestandsdatenerfassung und -pflege hinausgeht, so dass immer mehr der vom GIS-System bereit gestellten Werkzeuge Anwendung finden. Die Un-

terstützung einzelner Prozessschritte durch Geographische Informationssysteme basiert auf den jeweils benötigten GIS-Funktionalitäten, die bei (DVGW, 2003) auch als GIS-Bausteine bezeichnet werden. Im Wesentlichen handelt es sich bei diesen GIS-Bausteinen um Leistungsmerkmale von GIS-Systemen, die auch der detaillierten Beschreibung von Systemanforderungen gegenüber den GIS-Herstellern dienen können.

In diesem Kapitel werden die für Energieversorgungsunternehmen relevanten Funktionalitäten in Anlehnung an das Referenzmodell für GIS-gestützte Geschäftsprozesse beschrieben. Als wesentliche Informationen zu den einzelnen GIS-Bausteinen werden die benötigten Eingangsdaten (Input), vom System geforderte Funktionalitäten und durch das GIS zu generierende Ergebnisse (Output) graphisch dargestellt und die mit der GIS-Funktionalität verbundenen Ziele beschrieben. Die Beschreibung erfolgt in Anlehnung an (DVGW, 2003) und in alphabetischer Reihenfolge. Weitere, jedoch für den Bereich Störungsmanagement nicht relevanten Funktionalitäten werden zur Vervollständigung im Anhang A: Funktionalitäten von GIS aufgelistet.

#### 4.2.2.1 Netzverfolgung

Eine topologisch und logisch korrekte Erfassung des Versorgungsnetzes setzt die Funktion der Netzverfolgung als eine der wichtigsten Funktionen, die ein Geographisches System zu bieten hat, voraus. Die Ermittlung der temporär nicht versorgten Kunden bei Störungen oder die Selektion eines Netzbereiches für dessen Bereiche sind beispielhafte Anwendungen für die Netzverfolgung. Des Weiteren sind Funktionalitäten erforderlich, die das Netz aufgrund unterschiedlicher Anlagen- und Schieberzustände analysieren. Somit lassen sich die Ziele der Netzverfolgung folgendermaßen definieren:

- Ermittlung von temporär nicht versorgter Kunden und
- Datenselektion (z.B. Netz im Anschluss an einen Knoten)

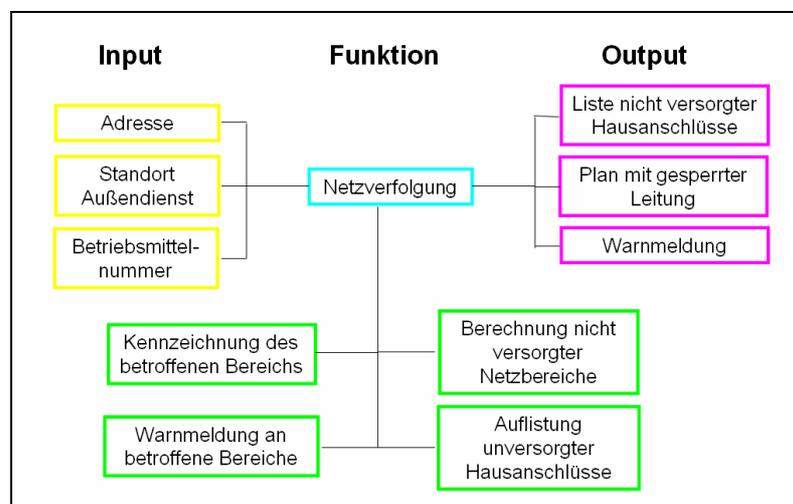


Abbildung 4.1: Funktion „Netzverfolgung“ nach (DVGW, 2003)

#### 4.2.2.2 Plotten

Obwohl die Zeit des papierlosen Büros angestrebt wird, so lässt sich an vielen Stellen die Produktion von analogen (Papierplänen) und digitalen Drucken (z.B. Rasterdaten, PDF-Dateien) nicht ganz vermeiden. Diese Funktion zählt jedoch zu den Basiswerkzeugen eines GIS-Systems, die von jeder Softwarelösung bereitgestellt wird. Ziel der Plottfunktionalität wird nach (DVGW, 2003) in den folgenden Punkten zusammengefasst:

- Analoge Ausgabe von Plänen und Kartenausschnitten (Bestands- und Übersichtspläne),

- Erzeugung von Rasterdaten in digitalen, standardisierten Formaten (z.B. JPG, TIFF, etc.) sowie
- interne und externe Auskunftserteilung, Dokumentation und Archivierung.

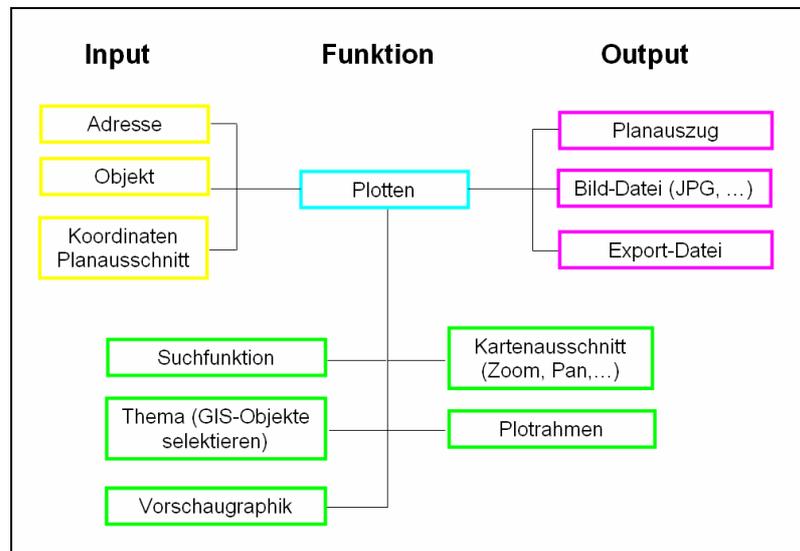


Abbildung 4.2: Funktion „Plotten“ nach (DVGW, 2003)

#### 4.2.2.3 Sachdaten erfassen / editieren

Ein Geographisches Informationssystem erlaubt es, zu den jeweiligen Grafik-Objekten auch Attribute oder Sachdaten abzulegen. Attributive Daten können erfasst und auch wieder geändert werden, wobei nach Wunsch Informationen über die Änderung der Daten zusätzlich erfasst werden können. Folgende Ziele lassen sich über Erfassungs- und Editierfunktion von GIS nach (DVGW, 2003) herausstellen:

- Eindeutige Zuordnung der Sachdaten zu den GIS-Objekten,
- Redundanzfreie (oder möglichst redundanzarme) Verwaltung der Sachdaten und
- Unterstützung zur Planung von Instandhaltung und Störfallmanagement.

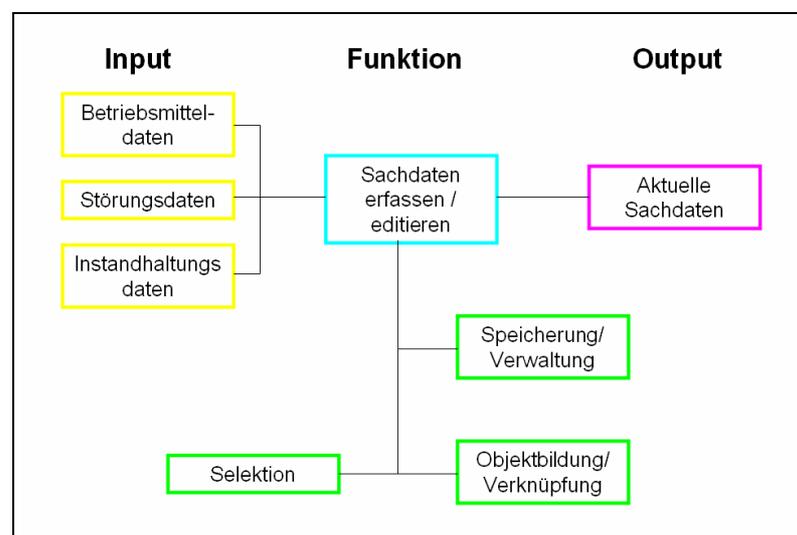


Abbildung 4.3: Funktion „Sachdaten ändern/ editieren“ nach (DVGW, 2003)

#### 4.2.2.4 Schnittstelle

Der Begriff Schnittstelle beschränkt sich hierbei allein auf die Fähigkeit, Daten aus anderen Systemen zu lesen und auch für andere Systeme auszugeben. Eine Umsetzung von Schnittstellen kann auf unterschiedlichen Technologien basieren (Schnittstellenprogrammierung, Middleware, Workflow-Managementsysteme, etc.).

Eine detaillierte Beschreibung der Schnittstellen ist bei der Prozessbeschreibung für Störfallmanagement aufgeführt. Dort werden auch die Ziel- bzw. Quellsysteme beschrieben und die Art der Schnittstelle näher spezifiziert. Allgemeine Ziele beim Einsatz einer Schnittstelle aus Sicht eines Energieversorgers sind laut (DVGW, 2003):

- Datenübergabe bzw. -übernahme von und zu anderen IT-Systemen und
- Datenaustausch mit Kommunen und externen Dienstleistern.

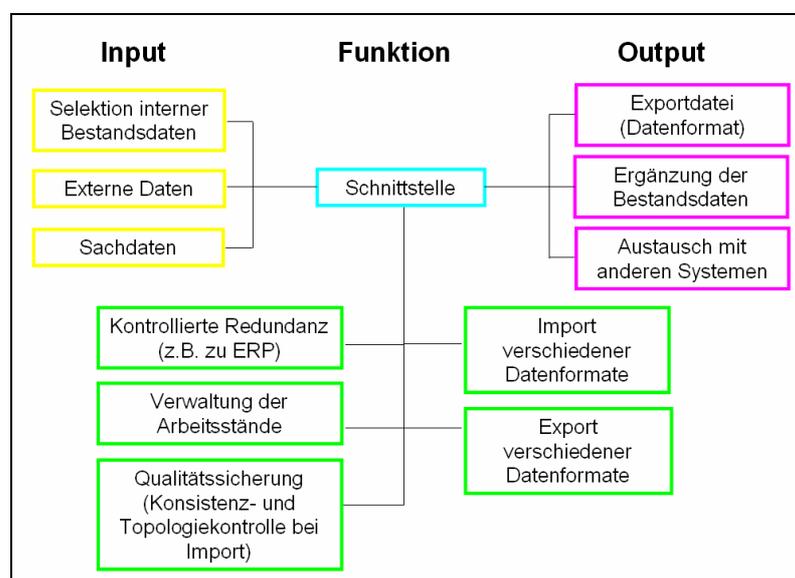


Abbildung 4.4: Funktion „Schnittstelle“ nach (DVGW, 2003)

#### 4.2.2.5 Tourenplanung

Gerade im Prozess Störfallmanagement ist der kürzeste Weg zu dem schadhafte Objekt von großer Bedeutung. Auch für die Planung von Instandhaltungstätigkeiten auf Basis eines GIS-Straßenverzeichnisses der Betriebsmittel kann das GIS mit geeigneten Funktionen die geforderten Dienste bereitstellen. Ziele, die laut (DVGW, 2003) durch die Tourenplanung erreicht werden können:

- Routenberechnung zur Optimierung der Instandhaltung und
- Unterstützung der Störtrupps beim Auffinden der Störstellen

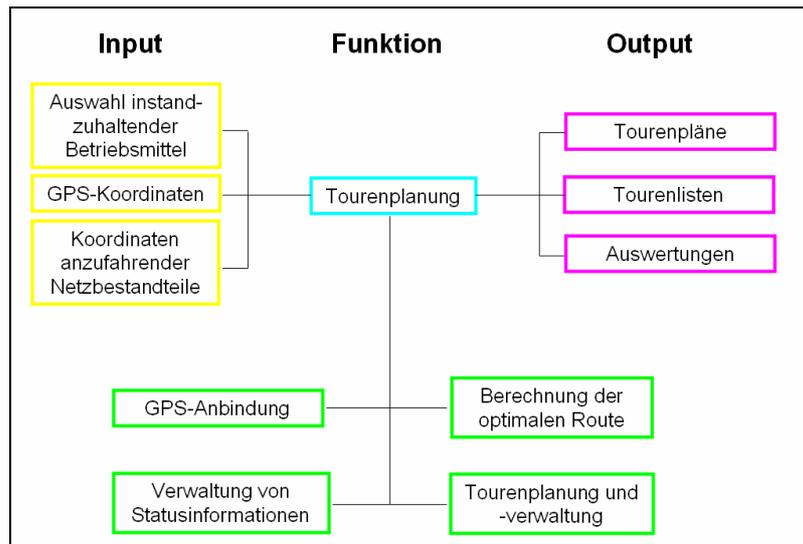


Abbildung 4.5: Funktion „Tourenplanung“ nach (DVGW, 2003)

#### 4.2.2.6 Visualisierung

Die Visualisierung von Daten ist eine der Basisfunktionalitäten eines Geographischen Informationssystems. Sie kann an entsprechend ausgestatteten Arbeitsplätzen erfolgen. Hierbei ist jedoch zu gewährleisten, die Präsentation der Daten so zu gestalten, dass sie für jeden Anwender genau auf die Erfordernisse seiner Tätigkeit zugeschnitten ist. Das Angebot der GIS-Hersteller reicht von Vollarbeitsplätzen bis hin zu internet- oder intranetbasierten Auskunftslösungen und mobilen Lösungen. Es wird ein unternehmensweiter Zugriff auf skalierten Arbeitsplätzen ermöglicht.

Des Weiteren sind im Rahmen der genannten Skalierbarkeit für die unterschiedlichen Nutzergruppen (Experte, Anwender, Betrachter) auch verschiedene Zugriffsmöglichkeiten zu bieten. Für reine Betrachter, die nur selten auf die GIS-Daten zugreifen, bietet sich eine vordefinierte Aufbereitung der Daten in Form von Informationsprodukten (z.B. Bestandsdaten, Übersichtsdaten, Schemadarstellung), die geringfügig angepasst werden können, an. Andere Nutzergruppen hingegen benötigen über das reine Betrachten hinausgehende Funktionen zur Erstellung eigener Informationsprodukte. Diese Flexibilität und Vernetzung von Funktionen wird in den weiteren GIS-Bausteinen ebenfalls deutlich. Die mit der Visualisierung verbundenen Ziele lassen sich nach (DVGW, 2003) in drei Punkten zusammenfassen:

- Darstellung der aktuellen Bestands- und Übersichtsdaten und weiterer Informationsprodukte (thematische Karten, Grundkarte, etc.),
- Zugriff auf GIS-Daten über verschiedene Arbeitsplätze (Online-Zugriff) und
- Mobile GIS (Einsatz vor Ort).

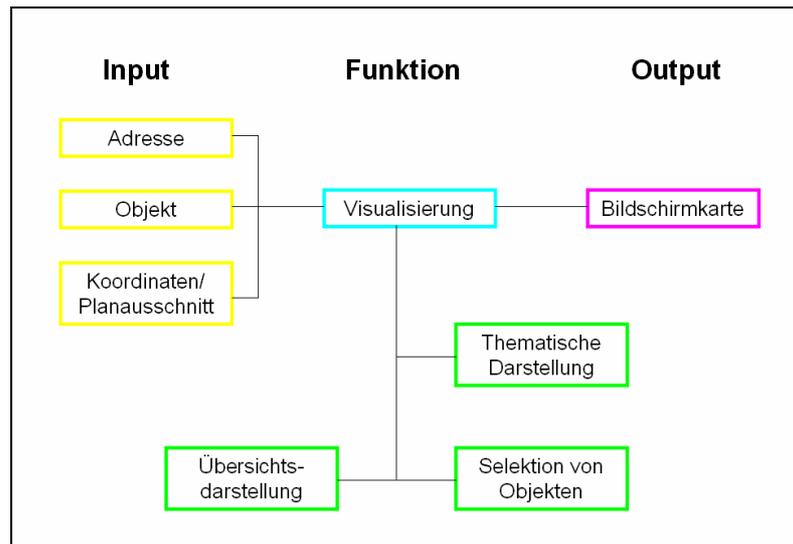


Abbildung 4.6: Funktion „Visualisierung“ nach (DVGW, 2003)

## 4.3 Ansätze zur Nutzung raumbezogener Informationen

Zur Einbindung von geographischen Daten in Geschäftsprozesse gibt es verschiedene technische Realisierungsansätze. Dieses Kapitel stellt die Möglichkeiten der GIS-Nutzung über Schnittstellen, Kopplungen, Integration sowie einen dienstbasierten Ansatz dar. Die Diskussion der einzelnen Varianten und eine Bewertung werden am Schluss des Kapitels geführt.

### 4.3.1 Variante 1 - Schnittstelle

Eine Schnittstelle (englisch: interface) ist allgemein bezeichnet ein Teil eines Systems, das dem Austausch von Informationen, Energie oder Materie mit anderen Systemen dient. Die Schnittstellenbeschreibung ist die zu einer Schnittstelle zugehörige Menge von Regeln (KnowLibrary, 2007b). Unterstützen Komponenten oder Module die gleiche standardisierte Schnittstelle, so lassen sie sich gegeneinander austauschen und man bezeichnet sie als kompatibel (Wikipedia, 2006d).

Im Softwarebereich unterscheidet man hinsichtlich Daten- und Funktionsschnittstellen. Eine Datenschnittstelle, auch Offline-Kommunikation genannt, regelt den klassischen Datenaustausch zwischen zwei Programmen, wird also zum Zugriff auf gespeicherte Daten (interne Datenschnittstellen) oder auch zum Austausch von Daten (externe Datenschnittstellen) verwendet. In den meisten Fällen findet der Austausch über standardisierte Datenformate statt (Geoinformatik-Lexikon, 2007). Entsprechend beschreibt das Schnittstellenformat die Codierung der Daten, die in einer Schnittstelle transportiert werden. DXF-Format, EDBS-Format sind Beispiele für Datenformate für Schnittstellen aus dem GIS-Bereich (Bill, Zehner 2001).

Bei Funktionsschnittstellen wird definiert, wie Daten und Anweisungen (Kommandos) zwischen verschiedenen Komponenten und Prozessen ausgetauscht werden. Man unterscheidet nach (Wikipedia, 2006d) hier zwei Arten:

- Zugriff auf Systemroutinen (Interprozesskommunikation) und
- Verbindung einzelner Softwarekomponenten (Module) eines Programms bzw. programmübergreifende Schnittstellen.

Schnittstellen zur Interprozesskommunikation (englisch: inter-process communication, IPC) ermöglichen den Informationsaustausch bzw. die Datenübertragung zwischen verschiedenen Programmen auf dem gleichen oder einem anderen Computer. Beispiele für netzwerkweite Kommunikationsschnittstellen sind Remote Procedure

Call, DCOM, RMI oder CORBA, aber auch ODBC und JDBC. Netzwerkprotokolle wie HTTP oder TCP können als IPC-Schnittstellen angesehen werden.

Schnittstellen für Programmkomponenten stellen eine formale Deklaration dar. Sie zeigt die vorhandenen Funktionen und deren Ansprache. Besitzen Module die gleiche Schnittstelle, können sie gegeneinander ausgetauscht werden. Diese Weise ermöglicht es auch, verschiedene Komponenten gleichzeitig zu entwickeln, ohne dass die erste fertig sein muss, um die zweite zu übersetzen. Die Modularisierung der Softwarearchitektur steht also im Vordergrund.

Auf die Erläuterung der Bedeutung einer Softwareschnittstelle in der objektorientierten Programmierung ist an dieser Stelle verzichtet worden, da dies für die Beurteilung im Folgenden keine Verwendung findet.

Der Vollständigkeit halber werden hier auch die Benutzerschnittstellen (oder Mensch-Maschine-Schnittstellen) aufgeführt. Sie stellen einen Punkt dar, an dem ein Mensch mit einem Gerät interagiert. Das können sowohl Bedienelemente einer Stereoanlage oder auch graphische Benutzeroberflächen (englisch: Graphical User Interface, GUI) sein (Wikipedia, 2006d).

### 4.3.2 Variante 2 - Kopplung

Das Wort Kopplung (auch Koppelung) wird vom Verb koppeln (verbinden) abgeleitet (Wikipedia, 2006e). In der Informatik versteht man hierunter die Verknüpfung von verschiedenen Anwendungen. Meistens laufen die Programme parallel ab. Es gibt jedoch Möglichkeiten, das jeweils andere Programm in einem bestimmten Kontext aufzurufen. Schnittstellen werden bei dieser Kommunikation also gekoppelt. Eine Unterscheidung wird hier hinsichtlich der Art der Datenübertragung gemacht (KnowLibrary, 2007a). Hierfür gibt es nach (Wikipedia, 2006e) folgende Möglichkeiten:

- Kopplung durch formatierte Dateien: Bei dieser Methode werden einfache Daten (so genannte Flatfiles) zur Übertragung genutzt. Alle tangierten Anwendungen müssen das verwendete Format verstehen. Tabulatorgetrennte bzw. kommagetrennte oder XML-basierte Dateiformate werden häufig verwendet.
- Kopplung mit Konverter: Der Konverter löst die Problematik, dass die sich verschiedenen Anwendungen auf ein Format zum Austausch einigen müssen. Er wandelt ein Datenformat in ein anderes um. Es wird versucht, die Semantik der Daten beizubehalten, was meist jedoch nicht vollständig möglich ist. Im Bereich von freier Software gibt es viele Beispiele für Konverter. So wandelt beispielsweise das Programm shp2dxf eine Shape-Datei in eine DXF-Datei um.
- Kopplung durch Kopplungsprozeduren: Bei dieser Variante ruft der Client beim Server eine Funktion auf. Mit ihr werden dann die Daten übertragen. Auch ist über diese Art der Kopplung das direkte Starten einer Logik auf der Serverseite möglich. Jedoch muss meist eigens für diese Methode der Quelltext von einem oder von beiden Anwendungen angepasst werden. Es existieren unterschiedliche Techniken in den verschiedenen Programmiersprachen und Konzepten, um mit dieser Kopplung zu arbeiten. RPC oder RFC, CORBA, RMI, SOAP bzw. Web Service, .NET Remoting, COM, DCOM seien als Beispiele genannt.
- Die Kopplung im Sinne der objektorientierten Programmierung wird hier nicht aufgeführt, da sie für den zu bearbeitenden Kontext nicht relevant ist.
- Kopplung durch menschlichen Eingriff: Dies ist keine Kopplung im engeren Sinne und wird hier nur der Vollständigkeit halber mit aufgeführt. Bei dieser Kopplung werden die Informationen über oder durch einen Menschen von einem System ins andere transportiert.

Je nach Grad der Kopplung spricht man auch von einer engen oder losen Kopplung. Beide Varianten werden im Folgenden diskutiert.

### 4.3.3 Variante 3 - Integration

Die Integration hat ihren begrifflichen Ursprung im Lateinischen (*integer*) und Griechischen (*entagros*) mit der Bedeutung zur Herstellung eines Ganzen bzw. sämtliche Aspekte der Ganzheitsbildung. Die Integration in der Informatik dient zur Verknüpfung von verschiedenen Anwendungen unter der Prämisse, Schnittstellen zu verringern bzw. zu vermeiden. Die Integration wird auch als enge Kopplung bezeichnet. Bei der Integration lassen sich drei Ebenen unterscheiden: die Funktionsintegration, die Datenintegration und die Geschäftsprozessintegration.

Werden mehrere Applikationen in einer Anwendung zusammengefasst, so bezeichnet man diese Art der Integration als Funktionsintegration. Eine Mehrfachimplementation von Funktionen wird hierdurch verhindert und Schnittstellen zur Datenübergabe vermieden.

Der Datenintegration liegt ein einheitliches Datenmodell inklusive Semantik zugrunde, auf das alle beteiligten Applikationen zugreifen, um Redundanzen zu vermeiden.

Die Integration von verschiedenen Funktionen entlang der Geschäftsprozesse über eine Integrationsplattform wird als Geschäftsprozessintegration (englisch: *enterprise application integration, EAI*) bezeichnet. Hier kann Standardsoftware zum Einsatz kommen, da die einzelnen Applikationen, die die Funktionen bereitstellen, in sich unverändert bleiben (Wikipedia, 2006f).

Des Weiteren sind bei Integrationsszenarien auf Systemebene auch die Methoden der Implementierung von Bedeutung, die in der folgenden Grafik dargestellt werden.

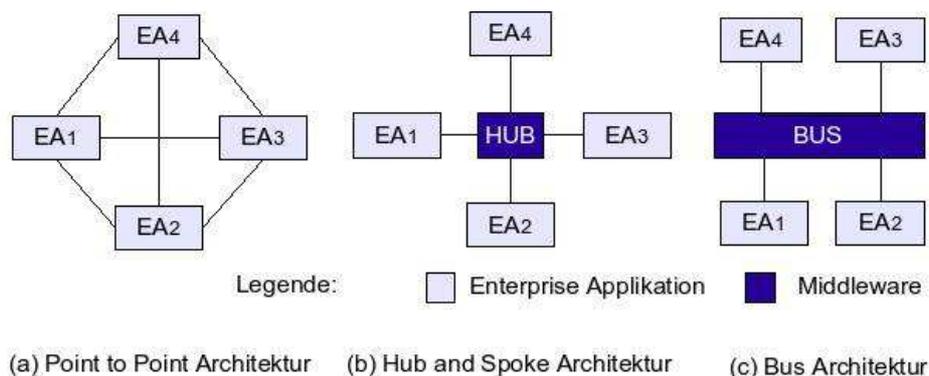


Abbildung 4.7: Integrationsstopologien (Bator, 2007)

Die einfachste Variante der Integration ist die direkte Verbindung der Applikationen. Sie wird Point-to-Point- oder Peer-to-Peer-Architektur (Abbildung (a)) genannt. Sie ist ohne Middleware zu realisieren, benötigt aber Adapter zur Verbindung der Applikationen untereinander. Im ungünstigsten Fall müssen die Adapter aufgrund von Applikationsänderungen neu implementiert werden. Dies entfällt bei der Hub and Spoke (Nabe-Speiche-Architektur in Abbildung (b)) bzw. bei der Bus-Architektur (Abbildung (c)). In diesen Architekturen wird ein gemeinsames Datenformat definiert, in das alle Nachrichten konvertiert werden können. Hub und Bus können das Format interpretieren und die Nachricht an die gewünschte Applikation weiterleiten. Bei diesen beiden Architekturen ist lediglich die Entwicklung und Pflege eines Adapters notwendig. Die Bus Architektur unterscheidet sich von der Hub and Spoke Architektur dadurch, dass der Bus eine rein logische Architektur darstellt und Provider und Requester rein logisch voneinander trennt (Bator, 2007).

Das Informations-Portal wird hier abschließend als Sonderform der Integration aufgeführt. Der Ausdruck Portal (lat. *porta*, Pforte) steht nach (Wikipedia, 2006f) in der Informatik für einen zentralen Zugang, der einen Zugriff auf individuell zugeschnittene, unternehmensinterne und externe Informationen und Dienste ermöglicht. Als Form der Integration steht auch beim Portal der Integrationsaspekt und somit das Bereitstellen von applikationsübergreifenden Leistungen und nicht die technische Implementierung (z.B. webbasiert) im Vordergrund. So liegt

es nahe, beim Aufbau eines Portals entweder auf eine EAI-basierende Infrastruktur zurückzugreifen oder eine Portal-Standard-Software zu benutzen, die sich der EAI bedient (Wikipedia, 2006f).

#### 4.3.4 Variante 4 - dienstbasierter Ansatz

Eine weitere Möglichkeit der Nutzung von GIS-Daten in Geschäftsprozessen ist der Einsatz von Diensten (englisch: Services), die auch als lose Kopplung bezeichnet werden können. In der Regel sind diese Dienste webbasiert, so dass man von Web Services spricht. Sie stellen die derzeit bestmögliche Realisierungsoption einer Service-orientierten Architektur dar. Die drei Bausteine in diesem Konzept, in welchem ein Dienst (Service) im Mittelpunkt steht, sind Kommunikation, Dienstbeschreibung und Verzeichnisdienst (Siehe auch Kapitel 3.3.2). Die Web Services Architektur manifestiert sich somit im Publish-Find-Bind-Paradigma. So werden die Metadaten eines Web Service von einem Service-Anbieter mit Hilfe einer Service-Registrierung veröffentlicht (publish). Ein Service-Konsument hat die Möglichkeit, eine solche Service-Registrierung nach einem für sich geeigneten Service zu durchsuchen. Wird ein passender Service gefunden (find), kann dieser vom Service-Konsumenten direkt verwendet werden (bind) (Kiehle, 2006). In der Web Service-Technologie haben sich bereits Standards für die drei Komponenten durchgesetzt. SOAP wird in der Nachrichtenkommunikation eingesetzt. WSDL dient zur Beschreibung der zu publizierenden Web Services und der Verzeichnisdienst wird vom Standard UDDI abgebildet. Neben den beschriebenen Standards kommen auch die beiden Internet-Standards HTTP und XML zum Einsatz. Mit den genannten Standards wird eine Unabhängigkeit von Plattform und Programmiersprachen erreicht (Dostal et al., 2005).

Außerdem erfolgt eine Kommunikation ausschließlich mit den notwendigen Web Services mit Anbindung zur Laufzeit, so dass auch von einer losen Kopplung gesprochen wird. Mehrere Web Services können über eine Prozesssteuerungssprache zu einem Prozess orchestriert werden, der wiederum über seine WSDL-Beschreibung selbst als Web Service angesprochen und verwendet werden kann.

#### 4.3.5 Bewertung der Varianten

Zwischen den einzelnen Konzepten gibt es eine hierarchische und assoziative Beziehung, die im Zusammenhang erläutert werden.

Die Kommunikation beginnt auf der physischen Ebene von Bits und Signalen mit einem Übertragungskanal, der Nachrichten übermittelt. Ist der Kanal öffentlich und vordefiniert, so bezeichnet man diesen Kanal samt der Struktur der darüber übermittelten Nachrichten als Schnittstelle. Ein System oder eine Applikation kann mehrere Schnittstellen implementieren. Kommunizieren mehrere Systeme über mehrere Schnittstellen, so heißt die Gesamtheit aller Schnittstellen dieser Systeme Kopplung. Bei einer Kopplung laufen die Programme nebeneinander. Es gibt aber Möglichkeiten z.B. über eine Schaltfläche das jeweils andere Programm in einem bestimmten Kontext aufzurufen. Beispiel: „Ruf das GIS auf und zeige mir mein gestörtes Betriebsmittel“. Hinsichtlich der Struktur von Kopplungen wird in lose und enge Kopplung unterschieden. Bei einer engen Kopplung spricht man auch von Integration. Es wird also mit Hilfe von ganz speziellen Schnittstellen aus mehreren Systemen auf einer bestimmten (Abstraktions-) Ebene ein neues eigenes System zusammengebaut. In diesem Sinne sollen sich die einzelnen Systeme in dieses (neue) gesamte oder übergreifende System integrieren. Bei einer Integration hat man normalerweise einen Programmteil (ein eingebettetes Kartenfenster) oder einfach nur die Daten der jeweils anderen Anwendung in der Hauptanwendung sichtbar, so dass optisch die Meinung entsteht, dass es sich nur um ein Programm handelt. SOA ist eine ganz spezifische Strukturierung von Schnittstellen zwischen Softwaresystemen, die auf einer Trennung von Schnittstellendefinition und Schnittstellenimplementierung beruht und damit eine lose Kopplung realisieren möchte. Das bedeutet, dass ein System nur mit Kenntnis der Schnittstellendefinition alleine bereits die Schnittstelle aktivieren kann.

Ein wichtiger Punkt zur Unterscheidung zwischen der Geschäftsprozessintegration (EAI) und (echter) SOA sind Aspekte zur Abbildung von Ort und Art der Prozesslogik. Werden die (vor)strukturierten Aktivitäten explizit modelliert und von einer "Prozess Maschine" ausgeführt, so sprechen wir von Geschäftsprozessintegration. In einer "echten" SOA gibt es "die" Applikation als solche nicht mehr, da die IT-Landschaft in Prozesse und Services "zerfällt", ohne dass eine Applikation von einer anderen unterschieden werden könnte. Des Weiteren basiert eine typische EAI-Lösung auf verteilten Technologien, wie beispielsweise DCOM, CORBA und MOM, somit

erhöhen sich im Gegensatz zur Anwendung von Web Services die Konzeption, die Entwicklung und nicht zuletzt die Wartung der Softwarelandschaft. Aufgrund der losen Kopplung als Kommunikation zwischen einzelnen Services findet man eine wesentlich höhere Flexibilität als bei traditionellen EAI-Lösungen. Weiterhin erlauben Web Services die Zerlegung komplexer Funktionalitäten in kleine unabhängige Einheiten, die in verschiedenen Geschäftsprozessen verwendet werden können. Somit wird die Zusammensetzung von Geschäftsprozessen leichter und effizienter.

Somit kann das Ziel der technologischen Kommunikation mit geographischen Daten bzw. Informationen nur über die Web Services in der Service-orientierten Architektur erfolgen, um einerseits ein Höchstmaß an Flexibilität und Wiederverwendbarkeit zu erreichen und um sich andererseits in die Technologien der „Nicht-Geo“-Informatik einzugliedern bzw. sich dem allgemeinen Bereich der Informatik anzupassen.

## 4.4 DVGW-Referenzmodell für GIS-gestützte Geschäftsprozesse

Um den vorangegangenen technischen Realisierungsmöglichkeiten Rechnung zu tragen, wird nun ein Beispiel für den inhaltlichen Kontext diskutiert. Da diese Arbeit ihren Schwerpunkt auf der Bearbeitung mit Standards in jeglicher Hinsicht hat, bietet sich für den Inhalt eines Geschäftsprozesses auch die Analyse eines entsprechenden Fachwerkes an. Der Deutsche Verein des Gas- und Wasserfaches e.V. (DVGW) beschäftigt sich im Rahmen seiner Vereinstätigkeit neben Handlungshinweisen und Merkblättern zum Arbeiten mit den Medien Gas und Wasser auch mit deren Abbildung in Geographischen Informationssystemen. So ist beispielsweise im Jahre 1990 der Technische Hinweis GW 122 „Netzinformationssystem – Aufbau und Fortführung mit Hilfe der grafischen Datenverarbeitung (GDV)“ erstmalig erschienen. Dieser Hinweis enthält für Versorgungsunternehmen mit den Sparten Gas und Wasser wichtige Informationen zur Einführung und nachfolgender Pflege eines GIS-Systems. Als ein weiterer Schritt zur Erleichterung der Leitungsdokumentation hat der DVGW für seine Mitglieder ein Datenmodell, das Gas-Wasser-Netzinformationssystem (GAWANIS-) Datenmodell, entworfen, das die Basis für alle folgenden Hinweise und Mitteilungen im Bereich geographische Informationsverarbeitung bildet.

Je nach Größe der Energieversorgungsunternehmen befindet sich die Ersterfassung der geographischen Daten in der Abschlussphase. Aufgrund dieser neuen Situation wurde im Jahr 2002 der Hinweis GW 119 „Verbesserung von Geschäftsprozessen durch die Einbindung von GIS“ unter Federführung des DVGW-Fachausschusses „Technische GIS“ vom Arbeitskreis „NIS/GIS“ erarbeitet. Dieser Hinweis war Ausgangspunkt für die „Studie zur Nutzung von digitalen GIS-Daten in prozessunterstützten Geschäftsabläufen eines Gas- und Wasserversorgungsunternehmens“. Es erfolgte erstmalig eine zusammenfassende Darstellung über den Verwendungsumfang raumbezogener Daten in deutschen Versorgungsunternehmen, die der DVGW in einem Referenzmodell im Jahr 2003 veröffentlichte. Ende 2005 wurde der Extrakt „DV-gestütztes Störfallmanagement und Schadensstatistik unter Einbindung von GIS“ aus dem Referenzmodell als Technische Mitteilung GW 133 publiziert. (DVGW, 2005)

Durch die Beteiligung von 12 DVGW-Mitgliedsunternehmen erhielt die Studie einen Referenzcharakter. Die Auswertung resultiert in der Modellierung von 5 Haupt- und 7 Teilprozessen, die durch Erhebungen bei den Versorgungsunternehmen kommuniziert, diskutiert und abgestimmt wurden. Zusätzliche Anmerkungen der Versorgungsunternehmen zu den Ergebnissen sind ebenso im Referenzmodell berücksichtigt. Da dieses Referenzmodell in Deutschland das einzige seiner Art ist und sämtliche Geschäftsprozesse, die einen Geobezug aufweisen, untersucht, bietet sich im Rahmen dieser Arbeit eine Analyse des Werks an. Den Kern des Referenzmodells bilden nach (DVGW, 2003) die folgenden vier Punkte:

- Dokumentation der Referenzprozesse als graphische Vorgangskettendiagramme inklusive der Beschreibung aller Arbeitsschritte,
- Darstellung der erforderlichen Systemschnittstellen je Prozess zwischen GIS und anderen Standardsystemen in Form von objekthaften Kommunikationsflüssen,
- Auflistung der Geodaten, die zur Unterstützung eines Arbeitsschrittes durch ein GIS benötigt werden sowie

- Gruppierung der zur Prozessunterstützung notwendigen GIS-Funktionen zu GIS-Bausteinen (Funktionsgruppen).

Unter Berücksichtigung der Kernpunkte sind vier wesentliche Einzelziele im Zuge der Erstellung des Referenzmodells erreicht worden. Es handelt sich zunächst um die Darstellung normalisierter Geschäftsprozessmodelle für Prozesse mit geodätischer Referenz. Durch diesen prozessorientierten Einsatz eines Geographischen Informationssystems sind die zugehörigen Optimierungspotentiale erfasst worden. Definition der prozessrelevanten GIS-Daten sowie die Empfehlung für eine prozessorientierte GIS-Integration mit anderen Standard-IT-Systemen in der Softwarelandschaft von Energieversorgungsunternehmen sind gleichermaßen dokumentiert. Zum Erreichen der genannten Einzelziele ist ein Zusammenwirken verschiedener Aspekte wichtig. Der (DVGW, 2003) nennt die folgenden Anforderungen als relevant für die Erschließung des Optimierungspotentials.

- Bereitstellung von Prozesslösungen durch GIS-Anbieter,
- einheitliche, standardisierte und offene Datenhaltungssysteme mit Zugriffsmöglichkeiten für alle bedeutsamen Systeme und
- intelligente Schnittstellen zwischen den wichtigsten Systemen unter Einsatz modernster Informationstechnologie.

Das Referenzmodell ist zunächst in Kern-, Haupt- und Teilprozesse, die hierarchisch gegliedert sind, unterteilt. So werden den übergeordneten Kernprozessen, oft auch als Unternehmensbereiche eines Versorgungsunternehmens bezeichnet, die Detaillierungsstufen Haupt- und Teilprozesse zugeordnet. Hauptprozesse durchlaufen meist mehrere Abteilungen, wobei ihr Ablauf einmalig innerhalb des Unternehmens ist. Sie zeichnen sich zudem durch eine Wertschöpfung für das Unternehmen aus. Teilprozesse bilden einen begrenzten Ablauf ab und unterstützen die Hauptprozesse an verschiedenen Stellen. Abbildung 4.8 zeigt die Zusammenhänge der Prozessstufen. Die graphische Darstellung der Geschäftsprozesse in einem Versorgungsunternehmen gliedert die Kernbereiche in die Hauptfarben blau („Marketing und Vertrieb“), gelb („Planen und Bauen“) und rot („Betreiben und Instandhalten“). Die Hauptprozesse orientieren sich farblich an den Kernprozessen. Die innerhalb der Kernprozesse auftretenden Teilprozesse sind gleichermaßen farblich gekennzeichnet (z.B. erfolgt die Auskunftserteilung sowohl im Kernprozess „Planen und Bauen“ als auch im Kernprozess „Betreiben und Instandhalten“). Der Vollständigkeit halber sind zu den Kernprozessen zählende, aber wegen fehlender GIS-Relevanz nicht beschriebene Hauptprozesse ebenfalls erwähnt („Kundenbetreuung“, „Planen und Bauen von Anlagen“ sowie „Netzführung“).

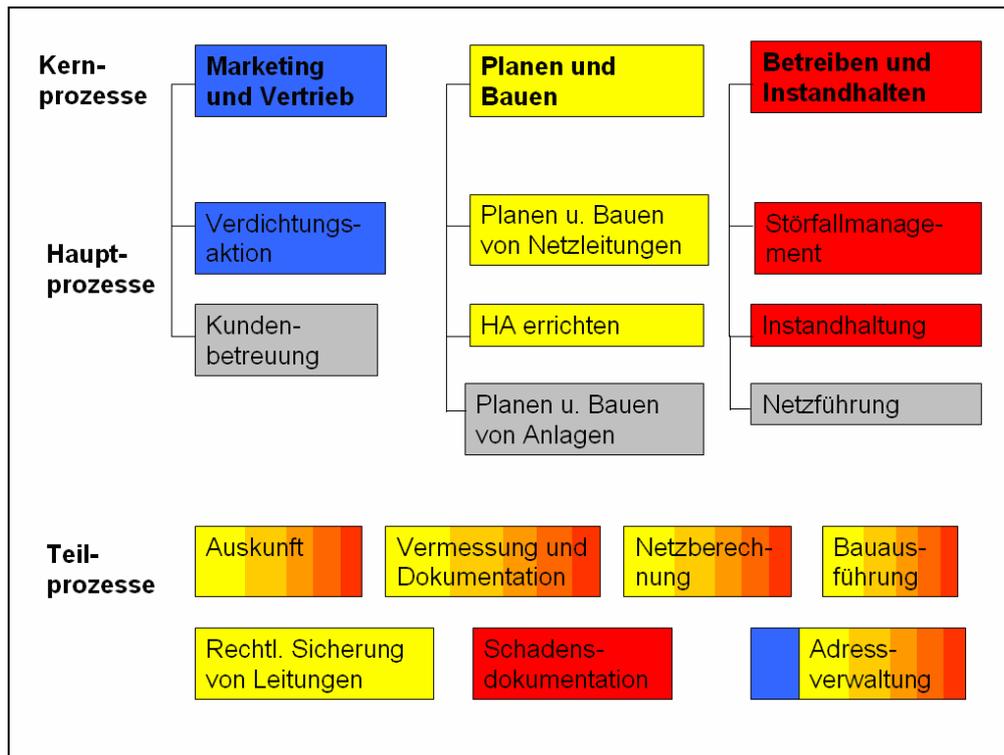


Abbildung 4.8: Zuordnung von Haupt- und Teil- zu Kernprozessen nach (DVGW, 2003)

Es folgt eine Definition und Abgrenzung der Geschäftsprozesse in der Versorgungswirtschaft. Jeder Prozess wird im Einzelnen untersucht und in Fließtext, Grafik und tabellarischer Form samt den beteiligten Systemen und Schnittstellen beschrieben. Die prozessunterstützenden GIS-Funktionalitäten und die Strukturierung der prozessrelevanten GIS-Daten werden gleichermaßen aufgeführt. Je Prozess wird abschließend das Nutzungspotential durch den gezielten GIS-Einsatz beschrieben.

Das Optimierungspotential wird laut (DVGW, 2003) durch verschiedene Prozesseigenschaften ermittelt. Dazu gehören Strukturiertheit, Ausführungshäufigkeit, beteiligte Ressourcen und die Prozessqualität. Der Hauptprozess Störfallmanagement nimmt unter allen Prozessen hinsichtlich dieser Kriterien den mittleren Platz ein und wird aus diesem Grund Basis für detailliertere Betrachtungen hinsichtlich der GIS-Integration.

#### 4.4.1 Beschreibung des Prozessmodells Störfallmanagement

In diesem Kapitel wird der Hauptprozess Störfallmanagement im Detail betrachtet. Nach einer allgemeinen Prozessbeschreibung wird der Ablauf in einer graphischen Darstellung als Ablaufdiagramm aufgezeigt und im Anschluss tabellarisch beschrieben.

Bei einer Störung des Versorgungsnetzes besteht das Ziel, die Entstörung sofort durchzuführen, damit die Netze schnell wieder in Betrieb genommen werden können. So ist im Hauptprozess Störfallmanagement das Wissen über Art und Lage der Betriebsmittel samt Netzlogik von großer Bedeutung. Zusätzlich werden Informationen über die Auswirkung der Störung auf den Versorgungszustand in der Netzleitstelle benötigt. Die Ermittlung von Sperrstrecken sowie notwendige Absperrmaßnahmen im Versorgungsnetz und die daraus resultierenden betroffenen Kunden sind Informationen von immenser Wichtigkeit für den Einsatzleiter. IT-Systeme liefern hier wertvolle Unterstützung, die auch im Bereich der Koordination von Mitarbeitern und Fahrzeugen angesehen werden kann. Beispielsweise müssen die Entstörtruppe beim Medium Gas nach DVGW-Vorschrift innerhalb 30 Minuten am Störungsort eintreffen. Anmerkungen der bei der Studie partizipierenden Versorgungsunternehmen waren, dass das Störfallmanagement mit der Aufnahme der Störungsmeldung beginnt und mit der Gefahrenabwehr und Beseitigung der Störung (z.B. Gas-/Wasser-Austritt) endet. Tiefbaumaßnahmen werden im Rahmen der schadensbedingten Instandsetzung abgewickelt.

Zusammenfassend sind nach (DVGW, 2003) folgende Funktionalitäten mit GIS-Relevanz zur Unterstützung im Störfallmanagement notwendig:

- Annahme, Bearbeitung und Auswertung der Störungen (mit einer evtl. Integration mit ERP-Systemen für nachgeschaltete Instandsetzungsarbeiten) sowie die Erfassung aller schadensrelevanten Daten.
- Management von Bereitschaftsdienst und Wachbetrieb (z.B. Tätigkeiten, Mitarbeiter, Fahrzeuge, Geräte).
- Fahrwegoptimierung für Einsatzfahrzeuge.
- Lokalisierung der Störungsorte über die Adresse (Straße und Hausnummer) mit Einsatz von mobilem GIS.
- Darstellung der Störung im GIS.
- Übersicht der gestörten Leitungen und Anlagen (Technische Daten, Lage, Netzverfolgung).
- Ermittlung unversorgter Abnehmer durch eine Netzverfolgung.
- Erkennung von Großstörungen durch Darstellung aller Störungen.
- Analyse der Störungen (z.B. Ursache, Dauer).
- Verknüpfung von Störungsort und beschädigten Versorgungsobjekten.
- Weiterleitung der Informationen zur Instandhaltung.
- Unterstützung zur Anfertigung der DVGW-Schadensstatistik.

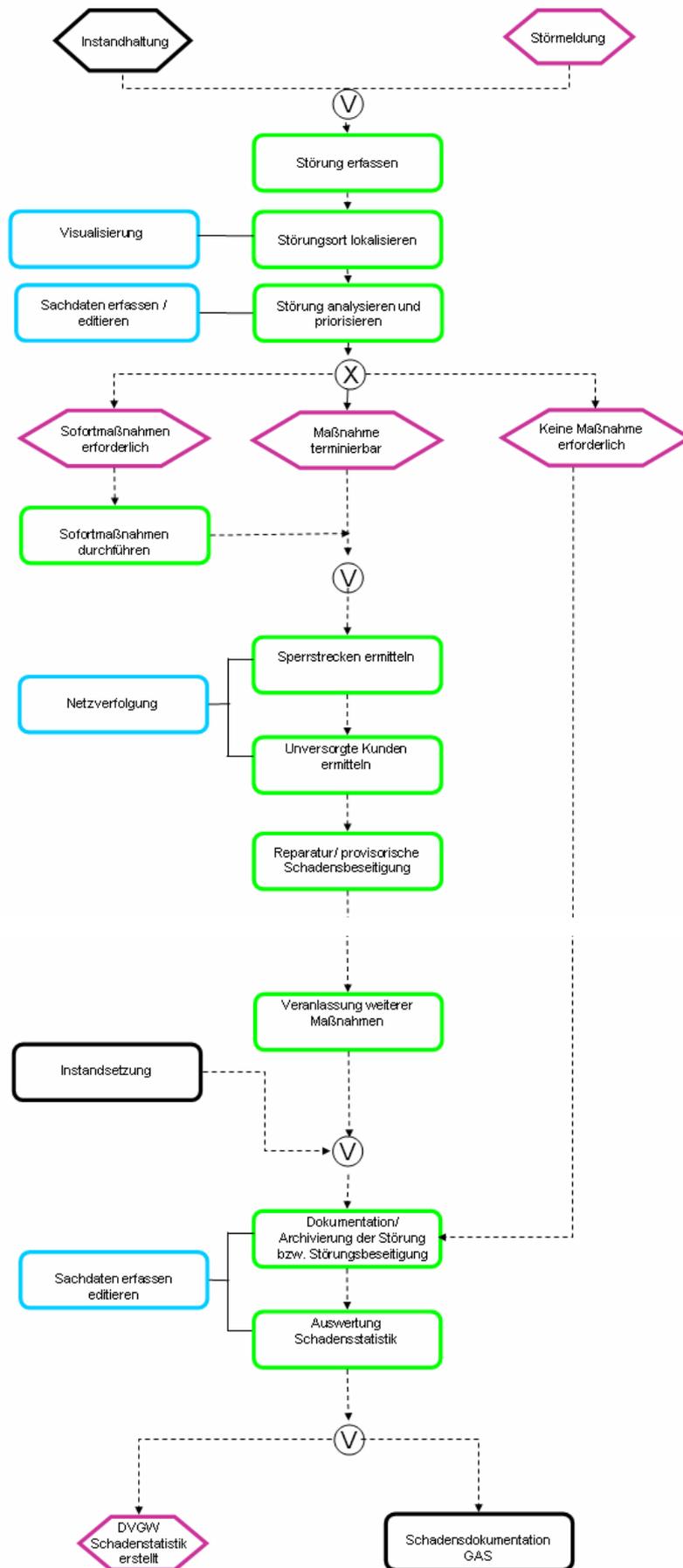


Abbildung 4.9: Darstellung des Ablaufes Störfallmanagement nach (DVGW, 2003)

Es folgt eine tabellarische Ablaufbeschreibung als Beschreibung der Grafik.

Typ	Aktion	Erläuterung
Prozessschnittstelle	Instandhaltung	Der Prozess Störmeldung wird vom Prozess Instandhaltung dann angestoßen, wenn im Rahmen der Instandhaltungsarbeiten Störungen an den Betriebsmitteln festgestellt worden sind.
Ereignis	Störmeldung	Das Eingehen einer Störmeldung ist der Beginn des Prozesses Störfallmanagement. Die Art, wie eine Störmeldung eingeht, ist vielfältig und umfasst neben dem Anruf durch den Kunden auch die Störmeldung auf der Baustelle, die von Dritten gemeldet wird, etc. Allen Störmeldungen gemeinsam ist, dass das VU über die Störung eines Betriebsmittels in Kenntnis gesetzt wird.
Vorgang	Störung erfassen	Gem. der gesetzlichen Vorgaben sind eingehende Störmeldungen zu erfassen. In der Störmeldung enthalten sein müssen neben der Uhrzeit und des Inhaltes der Störungsmeldung auch die Lage, die Art, etc. festgehalten werden.
Vorgang	Störungsort lokalisieren	Der Ort der Störung wird durch die Abfrage von Straße und Hausnummer, ggf. auch durch eine Betriebsmittelnnummer (Zählernummer, Hydranten-/Schiebernummer, etc.) beschrieben und auf einem Lageplan lokalisiert. Mehrfachmeldungen eines Schadens und Störmeldungen, die auf einem bereits erfassten Schaden beruhen, lassen sich im Rahmen der Lokalisierung besser beurteilen.
Vorgang	Störung analysieren und priorisieren	Die Störung wird hinsichtlich ihrer Dringlichkeit analysiert und priorisiert. Erfolgt die Erfassung über sog. Maßnahmenkataloge, so ist die Analyse und Priorisierung das Ergebnis einer automatischen Bewertung der vorliegenden Informationen.
Ereignis	Sofortmaßnahmen erforderlich	Je nach Aussagen der Störmeldung kann es erforderlich sein, Sofortmaßnahmen einzuleiten.

Typ	Aktion	Erläuterung
Vorgang	Sofortmaßnahmen durchführen	Meist werden Sofortmaßnahmen dann erforderlich, wenn Gefahr im Verzug ist. Hierzu gehören Blaulichtfahrt, Notabsperungen, großflächige Schaltmaßnahmen durch die Netzleitstellen, etc.
Ereignis	Maßnahmen terminierbar	Nicht immer muss eine Störung sofort beseitigt werden. Dabei stellt die Terminierung das Gegenteil der Sofortmaßnahme dar.  Die Terminierung kann Minuten, Stunden aber auch Tage umfassen, wenn die gemeldete Störung z.B. unkritisch ist.
Ereignis	keine Maßnahmen erforderlich	Wird im Rahmen der Aufnahme der Störmeldung festgestellt, dass das VU für die gemeldete Störung nicht verantwortlich ist, werden meist keine weiteren Maßnahmen durchgeführt. I.d.R. wird der Meldende an die zuständige Stelle verwiesen.
Vorgang	Sperrstrecken ermitteln	Die Sperrstrecken werden entweder durch Auswertung der Bestandsdaten ermittelt. Dies geschieht durch eine Netzverfolgung.
Vorgang	unversorgte Kunden ermitteln	Über die Netzverfolgung können diejenigen Kunden ermittelt werden, die von einer Netzsperrung betroffen wären und nicht versorgt würden.  Durch Ermittlung von Absperralternativen kann untersucht werden, inwiefern die Versorgung spezieller Kunden durch Vornahme einer anderen Absperrmaßnahme gewährleistet werden kann.
Vorgang	Reparatur / provisorische Instandsetzung	Vor Ort erfolgt nach Absperrung die Reparatur bzw. die provisorische Instandsetzung. Die Entstörungscrew ist nicht zuständig für die endgültige Wiederherstellung des Versorgungszustandes.

Typ	Aktion	Erläuterung
Vorgang	Veranlassung weiterer Maßnahmen	Mit Absperrung des Gas-/Wasseraustritts endet i.d.R. die Tätigkeit der Entstörungsscrew auf der Baustelle. Die endgültige Wiederherstellung der Versorgung sowie der Schutz des Betriebsmittels durch Isolation, Tiefbaumaßnahmen, Oberflächenwiederherstellung, etc. erfolgt grundsätzlich nicht durch die Entstörungsscrew. Die Weiterleitung der Maßnahmen erfolgt IT-unterstützt.
Prozessschnittstelle	Instandsetzung	Die endgültige Wiederherstellung eines Versorgungszustandes erfolgt im Rahmen der Instandsetzung. In diesem Prozess wird auch entschieden, ob eine Reparatur oder eine Sanierung erfolgen soll.
Vorgang	Dokumentation / Archivierung der Störung bzw. der Störungsbeseitigung	Die Störung wird komplett dokumentiert. Hierzu gehört neben der Erstellung von Schadensskizzen auch die Archivierung aller mit der Entstörung betroffenen Maßnahmen (wer hat was wann gemacht).  In diesem Zusammenhang wird die rechtlich vorgeschriebene Dokumentation der Störung komplettiert und archiviert.
Vorgang	Auswertung / Schadenstatistik	Die vorliegenden Daten der Störung / des Schadens werden ausgewertet und der Schadenstatistik zugeführt.
Ereignis	DVGW-Schadenstatistik erstellt	Aus der Gesamtheit aller vorliegenden Störungen / Schadensinformationen wird die DVGW-Schadenstatistik erstellt.
Prozessschnittstelle	Schadensdokumentation	Jede Störung, die je an einem Betriebsmittel aufgetreten ist, wird dem Teilprozess Schadensdokumentation zugeführt. Ziel ist die Langzeit-Archivierung der Störmeldungsdaten zwecks Entscheidung über Reparatur oder Sanierung im Rahmen von Rehabilitationsmaßnahmen.

Tabelle 4.4: Beschreibung des Ablaufes Störfallmanagement nach (DVGW, 2003)

In der Ablaufbeschreibung für den Hauptprozess Störfallmanagement lassen sich von diesem Schnittstellen zu weiteren IT-Systemen erkennen. Es werden das Geographische Informationssystem, die Verbrauchsabrechnung bzw. das CRM, die Netzberechnung, das Netzleitsystem und das Instandhaltungsprogramm für die Erzeugung von weiteren Informationen benötigt.

Das GIS-System hat zu Beginn des Prozessablaufes die Aufgabe der Visualisierung von Adresse und/oder Betriebsmittel auf dem Bestandsplan des Versorgungsgebietes. Dies erfolgt in der Regel durch Eingabe von Adres-

se, Betriebsmittelnummer oder ähnlichem. Im weiteren Prozessverlauf kommen topologische Abfragen dazu. Es handelt sich um die Leitungsverfolgung zur Ermittlung von Sperrstrecken und den damit verbundenen Absperrorganen sowie um die Ermittlung der unversorgten Kunden (Versorgungsknoten), die an den abgesperrten Leitungen (Versorgungskanten) hängen.

Die Verbrauchsabrechnung oder auch das CRM wird zur Identifizierung von so genannten „gesperrten Kunden“ benötigt, da es sich in diesem Fall nicht um einen untypischen Betriebszustand, sondern lediglich um eine gewollte Nicht-Versorgung eines einzelnen Kunden handelt.

Das Netzleitsystem wird zum Schalten der Sperrstrecken bzw. zur Freischaltungsbestätigung für den Beginn der Arbeiten am Netz herangezogen.

Die Netzberechnung kann zur Berechnung des Störungsausmaßes oder zur Ermittlung der Folgen aufgrund einer Absperrmaßnahme in den Versorgungsnetzen zugezogen werden.

Zum Instandhaltungsprogramm, das meist ein Baustein eines ERP-System darstellt, empfiehlt es sich, eine Kommunikation herzustellen, die es erlaubt, die Abarbeitung der Tätigkeiten zur Instandsetzung (Personal, Dauer, Geräte, Material,...) zu dokumentieren.

Die Abbildung 4.10 zeigt das Zusammenspiel zwischen dem Störfallmanagement und den zur Abarbeitung desselben Prozesses notwendigen weiteren IT-Komponenten.

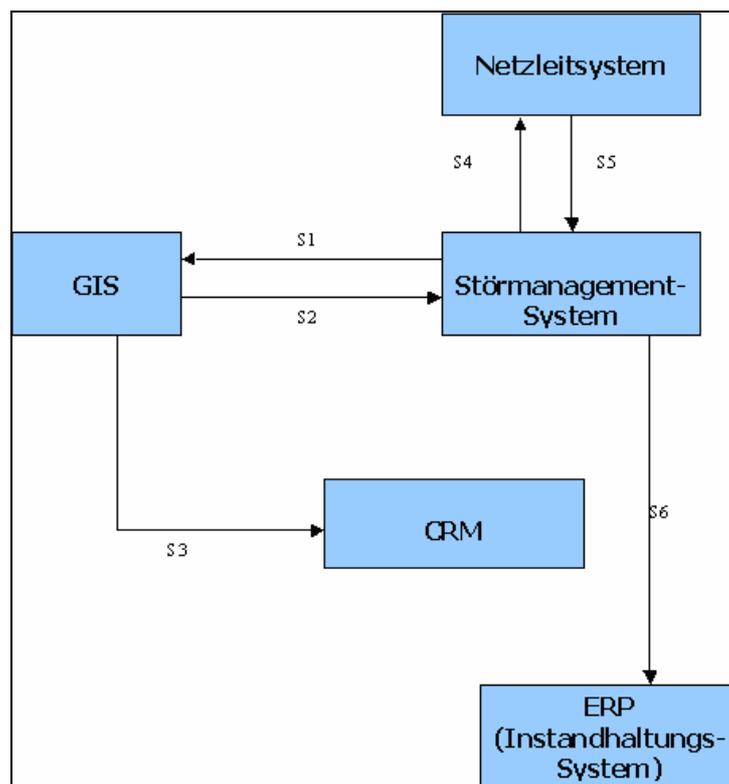


Abbildung 4.10: Schnittstellen vom/zum Störfallmanagement (DVGW, 2003)

Die Beschreibung der mit SX gekennzeichneten Schnittstellen wird auf den folgenden Seiten tabellarisch aufgeführt.

Schnittstelle	Von	Nach	Objekte	Beschreibung
S1	Störfallmanagement	GIS	Defektes Objekt - Adresse	Das defekte Objekt wird über seine objektspezifische Adresse identifiziert und im GIS visualisiert. Ist zum Zeitpunkt der Störungsmeldung eine eindeutige Zuordnung zu einem Betriebsmittel nicht möglich, erfolgt die Zuordnung zu einer Adresse. Im Laufe der Weiterbearbeitung der Störung erfolgt eine genauere Zuordnung und später eine eindeutige Zuweisung der Störung zu einem Betriebsmittel.
S2	GIS	Störfallmanagement	Absperrorgane	Im GIS ermittelte Sperrstrecken werden zusammen mit den Absperrorganen dem Störfallmanagement übergeben.
S3	GIS	CRM	Aufgrund der Absperrung betroffene Anschlussadressen	Durch Netzverfolgung ermittelte und wegen Sperrung unversorgte Anschlüsse werden im CRM genutzt, um ggf. Kunden zu informieren bzw. auf besonders kritische Kunden (z.B. Dialysestationen, etc.) hinzuweisen.
S4	Störfallmanagement	Netzleitsystem	Sperrstrecke Absperrorgane Kenndaten der Sperrung - Datum - Uhrzeit - Dauer	Information und Abstimmung über das Schalten der Sperrstrecke
S5	Netzleitsystem	Störfallmanagement	Freischaltungsbestätigung	Wurde von der Netzleitstelle eine Freischaltung durchgeführt (fernwirktechnisch oder manuell), so wird eine Freischaltungsbestätigung an das Störfallmanagement weitergeleitet. Erst ab dem Zeitpunkt des Eingangs der Freischaltungsbestätigung darf an der betroffenen Leitung gearbeitet werden.
S6	Störfallmanagement	IH-System	Defektes Objekt Sperrstrecke Absperrorgane Schadensbeschreibung	Übergabe der zur Abarbeitung anstehenden Tätigkeiten an die Instandsetzung.

Tabelle 4.5: Erläuterung der Schnittstellen nach DVGW (2003)

Abschließend sind in der Beschreibung des Hauptprozesses Störfallmanagement im Referenzmodell für GIS-gestützte Geschäftsprozesse die Optimierungspotentiale, die laut (DVGW, 2003) durch die GIS-Integration in diesem Arbeitsablauf erzielt werden können, dokumentiert.

Die Eingabe der Adresse wird erleichtert, da sie im GIS als Codeliste vorgehalten werden kann. Mehrfachmeldungen zu einer Störung können schneller erkannt werden. Ebenso wird das Erkennen von Störanfälligkeiten bestimmter Netzbereiche vereinfacht.

Das Auffinden und Darstellen selbiger Objekte auf Lage- oder Themenplänen kann durch GIS automatisch und schnell erfolgen. Zu diesen Objekten zählen Störungen, Absperrorgane, Hausanschlüsse sowie Ressourcen. Auch das automatische Erzeugen einer Adressliste zur Weiterverarbeitung beispielsweise als Serienbrief im Vorfeld einer Absperrung ist eine große Erleichterung für die Versorgungsunternehmen.

Durch die Visualisierung von Lage und Bearbeitungsstand der Störung wird die Ressourcenplanung erleichtert. Die Fahrwegoptimierung über GIS-Funktionalität „kürzester Weg“ kann ebenso als Nutzen gesehen werden.

#### 4.4.2 Beurteilung des Prozessmodells

Der DVGW unterstützt mit seinen technischen Mitteilungen die Versorgungsunternehmen bei ihren täglichen Aufgaben auch im Bereich der geographischen Informationsverarbeitung. Die Hinweise reichen von der GIS-Einführung über die Netzdokumentation bis hin zur Verbesserung von Geschäftsprozessen durch die Einbindung von GIS-Systemen. Das spiegelt einerseits den aktuellen Trend im Bereich der Prozessorientierung aus der allgemeinen Informatik wider und zeigt auf der anderen Seite die dringende Notwendigkeit der Eingliederung des GIS-Systems in diesen Bereich. Das DVGW-Referenzmodell ist derzeit in Deutschland das einzige seiner Art und erlangt durch die Mitwirkung von 12 DVGW-Mitgliedsunternehmen den Charakter einer Studie, die in diesem Kapitel beurteilt wird.

Sehr gut ist die inhaltliche Aufteilung sämtlicher Geschäftsprozesse eines Energieversorgungsunternehmens in Kern-, Haupt- und Teilprozesse. Diese Struktur greift über das gesamte Unternehmen und so lassen sich alle GIS-relevanten Prozesse herausfiltern und betrachten. Auch die Prozessbeschreibung an sich ist sehr strukturiert und detailliert aufgeführt. Durch die graphische Darstellung des Prozessablaufs werden Zusammenhänge auch gerade zu anderen Systemen bzw. Prozessen deutlich.

Im untersuchten Hauptprozess Störfallmanagement als Bestandteil des Kernprozesses „Betreiben und Instandhalten“ sind jedoch einige Ergänzungen erforderlich. Betrachtet man zunächst die Ziele, die den Hauptprozess Störfallmanagement von anderen Prozessen abgrenzen, so finden sich zwei Funktionalitäten wieder, die eher dem Bereich der Instandhaltung zuzuordnen sind. Die Verwaltung von Ressourcen (Mitarbeiter, Fahrzeuge) sowie die Verbesserung der Kapazitätsauslastung in störungsfreien Zeiten durch Einplanung von Routinearbeiten werden zwar vom Störfallmanagement berührt, haben ihren Schwerpunkt aber in der Instandhaltung und sind daher nicht im Störfallmanagement anzusiedeln. Die Ressourcen-Verwaltung findet ihren Einsatz in der Arbeitsvorbereitung, in der die täglichen Arbeiten geplant und dokumentiert werden. Bei dem Thema Kapazitätsauslastung stellen sich zwei Diskussionspunkte. Abhängig von Unternehmensgröße und -Struktur gibt es bei einigen Unternehmen ein oder mehrere Teams, die ausschließlich für Entstörungen zuständig sind. In diesem Fall ist eine Kapazitätsplanung im Zusammenhang mit weiteren Tätigkeiten notwendig. Wird jedoch für die Entstörung ein Team vorgehalten, das zu diesem Zeitpunkt einer geplanten Tätigkeit nachgeht, so ist zwar auch eine Überarbeitung des Arbeitsplans erforderlich, jedoch nicht im Sinne einer Verbesserung der Kapazitätsauslastung.

Aus den geschilderten Gründen ist das unter GIS-relevante Funktionen aufgelistete Management des Wachbetriebs und Bereitschaftsdienstes (Personal, Fahrzeuge, Tätigkeiten, etc.) für das Störfallmanagement an der Stelle nicht notwendigerweise korrekt eingeordnet.

Die Optimierungspotentiale, die durch die Einbindung von GIS im Störfallmanagement erreicht werden, sind vollständig und bedürfen lediglich einiger Randbemerkungen. Das Dokumentieren der Adresse der Störungsstelle wird zwar erleichtert, da nur die im GIS abgelegte Adresse ausgewählt wird. Um eine reibungsfreie Kommunikation mit anderen Prozessen zu gewährleisten, müssen sämtliche Adressdatensätze aus weiteren Anwendungen aufeinander abgestimmt bzw. abgeglichen werden. Ein solcher Adressenabgleich hängt auch von den unter-

nehmensinternen Abläufen ab und wird hier nicht weiter diskutiert. Die Erleichterung der Erkennung von Mehrfachmeldungen einer Störung wird gleichermaßen als Verbesserung aufgeführt. In diesem Zusammenhang ist es wichtig, die Beziehung von Störmeldung und Störung zu betrachten. Die Störmeldung wird im Referenzmodell nicht als eigenes Objekt sondern als Attribut zur Störung abgebildet. In diesem Fall ist es schwierig, die n:1-Beziehung zwischen Störmeldung und Störung zu dokumentieren. Es können durchaus für eine Störung mehrere Meldungen (Kunde A am Anfang der Straße und Kunde B am Ende der Straße) eingehen, die alle dokumentiert und für die Statistik ausgewertet werden müssen. So empfiehlt es sich, zunächst die Meldung zu erfassen und eine mit der Meldung verknüpfte Störung anzulegen. Alle weiteren eingehenden Meldungen werden erfasst und über die Topologie des Versorgungsnetzes wird geprüft, ob sie einer bereits dokumentierten Störung zugeordnet werden können.

Die Analyse und Priorisierung einer Störung hinsichtlich ihrer Dringlichkeit ist sicherlich ein Aspekt in der Ablaufbeschreibung, der kontrovers diskutiert werden kann. Die Erfassung kann über Maßnahmenkataloge erfolgen, die dann automatisch eine Bewertung der vorliegenden Informationen ausgeben. Unproblematisch ist dies sicherlich für einen Einsatzleiter oder einen fachlich versierten Mitarbeiter, der das Ergebnis dieser Beurteilung abschließend selbst beurteilen kann. Schwieriger wird es bei der Priorisierung bereits im Call-Center bzw. in der Störmeldeannahme, die nicht zwingend fachlicher Natur ist und somit der Hintergrund für eine Bewertung der automatischen Priorisierung nicht gegeben ist.

Bei der Einleitung von Maßnahmen zur Entstörung ist es sinnvoll, den Prozess Instandhaltung mit zu betrachten. Sobald eine Maßnahme notwendig wird, kommt die Arbeitsplanung samt Mitarbeitern, Geräten und Fahrzeugen mit dazu. Dies ist ein klassischer Bereich im Prozess Instandhaltung, so dass bei der Realisierung der beiden Prozesse über eine Weiterleitung der Störungsinformation in den Instandhaltungsprozess nachgedacht werden sollte. Es könnte beispielsweise eine automatische Erzeugung eines technischen Arbeitsauftrages mit dem Attribut „aus Störung“ sein, der dann weiter abgearbeitet wird und nach Abschluss die Information für die Fertigstellung an den Störungsprozess wieder zurück schickt. Nach diesem Vorgehen würde auch die Netzverfolgung und Sperrstreckenermittlung in diesen Bereich fallen.

Die dargestellte Schnittstellenthematik vom Prozess zu den weiteren benötigten Anwendungen, wie GIS, Netzleitsystem, CRM und ERP (Instandhaltungssystem) sind nicht stimmig im Referenzmodell dokumentiert. Die Kommunikation zwischen den Anwendungen stimmt nicht mit der Grafik und den detaillierten Schnittstellenbeschreibungen überein. Es wird eine Übersicht der Schnittstellen aufgeführt, die sich nicht mit der danach folgenden graphischen und tabellarischen Auflistung deckt. So werden die Programme Netzberechnung und Verbrauchsabrechnung in der Übersicht erwähnt, aber in der Beschreibung nicht weiter betrachtet. Umgekehrt erscheinen die Anbindungen zu CRM, Netzleitsystem und Instandhaltung nicht in der Übersicht. Für weitere Untersuchungen wird die Gesamtheit aller kommunizierenden Systeme betrachtet und dieser Punkt obsolet.

## 4.5 Aktuelle Marktsituation für Störfallmanagementlösungen

Für den Bereich Störfallmanagement existieren Softwarelösungen, die sich hinsichtlich Funktionalität und Architektur unterscheiden. Es folgt eine Auflistung der bekanntesten Module für das Störfallmanagement. Die Recherche wurde ausschließlich unter Firmen durchgeführt, die ihren Ursprung im Bereich der Geographischen Informationssysteme, also im technischen und nicht im kaufmännischen Bereich haben. Die zur jeweiligen Lösung zugehörige Betrachtung hinsichtlich Funktionalität und Architektur befindet sich in Anhang B: Beschreibung der existierenden Lösungen. Die Liste hat eine alphabetische Reihenfolge und basiert auf einer Internetrecherche mit Stand April 2007.

- ArcFM UT mit Utilities Business Package
- Geo EAM
- GeoXtension
- G!NIUS

- IAM
- Integriertes Störmanagement
- NOA
- TBM
- TOMS

Nahezu alle Lösungen arbeiten in einer EAI-Umgebung. Web Services werden nur von sehr wenigen unterstützt, wobei keine Aussagen zu Umsetzungen gemacht werden. Lediglich eine Softwarelösung arbeitet mit der Prozesssteuerungssprache BPEL zur Orchestrierung von Web Services. Das Pilotprojekt, als bisher einziges Projekt mit diesem Tool, gibt es für die Beleuchtungsinstandhaltung. Somit fehlt es an einer Lösung für Störungsmanagement, die sich der ausgewählten Variante zur Architektur (SOA) und den dazugehörigen Kernstandards bedienen.

## 4.6 Bewertung

Das beschriebene Referenzmodell zur Einbindung von GIS in Geschäftsprozesse von Energieversorgungsunternehmen ist in den Jahren 2002/2003 entstanden und bisher nicht weiter aktualisiert oder überarbeitet worden. Es existiert lediglich der Technische Hinweis GW 133 aus dem Jahr 2005, welcher bisher jedoch nicht zu einer Überarbeitung des GAWANIS-Datenmodells geführt hat. So haben sich Punkte informationstechnischer und inhaltlicher Art geändert, die in diesem Kapitel aufgearbeitet werden. Des Weiteren wird auf zu diesem Zeitpunkt erhältliche Softwarelösungen eingegangen und geschlussfolgert.

Beginnend mit der fachlichen Seite des Referenzmodells gibt es den Diskussionspunkt der Aufgabenverteilung zwischen den Modulen Instandhaltung (insbesondere die Arbeitsvorbereitung) und Störfallmanagement. Hier muss in zwei Fällen die Bearbeitungstiefe der Entstörung in den genannten Modulen getrennt betrachtet werden. Zunächst für große Energieversorgungsunternehmen und solche, die einen Entstördienst haben, der ausschließlich für diese Tätigkeit vorgesehen ist. In diesem Fall macht es Sinn, für den Entstördienst eine separate, im Störfallmanagement geführte Ressourcenverwaltung und Einsatzplanung vorzuhalten. Dies ist für kleinere Unternehmen und/oder solche, die Mitarbeiter aus den geplanten Tätigkeiten für Störungen abziehen nicht erforderlich. Hier ist es sinnvoll, diese Funktionalität aus dem Störfallmanagement hinaus in die Arbeitsvorbereitung der Instandhaltung zu verlagern, da in dem Modul ohnehin Ressourcen und Einsätze verwaltet und Routen geplant werden.

Des Weiteren sind bei der Betrachtung im Störfallmanagement die Störung und der Schaden als Objekt in der Datenbank abgelegt. Die Störmeldung existiert als Attribut der Störung, was jedoch zum heutigen Zeitpunkt nicht mehr sinnvoll erscheint, da es erforderlich ist, mehrere Meldungen zu einer Störung entgegen nehmen zu können. Auch resultiert nicht immer eine Störung aus einer Meldung, die Meldung an sich muss jedoch erfasst werden. Somit sollte die Störmeldung als eigenes Objekt im Datenmodell vorgesehen werden.

Auch nach (Hauffe, 2004) gibt es im Störungsmanagement ein Zusammenspiel zwischen den in Abbildung 4.11 dargestellten Komponenten, bei denen die Meldung der Störung ebenso eine höhere Rolle spielt.

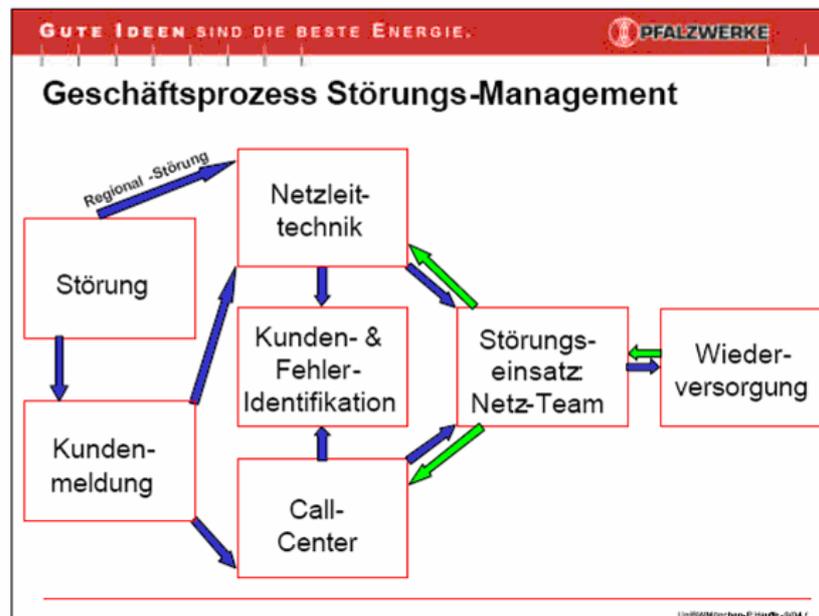


Abbildung 4.11: Geschäftsprozess Störungs-Management (Hauffe, 2004)

Eine durchaus diskussionswürdige Fragestellung, die im Referenzmodell angesprochen wird, ist die Erweiterung des GIS um prozessspezifische Aspekte im Störfallmanagement, wie beispielsweise die Dokumentation der Störung. Ein Geographisches Informationssystem um Funktionen zu erweitern, die nicht seiner klassischen Funktionalität entsprechen, macht in einem Zeitalter von modernen Kommunikationstechnologien zwischen Softwarekomponenten wenig Sinn, zumal die „erzwungenen“ Funktionen vermutlich hardcodiert und nachträglich, teilweise sogar auch umständlich programmiert werden müssten, weil eben solche Funktionalitäten in einem GIS-System nicht vorgesehen sind. Also wird eine Applikation oder Anwendung, die den Störfallprozess und auch weitere technische Arbeitsprozesse abbilden und mit dem GIS kommunizieren kann, notwendig.

Die unter Kapitel 4.2.2 beschriebenen GIS-Funktionalitäten sind nicht in Gänze für das Störungsmanagement relevant. Die größte Bedeutung besitzen die Funktionen Visualisierung, um die Störungen in der Karte zu lokalisieren, und Netzwerkverfolgung, zunächst im Sinne einer topologischen Auswertung, um verschiedene Beziehungen zwischen Objekten während der Entstörungsbearbeitung zu gewährleisten. Folgende Beziehung ist dabei von großer Wichtigkeit: Melder bzw. Abnahmestelle und Leitungsnetz, um das übergeordnete Netzobjekt bei der Störung zu finden und um eine Übersicht über das Ausmaß der Störung zu bekommen. Sind für den mobilen Einsatz keine entsprechenden Geräte vorgesehen, ist die Plottfunktionalität zur Kartenerstellung ebenfalls relevant. Die Touren- oder Routenplanung stellt eine wichtige Funktion für den Entstördienst bzw. die Arbeitstrupps aus der Instandhaltung dar. Voraussetzung hierfür sind mobile Geräte, die dies unterstützen, sowie eine entsprechende Berücksichtigung im Arbeitsprozess. Die mobile GIS-Unterstützung wird im Weiteren nicht detaillierter betrachtet.

Technische Themen, wie Datenstrukturen, Kopplungen und überbetrieblicher Datenaustausch entsprechen im Referenzmodell nicht mehr dem Stand der Technik, da zwischenzeitlich vom OGC und weiteren Organisationen einige Standards diesbezüglich anerkannt worden sind. So gibt es für das Speichern von Geodaten eine entsprechende OGC-Spezifikation (OGC Simple Features), die eine einheitliche Datenablage sowohl von Attributen als auch von den Geometrien regelt. Selbstverständlich muss die verwendete Datenstruktur dokumentiert und offen gelegt sein, so dass jederzeit auf die Informationen zugegriffen werden kann. Wichtige Daten für das Störfallmanagement sind primär die Kenntnisse über Kataster, das Leitungsnetz und die Kunden.

Die technologische Basis sollte, wie in Kapitel 0 beschrieben, durch Flexibilität und Wiederverwendbarkeit geprägt sein, so dass eine Service-orientierte Architektur mit Web Services das Ziel für eine Realisierungs Umgebung ist. Des Weiteren hat damit die Geoinformatik die Chance, sich der allgemeinen Informatik anzunähern. Entsprechend fällt auch die Bewertung der erhältlichen Lösungen für das Störfallmanagement aus. Die meisten

Tools arbeiten in der Kommunikation noch über XML-Schnittstellen in einer EAI-Umgebung. Nur sehr wenige bieten die Kommunikation über Web Services an. Nur eine Softwarelösung verwendet die Standards der Web Services mit BPEL und hier gibt es nur eine Pilotapplikation für den Bereich der Beleuchtungsinstandhaltung.

Erwähnenswert ist in diesem Zusammenhang ist die OWS-4 Initiative des OGC. Dort wird versucht die Geoinformatik mit den Standards SOAP, WSDL, UDDI und BPEL in Verbindung zu bringen. Bisher ist aus den 4 OWS-Initiativen jedoch nur ein discussion paper entstanden, das nicht den Bereich Störfallmanagement abdeckt und dessen Ergebnis nicht für die Öffentlichkeit zugänglich ist.

Bestreben ist es, (Teil-) Prozesse aus dem Störfallmanagement zu entwickeln, die das Fachliche aus dem Störfallmanagement abdecken und technologisch einen höchstmöglichen Grad an Wiederverwendbarkeit aufweisen, so dass sie beliebig weiter verwendet werden können.

## 5 Umsetzung des gewählten Ansatzes

In den folgenden Kapiteln wird der zur technischen Realisierung einer Geodatenanbindung in Geschäftsprozessen ausgewählte Ansatz mit Web Services in einer Service-orientierten Architektur prototypisch umgesetzt. Als Beispielprozess dient das Störfallmanagement aus dem Referenzmodell des DVGW. Der Prozess wird geringfügig modifiziert und die einzelnen Funktionspakete in Web Services unterteilt. Die Störungslokalisierung wird als BPEL-Prozess bestehend aus der Orchestrierung eines geographischen und eines nichtgeographischen Web Service umgesetzt.

### 5.1 Modifikation des Referenzmodells für Störfallmanagement

In diesem Kapitel wird ein modifiziertes Geschäftsprozessmodell für Störfallmanagement beschrieben, welches auf der Technischen Mitteilung GW 133 und dem entsprechenden Part im Referenzmodell für die Einbindung von GIS in Geschäftsprozesse basiert und zusätzlich Modifikationen aus den Bewertungen der vorherigen Kapitel beinhaltet. Das neue Modell ist teilweise konkreter und teilweise gröber strukturiert als die Eingangsmodelle und enthält Erweiterungen. Im bearbeiteten Modell läuft der Entstörungsprozess wie folgt ab:

Eine Störmeldung geht beim Energieversorgungsunternehmen ein. Dies kann viele Hintergründe haben. Eine Störmeldung kann durch einen Kunden eingehen oder einem Ereignis in der Instandhaltung oder Bauphase folgen. Die Meldung wird mit Störungsdaten nach GW 133 durch einen Mitarbeiter (meist aus der Leitwarte oder bei größeren Unternehmen aus dem Call-Center) erfasst, welcher die erforderlichen Informationen eingibt und im Anschluss den Störungsort anhand der Adresse unter Zuhilfenahme eines GIS lokalisiert. Daraufhin wird entschieden, ob es sich um eine Meldung zu einer bereits bekannten Störung oder um eine neue Störung handelt. Bei ersterem wird die Meldung mit der bestehenden Störung verknüpft. Bei letzterem wird ein neuer Störungsdatensatz generiert.

Dann erfolgen eine Störungsanalyse und eine Priorisierung der Störung in der Applikation für Störfallmanagement, die meist durch den Einsatzleiter vorgenommen wird. Somit werden sämtliche Informationen zur Störung an den Einsatzleiter weitergeleitet, welcher hinsichtlich weiterer Maßnahmen entscheidet. Zur Auswahl stehen hier die drei Varianten: keine Maßnahme erforderlich, Sofortmaßnahme erforderlich und Maßnahme terminierbar. Ist keine Maßnahme notwendig, so ist die Bearbeitung an dieser Stelle beendet und die Dokumentation wird geschlossen.

Bei einer Sofortmaßnahme informiert der Einsatzleiter umgehend sofern nötig die Gefahr abwehrenden Institutionen (Feuerwehr, Polizei,...) und den Entstördienst, der eine Erstsicherung vor Ort vornimmt. Dies ist besonders für das Medium Gas von entscheidender Bedeutung, da nach dem Regelwerk des DVGW seit Mitte 2004 vorgeschrieben ist, dass die Arbeiten zur Behebung einer Störung in der Gasversorgung in bebautem Gebiet innerhalb von 30 Minuten nach Eingang der Meldung aufgenommen sein müssen. Des Weiteren wird mit Benachrichtigung des Entstörtrupps automatisch eine Weiterleitung der Informationen an die Instandhaltung gesendet, wo wiederum automatisch ein technischer Arbeitsauftrag angelegt wird, den der Entstördienst nach oder während seiner Tätigkeit befüllt. Sind nach der Erstsicherung weitere Eingriffe erforderlich, geht der Arbeitsauftrag in die planbare Instandhaltung bzw. in die Arbeitsvorbereitung über und wird dort abgearbeitet und beendet. Ist eine terminierbare Maßnahme zu treffen, so wird eine Weiterleitung an einen Verantwortlichen im Funktionsbereich Instandhaltung initiiert, der einen Arbeitsauftrag mit entsprechenden Eigenschaften anlegt, welcher die Aktivitäten in der Arbeitsvorbereitung mit berücksichtigt. Bei sämtlichen Aktivitäten, die zur Abarbeitung dem Instandhaltungsbereich zugeordnet werden, ist eine Rückmeldung über die Fertigstellung bzw. Behebung der Störung erforderlich, um die Störung im System zu komplettieren.

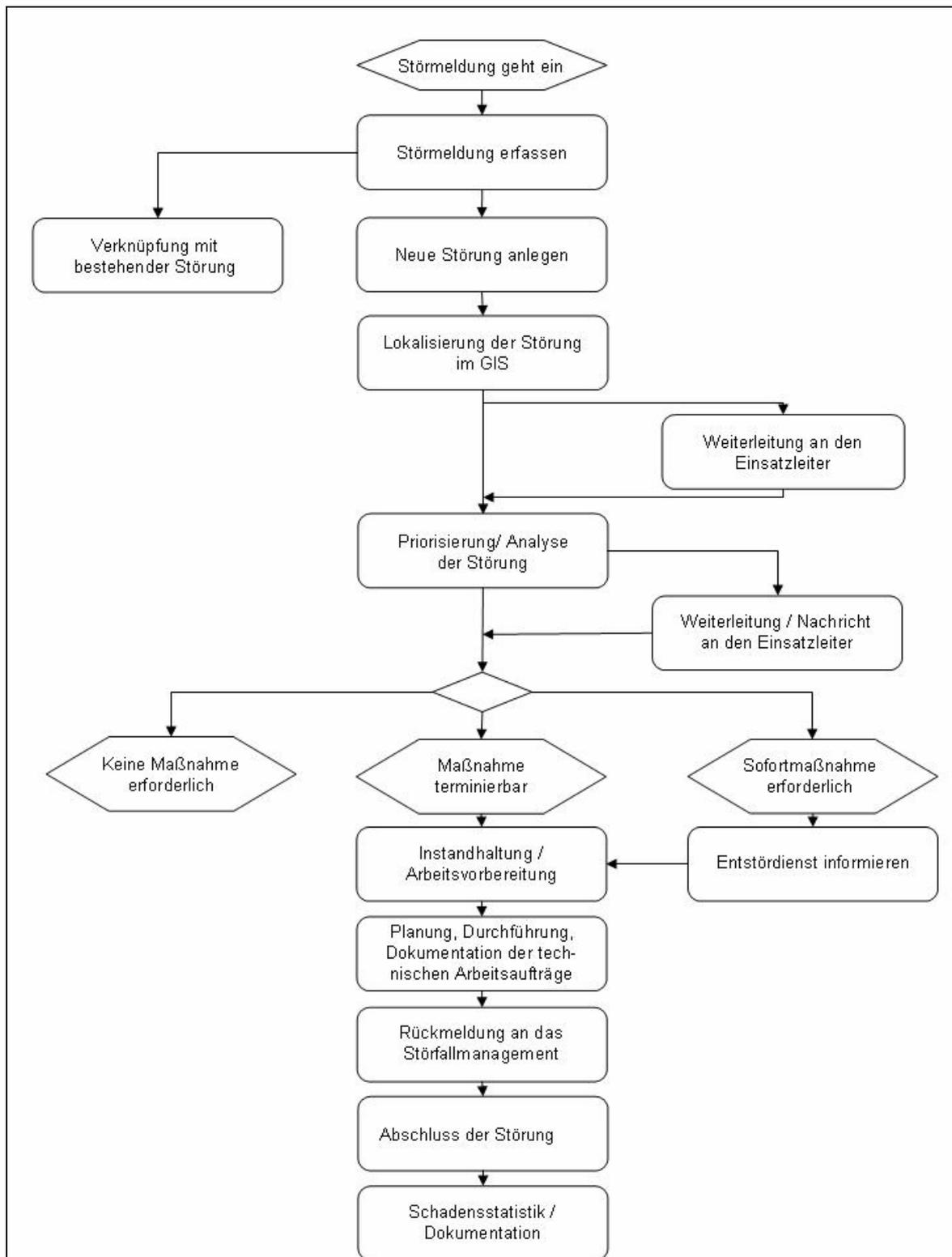


Abbildung 5.1: Ablauf des modifizierten Störfallprozesses

Der Rückmeldung zum Störfallmanagement folgt eine Überprüfung der Störungsdokumentation und nachdem alle Pflichtattribute gefüllt sind, kann der Vorgang abgeschlossen und archiviert werden bzw. für statistische Auswertungen zur Verfügung stehen. Die Überprüfung der Vollständigkeit der Dokumentation kann im Verlauf auch über Pflichtattribute automatisch erfolgen.

Der zuvor beschriebene Prozessablauf für das Störfallmanagement ist als Grafik in der Abbildung 5.1 zu sehen und dient als Basis für die weitere Bearbeitung und Untersuchung.

Der modifizierte Prozess Störungsmanagement ist in einigen Bereichen feiner und in anderen gröber gegliedert. Zu Prozessbeginn wird der Verlauf vom Eingang der Meldung bis zur Priorisierung der Störung feiner abgebildet, da im neuen Modell die Störungsmeldung an sich einen höheren Stellenwert bekommt, indem sie ein eigenes Objekt darstellt. Im weiteren Verlauf wird sie entweder einer bereits bestehenden Störung zugeordnet oder aus ihr wird eine neue Störung erzeugt. Die Option Informationen oder Bearbeitungszustände an weitere Mitarbeiter (z.B. den Einsatzleiter) zu senden, ist ebenfalls eine Neuerung im modifizierten Modell. Ergänzend soll an dieser Stelle angemerkt werden, dass in dem oben beschriebenen Prozessmodell sämtliche Tätigkeiten, wie beispielsweise Reparaturen planen, Sperrstrecken und unversorgte Kunden ermitteln und benachrichtigen, aus der Instandhaltung heraus über die Planung und Abarbeitung der Arbeitsaufträge durchgeführt werden. Diese Aktivitäten werden nicht im Störfallmanagement koordiniert oder abgearbeitet. Somit ist dieser Block in der Entstörung nur sehr grob dargestellt.

Das Zusammenspiel der verschiedenen Softwarekomponenten im veränderten Modell für Störfallmanagement zeigt Abbildung 5.2.

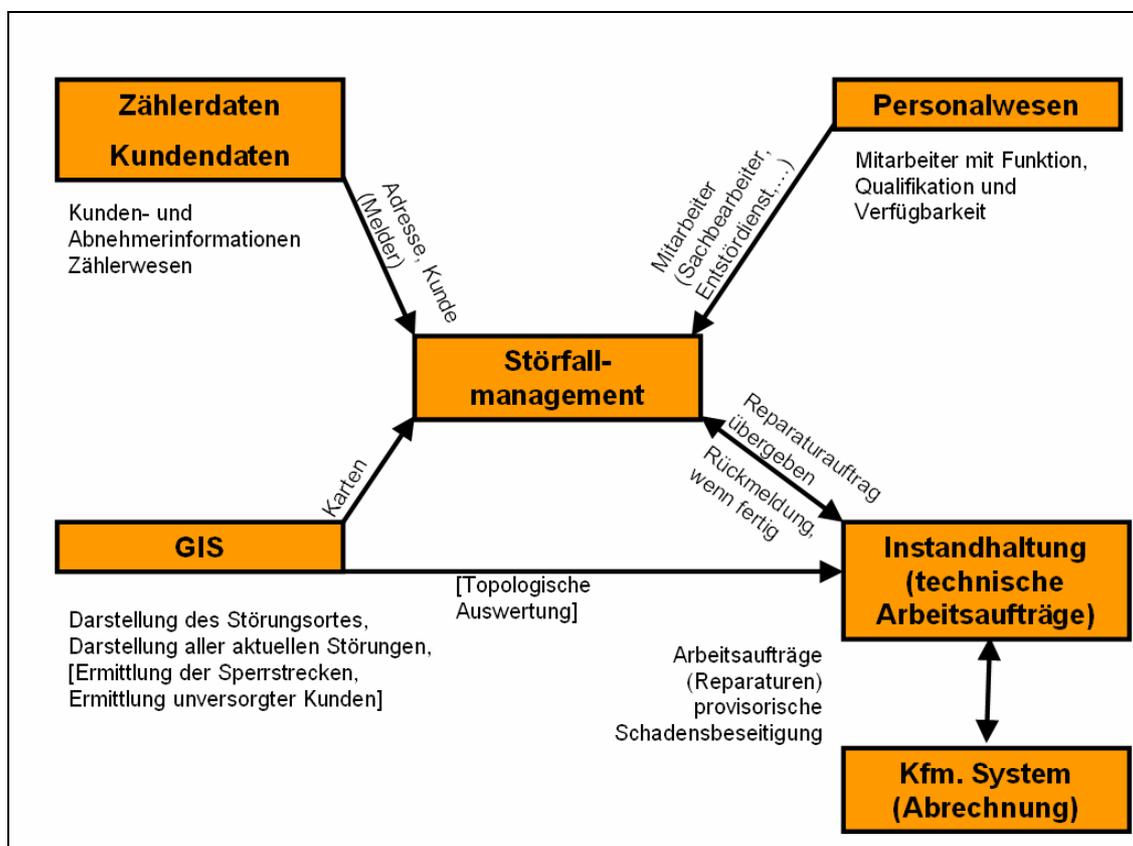


Abbildung 5.2: Zusammenhänge bei der Abarbeitung im Störfallmanagement

Die Verbindung vom GIS zur Instandhaltung ist eingeklammert dargestellt, ebenso die Funktionalitäten, die das GIS für diesen Bereich bereitstellt, das sie das Störfallmanagement nur indirekt betreffen. Großstörungen, die über das Netzleitsystem gemeldet werden, sind in der Grafik und im weiteren Verlauf nicht berücksichtigt.

Der modifizierte Störfallprozess wird im nächsten Kapitel in einzelnen Web Services unterteilt, welche verschiedene Funktionsbereiche im Gesamtkontext abdecken und vom Prozess an der entsprechenden Stelle angesprochen werden.

## 5.2 Unterteilung des Gesamtprozesses in Web Services

Um den Prozessablauf für Störfallmanagement in einer Service-orientierten Architektur abzubilden, wird der gesamte Prozess zunächst in einzelne Web Services, welche jeweils Teilfunktionalitäten abarbeiten, unterteilt. Bei dieser Einteilung wird hinsichtlich 2 Kriterien differenziert: unternehmensspezifische und prozessspezifische Services. Diese Unterteilung ist für die Umsetzung in die Praxis von großer Relevanz, da nur für prozessspezifische Services eine Allgemeingültigkeit über Standards erreicht werden kann. Des Weiteren lässt sich im Ablauf des Störfallmanagements erkennen, dass an verschiedenen Stellen die Interaktion mit einem Benutzer (Mitarbeiter der Leitwarte, Einsatzleiter, Entstördienst,...) notwendig ist. Da jedoch sowohl die Web Services als auch die Prozesssteuerungssprache BPEL für die Kommunikation von Maschinen entwickelt wurden, wird eine zusätzliche Sprach-Komponente erforderlich. Für den so genannten Human Task (Benutzerinteraktion im Prozess) wird die Spracherweiterung BPEL4people eingesetzt, die zurzeit als Whitepaper der Firmen SAP und IBM existiert. Das dritte Unterkapitel ist dem Bereich Benutzereingabe bzw. Human Task gewidmet.

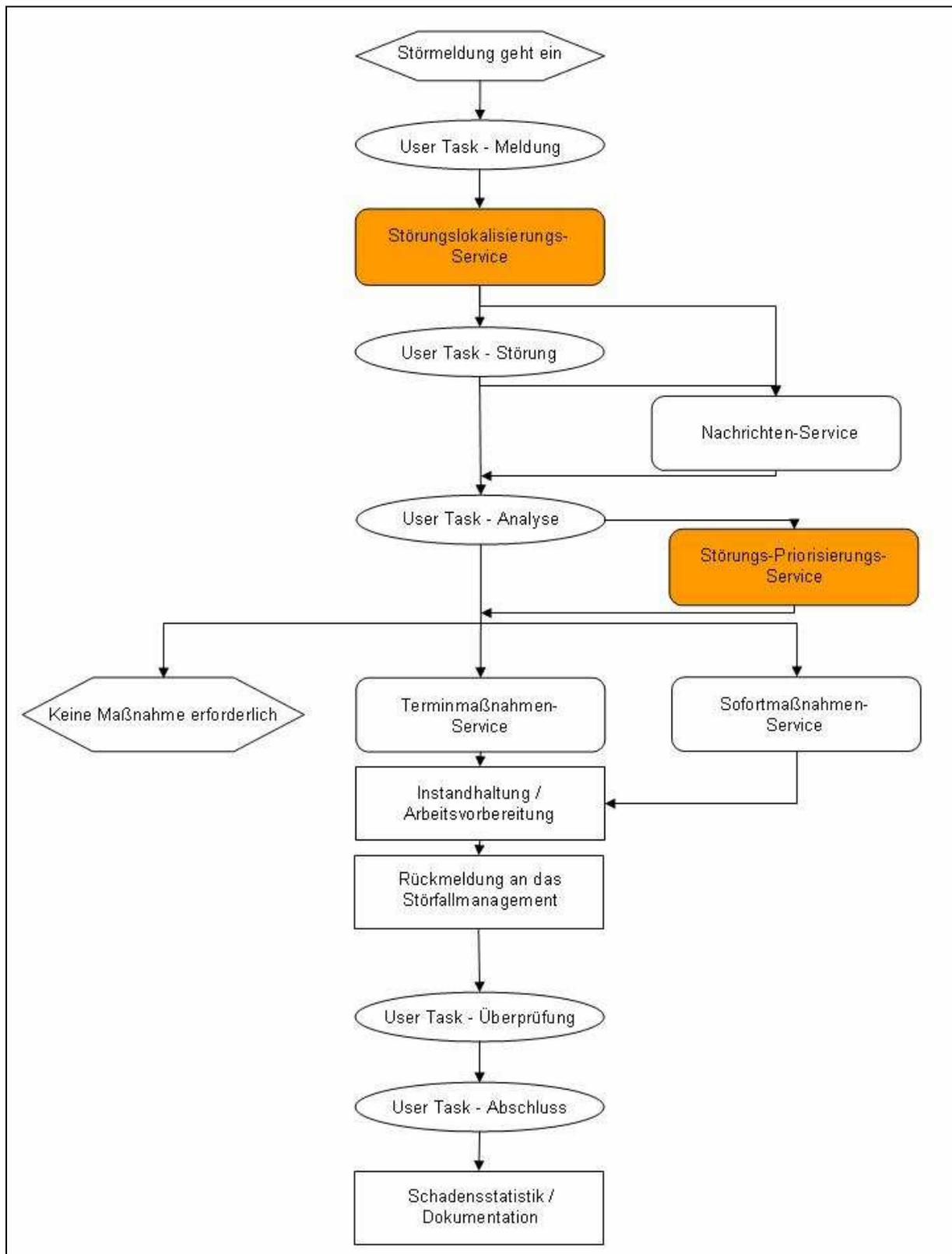


Abbildung 5.3: Ablauf modifiziertes Störungsmanagement mit Services

In der nachfolgenden Tabelle wird der oben dargestellte Ablauf inklusive der Funktionskomponenten näher beschrieben.

Typ	Name	Erläuterung
Ereignis	Störmeldung geht ein	Eine Störmeldung startet den Prozess Störungsmanagement und kann auf verschiedene Arten beim Unternehmen eingehen. Neben dem Anruf oder einer Benachrichtigung von Kunden bzw. Passanten können auch im Rahmen von Bautätigkeiten oder Instandhaltungsarbeiten Störungen an Betriebsmitteln auftreten.
Human Task	Meldung	Es wird eine neue Störungsmeldung angelegt und alle relevanten Daten (Uhrzeit, Inhalt, Lage, Art,...) dazu erfasst.
Web Service	Störungslokalisierungs-Service	Der Störungslokalisierungs-Service ermöglicht die Darstellung der aktuellen Störungsmeldung in einem Lageplan. Über die Adresse bzw. ggf. die Betriebsnummer, sofern bekannt, wird der Ort der Störung lokalisiert.
Web Service	Nachrichten	Je nach Organisationsstruktur des Unternehmens ist es notwendig, den Einsatzleiter über die aktuell vorliegende Störung zu informieren, welches an zwei Stellen im Ablauf erfolgen kann. Des Weiteren kann der Einsatzleiter selbst auch am Ablauf der Entstörung beteiligt sein, so dass er verschiedene User Tasks selbst durchführt.
Human Task	Störung	Der Mitarbeiter trifft die Entscheidung, ob die eingegangene Meldung einer bereits bestehenden Störung zugeordnet wird, oder ob ein neuer Datensatz für eine Störung generiert wird.
Human Task	Analyse	Anhand von Maßnahmenkatalogen, welche in Auswahlfeldern abgebildet sein können, wird die Störung hinsichtlich ihrer Dringlichkeit analysiert und priorisiert. Die Kombination der Auswahlfelder ist das Ergebnis und stößt die erforderlichen Services an.
Web Service	Störungs-Priorisierungs-Service	Falls im Prozess notwendig, werden durch diesen Web Service alle aktuellen Störungen im Übersichtsplan angezeigt, welches eine zusätzliche Option zur Priorisierung der Störung darstellt.
Ereignis	Keine Maßnahme erforderlich	Ist keine Maßnahme erforderlich, so endet der Störungsprozess an dieser Stelle.
Web Service	Terminmaßnahmen-Service	Dieser Web Service leitet sämtliche Informationen an die Arbeitsvorbereitung (Instandhaltung) weiter und generiert einen technischen Arbeitsauftrag zur Dokumentation der Entstörungstätigkeit (siehe weiter Modul Instandhaltung).

Typ	Name	Erläuterung
Web Service	Sofortmaßnahmen-Service	Ja nach Grad der Dringlichkeit informiert der Sofortmaßnahmen-Service die Feuerwehr (für Blaulichtfahrten) sowie umgehend den Entstördienst. Dies kann auf verschiedenen Wegen erfolgen. Des Weiteren wird ein technischer Arbeitsauftrag in der Instandhaltung angelegt, der vom Entstördienstmitarbeiter nach/während seiner Tätigkeit befüllt wird. Ist eine weitere Maßnahme erforderlich, so erfolgt eine Weiterleitung an die Arbeitsvorbereitung (Instandhaltung). Eine Rückmeldung an das Störfallmanagement kann zusätzlich auch nach Abschluss der Tätigkeit durch den Entstördienst erfolgen.
Modul	Instandhaltung/ Arbeitsvorbereitung	Im Modul Instandhaltung wird die Planung, Durchführung und Dokumentation sämtlicher Arbeiten an den Betriebsmitteln geführt. Dies beinhaltet auch die Ermittlung von Sperrstrecken und den daraus resultierenden unversorgten Kunden.
(Web Service)	Rückmeldung an das Störfallmanagement	Nach Abarbeitung des Entstörungsarbeitsauftrages wird eine Rückmeldung an das Modul Störungsmanagement geschickt.
(Web Service)	Sperrstrecken ermitteln	Die Sperrstrecken werden über eine Netzwerkverfolgung ermittelt und als Karte ausgegeben.
(Web Service)	unversorgte Kunden ermitteln	Über die Netzverfolgung können diejenigen Kunden ermittelt werden, die von einer Netzsperrung betroffen wären und nicht versorgt würden.  Durch Ermittlung von Absperralternativen kann untersucht werden, inwiefern die Versorgung spezieller Kunden durch Vornahme einer anderen Absperrmaßnahme gewährleistet werden kann.
Human Task/ Web Service	Überprüfung	Es wird überprüft, ob der Störungsprozess vollständig dokumentiert ist. Selbstverständlich kann diese Überprüfung auch anhand von Pflichtfeldern, die während des Ablaufes kontrolliert werden oder zum Schluss des Prozesses durch einen Automatismus durchgeführt werden.
Human Task	Abschluss	Der Vorgang der Entstörung wird abgeschlossen und als nicht manipulierbarer Datensatz in einer Datenbank zur Auswertung archiviert.
Anwendung	Schadensstatistik / Dokumentation	Aus den bestehenden Daten zu aufgetretenen Störungen können Analysen und Statistiken angefertigt werden, um die Strategien für verschiedene Bereiche (Instandhaltung, Materialverwendung,...) zu erstellen.

Tabelle 5.1: Beschreibung der Komponenten im modifizierten Störfallmanagement

### 5.2.1 Unternehmensspezifische Services

Bei den folgenden Services fließen unternehmensspezifische Faktoren mit in die Umsetzung ein, so dass hier keine Allgemeingültigkeit aufgestellt und diese Services nicht generell abgebildet werden können. Das bedeutet für diesen Teil der Anwendung für das Störfallmanagement, dass er entweder durch eine feste Programmierung des Herstellers oder durch Definition des Unternehmens erstellt bzw. umgesetzt wird. Die genannten Services werden hier nur aufgeführt, jedoch nicht weiter behandelt oder umgesetzt.

- **Terminmaßnahmen-Service**  
leitet die Informationen an die Instandhaltung bzw. Arbeitsvorbereitung weiter, legt einen zugehörigen neuen technischen Arbeitsauftrag an und gibt eine Rückmeldung an das Modul Störfallmanagement, wenn die Abarbeitung des technischen Arbeitsauftrages erfolgt ist.
- **Sofortmaßnahmen-Service**  
informiert umgehend den Entstördienst über ein noch festzulegendes Medium (SMS, Mail, Sprachmailbox,...) und legt für die auszuführende Tätigkeit einen entsprechenden technischen Arbeitsauftrag an, den der Entstördienst mit Daten füllt. Ist eine weitere Maßnahme notwendig, so wird automatisch diese Information an die Arbeitsvorbereitung weitergeleitet. Letzteres kann auch in einem eigenen Service abgebildet werden. In jedem Fall erfolgt ein Feedback an das Störungsmodul.
- **Sperrstecken-Service als Sub-Funktion in der Instandhaltung**  
ermittelt die Sperrstrecken anhand von Störungs- (oder Instandhaltungs-) Daten und gibt einen Plan bzw. eine Karte aus. Der Sperrstrecken-Service wird der Instandhaltung bzw. Arbeitsvorbereitung zugeordnet.
- **Unversorgte-Kunden-Service als Sub-Funktion in der Instandhaltung**  
ermittelt die unversorgten Kunden anhand von Störungs- (oder Instandhaltungs-) Daten und gibt eine Auflistung derer aus, welche wiederum an ein Textprogramm weitergeleitet werden kann und einen zugehörigen Serienbrief ausgibt. Der Unversorgte-Kunden-Service wird der Instandhaltung bzw. der Arbeitsvorbereitung zugeordnet.
- **Nachrichten-Services**  
die modulübergreifend agieren, zählen ebenso zu den unternehmensspezifischen Services, da sie mit unterschiedlicher Hard- und Software realisiert werden können.

### 5.2.2 Prozessspezifische Services

Bei den prozessspezifischen Services, handelt es sich um Services, die prozess- und nicht unternehmensabhängig sind, was bedeutet, dass ein Service bei jedem Unternehmen im Bereich Störfallmanagement genutzt werden kann. Die folgend aufgeführten Services werden aufgrund ihrer allgemeinen Gültigkeit näher betrachtet und auch umgesetzt. Bei dieser Betrachtung sind die prozessspezifischen Services mit den Services, die GIS-Daten nutzen, gleichzusetzen.

- **Störungslokalisierungs-Service**  
zeigt die aktuelle Störung auf einer Karte an, die anhand einer Adresse lokalisiert wird.
- **Störungs-Priorisierungs-Service (optional)**  
zeigt alle aktuellen Störungen des Netzes auf einer Karte an und markiert die zeitlich letzte Störung.

### 5.2.3 Human Task

Die Benutzereingaben werden im Prozessmodell über den so genannten Human Task abgewickelt. Es werden fünf Interaktionen mit dem Benutzer wie folgt durchgeführt.

- **Human Task – Meldung**  
der Mitarbeiter legt einen neuen Datensatz für eine Störungsmeldung an und trägt alle erforderlichen Informationen (Uhrzeit, Inhalt, Lage, Art,...) ein.
- **Human Task – Störung**  
der Mitarbeiter entscheidet, ob es sich um eine Meldung zu einer bestehenden Störung oder um eine Meldung zu einer neuen Störung handelt und agiert entsprechend, d.h. er ordnet im ersten Fall die Meldung einer existierenden Störung zu und legt im zweiten Fall einen neuen Datensatz für eine Störung an.
- **Human Task – Analyse**  
der Mitarbeiter oder Einsatzleiter stößt die Analyse bzw. Priorisierung der Störung über das Ausfüllen von Maßnahmenfeldern oder Katalogen an.
- **Human Task – Überprüfung (optional)**  
der Mitarbeiter überprüft, ob alle Informationen ordnungsgemäß abgelegt und der Vorgang korrekt abgearbeitet worden ist. Dieser User Task kann durch eine automatische Überprüfung anhand von Pflichtattributen ersetzt werden.
- **Human Task – Abschluss**  
der Mitarbeiter schließt den Vorgang der Störung und löst somit den Archivierungsvorgang aus, der die Informationen für statistische Zwecke zur Verfügung hält.

Die Abbildung mit Human Tasks wird im weiteren Verlauf nicht näher betrachtet, da es sich bei der Erweiterung BPEL4people noch nicht um einen anerkannten Standard handelt.

## 5.3 Verwendete Software-Komponenten

Für die Umsetzung des Prozesses Störfallmanagement sind verschiedene Softwarekomponenten erforderlich. Dieses Kapitel beschreibt alle notwendigen Softwarekomponenten, die zur Implementierung für den Störfall-Prozess notwendig sind.

Um Geschäftsprozesse in der EDV durchführen zu können werden eine entsprechende Hard- und Softwareumgebung benötigt. Eine funktionierende Hardware wird an dieser Stelle vorausgesetzt und nicht weiter betrachtet. Für die Softwarelandschaft sind Komponenten erforderlich, um die Service-orientierte Architektur mit Web Services aufzubauen. Somit werden Softwareprodukte benötigt, die Web Services erstellen, registrieren, suchen, orchestrieren und nutzen können. Da es sich bei der Umsetzung des Prozesses Störfallmanagement um einen unternehmensinternen Prozess ohne Anbindung an weitere Geschäftspartner oder ähnliches handelt, wird der Verzeichnisdienst nicht in die weitere Betrachtung mit einbezogen. Des Weiteren wird eine Softwarelösung zur raumbezogenen Datenhaltung und Darstellung sowie zur Generierung von Informationen nach der WMS-Spezifikation gebraucht. Die Erfassung von Geodaten spielt in diesem Prozess keine Rolle und erfordert somit auch keine EDV-Unterstützung.

Beginnend mit der Geokomponente ist eine Datenbank erforderlich, in der sämtliche Geodaten, Grafik und Attribute, abgelegt werden können. Außerdem wird ein Visualisierungstool zum Betrachten der GIS-Daten sowie deren Aufbereitung und Bereitstellung nach der WMS-Spezifikation erforderlich. Für die Verarbeitung der

„Nicht-Geodaten“ sind eine Entwicklungsumgebung zur Erstellung und Bereitstellung der Web Services sowie eine Softwarekomponente zur Orchestrierung der einzelnen Services zu einem Geschäftsprozess nötig.

Zusammenfassend werden demnach ein Datenbankmanagementsystem samt raumbezogener Datenhaltungskomponente, eine Visualisierungskomponente für die Geodaten, eine Komponente für die Webfähigkeit dieser Daten, ein Applikations-Server zur Bereitstellung aller Daten, eine Entwicklungsumgebung für die Erstellung der Web Services sowie eine Komponente zur Orchestrierung der Web Services zu einem Geschäftsprozess benötigt.

Für die genannten Softwaregebiete kommt ein Gesamtpaket der Firma Oracle zum Einsatz. Die Datenbank Oracle 10g samt der Spatial-Komponente wird zur Datenhaltung eingesetzt. Die Visualisierung und Bereitstellung der Daten fürs Web nach der WMS-Spezifikation erfolgt durch den Oracle Application Server samt dem zusätzlichen Tool MapViewer. Als Entwicklungsumgebung bietet sich der Oracle JDeveloper an, der als Plug-In den Oracle BPEL-Designer zur Orchestrierung der Web Services enthält und das Verbindungsglied zwischen den einzelnen Softwarekomponenten darstellt.

Die Gründe für den Einsatz der Oracle-Software sind vielfältig. Zunächst gilt Oracle als der größte Datenbankhersteller mit nahezu 40% Marktanteil (Heise, 2005). Neben dem Aspekt, dass alles aus einer Hand kommt, sind doch sämtliche Pakete im Internet als Download für Testzwecke frei erhältlich. Der JDeveloper ist seit 2005 sogar generell kostenfrei (Oracle, 2005b). Des Weiteren bietet Oracle bisher als einziger Datenbankhersteller mit dem MapViewer ein Tool an, das die Visualisierung von Geodaten auch ohne GIS ermöglicht. Im Oktober 2006 hat Oracle die SOA Suite Release 3 auf den Markt gebracht, die sämtliche für eine Service-orientierte Architektur notwendigen Komponenten bereits beinhaltet. Der darin integrierte BPEL Process Manager wurde nach einer Fachstudie zum Vergleich von BPEL Laufzeitumgebungen als kommerzielles Produkt am Besten bewertet (Hantschel, Ruf, Strotbek, 2006).

### 5.3.1 Oracle Database 10g

Im Rahmen dieser Arbeit wird die Oracle Database 10g in der Version 10.1.2 zur Speicherung der Geodaten verwendet. Es werden sowohl Geometrien als auch Attributdaten in Oracle abgelegt. Außerdem benötigt das Tool MapViewer von Oracle, das zur Darstellung der GIS-Daten verwendet wird, eine Anbindung an die Oracle Datenbank.

Die Ablage von räumlichen Daten ist in Kapitel 3.2.2 ausführlich beschrieben, so dass an dieser Stelle darauf verwiesen wird.

Der folgende Screenshot zeigt die Tabelle Hauskoordinaten im Benutzer Kataster.

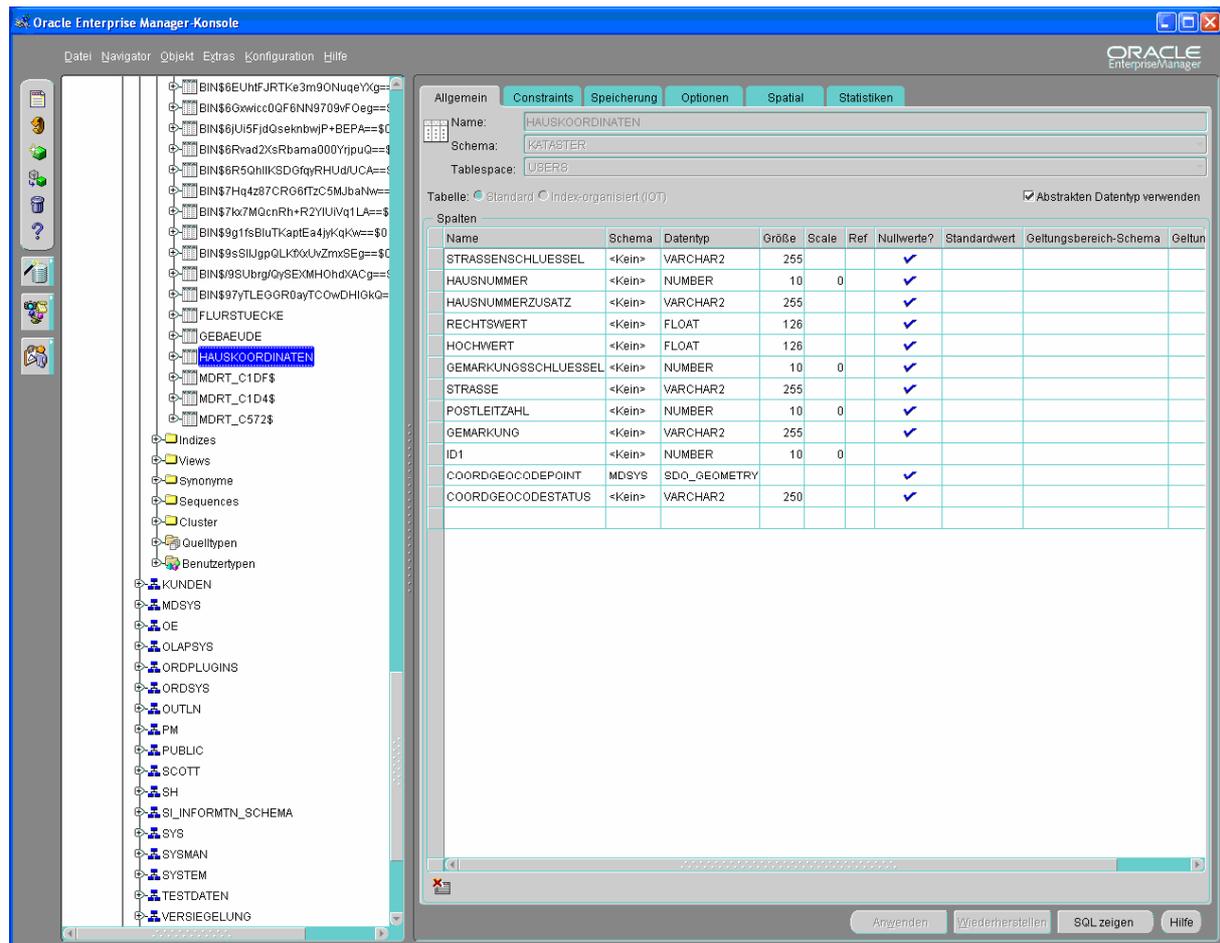


Abbildung 5.4: Tabellenansicht (Hauskoordinaten) in der Oracle-Datenbank

### 5.3.2 Oracle SOA Suite 10g

Oracle stellt die Oracle SOA Suite als umfassende und zur Laufzeit integrierbare Software Suite dar, die es ermöglicht, eine Service-orientierte Architektur aufzubauen, bereit zu stellen und zu managen. Hierzu zählen die Entwicklung von Service-orientierten Applikationen, deren Integration mit IT-Systemen sowie die Prozess-Orchestrierung von Services und Human Workflow. Die SOA Suite kann in einer heterogenen Umgebung installiert werden, so dass eine Service-orientierte Architektur auch schrittweise erreicht werden kann. Sie besteht als Best-of-breed-Lösung aus einer Komposition von verschiedenen Werkzeugen, die sich jeweils für eine Teilfunktion am Besten eignen. Im Folgenden werden die Komponenten aufgelistet, die in der nächsten Abbildung im graphischen Zusammenhang gebracht werden (Oracle, 2006b), (Oracle, 2006d).

- Integrated Service Environment (ISE) zur Entwicklung von Services
- BPEL Process Manager zur Orchestrierung von Services zu Geschäftsprozessen
- Enterprise Service Bus (ESB) zur Verbindung von bestehenden IT-Systemen und Geschäftspartnern
- Oracle Business Rules, um dynamische Entscheidungen zur Laufzeit zu treffen
- Oracle Application Server Business Activity Monitoring zur Überwachung von Services und verschiedenen Ereignissen sowie die Bereitstellung von Echtzeit-Informationen zum Stand des Unternehmens, der Geschäftsprozesse, Mitarbeiter und Systeme
- Web Services Management zur Verwaltung sämtlicher Informationen zu Authentifizierung und Autorisierung

- Web Services Registry (UDDI) zur Regelung des Lebenszyklus eines Web Services
- Oracle Application Server zur Bereitstellung einer Java 2 Enterprise Edition (J2EE) Umgebung für sämtliche J2EE-Applikationen

Die für die Umsetzung des Störfallmanagement notwendigen Komponenten sind der Application Server, der BPEL Process Manager und der JDeveloper. Diese drei Komponenten der SOA Suite werden in den folgenden Kapiteln näher erläutert. Verwendet wurde die SOA Suite 10g im Release 3.



Abbildung 5.5: Architektur der Oracle SOA Suite (Oracle, 2006b)

### 5.3.3 Oracle Application Server 10g

Der Oracle Application Server ist ein standardbasierter Applikations-Server, der eine umfassende und integrierte Plattform für den Betrieb von Webseiten, J2EE-Applikationen und Web Services bietet (Oracle, 2006c). Verschiedene Standards, die in einer Service-orientierten Architektur eine Rolle spielen, werden vom Oracle Application Server unterstützt: WS-Reliable Messaging, WS-Security, WS-Federation, Web Services Metadata, WSIF and REST web services. Die Oracle SOA Suite ist nicht notwendigerweise an den gleichnamigen Applikations-Server gebunden, sondern kann auch mit anderen führenden Servern zusammenarbeiten (Oracle, 2006d).

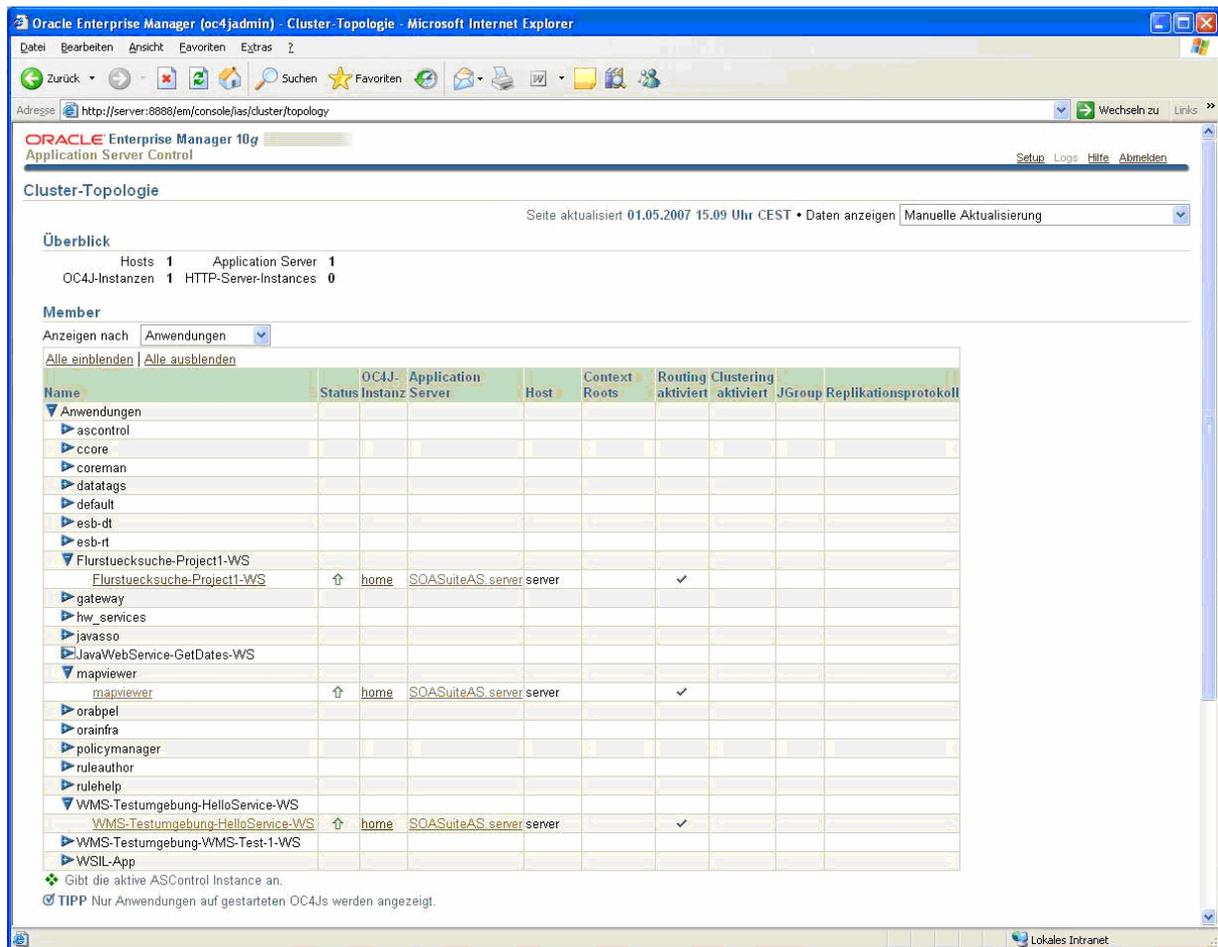


Abbildung 5.6: Oracle Application Server mit laufenden Anwendungen

Eine wichtige Komponente des Application Servers ist der MapViewer, der mit zur Auswahl der Oracle-Technologien beigetragen hat und dem das folgende Kapitel gewidmet ist.

### 5.3.4 Oracle Application Server 10g MapViewer

Der Oracle Application Server 10g Map Viewer, im Folgenden abgekürzt mit MapViewer, ist eine Java-Komponente als Add-On zum Oracle Application Server, die zusätzlich installiert und konfiguriert wird. Der MapViewer dient zur Darstellung raumbezogener Informationen. In Java entwickelt läuft das Tool in einer J2EE-Umgebung (Oracle, 2005a) und unterstützt die WMS-Spezifikation des OGC aktuell in der Version 1.1.1. Nach dem „Compliance Testing“ darf der MapViewer ab der Version 10.1.2 den Titel „OGC Compliant“ tragen. Die Requests GetCapabilities, GetMap und GetFeatureInfo werden unterstützt. Die optionale Funktionalität des Styled Layer Descriptors ist in diesem Release noch nicht bereitgestellt, ebenso die Funktion als Cascading Map Server (Oracle 2006a).

Alle Anfragen an den MapViewer werden mit der HTTP POST Methode gesendet, wobei die Requests sowohl Kartenanfragen als auch administrative Anfragen, wie das Auflisten von vordefinierten Karten aus der Datenquelle sein können. Der Inhalt der Requests wird in XML verpackt. Hat der MapViewer eine Kartenanfrage erhalten, so holt er sich alle relevanten räumlichen Daten sowie die Metadaten zur Darstellung aus der Datenbank und gibt als Antwort eine Karte aus, die in einem Standard-Browser aufgerufen werden kann. Die folgende Abbildung zeigt die MapServer Architektur (Oracle, 2005a).

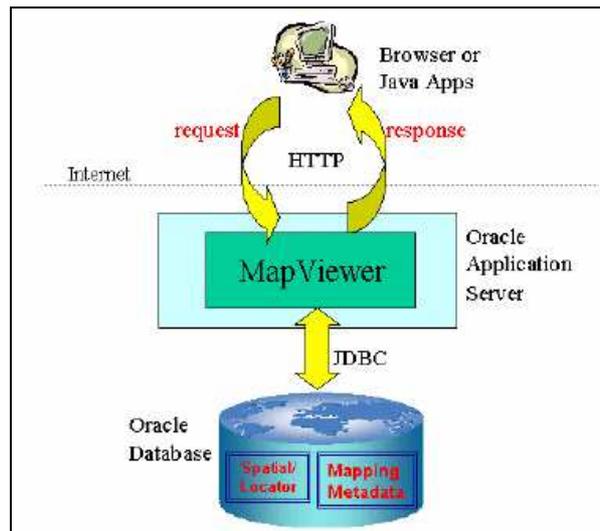


Abbildung 5.7: MapViewer Architektur (Oracle, 2005a)

Der MapViewer besteht aus den folgenden Bestandteilen (Oracle, 2006a):

- einer Java Bibliothek als Core Rendering Engine (SDOVIS), die die kartographische Darstellung generiert. Ein Servlet wird bereitgestellt, um die Funktionen im Web zu publizieren.
- einem Set von API's (application programming interfaces), die einen programmierbaren Zugang zu den MapViewer-Features erlauben. Diese API's beinhalten XML, Java, PL/SQL und ein AJAX-basierendes Javascript API.
- ein Map Builder-Tool, das die Erstellung von Symbolen und die Definition von Darstellungsregeln sowie das Editieren und Erstellen von MapViewer-Objekten erlaubt (siehe Abbildung 5.8).
- Oracle Map, das Map Cache und FOI (feature of interest) Server beinhaltet, die die Entwicklung von interaktiven räumlichen Web Applikationen erleichtern.

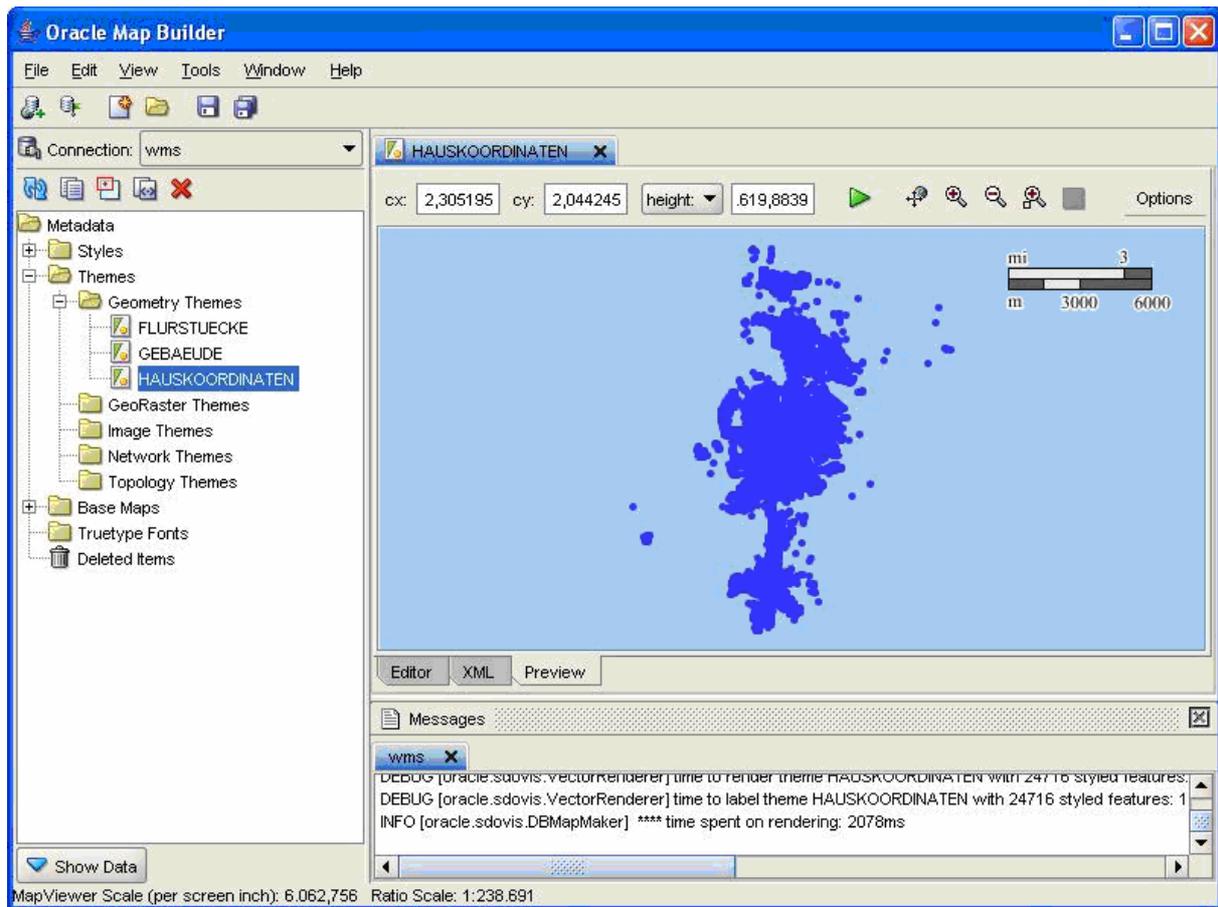


Abbildung 5.8: Screenshot Map Builder-Tool

Im Rahmen des Prozesses Störfallmanagement wird der MapViewer in der Version 10.1.3.1 zur Erstellung sämtlicher Karten verwendet.

### 5.3.5 Oracle BPEL Process Manager

Der Oracle BPEL Process Manager ist eine Laufzeit- und Steuerungsumgebung für BPEL-Prozesse, die in Java entwickelt wurde und dementsprechend auf allen Applikations-Servern läuft, die den J2EE-Standard unterstützen. Vier Komponenten, die in der folgenden Abbildung gezeigt werden, bilden den Oracle BPEL Process Manager:

- der BPEL Designer,
- der BPEL Server,
- die BPEL Console und
- eine Datenbank.

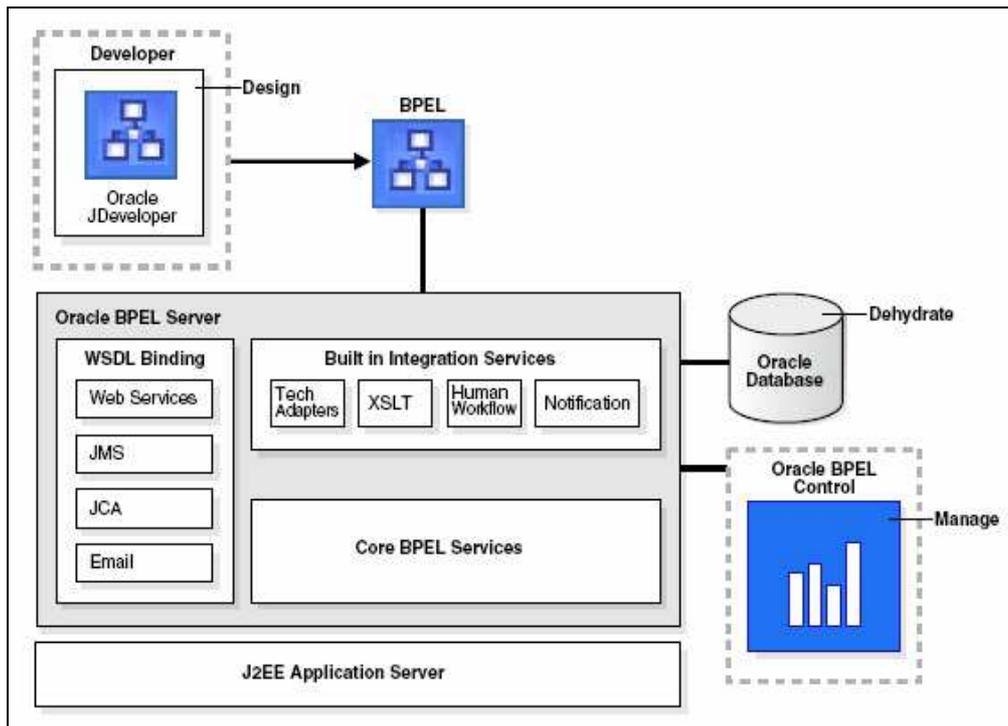


Abbildung 5.9: Oracle BPEL Process Manager (Oracle 2006c)

Der BPEL-Designer ist für die Implementierung der BPEL-basierten Prozesse zuständig und stellt graphische und benutzerfreundliche Möglichkeiten zur Erstellung der Prozesse zur Verfügung, so dass der BPEL-Code nicht komplett von Hand geschrieben werden muss. Wahlweise ist er in den Softwareprodukten Eclipse (ab Version 3.x) oder Oracle JDeveloper als Plug-In integriert (Juric, et al., 2006).

Der BPEL-Server stellt die Laufzeitumgebung zur Bereitstellung, auch Deployment genannt, und Ausführung der BPEL-Prozesse bereit. Neben BPEL werden noch weitere Web-Service-Technologien unterstützt, wie beispielsweise WS-Security. Er besteht aus einer Core BPEL Engine, dem eigentlichen Kern, der die oben genannten Funktionalitäten bereitstellt, den WSDL Bindings, die für die Kommunikation außerhalb von SOAP (z.B. Email) eine wichtige Rolle spielen, und Integration Services, die eine Transformation der im BPEL-Prozess ausgetauschten XML-Dokumente ermöglichen (Juric, et al., 2006).

Die Verwaltung, das Monitoring und Debugging der Prozesse übernimmt die BPEL Console, als webbasiertes Interface zur Überwachung der deployten BPEL-Prozesse. Dazu zählen im Einzelnen die visuelle Darstellung des Prozessablaufes, Kontrollmöglichkeiten, Fehlersuche- und Beseitigung, Prozesshistorie und das Managen von BPEL-Domänen und deren Konfiguration (Juric, et al., 2006).

Die Datenbank wird vom BPEL Server genutzt, um BPEL Schemata und die Zustände der Prozesse abzulegen. Dies wird insbesondere bei asynchronen Prozessen verwendet, bei denen ein Partner einen Web Service aufgerufen hat und auf die Antwort gewartet wird. Während der Wartezeit wird der Prozesszustand in der Datenbank abgelegt, um Server Ressourcen freizugeben. Dies wird, wie in der folgenden Abbildung demonstriert, als Dehydration bezeichnet. Erhält die Core BPEL Engine die Antwort, werden die notwendigen Informationen zur weiteren Verarbeitung wieder aus der Datenbank geholt, was als Hydratation benannt wird (Juric, et al., 2006). Die folgende Abbildung zeigt das Zusammenspiel der Komponenten im Oracle BPEL Process Manager, der in dieser Arbeit in der Version 10.1.3.1.0 zum Einsatz kommt.

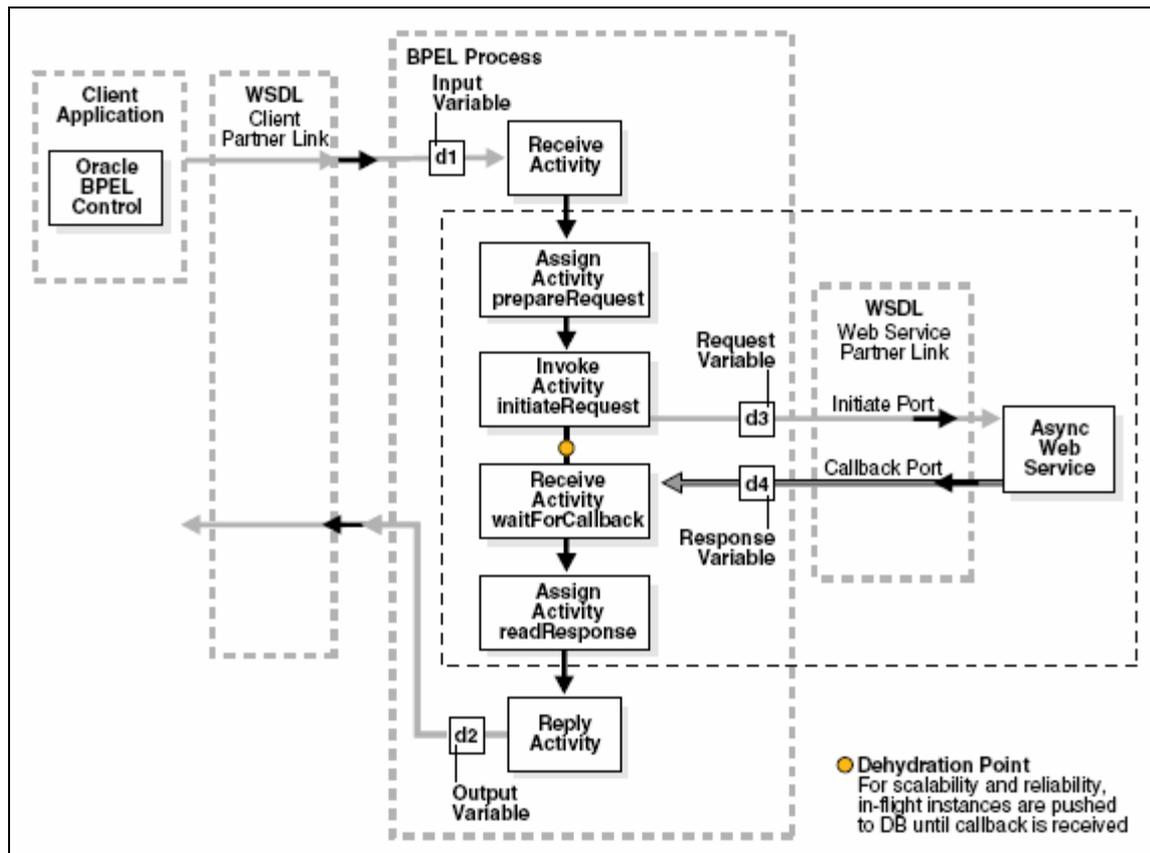


Abbildung 5.10: BPEL-Prozess mit Aufruf eines asynchr. Web Services (Oracle, 2006d)

### 5.3.6 Oracle JDeveloper 10g

Der JDeveloper von Oracle stellt die Entwicklungskomponente der SOA Suite als Java-Entwicklungsumgebung (integrated development environment IDE) dar (Oracle, 2007b) und bietet Funktionalitäten, um Service-basierte Applikationen zu modellieren, zu erstellen, aufzubauen, zu orchestrieren, testen, bereitzustellen und zu verwalten. Der JDeveloper unterstützt die SOA-Prinzipien und XML-Web-Service-Standards sowie traditionelle Java, J2EE und PL/SQL Komponenten (Oracle, 2006d).

Applikationen, die mit dem JDeveloper entwickelt wurden, können an jede JDBC-Datenquelle angebunden werden und laufen auf jedem J2EE-kompatiblen Applikations-Server. Die aktuellen Java-Standards (J2SE 5.0, J2EE 1.4, EJB 3.0) werden in der verwendeten Version 10.1.3.1. unterstützt. Der JDeveloper selbst ist ein Java-basierendes Tool, das auf verschiedenen Betriebssystemen (Windows, Linux, Mac, Unix,...) aufsetzen kann. Die Entwicklung kann je nach Benutzer unterschiedlich ablaufen. Es wird neben der reinen Code-Eingabe auch ein visuelles Benutzerinterface mit Drag-and-Drop-Funktion unterstützt und auf Wunsch können auch Assistenten zum Einsatz kommen. Diese erzeugen automatisch Metadaten wie beispielsweise ein WSDL-Dokument. Bei der Erstellung von Web Services lässt der JDeveloper zwei Varianten zu, die Bottom-Up und die Top-Down-Variante. Die Top-Down-Entwicklung basiert auf einem vorhandenen WSDL-File und generiert die für einen Web Service notwendigen weiteren Dokumente. Der Bottom-Up-Ansatz benötigt als Basis für die Web-Service-Entwicklung Java-Klassen, Remote Interfaces von EJB 2.0 bzw. 2.1 oder ADF Business Components Service Session Beans Wrapped als EJBs (Oracle, 2007b), (Oracle, 2006d).

Im Weiteren können mit dem JDeveloper Verbindungen zu anderen Softwarekomponenten wie beispielsweise Applikations-Server und Datenbank, definiert werden. Wobei diese Verbindungen nicht zwingend an Oracle-Produkte gebunden sind. Auch für diese Arbeitsschritte werden Assistenten angeboten, die abschließend einen Verbindungstest anbieten und falls notwendig eine Fehlerbehebung ermöglichen.

Die folgende Abbildung zeigt ein BPEL-Projekt im JDeveloper. Die vierteilige Ansicht zeigt die Projektstruktur im Application Navigator, die Dokumentstruktur des BPEL-Dokumentes, die graphische Ansicht des BPEL-Projektes sowie ein Log-Fenster.

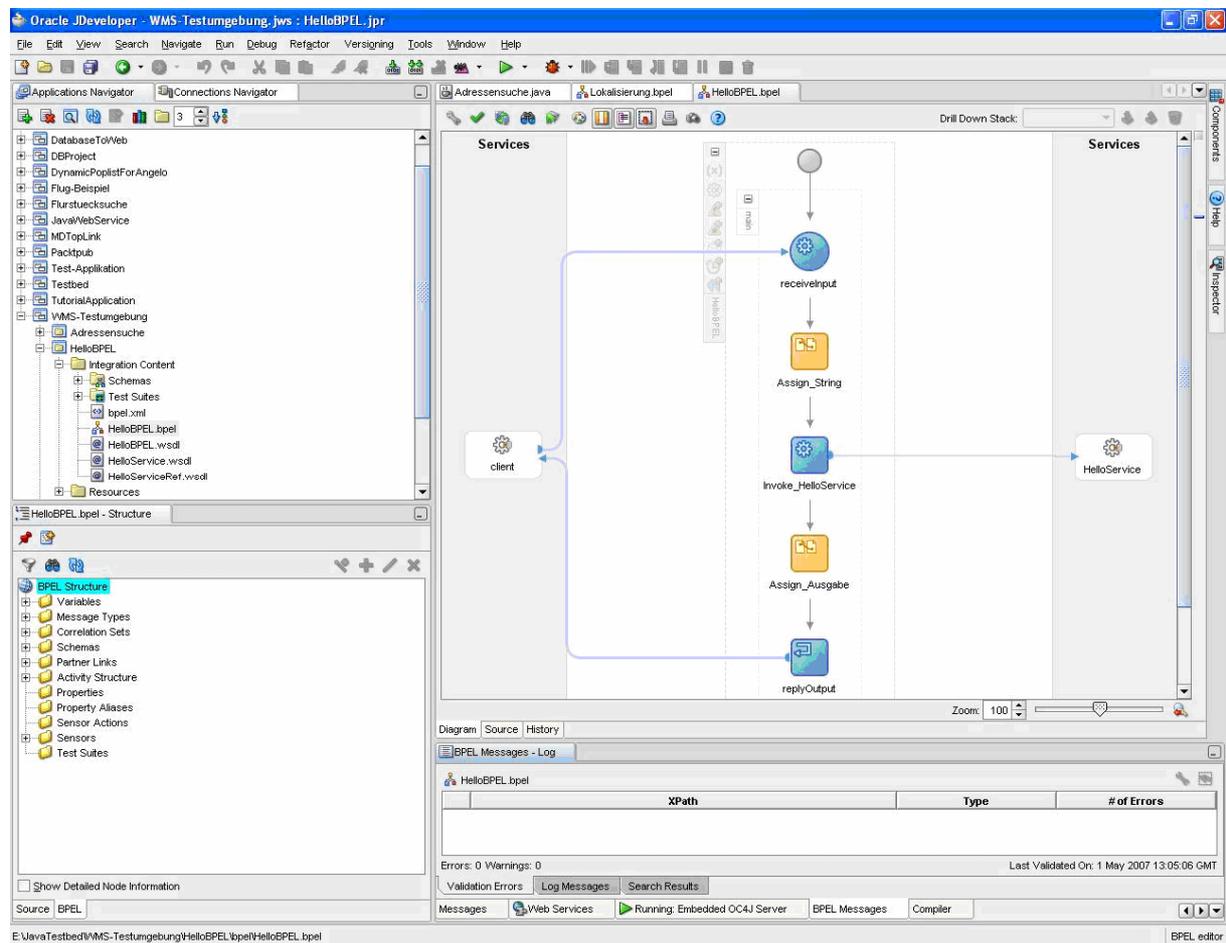


Abbildung 5.11: BPEL-Projekt im JDeveloper

## 5.4 Verwendete Daten

In diesem Kapitel werden die Daten beschrieben, welche zur Umsetzung benötigt werden. Dabei lassen sich die Daten in zwei Kategorien unterteilen. Einerseits werden Daten nur zur Visualisierung und besseren graphischen Darstellung benötigt und andererseits werden Daten als Input für den Prozess verwendet. Die Katasterdaten und auch die Leitungsdaten dienen dem zuerst genannten Zweck. Die Hauskoordinaten, als Zusatz zu den Katasterdaten, sorgen für die Eingangsdaten im Entstörungsprozess, da auf Basis von Straße, Hausnummer und eventuell Hausnummerzusatz der Melder bzw. die Abnahmestelle oder Störungsstelle lokalisiert werden soll. Da diese Umsetzung als Schwerpunkt die Verwendung von geographischen Informationen in dem Geschäftsprozess Störfallmanagement hat, besteht vor diesem Hintergrund keine Notwendigkeit störungsrelevante Tabellen und Attribute zu erzeugen.

Sämtliche Daten sind eigens für die Umsetzung erfasst worden, wobei die Hilfsmittel UML zur Modellierung der Leitungsdaten und das Geographische Informationssystem GeoMedia zur Erfassung der GIS-Daten dienen. Die folgende Abbildung zeigt die Darstellung der erfassten GIS-Daten in Geomedia.

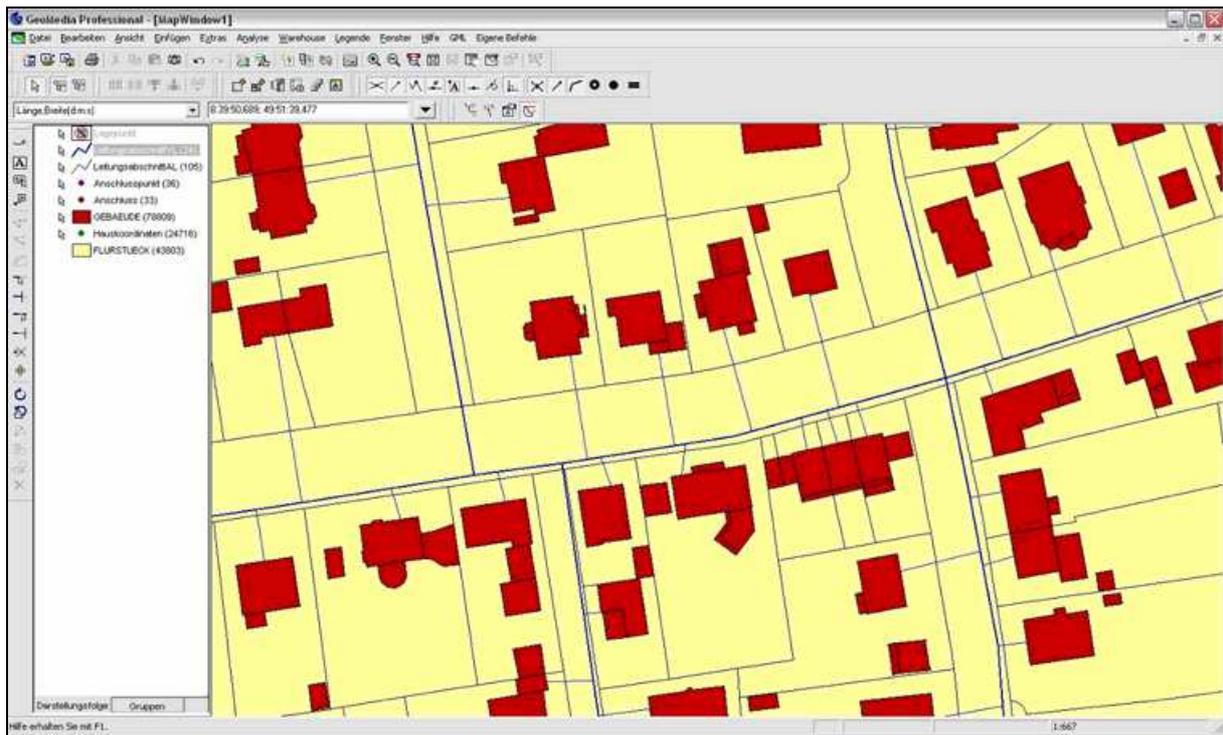


Abbildung 5.12: Visualisierung der erzeugten GIS-Daten in Geomedia

Das Verfahren zur Erfassung der Daten ist für die drei Fachdaten gleich, so dass die Beschreibung zur Erstellung lediglich bei den Hauskoordinaten, die für den Prozess notwendig sind, erfolgt. In den folgenden Kapiteln werden die Hauskoordinaten, die Katasterinformationen und die Leitungsdaten näher beschrieben.

### 5.4.1 Hauskoordinaten

Die Hauskoordinaten als Zusatzinformation zu den Katasterdaten stellen für diesen Prozess die wichtigste Datenquelle dar, da sie die Basis für die Lokalisierung liefern. Die Hauskoordinaten beinhalten eine Tabelle mit selbigem Namen. Relevanter Inhalt dieser Tabelle sind die Attribute Strasse, Hausnummer, Hausnummerzusatz, Rechtswert und Hochwert, wodurch einer Adresse Gauß-Krüger-Koordinaten zugeordnet werden. Weitere Attribute sind der Abbildung 5.13: Tabellenansicht Hauskoordinaten in ORACLE zu entnehmen, die die Struktur und Definition der Tabelle Hauskoordinaten zeigt.

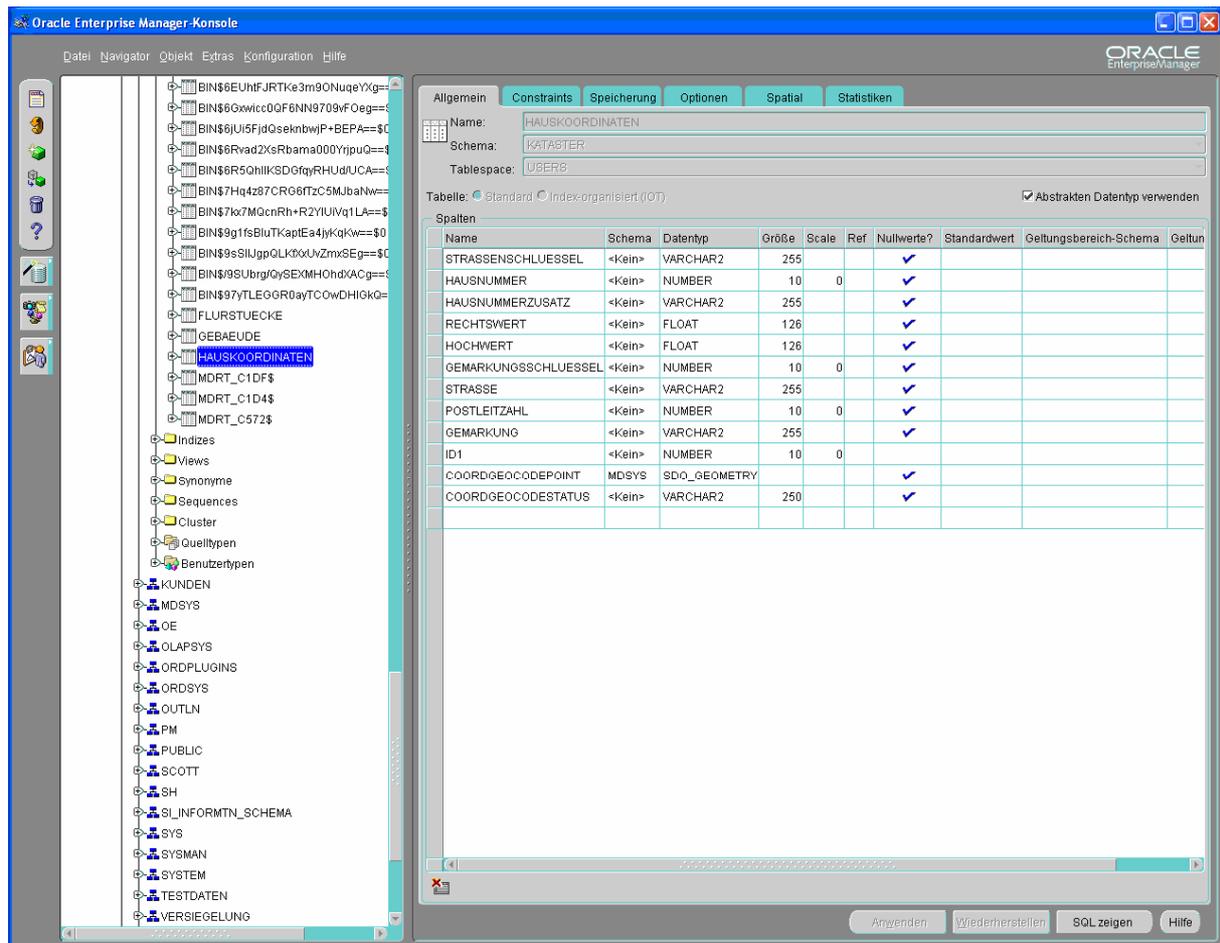


Abbildung 5.13: Tabellenansicht Hauskoordinaten in ORACLE

Die Daten werden nach der Erfassung in GeoMedia mit der dort erzeugten Importroutine über den SQL-Loader in den entsprechenden ORACLE-Benutzer eingelesen. Im Anschluss daran wird das Koordinatensystem dem Geometriefeld zugewiesen. Die Zahl „82027“ steht für das Gauß-Krüger-Koordinatensystem. Dieser Schritt erfolgt über ein SQL-Statement im Tool SQLplus von Oracle. Der SQL-Befehl ist nachfolgend aufgeführt.

```
Update HAUSKOORDINATEN a set a.COORDGEOCODEPOINT.SDO_SROD=82027
```

Es folgt eine Indizierung der Daten nach dem R-Tree-Verfahren, damit der WMS-Service auf die Hauskoordinaten zugreifen kann. Über den nachfolgenden SQL-Befehl wird der Index gesetzt.

```
CREATE INDEX "KATASTER"."HAUSKOORDINATEN"
ON "KATASTER"."HAUSKOORDINATEN" ( "COORDGEOCODEPOINT" )
INDEXTYPE IS MDSYS.SPATIAL_INDEX PARAMETERS ( ' SDO_INDX_DIMS=2
LAYER_GTYPE="COLLECTION" ' )
```

Nach diesen beiden Schritten können die Daten im Map Builder-Tool, welches als graphischer Editor für die geographischen Daten in ORACLE dient, visualisiert werden. In diesem Tool werden ebenfalls sämtliche Einstellungen zur Darstellung der Hauskoordinaten im WMS getroffen, welche in der folgenden Abbildung 5.14: Oracle Map Builder mit visualisierten Hauskoordinaten dargestellt sind.

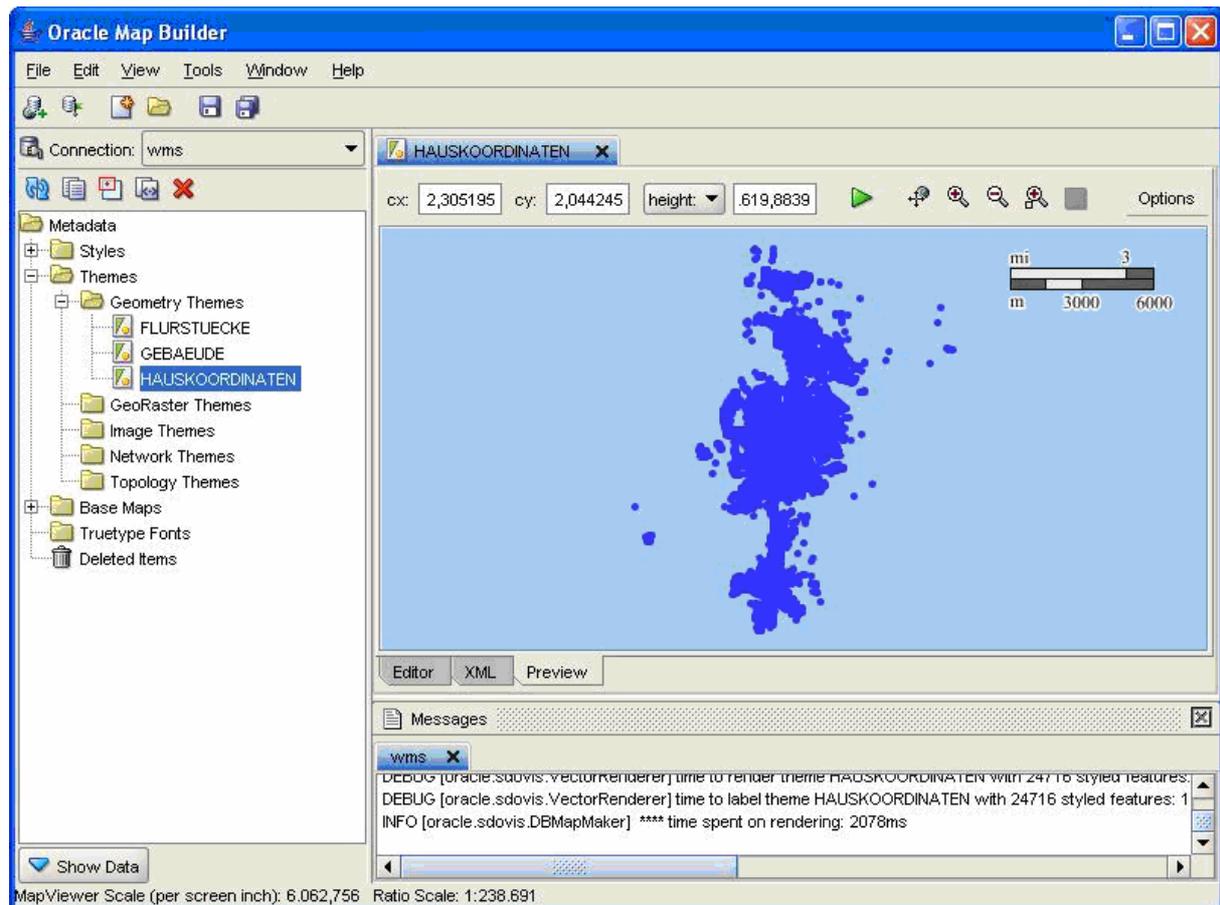


Abbildung 5.14: Oracle Map Builder mit visualisierten Hauskoordinaten

Nachdem den Daten im Map Builder eine Darstellungsart zugeordnet worden ist, sind sie über die Anwendung MapViewer als WMS-Karte abrufbar. Die zuvor genannten Arbeitsschritte sind ebenfalls für die folgenden Themen Flurstücke und Gebäude als Katasterdaten sowie LeitungsabschnittVL und LeitungsabschnittAL als Leitungsdaten durchgeführt worden, so dass diese beiden Themen im Folgenden lediglich beschrieben werden.

### 5.4.2 Katasterinformationen

Katasterinformationen wie beispielsweise Flurstücke und Gebäude, bilden die Grundkarte oder auch Basis für sämtliche Darstellungen des Versorgungsnetzes. Da diese grundlegenden Daten von hoher Aktualität sein müssen, erfassen die meisten Energieversorgungsunternehmen neben den vom Katasteramt gelieferten ALK-Daten ihre eigenen Daten in Bereichen, die noch nicht öffentlich vermessen worden sind. Beispielsweise werden Neubaugebiete beim Energieversorger in der Regel vor den ALK-Daten erfasst, um die Thematik Hausanschlüsse korrekt abwickeln zu können.

Die Katasterinformationen mit den Themen Flurstücke und Gebäude sind als Basiskarte für die Darstellung der Störungsstelle vorgesehen. Da auf die Attribute dieser Daten nicht zugegriffen wird, sind sie nach der Vorgabe des Katasteramtes für ein Mustergebiet erfasst und anonym dargestellt worden. Beide Themen bilden jeweils einen Layer, der über den MapViewer visualisiert werden kann.

### 5.4.3 Leitungsdaten

Die Leitungsdaten stellen für ein Energieversorgungsunternehmen die Geodaten mit dem größten Wert dar, da in ihnen die gesamte Kenntnis über das Versorgungsnetz liegt. Bei der Erfassung werden im Allgemeinen Fachschalen des jeweiligen GIS-Systems verwendet, die sich in Kleinigkeiten unterscheiden, das Netz jedoch in all seinen Zusammenhängen abbilden. Für die Abbildung eines Versorgungsnetzes der Medien Gas und Wasser hat der DVGW ein allgemeines Datenmodell, GAWANIS genannt, entwickelt. Diese 70%-Lösung soll den Unter-

nehmen, als Basis für die eigene Netzdarstellung dienen. Nach Angaben des DVGW setzen etwa 100 Unternehmen dieses Modell als Basis ein. Einige wenige Hersteller haben sich beim DVGW mit ihren Fachschalen für dieses Modell zertifizieren lassen. Die GIS-Systeme sisNET, Smallworld, SICAD UT und Geograt haben ihre Fachschalen Gas und Wasser vom DVGW prüfen lassen.

Da die Leitungsdaten für die Abbildung und Durchführung des Prozesses nur eine untergeordnete Rolle spielen, weil sie nur zur Visualisierung dienen, werden sie für das Untersuchungsgebiet erfasst. Die Erfassung erfolgt nach dem GAWANIS-Datenmodell des DVGW. Das erstellte UML-Diagramm zeigt die zu erfassenden Objekte samt ihren Attributen und ihrer Beziehung untereinander.

Weitere Auszüge des GAWANIS-Datenmodells als UML-Diagramm befinden sich in Anhang J: UML-Diagramm des GAWANIS-Modells (Auszug).

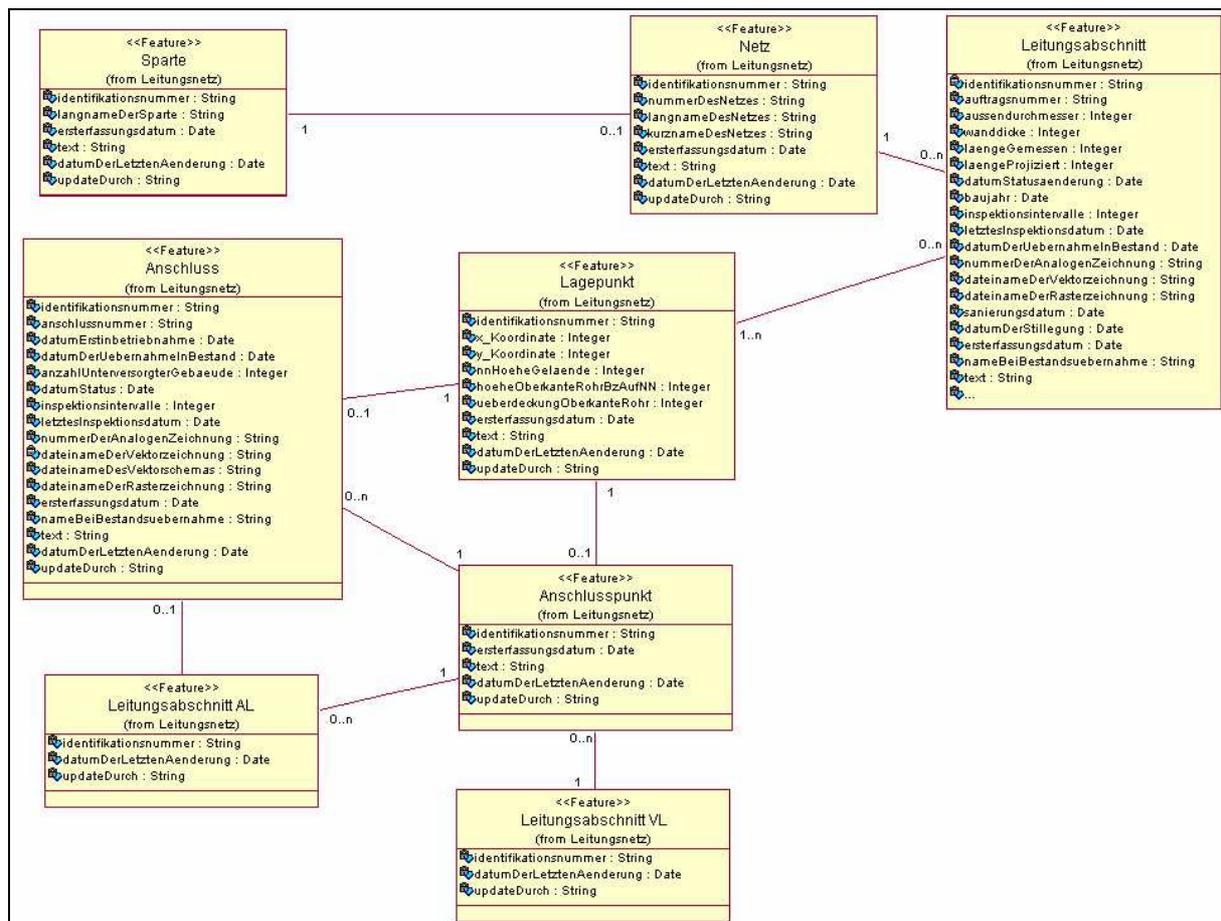


Abbildung 5.15: UML-Diagramm der Leitungsdaten (Auszug)

Für die spätere Darstellung des Versorgungsnetzes sind die dafür wesentlichen Elemente Leitungsabschnitt VL (Versorgungsleitung) und Leitungsabschnitt AL (Anschlussleitung) für das Testgebiet in dem Geographischen Informationssystem GeoMedia jedoch ohne weitere Attribute erfasst worden.

## 5.5 Ablaufbeschreibung des Störungslokalisierungs-Service

Der Störungslokalisierungs-Service wird als Beispiel für einen prozessspezifischen Web Service näher betrachtet und realisiert. Er dient im Allgemeinen dazu, anhand einer Adresse und ggf. Betriebsmittelbezeichnung eine Karte zu retournieren, die diese Adresse im Mittelpunkt zeigt. Diese Funktionalität wird nun weiter verfeinert. Um die angestrebte Wiederverwendbarkeit und die Kommunikation zwischen geographischen und nichtgeogra-

phischen Web Services zu erreichen, wird der Störungslokalisierungs-Service in eben solche unterteilt, die dann über eine Prozesssteuerungssprache BPEL zu einem Prozess zusammengeführt werden.

### 5.5.1 Web Services

Ein „Adressen-Service“ soll als nichtgeographischer Web Service mit einer Adresse als Input die zugehörige Hauskoordinate aus der Datenbank ermitteln, also eine Attributabfrage durchführen. Der „WMS-Service“, als weiterer Web Service geographischer Art, wird um diese Hauskoordinate einen quadratischen Umring mit einer Kantenlänge von 200m x 200m ermitteln. Dazu errechnet der Service die Koordinaten der linken, unteren und rechten, oberen Ecke der BoundingBox und lässt diese Werte in die Erstellung einer WMS-URL einfließen. Diese URL wird als Anfrage an den Web Map Server weitergeleitet.

Um die Informationen über die Karte zu versenden bzw. zu empfangen wird der OGC-Standard WMS und für die allgemeine Kommunikation mit den Web Services die WSDL-Datei verwendet. Da ein Geo Web Service nicht dieselben Kommunikationsstandards nutzt, wie ein allgemeiner Web Service, muss ein Weg gefunden werden, um den geographischen mit dem nichtgeographischen Web Service kommunizieren zu lassen bzw. eine Lösung, mit der beide Web Services orchestriert werden können.

Mit Hilfe der genutzten Entwicklungsumgebung wird die gewünschte Funktionalität der Web Services im ersten Schritt erstellt. Dabei kann es sich beispielsweise um Java-Klassen, EJB-Komponenten von Sun, COM-Komponenten von Microsoft oder auch Visual Basic Anwendungen handeln. Da der Oracle JDeveloper eine Java-Entwicklungsumgebung anbietet, werden die Web Services über Java-Klassen als J2EE-Web Services entwickelt. Mit der Erstellung in Java bedeutet es, dass sämtliche Funktionen beider Web Services als Methoden in jeweils einer Java-Klasse für den geographischen und den nichtgeographischen Web Service abgebildet werden. Toolkits im JDeveloper erzeugen automatisiert die WSDL-Datei, die spezifiziert, welche Funktionen durch den Web Service angeboten werden, sowie das Binding zwischen der WSDL-Schnittstelle und der eigentlichen Implementierung. Zur Ausführung des Web Services wird eine entsprechende Runtime-Umgebung benötigt, welche über das http-Protokoll eingehende SOAP-Anfragen an die eigentliche Implementierung weiterleitet. Die gewählte SOA Suite von Oracle stellt die oben genannten Werkzeuge zum Erstellen eines Web Service auf Java-Basis bereit.

### 5.5.2 BPEL-Prozess

Mithilfe der Prozesssteuerungssprache BPEL werden beide Web Services zusammengeführt bzw. orchestriert, so dass als Ergebnis der Störungslokalisierungs-Service erstellt wird, der als BPEL-Prozess ebenfalls als Web Service genutzt werden kann. Ein Client liefert eine Adresse als Input für den „Adressen-Service“, der daraufhin Rechts- und Hochwert der Hauskoordinate zurückgibt. Diese dient wiederum als Input für den WMS-Service, dessen erste Funktion die Berechnung der BoundingBox ist. Das Ergebnis wird an die zweite Funktion, die Erstellung der WMS-URL, übergeben und diese stellt anhand der WMS-Spezifikation des OGC einen String zusammen, der die URL zur Abfrage der Karte (GetMap-Funktionalität) enthält. Der String wird als Output des Prozesses zurück an den Client geschickt.

Die Erstellung der Web Services und die Zusammenführung dieser zu einem Störungslokalisierungs-Service werden in den folgenden zwei Kapiteln detailliert erläutert.

## 5.6 Erstellung der Web Services

In diesem Kapitel wird die Erstellung des Web Services vorgestellt. Bevor die beiden relevanten Web Services näher beschrieben werden, soll anhand eines Beispiel-Service die allgemeine Vorgehensweise zur Erstellung von Web Service unter Zuhilfenahme der im Abschnitt 5.3 aufgeführten Softwarekomponenten beschrieben werden.

### 5.6.1 Erstellung eines allgemeinen Web Service

Bei der Erstellung eines Web Services kommen die Softwareprodukte JDeveloper, welcher den Web Service an sich erzeugt, und Oracle Application Server, der zur Bereitstellung des Web Services für den Client dient, zum Einsatz. Die generelle Vorgehensweise zur Erstellung eines Web Services wird anhand des „Hello World“-Web Services erläutert. Abbildung 5.16 zeigt die relevanten Komponenten im Zusammenspiel, welche bei der Erstellung, Bereitstellung und Test bzw. Nutzung von Web Services partizipieren.

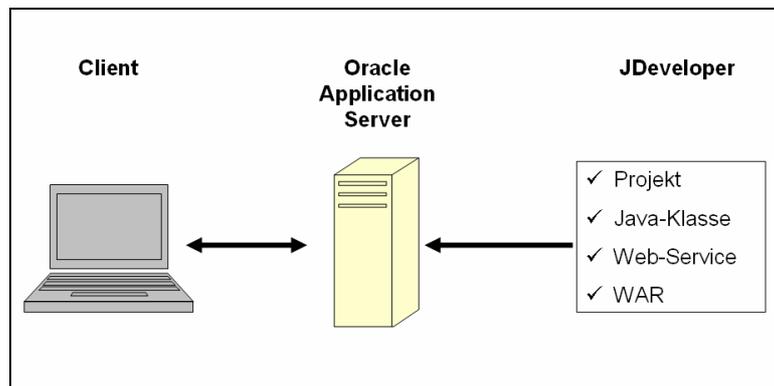


Abbildung 5.16: Komponenten im Zusammenspiel

Es sind vier Schritte zur Entwicklung eines Web Services in der verwendeten Oracle-Umgebung erforderlich. Zunächst wird im JDeveloper ein neues Projekt angelegt, in welchem die Java-Klasse, die die Funktionalität des Web Services beinhaltet, erstellt werden kann. Für den „Hello World“-Web Service ist der nachfolgende Code notwendig und in Abbildung 5.17 aufgeführt. Die Java-Klasse kann entweder im JDeveloper programmiert oder dahin importiert werden.

```
23 HelloService.java
package helloservice;

public class HelloService {
    public String sayHello (String name) {
        return ("Hello "+ name);
    }
}
```

Abbildung 5.17: Code-Beispiel für den „Hello World“-Web Service

An dieser Stelle endet die Entwicklung eines Web Services, da lediglich die Funktionalität erstellt werden muss. Alle weiteren Schritte werden mit dem JDeveloper durchgeführt, welcher mit einem Assistenten die Erstellung erleichtert. Nach Fertigstellung der zur Verfügung stehenden Funktionalität wird der Assistent aufgerufen, welcher einige Angaben erfordert. In diesem Fall bekommt der Web Service den Namen „HelloService“ und die Funktionalität der Methode „sayHello“ aus der bestehenden Java-Klasse zugewiesen. Weiterhin wird ein Endpunkt angegeben, unter dessen URL der Service später erreichbar sein soll. Die Erstellung des einfachen Web Service ist hiermit abgeschlossen. Über den Assistenten werden ebenfalls alle notwendigen Standarddokumente für den Web Service erzeugt. Die wesentlichen Dateien werden im Folgenden erläutert und sind in der Abbildung 5.18 in der Projektstruktur dargestellt:

- das WSDL-Dokument zur Beschreibung des Web Services (*HelloService.wsdl*),

- ein Interface,
- das Deployment-Profil **WebServices.deploy**, welches alle Informationen über die Source-Dateien (siehe Abbildung 5.19: Source-Dateien des "HelloService" -Web Service) zur Bereitstellung des Services inklusive einer WAR-Datei (Web Application Archive) beinhaltet,
- eine XML-Datei **HelloService-java-wsdl-mapping.xml**, die die über den Web Service ausgetauschten Objekte wenn notwendig nach XML konvertiert und wieder zurück,
- eine XML-Datei **oracle-webservices.xml**, die spezielle Informationen für einen OC4J (Oracle Container for Java) enthält,
- die Datei **web.xml**, die die Konfiguration der J2EE Web Komponenten ermöglicht und
- eine XML-Datei **Webservices.xml**, die Informationen zur Beschreibung des Services und zum Mapping beinhaltet.

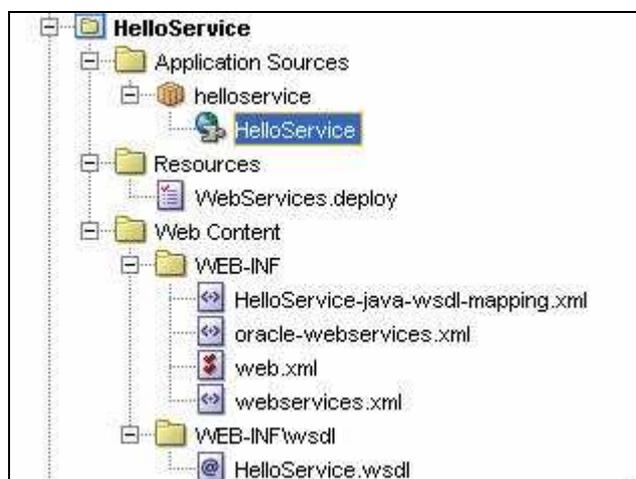


Abbildung 5.18: Projektstruktur des "HelloService"-Web Service

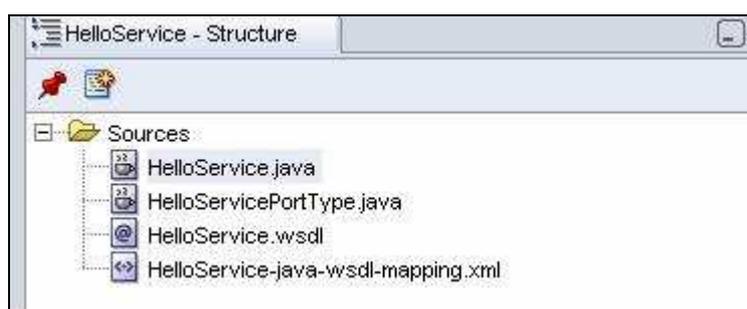


Abbildung 5.19: Source-Dateien des "HelloService" -Web Service

Der eigentliche Web Service ist somit fertig gestellt und wird auf dem Oracle Application Server bereitgestellt. Diese Bereitstellung kann ebenfalls über den JDeveloper initiiert werden, wenn im selbigen eine Verbindung zum Application Server erstellt wurde. Entsprechende Angaben werden über das Editieren der Datei **WebServices.deploy** gemacht. Dann wird darüber das Deployment gestartet, welches die WAR-Datei erzeugt, diese auf dem zugewiesenen Application Server ablegt und daraus den Web Service bereitstellt. Die WAR-Datei enthält alle Dateien, die für den Betrieb des Web Services nötig sind: die Java-Klasse, das Interface, das WSDL-Dokument und die Datei **web.xml**.

Der „Hello World“-Web Service liegt nun auf dem Oracle Application Server bereit und kann unter der lokalen URL

```
http://server:8888/WMS-Testumgebung-HelloService-context-  
root/HelloServiceSoapHttpPort
```

getestet werden. Die Abbildung 5.20 zeigt einen Testclient, mit dessen Hilfe die Funktionalität des Web Services überprüft werden kann. Der Methode „sayHello“ wurde der String „WORLD“ übergeben. Die SOAP-Antwort des Web Services liefert den String „Hello WORLD“ zurück.

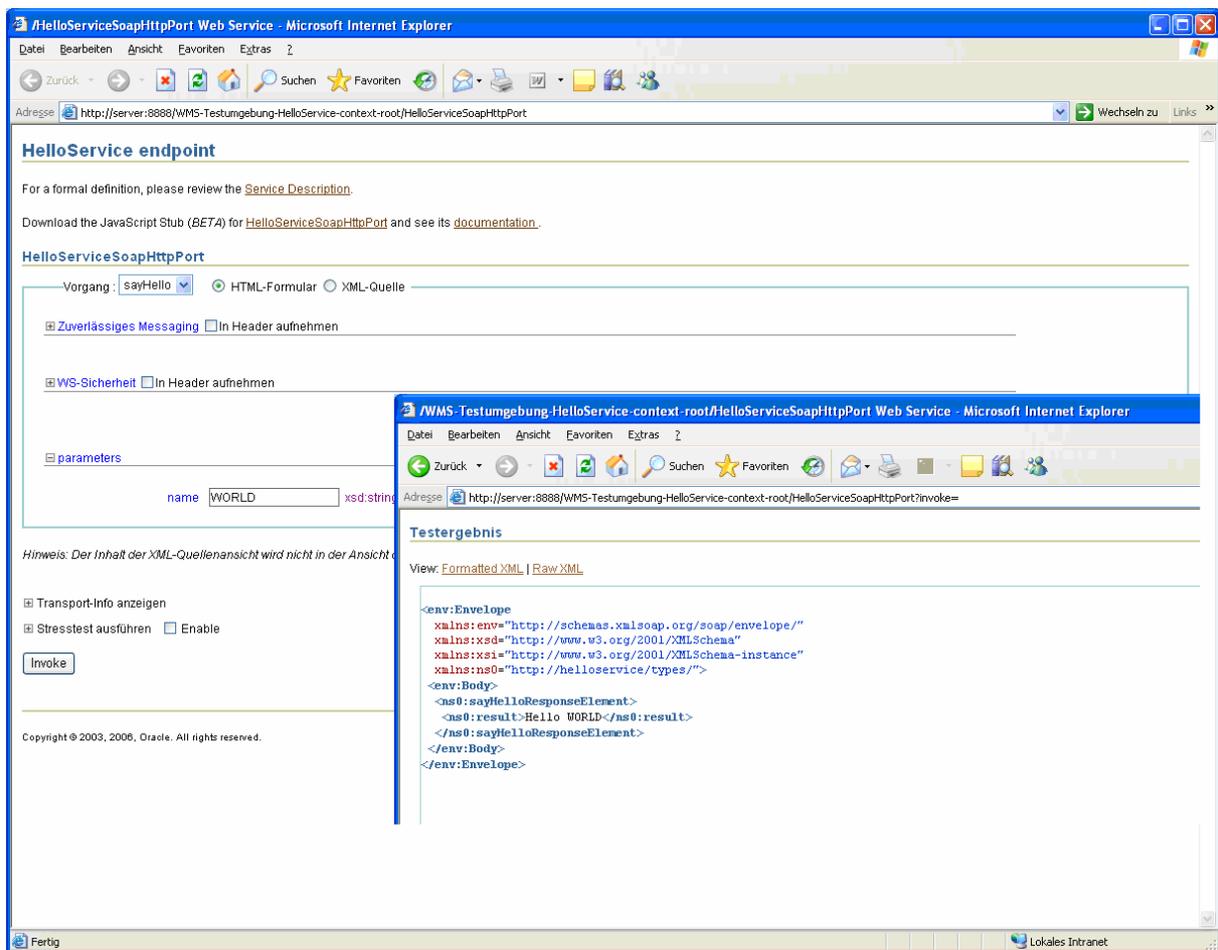


Abbildung 5.20: Test und Ergebnis des "Hello World"-Web Service

Nach diesem Test ist die Erreichbarkeit und Funktionalität des Web Services sichergestellt, so dass er über eine URL beispielsweise von einem BPEL-Prozess aufgerufen werden kann.

## 5.6.2 Beschreibung des Adressen-Service

Der Adressen-Service dient zur Filterung bzw. Auswahl von Straße und Hausnummer sowie gegebenenfalls einem Hausnummerzusatz. Diese Informationen stellen die Basis für den weiteren Prozessverlauf, also die Lokalisierung der Störungsstelle dar.

Die Java-Klasse, die diesem Web Service zugrunde liegt, enthält verschiedene Funktionen. Die folgenden öffentlichen (public) Methoden können in Abhängigkeit voneinander beim Adressen-Service aufgerufen werden:

- getStrasse,
- getHausnummer,
- getHausnummerzusatz,
- getRechtswert und
- getHochwert.

Die beiden Methoden-Gruppen sind jeweils durch ein Beispiel (getHausnummer und getRechtswert) in den folgenden zwei Abbildungen dargestellt.

```
/**
 * Fragt alle möglichen Hausnummern zur ausgewählten Strasse aus der Datenbank
 * @param strasse Name der ausgewählten Strasse
 * @return Nummern der Hausnummern, die in der Strasse vorkommen
 * @wehmethod
 */
public String getHausnummer(String strasse) {
    verbinden();
    Vector hausn = new Vector();
    try {
        // Abfrage der Datenbank: Gebe alle möglichen Strassenamen zurück
        String query = "SELECT DISTINCT hausnummer FROM hauskoordinaten WHERE strasse = '" + strasse + "'";
        s = conn.createStatement();
        rs = s.executeQuery(query);
        // Abfrage speichern
        while (rs.next()) {
            hausn.addElement(rs.getString("hausnummer"));
        } // Ende while
    } catch (SQLException sqle) {
        System.out.println("Die Datenbankabfrage nach der Hausnummer konnte nicht ausgeführt werden.");
        sqle.printStackTrace();
    } // Ende catch
    schliessen();
    String [] hausn1 = makeArray(hausn);
    String [] hausnummer = hausnummerSortieren(hausn1);
    String hausmul = makeString2(hausnummer);
    return hausmul;
} // Ende abfragen()
```

Abbildung 5.21: Java-Klasse "Adresse" mit der "getHausnummer"-Methode

```

/**
 * Gibt den Rechtswert zur ausgewählten Strasse und Hausnummer und HSNZusatz aus der
 * Datenbank zurueck
 * @param strasse Ausgewaehlte Strasse
 * @param hausnummer Ausgewaehlte Hausnummer
 * @param hausnummerzusatz Ausgewählter Hausnummerzusatz
 * @return Moegliche Hausnummerzusätze in Abhaengigkeit von Strasse und
 *         Hausnummer
 * @webmethod
 */
public String getRechtswert(String strasse, String hausnummer, String hausnummerzusatz) {
    verbinden();
    Vector rechtsw = new Vector();
    try {
        // Abfrage der Datenbank: Gebe den Rechtswert zurueck
        String query = "SELECT rechtswert FROM hauskoordinaten WHERE strasse = '"
        + strasse + "' AND hausnummer ='" + hausnummer + "' AND hausnummerzusatz ='" + hausnummerzusatz + "'";
        s = conn.createStatement();
        rs = s.executeQuery(query);
        // Abfrage speichern
        while (rs.next()) {
            rechtsw.addElement(rs.getString("rechtswert"));
        } // Ende while
    } catch (SQLException sqle) {
        System.out.println("Die Datenbankabfrage nach dem Hausnummerzusatz konnte nicht ausgefuehrt werden.");
        sqle.printStackTrace();
    } // Ende catch
    schliessen();
    String rechtswert = makeString1(rechtsw);
    return rechtswert;
} // Ende abfragen()

```

Abbildung 5.22: Java-Klasse "Adresse" mit der "getRechtswert"-Methode

Der vollständige Code der Java-Klasse Adresse.java befindet sich in Anhang C: Code der Java-Klasse Adresse.java.

Aus der Java-Klasse wird mithilfe des JDevelopers ein J2EE Web Service erzeugt, welcher über den selbigen auf dem Application Server über das so genannte "Deployen" bereitgestellt wird. Steht der Service zur Verfügung, kann er über eine Testseite auf seine Funktionalität und Lauffähigkeit überprüft werden. Die folgende Abbildung zeigt den Test der „getHausnummerzusatz“-Methode, die als Input die Variablen Strasse und Hausnummer benötigt. Die Testseite stellt eine Eingabemöglichkeit für die Variablen bereit.

Der vollständige Code der WSDL-Datei des Adressen-Service befindet sich in Anhang E: Code der WSDL-Datei des Adressen-Service.

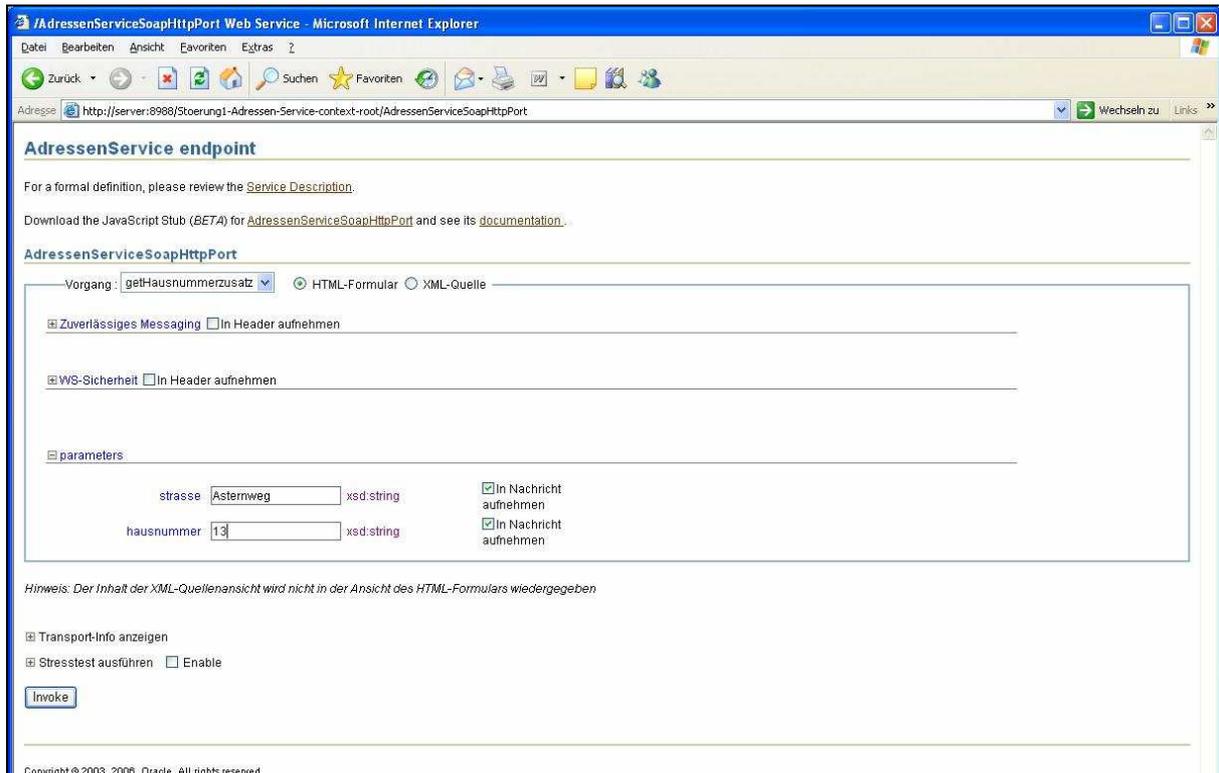


Abbildung 5.23: Test der "getHausnummerzusatz"-Methode des Adressen-Service

Durch einen Klick auf den Button „Invoke“ wird die Anfrage an den Web Service gesendet. Das Ergebnis ist eine SOAP-Nachricht in XML (formatiert oder unformatiert) mit dem zugehörigen „response“-Element, das die Hausnummerzusätze A und B zum Asterweg 13 zeigt. In Abbildung 5.24: Testergebnis der "getHausnummerzusatz"-Methode wird die SOAP-Nachricht als Antwort des Web Service dargestellt.

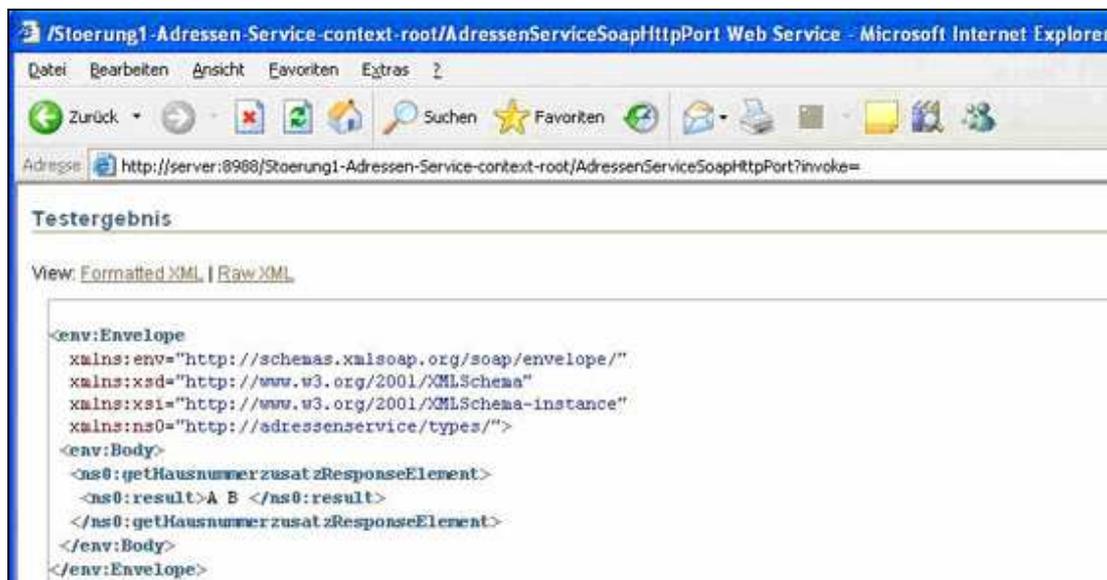


Abbildung 5.24: Testergebnis der "getHausnummerzusatz"-Methode

Der Adressen-Service steht funktionsfähig auf dem Application-Server bereit und kann für die weitere Bearbeitung angesprochen werden.

### 5.6.3 Beschreibung des WMS-Service

Der WMS-Service stellt das Herzstück des Störungsprozesses dar. Anhand von Eingangskordinaten stellt er eine WMS-Abfrage zusammen und gibt die zugehörige URL zurück, unter welcher die abgefragte Karte aufgerufen werden kann.

Basis dieses Web Services ist eine Java-Klasse mit fünf Funktionalitäten. Vier Methoden errechnen anhand einer Eingangskordinate die BoundingBox von 200mx200m und geben den Rechts- und Hochwert der linken unteren sowie rechten oberen Ecke der BoundingBox als Koordinatenpaare zurück. Die folgende Abbildung zeigt die Java-Klasse „WMS“ samt Variablendeklaration und den vier Methoden zur Berechnung von minx, miny, maxx und maxy als Eckkoordinaten der BoundingBox. Als Input dient eine Eingangskordinate, welche in diesem Fall eine Hauskordinate ist. Diese Eingangskordinate wird mit den Variablen mittex und mittey als Mittelpunkt der darzustellenden Karte gesetzt und die linke untere bzw. rechte obere Ecke werden daraus ermittelt. Für die „min“-Werte wird jeweils der Wert 100 von dem Eingangswert abgezogen. Die „max“-Werte errechnen sich auch aus der Eingangskordinate zuzüglich 100. Das Ergebnis wird über „return“ wieder zurückgegeben.

```
public double getBBoxminx(double mittex) {
    minx = mittex - 100;
    return minx;
}

public double setBBoxminy (double mittey) {
    miny = mittey - 100;
    return miny;
}

public double setBBoxmaxx (double mittex) {
    maxx = mittex + 100;
    return maxx;
}

public double setBBoxmaxy(double mittey) {
    maxy = mittey + 100;
    return maxy;
}

// Ende setBoundingBox()
```

Abbildung 5.25: Java-Klasse WMS mit Methoden zur Berechnung der Boundingbox

Die Methode „getMVURL“ setzt eine URL als String zur Abfrage eines WMS-Kartenausschnittes mit der zuvor berechneten BoundingBox zusammen und retourniert diese URL in einer SOAP-Nachricht an den Client. Das folgende Codebeispiel zeigt die Zusammenstellung der WMS-URL, wie sie im Kapitel Web Map Server (WMS) beschrieben wird.

```
/**
 * Fragt die Karte beim MapViewer ab
 * @return URL des WMS des MapViewers
 * @webmethod
 */
public String getMVURL(double minx, double miny, double maxx, double maxy) {
    mvURL = "http://localhost:8888/mapviewer/wms?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1" +
        "&width=" + breite + "&height=" + hoehe + "&Bbox=" +
        minx + "," + miny + "," + maxx + "," + maxy + "&Layers=" + layerMV +
        "&SRS=EPSG:31467&format=image/png8&TRANSPARENT=TRUE";
    return mvURL;
}

// Ende getMVURL()
```

Abbildung 5.26: Java-Klasse WMS mit Methode "getMVURL"

Der vollständige Code der Java-Klasse WMS.java befindet sich in Anhang C: Code der Java-Klasse Adresse.java.

Aus der Java-Klasse wird im JDeveloper ein J2EE-Web Service erzeugt und auf dem in der Systemumgebung integrierten Application Server der SOA Suite von ORACLE bereitgestellt, so dass er dort angesprochen werden kann. Dies ist eine Standardfunktionalität des JDeveloper in der SOA-Suite. Der vollständige Code der WSDL-Datei des WMS-Service befindet sich in Anhang F: Code der WSDL-Datei des WMS-Service. Die folgende URL ist das Ergebnis einer Anfrage an den WMS-Service, welches in einer SOAP-Nachricht geliefert und in Abbildung 5.27: Ergebnis des WMS-Service im InternetExplorer visualisiert wird. In der Karte werden Katasterinformationen sowie Hauskoordinaten und das Leitungsnetz dargestellt.

```
http://localhost:8888/mapviewer/wms?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&width=200&height=200&Bbox=3475666.96,5524787.19,3475866.96,5524987.19&Layers=Flurstuecke,Gebaeude,Leistungsabschnittv1,Leistungsabschnittal&SRS=EPSG:31467&format=image/png8&TRANSPARENT=TRUE
```

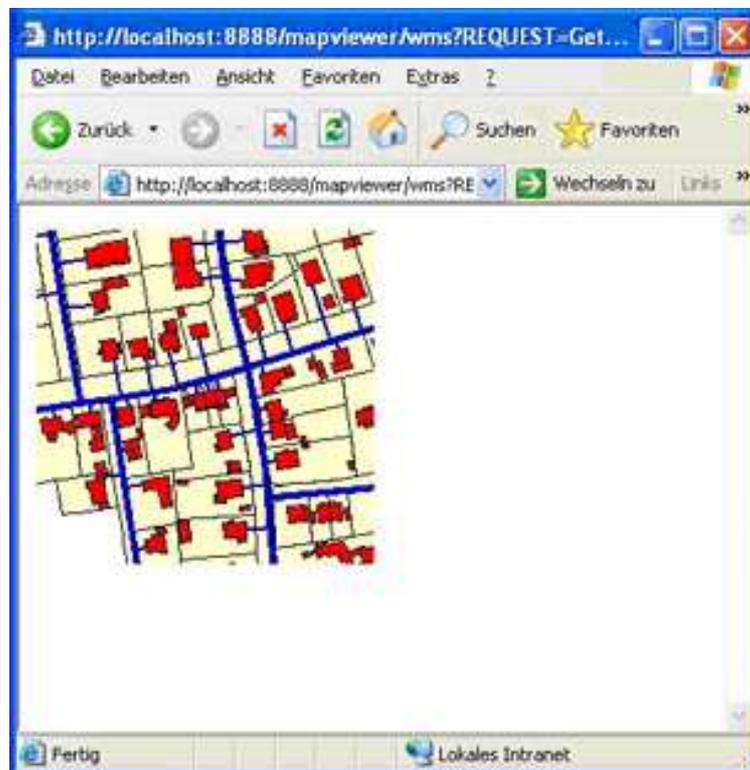


Abbildung 5.27: Ergebnis des WMS-Service im InternetExplorer

Der WMS-Service steht genauso wie der Adressen-Service auf dem Application-Server zur Verfügung. Über BPEL können beide Web Services angesprochen und zu einem Prozess orchestriert werden. Diese Vorgehensweise wird im nächsten Kapitel beschrieben.

## 5.7 Orchestrierung der Web Services mit BPEL

In diesem Kapitel werden die beiden zuvor erstellten Web Services im Rahmen einer Orchestrierung zusammengeführt. Den Teil der Orchestrierung übernimmt die Prozesssteuerungssprache BPEL. Analog zum vorherigen Kapitel wird zunächst an einem Beispiel die Erstellung eines BPEL-Prozesses, der einen Web Service aufruft, verdeutlicht und im Anschluss daran die Zusammenführung des Adressen-Service und des WMS-Service mithilfe von BPEL gezeigt.

### 5.7.1 Erstellung eines allgemeinen BPEL-Prozesses

Anhand eines einfachen „HelloBPEL“-Prozesses soll die Funktionsweise und das Zusammenspiel der einzelnen BPEL-Komponenten betrachtet werden. Der „HelloBPEL“-Prozess bekommt einen Namen als Input von einem Client und ruft die Methode „sayHello“ vom „Hello World“-Web Service auf. Der Web Service liefert Informationen als String an den Prozess zurück, der wiederum die Antwort an den Client weiterleitet. Hierbei kommen die in Kapitel 5.3.5 beschriebenen Softwareprodukte zum Einsatz. Der BPEL Designer, hier als Plug-In im JDeveloper verwendet, dient zur Erstellung des BPEL-Projektes. Zu Projektbeginn wird zunächst festgelegt, ob es sich um einen synchronen oder asynchronen Prozess handelt. Das Beispiel retourniert die Information sofort zurück an den Client, weshalb sich für einen synchronen Prozess entschieden wird. Im Anschluss an diese Entscheidung erzeugt der JDeveloper selbstständig die für einen BPEL-Prozess relevanten Dateien, die in Tabelle 5.2 einzeln samt ihrer Beschreibung aufgeführt werden. Je nach Prozessablauf werden weitere Dateien angelegt, so wird beispielsweise je PartnerLink-Element eine eigene WSDL-Datei erzeugt.

Dateiname	Funktion/Beschreibung
HelloBPEL.jpr	Projektdatei des JDeveloper
bpel.xml	beinhaltet Metadaten über den BPEL-Prozess
build.xml	Skript für die Kompilierung und das Deployment
HelloBPEL.bpel	Definition des BPEL-Prozesses
HelloBPEL.wsdl	Beschreibung des BPEL-Prozesses als Web Service

Tabelle 5.2: Struktur eines BPEL-Projektes im BPEL-Designer

Der bisher leere Prozess wird durch Aktivitäten ergänzt. Im Rahmen des „HelloBPEL“-Prozesses werden die Aktivitäten „Assign“ und „Invoke“ verwendet. Die „Assign“-Aktivität leitet in diesem Fall die Eingangsvariable an den mit der Aktivität „Invoke“ aufgerufenen Web Service weiter und liefert als zweiten Baustein die Antwort vom Web Service in die Output-Variable zurück in den Prozess. Die Abbildung 5.28: Design des "HelloBPEL"-Prozesses zeigt die Eingabemaske für den „Invoke“-Befehl. Neben der Arbeit mit graphischen Editoren zur Definition des BPEL-Prozesses kann der Ablauf auch direkt als Code eingegeben werden. Abbildung 5.29: Code des "HelloBPEL"-Prozesses zeigt die Ansicht zur Code-Eingabe.

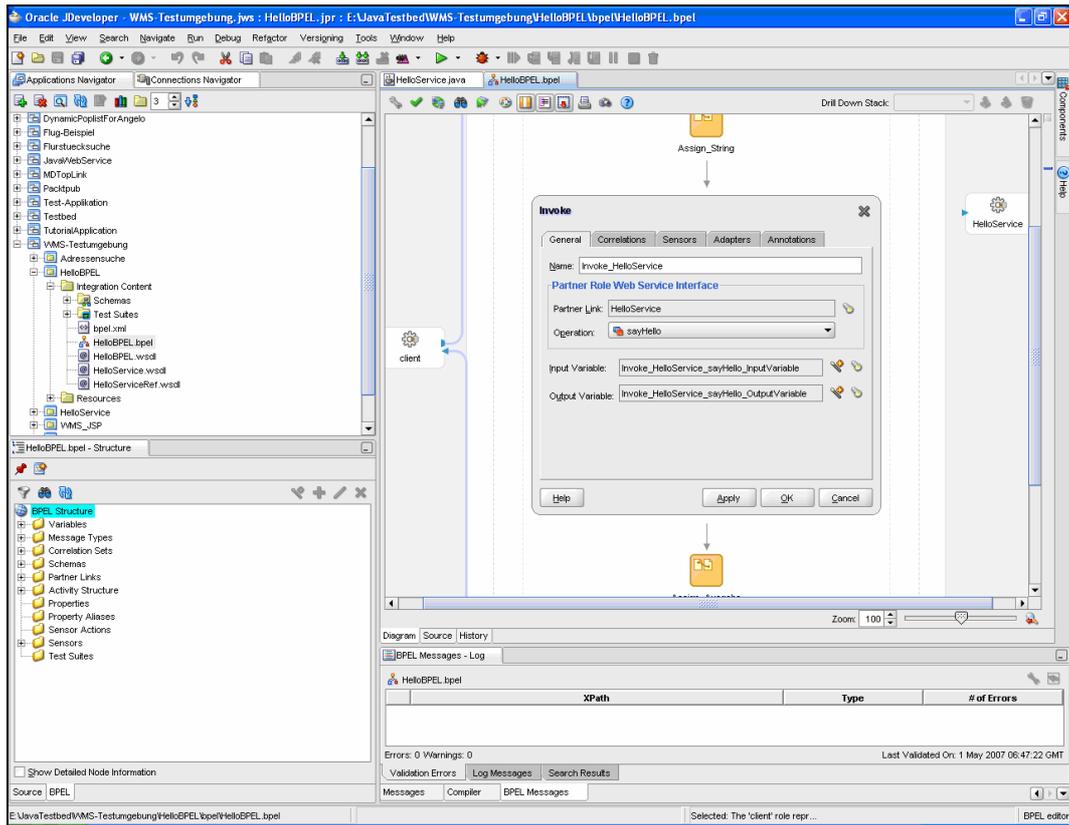


Abbildung 5.28: Design des "HelloBPEL"-Prozesses

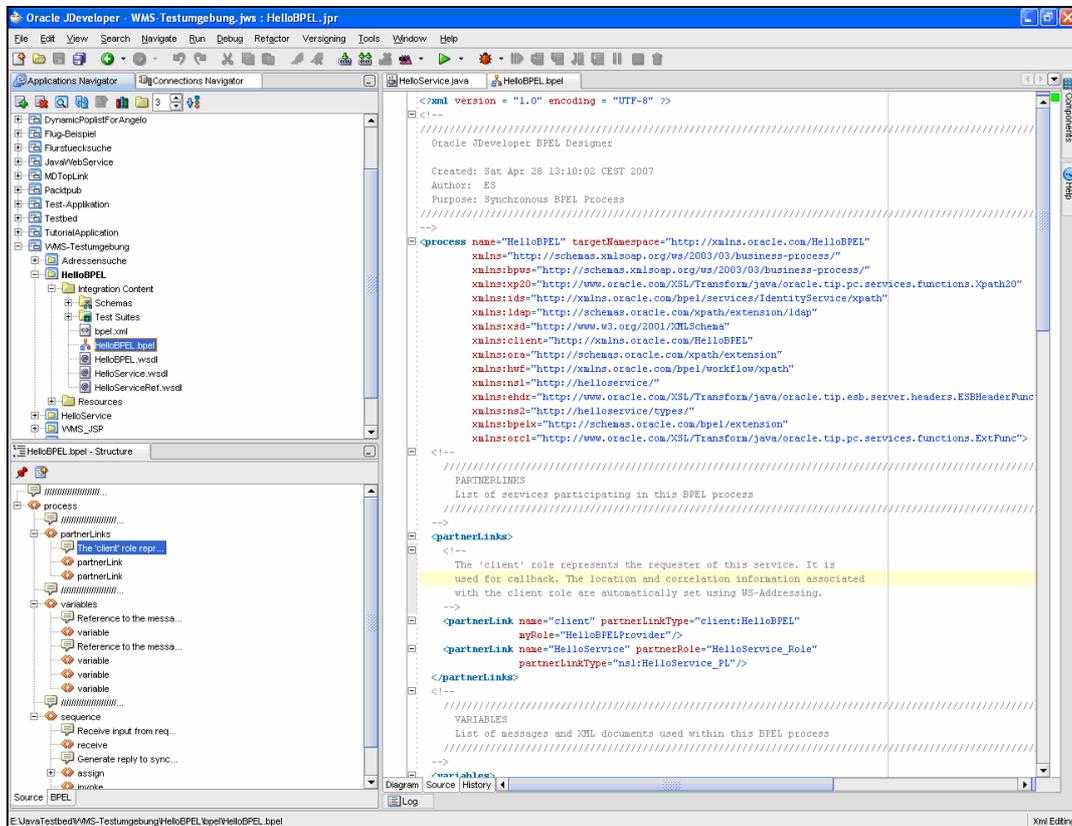


Abbildung 5.29: Code des "HelloBPEL"-Prozesses

Die nachfolgende Abbildung zeigt den gesamten „HelloBPEL“-Prozess als synchronen Ablauf im BPEL-Designer. Der Prozess besteht aus den Elementen: RecieveInput, Assign\_String, Invoke\_HelloService, Assign\_Ausgabe und ReplyOutput. Dem Prozess werden zwei Partner-Links Client und HelloService zur Kommunikation mit selbigen zugewiesen.

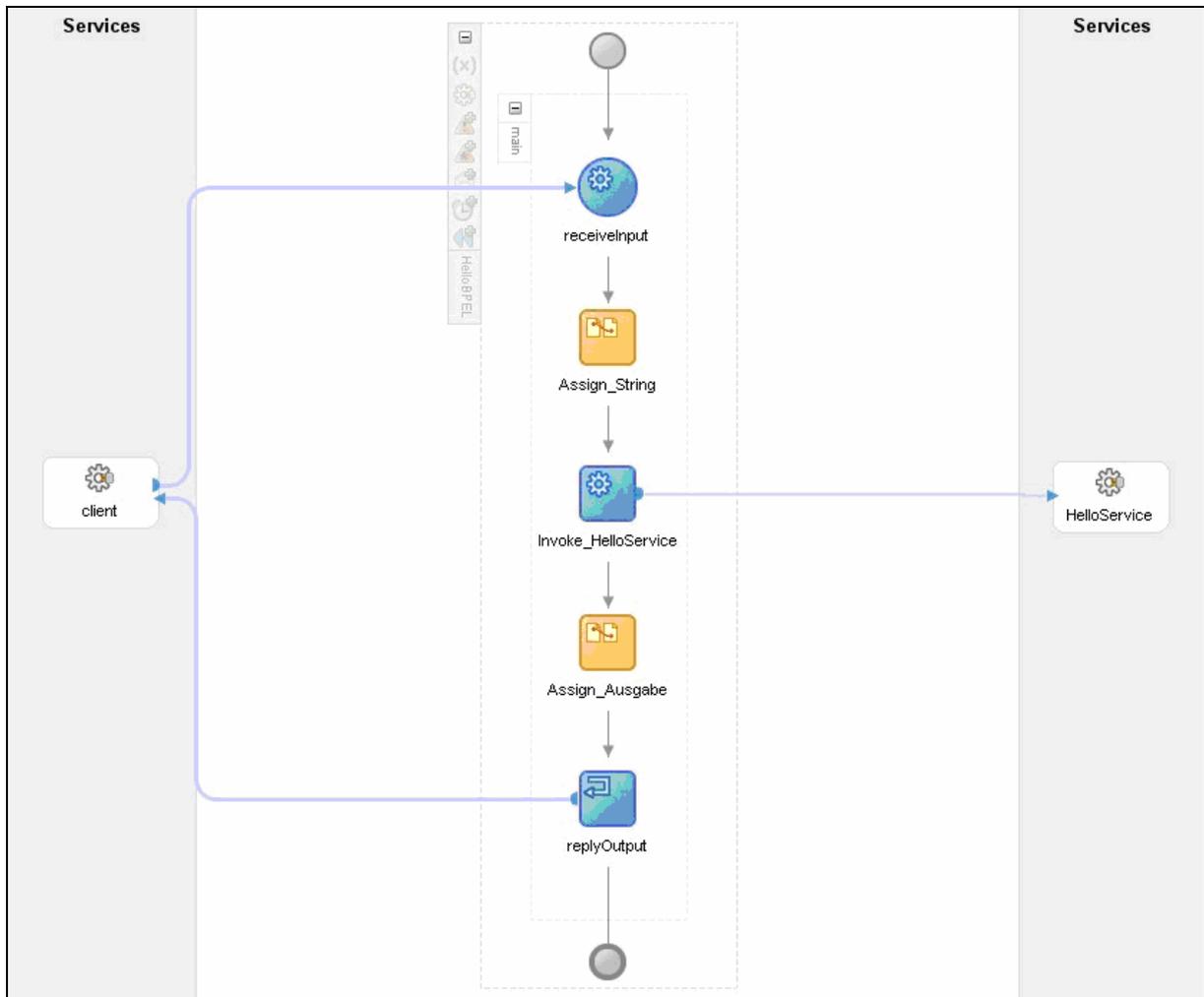


Abbildung 5.30: Synchroner Beispielprozess "HelloBPEL"

Nachdem der Prozess fertig konfiguriert ist, kann er auf der BPEL-Console bereitgestellt werden. Die BPEL Console dient sowohl zur Administration der verschiedenen Prozesse samt zugehörigen Instanzen als auch als Testumgebung für die Prozesse. Die Testumgebung für den Beispiel-Prozess HelloBPEL ist in Abbildung 5.31 dargestellt. Um den Prozess zu testen, wird in der Eingabemaske der String „WORLD“ eingegeben.

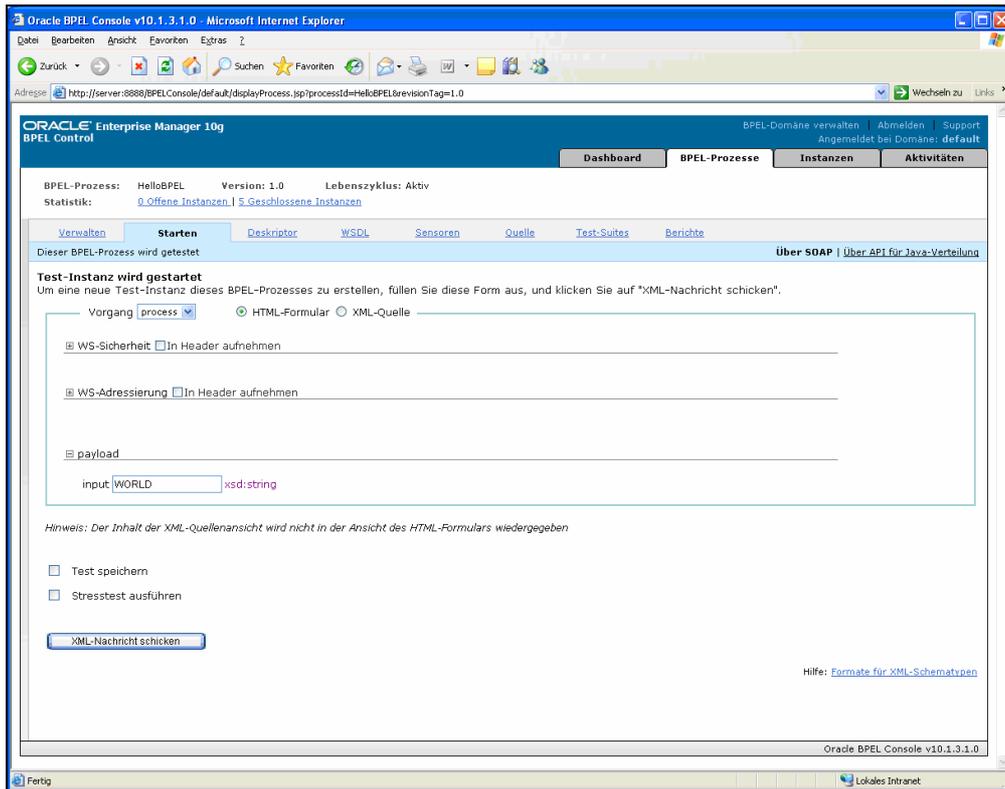


Abbildung 5.31: Test des "HelloBPEL"-Prozesses

Als Ergebnis liefert der Prozess eine SOAP-Nachricht mit den String „Hello World“ in XML zurück wie der Screenshot aus Abbildung 5.32 zeigt.

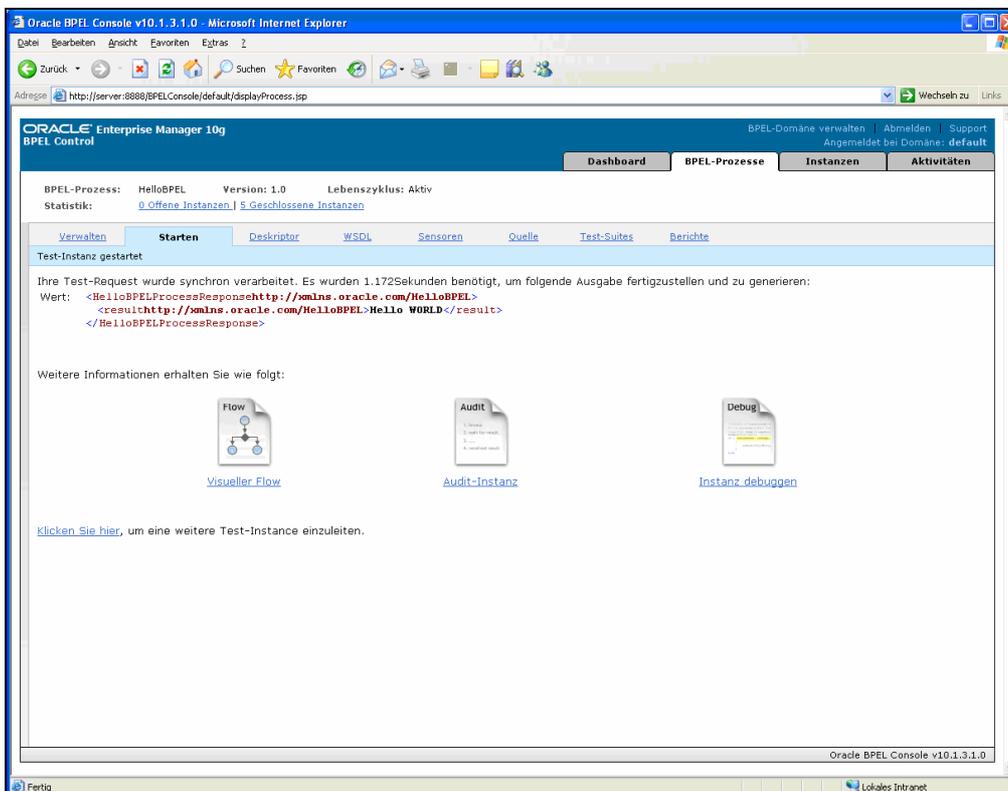


Abbildung 5.32: Ergebnis des "HelloBPEL"-Prozesses

Zusätzlich besteht die Möglichkeit, den Verlauf einer Instanz graphisch (Flow) oder als Audit-Trail wie in der Abbildung 5.33 dargestellt anzuzeigen.

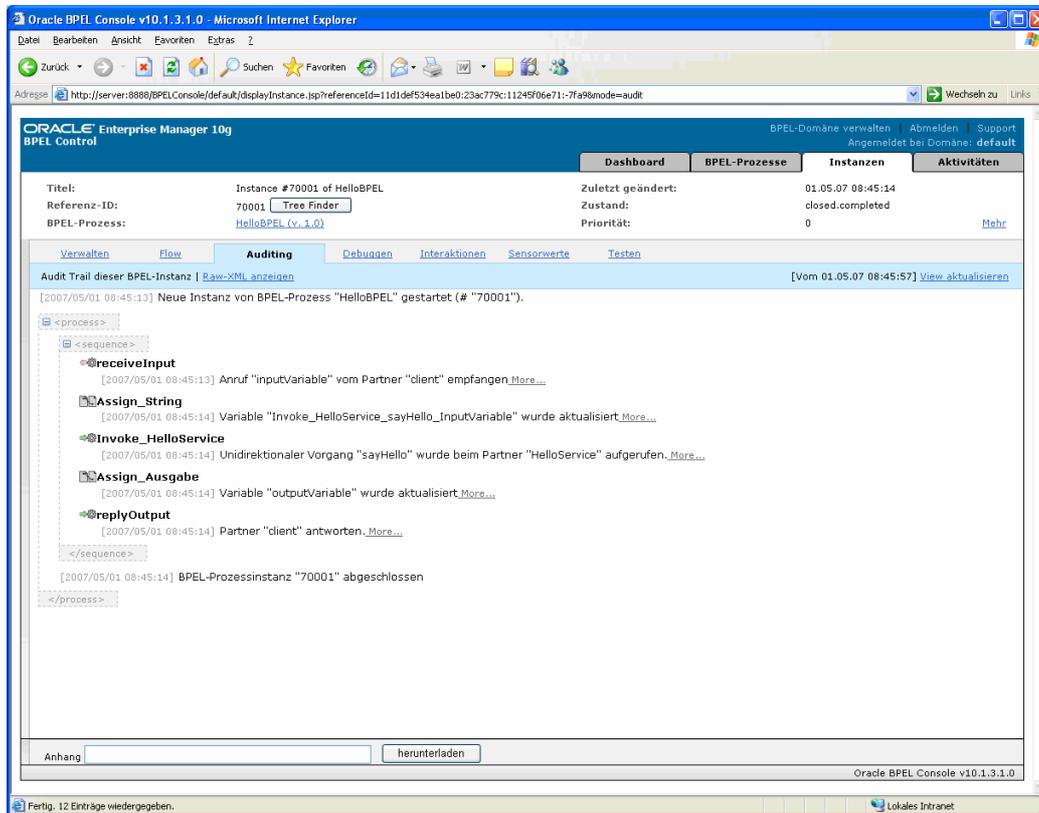


Abbildung 5.33: Auditing des "HelloBPEL"-Prozesses

Nachdem die Erstellung eines synchronen Beispielprozesses am „HelloBPEL“-Prozess verdeutlicht worden ist, wird im folgenden Kapitel die Orchestrierung der für den Bereich Störfallmanagement entwickelten Web Services beschrieben.

## 5.7.2 Störungslokalisierung mit BPEL

In der Störungslokalisierung mit BPEL werden die beiden Web Services, welche in den Kapiteln Beschreibung des Adressen-Service und Beschreibung des WMS-Service erläutert wurden, zu einem Prozess zusammengeführt, was als Orchestrierung bezeichnet wird. Im synchronen Prozess bekommt der Adressen-Service als Input eine Adresse und liefert die zugehörige Hauskoordinate zurück, die wiederum dem WMS-Service als Eingangswert zur Berechnung der BoundingBox für die Kartendarstellung dient, für die eine URL zurück gegeben wird. Zum Einsatz kommt hier ebenfalls der JDeveloper mit integriertem BPEL-Designer sowie der Application-Server auf dem der Prozess bereitgestellt wird. Die BPEL-Console dient zur Administration und zum Testen des erzeugten Prozesses.

Der gesamte Prozess bildet sich graphisch wie in der folgenden Abbildung dargestellt ab. Eine detaillierte Erläuterung des Prozessablaufes wird im Anschluss an die Graphik aufgeführt.

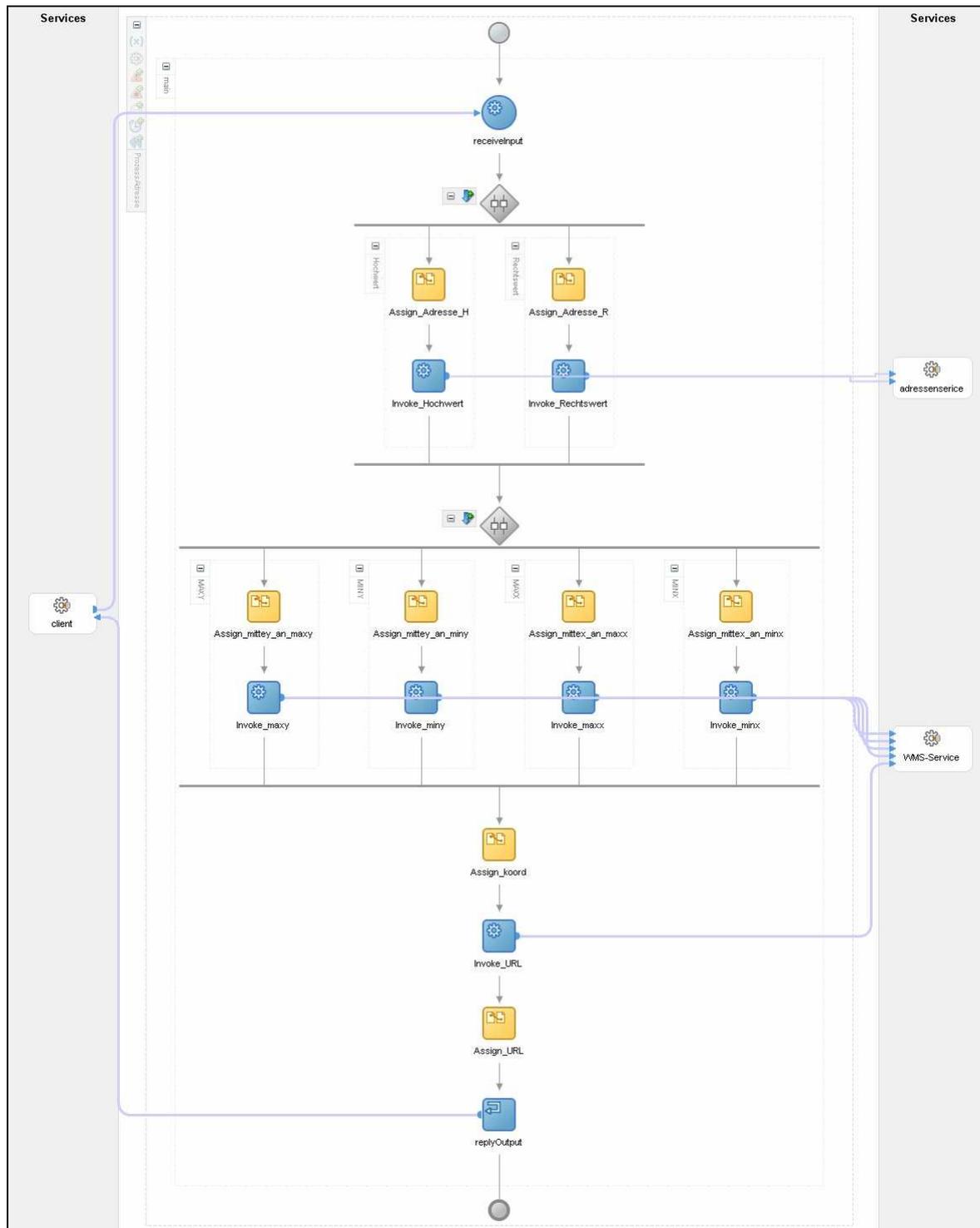


Abbildung 5.34: Ablauf der Lokalisierung in BPEL

Der Client startet den Prozess mit den Eingabewerten für die Variablen Strasse, Hausnummer und Hausnummerzusatz. Parallel zueinander werden der zugehörige Rechts- und Hochwert jeweils über den Aufruf des Adressen-Service aus der Datenbank ermittelt. Im Anschluss wird ebenfalls parallel zueinander die Berechnung der Min- und Max-Werte der BoundingBox gestartet. Der Adressen-Service liefert beide Werte als String zurück, so dass sie vor der weiteren Verwendung zur Berechnung der BoundingBox in eine Zahl umgewandelt werden müssen. Dies geschieht über die Assign-Funktion für jede Methode. Über die vier parallelen Invoke-Funktionen wird der WMS-Service mit den vier Berechnungsmethoden angesprochen und liefert vier einzelne Koordinatenwerte (minx, miny, maxx, maxy) zurück. Diese Koordinatenwerte werden über die Assign\_koord-Funktion den vier

Variablen, welche als Input für die URL-Zusammenstellung dienen, zugewiesen. Die darauf folgende `invoke_URL`-Funktion ruft die Methode „`getMVURL`“ des WMS-Service auf, übergibt die vier notwendigen Eingangsparameter und liefert die komplette URL zur Darstellung eines WMS zurück. Dieses Ergebnis wird wiederum der Output-Variable zugeordnet, die die URL als Inhalt einer SOAP-Nachricht an den Client retourniert.

Der vollständige Code der WSDL-Datei des BPEL-Prozesses befindet sich in Anhang G: Code der WSDL-Datei des BPEL-Prozesses, die zugehörige Schemadatei zum Prozess in Anhang H: Code der Schemadatei XSD zum BPEL-Prozess und der Prozess-Code in Anhang I: Code des BPEL-Prozesses.

Nach Fertigstellung des Prozesses wird dieser auf dem Application-Server bereitgestellt. Mit dem Deployment ist der Prozess verfügbar und kann angesprochen oder getestet werden. In der BPEL Console wird eine Test-Instanz gestartet, welche in der folgenden Abbildung gezeigt wird. Über die Eingabe der Parameter Strasse, Hausnummer und Hausnummerzusatz wird der Prozess gestartet.

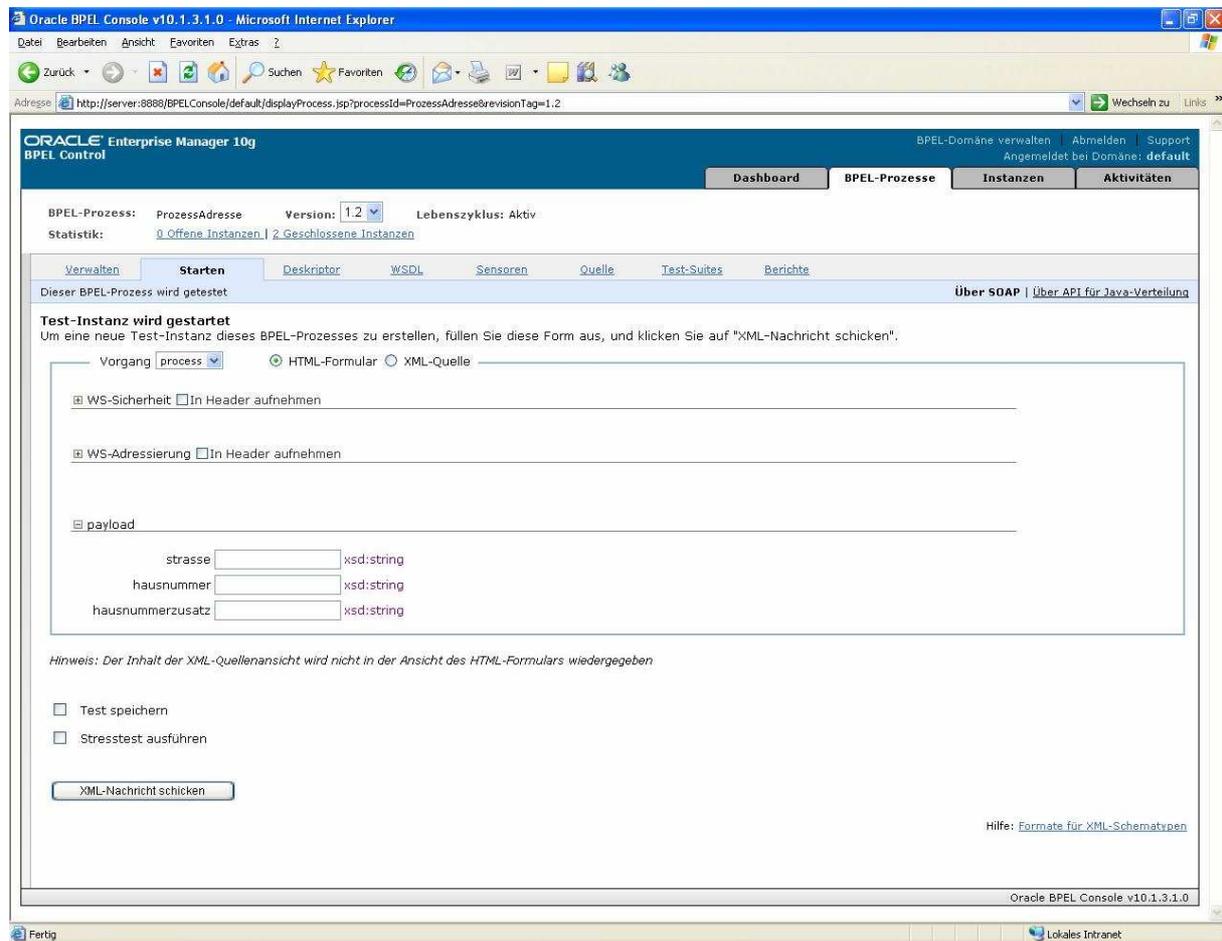


Abbildung 5.35: Aufruf der Testinstanz zum Lokalisierungs-Prozess

Als Ergebnis liefert die Testinstanz eine SOAP-Nachricht, welche die WMS-URL beinhaltet. Der folgende Screenshot zeigt das Ergebnis des Test-Request in der BPEL-Console.

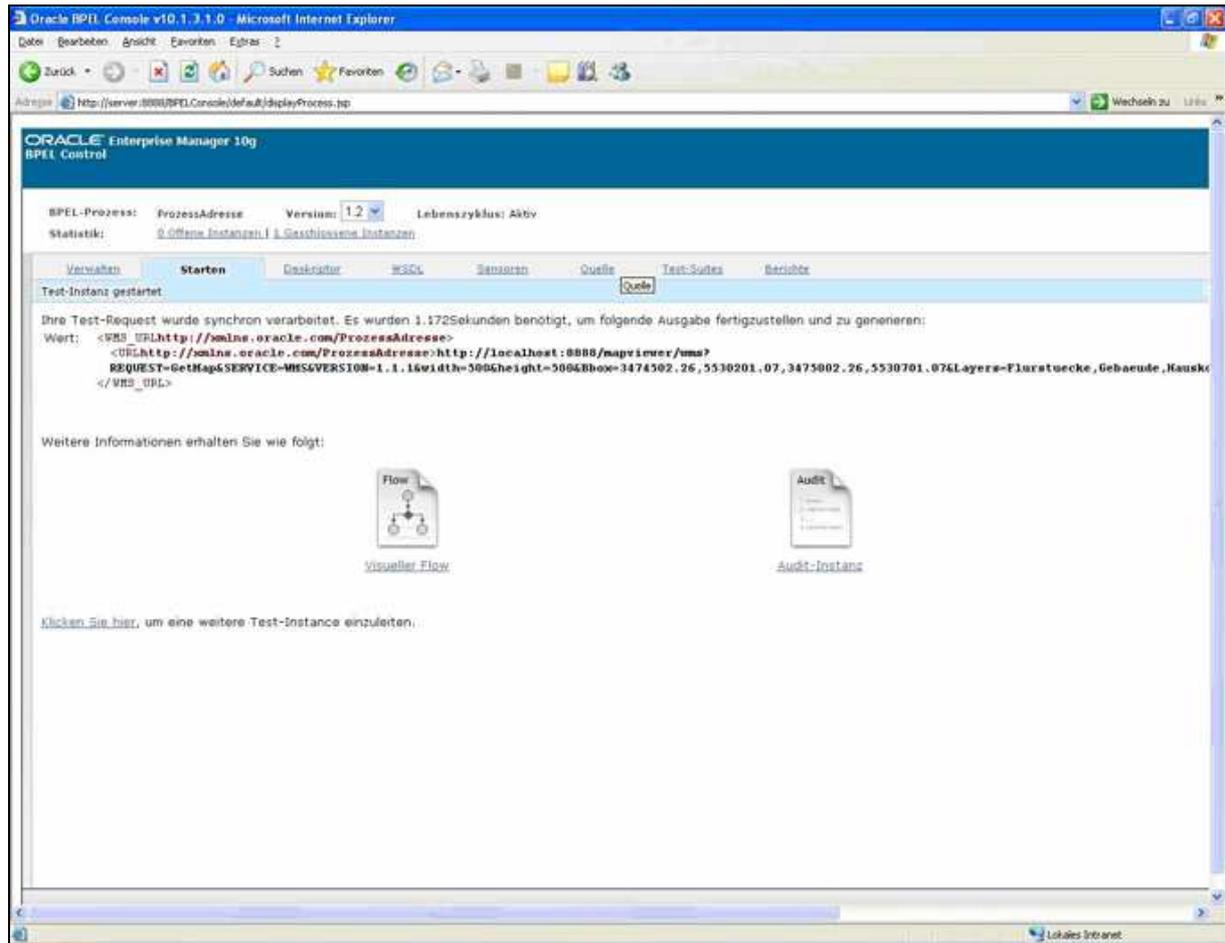


Abbildung 5.36: SOAP-Nachricht mit URL als Ergebnis des Tests

Im Folgenden sind die Ergebnis-URL, sowie deren zugehörige Darstellung in einem Browser dargestellt. Die Karte zeigt einen Ausschnitt von 200m x 200m um die Jahnstraße 105A.

```
http://localhost:8888/mapviewer/wms?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&width=200&height=200&Bbox=3475666.96,5524787.19,3475866.96,5524987.19&Layers=Flurstuecke,Gebaeude,Leitungsabschnittv1,Leitungsabschnittal&SRS=EPSG:31467&format=image/png8&TRANSPARENT=TRUE
```



Abbildung 5.37: Ergebnis-URL als Graphik im Browser

## 5.8 Bewertung

Der erstellte BPEL-Prozess zur Lokalisierung einer Störung kann nach den folgenden Kriterien bewertet werden: Erstellung, Softwareprodukte, Inhalt und Funktionalität.

Die Erstellung der Web Services und des BPEL-Prozesses erfolgt in einer gemeinsamen Oberfläche im JDeveloper, in dem der BPEL-Designer als Plug-In verfügbar ist. Lediglich für die Erstellung der Java-Klassen als Basis für die beiden Web Services Adressen-Service und WMS-Service ist Expertenwissen im Sinne der Java-Programmierung erforderlich. Die Orchestrierung der Web Services zu einem Prozess kann entweder über graphische Editoren oder über direkte Codeeingabe erfolgen. Während der Bearbeitung hat sich der Zugriff auf beide Sichtweisen als sehr vorteilhaft für die Prozesserstellung gezeigt, da nach Bedarf zwischen Diagramm- und Source-Fenster gewechselt werden kann. Von Vorteil ist zumindest eine grundlegende Kenntnis der Prozesssteuerungssprache BPEL, die wie Java als Programmiersprache angesehen werden kann, da die Bezeichnung der Elemente im BPEL-Designer analog zur Sprache ist. Kenntnis über die komplette Syntax von BPEL ist nicht in vollem Umfang erforderlich.

Alle weiteren Schritte zur Erstellung der Web Services samt dem Deployment werden über Assistenten abgewickelt. Ebenso verhält es sich mit der Bereitstellung des BPEL-Prozesses auf dem Application-Server. Über den Application-Server können die Web Services und über die BPEL Console die Prozesse administriert und getestet werden. Die Bereitstellung von Testinstanzen inklusive Eingabemasken hat die Überprüfung der Funktionalitäten von Web Services und Prozess immens beschleunigt, da nicht eigens dafür eine separate Eingabeoberfläche erstellt werden musste. Die Tools der SOA-Suite, insbesondere der JDeveloper mit seinen graphischen Editoren und Verbindungen zu allen weiteren Komponenten der SOA-Suite, haben sich als sehr bedienerfreundlich und für den getesteten Zweck als stabile Lösung bewiesen.

Inhaltlich ist der Störungs-Lokalisierungs-Service als synchroner Prozess abgebildet worden, der wiederum als Web Service von anderen Prozessen angesprochen werden kann und somit in den Gesamtkontext eines Störungsprozesses in eine Service-orientierte Architektur integriert werden kann. Der Lokalisierungs-Prozess ist durch die direkte Antwort des GIS als synchron angenommen worden. Bei einer Kommunikation mit mehreren,

dezentral platzierten Geodaten ist sicherlich ein asynchroner Prozess mit einer Callback-Funktion eine bessere Lösung, da bei diesem im Falle einer ausbleibenden Antwort von einem Web Service der Prozess samt Zustand in einer Datenbank dehydriert wird und erst nach erfolgreicher Antwort weiter abläuft. Da der Schwerpunkt der Umsetzung auf der Anbindung von Geodaten in einem Geschäftsprozess liegt, sind weitere Prozesselemente, wie beispielsweise die Fehlerbehandlung im Ablauf vernachlässigt worden. Um den Charakter von Web Services als reine Kommunikation unter Maschinen zu wahren und ausschließlich Standards zu verwenden, sind keine Human Task Elemente in den Prozess integriert worden. Zwar bietet der BPEL-Designer Human Task Elemente angelehnt an die BPEL-Erweiterung BPEL4people an, jedoch existiert diese Erweiterung zurzeit noch als Whitepaper der Firmen IBM und SAP (Kloppmann, et al., 2005).

Als Basis für die Lokalisierung sind Hauskoordinaten, die beim Katasteramt erworben bzw. abgefragt werden können, vorausgesetzt worden, da sich aus Erfahrungswerten gezeigt hat, dass die Lokalisierung über eine Adresse in nahezu allen Fällen im Störungsmanagement durchgeführt wird. Durch die Trennung der Funktionalität von Adressen- und WMS-Service, kann der WMS-Service zur Lokalisierung von jeglichen geographischen Objekten, denen eine Koordinate zugeordnet werden kann, eingesetzt werden.

## 6 Evaluierung der Zielsetzungen

Die Zielsetzung zur Entwicklung und Realisierung eines Geschäftsprozesses mit Geodatennutzung mittels Standards in einer aktuellen Software-Architektur wurde im Kapitel 5 Umsetzung des gewählten Ansatzes detailliert beschrieben und wird daher im folgenden Kapitel nur kurz zusammengefasst. Somit wurden die Fragen zur möglichen Integration der räumlich expliziten Funktionen in die Unternehmens-IT samt einer konkreten Umsetzung beantwortet. Im Weiteren ist zu diskutieren, inwiefern sich sowohl die in Form von Web Services beschriebenen Teilprozesse als auch die mit BPEL erstellten Abläufe wiederverwenden lassen und welche Vor- und Nachteile sich aus der Verwendung von Standards bei der Abbildung von Geschäftsprozessen bei Energieversorgungsunternehmen ergeben. Ebenso wird die Entwicklung eines übergeordneten Konzeptes für die Nutzung von Geodaten in ausgewählten Geschäftsprozessen diskutiert und bewertet.

### 6.1 Prototypische Realisierung eines Geschäftsprozesses mit Geodatennutzung

Das Ziel der Entwicklung und prototypischen Umsetzung eines Geschäftsprozesses mit Geodatennutzung wurde erreicht. Wie einleitend in den Zielen gefordert, sind in der Umsetzung ausschließlich existierende Standards verwendet worden. Des Weiteren wurde die Service-orientierte Architektur als Gesamtumgebung für die Realisierung gewählt. So hat die technische Umsetzung des Teilprozesses Störungslokalisierungs-Service aus dem Prozess Störfallmanagement eines Energieversorgungsunternehmens gezeigt, dass mit der Web Service Technologie geographische Dienste mit nichtgeographischen Diensten zu einem Prozess orchestriert werden können. Die ausführliche Beschreibung der Realisierung ist Kapitel 5 Umsetzung des gewählten Ansatzes zu entnehmen. So wird die Frage einer möglichen Integration von räumlich expliziten Funktionen in die Unternehmens-IT positiv beantwortet. Jedoch ist kein kompletter Hauptprozess sondern nur ein Teilprozess und zwar die Störungslokalisierung des Hauptprozesses Störungsmanagement umgesetzt worden. Dies begründet sich in zwei Aspekten. Einerseits ist es nicht sinnvoll, einen kompletten Prozess von Anfang bis Ende abzubilden, da zu viele unternehmensspezifische Faktoren Einfluss nehmen und eine Verallgemeinerung nicht möglich wäre. Dies wird in Kapitel Prozessvorgabe durch übergeordnete Institutionen weiter bearbeitet. Weiterhin lag der Schwerpunkt der Arbeit auf der Entwicklung und Realisierung der Geodatennutzung in Prozessen und nicht in der Programmierung des Ablaufes eines bestimmten Prozesses. Die verbleibenden Forschungsfragen und Ziele lassen sich auf den bearbeiteten Teilprozess übertragen und werden in den folgenden Kapiteln diskutiert.

### 6.2 Wiederverwendbarkeit als Beitrag zur künftigen Nutzung

Die Wiederverwendbarkeit von Web Services ist in verschiedenen ihrer Merkmale begründet. So kann zunächst die Kapselung erwähnt werden, die den Web Service eine in sich abgeschlossene Funktionalität sein lässt, die eine Aufgabe erfüllt. Über die mitgeführten Metadaten ist der Web Service selbstbeschreibend und kann über eine lose Kopplung, die zur Laufzeit stattfindet, angesprochen werden. Operationen und Nachrichten des Web Service werden über allgemein anerkannte Protokolle, beispielsweise HTTP, unterstützt. Des Weiteren können die Web Services in weitere Web Services zerlegt werden oder zu einem neuen Web Service zusammengestellt werden. Die beiden entwickelten Web Services „Adressen-Service“ und „WMS-Service“ können somit losgelöst voneinander weiteren Anwendungen zugeordnet werden. Dies kann über den Aufruf der jeweiligen WSDL-Datei aus einem BPEL-Prozess heraus erfolgen. Der Grad der Wiederverwendung differiert zwischen den erstellten Services.

Der „Adressen-Service“ dient als nichtgeographischer Web Service dazu, anhand von bestimmten Eingabekriterien (hier die Adresse) einen Datensatz in der Datenbank ausfindig zu machen und ein Attribut dieses Datensatzes auszugeben, welches im Beispiel der Rechts- und Hochwert der Hauskoordinate ist. In allgemeiner Form kann der Web Services als „Attributausgabe“-Service bezeichnet werden. Die Ein- und Ausgabeparameter lassen sich nach Bedarf modifizieren.

Der geographische „WMS-Service“ liefert anhand einer Eingangskoordinate eine darauf zentrierte Karte mit Kataster, Hauskoordinaten und Leitungen zurück. Diese Funktionalität ist auch bei anderen Geschäftsprozessen

von Relevanz. So werden beispielsweise neben den Kunden auch Betriebsmittel aus der Instandhaltung heraus lokalisiert. Wird dieser Geo Web Service via Internet bereitgestellt, ist sogar eine Nutzung der Funktionalität von externen Prozessen, die auf dieselben Daten zugreifen denkbar. Für die beschriebene Verwendung ist keine Anpassung des WMS-Service erforderlich. Soll der Kartenausschnitt, den der WMS-Service liefert, verändert werden, so muss die Berechnung der BoundingBox als Methode in der Java-Klasse angepasst werden. Weitere Änderungen in der Java-Klasse sind erforderlich, wenn andere Themen in der Karte angezeigt werden sollen. Die dargestellten GIS-Layer können ebenfalls in einer Java-Methode zusammengesetzt werden und die WMS-URL komplettieren. Weiterhin denkbar ist die Verwendung von Katasterdaten wie beispielsweise ALK-Daten oder Orthophotos, direkt vom Katasteramt, so dass lediglich die Fachdaten des Versorgungsunternehmens lokal vorhanden sind. Aktuell gibt es bereits Katasterämter, die ihre Geodaten in Form eines WMS-Service zur Verfügung stellen, so dass kein Sekundärdatenbestand beim Kunden unterhalten werden muss. Eine denkbare Umsetzung wäre ein weiterer Web Service, der die WMS-URL der Katasterdaten liefert, welche wiederum in einem Web Service mit der bestehenden WMS-URL der Fachdaten überlagert wird.

Da sich ein BPEL-Prozess aus Web Services zusammensetzt und selbst wie ein Web Service behandelt werden kann, ist auch hier die Wiederverwendbarkeit anzutreffen, die jedoch nicht so hoch ist wie bei einem Web Service, der nur ein Funktionspaket enthält. Der entwickelte Störungs-Lokalisierungs-Prozess sendet aufgrund einer eingegebenen Adresse eine Karte zurück. Diese Funktionalität bietet sich auch zur Nutzung bei der Kundenbetreuung an. Beispielsweise kann dieser Prozess ohne Modifizierung neben der Störungslokalisierung auch bei der Lokalisierung der Adresse für einen neuen Hausanschluss oder Abnehmerwechsel verwendet werden. Des Weiteren kann dieser Prozess bei Unternehmen, die die gleiche Oracle-Datenbank samt Datenstruktur besitzen bereitgestellt und verwendet werden. Nutzen die Unternehmen eine andere Struktur oder andere GIS-Themen, so ist lediglich eine geringe Anpassung der Java-Klassen hinsichtlich der Datenbankverbindung und den Spaltenbezeichnungen notwendig. Die PartnerLinks sind Elemente, die im BPEL-Code verändert werden müssen.

### 6.3 Tauglichkeit der verwendeten Standards

Um ein höchstmögliches Maß an Flexibilität und Wiederverwendbarkeit zu erreichen, bauen die Umsetzungen und Untersuchungen dieser Aufgabenstellung auf Standards und Spezifikationen der zuständigen Institutionen auf. In den folgenden zwei Abschnitten wird heraus gearbeitet, inwiefern die verwendeten Standards den Anforderungen sowohl in der verwendeten Technologie als auch in der Abarbeitung von Geschäftsprozessen mit Geodatenanbindung genügen.

Durch die Verwendung von offenen Standards in der Web Service Technologie entsteht ein klarer Kostenvorteil, da zu diesen auch die allgegenwärtigen internetbasierten Technologien gehören, wodurch Lizenzkosten vermieden werden und ein Einstieg in diese Thematik im Vergleich zu neuen Technologien recht gering ist. Da üblicherweise HTTP zur Datenübertragung verwendet wird, gibt es selten Probleme mit Firewalls, die bei vergleichbaren Übertragungsmöglichkeiten wie CORBA, DCOM oder auch Java RMI auftreten. HTTP ist jedoch nur eine von vielen denkbaren Übertragungsarten, da Web Services nicht an HTTP gebunden sind, sondern auch andere Protokolle, wie beispielsweise SMTP nutzen können. Durch die Nutzung dieser offenen Standards wird die Web Service Technologie in einer Service-orientierten Umgebung eine offene und flexible Architektur, die unabhängig von den verwendeten Plattformen und deren Programmiersprachen sowie den Protokollen ist. Dadurch sind beispielsweise Windows-C#-Clients in der Lage, hinter einer Firewall mit Java-Servern einer Linux-Umgebung zu kommunizieren (Wikipedia, 2006g). Das ermöglicht eine Interoperabilität im und durch das Internet.

Eine große Herausforderung besteht derzeit noch darin, dass die echte Interoperabilität von Web Services erst dann erreicht ist, wenn entsprechende Implementationen diese zeigen. Das bedeutet, dass wesentlich komplexere Anwendungen als das für diese Aufgabenstellung umgesetzte Beispiel realisiert werden müssen, um die zugehörigen komplexeren Standards für Web Services entwickeln, testen und anwenden zu können.

Die Web Services Interoperability Organisation (WS-I) leistet einen großen Beitrag zur Erweiterung der Kernspezifikationen beispielsweise in den Bereichen Sicherheit, Komposition, Management und Interoperabilität. Der Unterschied vom WS-I zu anderen Gremien liegt darin, dass die Organisation versucht, bestehende Standards verschiedener Gremien zu einem so genannten "Basic Profile" zusammenzufassen. Es verwendet also einige

Spezifikationen aus einem Bereich, die Kommunikation und Austausch von Daten und Funktionalität der verschiedenen Implementierungen auf Basis dieser Profile gewährleisten sollen. Weiterhin spielen auch die weniger greifbaren Faktoren wie Güte bzw. Qualität und Verfügbarkeit eines Web Services eine große Rolle. So sind die Spezifikationen, die über den Grundbedarf hinausgehen, sehr jung und unterliegen noch dem Reifeprozess. Im Folgenden werden solche Spezifikationen aus den Bereichen des Web Services Standards Stack aufgelistet (InnoQ, 2007): Messaging Specifications, Metadata Specifications, Security Specifications, Reliability Specifications, Resource Specifications, Management Specifications, Business Process Specifications, Transaction Specifications, Presentation Specifications.

Ein weiteres Augenmerk liegt laut (Wikipedia, 2006g) auf der Performance von Web Service Technologien, die durch XML, Parsen und die Dateigröße negativ beeinflusst wird. Derzeit ist der Verwaltungsaufwand bei stark verteilten Systemen groß und der Overhead um die eigentlichen Nutzdaten teilweise erheblich. Die Erweiterung BPEL4people der Prozesssteuerungssprache BPEL ist ebenfalls ein Schritt in ein weiteres Standardisierungsfeld. Da Web Services an sich nur für die Kommunikation unter Maschinen gedacht sind, eine menschliche Interaktion jedoch meist der Startpunkt für einen Prozess ist bzw. sich nie ganz vermeiden lässt, gewinnt das von IBM und SAP eingereichte Whitepaper immer mehr an Bedeutung.

WS-BPEL 2.0 an sich als noch recht junger Standard mit Release im April 2007, ist noch nicht oder bisher nur in Extrakten in die Softwareprodukte eingebracht worden, so dass über die Verwendung dieses Standards in der Version 2.0 keine Untersuchungen gemacht werden können. Die neue Version beinhaltet Detail-Verbesserungen, ist jedoch nicht kompatibel zu den Vorgängerversionen, was wiederum einen Migrationsaufwand mit sich bringt.

Für das umgesetzte Beispiel haben sowohl die GIS-Funktionalität (reine Visualisierung) als auch die geforderten Abläufe in BPEL ausgereicht. Zwar wird die aktuelle Störung im Kartenmittelpunkt ohne Symbol oder Markierung symbolisiert, was jedoch mit einer geeigneten Beschriftung der Grundkarte kompensiert werden kann, da es sich ohnehin um nur eine Störung, die es zu lokalisieren gilt, handelt.

Betrachtet man jedoch komplexere Abläufe oder Services, so stoßen die verwendeten Standards WMS und BPEL an ihre Grenzen. In der entwickelten Störungslokalisierung genügt die GetMap-Anfrage der Web Map Service Spezifikation, um die gewünschte Störung auf einer Karte zu lokalisieren.

Betrachtet man den weiteren Ablauf des Entstörungsprozesses, so wird mindestens eine weitere Stelle, an der im Prozess mit Geodaten gearbeitet werden muss, relevant: der Störungs-Priorisierungs-Service, der eine Karte mit allen aktuellen Störungen im Versorgungsgebiet liefert. Die Ausgabe der Karten-URL stellt dabei keine zusätzliche Aufgabe dar, da sie im Beispielprozess bereits umgesetzt wurde. Vielmehr muss eine Lösung für die Speicherung von Störungslokalitäten im Zusammenhang mit gemeldeten Störungen erfolgen, die zusätzlich zu den Basisdaten der Grundkarte und des Leitungsnetzes auf der Karte eingeblendet werden. Existierende Störungen, beispielsweise die vom Entstördienst, werden auf einem mobilen Gerät erfasst und in den Hauptdatenbestand übertragen. Sie können als weiteres Thema im Map Builder angelegt und über den Map Viewer via WMS bereitgestellt werden.

Komplexer wird die Darstellung des Störungsmelders, der zu diesem Zeitpunkt des Entstörprozesses auch den Störungsort darstellt. Mit existierenden Störungen, die über Symbole in der Karte angezeigt werden, fällt es dem Betrachter zunehmend schwerer im Gesamtkontext die Stelle zu lokalisieren, die zwar im Mittelpunkt der Karte liegt, jedoch ohne weitere Markierung dargestellt wird. Somit ist es für diese Art der Lokalisierung erforderlich, dass die Koordinate der aktuellen Melderadresse entweder in einem eigenen Thema oder gemeinsam mit allen anderen Störungen abgelegt wird. In diesem Fall wird neben der reinen Visualisierung von Geodaten eine Erstellung bzw. Bearbeitung als Speicherung von Koordinaten derselbigen vorgenommen. Diese Funktionalität muss nicht notwendigerweise in einem GIS vorgenommen werden, sondern kann entweder über einen Web Service oder einen BPEL-Prozess abgedeckt werden.

Betrachtet man weitere Abläufe im Entstörprozess, die jedoch im Rahmen der Modifikation des Referenzmodells dem Bereich Instandhaltung zugeordnet wurden, so stellt die Netzverfolgung für Sperrstrecken oder unver sorgte Kunden ausgehend von einem gestörten Betriebsmittel neben der Visualisierung eine reine GIS-Funktionalität dar, die über einen Web Service bereit gestellt werden müsste. Lösung kann hier eine Abfrage in

Spatial SQL sein, welches die räumliche Abfrage direkt in der Datenbank von Oracle vornimmt. Ergebnis ist eine Liste aller Elemente, die bei der Netzverfolgung ermittelt wurden, welches für die Abfrage für unversorgte Kunden ausreichend ist, um daraus wiederum einen Serienbrief zu generieren. Um eine dazu passende Karte zu erhalten, muss entweder die Netzverfolgung ebenso wie WMS oder WFS als Standard Voraussetzung sein, oder die ausgegebene Liste auf entsprechendem Kartenmaterial graphisch dargestellt werden, was sich als eine komplexere Aufgabenstellung im Vergleich zur reinen Liste erweist und mit den bisher existierenden Standards nicht noch umgesetzt werden kann.

Bei der Einteilung des Störfallmanagements in einzelne Web Services, konnte die menschliche Interaktion, die originär nicht von Web Services und BPEL unterstützt wird, nicht vollkommen außer acht gelassen werden, da diese in nahezu jedem Arbeitsschritt erforderlich ist. Somit gewinnt auch an dieser Stelle die Erweiterung der Prozesssteuerungssprache BPEL4people eine große Bedeutung.

## 6.4 Prozessvorgabe durch übergeordnete Institutionen

Weiterhin lässt sich aus der Wiederverwendbarkeit schlussfolgern, dass generell die Möglichkeit besteht, ein Basis-Prozessmodell, wie es der DVGW anbietet, für alle Energieversorgungsunternehmen zur Nutzung zu entwickeln, was aufgrund der Anwendung von Standards theoretisch machbar sein sollte. Dem entgegen sprechen verschiedene Punkte, die bei der Gestaltung und Entwicklung von Geschäftsprozessen hinreichend bedacht werden müssen.

Die betriebenen Sparten und die Struktur des Versorgungsgebietes, wie sie im Kapitel 2.2.1 Strukturierung und Arten von Energieversorgungsunternehmen erläutert wurden, müssen als relevante Aspekte betrachtet werden. So gibt es auf der einen Seite Flächenversorger, die in kleineren Städten oder Überlandgebieten oft nur ein Medium betreuen. Auf der anderen Seite stehen die Stadtwerke, die meist mehrere Versorgungsmedien in einem überschaubaren Gebiet betreiben. Die Arbeitsabläufe werden in beiden Varianten oft anders gestaltet. So verhält sich beispielsweise das Störungsmanagement für Gasgeruch im Detail anders als für einen Stromausfall. Die Störmeldung an sich kommt beim größeren Unternehmen in einem Call-Center und beim kleineren direkt in der Leitwarte an. Diese Unterschiede in der Bearbeitung sind zu gravierend, als dass man ein einheitliches Prozessmodell für alle Energieversorgungsunternehmen entwickeln und vorschreiben könnte.

Jedoch lassen sich für alle prozessspezifischen Web Services in einem Prozess einige allgemeine Web Services oder Prozesse definieren. Voraussetzung ist die Verwendung von Standards in den Web Services bzw. Prozessen, wie beispielsweise die Anbindung von GIS-Daten über WMS. Um die Web Services oder Prozesse auf andere Unternehmen übertragen zu können, müssten lediglich einige Parameter bezüglich Datenbankbindung und Feld- bzw. Spaltenbezeichnungen angepasst werden. Die Entwicklung allgemeingültiger Web Services oder Teilprozesse kann eine Aufgabe von Institutionen oder Verbänden (z.B. der DVGW) sein, um einerseits eine Akzeptanz bei den Unternehmen und andererseits eine Unabhängigkeit von Herstellern zu erreichen.

## 6.5 Bewertung

Abschließend zu den Untersuchungen zur Verwendung von Standards bei einer Anbindung von Geodaten an die Unternehmens-IT zur Durchführung von Geschäftsprozessen und der Wiederverwendbarkeit der dafür entwickelten Web Services und Prozesse sowie der Prozessvorgabe durch eine übergeordnete Institution lassen sich die nachstehenden Bewertungen schlussfolgern.

Die Web Service Technologie bietet der Geoinformatik die Chance, sich aus ihrem bisherigen Inseldasein zu lösen und ein Teil der allgemeinen Informatik mit Geobezug zu werden. Dies ist zur fließenden Abarbeitung der Geschäftsprozesse eines Energieversorgungsunternehmens ohne Medienbrüche erforderlich. Die Kapselung von einzelnen Funktionsbausteinen in Web Services bietet eine hohe Wiederverwendbarkeit, so dass alleine durch diesen Vorteil die gesamte Technologie einen Mehrwert gegenüber anderen darstellt. Aufgrund der Plattform- und Programmiersprachen-Unabhängigkeit kann jeder Entwickler und Anwender diese Technologie nutzen, ohne seine bisherige IT-Infrastruktur verändern oder anpassen zu müssen.

Inhaltlich gesehen ist die Generierung der WMS-URL bzw. einer Karte aufgrund einer Eingangskoordinate keine neue Funktion, die Umsetzung dieser Funktionalität über nichtgeographische Web Services jedoch schon, da der Schwerpunkt nicht auf dem Inhalt sondern auf der verwendeten Technologie liegt. Benötigt man neben der reinen Kartendarstellung auch echte GIS-Funktionen, so bedeutet dies, dass sämtliche GIS-Funktionalitäten als einzelne Web Services innerhalb der Web Service Technologie zur Verfügung stehen müssen: Ein GIS-System wäre in einer Service-orientierten Architektur eine fachliche Gruppierung von Web Services, die Geodaten verarbeiten und zur Laufzeit angesprochen werden. Hier besteht weiterer Forschungsbedarf, da weitere Standards neben WMS und WFS benötigt werden, um die GIS-Funktionalität über einen Web Service für jeden Nutzer anwendbar zu machen.

Alleine für die Visualisierung der Störungsstelle ist kein GIS-System erforderlich, es reichen eine räumliche Datenbank (hier: Oracle Datenbank) und ein entsprechender Map Server (Map Viewer auf dem Oracle Application Server) aus, um über die bestehenden OGC-Standards WMS und WFS eine Karte zu generieren. Der weitere Aufwand zur Erzeugung eines geographischen Web Services nach den Regeln der Web Service Technologie ist zwar zum heutigen Zeitpunkt mit der Verpackung des URL-Abrufes in beispielsweise einer Java-Klasse noch etwas umständlich, aber erforderlich, da für das Zusammenspiel der Web Services generell und für die Orchestrierung mit BPEL eine Beschreibung des Web Service (WSDL-Datei) notwendig ist. Da jedoch die Zusammenarbeit zwischen W<sup>3</sup>C und dem OGC durch dessen Beitritt zum W<sup>3</sup>C im Januar 2007 stark voranschreitet, ist zu erwarten, dass für diese Schritte neue Lösungswege geschaffen werden. Des Weiteren sind OWS-Initiativen aktiv, die die Anpassung von geographischen Web Services an die allgemeine Informatik untersuchen. Diese Initiativen haben bisher aber noch keine konkreten Vorgehensweisen sondern lediglich so genannte discussion papers hervorgebracht.

Die Prozessvorgabe über einzelne Web Services, welche auch Teilprozesse sein können, ist ein weiteres anzustrebendes Ziel. Hier besteht noch sehr hoher Bedarf an Analyse der prozessspezifischen Services für die komplette Abwicklung der Geschäftsprozesse eines Energieversorgungsunternehmens. Ein Test des entwickelten Störungslokalisierungs-Service in einer anderen Umgebung, um beispielsweise ein anderes Unternehmen zu simulieren, war zum Zeitpunkt der Umsetzung nicht möglich, da die Web Service-Technologie für diesen Bereich noch keine Anwendungen in der Praxis bietet.

## 7 Abschlussbewertung und Ausblick

Durch diese Arbeit wird ein Beitrag zur Eingliederung der Geoinformatik in die allgemeine Informatik geleistet. Diese Arbeit untersucht die standardbasierte Einbindung von räumlich expliziten Funktionen in die Unternehmens-IT. Als fachlicher Hintergrund dient der Geschäftsprozess Störfallmanagement des Kernprozesses Betreiben und Instandhalten bei Energieversorgungsunternehmen.

Es werden allgemein die bestehenden Möglichkeiten zur Anbindung von Geodaten in die Geschäftsprozesse untersucht. Voraussetzungen für die Untersuchungen ist die Verwendung von Standards sowohl auf Seiten der Geoinformatik als auch auf Seiten der allgemeinen Informatik. Ergebnis dieser technischen Diskussion ist eine Anbindung der Geodaten in die Geschäftsprozesse mit der Web Service Technologie in einer Service-orientierten Architektur, da mit dieser Umsetzung der höchstmögliche Grad an Wiederverwendung und Flexibilität erreicht werden soll.

Um dieses Ergebnis in eine konkrete Umsetzung münden zu lassen, werden auf inhaltlicher Seite die Geschäftsprozesse von Energieversorgungsunternehmen anhand des Referenzmodells für GIS-gestützte Geschäftsprozesse, herausgegeben im Jahre 2003 vom DVGW, herausgearbeitet. Auf diesem Modell aufbauend hat der DVGW im Dezember 2005 einen Technischen Hinweis für DV-gestütztes Störfallmanagement veröffentlicht. Der Hauptprozess Störfallmanagement aus dem Kernprozess Betreiben und Instandhalten dient als Basis für die Realisierung. Der Prozess wird inhaltlich diskutiert und im Ablauf modifiziert sowie in die Struktur der Web Service Technologie gebracht, welches eine Aufteilung der einzelnen Arbeitsschritte und Funktionen in Web Services beinhaltet. Um eine Allgemeingültigkeit für die Umsetzung zu erreichen, werden die einzelnen Funktionsbausteine in drei Kategorien unterteilt, von denen die prozessspezifischen Web Services mit Geobezug aufgrund ihrer allgemeinen Gültigkeit zur Umsetzung gebracht werden. Es handelt sich um die Störungslokalisierung zu Beginn des Störfallprozesses, die auf der Adresseingabe der Störmeldung basiert. Es wird eine Realisierung der Störungslokalisierung über zwei separate Web Services, die in einem BPEL-Prozess orchestriert werden, angestrebt, die in einer ORACLE-Umgebung umgesetzt werden soll. In dieser Arbeit wird ein Prototyp der Störungslokalisierung von einer Adresse auf einer Grundkarte mit Leitungsnetz entwickelt. Über Java-Klassen wird die Zusammensetzung und Abfrage der WMS-URL in allgemeine Web Services realisiert. Somit wurde ein möglicher Weg zur konkreten Umsetzung aufgezeigt.

Die verwendeten Standards sowohl aus der Geoinformatik als auch aus der allgemeinen Informatik haben für den Zweck der Visualisierung von GIS-Daten in einem Geschäftsprozess ausgereicht. Forschungsbedarf wird in der Kapselung echter GIS-Funktionalitäten in Web Services dahingehend gesehen, dass ein GIS-System zukünftig aus einer fachlichen Gruppierung von Web Services besteht, die über ihre Beschreibung (WSDL-Datei) aufgerufen werden können. Der Forschungsbedarf gliedert sich in die technische Realisierung und in die Weiterentwicklung bestehender sowie die Neuentwicklung weiterer Standards auf diesem Gebiet, da beispielsweise für eine Netzverfolgung die OGC-Standards WMS oder WFS nicht mehr genügen. Die Weiterentwicklung der Standards gilt sowohl für die Geoinformatik als auch für die Web Service Technologie, da bei dieser noch jungen Technologie weitere Spezifikationen in Richtung Sicherheit, Transaktionen, Dienstgüte und Verrechnung anzustreben sind. Erste Ansätze zeigen sich bereits in der Aufgabenstellung des OGC, welches Initiativen zur Eingliederung und Anpassung von Geo Web Services an „normale“ Web Services gebildet hat. Eine zum Zeitpunkt der Niederschrift gestartete Initiative ist OWS-5, deren Vorgänger jedoch bisher nur so genannte Recommendation Papers hervorgebracht haben. Auch laut (Kiehle, 2006) ist der Aufbau eines Geodienstes für den produktiven Einsatz bislang noch nicht sinnvoll. Ein weiterer Schritt in die angestrebte Verschmelzung von Geoinformatik mit der allgemeinen Informatik ist der Beitritt des OGC zum W<sup>3</sup>C im Januar 2007. Aus dieser Zusammenarbeit sind sicherlich auch grundlegende Erkenntnisse für die gewünschte Entwicklung zu erwarten.

Die erhoffte Wiederverwendbarkeit einzelner Services und des gesamten Prozesse hat die Erwartungen an die Technologie erfüllt. Die Analyse zur Wiederverwendbarkeit ergibt, dass sowohl der nichtgeographische Web Service, der zu einer eingegeben Adresse eine Koordinate aus der Datenbank holt, sowie der geographische Web Service, der aufgrund einer Koordinate eine URL mit einer Karte liefert, in Geschäftsprozessen anderer Hauptprozesse Anwendung finden. Auch der BPEL-Prozess als solches, der anhand einer Adresse eine Karten-URL

ausgibt, kann im Unternehmen an anderer Stelle weitere Verwendung finden. Des Weiteren kann bei einer öffentlichen Bereitstellung auch von extern auf die Web Services oder den Prozess zugegriffen werden. Ebenso können extern vorgehaltene Informationen, die durch Web Services bereitgestellt werden, in eigenen Prozessen eingebunden werden.

Ein wesentlicher Erfolgsfaktor für die Web Service Technologie in einer Service-orientierten Architektur ist, dass jeder Nutzer und Entwickler von Web Services gezwungen wird, die bestehenden Standards einzuhalten, welches in dieser Arbeit gleichermaßen die Voraussetzung war. Für eine unternehmensübergreifende Anwendung von Web Services ist der Einsatz von Standards somit unabdingbar.

## 8 Literaturverzeichnis

- AED-SICAD** (2007a): Webseite der Firma AED-SICAD; [www.aed-sicad.de](http://www.aed-sicad.de); (14.04.2007).
- AED-SICAD** (2007b): White Paper ArcFM™ UT Integrator; [http://www.aed-sicad.de/pages/produkte/applikationen/utilities/ut\\_on\\_arcgis/dateien/20050902-WhitePaperArcFM-UT%20Integrator-d.pdf](http://www.aed-sicad.de/pages/produkte/applikationen/utilities/ut_on_arcgis/dateien/20050902-WhitePaperArcFM-UT%20Integrator-d.pdf); (14.04.2007).
- Bartelme** (2005): Geoinformatik: Modelle, Strukturen, Funktionen; Springer Verlag; Berlin; 2005.
- Bator** (2007): Enterprise Application Integration; <http://www.devmatic.de/main.php?lang=de&page=0&article=articles/eai.html&full=1>; (20.02.2007).
- Bauer** (2006): Entwurf und Bewertung von Realisierungsvarianten für ein verteiltes Monitorsystem auf Basis von Web Services; Diplomarbeit an der Uni Siegen; unveröffentlicht; [http://www.bs.informatik.uni-siegen.de/web/wismueller/arbeiten/2006\\_bauer.pdf](http://www.bs.informatik.uni-siegen.de/web/wismueller/arbeiten/2006_bauer.pdf); (16.05.2007).
- Behr** (2002): Strategisches GIS-Management – Grundlagen, Systemeinführung und Betrieb; Herbert Wichmann Verlag; Heidelberg; 2000.
- BERIT** (2007): TOMS – Betriebstechnisches Informationssystem, [www.berit.de](http://www.berit.de), [http://www.lids.de/de/index.php?load=berit\\_toms&PHPESSID=e5894ba41ed478f4b8936d0abafa48c1](http://www.lids.de/de/index.php?load=berit_toms&PHPESSID=e5894ba41ed478f4b8936d0abafa48c1), (05.04.2007).
- Bien** (2004): Das XML der Komponenten; in: javamagazin; 11/2004; S. 37-41; ISSN 169-795X.
- Bill, Zehner** (2001): Lexikon der Geoinformatik; Herbert Wichmann Verlag; Heidelberg; 2001.
- Dhesiaseelan** (2004): What's New in WSDL 2.0?; Fachartikel; <http://www.xml.com/pub/a/ws/2004/05/19/wsd12.html?page=1>; (16.05.2007).
- Donaubauer** (2004): Interoperable Nutzung verteilter Geodatenbanken mittels standardisierter Geo Web Services; Dissertation an der TU München; 2004.
- Dostal, et al.** (2005): Service-orientierte Architekturen mit Web Services; Spektrum Akademischer Verlag; München; 2005.
- DVGW** (2002): Technische Mitteilung Hinweis GW 119; Januar 2002; Verbesserung von Geschäftsprozessen durch die Einbindung von GIS-Systemen; Bonn, 2002.
- DVGW** (2003): GAWANIS - Referenzmodell für GIS-gestützte Geschäftsprozesse; Januar 2003; Bonn; 2003.
- DVGW** (2005): Technische Mitteilung Hinweis GW 133; Dezember 2005; DV-gestütztes Störfallmanagement und Schadensstatistik unter Einbindung von GIS; Bonn; 2005.
- EC-GIS** (2007): The Infrastructure For Spatial Information In Europe; <http://www.ec-gis.org/inspire/home.html>; (23.02.2007).
- Eberhardt, Fischer** (2003): Web Services; Karl Hanser Verlag; München; 2003.
- EnWG** (1998): Gesetz über die Elektrizitäts- und Gasversorgung Ausfertigungsdatum: 24. April 1998; Verkündungsfundstelle: BGBl I 1998, 730.
- ESN** (2007a): TBM Technischer Betriebs-Manager; Produktflyer; <http://www.esn.de/Produkte/Betriebsfuehrung/pdf/TBM.pdf>; (05.04.2007).
- ESN** (2007b): TBM Technischer Betriebs-Manager; Produktbroschüre; [http://www.esn.de/Produkte/Betriebsfuehrung/pdf/TBM\\_Broschuere.pdf](http://www.esn.de/Produkte/Betriebsfuehrung/pdf/TBM_Broschuere.pdf); (05.04.2007).

- ESRI** (2007): ArcNews online; SAP and ESRI Collaborate on Enterprise Services; <http://www.esri.com/news/arcnews/fall06articles/sap-and-esri.html>; (15.05.2007).
- FCIT** (2007a): Webseite zur Lösung GeoXtension; <http://www.geoxtension.de>; (14.05.2007).
- FCIT** (2007b): Systemübergreifende Geschäftsprozesse bei den Stadtwerken Ulm/Neu-Ulm mit BPEL; Projektbericht; [http://www.geoxtension.de/doc/sapphire\\_May10\\_ger.pdf](http://www.geoxtension.de/doc/sapphire_May10_ger.pdf); (14.04.2007).
- FCIT** (2007c): Kooperation mit ESN; <http://www.geoxtension.de/doc/ESN.PDF>; (14.04.2007).
- Ferdinand** (2005): Bearbeitung und Steuerung eines GIS-Prozesses mit Hilfe von Web-Services und BPEL; Diplomarbeit an der TU Darmstadt; unveröffentlicht, 2005.
- Francica** (2007): SAP Takes Another Hard Look at the GIS Market; Directions Magazine; [http://www.directionsmag.com/article.php?article\\_id=2203&trv=1](http://www.directionsmag.com/article.php?article_id=2203&trv=1); (15.05.2007).
- GDMcom** (2007): Webseite der Firma GDMcom; <http://www.gdmcom.de>; (05.04.2007).
- Geoinformatik-Lexikon** (2007): Datenschnittstelle; <http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=458>; (15.05.2007).
- George, Westermann** (2006): Model Driven Architecture in einer service-orientierten Anwendungslandschaft, JavaSpektrum 1/2006; S. 18-21;2006.
- GISZH** (2007): GIS-Zentrum des Kantons Zürich; [http://www.giszh.zh.ch/internet/bd/arv/gis-de/was\\_ist\\_gis\\_datenformen.html](http://www.giszh.zh.ch/internet/bd/arv/gis-de/was_ist_gis_datenformen.html); (20.02.2007).
- GIUP** (2007): Geographische Institut der Universität Bonn; das GIS-Tutorial; <http://www.giub.uni-bonn.de/gistutor/>; (20.02.2007).
- GreenGate** (2007a): Funktionsumfang GS-Service; Leistungsbeschreibung; [http://www.greengate.de/Funktionsumfang\\_GS.pdf](http://www.greengate.de/Funktionsumfang_GS.pdf); (05.04.2007).
- GreenGate** (2007b): Betriebs-Management mit GS-Service; Funktionsbeschreibung; [http://www.greengate.de/Betriebsmanagement\\_mit\\_GS.pdf](http://www.greengate.de/Betriebsmanagement_mit_GS.pdf); (05.04.2007).
- Hansen, Neumann** (2005): Wirtschaftsinformatik 2; Informationstechnik; Lucius & Lucius GmbH; Stuttgart; 2005.
- Hantschel, Ruf, Strotbek** (2006): Vergleich von BPEL Laufzeitumgebungen; Fachstudie Nr. 54 am Institut für Architektur von Anwendungssystemen an der Universität Stuttgart; 2006.
- Hauffe** (2004): GIS-Integration in die Geschäftsprozesse eines regionalen Energieversorgers; Vortrag zum AGIS-Seminar am 16.09.2004; [http://www.agis.unibw-muenchen.de/HTML/Weiterbildung/15\\_Hauffe.pdf](http://www.agis.unibw-muenchen.de/HTML/Weiterbildung/15_Hauffe.pdf); (28.05.2006).
- Hauser, Löwer** (2004): Web Services - Die Standards; Galileo Press GmbH, Bonn; 2004.
- Heise** (2005): Oracle dominiert Datenbank-Markt; <http://www.heise.de/newsticker/meldung/48029>; (15.03.2007).
- IMAGI** (2006): Leitfaden für Aufbau und Betrieb webbasierter Geo-Dienste; Informationsschrift des Interministeriellen Ausschusses für Geoinformationswesen (IMAGI); Frankfurt, 2006.
- InnoQ** (2007): Web Services Standards Overview; <http://www.innoq.com/soa/ws-standards/poster/innoQ%20WS-Standards%20Poster%202007-02.pdf>; (19.05.2007).
- Intergraph** (2007a): Webseite zu G!NIUS; <http://www.intergraph.de/ginius/>; (14.04.2007).
- Intergraph** (2007b): Produktinformation zu G!NIUS; <http://www.intergraph.de/GINIUS/GINIUS-Basis-Flyer.pdf>; (14.04.2007).

- ITS** (2007a): Strategisches und Operatives Asset-Management in der Versorgungswirtschaft; [http://www.its-consult.de/images/Hauptseite/ITS\\_Asset\\_Management.pdf](http://www.its-consult.de/images/Hauptseite/ITS_Asset_Management.pdf); (05.04.2007).
- ITS** (2007b): IST MAGIK News; Das Kundenmagazin der ITS Informationstechnik Service GmbH; Ausgabe 6; Oktober 2006; [http://www.its-service.de/pdf/magik\\_news/ITS-MAGIK-NEWS-10-2006.pdf](http://www.its-service.de/pdf/magik_news/ITS-MAGIK-NEWS-10-2006.pdf); (05.04.2007).
- Juric, et al.** (2006): Business Process Execution Language for Web Services; Second Edition; Packt Publishing; Birmingham; ISBN 1-904811-81-7; 2006.
- Kiehle** (2006): Entwicklung einer Geodateninfrastruktur zur regelbasierten Ableitung von Geoinformationen aus distributiven Datenbeständen; Dissertation an der RWTH Aachen; 2006.
- Kleber** (2005): Beitrag zur ALKIS-Implementierung in Hessen und Untersuchung der Anwendersicht; Dissertation an der TU Darmstadt; 2005.
- Kleber** (2006): Standards und Technologien zur Verteilung von Geodaten; Vortrag an der Humboldt Universität zu Berlin; unveröffentlicht; (25.11.2006).
- Kloppmann, et al.** (2005): WS-BPEL Extension for People – BPEL4People; A Joint White Paper by IBM and SAP; July 2005; <ftp://www6.software.ibm.com/software/developer/library/ws-bpel4people.pdf>; (08.05.2007).
- KnowLibrary** (2007a): Kopplung in der Informatik; [http://kopplung.know-library.net/#Kopplung\\_in\\_der\\_Informatik](http://kopplung.know-library.net/#Kopplung_in_der_Informatik); (22.02.2007).
- KnowLibrary** (2007b): Schnittstelle, (<http://schnittstelle.know-library.net/>), (20.2.2007).
- Manhart** (2006): Serviceorientierte Architekturen – Grundlegende Konzepte, Fachartikel; <http://www.tecchannel.de/entwicklung/grundlagen/456248/>; (15.02.2007).
- Mettenmeier** (2007): Smallworld GIS – Fachschale GAS; <http://www.mettenmeier.de/mettenmeier/ge-smallworld-fachschale-gas.htm>; (20.02.2007).
- Mintert** (1998): Auszeichnungssprachen im World Wide Web; <http://www.mintert.com/xml/mlweb/MarkUpLang.html>; (01.02.2007).
- MGIS** (2007a): NOA – die Plattform für den Betrieb von Versorgungsnetzen; [http://www.mgis.de/downloads/NOA\\_%20Flyer\\_2005-10.pdf](http://www.mgis.de/downloads/NOA_%20Flyer_2005-10.pdf); (05.04.2007).
- MGIS** (2007b): Firmenprofil MGIS; [http://www.mgis.de/downloads/MGIS\\_Flyer\\_Image.pdf](http://www.mgis.de/downloads/MGIS_Flyer_Image.pdf); (05.04.2007).
- Näf** (2007): XML-Einführung; <http://www.swisseduc.ch/informatik/material/xml-einfuehrung/index.html/einfuehrung/>; (10.02.2007).
- Noack** (2001): XML 1.0 Grundlagen; Herdt-Verlag; RRZN / Universität Hannover; 2001.
- NRW** (2007): Geodatenbasisportal Nordrhein Westfalen; <https://www.geobasis.nrw.de/>; (23.02.2007).
- OASIS** (2007): Web Services Business Process Execution Language Version 2.0, Committee Draft, 25 January, 2007; <http://www.oasis-open.org/committees/download.php/22036/wsbpel-specification-draft%20candidate%20CD%20Jan%2025%202007.pdf>; (02.03.2007).
- OGC** (2002): Web Feature Service Implementation Specification; OpenGISConsortium; Version 1.1.1; 2002.
- OGC** (2007): OpenGeospatialConsortium; <http://www.opengeospatial.org>; (26.02.2007).
- OMG** (2003): Homepage der Object Management Group; <http://www.omg.org>; (20.12.2006)
- Oracle** (2003): Oracle Spatial User's Guide and Reference; Release 10.1; 2003.
- Oracle** (2005a): Oracle Application Server 10g (10.1.2) MapViewer; Oracle White Paper; 2005.

- Oracle** (2005b): Oracle JDeveloper and Oracle Application Development Framework Pricing FAQ; <http://www.oracle.com/technology/products/jdev/htdocs/jdevpricefaq.html>; 2005.
- Oracle** (2006a): Oracle Application Server MapViewer User's Guide Release 10.1.3.1; 2006.
- Oracle** (2006b): Oracle Oracle SOA Suite Quick Start Guide 10g (10.1.3.1.0); 2006.
- Oracle** (2006c): Oracle BPEL Process Manager Quick Start Guide 10g (10.1.3.1.0); 2006.
- Oracle** (2006d): Oracle SOA Suite Developer's Guide 10g (10.1.3.1.0); 2006.
- Oracle** (2007a): Oracle Corporation; <http://www.oracle.com/>; (19.02.2007).
- Oracle** (2007b): Oracle JDeveloper Overview; Oracle White Paper; 2007.
- Portele** (2005): GML and Web Services – Enabling the Web Geographically; Vortrag auf dem Interoperability Day; <http://www.amfm.it/eventi/2005/interoperabilita/interventi/portele.pdf>; Rom; 2005.
- Pritschet** (2005): Konzeption und Entwicklung eines GIS-Praktikums auf Basis von OGC Web Feature Services; Diplomarbeit an der TU München; unveröffentlicht; 2005.
- Reed** (2004): Integrating Geospatial Standards and Standards Strategies into Business Process; Open GIS Consortium (OGC) White Paper; [portal.opengeospatial.org/files/?artifact\\_id=5098](portal.opengeospatial.org/files/?artifact_id=5098); (07.11.2006).
- Richter, et al.** (2005): Technology Guide SOA; Themenarbeit; software design & management AG; 2005.
- Scheer, Werth** (2005): Geschäftsprozessmanagement und Geschäftsregeln; Heft 183; Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI); 2005.
- Schrenner** (2005): ERP und GIS in EVU; Masterthesis an der Paris-Lodron-Universität Salzburg; unveröffentlicht; 2005.
- Schmietendorf** (2005): Vorlesung Web Services; [http://ivs.cs.uni-magdeburg.de/~schmiete/lehre/vorlesung/paper/ss05/web\\_services\\_ss05\\_324.pdf](http://ivs.cs.uni-magdeburg.de/~schmiete/lehre/vorlesung/paper/ss05/web_services_ss05_324.pdf); (02.03.2007).
- Schwickert, Fischer** (1996): Der Geschäftsprozess als formaler Prozess – Definition, Eigenschaften, Arten; in Arbeitspapiere WI; Nr. 4/1996; Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes-Gutenberg Universität Mainz; 1996.
- Siemen** (2006): Die Integration von TOMS und Smallworld GIS; Vortrag auf der TOMS Roadshow der Firmen Mettenmeier und BERIT; Leipzig, Düsseldorf; September 2006.
- Soutschek** (2007): Die digitale Erde – Die Vision wird Wirklichkeit; [http://www.gis.bv.tum.de/images/Skripten/AndereDokumente/rtgis\\_article\\_google\\_final.pdf](http://www.gis.bv.tum.de/images/Skripten/AndereDokumente/rtgis_article_google_final.pdf); (20.02.2007).
- Steinmann** (2005): Dokumentation alleine war einmal!; Fachartikel in der Zeitschrift EMW; Heft 6/05; 2005.
- Stockwald** (2000): Aktualisierung hybrider geografischer Informationssysteme bei Energieversorgungsunternehmen durch amtliche Geobasisdaten; Dissertation an der Technischen Universität München; 2000.
- Stümpert** (2003): Vorlesung Angewandte Informatik II; <http://www.aifb.uni-karlsruhe.de/Lehre/Sommer2003/AngInfo2/AI2-4-SOAP.pdf>; (01.03.2007).
- Theling, Loos, Sommerrock** (2005): Marktübersicht zu ERP-Literatur, Paper 21 des Lehrstuhl für Wirtschaftsinformatik und BWL an der Johannes Gutenberg Universität Mainz; [http://isym.bwl.uni-mainz.de/downloads/Publikationen/Theling\\_Loos\\_Sommerrock\\_2005\\_Marktuebersicht\\_ERP\\_Literatur.pdf](http://isym.bwl.uni-mainz.de/downloads/Publikationen/Theling_Loos_Sommerrock_2005_Marktuebersicht_ERP_Literatur.pdf); (06.08.2006).
- Ubisense** (2007): Webseite der Firma Ubisens AG; [www.ubisense-consulting.de](http://www.ubisense-consulting.de); (14.04.2007).

**W<sup>3</sup>C** (2003): SOAP Version 1.2 Part 0: Primer; W3C Recommendation 24 June 2003; <http://www.w3.org/TR/soap12-part0/>; (01.03.2007).

**W<sup>3</sup>C** (2006): Web Services Description Language (WSDL) Version 2.0 Part 0: Primer; W3C Candidate Recommendation 27 March 2006; <http://www.w3.org/TR/2006/CR-wsd120-primer-20060327/>; (01.03.2007).

**W<sup>3</sup>C** (2007): World Wide Web Consortium; <http://www.w3.org/>; (01.02.2007).

**Wikipedia** (2006a): ERP-Systeme;

[http://de.wikipedia.org/wiki/Enterprise\\_Resource\\_Planning](http://de.wikipedia.org/wiki/Enterprise_Resource_Planning); (31.10.2006).

**Wikipedia** (2006b): Geoinformationssystem;

[http://de.wikipedia.org/wiki/Geoinformationssystem#NIS:\\_Netzinformationssystem](http://de.wikipedia.org/wiki/Geoinformationssystem#NIS:_Netzinformationssystem); (31.11.2006).

**Wikipedia** (2006c): W3C; <http://de.wikipedia.org/wiki/W3C>; (01.02.2007).

**Wikipedia** (2006d): Schnittstelle; <http://de.wikipedia.org/wiki/Schnittstelle>; (20.03.2007).

**Wikipedia** (2006e): Kopplung; [http://de.wikipedia.org/wiki/Kopplung\\_%28Softwareentwicklung%29](http://de.wikipedia.org/wiki/Kopplung_%28Softwareentwicklung%29); (25.12.2006).

**Wikipedia** (2006f): Integration; [http://de.wikipedia.org/wiki/Integration\\_%28Software%29](http://de.wikipedia.org/wiki/Integration_%28Software%29); (25.12.2006).

**Wikipedia** (2006g): Web Service; [http://de.wikipedia.org/wiki/Web\\_Service](http://de.wikipedia.org/wiki/Web_Service); (25.12.2006).

## Anhang A: Funktionalitäten von GIS

Es werden weitere Funktionalitäten, die als Voraussetzung zur Einbindung von GIS in Geschäftsprozesse gesehen werden, aufgeführt. Die Beschreibung erfolgt in Anlehnung an (DVGW, 2003) und in alphabetischer Reihenfolge.

### Daten analysieren

Die Analyse der im GIS vorgehaltenen Daten deckt einen weiten Bereich ab. Sie bietet von der einfachen thematischen Darstellung (z.B. Markierung von Objekten, die bestimmte Kriterien erfüllen) bis hin zur Verschneidung der Netzinformationen mit systemexternen Daten (z.B. Lifestyle- oder Haushaltsdaten). Qualitätsprüfung der Daten, Aggregation (z.B. Schäden pro km Netz, Kosten pro Hausanschluss) und Flächenverschneidung sind weitere Anwendungsbereiche der Analysefunktion eines Geographischen Informationssystems.

Ziele, die laut (DVGW, 2003) durch diese Funktion erreicht werden können sind:

- Informationsverdichtung,
- Datenübergabe für Managementinformationen (z.B. Geschäftsbericht) und
- Unternehmensdaten (z.B. Leitungslängen, gebaute Leitungslängen pro Jahr).

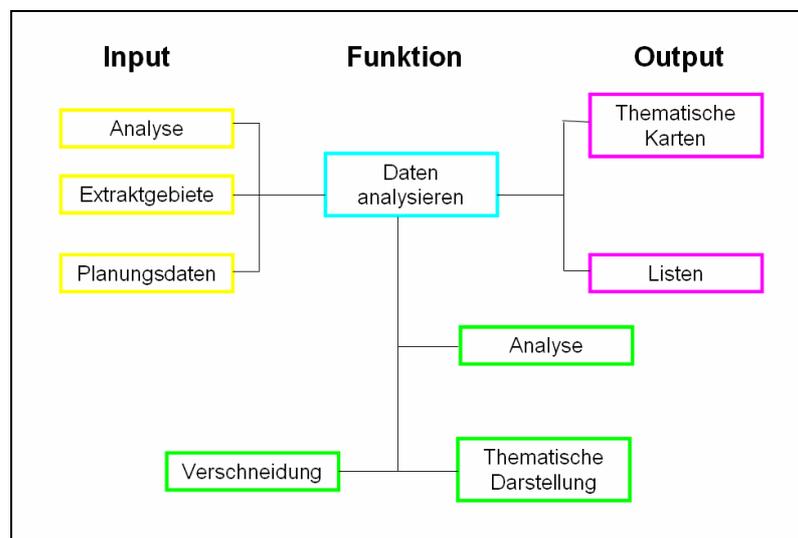


Abbildung: Funktion „Daten analysieren“ nach (DVGW, 2003)

## Konstruktion / Datenerfassung

Für die Erfassung der Leitungsdaten im GIS müssen entsprechende Funktionalitäten bereitgestellt werden. Einerseits werden für die Übernahme der Vermessungsdaten in den GIS-Datenbestand Tools benötigt, die aus den Punktwolken Objekte und Netzstrukturen konstruieren. Andererseits werden neue Objekte aus Feldbüchern oder Handskizzen erstellt, so dass auch diese Art der Datenerfassung abgedeckt sein sollte. Hierzu muss das GIS übliche Standardfunktionen bieten. Die Funktionsziele sind hierbei wie folgt definiert:

- Übernahme von Vermessungsdaten,
- Erzeugung von Objekten aus Punktwolken und
- Konstruktion neuer GIS-Objekte.

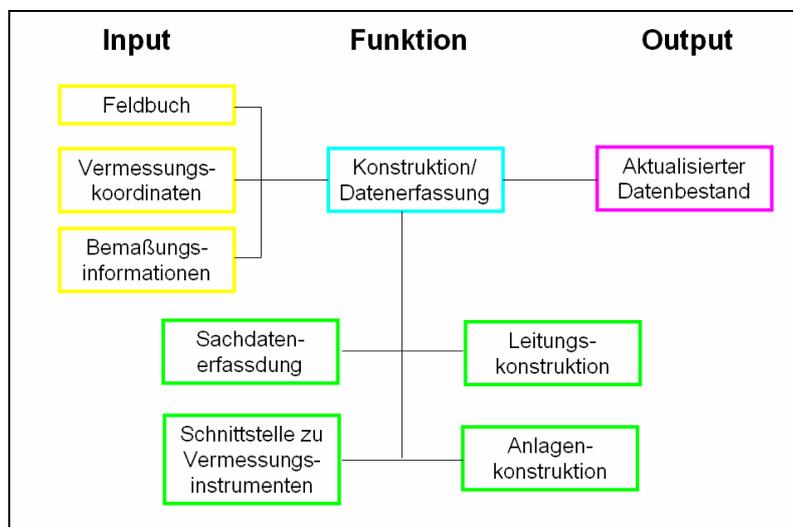


Abbildung: Funktion „Konstruktion/ Datenerfassung“ nach (DVGW, 2003)

## Längen- und Massenermittlung

Um die beschriebenen Prozesse durchgängig zu unterstützen ist eine Zusammenführung von kaufmännischen und technischen Informationen unumgänglich. Für die Kostenschätzung von geplanten oder künftigen Maßnahmen ist eine Längen- und Massenermittlung im GIS-System ein ideales Werkzeug. Auch die Kontrolle von Abrechnungen mit Baufirmen und die Materialverwaltung können mit der Integration dieser Funktion optimiert werden. Zusammengefasst sind diese Ziele von Bedeutung:

- Auflistung von Längen und Massen zu geplanten/gebauten Netzobjekten und
- Übergabe der ermittelten Daten zur weiteren Bearbeitung in andere IT-Systeme (z.B. ERP, Bausoftwarelösungen).

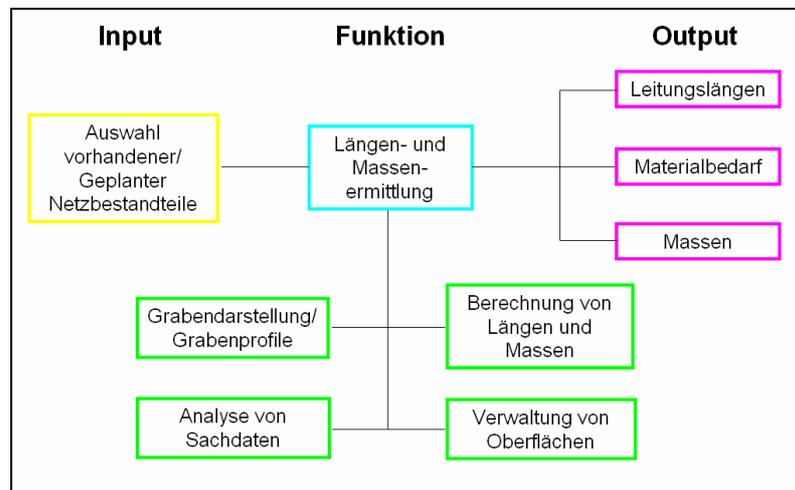


Abbildung: Funktion „Längen- und Massenermittlung“ nach (DVGW, 2003)

## Planungsfunktionalitäten

Aufgaben im Bereich der Netzplanung erfordern, je nach Integrationsgrad einer möglichen Lösung (z.B. Bau-softwarelösungen, AVA-Programme), einige Funktionen, die über den Standard gängiger GIS hinausgehen (z.B. Verwaltung von Planungsvarianten innerhalb des GIS-Datenbestandes). Die Anforderungen reichen von einfacher Längenberechnung bis hin zur Verwaltung von Standard-Grabenprofilen und Formstücken, von der automatisierten Weitergabe von Daten zur Materialplanung/-reservierung und Kostenabschätzung (insbesondere für Ausschreibungen) bis hin zum Controlling. Folgende Ziele zeichnen sich laut (DVGW, 2003) hinsichtlich der Planungsfunktionalitäten ab:

- Integration DV-gestützter Planungen und
- Effizientere Abwicklung von Planungsaufgaben.

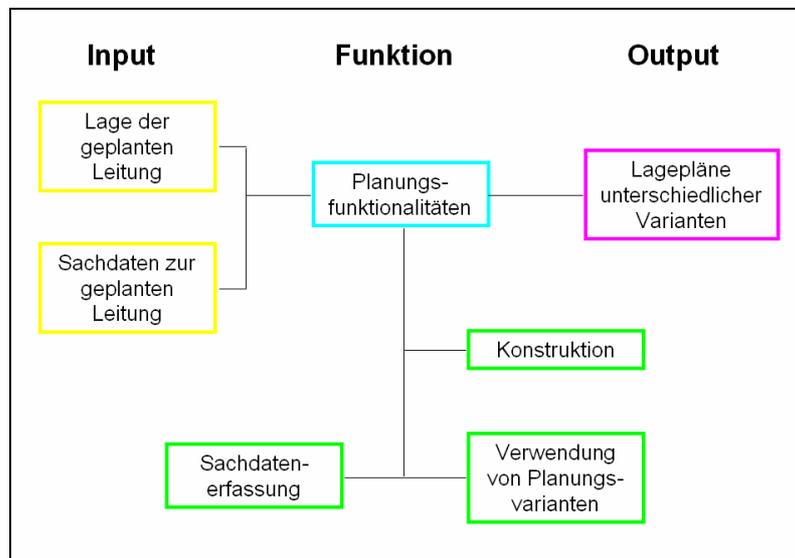


Abbildung: Funktion „Planungsfunktionalität“ nach (DVGW, 2003)

## Anhang B: Beschreibung der existierenden Lösungen

### ArcFM UT mit Utilities Business Package

Das Unternehmen AED-SICAD setzt mit seiner Philosophie auf die GIS-Lösungen SICAD und ArcGIS sowie auf eine ORACLE-Datenbank auf. Die Software ArcFM UT ist eine speziell auf Versorger zugeschnittene Lösung für ein Netzinformationssystem (AED-SICAD, 2007a). Zur Integration von IT-Systemen bietet AED-SICAD zwei Szenarien mit entsprechenden Produkten zur Auswahl an. Handelt es sich um Integration der Daten (EAI-Funktion) auf Datenbankebene, so kommt der UT Integrator-EAI ins Spiel. Möchte man systemübergreifende Prozesse mit Rollenverteilung realisieren, so finden die Utilities Business Packages auf Basis SAP/EP oder GIS-Portal Verwendung. Die Utilities Business Packages sind Bausteine, die bestimmte Teilprozesse in ihrer Abarbeitungsreihenfolge und mit den auf den Prozess-Schritt zugeschnittenen Informationen unterstützen (AED-SICAD, 2007b). Die Lösung auf der Basis SAP Enterprise Portal beruht auf der iView-Technologie von SAP, die die iViews als elementare Bausteine, zugeschnitten auf definierte Teilfunktionen eines IT-Systems, darstellen. Abbildung zeigt das Prinzip. Neben dem Datenzugriff und der dazugehörigen Businesslogik ist auch eine Präsentationskomponente für die Benutzerschnittstelle inbegriffen. Für die verschiedenen SAP-Komponenten existiert bereits eine umfangreiche Anzahl von iViews, die für ArcFM UT sind momentan noch in der Entwicklung sind. Das GIS Portal setzt auf einzelnen Web Services auf, die in einem Portal gebündelt und mit Interaktionsmöglichkeiten ausgestattet werden können.

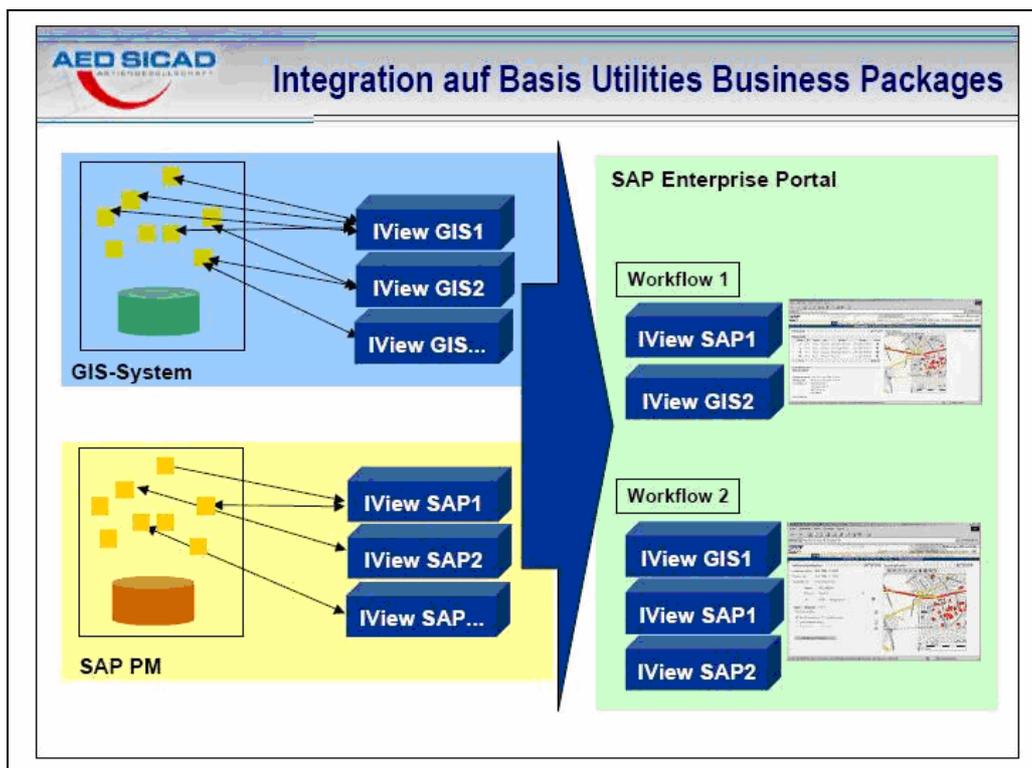


Abbildung: Prinzip Business Packages (AED-SICAD, 2007b)

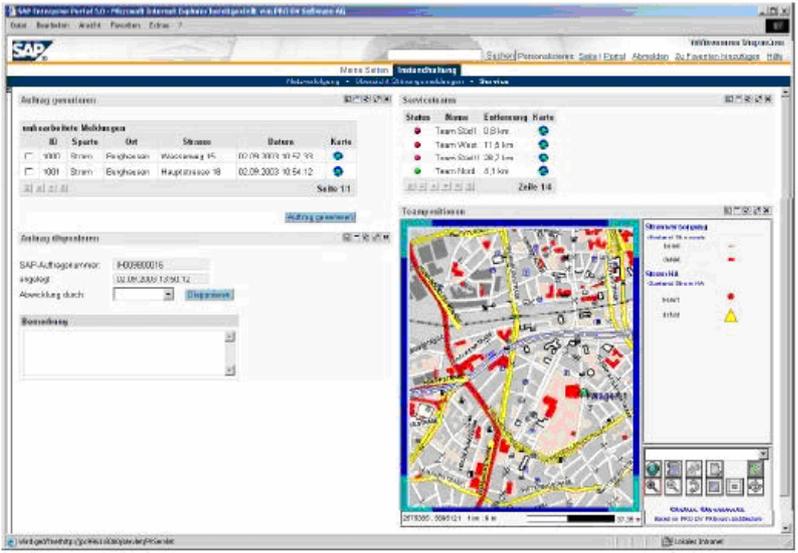
Auf Basis des Lösungsansatzes Utilities Business Packages können alle gewünschten Prozesse, so wie beispielsweise das Störungsmanagement, abgebildet werden.



## Utilities Business Packages: Beispiel

### Beauftragung eines Serviceteams

- ▶ Übersicht der Meldungen, Aufträge und Teams
- ▶ Zusammenstellung der Daten zu einem Auftrag
- ▶ Beauftragung eines Teams



The screenshot displays the SAP Utilities Business Packages interface. It features a central map showing a city street grid with red and yellow markers indicating service locations. To the left of the map, there are data tables for 'Antrag genehmigen' (Request Approval) and 'Antrag abgeben' (Request Submission). The 'Antrag genehmigen' table has columns for 'ID', 'Sparte', 'Ort', 'Strom', 'Datum', and 'Karte'. The 'Antrag abgeben' section includes input fields for 'SAP-Auftragsnummer', 'Angebot', and 'Abwicklung durch', along with a 'Bestätigung' dropdown menu. On the right side, there is a 'Service-Teams' table with columns for 'Status', 'Name', 'Entfernung', and 'Karte'. Below this table, there are options to 'Team zuordnen' (Assign Team) and 'Team freigegeben' (Release Team).

Abbildung: Utilities Business Packages auf Basis SAP/EP (AED-SICAD, 2007b)

## Geo EAM

Die Partnerschaft von ESRI und SAP startete 2005 mit einem gemeinsamen Projekt namens Sagres zum Thema Geo EAM (geo-enterprise asset management). Bisher wurden Geschäftsprozesse, Rollen und damit verbundenen Aufgaben analysiert. Darauf aufbauend entstand bis Mitte Juni 2006 eine Architektur für den Prototyp, welche in Abbildung: Architektur des Prototyps von Sagres (ESRI, 2007) dargestellt wird. Laut (ESRI, 2007) liegt eine Service-orientierte Architektur mit Web Services diesem Prototyp zugrunde.

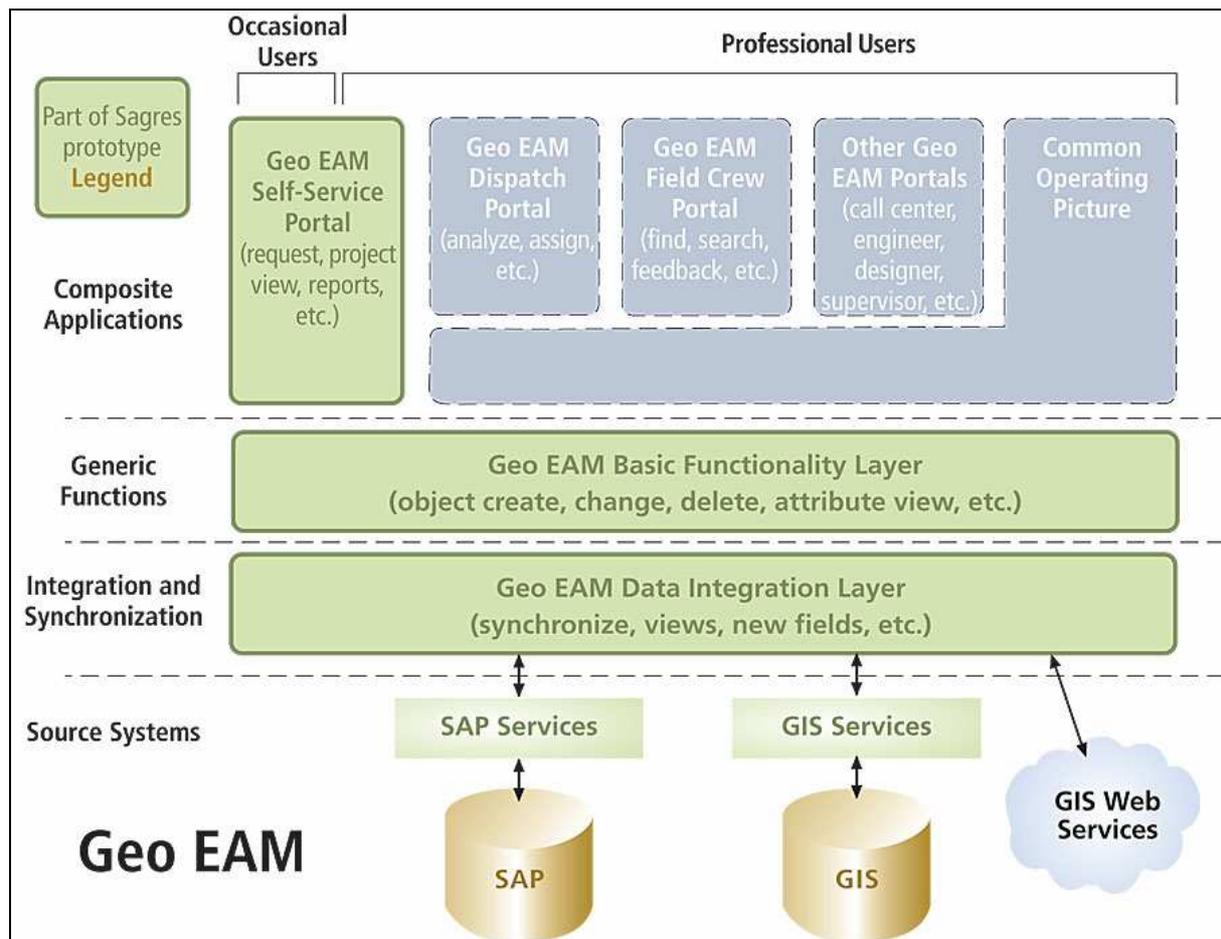


Abbildung: Architektur des Prototyps von Sagres (ESRI, 2007)

Hauptbestandteil von Geo EAM ist die SAP NetWeaver Plattform. ArcGIS Server ist als Geokomponente für die Anwendung unter SAP NetWeaver zertifiziert und lauffähig. SAP NetWeaver verwendet das Prinzip der Web Services, die zusätzlich auf so genannte Enterprise Services, als SAP-spezifischer Begriff, erweitert werden und im Baukastenprinzip zu geschäftsorientierten Anwendungen (Composite Applications) zusammengesetzt werden.

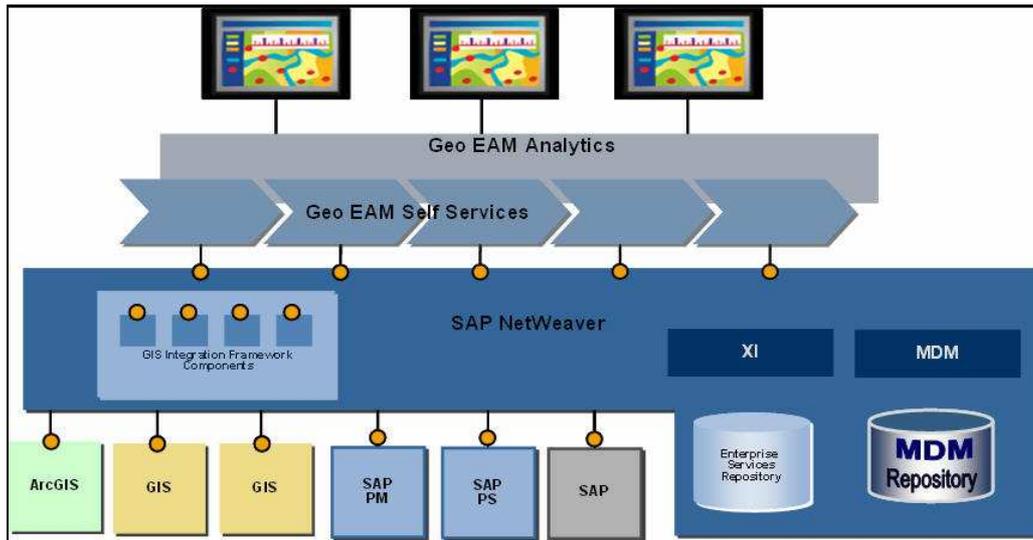


Abbildung: SAP NetWeaver Plattform mit GIS-Integration, (Francica, 2007)

Zum Zeitpunkt der Recherche gab es zum Prototyp noch keine Umsetzung beim Kunden für einen oder mehrere Geschäftsprozesse.

### GeoXtension

Die Fichtner Consulting und IT AG (FCIT) hat zur Integration von Geodaten in Prozesse und/oder Webanwendungen das Tool GeoXtension entwickelt. GeoXtension ermöglicht die Erstellung interaktiver Webapplikationen, fachlich nicht festgelegt, also auch für Störungsmanagement einsetzbar, und ist als eine offene dreischichtige J2EE-Anwendung konzipiert, die mit den Standards, wie sie in Abbildung aufgezeigt sind, arbeitet. Zur Beschreibung von Graphiken und graphischen Anwendungen kommt SVG zum Einsatz, XML/SOAP dienen zum Austausch von Daten und Diensten im Web, HTTP wird zur Datenübertragung im Web und J2EE zur Entwicklung der Unternehmensanwendungen verwendet. Die Geodaten sind nach der OGC Simple Features Spezifikation gespeichert und mit SQL kann auf die Datenbank zugegriffen werden. BPEL dient zur Zusammenstellung von Diensten von Transaktionsprozessen (FCIT, 2007a).



Abbildung: Verwendung von Standards bei GeoXtension (FCIT, 2007a)

Beim Pilotprojekt für die Stadtwerke Ulm ist die Beleuchtungsinstandhaltung mit GeoXtension realisiert worden. Der Datenfluss zur Anwendung ist der folgenden Abbildung dargestellt.

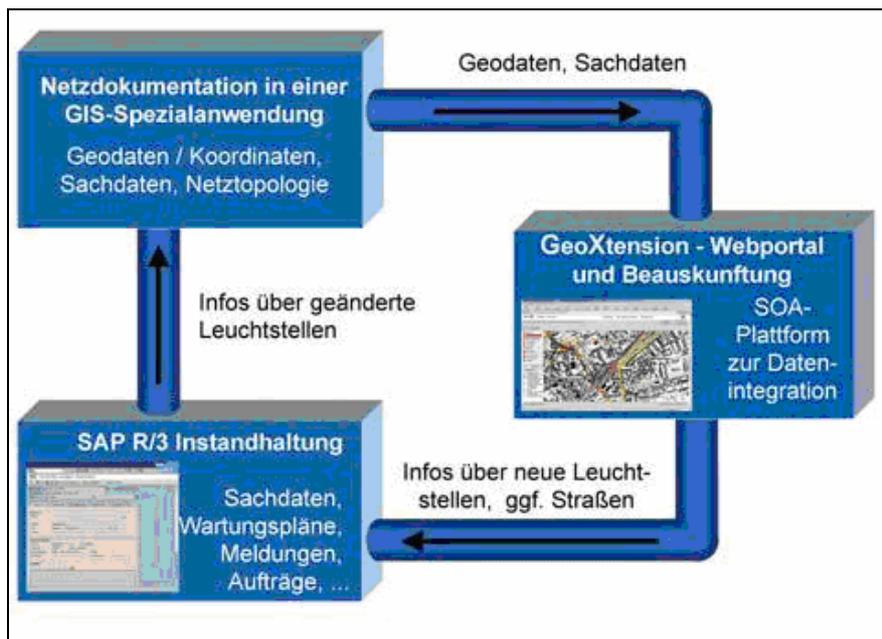


Abbildung: Datenfluss beim Pilotprojekt in Ulm (FCIT, 2007b)

## G!NIUS

G!NIUS stellt die vorkonfigurierte GIS-Standard-Lösung der Intergraph Corporation (G/Net) mit Erweiterung um prozessabbildende Funktionen des Produktes GRIPS von der Tochtergesellschaft Poppenhäger Grips GmbH dar. Alle Prozesse mit Geodatenbezug für die Bereiche Planung, Bau, Betrieb und Pflege der Betriebsmittel, Flächenverwaltung und Beauskunftung oder für Einsatzsteuerung mobiler Netztechniker, Störungsmanagement und Ressourcenverwaltung werden von G!NIUS als skalierbarer, modularer Lösungsansatz abgebildet. Dies gilt für sämtliche Medien, die sich in der in folgen Abbildung skizzierten Technologie der Lösungen wieder finden. Der Fokus liegt dabei auf den Bereichen Netzplanung, Netzbau und Netzbetrieb.

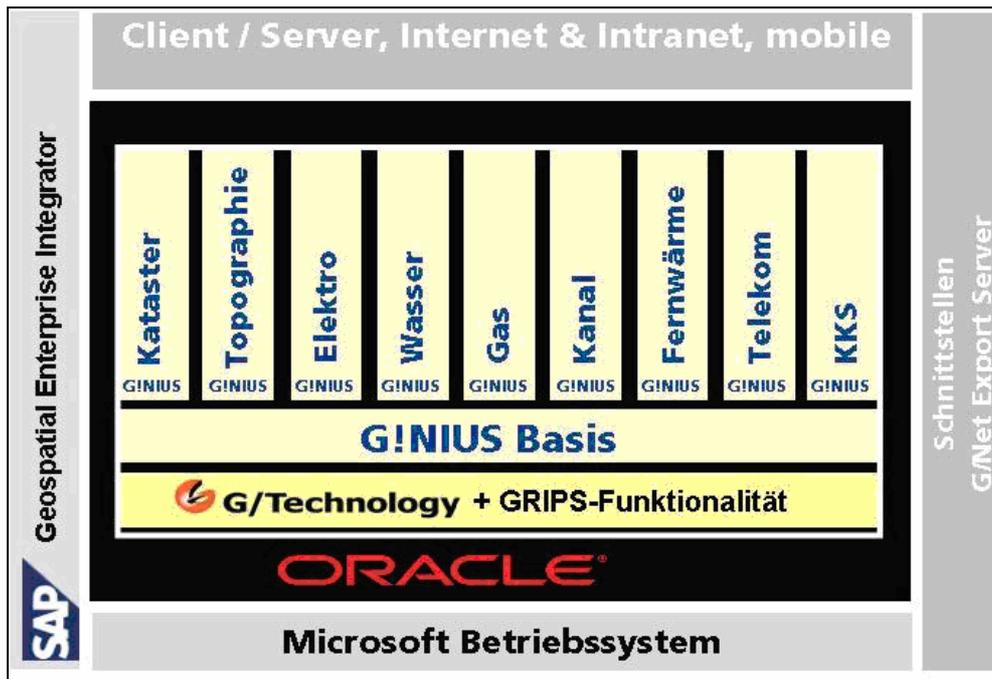


Abbildung: Technologie von G!NIUS (Intergraph, 2007a)

Basis der Lösung ist eine ORACLE-Datenbank. Standardschnittstellen zu SAP, SCADA und der Netzberechnung sind vorhanden, welche neben der Architektur von G!NIUS dargestellt werden. Das Störungsmanagement beinhaltet die folgenden Funktionalitäten: Störungserfassung, -zuweisung, Navigation, GPS, Wireless Kommunikation und Statistik (Intergraph, 2007b).

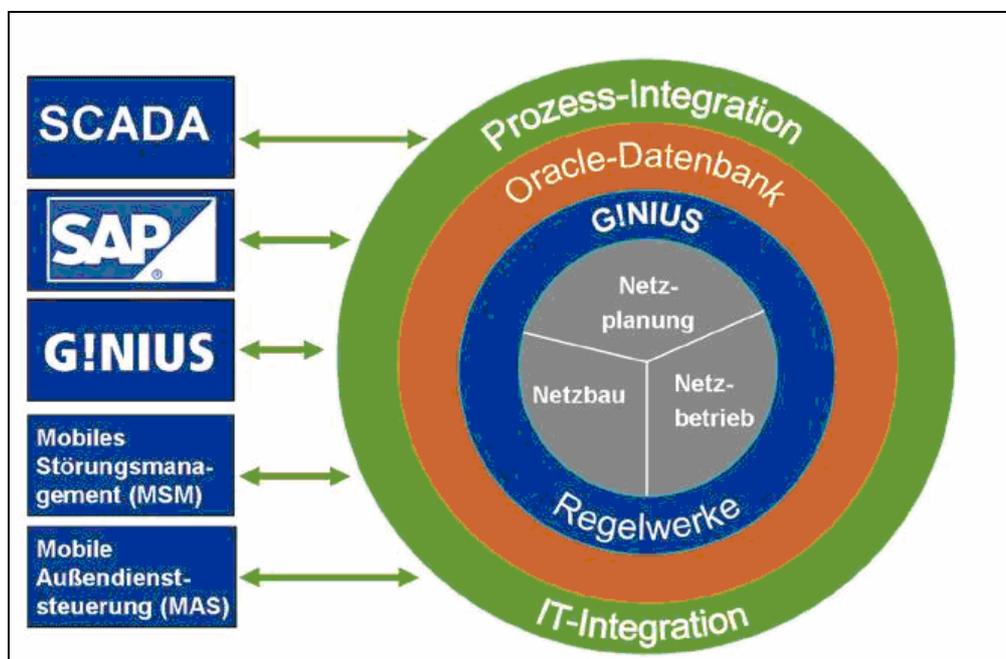


Abbildung: Architektur von G!NIUS (Intergraph, 2007b)

## IAM

IAM stellt das Integrated Asset-Management als Lösung der Firma Informationstechnik Consult GmbH, die als Tochter der Informationstechnik Service GmbH aufgestellt ist. Diese Produktfamilie bilden Asset Manager, Asset Monitor, Integration Manager und der Location Manager (ITS, 2007a). Der Asset Manager erlaubt Steuerung und Kontrolle der operativen Prozesse, der Asset Monitor dient zur Prognose, Optimierung und Monitoring der Strategien, der Integration Manager dient zur Kommunikation zwischen GIS und ERP und der Location Manager ist für den Ablauf der mobilen Prozesse zuständig. Die folgende Abbildung zeigt das Zusammenspiel der Bausteine mit GIS und ERP sowie zusätzlich mit einer mobilen Komponente. Ebenso wird in derselben Abbildung die fachliche Ausrichtung der Gesamtlösung für Planung/Projektierung, Bau, Entstörung, Wartung, Auskunft und Dokumentation herausgestellt.

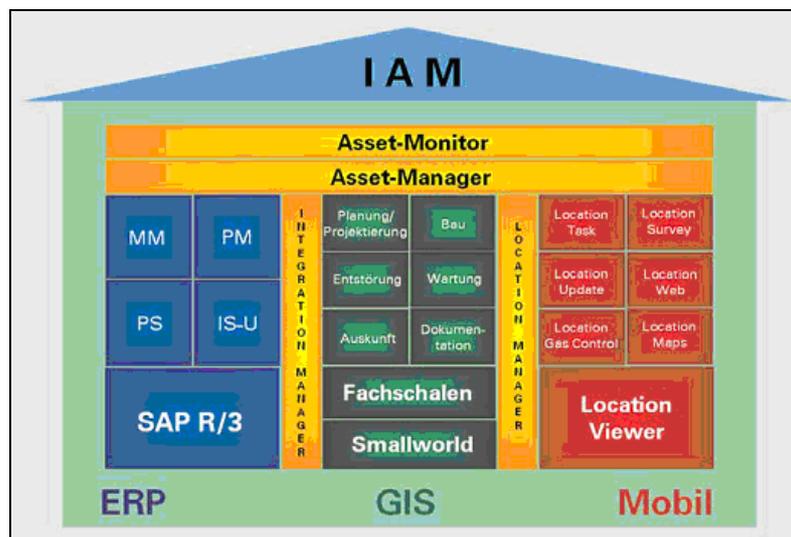


Abbildung: Integriertes Asset Management bei der ITS (ITS, 2007a)

Die Kommunikation zwischen den einzelnen Bestandteilen kann sowohl über einfache Punkt-zu-Punkt-Kopplungen als auch über Middleware-Lösungen, in deren Zusammenhang SAP NetWeaver XI und IBM WebSphere genannt werden, erfolgen. Für die Kopplung von SAP R/3 mit dem GIS Smallworld gibt es einen separaten Smallworld Business Integrator (SBI), auf den der Integration Manager aufgesetzt wird (ITS, 2007b).

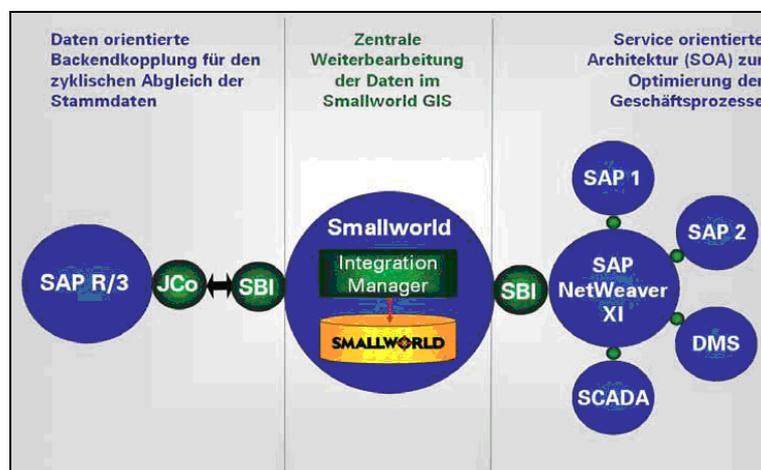


Abbildung: Integrationsarchitekturen bei IAM (ITS, 2007b)

## Integriertes Störmanagement

Die GDMcom GmbH, ein Unternehmen der VNG Verbundnetz Gas AG, stellt neben einem Digitalen Meldesystem Pipelineüberwachung und einem Dokumentenmanagement auch ein Tool für die Abwicklung von integriertem Störmanagement bereit (GDMcom, 2007). Die Software unterstützt dabei alle Phasen eines auftretenden Störfalles, von der Aufnahme der Meldung bis zur satellitengestützten Koordination und Navigation der Einsatzkräfte. Bei der Aufnahme der Störmeldung über eine Web-Anwendung wird gleichzeitig ein Reparaturauftrag im angebundenen ERP-System angelegt. Die Standorte der Fahrzeuge im Versorgungsgebiet werden automatisch per SMS abgefragt und im integrierten GIS-Fenster der Anwendung visualisiert. Der Einsatzleiter kann somit das nächstliegende Fahrzeug zur Schadensstelle als Erstsicherer schicken. Die Navigationssoftware des ausgewählten Einsatzfahrzeuges erhält die Koordinaten per SMS und leitet den Fahrer zur Störungsstelle. Dabei kann er jederzeit die nötigen Informationen und Pläne zur Störung über das mobile GIS abrufen (GDMcom, 2007).

Beim integrierten Störmanagement wird eine J2EE-konforme Web-Anwendung in einer 3-Stufen-Architektur abgebildet. Es kann ein beliebiger J2EE-Applikationsserver eingesetzt werden. Zur Datenhaltung wird eine serverfähige SQL-Datenbank benötigt. Die Kommunikation der J2EE-Anwendung mit Back-End-Systemen erfolgt über bordeigene Mittel des Applikationsservers und der Schnittstellen, welche durch die angebundenen Fremdsysteme bereitgestellt werden. Eine asynchrone Kommunikation mit den beteiligten Systemen wird über die JMS-Komponente des Applikationsserver ermöglicht (GDMcom, 2007).

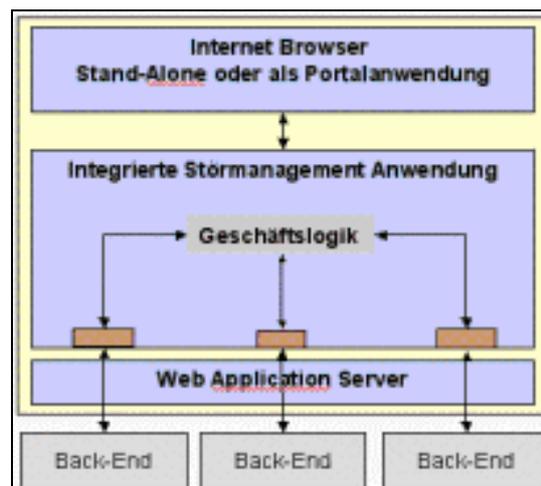


Abbildung: Architektur des integrierten Störmanagements (GDMcom, 2007)

## NOA

NOA stellt als Network Operations Application die Lösung für das Störungsmanagement der Firma Gesellschaft für Consulting und Innovative Software mbH (MGIS), die seit Oktober 2006 von dem Unternehmen Ubisense AG übernommen wurde. Das Softwarepaket fand seinen Ursprung in einem Projekt, welches 2004 von den Kommunalen Wasserwerken Leipzig initiiert wurde (Ubisense, 2007).

NOA ermöglicht die Integration von SAP (R/3 und Netweaver), Geoinformationssystemen (GIS), Netz- und Prozessleittechnik (NLS/PLS) mit weiteren Systemen wie z. B. Workforcemanagement, Flottenmanagement, Routenplaner, Dokumentenmanagement und Office. Bei der Datenhaltung kann der Kunde zwischen ORACLE, MS-SQL-Server und IBM DB2 wählen. NOA selbst basiert auf einer 3-Tier-Architektur, bei der laut Produktbeschreibung Microsoft .Net und ausschließlich Web Services zum Einsatz kommen. NOA kann laut Hersteller in einer EAI oder auch SOA zum Einsatz kommen.

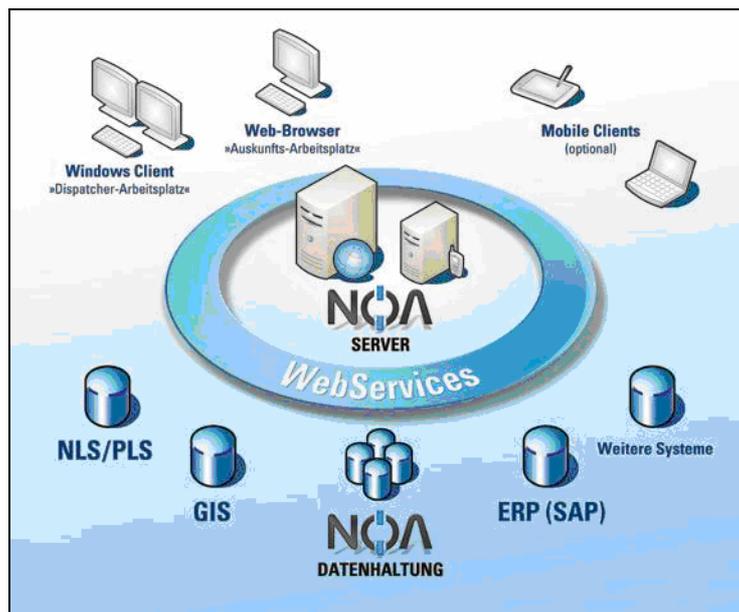


Abbildung: Technologie und Systemarchitektur von NOA (MGIS, 2007a)

Die zunächst für das Störungsmanagement konzipierte Lösung ist auf die Bereiche Planen und Bauen ausgeweitet worden und als Gesamtpaket erhältlich.

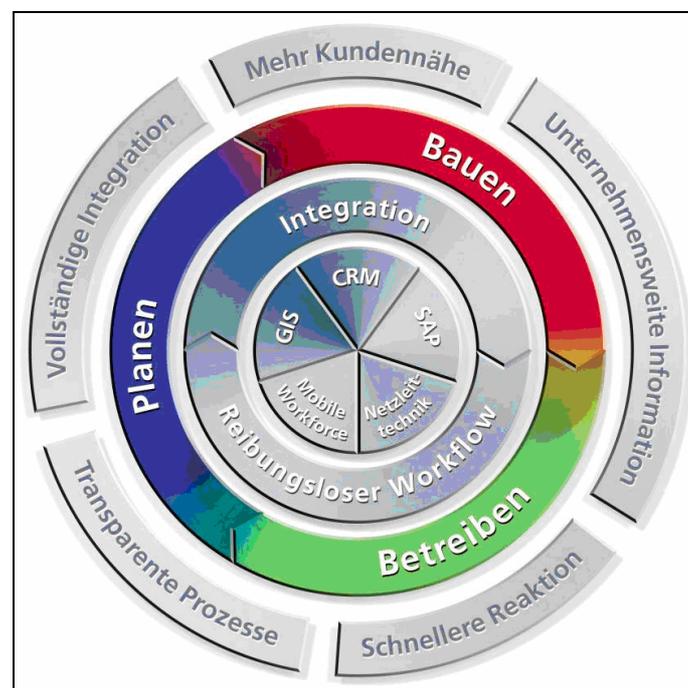


Abbildung: Die Lösung NOA (MGIS, 2007b)

## TBM

TBM stellt das Betriebsführungssystem „Technischer Betriebs-Manager“ der Firma EnergieSystemeNord GmbH (ESN) dar. Laut Produktbeschreibung können mit ihm die Medien Strom, Gas, Wasser, Fernwärme und Kanal verwaltet werden. Der TBM dient als Bindeglied zwischen dem Geographischen Informationssystem, dem Leitsystem und der kaufmännischen Welt.

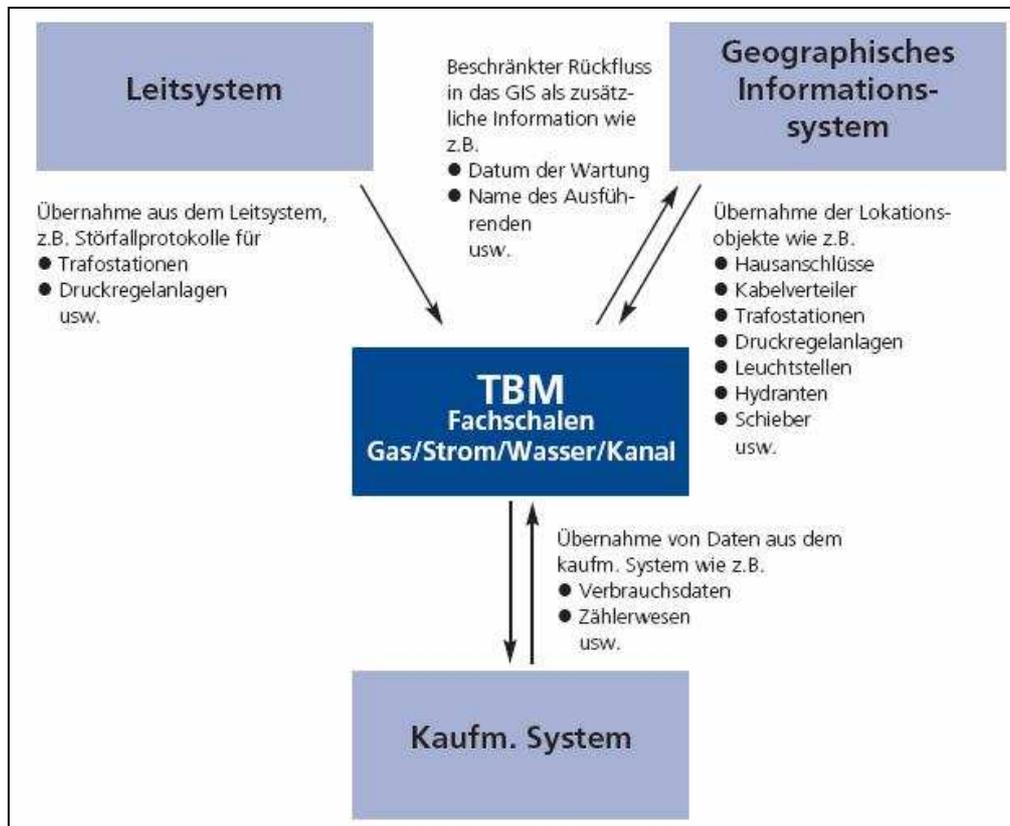


Abbildung: Systemintegration mit dem TBM (ESN, 2007a)

Bestandteile des TBM sind Betriebsführung, Aufgabenplanung, Ereignisverwaltung, Terminplanung, Objektverwaltung, Personalplanung, Betriebsmittel, PDA und Berichtswesen. Der Schwerpunkt liegt bei dem Themenbereich Wartung und Instandhaltung. Das Störungsmanagement wird zwar als Begriff erwähnt, in den Produktbeschreibungen jedoch nicht näher erläutert. Der Technische Betriebs-Manager basiert auf der Betriebsführungssoftware GS (Version 3.0) der Firma GreenGate AG, welche eine Client-Server-Lösung unter Windows ist. Favorisiert wird eine Firebird-SQL-Datenbank, die optional durch ORACLE oder MS-SQL-Server ausgetauscht werden kann. Des Weiteren wirbt der Anbieter mit den Schlagworten objektorientierter Aufbau, Mandantenfähigkeit, Mehrsprachigkeit sowie einer dynamisch erweiterbaren Datenstruktur. Sämtliche Kommunikation wird über COM/DCOM Schnittstellen realisiert (GreenGate, 2007b), welche zu den GIS-Systemen SICAD/SD, sis-Net, MapInfo, ArcGIS, MapObjects, Magellan, Tiffany, Smallworld, STRAKAT und Topobase bereits erstellt wurden. Laut Produktbeschreibung (GreenGate, 2007a) ist eine Schnittstelle zu dem ERP-System SAP R/3 in Vorbereitung. Die Schnittstellen zum ERP- und zum GIS-System werden über die von der ESN angebotenen Standardschnittstellen „ESN GIS Connect“ und „ESN TBM Interface“ umgesetzt.

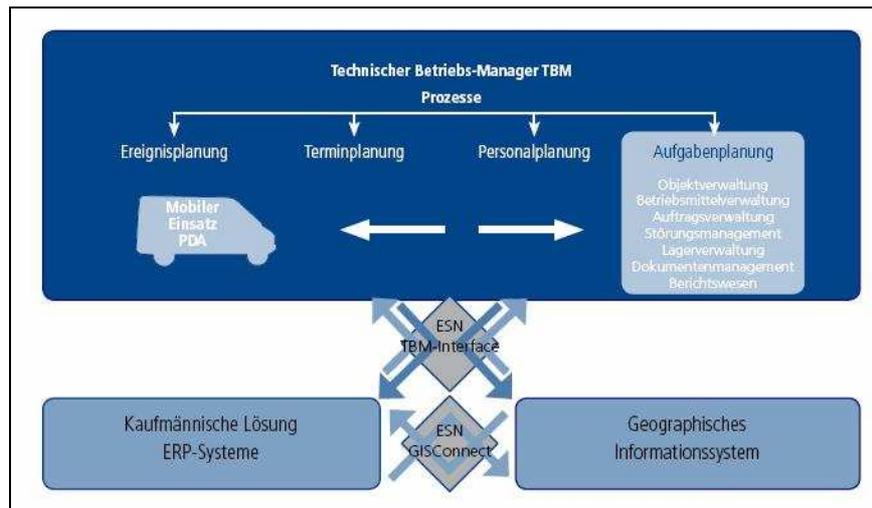


Abbildung: Schnittstellen beim TBM (ESN, 2007b)

Zwischen der ESN und FCIT gibt es seit November 2006 eine Partnerschaft bezüglich der Anwendung GeoX-tension (FCIT, 2007c). Ob diese Kooperation im Produktbereich Einfluss auf den TBM hat, ist noch nicht bekannt.

## TOMS

TOMS steht für „Technical Operational and Maintenance System“ und ist das Betriebstechnische Informationssystem (TIS) der Firma BERIT GmbH, welches die Lücke zwischen SAP, GIS, SCADA und der Netzberechnung schließt. Es bietet Module zu Netzplanung, Projektierung, Bau, Instandhaltung und Störungsmanagement an, welche gemeinsam und untereinander vernetzt oder auch separat genutzt werden können (BERIT, 2007). Die Systemintegration von TOMS mit anderen IT-Komponenten wird in der folgenden Abbildung dargestellt.

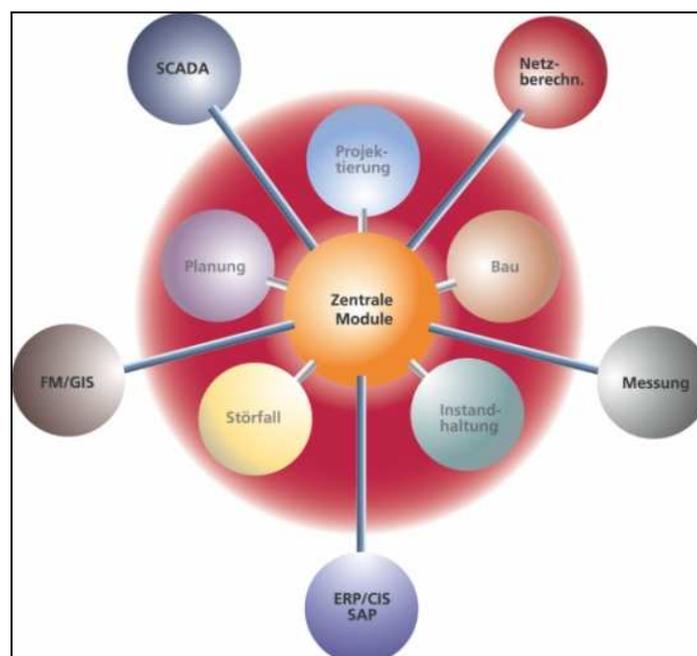


Abbildung: Systemintegration mit TOMS (BERIT, 2007)

Das Modul Störfallmanagement verfügt über eine Anbindung an das GIS, welches LIDS von BERIT (Bentley MicroStation mit einer Oracle-Datenbank) oder Smallworld von General Electric sein kann, und an das Netzleitsystem. Neben dem Handling und Beheben von Störungen und Netzausfällen können auch Abschaltungen mit dem Tool geplant und durchgeführt werden. Den Datenfluss im Störfallmanagement zeigt die folgende Abbildung.

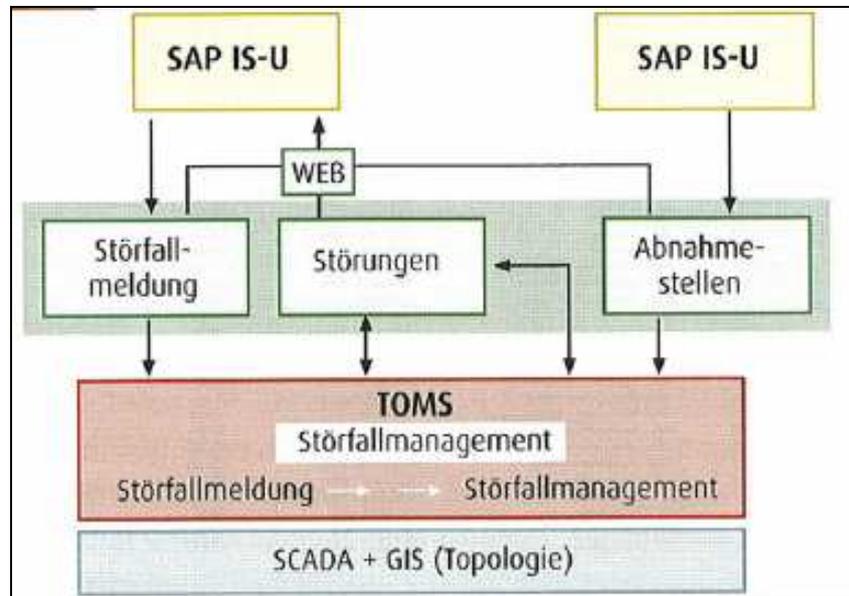


Abbildung: Datenfluss im Störungsmanagement bei TOMS (Steinmann, 2005)

TOMS wurde für einen tschechischen Regionalversorger im Strombereich entwickelt und zunächst auf die anderen Medien (Gas, Fernwärme, Wasser) und dann auf die deutschen Gegebenheiten angepasst. TOMS arbeitet in einer 3-Schicht-Architektur mit Java basierenden Diensten. Die Standards J2EE, XML und SOAP werden zur Kommunikation mit SAP verwendet. Seit der Version 3.2 steht eine Vielzahl an Funktionen mit Web Services bereit (Siemen, 2006).

## Anhang C: Code der Java-Klasse Adresse.java

```
package adressenservice;

// Import fuer Java
import java.util.*;
// Import fuer Datenbank
import java.sql.*;

/**
 * <p>Überschrift: Adressen-Service</p>
 * <p>Beschreibung: Fragt Straßennamen und Hausnummer mit Hausnummerzusatz aus der Daten-
bank ab.</p>
 * <p>Copyright : Copyright(c) 2007 </p>
 * @author Edda Steinmann
 * @version 1.0
 */
public class Adresse {

    // Variablen fuer die Datenbank
    private Connection conn;
    private Statement s;
    private ResultSet rs;
    /**
     * Fragt alle möglichen Strassennamen aus der Datenbank ab
     *
     * @return Strasse
     * @webmethod
     */
    public String getStrasse() {
        verbinden();
        Vector stras = new Vector();
        try {
            // Abfrage der Datenbank: Gebe alle möglichen Strassennamen zurueck
            String query = "SELECT DISTINCT strasse FROM hauskoordinaten";
            s = conn.createStatement();
            rs = s.executeQuery(query);
            // Abfrage speichern
            while (rs.next()) {
                stras.addElement(rs.getString("strasse"));
            } // Ende while
        } catch (SQLException sqle) {
            System.out.println("Die Datenbankabfrage nach der Strasse konnte nicht ausgefuehrt werden.");
            sqle.printStackTrace();
        } // Ende catch
        schliessen();
        String strasse = makeString1(stras);
        return strasse;
    }
    // Ende abfragen()

    /**
     * Fragt alle möglichen Hausnummern zur ausgewählten Strasse aus der Datenbank
     * @param strasse Name der ausgewaehlten Strasse
     * @return Nummern der Hausnummern, die in der Strasse vorkommen
     * @webmethod
     */
    public String getHausnummer(String strasse) {
        verbinden();
        Vector hausn = new Vector();
```

```

try {
    // Abfrage der Datenbank: Gebe alle möglichen Strassennamen zurueck
    String query = "SELECT DISTINCT hausnummer FROM hauskoordinaten WHERE strasse = " +
strasse + """;
    s = conn.createStatement();
    rs = s.executeQuery(query);
    // Abfrage speichern
    while (rs.next()) {
        hausn.addElement(rs.getString("hausnummer"));
    } // Ende while
} catch (SQLException sqle) {
    System.out.println("Die Datenbankabfrage nach der Hausnummer konnte nicht ausgefuehrt wer-
den.");
    sqle.printStackTrace();
} // Ende catch
schliessen();
String [] hausn1 = makeArray(hausn);
String [] hausnummer = hausnummerSortieren(hausn1);
String hausnu1 = makeString2(hausnummer);
return hausnu1;
} // Ende abfragen()

/**
 * Fragt alle möglichen Hausnummernzusätze zur ausgewählten Strasse und Hausnummer aus der
 * Datenbank
 * @param strasse Ausgewaehlte Strasse
 * @param hausnummer Ausgewaehlte Hausnummer
 * @return Moegliche Hausnummerzusätze in Abhaengigkeit von Strasse und
 * Hausnummer
 * @webmethod
 */
public String getHausnummerzusatz(String strasse, String hausnummer) {
    verbinden();
    Vector hausnzus = new Vector();
    try {
        // Abfrage der Datenbank: Gebe alle möglichen Hausnummerzusätze zurueck
        String query = "SELECT hausnummerzusatz FROM hauskoordinaten WHERE strasse = " + stras-
se + " AND hausnummer =" + hausnummer + " AND hausnummerzusatz is not Null";
        s = conn.createStatement();
        rs = s.executeQuery(query);
        // Abfrage speichern
        while (rs.next()) {
            hausnzus.addElement(rs.getString("hausnummerzusatz"));
        } // Ende while
    } catch (SQLException sqle) {
        System.out.println("Die Datenbankabfrage nach dem Hausnummerzusatz konnte nicht ausge-
fuehrt werden.");
        sqle.printStackTrace();
    } // Ende catch
    schliessen();
    String hausnummerzusatz = makeString1(hausnzus);
    return hausnummerzusatz;
} // Ende abfragen()

/**
 * Gibt den Rechtswert zur ausgewählten Strasse und Hausnummer und HSNZusatz aus der
 * Datenbank zurueck
 * @param strasse Ausgewaehlte Strasse
 * @param hausnummer Ausgewaehlte Hausnummer
 * @param hausnummerzusatz Ausgewählter Hausnummerzusatz

```

```

* @return Moegliche Hausnummerzusätze in Abhaengigkeit von Strasse und
* Hausnummer
* @webmethod
*/
public String getRechtswert(String strasse, String hausnummer, String hausnummerzusatz) {
    verbinden();
    Vector rechtsw = new Vector();
    try {
        // Abfrage der Datenbank: Gebe den Rechtswert zurueck
        String query = "SELECT rechtswert FROM hauskoordinaten WHERE strasse = " + strasse + "
AND hausnummer = " + hausnummer + " AND hausnummerzusatz = " + hausnummerzusatz + """;
        s = conn.createStatement();
        rs = s.executeQuery(query);
        // Abfrage speichern
        while (rs.next()) {
            rechtsw.addElement(rs.getString("rechtswert"));
        } // Ende while
    } catch (SQLException sqle) {
        System.out.println("Die Datenbankabfrage nach dem Hausnummerzusatz konnte nicht ausge-
fuehrt werden.");
        sqle.printStackTrace();
    } // Ende catch
    schliessen();
    String rechtswert = makeString1(rechtsw);
    return rechtswert;
} // Ende abfragen()

/**
* Gibt den Hochwert zur ausgewählten Strasse und Hausnummer und HSNZusatz aus der
* Datenbank zurueck
* @param strasse Ausgewaehlte Strasse
* @param hausnummer Ausgewaehlte Hausnummer
* @param hausnummerzusatz Ausgewählter Hausnummerzusatz
* @return Moegliche Hausnummerzusätze in Abhaengigkeit von Strasse und
* Hausnummer
* @webmethod
*/
public String getHochwert(String strasse, String hausnummer, String hausnummerzusatz) {
    verbinden();
    Vector hochw = new Vector();
    try {
        // Abfrage der Datenbank: Gebe den Rechtswert zurueck
        String query = "SELECT hochwert FROM hauskoordinaten WHERE strasse = " + strasse + "
AND hausnummer = " + hausnummer + " AND hausnummerzusatz = " + hausnummerzusatz + """;
        s = conn.createStatement();
        rs = s.executeQuery(query);
        // Abfrage speichern
        while (rs.next()) {
            hochw.addElement(rs.getString("hochwert"));
        } // Ende while
    } catch (SQLException sqle) {
        System.out.println("Die Datenbankabfrage nach dem Hausnummerzusatz konnte nicht ausge-
fuehrt werden.");
        sqle.printStackTrace();
    } // Ende catch
    schliessen();
    String hochwert = makeString1(hochw);
    return hochwert;
    //was passiert mit Feldern, die nicht gefüllt sind????
} // Ende abfragen()

/**

```

```

* Methode die den Inhalt eines Vectors in ein Array mit String schreibt
*
* @param vector Vector mit den Werten
* @return Array mit den Werten des uebergebenen Vectors
*/
private String[] makeArray(Vector vector) {
    Vector eingang = vector;
    int laenge = eingang.size();
    String [] ausgang = new String[laenge];
    for (int i=0; i<ausgang.length; i++) {
        ausgang[i] = eingang.elementAt(i).toString();
    } // Ende for
    return ausgang;
} // Ende makeArray()

/**
* Methode die den Inhalt eines Vectors in einen String schreibt
*
* @param vector Vector mit den Werten
* @return String mit den Werten des uebergebenen Vectors durch Leerzeichen getrennt
*/
private String makeString1(Vector vector) {
    Vector eingang = vector;
    int laenge = eingang.size();
    String ausgang = new String();
    for (int i=0; i<laenge; i++) {
        ausgang += eingang.elementAt(i).toString() + " ";
    } // Ende for
    return ausgang;
} // Ende makeArray()

/**
* Methode die den Inhalt eines Arrays in einen String schreibt
*
* @param string String mit den Werten
* @return String mit den Werten des uebergebenen Arrays durch Leerzeichen getrennt
*/
private String makeString2(String[] string) {
    String[] eingang = string;
    int laenge = eingang.length;
    String ausgang = new String();
    for (int i=0; i<laenge; i++) {
        ausgang += eingang[i] + " ";
    } // Ende for
    return ausgang;
} // Ende makeArray()

/**
* Stellt die Verbindung mit der Datenbank her
*/
private void verbinden() {

    try {
        Class.forName("oracle.jdbc.OracleDriver");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        System.exit(1); // TODO
    }
}
// Anmeldung an die Datenbank
try {

```

```
    conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "kataster", "system");
  } // Ende try
  catch (SQLException sqle) {
    System.out.println("Es konnte keine Datenbankverbindung hergestellt werden.");
    System.out.println(sqle);
  } // Ende catch
} // Ende verbinden()

/**
 * Schliesst die Verbindung mit der Datenbank
 */
private void schliessen() {
  try {
    rs.close();
    s.close();
    conn.close();
  } // Ende try
  catch (SQLException sqle) {
    System.out.println("Die Datenbankverbindung konnte nicht geschlossen werden.");
    System.out.println(sqle);
  } // Ende catch
} // Ende schliessen()

/**
 * Sortiert die zurueckgegebenen Werte der Hausnummer wie Integer aufsteigend
 * @param werte Unsortierte Hausnummer
 * @return Sortierte Hausnummern
 */
private String[] hausnummerSortieren(String [] werte) {
  int [] wertelnt = new int [werte.length];
  for(int i=0; i<werte.length; i++) {
    int zahl = Integer.parseInt(werte[i]);
    wertelnt[i] = zahl;
  }
  Arrays.sort(wertelnt);
  for(int j=0; j<werte.length; j++) {
    werte[j] = String.valueOf(wertelnt[j]);
  }
  return werte;
} // Ende sortieren

/**
 * Gibt den Hausnummer als Integer als String zurueck
 *
 * @param hausnummer
 * @return Hausnummerzusatz
 */
private int getHausnummerzusatz(String hausnummer) {
  StringTokenizer st = new StringTokenizer(hausnummer, "/");
  int hausnummerzusatz = Integer.valueOf(st.nextToken()).intValue();
  return hausnummerzusatz;
} // Ende getHausnummerzusatz()

public Adresse() {
}
}
```

## Anhang D: Code der Java-Klasse WMS.java

```

package wmservice;
import java.sql.SQLException;
import java.util.Vector;
/**
 * <p>Überschrift: WMS Service</p>
 * <p>Beschreibung: Diese Klasse nutzt die Hauskoordinate und gibt die WMS-URL aus.</p>
 * <p>Copyright : Copyright(c) 2007 </p>
 * @author Edda Steinmann
 */
public class WMS {
// Instanzvariablen
private double minx;
private double miny;
private double maxx;
private double maxy;
private int breite = 200;
private int hoehe = 200;
private String mvURL;
private String layerMV = "Flurstuecke,Gebaeude,Leistungsabschnittv,Leistungsabschnittal";
/**
 * Setzt die BoundingBox fuer das entsprechende Flurstueck in GK-Koordinaten
 * @param
 * Koordinaten vom Gesamtgebiet: &BBOX=3464666,5516491,3483169,5535899&
 * @webmethod
 */
public double getBBoxminx(double mittex) {
minx = mittex - 100;
return minx;
}
public double setBBoxminy (double mittey) {
miny = mittey - 100;
return miny;
}
public double setBBoxmaxx (double mittex) {
maxx = mittex + 100;
return maxx;
}
public double setBBoxmaxy(double mittey) {
maxy = mittey + 100;
return maxy;
}
// Ende setBoundingBox()
public String getMVURL(double minx, double miny, double maxx, double maxy) {
mvURL =
"http://localhost:8888/mapviewer/wms?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1" +
"&width=" + breite + "&height=" + hoehe + "&Bbox=" +
minx + "," + miny + "," + maxx + "," + maxy + "&Layers=" + layerMV +
"&SRS=EPSG:31467&format=image/png8&TRANSPARENT=TRUE";
//System.out.println(mvURL);
return mvURL;
}
// Ende getMVURL()
public WMS() {
}
}

```

## Anhang E: Code der WSDL-Datei des Adressen-Service

```
<definitions
  name="adressenserice"
  targetNamespace="http://adressenservice/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://adressenservice/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns0="http://adressenservice/types/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
>
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespa-
ce="http://adressenservice/types/"
      elementFormDefault="qualified" xmlns:tns="http://adressenservice/types/"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:soap11-
enc="http://schemas.xmlsoap.org/soap/encoding/">
      <element name="getHausnummerElement">
        <complexType>
          <sequence>
            <element name="strasse" type="string" nillable="true"/>
          </sequence>
        </complexType>
      </element>
      <element name="getHausnummerResponseElement">
        <complexType>
          <sequence>
            <element name="result" type="string" nillable="true"/>
          </sequence>
        </complexType>
      </element>
      <element name="getHausnummerzusatzElement">
        <complexType>
          <sequence>
            <element name="strasse" type="string" nillable="true"/>
            <element name="hausnummer" type="string" nillable="true"/>
          </sequence>
        </complexType>
      </element>
      <element name="getHausnummerzusatzResponseElement">
        <complexType>
          <sequence>
            <element name="result" type="string" nillable="true"/>
          </sequence>
        </complexType>
      </element>
      <element name="getHochwertElement">
        <complexType>
          <sequence>
            <element name="strasse" type="string" nillable="true"/>
            <element name="hausnummer" type="string" nillable="true"/>
            <element name="hausnummerzusatz" type="string" nillable="true"/>
          </sequence>
        </complexType>
      </element>
      <element name="getHochwertResponseElement">
        <complexType>
```

```

        <sequence>
            <element name="result" type="string" nillable="true"/>
        </sequence>
    </complexType>
</element>
<element name="getRechtswertElement">
    <complexType>
        <sequence>
            <element name="strasse" type="string" nillable="true"/>
            <element name="hausnummer" type="string" nillable="true"/>
            <element name="hausnummerzusatz" type="string" nillable="true"/>
        </sequence>
    </complexType>
</element>
<element name="getRechtswertResponseElement">
    <complexType>
        <sequence>
            <element name="result" type="string" nillable="true"/>
        </sequence>
    </complexType>
</element>
<element name="getStrasseElement">
    <complexType>
        <sequence/>
    </complexType>
</element>
<element name="getStrasseResponseElement">
    <complexType>
        <sequence>
            <element name="result" type="string" nillable="true"/>
        </sequence>
    </complexType>
</element>
</schema>
</types>
<message name="Adressenservice_getHausnummer">
    <part name="parameters" element="tns0:getHausnummerElement"/>
</message>
<message name="Adressenservice_getHausnummerResponse">
    <part name="parameters" element="tns0:getHausnummerResponseElement"/>
</message>
<message name="Adressenservice_getHausnummerzusatz">
    <part name="parameters" element="tns0:getHausnummerzusatzElement"/>
</message>
<message name="Adressenservice_getHausnummerzusatzResponse">
    <part name="parameters" element="tns0:getHausnummerzusatzResponseElement"/>
</message>
<message name="Adressenservice_getHochwert">
    <part name="parameters" element="tns0:getHochwertElement"/>
</message>
<message name="Adressenservice_getHochwertResponse">
    <part name="parameters" element="tns0:getHochwertResponseElement"/>
</message>
<message name="Adressenservice_getRechtswert">
    <part name="parameters" element="tns0:getRechtswertElement"/>
</message>
<message name="Adressenservice_getRechtswertResponse">
    <part name="parameters" element="tns0:getRechtswertResponseElement"/>
</message>
<message name="Adressenservice_getStrasse">
    <part name="parameters" element="tns0:getStrasseElement"/>
</message>

```

```
<message name="Adressenservice_getStrasseResponse">
  <part name="parameters" element="tns0:getStrasseResponseElement"/>
</message>
<portType name="adressenservice">
  <operation name="getHausnummer">
    <input message="tns:Adressenservice_getHausnummer"/>
    <output message="tns:Adressenservice_getHausnummerResponse"/>
  </operation>
  <operation name="getHausnummerzusatz">
    <input message="tns:Adressenservice_getHausnummerzusatz"/>
    <output message="tns:Adressenservice_getHausnummerzusatzResponse"/>
  </operation>
  <operation name="getHochwert">
    <input message="tns:Adressenservice_getHochwert"/>
    <output message="tns:Adressenservice_getHochwertResponse"/>
  </operation>
  <operation name="getRechtswert">
    <input message="tns:Adressenservice_getRechtswert"/>
    <output message="tns:Adressenservice_getRechtswertResponse"/>
  </operation>
  <operation name="getStrasse">
    <input message="tns:Adressenservice_getStrasse"/>
    <output message="tns:Adressenservice_getStrasseResponse"/>
  </operation>
</portType>
<binding name="adressenserviceSoapHttp" type="tns:adressenservice">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getHausnummer">
    <soap:operation soapAction="http://adressenservice//getHausnummer"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="getHausnummerzusatz">
    <soap:operation soapAction="http://adressenservice//getHausnummerzusatz"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="getHochwert">
    <soap:operation soapAction="http://adressenservice//getHochwert"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="getRechtswert">
    <soap:operation soapAction="http://adressenservice//getRechtswert"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
</wsdl:binding>
</wsdl:service>
```

```
</operation>
<operation name="getStrasse">
  <soap:operation soapAction="http://adressenservice//getStrasse"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
</binding>
<service name="adressenservice">
  <port name="adressenserviceSoapHttpPort" binding="tns:adressenserviceSoapHttp">
    <soap:address location="http://127.0.0.1:8888/Stoerung1-Adressen-Service-context-
root/adressenserviceSoapHttpPort"/>
  </port>
</service>
</definitions>
```

## Anhang F: Code der WSDL-Datei des WMS-Service

```
<definitions
  name="WMS-Service"
  targetNamespace="http://wmsservice/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://wmsservice/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns0="http://wmsservice/types/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
>
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://wmsservice/types/"
      elementFormDefault="qualified" xmlns:tns="http://wmsservice/types/"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:soap11-enc="http://schemas.xmlsoap.org/soap/encoding/">
      <element name="getBBoxminxElement">
        <complexType>
          <sequence>
            <element name="mittex" type="double"/>
          </sequence>
        </complexType>
      </element>
      <element name="getBBoxminxResponseElement">
        <complexType>
          <sequence>
            <element name="koordminx" type="double"/>
          </sequence>
        </complexType>
      </element>
      <element name="getMVURLElement">
        <complexType>
          <sequence>
            <element name="minx" type="double"/>
            <element name="miny" type="double"/>
            <element name="maxx" type="double"/>
            <element name="maxy" type="double"/>
          </sequence>
        </complexType>
      </element>
      <element name="getMVURLResponseElement">
        <complexType>
          <sequence>
            <element name="result" type="string" nillable="true"/>
          </sequence>
        </complexType>
      </element>
      <element name="setBBoxmaxxElement">
        <complexType>
          <sequence>
            <element name="mittex" type="double"/>
          </sequence>
        </complexType>
      </element>
      <element name="setBBoxmaxxResponseElement">
```

```

    <complexType>
      <sequence>
        <element name="koordmaxx" type="double"/>
      </sequence>
    </complexType>
  </element>
  <element name="setBBoxmaxyElement">
    <complexType>
      <sequence>
        <element name="mitty" type="double"/>
      </sequence>
    </complexType>
  </element>
  <element name="setBBoxmaxyResponseElement">
    <complexType>
      <sequence>
        <element name="koordmaxy" type="double"/>
      </sequence>
    </complexType>
  </element>
  <element name="setBBoxminyElement">
    <complexType>
      <sequence>
        <element name="mittey" type="double"/>
      </sequence>
    </complexType>
  </element>
  <element name="setBBoxminyResponseElement">
    <complexType>
      <sequence>
        <element name="koordminy" type="double"/>
      </sequence>
    </complexType>
  </element>
</schema>
</types>
<message name="WMSService_getBBoxminx">
  <part name="parameters" element="tns0:getBBoxminxElement"/>
</message>
<message name="WMSService_getBBoxminxResponse">
  <part name="parameters" element="tns0:getBBoxminxResponseElement"/>
</message>
<message name="WMSService_getMVURL">
  <part name="parameters" element="tns0:getMVURLElement"/>
</message>
<message name="WMSService_getMVURLResponse">
  <part name="parameters" element="tns0:getMVURLResponseElement"/>
</message>
<message name="WMSService_setBBoxmaxx">
  <part name="parameters" element="tns0:setBBoxmaxxElement"/>
</message>
<message name="WMSService_setBBoxmaxxResponse">
  <part name="parameters" element="tns0:setBBoxmaxxResponseElement"/>
</message>
<message name="WMSService_setBBoxmaxy">
  <part name="parameters" element="tns0:setBBoxmaxyElement"/>
</message>
<message name="WMSService_setBBoxmaxyResponse">
  <part name="parameters" element="tns0:setBBoxmaxyResponseElement"/>
</message>
<message name="WMSService_setBBoxminy">
  <part name="parameters" element="tns0:setBBoxminyElement"/>

```

```
</message>
<message name="WMSService_setBBoxminyResponse">
  <part name="parameters" element="tns0:setBBoxminyResponseElement"/>
</message>
<portType name="WMS-Service">
  <operation name="getBBoxminx">
    <input message="tns:WMSService_getBBoxminx"/>
    <output message="tns:WMSService_getBBoxminxResponse"/>
  </operation>
  <operation name="getMVURL">
    <input message="tns:WMSService_getMVURL"/>
    <output message="tns:WMSService_getMVURLResponse"/>
  </operation>
  <operation name="setBBoxmaxx">
    <input message="tns:WMSService_setBBoxmaxx"/>
    <output message="tns:WMSService_setBBoxmaxxResponse"/>
  </operation>
  <operation name="setBBoxmaxy">
    <input message="tns:WMSService_setBBoxmaxy"/>
    <output message="tns:WMSService_setBBoxmaxyResponse"/>
  </operation>
  <operation name="setBBoxminy">
    <input message="tns:WMSService_setBBoxminy"/>
    <output message="tns:WMSService_setBBoxminyResponse"/>
  </operation>
</portType>
<binding name="WMS-ServiceSoapHttp" type="tns:WMS-Service">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getBBoxminx">
    <soap:operation soapAction="http://wmsservice//getBBoxminx"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="getMVURL">
    <soap:operation soapAction="http://wmsservice//getMVURL"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="setBBoxmaxx">
    <soap:operation soapAction="http://wmsservice//setBBoxmaxx"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="setBBoxmaxy">
    <soap:operation soapAction="http://wmsservice//setBBoxmaxy"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="setBBoxminy">
    <soap:operation soapAction="http://wmsservice//setBBoxminy"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
</wsdl:binding>
</wsdl:portType>
</wsdl:service>
</wsdl:definitions>
```

```
</output>
</operation>
<operation name="setBBoxminy">
  <soap:operation soapAction="http://wmsservice//setBBoxminy"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
</binding>
<service name="WMS-Service">
  <port name="WMS-ServiceSoapHttpPort" binding="tns:WMS-ServiceSoapHttp">
    <soap:address location="http://127.0.0.1:8888/Stoerung1-WMS-Service-context-
root/WMSServiceSoapHttpPort"/>
  </port>
</service>
</definitions>
```

## Anhang G: Code der WSDL-Datei des BPEL-Prozesses

Code der WSDL-Datei des BPEL-Prozesses (hier ProzessAdresse genannt)

```

<definitions
  name="ProzessAdresse"
  targetNamespace="http://xmlns.oracle.com/ProzessAdresse"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
  xmlns:adr="http://xmlns.oracle.com/Stoerung1/Adressen-Service/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:client="http://xmlns.oracle.com/ProzessAdresse"
  xmlns:wms="http://xmlns.Oracle.com/Stoerung1/WMS-Service/"
>
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://xmlns.oracle.com/ProzessAdresse" schemaLocati-
on="ProzessAdresse.xsd"/>
    </schema>
  </types>
  <message name="ProzessAdresseRequestMessage">
    <part name="payload" element="client:adresse"/>
  </message>
  <!--<message name="WMSService_setBoundingBox">
    <part name="payload2" element="wms:koordinaten"/>
  </message-->
  <message name="ProzessAdresseResponseMessage">
    <part name="payload1" element="client:WMS_URL"/>
  </message>
  <portType name="ProzessAdresse">
    <operation name="process">
      <input message="client:ProzessAdresseRequestMessage"/>
      <output message="client:ProzessAdresseResponseMessage"/>
    </operation>
  </portType>
  <plnk:partnerLinkType name="ProzessAdresse">
    <plnk:role name="ProzessAdresseProvider">
      <plnk:portType name="client:ProzessAdresse"/>
    </plnk:role>
  </plnk:partnerLinkType>
</definitions>

```

## Anhang H: Code der Schemadatei XSD zum BPEL-Prozess

```
<schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="http://xmlns.oracle.com/ProzessAdresse"
  xmlns="http://www.w3.org/2001/XMLSchema">

  <element name="adresse">
    <complexType>
      <sequence>
        <element name="strasse" type="string"/>
        <element name="hausnummer" type="string"/>
        <element name="hausnummerzusatz" type="string"/>
      </sequence>
    </complexType>
  </element>
  <element name="WMS_URL">
    <complexType>
      <sequence>
        <element name="URL" type="string"/>
      </sequence>
    </complexType>
  </element>
</schema>
```



```

<variables>
  <!-- Reference to the message passed as input during initiation -->
  <variable name="inputVariable"
    messageType="client:ProzessAdresseRequestMessage"/>
  <!-- Reference to the message that will be returned to the requester-->
  <variable name="outputVariable"
    messageType="client:ProzessAdresseResponseMessage"/>
  <variable name="Invoke_Rechtswert_getRechtswert_InputVariable"
    messageType="ns2:Adressenservice_getRechtswert"/>
  <variable name="Invoke_Rechtswert_getRechtswert_OutputVariable"
    messageType="ns2:Adressenservice_getRechtswertResponse"/>
  <variable name="Invoke_Hochwert_getHochwert_InputVariable"
    messageType="ns2:Adressenservice_getHochwert"/>
  <variable name="Invoke_Hochwert_getHochwert_OutputVariable"
    messageType="ns2:Adressenservice_getHochwertResponse"/>
  <variable name="Invoke_URL_getMVURL_InputVariable"
    messageType="ns1:WMSservice_getMVURL"/>
  <variable name="Invoke_URL_getMVURL_OutputVariable"
    messageType="ns1:WMSservice_getMVURLResponse"/>
  <variable name="Invoke_minx_getBBoxminx_InputVariable"
    messageType="ns1:WMSservice_getBBoxminx"/>
  <variable name="Invoke_minx_getBBoxminx_OutputVariable"
    messageType="ns1:WMSservice_getBBoxminxResponse"/>
  <variable name="Invoke_maxx_setBBoxmaxx_InputVariable"
    messageType="ns1:WMSservice_setBBoxmaxx"/>
  <variable name="Invoke_maxx_setBBoxmaxx_OutputVariable"
    messageType="ns1:WMSservice_setBBoxmaxxResponse"/>
  <variable name="Invoke_minx_setBBoxminx_InputVariable"
    messageType="ns1:WMSservice_setBBoxminx"/>
  <variable name="Invoke_minx_setBBoxminx_OutputVariable"
    messageType="ns1:WMSservice_setBBoxminxResponse"/>
  <variable name="Invoke_maxx_setBBoxmaxx_InputVariable"
    messageType="ns1:WMSservice_setBBoxmaxx"/>
  <variable name="Invoke_maxx_setBBoxmaxx_OutputVariable"
    messageType="ns1:WMSservice_setBBoxmaxxResponse"/>
</variables>
<!--
  //////////////////////////////////////
  ORCHESTRATION LOGIC
  Set of activities coordinating the flow of messages across the
  services integrated within this business process
  //////////////////////////////////////
-->
<sequence name="main">
  <!-- Receive input from requestor. (Note: This maps to operation defined in ProzessAdresse.wsdl) --
  >
  <receive name="receiveInput" partnerLink="client"
    portType="client:ProzessAdresse" operation="process"
    variable="inputVariable" createInstance="yes"/>
  <!-- Generate reply to synchronous request -->
  <flow name="Flow_1">
    <sequence name="Sequence_1">
      <assign name="Assign_Adresse_H">
        <copy>
          <from variable="inputVariable" part="payload"
            query="/client:adresse/client:strasse"/>
          <to variable="Invoke_Hochwert_getHochwert_InputVariable"
            part="parameters" query="/ns3:getHochwertElement/ns3:strasse"/>
        </copy>
        <copy>
          <from variable="inputVariable" part="payload"
            query="/client:adresse/client:hausnummer"/>

```

```

    <to variable="Invoke_Hochwert_getHochwert_InputVariable"
        part="parameters"
        query="/ns3:getHochwertElement/ns3:hausnummer"/>
    </copy>
    <copy>
    <from variable="inputVariable" part="payload"
        query="/client:adresse/client:hausnummerzusatz"/>
    <to variable="Invoke_Hochwert_getHochwert_InputVariable"
        part="parameters"
        query="/ns3:getHochwertElement/ns3:hausnummerzusatz"/>
    </copy>
</assign>
<invoke name="Invoke_Hochwert" partnerLink="adressenserice"
    portType="ns2:adressenserice" operation="getHochwert"
    inputVariable="Invoke_Hochwert_getHochwert_InputVariable"
    outputVariable="Invoke_Hochwert_getHochwert_OutputVariable"/>
</sequence>
<sequence name="Sequence_1">
    <assign name="Assign_Adresse_R">
    <copy>
    <from variable="inputVariable" part="payload"
        query="/client:adresse/client:strasse"/>
    <to variable="Invoke_Rechtswert_getRechtswert_InputVariable"
        part="parameters"
        query="/ns3:getRechtswertElement/ns3:strasse"/>
    </copy>
    <copy>
    <from variable="inputVariable" part="payload"
        query="/client:adresse/client:hausnummer"/>
    <to variable="Invoke_Rechtswert_getRechtswert_InputVariable"
        part="parameters"
        query="/ns3:getRechtswertElement/ns3:hausnummer"/>
    </copy>
    <copy>
    <from variable="inputVariable" part="payload"
        query="/client:adresse/client:hausnummerzusatz"/>
    <to variable="Invoke_Rechtswert_getRechtswert_InputVariable"
        part="parameters"
        query="/ns3:getRechtswertElement/ns3:hausnummerzusatz"/>
    </copy>
    </assign>
    <invoke name="Invoke_Rechtswert" partnerLink="adressenserice"
        portType="ns2:adressenserice" operation="getRechtswert"
        inputVariable="Invoke_Rechtswert_getRechtswert_InputVariable"
        outputVariable="Invoke_Rechtswert_getRechtswert_OutputVariable"/>
    </sequence>
</flow>
<flow name="Flow_2">
    <sequence name="Sequence_4">
    <assign name="Assign_mittey_an_maxy">
    <copy>
    <from expres-
sion="number(bpws:getVariableData('Invoke_Hochwert_getHochwert_OutputVariable','parameters',/n
s3:getHochwertResponseElement/ns3:result'))"/>
    <to variable="Invoke_maxy_setBBoxmaxy_InputVariable"
        part="parameters" query="/ns4:setBBoxmaxyElement/ns4:mitty"/>
    </copy>
    </assign>
    <invoke name="Invoke_maxy" partnerLink="WMS-Service"
        portType="ns1:WMS-Service" operation="setBBoxmaxy"
        inputVariable="Invoke_maxy_setBBoxmaxy_InputVariable"
        outputVariable="Invoke_maxy_setBBoxmaxy_OutputVariable"/>

```

```

</sequence>
<sequence name="Sequence_3">
  <assign name="Assign_mittey_an_miny">
    <copy>
      <from expression="number(bpws:getVariableData('Invoke_Hochwert_getHochwert_OutputVariable','parameters',/ns3:getHochwertResponseElement/ns3:result'))"/>
      <to variable="Invoke_miny_setBBoxminy_InputVariable"
        part="parameters" query="/ns4:setBBoxminyElement/ns4:mittey"/>
    </copy>
  </assign>
  <invoke name="Invoke_miny" partnerLink="WMS-Service"
    portType="ns1:WMS-Service" operation="setBBoxminy"
    inputVariable="Invoke_miny_setBBoxminy_InputVariable"
    outputVariable="Invoke_miny_setBBoxminy_OutputVariable"/>
</sequence>
<sequence name="Sequence_2">
  <assign name="Assign_mittex_an_maxx">
    <copy>
      <from expression="number(bpws:getVariableData('Invoke_Rechtswert_getRechtswert_OutputVariable','parameters',/ns3:getRechtswertResponseElement/ns3:result'))"/>
      <to variable="Invoke_maxx_setBBoxmaxx_InputVariable"
        part="parameters" query="/ns4:setBBoxmaxxElement/ns4:mittex"/>
    </copy>
  </assign>
  <invoke name="Invoke_maxx" partnerLink="WMS-Service"
    portType="ns1:WMS-Service" operation="setBBoxmaxx"
    inputVariable="Invoke_maxx_setBBoxmaxx_InputVariable"
    outputVariable="Invoke_maxx_setBBoxmaxx_OutputVariable"/>
</sequence>
<sequence name="Sequence_2">
  <assign name="Assign_mittex_an_minx">
    <copy>
      <from expression="number(bpws:getVariableData('Invoke_Rechtswert_getRechtswert_OutputVariable','parameters',/ns3:getRechtswertResponseElement/ns3:result'))"/>
      <to variable="Invoke_minx_getBBoxminx_InputVariable"
        part="parameters" query="/ns4:getBBoxminxElement/ns4:mittex"/>
    </copy>
  </assign>
  <invoke name="Invoke_minx" partnerLink="WMS-Service"
    portType="ns1:WMS-Service" operation="getBBoxminx"
    inputVariable="Invoke_minx_getBBoxminx_InputVariable"
    outputVariable="Invoke_minx_getBBoxminx_OutputVariable"/>
</sequence>
</flow>
<assign name="Assign_koord_an_url">
  <copy>
    <from variable="Invoke_minx_getBBoxminx_OutputVariable"
      part="parameters"
      query="/ns4:getBBoxminxResponseElement/ns4:koordminx"/>
    <to variable="Invoke_URL_getMVURL_InputVariable" part="parameters"
      query="/ns4:getMVURLElement/ns4:minx"/>
  </copy>
  <copy>
    <from variable="Invoke_maxx_setBBoxmaxx_OutputVariable"
      part="parameters"
      query="/ns4:setBBoxmaxxResponseElement/ns4:koordmaxx"/>
    <to variable="Invoke_URL_getMVURL_InputVariable" part="parameters"
      query="/ns4:getMVURLElement/ns4:maxx"/>
  </copy>

```

```
<copy>
  <from variable="Invoke_miny_setBBoxminy_OutputVariable"
    part="parameters"
    query="/ns4:setBBoxminyResponseElement/ns4:koordminy"/>
  <to variable="Invoke_URL_getMVURL_InputVariable" part="parameters"
    query="/ns4:getMVURLElement/ns4:miny"/>
</copy>
<copy>
  <from variable="Invoke_maxy_setBBoxmaxy_OutputVariable"
    part="parameters"
    query="/ns4:setBBoxmaxyResponseElement/ns4:koordmaxy"/>
  <to variable="Invoke_URL_getMVURL_InputVariable" part="parameters"
    query="/ns4:getMVURLElement/ns4:maxy"/>
</copy>
</assign>
<invoke name="Invoke_URL" partnerLink="WMS-Service"
  portType="ns1:WMS-Service" operation="getMVURL"
  inputVariable="Invoke_URL_getMVURL_InputVariable"
  outputVariable="Invoke_URL_getMVURL_OutputVariable"/>
<assign name="Assign_URL">
  <copy>
    <from variable="Invoke_URL_getMVURL_OutputVariable" part="parameters"
      query="/ns4:getMVURLResponseElement/ns4:result"/>
    <to variable="outputVariable" part="payload1"
      query="/client:WMS_URL/client:URL"/>
  </copy>
</assign>
<reply name="replyOutput" partnerLink="client"
  portType="client:ProzessAdresse" operation="process"
  variable="outputVariable"/>
</sequence>
</process>
```

## Anhang J: UML-Diagramm des GAWANIS-Modells (Auszug)

Auszüge aus dem erstellten UML-Diagramm der konzeptuellen Modellierung von Leitungsdaten nach dem GAWANIS-Datenmodell.

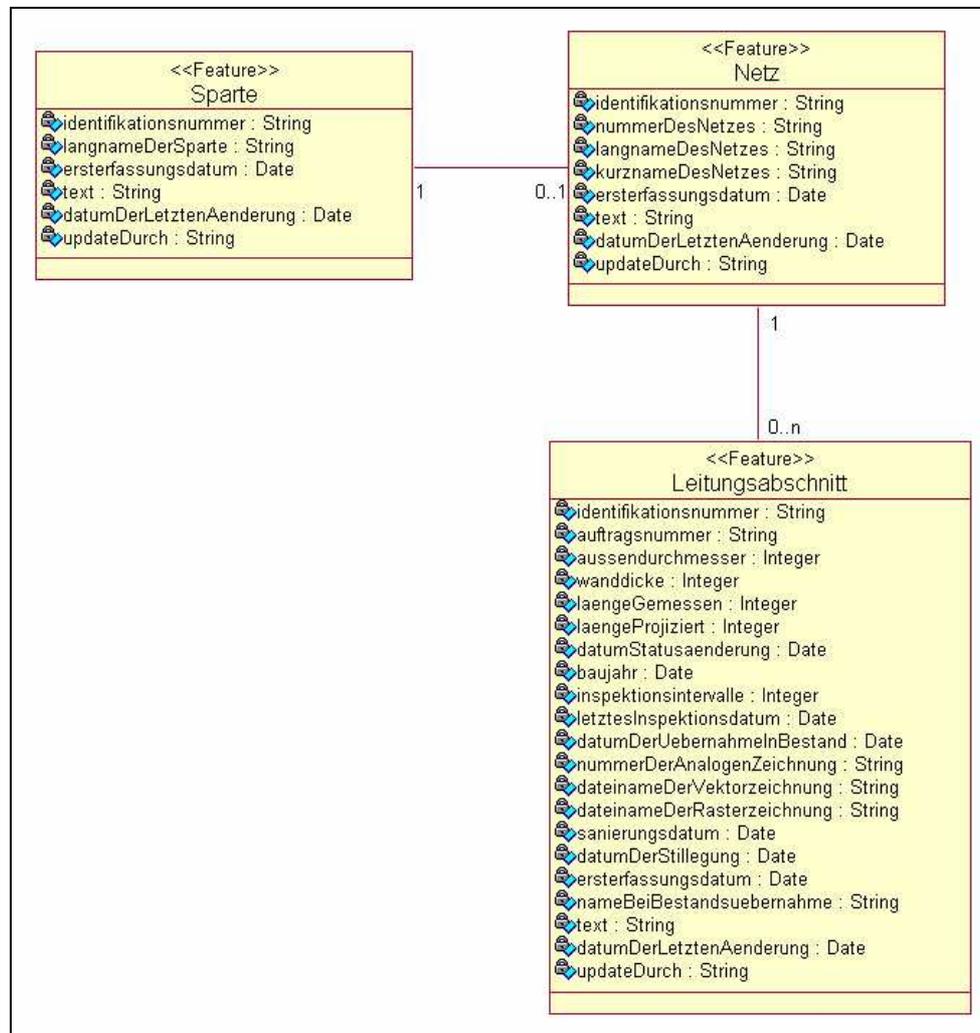


Abbildung: Zusammenspiel im GAWANIS-Datenmodell zwischen Sparte, Netz und Leitungsabschnitt

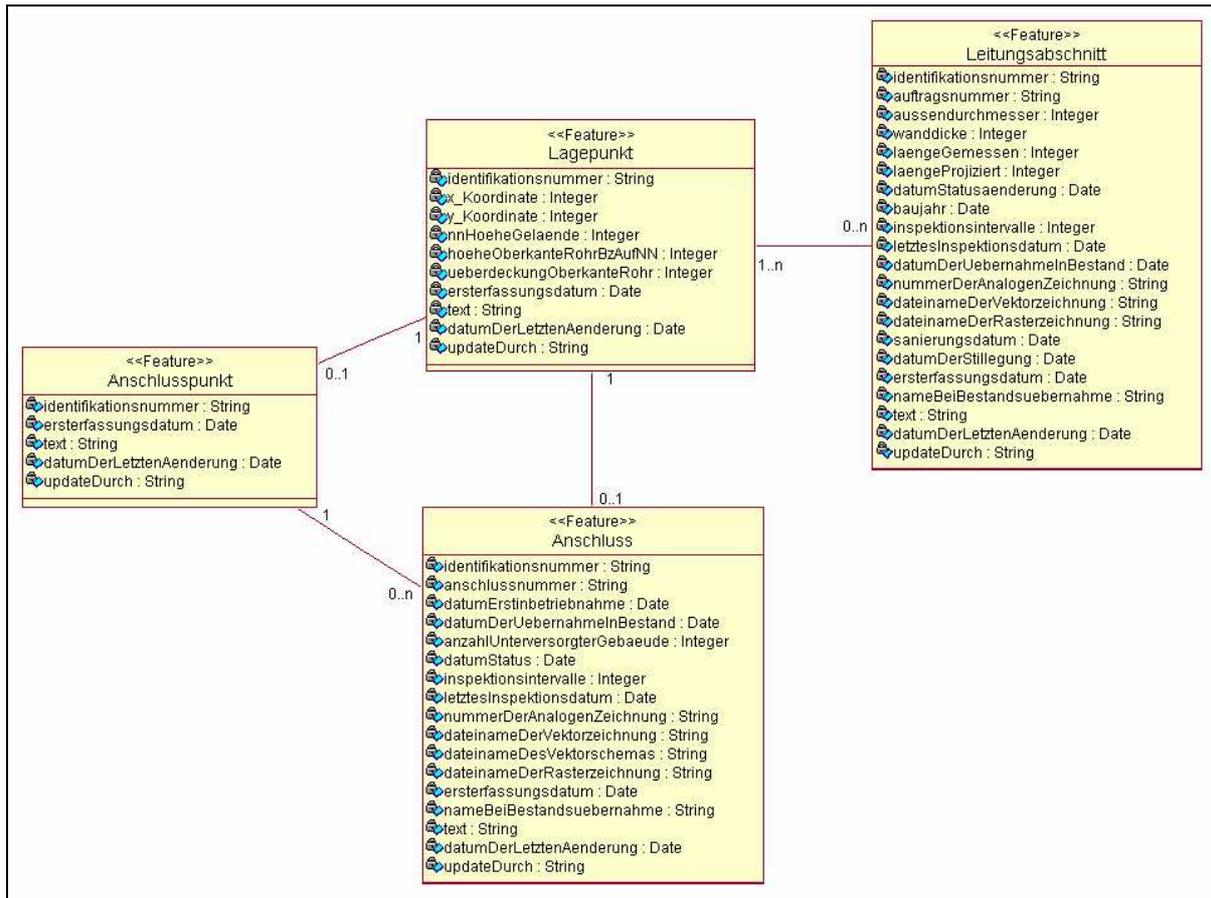


Abbildung: Zusammenspiel im GAWANIS-Datenmodell zwischen Anschlusspunkt, Anschluss, Lagepunkt und Leitungsabschnitt