

---

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN

FAKULTÄT FÜR BAUINGENIEUR- UND VERMESSUNGSWESEN

# **Zur kontextbasierten Visualisierung von Geodaten auf Basis von standardisierten Webdiensten**

Iris Wiebrock

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.) genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Kurt Brunner  
1. Berichterstatter: Univ.-Prof. Dr.-Ing. Wolfgang Reinhardt  
2. Berichterstatter: Univ.-Prof. Dr.-Ing. Alexander Zipf

Diese Dissertation wurde am 17. Dezember 2010 bei der Universität der Bundeswehr München eingereicht.

Tag der mündlichen Prüfung: 06. Mai 2011



---

## Zusammenfassung

Die Nutzung von Kartendarstellungsdiensten im Web hat sich in den letzten Jahren stark erhöht. Der Nutzer findet auf Anfrage Informationen zu Ortschaften, Restaurants oder Sehenswürdigkeiten. Doch vor allem im Web wird eine große Vielfalt an Geodaten zur Verfügung gestellt, aus denen relevante Daten gefiltert werden müssen. Außerdem verwendet der Nutzer verschiedene Ausgabegeräte und die häufige Änderung der Umgebung des Nutzers beeinflusst die Geodatenvisualisierung. Aus diesem Grund besteht eine Notwendigkeit an kontextbezogenen Anwendungen.

Kontextbezogene Anwendungen passen sich abhängig von der Umgebung ihres Nutzers in Informationsselektion und Signaturierung an. Ein Hauptproblem von kontextsensitiven Systemen ist die Nutzung verschiedener Kontextkodierungen und der Weg, Zugriff auf diese Ressourcen zu erhalten. Jedes System hat sein eigenes Format und seine eigene Kommunikation. Dieser Sachverhalt kann teilweise durch die Verwendung von Normen und Standards wie dem Web-Kartendarstellungsdienst, Web Map Service (WMS) definiert in ISO 19128 – Web map server interface gelöst werden. Diese Norm hat sich in der Welt der Geodienste etabliert. Die Erzeugung der Darstellung mit Hilfe eines WMS erfolgt jedoch ohne Berücksichtigung des Kontextkonzeptes. Aus diesem Grund soll in dieser Arbeit im Bereich Geodatenvisualisierung eine Erweiterung von standardisierten Diensten durchgeführt werden.

Die Arbeit konzentriert sich auf ein Konzept zur kontextbasierten Geodatenvisualisierung für verschiedene Nutzer in Bezug auf ihre spezifischen Anforderungen. Als Grundlage zur Kontextmodellierung werden verschiedene Kontextmodelle aus der Literatur genutzt und anhand zweier Szenarios aus den Bereichen Freizeit und Militär im Bereich Geodatenvisualisierung geprüft und erweitert. Auf dieser Grundlage wird ein Kontextmodell für verschiedene Nutzergruppen definiert. Ferner wird der Einfluss des Kontextmodells auf den Darstellungsprozess analysiert. Aufgrund der kontextsensitiven Anforderung muss auch ein Portrayalmodell, welches die Darstellungsregeln und Signaturen zur Darstellung enthält, um Kontextelemente erweitert werden.

Außerdem werden zur Unterstützung des kontextabhängigen Darstellungsprozesses Regeln definiert, die die Geodaten filtern und die Signaturierung anpassen. Dieser Prozess wird in einem erweiterten standardisierten Darstellungsdienst WMS realisiert (WMS<sup>+</sup>) und beinhaltet Operationen zur Abfrage der Kontextschemadatei, zur Übertragung einer Kontextdatei an den Dienst und zur Erzeugung der Darstellung der Geodaten in Abhängigkeit des Kontextes. Das vorgeschlagene Konzept wurde anschließend in praktischen Tests mit Hilfe eines Prototyps überprüft und mit Anwendern diskutiert.

Durch die Verwendung von Kontextinformationen wird die Visualisierung der Geoinformationen für den Nutzer und seine Situation angepasst. Das vorgeschlagene Konzept in dieser Arbeit unterstützt die Visualisierung von Geodaten mit Hilfe von Standards und in Abhängigkeit des Kontextes.

## Abstract

The use of Web Map Services has greatly increased in the last decade. A common scenario is where the user interacts with a web site to retrieve information about objects of interest like towns, restaurants and resort sights. Behind the user interfaces (e.g. web application clients), there is a wide range of geodata that have to be filtered for relevance. Besides the user can utilize different display devices and his environment can change frequently which also influences the geodata visualization. Therefore there is need for context-sensitive applications.

Depending on the user environment the context-sensitive applications have to adapt their data selection and symbolization. The main problem of context-sensitive systems is the usage of different context encodings and the access to the various different data resources. Every system has its own format and means of communication. The proprietary issues of concern can be partly resolved through usage of standards like the Web Map Service (WMS), which is the ISO 19128 – Web map server interface. This standard is well-established in the geoservices world. However, the generation of geodata visualizations with the help of a WMS does not consider the concept of context. For this reason the research work presented in this dissertation extends the existing standardized geodata visualization services.

The research work focuses on a concept for context-based geodata visualization for different users with regards to their specific needs. For modelling the context concept, two scenarios from the civilian (leisure) and military area are analyzed and extended. In addition, context models are defined for various user groups and the influence of these context models on the portrayal process is analyzed. As a result of the context-sensitive requirement a portrayal model which contains portrayal rules and symbols is extended to include context elements.

For the support of the context-dependent portrayal process, rules are defined which filter the geodata and adapt the symbolization. This process is realized as extended standardized WMS (WMS<sup>+</sup>) with operations to request a context schema, to transfer a context file to the service and to generate the geodata display depending upon the context. The proposed concept has been practically proved in a prototype implementation and discussed with the users.

The visualization of the geodata for the user, based on his current situation, is adapted by the use of context information. The concept suggested by this dissertation supports visualization that is based on standards and context sensitive.

## Inhaltsverzeichnis

<b>1.</b>	<b>EINFÜHRUNG.....</b>	<b>9</b>
1.1.	MOTIVATION.....	9
1.2.	ZIELSETZUNG DER ARBEIT .....	10
1.3.	AUFBAU DER ARBEIT .....	11
<b>2.</b>	<b>PORTRAYAL UND PORTRAYALWEBDIENSTE.....</b>	<b>13</b>
2.1.	TECHNOLOGIEN UND STANDARDS FÜR WEB SERVICES.....	13
2.1.1.	Web Service Architektur .....	13
	<i>Kommunikationsebene - Communications.....</i>	<i>14</i>
	<i>Nachrichtenebene - Messages.....</i>	<i>14</i>
	<i>Beschreibungsebene - Descriptions.....</i>	<i>16</i>
	<i>Prozessdefinitionsebene - Processes .....</i>	<i>16</i>
2.1.2.	„Publish-Find-Bind“-Prinzip und SOA-Definition .....	17
2.1.3.	Standards für Geoinformationsdienste .....	17
2.2.	PRÄSENTATION VON GEOINFORMATIONEN: DEFINITION PORTRAYAL ..	20
2.2.1.	Existierende Signaturenkataloge .....	23
	<i>ATKIS-Signaturenkataloge.....</i>	<i>23</i>
	<i>MIL-DTL-89045 GeoSym .....</i>	<i>24</i>
	<i>COP und REP .....</i>	<i>25</i>
2.2.2.	Ansätze zur Visualisierung .....	26
	<i>Desktop-GIS gestützte Visualisierung.....</i>	<i>26</i>
	<i>Web gestützte Visualisierung (Web-Mapping) und dienstebasierte Visualisierung .....</i>	<i>29</i>
2.3.	STANDARDS UND NORMEN IM BEREICH PORTRAYAL UND PORTRAYALWEBDIENSTE.....	31
2.3.1.	ISO Portrayalmodell (Informationsmodell).....	33
2.3.2.	OGC Regeln und Signaturen .....	39
2.3.3.	Portrayalwebdienst .....	39
	<i>Portrayalwebdienste allgemein und OGC portrayal pipeline.....</i>	<i>40</i>
	<i>Portrayalwebdienst ISO 19128 - WMS.....</i>	<i>41</i>
	<i>Portrayalwebdienst OGC SLD-WMS.....</i>	<i>42</i>
2.3.4.	Verwaltung der Regeln und Signaturen.....	45
2.3.5.	Speicherung von Metainformationen für einen Portrayalwebdienst .....	45
2.3.6.	Digitale Ausgabeformate .....	46
2.4.	FAZIT .....	46
<b>3.</b>	<b>KONTEXTABHÄNGIGE VISUALISIERUNG VON GEODATEN .....</b>	<b>47</b>
3.1.	KONTEXTSENSITIVE SYSTEME .....	47
3.1.1.	Adaptive Systeme im Bereich Informatik .....	47
	<i>Klassifikation adaptiver Systeme.....</i>	<i>48</i>
	<i>Nutzermodellierung .....</i>	<i>49</i>
3.1.2.	Kontext und kontextsensitive Systeme .....	51
	<i>Kontextinformationen/-gewinnung.....</i>	<i>53</i>
	<i>Kontextmodellierung.....</i>	<i>54</i>
3.2.	KONTEXTSENSITIVE SYSTEME UND GEODATENVISUALISIERUNG .....	57

<i>Allgemeine Geoanwendungen</i> .....	58
<i>Mobile Geoanwendungen</i> .....	60
<i>Ubiquitous Public Access</i> .....	64
3.3. FAZIT.....	65
<b>4. KOMPONENTEN ZUR KONTEXTABHÄNGIGEN GEODATENVISUALISIERUNG.....</b>	<b>67</b>
4.1. KOMPONENTENDEFINITION.....	67
4.2. ANWENDUNGSSZENARIOEN.....	68
4.2.1. Anwendungsszenario Bereich Freizeit: Freizeitplanung und Notfallsituation.....	69
4.2.2. Anwendungsszenario Bereich Militär: Marsch.....	70
4.2.3. Nutzerrollen.....	72
4.3. KONTEXT.....	75
4.3.1. Primärkontext: Modellierungsebenen Kontextmodell.....	76
<i>Konzeptionelles Modell: UML – Klassendiagramm</i> .....	76
<i>Logisches Modell: Modellierung in XML Schema (XSD)</i> .....	80
4.3.2. Einfluss des Kontextmodells auf die Geoobjektauswahl, Signaturierung, Bildschirmanzeige und Werkzeuge.....	82
4.3.3. Sekundärkontext.....	84
4.4. KONTEXTABHÄNGIGES PORTRAYALMODELL.....	87
4.5. DISKUSSION.....	89
<b>5. KONTEXTABHÄNGIGE GEODATENVISUALISIERUNG FÜR NUTZER EINES STANDARDISIERTEN PORTRAYALWEBDIENSTES.....</b>	<b>91</b>
5.1. KONTEXTABHÄNGIGER PORTRAYALPROZESS UND WERKZEUGAUSWAHL.....	91
5.2. ADAPTIONSREGELN UND ANGEPASSTE „PORTRAYAL PIPELINE“.....	99
5.2.1. Adaptionsregeln: Anpassung der Präsentation.....	99
5.2.2. Adaptionsregeln: Auswahl der Werkzeuge.....	102
5.2.3. Angepasste „Portrayal Pipeline“.....	103
5.3. ERWEITERUNGEN DES WMS (WMS <sup>+</sup> ): OPERATIONEN (GESAMTFRAMEWORK).....	104
5.3.1. DescribeContextType Operation.....	105
5.3.2. GetContext Operation.....	105
5.3.3. TransactContext Operation.....	106
5.3.4. Erweiterte GetMap Operation.....	108
5.3.5. Erweiterte GetCapabilities Operation.....	109
5.4. DISKUSSION.....	110
<b>6. PRAKTISCHE ERGEBNISSE.....</b>	<b>113</b>
6.1. ARCHITEKTUR DES PROTOTYPSYSTEMS.....	113
6.2. TECHNOLOGIEN DES PROTOTYPSYSTEMS.....	115
6.2.1. Oracle Spatial Datenbank.....	115
6.2.2. Server: Portrayalwebdienst.....	117
6.3. ERGEBNISSE DER TESTS UND DISKUSSION.....	118

---

<b>7.</b>	<b>ZUSAMMENFASSUNG UND AUSBLICK .....</b>	<b>123</b>
<b>8.</b>	<b>ABKÜRZUNGSVERZEICHNIS.....</b>	<b>125</b>
<b>9.</b>	<b>LITERATURVERZEICHNIS .....</b>	<b>127</b>
<b>10.</b>	<b>ABBILDUNGSVERZEICHNIS.....</b>	<b>135</b>
<b>11.</b>	<b>TABELLENVERZEICHNIS .....</b>	<b>139</b>
<b>12.</b>	<b>ANHANG.....</b>	<b>141</b>
12.1.	XSDS (KONTEXTSCHEMA) UND XML-BEISPIELDATEI (KONTEXT) .....	141
12.2.	ERWEITERUNG GETCAPABILITIES DOKUMENT .....	144
12.3.	ADAPTIONSREGELN .....	144
12.4.	XSQL ACTION ELEMENT HANDLER UND XMLTYPE FUNKTIONEN.....	147
12.5.	UML NOTATIONEN UND ANDERE DARSTELLUNGEN .....	149
<b>DANK</b>	<b>.....</b>	<b>151</b>
<b>LEBENS LAUF</b>	<b>.....</b>	<b>152</b>





---

# 1. Einführung

## 1.1. Motivation

Die Nutzung von Kartendarstellungsdiensten im Web ist in den letzten Jahren stark gewachsen. Auf verschiedenen Internetseiten stellen Anbieter wie Map24 (<http://www.de.map24.com>) oder Google Maps (<http://maps.google.de>) dem Nutzer digitale Kartendarstellungen als Dienste zur Verfügung. Somit benötigt der Nutzer kein lokales GIS-System und findet auf Dienstanfrage Informationen zu Ortschaften, Restaurants oder Sehenswürdigkeiten.

Neben der Zunahme von Diensten steigen auch die Geodatenmengen an. Ziel ist es somit, einen Zugang zu verteilten, heterogenen Geodaten und Geodiensten für möglichst viele Anwender zu schaffen, um diesen eine einfache und direkte Nutzung von Geodaten zu ermöglichen. Erst Normen bzw. Standards ermöglichen das reibungslose Zusammenspiel heterogener Geodaten und Geodienste. In DONAUBAUER (2004) wird zum Beispiel der Fokus auf die Verfügbarkeit von Geodaten über das Internet auf Basis internationaler Normen und Standards gelegt. Unter anderem wurde auch die Nutzung von Web-Kartendarstellungsdiensten, Web Map Service (WMS), durch Definition von Operationen in der Norm ISO 19128 – Web map server interface (ISO (2005b)) standardisiert. Diese Norm definiert die Anfragen an einen Dienst zur Erstellung eines digitalen Kartenbildes aus verschiedenen Geodatenquellen.

Der „GeoViewer“, eine GeoPortal.Bund-Teilkomponente, bietet einen Einstieg in die an GeoPortal.Bund angeschlossenen und integrierten Dienste und Daten und ist ein Beispiel für einen Client, der die Nutzung von standardisierten Kartendarstellungsdiensten im Internet (WMS) umgesetzt hat. GeoPortal.Bund ist der Einstiegspunkt der GDI-DE im Internet und stellt somit die zentrale Informationsplattform der GDI-DE dar. Das Gesetz über den Zugang zu digitalen Geodaten (Geodatenzugangsgesetz - GeoZG) dient dem Aufbau der nationalen Geodateninfrastruktur (GDI-DE). Es schafft den rechtlichen Rahmen für den Zugang zu und die Nutzung von Geodaten, Geodatendiensten und Metadaten. So soll hier neben Diensten wie einem Such- und Downloaddienst auch ein Darstellungsdienst bereitstehen (GEOZG (2009)). Das GeoPortal.Bund stellt dem Nutzer aktuell über 650 Einzellayer (Karten) und über 35.000 Metadateneinträge aus vielen verschiedenen Geobereichen zur Verfügung. Die Zugriffszahlen steigen kontinuierlich und liegen Mitte 2010 bei rund 8000 pro Tag (GEOPORTAL.BUND (2010)).

Jedoch wirft die große Anzahl an Geodatenbeständen für die Verwaltung, Wirtschaft und Wissenschaft die Frage auf, wie diese nutzer- und aufgabenorientiert verwendet werden können. Normalerweise kommen Nutzer aus unterschiedlichen Anwendungsbereichen und haben verschiedene Aufgaben wie zum Beispiel Sichtbarkeitsanalysen oder Routenplanungen zu erfüllen. Ferner haben Nutzer von Geodaten und Geowebdiensten unterschiedliche Systeme (PC Monitor, TabletPC, ...) und befinden sich in wechselnden Umgebungen und somit sich dynamisch ändernden Lichtverhältnissen. Aus diesen Gründen bestehen verschiedene Nutzer- bzw. Situationsanforderungen an die Präsentation von Geoinformationen.

Ein Beispiel für eine Anwendung im Freizeitbereich im Bereich kontextsensitiver Visualisierung ist die Durchführung einer Radwanderung. Abhängig von der Aufgabe erfolgen eine Auswahl der notwendigen Geodaten und eine Änderung von Parametern der Visualisierung. Eine Aufgabe des GIS-Analysten ist hierbei die Routenplanung. Für eine Routenplanung einer Radwanderung muss eine GIS Analyse durchgeführt werden. Der Nutzer (Analyst) verfügt über verschiedene Basisdaten wie Gelände- (TIN) und

Radwegdaten (mit zusätzlichen Attributen wie Zustand und Schwierigkeitsgrad). Mit Hilfe von Analysewerkzeugen können verschiedene mögliche Routen anhand entsprechender Nutzerkriterien berechnet und zugeordnet werden. Relevante Kriterien sind z. B. Tourart (Mountainbike, Sportradfahren, ...) und POIs (Sehenswürdigkeiten, Rastmöglichkeiten, Bahnhöfe, Parkplätze, Badeplätze, ...). Ferner stellt der Analyst die zuvor berechneten Routen bereit. Ein anderer Nutzer (Auskunftsuchender) führt die Radwanderung durch und muss hierfür die Route abhängig von verschiedenen Kriterien des Nutzers selektieren. Nach der Selektion kann die Durchführung der Radwanderung mit Anzeige der Route während der Wanderung im Gelände beginnen. Ein zweites Beispiel verdeutlicht die Visualisierungsanforderung an eine bestimmte Situation. Verändert sich die Umgebung des Nutzers zum Beispiel von Tag zu Nacht, sollte sich auch die Kartendarstellung aufhellen, um eine weitere Kartennutzung möglich zu machen. Folglich sind verschiedene Präsentationen wie Tag- und Nachtsicht notwendig.

Aus diesem Grund profitiert der Nutzer von der adaptiven Visualisierung von Geodaten in Abhängigkeit seiner Aufgabe und Situation. Gerade mit dem WWW und dem damit verbundenen Anstieg an ein größer werdendes verschiedenartiges Publikum ist der Bedarf nach Adaption und somit die Forschung in diesem Bereich gestiegen. Der Adaptionbedarf reicht im Bereich der Geodatenvisualisierung vom Nutzerinterface über die Geodatenauswahl bis hin zur Signaturierung der Geodaten. Auch Fragestellungen welche Anforderungen auf welche Art die Präsentation beeinflussen oder auf welche Weise die kontextbasierte Präsentation standard- und webbasiert erzeugt werden kann sind offen.

Jetzige Kartendarstellungsdienste im Internet wie Google Maps treffen die Geodatenauswahl zum Beispiel schon nutzerbasiert. Dieser Dienst erlaubt eine Einstellung des Nutzers zur Fortbewegungsart (Autofahrer oder Fußgänger) oder die Wahl Autobahnen zu vermeiden, um auf dieser Basis die Route vom Start- zum Zielort zu berechnen. Doch zurzeit existieren nur einzelne proprietäre Lösungen und jedes System hat sein eigenes Format und seine eigenen Protokolle. Außerdem werden keine einheitlichen Nutzer- und Situationsmodelle verwendet. Zusammengefasst existiert kein Gesamtrahmen bzw. -konzept zur Erzeugung von nutzer- und situationsabhängigen Präsentationen mithilfe von standardisierten Geowebdiensten.

## **1.2. Zielsetzung der Arbeit**

Die Arbeit soll auf dem Gebiet der „Kontextbasierten Visualisierung für Darstellungsdienste im Web“ einen Beitrag leisten. Diese Arbeit verfolgt das Ziel einen Rahmen zu entwickeln und zu prüfen, der es erlaubt spezifische für verschiedene Nutzergruppen angepasste Geodatenpräsentationen mithilfe von standardisierten Geowebdiensten zu erzeugen. Dabei sind vor allem folgende Forderungen zu erfüllen:

- Kontextbasiert: Erstellung eines Kontextmodells basierend auf Nutzergruppen definiert an einzelnen Beispielen und Berücksichtigung des Kontextes (Nutzer und seine Situation) bei der Präsentation von Geodaten
- Webbasiert: Berücksichtigung existierender standardisierter Geowebdienste
- Standardbasiert: Erweiterung bestehender Normen und Standards unter Berücksichtigung der Implementierungsmöglichkeiten

---

Es wird in dieser Arbeit der Schwerpunkt auf die Visualisierung der Geodaten gelegt. Generalisierung, Verdrängung und andere Verfahren, um eine endgültige digitale Karte herzustellen, werden nicht berücksichtigt.

Der zur Erreichung dieser Ziele formulierte Lösungsansatz stützt sich einerseits auf die Arbeiten und Fortschritte im Bereich Web-Mapping und Standardisierung und zum anderen auf die Arbeiten im Bereich des Kontextes. Um die definierten Forderungen zu unterstützen, muss ein Kontextmodell basierend auf einzelnen Szenarien definiert werden. Außerdem müssen die Regeln zur Signaturierung der Geodaten erweitert werden, um den Kontext berücksichtigen zu können. Ein kontextabhängiger Portrayalprozess mit Geodatenselektion und Anpassung der Signaturierung entsprechend des Kontextes ist notwendig, um eine nutzerspezifische Präsentationen zu erhalten. Auf verschiedenen Stufen des Visualisierungsprozesses werden Adaptionsregeln ausgeführt und die endgültige nutzerspezifische Präsentation wird entsprechend des Kontextmodells via „Web Map Service“ generiert und anschließend auf dem Clientbildschirm dargestellt. Hierfür sind eine Definition der Adaptionsregeln und eine Erweiterung des standardisierten Darstellungsdienstes notwendig.

Das vorgeschlagene Konzept verbessert die Nutzbarkeit von Darstellungsdiensten und unterstützt die „Entscheidungsunterstützungsaktivitäten“ von verschiedenen Nutzergruppen entsprechend ihres Kontextes. Durch die dynamische Anpassung der Darstellung der Geodaten nach Nutzer und Aufgabe ist z.B. eine Unterstützung im Bereich „decision support“ möglich, um situationsbezogene Entscheidungen treffen zu können. Außerdem wird so auch der Trend in Richtung Benutzerfreundlichkeit („Usability“) unterstützt. Aufgrund des standardbasierten Ansatzes wird die Nutzung von verschiedenen Datenquellen ermöglicht.

### **1.3. Aufbau der Arbeit**

Die Grundlagen und der Stand der Forschung und Technik werden in Kapitel 2 und 3 beschrieben. In Kapitel 2 (Portrayal und Portrayalwebdienste) werden zu Beginn Technologien und Standards für Web Services beschrieben. Anschließend wird der Begriff „Portrayal“, existierende Signaturenkataloge und Ansätze zur Visualisierung erläutert. Auf dieser Grundlage werden die verschiedenen Normungs- und Standardisierungsaktivitäten der ISO und OGC im Bereich Portrayal und Portrayaldiensten beschrieben und analysiert. In Kapitel 3 (Kontextabhängige Visualisierung von Geodaten) werden zu Beginn kontextsensitive Systeme erläutert. Dies beinhaltet adaptive Systeme im Bereich Informatik und Kontext und kontextsensitive Systeme. Im Anschluss wird auf kontextsensitive Systeme im Bereich Geodatenvisualisierung eingegangen. Die eigenen Arbeiten werden in Kapitel 4 und folgenden vorgestellt. In Kapitel 4 (Komponenten zur kontextabhängigen Geodatenvisualisierung) werden die einzelnen notwendigen Komponenten definiert und beschrieben. Es wird anhand von Anwendungsszenarien ein Kontextmodell entwickelt und der Einfluss des Kontextmodells auf die Geodatenauswahl, Signaturierung, Bildschirmanzeige und Werkzeuge analysiert. Um dieses Modell im kontextabhängigen Portrayalprozess verwenden zu können, muss das Portrayalmodell erweitert werden. Die kontextabhängige Geodatenvisualisierung für Nutzer eines standardisierten Portrayalwebdienstes wird in Kapitel 5 erläutert. Es wird ein kontextabhängiger Portrayalprozess empfohlen, um eine kontextabhängige Präsentation erstellen zu können. Hierfür ist die Definition von Adaptionsregeln notwendig. Im Anschluss wird eine angepasste „Portrayal pipeline“ beschrieben und die Operationen für einen erweiterten Darstellungsdienst erläutert. In Kapitel 6 (Praktische

Ergebnisse) wird der Prototyp erläutert und die Tests bewertet. Die Ergebnisse werden in Kapitel 7 zusammengefasst und ein Ausblick gegeben. In Anhang 12.1 finden sich Schema und Beispieldatei für den in dieser Arbeit definierten Kontext. Die Erweiterung des „GetCapabilities“-Dokuments ist in Anhang 12.2 zu finden. In Anhang 12.3 sind Adaptionenregeln im Allgemeinen und als SQL-Abfragen für Oracle definiert. Zur Umsetzung des Prototypsystems wurde das Oracle XDK (XML Developer's Kit) verwendet. Dies ist eine Sammlung von Tools für die Schnittstelle von XML und (Oracle-) Datenbank. Oracle XDK beinhaltet unter anderem XMLType Funktionen und XSQL Action Element Handler. Diese sind in Anhang 12.4 zu finden. Im Anhang 12.5 finden sich verwendete UML-Notationen.

## 2. Portrayal und Portrayalwebdienste

Im Folgenden werden die Grundlagen im Bereich Portrayal sowie der Stand der Forschung und Technik zu den Themengebieten, die in dieser Arbeit berührt werden, zusammengefasst. Dies beinhaltet Grundlagen, Technologien und Standards für Web Services im Bereich Informatik allgemein und Standards und Normen im Bereich Portrayal im speziellen.

### 2.1. Technologien und Standards für Web Services

In diesem Kapitel werden allgemeine Technologien für Web Services wie Dienstarchitekturen beschrieben, da diese die Basis für die Webdienste im Geobereich darstellen. Das WWW-Konzept von Berners-Lee aus dem Jahre 1985 basiert auf den Komponenten: Webserver und Webclient. Ein Webserver ist im allgemeinen „eine Serverapplikation, welche auf Anfrage im HTTP-Protokoll reagiert und die entsprechenden Antworten generiert“ (WALTER (2008)). Ein Webclient stellt die Anfrage an den Webserver und zeigt die Antwort formatiert z.B. in HTML (HyperText Markup Language) an. Die Kommunikation findet über das Vermittlungsprotokoll HTTP (Hypertext Transfer Protocol) statt.

Verschiedene Organisationen beschäftigen sich mit der Spezifizierung im Bereich Web und Web Services. Die Internet Engineering Task Force (IETF) begann 1986 ihre Arbeit. Zur Internetprotokollfamilie der IETF gehören beispielsweise das Anwendungsprotokoll HTTP/1.1 (IETF (1999)) zur Übertragung von Web-Inhalten oder MIME (Multipurpose Internet Mail Extensions) (IETF (1996)), welches ermöglicht, Informationen über den Typ der übermittelten Daten zwischen Client und Server auszutauschen. Das World Wide Web Consortium (W3C) wurde 1994 gegründet, um Standards zu entwickeln, die den Fortschritt des Webs fördern und die Interoperabilität sicherstellen sollen. Bekannteste Standards sind HTML (W3C (1999)), XML (W3C (2000)) und XML Schema (W3C (2004c)). HTML ist eine Sprache, die es ermöglicht, Informationen im Internet zu präsentieren. Nähere Erläuterungen zu XML und XML Schema sind im Anschluss zu finden.

#### 2.1.1. Web Service Architektur

Web Services erleichtern den Zugriff auf verschiedene Daten über wohldefinierte Schnittstellen. Der standardisierte Austausch findet mit Hilfe von XML kodierten Daten über bekannte Datenübertragungsprotokolle wie HTTP statt. Vorteile von Web Services sind des Weiteren die Kontrolle der Daten und des Layouts, die Weiterverarbeitung der Daten und die Plattformunabhängigkeit. Ein Nachteil bei der Verwendung von Web Services kann zum Beispiel die Performanz sein.

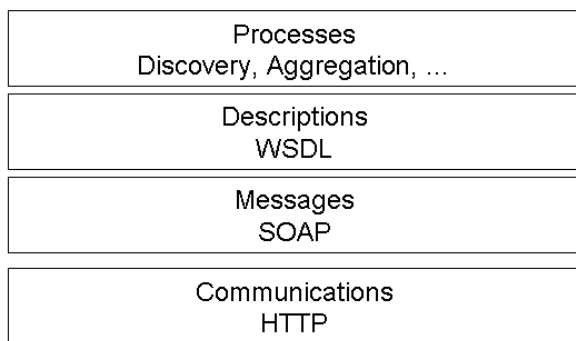
Der Zugriff auf Web Services erfordert deren eindeutige Bezeichnung, Informationen über ihren Ort sowie geeignete Zugriffsmöglichkeiten. Im Web sind besonders zwei Standards der IETF von Bedeutung: Das oben erwähnte HTTP Protokoll als Transportprotokoll und der Uniform Resource Identifier (URI) zur eindeutigen Identifizierung einer (Daten)quelle.

**Definition 2.1 (URI, Uniform Resource Identifier)** *“Ein Uniform Resource Identifier (URI) („einheitlicher Bezeichner für Ressourcen“) ist ein Identifikator und besteht aus einer Zeichenfolge, die zur Identifizierung einer Ressource dient.“ (IETF (2005))*

Das W3C (W3C (2002)) definiert Web Services als „a software application identified by a URI, whose

*interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols.”*

Im Anschluss wird die W3C Web Service Architektur beschrieben. Sie wird durch verschiedene Ebenen definiert: Kommunikationsebene, eine Nachrichtenebene, eine Beschreibungsebene und eine Prozessdefinitionsebene (Abbildung 1). Die Beschreibungsebene („Descriptions“) beschreibt Nachrichten, die zwischen dem Provider und dem Nutzer ausgetauscht werden. Eine Nachricht muss an ein Transportprotokoll („Communications“) und an Nachrichtenformat („Messages“) gebunden werden.



**Abbildung 1: Web Service Architektur nach W3C (2004a)**

Die einzelnen Ebenen werden im Weiteren erläutert, wobei zuerst Ziel und Inhalt und anschließend die Anwendung der Ebene beschrieben werden.

#### ***Kommunikationsebene - Communications***

Das Kommunikationsprotokoll regelt, wie zwischen Webclient und Webserver Informationen ausgetauscht werden können. Dazu sind Steuerungs- und Kontrollmechanismen für die Verbindung und den Datenaustausch implementiert.

Das *HTTP* (HyperText Transfer Protocol) ist ein Transportprotokoll zur Übertragung von Daten über ein Netzwerk. Es wird hauptsächlich eingesetzt, um Webseiten aus dem World Wide Web (WWW) in einen Webbrowser zu laden. Neben HTTP existieren andere Transportprotokolle wie *SMTP* (Simple Mail Transfer Protocol) oder *FTP* (File Transfer Protocol) (TANNENBAUM (2003)).

#### ***Nachrichtenebene - Messages***

Die „Message“ (Nachricht) ist eine Dateneinheit und wird zwischen Provider Agents (Akteur, welcher Daten oder Information anfragt) und Requester Agents (Akteur, welcher Informationen bereitstellt) ausgetauscht.

Es gibt verschiedene Nachrichtenformate. Als Nachrichtenformat kann zum Beispiel *HTTP* verwendet werden. Die Übertragungsmechanismen der Daten sind u. a. GET und POST. Bei einer GET-Anfrage wird der Datenteil durch das Sonderzeichen ? getrennt an die Adresse angefügt und die einzelnen Key-Value-Paare durch & getrennt.

Beispiel: `http://rechnername/scriptname?feld1=wert1&feld2=wert2`

Ein Vorteil ist hierbei, dass der erzeugte Request vollständig sichtbar ist und in der Browser-History angezeigt werden kann. Probleme kann es hingegen bei der Übermittlung sehr langer Datenblöcke geben. In diesem Fall kann die Anfrage POST verwendet werden. Hier wird parallel zum HTTP-Request zur Adresse ein separater Übermittlungskanal für die Daten genutzt. Mit der Version 1.1 stellt HTTP zusätzlich die Methode PUT und DELETE bereit, um HTTP-Dateien auf einem Server abzulegen oder zu löschen.

*SOAP* (ursprünglich für Simple Object Access Protocol, aber Akronym wird seit der aktuellen Version (1.2) nicht mehr gebraucht) ist eine weitere Möglichkeit der Nachrichtenübertragung. Sie wurde entwickelt, um XML-Dokumente zu transportieren. Ein XML-Dokument wird dabei mit einem „Umschlag“ versehen, der die Daten mit Verwaltungsinformationen anreichert. *XML* (Extensible Markup Language) (W3C (2000)) ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten. Das W3C definiert ein Datenobjekt als ein XML-Dokument „, wenn es im Sinne der *XML-Spezifikation wohlgeformt ist.* (W3C (2000)). „*Ein wohlgeformtes XML-Dokument kann darüber hinaus gültig sein, sofern es bestimmten weiteren Einschränkungen genügt. ... Ein XML-Dokument ist gültig, wenn es eine dazugehörige Dokumenttyp-Deklaration besitzt und wenn sich das Dokument an die darin formulierten Beschränkungen hält.*“ (W3C (2000)).

Die Vorteile von XML sind vielfältig. XML ist erweiterbar und plattformunabhängig. XML-Dateien bestehen aus Text und können, wenn notwendig, auch vom Menschen gelesen werden. Außerdem ist XML modular und der Nutzer ist somit in der Lage ein neues Dokumentenformat zu definieren, indem er andere Formate kombiniert. Bei den Nachteilen kann vor allem die Dateigröße genannt werden, denn XML-Dateien sind fast immer größer als vergleichbare binäre Formate. Eine XML Einführung bietet zum Beispiel RRZN (2002). Weitere W3C-Empfehlungen die in diesem Zusammenhang genannt werden sollten sind: *XML Schema* (W3C (2004c), W3C (2004d)) und *XSLT* (XSL Transformations) (W3C (1999)). Die W3C-Empfehlung *XML Schema* ist ebenfalls eine wichtige Komponente bei Webservice-Spezifikationen. Es definiert mit Hilfe von Regeln logische Strukturen für XML-Dokumente. So kann ihre Dokumentation und Validierung ermöglicht werden. Näheres ist zum Beispiel in VAN DER VLIST (2002) zu finden. *XSLT* ist eine Transformationssprache, die für das Umstellen, Hinzufügen und Löschen von Tags und Attributen verwendet wird. Durch die Verwendung von XSLT kann mit Hilfe von Umwandlungsregeln ein XML-Dokument in ein anderes Dokument transformiert werden. Näheres zu XSLT ist in IRWIN and CHURCHER (2002) und MATHIAK et al. (2005) zu finden.

Eine SOAP-Nachricht ist eine XML-Nachricht und ist wie folgt aufgebaut: SOAP Envelope, SOAP Header und SOAP Body. Der SOAP Envelope („Umschlag“) enthält die eigentliche Nachricht („Brief“). Der SOAP Header ist ein optionales Element. Das Übertragen von Sicherheitsinformationen ist ein typisches Anwendungsgebiet für den SOAP Header. Der SOAP Body ist zwingend erforderlich und enthält die zu übertragenden Informationen (MELZER et al. (2007)). Mit SOAP lassen sich alle Informationen verschicken, die sich als XML-Dokumente darstellen lassen (wie HTML-Seiten, Bestellformulare, ...). Die Wahl des Kommunikationsprotokolls (HTTP, SOAP, ...) hängt von der jeweiligen Anwendung ab, doch da HTTP überall zur Verfügung steht, wird es hauptsächlich verwendet. Bei SOAP können Objekte jedoch nie direkt angesprochen werden. Dies ist einer der Gründe weshalb es zu SOAP einfachere Alternativen, wie zum Beispiel *REST* (REpresentational State Transfer) gibt. REST wurde 2000 von R. T. Fielding be-

schrieben (FIELDING (2000)). Es ist kein eigenes Protokoll, sondern ein Architekturstil, welcher definiert wie existierende Protokolle verwendet werden können. REST fokussiert sich im Gegensatz zu SOAP auf das WWW. Eine Anwendung mit REST-Architekturstil verwaltet eine Menge von Ressourcen. FIELDING (2000) definiert eine Ressource als jegliche Information, die benannt werden kann. Dies können Dokumente oder Dienste, aber auch reale Objekte wie Personen sein. Jeder Zugriff erfolgt indirekt über die der Ressource zugeordnete URI. Erreichbar sind Webservices über einen eindeutigen URI (s. Definition 2.1). Somit erfüllen sowohl SOAP als auch REST diese Anforderung.

### ***Beschreibungsebene - Descriptions***

Damit die Nachrichten interdisziplinär verwendet werden können, benötigen sie eine strikte Vereinbarung in der sogenannten Beschreibung (Description) in einer maschinen-verarbeitbaren Sprache. Eine Beschreibungsdatei enthält Informationen über das Interface und die Semantik, wie der Web Service aufgerufen werden kann.

Web Service Interfaces werden mit Hilfe von *WSDL* (Web Services Definition Language) beschrieben. Hierbei gibt vier verschiedene Informationen über Web Services: Interfaceinformationen (Beschreibung aller Operationen/Funktionen), Informationen zu den Datentypen für die ankommenden und abgehenden Nachrichten dieser Operationen, „binding information“ (um die spezifizierten spezifische Web Services aufrufen zu können) und Adresseninformationen zur Lokalisierung des spezifischen Web Services (SAYAR et al. (2005)).

### ***Prozessdefinitionsebene - Processes***

In der Ebene „Processes“ werden Prozesse definiert. Beispiele für Prozesse sind „Discovery“ und „Aggregation“ von Web Services. Discovery dient dem Auffinden von WebServices, wohingegen die Aggregation eine Komposition von einem Satz von Services ist, um ein gemeinsames Ziel zu erreichen.

Zum Auffinden von Web Services (Discovery) gibt es verschiedene Möglichkeiten. Hier sind zwei XML Register Standards zu nennen: *UDDI* (Universal Description, Discovery and Integration) (OASIS (2002)) und *ebXML* (Electronic Business using XML) (OASIS (2005)). *UDDI* ist ein Verzeichnisdienst zur Registrierung und Recherche von Web-Services und sehr einfach strukturiert. *UDDI* speichert keinen Inhalt und hat somit keine Repository, sondern nur Metadaten mit „Pointer“ zum Inhalt. So können zum Beispiel Auflistungen von Diensten gespeichert werden. *ebXML* hingegen hat ein integriertes Repository und Registry und speichert Inhalt und Metadaten. Außerdem ist *ebXML* komplexer und bietet ein erweiterbares und sicheres Informationsmanagement von jeglichem Typ. Das Informationsmodell ist im Gegensatz zu *UDDI* erweiterbar (EBXMLWIKI (2009)). Die *ebXML* Version 2.1 von OASIS entspricht der ISO/TS 15000. Ein Vergleich einzelner Elemente ist unter *EBXMLRR* (2009) zu finden.

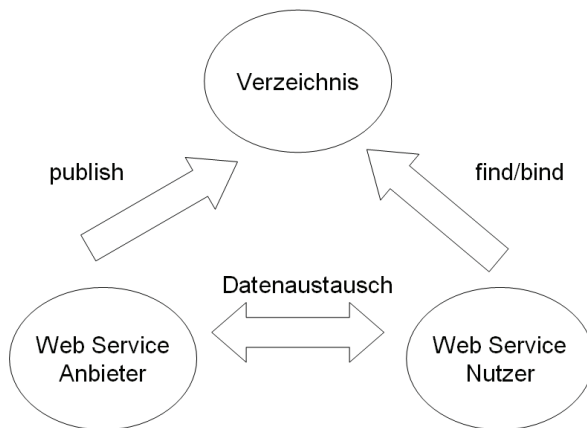
Neben der „Discovery“ gibt es auch die „Aggregation“ von Web Services. Die Aggregation ist die Komposition von einem Satz von Services, um ein gemeinsames Ziel zu erreichen. Die Aggregation ist entweder beschränkt bzw. unbeschränkt (englisch „constrained“ bzw. „unconstrained“) abhängig davon, wie viel Kontrolle über Interaktionen notwendig ist, um ein Ziel zu erreichen. Bei einer unbeschränkten Aggregation gibt es keine Kontrolle der Interaktionen zwischen den Diensten. Beispiele sind einfache Gruppierungen von Diensten. Bei den beschränkten Aggregationen hingegen werden Beschränkungen für die Dienste definiert. Dies beinhaltet vor allem die Funktionalitäten die sie unterstützen müssen oder die In-



teraktionen zwischen den Diensten (KHALAF and LEYMANN (2003)).

### 2.1.2. „Publish-Find-Bind“-Prinzip und SOA-Definition

Zum Bereich der Web Service-Technologien gehört das „Publish-Find-Bind“-Prinzip (KUSCHKE and WÖLFEL (2002)). Hier können Anbieter ihre Dienste in einem Verzeichnis veröffentlichen (*publish*), worin der Nutzer nach einem Dienst sucht. Hat der Nutzer einen Dienst gefunden (*find*), erhält er eine Referenz zur Beschreibung des Dienstes. Auf diese Beschreibung kann wiederum zugegriffen werden und er kann den Dienst in seine Anwendung einbinden (*bind*) (Abbildung 2).



**Abbildung 2: Web Service Architektur**

Die Wiederverwendbarkeit von Diensten ist einer der zentralen Vorteile, die serviceorientierte Architekturen (SOA) versprechen. MELZER et al. (2007) definiert den Begriff SOA (Service-orientierte Architektur) als „eine Systemarchitektur, die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wieder verwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht.“ Mit Hilfe von Web Services kann eine SOA umgesetzt werden. Die Komponenten einer SOA sind: Kommunikation, Dienstbeschreibung und Verzeichnisdienst. Hierfür werden zurzeit SOAP, WSDL und UDDI verwendet. Diese Komponenten entsprechen in Abbildung 1 den Ebenen Kommunikation (HTTP), Nachrichten (SOAP), Beschreibung (WSDL) und Prozess (UDDI).

### 2.1.3. Standards für Geoinformationsdienste

Im Bereich der Standardisierung in der Informatik sind Organisationen wie *W3C* (World Wide Web Consortium), *OMG* (Object Management Group) oder *OASIS* (Organization for the Advancement of Structured Information Standards) zu nennen. Standards legen in der Geoinformation die Interoperabilität im Bereich Datenakquisition, Prozessierung, Analyse, Zugriff, Portrayal und Datentransfer fest. Sie stellen eine Struktur für Anwendungsentwicklungen in Bezug zu geographischen Daten und in Beziehung zu existierenden Standards in der Informationstechnologie und Kommunikation bereit. Die beiden wichtigsten Organisationen, die sich mit der Standardisierung im Geobereich beschäftigen sind: *ISO* (International Organization for Standardization) TC211 und *OGC* (Open Geospatial Consortium). *ISO* verfasst de jure Standards (Normen) und ist somit eine Standardisierungsorganisation und *OGC* definiert de facto Stan-

dards. ISO und OGC teilen die gleichen Ziele und aus diesem Grunde werden bewährte OGC Standards auch in den ISO-Prozess eingebracht. Das OGC wurde 1994 gegründet. Ferner ist die multinationale Organisation *DGIWG* (Defense Geospatial Information Working Group) zu nennen. Sie spezifiziert Geostandards für militärische Organisationen.

Die Gruppe der ISO 191xx definiert Normen für Geodaten. Die relevanten ISO-Normen im Bereich Web-Dienste sind die ISO 19101 – Reference model document (ISO (2000)) und die ISO 19119 – Services (ISO (2005c)). In der ISO 19101– Reference model document (ISO (2000)) werden verschiedene Typen von Geoinformationsdiensten definiert, die sich aus sechs generellen Klassen von „Information Technology Services“ ableiten lassen. Diese sind:

- *Geographic human interaction services*
- *Geographic model/information management services*
- Geographic workflow/task management services (nicht relevant in ISO 19000 Serie)
- *Geographic processing services*
- Geographic communication services (nicht relevant in ISO 19000 Serie)
- Geographic system management services (nicht relevant in ISO 19000 Serie)

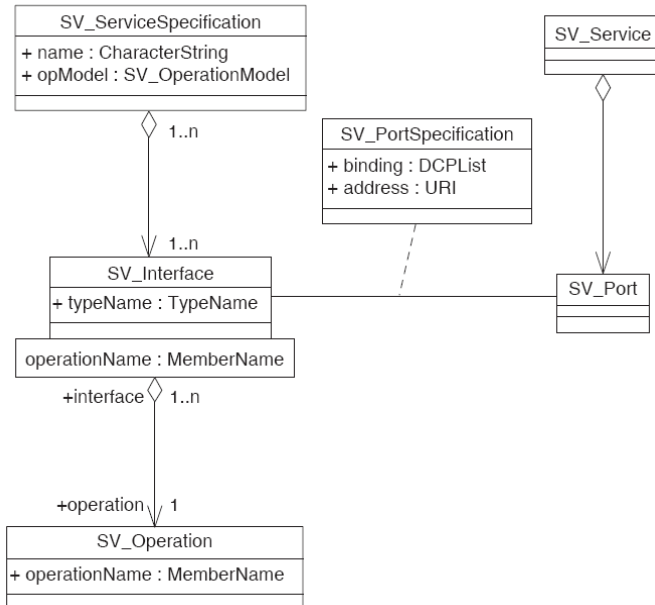
In ISO 19119 – Services (ISO (2005c)) ist definiert, wie Services spezifiziert werden müssen (u. a. Interfaces und Service-Metadaten), um Interoperabilität zu gewährleisten. Auf dieser Basis sollen verschiedene Services entwickelt werden. Im Folgenden werden die Begriffe Dienst, Schnittstelle und Operation laut ISO 19119 – Services (ISO (2005c)) definiert. Diese entsprechen den gebräuchlichen Definitionen und werden vollständigshalber im Weiteren vorgestellt.

**Definition 2.2 (Service (Dienst))** *“A Service is a distinct part of the functionality that is provided by an entity through interfaces”.* (ISO (2005c))

**Definition 2.3 (Interface (Schnittstelle))** *“An Interface is a named set of operations that characterize the behavior of an entity.”* (ISO (2005c))

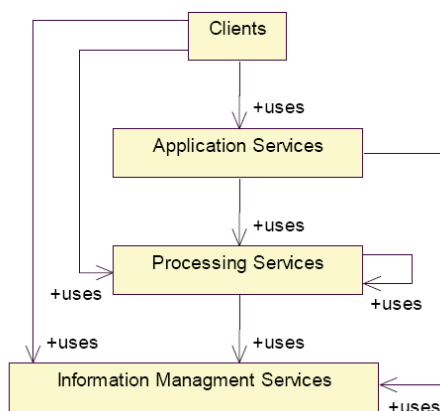
**Definition 2.4 (Operation)** *“An Operation is a specification of a transformation or query that an object may be called to execute.”* (ISO (2005c)). Jede Operation hat einen Namen und eine Liste von Parametern.

Diese werden in Abbildung 3 dargestellt. Services (SV\_ServiceSpecification) werden über einen Satz von Interfaces (SV\_Interfaces) spezifiziert. Diese wiederum haben einen Satz von Operationen (SV\_Operation). Interfaces werden als Ports (SV\_Port) implementiert, welche als Services (SV\_Service) dem Nutzer zugänglich gemacht werden. Die Implementierung (SV\_PortSpecification) beinhaltet die Implementierung der plattformspezifischen Spezifikation (binding) und die Methode den Service zum Beispiel über eine Adresse zu identifizieren (address).



**Abbildung 3: Servicedefinition nach ISO 19119 (ISO (2005c))**

Im „OGC Reference Model 2.0“ werden OGC Dienste auf der Grundlage von serviceorientierten Architekturen (OGC (2008a)) nach ISO 19119 – Services (ISO (2005c)) definiert. Ein Prinzip der OGC ist die Nutzung von offenen Internetstandards und Datentransferstandards wie XML und eine Organisation in „Multi-Tiers“ Architekturen. Diese sind: Clients, Application Services (geographic human interaction services wie web portal services, ...), geographic processing services (wie WPS (Web Processing Service)) und geographic model/information management services (wie WFS (Web Feature Service), WMS, ...) (Abbildung 4).



**Abbildung 4: Service tiers, OGC Reference Model 2.0 (OGC (2008a))**

OGC Web Services basieren ursprünglich auf WWW-Standards wie HTTP und XML. Seit kurzer Zeit werden OWS-Dienste (Open Web Services) mit Hilfe der Standards WSDL und SOAP beschrieben. SAYAR et al. (2005) schlagen ein Verfahren vor, um OGC Services mit Hilfe von WSDL Dokumenten zu

beschreiben und definieren XML Schema für alle Abfragen und Antworten von OWS-Diensten am Beispiel des OGC WMS. Jedoch gibt es bei der OGC einige „entwicklerspezifische“ Dienstparameter, so dass jeder Anbieter eines Dienstes ein etwas unterschiedliches WSDL-Dokument bereitstellt. SAYAR et al. (2006) integrierte die neue Technologie AJAX (Asynchronous JavaScript and XML, s. Kap. 2.2.2) in einen OGC WMS und WFS. Sie kamen zu dem Ergebnis, dass durch AJAX die Performanz von WMS – Services verbessert werden kann. Nachteil jedoch ist die zusätzliche Arbeit für die Entwicklung auf Browserseite.

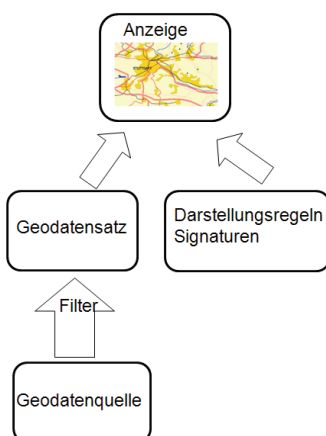
Viele OGC Standards werden in einer GDI (Geodateninfrastruktur) wie z.B. INSPIRE verwendet. Geo-Portale sind Einstiegsknoten im Internet/Intranet, die es autorisierten Benutzern erlauben, im Portal registrierte Geodaten, Geodienste und Anwendungen über das Internet/Intranet zu nutzen. Das GeoPortal ist üblicherweise der Zugang zu einer GDI und somit der Vermittler zwischen Nutzern (Clients) und Geodaten und Geodiensten. Näheres ist zum Beispiel in TAIT (2005) und TATNALL (2005) beschrieben. Beispiele für verteilte Dienstarchitekturen, die einen Darstellungsdienst beinhalten, sind unter anderem im nationalen Bereich der Deutschlandviewer der GDI-DE und im internationalen Bereich INSPIRE (Infrastructure for Spatial Information in Europe).

Im Anschluss werden verschiedene Ansätze zur Präsentation von Geoinformationen (Portrayal) beschrieben. Auf dieser Grundlage und den zuvor beschriebenen Technologien und Standards für Web Services werden in Kapitel 2.3 Standards und Normen im Bereich Portrayal und Portrayalwebdienste vorgestellt.

## 2.2. Präsentation von Geoinformationen: Definition Portrayal

Der Begriff „Portrayal“ wird in ISO 19117 – Portrayal als „*presentation of information to humans*“ (ISO (2005a)) definiert. Die im Weiteren verwendeten Begriffe werden für diese Arbeit wie folgt definiert:

**Definition 2.5 (Portrayal)** *Portrayal ist die Präsentation bzw. Visualisierung von Informationen und beschränkt sich in dieser Arbeit auf die digitale Präsentation von Geoinformationen. Portrayal beinhaltet den gesamten Prozess der Transformation der Geodaten in eine für den Menschen sichtbare Anzeige. Beim Portrayalprozess werden zunächst Geodaten aus einer Geodatenquelle gefiltert (Geodatensatz). Anschließend wird mithilfe von Darstellungsregeln und Signaturen eine sichtbare Anzeige des Geodatensatzes erzeugt (Abbildung 5).*

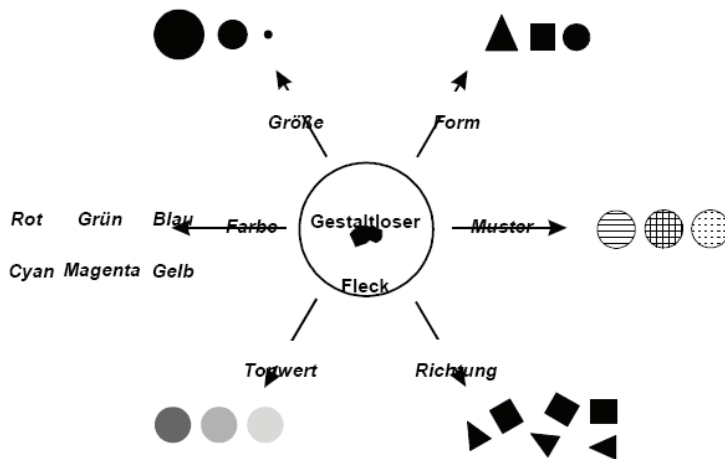


**Abbildung 5: Portrayalprozess - Präsentation von Geoinformationen**

Eine Aufgabe der Kartographie ist die „*Veranschaulichung von Geoinformationen durch kartographische Darstellungen*“ (HAKE et al. (2002)). Sie ist die zielgerichtete und möglichst störungsfreie Übermittlung von räumlichen Informationen an den Kartenbenutzer und bedient sich dazu einer Fülle von kartographischen Gestaltungsregeln. Mit der Arbeit „*The Look of Maps*“ legte ROBINSON (1952) den Grundstein für umfangreiche wissenschaftliche Untersuchungen in Wirksamkeit und Ästhetik der kartographischen Gestaltung. Es folgten Arbeiten, die sich mit der graphischen Variation von Zeichen beschäftigten.

**Definition 2.6 (Zeichen)** „*Ein Element aus einer zur Darstellung von Information vereinbarten endlichen Menge von verschiedenen Elementen.*“ (DIN (1988))

BERTIN (1973) zum Beispiel definiert verschiedene allgemeine graphische Variablen, die die Darstellung von Zeichen beeinflussen. Diese Variablen beinhalten die geographische Position eines Zeichens (engl. mark) auf der Oberfläche, welches in zwei Dimensionen variieren kann (x und y). Ein Zeichen kann weiter graphisch variieren in *Größe*, *Tonwert*, *Muster (Textur)*, *Farbe*, *Richtung* und *Form* (nach BERTIN (1973)). Die graphischen Variablen sind in Abbildung 6 dargestellt.



**Abbildung 6: Graphische Variablen nach BERTIN (1973) aus NEUDECK (2001)**

Diese Variablen können um einen Satz von Bildschirmvariablen zur Beschreibung für Symbole der web-basierten Visualisierung im Zeitalter der digitalen Kartographie erweitert werden (LI and KRAAK (2002)). Die Bildschirmvariablen sind *blur* (Unschärfe), *focus* (Schärfe) und *Transparenz*. In diesem Zusammenhang sind auch wissenschaftliche Untersuchungen zu nennen, die sich mit der Gestaltung am digitalen Bildschirm beschäftigen (BROWN (1993), BRUNNER and NEUDECK (2002)). NEUDECK (2001) analysierte die Bildschirmvisualisierung für topografische Karten und gab Vorschläge zur Gestaltung. Für dynamische Geodaten werden diese um das Element *motion* (Bewegung) (WILKINSON (1999)) ergänzt. Für eine ausführlichere Beschreibung kann das Element *motion* mit den Attributen „direction“ (Richtung), „speed“ (Geschwindigkeit) und „acceleration“ (Beschleunigung) erweitert werden.

Zur Präsentation von Geodaten werden kartographische Gestaltungsmittel benötigt, die definieren, wie verschiedene Geobjekte dargestellt werden sollen. Kartographische Gestaltungsmittel sind neben den

Grundelementen Punkt, Linie und Fläche, zusammengesetzte Zeichen wie Signatur oder Schrift (HAKE et al. (2002)).

**Definition 2.7 (Signatur, engl. symbol)** „*Signaturen, auch Kartenzeichen oder Symbole genannt, sind abstrahierte Objektbilder oder konventionelle Zeichen die man allen graphischen Variationen unterziehen kann... .*“ (HAKE et al. (2002))

Eine Signatur besteht aus verschiedenen Signaturparametern. Diese bestehen zum einen aus graphischen Variationen wie Größe, Farbe, ... und zum anderen aus den Bildschirmvariablen und Transparenz. Die Punkt- und Liniensignaturen können mit allen Variablen beschrieben werden. Die Größe ist bei einer Liniensignatur mit dem Wert Breite definiert. Eine Flächensignatur ist über die Bildschirmvariablen, Tonwert, Textur und Farbe für die Füllung und den Attributen der Liniensignatur für die Linienbreite (Außenlinie) definiert. Das Element Textur hat zwei Attribute (Typ und Abstand). Der Typ kann vordefiniert sein (wie gestrichelt etc.) oder hat eine Referenz zum Zeichen. Der Abstand spezifiziert den Abstand zwischen der Textur.

Um Geodaten darstellen zu können ist es notwendig, ein digitales kartographisches Modell (DKM) zu erzeugen. Hierfür muß ein Regelwerk für die inhaltliche und graphische Gestaltung definiert werden (HAKE et al. (2002)). Das Portrayalmodell beschreibt in dieser Arbeit ein Modell, welches Darstellungsobjekte festlegt, indem es Geobjekte bestimmten Signaturen zuordnet.

**Definition 2.8 (Portrayalmodell, engl. portrayal model)** *Das Portrayalmodell beinhaltet eine Sammlung von Darstellungsregeln (Definition 2.9) und Signaturen.*

**Definition 2.9 (Darstellungsregel, engl. portrayal rule)** *Eine Darstellungsregel beinhaltet einen Regelausdruck mit einer Abfrageanweisung („query statement“) und eine Referenz zu einer Signatur. Die Regeln legen über eine Referenz die Signatur (Definition 2.7) für die einzelnen Objektarten (feature\_type) in Abhängigkeit ihres Objekttyps („delineation“) (Punkt, Linie, Fläche), bestimmter Attribute (attributes) und möglicher weiterer Bedingungen fest (Abbildung 7). Wenn die Abfrageanweisung erfüllt ist, kann die Darstellungsaktion mit der referenzierten Signatur durchgeführt werden. Die Priorität (priority) legt die Signaturrenderingordnung relativ zu den anderen Signaturen der Darstellung fest.*

In der Abbildung 7 ist eine allgemeine Darstellungsregel abgebildet.

**IF** (feature\_type AND/OR attributes AND delineation=TRUE) **THEN** (Symbol) **WITH** priority

Abfrageanweisung Signaturreferenz

**Abbildung 7: Allgemeine Darstellungsregel**

In der Praxis werden einzelne Darstellungsobjektarten zusammen mit ihrer Ableitung aus dem Objektmodell und der Regeln für die Gestaltung definiert und in einem Signaturenkatalog zusammengefasst.

**Definition 2.10 (Signaturenkatalog, engl. symbol catalogue)** *Der Signaturenkatalog besteht aus zwei Teilen (Darstellungsregelwerk und Signaturspezifikation). Des Weiteren werden alle Signaturen in einer*

*Signaturspezifikation (Definition 2.11) definiert.*

**Definition 2.11 (Signaturspezifikation, engl. symbol specification)** *Die Signaturspezifikation besteht aus einer Sammlung von einzelnen Signaturen.*

Die Darstellungsregeln und Signaturen werden in existierenden Signaturenkatalogen in verschiedenen Formen wie Tabellen im PDF Format beschrieben und liegen im Normalfall nicht basierend auf einem einheitlichen Portrayalmodell schematisch vor. Um die Möglichkeit aufzuzeigen, wie die Portrayalregeln nach Abbildung 7 schematisch abgebildet werden können, werden beispielhaft Signaturenkataloge im Anschluss beschrieben und analysiert.

Ferner werden Technologien gegenübergestellt, die verwendet werden, um die Signaturenkataloge technisch in einem System umzusetzen, um so neben Vorlagen für den Druck analoger Karten vor allem digitale Darstellungen von Geoinformationen zu erstellen.

### **2.2.1. Existierende Signaturenkataloge**

In diesem Abschnitt werden zwei existierende Signaturenkataloge (SK) aus dem zivilen und dem militärischem Bereich beispielhaft dargestellt.

#### ***ATKIS-Signaturenkataloge***

Als Beispiel für Signaturenkataloge in zivilen Behörden Deutschlands werden die *ATKIS-Signaturenkataloge* beschrieben. Diese werden von der AdV (Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland) veröffentlicht. Mit Hilfe dieser Signaturenkataloge werden zur Darstellung des Datenbestandes des ATKIS Basis-DLMs bzw. des DLM50-Datenbestandes verschiedene Auflösungen von digitalen Topographischen Karten abgeleitet. ATKIS-SK10 definiert die Darstellungsregeln und Symboliken für die digitale Topographische Karte 1:10 000 (DTK10), ATKIS-SK25 für die digitale Topographische Karte 1:25 000 (DTK25) (AdV (2005a)) und ATKIS-SK50 für die Digitale Topographische Karte 1:50 000 (DTK50) (AdV (2005b)). Teil A der Signaturenkataloge enthält die Regeln, nach denen die einzelnen Basis-DLM Objektarten bzw. die Kombination mehrerer Basis-DLM-Objektarten (z.B. komplexe Objekte) in Abhängigkeit ihres Objekttyps (Punkt, Linie oder Fläche), bestimmter Attribute/Attributwerte, Referenzbedingungen und/oder zu berechnender Werte signaturiert werden. Das Beispiel in Abbildung 8 zeigt einen Ausschnitt des ATKIS-SK-50 Teil A. Hier wird für die Objektart „Straße“ mit dem Objekttyp Linie („Obj-Typ L“) und verschiedenen Attributwerten wie „FKT“ (Funktion) oder „ZUS“ (Zustand) die Signaturnummer („SN“) 3070 festgelegt. Ferner müssen in einer Karte zum Beispiel Informationen wie geographische Namen in Form von Kartenschriften enthalten sein. Da diese nicht automatisch platziert werden, sind sie interaktiv als Präsentationsobjektart („POA“) zu bilden. In Abbildung 8 ist ein textförmiges Präsentationsobjekt (1001) mit Schriftinhalt („SI“) und Signaturnummer („SN“) zu definieren. Außerdem wird die Angabe der Darstellungspriorität („DP“) festgelegt.

ATKIS-Signaturenkatlog 1 : 50 000 (ATKIS-SK50), Version 1.2, Teil A: Ableitungsregeln						Objektbereich 3000 (Verkehr)	Stand: 01.02.2005	Seite 26 (von 91)						
Objektart	Obj- Typ	Für Kartensignatur relevante Attribute/Attributwerte, berechnete Werte oder Referenzen			SN	DP		Zusätzliche für Präsentationsobjekte relevante Attribute/Attributwerte oder berechnete Werte	ergibt Präsentationsobjekt					
						L	F		POA	SI	SN	DP		
											L	F		
3101_Straße	L	FKT	2301	Straßenverkehr	3070			keine	1001	[GN/ZM]	8260	35	---	
		FTR	2000	getrennt				keine						
		WDM	1305	Landesstraße, Staatsstraße										
		ZUS	1100	in Betrieb										
		KRO		keine Ref. oben zu OA 3513										
		KRO		keine Ref. oben zu OA 3517					1001	KN <sup>11</sup>	8550	35	---	

**Abbildung 8: Teil A, ATKIS-SK-50 Beispiel, ATKIS-SK-Spezifikation (ADV (2005b))**

Die Signaturnummer („SN“) bildet die Referenz zu Teil B. Teil B beschreibt alle vorkommenden Kartensignaturen, identifizierbar über die Signaturnummer. Die Beschreibung umfasst die Bedeutung, die Signaturdarstellung und ihre Farbgebung sowie Zusätze (Abbildung 9). Zusätze können zum Beispiel die Richtung der Signatur festlegen.

SN	Bedeutung	Signaturdarstellung		Farbgebung		Zusätze, Hinweise, Bemerkungen
		vergrößert (Maße in 1/100 mm)	maßstäblich	Linie / Signatur	Fläche / Decker	
3070	Landes- / Staats- straße mit getrennten Fahrbahnen			schwarz	straßen- gelb	

**Abbildung 9: Teil B, ATKIS-SK-50 Beispiel, ATKIS-SK-Spezifikation (ADV (2005b))**

Eine schematische Abbildung der Ableitungsregeln (Abbildung 8) in Darstellungsregeln nach Definition 2.9 ist möglich und ist in Abbildung 10 dargestellt.

```
Regel: IF (Objektart AND Attributwerte AND Objekttyp = TRUE) THEN (Signatur = ...) WITH
Darstellungspriorität = ...
```

**Abbildung 10: Schematische Abbildung der Ableitungsregeln nach ATKIS-SK aus Abb. 8 nach Definition 2.9**

### MIL-DTL-89045 GeoSym

Im militärischen Umfeld wird für digitale Darstellungen von Geodaten der „de facto Standard“ MIL-DTL-89045 „GeoSym“ (Geospatial Symbols for Digital Displays) verwendet (DOD (1998)). Diese wurde zur Signaturierung von VPF-Produkten (Vector Product Format) wie DNC (Digital Nautical Chart) oder Vector Map Level 0 zur Darstellung am Bildschirm konzipiert. Das Vector Product Format ist ein militärischer Standard der US Verteidigung (U.S. Department of Defense, DOD) für vektorbasierte digitale Kartenprodukte und ist im militärischen Standard „MIL-STD-2407“ beschrieben. Es wurde als Teil des DIGEST in Form des Vector Relational Format (VRF) (Annex C) angenommen (DGIWG (2000)). DIGEST FACC (Teil 4) ist ein „dictionary“ („Wörterbuch“) für Objekte, Attribute und Attributwerte, orga-



nisiert in einem standardisierten Kodierungssystem. Ein Ausschnitt aus der GeoSym ist in Abbildung 11 dargestellt.

FCode: AP030 - road, Delineation: Line, Display Priority: 8



Product	Feature Description	Symbol Assignment Expression	Orient	Source	Source ID
VML2	road - operational, hard surface, national/state	wtc=1 AND eks=28 AND ltn>=2 AND med=1 AND rst=1		MIL-STD-2402	1P030L002

**Abbildung 11: GeoSym Beispiel Vmap Level 2, GeoSym-Spezifikation (DOD (1998))**

Es ist für das Produkt VMapLevel 2 („VML2“) mit der Objektart „road“ („feature description“) und verschiedenen Attributwerten wie „wtc“ („weather conditions“) oder „eks“ („existence“) die Signaturnummer 1315 festgelegt. Mehrere Attributwerte werden mit „and“ und „or“ miteinander verbunden. „Delineation“ beschreibt die Objektart (Punkt, Linie, Fläche). „Display Priority“ legt die Darstellungspriorität (0-9) fest, wobei Symbole mit der Darstellungspriorität 0 zuerst dargestellt werden. Auch in diesem Fall ist eine schematische Abbildung der Tabelle (Abbildung 11) in Darstellungsregeln nach Definition 2.9 möglich. Die Abfrageanweisung der Darstellungsregel nach Definition 2.9 besteht aus den FACC kodierten Objektarten (F-Code) (feature\_type), den Attributwerten (attribute) und der „Delineation“.

```
Regel: IF (feature_type AND attribute AND Delineation = TRUE) THEN (Symbol = ...) WITH Priority = ...
```

**Abbildung 12: Schematische Abbildung von Abb. 11 nach Definition 2.9**

Ist die Abfrageanweisung wahr, wird auf eine Symbolreferenz mit Darstellungspriorität (0-9) verwiesen, wobei Symbole mit der Darstellungspriorität 0 zuerst dargestellt werden. Die Symboldarstellung liegen außerdem bei der GeoSym im Format WebCGM (s. 2.3.6) vor.

Mit diesen zwei Beispielen konnte gezeigt werden, dass man Signaturenkataloge schematisch nach der Definition 2.9 umsetzen kann.

### **COP und REP**

Des Weiteren sind in diesem Zusammenhang die domänenübergreifenden Konzepte *COP* (Common Operational Picture) und *REP* (Recognized Environmental Picture) zu nennen. *COP* ist ein gemeinsames operationelles Lagebild und beinhaltet somit die Basis für vernetztes Planen und Handeln aller Führungsebenen und Aufgabenbereiche. *COP* ist die Abkürzung für „Common Operational Picture“. *Common* bedeutet eine Darstellung derselben Informationen auf unterschiedlichen Computern zu ermöglichen. Es werden die dargestellten Informationen dazu genutzt, Entscheidungen bei einer bestimmten Operation zu unterstützen (*operational*). Außerdem werden die Informationen graphisch in einem Bild dargestellt (*picture*). *COP* ist definiert als „A single display of relevant information within a commander’s area of interest tailored to the user’s requirements and based on common data and information shared by more than one command“ (U.S. GOVERNMENT PRINTING OFFICE (2008)). Neben Aufklärungsinformationen

beinhaltet *COP* Umweltinformationen, die im *REP* dargestellt werden. Diese Umweltinformationen, auch als GEOMETOC-Informationen bezeichnet, beinhalten alle relevanten räumlichen (GEO), meteorologischen (MET) und ozeanographischen (OC) Daten. REP legt drei Stufen der Weitergabe von GEOMETOC-Daten fest. Dies sind Sammlung, Fusion und Publikation (TEUFERT and TRABELSI (2006)).

Es sind große Datenmengen zu verarbeiten, um ein COP zu erstellen. Hierbei entsteht das Produkt manuell, indem die Informationen aus dem Grundbestand extrahiert werden, die Darstellung festgelegt wird und es zum jeweiligen Nutzerkreis übertragen wird. Für eine bestimmte Entscheidung werden jedoch in den meisten Fällen nur Teile dieses Produktes benötigt. Aus diesem Grund sollte das COP einzelnen Nutzergruppen und seinen Bedürfnissen anpassbar sein. Der Informationsüberfluss sollte über Datenfilter begrenzt werden und eine situationsabhängige Darstellung kann zum Beispiel über Nutzereingaben (wie Nutzerinteresse) oder Highlighting von Objekten erreicht werden. Erste Vorschläge sind in MULGUND and LANDSMAN (2007) zu finden. COP und REP zeigen Ansätze für ein gemeinsames operationelles La-gebild, welches jedoch durch Informationsfilterung und Anpassung an die Darstellung an Nutzbarkeit gewinnen kann.

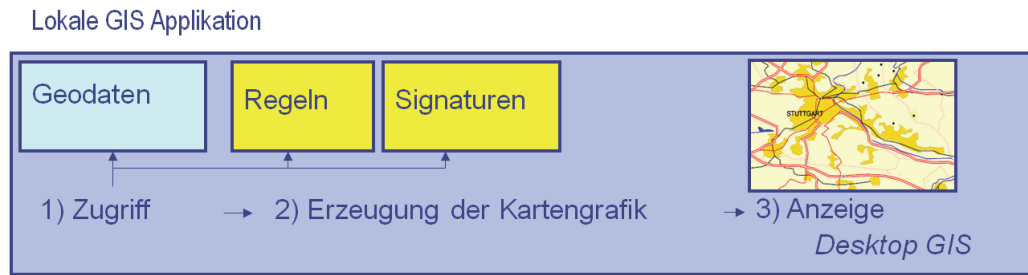
### **2.2.2. Ansätze zur Visualisierung**

Signaturenkataloge werden innerhalb eines Systems implementiert, um Geodaten zu visualisieren. Es werden im Folgenden zwei Visualisierungsansätze (Desktop-GIS und web-gestützte Lösung („Web-Mapping“)) vorgestellt, die die Darstellungsregeln und die Geodaten verarbeiten können, um die Geodaten zum Beispiel am Bildschirm darstellen zu können. In der Praxis werden die Signaturen in Signaturbibliotheken verwaltet.

#### ***Desktop-GIS gestützte Visualisierung***

BILL (1999) definiert ein GIS (Geo-Informationssystem) als „*ein rechnergestütztes System, 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 graphisch präsentiert werden.*“. Ein Desktop-GIS sind laut BILL (1999) GIS-Systeme „*die am Arbeitsplatz des Sachbearbeiters stehen*“. Beispiele für kommerzielle Desktop-GIS-Systeme sind z. B. ArcGIS von der Firma ESRI oder GeoMedia von der Firma Intergraph (HARZER (2008)). Neben den kommerziellen gibt es frei verfügbare und open-source Systeme wie Grass GIS oder gvSIG. Ein Überblick über die vorhandenen frei verfügbaren und open-source Systeme und ihre Funktionalitäten ist in STEINIGER and BOCHER (2009) zusammengefasst.

In einem Desktop-GIS werden z. B. zur Erstellung eines Geodatenproduktes Darstellungsregeln (*portrayal rules*) und Signaturen (*symbols*) verwendet, die in einem Signaturenkatalog (Kapitel 2.2.1) definiert werden. Die Darstellungsregeln sind hierbei direkt im GIS implementiert und die Signaturen werden in einer herstellereigenen Signaturbibliothek verwaltet. Die Regeln werden zusammen mit den Geodaten in einem lokalen GIS ausgeführt und abhängig davon die entsprechenden Signaturen ausgewählt und eine graphische Repräsentation erzeugt, die z. B. als digitales Kartenprodukt im GIS am Bildschirm angezeigt wird (Abbildung 13).



**Abbildung 13: Visualisierung von Geodaten in einem Desktop-GIS**

Als Beispiel wie die Visualisierung von Geoinformationen in einem GIS realisiert werden kann, werden im Folgenden die neuen Funktionalitäten in ESRI ArcGIS 9.2 dargestellt. In früheren Versionen wurde die Darstellung der Geodaten in einer mxd-Datei (ArcGIS-Projektdatei) gespeichert. Seit Version 9.2 sind kartographische Repräsentationen („Cartographic Representations“) erlaubt. Diese ermöglichen verschiedene Karten aus einer Geodatenbasis produzieren zu können. Die Daten werden unter Nutzung einer flexiblen, regelbasierten Struktur symbolisiert. Sie werden als zusätzliche Informationen zu einem Objekt oder einer Objektklasse verwendet. Zwei Spalten werden zum Objekt und somit an die „feature class“-Tabelle angefügt:

- RuleID Spalte: Die Repräsentationsregel wird in einer zusätzlichen Spalte für jedes Objekt bzw. Objektklasse als Integerwert referenziert. Die Regel an sich wird in „geodatabase“-Systemtabellen gespeichert.
- Override Spalte: Hier werden die objektspezifischen „overrides“ zu Repräsentationsregeln gespeichert. „Overrides“ sind eine Erweiterung zur Repräsentationsregel und definieren Ausnahmen, so dass Objekte verschieden dargestellt werden, die dieselbe Regel teilen. Sie ändern die Eigenschaften einzelner Objekte und können zum Highlighting von bestimmten Objekten genutzt werden. Sie werden in binary large object (BLOB) gespeichert.

Ein Beispiel einer „feature class“-Tabelle ist im Folgenden (Abbildung 14) dargestellt.

Objektart	RuleID	Override
Autobahn	1	BLOB
Hauptstrasse	2	BLOB

**Abbildung 14: „Feature Class“-Tabelle mit den zusätzlichen Spalten „RuleID“ und „Override“**

Der Workflow der kartographischen Repräsentation ist in Abbildung 15 abgebildet. Es wird je nach Objekt und deren Attributen eine bestimmte Regel verwendet. Diese beinhaltet die Signaturen des Objektes wie hier das rote Kreuz. Im Anschluss können eventuelle „Overrides“ durchgeführt werden. „Overrides“

können die Eigenschaften (wie Größe, Farbe, ...), der Gestalt oder der Position (unabhängig von Objektgeometrie) der Objekte verändern. Im Beispiel in Abbildung 15 wird das rote Kreuz mit einem Versatz zur ursprünglichen Position (schwarzer Punkt) dargestellt. Außerdem ist zu erkennen, dass sich die Definitionen von der Definition 2.9 unterscheiden.

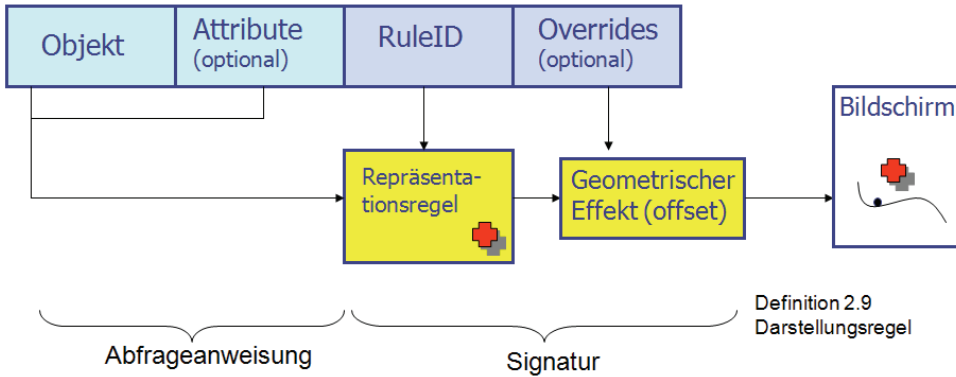
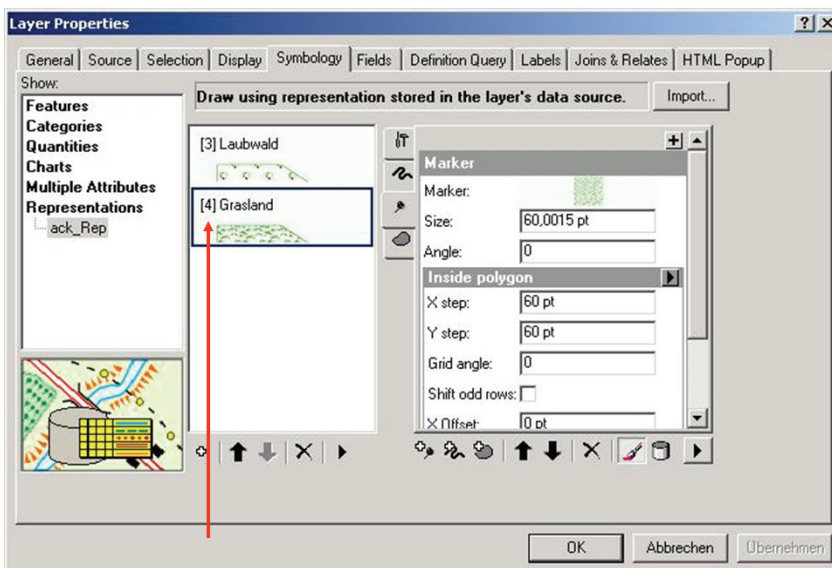


Abbildung 15: Kartographische Repräsentation - Workflow in ArcGIS 9.2

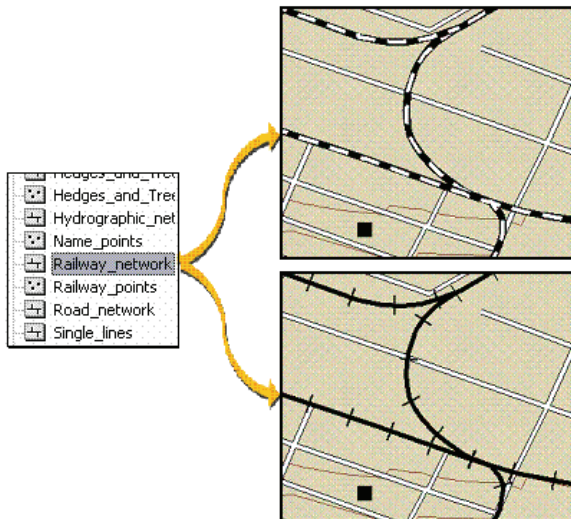
Abbildung 16 zeigt die Layereigenschaften mit der kartographischen Repräsentation für Laubwald und Grasland.



RuleID

Abbildung 16: Erstellung der kartographischen Repräsentation in ArcGIS 9.2

Bei multiplen Repräsentationen wird dasselbe Phänomen unter verschiedenen Aspekten in einem System unterschiedlich präsentiert. Multiple Repräsentationen können mit einzelnen Objektklassen assoziiert sein, so dass dieselben Daten unterschiedlich präsentiert werden können (Abbildung 17).



**Abbildung 17: Multiple Repräsentationen derselben Phänomene (REPRESENTATIONS (2009))**

In ArcGIS 9.2 werden die Repräsentationsregeln getrennt von den Daten gespeichert bzw. verwaltet und somit ist eine verteilte Architektur möglich. Die Lösung ist proprietär, welches ein Nachteil ist, wenn Repräsentationen zum Beispiel zwischen GIS Systemen ausgetauscht werden sollen. Außerdem sind die „overrides“ in BLOBs gespeichert und sind somit nicht abfragbar. Ferner unterscheidet sich die Definition in 2.9 von dem Konzept einer Repräsentationsregeln in ArcGIS 9.2 (Abbildung 15).

Ein weiterer seit den 90er Jahren existierender Ansatz um Geodaten zu visualisieren, ist das „Web-Mapping“ und die dienstbasierte Visualisierung, worauf im Weiteren eingegangen wird.

### **Web gestützte Visualisierung (Web-Mapping) und dienstbasierte Visualisierung**

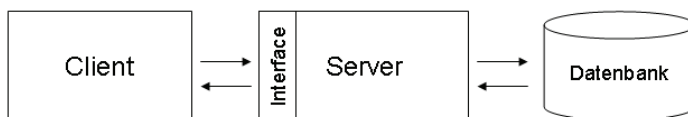
„Web mapping is the process of designing, implementing, generating and delivering maps on the World Wide Web.“ (NEUMANN (2008)). Der „Web-Mapping“-Prozess basiert auf Web-Service Architekturen. 1993 wurde der erste „Mapserver“ auf Basis von CGI/Perl entwickelt, der eine Ausgabe der Karten in GIF unterstützt (Xerox Parc Map Viewer) (PUTZ (1994)). Ein Jahr später wurde der Online Atlas (National Atlas of Canada) freigegeben (KRAMERS (2006)). Der Adresssuch- und Routendienst auf der Seite „Mapquest.com“ wurde im Jahre 1996 online gestellt. Nach Übernahme der Firma Argus veröffentlichte 1996 Autodesk „MapGuide 2.0“. In diesem Jahr wurde auch die erste GeoMedia „Web Map Version 1.0“ von Intergraph ausgeliefert, die das Vektorformat ActiveCGM unterstützt. Ein Jahr später entwickelte NASA den „UMN MapServer 1.0“, um Fernerkundungsdaten über das Web anbieten zu können. Die Version des „UMN MapServer 3.6“ (2002) beinhaltet auch eine WMS-Unterstützung (WEBMAPPING (2009)). Im Jahr 1998 stieg ESRI mit dem „MapObjects Internet Map Server“ in die web gestützte Visualisierung ein. Durch Google bekam 2005 der Nutzer die Möglichkeit auf eigenen Webseiten ohne die Verwendung eines eigenen Servers Google Kartendienste einbinden zu können. Seit 2005 ist der Trend in Richtung Web 2.0 zu spüren. Der Begriff Web 2.0 wurde Ende des Jahres 2004 durch Mitarbeiter des Verlags O'Reilly eingeführt (O'REILLY (2005)), um damit auf einen neuen Trend im World Wide Web hinzuweisen. Bei einem Web 2.0 Projekt können Nutzer auf der ganzen Welt Informationen über gemeinsame Werkzeuge beitragen. Im Bereich des Web Mapping ist zum Beispiel „Wikimapia“

(<http://wikimapia.org>) zu nennen, welches 2006 freigeschaltet wurde. Hier werden „GoogleMaps“ und „Wikipedia“ miteinander kombiniert. In Kapitel 3.2 werden Beispiele für Web 2.0 Anwendungen aufgezeigt. Immer mehr Geodaten werden über sogenannte Geodienste bereitgestellt und visualisiert. Der Geodienst ist wie folgt definiert.

**Definition 2.12 (Geodienst)** „Ein Geodienst stellt über Schnittstellen Funktionalität für die Nutzung von Geodaten bereit. Die Nutzung von Geodaten umfasst den Zugriff auf sowie die Erfassung, Manipulation, Transformation, Analyse und die Präsentation von Geodaten.“ (DONAUBAUER (2004))

**Definition 2.13 (Geo Web Service)** „Ein Geodienst, der seine Funktionalitäten über das Internet bereitstellt ist ein Geo Web Service“. (DONAUBAUER (2004))

Bei einem Geodienst startet der Nutzer per Client eine Anfrage („Request“) an einen bestimmten Dienst wie Geodatendienst oder Kartendienst im Internet. Dieser wertet die Anfrage aus, greift auf die entsprechenden Daten zu und gibt diese an den Client zurück (Abbildung 18). Ausgabemedien sind bei einem Kartendienst vor allem digitale Darstellungen am Bildschirm. Jedoch können auch Vorlagen für gedruckte Karten erzeugt werden.



**Abbildung 18: Datenbankdienste nach BARTELME (2005)**

Für die Verwendung von Geodiensten werden verschiedene client- oder serverseitige Technologien verwendet. Die clientseitige Programmierung für dynamische Seiten bietet eine direkte Interaktion mit dem Benutzer. Kerntechniken sind hier u. a. *Java Applets*, *JavaScript*, *Flash* und *AJAX* (Asynchronous JavaScript and XML). AJAX ist keine eigene Technologie, sondern nutzt verschiedene Technologien: XML, JavaScript, HTTP und XHTML. AJAX ist ein Entwicklungsmodell für browserbasierte Webanwendungen. AJAX nutzt zum Nachrichtenaustausch im Gegensatz zu Web Services HTTP basiertes „XMLHttpRequest Protocol“ und hat den Vorteil, dass die Daten manipuliert werden können ohne noch einmal die gesamte Seite im Web Browser rendern zu müssen. Ein Beispiel für eine AJAX-basierte Anwendung ist Google Maps. Wesentlich komplexere Programmierungen für dynamische Seiten finden serverseitig statt, insbesondere weil hier Datenbankanbindungen genutzt werden können. Techniken sind hier zum Beispiel *CGI* (Common Gateway Interface), *PHP* (PHP: Hypertext Preprocessor), *ASP* (Active Server Pages), *Servlets* (Java-Pendant zu CGI oder ASP) und *JSP* (Java Server Pages). Bei Servlets ist das HTML in Java-Code eingebettet und bei JSP umgekehrt. Details zu den Technologien sind in WALTER (2008) zu finden. Grundlegende Techniken für webgestützte GIS Architekturen sind in LEUKERT and REINHARDT (2000) beschrieben.

Ein spezieller Geowebdienst ist der Portrayalwebdienst. Er wird wie folgt definiert.

**Definition 2.14 (Portrayalwebdienst)** Ein Portrayalwebdienst ist ein Geo Web Services der den Portrayalprozess durchführt. Er bereitet die Geoinformationen grafisch auf, um zum Beispiel eine Anzei-



ge am Bildschirm zu ermöglichen.

Im Folgenden wird die Funktionalität eines Portrayalwebdienstes beschrieben, der eine Grafik erzeugt. Die grafische Aufbereitung von Geoinformationen erfordert zwei Hauptkomponenten: Geodaten und die Umsetzung eines Signaturenkatalogs (Regeln und Signaturen) (Abbildung 19). Hierbei wird zuerst ein bestimmter Geodatenausschnitt mit der Darstellung nach einem bestimmten Signaturenkatalog als Grafik am Server angefragt (Request). Es werden die Geodaten und die Darstellungsregeln und ihre referenzierten Signaturen vom Portrayalwebdienst selektiert (1 Zugriff). Anschließend werden die Darstellungsregeln ausgeführt, um so die Signaturen für eine Geodatenauswahl zu erhalten. Anschließend erzeugt der Dienst die Grafik (2 Erzeugung der Kartengrafik) und sendet die Antwort („Response“) an den Client zurück, der diese anzeigt (3). Die Technologie wird detailliert im Kapitel 2.3.3 erläutert.

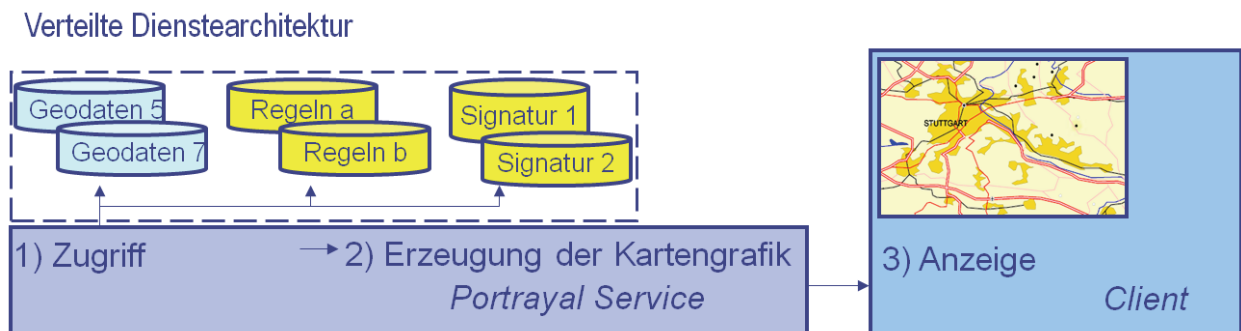


Abbildung 19: Visualisierung von Geodaten mit Hilfe eines Portrayalwebdienstes

Der Vorteil einer standardisierten verteilten Dienstarchitektur gegenüber einem Desktop-GIS ist, dass die Geodaten, Signaturen und Regeln verteilt vorliegen und durch den standardisierten Zugriff von jedermann mit Zugang zum Intranet/Internet genutzt werden können. Jeder Nutzer benötigt nur einen Browser (Client) und es muss kein Desktop-GIS System lokal vorhanden sein. Auf Basis der zuvor beschriebenen Prozesse zur Visualisierung von Geoinformationen, werden im Anschluss Normen und Standards in diesem Bereich beschrieben.

### 2.3. Standards und Normen im Bereich Portrayal und Portrayalwebdienste

Das Ziel der Arbeit ist die Verwendung eines Portrayalwebdienstes zur kontextabhängigen Darstellung von Geoinformationen am Bildschirm. Der Zugriff auf einen Portrayalwebdienst muss über standardisierte Operationen definiert werden. Zur Visualisierung von Geoinformationen werden Regeln vom Portrayalwebdienst überprüft, um eine entsprechende Signaturierung zu erhalten. Hierfür muss eine standardisierte Kodierung für die Darstellungsregeln definiert werden, um einen Austausch von Regeln zu ermöglichen und diese systemübergreifend verwenden zu können. Für diese Anforderungen sind verschiedene Normen und Standards notwendig. Als Basis für die Arbeit werden ISO Normen und OGC Standards berücksichtigt. Die bisherigen Aktivitäten der Standardisierungsgremien im Bereich „Portrayal“ werden aus diesem Grund im Folgenden beschrieben.

Das OGC definierte 1995 seine eigene „OpenGIS architecture“. 1999 startete die Web Mapping Testbed

Phase 1 (WMT1) und somit erste Prototypen, die sich mit der Standardisierung von Web Map Services beschäftigten (OGCHISTORY (2009)). 2000 wurde die „WebMap Server Interface Specification“ verabschiedet. Im Jahre 2004 übernahm die ISO diese Spezifikation als Norm (ISO (2005b)). 2002 entstand die erste OGC SLD-Version, die sich mit der Umsetzung von Darstellungsregeln und Signaturen in XML beschäftigten.

Es gibt viele verschiedene Normen und Standards aus ISO und OGC im Bereich „Portrayal“ und „Portrayalwebdiensten“. Die einzelnen relevanten Bereiche sind im Folgenden aufgelistet:

- Portrayalmodell (Regeln und Signaturen, Informationsmodell)
- Regeln und Signaturen (XML Implementierungsspezifikation)
- Portrayalwebdienste (Standardisierte Anfrageoperationen und deren Parameter, Antwortdefinitionen des Dienstes)
- Verwaltung der Regeln und Signaturen und deren dienstbasierter Zugriff
- Speicherung von Metainformationen für einen Portrayalwebdienst
- Digitale Ausgabeformate

In der nachstehenden Tabelle 1 sind die Normen und Standards im Bereich Portrayal zusammengefasst.

**Tabelle 1: Standards und Normen im Bereich Portrayal (Stand 02/10)**

Art	Referenznummer	Name	Version	Datum
Portrayalmodell	ISO 19117	Portrayal – ISO Norm	ISO 19117: 2004	2004-01-06
	ISO 19117 Rev	ISO/CD 19117: Portrayal	ISO 19117: 2009	2009-12-03
Regeln und Signaturen	OGC 05-077r4	Symbology Encoding Implementation Specification – Standard	1.1.0 (revision 4)	2006-07-21
Portrayalwebdienst (Basis)	ISO 19117	Portrayal – ISO Norm	ISO 19117: 2004	2004-01-06
	OGC 06-042	OpenGIS Web Map Server Implementation Specification – Standard	1.3.0	2006-03-15
	ISO 19128	WMS – ISO Norm	ISO 19128: 2005	2005-01-25
Portrayalwebdienst (erweitert, mit Möglichkeit zur Verwendung von Darstellungsregeln und Signaturen)	OGC 05-078r4	Styled Layer Descriptor Profile of the Web Map Service: Implementation Specification – Standard	1.1.0 (revision 4) (erweitert WMS Version 1.3)	2007-06-29
Verwaltung – Katalogdienst	OGC 07-006r1	OGC™ Catalogue Services Specification –	2.0.2 with Corri-	2007-02-23



		Standard	gendum 2	
Verwaltung - Signaturmanagementdienst	OGC 04-040	Style Management Services for Emergency Mapping Symbology – Diskussionspapier	1.0	2005-02-14
Verwaltung - Signatur- und Regelmanagement	OGC 05-112r1	Symbology Management-Diskussionspapier	0.3	2006-07-26
Verwaltung – Registerdienst	OGC 07-110r4	CSW-ebRIM Registry Service - Part 1: ebRIM profile of CSW - Standard	1.0.1	2009-02-05
Verwaltung - Registerdienst	OGC 07-144r4	CSW-ebRIM Registry Service - Part 2: Basic Extension Package – Standard	1.0.0	2009-02-05
Verwaltung - Registerinformationsmodell	ISO 15000-3	ebXML Registry Information Model – ISO Norm	ISO/TS 15000-3:2004	2004-05-14
Verwaltung – Registerdienst	ISO 15000-4	ebXML Registry Services Specification – ISO Norm	ISO/TS 15000-4:2004	2004-05-14
Speicherung von Metainformationen für einen Portrayalwebdienst (Web Map Context)	OGC 05-005	Web Map Context	1.1.0	2005-01-19
Digitale Ausgabeformate	-	Rasterformate wie JPEG Vektorformate wie SVG	-	-

Die verschiedenen Arten aus Tabelle 1 werden in den folgenden Kapiteln erläutert.

### 2.3.1. ISO Portrayalmodell (Informationsmodell)

In ISO 19117 – Portrayal (ISO (2005a)) wird eine Methodik zur Präsentation von Objekten eines Datensatzes über Darstellungsregeln beschrieben. Durch den Standard sollen alle Signaturenkataloge einheitlich verwendet werden können. Es beinhaltet keine Standardisierung von kartographischen Signaturen.

Die Norm definiert folgende Komponenten, die notwendig sind um Geodaten in einer Form zu präsentieren, die von Menschen verstanden werden kann:

- Portrayal Catalogue (Portrayalkatalog, welcher die Darstellungsregeln beinhaltet)
- Portrayal Specification (Portrayalspezifikation, Sammlung Operationen inkl. Signaturen zur Darstellung einer Objektinstanz)
- Portrayal Service (Portrayalwebdienst, Dienst zur grafischen Aufbereitung)

Der *Portrayal Catalogue* und die *Portrayal Specification* werden benötigt, um den Signaturenkatalog standardisiert abbilden zu können. Der *Portrayal Catalogue* beinhaltet die Darstellungsregeln, die eine Referenz zur *Portrayal Specification* besitzen. Der *Portrayal Service* entspricht der Definition 2.14.

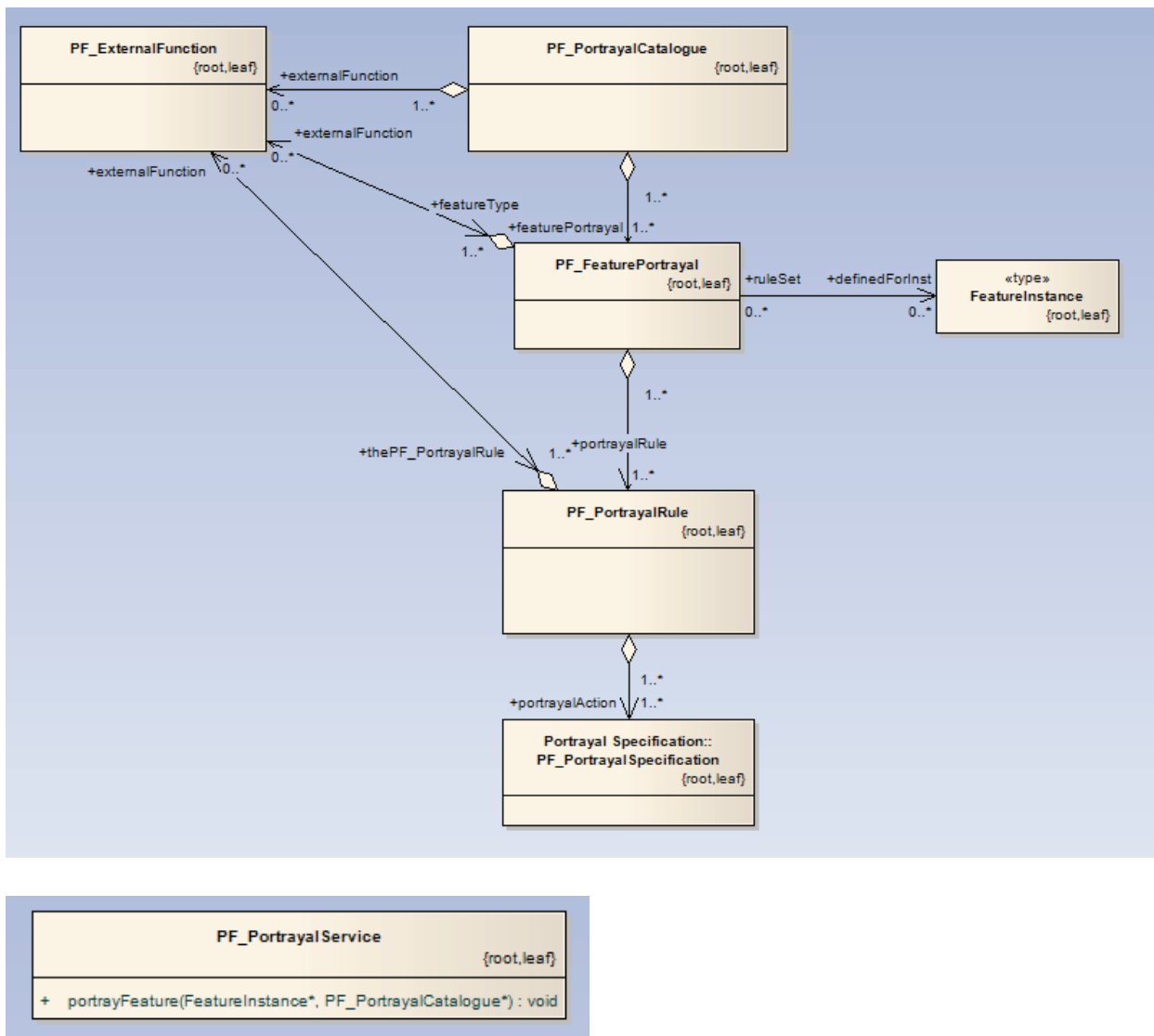


Abbildung 20: Überblick der ISO 19117 – Portrayal, modifiziert

Abbildung 20 gibt einen Überblick über die Komponenten der ISO 19117 – Portrayal (ISO (2005a)). Die *Portrayal Catalogue* Klasse beinhaltet einen Satz von *Feature Portrayal* Objekten und optional *External Function* Objekten. *Feature Portrayal* ist eine Sammlung von *Portrayal rules* (Darstellungsregeln) zur Objektpräsentation. Die korrespondierenden Objekte (*Feature Instance*) werden abhängig von ihren Darstellungsregeln symbolisiert. Die Darstellungsregeln testen die Attribute von Objektinstanzen in einem Datensatz und geben wahr oder falsch zurück. Wird wahr zurückgegeben, wird die *Portrayal specification* die durch die Darstellungsregel referenziert ist, verwendet. Die *Portrayal specification* definiert eine Sammlung von Operationen, um das Erstellen der Kartengrafik zu ermöglichen. *External functions* sind Funktionen, die nicht Teil des Anwendungsschemas der darzustellenden Geodaten sind (wie Rotation einer Karte über die aktuelle Position des Autos bei Autonavigation). Der *Portrayaldienst* ist ein Dienst zur

Präsentation der Objektinstanzen mithilfe der Operation „portrayFeature“. Diese Operation referenziert eine oder mehrere Objektinstanzen und einen Portrayalkatalog. Genauere Definitionen von Operationen des Dienstes und deren Parametern sind nicht Inhalt dieser Norm.

Anhand eines Beispiels soll im Folgenden die Methodik der ISO 19117 – Portrayal (ISO (2005a)) zur Präsentation von Objekten eines Datensatzes beschrieben werden. Die Darstellungsregeln werden im Portrayalkatalog gespeichert. Sie bestehen aus einem „Query statement“ (Abfrageanweisungen) und mehreren „action statements“ (Operation wie „Zeichne eine Kurve“ - „drawCurve“). In Abbildung 21 wird ein Beispiel für eine Darstellungsregel gegeben. Der Datensatz beinhaltet mehrere Instanzen der Objektklasse „Straße“. Die Objektklasse beinhaltet ein Attribut „Klassifikation“, mit den Werten „Landstraße“ oder „Autobahn“. Die *Portrayal specification* heißt TK50\_Spez. Die Regeln lauten wie folgt:

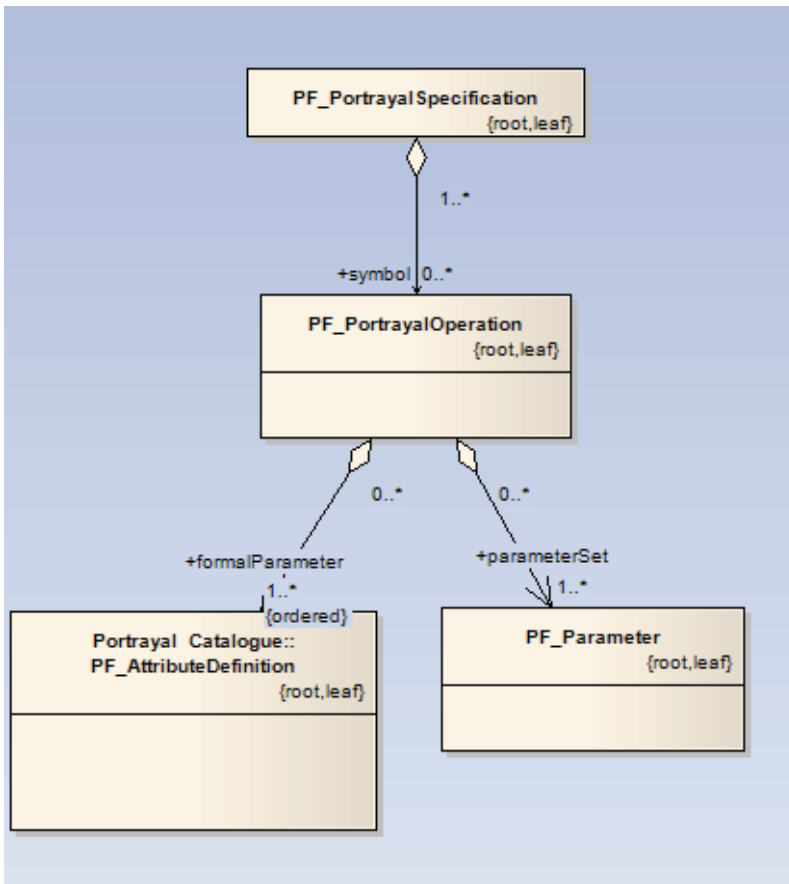
```
IF (Strasse.Klassifikation = "Autobahn") THEN drawCurve("TK50_Spez.Rote_Linie",Strassensegment)
IF (Strasse.Klassifikation = "Landstraße") THEN drawCurve("TK50_Spez.Gelbe_Linie",Strassensegment)
```

Abfrageanweisung
Operation
Portrayalspezifikation
Geometrie

**Abbildung 21: Regeln nach ISO 19117 - Portrayal**

„Then“ trennt die Bedingung von der Aktion. Die Operation „drawCurve“ zeichnet eine Kurve mit der Geometrie von Straßensegment und der Farbe, Strichstärke, ... definiert in der *Portrayal specification* TK50\_Spez.

Die *Portrayal specification* definiert eine Sammlung von Operationen (*Portrayal Operation*) wie „drawCurve“, um das Erstellen der Kartengrafik zu ermöglichen und beinhaltet eine Methodik die Symbolattribute definiert. Die Operationen vereinbaren einen Satz von *formal parameters*, welche für den Renderingprozess der Grafik benötigt werden (Abbildung 22). Die Signaturattribute der Parameter für eine Linie sind hierbei beispielsweise Farbe und deren Typ (RGB-Wert). Die *formal parameters* sind mit den *parameter sets* assoziiert. Es werden Werte für die Parameter in den *parameter sets* festgelegt, wie z. B. in Abbildung 22 für Rote\_Linie die Farbe Rot (RGB-Wert „255 0 0“).



```

Operation
{
name="drawCurve"
formal_parameters=
{
ad21: AttributeDefinition (name="color", type="Sequence(Int)")
...
}}

ParameterSet
{
label="Rote_Linie"
actual_parameters=
{
AttributeValue (value="255 0 0",attribute_type="ad21")
...
}}

ParameterSet
{
label="Gelbe_Linie"
actual_parameters=
{
AttributeValue (value="255 255 0",attribute_type="ad21")
...
}}
  
```

Abbildung 22: Portrayal Specification (vereinfacht) und Beispiel

Die Darstellungsregeln haben eine Relation zu einer oder mehreren *Portrayal specifications*. Dies macht es möglich, denselben Datensatz mit unterschiedlichen Signaturenkatalogen auch verschieden darzustellen ohne den Datensatz selbst zu verändern.

Aber der Portrayalstandard hat Grenzen wie:

- Das Portrayalkonzept ist sehr abstrakt und unzureichend definiert. Es ist eine genauere Definition des Portrayalkonzeptes notwendig.
- Keine klare Struktur oder konzeptionelles Modell für die Portrayalregeln. Der Standard enthält Beispiele, jedoch gibt es keine Definition der Struktur. Er beschreibt nicht die Komposition von Portrayalregeln. Es ist eine Überarbeitung des konzeptionellen Modells für Darstellungsregeln notwendig.
- Kein Modell für die Signaturen. Im Portrayalstandard werden *formal parameters* definiert, die die Signaturparameter (wie Farbe, Strichstärke, ...) festlegen. Jedoch ist eine Spezifizierung der Signaturparameter für die einzelnen Objektarten (Punkt, Linie, Fläche) in einem Signaturmodell sinnvoll.

Aus diesem Grund schlug die DGIWG eine Revision der ISO 19117 – Portrayal (ISO (2009b)) vor. Der Autor partizipiert in DGIWG. Die Revision ist zurzeit im Stadium CD (committee draft) und geht über den existierenden Standard im Bereich Portrayalregeln und Signaturen hinaus.

Der definierte Portrayalprozess verwendet ein Portrayalmodell bestehend aus Portrayalfunktionen (*Portrayal function*) und Signaturdefinitionen (*Symbol definition*) (Abbildung 23). Die Portrayalfunktion definiert ein „Mapping“ (Zuordnung) der Geoobjekte (*Feature Data*) zu den Signaturen mithilfe der portrayal engine. Hierbei assoziiert ein „pointer“ (symbol reference) in der Portrayalfunktion die Geoobjekte mit einer spezifischen Signatur, welche in der „symbol definition“ definiert sind. Das „Mapping“ kann auf verschiedene Weise definiert werden. Eine Möglichkeit ist die Definition von Darstellungsregeln. Sie sollen gegen die Attributwerte der Objektinstanzen im Datensatz getestet werden. Diese sollte ein „true“ oder „false“ zurückgeben. Wenn die Abfrageanweisung „true“ (wahr) zurückgibt, soll die entsprechende „symbol specification“ verwendet werden.

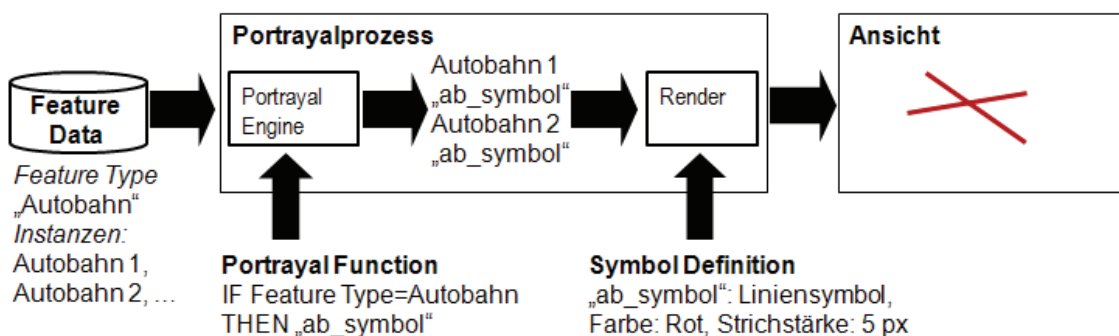


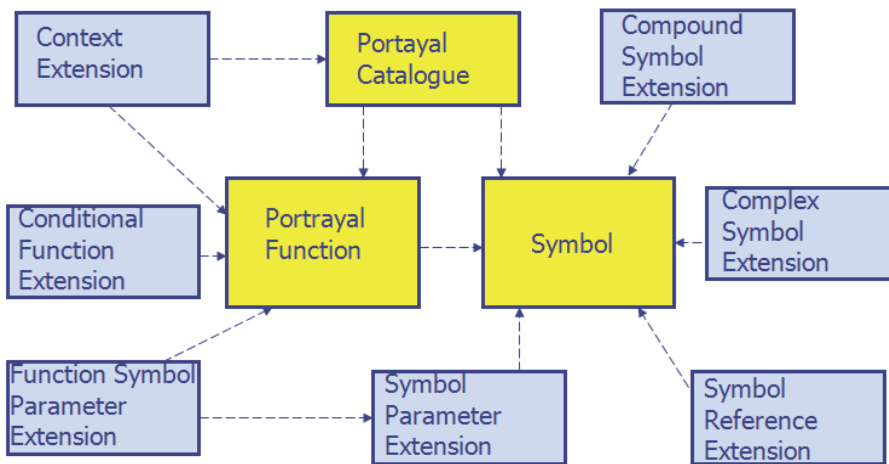
Abbildung 23: Portrayalprozess nach CD ISO 19117 – Portrayal (ISO (2009b))

Der Renderingprozess erzeugt auf Basis der „symbol definition“ eine Darstellung des Geoobjektes. In

Abbildung 23 ist der Portrayalprozess anhand eines Beispiels dargestellt. Es wird mithilfe der portrayal engine die Objektinstanz „Autobahn 1“ der Signatur „ab\_symbol“ zugeordnet. Anschließend über den Renderingprozess die Signatur „ab\_symbol“ auf das Objekt „Autobahn 1“ angewendet.

Das Portrayalmodell wird in der CD ISO 19117 – Portrayal (ISO (2009b)) mithilfe eines „Core and extension models“ definiert (Abbildung 24). Das Basismodell (Core model) besteht aus den zuvor beschriebenen Portrayalfunktionen und Signaturen.

Der Portrayal Catalogue beinhaltet die „Portrayal function“ und „symbols“. Die Signaturierung (*symbols*) wird über die Elemente Punkt-, Linien-, Flächen- und Textsymbole und deren graphischen Variablen (Stil, Füllung, ...) definiert.



**Abbildung 24: „Core (gelb) and Extension (blau) Packages“ mit Abhängigkeiten nach ISO 19117 – Portrayal (ISO (2009b))**

Das „extension model“ (Erweiterungsmodell) besteht aus verschiedenen Erweiterungen für die Basismodelle. Mithilfe der „conditional function“ können neben der Objektart auch Objektattribute, -geometrie, ... einbezogen werden. Des Weiteren kann der Kontext als Erweiterung des Kernmodells berücksichtigt werden. Kontextwerte können hierbei mit der Erweiterung „context extension“ in die „portrayal functions“ mithilfe einer Parameterliste (PN\_PortrayalFunctionContextualParameterList) verwendet werden. Mit der Parameterliste sollen Kontextelemente als Parameter wie zum Beispiel die Attributwerte der Geoobjekte im Kernmodell in die „portrayal functions“ integrierbar sein. Jedoch gibt die Definition nur einen kleinen Rahmen vor. Die genaue Umsetzung in einem System oder die Definition einzelner Kontextelemente fehlt und muss detaillierter in der Arbeit definiert werden.

Die Erweiterung des „symbol“ Packages definiert verschiedene Komponenten („compound symbols“) und komplexe Symbole wie Punktsymbole über eine Linie oder schraffierte Flächen („complex symbols“). Ein weiteres Erweiterungsmodell des „symbols“ beinhaltet ein Konzept zur Nutzung externer Symbolkomponenten („symbol reference“). Eine weitere Möglichkeit ist die Nutzung von Symbolparametern mit der Erweiterung „symbol parameter“ und „function symbol parameter“. Die Erweiterung „symbol parameter“ erlaubt die Verwendung verschiedener Symbolparameter wie Farbe, ..., wohingegen die „function symbol parameter“ Erweiterung es ermöglicht, Attributinformationen von Objekten an

Symbole zu übermitteln.

Die ISO 19117 – Portrayal (ISO (2009b)) definiert im Gegensatz zur ursprünglichen Norm keinen Portrayalwebedienst, da dieser in einer separaten Norm spezifiziert werden sollte. Des Weiteren sind auch keine Regeln zur Fertigstellung einer kartenähnlichen Darstellung wie Verdrängung oder Generalisierung definiert. Ferner definiert die Norm nur 2D- und keine 3D-Signaturmodelle.

### 2.3.2. OGC Regeln und Signaturen

Um die Darstellungsregeln und Signaturen in einer servicebasierten Webarchitektur verwenden zu können, müssen sie maschinenlesbar sein. Eine maschinenlesbare und für Webservices geeignete Sprache ist XML. OGC Symbology Encoding (SE) (OGC (2006a)) definiert eine XML Implementierungsspezifikation für Darstellungsregeln und Signaturen. Das Ziel vom „Symbology Encoding“ ist es, das Styling einer Objektart und einzelner Attribute von anderen relevanten Informationen isoliert zu beschreiben. Diese Styles werden auf spezifische Objekttypen oder Coveragetypen unabhängig vom Dienst angewendet. Die Elemente definieren die Regeln in Form von Filtern (basiert auf ISO 19143 – Filter Encoding (ISO (2010)) und Signaturen „Symbolizer“ (Punkt-, Linien-, Polygon-, Text- und Rastersymbole). Die Symbolizer werden in die Regeln eingebettet. Die Regeln werden verwendet, um Renderinginstruktionen abhängig von Objekteigenschaftsbedingungen und weiteren Eigenschaften wie Kartenmaßstab durchzuführen. Ein Beispiel ist in Abbildung 25 dargestellt. Der „PolygonSymbolizer“ beschreibt ein Polygon mit schwarzem Rand und roter Füllung. Dieser „PolygonSymbolizer“ wird angewendet, wenn das Objekt „Haus“ das Attribut A=1 (Filter) besitzt.

```

<FeatureTypeStyle version="1.1.0">
  <FeatureTypeName>Haus</FeatureTypeName>           Abfrageanweisung („feature type“)
  <Rule>
    <Filter>(A=1)</Filter>                             Abfrageanweisung („attributes“)
    <PolygonSymbolizer>
      <Fill> <SvgParameter name="fill">#red</SvgParameter> </Fill>
      <Stroke> <SvgParameter name="stroke">#black</SvgParameter> </Stroke> } Signaturdefinition
    </PolygonSymbolizer>
  </Rule>
  ...
  <FeatureTypeName>Strassen</FeatureTypeName>
  ...
</FeatureTypeStyle>

```

Abbildung 25: „FeatureTypeStyle“-Element im Symbology Encoding

Es werden in dieser Spezifikation nicht alle Eigenschaften, die in der CD ISO 19117 – Portrayal definiert sind unterstützt. Es gibt keine Trennung von Regeln und Signaturen. Des Weiteren werden keine komplexen Signaturen unterstützt. Eine Zusammenfassung mehrerer Symbole zu einer Gruppe ist ebenfalls nicht möglich. Auch die Verwendung von „formal parameters“, um Signaturparameter zu verwenden wird nicht unterstützt. Ferner sollten laut CD ISO 19117 – Portrayal Signaturreferenzen unterstützt werden. Um diese Mängel zu beheben, arbeitet die OGC zurzeit an einer neuen SE Version.

### 2.3.3. Portrayalwebedienst

Ziel dieser Arbeit ist es, eine kontextbasierte Visualisierung auf Basis von standardisierten Web-Dienste-

Architekturen zu ermöglichen. Folglich müssen die vorhandenen Web-Dienste-Architekturkonzepte der Standardisierungsgremien wie ISO und OGC analysiert werden. Portrayalwebdienste basieren auf dem Konzept der Geodienste. Aus diesem Grund werden in diesem Kapitel die ISO und OGC Referenzmodelle für service-orientierte Architekturen allgemein und der Portrayalwebdienst speziell vorgestellt.

### *Portrayalwebdienste allgemein und OGC portrayal pipeline*

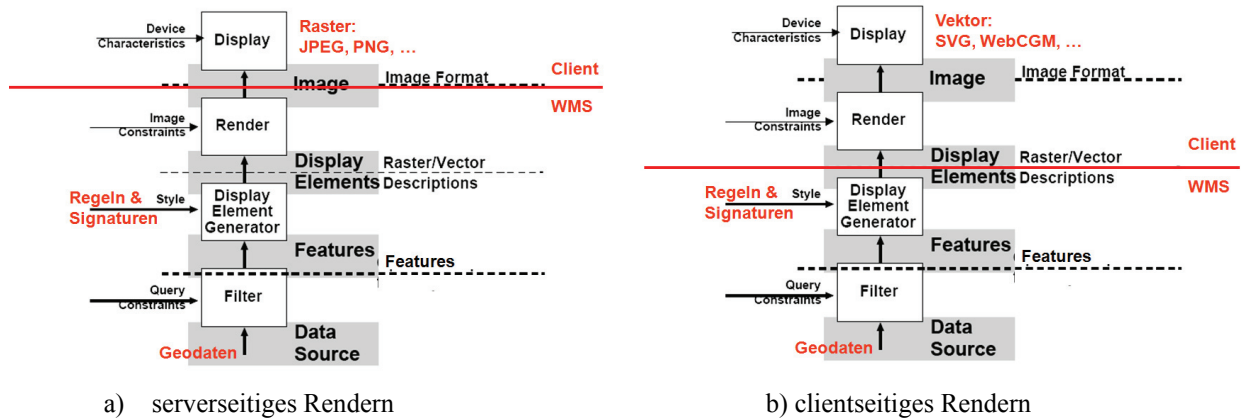
Grundlage der Portrayalwebdienste bilden die Konzepte der allgemeinen Geodienste beschrieben im vorherigen Abschnitt. Der Zusammenhang zwischen den allgemeinen ISO Geodiensten und den ISO Portrayaldiensten ist in Tabelle 2 dargestellt. Die sechs generellen Klassen von „Information Technology Services“ wurden in Kapitel 2.1.3 definiert.

**Tabelle 2: Gegenüberstellung Geodienste allgemein und Portrayaldienste**

Information Technology Services nach ISO 19101– Reference model document (ISO (2000))	Relevante ISO Standards im Bereich Portrayaldienste
Geographic human interaction services	ISO 19128 – Web map server interface (ISO (2005b)): „geographic viewer (client service)“: Dieser „Viewer“ erlaubt dem Nutzer mit den Kartendaten zu interagieren, wie Abfrage von Visualisierung oder „Querying“.
Geographic model/information management services	ISO 19128 – Web map server interface ((ISO (2005b)): map access service  ISO 19117 – Portrayal (ISO (2005a)): Portrayal Service - Visualisiert die Objektinstanz(en) und führt die Operationen aus, die in der Portrayalspezifikation definiert sind.

Nach dem Konzept des OGC ist es möglich, verteilte standardisierte Geowebdienste zu definieren und damit Interoperabilität zu gewährleisten. Die allgemeine „Portrayal pipeline“ des OGC (Abbildung 26) basiert ebenfalls auf diesem dienstebasierten Konzept. Zuerst selektiert der „Filter“ Prozess die Geodaten (*Data Source* -> *Features*), die dargestellt werden sollen. Der „Style“ beinhaltet die Beschreibung der Geodaten in graphischen Präsentationen (Regeln & Signaturen). Bei der Generierung mithilfe des „Display Element Generators“ werden die Darstellungselemente (*Display Elements*) aus den selektierten Geodaten und ihrem „Style“ erzeugt.





**Abbildung 26: OGC Portrayal Pipeline nach OGC Reference Model 2.0. (OGC (2008a)), modifiziert**

Der *Rendering* Prozess erstellt die gerenderte (graphisch aufbereitete) kartenähnliche Darstellung aus der Sequenz von Darstellungselementen. Dies kann server- oder clientseitig geschehen. Beim serverseitigen Rendern (Abbildung 26 a) erzeugt der WMS Service die Darstellung in Rasterformat wie JPEG oder PNG. Anschließend wird die Darstellung an den Client gesendet und auf einem Gerät (wie Web Browser) sichtbar gemacht (*Display*). Beim clientseitigen Rendern (Abbildung 26 b) übernimmt der Client neben dem *Display* den *Rendering*-Prozess. Der Dienst sendet dem Client die Sequenz von Darstellungselementen als Vektorformat wie SVG oder WebCGM. Dieser verfügt z. B. über ein Plug-In, um die Vektordaten zu rendern und so am Bildschirm darstellen zu können.

Zur Umsetzung der Konzepte spielt der Portrayaldienst WMS auf Serverseite eine wichtige Rolle und wird aus diesem Grund im Anschluss vorgestellt (Abbildung 26). Ferner werden neben der Verwaltung von Regeln und Signaturen (2.3.4) in 2.3.5 und 2.3.6 clientseitige Technologien im Bereich Portrayalwebdienste erläutert.

### *Portrayalwebdienst ISO 19128 - WMS*

Es wird in der Norm *ISO 19128 – WMS Server Interface* (ISO (2005b)) zwischen folgenden WMS-Klassen unterschieden:

- „Basic WMS“
- „Queryable WMS“

Der „Basic WMS“ erzeugt kartenähnliche Darstellungen, üblicherweise im Rasterformat wie JPEG. Der Server unterteilt die Geodatenvisualisierung in Layern und ermöglicht eine Anzahl von vordefinierten Signaturen, in welchen die Layer dargestellt werden sollen. „Styled layer“ ist eine Kombination von „Layer“ und „Style“. Der „Layer“ ist eine Basiseinheit von geographischen Informationen und kann eine Sammlung von möglicherweise gemischten Objekttypen sein. Der „Style“ definiert, wie der Layer symbolisiert wird. Aus den einzelnen „styled layers“ wird übereinander eine kartenähnliche Darstellung erzeugt. Die Norm *ISO 19128 – WMS Server Interface* (ISO (2005b)) basiert auf der OGC WMS Version 1.3 und beinhaltet keinen Mechanismus für nutzerdefinierte Symbolisierungen von Objekten.

Der „Basic WMS“ beinhaltet neben den üblichen Basisserviceelementen (wie Versionisierung) folgende Operationen:

- `getCapabilities`: Liefert Informationen über die Fähigkeiten des WMS zurück (Beschreibung des Dienstes über seine Metadaten, Informationen zu Inhalt der Layer und Anfrageparameter)
- `getMap`: Liefert ein georeferenziertes Rasterbild (Karte) zurück

Der „Queryable WMS“ hingegen unterstützt neben allen Operationen des „Basic WMS“ zusätzlich die Operation:

- `getFeatureInfo`: Beantwortet Anfragen zu einzelnen Objekten (Features) aus dem dargestellten Kartenausschnitt. Diese Operation soll den WMS Client mit zusätzlichen Informationen über die Objekte im Kartenbild unterstützen. So können zum Beispiel bei einem Klick des Nutzers an einer bestimmten Position im Bild, die Koordinateninformationen zurückgegeben werden.

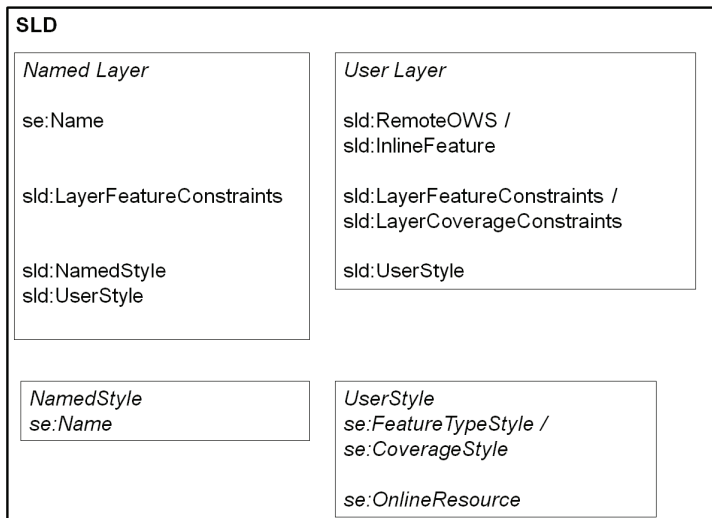
Die Norm nutzt Styles und Layers, welche dem WMS bekannt und durch den Namen eindeutig identifizierbar sind. Diese werden als „Named Layers“ bzw. „Named Styles“ bezeichnet. Durch eine Kombination von diesen beiden entsteht beim WMS 1.3 der „Styled Layer“. Die WMS Spezifikation definiert eine Sequenz von „styled layers“ wie:

```
http:// ... REQUEST=GetMap& ... &LAYERS=Rivers,Roads&STYLES=blue,red& ...
```

Es wird dabei nur die erzeugte Grafik und nicht die Daten selbst zurückgegeben. Generell werden Darstellungen im Rasterformat PNG, GIF oder JPEG Format oder in den Vektorformaten SVG oder WebCGM Format gerendert. Die Formate sind jedoch nicht zwingend vorgeschrieben. Es werden laut Norm alle MIME Formate unterstützt. Diese Version des Web Map Service unterstützt nur serverseitig vorkonfigurierte Signaturierungen. Für die Verwendung von benutzerseitig konfigurierbaren Präsentationvorschriften für die Geodaten bietet das OGC ein erweitertes Profil des Web Map Service. Dieses wird im nächsten Abschnitt beschrieben.

### ***Portrayalwebdienst OGC SLD-WMS***

Der „Style“ (Regeln und Signaturen) für einen WMS (Abbildung 26, Regeln & Signaturen) können auch über eine XML-kodierte Sprache die sogenannte *SLD* (Styled Layer Descriptor) beschrieben werden. Der OGC Standard *Styled Layer Descriptor profile of the Web Map Service* (OGC (2007b)) ist ein Profil des OGC Standards WMS 1.3 und unterstützt die Verwendung von SLDs. SLD stellt den Zusammenhang zwischen „Styling“ von Objekten, definiert in OGC Standard *SE* „OGC Symbology Encoding Version 1.1.0“ (OGC (2006a)) (s. 2.3.2) und den WMS-Layern, bereit (LayerFeatureConstraints, Abbildung 27). Allgemeines zu SLD ist in Li et al. (2004) zu finden.



**Abbildung 27: Beziehung SE und SLD**

SLD unterstützt neben den „Named Layers“ und „Named Styles“ (s. Portrayalwebdienst ISO 19128 - WMS) auch eine nutzerdefinierte Auswahl von Objekten in einem Layer („UserLayer“) (Abbildung 27). Das „LayerFeatureConstraints“-Element spezifiziert, welche Objekte welchen Objekttyps in einem Layer enthalten sind. Die Layer können nutzerdefiniert über zuvor festgelegte („NamedStyle“) oder neu definierte („UserStyle“) Signaturen präsentiert werden. Ein UserStyle erlaubt eine Festlegung der Signaturierung systemextern und als Austauschformat. Ein SLD Beispiel mit „NamedLayer“ (*Bebauung*) und „UserStyle“ (*Topo10*) ist in Abbildung 28 dargestellt.

```

<StyledLayerDescriptor ...>
  <NamedLayer>
    <Name>Bebauung</Name>
    <UserStyle>
      <Name>Topo10</Name>
      <IsDefault>1</IsDefault>
      SE Informationen (s. Abb. 25):
      <se:FeatureTypeStyle>
        <se:FeatureTypeName>Haus</se:FeatureTypeName>
        ...
      </se:FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>

```

**Abbildung 28: SLD mit SE-Elementen**

Der Portrayaldienststandard SLD-WMS unterscheidet zwischen „integrated“ und „component“ Diensten (Abbildung 29). Ein „integrated“ SLD-WMS greift auf interne Geodaten zu. Im Gegensatz hierzu ist der „component“ SLD-WMS, wie zum Beispiel der Feature Portrayal Service (FPS), eine „portrayal engine“ für *GML*- (Geography Markup Language)-Daten und stellt selbst keine eigenen Layer zur Verfügung.

Generell muss ein SLD-WMS alle Klassen des „Basic WMS“ unterstützen. Die Eigenschaften der beiden Arten von SLD-WMS werden im Folgenden aufgelistet:

- „Integrated SLD-WMS“: Unterstützt „UserStyles“, die „DescribeLayer“ Operation und entweder „GetCapabilities“ (WFS) und „DescribeFeatureType“ oder „GetCapabilities“ (WCS) und „DescribeCoverage“ Operationen
- „Component SLD-WMS“: Unterstützt „UserLayers“ und „UserStyles“. Es wird zwischen dem „Feature Portrayal Service“ und dem „Coverage Portrayal Service“ unterschieden:
  - „Feature Portrayal Service“: Unterstützt zusätzlich „RemoteWFS“
  - „Coverage Portrayal Service“: Unterstützt zusätzlich „RemoteWCS“

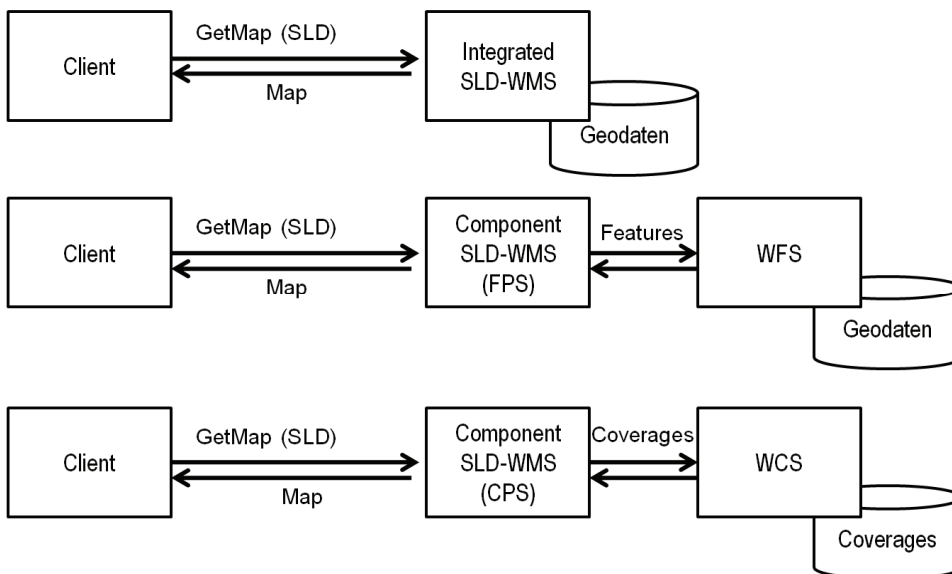


Abbildung 29: „Integrated SLD-WMS“ und „Component SLD-WMS“

Für mobile Anwendungen sind die SLD Definitionen unzureichend. Aus diesem Grund erweitert BRINKHOFF (2005) die SLD Basis Spezifikation um CSS Charakteristiken, um Anforderungen für LBS-Anwendungen unterstützen zu können. Er definiert hierbei externe Parameter für LBS, die die Darstellungseigenschaften für den jeweiligen Standpunkt verändert. ZIPF (2005) nutzt SLD für die dynamische Erstellung von nutzer- und kontextabhängigen mobilen Karten. Er beschreibt den generellen Weg um SLD in mobilen Anwendungen zu nutzen, aber es ist immer noch offen wie die SLD-Datei für bestimmte Parameter (wie Farbe, Transparenz, ...) adaptiert werden kann.

Zusammenfassend existieren erste Lösungen zur Erweiterungen der SLD Spezifikation. Jedoch hat die *ISO 19128 – WMS Server Interface* (ISO (2005b)) Norm und der *Styled Layer Descriptor profile of the Web Map Service* (OGC (2007b)) Spezifikation Grenzen. Die Interfacedefinitionen sind für die Zielsetzung der Arbeit nicht ausreichend, da Kontextinformationen keine Berücksichtigung finden und für die

Arbeit erweitert werden müssen.

### 2.3.4. Verwaltung der Regeln und Signaturen

Zur Speicherung von Regeln und Signaturen können Kataloge genutzt werden. Ein Katalog ist ein Container von unabhängigen Einheiten von Informationen. Ein Featurekatalog beinhaltet laut ISO 19110 – *Methodology for feature cataloguing* zum Beispiel Definitionen und Beschreibungen zur Objektklasse, Attributen und Assoziationen zwischen Objektklassen (ISO (2005d)). Ein Portrayalkatalog enthält laut Definition in dieser Arbeit Darstellungsregeln und Signaturen. In *Symbology Management 0.3.0 – IPR* (OGC (2006b)) wird der Zugriff auf Signaturenkataloge (Definition 2.10) über die standardisierte Schnittstelle *OGC Catalogue Service for the Web (CSW) in Catalogue Services Specification 2.0.2.* (OGC (2007a)) definiert.

Ein Register ist nach ISO/TC 211 N 2165 – *The applicability of ebXML RIM and advise to the committee according to resolution 358 from Riyadh* (ISO (2007)) ein Katalog, jedoch ein spezieller, denn er unterstützt zusätzliche Registrierungsprozeduren, wie Lebenszyklusmanagement von registrierten Einheiten oder Versionisierung. Um einen Registerdienst zu definieren und so auf den Inhalt von Registern zuzugreifen und diese verwalten zu können, hat das OGC ihren Katalogdienst (CSW) um einen Registerdienst erweitert. Dieser ist in der OGC Spezifikation als „CSW-ebRIM Registry Service - ebRIM profile of CSW“ (OGC (2009a), OGC (2009b)) (CSW-ebRIM) definiert. Inhalt dieser Spezifikation ist ein ebXML RIM als Informationsmodell und CSW als Service-Schnittstelle. Jedoch ist die Nutzung von ebXML RIM als Informationsmodell (s. Kapitel 2.2.1) und ebXML RS (Registry Service) als Service-Schnittstelle zu bevorzugen, denn CSW-ebRIM ist ein de facto Standard und eine geospezifische Schnittstelle. EbXML RIM/RS basiert hingegen auf einer Norm (ISO 15000) und ist in allgemeinen IT-Architekturen gängiger und somit interoperabel mit nicht geospezifischen IT-Welten. EbXML RIM/RS muss hierfür allerdings für räumliche Abfragen etc. erweitert werden. Hieran arbeitet die OGC seit Herbst 2008. Die Register stehen nicht im Fokus dieser Arbeit und werden hier der Vollständigkeit halber beschrieben.

### 2.3.5. Speicherung von Metainformationen für einen Portrayalwebdienst

Die *OGC Web Map Context (WMC)* (OGC (2005a)) Spezifikation beschreibt ein XML-Schema zum Abspeichern und Laden von allen Metainformationen in einem OWS-Kontext. Das ermöglicht das Abspeichern einer Sitzung bevor sie beendet wird. In einer Sitzung werden alle geladenen WMS, deren Ebenen und ob sie aktiviert sind, gespeichert. Die Parameter einer WMC-Datei sind in Tabelle 3 dargestellt. Die Elemente werden in einem XML-Dokument gespeichert. Diese Spezifikation speichert Metainformationen zu Diensten und unterstützt somit nicht die Speicherung von Kontextinformationen des Nutzers.

**Tabelle 3: Web Map Context (WMC) Parameter**

Elemente	WMC Parameter	Beschreibungen
Generelle Elemente (General)	BoundingBox, SRS	Kartenausschnitt, Koordinatenreferenzsystem
	Window	Fenster (Größe in Pixel)
	Title, Abstract, Keywords	Titel, Kurzbeschreibung, Schlüsselwörter
	ContactInformation	Kontaktinformationen

	LogoURL, DescriptionURL Extension	Logo URL, URL zu Webpage mit Erläuterungen Erweiterungen
Layerliste (LayerList)	Layer with Server, Name, Title, Abstract SRS, DimensionList DataURL, MetadataURL FormatList, StyleList	Layer mit Server, Name, Titel, Kurzbeschreibung Koordinatenreferenzsystem, DimensionList URL der Daten, URL der Metadaten Liste der Bildformate, Liste mit Signaturen

### 2.3.6. Digitale Ausgabeformate

Die Ausgabe des Portrayaldienstes ist eine grafische Aufbereitung von Geoinformationen zur Anzeige auf einem Bildschirm im Raster- oder Vektorformat. Im Web-Bereich sind die gängigsten Rasterformate *JPEG* (Joint Photographic Experts Group), *GIF* (Graphics Interchange Format) und *PNG* (Portable Network Graphics). Nähere Informationen sind in LIPP (1997) zu finden. Um Rasterdaten georeferenziert abspeichern zu können, gibt es Formate wie *GeoTIFF*, *JPEG2000* oder *GeoPDF*. Hier wird die Georeferenzierung im allgemeinen im Header der Datei abgespeichert. Die Anzeige dieser Formate im Browser setzt ein entsprechendes PlugIn voraus. Ein Beispiel für ein PlugIn für JPEG2000 ist jp2IE (JPEG2000 ActiveX Control for Internet Explorer) (JP2IE (2009)). Bei den Vektorformaten sind für die Visualisierung von Geodaten im Web folgende zwei Formate geläufig. Dies ist zum einen der W3C Standard *SVG* (Scalable Vector Graphics) (W3C (2006)) und zum anderen das „W3C Recommendation“ Dokument und der OASIS Standard *WebCGM* (Web Computer Graphics Metafile) (W3C (2007)). SVG ist eine XML-basierte 2D-Vektorgrafik. WebCGM liegt zurzeit in der Version 2.0 vor und ist eine binär-basierte 2D-Vektorgrafik. Detaillierte Beschreibungen sind in DUCE et al. (2002) zu finden. Einen Vergleich von SVG und WebCGM/ActiveCGM ist in WIEBROCK and REINHARDT (2004) und WIEBROCK and REINHARDT (2005) beschrieben. Weitere Informationen zu Web-Mapping Formaten sind in KRAAK and BROWN (2000) zu finden. Ein jüngeres Vektorformat für die Visualisierung von Geodaten ist der OGC Standard *KML* (Keyhole Markup Language) (OGC (2008b)) zur Modellierung und Speicherung von geographischen Features wie Punkten, Linien, Bildern und Polygonen in XML.

## 2.4. Fazit

In diesem Kapitel wurde auf die Grundlagen von Portrayal und Portrayalwebdiensten eingegangen. Weiterhin wurden die Bestrebungen in der Standardisierung erläutert und mögliche Anwendungsgebiete allgemein sowie Anwendungsbeispiele aus dem kommerziellen Bereich und der Forschung zusammengefasst. Es zeigt sich, dass Portrayaldienste wie der WMS aus technischer Sicht dem Nutzer in vielfältiger Form zur Verfügung gestellt werden können und bereits viele Dienste angeboten werden. Die Nutzeranforderungen an die Geodatenvisualisierung unterscheiden sich abhängig von der Nutzersituation (Kontext). Die vorgestellten standardisierten Portrayaldienste und Projekte hingegen unterstützen diese Anforderung nicht. Das Ziel der Untersuchungen ist es, diese Anforderungen mithilfe eines erweiterten Portrayaldienstes mit Kontextinformationen zu erfüllen. Aus diesem Grund wird der Begriff Kontext und kontextsensitive Systeme im Zusammenhang mit Geodatenvisualisierung im nächsten Kapitel erläutert.

### 3. Kontextabhängige Visualisierung von Geodaten

In diesem Kapitel werden die Grundlagen und der Stand der Forschung und Technik im Bereich adaptive und kontextsensitive Systeme im Allgemeinen und im Bereich Geodatenvisualisierung im Besonderen beschrieben.

#### 3.1. Kontextsensitive Systeme

##### 3.1.1. Adaptive Systeme im Bereich Informatik

Der Begriff „Adaption“ kommt in mehreren Bereichen wie Biologie, Medizin oder sogar Kunst vor. In dieser Arbeit ist der Schwerpunkt auf die Sicht der Informatik gelegt. Eine Adaption hat mehrere Vorteile wie zum Beispiel Flexibilität gegenüber Nutzerbedürfnissen und Umweltparametern und somit Verbesserung der Qualität und Gebrauchstauglichkeit. Der Begriff Adaption aus Sicht der Informatik wird im Lexikon Bertelsmann Verlag wie folgt definiert:

**Definition 3.1 (Adaption)** *„Adaption, Selbstanpassung. Ein System heißt adaptiv, wenn es sich in seiner Funktion selbsttätig an veränderte Umgebungsbedingungen anpassen kann. Es handelt sich meist um kybernetische Systeme bzw. Roboter, die durch entsprechende Ausrüstung (z.B. Sensoren) und Steuerung in der Lage sind, flexibel zu reagieren.“* (SCHIMPF and ULLFORS (1994))

Hierbei unterscheidet man zwischen zwei verschiedenen Arten von Adaptionen: Customizing und Personalisierung (KUIPER et al. (2006)).

**Definition 3.2 (Customizing)** *Customizing basiert auf expliziten Daten. Zum Beispiel kann ein Nutzer Anwendungsdaten oder das Layout adaptieren.*

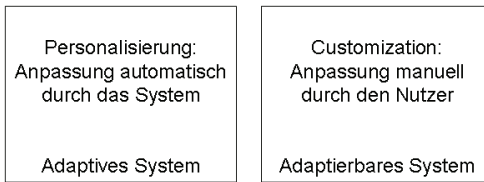
**Definition 3.3 (Personalisierung)** *Personalisierung wird vom System vollzogen und basiert auf impliziten Daten, wie Nutzerverhalten und/oder expliziten Daten wie Information aus einem eingegebenen Formular.*

**Definition 3.4 (adaptives System, engl. adaptive system)** *„Ein System ist dann adaptiv, wenn es sich selbstständig an veränderte Bedingungen anzupassen vermag.“* LEUTNER (2002)

Das bedeutet, dass eine automatische Änderung der Eigenschaften in Abhängigkeit der Nutzerbedürfnisse stattfindet. Ein Beispiel für ein adaptives System ist eine thermostatgeregelte Heizung.

**Definition 3.5 (adaptierbares System, engl. adaptable system)** *„Ein System ist adaptierbar, wenn es durch externe Eingriffe an veränderte Bedingungen angepasst werden kann.“* LEUTNER (2002)

Hier hat der Nutzer die Möglichkeit selbst in das System einzugreifen, wie zum Beispiel beim Verstellen eines Autositzes. Zur Verdeutlichung sind die Unterschiede zwischen einem adaptiven und einem adaptierbaren System in Abbildung 30 dargestellt.



**Abbildung 30: Adaptive und adaptierbare Systeme**

**Definition 3.6 (adaptiver Web Service)** *Ein adaptiver Dienst, der seine Funktionalitäten über das Internet bereitstellt ist ein adaptiver Web Service.*

Eine Klassifikation adaptiver Systeme wird im Folgenden vorgestellt.

### *Klassifikation adaptiver Systeme*

Wesentliche Forschungsgebiete im Bereich der adaptiven Systeme sind nach SPECHT (1998):

- „Adaptives Hypermedia“ (AH) (Systeme, die Eigenschaften von Benutzern in ein Benutzermodell aufnehmen und durch Anwendung des Modells ihre sichtbaren Inhaltsobjekte benutzerspezifisch anpassen)
- „Adaptive User Interfaces“ (AUI) (Anpassung des Interfaces oder der Interaktionsform)
- „Intelligente Tutorielle Systeme“ (ITS) (Anpassung tutorieller Strategien oder adaptiver Hilfefunktionen)
- „Intelligentes Multimedia“ (IMM) (Kombination und der gezielte Einsatz unterschiedlicher Medien und Präsentationsformen)
- „Intelligente (Lehr)agenten“ (IA) (Softwareeinheiten (Agenten) welche in der Lage sind, bestimmte Operationen im Auftrag eines Benutzers durchzuführen)

Adaptives Hypermedia (AH) ist zurzeit einer der populärsten Bereiche der Anwendung adaptiver Methoden. User Interface (AUI) spielen im Bereich der Geodatenvisualisierung auch eine Rolle. KRAAK and ORMELING (1996) zum Beispiel beschreiben Karten als Interface zu einem GIS. Aus diesem Grunde werden im Anschluss diese beiden Bereiche vorgestellt.

*"By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user"* (BRUSILOVSKY (1996)). Hypermedia-Systeme sind Informationssysteme, die heute häufig als verteilte Systeme realisiert werden. Sie bieten dem Anwender Zugang zu verteilten Informationen. Das World Wide Web ist das bekannteste Beispiel eines Hypermedia-Systems und zurzeit das größte existierende verteilte System. Einen sehr detaillierten Überblick über adaptive Methoden und Techniken gibt BRUSILOVSKY (2001). „Adaptive hypermedia“ (AH) ist eine Alternative zu traditionellen Systemen, wo jedem Nutzer dieselbe Ansicht und derselbe Inhalt geboten werden. Begonnen hat die Forschung im Bereich „Adaptive hypermedia“ in mehreren Forschungseinrichtungen in den 90er Jahren isoliert und wurde



ab 1996 zusammengeführt. Die Bereiche reichen hierbei von „educational hypermedia“ Systemen über „online“ Informationssystemen (wie virtuelle Museen) bis zu „Systemen mit personalisierten Ansichten“ (wie MyYahoo oder MyNetscape) (BRUSILOVSKY (2001)). Die Grundlage bilden die Nutzereigenschaften aus einem Nutzermodell. Viele Wissenschaftler sind sich über die verschiedenen Ebenen der Adaption in einem AH einig:

- Inhaltsanpassung (Text- und Multimediaanpassung)
- Präsentationsanpassung
- Strukturanpassung (Anzahl und Reihenfolge der Links)

Zukünftige Trends sieht der Autor in der Erweiterung auf andere Anwendungen, in neuen Technologien wie natürlicher Sprache oder neuen Architekturen wie Autorenwerkzeugen oder komponentenbasierten Frameworks. Auch Nutzermodellserver werden eine immer größere Rolle spielen. Ein Beispiel für ein AH-System beschreibt KOCH (2000). Sie definiert einen UML-basierten Ansatz für eine adaptive Hypermediaanwendung. Regeln werden unter Berücksichtigung des Ziels klassifiziert und beinhalten Akquisitions- und Adaptionregeln. Das Domänenmodell verwaltet die Basisnetzwerkstruktur des Hypermediasystems (Mechanismen wie Links und Knoten, Relationen und Navigation). Das Nutzermodell verwaltet die Menge an Nutzern basierend auf Nutzerattributen mit dem Ziel die Anwendung zu personalisieren.

Auch die Adaptive User Interfaces (AUI) spielen eine wichtige Rolle. Seit den frühen 90er Jahren gibt es eine Vielzahl von adaptiven Nutzerinterfaces. UI Standards steigern die Konformität, aber dieses schränkt auch die Adaption ein. So muss zwischen Flexibilität und Standardisierung abgewogen werden (NIELSEN (1993), MELZER et al. (2007)). Einen guten Überblick über dieses Thema bieten SCHNEIDERHUFSCHMIDT et al. (1993) und BROWNE et al. (1990).

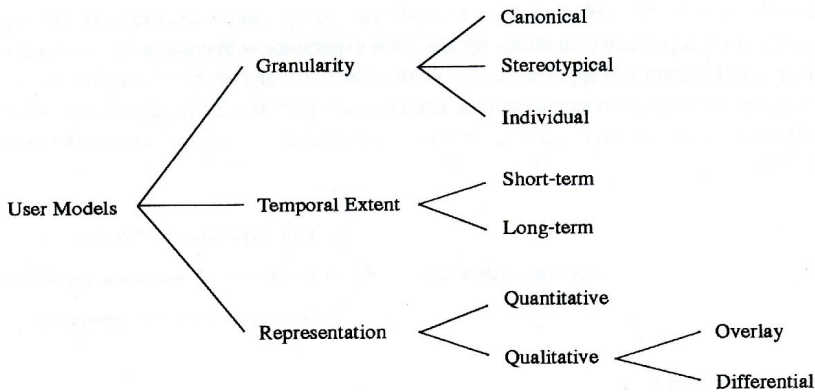
### **Nutzermodellierung**

GARCIA-BARRIOIS et al. (2005) beschreiben verschiedene Typen der Personalisierung, wobei hier Personalisierung als Adaption für einen bestimmten Nutzer definiert ist. Die Typen sind im Folgenden aufgelistet:

- Explizit (Basis sind Nutzerprofile) vs. implizit (System beobachtet Nutzer)
- Wahrnehmbar vs. versteckt (Nutzerinterface wird beeinflusst bzw. nicht)
- Prädiktiv (vorausschauend) vs. deterministisch (zu jedem Zeitpunkt ist der nächste Handlungsschritt eindeutig definiert)
- Kontrolliert vs. unkontrolliert (Nutzer kann System beeinflussen bzw. nicht beeinflussen)
- Individuell (Adaption für spezifischen Nutzer) vs. stereotypisch (Adaption für Gruppe)

In Abbildung 31 werden die Dimensionen eines Nutzermodells dargestellt. Der *Grad der Individualisierung (Granularität)* reicht von einem Modell für alle Nutzer (kanonisch) über stereotypische Nutzergruppen bis hin zur Modellierung jedes individuellen Nutzers. BRUSILOVSKY and MILLÁN (2007) unterscheiden zwei verschiedene Arten der Modellierung „feature-based modelling“ (Modellierung spezifischer Merkmale einzelner Nutzer (wie Wissen oder Ziele)) und „stereo-based modelling“ (Clustern aller mögli-

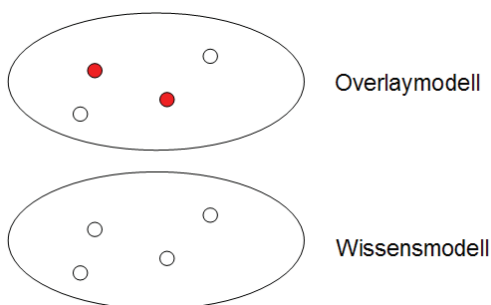
chen Nutzer in verschiedene Gruppen).



**Abbildung 31: Dimensionen eines Nutzermodells aus SCHNEIDER-HUFSCHMIDT et al. (1993)**

Ein neuerer Ansatz ist die Kombination beider Ansätze. Diesen nutzt man vor allem beim Kaltstartproblem mit neuen Nutzern. Hier werden Stereotypen genutzt, um individuelle „feature-based“-Modelle zu initialisieren. Nutzer werden in kollaborativen Familien zusammengruppiert. Einmalnutzer bekommen kein eigenes Profil, sondern erst wenn diese das System öfter nutzen. Ein weiterer Aspekt ist die *Zeit in welcher bestimmte Daten neu erfasst werden müssen* (kurzzeitig bis langfristig). Dies hängt davon ab, ob die Eigenschaften stabil sind oder sich regelmäßig ändern. Die *Repräsentation* (dritter Aspekt) kann quantitativ über Werte wie Fehlerraten oder qualitativ als zum Beispiel Overlaymodell beschrieben werden.

Overlaymodelle sind Untermengen von Wissensmodellen. Das Overlaymodell repräsentiert zum Beispiel den Wissensgrad eines Nutzers eines bestimmten Teils einer Domäne. So wird festgehalten wie viel ein Nutzer zu einem Wissenselement weiß. Die rot ausgefüllten Kreise in Abbildung 32 repräsentieren zum Beispiel Gebiete des Wissensmodells, welche vom Nutzer noch erlernt werden müssen. Das adaptive System nutzt dieses Modell, um die Inhalte dem Nutzerwissen anzupassen. Es werden bei einem Overlaymodell binäre Werte (name-value pairs) oder Gewichte festgelegt. Der Name weist auf das Domänenmodellkonzept und der Wert repräsentiert die Stufe des Wissens. Das Overlaymodell kann neben dem Nutzerwissen auch für andere Bereiche der Nutzermodellierung wie zum Beispiel der Nutzerziele Verwendung finden.



**Abbildung 32: Overlaymodell und Wissensmodell**

In BRUSILOVSKY and MILLÁN (2007) wird definiert, welche Elemente für die Nutzermodellierung eine Rolle spielen können. Dies können im Einzelnen sein:

- Nutzerwissen
- Nutzerinteressen
- Nutzerziele und –aufgaben
- Nutzerhintergrund
- Persönliche Eigenschaften des Nutzers

*Nutzerwissen* (user's knowledge): Hier werden skalare Modelle (Verwendung von Einzelwerten wie Experte, Anfänger, ...) oder Overlaymodelle verwendet.

*Nutzerinteressen* (user interest): Hier werden gewichtete Schlüsselwörter in „keyword-level models“ benutzt. Mächtiger als diese Modelle sind die „concept-level models“. Die Interessen werden separat modelliert. Zum Beispiel könnte ein adaptives Museumssystem die Interessen an Künstler, Stil, ... unabhängig voneinander modellieren.

*Nutzerziele und -aufgaben* (user's goal and task): Basis und somit das Kernstück für individuelle Nutzermodelle ist das Stereotypmodell (Clustern in Nutzergruppen). Der Nutzer kann das Modell bei einer Adaption modifizieren. Individuelle Nutzermodelle sind Überlagerungen („overlay“) auf die Nutzeraufgabenhierarchie. Diese Aufgaben können zum Beispiel in einem Zielkatalog definiert werden und der Nutzer selektiert aus diesem seine Ziele (VASSILEVA (1996)).

*Nutzerhintergrund* (user's background): Der Nutzerhintergrund ist die Definition von Elementen außerhalb der Kerndomäne. Sie werden vor allem zur Anpassung des Inhalts herangezogen. Hier kann ein einfaches Stereotypmodell verwendet werden. Ein Nutzer wird zu einer Klasse von vielen zugeordnet. Der Stereotyp enthält Fakten, Eigenschaften oder Vorlieben, die nicht unabhängig voneinander auftreten. Die Eigenschaften des Nutzerhintergrundes sind normalerweise explizite und stabile Eigenschaften.

*Persönlicher Charakter des Nutzers* (user's individual traits): Dieser definiert den Nutzer als Individuum und unterscheidet zwischen „cognitive styles“ (kognitiver Stil) und „learning styles“ (Lernstil). Der kognitive Stil beschreibt den individuellen bevorzugten und gewohnten Ansatz zur Organisation und Repräsentation von Informationen. Der Lernstil definiert den Weg wie Personen vorzugsweise lernen. Dies sind stabile Eigenschaften, die sich kurzfristig nicht ändern.

Jedoch kann in solchen Systemen neben dem Nutzer an sich zum Beispiel auch die Situation des Nutzers wie sein Standort eine Rolle spielen. Aus diesem Grund werden im Anschluss kontextsensitive Systeme definiert.

### **3.1.2. Kontext und kontextsensitive Systeme**

SCHMIDT (2002) beschreibt den Kontext allgemein mit Hilfe eines dreidimensionalen Systems mit den Dimensionen: Nutzer und Gerät (*self*), Nutzeraktivitäten (*activity*) und seine Umgebung (*environment*).

**Definition 3.7 (Kontext)** *”Any information that can be used to characterise the situation of an entity (person, place, or object).” DEY (2001)*

In vielen Arbeiten wird das „Context-aware systems“ (kontextsensitives System) als sich dynamisch veränderndes oder adaptives System charakterisiert. „Context-aware systems“ wurde erstmals von SCHILIT and THEIMER (1994) definiert:

*”Such context-aware systems adapts according to the location of use, the collection of nearby people, hosts, and accessible services, as well as to changes to such things over time. A system with these capabilities can examine the computing environment and react to changes to the environment.”*

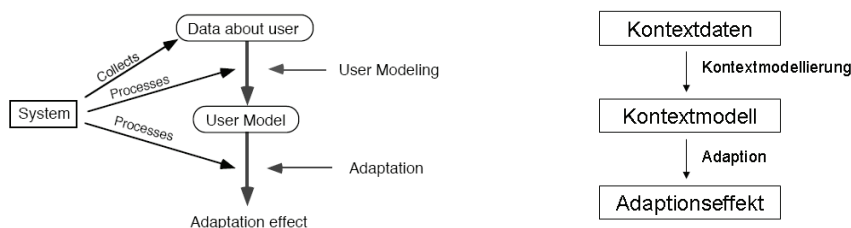
Diese Definition schränkt die Definition auf die Verwendung von Kontext und die Anpassung des Systems basierend auf dem Kontext ein. Diese Definition lässt zum Beispiel keine Speicherung oder das Anzeigen von Kontext zu. Aus diesem Grund ist für DEY and ABOWD (2000) ein System kontextsensitiv, *”if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task”*.

**Definition 3.8 (Kontextsensitiver Dienst / kontextsensitive Anwendung)** *There are three categories that context-aware applications may support:*

- *“presentation of information and services to a user;*
- *automatic execution of a service; and*
- *tagging of context to information for later retrieval”.*

(DEY and ABOWD (2000))

Für eine Adaption ist es wichtig, die jetzige Situation (Kontext) und den Nutzer zu modellieren. JAMESON (2001) beschreibt die Kombination der Modellierung von Kontext und Nutzer. In Abbildung 33 a ist das Nutzermodell und seine Adaption nach BRUSILOVSKY (1996) für ein adaptives System beschrieben. Hier werden im System die Daten gesammelt und prozessiert um auf dieser Grundlage das Nutzermodell zu definieren. Das System prozessiert mit Hilfe des Nutzermodells die Adaption. In Abbildung 33 b ist der Ablauf in einem kontextsensitiven System definiert. Hier werden zuerst die Kontextdaten zum Beispiel durch Sensoren gewonnen und ein Kontextmodell erstellt, auf dessen Basis die Adaption durchgeführt wird.



a) Nutzermodell nach Brusilovsky (1996)

b) Kontextmodell

**Abbildung 33: Adaption basierend auf Nutzer- bzw. Kontextmodell**

### *Kontextinformationen/-gewinnung*

SCHMIDT (2002) beschreibt in seiner Arbeit die Kontextakquisition durch Nutzung verschiedener Sensoren zur Erkundung des Nutzers und seiner Umgebung (Abbildung 34). Ein Ziel ist hierbei einen Überblick über mögliche Sensortechnologien und Abstraktionsmethoden zu geben. Ortsinformationen (Lokalisierung), Zeit, optische oder auch akustische Kontextinformationen, die mit Hilfe von Sensoren messbar sind werden als Low-Level-Kontextinformationen bezeichnet.

<i>Sensing Technologies</i>
Light and Vision
Audio
Movement and Acceleration
Location and Position
Magnetic Field and Orientation
Proximity, Touch and User Interaction
Temperature, Humidity and Air Pressure
Weight
Motion Detection
Gas-Sensors and Electronic Noses
Bio-Sensors
Zero-Power Sensors

**Abbildung 34: Erkundungstechnologien nach SCHMIDT (2002)**

Neben Low-Level-Kontextinformationen gibt es die High-Level-Kontextinformationen (CHEN and KOTZ (2000)). Diese sind Kombinationen von Low-Level-Kontext. Hierfür werden feste Berechnungsregeln definiert, statistische Auswertung, Mustererkennung durchgeführt oder künstliche Intelligenz verwendet. Um Kontexte vorherzusagen, können verschiedene Verfahren verwendet werden. Dies sind neuronale Netze, Bayesche Netze sowie Markov Modelle (PETZOLD (2005)). Die gerichteten Wirkeinflüsse von Kontextfaktoren lassen sich als bedingte Wahrscheinlichkeiten interpretieren. Probabilistische Graphische Modelle (engl. Probabilistic Graphical Models (PGMs)) sind eine Mischung aus Wahrscheinlichkeitstheorie und Graphentheorie. Es gibt verschiedene Ausprägungen von PGMs die auf Hidden-Markov-Modellen oder Bayesschen Netzwerken basieren (BEIERLE and KERN-ISBERNER (2006)).

Die Deutsche Forschungsgemeinschaft (DFG) finanziert einen Sonderforschungsbereich (SFB) um „Umgebungsmodelle für mobile kontextbezogene Systeme“ (Nexus) an der Universität Stuttgart zu definieren. Durch eine Integration von Teilmodellen entsteht ein umfassendes und verteilt verwaltetes Gesamtmodell, das durch entsprechende Schnittstellen einer Vielzahl von kontextbasierten Anwendungen als Basis dient (ROTHERMEL (2008)). Die Forschung beinhaltet neben der Erstellung und Verwaltung von großen und hochdynamischen Kontextmodellen, die Integration der Sensordaten (Position, Zeit, Temperatur, Helligkeit) in das Umgebungsmodell und die Präsentation und Interaktion (Abbildung 35). „*Hier steht die Visualisierung der häufig detaillierten Objekte des Umgebungsmodells auf unterschiedlichsten Endgeräten wie auch die kontextbezogene Interaktion von Benutzern im Vordergrund.*“ (ROTHERMEL et al. (2006)).



**Abbildung 35: Thematische Schwerpunkte bei Nexus (ROTHERMEL et al. (2006))**

Um ein Nutzermodell generieren zu können, müssen explizite und implizite Ressourcen verwendet werden (DOMIK and GUTKAUF (1994)). Für implizierte Modellierung wird der Nutzer beim Verwenden des Systems beobachtet, z.B. durch Mausklicks oder Augenbewegungen. Wohingegen beim expliziten Modellieren dem Nutzer direkte Fragen z. B. durch Fragebögen gestellt werden. Eine explizite Ressource wird definiert als *"A resource defined as a constraint or limitation on scheduling which is directly requested by a user in the definition of scheduling requirements."* (HAGOPIAN and MAXWELL (1996)). Eine implizite Ressource wird definiert als *"A resource defined as a constraint or limitation on scheduling which is not directly requested by a user in the definition of scheduling requirements, but which is derived from the requirements as defined by the user."* (HAGOPIAN and MAXWELL (1996)). Jedoch existiert bei jedem neuen Nutzer eines Systems das Kaltstartproblem. Ein neuer Nutzer eines Systems kann zu Beginn über eine schlechte Systemperformanz verfügen bis das Feedback des neuen Nutzers ausreichend ist. Aus diesem Grund nutzen zum Beispiel ZIGORIS and ZHANG (2006) explizite und implizite Ressourcen, um das Nutzerprofil zu erzeugen.

### **Kontextmodellierung**

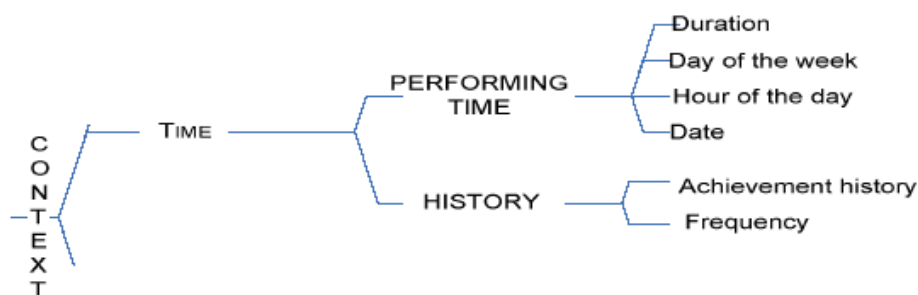
In STRANG and LINNHOFF-POPIEN (2004) werden verschiedene Kontextmodellierungsansätze beschrieben:

- Key-Value Models
- Markup Scheme Models
- Graphical Models
- Object Oriented Models
- Logic Based Models
- Ontology Based Models

Ein einfacher Weg zur Modellierung von Kontext ist die Definition von *key-value models*. *Markup scheme models* nutzen Profile mit Tags, Attributen und deren Werten. Ein typisches Beispiel solcher Profile sind Composite Capabilities/Preference Profile (CC/PP) vom W3C. Ein anderer Weg zur Kontextmo-

dellierung sind *graphical models* (graphische Modelle wie UML) oder Nutzung von *object-oriented techniques* (wie Wiederverwertbarkeit oder Vererbung). Existierende Ansätze nutzen verschiedene Objekte, um verschiedene Kontexttypen zu repräsentieren (wie Temperatur, Ort, ...). Logisch basierte Modelle (*logic based*) haben einen hohen Grad an Formalität. Facts, expressions und rules werden genutzt, um ein Kontextmodell zu definieren. Wegen der hohen und formalen Aussagekraft und zur Anwendung von Reasoningtechniken sind Ontologien (*ontology based*) ein sehr aussichtsreiches Instrument zur Modellierung von kontextuellen Informationen.

SAIDANI and NURCAN (2007) verfolgen den Ansatz, den Kontext in einem dreidimensionalen Raum  $S = \langle \text{ASPECTS, FACETS, ATTRIBUTES} \rangle$  zu beschreiben. Kontext kann in verschiedene Aspekte (aspects) wie Ort, Zeit (time), ... unterteilt werden. Facets (für Zeit zum Beispiel Historie (history)) sind mit Attributen (frequency) beschrieben (Abbildung 36).



**Abbildung 36: Kontextbaum nach SAIDANI and NURCAN (2007) für den Aspekt Zeit (time)**

In TRAN et al. (2008) wird ein Ontologiemodell zusammen mit Adaptionenregeln im Bereich „Hypermediasysteme“ definiert. Die Ontologie stellt eine Basis für das Wissen für die Adaptionenregeln bereit und beinhaltet die verschiedenen Modelle wie Systemmodell oder Aufgabenmodell.

Ein formaler Ansatz um eine kontextsensitive Anwendung zu beschreiben wird von SCHMIDT and GELLERSEN (2001) definiert. Diese definieren eine

Kontextsensitive Anwendung  $A = (\Omega, \Theta, R)$  mit

$\Omega = \{k_1, k_2, \dots, k_n\}$ : Menge der verwendeten Kontexte und

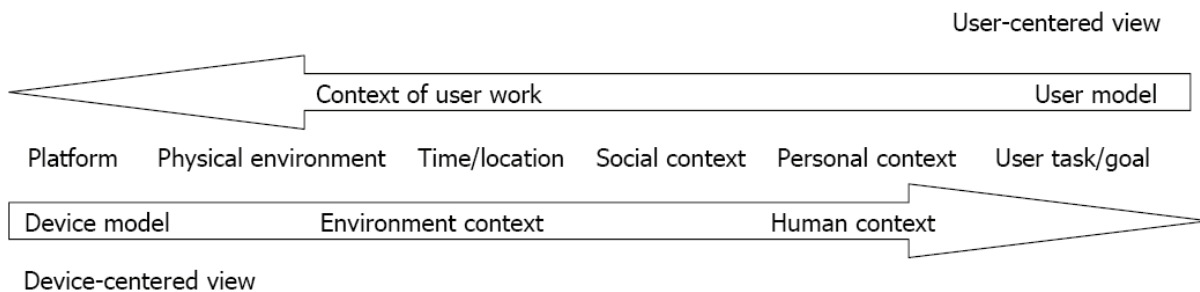
$\Theta = \{f_1, f_2, \dots, f_m\}$ : Menge der anwendungsbezogenen Funktionen und

$R: P(\Omega) \rightarrow P(\Theta)$ : Relation (Mapping von Potenzmenge des Kontextmenge und der Potenzmenge der Funktionsmenge).

Eine Beispielanwendung in dieser Publikation definiert ein Mobiltelefon mit Kontextbezug. Die Menge der Kontexte ist hierbei:  $\Omega = \{\text{Handy in Tasche, Handy in Hand, Handy in Besprechung}\}$  und die Menge der verwendeten Funktionen für die Mitteilung einer Nachricht sind  $\Theta = \{\text{Akustisch, Vibration, Visuell}\}$ . Die Relationen werden wie folgt definiert  $R = \{(\text{Handy in Hand, Visuell}), (\text{Handy in Besprechung, Vibration}), (\text{Handy in Tasche, Akustisch})\}$ . Dies bedeutet z. B. wenn das Handy sich in der Hand befindet, wird

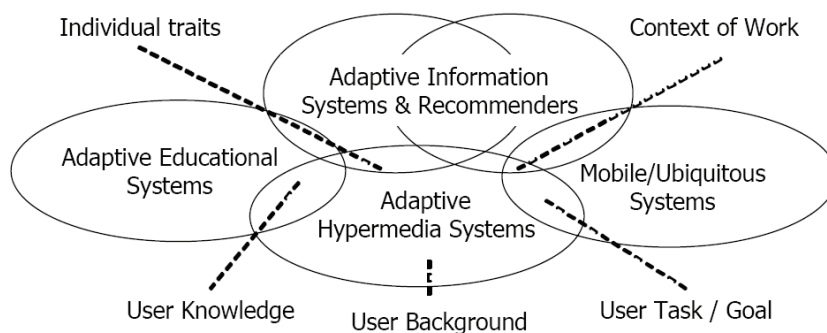
eine neue Mitteilung durch Blinken angezeigt. Die Autoren versuchen eine abstrakte Beschreibung für eine kontextbasierte Anwendung zu definieren. Jedoch definiert die Beschreibung nicht, in welcher Form die Kontexte die Funktionen der Anwendungen beeinflussen.

In BRUSILOVSKY and MILLÁN (2007) wird ein kurzer Ausblick gegeben, welche Elemente für die Kontextmodellierung eine Rolle spielen können. Dies sind zum Beispiel Elemente wie Nutzerplattform (System, Bandbreite, ...) oder Nutzerstandort. Bisher hatte die Forschung eine nutzerorientierte Sichtweise (user-centered view). Dieses hat sich im Bereich „mobile and ubiquitous computing“ geändert. Man kann hier von einer systemorientierten Sichtweise sprechen (device-centered view) (Abbildung 37). Der nutzerorientierte Ansatz sieht die Nutzeraufgabe („user task“) als Ziel des Nutzermodells und nicht den Kontext. Die systemorientierte Sichtweise unterscheidet verschiedene Kontexte wie „environment context“ (räumlich-zeitliche Aspekte und physikalische Konditionen) oder „human context“ (persönlicher Kontext, sozialer Kontext und Nutzeraufgabe).



**Abbildung 37: Kontext aus nutzer- und systemorientierter Sichtweise (BRUSILOVSKY and MILLÁN (2007))**

Die populärsten Eigenschaften sind wie im letzten Kapitel beschrieben: *Nutzerwissen* (user's knowledge), *Nutzerinteressen* (user interest), *Nutzerziele und -aufgaben* (user's goal and task), *Nutzerhintergrund* (user's background), *Persönliche Eigenschaften des Nutzers* (user's individual traits) und der *Kontext* (context of work). Jedes einzelne adaptive System beinhaltet einen Teil dieser Liste (BRUSILOVSKY and MILLÁN (2007)) (Abbildung 38).



**Abbildung 38: Nutzereigenschaften mit verschiedenen adaptiven Websystemen (BRUSILOVSKY and MILLÁN (2007))**



Nicht zufrieden mit solch einer generellen Definition und um die Komplexität zu reduzieren, versuchen viele Wissenschaftler den Kontext abhängig von einer spezifischen Anwendung zu definieren. Ein Beispiel für adaptive Web-Anwendungen beschreiben ZIEGLER et al. (2005) in fünf Kategorien:

- *Benutzer & Rolle* (Definition individueller Benutzer oder Gruppen von Benutzern anhand ihrer unterschiedlichen Rollen.)
- *Aufgabe* (Arbeitsaufgaben oder persönliche Zielsetzungen.)
- *Ort* (Beschreibung von Ortsmodellen. Ein Beispiel ist die Position im Netzwerk (Intranet / externer Zugriff.)
- *Zeit* (Zeitinformationen.)
- *Gerät* (Geräteinformationen, wie z.B. der Gerätetyp (Desktop PC, PDA) oder Display-Einstellungen.)

Auch für den Bereich Geodatenvisualisierung und deren Anwendungen muss das Kontextmodell mit seinen Elementen spezifiziert werden. Aus diesem Grund wird speziell dieser Bereich im nächsten Kapitel erläutert.

### 3.2. Kontextsensitive Systeme und Geodatenvisualisierung

Seit rund 500 Jahren produzieren Kartographen vorwiegend traditionelle Papierkarten. Traditionelle analoge Karten können immer nur Informationen in einer Form wiedergeben. Eine interaktive, nutzerspezifische Veränderung und Anpassung der Darstellung ist nicht möglich. Computer bieten eine neue Form der Informationsübertragung. Es wird eine interaktive und adaptive Informationsdarstellung möglich. Sie werden in der Kartographie seit den 50er Jahren eingesetzt und fanden ihren Durchbruch in den 90er Jahren durch Internet und webbasierte Dienste. In den letzten Jahren erfolgte eine immer größere Einbeziehung des Nutzers im WWW. Ein Beispiel hierfür ist das Web 2.0 (WEB2.0 (2009)). Web 2.0 Beispiele aus dem Geobereich sind im Folgenden aufgeführt:

- MapMixer/Yahoo! (Zusammenstellung einer Karte aus mehreren Karten)
- Loopt (Kartentool für das Mobiltelefon)
- YourStreet (Neuigkeiten aus der Nachbarschaft)
- MyMapChat (Markierung von Orten auf Karten mit Kommentaren)
- Pushpin (Einbindung Karten für eigene Webseiten)
- GeoCommons (Geodaten sammeln und austauschen)
- Placeopedia (Wikipedia-Artikel mit geographischen Informationen verknüpfen)

Der Zweck einer Kartennutzung geht von der reinen Visualisierung bis hin zur Kommunikation. MACEACHREN and TAYLOR (1994) definierten die Visualisierung über die Kartennutzung („*visualization in terms of map use*“) (Abbildung 39).

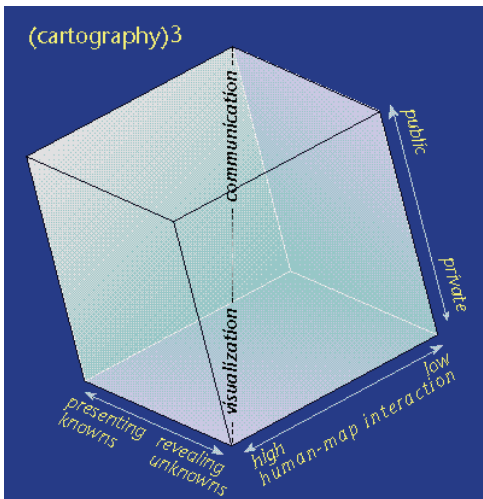


Abbildung 39: „Visualization in terms of map use“ (MACEACHREN and TAYLOR (1994))

Sie unterscheiden hierbei, welche dieser Aktivitäten hervorgehoben wird und definieren einen dreidimensionalen Raum mit:

- *Zielpublikum* (öffentliches Publikum bis private Individuen)
- *Ziel der Kartennutzung* (simple Informationsbereitstellung bis Informationserkundung)
- *Flexibilität/Interaktion der Kartennutzung* (niedrige statische bis hohe dynamische Karten)

Die digitale Umgebung lässt laut MENG (2003) die Möglichkeit zu, die Nutzeranforderungen in den Kartengestaltungsprozess zu integrieren und somit adaptive Karten herzustellen.

**Definition 3.9 (adaptive Karte)** Eine digitale Karte wird als „adaptiv“ bezeichnet, wenn sie sich selbstständig an veränderte Bedingungen abhängig von Kontextparametern anzupassen vermag.

MENG (2004) definiert das Egozentrum einer egozentrischen Karte als „eine Zusammenfassung der Nutzereigenschaften, der Kartenzwecke, der Kontextaspekte sowie deren Wechselwirkungen miteinander“. Das bedeutet, im Vergleich zu traditionellen Karten ist eine egozentrische Karte adaptiv und auf speziellen Nutzerbedürfnisse zugeschnitten und bietet daher eine „one-to-one“ statt „one-fit-for-all“-Lösung an. Gestaltungsmethoden für egozentrische Karten werden in MENG (2004) gegeben.

### Allgemeine Geonanwendungen

Für die Erstellung von adaptiven Präsentationen ist es wichtig, den Nutzer und seine Situation zu modellieren und auf dieser Grundlage eine Anpassung an seine spezifische Geodatenauswahl, deren Signaturierung (portrayal) und ein angepasstes Interface zu berücksichtigen (SARJAKOSKI and SARJAKOSKI (2008)).

In der Publikation von YU et al. (2003) wurde ein Ansatz beschrieben, um Nutzermodelle durch die Verwendung einer Geodatenbasis zu erzeugen. Hierbei sind drei verschiedene Parameter definiert worden, die einen „visiting point“ (Abbildung 40) definieren:

- Geographisches Gebiet (*area*)
- Inhaltskategorien (*content category*) und
- Themenlayer (*theme layers*).

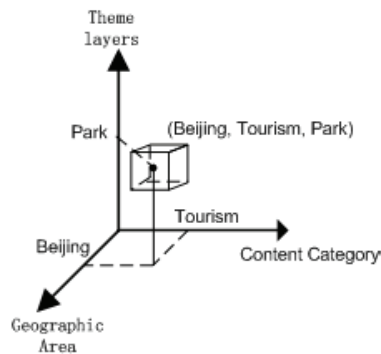


Abbildung 40: „Visiting points“ nach YU et al. (2003)

Im Beispiel in Abbildung 40 wird der visiting point „g<sub>1</sub> (Beijing, Tourism, Park)“ definiert. Diese „visiting points“ wurden in einem Test erfasst und statistisch ausgewertet. Hieraus konnten verschiedene Nutzermodelle mit Hilfe von Bayesche Netzen (Kapitel 3.1.2) abgeleitet werden. Basierend auf diesen Nutzermodellen schlugen die Autoren zwei verschiedene Implementierungsansätze vor. Dies ist die Adaption der Geodatenbasis selbst und ein Geovisualization(web)dienst mit Adaptionfunktionalität.

In MOISUC et al. (2006) wird die Personalisierung zur Geodatenselektion und angepassten Repräsentation durch einen modellbasierten Ansatz durchgeführt. Das ASTIS (adaptive spatio-temporal information system) ist zum Studieren und Verwalten von Naturgefahren erstellt worden. Es beinhaltet drei Module: *Daten-, Präsentations- und Adaptionsmodul*. Das Datenmodul speichert die Daten und initiiert das Datenmodell. Die Beschreibung der Nutzer wird in Nutzerprofilen definiert. Die Definition der Adaption und deren Ausführung werden im Adaptionsmodul realisiert. Das Präsentationsmodul transformiert die Daten vom Speicher- in das Visualisierungsformat und präsentiert die Daten mit Hilfe eines Visualisierungsinterfaces.

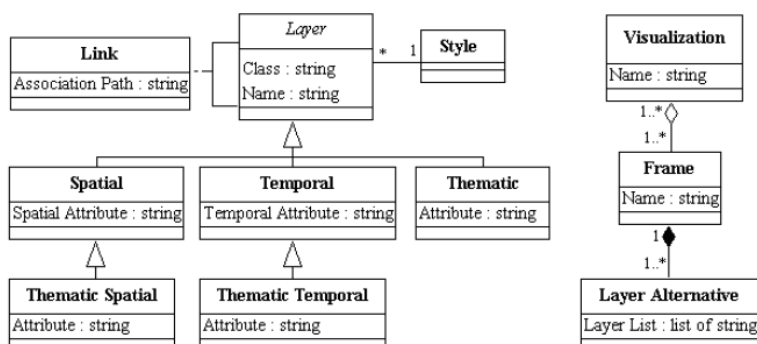


Abbildung 41: Präsentationsmodul nach MOISUC et al. (2006)

Das Präsentationsmodul (Abbildung 41) besteht aus Layern (*layers*), die thematische, temporale und räumliche Informationen beinhalten können. Ein *frame* kann aus mehreren Layern bestehen und wird mit Hilfe des Moduls visualisiert. Außerdem werden in dem System zwei visuelle Mechanismen unterstützt: *masking* und *highlighting*. Beim *masking* werden alle Layer eines nicht relevanten Frames ausgeblendet. Das *highlighting* hingegen bezieht sich nur auf ein Layer, welches hervorgehoben werden kann. Das System berücksichtigt zurzeit den Nutzer und seine Expertise. In der Zukunft ist eine Erweiterung durch Kontextinformationen geplant.

Ein adaptives GIS wird in PETIT et al. (2006) beschrieben. Die Autoren unterteilen den Kontext in Nutzerkontext, geographischen Kontext und Systemkontext. Anwendung ist hier die maritime Navigation (Abbildung 42). Das Modell identifiziert verschiedene Bereiche / Regionen des geographischen Kontexts. Dies sind der Standort des Nutzers (U), verfügbare Daten (D), prozessierte Daten (P) und Interesse einer bestimmten Region (S). In einem gewöhnlichen GIS sind U, P und D statisch. Dies ist hier jedoch nicht der Fall.

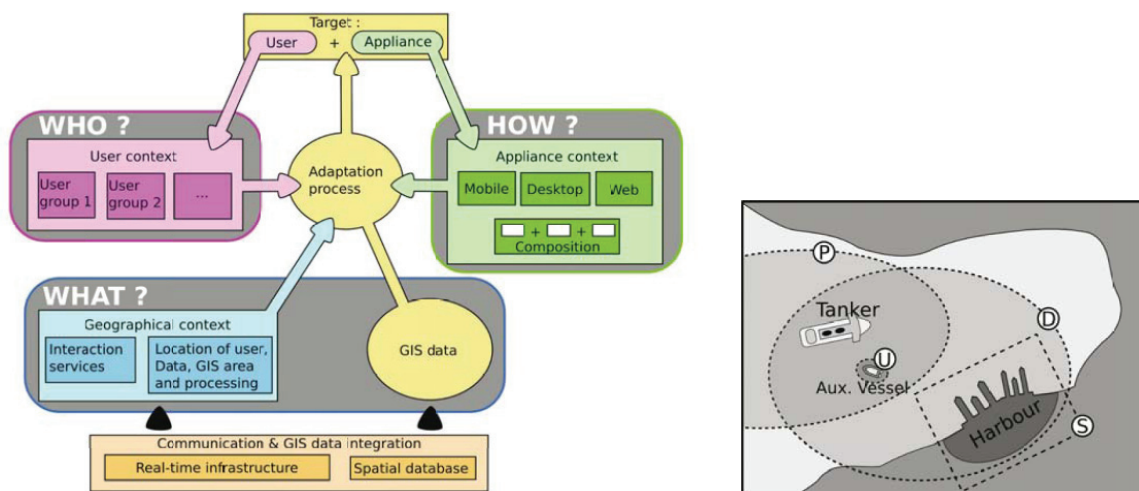


Abbildung 42: Adaptives GIS nach PETIT et al. (2006)

In AOIDH et al. (2006) wird ein personalisiertes GIS Interface vorgeschlagen. Informationen sollen über ein personalisiertes Karteninterface im Bereich archäologisches Kulturerbe erreichbar sein. Die Nutzer sind in diesem Fall Touristen, fortgeschrittene Nutzer und Expertennutzer. Die Nutzerinteraktionen werden implizit erfasst und beziehen sich auf räumlich-bezogene und nicht räumlich-bezogene Interaktionen. Diese beinhalten Mausektionen, Zoom und Pan und explizierte Suche und Objektplatzierung. Hierbei wurden Adaptionen im Bereich Inhalt und Interface umgesetzt. So beeinflusst zum Beispiel das Nutzermodell die Größe der Fenster. Hat zum Beispiel ein Nutzer weniger Wissen von geographischen Daten (Tourist) wird ein kleineres Fenster für die Karte und größeres Fenster für Erklärungen angezeigt. Eine Besonderheit des Systems ist eine „Web Search Engine“, die das Ranking der Resultate nutzerabhängig für räumliche und nicht-räumliche Daten durchführt.

### Mobile Geoanwendungen

Einen Überblick über die kontextsensitive mobile Forschung bieten CHEN and KOTZ (2000). Dies beinhaltet einerseits die Kontextinformationsgewinnung für den Standort, aber auch Zeit und Objekte in der

näheren Umgebung. Des Weiteren werden Datenstrukturen für Kontextmodelle im mobilen Bereich vorgestellt. Gründe für die Anpassung und somit die Förderung der Gebrauchstauglichkeit vor allem in mobilen Anwendungsbereich sind die Grenzen bei der Datengröße und kleine Displays. Geovisualisierungsdienste und die Möglichkeit ihre Gebrauchstauglichkeit zu testen ist in MENG (2006) beschrieben. Folgende Adaptionmethoden sind laut REICHENBACHER (2008) für mobile Anwendungen einsetzbar:

- Selektionsmethode (Karteneinhalt reduzieren, Informationsdichte)
- Priorisierungsmethode (Mit Hilfe von Prioritätsklassen)
- Substitutionsmethode (Ersatz von verschiedenen Visualisierungsformen)
- Symbolisierungsmethode (Verschiedene Symbole)
- Konfigurationsmethode (Verschiedenen Maßstäbe)
- Kodierungsmethode (Vektor oder Raster)

Im Folgenden werden relevante Arbeiten und Projekte im mobilen Anwendungsbereich und Kontext vorgestellt. Der Fokus in REICHENBACHER (2004) basiert auf der Ausarbeitung einer adaptiven Methode zur Visualisierung von geographischen Informationen für die mobile Nutzung. Der Kontext (C) kann hier als Funktion wie folgt beschrieben werden:

$$C = f(l, t, us, ac, in, te) | co$$

Die Elemente sind hierbei: l = location (Position als Koordinatenpaar, Ortsname oder Adresse), t = time (exakte Zeit, Zeitintervall oder Tageszeit), us = user (Identität oder Nutzergruppe), ac = activity (Aktion und Ziel), in = information, te = technology (wie Gerät, Netzwerktyp und Bandbreite), co = constraints (gültig für den Kontext).

In Abbildung 43 ist der Geovisualisierungsdienst für die adaptive Visualisierung von geographischen Informationen für die mobile Nutzung dargestellt. Die Kontextparameter werden in eine SOAP-Message eingebunden und vom Client an den Server gesendet. Die benötigten Geodaten werden anschließend von einem OGC WFS-Dienst abgefragt. Diese GML-Daten werden serverseitig über XSLT nach SVG umgewandelt. Hierbei werden weitere Adaptionen im SVG Document Object Model (DOM) durchgeführt. Die SVG-Datei wird anschließend an den Client zurückgesendet.

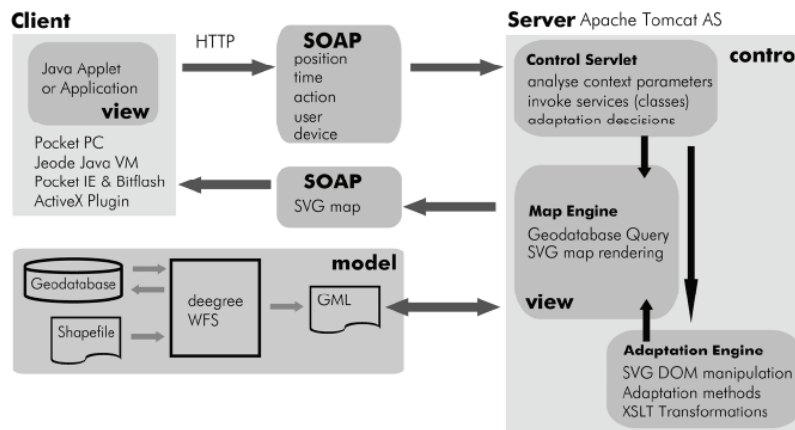


Abbildung 43: Geovisualisierungsdienst nach REICHENBACHER (2004)

Die strikte Einhaltung von OGC Standards in Reichenbacher ist nicht gegeben. So sind zum Beispiel das SOAP-Protokoll und somit die Kartenabfrage nicht standardkonform. Auch eine Einbindung eines OGC WMS oder die Unterstützung von SLD ist nicht vorhanden. Weitere Forschung ist laut REICHENBACHER (2003) außerdem in der Modellierung des Kontextes und somit verschiedener Domänen, der Entwicklung von Adaptionsmethoden und -regeln erforderlich. Auch eine Gewichtung der einzelnen Inputparameter ist notwendig.

Zipf und Jöst schlagen in ihrer Arbeit ein integriertes Situationsmodell mit verschiedenen Typen von Kontexten für einen adaptiven mobilen GI Dienst vor (ZIPF (2005), ZIPF and JÖST (2006)). Das System *MAPPER* berücksichtigt bei der Erstellung von personalisierten Karten im mobilen Bereich die Restriktion von Bildschirmgröße und begrenzter Bandbreite. Alle Aktionen des Nutzers werden implizit erfasst und das Nutzermodell dynamisch aktualisiert (WEAKLIAM et al. (2008)). Das EU-Projekt *CRUMPET* (Creation of user-friendly mobile services personalized for tourism) definiert einen mobilen Touristeninformationsdienst. Hier wird abhängig von einem adaptiven Benutzermodell (inklusive Endgerät) eine personalisierte und ortsabhängige Auswahl von Tourismusinformationen getroffen und dargestellt (SCHMIDT-BELZ et al. (2003)). In OPPERMANN and SPECHT (1999) unterstützt ein mobiler Museumsführer angepasste Informationen und das angepasste Lernen von Besuchern während einer Kunstaussstellung.

Sarjakoski und Nivala entwickelten eine Klassifikation für Kontext für mobile Dienste (NIVALA and SARJAKOSKI (2003), SARJAKOSKI and NIVALA (2005)). Die Kategorien werden in der Tabelle 4 generellen Kontextkategorien gegenübergestellt.

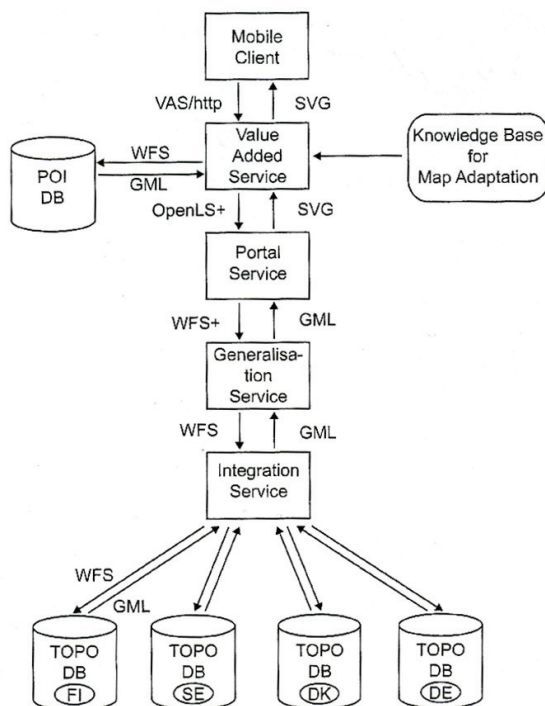
Tabelle 4: Kontextkategorien nach NIVALA and SARJAKOSKI (2003)

General context categories (Generelle Kontextkategorien)	Context categories for mobile maps (Kontextkategorien für mobile Dienste)
Computing	System
User	Purpose of use, User, Social, Cultural

Physical	Location, Physical surroundings, Orientation
Time	Time
History	Navigation History

Diese Kategorisierung beinhaltet das System des Nutzers, seine Aufgaben, physikalische Konditionen wie Standort und die Umgebung sowie Zeit und Navigationsverlauf. Die Modellierung der Nutzeraufgabe wird als brauchbarer Ansatz gesehen. „Modelling of user task is considered as a feasible approach that helps to reduce the great diversity of user characteristics influencing the usability of maps“ (MENG (2003)). So kann zum Beispiel auch das LOD (Level of detail) nach Nutzerinteresse angepasst werden.

Im EU-Projekt mit dem Titel „Geospatial Info-Mobility Service by Real-Time Data-Integration and Generalisation“ (GiMoDig) wurde ein Generalisierungsdienst für mobile Anwendungen definiert. Dieser basiert auf den OGC Spezifikationen WFS und OpenLS und wurde im Projekt erweitert (WFS+ und OpenLS+) (Abbildung 44).



**Abbildung 44: GiMoDig Service architecture nach SARJAKOSKI and SARJAKOSKI (2007)**

Dieser Generalisierungsdienst berücksichtigt Kontextinformationen des Nutzers. Diese werden durch einen Value-Added Service (VAS) an den WFS weitergegeben. Die Werte beziehen sich u. a. auf Anwendungsfall, Gerät, Alter und Position des Nutzers (SARJAKOSKI and SARJAKOSKI (2007)).

Auch JÖST (2009) schlägt ein Konzept und eine Architektur für die Berechnung von nutzeradaptiven



Stadtrundgängen durch eine Erweiterung der *OGC OpenLocation Spezifikation* (OGC (2005c)) in einer „OpenLS Tour Specification“ vor. Außerdem werden verschiedene zusätzliche *OGC Web Processing Services* (OGC (2007c)) kombiniert. Diese berechnen die personalisierten Routen zum Beispiel durch Gewichtung der POIs. Basisinformationen der *OGC OpenLocation Spezifikation* (OGC (2005c)), kodiert in XML abstrakten Datentypen, werden durch zusätzliche Datentypen wie „Graph“ u. a. bestehend aus Knoten, Kanten und Gewichtung erweitert.

### Ubiquitous Public Access

Die nächste Generation nach den mobilen Geoanwendungen (vorheriger Abschnitt) ist das „Ubiquitous Public Access“. Die Anforderungen sind im Entwurfspapier zu ISO 19154 – *Standardization Requirements for Ubiquitous Public Access* zu finden. Hier wird Ubiquitous Public Access (UPA) definiert als „a form of geographic information services produced and consumed by public users in any place at any time for any device.“ Diese Generation entstand im Jahr 2000 durch die Entwicklungen kleiner Hardwaregeräte, eingebettete Systeme und drahtlose Kommunikation. Ein wesentlicher Unterschied neben der größeren Nutzerbeteiligung (Produktion, Verteilung und Nutzung der Geodaten) besteht auch in der Berücksichtigung der Situation und Besonderheiten des Nutzers. Die Anforderungen an eine Standardisierung der UPA sind unter anderem Profile für Datenmodelle und Dienste (Abbildung 45).

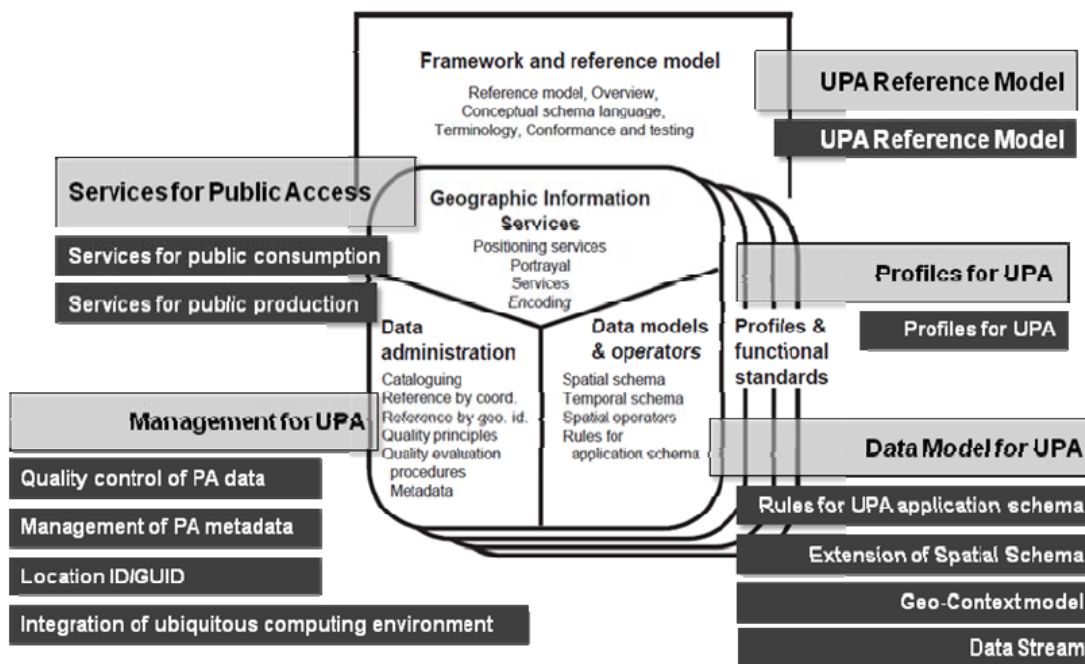


Abbildung 45: Standardisierung der Anforderungen an UPA (ISO 19154)

Die Standortinformationen eines Nutzers ändert sich sobald er sich zum Beispiel in einem anderen Raum wie „drinnen“ („Indoor“-Raum) nach „draußen“ (euklidischer Raum) und „draußen“ zu „Auto“ (Straßenraum) bewegt. Die Unabhängigkeit von unterschiedlichen Standortrepräsentationsmethoden ist das Ziel der ISO 19151 – *Logical location identification scheme* (ISO (2009a)). Die ISO definiert ein logisches „u-position naming scheme“ und ein Interface für Operationen, um diese zu nutzen. Die „u-position“ besteht aus einer ID (URI, Beispiel: `uri:upos://scientists.org/charles_darwin`), einer Zeitangabe und der Posi-



---

tion. Auf Serverseite werden die u-pos (Beispiel: darwin\_office) mit „physical location“ (Beispiel: E1271213, N03750015) und „reference system type“ (Beispiel: CRS) verknüpft. Die Operationen unterstützen die Erzeugung, das Lesen, die Aktualisierung und das Löschen von „u-position“. Diese ISO 19151 ignoriert jedoch andere ISO Normen wie WMS und OGC Spezifikationen.

### 3.3. Fazit

In diesem Kapitel wurde auf den Bereich des Kontextes eingegangen. Ferner wurden mögliche Anwendungsgebiete allgemein sowie Anwendungsbeispiele aus dem kommerziellen Bereich und der Forschung zusammengefasst. Durch die Zunahme von Daten und Nutzern ist eine Anpassung der Informationen wichtig. Aus diesem Grunde spielt der Kontext eines Nutzers in den mobilen Anwendungen eine relevante Rolle und es wurde gezeigt, dass einige Arbeiten zu diesem Thema geleistet worden sind. Doch speziell im Bereich von standardisierten Webumgebungen besteht noch Forschungsbedarf. Ein Hauptproblem in den verschiedenen Ansätzen von kontextsensitiven Systemen ist die Nutzung verschiedener Kontextkodierungen und der Weg Zugriff auf diese Ressourcen zu erhalten. Jedes System hat sein eigenes Format und seine eigene Kommunikation. Dieses sollte sich laut BALDAUF et al. (2007) durch eine Standardisierung der Formate und Protokolle verändern. *“In our opinion web service technologies seem to be an appropriate solution to achieve that aim as they provide standardised methods for service description and access.”* (BALDAUF et al. (2007)). Dieser Nachteil soll im Bereich Geodatenvisualisierung durch die Erweiterung von standardisierten Diensten in dieser Arbeit behoben werden. Als Grundlage zur Kontextmodellierung werden im Folgenden verschiedene zuvor beschriebene Kontextmodelle aus der Literatur genutzt und anhand zweier Szenarien im Bereich Geodatenvisualisierung geprüft und gegebenenfalls erweitert.



## 4. Komponenten zur kontextabhängigen Geodatenvisualisierung

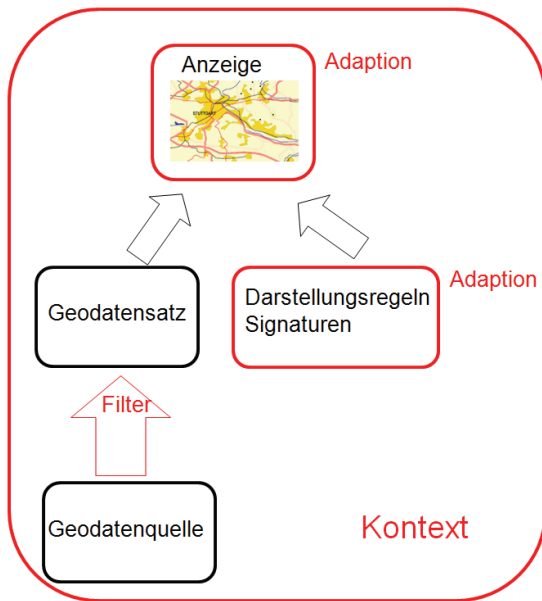
LEUTNER (2002) definiert ein System als *adaptiv*, „wenn es sich selbstständig an veränderte Bedingungen anpassen vermag“ (Kap. 3.1). Das bedeutet, es findet eine automatische Änderung der Eigenschaften in Abhängigkeit der Nutzerbedürfnisse statt. Die Anpassung der Visualisierung in einer Webumgebung abhängig vom Kontext kann in einem adaptiven System (Anpassung bzw. Auswahl der Geodaten, Werkzeuge (Navigation und Analyse), Bildschirmanzeige und Signaturierung) realisiert werden. Hierfür soll in dieser Arbeit ein standard- und webbasiertes Konzept zur kontextabhängigen Visualisierung von Geodaten erarbeitet werden. Um Daten von unterschiedlichen Quellen nutzen zu können, soll das Konzept auf Standarddiensten basieren, um Interoperabilität zu gewährleisten. Ausgangspunkt der Untersuchungen ist das Portrayalmodell und die Dienstarchitektur beschrieben in den vorangehenden Kapiteln. Auf dieser Basis sind im Einzelnen folgende Schritte notwendig:

- Definition der Komponenten zur kontextabhängigen Geodatenvisualisierung
- Analyse der Portrayalanforderungen am Beispiel zweier Szenarien aus den Bereichen Freizeit und Militär inklusive Nutzerrollendefinition
- Modellierung eines Kontextmodells und Identifizierung von weiteren notwendigen Informationen, um das Kontextmodell verwenden zu können
- Definition des Einflusses des Kontextmodells auf die Geobjektauswahl, Signaturierung, Bildschirmanzeige und Werkzeuge
- Erweiterung des Portrayalmodells aufgrund der kontextsensitiven Anforderung
- Definition von Regeln zur Unterstützung des Portrayalprozesses

### 4.1. Komponentendefinition

Die Komponenten zur kontextabhängigen Geodatenvisualisierung werden im Folgenden beschrieben. Das originale Portrayalmodell und die Portrayalwebdienstarchitektur beschrieben in Kapitel 2 und die generelle Idee des Kontextes und der Modellierung in Kapitel 3 sind hierfür die Basis. Um eine kontextabhängige Visualisierung zu erzeugen, ist ein kontextabhängiger Portrayalprozess notwendig der folgende Daten verwendet (Abbildung 46):

- Geodaten
- Kontextsensitive Darstellungsregeln und Signaturen
- Kontextdaten
- Regeln zur Unterstützung des Portrayalprozesses (Filter und Adaption)



**Abbildung 46: Kontextabhängiger Portrayalprozess**

Im kontextabhängigen Portrayalprozess können permanent aktuelle Kontextinformationen direkt verwendet oder abgeleitet werden. Mit Hilfe der Kontextinformationen werden die Geodaten gefiltert und die kontextsensitiven Darstellungsregeln und Signaturen adaptiert, um eine Präsentation der Geodaten zu erzeugen (Abbildung 46). Ferner werden aus den gesamten Werkzeugen kontextabhängig die notwendigen Werkzeuge selektiert.

Die beschriebenen Instanzen sollen auf Modellelementen einer Modellebene basieren. Somit beschreibt die Modellebene welche Instanzen auf Instanzebene existieren können und wie diese aussehen. Aus diesem Grund besteht die Notwendigkeit die Kontextinformationen zu modellieren (4.3). Ferner muss im Folgenden ein kontextabhängiges Portrayalmodell (4.4) spezifiziert werden, um kontextsensitive Darstellungsregeln und Signaturen verwenden zu können. Diese Ergebnisse werden in Kapitel 4.5 diskutiert.

Im nächsten Kapitel 4.2 werden die Anwendungsszenarien definiert (4.2.1 und 4.2.2), um daraus die Nutzer klassifizieren und deren Rollen definieren zu können (4.2.3).

## 4.2. Anwendungsszenarien

Für die Arbeit sind zwei verschiedene Anwendungsszenarien aus dem Bereich Freizeit und Militär ausgewählt worden. Den Nutzerklassen der Szenarien werden über Rollen Aufgaben zugeordnet.

**Definition 4.1 (Rolle)** *Eine Rolle ist eine Bündelung von Zielen, Aktivitäten/Aufgaben und Rechten. Diese werden einer Nutzerklasse zugeordnet.*

Dabei werden folgende Einschränkungen für diese Arbeit festgelegt. Es werden in dieser Arbeit 2D-Vektordaten kontextbasiert visualisiert, wohingegen 3D-Darstellungen und Textdarstellung nicht berücksichtigt werden.

#### 4.2.1. Anwendungsszenario Bereich Freizeit: Freizeitplanung und Notfallsituation

Im Rahmen dieses Szenarios soll eine Radwanderung geplant und durchgeführt werden. Hierbei soll per Rad die Landschaft erkundet, einzelne Sehenswürdigkeiten besucht und in Gasthäusern übernachtet werden. Die Nutzerrollen in diesem Szenario sind *GIS Analyst Radwanderung* welche Radtouren plant und *Auskunftssuchender Radwanderung* welche die Radwanderung durchführt. Ein Nutzer der Rolle *Auskunftssuchender Radwanderung* möchte in diesem Szenario eine Radtour angepasst an seine Bedürfnisse angezeigt bekommen. Die Auswahl der Radtour hängt von verschiedenen Nutzerbedürfnissen des Nutzers der Rolle *Auskunftssuchender Radwanderung* ab. Dies sind Kriterien wie Tourart (Mountainbike, Sportradfahren, ...) und den POIs (Sehenswürdigkeiten, Rastmöglichkeiten, Bahnhöfe, Parkplätze, Badeplätze, ...). Auch die Länge und Dauer der Radstrecke und die Kondition (überdurchschnittlich, durchschnittlich, unterdurchschnittlich) des Nutzers spielen eine Rolle für die Auswahl und Darstellung der Route.

Eine Nutzerrolle definiert einzelne Aufgaben, die im Folgenden beschrieben werden. Die Aufgaben der Rolle *GIS Analyst Radwanderung* ist die 'Analyse' der Geodaten und 'Bereitstellung' der Routen und Geodaten. Ein Nutzer der Rolle *GIS Analyst Radwanderung* führt zu Beginn die Routenberechnung aufgrund einer 'Analyse' verschiedener topographischer Basisdaten (Wälder, Straßen, Wege und Siedlungen), Geländedaten (TIN), Radwegedaten (mit zusätzlichen Attributen wie Zustand und Schwierigkeitsgrad) und POIs mit GIS Analysewerkzeugen auf einem PC mit Breitbandanschluß durch (Abbildung 47 a). Auf dieser Grundlage erzeugt der Nutzer die Routen und klassifiziert diese nach den oben genannten Kriterien (Tourart,...). Die Layerreihenfolge und Signaturierungen können durch den Nutzer geändert werden. Die Nutzer der Rolle *GIS Analyst Radwanderung* haben Werkzeuge wie Layer ein- und auszublenen und einfache Navigation wie Zoom, Pan, ... zur Verfügung. Nach der Erstellung der Routen erfolgt durch den Nutzer der Rolle *GIS Analyst Radwanderung* eine 'Bereitstellung' der Routen und Geodaten zum Beispiel über einen Webdienst.

Die Aufgaben der Rolle *Auskunftssuchender Radwanderung* sind 'Selektion' der Route und 'Anzeige' der Routen und anderer relevanter Geodaten. Ein Nutzer der Rolle *Auskunftssuchender Radwanderung* meldet sich beim Webdienst an und es werden in Abhängigkeit der oben genannten Kriterien mehrere Routen angezeigt. Nach der 'Selektion' durch den Nutzer der Rolle *Auskunftssuchender Radwanderung* kann die Durchführung der Radwanderung mit 'Anzeige' der Route und anderen relevanten Geodaten (topographische Daten, Höhenlinien, POIs) als Hintergrundkarte im Gelände beginnen (Abbildung 47 b). In vordefinierter Layerreihenfolge und Signaturierungen werden die relevanten Informationen wie die Radwanderroute, die Höhenlinien des Geländes und POIs auf einem PDA mit UMTS Verbindung dargestellt. Die Nutzer der Rolle *Auskunftssuchender Radwanderung* verfügen über Werkzeuge wie Layer ein- und auszublenen und einfache Navigation. Ferner können Informationen zu den POIs abgefragt werden.

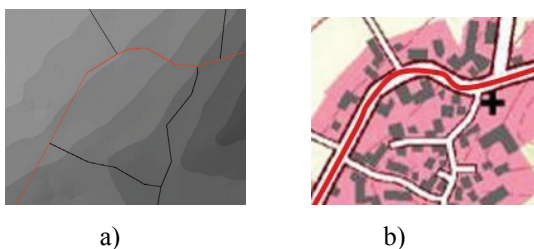


Abbildung 47: Routenplanung (a) und Routenanzeige (b)

Im Weiteren sollen für dieses Szenario Beispiele für eine Änderung der Situation des Nutzers der Rolle *Auskunftssuchender Radwanderung* während der Radwanderung gegeben werden, um die Notwendigkeit einer Anpassung der Präsentation zu zeigen. Es kann sich zum Beispiel während der Radwanderung die Kondition des Nutzers verändern. Er wird müde und eine kürzere Route soll selektiert werden. Dieses Beispiel führt zu einer Änderung der Route. Ein weiteres Beispiel ist eine Notfallsituation. Es entsteht überraschend ein Unwetter. Durch das lichtschwache Display des PDA kann nur sehr schwer zwischen Route und den braunen Höhenlinien unterschieden werden und eine Anpassung der Signaturierung ist notwendig. Die Darstellungsanforderungen haben sich durch die Änderung der Situation des Nutzers gewandelt. Außerdem sind verschiedene Ansichten wie Tag- und Nachtansicht notwendig (Abbildung 48). Wandeln sich die Lichtverhältnisse von Tag zu Nacht, muss sich auch die Präsentation verändern und in diesem Fall zum Beispiel heller und das Display dunkler werden (Anpassung der Bildschirmanzeige). Es wäre wünschenswert, wenn die Graphik auf das Darstellungsmedium bzw. auf die äußeren Einflüsse (helles Umgebungslicht) abgestimmt wäre und wesentliche Informationen "auf einen Blick" identifizierbar wären.



a)

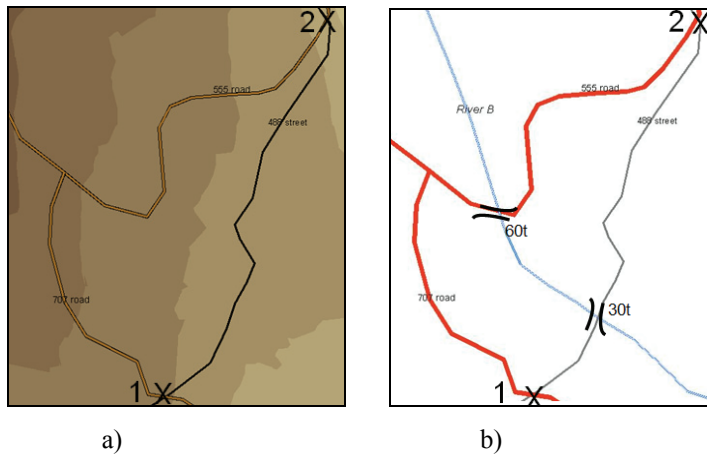
b)

**Abbildung 48: Tag- (a) und Nachtansicht (b) (GPSMAGAZINE (2007))**

#### 4.2.2. Anwendungsszenario Bereich Militär: Marsch

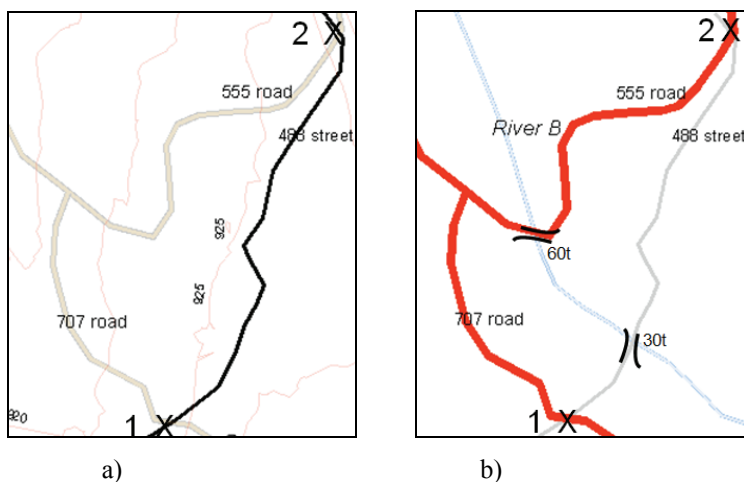
Bei diesem Anwendungsszenario im Bereich Militär müssen Soldaten und schweres Gerät von Punkt 1 zu Punkt 2 transportiert werden. Zu diesem Zweck sind verschiedene kartenähnliche Darstellungen zur Marschplanung zu erstellen. Die Nutzerrollen in diesem Szenario sind *GIS Analyst Marsch*, *Auskunftssuchender Marsch Soldat* und *Auskunftssuchender Marsch Gerät*.

Für die Rollen werden im Folgenden die einzelnen Aufgaben beschrieben. Die Aufgaben der Rolle *GIS Analyst Marsch* ist die 'Analyse' der Geodaten mithilfe von GIS Analysewerkzeugen und 'Bereitstellung' der Tracks und Geodaten. Auf Basis von digitalen Geländemodellen und Vektordaten wird die Marschplanung für Soldaten erstellt. Hierfür sind Anforderungen wie erreichbare Tracks und so wenig vertikale Höhendifferenzen wie möglich zu erfüllen. Aus diesem Grund zeigt die erste Karte in Abbildung 49 a den Start- und Endpunkt (1 & 2), die möglichen Tracks (schwarze und gelbbraune Linien) und die braunschattierten Höhenunterschiede, um eine Situationsanalyse zu unterstützen.



**Abbildung 49: Marschplanung Soldat (a) und Marschplanung Gerät (b)**

Um eine Marschplanung mit schwerem Gerät durchführen zu können, müssen Geodatenobjekte wie mögliche Tracks (rot und grau), mögliche Hindernisse wie Flüsse (blau) und Brücken (schwarz) und ihre maximale Auslastung in einer Karte dargestellt werden (Abbildung 49 b). Auch Darstellungen der Ressourcen, Einheiten und sicheren und unsicheren Zonen geben Unterstützung bei der Planung der Tracks. Die Tracks werden durch den Nutzer der Rolle *GIS Analyst Marsch* auf einem PC mit Satellitenverbindung erstellt und zum Beispiel über einen Webdienst zusammen mit weiteren Geodaten bereitgestellt. Die Layerreihenfolge und Signaturierungen können durch den Nutzer der Rolle *GIS Analyst Marsch* geändert werden. Es können Layer ein- und ausgeblendet und einfache Navigation in der Karte durchgeführt werden.



**Abbildung 50: Anzeige „Marsch Soldat“ (a) und „Marsch Gerät“ (b)**

Die Aufgaben der Rolle *Auskunftssuchender Marsch Soldat* ist die ‘Anzeige‘ des ausgewählten Tracks für den Marsch der Soldaten (schwarz) und die notwendigen Geodaten (verschiedene thematische Informationen, Höhenlinien (rot) und Standorte (1 & 2)) wie in Abbildung 50 a dargestellt über eine Satellitenverbindung. Die Aufgaben der Rolle *Auskunftssuchender Marsch Gerät* ist die ‘Anzeige‘ des zu verwendenden Tracks (rot) wie in Abbildung 50 b dargestellt mit mögliche Hindernisse und Brücken mit ihrer ma-

ximalen Auslastung. Diese werden in vordefinierter Layerreihenfolge und Signaturierungen auf verschiedenen Geräten wie PC Monitor oder TabletPC angezeigt. Die jeweiligen zu verwendenden Tracks für den Marsch werden durch eine geringere Transparenz als die weiteren Geodaten hervorgehoben. Die Nutzer der Rolle *Auskunftssuchender Marsch Soldat* und Rolle *Auskunftssuchender Marsch Gerät* verfügen über Werkzeuge wie Layer ein- und auszublenden und einfache Navigation wie Zoom, Pan, ....

Nach der Beschreibung der Szenarien ist es möglich, für die einzelnen Nutzerrollen ihre Anforderungen an Geodatenauswahl, Anpassung der Signaturierung, Bildschirmanzeige und Auswahl der Werkzeuge des Systems zu definieren. Die Anforderungen werden im nächsten Abschnitt beschrieben.

### 4.2.3. Nutzerrollen

Es wurden im Bereich Radwanderung die Nutzerrollen „Auskunftssuchender Radwanderung (View bicycle)“ und „GIS Analyst Radwanderung (GIS analysis bicycle)“ definiert. In Tabelle 5 werden die Geodatenauswahl, die Signaturierung/Bildschirmanzeige und die notwendigen Werkzeuge der einzelnen Nutzerrollen im Bereich Radwanderung dargestellt.

**Tabelle 5: Nutzerrolle – Geodaten, Signaturierung/Bildschirmanzeige und Werkzeuge, Radwanderung**

Nutzerrolle	Geodaten	Signaturierung/Bildschirmanzeige	Werkzeuge
Auskunftssuchender Radwanderung (View bicycle)	Verschiedene thematische Informationen (Siedlung, Vegetation, ...) nach ATKIS Objektbereichen (Bsp. Siedlung 2000), -klassen (Bsp. Baulich geprägte Flächen 2100) und -arten (Bsp. Wohnbaufläche 2111) Höhenlinien des Geländes Routen abhängig von Kriterien POIs	Signaturierung der Geodaten abhängig vom Objekttyp (punkt-, linien-, flächenförmig) Transparenz Darstellungsänderung bei verschiedenen Lichtverhältnissen (drinnen/draußen) inkl. Änderung der Bildschirmanzeige (helleres/dunkleres Display) Darstellung auf verschiedenen Geräten UMTS: 384 kbit/s	Vordefinierte Signaturierung Vordefinierte Layerreihenfolge Ein- und Ausblenden von Layern Einfache Navigationswerkzeuge (Zoom, Pan, ...) Abfrage von Informationen wie nächster Rastplatz (POIs) Suche und Selektion einer Radwandererroute  Rechte: Leserechte Geodaten und Routen
GIS Analyst Radwanderung (GIS analysis bicycle)	Verschiedene thematische Informationen nach Objektklassen und -arten Digitales Geländemodell	Signaturierung abhängig vom Objekttyp (punkt-, linien-, flächenförmig) Attributabhängige Signaturierung von Objekten (Bsp. Wohnbauflä-	Änderung der Signaturierung Änderung der Layerreihenfolge Ein- und Ausblenden von



	Radwege POIs	che 2111 BEB=1000) Transparenz Highlighting Darstellung auf PC Bildschirm DSL6000	Layern Einfache Navigationswerkzeuge und GIS Analysewerkzeuge für die Routenplanung Erzeugung von Routen (Analyseergebnisse) Bereitstellung der Routen und Geodaten  Rechte: Schreibrechte Routen, Leserechte Geodaten
--	-----------------	---	---

Es wurden im Bereich Militär die Nutzerrollen „Auskunftssuchender Marsch Soldat (View march soldier)“, „Auskunftssuchender Marsch Gerät (View march equipment)“ und „GIS Analyst Marsch (Operational commander C2 march)“ definiert (Kap. 4.2.2). Die Geodaten, Signaturierung/Bildschirmanzeige und Werkzeuge der Nutzerrolle für den Marsch sind in Tabelle 6 dargestellt.

**Tabelle 6: Nutzerrollen – Geodaten, Signaturierung/Bildschirmanzeige und Werkzeuge, Marsch**

Nutzerrollen	Geodaten	Signaturierung/Bildschirmanzeige	Werkzeuge
Auskunftssuchender Marsch Soldat (View march soldier)	Verschiedene thematische Informationen (Siedlung, Vegetation, ...) nach Objekthaupt- und -nebenkategorie GeoInfo Objektartenkatalog (Bsp. A Künstliche Bodenbedeckung, Bsp. AL Verschiedenartige Objekte), Objektarten (Bsp. Siedlung AL 105), hauptsächlich Straßenobjekte  Höhenlinien des Geländes  Tracks  Standorte	Signaturierung der Geodaten abhängig vom Objekttyp (punktförmig, linienförmig, flächenförmig)  Attributabhängige Signaturierung (Bsp. Siedlung AL 105 PPT=000)  Tracks hervorgehoben  Transparenz  Darstellungsänderung bei verschiedenen Lichtverhältnissen (drinnen/draußen) inkl. Änderung der Bildschirmanzeige (helleres/dunkleres Display)  Darstellung auf verschiedenen Geräten  Satellitenübertragung entspricht DSL1000	Vordefinierte Signaturierung  Vordefinierte Layerreihenfolge  Ein- und Ausblenden von Layern  Einfache Navigationswerkzeuge (Zoom, Pan, ...)  Abfrage von Informationen (Geobasisdaten, Fachdaten, Analyseergebnisse)  Rechte: Leserechte Geodaten und operationelle Daten wie Tracks

<p>Auskunftssuchender Marsch Gerät (View march equipment)</p>	<p>Verschiedene thematische Informationen (Siedlung, Vegetation, ...) nach Objekthaupt- und –nebenkategorie GeoInfo Objektartenkatalog (Bsp. A Künstliche Bodenbedeckung, Bsp. AL Verschiedenartige Objekte), Objektarten (Bsp. Siedlung AL 105), hauptsächlich Straßenobjekte, mögliche Hindernisse und Brückenobjekte mit ihrer maximaler Auslastung</p> <p>Tracks Standorte</p>	<p>Signaturierung der Geodaten abhängig vom Objekttyp (punktförmig, linienförmig, flächenförmig)</p> <p>Attributabhängige Signaturierung (Bsp. Siedlung AL 105 PPT=000)</p> <p>Tracks hervorgehoben</p> <p>Transparenz</p> <p>Darstellungsänderung bei verschiedenen Lichtverhältnissen (drinnen/draußen) inkl. Änderung der Bildschirmanzeige (helleres/dunkleres Display)</p> <p>Darstellung auf verschiedenen Geräten</p> <p>Satellitenübertragung entspricht DSL1000</p>	<p>Vordefinierte Signaturierung</p> <p>Vordefinierte Layerreihenfolge</p> <p>Ein- und Ausblenden von Layern</p> <p>Einfache Navigationswerkzeuge (Zoom, Pan, ...)</p> <p>Abfrage von Informationen (Geobasisdaten, Fachdaten, Analyseergebnisse)</p> <p>Rechte: Leserechte Geodaten und operationelle Daten wie Tracks</p>
<p>GIS-Analyst Marsch (Operational commander C2 march)</p>	<p>Verschiedene thematische Informationen nach Objektklassen und –arten</p> <p>Digitales Geländemodell</p> <p>Standorte</p> <p>Ressourcen, Einheiten und Zonen (sicher, unsicher)</p>	<p>Signaturierung abhängig vom Objekttyp (punkt-, linien-, flächenförmig)</p> <p>Transparenz</p> <p>Highlighting</p> <p>Signaturierungen von Ressourcen, Einheiten und Zonen (sicher, unsicher)</p> <p>Darstellung auf PC Bildschirm</p> <p>Satellitenübertragung entspricht DSL1000</p>	<p>Änderung der Signaturierung</p> <p>Änderung der Layerreihenfolge</p> <p>Ein- und Ausblenden von Layern</p> <p>Einfache Navigationswerkzeuge und GIS Analysewerkzeuge für die Planung der Tracks (Sichtbarkeitsanalysen)</p> <p>Fachdatenerzeugung von Tracks (Analyseergebnisse)</p> <p>Bereitstellung der Tracks und Geodaten</p> <p>Rechte: Schreibrechte operationelle Daten, Leserechte Geodaten</p>

Aus der Tabelle 5 und Tabelle 6 ergeben sich für die Nutzerrollen neben einer eindeutigen NutzerID für jeden Nutzer der Nutzerrolle verschiedene Kontextelemente (Anwendungsbereich, genutztes System und die Umgebung des Nutzers dieser Nutzerrolle). Diese sind in Tabelle 7 aufgelistet.

**Tabelle 7: Nutzerrollen und ihre Eigenschaften**

Bereich	Nutzerrolle	System	Umgebung: Situation / Physikalische Konditionen
Freizeit	Auskunftssuchender Radwanderung (View bicycle)	Verschiedene Geräte wie PDA, mobil (drinnen/draußen), UMTS: 384 kbit/s	Position, Datum, Tageszeit, Lichtverhältnisse, Wetter
Freizeit	GIS Analyst Radwanderung (GIS analysis bicycle)	PC, stationär, Bandbreite, DSL 6000	-
Militär	Auskunftssuchender Marsch Soldat (View march soldier)	Verschiedene Geräte wie PDA oder TabletPC, mobil (drinnen/draußen), Satellitenübertragung	Position, Datum, Tageszeit, Lichtverhältnisse, Wetter
Militär	Auskunftssuchender Marsch Gerät (View march equipment)	Verschiedene Geräte wie PDA oder TabletPC, mobil (drinnen/draußen), Satellitenübertragung	Position, Datum, Tageszeit, Lichtverhältnisse, Wetter
Militär	GIS Analyst Marsch (Operational commander C2 march)	PC, stationär, Satellitenübertragung	-

Diese Kontextelemente müssen zur einheitlichen Verwendung in einem Modell definiert werden. Das Kontextmodell, welches die Nutzerrollen der beiden Szenarios unterstützen soll, wird im Anschluss dargestellt.

### 4.3. Kontext

In diesem Kapitel wird das Kontextmodell definiert (4.3.1). In 4.3.2 wird der Einfluss des Kontextmodells auf die Geodatenauswahl, Signaturierung, Bildschirmanzeige und Werkzeuge des Systems erläutert. Hierfür werden folgende Definitionen festgelegt:

**Definition 4.2 (Kontextabhängiges Portrayalmodell)** *Das kontextabhängige Portrayalmodell definiert die Portrayalregeln und Signaturen in Abhängigkeit vom Kontext. Es bildet die Basis, um auf Grundlage des Kontextes die Signaturierung zu erhalten.*

Der Kontext wird verwendet, um Geodaten kontextabhängig auszuwählen und zusammen mit dem kontextabhängigen Portrayalmodell eine kontextabhängige Darstellung zu erzeugen.

**Definition 4.3 (Primärkontext)** *Der Primärkontext wird im Kontextmodell (Definition 4.4) definiert.*

*Beispielelemente sind Rolle und System des Nutzers.*

**Definition 4.4 (Kontextmodell)** *Das Kontextmodell beschreibt verschiedene Nutzer und ihre spezifische Situation.*

**Definition 4.5 (Sekundärkontext)** *Der Sekundärkontext ist abhängig vom Primärkontext bzw. wird davon abgeleitet. Zum Beispiel können Präferenzen des Benutzers wie bestimmte Geoobjekte aus seinem Bereich und seinen Rollen über Listen im Sekundärkontext abgeleitet werden.*

### 4.3.1. Primärkontext: Modellierungsebenen Kontextmodell

Im Folgenden wird das Kontextmodell in verschiedenen Ebenen definiert. Das konzeptionelle Modell und somit die Elemente, ihre Beziehungen und Funktionen werden in einem UML Klassendiagramm beschrieben. Das logische Modell definiert eine Abbildung in Datenstrukturen in einem XML Schema (XSD Format). Die Datenbankstruktur wird am Beispiel von einer Oracle-DB in Kapitel 6 beschrieben.

#### *Konzeptionelles Modell: UML – Klassendiagramm*

Das konzeptionelle Schema spezifiziert die sachlich-logische Ebene und ist Grundlage für den Entwurf der übrigen Schemata. Das konzeptionelle Schema beschreibt die Struktur der gespeicherten Daten. Wie zuvor erläutert, ist es für eine kontextbasierte Anpassung der Präsentation wichtig, den Kontext zu modellieren. Allgemein können laut Kapitel 3.1.2 die Kontextelemente in *technischen Kontext* (wie Netzwerkverbindungen, ...), *persönlichen Kontext* (wie soziale Verhältnisse), *physikalischen Kontext* (wie Temperatur, Lichtverhältnisse etc.) und *zeitlichen Kontext* (wie Tageszeit, Datum) beschrieben werden.

In diesem Abschnitt wird ein spezifisches Kontextmodell (mit verschiedenen Aspekten) anhand zweier Szenarien definiert. Dies basiert auf den Modelldefinitionen von Ziegler, Lohmann und Kaltz, Sarjakoski und Nivala und Reichenbacher (Kap. 3). Im Anschluss wird das Kontextmodell mithilfe der Anwendungsszenarien aus Kapitel 4.2 überprüft und weiterentwickelt bzw. vervollständigt mit dem Ziel das Modell zu verbessern.

ZIEGLER et al. (2005) definieren für eine adaptive Webanwendung fünf Kategorien von Kontext (Kap. 3.1.2). NIVALA and SARJAKOSKI (2003) entwickelten eine Klassifikation für Kontext für mobile Dienste (Kap. 3.2). REICHENBACHER fokussiert sich auf die ausführliche Darstellung einer Adaptionmethode für die Visualisierung von geographischen Informationen für die mobile Nutzung (REICHENBACHER (2004)) (Kap. 3.2). Die Modellierung verschiedener Kontextbereiche unter Berücksichtigung verschiedener Szenarios wurde jedoch nicht behandelt. Des Weiteren existiert keine Konformität zu standardisierten Webdiensten. Der Ansatz von Reichenbacher kann aus diesem Grund nur als Basis für die Anwendungsszenarien verwendet werden. Für die Erweiterung und Konkretisierung des Kontextmodells sind Erweiterungen für verschiedene Bereiche und deren Rollen notwendig. Es müssen spezifische Anforderungen berücksichtigt werden. Dies sind zum einen die typischen spezifischen Aufgaben für die Nutzergruppen und zum anderen muss das Modell für die Nutzung in standardisierten Portrayalwebdiensten geeignet sein, wobei nur bestimmte Anpassungen möglich und sinnvoll sind. Die zuvor dargestellten Modelle sind für die Ziele der Arbeit nicht ausreichend und sie müssen zu einem Gesamtmodell erweitert werden. In Tabelle 8 werden diese Kontextmodelle gegenübergestellt und in einem eigenen Gesamtmodell basierend auf Tabelle 7 zusammengefasst. Die Elemente des Gesamtmodells werden im Anschluss durch Unterele-

mente verfeinert.

**Tabelle 8: Modell Ziegler, Nivala und Sarjakoski, Reichenbacher und eigenes Gesamtmodell**

Ziegler, Lohmann und Kaltz	Nivala und Sarjakoski	Reichenbacher	Gesamtmodell
Benutzer & Rolle	Purpose of use, User, Social, Cultural	Nutzer (Identität oder Nutzergruppe)	Nutzeridentität (ID), Bereich
Aufgabe (Arbeitsaufgaben, persönliche Zielsetzungen)		Nutzeraktivität (Aktion und Ziel)	Rolle
Ort und Zeit	Location	Nutzersituation: Position als Koordinatenpaar, Ortsname oder Adresse	Situation
	Time	Zeit: exakte Zeit, Zeitintervall oder Tageszeit	
Gerät	System	Nutzersystem: technology (wie Gerät, Netzwerktyp und Bandbreite)	System
-	Physical surroundings	Information, die den Nutzer und das Nutzersystem umgibt wie Wetter, ...	Physikalische Konditionen
-	Navigation History	-	-

Das Gesamtmodell muss in statische und dynamische Elemente klassifiziert werden. Die statischen Elemente ändern sich normalerweise nicht in einer Anwendung, wobei die dynamischen Elemente immer wieder aktualisiert werden müssen und der kontextabhängige Portrayalprozess erneut erfolgen muss. Desweiteren beschreibt dieses Modell den sogenannten Primärkontext (Definition 4.2). Der Sekundärkontext (Definition 4.4) definiert hingegen Elemente, die sich aus dem Primärkontext ableiten lassen. Im Folgenden werden die statischen und dynamischen Elemente des Kontextmodells definiert und die einzelnen Elemente des Gesamtmodells aus Tabelle 8 weiter verfeinert.

Das Kontextmodell basiert auf der Basis der zuvor identifizierten Anforderungen. Der Nutzer hat einen „Anwendungsbereich“ und bestimmte Aufgaben definiert in „Rollen“. Das „System“ spielt ebenfalls eine wichtige Rolle und deren Eigenschaften wie Bildschirmgröße oder Auflösung werden berücksichtigt. Die Informationen zur Situation in der sich der Nutzer befindet beinhalten zum Beispiel das Wetter (Nebel). Außerdem existieren „physikalische Konditionen“ wie Licht (Tag/Nacht). Wenn dem Nutzer nur eine kleine Bandbreite zur Verfügung steht, dann sollten nur relevante Geodaten an den Nutzer geschickt und visualisiert werden. Die dynamischen Elemente können sich im Gegensatz zu den statischen im Laufe einer Anwendung immer wieder verändern. Das System ist hierbei ein Sonderfall, da es sowohl zu den statischen als auch zu den dynamischen Elementen gehören könnte, da sich das System während einer Anwendung ändern könnte. Dies ist jedoch selten der Fall und aus diesem Grund ist das Element „System“

den statischen Elementen zugeordnet. Die Elemente des Kontextmodells sind in WIEBROCK and REINHARDT (2008b) definiert und in Tabelle 9 (statische Elemente) und Tabelle 10 (dynamische Elemente) zusammengestellt.

**Tabelle 9: Statische Elemente**

Element	Definition	Beispiel
ID	ID zur Identifikation des Nutzers.	123
Bereich	Gebiet des Anwendungsbereiches und der Interessen. Dies sind in diesem Fall Freizeit und Militär.	Militär
Rolle	Spezifiziert die Nutzerrolle die mit einer bestimmten Aufgabe verbunden sind.	Auskunftsuchender Marsch Soldat
System	Parameter welche genutzt werden um die Geodaten zu visualisieren. Dies sind Systemnutzung, Bildschirmtyp, -auflösung, -größe und Bandbreite.	Nutzung: Mobil Gerät: TabletPC Netzwerk: Satellitenübertragung

**Tabelle 10: Dynamische Elemente**

Element	Definition	Beispiel
Situation	Standort des Nutzers (Position) und Zeit.	9,5 51,3 12:00:00
Physikalische Konditionen	Mittel- und kurzweilig, beeinflusst von Wetter und Lichtverhältnissen.	Tag Sonnig

Es ist wichtig, ein passendes Modell zu finden, welches die Beschreibung der Elemente erlaubt und gleichzeitig erweiterbar ist. Das verwendete Modell basiert auf dem Ansatz von SAIDANI and NURCAN (2007), welches im Kapitel 3.1.2 beschrieben wurde. Sie definieren einen dreidimensionalen Raum (space)  $S = \langle \text{ASPECTS}, \text{FACETS}, \text{ATTRIBUTES} \rangle$ . Der Kontext wird in verschiedene Elemente (aspects) unterteilt und ist identisch mit den zuvor definierten statischen und dynamischen Elementen. Facets werden mit Attributen beschrieben. Die folgende Tabelle 11 zeigt die Kontextelemente basierend auf Tabelle 9 und Tabelle 10. Die Wertebereiche der Attribute sind aus eigenen Überlegungen entstanden.

**Tabelle 11: Kontextelemente**

Aspect	Facets	Attributes
User (Nutzer)	Identification (Identifizierung)	ID
	Preferences (Präferenzen)	Bereich - domain (leisure, mili) Rolle - role (View bicycle, GIS Analysis bicycle, View March soldier, View March equipment, Operational commander march)
System	Device- hardware (Gerät)	Systemnutzung – system use (mobile, stationary)
		Ausgabemedium - Device type (PC, TabletPC, Note-

		book, PDA)
		Bildschirmgröße - Size (PC: 15''-40'', TabletPC: 7''-14'', Notebook: 12,1''-20,1'', PDA: 2,6''-5'')
		Auflösung - Resolution (PC: 800x600 – 1024x768, TabletPC: 800x600 - 1400x1050, Notebook: 800x600 - 1680x1050, PDA: 320x240, 480x320, 640x480)
	Band width (Bandbreite)	Analog (56 Kbit/s), ISDN (64 Kbit/s), UMTS (384 kbit/s), DSL 1000 (1024 Kbit/s), DSL 2000 (2048 Kbit/s), DSL 3000 (3072 Kbit/s), DSL 6000 (6016 Kbit/s), DSL 16000 (16000 Kbit/s)
Environment (Umgebung) (dynamic)	Situation – position (Standort)	Koordinaten - Coordinate : Sequence<Number>
	Situation – time (Zeit)	Datum - Date (YYYYMMDD)
		Tageszeit- Time of the day (hhmmss)
	Physical (Physikalisch)	Lichtverhältnisse - Light condition (day, night, twilight)
Wetter - Weather (rain, sun, fog, snow, clouds)		

Neben den beschriebenen Anpassungen kann das graphische Nutzerinterface adaptiert werden. Die Funktionen und die Größe, Anzahl und Orientierung (Quer-, Längsformat) der Fenster können abhängig von den Nutzerrollen angepasst werden. Dies wird jedoch in der weiteren Arbeit nicht berücksichtigt.

Die zuvor definierten Elemente aus Tabelle 11 bieten die Basis für das UML Kontextmodell (Abbildung 51). UML ist die Datenbeschreibungssprache für internationale GI-Normen. ISO 19103 - *Conceptual schema language* definiert Regeln und eine Anleitung für die Nutzung der konzeptuellen Schemasprache UML für die ISO 19100 Serie. Die Notationen finden sich in Anhang 12.5.

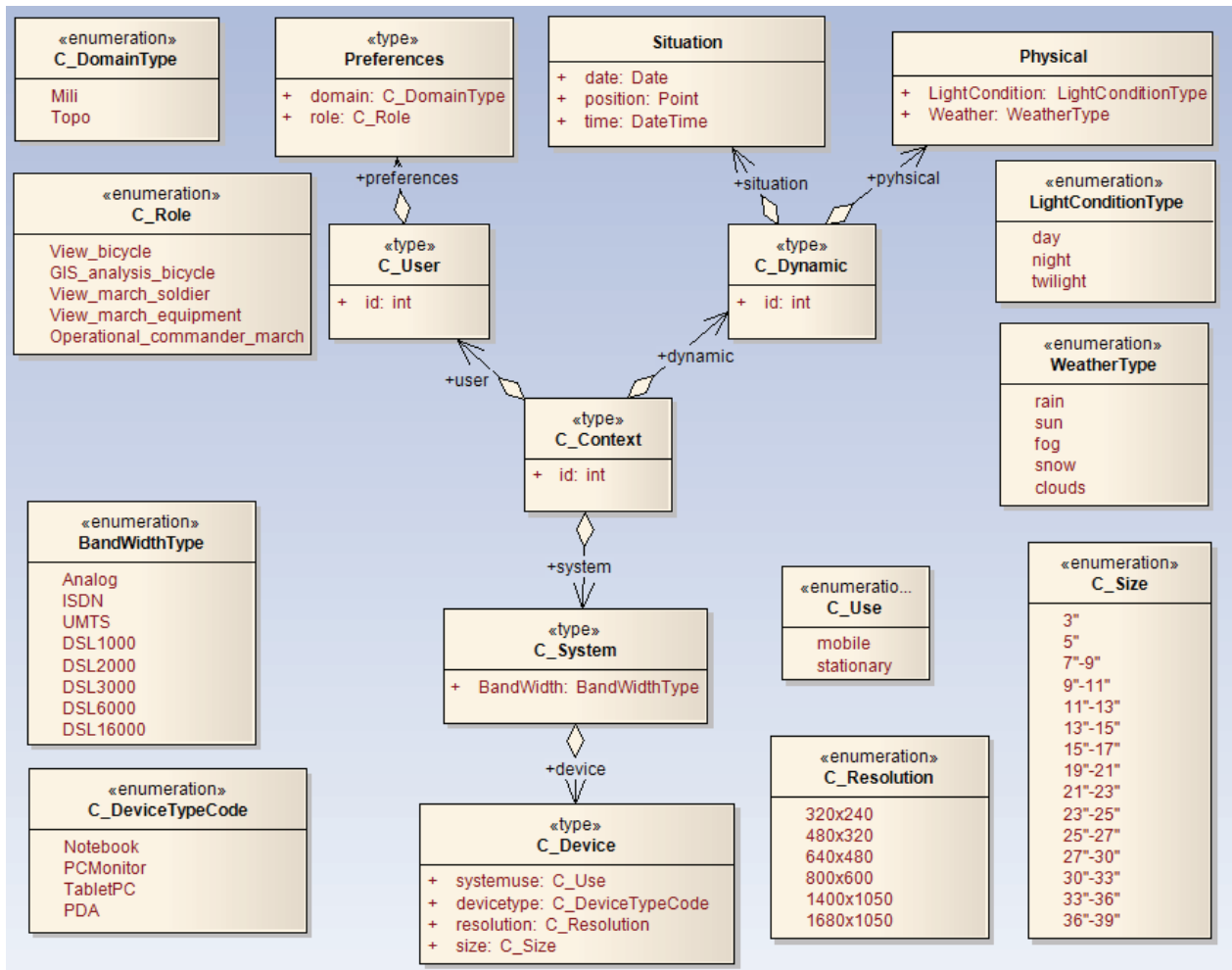


Abbildung 51: Kontextmodell in UML

### Logisches Modell: Modellierung in XML Schema (XSD)

Das zuvor beschriebene Konzept wird in *XML schema* (XSD) als logisches Modell implementiert, um Nutzergruppenmodellinformationen in einer Webumgebung verwenden zu können. Der Vorteil von XML Schema ist die Verwendung der XML Syntax, die Erweiterbarkeit und die Unterstützung verschiedener Datentypen (wie Integer, Zeichenketten, ...). Außerdem kann mit definierten Nutzerprofilen eine Schemavalidierung durchgeführt werden, um zu erkennen ob das Profil Gültigkeit besitzt. Hierfür kann nicht der OGC WMC Standard verwendet werden. Es werden im WMC-Dokument Elemente des WMS gespeichert, die wiederverwendet werden können. Für eine Berücksichtigung der zuvor definierten Kontextelemente müsste das WMC-Dokument (Kapitel 2.3.5) erweitert werden. Es gibt laut Standard ein „Extension“ Element, welches jedoch nur herstellerspezifische Informationen beinhalten kann und laut Standard keine neuen Elemente enthalten sollte. In Tabelle 12 wird das XML Schema für die Kontextinformationen tabellarisch dargestellt.



**Tabelle 12: XML Schema (Kontext)**

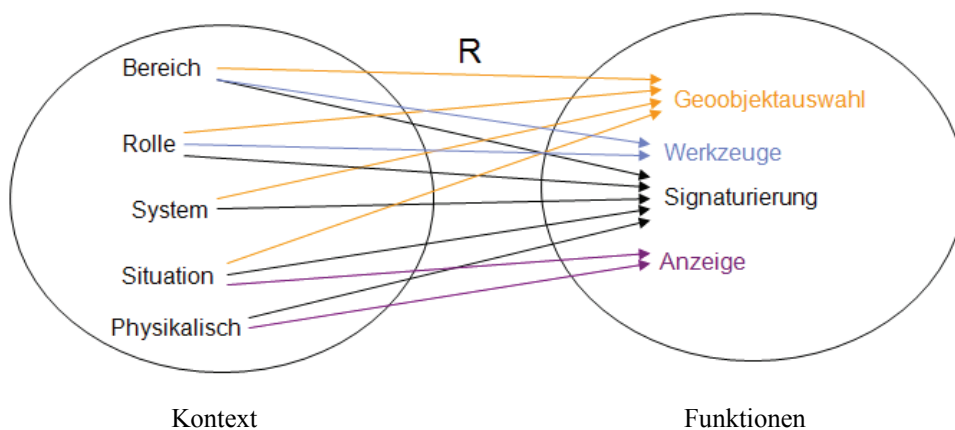
Typen	Elemente
ComplexType „C_Context“	Name="id", type="integer" Name="dynamic", type="C_Dynamic" Name="system", type="C_System" Name="C_User", type="C_User"
ComplexType „C_Dynamic“	Name="id", type="integer" Name="Situation", type="Situation" Name="Physical", type="Physical"
ComplexType „C_System“	Name="BandWidth", type="BandWidthType" Name="C_Device", type="C_Device"
ComplexType „C_User“	Name="id", type="integer" Name="Preferences", type="Preferences"
ComplexType „C_Device“	Name="devicetype", type="C_DeviceTypeCode" Name="resolution", type="C_Resolution" Name="size", type="C_Size" Name="systemuse", type="C_Use"
ComplexType „Preferences“	Name="domain", type="C_DomainType" Name="role", type="C_Role"
ComplexType „Situation“	Name="date", type="xs:date" Name="position", type="gml:PointType" Name="time", type="xs:dateTime"
ComplexType „Physical“	Name="LightCondition", type="LightConditionType" Name="Weather", type="WeatherType"
Simple Type „C_DeviceTypeCode“	Enumeration values="Notebook, PCMonitor, TabletPC, PDA"
Simple Type „C_Resolution“	Enumeration values="320x240, 480x320, 640x480, 800x600, 1400x1050, 1680x1050"
Simple Type „C_Role“	Enumeration values="GIS_analysis_bicycle, View_bicycle, View_march_soldier, View_march_equipment, Operational_commander_march"
Simple Type „BandWidthType“	Enumeration values="Analog, ISDN, UMTS, DSL1000, DSL2000, DSL3000, DSL6000, DSL16000"
Simple Type "C_DomainType"	Enumeration values ="Mili, leisure"
Simple Type "LightConditionType"	Enumeration values ="day, night, twilight"
Simple Type "WeatherType"	Enumeration values ="rain, sun, fog, snow, clouds"
Simple Type "C_Size"	Enumeration values ="3", 5", 7"-9", 9"-11""", 11"-13", 13"-15", 15"-17", 19"-21", 21"-23", 23"-25", 25"-27", 27"-30", 30"-33", 33"-36", 36"-39"""
Simple Type "C_Use"	Enumeration values ="mobile, stationary"

Das vollständige XML Schema und eine XML-Beispieldatei, die dem XSD Schema folgt ist im Anhang 12.1 zu finden.

### 4.3.2. Einfluss des Kontextmodells auf die Geobjektauswahl, Signaturierung, Bildschirmanzeige und Werkzeuge

Der Einfluss des Kontextmodells auf die Funktionen zur Erstellung einer kontextabhängigen Visualisierung mit Werkzeugauswahl wird in diesem Kapitel zusammengefasst. Die Auswahl der Funktionen (Geobjektauswahl, Signaturierung, Bildschirmanzeige und Werkzeuge) ist ein Resultat der Anwendungsszenarien (4.2). Um diese Relationen zu definieren, wird der Ansatz von SCHMIDT and GELLERSEN (2001) (Kap. 3.1.2) auf den verwendeten Kontext angewendet. Sie definieren eine Anwendung  $A = (\Omega, \Theta, R)$ .

Die Aspekte des Kontextmodells sind  $k_1 = \text{Bereich}$ ,  $k_2 = \text{Rolle}$ ,  $k_3 = \text{System}$ ,  $k_4 = \text{Physikalisch}$  und  $k_5 = \text{Situation}$ . Die Funktionen sind  $f_1 = \text{Geobjektauswahl}$ ,  $f_2 = \text{Signaturierung}$ ,  $f_3 = \text{Anzeige}$  und  $f_4 = \text{Werkzeuge}$  (Abbildung 52). Der Weg wie die Aspekte des Kontextmodells die Funktionen beeinflussen fehlt in SCHMIDT and GELLERSEN (2001). Auf Grundlage der Relationen werden die Adaptionenregeln und der kontextabhängige Portrayalprozess mit Werkzeugauswahl im nächsten Kapitel beschrieben.



**Abbildung 52: Relationen zwischen Kontextmodell und Funktionen**

Die Relationen zwischen Kontextmodell und Funktionen sind:

$R = \{(\text{Bereich}, \text{Geobjektauswahl}), (\text{Bereich}, \text{Signaturierung}), (\text{Bereich}, \text{Werkzeuge}),$

$(\text{Rolle}, \text{Geobjektauswahl}), (\text{Rolle}, \text{Signaturierung}), (\text{Rolle}, \text{Werkzeuge}),$

$(\text{System}, \text{Geobjektauswahl}), (\text{System}, \text{Signaturierung}),$

$(\text{Situation}, \text{Geobjektauswahl}), (\text{Situation}, \text{Signaturierung}), (\text{Situation}, \text{Anzeige}),$

$(\text{Physikalisch}, \text{Signaturierung}), (\text{Physikalisch}, \text{Anzeige})\}$

Die Relationen sind mangels vorhandener Forschungsergebnisse aus der Literatur durch eigene Überle-

gungen entstanden. Sie werden in Tabelle 13 explizit durch Beispiele begründet.

**Tabelle 13: Relationen und Begründungen mit Beispielen**

Relation	Begründung/Beispiel
Bereich, Geobjektauswahl	In vielen Fällen stimmen die benötigten Geobasisdaten (Vegetation, Gewässer, ...) für verschiedene Bereiche überein. Jedoch werden vor allem unterschiedliche spezifische Geodaten in den Anwendungsbereichen benötigt, zum Beispiel für den Bereich „Topo“ Radtouren und für „Militär“ Tracks.
Bereich, Signaturierung	Jeder Anwendungsbereich hat seine eigenen Signaturenkataloge. Im Bereich „Topo“ ist zum Beispiel „ATKIS SK“ und beim Militär „GeoSym“ zu nennen (s. Kap. 2.2.1).
Bereich, Werkzeuge	Jeder Anwendungsbereich nutzt spezielle Werkzeuge. Im Bereich „Topo“ werden eher allgemeine Werkzeuge wie zum Beispiel „kürzeste Verbindung“ verwendet. Im Bereich „Militär“ sind es spezielle GIS-Werkzeuge für Geländeberechnungen (wie Sichtlinie) oder Missions- und Logistikplanungen.
Rolle, Geobjektauswahl	Je nach Rolle wird eine andere Geobjektauswahl getroffen. Der „Auskunftssuchende Marsch Soldat“ verwendet hauptsächlich Straßenobjekte. Der „Auskunftssuchende Marsch Gerät“ hingegen benötigt neben Straßenobjekten auch mögliche Hindernisse und Brückenobjekte.
Rolle, Signaturierung	Die Signaturierung wird durch die Rolle beeinflusst. Der „GIS Analyst Radwanderung“ verwendet zum Beispiel „ATKIS-SK“, wohingegen die Signaturierung des „Auskunftssuchenden Radwanderung“ für die Radwanderung auf Fernwirkung abgestimmt sein muss und deshalb größere Signaturen verwendet werden.
Rolle, Werkzeuge	Der „Auskunftssuchende Radwanderung“ benötigt einfache Navigationswerkzeuge (Zoom, Pan, ...) oder Ein- und Ausblenden von Layern. Der Nutzer „GIS-Analyst Radwanderung“ hingegen verwendet neben einfachen Navigationswerkzeugen auch GIS Analysewerkzeuge wie z. B. „Sichtbarkeitsanalysewerkzeuge“ für die Planung der Tracks.
System, Geobjektauswahl	Die Geobjektauswahl hängt vom System ab. Wenn ein PDA als Darstellungsmedium verwendet wird („Auskunftssuchender Radwanderung“) werden weniger Geodaten als bei einem größerem Bildschirm („GIS-Analyst Radwanderung“) ausgewählt, da der Bildschirm über weniger Darstellungsfläche verfügt.
System, Signaturierung	Die Signaturierung variiert je nach System. So kann zum Beispiel ein PDA („Auskunftssuchender Radwanderung“) keine komplexen Symbole in hoher Auflösung darstellen wie ein größerer Bildschirm („GIS-Analyst Radwanderung“).
Situation, Geobjektauswahl	Durch die Änderung der Position des Nutzers (Situation) wie „Auskunftssuchender Radwanderung“ verändert sich ebenfalls die Geobjektauswahl in seiner näheren Umgebung.
Situation, Signaturierung	Durch die Änderung der Zeit (Situation) kann sich die Beleuchtung ändern (Tag/Nacht) und eine andere Signaturierung wird notwendig („Auskunftssuchender Radwanderung“).
Situation, Anzeige	Durch die Änderung der Zeit (Situation) und somit der Beleuchtung muss der Bildschirm zum Beispiel am Tag abgedunkelt werden („Auskunftssuchender Radwanderung“).
Physikalisch, Signaturierung	Wetterbedingt (Physikalisch) zum Beispiel durch Nebel kann sich die Beleuchtung ändern und eine andere Signaturierung wird notwendig („Auskunftssuchender Radwanderung“).
Physikalisch, Anzeige	Neben der Änderung der Signaturierung kann wetterbedingt (Physikalisch) der Bildschirm abdunkelt werden („Auskunftssuchender Radwanderung“).

Auf Basis der verschiedenen Einflüsse der Kontextelemente auf die Geodaten, Signaturierung, Bildschirmanzeige und Werkzeuge wird im Folgenden festgelegt, welche Informationen zusätzlich benötigt werden, um eine kontextbasierte Darstellung erzeugen zu können.

Die Elemente Bereich, Rolle, System und Situation beeinflussen die Geodatenauswahl (Abbildung 52). Für die Elemente Bereich, Rolle und System des Nutzers wird eine Objektauswahlliste benötigt. Aus der Situation (Position) kann der Ausschnitt als BoundaryBox (BBox) für die Geodaten abgeleitet werden. Es werden für die Geodatenauswahl folgende Informationen benötigt:

- Listen mit Objektauswahl abhängig von Bereich, Rolle und System
- BBox

Die Signaturierung ist von den Elementen Bereich, System, Situation, Physikalisch und der Rolle abhängig (Abbildung 52). Für die Elemente Bereich, System, Situation und Physikalisch muss das Portrayalmodell aus Kapitel 2 erweitert werden. Für die Rolle werden einzelne Änderungsanweisungen benötigt. Diese führen zu Änderungen einzelner Darstellungsregeln. Es werden für die Anpassung der Signaturierung folgende Informationen benötigt:

- Regeln und Signaturen basierend auf dem kontextabhängigen Portrayalmodell abhängig von Bereich, System, Situation und Physikalisch
- Listen mit einzelnen Änderungsanweisungen abhängig von der Rolle

Die Elemente Situation und Physikalisch beeinflussen die Bildschirmanzeige (Abbildung 52) und somit die Helligkeit des Bildschirms. Es werden keine weiteren Informationen für diese Anpassung benötigt.

Die Werkzeuge sind von den Elementen Bereich und Rolle abhängig. Dies erfordert eine Liste (Abbildung 52) der Werkzeuge:

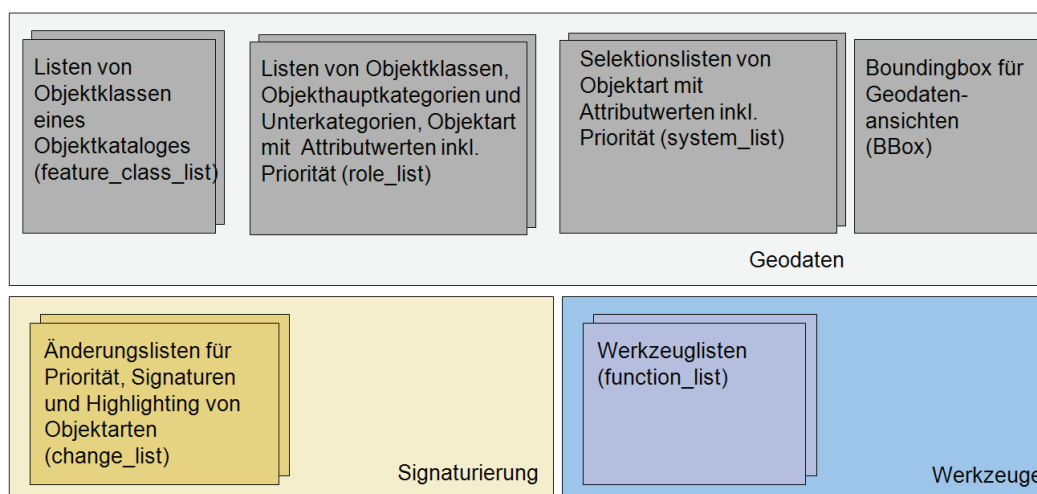
- Liste mit Werkzeugen abhängig von Bereich und Rolle

Zusammenfassend werden für die Erzeugung einer kontextbasierten Darstellung neben dem Primärkontext zusätzliche Informationen benötigt. Diese sind im Sekundärmodell (4.3.3) und im kontextabhängigen Portrayalmodell (Kap. 4.4) enthalten.

### **4.3.3. Sekundärkontext**

Der Sekundärkontext ist abhängig von Informationen des Primärkontextes. Die Listen sind nach Elementen des Kontextmodells geordnet. Es sind zum Beispiel im Sekundärkontext im Fall der verwendeten Szenarien zwei Listen für Objektklassen eines Objektkataloges geordnet nach den Bereichen „mili“ und „freizeit“ vorhanden. Hat der Nutzer das Profil Bereich „mili“, wird im kontextabhängigen Portrayalprozess die entsprechende Liste ausgewählt und verwendet. Die BBox kann über den Primärkontext Position abgeleitet oder eingegeben werden. Im kontextabhängigen Portrayalprozess werden die Listen selektiert, die dem jeweiligen Primärkontext entsprechen. Der Sekundärkontext besteht aus den folgenden Elementen (Abbildung 53):

- Liste mit Objektklassen eines Objektkataloges (feature\_class\_list)
- Liste mit Objektklassen, Objekthauptkategorien und Unterkategorien, Objektart mit Attributwerten inklusive Priorität (role\_list)
- Selektionsanweisungen für Objektart und Attributwerte inklusive Priorität (system\_list)
- Änderungsanweisungen für Priorität, Signaturen und Highlighting von Objektarten (change\_list)
- BoundingBox über System (kann aus Position abgeleitet (Primärkontext) oder eingegeben werden) (BBox)
- Liste mit Werkzeugen (function\_list)



**Abbildung 53: Sekundärkontext**

Abbildung 54 zeigt die Elemente des Sekundärkontextes in Abhängigkeit zum Primärkontext.

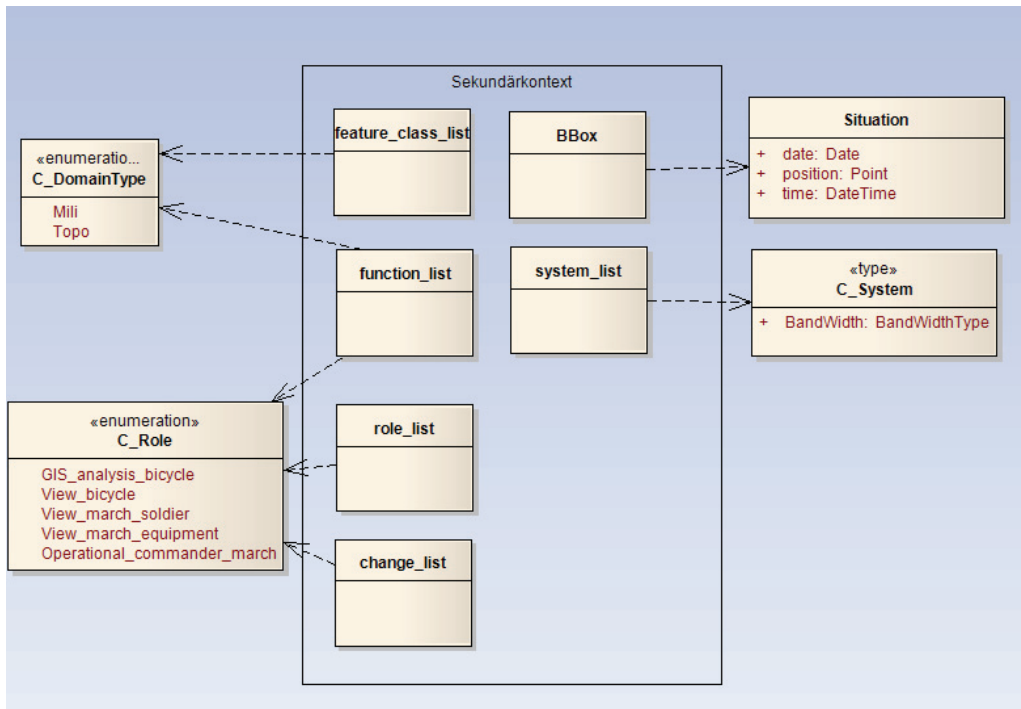


Abbildung 54: Sekundärkontext in Abhängigkeit zum Primärkontext

In Abbildung 55 ist der Workflow der Kontextinformationen dargestellt. Der Primär- und Sekundärkontext wird als Basis verwendet, um die Geodaten, die Signaturierung, die Bildschirmanzeige und die Werkzeuge anzupassen. Der Primärkontext wird im Kontextmodell (Kap. 4.3.1) modelliert. Dynamisch ändernde Informationen werden aktuell über die Kontextinformationen in den Primärkontext eingefügt. Der Sekundärkontext ist abhängig von Elementen des Primärkontextes.

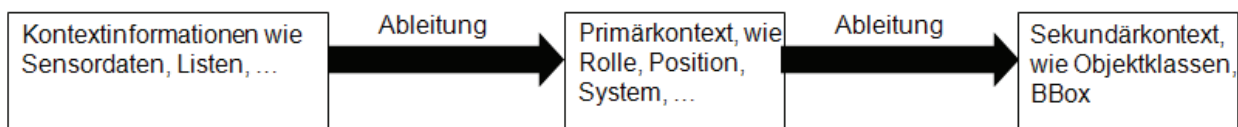


Abbildung 55: Workflow der Kontextinformationen

Kontextinformationen werden aus Sensordaten abgeleitet oder in Listen geführt. Dies beinhaltet sind folgende Daten:

- Position
- Definitionen des Sonnenauf- und -untergangs (Tageszeit)

Diese können zum Beispiel in Listenform (Sonnenauf- und -untergang) vorliegen oder als Sensordaten wie die Position des Nutzers. Um das Kontextmodell mit Informationen zu füllen, muss eine adaptive Anwendung in der Lage sein, mit ihrer Umgebung zu kommunizieren und die von der Umgebung bereitgestellten Daten und Funktionen in ihr Kontextmodell zu integrieren. Hierfür müssen zum Beispiel auch

Rohdaten von Sensoren aufbereitet werden, wie in Kapitel 3.1.2 Kontextinformationsgewinnung beschrieben.

#### 4.4. Kontextabhängiges Portrayalmodell

Das allgemeine Portrayalmodell ist in Kapitel 2 definiert worden. Dieses beinhaltet Portrayalregeln und Signaturen, um zu definieren wie die Daten dargestellt werden sollen (Beispiel: `IF feature_type=abc AND attribute=RTT THEN Symbol=123`). Das Portrayalmodell wird mit kontextbasierten Informationen erweitert, um eine kontextbasierte Visualisierung zu unterstützen und ist aus diesem Grund in drei Ebenen unterteilt:

- Portrayalregeln Ebene1 (Portrayal RuleSet): Gruppe von Portrayalregeln Ebene 2
- Portrayalregeln Ebene2 (Portrayal RuleSubSet): Gruppe von Portrayalregeln Ebene 3
- Portrayalregeln Ebene3 (Portrayal Rule): Einzelne Portrayalregeln

Die Portrayalregeln Ebene1 beinhalten alle Portrayalregeln Ebene2, gruppiert nach Bereich und System des Kontextes (statische Elemente). Die Portrayalregeln Ebene2 beinhalten alle Portrayalregeln Ebene3, gruppiert nach Situation und Physikalisch des Kontextes (dynamische Elemente). Die Portrayalregeln Ebene3 beinhalten die Portrayalregeln an sich (Abbildung 56).

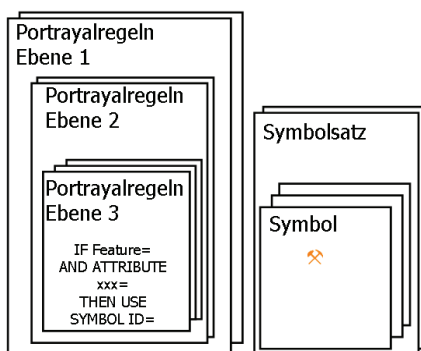
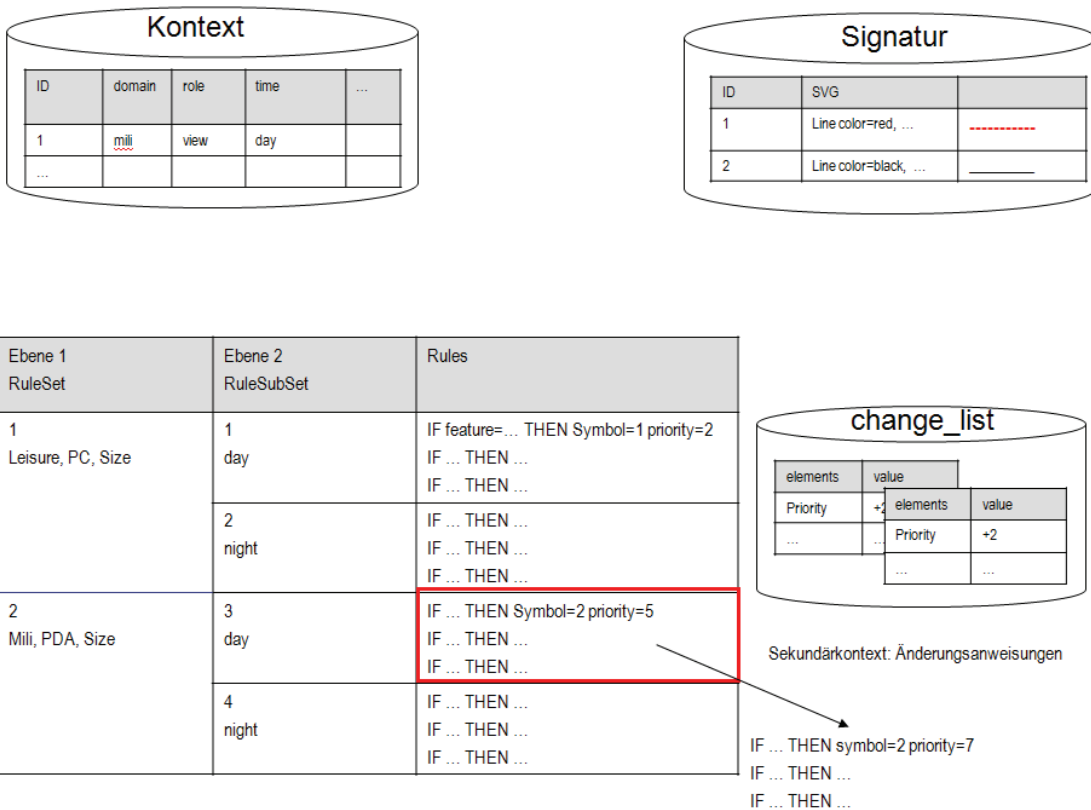


Abbildung 56: Kontextabhängiges Portrayal-Modell

Im kontextabhängigen Portrayalprozess werden Portrayalregeln Ebene1 selektiert und basierend darauf Portrayalregeln Ebene2 und Portrayalregeln Ebene3 ausgewählt. Bestimmte einzelne Portrayalregeln der Portrayalregeln Ebene3 werden je nach Rolle des Nutzers angepasst. Die Nutzerrolle beeinflusst zum Beispiel die Darstellungspriorität und die Signaturen bzw. Signaturparameter. Mit einer Darstellungsänderungsliste für Priorität, Signatur und Highlighting von Objektarten wird die Darstellung von Objekten angepasst (Abbildung 57).



**Abbildung 57: Darstellungsänderungsliste**

Ein Portrayal RuleSet ist wie folgt als Gruppe von Portrayal RuleSubSets definiert:

```
Portrayal RuleSet: {Portrayal RuleSubSet 1, Portrayal RuleSubSet 2 ...}
```

Um das geeignete Portrayal RuleSet zu erhalten, werden Adaptionenregeln (definiert in Kapitel 5.2) nach folgenden Parametern ausgewertet. Die beschriebenen Parameter können zum Beispiel nach Definition der CD ISO 19117 in einer Kontextparameterliste vorliegen.

*Bereich:* Anwendungs- bzw. Nutzerbereich (wie Militär)

*System:* System (wie TabletPC)

Ein Portrayal RuleSubSet ist wie folgt als Gruppe von Portrayal Rules definiert:

```
Portrayal RuleSubSet: {Portrayal Rule 1, Portrayal Rule 2 ...}
```

Um ein passendes Portrayal RuleSubSet zu erhalten, werden Regeln (definiert in Kapitel 5.2) nach folgenden Parametern ausgewertet:

*Portrayal RuleSet name:* Name des Portrayal Rule Sets, welches dem Portrayal Rule Sub Set angehört



---

*Situation*: Situation (wie Zeit)

*Physikalisch*: Physikalisch (wie Nebel)

Diese Parameter sind jedoch nur relevant, wenn der Nutzer das System mobil verwendet. Bestimmte Portrayal Rules werden je nach Rolle des Nutzers einzeln adaptiert.

Die Portrayalregeln referenzieren die Signatur. Zwei verschiedene Ebenen von Symbolen wurden identifiziert (Abbildung 56):

- SymbolSet (Gruppen von Signaturen)
- Symbols (Einzelne Signaturen)

Mit einer Änderungsliste für Priorität, Signaturen und Highlighting von Objekttypen können die Signaturparameter adaptiert werden. Ein eigenes SymbolSet für eine bestimmte Rolle ist eine mögliche Alternative.

## 4.5. Diskussion

Um eine kontextbasierte Visualisierung von Geodaten erzeugen zu können, mussten die hierfür notwendigen verschiedenen Komponenten in Kapitel 4 definiert und untersucht werden. Zu Beginn wurde das Kontextmodell definiert. Hierzu wurden Kontextmodelle aus der Literatur als Basis verwendet und anhand von zwei Szenarien überprüft und erweitert, um festzustellen ob die Modelle aus der Literatur die Anforderungen an die kontextbasierte Visualisierung von Geodaten erfüllen. Zweifelsohne kann dieses Kontextmodell nicht alle Nutzergruppen und deren Aufgaben im Bereich kontextbasierte Visualisierung von Geodaten abdecken. Allerdings ist das Kontextmodell sehr generisch definiert worden und durch eine Diskussion mit Anwendern die Nutzbarkeit des in dieser Arbeit definierten Kontextmodells bestätigt worden. Die Struktur des Kontextmodells kann auf verschiedene Weise definiert werden. In dieser Arbeit ist eine Aufteilung in Primär- und Sekundärkontext gewählt worden, da sich die Elemente des Sekundärkontextes aus den Elementen des Primärkontextes ableiten lassen. Es ist jedoch auch möglich, alle Elemente in einem gesamten Kontextmodell zu definieren. Durch die Umsetzung in XSD Schema und XML ist die Verwendung in einer Dienstenumgebung möglich. Außerdem ist auf diesem Wege eine Erweiterungsmöglichkeit des Kontextmodells um weitere Elemente gegeben.

Als nächster Schritt ist die Analyse des Einflusses des Kontextmodells auf die Geoobjektauswahl, Signaturierung, Bildschirmanzeige und Werkzeuge durchgeführt worden. Es sind Relationen durch eigene Überlegungen entstanden und durch Beispiele begründet worden. Das Ergebnis des Einflusses des Kontextes auf die Signaturierung bildete die Grundlage zur Erweiterung des generellen Portrayalmodells. Eine Erweiterung zum kontextabhängigen Portrayalmodell ist notwendig, um den Kontext bei der Visualisierung berücksichtigen zu können. Die Struktur des Modells kann auf verschiedene Weise definiert werden. Die Darstellungsregeln können abhängig vom Kontext gruppiert werden. Anhand des aktuellen Kontextes wird ausgewählt, welche Regeln verwendet werden sollen. Um das gleiche Ergebnis zu erzielen, kann der Kontext aber auch Teil einzelner Darstellungsregeln sein. Im ersten Fall werden Bedingungen

genutzt, um die Verwendbarkeit der Regel in Abhängigkeit vom Kontext zu prüfen. Im zweiten Fall werden die Kontextelemente als Bedingung in der Darstellungsregel verwendet. Die Regeln legen über eine Referenz die Signatur für die einzelnen Objektarten in Abhängigkeit bestimmter Attribute und des Kontextes fest. Wenn die Abfrageanweisung erfüllt ist, kann die Darstellungsaktion mit der referenzierten Signatur durchgeführt werden. In dieser Arbeit wurde eine Kombination der beiden Ansätze verwendet. Das kontextabhängige Portrayalmodell ist in hierarchischen Ebenen organisiert. Die Ebenen werden nach den Kriterien statische und dynamische Elemente definiert und anhand des aktuellen Kontextes wird ausgewählt, welche Regeln verwendet werden sollen. Ferner werden einzelne Regeln abhängig von der Rolle des Nutzers verändert. Diese Lösung des hierarchischen Aufbaus ist durch den Dienst relativ schnell auswertbar.

Zusammenfassend wurde in diesem Kapitel durch die Definition der verschiedenen Komponenten zur kontextbasierten Visualisierung die Basis für das Ziel der Arbeit gelegt. Im Weiteren wird der kontextabhängige Portrayalprozess und die Werkzeugauswahl definiert und die Erweiterung des standardisierten Darstellungsdienstes erläutert.

## 5. Kontextabhängige Geodatenvisualisierung für Nutzer eines standardisierten Portrayalwebdienstes

In diesem Kapitel wird zu Beginn der kontextabhängige Portrayalprozess mit Werkzeugauswahl vorgestellt (5.1). Anschließend werden die Adaptionenregeln im Detail und die angepasste „Portrayal Pipeline“ definiert (5.2). Die Erweiterungen eines WMS (WMS<sup>+</sup>) mit Operationen werden in Kapitel 5.3 dargestellt. Es soll eine offene und modularisierte Architektur definiert werden. Eine strikte Teilung von kontextabhängigem Portrayalmodell, Kontextmodell und kontextabhängigem Portrayalprozess ist wichtig, um die Modularisierung der Systemkomponenten sicherzustellen. In Kapitel 5.4 werden die Ergebnisse des Kapitels diskutiert.

### 5.1. Kontextabhängiger Portrayalprozess und Werkzeugauswahl

Abbildung 58 stellt den kontextabhängigen Portrayalprozess dar. Die Geodaten und das kontextabhängige Portrayalmodell werden mithilfe des Primärkontextes und Sekundärkontextes adaptiert und eine Bildschirmanzeige (Display) erzeugt. Danach kann diese Anzeige auch am Bildschirm angepasst werden. Die Werkzeuge werden je nach Kontext selektiert.

**Definition 5.1 (Auswahl Geodaten)** Die Geodaten werden mithilfe des kontextabhängigen Portrayalprozesses auf Basis des Primär- und Sekundärkontextes ausgewählt.

**Definition 5.2 (Kontextabhängiges Portrayalmodell)** Das kontextabhängige Portrayalmodell ist eine Erweiterung des allgemeinen Portrayalmodells. Aus dem kontextabhängigen Portrayalmodell werden auf Basis des Primär- und Sekundärkontextes die notwendigen angepassten Signaturierungen (Regeln und Signaturen) abgeleitet.

**Definition 5.3 (Auswahl Werkzeuge)** Die Werkzeuge werden mithilfe des kontextabhängigen Portrayalprozesses auf Basis des Primärkontextes ausgewählt.

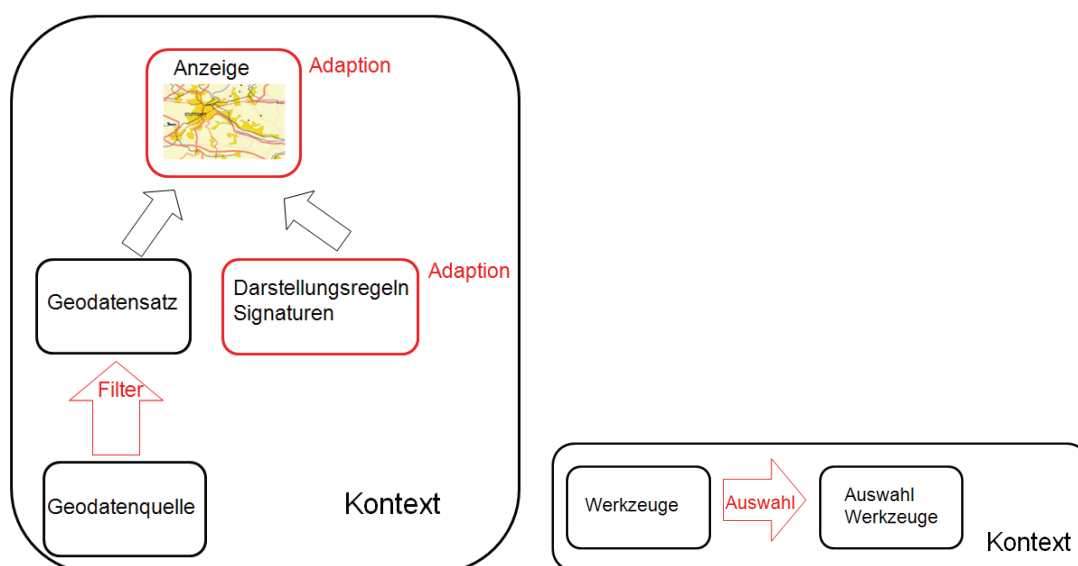


Abbildung 58: Kontextabhängiger Portrayalprozess mit Werkzeugauswahl

Im nächsten Schritt ist es notwendig die Adaptionenregeln zu definieren. Es werden vier verschiedene Typen von Adaptionenregeln definiert, die die vier Adaptionsschritte für die Geoobjekte, die Signaturierung, die Anzeige und die Werkzeuge (Kap. 4.3.2) durchführen.

Die vier Typen (Abbildung 58) sind:

- *Objektfilterregeln für die Geoobjekte*
- *Portrayaladaptionenregeln für die Signaturierung*
- *Graphische Adaptionenregeln für die Anzeige*
- *Funktionale Auswahlregeln für die Werkzeuge*

**Definition 5.4 (Objektfilterregeln)** *Diese Regeln selektieren aus dem gesamten Quellgeodatenbestand kontextbasiert die für den Nutzer relevanten Geodaten in einem Geodatensatz.*

**Definition 5.5 (Portrayaladaptionenregeln)** *Diese Regeln adaptieren die kontextabhängigen Regeln und Signaturen.*

**Definition 5.6 (Graphische Adaptionenregeln)** *Diese Regeln passen je nach Kontext die Anzeige des Bildschirms auf der Clientseite an.*

**Definition 5.7 (Funktionale Auswahlregel)** *Diese Regel selektiert je nach Kontext die Werkzeuge, welche dem Nutzer bereitstehen.*

Die Regeln des kontextabhängigen Portrayalprozesses basieren auf einer „Bedingung“ (condition) definiert durch den Kontext und eine „Aktion“ (action) (IF (Bedingung) THEN (Aktion)). Das Resultat der Bedingung kann wahr oder falsch sein. Eine Aktion wird angestoßen, wenn die Bedingung als wahr evaluiert wurde. Näheres zu den Adaptionenregeln ist in 5.2 zu finden.

Der kontextabhängige Portrayalprozess wird im nächsten Abschnitt detailliert in Form von „Business Workflow Diagrammen“ dargestellt. Hierfür wurde das Programm Enterprise Architect verwendet. Außerdem soll der kontextabhängige Portrayalprozess mit Hilfe eines Beispiels verdeutlicht werden. Für eine Aufgabe (Ansicht Marschplanung von schwerem Gerät) hat der Nutzer einen Tablet\_PC im Feld zur Präsentation der Geodaten.

Zuerst beeinflussen die *Objektfilterregeln* die Geodatenfilterung (Daten → Auswahl Geodaten). Ein Beispiel für diese Regeln wird in Tabelle 14 aufgezeigt. Basierend auf einer Liste für einen bestimmten Bereich (definiert im Sekundärkontext) filtern *Objektfilterregeln 1* den Objektkatalog für diesen Bereich. Die *Objektfilterregeln 2* selektieren mit Hilfe einer Objektklassenliste die Objektkategorien, Unterkategorien und Objektarten relevant für eine spezifische Rolle mit einer bestimmten Priorität. Diese Selektion wird mit *Objektfilterregeln 3* für das jeweilige System des Nutzers gefiltert. Hierfür wird die Priorität der Objekte wichtig. Ein Schwellwert für die Prioritäten bestimmt, welche Objekte selektiert werden sollen. Die *Objektfilterregeln 4* berücksichtigen die Situation des Nutzers und ein Ausschnitt wird über die bounding box für die Präsentation definiert (Abbildung 59).

**Tabelle 14: Objektfilterregeln**

<b>Adaptionsregeln</b>	<b>Beispiel Primärkontext</b>	<b>Beispiel Sekundärkontext</b>
Objektfilterregeln 1	Bereich: Militär (statisch)	Objektklassen aus GeoInfoOK (Objektartenkatalog Militär)
Objektfilterregeln 2	Rolle: Auskunftssuchender Marsch Gerät (statisch)	Flüsse, Straßen, Brücken
Objektfilterregeln 3	System: Tablet PC (statisch)	Selektionsanweisungen
Objektfilterregeln 4	Position: 9,5 51,3 12:00:00 (dynamisch)	BBOX

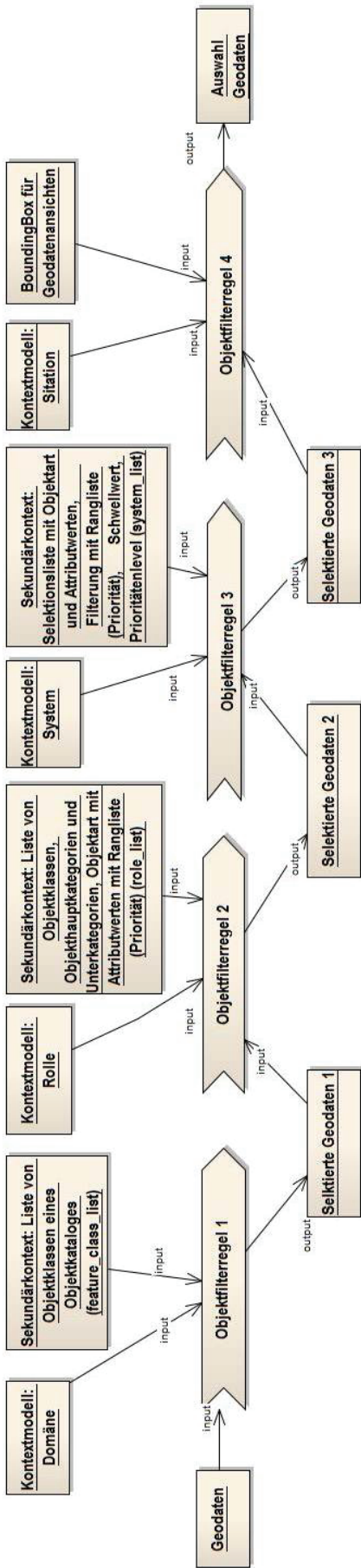


Abbildung 59: Objektfilterregeln

Ein Beispiel für die *Portrayaladaptionsregeln* wird in Tabelle 15 aufgezeigt. Der Bereich und das System des Nutzers sind statische Aspekte des Kontextmodells. Diese Komponenten sind die Basis für die Selektion der Portrayalregeln Ebene1 mithilfe der *Portrayaladaptionsregeln 1*. Diese Portrayalregeln Ebene1 sollten abhängig von diesen Informationen klassifiziert werden. Die *Portrayaladaptionsregeln 2* filtern die Portrayalregeln Ebene2 aus den Portrayalregeln Ebene 1. Wenn das System „mobil“ genutzt wird, werden die Nutzersituation und der physikalische Aspekt berücksichtigt. Zum Beispiel definiert eine Liste von Sonnenauf- und -untergang (Tageszeit) die Grenze zwischen Tag und Nacht. Die Basis der *Portrayaladaptionsregeln 3* sind die Regeln in „Portrayalregeln Ebene2“. Bestimmte „Portrayalregeln Ebene3“ müssen auf Basis der Nutzerrolle angepasst werden. Dieses beeinflusst die Darstellungspriorität und die Signatur oder bestimmte Signaturparameter. Mit Hilfe einer Änderungsliste für Priorität, Symbole und Highlighting von Objektarten wird die Signaturierung von Objekten angepasst (Abbildung 60).

**Tabelle 15: Portrayaladaptionsregeln**

<b>Adaptionsregeln</b>	<b>Beispiel Primärkontext</b>	<b>Beispiel Sekundärkontext</b>
Portrayaladaptionsregeln 1	Bereich: Militär  System: Tablet PC (statisch)	-
Portrayaladaptionsregeln 2	System: Mobil (statisch)  Situation: Zeit und Physikalisch: Wetter wie Nebel (dynamisch)	-
Portrayaladaptionsregeln 3	Rolle: Auskufsuchender Marsch Gerät (statisch)	Änderungsanweisungen

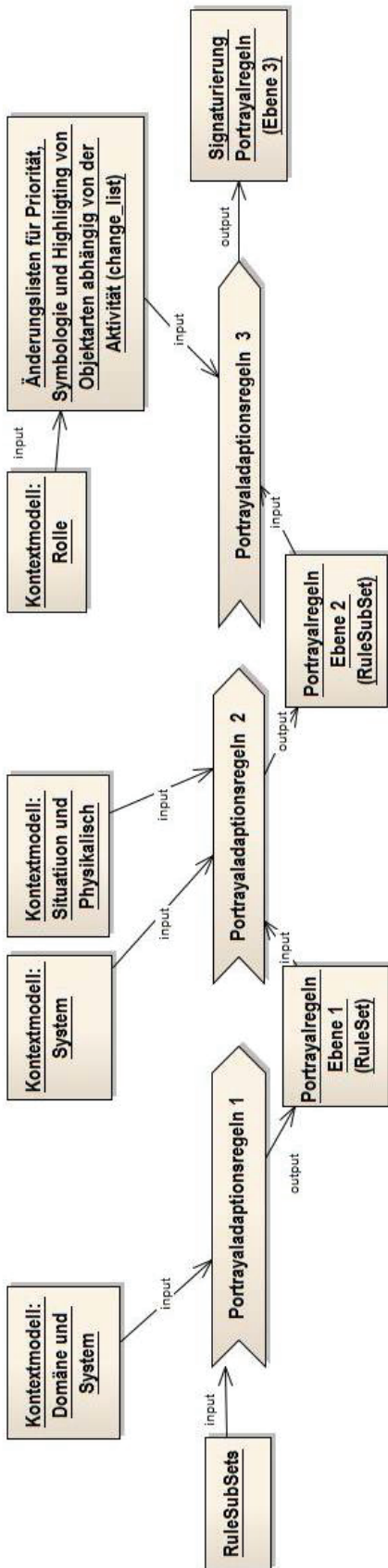


Abbildung 60: Portrayaladaptionregeln



Die Präsentation wird generiert (gerendert) basierend auf der Signaturierung und der Auswahl an Geodaten. Dieses kann durch *graphische Adaptionsregeln* (Änderung des Bildschirms wie Helligkeit) beeinflusst werden. Ein Beispiel für diese Regeln wird in Tabelle 16 aufgezeigt. Mit Hilfe der Signaturierungen produziert die Applikation eine Anzeige. Diese kann von graphischen Adaptionsregeln (wie Helligkeit des Gerätebildschirms) beeinflusst werden. Die *graphischen Adaptionsregeln 1* beeinflussen die Anzeige abhängig von der Situation des Nutzers wie Zeit. Die Liste der Definition von Sonnenauf- und -untergang (Zeit des Tages) definiert den Übergang zwischen Tag und Nacht. Des Weiteren beeinflussen die *graphischen Adaptionsregeln 2* die Anzeige abhängig von der physikalischen Umgebung. Die Information wie Wetterinformationen kann vom Nutzer selber oder mithilfe von Sensoren erfasst werden (Abbildung 61).

**Tabelle 16: Graphische Adaptionsregeln**

<b>Adaptionsregeln</b>	<b>Beispiel Primärkontext</b>	<b>Beispiel Sekundärkontext</b>
Graphische Adaptionsregeln 1	Situation: Zeit (dynamisch)	-
Graphische Adaptionsregeln 2	Physikalisch: Nebel (dynamisch)	-

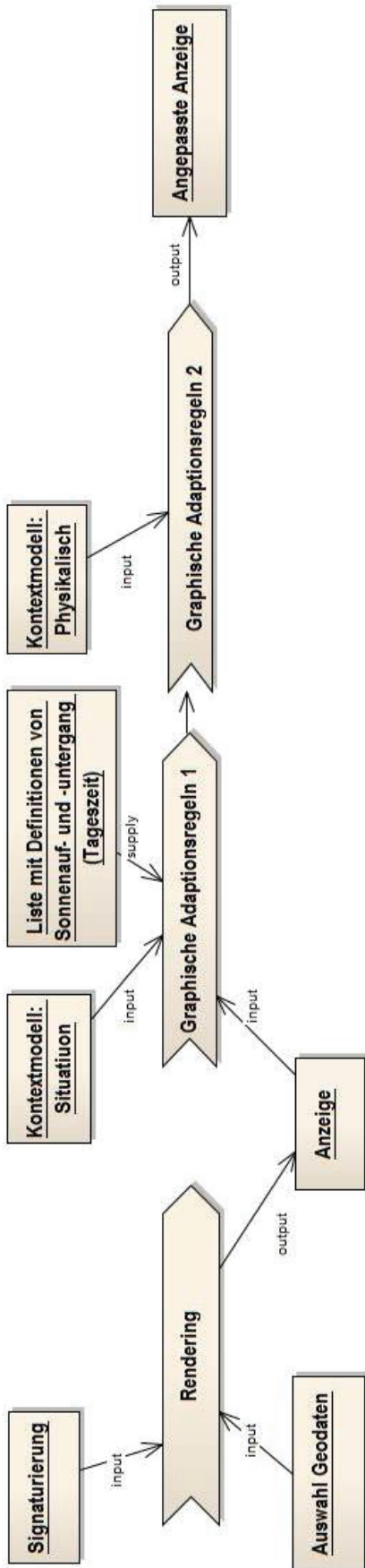
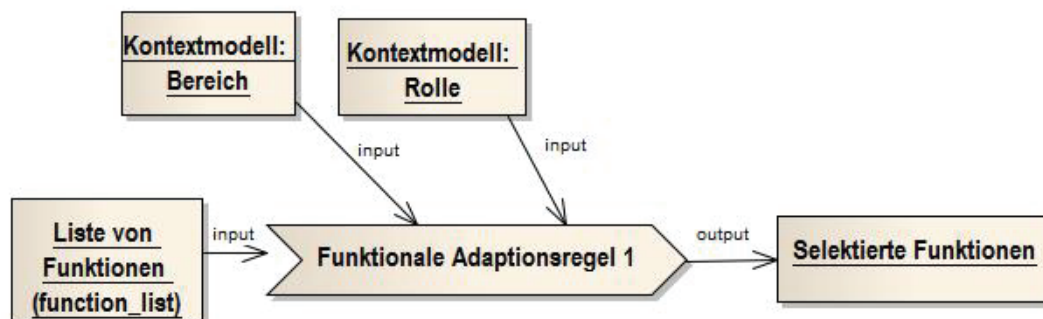


Abbildung 61: Graphische Adaptionsregeln mit Renderingprozess

Mit Hilfe der *Funktionalen Auswahlregel 1* werden die Werkzeuge selektiert, welche dem Bereich und Rolle des Kontextes angehören (Tabelle 17 und Abbildung 62).

**Tabelle 17: Funktionale Auswahlregel**

Adaptionsregeln	Beispiel Primärkontext	Beispiel Sekundärkontext
Funktionale Auswahlregel 1	Bereich: Militär (statisch)  Rolle: Auskunftsuchender Marsch Gerät (statisch)	-



**Abbildung 62: Funktionale Auswahlregel**

Dieser vorgestellte Ansatz wurde anhand einer Prototypimplementierung getestet (Kap. 6). Des Weiteren muss es Aquisitions(erfassungs)regeln, um das Kontextmodell zu aktualisieren. Mit Eingabe von Nutzern oder Sensoren werden Änderungen des Kontextes aktualisiert. Nebenbei ist es möglich, das Nutzerinterface zu adaptieren. Diese Aspekte werden hier nicht weiter behandelt.

## 5.2. Adaptionsregeln und angepasste „Portrayal pipeline“

Im Folgenden werden die verschiedenen Adaptionsregeln vorgestellt. Hierbei wird zwischen Anpassung der Präsentation und der Auswahl der Werkzeuge unterschieden. Ferner wird die angepasste „portrayal pipeline“ vorgestellt.

### 5.2.1. Adaptionsregeln: Anpassung der Präsentation

Die Anpassung der Präsentation der Geodaten besteht aus dem Prozess der Auswahl an Geodaten zu einem Geodatensatz, der Anpassung der Signaturierung, des Renderingprozesses zur Erzeugung der Präsentation und der Anpassung der Bildschirmanzeige auf Clientseite. Im Folgenden werden die verschiedenen Adaptionsregeln in Pseudocode beschrieben (Abbildung 63).

```

1 To ADAPT GEOVIEW
2   begin
3     ADAPT GEODATA (primary context, secondary context, geodata base);
      Output: FeatureSet;//Selected features
4     ADAPT PORTRAYAL (primary context, secondary context, context based portrayal);
      Output: Adapted portrayal rules;
5     RENDER VIEW (FeatureSet, adapted portrayal rules);
      Output: View;
6     ADAPT GRAPHIC (primary context, secondary context);
      Output: Display;
7   end;

```

**Abbildung 63: Anpassung der Präsentation in Pseudocode**

Erläuterung „ADAPT GEODATA” Prozess

Die *Objektfilterregeln* bestehen aus vier Hauptregeln. Die erste Regel selektiert die Geodaten nach dem Bereich (Code 3.9). Die zweite Regel selektiert die Geodaten nach der Rolle. Hierbei sind auch mehrere Rollen eines Nutzers über eine Gewichtung zu berücksichtigen (Code 3.11). Die dritte Regel selektiert die Geodaten nach dem System (Code 3.13). Die vierte Regel selektiert die Geodaten nach dem dynamischen Element Situation (Position) und der Boundingbox (Code 3.16) (Abbildung 64).

```

3.1 get: domain as: context.user.preferences.domain
      from: primary context
3.2 get: role as: context.user.preferences.role
      from: primary context
3.3 get: system type as: context.system.device.type
      from: primary context
3.4 get: system size as: context.system.device.size
      from: primary context
3.5 get: system resolution as: context.system.device.resolution
      from: primary context
3.6 get: bandwidth resolution as: context.system.bandwidth
      from: primary context
3.7 get: position as: context.environment.situation.position
      from: primary context
3.8 get: feature_class_list from: secondary context //usage: context.user.preferences.domain
3.9 get: features as: GeodataSet1
      from: feature_class_list in: geodata base
3.10 get: role_list from: secondary context //usage: context.user.preferences.role
3.11 get: features as: GeodataSet2
      from: role_list in: GeodataSet1
3.12 get: system_list from: secondary context //usage: context.system.device.type and
      context.system.device.size and context.system.device.resolution and
      context.system.bandwidth
3.13 get: features as: GeodataSet3
      from: system_list in: GeodataSet2
3.14 get: position
      from: context.environment.situation.position and context.system.device.size
3.15 calculate: bbox
      from: position and device size
3.16 get: features as: FeatureSet
      from: role_list where: INSIDE bbox or COVEREDBY position

```

**Abbildung 64: „Adapt Geodata” in Pseudocode**

Erläuterung „ADAPT PORTRAYAL” Prozess

Die *Portrayaladaptionsregeln* bestehen aus drei Hauptregeln, die im Folgenden dargestellt werden. Die erste Regel selektiert das passende Portrayal RuleSet (s. Kapitel 4.2) abhängig von den statischen Elementen Bereich und System (Code 4.10). Die zweite Regel selektiert das passende Portrayal Rule Sub Set (s. Kapitel 4.2) abhängig von den dynamischen Elementen Situation und Physikalisch, wenn das System mobil statt stationär genutzt wird (Code 4.11). Die dritte Regel ist eine Gruppe von Regeln, die je nach Rolle die einzelnen Portrayalregeln und die Symbole und deren Komponenten anpasst (Code 4.13) (Abbildung 65).

```

4.1 get: domain as: context.user.preferences.domain
    from: primary context
4.2 get: system type as: context.system.device.type
    from: primary context
4.3 get: system size as: context.system.device.size
    from: primary context
4.4 get: system resolution as: context.system.device.resolution
    from: primary context
4.5 get: system use as: context.system.device.systemuse
    from: primary context
4.6 get: time as: context.environment.situation.time
    from: primary context
4.7 get: weather information as: context.environment.physical.weather
    from: primary context
4.8 get: lightCondition information as: context.environment.physical.lightCondition
    from: primary context
4.9 get: role as: context.user.preferences.role
    from: primary context
4.10 get: RuleSet //usage: context.user.preferences.domain and context.system.device.type and
    context.system.device.size and context.system.device.resolution
    from: context based portrayal
4.11 if: context.system.device.systemuse = MOBILE
    then begin
    get: RuleSubSet //usage: context.environment.situation.time and
    context.environment.physical.weather and context.environment.physical.lightCondition
    from: RuleSet
    else: RuleSet := RuleSubSet
    end
4.12 get: change_list as Number from: secondary context //usage: context.user.preferences.role
4.13 get: adapted portrayal rules from: RuleSubSet
    where: change_list ID = Number //adapt portrayal rules defined in change_list

```

**Abbildung 65: „Adapt Portrayal” in Pseudocode**

Erläuterung „ADAPT GRAPHIC” Prozess

Die *graphischen Adaptionsregeln* bestehen aus zwei Hauptregeln, die im Anschluss aufgezeigt werden. Die Anpassung erfolgt auf dem Client/Gerät. Die erste Regel passt die Ansicht abhängig von Situation (Zeit) an (Code 6.4). Die zweite Regel passt die Ansicht abhängig von physikalischen Konditionen an (Code 6.5) (Abbildung 66).

```

6.1 get: time as: context.environment.situation.time
    from: primary context
6.2 get: weather information as: context.environment.physical.weather
    from: primary context
6.3 get: lightCondition information as: context.environment.physical.lightCondition
    from: primary context
6.4 if: context.environment.situation.time < SetTimeLimit | > SetTimeLimit
    then: lighten display | shade display;
6.5 if: context.environment.physical.weather < SetWeatherLimit | > SetWeatherLimit and
context.environment.physical.lightCondition < LightConditionLimit | > SetLightConditionLimit
    then: lighten display | shade display;

```

Abbildung 66: „Adapt Graphic“ in Pseudocode

### 5.2.2. Adaptionsregeln: Auswahl der Werkzeuge

Die Auswahl der Werkzeuge besteht aus einer Selektion der passenden Werkzeuge abhängig von den statischen Elementen Bereich und Rolle entsprechend der definierten Werkzeuge aus Kapitel 4.2 (*Funktionale Auswahlregel*). Im Folgenden wird die Adaptionsregel in Pseudocode beschrieben (Abbildung 67).

```

1 To ADAPT FUNCTIONS (primary context, secondary context)
Output: selected functions
2 get: domain as: context.user.preferences.domain
    from: primary context
3 get: role as: context.user.preferences.role
    from: primary context
4 get: function_list from: secondary context //usage: context.user.preferences.domain and
context.user.preferences.role
5 get: functions
    from: function_list

```

Abbildung 67: „Adapt Functions“ in Pseudocode

Im Anhang 12.3 werden die Regeln jeweils zu Beginn generell und anschließend in Beispielen in SQL-Anweisungen definiert und zum besseren Verständnis mithilfe eines Beispiels veranschaulicht. Eine weitere Möglichkeit besteht in der Definition der Regeln in ISO 19143 – Filter Encoding (ISO (2010)). Ein Filter ist vergleichbar zur „select-from-where“ SQL-Anfrage. Der XML kodierte Filter kann in eine „select-from-where“ SQL-Anfrage transformiert werden, um Daten aus einer Datenbank abzurufen. In Abbildung 68 ist ein Beispiel einer SQL-Abfrage mit der Umsetzung in Filter Encoding (FE) dargestellt. Der „typeName“ definiert die Daten, die abgefragt werden sollen. Der Filter „PropertyIsEqualTo“ mit den Attributen „PropertyName“ und „Literal“ definiert die Abfragewerte.

```

SQL-Abfrage:
SELECT CASE WHEN domain=mili and device=pda THEN SELECT * FROM RuleSets

Filter Encoding:
<Query typeName="RuleSets">
<Filter>
<And>
<PropertyIsEqualTo>
<PropertyName>domain</PropertyName>
<Literal>mili</Literal>
</PropertyIsEqualTo>
<PropertyIsEqualTo>
<PropertyName>device</PropertyName>
<Literal>pda</Literal>
</PropertyIsEqualTo>
</And>
</Filter>
</Query>

```

**Abbildung 68: Abfrage in SQL und Filter Encoding**

Die Verwendung der Adaptionenregeln und ihre Auswirkungen werden in Kapitel 5.2.3 beschrieben. Mit Hilfe von Akquisitionsregeln kann das Kontextmodell aktualisiert werden. Dieses ist in dieser Arbeit nicht näher berücksichtigt worden. Der gesamte Prozess und die Verwendung der zuvor aufgelisteten erforderlichen Informationen sind im nächsten Kapitel beschrieben.

### 5.2.3. Angepasste „Portrayal Pipeline“

Für eine kontextbasierte Visualisierung von Nutzern eines Portrayaldienstes muss die Portrayal Pipeline aus Kapitel 2.3.3 (Abbildung 69 b) erweitert werden, um so den kontextabhängigen Portrayalprozess zu realisieren.

Diese angepasste „portrayal pipeline“ ist in fünf Repräsentationskomponenten (Daten, Datensatz, kontextabhängige Regeln und Signaturen, Signaturierung, Anzeige) und drei Typen von Adaptionenregeln (Pfeile) unterteilt (Abbildung 69 a). Der kontextabhängige Portrayalprozess verwendet drei verschiedene Typen von Adaptionenregeln, um eine Präsentation erzeugen zu können. Zuerst beeinflussen die *Objektfilterregeln* die Geodatenfilterung (Daten → Datensatz). Des Weiteren regulieren die *Portrayaladaptionenregeln* die Anpassung der Signaturierung (Kontextabhängige Regeln und Signaturen → Signaturierung). Mithilfe der Signaturierung und des Datensatzes erzeugt die Anwendung die kontextabhängige Präsentation als Anzeige auf dem Bildschirm (Client). Dieser kann durch *Graphische Adaptionenregeln* (wie Bildschirmhelligkeit) beeinflusst werden. Die *funktionale Auswahlregel* wird separat ausgeführt. Diese Regel selektiert je nach Kontext die Werkzeuge der Anwendung. Der kontextabhängige Portrayalprozess wird über einen Portrayalwebdienst und einen Client realisiert.

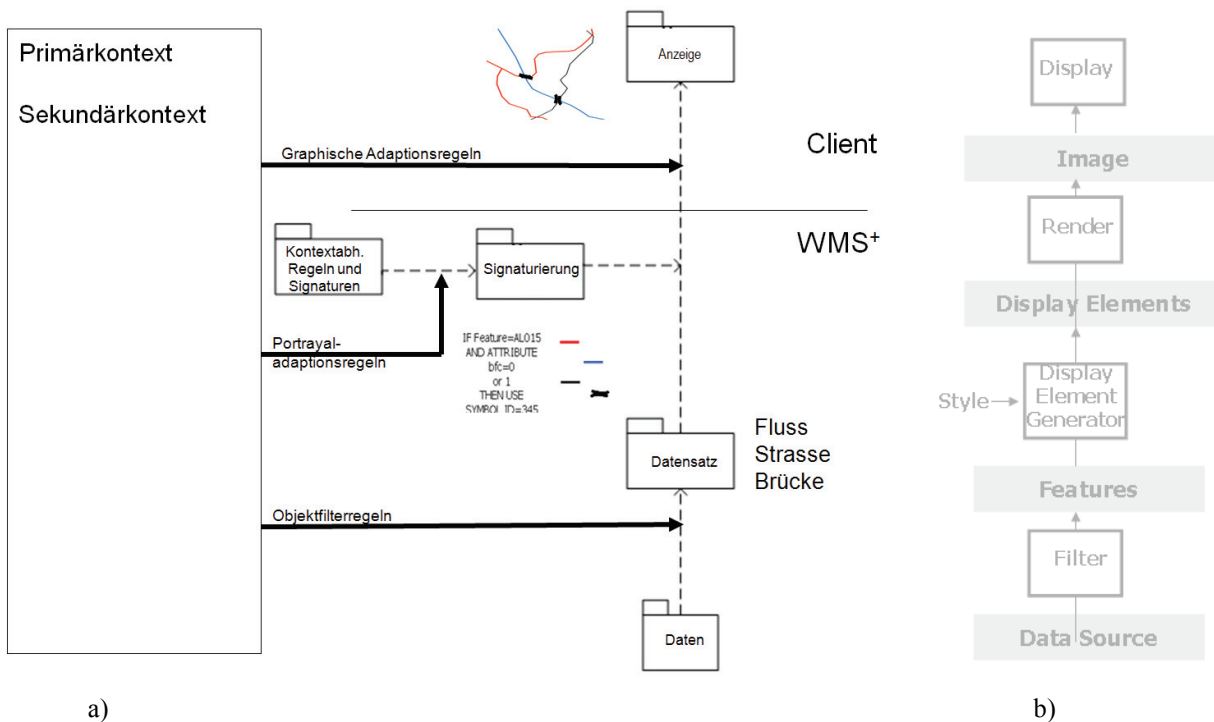


Abbildung 69: Angepasste „Portrayal Pipeline“ (a), Original (b)

Der Portrayalwebdienst kann über einen erweiterten WMS Dienst (WMS<sup>+</sup>, Abbildung 69 a) realisiert werden. Dieser Aspekt wird in Kapitel 5.3 näher beschrieben.

### 5.3. Erweiterungen des WMS (WMS<sup>+</sup>): Operationen (Gesamtframework)

Um eine kontextbasierte Geodatenvisualisierung mittels eines WMS durchzuführen, müssen Erweiterungen des Dienstes beschrieben werden. Nach Empfehlung der ISO 19154 – *Standardization Requirements for Ubiquitous Public Access* (Kapitel 3.2, Ubiquitous Public Access) wird hierfür ein Profil eines Dienstes benötigt (Abbildung 45). In diesem Kapitel wird eine WMS-Erweiterung (WMS<sup>+</sup>) definiert. Zuerst werden die Funktionen beschrieben, die der Dienst erfüllen muss. Anschließend werden diese als Operationen mit Aufruf- und Antwortparametern definiert. Der Dienst soll folgende Funktionen und somit Operationen beinhalten:

- Abfrage Schemabeschreibung eines Kontexttypen („ContextTypes“) über die zusätzliche Operation „DescribeContextType“
- Abfrage von Kontextdokumenten über die zusätzliche Operation „GetContext“
- Einfügen, aktualisieren und löschen (insert, update, and delete) von Kontextinformationen über die zusätzlichen Operationen „TransactContext“
- Abfrage der Präsentation abhängig von Kontextdokument über die erweiterte WMS-Operation „GetMap“
- Abfrage der Metadaten des Dienstes und der Daten über die erweiterte WMS-Operation „GetCa-



pabilities“

### 5.3.1. DescribeContextType Operation

Die „DescribeContextType“ Antwort gibt eine Schemabeschreibung zu einem bestimmten Kontexttypen („ContextTypes“) und ihren Attributen zurück (Abbildung 70).

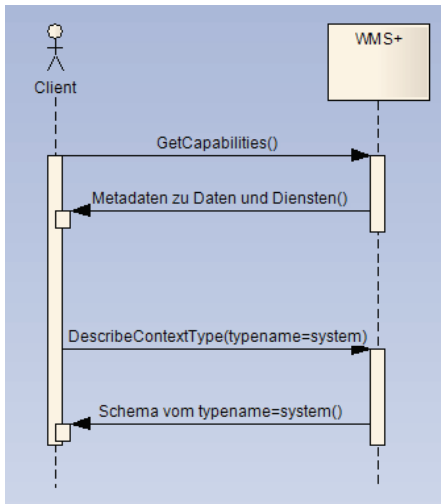


Abbildung 70: DescribeContextType Operation

Die Kontexttypen entsprechen den Elementen aus Kapitel 4.3.1. Die Schemabeschreibung definiert, wie der Dienst die Kodierung für Kontextinformationen für das Einfügen und die Aktualisierung erwartet und wie die Kontextinformationen vom Dienst als Ausgabe über den „GetContext“-Aufruf generiert werden. Im Sequenzdiagramm (Abbildung 70) werden zuerst die Metadaten des Dienstes und die Kontexttypen in einem XML Dokument über die Operation „GetCapabilities“ (Spezifizierung in Kapitel 5.3.5) aufgerufen. Über die Informationen in diesem Dokument ist es im Anschluss möglich, einen Aufruf zu starten der das Schema der Kontextdaten zurückgibt. Der Beispielaufruf sieht folgendermaßen aus:

[http://localhost:8080/server/wms?version=...&service=wms&request=DescribeContextType&typename=.](http://localhost:8080/server/wms?version=...&service=wms&request=DescribeContextType&typename=)

..

Die Antwort ist das Schema aller Kontexttypen. Es entspricht dem Inhalt des XML Schema aus Kapitel 4.3.1.

### 5.3.2. GetContext Operation

Die „GetContext“ Antwort liefert Kontextdateien zurück (Abbildung 71).

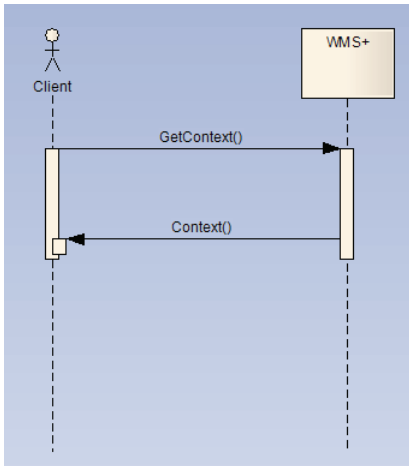


Abbildung 71: GetContext Operation

Der Beispielaufruf ist im Anschluss wie folgt definiert:

<http://localhost:8080/server/wms?version=...&service=wms&request=GetContext&typename=...>

Die Antwort muss valide zum XSD Kontextschema sein. Eine XML-Beispieldatei ist im Anhang zu finden.

### 5.3.3. TransactContext Operation

Die „TransactContext“ Operation bietet Funktionalitäten zum Einfügen, Aktualisieren und Löschen (insert, update, and delete) für Kontextinformationen an (Abbildung 72). Wenn die Transaktion abgeschlossen ist, wird ein XML-Antwortedokument mit Statusbericht an den Client gesendet.

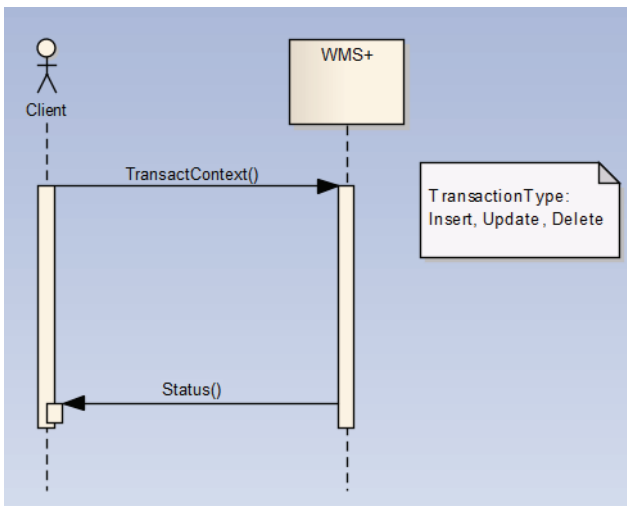


Abbildung 72: TransactContext Operation

Die XML Kodierung eines „TransactContext“-Aufrufs wird durch das XML Schema Fragment in Abbildung 73 definiert.

```

<xsd:element name="TransactContext" type="ctxt:TransactContextType"/>
<xsd:complexType name="TransactContextType"> <xsd:sequence>
<xsd:element ref="ctxt:LockId" minOccurs="0"/>
<xsd:choice minOccurs="0" maxOccurs="unbounded">
  <xsd:element ref="ctxt:Insert"/>
  <xsd:element ref="ctxt:Update"/>
  <xsd:element ref="ctxt>Delete"/>
</xsd:choice> </xsd:sequence>

```

**Abbildung 73: „TransactContext“-Aufruf**

Beim Einfügen neuer Kontextdokumente zum Dienst, sollte dieser einen eindeutigen Kennzeichner („unique identifier“) für alle neu hinzugefügten Kontextinstanzen besitzen bzw. zugewiesen bekommen. Die Organisation der eindeutigen Kennzeichner wird an einem Beispiel wie folgt vorgeschlagen: Context\_id=context1 (beinhaltet user1, system1, dynamic1). Dies soll mit dem Attribut „idgen“ im TransactContextdokument realisiert werden. Es definiert welche Methode der Identifizierung verwendet wird. Tabelle 18 zeigt mögliche Werte für das Attribut und deren Bedeutung.

**Tabelle 18: Idgen Wert**

idgen Wert	Aktion
GenerateNew (default)	Erzeugung eines eindeutigen Kennzeichner (ID)
UseExisting	Der Dienst verwendet die ID, welche im Kontextdokument definiert wird. Wenn die ID schon vorhanden ist, sollte der Dienst eine Fehlermeldung zurückgeben.

Bei der „Insert“-Transaktion wird ein XML Dokument an den Dienst versendet (Abbildung 74).

```

<?xml version="1.0"?>
<wms:Transaction version="1.1.0" service="WMS" ...>
  <wms:Insert idgen="GenerateNew">
    KONTEXTDOKUMENT
  </wms:Insert>
</wms:Transaction>

```

**Abbildung 74: „Insert“-Transaktion**

Die Antwort ist ein XML Dokument mit Statusbericht der Transaktion. Falls eine „Insert“-Operation durchgeführt wurde, muss der Dienst den Nutzer über alle neu erzeugten Elemente und deren Kennzeichner berichten (Beispiel <ctxt:totalInserted>5</ctxt:totalInserted>).

### 5.3.4. Erweiterte GetMap Operation

Die „GetMap“ Operation des WMS liefert eine kartenähnliche Darstellung an den Nutzer zurück. Diese Operation wird um verschiedene Parameter verändert, um eine Darstellung abhängig von der Kontextdatei erzeugen zu können (Abbildung 75).

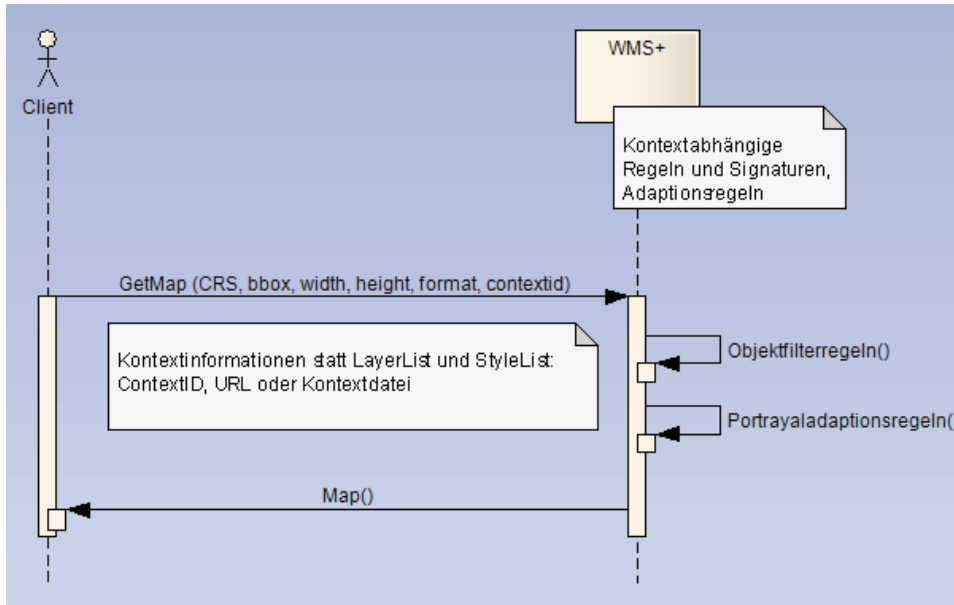


Abbildung 75: GetMap Operation

Der WMS Aufruf wird für die kontextsensitive Darstellung um die Parameter „Layer“ und „Style“ verringert und um die zwei Parameter „VERSION“ des Kontextschemas und „CONTEXT“ erweitert. Es gibt drei verschiedenen Varianten eine Kontextdatei in einem WMS Aufruf zu referenzieren. Bei der ersten Möglichkeit befindet sich das Kontextdokument auf dem Kontextdienst. Der Aufruf sieht in diesem Fall wie folgt aus:

<http://.../WMS?REQUEST=GetMap&VERSION=1.0&BBOX=9,51,9,52&CONTEXTID=context1&WIDTH=400&HEIGHT=400&FORMAT=PNG>

Liegt das Dokument auf einem außenstehenden Server sieht der Aufruf wie folgt aus:

<http://.../WMS?REQUEST=GetMap&Version=1.0&BBOX=9,51,9,52&CONTEXT=http://.../context.xml&WIDTH=400&HEIGHT=400&FORMAT=PNG>

Die dritte Variante bietet dem Nutzer die Möglichkeit seine Kontextdatei direkt in der URL einzubinden:

[http://.../WMS?REQUEST=GetMap&Version=1.0&BBOX=9,51,9,52&CONTEXT\\_BODY=<xml...>&WIDTH=400&HEIGHT=400&FORMAT=PNG](http://.../WMS?REQUEST=GetMap&Version=1.0&BBOX=9,51,9,52&CONTEXT_BODY=<xml...>&WIDTH=400&HEIGHT=400&FORMAT=PNG)

Nach Erhalt der „GetMap“-Operation extrahiert der Dienst alle datenbankrelevanten Informationen (Geodaten, Portrayalregeln, Signaturen und Kontextinformationen) mithilfe der Adaptionregeln (Objektfilter-

regeln, Portrayaladaptionsregeln) und erzeugt die Präsentation. Hierbei liegen die Adaptionsregeln im Dienst vor. Eine Alternative bietet die Definition von Adaptionsregeln über Filter Encoding, die dem Dienst bereitgestellt werden. Eine Übermittlung der Adaptionsregeln zusammen mit dem Kontextdokument des Nutzers mithilfe der „TransactContext“-Operation ist möglich, wenn der Dienst die Fähigkeit besitzt das „Filter Encoding“-Dokument in SQL-Abfragen umzuwandeln (s. Kapitel 5.2).

### 5.3.5. Erweiterte GetCapabilities Operation

Die „GetCapabilities“-Antwort liefert die Metadaten des Dienstes und der Daten als XML-Datei zurück an den Client (Abbildung 76).

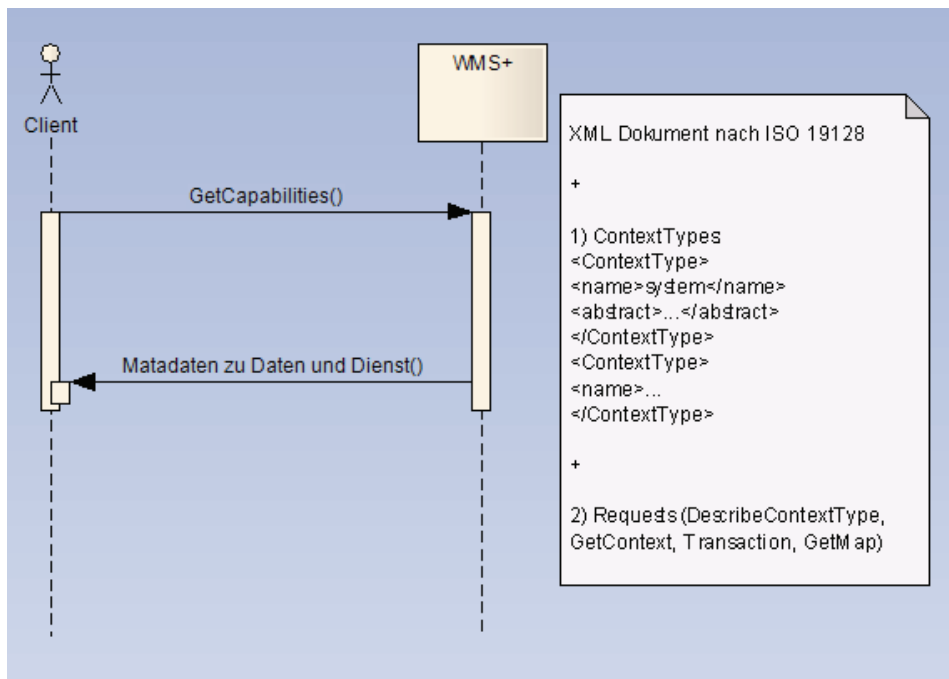


Abbildung 76: GetCapabilities Operation

Der „GetCapabilities“-Aufruf entspricht der WMS Spezifikation nach ISO (2005b). Die WMS Spezifikation erlaubt in ihrem „wms capabilities“ Schema erweiterte „Capabilities“ und „Operations“ (*extended capabilities and operations*). Diese werden mit `<_ExtendedCapabilities>`- oder `<_ExtendedOperations>`-Elementen im Servicemetadatenchema beschrieben. Die Antwort-XML-Datei des Kontextdienstes wird um die Datentypen („ContextType“) und die neu spezifizierten Operationen erweitert. Diese werden im Folgenden als XML Schema Fragment auszugsweise beschrieben:

1. Erweiterung der Capabilities um das Element „ContextType“ (Abbildung 77).
2. Erweiterung der Capabilities mit den Operationen „DescribeContextTyp“, „GetContext“, „TransactContext“ (Abbildung 78).

```

<element name="ContextType" substitutionGroup="wms:_ExtendedCapabilities">
  <complexType>
    <sequence>
      <element ref="wms:Name" minOccurs="0"/>
      <element ref="wms:Abstract" minOccurs="0"/>
    </sequence>
  </complexType>
</element>

```

**Abbildung 77: „ContextType“-Element im „Capabilities“-Dokument**

```

<element name="DescribeContextType" type="wms:OperationType"
substitutionGroup="wms:_ExtendedOperation"/>

<element name="GetContext" type="wms:OperationType"
substitutionGroup="wms:_ExtendedOperation"/>

<element name="Transaction" type="wms:OperationType"
substitutionGroup="wms:_ExtendedOperation"/>

```

**Abbildung 78: Erweiterte Operationen im „Capabilities“-Dokument**

Das vollständige Schema ist in Anhang 12.1 zu finden. Praktische Ergebnisse des vorgestellten Konzeptes wie die Architektur, die Technologien und Operationen des Prototypsystems werden in Kapitel 6 vorgestellt.

## 5.4. Diskussion

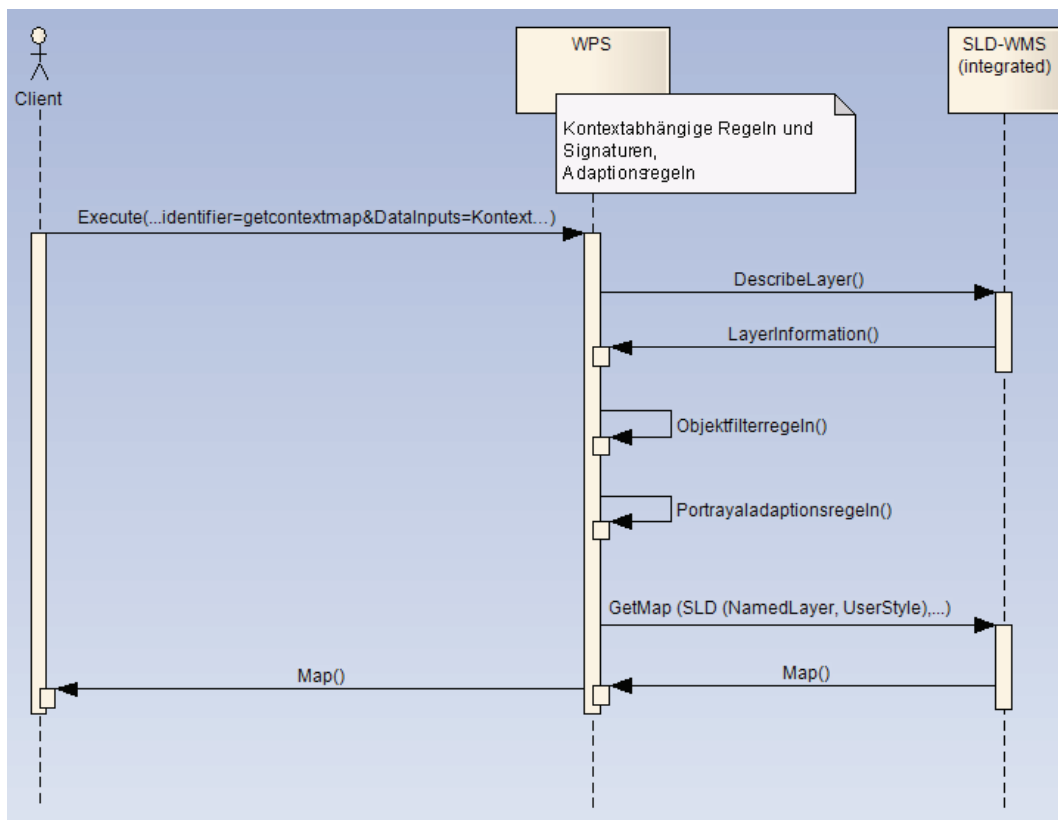
In Kapitel 5 wurde der Portrayalprozess untersucht. Ferner wurden auf Basis der Ergebnisse aus Kapitel 4 notwendige Adaptionenregeln definiert und der standardisierte Darstellungsdienst WMS erweitert. Zusammenfassend wurde ein Rahmen definiert der es erlaubt, spezifische für verschiedene Nutzergruppen angepasste Präsentation in einer Webumgebung standardbasiert zu präsentieren. Die Komponenten in Kapitel 4, der Prozess in 5.1 und die Adaptionenregeln in Pseudocode in 5.2 haben hierbei Allgemeingültigkeit und können auch in einem anderen kontextsensitivem System ohne Dienste realisiert werden.

Das hier vorgestellte Konzept ist sicher nur im begrenzten Maße allgemein verwendbar und auf andere Anwendungsbereiche übertragbar. Für ein anderes Szenario ist nach einer Analyse der Anwendung das Kontextmodell inklusive Rolle anzupassen. Durch die Realisierung des Kontextes in XSD Schema ist eine Erweiterung möglich. Ferner müssen die Adaptionenregeln angepasst werden damit eine Berücksichtigung der neuen Kontextelemente erfolgen kann. Außerdem muss das kontextabhängige Portrayalmodell hinsichtlich der Kontextelemente ergänzt werden. Die Verwendung anderer Geodatenbestände ist möglich. Hierfür ist es allerdings notwendig, die Listen im Sekundärkontext anzupassen.

Der kontextbasierte Portrayalprozess mit den verschiedenen Komponenten ist in dieser Arbeit durch eine Erweiterung des WMS umgesetzt worden. Eine Erweiterungsmöglichkeit des vorgestellten WMS Dienstes ist die Verwendung von SLDs als Kodierung für Regeln und Signaturen (Kap. 2.3.3). Ein Ansatz ist die Verwendung von SLDs mit integrierten Kontextinformationen. Hierbei ist es allerdings notwendig, die SLDs mit Kontextelementen zu erweitern. Eine Alternative zur Erweiterung des WMS ist die Spezifizierung eines *OGC Web Processing Services* (OGC (2007c)), da diese Spezifikation sehr abstrakt und somit universell verwendbar ist. Die neu spezifizierten Operationen „DescribeContext“, „GetContext“ und „TransactContext“ können im WPS wie im erweiterten WMS<sup>+</sup> (Kap. 5.3.1 -5.3.3) umgesetzt werden. Die erweiterte „GetMap“-Operation kann im WPS als „Execute“-Operation (Ausführung des Prozesses) realisiert werden. SLDs sind das Ergebnis des WPS und können anschließend in einem OGC SLD-WMS (Kap. 2.3.3) verwendet werden. Abbildung 79 zeigt ein Sequenzdiagramm eines WPS und eines „integrated“ SLD-WMS (Kap. 2.3.3). Der Client schickt eine Anfrage mit Kontextinformationen an den WPS:

[http://...?REQUEST=Execute&Version=1.0.0&identifier=getcontextmap&DataInputs=context, ...](http://...?REQUEST=Execute&Version=1.0.0&identifier=getcontextmap&DataInputs=context,...)

Mit der „Execute“-Operation werden dem WPS die notwendigen Eingabeparameter mit Identifikator-Attribut (identifier) „getcontextmap“ übergeben, der den Prozess definiert. Als Antwort wird ein SLD Dokument erzeugt, welches der WPS an einen „integrated“ SLD-WMS weiterleiten kann.



**Abbildung 79: WPS „Execute“ Operation und „integrated“ SLD-WMS**

Der SLD-WMS erzeugt die Darstellung und diese wird über den WPS an den Client zurückgesendet. Die

Informationen zur Geodatenquelle (Layer) erhält der WPS vom SLD-WMS über „DescribeLayer“. Die kontextabhängigen Regeln und Signaturen und die Adaptionenregeln werden im WPS verwendet, um die Layerauswahl mit den entsprechenden Signaturen in ein SLD umzuwandeln. Eine weitere Möglichkeit ist es dem Client das vom WPS erzeugte SLD zu senden, damit dieser einen „Get Map“-Aufruf inklusive SLD-Datei durchführen kann.

In dieser Arbeit wurde die Erweiterung des WMS gewählt, da die Verwendung von SLDs nicht alle Anforderungen an die Darstellungsregeln und Signaturen berücksichtigt (Definition 2.9). Zum einen unterstützt SLD bzw. SE nicht das Attribut *Objektyp* („delineation“) (Punkt, Linie, Fläche). Zum anderen werden nicht alle komplexen Signaturen unterstützt. Dies beinhaltet vor allem komplexe Liniensignaturen (Punktsymbol-Linien). Ferner referenziert eine Darstellungsregel in ISO 19117 – Portrayal (ISO (2009b)) eine Signatur mit der Möglichkeit der Komposition von „Untersignaturen“. Eine Zusammenfassung mehrerer Signaturen zu einer Gruppe ist in SE nicht möglich. Es können in SE mehrere „Symbolizer“ verwendet werden, aber es gibt keine Unterstützung hierarchischer Strukturen wie in der ISO 19117 – Portrayal (ISO (2009b)). Diese Einschränkungen in SLD können durch Workarounds behoben werden, so dass trotz der beschriebenen Einschränkungen eine Verwendung von WPS, SLD-WMS und SLD möglich ist.



## 6. Praktische Ergebnisse

Das in den vorherigen Kapiteln beschriebene Konzept wurde in einem Prototypsystem realisiert und getestet. In diesem Kapitel werden die Umsetzungen aus Sicht der Systemarchitektur (Kap. 6.1) sowie die Technologien und Operationen des Darstellungsdienstes (Kap. 6.2) beschrieben. In Kapitel 6.3 werden die Ergebnisse zusammengefasst und diskutiert.

### 6.1. Architektur des Prototypsystems

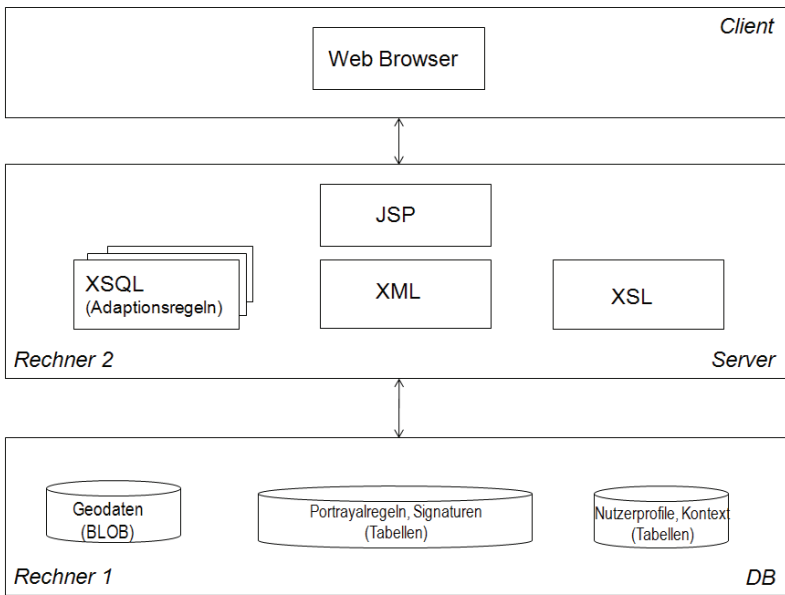
Im Folgenden wird die Architektur inklusive Hardware, Software und Datengrundlage beschrieben. Das Prototypsystem besteht aus zwei PCs (Server und Datenbank) und einen beliebigen PC mit Web Browser (Client) zur Anfrage an den Darstellungsdienst und zur Darstellung des Ergebnisses. Auf einem PC (Rechner 1) mit dem Betriebssystem Windows XP befindet sich die Oracle Spatial-Datenbank 9.2 (Details s. 6.2.1) mit den Datenbank-Tabellen für die

- Geodaten,
- Regeln und Signaturen,
- Kontextdaten und
- Listen.

Der Darstellungsdienst läuft auf einem PC (Rechner 2) mit dem Betriebssystem Windows 2000 Server und basiert auf folgenden Applikationen bzw. Dokumenten (Details s. 6.2.2):

- Apache Server 2 mit Tomcat 4.1 und Oracle XML Developer's Kit,
- JSP-, XSQL- und XSL-Dokumente (Adaptionsregeln) und
- GetCapabilities-Dokument.

Die Architektur des Prototypsystems ist in Abbildung 80 dargestellt. Über einen *Web Browser* kann der Nutzer eine Anfrage an den Dienst (*Server*) stellen. Dies geschieht über eine JSP-Seite, die die Parameter in der Abfrage an verschiedene XSQL-Dokumente weiterleitet. In diesen Dokumenten sind die Adaptionsregeln in SQL Anweisungen abgelegt. Durch diese SQL Anweisungen werden Anfragen an die verschiedenen Datenbanken (*DB*) gestellt und Adaptionen durchgeführt. Die dadurch entstandenen Darstellungselemente werden über den Server in XML Format zurückgegeben und per XSL im Vektorformat SVG umgewandelt. Die SVG-Datei wird an den Web Browser zurückgeschickt, der die Darstellungselemente über ein Plug-In in eine für den Menschen sichtbare Anzeige umwandelt.



**Abbildung 80: Architektur (DB – Server - Client)**

Im Folgenden werden die verschiedenen Datenbank-Tabellen vorgestellt. Zu Beginn werden die physikalischen Modelle für die Geodaten und Portrayalregeln beschrieben. Die Struktur der Geodaten entspricht dem GeoInfo Objektartenkatalog 25 des GeoInfoDBw und wird in Abbildung 81 als Auszug aus einer Oracletabelle dargestellt. Jedes Objekt hat eine eindeutige ID (Spalte FLDBID), Geometrie (Spalte GEOMETRY) und Attribute (wie Spalte PRO, ...).

Table: AA010F  
 freib3D | AGIS | TABLE | AA010F

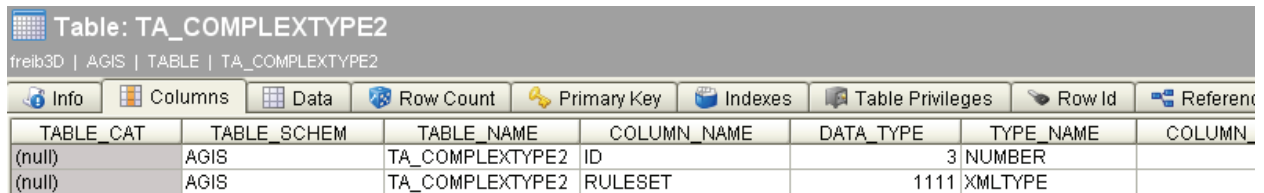
	FLDBID	PRO	GEOMETRY
1	1	Ton	3003, null, null, oracle.sql.ARRAY@a64642, or
2	2	Kies/Schotter	3003, null, null, oracle.sql.ARRAY@1dd040a, i
3	3	Kies/Schotter	3003, null, null, oracle.sql.ARRAY@12bed82, i
4	4	Kies/Schotter	3003, null, null, oracle.sql.ARRAY@f0a755, or
5	5	Kalkstein	3003, null, null, oracle.sql.ARRAY@22ed4b, or
6	6	Kalkstein	3003, null, null, oracle.sql.ARRAY@56291a, or
7	7	Kalkstein	3003, null, null, oracle.sql.ARRAY@1ea7821, i
8	8	Kalkstein	3003, null, null, oracle.sql.ARRAY@1a32fd1, o
9	9	Kalkstein	3003, null, null, oracle.sql.ARRAY@1a1732d, i
10	10	Sonstig	3003, null, null, oracle.sql.ARRAY@7c4246, or
11	11	Ton	3003, null, null, oracle.sql.ARRAY@13306ad, i
12	12	Gneis	3003, null, null, oracle.sql.ARRAY@56c528, or
13	13	Kies/Schotter	3003, null, null, oracle.sql.ARRAY@72cbb9, or
14	14	Kies/Schotter	3003, null, null, oracle.sql.ARRAY@176a8c8, c
15	15	Kies/Schotter	3003, null, null, oracle.sql.ARRAY@c07930, or
16	16	Kies/Schotter	3003, null, null, oracle.sql.ARRAY@96d92e, or

**Abbildung 81: Oracletabellenansicht**

Das Attribut **GEOMETRY** gibt den Geometriotyp an (Beispiel: 3003, null, null, oracle.sql.ARRAY@..., oracle.sql.ARRAY@...). Die erste Zahl definiert die Dimension der Koordinaten. Die zweite Ziffer beschreibt das lineare Bezugssystem. Mit dem Standardwert 0 wird kein Bezugssystem verwendet. Die restlichen Werte definieren den Geometriotyp gemäß dem OGC (2005b) Simple-Feature-

Modell (z.B. 01 für Punkt, 03 für einfaches Polygon, evtl. mit Löchern).

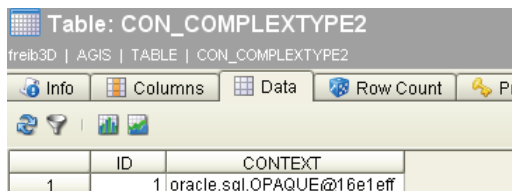
Die Struktur der Portrayalregeln entspricht der Definitionen in Kapitel 2. In Abbildung 82 werden die Regelsätze als *XML TYPE* abgelegt. Die Signaturen sind bei der Umsetzung des Prototypen zusammen mit den Portrayalregeln gespeichert.



TABLE_CAT	TABLE_SCHEM	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	COLUMN...
(null)	AGIS	TA_COMPLEXTYPE2	ID	3	NUMBER	
(null)	AGIS	TA_COMPLEXTYPE2	RULESET	1111	XMLTYPE	

**Abbildung 82: Portrayalregeln und Signaturen in Oracle-Tabelle**

Die Nutzerprofile (XML Kontextdatei) werden in einzelnen Spalten ebenfalls als *XML TYPE* abgelegt (Abbildung 83). Diese werden, bevor sie in die Datenbank übernommen werden, gegen das Kontextschema aus Kapitel 4 geprüft.



ID	CONTEXT
1	1   oracle.sql.OPAQUE@16e1eff

**Abbildung 83: Nutzerprofil in Oracle-Tabelle**

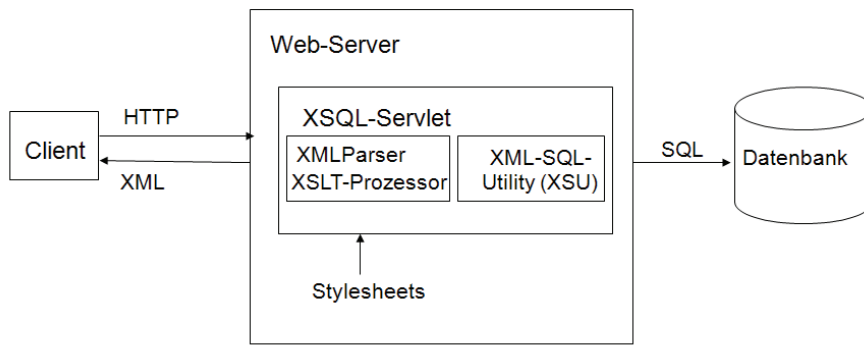
Die Adaptionenregeln sind in XSQL-Dokumenten abgelegt und werden im nächsten Kapitel näher beschrieben.

## 6.2. Technologien des Prototypsystems

In diesem Kapitel werden die Datenbanktechnologien (6.2.1) und die Operationen des Portrayalwebdienstes (6.2.2) beschrieben.

### 6.2.1. Oracle Spatial Datenbank

Zur Umsetzung des Prototypsystems wurde das Oracle XDK (XML Developer's Kit) verwendet. Dies ist eine Sammlung von Tools für die Schnittstelle von XML und (Oracle-) Datenbank. Oracle XDK beinhaltet unter anderem XML SQL Utility (XSU), XML-Parser, XSLT-Prozessoren und XSQL zum Erstellen von XML-Dokumenten aus SQL (Abbildung 84).



**Abbildung 84: Oracle XML Developer's Kit**

Für die Anfrage an den Darstellungsdienst wird eine HTTP-Anfrage über JSP (Kap. 2.2.2) gestellt (Beispiel: `http://servername:8080/context.jsp?GetMap&CONTEXTID=5`). Die JSP-Dateien nehmen die Parameter und leiten sie über `<jsp:include>` oder `<jsp:forward>` an eine XSQL-Seite weiter. In der XSQL-Datei sind die Adaptionsregeln als SQL Abfragen definiert (Abbildung 85).

```

<% String X1 = request.getParameter("CONTEXTID");%>
<% String whereTo = "/map.xsql?ID=" + CONTEXTID; %>
<jsp:forward page='<%= whereTo %>' />

```

**Abbildung 85: Auszug aus JSP-Datei**

Beispiele für XSQL Action Element Handler, die das Oracle XDK unterstützt, sind Abfragen (`query`) und Einfügen (`insert-param`) von Werten. Mit dem „query“- XSQL Action Element Handler werden SQL-Anweisungen ausgelöst und das Ergebnis in XML zurückgegeben. Der XSQL Action Element Handler „insert-param“ wird zum Beispiel zum Upload von XML-Dokumenten verwendet. In Tabelle 19 im Anhang werden die XSQL Action Element Handler und ihre Beschreibung dargestellt. Näheres zu Oracle XDK ist in OracleXDK (2007) zu finden.

Seit Oracle 9i steht XMLType als neuer Datentyp in der Datenbank zur Verfügung. XMLType-Daten können unstrukturiert als CLOB (Character Large Object) oder strukturiert in objektrelationalen Tabellen gespeichert werden. Um XML-Dokumente strukturiert in objektrelationalen Tabellen zu speichern, ist eine XML-Schemadefinition erforderlich. Das XML-Schema wird für die XMLType-Spalte registriert. Somit können nicht beliebige XML-Dokumente in der XMLType-Spalte abgelegt werden, sondern nur noch solche XML-Dokumente, die der XML-Schema-Definition entsprechen.

Diese Technologie wurde auch beim Prototypen eingesetzt. Das Kontextschema wurde in Oracle gespeichert und vor dem Einfügen neuer Kontextdokumente, wird das Dokument gegen das Schema geprüft. Oracle unterstützt verschiedene XMLType-Funktionen. „Select extract“ zum Beispiel liefert XML zurück und extrahiert einen oder mehrere Äste eines XMLType-Objekts.

```

<xsql:query rowset-element="eins">
select extractValue (ruleset, '//rule [@id="1"]/fill') as COLUMN_VALUE1 from ta_complextype2
</xsql:query>

```

**Abbildung 86: „Select extractValue“ XMLType-Funktion**

„Select extractValue“ liefert anhand eines Xpath-Ausdrucks den Wert eines Knotens (Abbildung 86). Diese Funktion kann zu langen Wartezeiten führen. Aus diesem Grund sollte ein Index festgelegt werden. In Tabelle 20 im Anhang werden die XMLType-Funktionen und ihre Beschreibung dargestellt.

### 6.2.2. Server: Portrayalwebdienst

Der Portrayalwebdienst wurde mit Apache Server 2 mit Tomcat 4.1 und Oracle XML Developer's Kit als erweiterter WMS umgesetzt. Die Ergebnisse der Test werden im nächsten Kapitel dargestellt. Im Folgenden wird die prototypische Umsetzung der einzelnen Operationen des Portrayalwebdienstes aus Kapitel 5.3 beschrieben.

Bei der „GetCapabilities“-Operation wird ein Dokument mit Metadaten des Dienstes und der Daten abgefragt. Das XML-Dokument liegt als Datei auf dem Server.

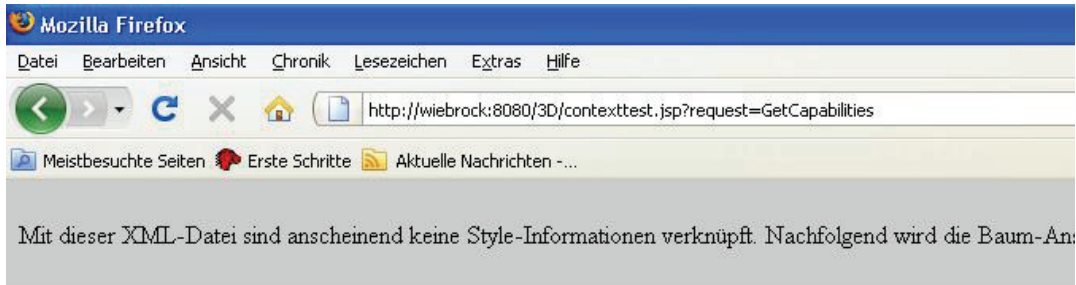
Die „DescribeContextType“-Operation gibt eine Schemabeschreibung der Kontexttypen zurück. Mit der SQL-Anweisung `SELECT x.RULESET.getschemaurl() as Document FROM tabel x where ID='{@X1}'` im XSQL-Dokument wird das Schemadokument von der Kontextdatenbank angefragt. Die „GetContext“-Operation gibt Kontextdokumente einzelner Nutzer zurück. Ein Beispiel einer Anfrage in SQL an die Kontextdatenbank ist `SELECT x.ruleset.getStringVal() asDocument FROM ta_complextype2 x WHERE ID='{@X1}'`.

Die „TransactContext“-Operation aktualisiert, löscht und fügt Kontextinformationen des Nutzers ein. Die Aktualisierung kann mit dem XSQL Action Element Handler (`updateXML`) und das Einfügen mit `insert-param` realisiert werden. Im Prototyp wurde die Operation durch eine einfache „Upload“-Funktion und nicht als „TransactContext“-Operation realisiert (`updateXML`). Das Kontextdokument wird nach einer Schemavalidierung in der Kontextdatenbank abgelegt.

Die „GetMap“-Operation gibt eine Präsentation abhängig vom Kontextdokument zurück. Zuerst wird über eine HTTP Anfrage die Verarbeitung einer serverseitig vorliegenden XSQL-Seite veranlasst und anschließend an den Dienst übergeben. Der wiederum extrahiert alle datenbankrelevanten Informationen (Geodaten, Portrayalregeln, Signaturen und Kontextinformationen) mithilfe der Adaptionregeln in SQL. Die Ergebnisse solcher Interaktionen werden nach dem kontextabhängigen Portrayalprozess in einem XML-Dokument zusammengefügt. Durch Angabe einer Transformationsvorschrift in Form eines XSL-Stylesheets innerhalb der XSQL-Pages wird das generierte XML-Dokumente in SVG umgewandelt und anschließend an den Anfragenden zurückgesendet. Die Ergebnisse der Operationen sind im folgenden Kapitel beschrieben.

### 6.3. Ergebnisse der Tests und Diskussion

Die prototypischen Tests wurden erfolgreich durchgeführt. In den Abbildung 87, Abbildung 88, Abbildung 89, Abbildung 90 und Abbildung 91 sind die Ergebnisse der einzelnen Operationen des Dienstes dargestellt.



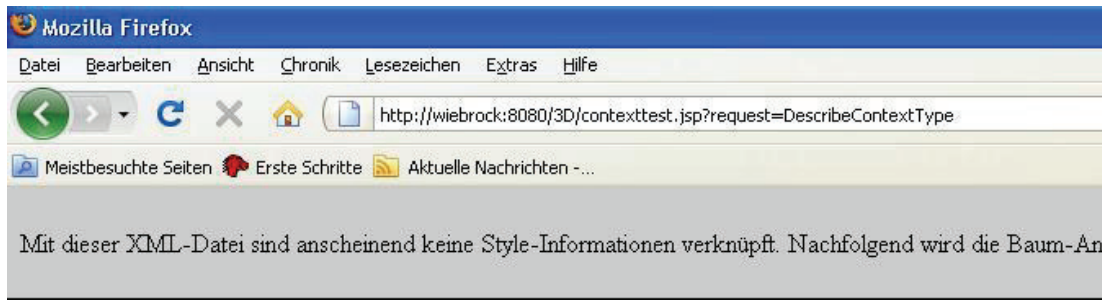
```

- <wms:WMS_Capabilities xsi:schemaLocation="http://www.opengis.net/ctxt context_capabilities.xsd">
- <wms:Service>
  <wms:Name>WMS extended</wms:Name>
  <wms:Title>WMS test</wms:Title>
  <wms:Abstract>wms reference implementation</wms:Abstract>
- <wms:KeywordList>
  >
  <wms:Keyword>wms</wms:Keyword>
  <wms:Keyword>AGIS</wms:Keyword>
  <wms:Keyword>context</wms:Keyword>
</wms:KeywordList>
  <wms:OnlineResource xlink:type="simple" xlink:href="http://wiebrock:8080/3D/">
+ <wms:ContactInformation></wms:ContactInformation>
  <wms:Fees>none</wms:Fees>
  <wms:AccessConstraints>none</wms:AccessConstraints>
</wms:Service>
- <wms:Capability>
- <wms:Request>
  - <wms:GetCapabilities>
    <wms:Format>application/vnd.ogc.wms_xml</wms:Format>
    + <wms:DCPType></wms:DCPType>
    </wms:GetCapabilities>
  + <wms:GetMap></wms:GetMap>
  + <DescribeContextType></DescribeContextType>
  + <GetContext></GetContext>
</wms:Request>
- <wms:Exception>
  <wms:Format>application/vnd.ogc.se_xml</wms:Format>
</wms:Exception>
+ <ContextType></ContextType>
- <wms:Layer queryable="0" cascaded="0" opaque="0" noSubsets="0">
  <wms:Title>Top level layer</wms:Title>
  <wms:Abstract>the top-level-layer</wms:Abstract>
- <wms:KeywordList>
  <wms:Keyword>deegree</wms:Keyword>
  <wms:Keyword>layer</wms:Keyword>
</wms:KeywordList>
  <wms:CRS>EPSG:4326</wms:CRS>
  <wms:BoundingBox CRS="EPSG:4326" minx="100 0" miny="0 0" maxx="100 0" maxy="0 0"/>

```

Abbildung 87: Get Capabilities Ergebnis



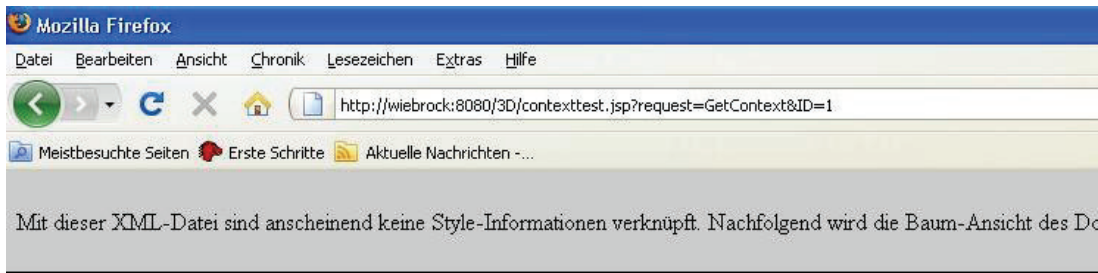


```

- <xs:schema>
  <xs:import namespace="http://www.opengis.net/gml" schemaLocation="C:\Dokumente und Einstellung
  <xs:element name="C_System" type="C_System"/>
- <xs:complexType name="C_System">
  - <xs:sequence>
    <xs:element name="BandWidth" type="BandWidthType"/>
    <xs:element name="C_Device" type="C_Device"/>
  </xs:sequence>
  </xs:complexType>
  <xs:element name="C_Dynamic" type="C_Dynamic"/>
- <xs:complexType name="C_Dynamic">
  - <xs:sequence>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="Situation" type="Situation"/>
    <xs:element name="Physical" type="Physical"/>
  </xs:sequence>
  </xs:complexType>
  <xs:element name="C_User" type="C_User"/>
- <xs:complexType name="C_User">
  - <xs:sequence>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="Preferences" type="Preferences"/>
  </xs:sequence>
  </xs:complexType>
  <xs:element name="C_Context" type="C_Context"/>
- <xs:complexType name="C_Context">
  - <xs:sequence>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="dynamic" type="C_Dynamic"/>
    <xs:element name="system" type="C_System"/>
    <xs:element name="C_User" type="C_User"/>
  </xs:sequence>

```

Abbildung 88: DescribeContextType Ergebnis



```

- <C_Context xsi:noNamespaceSchemaLocation="testneu_280110.xsd">
  <id>0</id>
  - <dynamic>
    <id>0</id>
    - <Situation>
      <date>2007-08-13</date>
      - <position>
        <gml:pos>45.256 -71.92</gml:pos>
      </position>
      <time>2009-12-09T09:30:45Z</time>
    </Situation>
    - <Physical>
      <LightCondition>day</LightCondition>
      <Weather>rain</Weather>
    </Physical>
  </dynamic>
  - <system>
    <BandWidth>DSL2000</BandWidth>
  - <C_Device>
    <devicetype>Notebook</devicetype>
    <resolution>800x600</resolution>
    <size>19"-21"</size>
  </C_Device>
</system>
- <C_User>
  <id>8</id>
  - <Preferences>
    <domai>Mili</domai>

```

Abbildung 89: GetContext Ergebnis

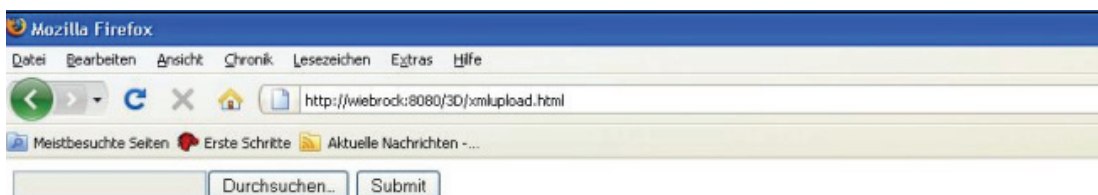
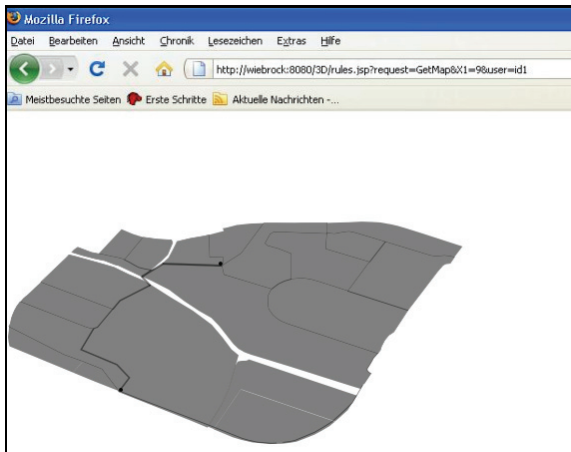


Abbildung 90: Upload Operation

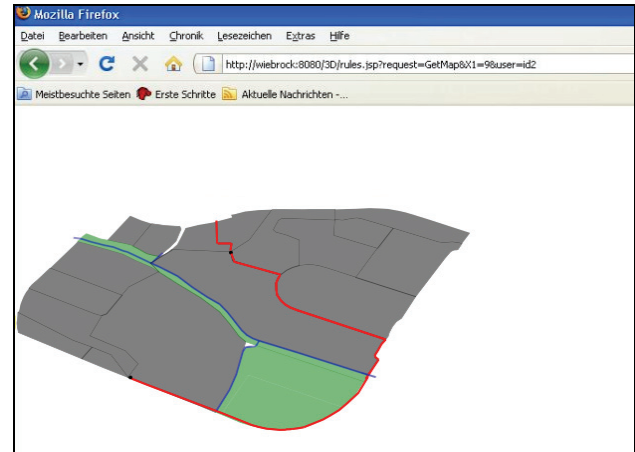
In Abbildung 91 a ist das Ergebnis der Anfrage des Nutzers der Rolle *Auskunftssuchender Marsch Soldat* und in Abbildung 91 b der Anfrage des Nutzers der Rolle *Auskunftssuchender Marsch Gerät* dargestellt. Die Anfrage des Nutzers der Rolle *Auskunftssuchender Marsch Soldat* zeigt alle GeoInfo-Objekthauptkategorien „Künstliche Bodenbedeckung“ in grau, die Straßen in schwarz, den Startpunkt und Endpunkt und die vorgeschlagene Route hervorgehoben in schwarz. Die Anfrage des Nutzers der



Rolle *Auskunftssuchender Marsch Gerät* zeigt neben die GeoInfo-Objekthauptkategorien „Künstliche Bodenbedeckung“ klassifiziert nach Objektarten „Siedlung“ in grau und „Park“ in grün. Außerdem werden mögliche Hindernisse wie ein Fluss in blau dargestellt. Die rote Linie stellt eine große ausgebaute Straße dar, die für den Transport von schwerem Gerät geeignet ist und somit als vorgeschlagene Route hervorgehoben dargestellt wird. Die Kontexte der Nutzer waren bei diesen Tests auf Seite des Dienstes gespeichert und wurden durch eine ID des Nutzers für die „GetMap“-Anfrage verwendet.



a) Auskunftssuchender Marsch Soldat



b) Auskunftssuchender Marsch Gerät

**Abbildung 91: Ergebnisse der Nutzeranfragen**

Zusammenfassend ist bei diesen Tests nicht die Vollständigkeit der Rolle oder des Kontextmodells relevant, sondern es wurde das vorgeschlagene Gesamtkonzept für eine kontextbasierte Visualisierung von Geodaten auf Basis von standardisierten Webdiensten auf Machbarkeit überprüft. Dieses Gesamtkonzept gibt einen Rahmen inklusive der Definition einzelner notwendiger Komponenten vor. Der Prozess der Anfrage inklusive der Anpassung und der Erzeugung der Präsentation verlief mit den Testdaten und den beispielhaft erstellten Kontexten erfolgreich. Ferner wurde das Ergebnis mit Anwendern diskutiert.



## 7. Zusammenfassung und Ausblick

Die große Anzahl an Geodatenbeständen für die Verwaltung, Wirtschaft und Wissenschaft wirft die Frage auf, wie diese nutzer- und aufgabenorientiert verwendet werden können. Die Ausführungen der vorliegenden Arbeit konzentrieren sich auf die Entwicklung eines Rahmens der es erlaubt, spezifische für verschiedene Nutzergruppen angepasste Präsentation in einer Webumgebung standardbasiert zu präsentieren.

Zu diesem Zweck wurde ein Kontextmodell für verschiedene Nutzergruppen basierend auf zwei Szenarien definiert. Desweiteren wurde das generelle Portrayalmodell zu einem kontextabhängigen Portrayalmodell erweitert, um den Kontext berücksichtigen zu können. Außerdem wurde ein kontextabhängiger Portrayalprozess definiert, um Präsentationen für verschiedene Kontexte anzupassen. Adaptionsregeln wurden auf verschiedenen Stufen des Portrayalprozesses analysiert und definiert und die endgültige kontextspezifische Präsentation wurde entsprechend des Kontextmodells generiert. Ferner wurde zu diesem Zweck ein standardisierter Darstellungsdienst (WMS) erweitert. Der Dienst unterstützt verschiedene Operationen. Dies sind „DescribeContextType“ zur Abfrage der Schemabeschreibung eines Kontexttypen und „GetContext“ zur Abfrage von Kontextdokumenten. Das Einfügen, Aktualisieren und Löschen von Kontextinformationen ist über die zusätzliche Operationen „TransactContext“ spezifiziert. Über die erweiterte WMS-Operation „GetMap“ wird die Präsentation abhängig vom Kontextdokument erzeugt. Die Abfrage der Metadaten des Dienstes und der Daten erfolgt über die erweiterte WMS-Operation „GetCapabilities“.

Die Realisierbarkeit des Konzeptes wurde anhand eines Prototyps gezeigt. Das Prototypsystem besteht zum einen aus einer Oracle-Datenbank in der die verschiedenen Quelldaten (wie Geodaten, kontextabhängige Regeln und Signaturen oder Adaptionsregeln) vorliegen. Der andere Teil des Systems ist der Dienst zur Erzeugung und der Client zur Anzeige der Präsentation. Ferner wurde das Ergebnis mit Anwendern diskutiert.

Vor allem im Bereich der Entscheidungsunterstützung spielt die Darstellung von Geodaten eine wichtige Rolle. Durch die Verwendung von Kontextinformationen kann die Visualisierung der Geoinformationen für den Nutzer und seine Situation angepasst werden. Eine nutzer- und situationsabhängige Visualisierung hat einen effektiven Mehrwert und hilft dem Nutzer situationsbezogene Entscheidungen schnell treffen zu können. Der vorgeschlagene Rahmen verbessert die Nutzbarkeit des Portrayaldienstes und unterstützt die „Entscheidungsunterstützungsaktivitäten“ von verschiedenen Nutzergruppen entsprechend ihres Kontextes. Das vorgeschlagene Konzept in dieser Arbeit unterstützt die Visualisierung von Geodaten mit Hilfe von Standards und in Abhängigkeit des Kontextes.

Das Konzept zur kontextbasierten Visualisierung von Geodaten auf Basis von standardisierten Webdiensten ist für einzelne Nutzergruppen definiert worden. Die Kontextelemente für andere Anwendungsbereiche und Aufgaben der Nutzer des erweiterten Portrayalwebdienstes sind nicht zwangsläufig durch das in dieser Arbeit definierte Kontextmodell abgedeckt. Es müssen für verschiedene Anwendungsbereiche die Kontextelemente und somit auch die Adaptionsregeln angepasst werden. Durch die Realisierung in XSD Schema ist eine Erweiterung möglich.

Die weiteren Forschungsbereiche sind vielfältig. Da verschiedene Endgeräte durch den Nutzer verwendet werden, muss sich die Visualisierung an diese Gegebenheit anpassen. Ein Bereich in der Kartographie ist sicherlich die Generalisierung von Geodaten, um diese zum Beispiel auf verschiedenen Bildschirmen

(wie PDA, Handy) optimal darstellen zu können. Der Bereich der Adaption von Nutzerinterfaces erfordert ebenfalls noch weiterführende Analysen.

Forschungsbedarf besteht weiterhin in der semantischen Interoperabilität. Es sind viele verschiedene Geodaten aus unterschiedlichen Quellen vorhanden und sollten diese kontextabhängig visualisiert werden, ist eine einheitliche Semantik notwendig. In diesem Zusammenhang ist auch das Gebiet „Semantic Web“ zu nennen, dessen Ziel es ist, Informationen für Maschinen „verstehbar“ zu machen und Begriffe und Beziehungen zwischen Begriffen (Ontologie) zu verwenden.

Auch im Bereich der Personalisierung bzw. Personalisierungsdienste sollte geforscht werden. Diese stellen personalisierte Funktionen zur Verfügung, die personalisiert aufgefunden, ausgewählt und zusammengestellt werden können. Dieser Aspekt spielt eine wichtige Rolle im „Web 2.0“. Dies beinhaltet gerade im Bereich Kontext die Nutzerakzeptanz, da die Preisgabe von Kontextinformationen durch den Nutzer ein Eingriff in die Privatsphäre darstellen kann.

Desweiteren müssen verschiedene Schnittstellen zwischen den Sensoren zur Aufnahme der Kontextinformationen und dem gesamten vorgeschlagenen System definiert werden. Hier ist zu prüfen, ob OGC genügend Unterstützung für diese Aufgabe bietet.

## 8. Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
ASP	Active Server Pages
CC/PP	Composite Capabilities/Preference Profile
CSW	Catalogue Service for the Web
ebXML	Electronic Business using XML
ebXML RIM	ebXML Register information model
FE	Filter Encoding
GeoInfoDBw	GeoInfo-Dienst der Bundeswehr
GeoRSS	Geo Really Simple Syndication
GEOSS	Global Earth Observation System of Systems
GIF	Graphics Interchange Format
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
INSPIRE	Infrastructure for Spatial Information in Europe
JPEG	Joint Photographic Experts Group
JSP	Java Server Pages
KML	Keyhole Markup Language
MIME	Multipurpose Internet Mail Extensions
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
OWS	Open Web Services
PHP	PHP: Hypertext Preprocessor
PNG	Portable Network Graphics
REST	REpresentational State Transfer
SE	Symbology encoding
SLD	Styled Layer Descriptor encoding
SOA	Service oriented architecture
SOAP	ursprünglich für Simple Object Access Protocol, aber Akronym wird seit der aktuellen Version (1.2) nicht mehr gebraucht
SVG	Scalable Vector Graphics

UDDI	Universal Description, Discovery and Integration
UML	Unified Modeling Language
UMN Mapserv	eine von der University of Minnesota entwickelte CGI-Applikation im Webserver
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WCS	Web Coverage Service
WebCGM	Web Computer Graphics Metafile
WFS	Web Feature Service
WMC	Web Map Context
WMS	Web Map Service
WSDL	Web Services Definition Language
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation

## 9. Literaturverzeichnis

ADV (2005a): *ATKIS-Signaturenkatalog 1: 25000 (ATKIS-SK25), Version 4.2, Bonn.*

ADV (2005b): *ATKIS-Signaturenkatalog 1: 50000 (ATKIS-SK50), Version 1.2, Bonn.*

AOIDH, E. M., KOINIS, A. and BERTOLOTTA, M. (2006): *Improving Archaeological Heritage Information Access Through a Personalised GIS Interface.* In: 6th International Symposium, W2GIS 2006, Hong Kong, China, S. 135-145.

BALDAUF, M., DUSTDAR, S. and ROSENBERG, F. (2007): *A survey on context-aware systems.* In: International Journal of Ad Hoc and Ubiquitous Computing Vol. 2, No. 4, S. 263 - 277.

BARTELME, N. (2005): *Geoinformatik - Modelle, Strukturen, Funktionen.* Springer, Berlin Heidelberg.

BEIERLE, C. and KERN-ISBERNER, G. (2006): *Quantitative Methoden I — Probabilistische Netzwerke.* In: Methoden wissenschaftlicher Systeme, Vieweg, S. 365-416.

BERTIN, J. (1973): *Graphische Seminologie. Diagramme, Netze, Karten.* Walter de Gruyter, Berlin, New York.

BILL, R. (1999): *Grundlagen der Geoinformationssysteme - Hardware, Software und Daten.* Wichmann Verlag, Heidelberg.

BRINKHOFF, T. (2005): *Towards a Declarative Portrayal and Interaction Model for GIS and LBS.* In: Proceedings 8th Conference on Geographic Information Science (AGILE 2005), Estoril, Portugal, S. 449-458.

BROWN, A. (1993): *Map design for screen display.* In: The Cartographic Journal, Vol. 30, S. 129-135.

BROWNE, D., TOTTERDELL, P. and NORMAN, M. (1990): *Adaptive user interfaces.* Academic Press, London.

BRUNNER, K. and NEUDECK, S. (2002): *Grafische und kartografische Aspekte der Bildanzeige.* In: Geowissenschaftliche Mitteilungen, Schriftenreihe der Studienrichtung Vermessungswesen und Geoinformation der Technischen Universität Wien, Vol. Heft 58, S. 77-84.

BRUSILOVSKY, P. (2001): *Adaptive Hypermedia.* In: User Modeling and User Adapted Interaction, S. 87-110.

BRUSILOVSKY, P. (1996): *Methods and techniques of adaptive hypermedia.* In: User Modeling and User Adapted Interaction, S. 87-129.

BRUSILOVSKY, P. and MILLÁN, E. (2007): *User Models for Adaptive Hypermedia and Adaptive Educational Systems.* In: P. Brusilovsky, A. Kobsa and W. Nejdl (Hrsg.): The Adaptive Web - Methods and Strategies of Web Personalization, Springer, Berlin Heidelberg, S. 3-53.

CHEN, G. and KOTZ, D. A. (2000): *A Survey of Context-Aware Mobile Computing Research*

DEY, A. K. (2001): *Understanding and using context.* In: Personal and Ubiquitous Computing, Vol. 5, S. 4-7.

DEY, A. K. and ABOWD, G. D. (2000): *Towards a Better Understanding of Context and Context-Awareness.* In: Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000),

DGIWG (2000): *The Digital Geographic Information Exchange Standard (DIGEST).*

DIN (1988): *44 300 - Informationsverarbeitung - Begriffe*

DOD (1998): *MIL-PRF-89045 (Performance Specification): Geospatial Symbology for Digital Display (GeoSym)*.

DOMIK, G. O. and GUTKAUF, B. (1994): *User Modeling for Adaptive Visualization Systems*. In: Proc. IE-EE Visualization, S. 217 - 223.

DONAUBAUER, A. (2004): *Interoperable Nutzung verteilter Geodatenbanken mittels standardisierter Geo Web Services*. Dissertation, Fakultät für Bauingenieur- und Vermessungswesen, Technischen Universität München.

DUCE, D., HERMAN, I. and HOPGOOD, B. (2002): *Web 2D Graphics File Formats*. In: COMPUTER GRAPHICS forum, Vol. 21, No. 1, S. 43-64.

EBXMLRR (2009): *Registry Capability Comparison Matrix*, [http://ebxmlrr.sourceforge.net/tmp/Registry\\_Capability\\_Matrix.html](http://ebxmlrr.sourceforge.net/tmp/Registry_Capability_Matrix.html) (zuletzt besucht am 23.11.2010).

EBXMLWIKI (2009): <http://ebxmlrr.sourceforge.net/wiki/index.php/Overview/comparisonWithUDDI>, (zuletzt besucht am 23.11.2010).

FIELDING, R. T. (2000): *Architectural Styles and the Design of Network-based Software Architectures*. Dissertation, University of California, Irving.

GARCIA-BARRIOIS, V., GÜTL, C. and TRUMMER, C. (2005): *Personalisation versus Adaptation? A User-centred Model Approach and its Application*. In: International Conference on Knowledge Management, S. 120 - 127.

GEOPORTAL.BUND (2010): *Nutzerumfrage Geoportal.Bund*, [http://www.geobranchen.de/images/stories/index.php?option=com\\_geonews&page=details&id=4491](http://www.geobranchen.de/images/stories/index.php?option=com_geonews&page=details&id=4491), (zuletzt besucht am 23.11.2010).

GEOZG (2009): *Geodatenzugangsgesetz - Gesetz über den Zugang zu digitalen Geodaten*, <http://www.bgblportal.de/BGBL/bgbl1f/bgbl109s0278.pdf>, zuletzt besucht am 23.11.2010).

GPSMAGAZINE (2007): *GPS Magazine*, [http://www.gpsmagazine.com/assets/review-loox/loox\\_n100\\_day\\_night\\_map.jpg](http://www.gpsmagazine.com/assets/review-loox/loox_n100_day_night_map.jpg), (zuletzt besucht am 23.11.2010).

HAGOPIAN, J. and MAXWELL, T. (1996): *Explicit and implicit resources: a simplified approach to user requirements modeling*, <http://www.op.dlr.de/SpaceOps/spops96/misplan/track3.htm>, (zuletzt besucht am 23.11.2010).

HAKE, G., GRÜNREICH, D. and MENG, L. (2002): *Kartographie*. de Gruyter, Berlin.

HARZER, B. (2008): *GIS-Report 2008/09*. Bernhard Harzer Verlag GmbH, Karlsruhe.

IETF (1999): *HyperText Transfer Protocol - HTTP/1.1*, Internet Engineering Task Force.

IETF (1996): *Multipurpose Internet Mail Extensions (MIME)*.

IETF (2005): *Uniform Resource Identifier (URI): Generic Syntax*, Internet Engineering Task Force.

IRWIN, W. and CHURCHER, N. (2002): *XML in the Visualisation Pipeline*. In: Selected papers from 2001 Pan-Sydney Area Workshop on Visual Information Processing (VIP2001), Sidney, Australia, S. 59-67.

ISO (2009b): *CD ISO 19117: Portrayal*.



- ISO (2009a): *CD ISO 19151: 2009 - Geographic information -- Logical location identification scheme.*
- ISO (2000): *ISO 19101: 2000 - Geographic information -- Reference model.*
- ISO (2005d): *ISO 19110: 2005 - Geographic information -- Methodology for feature cataloguing.*
- ISO (2005a): *ISO 19117:2005 - Geographic information -- Portrayal.*
- ISO (2005c): *ISO 19119: 2005 - Geographic information -- Services.*
- ISO (2005b): *ISO 19128:2005 - Geographic information -- Web map server interface.*
- ISO (2010): *ISO 19143:2005 - Geographic information -- Filter Encoding.*
- ISO (2007): *ISO/TC 211 N 2165 - The applicability of ebXML RIM and advise to the committee according to resolution 358 from Riyadh, study document.*
- JAMESON, A. (2001): *Modelling both the Context and the User.* In: Personal Ubiquitous Computing, Vol. 5, No. 1, S. 29-33.
- JÖST, M. (2009): *Personalized City Tours - An Extension of the OGC OpenLocation Specification.* Dissertation, Ruprechts-Karls-Universität Heidelberg.
- JP2IE (2009): <http://www.fnordware.com/j2k/>, (zuletzt besucht am 23.11.2010).
- KHALAF, R. and LEYMAN, F. (2003): *On Web Services Aggregation.* In: Technologies for E-Services, LNCS 2819, Springer-Verlag Berlin Heidelberg, S. 1-13.
- KOCH, N. (2000): *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process.* Dissertation, Informatik, Universität München.
- KRAAK, M. J. and BROWN, A. (2000): *Web Cartography - developments and prospects.* Taylor & Francis, London.
- KRAAK, M. J. and ORMELING, F. (1996): *Cartography: visualization of spatial data* Longman, Harlow, UK.
- KRAMERS, R. E. (2006): *The Atlas of Canada – User Centred Development.* In: W. Cartwright, M. P. Peterson and G. F. Gartner (Hrsg.): *Multimedia Cartography*, Springer, Berlin, S. 139–160.
- KUIPER, P. M., VAN DIJK, E. M. A. G. and BOERMA, A. K. (2006): *Adaptive Municipal e-forms.* In: Proceedings of Workshops held at the Fourth International Conference on Adaptive Hypermedia & Adaptive Web-based Systems (AH2006), Dublin, Ireland, S. 484-488.
- KUSCHKE, M. and WÖLFEL, L. (2002): *Web Services kompakt.* Spektrum Akademischer Verlag, Heidelberg Berlin.
- LEUKERT, K. and REINHARDT, W. (2000): *GIS-Internet architectures.* In: International Archives of Photogrammetry and Remote Sensing (IAPRS), Vol. 33, No. Part B4, S. 572-578.
- LEUTNER, D. (2002): *Adaptivität und Adaptierbarkeit multimedialer Lehr- und Informationssysteme.* In: L. Issing and P. Klimsa (Hrsg.): *Information und Lernen mit Multimedia und Internet: Lehrbuch für Studium und Praxis*, Beltz-Verlag, Weinheim, S. 115-125.

- LI, J., YANG, C., ZHANG, J. and YANG, J. (2004): *A new Web mapping architecture based on SLD and pattern*. In: Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International, S. 2893- 2895.
- LI, Z. L. and KRAAK, M.-J. (2002): *Web-based exploratory data analysis (Web-EDA)*. In: Visualisation meets spatial analysis, Proceedings of the ISPRS Commission II Symposium, Xi'an, China, S. 281-286.
- LIPP, T. W. (1997): *Grafikformate*. Microsoft Press, 1997, Unterschleißheim.
- MACEACHREN, A. and TAYLOR, D. R. F. (1994): *Visualization in Modern Cartography*. Pergamon, Great Yarmouth.
- MATHIAK, B., KUPFER, A. and NEUMANN, K. (2005): *Modellierung und kartographische Visualisierung von Geodaten mit XML-basierten Sprachen* In: Informatik - Forschung und Entwicklung, Vol. 20, No. 1-2, S. 24-32.
- MELZER, I., DOSTAL, W., JECKLE, M. and ZENGLER, B. (2007): *Service-orientierte Architekturen mit Web Services*. Spektrum Akademischer Verlag, München.
- MENG, L. (2004): *Methoden zur Gestaltung egozentrischer Karten*. In: Kartographische Schriften, Vol. 9, S. 113-119.
- MENG, L. (2003): *Missing Theories and Methods in Digital Cartography*. In: 21st International Cartographic Conference Durban 2003, S. 1887-1893.
- MENG, L. (2006): *Mobile Geovisualisierungsdienste und ihre Gebrauchstauglichkeit*. In: Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover, Vol. 263, S. 249-260.
- MOISUC, B., GENSEL, J., DEVOINE, P.-A. and MARTIN, H. (2006): *Designing Adaptive Spatio-temporal Information Systems for Natural Hazard Risks with ASTIS*. In: W2GIS 2006, Hong Kong, China, S. 146-157.
- MULGUND, S. and LANDSMAN, S. (2007): *User defined operational pictures or tailored situation awareness*. In: 12th International Command and Control Research and Technology Symposium, Newport, RI.
- NEUDECK, S. (2001): *Zur Gestaltung topografischer Karten für die Bildschirmvisualisierung*. Dissertation, Universität der Bundeswehr München.
- NEUMANN, A. (2008): *Web Mapping and Web Cartography*. In: S. Shekhar and H. Xiong (Hrsg.): *Encyclopedia of GIS*, Springer, S. 1261.
- NIELSEN, J. (1993): *Usability Engineering*. Boston, AP Professional.
- NIVALA, A. M. and SARJAKOSKI, L. T. (2003): *Need for Context-Aware Topographic Maps in Mobile Devices*. In: ScanGIS'2003 -Proceedings of the 9th Scandinavian Research Conference on Geographical Information Science, Espoo, Finland, S. 15-29.
- O'REILLY, T. (2005): *What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software* <http://www.oreilly.de/artikel/web20.html>, (zuletzt besucht am 23.11.2010).
- OASIS (2005): *ebXML Registry Information Model v3.0, OASIS Standard*.
- OASIS (2002): *UDDI Version 3.0, Spec Technical Committee Specification*.
- OGC (2009a): *OGC CSW-ebRIM Registry Service - Part 1: ebRIM profile of CSW 1.0.1*.

- OGC (2009b): *OGC CSW-ebRIM Registry Service – Part 2: Basic extension package 1.0.1*.
- OGC (2008b): *OGC KML 2.2.0*.
- OGC (2008a): *OGC Reference Model 2.0*.
- OGC (2007b): *OGC Styled Layer Descriptor profile of the Web Map Service Implementation Specification 1.1.0*.
- OGC (2006a): *OGC Symbology Encoding Version 1.1.0 (revision 4)*.
- OGC (2007a): *OpenGIS Catalogue Services Specification 2.0.2*.
- OGC (2005b): *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture*.
- OGC (2005c): *OpenGIS Location Service (OpenLS) Implementation Specification: Core Services*.
- OGC (2005a): *OpenGIS Web Map Context Implementation Specification*.
- OGC (2007c): *OpenGIS Web Processing Service 1.0.0*.
- OGC (2006b): *Symbology Management 0.3.0 - IPR*.
- OGCHISTORY (2009): <http://www.opengeospatial.org/ogc/historylong>, (zuletzt besucht am 23.11.2010).
- OPPERMANN, R. and SPECHT, M. (1999): *Adaptice Mobile Museum Guide for Information and Learning on Demand*. In: HCI International '99, S. 642-646.
- ORACLEXDK (2007): [http://download.oracle.com/docs/cd/B19306\\_01/appdev.102/b14252/adx\\_overview.htm](http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14252/adx_overview.htm), (zuletzt besucht am 23.11.2010).
- PETIT, M., RAY, C. and CLARAMUNT, C. (2006): *A Contextual Approach for the Development of GIS: Application to Maritime Navigation*. In: W2GIS 2006, S. 158-169.
- PETZOLD, J. (2005): *Zustandsprädiktoren zur Kontextvorhersage in ubiquitären Systemen*. Dissertation, Universität Augsburg.
- PUTZ, S. (1994): *Interactive information services using World-Wide Web hypertext*. In: CERN: First International Conference on the World-Wide Web, S. 273-280.
- REICHENBACHER, T. (2003): *Adaptive Methods for Mobile Cartography*. In: Proceedings of the 21st International Cartographic Conference ICC: Cartographic Renaissance, Durban, South Africa, S. 1311-1321.
- REICHENBACHER, T. (2004): *Mobile Cartography – Adaptive Information on Mobile Devices*. Dissertation, TU München.
- REICHENBACHER, T. (2008): *Mobile Usage and Adaptive Visualization*. In: S. Shekhar and H. Xiong (Hrsg.): *Encyclopedia of GIS*, Springer, S. 677-682.
- REPRESENTATIONS (2009): [http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An\\_overview\\_of\\_representations](http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An_overview_of_representations), (zuletzt besucht am 23.11.2010).

- ROBINSON, A. H. (1952): *Look of Maps - An examination of cartographic design*. Univ. of Wisconsin Pr., Madison, Wis.
- ROTHERMEL, K. (2008): *Kontextbezogene Systeme – die Welt im Computer modelliert*. In: Digitale Visionen, Springer Berlin Heidelberg, S. 31-42.
- ROTHERMEL, K., ERTL, T., FRITSCH, D., KÜHN, P. J., MITSCHANG, B., WESTKÄMPER, E., BECKER, C., DUDKOWSKI, D., GUTSCHER, A., HAUSER, C., JENDOUBI, L., NICKLAS, D., VOLZ, S. and WIELAND, M. (2006): *SFB 627 – Umgebungsmodelle für mobile kontextbezogene Systeme*. In: Informatik - Forschung und Entwicklung, Vol. 21(1-2), S. 105-113.
- RRZN (2002): *XML 1.0 - Grundlagen* Regionales Rechenzentrum für Niedersachsen (RRZN), Hannover.
- SAIDANI, O. and NURCAN, S. (2007): *Towards Context Aware Business Process Modelling*. In: 8th Workshop on Business Process Modeling, Development, and Support (BPMDS'07), Trondheim.
- SARJAKOSKI, L. T. and NIVALA, A.-M. (2005): *Adaptation to Context - A Way to Improve the Usability of Mobile Maps*. In: L. Meng, A. Zipf and T. Reichenbacher (Hrsg.): *Map-based Mobile Services, Theories, Methods and Implementations*, Springer Berlin Heidelberg New York, S. 107-123.
- SARJAKOSKI, L. T. and SARJAKOSKI, T. (2008): *User Interfaces and Adaptive Maps*. In: S. Shekhar and H. Xiong (Hrsg.): *Encyclopedia of GIS* Springer, Heidelberg, S. 1205-1212.
- SARJAKOSKI, T. and SARJAKOSKI, L. T. (2007): *A Real-Time Generalisation and Map Adaptation Approach for Location-Based Services*. In: W. A. Mackaness, A. Ruas and L. T. Sarjakoski (Hrsg.): *Generalisation of Geographic Information: Cartographic Modelling and Applications*, International Cartographic Association, Elsevier, S. 137-159.
- SAYAR, A., PIERCE, M. and FOX, G. (2006): *Integrating AJAX Approach into GIS Visualization Web Services*. In: IEEE International Conference on Internet and Web Applications and Services, ICIW'06, S. 169.
- SAYAR, A., PIERCE, M. and FOX, G. (2005): *Message Exchange for Web Service-Based Mapping Services*. In: *GeoInformatics '05*, Toronto, Canada
- SCHILIT, B. and THEIMER, M. (1994): *Disseminating active map information to mobile hosts*. In: IEEE Network, Vol. 8, S. 22-32.
- SCHIMPF, C.-A. and ULLFORS, C.-M. (1994): *Bertelsmann-Lexikon Informatik - EDV - Computertechnik*. Lexikon-Institut Bertelsmann, Gütersloh.
- SCHMIDT-BELZ, B., LAAMANEN, H., POSLAD, S. and ZIPF, A. (2003): *Location-based mobile tourist services - first user experiences*. In: *Proceedings of ENTER 2003. Int. Congress on Tourism and Communications Technologies*, Helsinki, Finland.
- SCHMIDT, A. (2002): *Ubiquitous Computing - Computing in Context*. Dissertation, Computing Department, Lancaster University, U.K..
- SCHMIDT, A. and GELLERSEN, H. (2001): *Nutzung von Kontext in ubiquitären Informationssystemen*, S. 83-90.
- SCHNEIDER-HUFSCHMIDT, M., KÜHME, T. and MALINOWSKI, U. (1993): *Adaptive User Interfaces: Principles and Practice*. Amsterdam: North-Holland.
- SPECHT, M. (1998): *Adaptive Methoden in computerbasierten Lehr/Lernsystemen*. Dissertation, GMD Forschungszentrum Informationstechnik, Sankt Augustin, Trier University.

- STEINIGER, S. and BOCHER, E. (2009): *An overview on current free and open source desktop GIS developments*. In: International Journal of Geographical Information Science, Vol. 24.
- STRANG, T. and LINNHOF-POPIEN, C. (2004): *A Context Modeling Survey*. In: First International Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp, S. 34-41.
- TAIT, M. G. (2005): *Implementing geoportals: applications of distributed GIS*.
- TANNENBAUM, A. S. (2003): *Computernetzwerke*. Prentice Hall Verlag, Harr bei München.
- TATNALL, A. (2005): *Portals, Portals Everywhere*. In: Web Portals: The New Gateways to Internet Information and Services, S. 1-14.
- TEUFERT, J. F. and TRABELSI, M. (2006): *Development of NATO's recognized environmental picture*. In: Proc. SPIE 6201, 62011D.
- TRAN, T., CIMIANO, P. and ANKOLEKAR, A. (2008): *A Rule-based Adaption Model for Ontology-based Personalization*. In: Advances in Semantic Media Adaptation and Personalization, Springer, S. 117-135.
- U.S. GOVERNMENT PRINTING OFFICE, W. D. (2008): *Field Manual 3-0, Operations*.
- VAN DER VLIST, E. (2002): *XML Schema*. O'Reilly, Sebastopol, CA.
- VASSILEVA, J. (1996): *A task-centered approach for user modeling in a hypermedia office documentation system*. In: User Modeling and User-Adapted Interaction 6 S. 185 - 224.
- W3C (2000): *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation.
- W3C (1999): *HTML 4.01 Specification*, W3C Recommendation.
- W3C (2006): *Scalable Vector Graphics (SVG) Full 1.2 Specification*, W3C Recommendation.
- W3C (2004a): *Web Services Architecture*, W3C Working Group Note.
- W3C (2002): *Web Services Description Requirements*, W3C Working Draft.
- W3C (2007): *WebCGM 2.0*, W3C Recommendation.
- W3C (2004c): *XML Schema Part 1: Structures*, W3C Recommendation.
- W3C (2004d): *XML Schema Part 2: Datatypes*, W3C Recommendation.
- W3C (1999): *XSL Transformations (XSLT) Version 1.0*, W3C Recommendation.
- WALTER, T. (2008): *Kompendium der Web-Programmierung*. Springer, Berlin.
- WEAKLIAM, J., WILSON, D. and BERLOTTO (2008): *Personalising Map Feature Content for Mobile Map Users*. In: M. e. al. (Hrsg.): Map-based mobile services – Design, Interaction and Usability, S. 125-145.
- WEB2.0 (2009): <http://www.web2null.de/category/geodaten/page/1> (zuletzt besucht am 23.11.2010).
- WEBMAPPING (2009): [http://en.wikipedia.org/wiki/Web\\_mapping](http://en.wikipedia.org/wiki/Web_mapping), (zuletzt besucht am 23.11.2010).
- WIEBROCK, I. and REINHARDT, W. (2008b): *A context-aware user group model for web portrayal services*. In: 5th International Conference on Geographic Information Science, Park City, Utah, USA, S. 325-330.

WIEBROCK, I. and REINHARDT, W. (2005): *Präsentation von Karten im Internet – Ein Vergleich der Vektorformate SVG und ActiveCGM*. In: Mitteilungen aus dem Bundesamt für Kartographie und Geodäsie, Vol. 34, S. 131-140.

WIEBROCK, I. and REINHARDT, W. (2004): *Presentation of maps on the Internet – A Comparison of the vector formats SVG and ActiveCGM*. In: 24th Urban Data Management Symposium, S. Proceedings on CD-ROM.

WILKINSON, L. (1999): *The grammar of graphics* Springer, New York.

YU, Z., WANG, Y. and REICHENBACHER, T. (2003): *User modeling for geo-database adaption*. In: Proceedings of the 21st International Cartographic Conference (ICC), S. Proceedings on CD-ROM.

ZIEGLER, J., LOHMANN, S. and KALTZ, J. W. (2005): *Kontextmodellierung für adaptive webbasierte Systeme*. In: Mensch & Computer 2005: Kunst und Wissenschaft – Grenzüberschreitungen der interaktiven ART, München, S. 181-189.

ZIGORIS, P. and ZHANG, Y. (2006): *Bayesian adaptive user profiling with explicit & implicit feedback*. In: Conference on Information and Knowledge Management, Arlington, Virginia, USA, S. 397 - 404.

ZIPF, A. (2005): *Using Styled Layer Descriptor (SLD) for the Dynamic Generation of User- and Context-Adaptive Mobile Maps – A Technical Framework*. In: 5th International Workshop W2GIS 2005, Proceedings, Lausanne, Switzerland, S. 183-193.

ZIPF, A. and JÖST, M. (2006): *Implementing adaptive mobile GI services based on ontologies: Examples from pedestrian navigation support*. In: Computers, Environment and Urban Systems, Vol. 30, No. 6, S. 784-798.

## 10. Abbildungsverzeichnis

Abbildung 1 : Web Service Architektur nach W3C (2004a).....	14
Abbildung 2 : Web Service Architektur .....	17
Abbildung 3 : Servicedefinition nach ISO 19119 (ISO (2005c)) .....	19
Abbildung 4 : Service tiers, OGC Reference Model 2.0 (OGC (2008a)).....	19
Abbildung 5 : Portrayalprozess - Präsentation von Geoinformationen .....	20
Abbildung 6 : Graphische Variablen nach BERTIN (1973) aus NEUDECK (2001).....	21
Abbildung 7 : Allgemeine Darstellungsregel .....	22
Abbildung 8 : Teil A, ATKIS-SK-50 Beispiel, ATKIS-SK-Spezifikation (ADV (2005b)) .....	24
Abbildung 9 : Teil B, ATKIS-SK-50 Beispiel, ATKIS-SK-Spezifikation (ADV (2005b)) .....	24
Abbildung 10 : Schematische Abbildung der Ableitungsregeln nach ATKIS-SK aus Abb. 8 nach Definition 2.9 .....	24
Abbildung 11 : GeoSym Beispiel Vmap Level 2, GeoSym-Spezifikation (DOD (1998)).....	25
Abbildung 12 : Schematische Abbildung von Abb. 11 nach Definition 2.9 .....	25
Abbildung 13 : Visualisierung von Geodaten in einem Desktop-GIS.....	27
Abbildung 14 : „Feature Class“-Tabelle mit den zusätzlichen Spalten „RuleID“ und „Override“ .....	27
Abbildung 15 : Kartographische Repräsentation - Workflow in ArcGIS 9.2.....	28
Abbildung 16 : Erstellung der kartographischen Repräsentation in ArcGIS 9.2.....	28
Abbildung 17 : Multiple Repräsentationen derselben Phänomene (REPRESENTATIONS (2009)).....	29
Abbildung 18 : Datenbankdienste nach BARTELME (2005) .....	30
Abbildung 19 : Visualisierung von Geodaten mit Hilfe eines Portrayalwebdienstes .....	31
Abbildung 20 : Überblick der ISO 19117 – Portrayal, modifiziert .....	34
Abbildung 21 : Regeln nach ISO 19117 - Portrayal .....	35
Abbildung 22 : Portrayal Specification (vereinfacht) und Beispiel.....	36
Abbildung 23 : Portrayalprozess nach CD ISO 19117 – Portrayal (ISO (2009b)).....	37
Abbildung 24 : „Core (gelb) and Extension (blau) Packages“ mit Abhängigkeiten nach ISO 19117 – Portrayal (ISO (2009b)).....	38
Abbildung 25 : „FeatureTypeStyle“-Element im Symbology Encoding.....	39
Abbildung 26 : OGC Portrayal Pipeline nach OGC Reference Model 2.0. (OGC (2008a)), modifiziert ..	41
Abbildung 27 : Beziehung SE und SLD .....	43
Abbildung 28 : SLD mit SE-Elementen .....	43
Abbildung 29 : „Integrated SLD-WMS“ und „Component SLD-WMS“ .....	44
Abbildung 30 : Adaptive und adaptierbare Systeme .....	48
Abbildung 31 : Dimensionen eines Nutzermodells aus SCHNEIDER-HUFSCHMIDT et al. (1993).....	50
Abbildung 32 : Overlaymodell und Wissensmodell.....	50
Abbildung 33 : Adaption basierend auf Nutzer- bzw. Kontextmodell .....	52
Abbildung 34 : Erkundungstechnologien nach SCHMIDT (2002) .....	53
Abbildung 35 : Thematische Schwerpunkte bei Nexus (ROTHERMEL et al. (2006)).....	54
Abbildung 36 : Kontextbaum nach SAIDANI and NURCAN (2007) für den Aspekt Zeit (time).....	55
Abbildung 37 : Kontext aus nutzer- und systemorientierter Sichtweise (BRUSILOVSKY and MILLÁN (2007)) .....	56
Abbildung 38 : Nutzereigenschaften mit verschiedenen adaptiven Websystemen (BRUSILOVSKY and MILLÁN (2007)).....	56
Abbildung 39 : „Visualization in terms of map use“ (MACEACHREN and TAYLOR (1994)) .....	58
Abbildung 40 : „Visiting points“ nach YU et al. (2003).....	59
Abbildung 41 : Präsentationsmodul nach MOISUC et al. (2006) .....	59

---

Abbildung 42 : Adaptives GIS nach PETIT et al. (2006).....	60
Abbildung 43 : Geovisualisierungsdienst nach REICHENBACHER (2004) .....	62
Abbildung 44 : GiMoDig Service architecture nach SARJAKOSKI and SARJAKOSKI (2007) .....	63
Abbildung 45 : Standardisierung der Anforderungen an UPA (ISO 19154) .....	64
Abbildung 46 : Kontextabhängiger Portrayalprozess .....	68
Abbildung 47 : Routenplanung (a) und Routenanzeige (b) .....	69
Abbildung 48 : Tag- (a) und Nachtsicht (b) (GPSMAGAZINE (2007)) .....	70
Abbildung 49 : Marschplanung Soldat (a) und Marschplanung Gerät (b).....	71
Abbildung 50 : Anzeige „Marsch Soldat“(a) und „Marsch Gerät“ (b).....	71
Abbildung 51 : Kontextmodell in UML .....	80
Abbildung 52 : Relationen zwischen Kontextmodell und Funktionen .....	82
Abbildung 53 : Sekundärkontext .....	85
Abbildung 54 : Sekundärkontext in Abhängigkeit zum Primärkontext.....	86
Abbildung 55 : Workflow der Kontextinformationen .....	86
Abbildung 56 : Kontextabhängiges Portrayal-Modell .....	87
Abbildung 57 : Darstellungsänderungsliste .....	88
Abbildung 58 : Kontextabhängiger Portrayalprozess mit Werkzeugauswahl .....	91
Abbildung 59 : Objektfilterregeln.....	94
Abbildung 60 : Portrayaladaptionsregeln .....	96
Abbildung 61 : Graphische Adaptionsregeln mit Renderingprozess.....	98
Abbildung 62 : Funktionale Auswahlregel .....	99
Abbildung 63 : Anpassung der Präsentation in Pseudocode.....	100
Abbildung 64 : „Adapt Geodata“ in Pseudocode.....	100
Abbildung 65 : „Adapt Portrayal“ in Pseudocode .....	101
Abbildung 66 : „Adapt Graphic“ in Pseudocode .....	102
Abbildung 67 : „Adapt Functions“ in Pseudocode .....	102
Abbildung 68 : Abfrage in SQL und Filter Encoding.....	103
Abbildung 69 : Angepasste „Portrayal Pipeline“ (a), Original (b) .....	104
Abbildung 70 : DescribeContextType Operation .....	105
Abbildung 71 : GetContext Operation.....	106
Abbildung 72 : TransactContext Operation.....	106
Abbildung 73 : „TransactContext“-Aufruf .....	107
Abbildung 74 : „Insert“-Transaktion .....	107
Abbildung 75 : GetMap Operation .....	108
Abbildung 76 : GetCapabilities Operation.....	109
Abbildung 77 : „ContextType“-Element im „Capabilities“-Dokument .....	110
Abbildung 78 : Erweitere Operationen im „Capabilities“-Dokument .....	110
Abbildung 79 : WPS „Execute“ Operation und „integrated“ SLD-WMS .....	111
Abbildung 80 : Architektur (DB – Server - Client) .....	114
Abbildung 81 : Oracletabellenansicht.....	114
Abbildung 82 : Portrayalregeln und Signaturen in Oracle-Tabelle .....	115
Abbildung 83 : Nutzerprofil in Oracle-Tabelle.....	115
Abbildung 84 : Oracle XML Developer's Kit.....	116
Abbildung 85 : Auszug aus JSP-Datei.....	116
Abbildung 86 : „Select extractValue“ XMLType-Funktion.....	117



---

Abbildung 87 : Get Capabilities Ergebnis .....	118
Abbildung 88 : DescribeContextType Ergebnis .....	119
Abbildung 89 : GetContext Ergebnis.....	120
Abbildung 90 : Upload Operation .....	120
Abbildung 91 : Ergebnisse der Nutzeranfragen.....	121



## 11. Tabellenverzeichnis

Tabelle 1 : Standards und Normen im Bereich Portrayal (Stand 02/10) .....	32
Tabelle 2 : Gegenüberstellung Geodienste allgemein und Portrayaldienste .....	40
Tabelle 3 : Web Map Context (WMC) Parameter.....	45
Tabelle 4 : Kontextkategorien nach NIVALA and SARJAKOSKI (2003).....	62
Tabelle 5 : Nutzerrolle – Geodaten, Signaturierung/Bildschirmanzeige und Werkzeuge, Radwanderung.	72
Tabelle 6 : Nutzerrollen – Geodaten, Signaturierung/Bildschirmanzeige und Werkzeuge, Marsch.....	73
Tabelle 7 : Nutzerrollen und ihre Eigenschaften .....	75
Tabelle 8 : Modell Ziegler, Nivala und Sarjakoski, Reichenbacher und eigenes Gesamtmodell.....	77
Tabelle 9 : Statische Elemente .....	78
Tabelle 10 : Dynamische Elemente.....	78
Tabelle 11 : Kontextelemente.....	78
Tabelle 12 : XML Schema (Kontext).....	81
Tabelle 13 : Relationen und Begründungen mit Beispielen.....	83
Tabelle 14 : Objektfilterregeln .....	93
Tabelle 15 : Portrayaladaptionsregeln .....	95
Tabelle 16 : Graphische Adaptionsregeln .....	97
Tabelle 17 : Funktionale Auswahlregel.....	99
Tabelle 18 : Idgen Wert.....	107
Tabelle 19 : XSQL Action Element Handler nach OracleXDK (2007) .....	147
Tabelle 20 : XMLType-Funktionen .....	148



## 12. Anhang

### 12.1. XSDs (Kontextschema) und XML-Beispieldatei (Kontext)

XSD Kontextschema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml">
  <xs:import namespace="http://www.opengis.net/gml" schemaLocation="C:\Dokumente und Ein-
stellungen\wiebrock.AGIS\Desktop\gml\3.1.1\base\geometryBasic0d1d.xsd"/>
  <xs:element name="C_System" type="C_System"/>
  <xs:complexType name="C_System">
    <xs:sequence>
      <xs:element name="BandWidth" type="BandWidthType"/>
      <xs:element name="C_Device" type="C_Device"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="C_Dynamic" type="C_Dynamic"/>
  <xs:complexType name="C_Dynamic">
    <xs:sequence>
      <xs:element name="id" type="xs:int"/>
      <xs:element name="Situation" type="Situation"/>
      <xs:element name="Physical" type="Physical"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="C_User" type="C_User"/>
  <xs:complexType name="C_User">
    <xs:sequence>
      <xs:element name="id" type="xs:int"/>
      <xs:element name="Preferences" type="Preferences"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="C_Context" type="C_Context"/>
  <xs:complexType name="C_Context">
    <xs:sequence>
      <xs:element name="id" type="xs:int"/>
      <xs:element name="dynamic" type="C_Dynamic"/>
      <xs:element name="system" type="C_System"/>
      <xs:element name="C_User" type="C_User"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="C_Device" type="C_Device"/>
  <xs:complexType name="C_Device">
    <xs:sequence>
      <xs:element name="devicetype" type="C_DeviceTypeCode"/>
      <xs:element name="resolution" type="C_Resolution"/>
      <xs:element name="size" type="C_Size"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="C_DeviceTypeCode">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Notebook"/>
      <xs:enumeration value="PCMonitor"/>
      <xs:enumeration value="TabletPC"/>
      <xs:enumeration value="PDA"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="Preferences" type="Preferences"/>
  <xs:complexType name="Preferences">
    <xs:sequence>
```

```

        <xs:element name="domai" type="C_DomainType"/>
        <xs:element name="role" type="C_Role"/>
        <xs:element name="use" type="C_Use"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Situation" type="Situation"/>
<xs:complexType name="Situation">
    <xs:sequence>
        <xs:element name="date" type="xs:date"/>
        <xs:element name="position" type="gml:PointType"/>
        <xs:element name="time" type="xs:dateTime"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Physical" type="Physical"/>
<xs:complexType name="Physical">
    <xs:sequence>
        <xs:element name="LightCondition" type="LightConditionType"/>
        <xs:element name="Weather" type="WeatherType"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="C_Resolution">
    <xs:restriction base="xs:string">
        <xs:enumeration value="320x240"/>
        <xs:enumeration value="480x320"/>
        <xs:enumeration value="640x480"/>
        <xs:enumeration value="800x600"/>
        <xs:enumeration value="1400x1050"/>
        <xs:enumeration value="1680x1050"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="C_Role">
    <xs:list>
        <xs:simpleType>
            <xs:union>
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration val-
ue="View_bicycle"/>
                        <xs:enumeration val-
ue="GIS_analysis_bicycle"/>
                        <xs:enumeration val-
ue="View_march_soldier"/>
                        <xs:enumeration val-
ue="View_march_equipment"/>
                        <xs:enumeration val-
ue="Operational_commander_march"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:union>
        </xs:simpleType>
    </xs:list>
</xs:simpleType>
<xs:simpleType name="BandWidthType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Analog"/>
        <xs:enumeration value="ISDN"/>
        <xs:enumeration value="UMTS"/>
        <xs:enumeration value="DSL1000"/>
        <xs:enumeration value="DSL2000"/>
        <xs:enumeration value="DSL3000"/>
        <xs:enumeration value="DSL6000"/>
        <xs:enumeration value="DSL16000"/>
    </xs:restriction>
</xs:simpleType>

```

```

        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="C_DomainType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Mili"/>
            <xs:enumeration value="Topo"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="LightConditionType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="day"/>
            <xs:enumeration value="night"/>
            <xs:enumeration value="twilight"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="WeatherType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="rain"/>
            <xs:enumeration value="sun"/>
            <xs:enumeration value="fog"/>
            <xs:enumeration value="snow"/>
            <xs:enumeration value="clouds"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="C_Size">
        <xs:restriction base="xs:string">
            <xs:enumeration value="3"/>
            <xs:enumeration value="5"/>
            <xs:enumeration value="7"-9"/>
            <xs:enumeration value="9"-11"/>
            <xs:enumeration value="11"-13"/>
            <xs:enumeration value="13"-15"/>
            <xs:enumeration value="15"-17"/>
            <xs:enumeration value="19"-21"/>
            <xs:enumeration value="21"-23"/>
            <xs:enumeration value="23"-25"/>
            <xs:enumeration value="25"-27"/>
            <xs:enumeration value="27"-30"/>
            <xs:enumeration value="30"-33"/>
            <xs:enumeration value="33"-36"/>
            <xs:enumeration value="36"-39"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="C_Use">
        <xs:restriction base="xs:string">
            <xs:enumeration value="mobile"/>
            <xs:enumeration value="stationary"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>

```

XML-Beispieldatei nach UML-Modell:

```

<?xml version="1.0" encoding="UTF-8"?>
<!--XML-Beispieldatei von XMLSpy generiert v2009 sp1 (http://www.altova.com)-->
<C_Context xsi:noNamespaceSchemaLocation="testneu_280110.xsd"
xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <id>0</id>
    <dynamic>
        <id>0</id>

```

```

    <Situation>
      <date>2007-08-13</date>
      <position>
        <gml:pos>45.256 -71.92</gml:pos>
      </position>
      <time>2009-12-09T09:30:45Z</time>
    </Situation>
    <Physical>
      <LightCondition>day</LightCondition>
      <Weather>rain</Weather>
    </Physical>
  </dynamic>
  <system>
    <BandWidth>DSL2000</BandWidth>
    <C_Device>
      <devicetype>Notebook</devicetype>
      <resolution>800x600</resolution>
      <size>19&apos;&apos;-21&apos;&apos;</size>
    </C_Device>
  </system>
  <C_User>
    <id>8</id>
    <Preferences>
      <domai>Mili</domai>
      <role>View_march_soldier</role>
      <use>mobile</use>
    </Preferences>
  </C_User>
</C_Context>

```

## 12.2. Erweiterung GetCapabilities Dokument

```

<?xml version="1.0" encoding="UTF-8"?> <schema xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:wms="http://www.opengis.net/wms" xmlns:ctxt="http://www.opengis.net/ctxt"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema" target-
Namespace="http://www.opengis.net/ctxt" elementFormDefault="qualified">
  <import namespace="http://www.opengis.net/wms" schemaLoca-
tion="wmscapabilities_1_3_0.xsd"/>
  <annotation> <documentation> Context version 1.0 </documentation> </annotation>
  <element name="ContextType" substitutionGroup="wms:_ExtendedCapabilities">
    <complexType>
      <sequence>
        <element ref="wms:Name" minOccurs="0"/>
        <element ref="wms:Abstract" minOccurs="0"/>
      </sequence>
    </complexType>
  </element>
  <element name="DescribeContextType" type="wms:OperationType" substitution-
Group="wms:_ExtendedOperation"/>
  <element name="GetContext" type="wms:OperationType" substitution-
Group="wms:_ExtendedOperation"/>
  <element name="TransactContext" type="wms:OperationType" substitution-
Group="wms:_ExtendedOperation"/>
</schema>

```

## 12.3. Adaptionenregeln

Die *Objektadaptionenregeln* bestehen aus vier Hauptregeln. Die erste Regel selektiert die Geodaten nach dem Bereich und wird generell folgendermaßen definiert:



1. Allgemein: **IF** domain = ... **THEN** use feature\_class\_list = ... (selection list of feature classes from object catalog) from geodata\_base as GeodataSet1

Diese Regel sieht anhand eines Beispiels in Oracle im Allgemeinen wie folgt aus:

```
Oracle SQL: CREATE VIEW GeodataSet1 AS SELECT CASE WHEN domain=... THEN SELECT * FROM geodata_base WHERE feature_class_list= ...
```

Diese Regel wird für alle Werte, die den Bereich annehmen können, in Oracle umgesetzt:

```
Beispiel Oracle SQL: CREATE VIEW GeodataSet1 AS SELECT CASE WHEN domain=mili FROM geodata_base WHERE feature_class_list=1
```

```
Beispiel Oracle SQL: CREATE VIEW GeodataSet1 AS SELECT CASE WHEN domain=topo FROM geodata_base WHERE feature_class_list=2
```

Die zweite Regel selektiert die Geodaten nach der Rolle. Hierbei sind auch mehrere Rollen eines Nutzers über eine Gewichtung zu berücksichtigen. Die Regel wird folgendermaßen definiert:

2. Allgemein: **IF** role= ... **THEN** use role\_list = ... (selection list of feature classes and feature types with priority dependent on role) from GeodataSet1 as GeodataSet=2

```
Oracle SQL: CREATE VIEW GeodataSet2 AS SELECT CASE WHEN role=... THEN SELECT * FROM GeodataSet1 WHERE role_list=...
```

```
Beispiel Oracle SQL: CREATE VIEW GeodataSet2 AS SELECT CASE WHEN role=viewer THEN SELECT * FROM GeodataSet1 WHERE role_list=1
```

Die dritte Regel selektiert die Geodaten nach dem System und wird folgendermaßen definiert:

3. Allgemein: **IF** system = ... **THEN** use system\_list = ... (selection list of feature classes and feature types with priority dependent on system) from GeodataSet2 as GeodataSet=3

```
Oracle SQL: CREATE VIEW GeodataSet3 AS SELECT CASE WHEN system=... THEN SELECT * FROM GeodataSet2 WHERE system_list=...
```

```
Beispiel Oracle SQL: CREATE VIEW GeodataSet3 AS SELECT CASE WHEN system=TabletPC THEN SELECT * FROM GeodataSet2 WHERE system_list=1
```

Die vierte Regel selektiert die Geodaten nach den dynamischen Element Situation (Position) und der Boundingbox der WMS-Anfrage und wird folgendermaßen definiert:

4. Allgemein: **SELECT** features **FROM** GeodataSet= 3 **WHERE** INSIDE bbox = ... or COVEREDBY position=...

```
Oracle SQL: SELECT * from GeodataSet= 3 WHERE SDO_RELATE ('masktype=INSIDE querytype=box' or 'masktype=COVEREDBY querytype=position')
```

```
Beispiel Oracle SQL: SELECT * from GeodataSet= 3 WHERE SDO_RELATE ('masktype=INSIDE querytype=9.4,9.5,51.2,51.3' or 'masktype=COVEREDBY querytype=9,5 51,3')
```

Die *Portrayaladaptionsregeln* bestehen aus drei Hauptregeln, die im Folgenden dargestellt werden. Die erste Regel selektiert das passende Portrayal Rule Set abhängig von den statischen Elementen Bereich

und System:

1. Allgemein: **IF** domain=... and device=... **THEN** use RuleSet=...

Oracle SQL: SELECT CASE WHEN domain=... and device=... THEN SELECT \* FROM RuleSet=...

Beispiel Oracle SQL: SELECT CASE WHEN domain=mili and device=pda THEN SELECT \* FROM RuleSet=2

Die zweite Regel selektiert das passende Portrayal Rule Sub Set abhängig von den dynamischen Elementen Situation und Physikalisch, wenn das System mobil statt stationär genutzt wird:

2. Allgemein: **IF** RuleSet=... and systemuse=mobile and time=... and weather=... and lightCondition=... **THEN** use RuleSubSet=... **ELSE** RuleSet=RuleSubSet

Oracle SQL: SELECT CASE WHEN RuleSet=... and systemuse=mobile and time=... and weather=... and lightCondition=... THEN SELECT \* FROM RuleSubSet=... **ELSE** RuleSet=RuleSubSet

Beispiel Oracle SQL: SELECT CASE WHEN RuleSet=2 and systemuse=mobile and time=12:00:00h and weather=sun and lightCondition=day THEN SELECT \* FROM RuleSubSet=1 **ELSE** RuleSet1=RuleSubSet1

Die dritte Regel ist eine Gruppe von Regeln, die je nach Rolle die einzelnen Portrayalregeln und die Symbole und deren Komponenten anpasst:

3. Allgemein: **IF** RuleSubSet=... and role=... **THEN** use change\_list=... (list with changes for priority, symbology and highlighting)

Oracle SQL: SELECT CASE WHEN RuleSubSet=... and role=... THEN SELECT \* FROM RuleSubSet=... WHERE change\_list=...

Beispiel Oracle SQL: SELECT CASE WHEN RuleSubSet=1 and role=mili THEN SELECT \* FROM RuleSubSet=1 WHERE change\_list=1

Die graphischen Adaptionsregeln bestehen aus zwei Hauptregeln, die im Anschluss aufgezeigt werden. Die Anpassung erfolgt auf dem Client/Gerät und es werden aus diesem Grund nur allgemeine Regeln dargestellt.

Die erste Regel passt die Ansicht abhängig von Situation (Zeit) an:

1. Beispiel: **IF** time=12:00 and sun\_list (timelist with sunrise and sunset) **THEN** lighten display

Die zweite Regel passt die Ansicht abhängig von Physikalisch an:

2. Beispiel: **IF** weather=rain and light\_condition=day **THEN** lighten display

Die *Funktionale Adaptionsregel* ist eine Regel, die im Folgenden dargestellt wird. Sie selektiert die passenden Werkzeuge abhängig von den statischen Elementen Bereich und Rolle entsprechend der definier-

ten Werkzeuge:

1. Allgemein: **IF** domain=... and role=... **THEN** use function\_list = ... (selection list of functions dependent on domain and role)

Oracle SQL: SELECT CASE WHEN domain=... and role=... THEN SELECT function\_list=...

Beispiel Oracle SQL: SELECT CASE WHEN domain=mili and role=analysis THEN SELECT function\_list=1

## 12.4. XSQL Action Element Handler und XMLType Funktionen

In Tabelle 19 werden die XSQL Action Element Handler und ihre Beschreibung dargestellt.

**Tabelle 19: XSQL Action Element Handler nach OracleXDK (2007)**

XSQL	Beschreibung
<xsql:action>	XSQLExtensionActionHandler Invoke a user-defined action handler, implemented in Java, for executing custom logic and including custom XML data in your XSQL page
<xsql:delete-request>	XSQLDeleteRequestHandler Delete an existing row in the database based on the posted XML document supplied in the request.
<xsql:dml>	XSQLDMLHandler Execute a SQL DML statement or a PL/SQL anonymous block.
<xsql:if-param>	XSQLIfParamHandler Conditionally include XML content or other XSQL actions.
<xsql:include-owa>	XSQLIncludeOWAHandler Include the results of a stored procedure that uses the Oracle Web Agent (OWA) packages in the database to generate XML.
<xsql:include-param>	XSQLGetParameterHandler Include a parameter and its value as an element in the XSQL page.
<xsql:include-posted-xml>	XSQLIncludePostedXMLHandler Include the XML document that has been posted in the request into the XSQL page.
<xsql:include-request-params>	XSQLIncludeRequestHandler Include all request parameters as XML elements in the XSQL page.
<xsql:include-xml>	XSQLIncludeXMLHandler Include arbitrary XML resources at any point in your page by relative or absolute URL.
<xsql:include-xsql>	XSQLIncludeXSQLHandler Include the results of one XSQL page at any point inside another.
<xsql:insert-param>	XSQLInsertParameterHandler Insert the XML document contained in the value of a single parameter.
<xsql:insert-request>	XSQLInsertRequestHandler Insert the XML document or HTML form posted in the request into a table or view.
<xsql:query>	XSQLQueryHandler Execute an arbitrary SQL statement and include its result in canonical XML format.
<xsql:ref-cursor-function>	XSQLRefCursorFunctionHandler Include the canonical XML representation of the result set of a cursor returned by a PL/SQL stored function.
<xsql:set-cookie>	XSQLSetCookieHandler Set an HTTP Cookie.
<xsql:set-page-param>	XSQLSetPageParamHandler Set an HTTP-Session level parameter. Set a page-level (local) parameter that can be referred to in subsequent SQL state-

	ments in the page.
<xsql:set-session-param>	XSQLSetSessionParamHandler Set an HTTP-Session level parameter.
<xsql:set-stylesheet-param>	XSQLStylesheetParameterHandler Set the value of a top-level XSLT stylesheet parameter.
<xsql:update-request>	XSQLUpdateRequestHandler Update an existing row in the database based on the posted XML document supplied in the request.

In Tabelle 20 werden die XMLType-Funktionen und ihre Beschreibung dargestellt.

**Tabelle 20: XMLType-Funktionen**

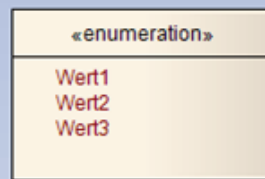
XML Funktion	Beschreibung
ExtractValue	Liefert anhand eines Xpath-Ausdrucks den Wert eines Knotens.
Extract	Extrahiert einen oder mehrere Äste eines XMLType-Objekts.
ExistsNode	Prüft, ob in einem XMLType-Objekt ein entsprechender Knoten vorliegt.
GetRootElement	Gibt das Root-Element der XMLType-Instanz zurück.
GetSchemaURL	Gibt die URL der registrierten XML-Schema-Definition zurück. Wurde kein XML-Schema registriert, so wird NULL zurück gegeben.
Transform	Transformiert die übergebene XMLType-Instanz unter Einsatz einer XSLT-Datei als XMLType-Instanz in HTML, Text etc.
UpdateXML	Ändert Werte in einem XMLType-Objekt.
XMLSequence	Ermöglicht die Verarbeitung mehrerer Knoten eines XMLType-Objekts.

## 12.5. UML Notationen und andere Darstellungen

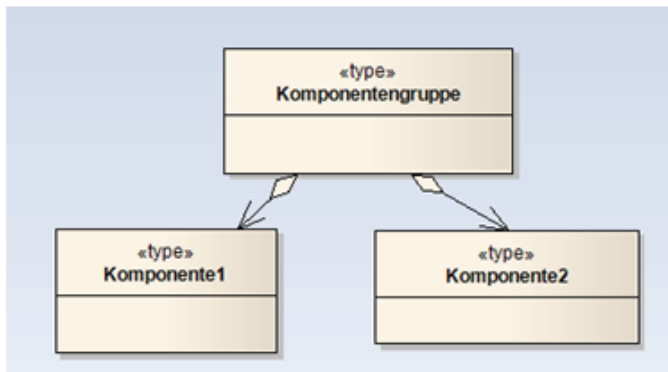
### Klasse



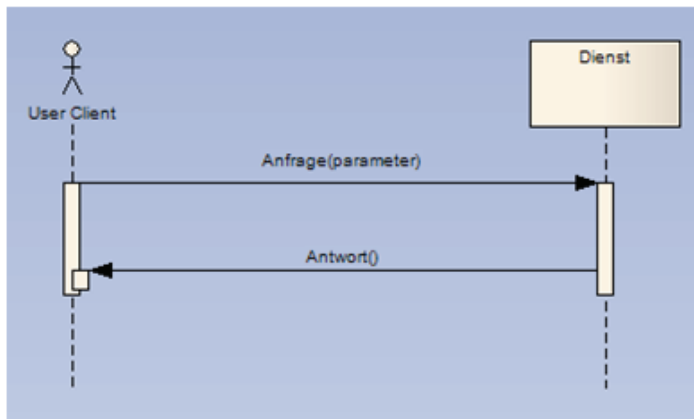
### Enumeration



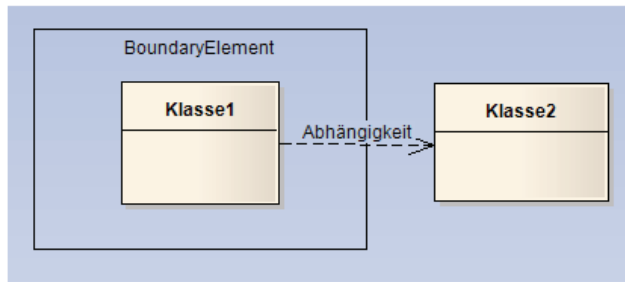
### Shared Aggregation (navigierbar)



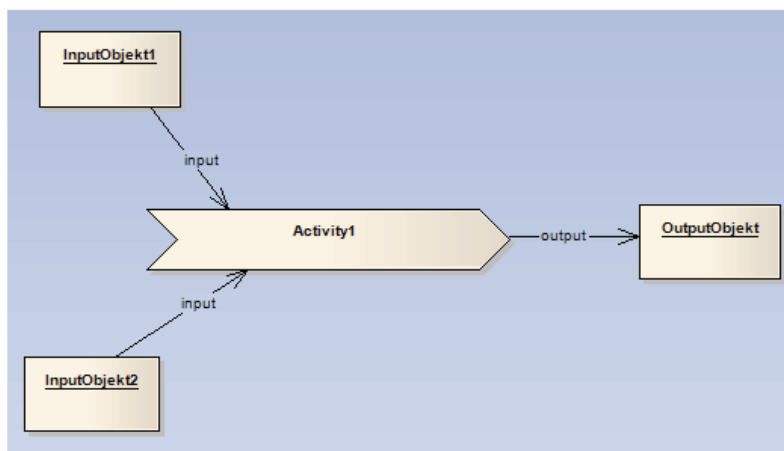
### Sequenzdiagramm



## Boundary Element



## Businessworkflow



## Dank

Mein besonderer Dank gilt Herrn Univ.-Prof. Dr.-Ing. Wolfgang Reinhardt für die Unterstützung während der Erstellung der Dissertation, für die wertvollen Denkanstöße und für die jederzeitige Bereitschaft zu fachlichen Diskussionen. Ferner danke ich Herrn Univ.-Prof. Dr.-Ing. Wolfgang Reinhardt und Uni.-Prof. i. R. Dr.-Ing. Wilhelm Caspary für die sehr lehrreiche Zeit an der AGIS und die Möglichkeit zur Promotion. Mein Dank gilt außerdem Herrn Univ.-Prof. Dr.-Ing. Alexander Zipf für die fachlichen Diskussionen und nützlichen Hinweise während der Erstellung der Arbeit. Dem Vorsitzenden des Promotionsverfahrens, Herrn Univ.-Prof. Dr.-Ing. Kurt Brunner, danke ich ebenfalls für die Bereitschaft den Vorsitz der Promotion zu übernehmen.

Mein Dank geht auch an das Amt für Geoinformationswesen der Bundeswehr (AGeoBw). Aus dem geförderten Projekt „Präsentation von Vektordaten des GeoInfoDienstes der Bundeswehr“ konnten Ergebnisse in die Dissertation einfließen.

Für das angenehme und motivierende Arbeitsumfeld und die vielen konstruktiven Gespräche möchte ich mich herzlichst bei meinen ehemaligen Kollegen insbesondere bei Dr.-Ing. Admire Kandawasvika, Stefan Strobel, Thorsten Bockmühl, Dr.-Ing. Stephan Mäs, Eva Nuhn und Dr.-Ing. Simone Stürmer bedanken.

Zu guter Letzt möchte ich mich bei meiner Familie und bei allen Freunden für die Unterstützung und Ermutigungen bedanken, die zur Fertigstellung meiner Dissertation beitrugen. Ich danke meinem Papa für das Korrekturlesen meiner Arbeit. Meinen Eltern möchte ich für die Unterstützung während meines Studiums danken.

## Lebenslauf

### **Iris Wiebrock**

Geboren am 23. Mai 1978 in Hannover

1990 bis 1994	Gymnasialstufe in Bangkok, Thailand, Ruamrudee International School, Swiss Section
1994 bis 1997	Gymnasiale Oberstufe in Hannover, Sophienschule , Abschluss Abitur
1997 bis 2002	Studium an der Universität Hannover, Abschluss Dipl.-Ing. Vermessungswesen
2003 bis 03/2010	Wissenschaftliche Mitarbeiterin der Arbeitsgemeinschaft GIS (AGIS) an der Universität der Bundeswehr München in Neubiberg im Bereich „Präsentation von Vektordaten des GeoInfoDienstes der Bundeswehr“, gefördert durch das Amt für Geoinformationswesen der Bundeswehr (AGeoBw)
Seit 05/2010	Selbstständig, WIEBROCK.geosolutions, Beratung im Bereich Geodatenvisualisierung und Geodienste