Einbindung dreidimensionaler Modelldaten in die Nexus-Plattform

Prüfer:	Prof. Dr. B. Mitschang
Betreuerin:	Dipl.-Inform. Daniela Nicklas
Beginn am:	15.04.2002
Beendet am:	15.10.2002
CR-Nummer:	C 2.4, H 2.3, H 2.8

Diplomarbeit-Nr. 2012

Einbindung dreidimensionaler Modelldaten in die Nexus-Plattform

Dietrich Müller

Inhaltsverzeichnis

1. Einleitung 7

1.1 Motivation 7
1.2 Nexus 8
1.3 Aufgabenbeschreibung 8
1.4 Aufbau der Arbeit 8

2. Die Nexus-Plattform 9

2.1 Architektur 9
2.2 Nexus-Dienste und -Anwendungen 10
2.3 Mehrwertdienste 11
2.4 Föderation 11
2.5 Augmented World Model (AWM) 12

3. Anforderungsanalyse 21

3.1 Untersuchung der Use-Cases 21
3.2 Anforderungen an bestehende Systeme 27
3.3 Analyse der Sprachen AWQL und AWML 28
3.4 Untersuchung der Anforderungen 29
3.5 Anforderungen an Anfragen 30
3.6 Anforderungen an die Position 30
3.7 Anforderungen an die Blickrichtung 30
3.8 Schematisches Architekturbild für die Erstellung von 3D Bildern 31
3.9 Anforderungen an das Ergebnis einer Anfrage 31
3.10 Analyse der Anforderungen an die Eingabe 32
3.11 Anforderungen an Anfrage- und Ergebnisparameter 32
3.12 Anforderungen an den Benutzer 33
3.13 Anforderungen an Funktionsumfang und Datenumfang 33
3.14 Anforderungen an die Schnittstelle 33
3.15 Anforderungen an die Kommunikation 34
3.16 Kommunikation zwischen MWD und 3D Bilddienst 36
3.17 Kommunikation zwischen MWD und Client 37
3.18 Anforderungen bezüglich der Dienstbereitstellung 37

4. Lösungsansätze 39

4.1 Datenmodellerweiterung 39
4.2 Anfrageerweiterung 39
4.3 Objektgebundene Erweiterung 40
4.4 Mehrwertdienst 40
4.5 Wrapper 44
4.6 **Zusammenfassung 45**

5. Die Bildgenerierung als Webservice 49

5.1 Definition des Webservice 49
5.2 Funktionsweise der Bildgenerierung 49

5.3 Anforderungen an den Webservice Bildgenerierung 51
5.4 Aufruf des Webservice Bildgenerierung 53
5.5 Ablauf einer Anfrage an die Bildgenerierung 54
5.6 Verarbeitung der Anfrage 56

6. Entwurf 57

6.1 Entwurf der Datenstruktur 57
6.2 Einlesen der Daten 62
6.3 Applikation 63
6.4 Realisierung der Koordinatentransformation 64
6.5 Konvertierung von WGS84 nach Gauß-Krüger 67
6.6 Realisierungsmöglichkeiten der Anwendung Bildgenerierung 69
6.7 Entwurf der Kommunikation 69

7. Zusammenfassung und Ausblick 75

7.1 Zusammenfassung 75
7.2 Ausblick 75

Kurzfassung

Das Nexus-Projekt hat die Intention, eine offene und globale Plattform für ortsbezogene Anwendungen zu entwicklen. Ein wichtiger Aspekt ist die Erweiterbarkeit des Nexus-Systems. Dies schließt Anwendungen, Dienste und Daten ein. Standarddienste und Mehrwertdienste tragen dem Gedanken der Integrierbarkeit heterogener Daten Rechnung, um externe Dienstleistungen in die Nexus-Infrastruktur einzubinden.

In der vorliegenden Diplomarbeit sollen dreidimensionale Modelldaten in die Nexus-Plattform eingebunden werden. Um das Ziel dieser Arbeit zu realisieren, werden zunächst die die Daten, die bestehenden Systeme und die Nexus-Architektur analysiert. Basierend darauf werden die Anforderungen herausgearbeitet, die den Leistungsumfang der Grundfunktionen feststellen. Anschließend werden die Realisierungskonzepte Wrapper und Mehrwertdienst den Anforderungen gegenübergestellt. Die favorisierte Möglichkeit soll prototypisch umgesetzt werden.

1 Einleitung

Die Einleitung dient dazu, den Leser mit dem Thema vertraut zu machen und dabei einen Ausblick auf den Inhalt geben. Der Rahmen wird duch die Motivation, das Forschungsprojekt Nexus, die Aufgabenbeschreibung gebildet. Der Ausblick über den Aufbau der Arbeit beendet die Einleitung.

1.1 Motivation

Die Entwicklung der letzten Jahre hat den Gedanken der Mobilität sehr gefördert. Mobilität beeinflusst mehr und mehr alle möglichen Anwendungsbereiche und soll an verschiedenen Orten für mobile Anwender einsetzbar sein. Ortsbasierte Anwendungen sind ein Beispiel, bei denen die Informationen vom jeweiligen Ort des Anwenders abhängt. Der momentane Aufenthaltsort ist ausschlaggebend für die Information, die der Anwender erhält. Informationen werden neben der Mobilität auch der Aktualität gerecht und können von verschiedenen Datenanbietern zur Nutzung eingebracht werden.

Vorstellbar sind ortsbasierte Anwendungen, wie z.B. ein Touristguide. Dieser könnte dazu dienen Informationen über sehenswerte Gebäude oder Museen und deren Ausstellungen in der Nähe, des Benutzers diesem auf dessen PDA mitzuteilen. Dazu muß aber auf die Daten extern zugegriffen werden, da die zu übertragenden Informationen noch zu umfangreich für die gegenwärtigen mobilen Geräteeinheiten sind.

Ortbasierte Anwendungen haben meistens ein großes Datenvolumen unterschiedlicher Art und Struktur zu bewältigen. Dies wird durch immer umfangreichere Anwendungen beeinflusst. Unterschiedlichste Endgeräte sollen den Anwender durch ein immer größer werdendes Leistungsspektrum unterstützen. Die Leistungssteigerung der Anwendungen bedingt neben gerätespezifischen Neuerungen die Fähigkeit, komplexere Daten verarbeiten zu können.

Komplexe Daten werden benötigt, um Strukturen und Objekte der realen Welt in einem Model, wie im Augmented World Model der Nexus-Architektur besser erfassen und darstellen zu können. Dazu reicht es nicht mehr aus bei der Repräsentation der Objekte sich nur auf die zweidimensionale Darstellung zu beschränken. Die Konsequenz ist ein höherer Aufwand bei Verarbeitung und Speicherung. Eine Erweiterung für die Speicherung von 3D-Daten in den gängigen Datenbanksystemen existiert noch nicht, dagegen können aber Geoinformationssysteme diese Daten speichern. Es stellt sich die Frage, wie man 3D-Daten in der Nexus-Architektur einbinden und nutzbar machen kann. Um eine Lösungsmöglichkeit für diese Problematik zu suchen, wurde nachfolgende Aufgabe (siehe 1.3) innerhalb des Nexus-Forschungsprojektes gestellt.

1.2 Nexus

Das Ziel des Forschungsprojektes Nexus besteht darin, Konzepte und Methoden zur Realisierung einer Plattform für ortsbezogene Anwendungen zu entwickeln (s. Einführung Kapitel 2). Dazu wird eine globale, offene Plattform entwickelt, die als Infrastruktur für solche Anwendungen dient. Grundlage hierfür bildet ein verteiltes, dynamisch gegenüber Veränderungen angepaßtes Modell der realen Welt. Erweitert wird es mit virtuellen Objekten. Die virtuellen Objekte werden mit geographischen Positionen oder mit realen Objekten verbunden. Sie enthalten Links zu bestehenden Informationsräumen, wie z.B. dem WWW. Der Zugriff auf diese Plattform wird durch einen hochverfügbaren Kommunikationsdienst realisiert, der verschiedene mobile Kommunikationssysteme integriert. Dies schließt einen Quality of Service und geographische Adressierungsmöglichkeiten ein. Un-umgänglich ist die Entwicklung neuer Föderationstechniken. Sie verbindet bestehende Informationsquellen mit den innerhalb des Forschungsprojekts entwickelten Datenbanken.

1.3 Aufgabenbeschreibung

In dieser Arbeit soll untersucht werden, ob die Einbindung dreidimensionaler Daten durch die Verwendung bestehender Systeme für die Speicherung von 3D Daten als Mehrwertdienst erreicht werden kann; oder ob durch die Entwicklung eines Wrappers bestehende Systeme als Spatial Model Server angesprochen werden können. Nach Entscheidung für eine der beiden Möglichkeiten soll diese dann prototypisch umgesetzt werden. Dabei sollen 3D Daten des Instituts für Photogrammetrie (ifp) zu Demonstrationszwecken in das realisierte System eingebunden werden.

1.4 Aufbau der Arbeit

Kalitel 2 enthält eine Einführung in die Nexus-Plattform. An dieses Kapitel schließt Kapitel 3 an, das in einer Analyse die bestenden Probleme aufzeigt. Im folgenden Kapitel 4 werden die Lösungsansätze diskutiert. Dies bildet die Grundlage für das Kapitel 5. In diesem Kapitel wird der favorisierte Ansatz weiter aufbereitet. Kapitel 6 enthält den Entwurf. Das folgende Kapitel 7 bildet den Abschluß der Arbeit und fasst diese zusammen.

2 Die Nexus-Plattform

Das vorhergehende Kapitel bestand aus einer Einleitung in die Thematik. Kapitel 2 dient dazu, die Infrastruktur von Nexus (An Open Global Infrastrukture for Spatial-Aware Applications) zu beschreiben. Ausgangspunkt für weitergehende Erklärungen bildet die Architektur mit ihren Schichten. Die Komponenten Föderation, Augmented World Model, Nexus-Dienste, Nexus-Anwendungen und Mehrwertdienste werden in den folgenden Unterkapiteln näher erläutert.

In die Nexus-Architektur sollen dreidimensionale Modelldaten eingebunden werden. Für die Generierung dieser Daten ist ein externer Dienst vorhanden, der nicht Teil der Architektur ist, sondern auf einem entfernten Rechner läuft. Um den Dienst in Nexus nutzen zu können, muß dieser in die Nexus-Architektur eingebunden werden. Für das Verständnis der Zusammenhänge wird in diesem Kapitel die Nexus-Architektur erläutert.

2.1 Architektur

Die Nexus-Architektur weist einige Charakteristiken auf, die nun erläutert werden.

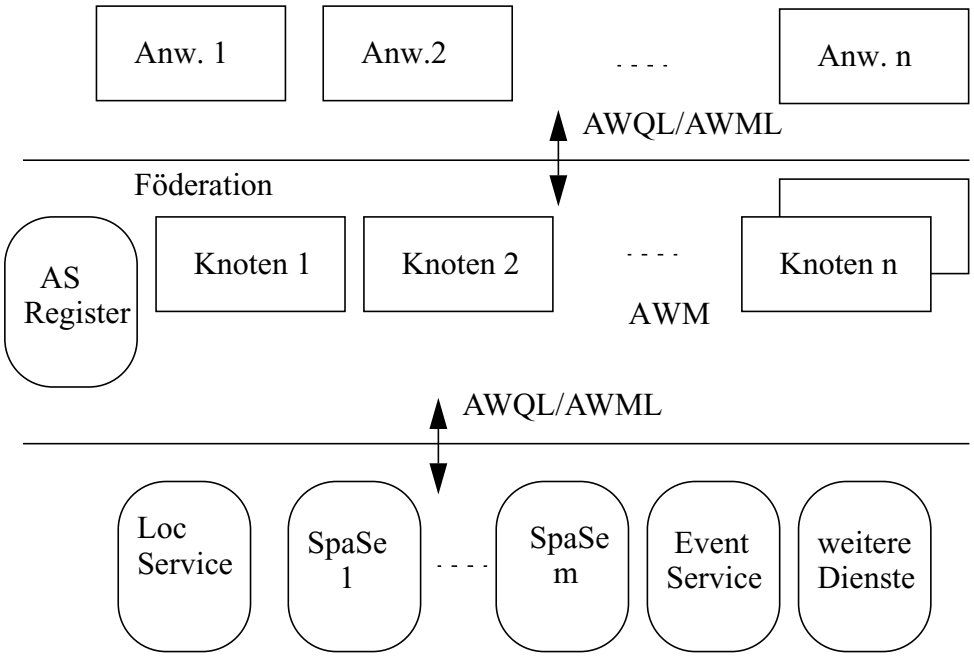

Abb.2.0: Nexus-Architektur (aus [Nicklas et al 2001])

Die Abbildung 2.0 zeigt die Nexus-Architektur. Sie besteht aus drei Schichten: Den Anwendungen, den Knoten und den Diensten.

Das Augmented World Model (AWM) ist die gemeinsame Datensicht der Anwendungen und der Plattform. Sie stellen Anfragen an Knoten der Föderation. Die Anwendungen greifen auf Dienste verschiedener Datenanbieter, deren Schnittstellen Nexus zur Verfügung stellt, zu. Dies wird konzeptionell über Knoten realisert. Aufgabe der Knoten ist die Weiterleitung der Anfragen an die Dienste. Das Area Service Register (ASR) gibt Auskunft über den zuständigen Server. Spatial Model Server (SpaSe) verwalten Informationen zu statischen Objekten und ermöglichen räumliche Anfragen über die Objekte. Der Location Service administriert Informationen mobiler Objekte. Zu den weiteren Diensten gehört der Event Service, der eine ereignisbasierte Kommunikation ermöglicht.

2.2 Nexus-Dienste und -Anwendungen

In diesem Unterkapitel werden die Nexus-Dienste und danach die Nexus-Anwendungen betrachtet.

2.2.1 Nexus-Dienste

Zu den Diensten von Nexus gehören der Location Service, der Event Service und die Spatial Model Server (s. Abb Nexus-Architektur). Die SpaSes ermöglichen räumliche Abfragen über Objekte der Augmented World. Ferner werden die Augmented Areas, die von den SpaSe administriert werden beim AS Register angemeldet, der die Informationen über die Augmented Areas speichert.

Der Location Service speichert Informationen der mobilen Objekte. Dazu gehören auch Objekte, die sich innerhalb anderer statischer (Gebäude) oder dynamischer (Schiffsrumpf) Objekte selbst (Personen) dynamisch bewegen.

Der Event Service, bestehend aus Observation und Notification Service, schafft die Voraussetzung für eine Kommunikation durch Events. Events bezeichnen das Auftreten von Ereignissen. Die Benachrichtigung über das Auftreten übernimmt der Notification Service, der Observation Ser-vice entscheidet über die Weiterleitung.

2.2.2 Nexus-Anwendungen

Nexus-Anwendungen sind ortsbasiert, deshalb hängt ihre Funktionalität vom gegenwärtigen Ort des Anwenders ab. Der Standpunkt des Anwenders wird durch dessen Position (hier: auf der Karte) bestimmt. Eine Nexus-Anwendung besteht aus folgenden Komponenten: der Benutzungsschnittstelle, der Anwendungslogik

und einer Komponente zur Verarbeitung von AWML und AWQL. In AWQL wird die Anfrage formuliert, in AWML das Ergebnis eingelesen. Die Anwendung greift mit der Nexus-Plattform auf Kommunikation die Nexus-Schnittstelle.

Beispiel für eine Nexus-Anwendung kann ein Touristguide sein für die Darstellung ortsbasierter Informationen. Der Guide kann den Weg zu sehenswerten Gebäuden, Museen beschreiben und bei Anforderung weiterführende Informationen ausgeben.

2.3 Mehrwertdienste

Mehrwertdienste dienen der Erweiterung der Funktionaltät und befinden sich auf einem Nexus-Knoten. Angesprochen werden sie über eine proprietäre Schnittstelle von den Anwendungen. Mehrwertdienste zählen nicht zu den Nexus-Diensten, da sie föderierte Daten bearbeiten oder auf Quellen außerhalb des Nexus-Systems zugreifen. Sie können eine beliebige Schnittstelle zu den Anwendungen anbieten. Mehrwertdienste stellen ihre Dienste den Nexus-Anwendungen zur Verfügung, im Vergleich zu den Nexus-Diensten (vgl oben), die von der Föderation verwendet werden.

2.4 Föderation

Die Föderation bietet den Anwendungen eine einheitliche Sicht auf die Daten und läuft auf den Nexus-Knoten. Zwei Schnittstellen, die Anwendungsschnittstelle und die Dienstschnittstelle, besitzt die Föderation. Über die Anwendungsschnittstelle werden Anfragen und Ergebnisse verarbeitet, über die Dienstschnittstelle werden die Nexus-Dienste mit der Föderation verbunden. Anfragen werden von der Föderation (s. nachfolgende Abbildung 2.1) entgegengenomen (AWQL) und zunächst die Zielregion der Anfrage ermittelt. Die gesuchten Objektklassen und die Zielregion werden an das Area Service Register (ASR) geschickt (AADL), das eine Liste der Augmented Areas (AAList) an die Föderationskomponente zurücksendet. Anschließend wird die Anfrage der Anwendungen zu den entsprechenden Spatial Model Server (SpaSe) geleitet (AWQL). Die Föderationskomponente sammelt die Antworten der Server (AWML) ein und faßt sie in einem Ergebnis zusammen. Das Gesamtergebnis wird anschließend an die Anwendungen zurückgesendet (AWML).

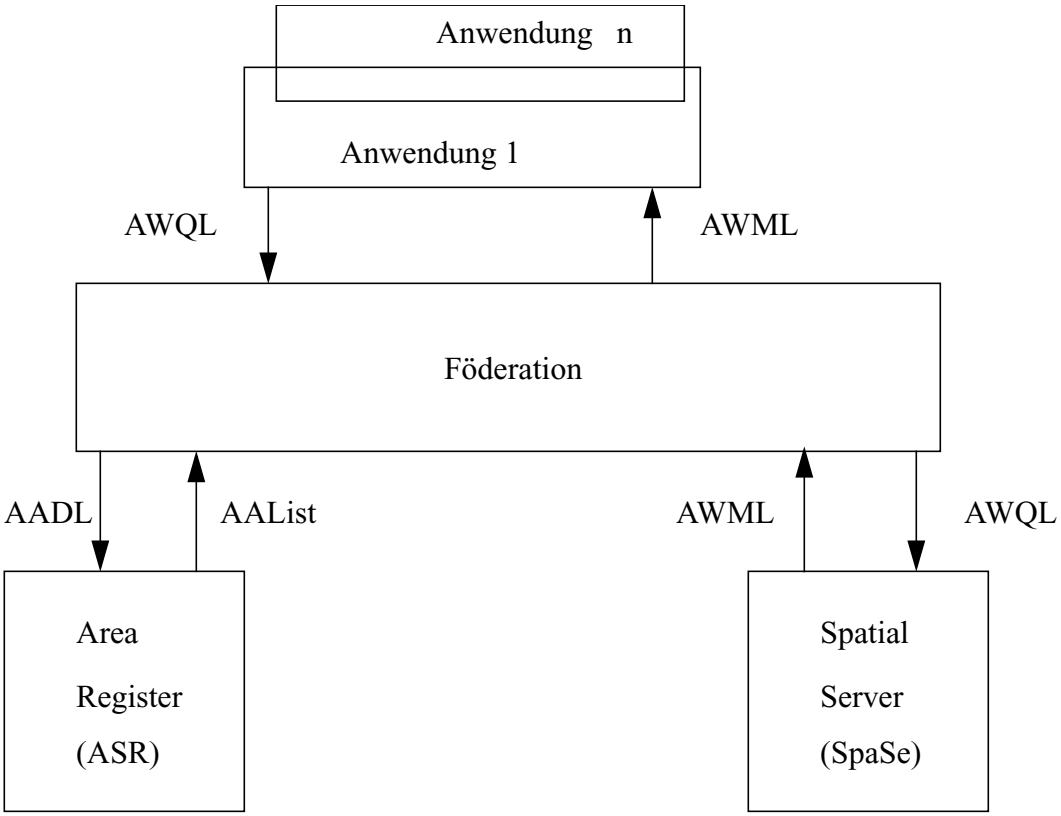

Abb.2.1: Ablauf einer Anfrage

2.5 Augmented World Model (AWM)

Das Augmented World Model, ein objektorientiertes Informationsmodell, beschreibt in einem Modell die reale Welt. Diese wird durch virtuelle Objekte angereichert (augmented) [Nicklas et al 2001]. Virtuelle Objekte können z.B. virtuelle Litfaßsäulen sein. Die folgende Abbildung 2.2 zeigt den Aufbau des Augmented World Models.

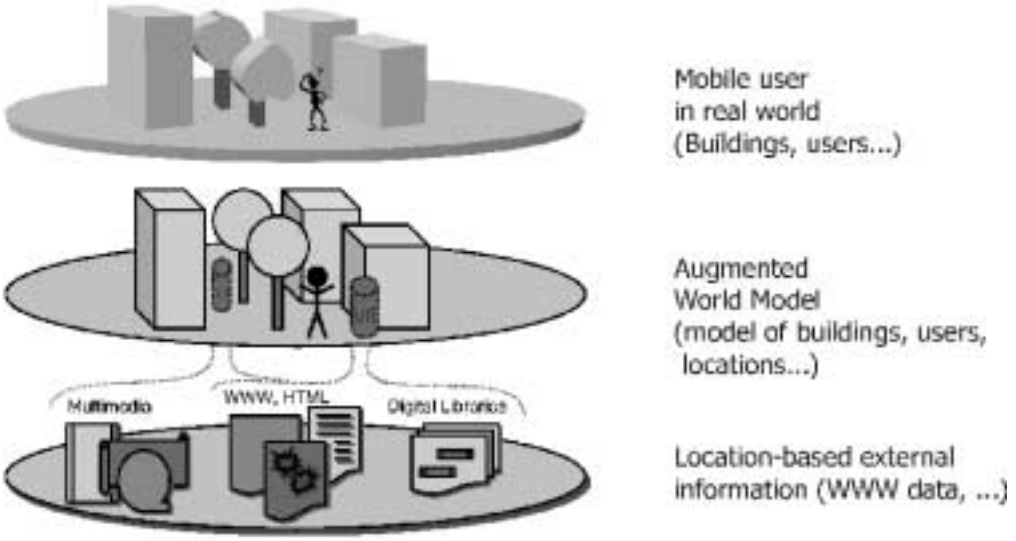

Abb.2.2: Aufbau des AWM [Nexus]

Kennzeichnend für das Modell sind die folgenden drei Ebenen: "top level", "middle level" und "bottom level". Die Ebenen charakterisieren sowohl die Verknüpfung der realen Welt zur Augmented World als auch die Verbindung zu den virtuellen Objekten [Nexus Vision]. Die Virtualität nimmt von oben nach unten je Ebene zu. Die oberste Ebene bildet einen möglichen Ausschnitt der realen Welt. Enthalten sind reale statische Objekte, wie Bäume, Häuser und reale mobile Objekte in Form von Benutzern. Im Vergleich dazu zeigt die mittlere Ebene die Augmented World mit realen Objekten, die physisch existieren. Erweitert ist die Ebene um virtuelle Objekte, die in der realen Welt nicht vorkommen. Auf derselben Ebene sind in der Abbildung virtuelle Litfaßsäulen (Virtual Information Tower (VIT) [ViLis 2000] zu sehen. Sie dienen als Verbindung zu externen Datenquellen. Die unterste Ebene enthält externe Datenquellen. Dazu gehören Multimedia, das WWW und digitale Bibliotheken.

Das Modell entsteht durch die Zusammenführung aller Augmented Areas [Hohl et al 1999]. Es stellt die Grundlage dar, die Nexus-Plattform flexibel und erweiterbar zu gestalten. Ferner können externe Datenquellen, wie z.B. dreidimensionale Daten auf entfernten Servern integriert werden. Der Zugriff kann durch einen Aufruf über das WWW realisiert werden. Zum Datenaustausch werden zwei Sprachen, AWQL und AWML, verwendet. Die Augmented World Query Language (AWQL) spezifiziert Anfragen, die Augmented World Modelling Language (AWML) dient zur Darstellung der Objekte und zur Repräsentation des Ergebnisses. Beide verwenden GML (Geographic Markup Language) [GML 2.0], für die Darstellung zweidimensionaler Geodaten. Ferner gibt es noch die Change

Report Language (CRL) zur Repräsentation des Ergebnisses von Einfüge-, Lösch-, und Änderungsoperationen, die Augmented Area Description Language (AADL) zur Beschreibung der Augmented Areas und die Map Predicate Language (MapPL) für die Generierung von Karten.

2.5.1 Augmented World Query Language (AWQL)

AWQL ist eine Query-Sprache, eine Abfragesprache, die SQL ähnelt. Die Mächtigkeit ist jedoch im Vergleich zu SQL bedeutend kleiner. Das Ergebnis einer Anfrage in AWQL ist eine Teilmenge der Augmented World. Anfragen werden von Anwendungen an die globale Föderation gestellt. Die Spezifikation von AWQL steht in [Schwarz 2001], sie enthält die fogenden Elemente:

"aas, scope, restriction, closest, filter, update, delete":

aas

Das "aas" Element ist eine Liste und enthält den Nexus Augmented Area Locator (NAAL). Der Locator bestimmt die betreffende Area und den entsprechenden Dienstanbieter. Die Anwendung kann den NAAL direkt bestimmen und kann den Suchraum begrenzen.

scope

Das Element "scope" besteht aus einer Liste von Extended Class Schemas. Sie enthält die Schemas, die die jeweilige Nexus-Anwendung versteht. Mit "scope" können Aliases für Extended Class Schemas definiert werden.

restriction

Mit "restriction" werden die Bedingungen der Objekte definiert, damit sie in die Ergebnismenge aufgenommen werden können. Dazu gehören "and", "or", "not", "in", "equal", "inside" und "overlaps". Sollen Anfragen mit mehreren Bedingungen verknüft werden, so geschieht dies mit "and", "or" und "not". Der Vergleich wird mit "equal" wird benötigt, wenn ein Attribut einen bestimmten Wert haben soll. Das Element "in" wird verwendet, um einem Attribut einen Wert aus einer bestimmten Menge zuzuordnen. "inside" und "overlaps" sind zweidimensionale, geometrische Prädikate. Sie dienen der Objektwahl innerhalb einer Region oder lassen auf diese Weise den Schnitt zwischen Regionen zu.

filter

Objekte müssen nicht immer alle möglichen Attribute besitzen. Sollen Objekte nur bestimmte Attribute aufweisen, wird das Element "filter" verwendet, um die Eigenschaften zu beschränken. Es kann zwischen "includes", excludes", "includeallother", "excludeallother" gewählt werden. Mit "include", bzw. "excludes" werden die Attribute ausgesucht bzw. verworfen. Wird "excludes" mit "includeallother" verbunden so erhält man alle Attribute, die nicht mit "excludes" ausgeschlossen werden. Es kann auch "includes" mit "excludeallother" kombiniert werden. Das "ex-cludeallother" Element bietet die Möglichkeit, alle Attribute auzgrenzen, die nicht in "includes" gesucht werden.

closest

Soll die Anzahl der Objekte begrenzt werden, geschieht das in der Anfrage mit closest. Das "num" Attribut nimmt die Anzahl der Objekte auf (z.B. num = 5 für die 5 nächstgelegensten Objekte in nachfolgender Beispielanfrage).

generalization

Diese Element wird verwendet, die geometrischen Attribute zu generalisieren. Die geometrischen Eigenschaften sind nur auf einer zweidimensionalen Ebene definiert. Das Ergebnis bleibt dementsprechend zweidimensional.

aggregation

Kleinere Objekte können zu größeren Objekten verbunden werden. Die geometrische Dimension wird davon nicht betroffen, d.h. mehrere zweidimensionale Objekte zu einem komplexeren dreidimensionalen Objekt verschmelzen zu lassen, führt im Ergebnis zu keinem dreidimensionalen Objekt.

update

Die Änderung von Werten bestimmter Attribute wird durch ein "update" Element realisiert. Die neuen Werte ersetzen die alten Werte.

Beispielanfrage zu AWQL

Die Anfrage soll alle Restaurants im Ergebnis zurückliefern, die in einem bestimmten Gebiet liegen. Es sind nur die fünf nächsten Objekte mit Namen und Position gesucht.

```
<awql>
  <restriction>
    <and>
```

```
      <!-- Die Objekte müssen Restaurant sein -->
      <equal>
          <attr name = "type"/>
          <nexusdata>restaurant</nexusdata>
      <equal>

      <!-- Die Objekte müssen sich innerhalb des Polygons befinden -->
      <inside>
        <gml>
            <polygon>
               <outerBoundaryIs>
                  <Linear Ring>
                      <coordinates> 40.0 40.0 40.0, 35.0 35.0, 35.0 35.0, 40.0 40.0,
                                    40.0
                      </coordinates>
                  </Linear Ring>
               </outerBoundaryIs>
            </polygon>
          </gml>
      </inside>
    </and>
  <restriction>
  <!-- Nur die 5 nächstern Objekte bzgl der angegebenen Position
  sollen zurückgeliefert werden-->
  <closest num = "5" acc = "1">
      <nexusdata>
          <gml>  ... GML representation ofan object describing my position ...
          </gml>
      </nexusdata>
  </closest>
  <!-- Alle Attribute bis auf Name und Position sollen herausgefiltert werden -->
  <filter>
     <includes>
        <attr name = "name"/>
        <attr name = "position"/>
     </includes>
     <excludeallother/>
  </filter>
</awql>
```

2.5.2 Augmented World Modeling Language (AWML)

Die Darstellung der Informationen wird in der Augmented World Modeling Language (AWML) verfasst. AWML ist ein XML-Dialekt und beschreibt die Augmented World. Ein AWML-Dokument ist also ein XML-Dokument. Der Inhalt

besteht aus den Objekten der Augmented World. Die Spezifikation ist in [Schwarz 2001]. AWML wird benötigt, um die Ergebnisse der Anfragen an das AWM zu beschreiben und um Daten zwischen den Diensten austauschen zu können. Damit ist es auch möglich, Elemente im Spatial Model Server einzufügen. Das folgende Beispiel ist ein mögliches Ergebnis der Anfrage (s.1.6.1).

```
<awml>
    <nexusobject type = "restaurant" nol = "nexus//...">
    <name>Italyrestaurant</name>
    <position>
        <gml>
            <Point>20,0 20,0</Point>
        </gml>
   </position>
   </nexusobject>
</awml>
```

2.5.3 Augmented Area Description Language (AADL)

Die Metainformationen des Area Service Registers werden mit Hilfe der Augmented Area Description Language (AADL) bearbeitet. Dazu bietet AADL Möglichkeiten, um Metainformationen einzufügen, zu löschen, abzufragen und zu aktualisieren.

Folgende Elemente gehören zu AADL:

"naal", "area", "awschema", "resultspec":

naal

Die Augmented Area muß als Ziel gewählt werden, um z.B neue Informationen einfügen zu können. Dies gilt auch für die anderen Operationen. Das "naal" Element dient als Identifikator.

area

Das "area" Element definiert die Ausdehnung der Region, die bearbeitet werden soll.

awschema

Eine Augmented Area enthält viele Objekte. Die Klassen der Objekte dieser Area wird mit dem "awschema" Element spezifiziert.

resultspec

In der Antwort wird zuerst eine Auswahl der Elemente "naal", "area", "awschema" im Element "resultspec" getroffen. Voreingestellt ist in einer AADL-Antwort nur "naal".

Das Beispiel zeigt eine AADL-Anfrage von der Föderationskomponente. In der Anfrage wird in einer bestimmten Region der Typ "shop" gesucht. Überschneiden sich die Region aus der Anfrage mit den Augmented Areas, wird eine Liste ausgegeben, die das Objekt mit der Klasse "shop" enthält. Zusätzlich wird die Ausdehnung der Augmented Areas angegeben, da im Element "resultspec" area = "yes" gesetzt wurde.

```
<aadl>
  <resultspec area = "yes"/>
  <area>
      <gml>
         <polygon>
           <outerBoundaryIs>
             <LinearRing>
                <coordinates>
                      40.0 40.0 40.0, 35.0 35.0, 35.0 35.0, 40.0 40.0, 40.0
                </coordinates>
             </LinarRing>
           </outerBoundaryIs>
         </polygon>
      </gml>
    </area>
    <awschema>
        <objectclass type = "shop" />
    </awschema>
</aadl>
```

Die Anfrage resultiert in einer Antwort, die aus einer Liste von <aadl> Elementen im AAList Format besteht. Die Antwort wird vom Area Service Register an die Föderationskomponente geschickt, da diese die Anfrage gestellt hat. Es ist genauso möglich, daß die Anfragen von einem Spatial Model Server kommen. Dies ist dann der Fall, wenn der Spatial Model Server auf einer existierenden Augmented Area ein Update durchführt.

Das Beispiel zeigt die Antwort im AAList Format:

```
<aalist>
  <aadl>
    <naal>
```

```
      nexus://baden-wuerttemberg.de/stuttgart/stadtmitte/shop
    </naal>
    <area>
      <gml>...</gml>
    </area>
  </aadl>
</aalist>
```

2.5.4 Abbildung der Objekte auf Klassen

Objekte werden in der Augmented World durch räumliche Positionen bestimmt. Dies gilt sowohl für statische als auch für dynamische Objekte. In der Regel weisen Objekte Unterschiede bezüglich ihrer Attributausprägung auf. Um eine Struktur zu generieren ist ein Schema notwendig, das eine Hierachie aufweist. In der Augmented World werden dazu "Class Schemas" definiert (s. Abb. 2.3), in denen die Klassen der Objekte festgelegt werden. Ein Standard Class Schema bildet die Grundlage aller Schemas und kann, falls erforderlich, erweitert werden. Dieses Standard Class Schema besteht aus einer Menge von Objektklassen.

Die Klassen haben ihren Ursprung in den Use-Cases und Szenarios [Nicklas & Mitschang 2001]. Das genaue Aussehen der Klassen ist in [Meßmer 2001] festgelegt. Sie besitzen Attribute, die für die Nexus-Anwendungen von Bedeutung sind. Nexus-Anwendungen und Nexus-Knoten können auf die definierten Attribute und Klassen zugreifen.

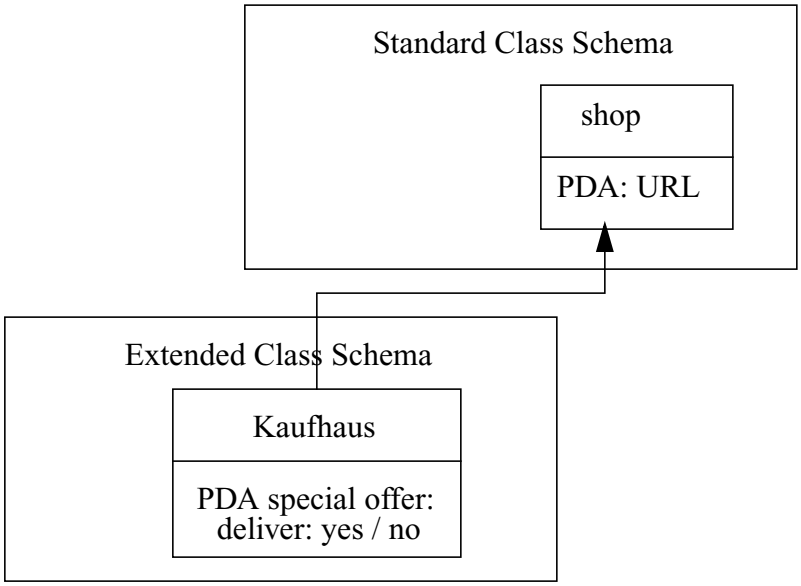

Abb.2.3: Beispiel zu Standar Class Schema

Das Beispiel in der Abbildung 2.3 zeigt, wie die Nexus-Anwendung das Attribut "PDA" vom Objekt "shop" des Standard Class Schemas erhält.

Das Standard Class Schema kann nicht alle möglichen Klassen und alle möglichen Attribute enthalten. Die Komplexität würde auf Kosten der Übersichtlichkeit sehr zunehmen und sich nachteilig auf die Struktur auswirken. Für die Erweiterung des Standard Class Schemas wird der Mechanismus der Extended Class Schemas eingefuhrt worden. Soll eine Anwendung über weitere Klassen verfügen, die noch nicht existieren, so können diese von der Klasse abgeleitet werden, die am nächsten zum Standard Class Schema stehen. Die Abbildung 1.6.4 zeigt diese Möglichkeit. Soll ein neuer Dienst Informationen für Kaufhäuser anbieten können, muß der Dienst eine neue Klasse definieren. Die Klasse wird in diesem Fall von der Klasse "shop" aus dem Standard Class Schema abgeleitet. In der Abbildung 1.6.4 ist aus Gründen der Übersichtlichkeit nur die Ableitung einer Extended Class Schema aus dem Standard Class Schema zu sehen. Dieses ist normalerweise Ausgangspunkt für mehrere Extended Class Schemas.

3 Anforderungsanalyse

Im Kapitel 2 ist die Nexus-Architektur vorgestellt worden. Das Kapitel der Anforderungsanalyse dient dazu, die Anforderungen herauszuarbeiten. Dazu wird die gegenwärtige Iteration der Nexus-Architektur zunächst analysiert. Untersuchungsgegenstand bilden zuerst die Use-Cases. Anhand der Nexus Use-Cases wird untersucht, ob eine Erweiterung auf 3D sinnvoll ist und welche Anforderungen sich aufgrund der Einbindung von dreidimensionalen Daten ergeben. Ausgehend davon liegen die Schwerpunkte auf den Sprachen AWQL und AWML, der Architektur, den Daten der Anfrage, der Position, der Eingabe, dem Ergebnis und auf der Kommunikation.

3.1 Untersuchung der Use-Cases

Festgestellt werden für die Use-Cases 1 bis 17 mögliche Vor- und Nachteile, die sich aus einer Erweiterung auf 3D ergeben. Die zugrunde gelegte Spezifikation (Use-Cases aus Diplomarbeit Nr. 1870) wird dabei nicht verändert.

Use-case 2 **Passive Environmental Information**

Spezifikation: Ein Liste sich ändernder Objekte wird dem User gezeigt, der sich innerhalb einer bestimmten Umgebung (z.B. Stadt, Ausstellung) bewegt. Die Objekte sind abhängig von seiner geographischen Position und seinen Interessen.

3D Daten bieten die Möglichkeit sich umfangreicher und genauer über eine spezielle Umgebung zu informieren. Eine 3D Repräsentation erhöht die wirklichkeitsgetreue Darstellung von Objekten der realen Welt. Damit hat es der Anwender einfacher, die realen Objekte in der realen Welt zu finden. Ferner können Positionsangaben, erweitert um 3D, besser spezifiziert werden. Dadurch kann sich die Suche nach Objekten oder die Wegbeschreibung z.B. zu einer Kunstausstellung verkürzen.

Use-case 3 **Active Environmental Information**

Spezifikation: Ein User will explizit beim Passieren eines Geschäftes informiert werden.

Der User bekommt während er ein Geschäft passiert explizit Informationen über das Sortiment oder ein bestimmtes Angebot. Mit Hilfe der 3D Daten ist es möglich, eine genauere Information bezüglich Angebot und Standort zu geben. Die 3D Daten können zum einen das Objekt (Angebot) enthalten, oder zum anderen die Beschreibung durch Stockwerk und Abteilung sein. Ferner kann mit 3D Daten die Repräsentation des Objektes selbst vielfältiger gestaltet werden. Diese verbesserte Objektpräsentation ist sowohl für “Client” als auch für “Customer” von Vorteil. Dies vereinfacht nebenbei die Werbung für das jeweilige Angebot und bietet neue Werbeformen, -kanäle und -plattformen.

Use-case 4 **Determining A User´s Current Position And His Motion Vector**

Spezifikation: Ein User erfragt die gegenwärtige Position eines anderen bestimmten Users und optional dessen Geschwindigkeit und Richtung mit der sich dieser bewegt.

Die gegenwärtige Position einer Person kann durch 3D Daten genauer und wirklichkeitsgetreuer dargestellt werden. 3D Daten unterstützen eine Richtungsangabe im Raum. Für die Geschwindigkeit, mit der sich die Person bewegt, bringen 3D-Daten jedoch keinen Vorteil.

Use-case 5 **Navigation**

Spezifikation: Ein User will wissen, wie er von seiner Position zu einem gewählten Zielpunkt navigieren muß. Alternativ könnte er auch die Route von einem Startposition zu einer bestimmten Zielposition wissen wollen.

Die Navigation mit 3D stellt an den Anwender höhere Anforderungen durch eine komplexere Navigation, bietet jedoch die Möglichkeit, Positionsangaben sehr viel genauer anzugeben. Die Applikation erhält ein Triple (AP, EP, Orientierung = Blickrichtung). Die Darstellung der Route soll in 3D erfolgen und die Richtungen “rechts” oder “links” können mit dreidimensionalen Daten räumlich erweitert werden. Die Navigationsfunktionen werden um “hoch” oder ”runter” für die räumliche Koordinate ergänzt. Die Darstellung der Route bei einer Navigation, z.B. innerhalb großer Gebäudekomplexe, kann durch 3D Daten sehr viel realitätsnaher dargestellt werden. Dies ist von Vorteil für die Suche in der realen Welt. Neben der reinen 3D Darstellung können nun die Direktiven “rechts” oder “links” um die Direktiven “hoch” oder “runter” vervollständigt werden.

Use-case 6 **GeoMail**

Spezifikation: Karten, die ursrprünglich für ein bereits ausverkaufte Opernvorstellung reserviert wurden, sind nun doch zu haben. Das Theater will den Touristen eine entsprechende Message senden. Verursacht ein Verkehrsunfall einen Verkehrsstau, so können mit Hilfe von GeoMail alle vom Stau betroffenen Fahrer informiert werden.

Eine Benachrichtigung mit 3D Informationen bringt für die Beteiligten (Sender und Empfänger der Nachricht) keine Vorteile. Die Versendung und der Empfang der Nachricht würde aufgrund komplexerer Daten länger dauern. Denkt man aber an die Adressierung von einzelnen Gebäudeteilen aus einem Gebäudekomplex, sind 3D Daten auf jeden Fall sehr nützlich. Eine Verwendung von 3D Daten nur bei Notwendigkeit würde zu einer Reduktion des Aufwandes führen. Die Notwendigkeit besteht dann, wenn mehrere Stockwerke vorhanden sind und kann über eine Abfrage geprüft werden.

Use-case 7 **Query A Service In A Certain Area**

Spezifikation: Ein User will eine Liste aller italienischen Restaurants, aller Postämter oder aller Museen in der Stadt haben.

Besteht das Ergebnis nur in einer Namensliste, z.B. von allen italienischen Restaurants, so kann dabei auf 3D Daten verzichtet werden. Damit verkürzt sich auf jeden Fall die Anfrage und das Ergebnis. In Gebäudekomplexen, die mehrere verschiedene Dienste aus unterschiedlichen Branchen anbieten, wie z.B. in einem Messezentrum, sind 3D Daten für die Orientierung hilfreich. Für große Gebäudeeinheiten, wie das SI-Zentrum oder der Trump Tower, die über sehr viele Stockwerke und damit über mehrere eigenständige Gebiete und Dienste verfügen, ermöglichen 3D Daten eine genauere Positionierung. Verschiedene Dienste können dem Nutzer auf diese Weise ein umfangreicheres Informationsangebot anbieten.

Use-case 8 **Retrieving Information By Pointing At Objects**

Spezifikation: Der User kann durch Zeigen mit einem entsprechenden Device (Telepointer) auf ein bestimmtes Objekt (z.B. Gebäude) Informationen über dieses erhalten.

3D Daten bieten die Möglichkeit, realitätsgetreuere Abbildungen zu erstellen. Damit wird, denkt man an die Wiedererkennung, eine Erleichterung für die “Zeigeoperation” geschaffen. Zeigen auf Objekte wird durch 3D erst richtig sinnvoll, da durch 3D eine komlexere Beschreibung der Objekte ermöglicht und die visuelle Wahrnehmung des Benutzers unterstützt wird.

Use-case 9 **Location-Aware Memo**

Spezifikation: Ein User plant seine Einkaufstour. Dafür schreibt er alle zu kaufenden Gegenstände auf eine Liste. Betritt der User das entsprechende Geschäft, wird er entsprechend informiert.

Für diese Anwendungsmöglichkeit bieten 3D Daten mehrere Vorteile. Zum einen kann der Einkaufsort besser beschrieben werden als nur über zweidimensionale Daten, vor allem wenn sich die Einkaufsgelegenheit oder die Abteilung in unterschiedlichen Stockwerk befindet. Damit wird der Suchvorgang beschleunigt und erleichtert. Weiters können die Produkte mi Hilfe von 3D Daten selbst umfangreicher dargestellt werden. Ansichten über das Produkt, bei Bedarf von mehreren Seiten, schaffen eine bessere Transparenz für den Kunden und können den Werbeaufwand reduzieren. Ferner erleichtert die 3D Darstellung die Wiedererkennung des Produktes in der realen Welt vor Ort.

Use-case 10 **Redirecting Requests for Communication To An Appropriate Terminal Device Close To The User's Current Location**

Spezifikation: Wann immer ein User versucht mit einem anderen User zu kommunizieren (z.B. mit dem Telefon), wird der Kommunikationspfad zu einem dem User nahen Endgerät zurückgeführt. Der User kann dann, sofern er möchte, das Gespräch dort entgegennehmen.

Zurückführende Requests benötigen per se keine zusätzliche Information in Form von 3D Daten. Für diese Anwendung ist eine Position, die durch x und y beschrieben wird, ausreichend. Zur Rückführung des Kommunikationspfades sind keine 3D Daten notwendig. Besteht eines der späteren Features in einer visuellen Präsentation des Pfades, so sind 3D Daten mit Sicherheit eine nützliche Erweiterung. Gerade in großen Gebäudekomplexen hilft eine dreidimensionale Darstellung des Kommunikationspfades die Orientierung zu vereinfachen.

Allerdings ist der Konflikt Privacy vs Location Awareness ist gerade bei dieser Anwendung ein nicht zu unterschätzendes Problem.

Use-case 11 **Virtual Warning Triangles**

Spezifikation: Ein Verkehrsunfall setzt ein virtuelles Warndreieck etliche hundert Meter vor dem Unfallort. Ankommende Fahrzeuge werden durch das Nexus-fähige Navigationssystem gewarnt.

Besteht das Ziel, virtuelle Warnhinweise für die Straßenzustände abzugeben, so ist dafür in erster Linie die zweidimensionale Position x und y für den Standort wichtig. Hier sind 2D Daten ausreichend. Mit Hilfe von 3D Daten könnten jedoch die

Warndreiecke effektvoller dargestellt werden. Die visuelle Aufnahmefähigkeit kann so beim Benutzer positiv beeinflusst werden. Diese Anwendung wäre auch speziell für Rettungseinheiten nutzbringend. Bei Abruf der Information könnten neben einer umfangreichen 3D Wegbeschreibung weitergehende Informationen, die Aussagen bezüglich spezieller Gebäudebeschaffenheiten oder Gebäudegefahren machen, enthalten sein. Je mehr vor der Begehung der Gefahrenstelle durch Rettungseinheiten bekannt ist, desto gezielter und erfolgreicher kann die Gefahrenbeseitigung ablaufen. Nicht nur innerhalb großer Gebäudeeinheiten, sondern auch auf der Straße unterstützen 3D Daten diese Anwendung.

Use-case 12 **Virtual Tickets**

Spezifikation: Ein Nexus User kauft ein Ticket bei einem virtuellen Kiosk einer Ausstellung. Während er den Eingang passiert, checkt das System, ob er eine gültige Eintrittskarte besitzt.

Die Realisierung virtueller Tickets mit 3D Daten führen für den Anwender zu keinem weiteren Vorteil.

Use-case 13 **Showing Conductions**

Spezifikation: Bei Betrachtung eines Mauerfläche wird ein spezieller Augmented Reality Bildschirm verwendet, der alle Leitungen innerhalb der betrachteten Fläche anzeigt.

Für die bloße Ansicht sind zunächst keine 3D Daten notwendig. Betrachtet man aber Mauerflächen die größer und höher sind, ist die Repräsentation mit 3D Daten hilfreich und erleichtert die Orientierung. Es wird eine größere Realitätsnähe der realen Welt erreicht, da Flächen mit 2D und die Höhe mit 3D Daten beschrieben werden kann.

Use-case 14 **Control Of Real Objects Via Their Model Representation**

Spezifikation: Die Veränderung des Objektzustands in der Modellwelt muß zu einer Änderung des entsprechenden Zustands in der realen Welt führen

3D Daten ermöglichen die Repräsentation realer Objekte sehr viel wirklichkeitsgetreuer als die Abbildung über zweidimensionalen Daten. Reale Objekte können mit 3D Daten besser dargestellt werden. Es kann wesentlich umfangreicher das Modell und dessen Kontrollmöglichkeiten beschrieben werden. Die Kontrolle realer Objekte, wie z.B. die Modellbahn, durch ein Handheld Device wird mit 3D Daten auf jeden Fall erleichtert.

Use-case 15 **Changing Objects**

Spezifikation: Dieses Use-Case ist das Gegenteil von Use-Case 13. Die Zustandsänderung eines Objekts der realen Welt führt zu einer Änderung des virtuellen Objekts.

Die Änderung des Zustands muß nicht zwangsläufig mit 3D Daten erfolgen. Objekte der realen Welt können aber mit 3D Daten einfacher und wirkungsvoller präsentiert werden. Wird ein mit 3D Daten abgebildetes Objekt der realen Welt verändert, sollte das wieder in einem dreidimensionalen Modell resultieren. Eine Reduktion auf ein zweidimensionales Modell würde den Realitätsgedanken nicht fördern.

Use-case 16 **Identifying Objects By Image Processing**

Spezifikation: Ein User übermittelt ein Bild eines Objekts aus der realen Welt zu einer Nexus Anwendung. Die Anwendung versucht das Objekt zu identifizieren, indem das Bild mit den Objekten der Augmented World abgeglichen wird.

Übermittelt der Benutzer ein zweidimensionales Bild, so sind die Objekte nur in zwei Dimensionen repräsentiert. Ist die gesamte Augmented World mit dreidimensionalen Objekten erstellt, wird der Abgleich mit dem zweidimensionalen Bild erschwert, da mit einem 2D Bild versucht wird, Objekte zu identifizieren, die dreidimensional visualisiert werden. Das Matching könnte dann vereinfacht werden, wenn die Ausgangsdaten bereits 3D Informationen beinhalten. Andererseits könnten von allen Objekten alle Ansichten gespeichert werden. Auf dieser Grundlage ist ein Abgleich vielversprechender und effizienter. Es erhöht sich im Gegenzug der Speicheraufwand pro Objekt um ein Mehrfaches, abhängig von der Anzahl der zu speichernden Ansichten. Die Entfernung, aus der ein Bild von einem Objekt erstellt wird und der Maßstab der Objekte, mit denen die Objekte der Augmented World verglichen wird, stellen ein weiteres Problem dar.

Use-case 17 **Events Addressing Different People With Different Purpose**

Spezifikation: Ein Einkaufszentrum will eintretende Kunden über ihre Tagesangebote informieren und schickt diesen eine Nachricht, die einen Überblick enthält. Im Gegensatz zu Use-Case 5 sind die Betreiber interessiert an der Anzahl der Kunden, die eintreten und an bestimmten Eigenschaften, die diese haben.

Besteht der Dienst aus einer Benachrichtigung in Form einer kurzen Mitteilung, mit dem Ziel nur textbasiert zu benachrichtigen, sind zweidimensionale Daten ausreichend. 3D Daten wären an dieser Stelle zu umfangreich und würden nur zu

einer Verzögerung führen, die nicht von Vorteil wäre. Umfangreiche Information durch Bildpräsentationen über das Angebot steht dabei nicht im Vordergrund. Ein Nachteil besteht aber darin, Kunden nicht visuell über die Angebote informieren zu können. 3D Daten sind gerade hier von Vorteil, da sie die Visualisierung der Angebote unterstützen.

Use-case 18 **Navigation Within Mobile Objects**

Spezifikation: Die Navigation innerhalb großer mobiler Objekte wie Schiffe oder Züge soll genauso möglich sein, wie die Navigation innerhalb statischer Objekte (Use-Case 4). User, die sich auf einem Schiff bewegen, das den Atlantischen Ozean überquert, sollen sich zurechtfinden können, wenn sie das Nexus Navigationssystem verwenden.

Für diese Anwendungsmöglichkeit sind 3D Daten nicht nur hilfreich, sondern sehr wichtig für die Umsetzung. Mit Hilfe dreidimensionaler Information kann die Navigation im Raum, wie z.B. im Inneren eines Schiffes oder eines großen Gebäudes überhaupt erst sinnvoll durchgeführt werden. Im ersten Fall handelt es sich um die Navigation innerhalb eines mobilen Objekts, im zweiten Fall um die Navigation innerhalb eines statischen Objektes. Die Bewegung in einem mobilen Objekt führt zu einer Komplexität bei einer Positionsbestimmung, die sich nicht nur auf die direkten Referenzpunkte eines Objektes, wie z.B. den Schiffsrumpf, bezieht, sondern auf die gesamte Umgebung. Problematisch wirken sich objektspezifische Daten aus, wenn über verschiedene Innenleben mobiler Objekte generalisiert wird. Solange es keine Standards hinsichtlich der Gebäudeorientierung gibt, bleibt ein Außenbezug von objektgebundenen Daten zu einer gesamten Umgebung jeweils spezifisch.

Neben diesen objektbezogenen Anforderungen ergeben sich noch weitere, die die Nexus-Architektur betreffen. Statische und dynamische Objekte müssen unterschiedlich verwaltet werden ([Nicklas & Mitschang 2001]), da die Spatial Model Server den Anforderungen zur Speicherung mobiler Objekte nicht gerecht werden. Location Server werden eingesetzt, um Informationen dynamischer Objekte zu verwalten. Ein weiteres Problem ergibt sich, wenn ein mobiles Objekt die Grenzen einer Augmented Area zu einer anderen überschreitet oder eine Augmented Area ganz verlassen wird. Die Objekte müssen an den nächsten zuständigen Server übergeben werden, ohne daß es zu einem Datenverlust kommt.

3.19 Anforderungen an bestehende Systeme

Dreidimensionale Daten führen zu einer größeren Komplexität der Daten an sich. Ferner steigen die Anforderungen an das System, umfangreichere Daten verarbeiten zu können. Der gegenwärtige Systemzustand wird dem aus mehreren Gründen nicht gerecht.

Generell können aus einem DBS oder GIS nur Informationen erfragt werden, die gespeichert sind. Standardanfragen (standard database queries) an die Datenbank sind möglich. Ferner können Anfragen, die Ort und Raum betreffen (location- and spatial-aware queries), bereits getätigt werden. Diese sind aber auf zweidimesionale Daten beschränkt. Positionsangaben sind ebenfalls nur in x, y Koordinaten möglich. Komplexe Anfragen, die einen Zeitaspekt (time-aware queries) enthalten, oder kombinierte Anfragen (combined queries) sind noch nicht möglich.

3.20 Analyse der Sprachen AWQL und AWML

AWML und AWQL sind in der gegenwärtigen Iteration nur auf zweidimensionale Daten ausgelegt. Beide setzen auf der Graphischen Markup Language [GML 2.0] auf. GML unterstützt in der derzeitigen Version standardmäßig keine 3D Daten. Die folgende Abbildung zeigt die Geometriehierachie.

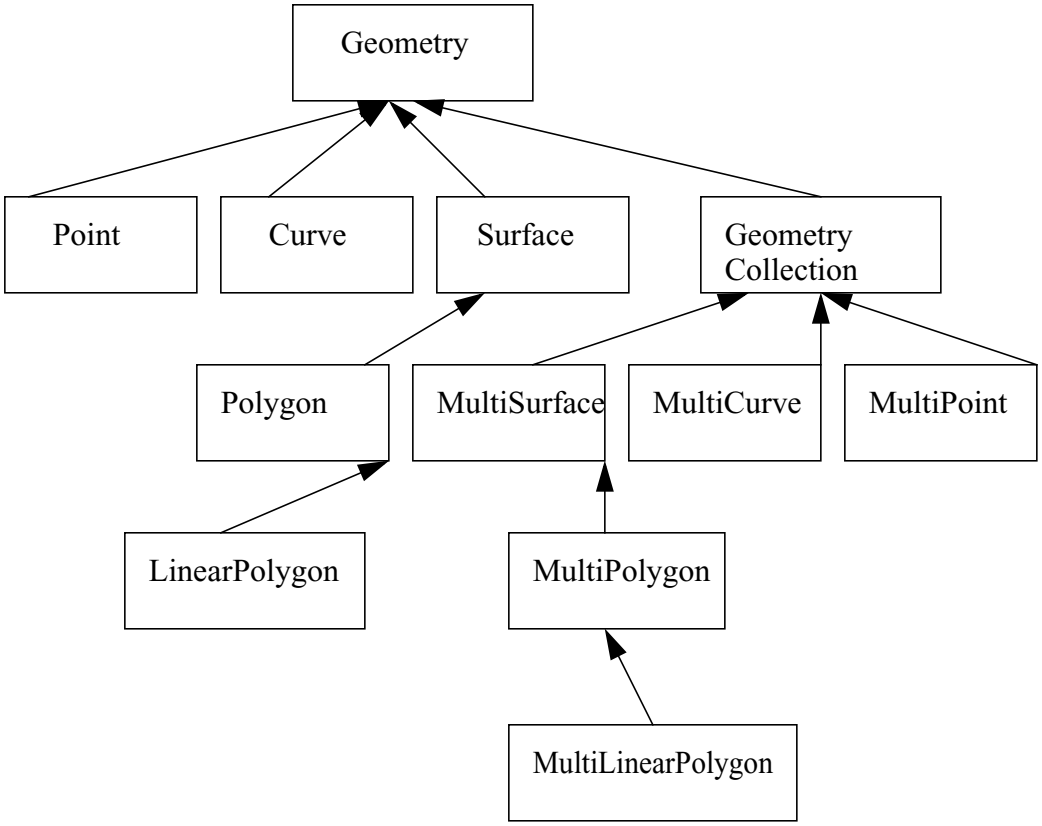

Abb.3.0: Geometriehierachie

Es können komplexe Strukturen oder Objekte, wie Gebäude mit den geometrischen Klassen aus Abbildung 3.0 noch nicht dargestellt werden, da die Klassen in der gegenwärtigen Iteration nur auf zwei Dimensionen ausgelegt sind. Die Geometry Klasse bildet den Ausgangspunkt für die Klassen Point, Curve, Surface

und GeometryCollection. Von diesen zweidimensionalen Klassen lassen sich weitere geometrische Klassen ableiten, wie z.B. die Polygon Klasse. Die Ableitungen sind ebenfalls zweidimensional.

Eine Präsentaion dreidimensionaler Daten in AWML, die für die Ergebnisdarstellung zuständig ist, ist nicht in 3D möglich, da noch keine 3D Objekte vorhanden sind. Es besteht die Möglichkeit, in AWQL die Attribute geometrischer Objekte zu ändern, in dem sie generalisiert oder aggregiert werden.

generalization

Dieses Element wird verwendet, um die geometrischen Attribute zu generalisieren. Die geometrischen Eigenschaften sind nur auf einer zweidimensionalen Ebene definiert. Das Ergebnis bleibt dementsprechend zweidimensional.

aggregation

Kleinere Objekte können zu größeren Objekten verbunden werden. Die geometrische Dimension wird davon nicht betroffen, d.h. mehrere zweidimensionale Objekte zu einem komplexeren dreidimensionalen Objekt verschmelzen zu lassen, führt im Ergebnis zu keinem dreidimensionalen Objekt.

Beide Objektänderungen resultieren im Ergebnis wieder zu einem Objekt gleicher Dimension. Objekte können also auf diese Weise nicht in der Dimension erweitert werden. Eine Transformation zweidimensionaler Objekte in dreidimensionale Objekte ist mit Änderungen, die auf geometrischen Attributen beruhen, nicht herbeizuführen.

Die Bildgenerierung erstellt Bilder auf Basis eines konkreten Umfeldes. Dazu sind Daten, notwendig, die den Inhalt beschreiben. Es handelt sich dabei um Parameter, die die Positon im Raum und die Bildqualität beeinflussen. Ferner wird eine horizontale und vertikale Blickrichtung für die Eingrenzung des Bildinhalts benötigt. Den Anforderungen der Positionsbestimmung und der Blickrichtung wird die gegenwärtige Iteration von AWQL nicht gerecht.

3.21 Untersuchung der Anforderungen

Für die Umsetzung ist es wichtig zuerst die Anforderungen festzustellen. Die Einbindung des Mehrwertdienstes Bildgenerierung bewirkt eine Erweiterung des Nexus-Systems, sowohl architektonisch, als auch hinsichtlich der Leistung. Die dafür notwendigen Anforderungen umfassen Anfrage, Ergebnis, Architektur, Kommunikation und Orientierung in einer Umgebung.

3.22 Anforderungen an Anfragen

Grundanforderung ist die Voraussetzung, Anfragen an den eingebundenen Dienst stellen zu können. Anfragen enthalten Daten unterschiedlicher Art. Darunter ist auch die Position, die den Ort oder das Objekt in seinem Umfeld lokalisiert. Die Position kann in zweidimensionalen Daten oder dreidimensionalen Daten angegeben werden. Ist eine Höhe nicht bekannt, besteht die Position nur aus zwei Dimensionen. Der Fall tritt ein, wenn z.B. der Benutzer über die lokale Höhe seiner Position nicht verfügt. Für die Generierung eines Bildes muß die Position nicht in dreidimensionalen Daten angegeben werden. Die Untersuchung der Use-Cases hat aber gezeigt, daß drei Dimensionen für die Orientierung im Raum sinnvoll ist. Ferner wirken Bilder realitätsnaher, wenn bei der Generierung über drei Dimensionen verfügt werden kann. Um eine Anfrage bezüglich der Position zu komplementieren, die nur zwei Dimensionen enthält, könnte man für den fehlenden Wert einen Defaultwert einsetzen.

3.23 Anforderungen an die Position

Ein Bild kann nur dann generiert werden, wenn die Daten in der Anfrage an den Dienst einer bestimmten Form genügen. Der Dienst erstellt das Bild auf Grundlage von Gauß-Krüger Daten. In diesem Koordinatensystem müssen die Daten der Position vorliegen. Die Position wird durch x, y und z festgelegt. Der Blickwinkel, der das Sichtfeld definiert, ist nicht an ein bestimmtes Koordinatensystem gebunden. Ausgangspunkt könnte z.B. die Anfrage eines Benutzers sein, der sich in einer bestimmten Umgebung aufhält und nähere Informationen über ein konkretes Objekt in Form eines Bildes haben möchte. Seine lokale Position wird durch WGS84 Koordinaten bestimmt. Soll diese Position als Grundlage für eine Bildgenerierung dienen, ist zuvor eine Koordinatentransformation von WGS84 nach Gauß-Krüger erforderlich.

3.24 Anforderungen an die Blickrichtung

Für die Orientierung im Raum wird die Blickrichtung benötigt, die auf 2 Arten übergeben werden kann. Als 2 Wertetripple (x,y,z), die miteinander verbunden einen Richtungsvektor im Raum bilden oder als Winkel (head, pitch, role). Die Bestimmung mit 2 Winkeln (head, pitch) ist ausreichend, da der letzte Winkel (role) einen Rotationswinkel vergleichbar mit einer seitlichen Flugrolle eines Flugzeugs beschreibt und gegenwärtig in keinem Use-Case und keiner Anwendung von Nexus verwendet wird.

3.25 Schematisches Architekturbild für die Erstellung von 3D Bildern

Nachfolgendes Bild zeigt die an der Kommunikation beteiligten Komponenten Föderation, Mehrwertdienst (MWD) und 3D Bilddienst.

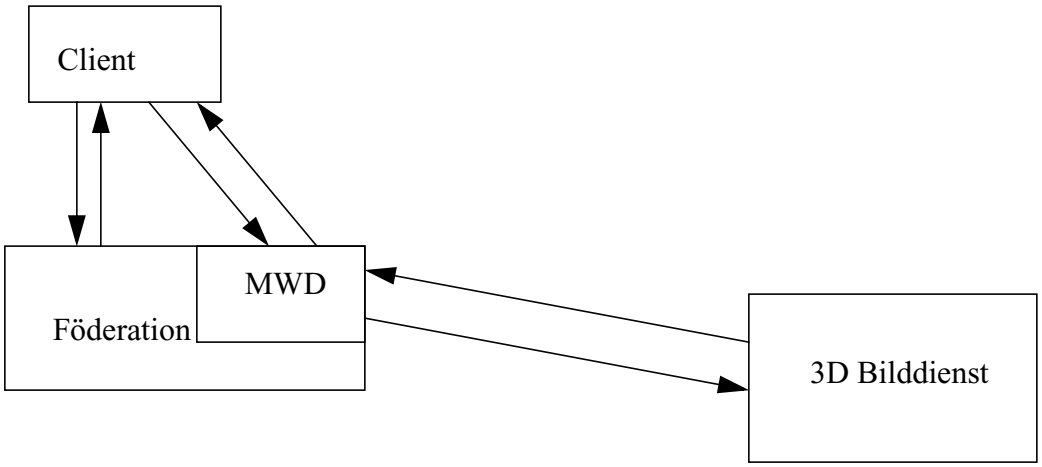

Abb.3.1:Ausschnitt aus der Nexus-Architektur mit 3D Bilddienst

Es werden jetzt Anforderungen an Anfrage, Eingabe, Ergebnis, Benutzer, Funktions-, Datenumfang und die zu definierende Schnittstelle für den Bilddienst untersucht.

3.26 Anforderungen an das Ergebnis einer Anfrage

Ergebnis einer Benutzer-Anfrage, die eine Positionsangabe und die Blickrichtung enthält, ist ein Bild, z.B. ein jpeg-Bild. Das Bild soll alle Objekte im Sichtfeld, das in etwa 60 Grad beträgt, darstellen. Ferner sollen unterschiedliche Bildformate möglich sein, da die Darstellung auf unterschiedlich großen Displays erfolgen soll. Andere Darstellungsformate wie tiff und png sind ebenfalls möglich. Besteht ein Nebenziel darin, die Übertragungszeit des Ergebnisses zu verkürzen, ist es sinnvoll, statt dem Bild nur die URL zurückzugeben. Die URL gibt dem Benutzer die Stelle an, an der das Bild gespeichert wird und ist mit einem Browser aufrufbar. Verfährt man auf diese Weise, spiegelt sich die Verkürzung in der Übertragung in einem Gewinn an Performanz wieder.

3.27 Analyse der Anforderungen an die Eingabe

Die Anfrage wird durch den Benutzer zur Bearbeitung an das System übergeben. Der Benutzer hat die Möglichkeit, seinen Standpunkt und die Blickrichtung einzugeben. Der Standpunkt besteht aus einer Positionsangabe. Bezüglich des Bildes kann die Art der Grafik, die Auflösung des Displays und eine Darstellung in Schwarz/Weiß bzw in Farbe (abhängig vom Leistungsumfang des Displays) ausgewählt werden. Der Default ist vorerst Farbe. Der Inhalt des Bildes kann über eine Attributselektion der Objekttypen spezifiziert werden.

3.28 Anforderungen an Anfrage- und Ergebnisparameter

In Anfrage und Ergebnis werden keine statischen Werte, sondern Parameter übergeben, um so dynamisch auf Änderungen reagieren zu können. Nachfolgend wird zunächst auf die Anfrage, dann auf das Ergebnis eingegangen.

3.28.1 Übergabe der Parameter in der Anfrage

Die folgende Tabelle zeigt Eingabeparameter und das Format

Parameter	Benutzereingabe
Standpunkt	Positionsangabe x und y zwingend, z optional
Blickrichtung	Blickwinkel horizontal, vertikal
Selektion der Objekttypen	real, virtuell
Art der Grafik	jpeg, tiff, png
Auflösung	width, height
Farbe bzw S/W	gray = no

Tabelle 1: Übergabeparamater

3.28.2 Übergabe der Ergebnisse

Das Ergebnis ist ein Bild, z.B. im jpeg-Format, welches vom 3D Dienst an den Client als SOAP-Response zurückgesendet wird, das wie die Anfrage ein XML-Dokument ist.

3.29 Anforderungen an den Benutzer

Jeder der die Anwendungen von Nexus nutzt, im der speziellen Nutzer des 3D Mehrwertdienstes, kann ein Benutzer sein, der über Erfahrung besitzt oder unerfahren ist. Um der Benutzerfreundlichkeit Rechnung zu tragen, sollte die Interaktion in einem gewissen Umfang abhängig vom Kenntnisstand der Benutzer gestaltet sein.

3.30 Anforderungen an Funktionsumfang und Datenumfang

Datenbasis bilden die dreidimensionalen Daten des Instituts für Photogrammetrie. Diese setzen zugleich die Grenzen der Möglichkeiten, Leistungsumfangs und der Bilder. Der Funktionsumfang enthält die Module, Funktionen und Berechnungen, die notwendig sind, um 3D Daten in die Nexus-Architektur einzubinden und darzustellen. Es ist dazu eine Schnittstelle notwendig, die die Parameter aus der XML-Anfrage übernimmt und an den Bilddienst weitergibt und das XML-Ergebnis entgegennimmt. Der Funktionsumfang des Dienstes besteht in der Generierung eines Bildes auf Basis der Anfrage. Für Anfrage und Ergebnis ist jeweils eine eigene DTD notwendig, um die Formate zu überprüfen. Ferner muß eine Anfragesprache vorhanden sein, die die Parameterwerte einer Anfrage an den Dienst weitergeben kann. Mit Hilfe der Anfragesprache kann auf die in der DTD definierten Elemente im XML-Dokuments zugegriffen werden. Darüberhinaus ist für den Transport von XML-Anfrage und XML-Ergebnis zwischen Client auf der einen Seite und dem Bilddienst auf der anderen Seite eine Kommunikationsstruktur erforderlich.

Der Funktionsumfang in der ersten Iteration besteht in der Entwicklung von DTD, Schnittstelle und Anfragesprache. In der zweiten Iteration ist das Ziel die Erstellung der Kommunikationsstruktur.

3.31 Anforderungen an die Schnittstelle

Es wird für die Datenübergabe von Anfrage und Antwort eine Schnittstelle benötigt. Sie dient der Übergabe der Parameter durch den Benutzer auf der einen Seite und auf der anderen Seite wird über sie die vom Institut für Photogrammetrie produzierten Bilder übergeben. Parameter sind Standpunkt, Blickrichtung (Winkel), Attributselektion, Art der Grafik, Auflösung und Schwarz/Weiß- bzw. Farbdarstellung. Diese werden als Ausgangslage für die Bildgenerierung an den

Bilddienst übergeben. Die Schnittstelle bekommt eine XML-Anfrage in Form eines XML-Dokumentes, das anhand der Anfrage-DTD auf Gültigkeit überprüft wird (= wellformed). Dazu muß eine Anfrage-DTD vorhanden sein. Der Bilddienst übergibt eine XML-Antwort, die ein Bild z.B. im jpeg-Format enthält. Dafür muß eine eigene Antwort-DTD entworfen werden.

Die Positionseingabe in der Anwendung erfolgt in WGS84 Koordinaten. Für die Schnittstelle ist das unwichtig, da über diese nur die Weitergabe der Parameter erfolgt. Die Berechnung der Bilder im Bilddienst erfolgt auf Basis von Gauß-Krüger Koordinaten. Generiert der Bilddienst aufgrund einer Eingabe in WGS84 ein Bild, basierend auf Gauß-Krüger Daten, würde dies zu einem Datenkonflikt führen, da beide Koordinatensysteme unterschiedliche Ursprungspunkte besitzen und die Orientierung in diesen unterschiedlich vorgenommen wird. Um eine Übereinstimmung zwischen beiden Koordinatensystemen zu erreichen, muß die Anfrage (WGS84) in das Gauß-Krüger Koordinatensystem konvertiert werden. Das Ergebnis dieser Konvertierung ist die Anfrage in Gauß-Krüger Koordinaten, mit der ein Bild in Gauß-Krüger Koordinaten berechnet wird. Eine Konvertierung des Bildes in das Koordinatensystem der Anfrage (WGS84) ist nicht mehr notwendig, solange das Ziel in der Bildberechnung liegt.

3.32 Anforderungen an die Kommunikation

Die Einbindung des Dienstes genügt allein nicht, um Anfragen stellen zu können und Antworten zu bekommen. Beide müssen an die Adressaten zugestellt werden können. Adressaten sind ein Client auf der einen Seite und der Bilddienst auf der anderen. Für Anfrage und Antwort treten Client und Dienst in Interaktion, indem sie miteinander kommunizieren. Eine Kommunikation bedarf einer Struktur, die zuerst aufgebaut werden muß. Client und Dienst müssen bekannt sein. Ferner müssen die Daten transportierbar sein. Dazu müssen diese serialisierbar sein. Der Einsatz eines standardisierten Protokolls vereinfacht den Transportvorgang. Verwendet man dabei das Simple Object Access Protocol (SOAP) und die Programmiersprache Java, wird dem Gedanken der Serialisierung am meisten Rechnung getragen.

Hotspots der Kommunikation sind zwischen Client und Föderation, zwischen Client und Mehrwertdienst, und zwischen Mehrwertdienst (MWD) und dem 3D Dienst.

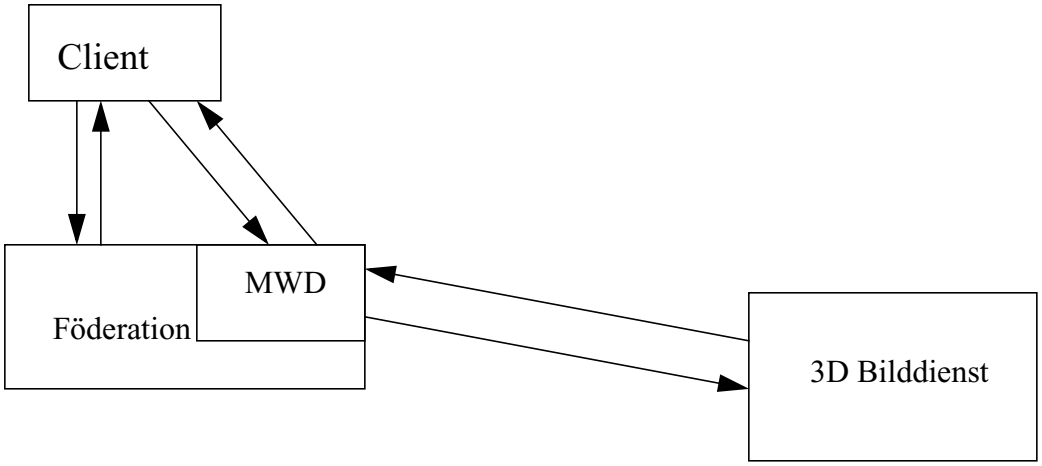

Abb.3.2: Gesamtbild der Kommunikation zwischen Client, Föderation und 3D Dienst

Im folgenden wird die Kommunikation an den eben genannten Punkten näher betrachtet.

3.33 Kommunikation zwischen MWD und 3D Bilddienst

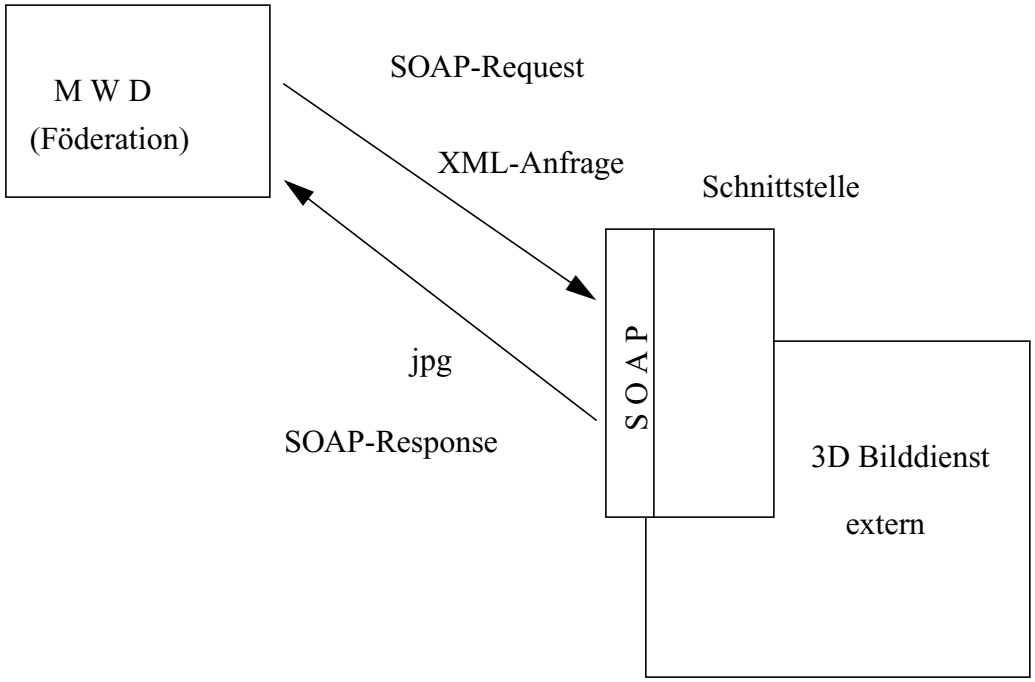

Abb.3.3: Kommunikation zwischen MWD und 3D Bilddienst

Die Kommunikation zwischen MWD und 3D Bilddienst basiert auf dem SOAP Protokoll. SOAP stellt einen einfachen Mechanismus zum Austausch von strukturierter und typisierter Information zwischen Rechnern in einer dezentralisierten, verteilten Umgebung zur Verfügung.. Ziel ist ein Mechanismus, um die interne Semantik einer Anwendung zu beschreiben. Es werden ein modulares Programmpaket sowie Mechanismen zur Verschlüsselung von Daten innerhalb von Modulen zur Verfügung gestellt. Daneben besteht die Möglichkeit, durch einen einfachen Mechanismus auf entfernten Rechnern Methoden aufzurufen [Apache SOAP 2001]. SOAP-Nachrichten werden in XML verfaßt.

Der Client ruft durch einen SOAP-Request den 3D Bilddienst auf einem entfernten Server auf. Die XML-Anfrage wird mit den Parametern für die Bildberechnung an den Dienst geschickt. Auf Grund der SOAP-Nachricht werden beim Bilddienst die entsprechenden Methode aufgerufen, um das Bild zu berechnen. Dabei erfolgt die Übergabe der aktuellen Parameterbelegung an den Dienst. Nach Ablauf der Methode wird dem Client das Bild durch eine SOAP-Response Nachricht zurückgeschickt.

3.34 Kommunikation zwischen MWD und Client

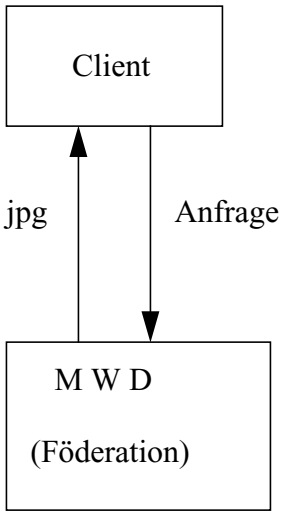

Abb.3.4

Abb.: Kommunikation zwischen MWD und Client

Der Client sendet seine Anfrage an den Mehrwertdienst. Die Anfrage wird in einem XML-Dialekt formuliert, den der Mehrwertdienst versteht. Dazu ist es notwendig eine XML-Anfragesprache zu entwerfen.

3.35 Anforderungen bezüglich der Dienstbereitstellung

Der Dienst für die Generierung der Bilder soll nach Einbindung in die Nexus-Plattform über das Web erreichbar sein. Dazu bietet sich für die Bereitstellung Apache als Webserver an. Das Deployment wird dann über den Webapplikations Server Tomcat [Tomcat] realisiert. Der Dienst ist in C++ implementiert. Die Folge ist, daß nun Tomcat nicht mehr eingesetzt werden kann, da dieser C++ Anwendungen nicht "deployen" kann. Die Verwendung von SOAP als Transportprotokoll bleibt sinnvoll, da SOAP C++ unterstützt. Es bietet sich auch gSOAP [Engelen & Kyle A. Gallivan] als Lösung an. Unter gSOAP versteht man eine Sammlung von Generator Tools für die Entwicklung von Web-Diensten und Anwendungen in C und C++. Sie verfügen über eine C/C++-transparente SOAP API, die auf diese Weise den Dienst für die Erstellung von Bildern an das WWW anbinden können.

4 Lösungsansätze

Das vorangehende Kapitel 3 dient dazu, die Probleme und Anforderungen herauszuarbeiten. Im folgenden werden entsprechende Lösungsmöglichkeiten diskutiert. Dazu gehören zunächst Erweiterungen am Datenmodell und an der Anfragesprache. Daran schließt eine objektgebundene Erweiterung an. Es folgen die beiden zentralen Lösungsansätze Wrappertechnologie und Mehrwertdienst. Dazu werden auch Webservices als Integrationtechnologie für Mehrwertdienste betrachtet. Die Zusammenfassung rundet das Kapitel ab.

4.1 Datenmodellerweiterung

Eine Datenmodellerweiterung, die komplexere Objekte und Strukturen in 3D darzustellen soll, kommt ohne höhere geometrische Modelle nicht aus. Dazu gehören auf jeden Fall die Würfel, Quader, Zylinder, Kegel und Kugel. Aus diesen einfachen Grundformen können größere 3D Strukturen aufgebaut werden. Können ganze Objekte gespeichert werden, so verringert sich der Aufwand für die Darstellung der Objekte und das Retrievel kann verkürzt werden, wenn ganze Objekte nicht mehr erst in kleinere Einheiten (z.B. Punkte) zerlegt werden müssen. Ferner ist eine Indexstruktur über Objektklassen und ganze Objekte sehr hilfreich. Es erhöht sich jedoch der Speicheraufwand für die Objekte aufgrund deren Größe und der Aufwand in Administration. Räumliche Indexe werden benötigt. Die Objektpräsentation ist abhängig von der Anzahl der möglichen Objekte und der Güte (Qualität) der Darstellung. Die Darstellungsmächtigkeit der Objekte wird jedoch begrenzt von der Vielfalt der geometrischen Objekte des erweiterten Modells.

4.2 Anfrageerweiterung

Komplexere Anfragen bieten umfangreichere Möglichkeiten, große 3D Strukturen zu erfassen. Dabei kommt man nicht umhin, GML, AWML und AWQL um 3D relevante Sprachelemente zu ergänzen. GML bietet die Möglichkeit, beliebige eigene Klassen und Strukturen zu definieren, 3D Objekte eingeschlosen. GML bildet die Grundlage für AWML und AWQL. Eine Erweiterung von GML impliziert indirekt eine Erweiterung sowohl von AWML als auch von AWQL. Dazu gehört die sprachliche Erweiterung, um sowohl räumliche Positionierung als auch den Eindruck einer dritten Dimension verarbeiten zu können. Notwendig sind z.B. räumliche Operatoren (davor, dahinter, innen) und räumliche Indexe. Ferner müssen für den 3D Effekt unterschiedliche Blickwinkel aus verschieden Psoitionen ermöglicht werden. Im Raum dynamische schnelle Objekte (innerhalb eines Gebäudes) erfordern die Speicherung und Kontrolle der Bewegungsgleichung,

während bei langsamen Objekten die Speicherung einzelner Schnappschüsse reicht. Räumlich ausgedehnte Objekte mit sich zeitlich ändernder räumlicher Ausdehnung, z.B. ein Verkehrsstau, benötigen eine räumliche und zeitliche Dimension, um temporalen Anfragen gerecht zu werden.

4.3 Objektgebundene Erweiterung

Nicht alle Objekte stehen im Mittelpunkt des Interesses für die Benutzer. Es gibt wichtigere Objekte, z.B. das Museum eines Touristguides, und weniger wichtige wie z.B. ein Baum in einer Straße. Es macht natürlich mehr Sinn für interessante Objekte (z.B. Ausstellung in einem bestimmten Stock des Museums) weitergehende Informationen abzulegen, die auf Abruf bereitstehen. Objekte, die nicht bedeutsam für den Benutzer sind, benötigen weder eine umfangreiche Darstellung (in 3D) noch tiefergehende Informationen über das Objekt. Dies führt zu einem erheblich geringeren Speicheraufwand. Im Forschungsprojekt DeepMap [Reuter] wird dieser objektorientierte Ansatz verfolgt.

Problematisch kann es unter Umständen sein, objektiv zu entscheiden wie wichtig ein Objekt ist. Dies könnte auch zu einem Verlust der Information führen.

4.4 Mehrwertdienst

Mehrwertdienste dienen dazu, die Funktionalität und bisherige Leistungsfähigkeit zu erweitern. Besteht das Ziel in der Verwendung leistungsarmerer Clients, oder stehen sowieso nur leistungsschwache Ressourcen bereit, könnte man durch Nexus-Anwendungen z.B. Daten auf einen Nexus-Knoten oder die Anwendungslogik auf einen Nexus-Knoten laden. Die Anfrageerstellung in AWQL und Umwandlung in AWML muß nun nicht mehr von den Clients ausgeführt werden. Es reduzieren sich auch die AWQL Aufrufe und als Folge die zu übertragenden Datenmengen. AWQL bietet jedoch in der gegenwärtigen Version nur einfache Anfragen. Das Ergebnis einer Anfrage kann sich so aus mehreren Teilergebnissen zusammensetzen. Ein Mehrwertdienst auf einem Nexus-Knoten könnte die Ergebnisse zusammenführen. Der Client wird auf diese Weise entlastet. Aufgrund der Entlastung kann nebenbei die Kommunikation an sich vereinfacht werden.

Nachfolgende Abbildung zeigt wo ein 3D Mehrwertdienst innerhalb der Nexus-Architektur plaziert ist.

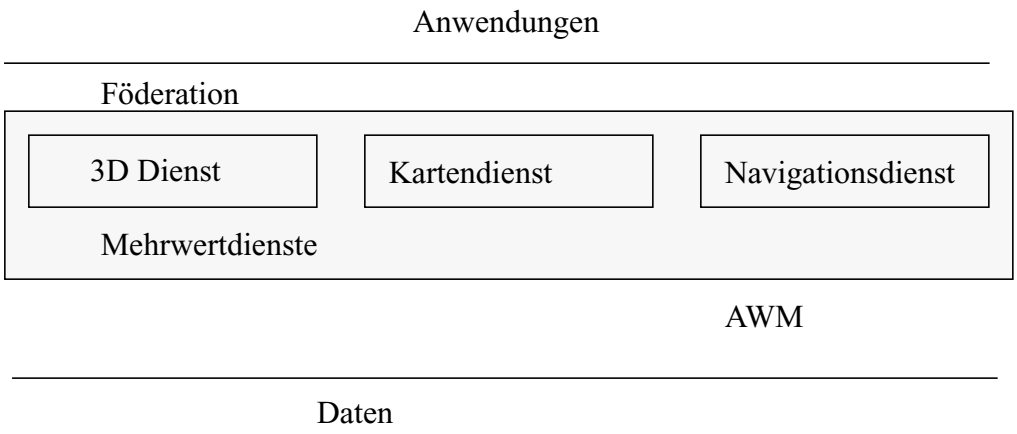

Abb.4.0: Einbindung der 3D Daten als Mehrwertdienst in Nexus

Angenommen, der 3D Dienst enthält die Funktionalitäten, die für die 3D Einbindung und 3D Darstellung notwendig sind, dann kann dieser 3D Mehrwertdienst wie die anderen Mehrwertdienste auf einen Nexus-Knoten geladen werden und damit dessen Funktionalität um 3D erweitern. Beispiele für Mehrwertdienste sind der Karten- und Navigationsdienst, die beide die Anwendungsschnittstelle nutzen, um Daten von der Föderation zu erhalten. Der 3D Mehrwertdienst kann auch auf die Föderation zugreifen und so 2D Daten integrieren. Die Nexus-Anwendungen können jedoch nur über ihre eigene proprietäre Schnittstelle auf solche Mehrwertdienste zugreifen, wie dies in der nachfolgenden Abbildung zu sehen ist.

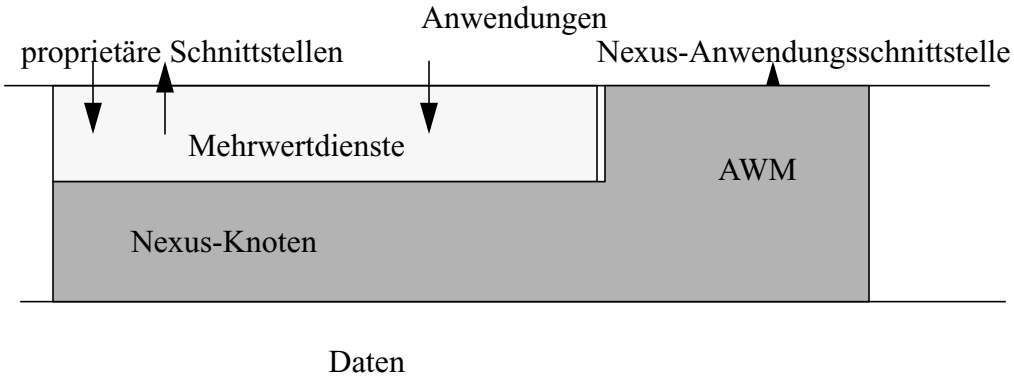

Abb.4.1: Angelehnt an Abbildung über den Aufbau der Föderation [Nicklas 2001]

Über die Anwendungsschnittstelle werden Anfragen und Ergebnisse verarbeitet. Anfragen werden in AWQL formuliert, Ergebnisse werden in AWML zurückgeliefert. Allerdings sind Requests in AWQL nur in 2D möglich während Replys in AWML in 3D möglich sind. Den unterschiedlichen Fähigkeiten von AWQL und

AWML muß auch der Mehrwertdienst gerecht werden. Der Sprachumfang von AWQL und AWML ändert sich durch den Mehrwertdienst nicht. Es könnte jedoch durch eine Erweiterung eine Anpassung vorgenommen werden, um eine sprachlich einheitliche Ebene zu schaffen. Im Vergleich zur 3D Einbindung in Nexus durch einen Wrapper ist die Konsquenz auf jeden Fall ein erhöhter Aufwand.

4.4.1 Integration bestehender Systeme als Mehrwertdienst für die Speicherung von 3D Daten

Bestehende Systeme für die Einbindung können sowohl Eigenentwicklungen, als auch Systeme wie Geoinformationssysteme (GIS) und Datenbanksysteme (DBS) umfassen. Eigenentwicklungen können ausgeschlossen werden, da keine bestehenden Systeme vorliegen, die dafür in Betracht kommen. Es bleiben GIS und DBS als weitere Möglichkeiten für die Speicherung von 3D Daten als Mehrwertdienst. DBS scheiden aus, da noch keine Erweiterungen für 3D Daten existieren. GIS können 3D Daten speichern und kommen für eine Einbindung in die Nexus-Architektur als bestehendes System in Frage. Allerdings stellt das AWM mit AWQL und AWML für diesen Mehrwertdienst weiterhin einen Gordischen Knoten dar. Reduziert man die Einbindung bestehender Systeme auf funktionale Einheiten, die nicht nur die Speicherung umfassen, so finden sich Einheiten, die sowohl in 2D wie auch in 3D Verwendung finden. Dazu gehören z.B. Multipolygonzüge

4.4.2 Webservices als Integrationstechnologie für Mehrwertdienste

Die Funktion, die das Web für die Interaktion zwischen Usern und Programmen hat, haben Webservices für Interaktionen zwischen Programmen. Webservices erlauben Lösungen schneller zu verbreiten und bieten weitergehende Anwendungsmöglichkeiten, vor allem im e-Business. Der Schlüssel für den erweiterten Horizont besteht in der Verwendung eines gewöhnlichen Kommunikationsmodells zwischen Programmen, das bestehende Standards wie HTTP, Extensible Marup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) und Universal Description Discovery and Integration (UDDI) verwendet. Webservices erlauben Applikationen und Services einfacher zu integrieren. Vorstellbar ist, unter Verwendung der bestehenden Internetstandards einen Mehrwertdienst, wie z.B. ein Service für die Bildgenerierung, in die Nexus-Architektur einzubinden. Der Service ermöglicht nicht nur auf der einen Seite die Einbindung von 3D Daten in die Architektur, sondern erlaubt auch auf der anderen Seite auf diese zuzugreifen. Die Integration erfolgt auf einem höheren Level des Protokollstacks. Die Basis der Dienste bilden Nachrichten, die z.B. Aufruf und Ergebnis der Dienste enthalten. Der Schwerpunkt liegt dabei mehr auf der Dienstsemantik als auf der Semantik des Netzwerkprotokolls. Eine lose Integration von Funktionen wird so ermöglicht. Webservices stellen eine Technologie

dar für das Deployment der Services. Zusätzlich werden Zugriffe über das Web auf die Services bereitgestellt. Unterstützende Arbeit bei der Implementierung von Webservices leisten Standards.

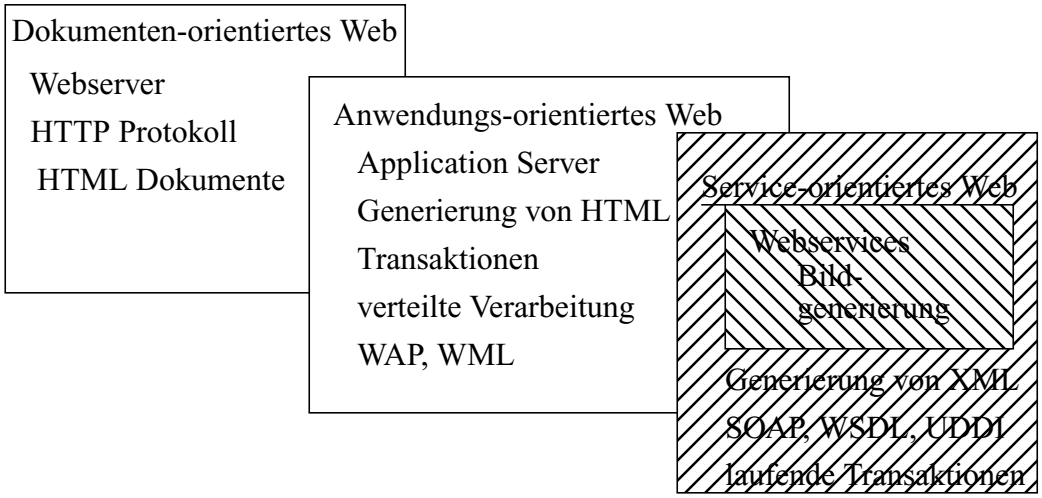

Abb.4.2: Zusammenhang von Dokumenten-, Anwendungs-, Serviceorientiertem Web in Anlehnung an ["dynamic e-business", GI-Regionalgruppe Hannover, 17 Januar 2001]

4.5 Wrapper

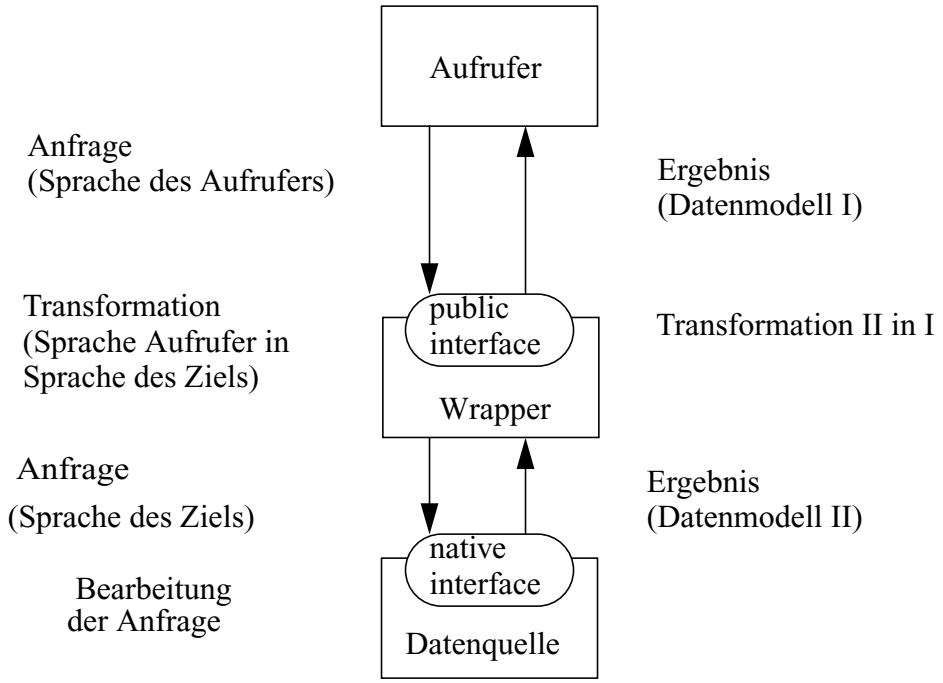

Abb.4.3: Allgemeine Darstellung eines Wrapers

Wrapper bieten die Möglichkeit, Daten anderen Ursprungs in ein System zu integrieren, ohne die Daten zu verändern. Ein Wrapper ist eine Komponente, die zwischen Aufrufer und Datenquelle sitzt. Der Wrapper nimmt den Request in der Sprache des Aufrufers entgegen, übersetzt diesen in eine entsprechende Anfrage, die die Sprache der Datenquelle ist. Anschließend wird der Request an die Datenquelle geschickt. Die Datenquelle bearbeitet den Request und gibt den Reply im eigenen Datenmodell zurück. Der Wrapper nimmt die Daten entgegen, übersetzt diese in das Datenmodell des Aufrufers, das für diesen interpretierbar ist und schickt die Daten als Ergebnis der Anfrage an den Aufrufer zurück. Mit Hilfe eines Wrappers kann auch die Sprache oder das Datenmodell bereichert werden. Dazu gehören auch Koordinaten für die Darstellung von 3D Daten. Ferner können Wrapper Funktionalitäten kompensieren, die fehlen. Die Umwandlung aus AWML, AWQL in die entsprechenden Formate für GIS ist durch Wrapper realisierbar. Es müssen die Daten vorher nicht entsprechend aufbereitet werden. Dies ist von großem Nutzen, da 3D Daten in der Regel eine große Datenmenge aufweisen.

Nachfolgende Abbildung soll verdeutlichen, wie der Wrapper zur Einbindung von 3D Daten in die Nexus-Architektur aussehen könnte.

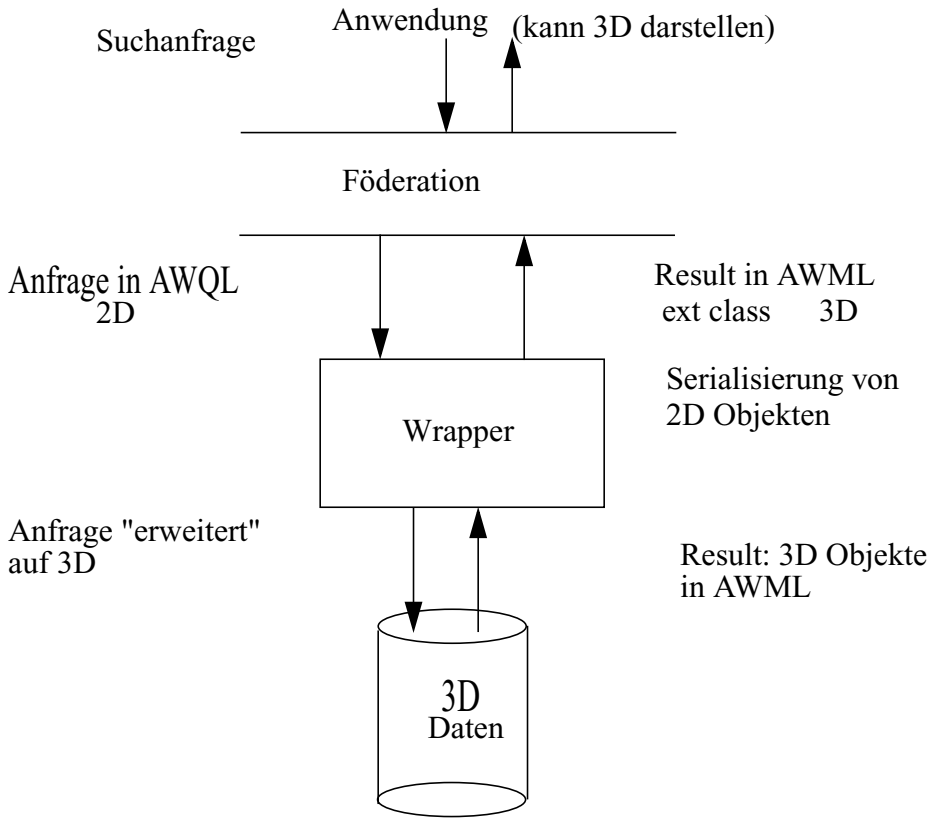

Abb.4.4: Realisierung des Wrappers in Nexus

Die Realisierung der Einbindung von 3D Daten durch einen Wrapper bietet mehrere Vorteile. Ortsbezogene Anwendungen werden durch den Wrapper unterstützt, da neue 3D Datenquellen leichter zu integrieren und zu ändern sind. Die Auswahl bestimmter Objekte mit bestimmten Attributen führt zu einer Datenreduktion. Der Wrapper fungiert als Brücke zwischen den Daten, die bereits dreidimensional vorliegen und den Anwendungen, die dreidimensional darstellen können. 3D Objekte sind über 2D Daten serialisierbar. Auf diese Weise können 3D Daten in die Nexus-Architektur integriert werden.

4.6 Zusammenfassung

Eine Erweiterung auf der Ebene der Sprache und des Datenmodells führt noch nicht zu einer vollständigen Beseitigung aller Probleme. Die Ergänzung der Funktionalitäten von 2D auf 3D kann prinzipiell über Wrapper und Mehrwertdienst erzielt werden. Nachfolgende Tabelle stellt Eigenschaften, Vor- und Nachteile einander gegenüber.

	Wrapper	Mehrwertdienst
Eigen-schaften	Erweiterung der Funktionalität Standalone	Erweiterung der Funktionalität Zusatzfunktion
Vorteile	+ bestehende Anwendung kann 3D mitbenutzen, ohne dies können zu müssen + AWM wird nicht verlassen + keine umfangreichen Änderungen auf Anwendungsseite + nur bei ASR einzutragen als eigener SpaSe verkäuflich	+ eigene Funktionalität innerhalb der Föderation nutzbar + Zugriff auf Föderation ermöglicht Integration von 2D Daten + weg von AWQL + andere, umfangreichere queries alle definierten Objekte der augmented world nutzbar + über MWD ist auch Zugriff auf externe Daten möglich
Nachteile	- alle Anfragen müssen in AWQL / AWML formulierbar / darstellbar sein - Föderation bleibt "Engpaß" - Vergrößerung der Architektur um eine Schicht	- Zusatzdienst (MWD) verursacht komplexere Verwaltung - Schnittstellenanpassungen notwendig - MWD stellt Dienst nur für spezielle Anwendung

Abb.4.5: Gegenüberstellung von Wrapper und Mehrwertdienst

Datenbanksysteme (DBS) oder Geoinformationssysteme (GIS) lassen sich erweitern, um räumliche Daten zu verwalten. Die Einbindung bestehender Systeme zur Speicherung von 3D Daten ist prinzipiell möglich, jedoch bleibt das Problem, daß in der offenen Nexus-Architektur AWML und AWQL sowohl von DBS als auch von GIS nicht standardmäßig unterstützt wird, bestehen. Wrapper können AWML und AWQL in das GIS- oder DBS-Format umwandeln. Für den Wrapper spricht auch, daß weder die Daten noch deren Format an der Datenquelle verändert werden müssen. Der Mehrwertdienst zeichnet sich vor allem dadurch aus, daß er über die Föderation auf bisherige 2D Daten und nun mit der Erweiterung auf 3D auch auf diese zugreifen kann. Vor dem Hintergrund der konkreten Anwendung ist die Möglichkeit, 3D Daten über einen Mehrwertdienst einzubinden, zu favorisieren. Erfolgt die Integration des Mehrwertdienstes als Webservice, so kann von folgenden Eigenschaften, die Webservices bieten, weiter profitiert werden: Webservices sind unabhängig von Objektmodell, Programmiersprache, Betriebssystem und Plattform. Der Service für die Bildgenerierung ist in C++ implemetiert und kann unabhängig von der Implemetierung anderer Services in die offene Architektur von Nexus integriert werden. Die Kommunikation beruht auf offenen Internet Standards. Webservices können dynamisch publiziert und gefunden werden. Mehrere Services können zu höherwertigen und komplexeren Services aufgebaut werden, die auch umfangreichere Dienste anbieten. Der Dienst für die Bildgener-

ierung könnte mit weitergehenden Funktionen versehen werden und Teil eines umfangreicheren Services sein. Die Integration z.B. in einen umfangreicheren Service zur Orientierung (s. Use-Cases) innerhalb einer bestimmten räumlichen Umgebung erscheint durchaus sinnvoll. Denkbar sind auch weitergehende Funktionen für die Bildbearbeitung oder Bildmanipulation.

5 Die Bildgenerierung als Webservice

Das vorangegangene Kaptel 4 hat dazu gedient, die Lösungsansätze darzustellen. Dieses Kapitel beschreibt den Webservice Bildgenerierung. Dazu gehören eine Defintion, die Funktionsweise der Bildgenerierung und die Integration in die Nexus-Plattform.

5.1 Definition des Webservice

Ein Webservice ist ein Interface, das eine Menge Operationen beschreibt, die mittels standardisierten Techniken über das Netzwerk erreichbar sind [Kreger]. Ein Web service wird durch ein Standard z.B. in formaler XML Notation beschrieben. Dies wird als Beschreibung des Service (service description) bezeichnet. Alle Details, die notwendig sind, um mit dem Service zu interagieren werden darin dokumentiert. Dazu gehören die Datenformate, Transportprotokolle und der Ort. Das Interface verbirgt Implementierungsdetails des Service. Dieser kann unabhängig von Software-, Hardwareplattform und der Implemetierungssprache benutzt werden. Eine lose Verknüpfung von Anwendungen, die auf Webservices basieren, wird auf diese Weise ermöglicht. Ferner sind Implementierungen der Services auch kom-ponentenweise und mit verschiedenen Technologien machbar. Webservices erfüllen eine oder mehrere Aufgaben.

Der Webservice Bildgenerierung hat die Aufgabe, ein Bild zu generieren und das Ergebnis zurückzugeben. Auslöser für die Generierung ist ein Request eines Clients, der die Anfrageparameter übergibt und das Ergebnis entgegennimmt.

5.2 Funktionsweise der Bildgenerierung

Die Bildgenerierung ist ein externer Dienst, die der Erstellung von Bildern dient. Sie bekommt eine Anfrage, die lokale Parameter der Umgebung enthält. Die Parameter dienen der Bildbeschreibung. Um ein Bild erstellen zu können, werden die Position, die Blickrichtung und die Bildparameter (Format, Auflösung, Farbe) benötigt. Die Koordinaten müssen als Gauß-Krüger Koordinaten vorliegen, ist dies nicht der Fall, ist eine Umrechnung notwendig. Wird die Bildgenerierung in NEXUS benutzt, ist eine Koordinatentransformation erforderlich, da in NEXUS die Position in World Geodetic System 1984 (WGS84) erfolgt.

Die Abbildung 5.0 zeigt die Zusamenhänge. Auf der einen Seite gibt es die Nexus-Architektur, in deren Rahmen ein Benutzer den Dienst zur Bildgenerierung verwenden möchte. Der externe Bilddienst ist nicht Teil der Nexus-Architektur. Im nachfolgenden Abschnitt werden die Möglichkeiten der Integration ausführlich erläutert.

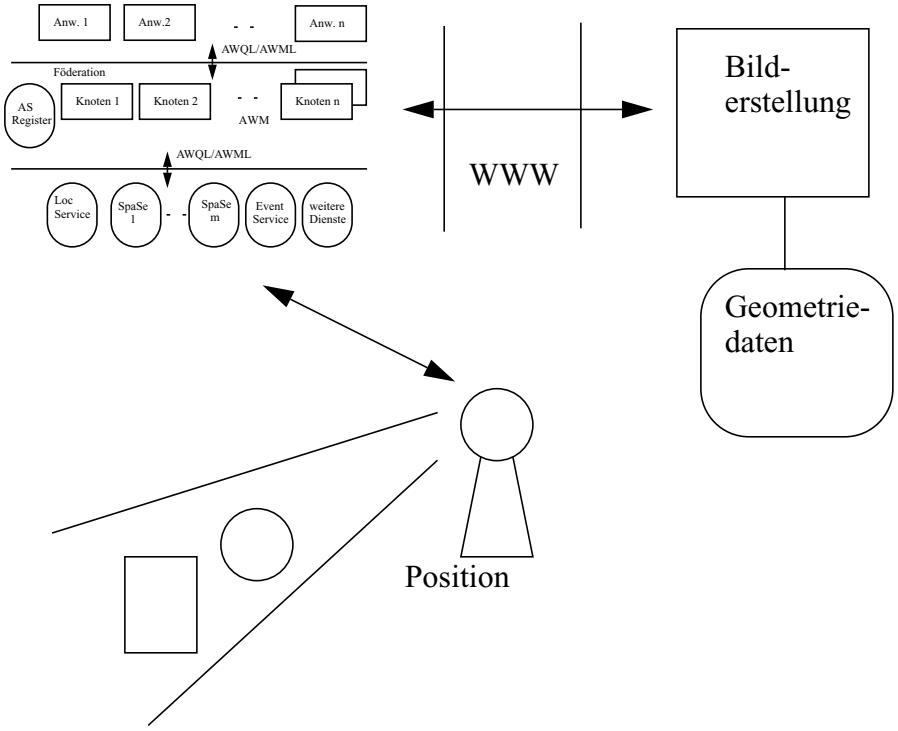

Abb.5.0: Zusammenhang von Nexus-Architektur und der Bildgenerierung

Die Daten der Position werden in einer Anfrage dem Dienst zugesendet. Dieser generiert unter Verwendung von geometrischen Strukturen und der Positionsbeschreibung ein Bild (s. Abb. 5.1). Dieses Bild basiert auf Gauß-Krüger Koordinaten (Breite: 3513268,942; Länge: 54044521,946; Höhe: 295,464) in Metern bezogen auf die Stadt Stuttgart. Diese Psition beschreibt den Blick in Richtung Stiftskirche. Man erkennt desweiteren im oberen rechten Drittel des Bildes die Königsstraße, die durch den Schloßplatz verläuft und die beiden Häuserzeilen teilt.

Abb.5.1: Ergebnis einer Anfrage [berechnet vom ifp]

5.3 Anforderungen an den Webservice Bildgenerierung

In diesem Unterkapitel werden die Möglichkeiten diskutiert, die Bildgenerierung über das WWW erreichbar zu machen.

5.3.1 Erreichbarkeit des Services Bildgenerierung

Die Anwendung ist in C++ implementiert. C++ Anwendungen, die über das Internet erreichbar sein sollen sind in der Regel nicht auf triviale Weise bereitstellbar. Die Ursache liegt in den fehlenden reinen C++ Technologien, die dafür einsetzbar sind. Die Konsequenz ist für C++ Entwickler eine andere Sprache zu verwenden, die als Interface zwischen dem C++ Code und dem Internet verwendet wird. Generelle Möglichkeiten, C++ Code zu integrieren sind z.B Java Servlets kombiniert mit dem Java Native Interface (JNI) oder Webserver Adapter wie z. B. Apache Module. Der Nachteil dieser Methoden besteht in der Notwendigkeit von anderen Sprachen oder langsameren Technologien wie CGI oder Java.

5.3.2 CGI

Die Verwendung von CGI bedeutet, für jede neue Anfrage an den Service für die Bildgenerierung einen neuen Prozeß starten zu müssen. Die Folge ist eine intensive Verwendung bestehender Ressourcen. Eine effiziente Zustandsspeicherung der CGI Programme wird verhindert. Eine Anfrage kann weitere verschiedene

CGI Prozesse auslösen. Multipliziert man die unterschiedlichen Pro-zesse mit der Anzahl der gleichzeitig möglichen, ist offensichtlich, daß die Performanz des Webservers beeinträchtigt wird.

5.3.3 JNI

Eine andere Methode könnten Java Servlets mit dem Java Native Interface darstellen, um Zugang zum C++ Code zu erhalten. Dies bedingt jedoch die Verwendung zweier Sprachen. Die Folge ist ein Performanzverlust, da eine weitere Schicht notwendig ist. Abbildung 5.2 verdeutlicht den Ansatz.

5.3.4 C++ Servlets

Der C++ Servlet Container hält ein Srevlet im Speicher. Alle Anfragen verwenden dieses eine Servlet. Der einzige Prozeß der gestartet wird, ist der Containerprozeß. Wird ein Servlet aktiv im Speicher gehalten, erlaubt der C++ Servlet Container den Code mit den Variablenbelegung während der Ausführung fest zu spezifizieren. Die Folge ist eine bessere Ausnutzung der Ressourcen eines Webservers. Im Vergleich dazu würde CGI bei jeder Programmausführung den Zustand reinitialisieren.

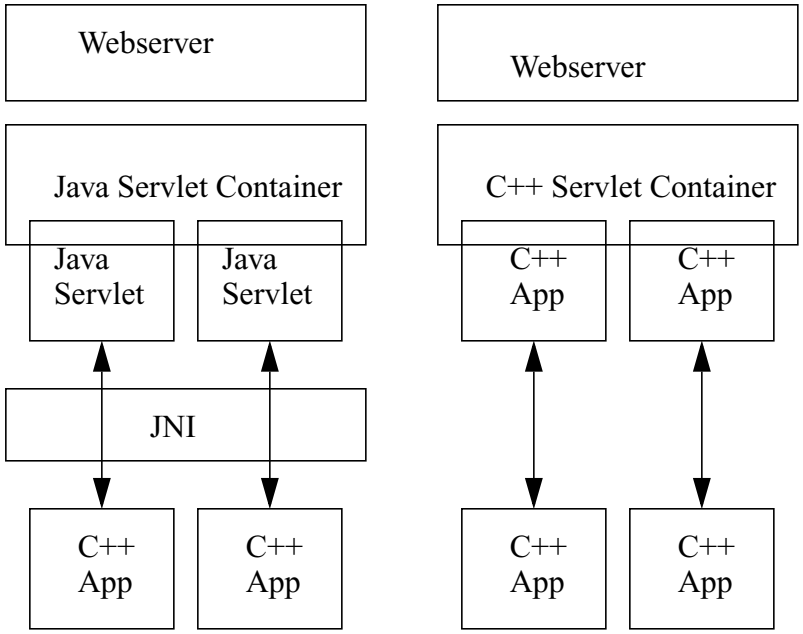

Abb.5.2: Ansatz unter Verwendung von JNI (links) und reiner C++-Ansatz (rechts) [Hinke]

Die Abbildung zeigt eine Gegenüberstellung zwischen dem JNI-Ansatz und dem reinen C++-Ansatz. Das Interface und der Verlust der Performanz fällt weg. Der C++ Container bietet eine effiziente, skalierbare und sehr portable Servlet API. Das Sprachdilemma, indem der Entwickler sich befindet, kann auf diese Weise umgangen werden. Die Integration von C++ Anwendungen mit dem Web ist ohne CGI machbar. Hinzu kommt, daß die Performanz erhöht wird, da die C++ Anwendung bereits kompiliert ist. Allerdings ist der reine C++ Ansatz nur sinnvoll, wenn Seiten, Formulare und Dokumente in das Web gehängt werden sollen. Für die komplexe und interaktive Anwendung der Bildgenerierung ist der Ansatz nicht sinnvoll, da eine umfangreichere Interaktion zugrunde liegt. Die Möglichkeit, die Anwendung über Webserver Adapter, wie z.B. ein Apache Module erreichbar zu machen, bleibt bestehen.

5.4 Aufruf des Webservice Bildgenerierung

Erreichbarkeit ist nicht das einzige Kriterium für den Webservice. Die Bildgenerierung soll mit einer Anfrage über das Netz aufgerufen werden können. Dazu muß ein Serviceaufruf vom User über das Internet gewährleistet werden. Netzwerkprotokolle werden verwendet, die allgemein "deployed" sind. Der de facto Standard für Services, die über das Internet erreichbar sein sollen, ist HTTP (s. Abb. 5.3). HTTP wird auch im Rahmen dieser Arbeit verwendet. Andere Internetprotokolle, SMTP und FTP eingeschlossen, spielen für diese Anwendungen eine untergeordnete Rolle.

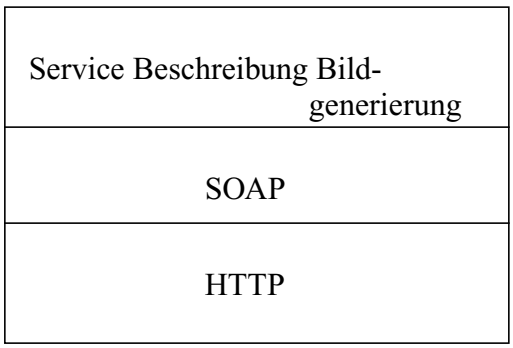

Abb.5.3: Protokollschichten für Webservices

Die Anfragen an die Bildgenerierung enthalten Daten, die XML-Formate aufweisen. SOAP wird für die Übertragung XML-basierter Nachrichten aus mehreren Gründen verwendet:

- SOAP ist ein einfacher XML-basierter Mechanismus für den Austausch von strukturierten Informationen zwischen Netzwerkanwendungen. SOAP enthält drei Teile: Zum einen eine Hülle, die den Rahmen definiert, um den Inhalt der Nachricht zu beschreiben. Des weiteren Regeln, um Instanzen der Datentypen aus der Anwendung auszudrücken. Der dritte Teil ist eine Vereinbarung, um Remote Procedure Calls (RPCs) und die dazugehörigen Antworten zu repräsentieren. SOAP kann auch mit anderen Netzwerkprotokollen außer HTTP verwendet werden. Dazu gehören SMTP, FTP, RMI über IIOP.

- SOAP ist ein standardisierter Mechanismus sowohl zum Ein- und Entpacken von Dokumenten als auch für den RPC im XML-Format.

- SOAP ist einfach, besteht aus einem HTTP POST mit einem XML Umschlag als Inhalt.

- Ein Standardmechanismus wird für den SOAP Kopf angeboten.

- SOAP Nachrichten unterstützen die Webservice Architektur durch Finde-, Binde-, und Operationen zum Publizieren.

Die Schicht der Servicebeschreibung enthält die Dokumente der Beschreibung des Services. Der de facto Standard für die XML-basierte Beschreibung ist WSDL (Web Service Description Language). WSDL definiert das Interface und den Mechanismus der Serviceinteraktion. Dies ist die minimale Standard Servicebeschreibung, die notwendig ist, um Operabilität zwischen Webservices zu unterstützen.

Ein Webservice wird definiert, indem er über das Netzwerk via SOAP zu erreichen ist und durch die Servicebeschreibung repräsentiert wird. Dies gilt auch für den Service der Bildgenerierung. Die drei Schichten, die in Abbildung 5.3 dargestellt sind, werden benötigt, um die Bildgenerierung als Webservice anbieten und verwenden zu könnnen. Abbildung 5.3 zeigt zugleich den minimalsten Aufbau. Enthalten sind die drei Schichten: HTTP (Netzwerkschicht) , SOAP (Schicht für XML-Übertragung) und die Servicebeschreibung (oberste Schicht).

5.5 Ablauf einer Anfrage an die Bildgenerierung

Grundlegende Voraussetzung für den Service Bildgenerierung und die Serviceanfrage ist die Fähigkeit SOAP-Nachrichten erstellen und parsen zu können. Ferner muß eine Kommunikation über ein Netzwerk möglich sein. Bestandteil der Kommunikation sind sowohl Empfangs- als auch Sende-Nachrichten.

Die eben genannten Funktionen werden oft von einem SOAP Server, der in einem Web Applikations Server läuft, ausgeführt. Alternativ dazu kann eine sprachspezifische Laufzeit-Bibliothek verwendet werden, die die Funktionen in einer API kapselt. Abbildung 5.4 zeigt die Integration mit SOAP anhand der Anfrage an den Service Bildgenerierung und die Antwort.

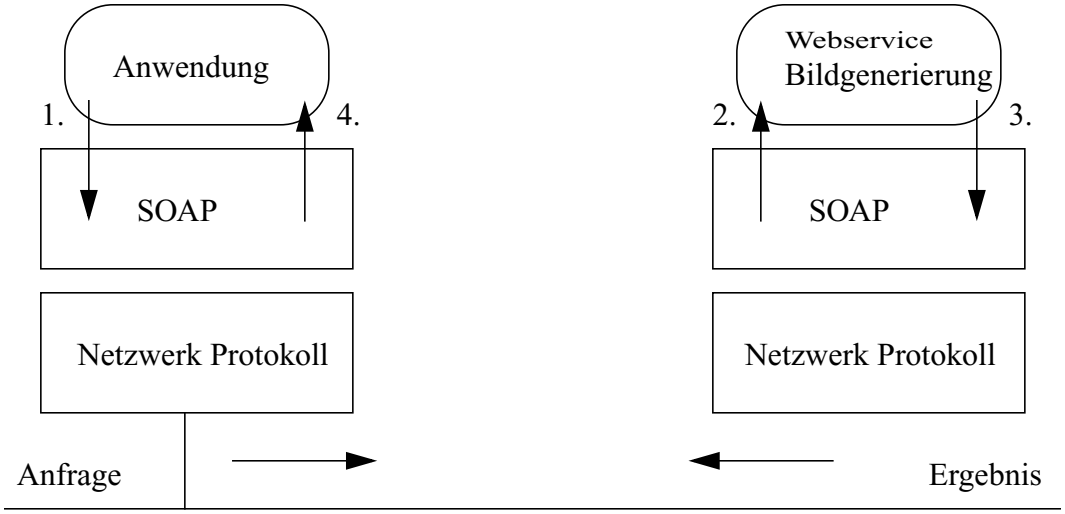

Abb.5.4: Anfrage an die Bildgenerierung

1. Die Anwendung z.B. eines PDAs in Nexus, erstellt eine SOAP-Nachricht. Die SOAP-Nachricht ist die Anfrage, die den Webservice Bildgenerierung mit einem Vorgang aufruft, den der Serviceanbieter ermöglicht. Der Vorgang "Bild erstellen" übernimmt die mit der in der Anfrage übergebenen Parameter im XML-Format und beginnt die Aufgabe abzuarbeiten. Die Anfrage wird zuammen mit der Netzwerkadresse des Serviceanbieters der Bildgenerierung an die SOAP Infrastruktur (z.B. ein SOAP-Client zur Laufzeit) übergeben. Die darunterliegende Netzwerk-schicht (HTTP) übernimmt die Nachricht, um sie über das Netzwerk an den Adressat zu übergeben.

2. Die Netzwerkinfrastruktur stellt die Nachricht der SOAP-Laufzeitumgebung zu (z.B. ein SOAP Server). Die Anfrage wird an den Webservice Bildgenerierung weitergeleitet. Die SOAP-Lauf-zeitumgebung ist verantwortlich für die Konvertierung der XML Nachricht in sparchabhängige Objekte, falls dies die Anwendung benötigt. Kontrolliert wird die Konvertierung über Schemas, die Teil der Nachricht sind.

3. Der Webservice Bildgenerierung ist verantwortlich für die Bearbeitung der Anfrage und Formulierung der Antwort. Die Antwort ist wie die Anfrage eine SOAP-Nachricht. Sie wird an die SOAP-Laufzeitumgebung mit der Adresse des Anfrage-Auslösers überstellt. Die SOAP-Laufzeit-umgebung sendet die Antwort zurück.

4. Die Antwort erhält zunächst die Netzwerkinfrastruktur des Auslöser der Anfrage. Sie wird an die SOAP-Infrastruktur weitergeleitet. Eventuell wird dabei die XML Nachricht in Objekte der Zielsprache konvertiert. Anschließend wird die Nachricht an die Anwendung zurüchgegeben, die den Webservice Bildgenerierung aufgerufen hat.

5.6 Verarbeitung der Anfrage

Der Service Bildgenerierung übernimmt die Parameter für die Verarbeitung aus der Anfrage, um mit den aktuellen Werten ein Bild zu generieren und das Ergebnis zurückzugeben. Dazu ist eine Schnittstelle notwendig. Sie dient der Formatbeschreibung, in der die Parameter übergeben werden. Für die Erstellung eines Bildes von einer bestimmten Position in einem konkreten Umfeld, wie z.B. in einer Stadt, sind die folgenden Parameter fundamental. Es muß der Blickwinkel, die Position und das Bildformat dem Service in der Anfrage überstellt werden. Der Blickwinkel definiert und begrenzt das Sichtfeld, das die visuell erfaßbaren Objekte enthält. Die Position lokalisiert den Ort im Kontext der Umgebung, der mit dem Bild näher beschrieben werden soll. Das Bildformat spezifiziert das Format (jpeg, gif) des Ergebnisses.

6 Entwurf

Im Kapitel 5 ist der Mehrwertdienst Bildgenerierung ausführlich beschrieben worden. Der Entwurf dient dazu, den favorisierten Lösungsansatz Mehrwertdienst weiter auszuarbeiten. Dazu sind eine Reihe von Entwurfsentscheidungen erforderlich, die ausschlaggebend für die weitere Umsetzung sind. Benötigt werden Entscheidungen hinsichtlich der Sprache, des Programmablaufs, der zugrundeliegenden Daten, des Ergebnisses, der Verarbeitung, der Kommunikation und der Architektur.

6.1 Entwurf der Datenstruktur

In diesem Abschnitt werden die Formate der Parameter entwickelt, die für die Besschreibung eines Bildes notwendig sind. Zuerst wird das Sichtfeld betrachtet. Anschließend werden die Parameter für die Position, die Blickrichtung und die für die Bildqualität (Auflösung, Format, Farbe) beschrieben.

6.1.1 Defintion des Sichtfeldes

Ein Bild zeigt einen Ausschnitt einer Umgebung. Es hat keine beliebig große Ausmaße, sondern weist Grenzen auf, die durch das Sichtfeld bestimmt wird. Der Inhalt zeigt alle Objekte, die gegenwärtig in einem bestimmten Rahmen, dem Sichtfeld, liegen. Der Ausschnitt und die Anzahl der Objekte werden durch die Entfernung, der Blickrichtung und der Position bestimmt.

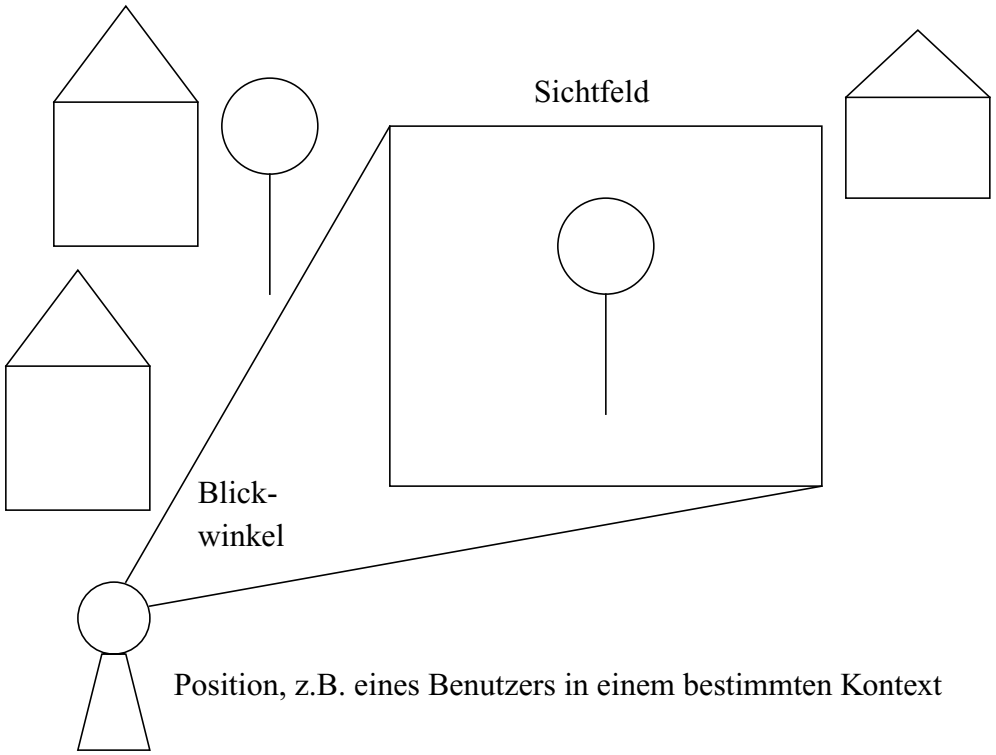

Abb.6.0: Sichtfeld in einer konkreten Umgebung

Die Entfernung, bzw. die Nähe, zu den Objekten bestimmt die Objektmenge. Ist die Position sehr nahe, verkleinert sich die Zahl der Objekte im Sichtfeld. Vergleichbar ist das mit dem Zoomvorgang eines Objektivs der Fotoausrüstung. Die Auswahl der Objekte kann mit der Blickrichtung geändert werden.

Blickrichtung

Die Blickrichtung kann durch 2 Wertetripple oder 2 Winkel beschrieben werden. Der Abstand zwischen den beiden Wertetripplen bildet einen Richtungsvektor. Es ist jedoch anzunehmen, daß sich die Benutzer unter der Eingabe von 2 Winkel für die Blickrichtung mehr vorstellen können, als unter 2 Wertetripplen. Auf Grund dieser Tatsache wird der Beschreibung der Blickrichtung durch Winkel gegenüber den Wertetripplen Vorzug gegeben. Es werden nur zwei Winkel benötigt. Der eine Winkel beschreibt die horizontale Richtung ("pitch"), der andere die vertikale Richtung ("head").

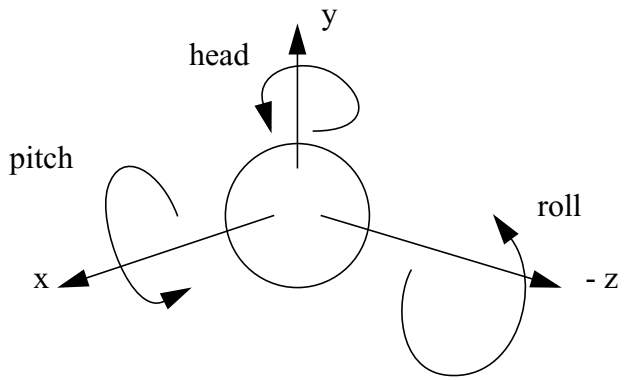

Abb.6.1: Orientierung im Raum nach Euler

Die vollständige Orientierung im Raum geht auf den Schweizer Mathematiker Leonard Euler (1707-1783) zurück und enthält neben den genannten Winkeln noch einen Rollwinkel ("roll") [Möller & Haines]. Im Sinne dieser Orientierung verläuft in der Grundposition die Blickrichtung entlang der negativen z-Achse während der Kopf entlang der y-Achse orientiert ist.

Der Rollwinkel ("roll") wird bei der Blickrichtung im Rahmen dieser Anwendung nicht benötigt, da eine seitliche Rollbewegung nach Untersuchung der Use-Cases hier keine Verwendung findet. In keinem der Use-Cases muß der Benutzer eine seitliche Rollbewegung ausüben.

Die Winkel der Blickrichtung sind unabhängig vom Koordinatensystem und brauchen nicht konvertiert zu werden. Für den vertikalen Winkel muß entweder das bestehende System erweitert (DEGREE.java, Direction.java) oder es müssen 2 neue Klassen eingeführt werden. Die Erweiterung birgt auch die Gefahr, den bestehenden Programmablauf zu stören. Für einen störungsfreien Betrieb ist es sicherer, den vertikalen Winkel durch eine eigene Winkelklasse (-90˚ < Winkel > 90˚) und dazugehörige Directionexception (90˚ < > -90˚) zu realisieren.

Position

Die Position ist für das zu generierende Bild wichtig, da sie den Ort im Umfeld lokalisiert. Sie bildet die Basis für die Blickrichtung. Eine Orientierung im Raum ist ohne eine Positionsbestimmung nicht möglich.

Die Eingabe der Position erfolgt durch den Benutzer (direkt oder automatisiert). Sie besteht aus drei Werten (x,y,z) und beschreibt die Position im WGS84-Koordinatensystem. Im WGS84-Koordinatensystem entspricht x der Breite, y der Länge

und z der Höhe. Die Höhe kann auch weggelassen werden, falls der Benutzer diese nicht kennt. Es wird dann eine Defaulthöhe von 1,80 m angenommen, um das Wertetripple zu komplementieren.

Weitere Bildparameter

Position und Blickrichtung reichen noch nicht für die Generierung eines Bildes aus. Sie bestimmen den Inhalt, während Auflösung, Art der Grafik, und Farbe die Bildqualität charakterisieren. Dazu gehört die Auflösung, die Farb- bzw S/W-Wahl und die Art der Grafik

Auflösung

Die Auflösung bestimmt die Größe des Bildes und zugleich die Schärfe. Sie wird mit Bildbreite (width) und Bildhöhe (height) beschrieben. Vorerst ist ein Defaultwert, 640x480, vereinbart worden. Die Auflösung beeinflusst indirekt die Übertragungszeit. Je größer sie gewählt wird, desto größer ist das Bildformat. Die Konsequenz ist eine längere Zeit, die für die Übertragung vom Service zum Empfänger benötigt wird.

Art der Grafik

Dieser Parameter dient dazu, dem Benutzer die Gelegenheit zu geben, die Darstellung zu beeinflussen. Es kann zwischen jpeg und png gewählt werden. Der Defaultwert ist jpeg.

Farbe bzw Schwarz/Weiß

Die Wahl von Farbe bzw. Schwarz/Weiß charakterisiert das Aussehen des Bildes. Es wird davon ausgegangen, daß die Bilder in Farbe sind. Das bedeutet, der Defaultwert ist Farbe (gray = no).

6.1.2 Realisierung der Datenübergabe

Die Datenübergabe erfolgt über ein Schnittstelle. Die Paramater, bestehend aus der Position, Blickrichtung und den Bildparametern werden aus der Anfrage an den Bilddienst gereicht. Dazu werden sie durch einen Parser ausgelesen und in Variablen für die weitere Verarbeitung abgelegt.

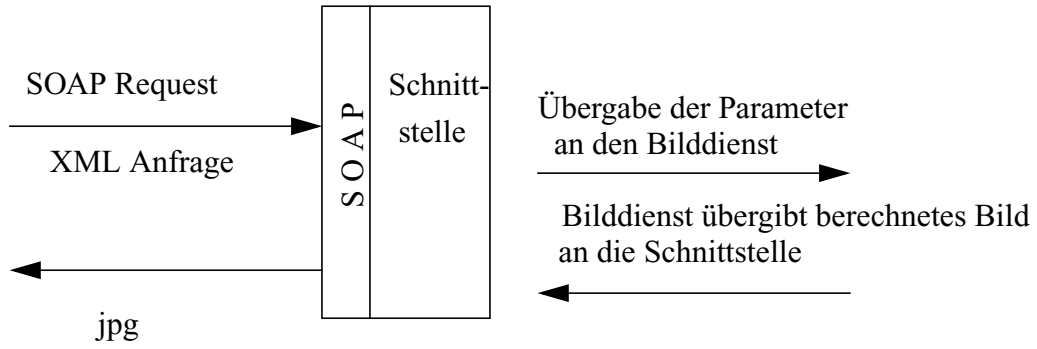

Abb.6.2: Schema der Schnittstelle

Die Abbildung 6.2 zeigt die Schnittstelle, die die Anfrage entgegenimmt, an den Bilddienst weiterreicht und das berechnete Bild von diesem wiederum zurücknimmt.

6.1.3 Datenformate

Die Daten liegen in einer bestimmten Form vor. Es handelt sich um das XML-Format. Das Format wird mit einer DTD überprüft, die auf die Anfrage spezialisiert ist.

```
<?xml version="1.0" encoding="US-ASCII"?>
<!ELEMENT 3DQueryFormat (imagespec, clipping?, angle, pos, objects)>
<!ELEMENT imagespec (imagesize, imageencoding)>
<!ELEMENT imagesize EMPTY>
<!ATTLIST imagesize
   width NMTOKEN #REQUIRED
   height NMTOKEN #REQUIRED>
<!ELEMENT imageencoding EMPTY>
<!ATTLIST imageencoding
   format (gif | jpeg) #REQUIRED
   gray (yes | no )                    "no">
<!ELEMENT  featurespec (#PCDATA)>
<!ATTLIST featurespec
   type NMTOKEN #REQUIRED>
<!ELEMENT clipping (nexusdata)>
```

```
<!ELEMENT nexusdata (wkt)>
<!ELEMENT wkt (#PCDATA)
<!ELEMENT angle EMPTY>
<!ATTLIST angle
   head  NMTOKEN #REQUIRED
   pitch  NMTOKEN #REQUIRED>
<!ELEMENT pos EMPTY>
<!ATTLIST pos
   x_pos NMTOKEN #REQUIRED
   y_pos NMTOKEN #REQUIRED
  z_pos NMTOKEN  #OPTIONAL>
<!ELEMENT objects EMPTY>
<!ATTLIST objects
   filter (((includes | includeallother)?, excludes?) |
              (includes?,( excludes | excludeallother)?))  #REQUIRED>
```

Diese DTD enthält die notwendigen Elemente und Attribute, um eine Anfrage an die Bildgenerierung zu stellen.

6.2 Einlesen der Daten

Die Daten für die Anfrage zur Erstellung eines Bildes liegen im XML-Format, z.B. als XML-File, vor. Das bedeutet auf der einen Seite, daß die Daten in einem standardisierten Format vorliegen, auf der anderen Seite heißt das, sie müssen zuerst geparst werden, um sie weiter zu verarbeiten. Ein XML-Parser übernimmt diese Aufgabe. Er liest aus dem XML-File die Elemente und Attribute aus und übergibt sie anschließend der Anwendung. Man kann die Parser in zwei Kategorien einteilen: validierend und nicht validierend.

Validierende Parser überprüfen, ob das Dokument in der richtigen Form (well-formed) vorliegt und prüfen das Dokument gegen die Data Type Definition (DTD).

Nichtvalidierende Parser prüfen das Dokument nur auf die richtige Form.

XML-Parser bieten die Möglichkeit, Daten im XML-Format einfach zu verarbeiten und können die Korrektheit der Formate prüfen. Es kann sowohl die XML-Konformität als auch die Form der XML-Daten geprüft werden. Die Form lässt sich mit Hilfe von XML-Schemas oder XML-DTDs definieren. Die wichtigsten standardisierten Schnittstellen für die XML-Verarbeitung sind das Document Object Model (DOM) und die Simple API für XML (SAX).

6.2.1 SAX

SAX [SAX] definiert Schnittstellen, um XML-Dokumente zu parsen und ist im Vergleich zum Document Object Model kein W3C-Standard. Das SAX-API stellt einen Mechanismus für eine serielle Verarbeitung von XML-Dokumenten bereit. Dokumente werden elementweise abgearbeitet.

6.2.2 DOM

Das Document Object Model [DOM] baut einen Baum auf, der die Struktur des XML-Dokuments repräsentiert. Das DOM-API liest den Baum aus oder ändert diesen. Im Gegensatz zu SAX ist von der World Wide Web Coalition (W3C) definiert.

6.2.3 Bewertung im Hinblick auf die Verwendung bei der Bildgenerierung

DOM baut immer zuerst einen Baum aus einem vorliegenden XML-Dokument. Im Vergleich zu SAX ist DOM langsamer, bietet dafür Operationen für die Manipulation im DOM-Baum. Gegenüber DOM liest der SAX-Parser sequentiell die Zeichen eines XML-Dokuments. SAX ist schnell, hat jedoch den Nachteil, daß kein Objekt zur Verfügung steht, auf das man nach dem Parsen zugreifen kann. Soll ein XML-Dokument nur einmal gelesen werden, ist SAX gegenüber DOM die bessere Wahl. Für die Bildgenerierung auf Basis von Daten im XML-Format reicht es aus, XML-Dokumente einmal sequentiell zu lesen und dabei die Elemente für die Verarbeitung auszulesen. Prinzipiell bieten sich sowohl SAX als auch DOM im Rahmen dieser Arbeit für eine Verwendung an, der SAX-Parser ist jedoch für diese Aufgabe dem DOM vorzuziehen.

6.2.4 Auswahl des Parsers

Das XML-File wird zuerst geparst, um die Parameter der Anwendung zur Verfügung zu stellen. Dafür wird SAX verwendet, denn es werden die Daten nur einmal gelesen. Der Zugriff auf die Daten erfolgt ebenfalls einmal. Ein weiterer Grund ist der Verzicht auf den Aubau eines Baumes, wie bei DOM, zu nennen.

6.3 Applikation

Die Anwendung wird nun kurz die Anwendung skizziert, um die unterschiedlichen Komponenten darzustellen, die an der Anwendung beteiligt sind.

Ablaufplan der Applikation (Skizziert):

Anwendung: Programmstart
Positionseingaben / Benutzereingaben (auf Korrektheit prüfen)
WKSPunkt erstellen (mit oder ohne Höhe)

Mehrwertdienst: Prüfen, ob WGS84- oder bereits GK-Koordinaten (keine Trans)
Test ob Anfrage schon gestellt (Hashfkt und Versionsnummer)
Aufruf der Bildberechnung (remote)

Bilddienst generiert mit den an ihn übergeben Daten das Bild.

Mehrwertdienst: Empfang des Ergebnisses

Anwendung: Programmende / Neue Position

Die Anwendung startet mit der Eingabe von Daten. Sie werden als XML-File vorgegeben. Mit den Positionsangaben wird ein WKSPunkt erstellt. Der Mehrwertdienst prüft, ob die Eingabe bereits als Gauß-Krüger Koordinaten vorliegen, falls dies nicht der Fall ist, muß nach Gauß-Krüger konvertiert werden. Wurde die Anfrage nicht schon einmal gestellt, wird nun der Bilddienst aufgerufen, der mit den Daten der Anfrage ein Bild generiert. Der Mehrwertdienst bekommt das Ergebnis zurück. Die Anwendung beendet das Programm.

6.3.1 Features

Ist eine Anfrage mit gleicher Position und gleichem Blickwinkel schon früher gestellt worden, soll diesselbe Anfrage nicht zu einer neuen Bildberechnung führen, sondern dann auf das im Cache abgelegte Bild zugreifen. Dazu muß im Cache das Bild über einen vorher bestimmten Zeitraum gehalten werden. Der Vergleich zwischen der neuen und den alten Anfragen kann mittels eine Hashfunktion z.B. über die Anfrageparameter und der Versionsnummer der Anfrage realisiert werden.

Datenreduktion führt auf jeden Fall zu einer Reduktion der zu übertragenden Volumina. Statt des Bildes in jpeg- oder gif-Format würde es auch genügen im Ergebnis nur die URL als String zu schicken. Die URL teilt dem Benutzer den Speicherort wo das Bild zu finden ist. Allerdings muß dafür Speicher bereitgestellt werden.

6.4 Realisierung der Koordinatentransformation

Die Bildberechnung erfolgt auf Basis von Gauß-Krüger-Koordinaten (derzeit Gauß-Krüger-Koordinaten im 3˚-Streifen-System, Datum Potsdam, bezogen auf das Bessel-Ellipsoid in den alten Bundesländern bzw. das Krasovsky-Ellipsoid in den neuen Bundesländern). Da die Positionseingabe und Positionsbestimmung auf Grundlage des World Geodetic System 1984 (WGS84) erfolgt, ist eine Transformation von WGS84 nach Gauß-Krüger nötig [Gendt, Oheim, Gehrke]. Diese kann direkt nach der Positionsbestimmung durchgeführt werden. Oft werden nur Transformationsparametersätze auf der Ebene eines Bundeslandes (z.B. für Branden-

burg bei WGEO von WASY) oder wahlweise für das Bessel– bzw. Krasovsky-Ellipsoid verwendet (Transdat, EasyTrans, ERDAS). Abweichungen der berechneten Gauß-Krüger-Koordinaten im Zentimeterbereich sind bereits zu erwarten, wenn Transformationsparameter eines 20-30 km entfernten Gebietes verwendet werden. Vor allem die Höhengenauigkeit nimmt stark ab, wenn keine lokal berechneten Transformationsparameter zur Verfügung stehen.

Es soll in der Anwendung Bildgenerierung nicht die lokal auf Stuttgart angepaßte Koordinatentransformation eingsetzt werden, um eine größere Flexibilität zu erreichen. Die Konsequenz ist eine stärkere Ungenauigkeit. Das ifp hat ein berechnetes Bild und die dazugehörigen Gauß-Krüger-Koordinaten (3513268,942; 5404521,946; 295,464) in Metern zur Verfügung gestellt. Es sind nicht die dazugehörigen Koordinaten in WGS84 vorhanden. Eine Prüfung der Transformationsfunktion kann so nicht durchgeführt werden, da für keine Position sowohl Gauß-Krüger- als auch WGS84-Koordinaten existieren. Die Koordinaten können für eine Rücktransformation nach WGS84 verwendet werden, um dann mit diesem Ergebnis wieder die Koordinaten nach Gauß-Krüger zu transformieren. Es kann jedoch kein Zwischenergebnis auf Richtigkeit überprüft werden. Ferner ist bei fehlerhaften Werten mit einer Akkumulation der Fehler zu rechnen. Enthält das Transformationsergebnis Ungenauigkeiten, die immer eine konstante Verschiebung aufweisen, könnte eine nachträgliche Fehlerkorrektur mit einem Korrekturfaktor den Fehler bereinigen. Dies muß vor der Bildberechnung erfolgen, da sonst unter Umständen ein Bild auf Basis falscher Werte berechnet wird.

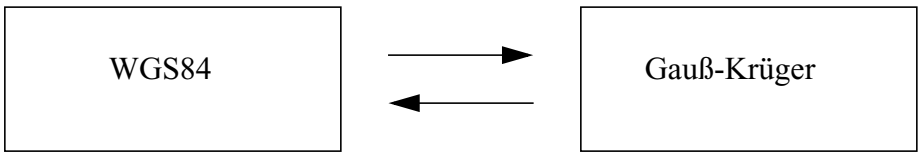

Abb.6.3: Direkte Koordinatentransformation

Die Abbildung 6.3 demonstriert eine direkte Koordinatentransformation. Beide Koordinatensysteme haben unterschiedliche Bezugssysteme. Die direkte Transformation ist so nicht möglich.

Das Ausgangskoordinatensystem ist WGS84. Es handelt sich dabei um ein kartesisches und erdmittelpunkt-zentriertes, rechtshändiges Koordinatensystem.

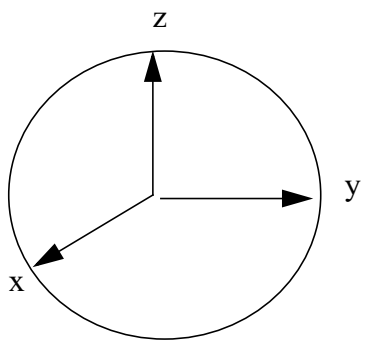

Abb.6.4: Skizze des WGS84

Die x-z-Ebene ist die Greenwhich-Meridian-Ebene, x der Schnittpunkt des Äquators mit dem Greenwhich-Meridian. Wird x um 90 Grad in östlicher Richtung um die z-Achse gedreht, ergibt sich y. Dieser Punkt liegt in der Äquator-Ebene vom Erdmittelpunkt zum Nordpol. Der z-Punkt wird durch die Senkrechte auf die beiden Ebenen gebildet.

Zielkoordinatensystem ist das Gauß-Krüger Koordinatensystem.

Das Gauß-Krüger Koordinatensystem (GK) wird über einem lokalem Bezugselipsoid definiert. Meistens verwendet man das Bessel-Ellipsoid (1841).

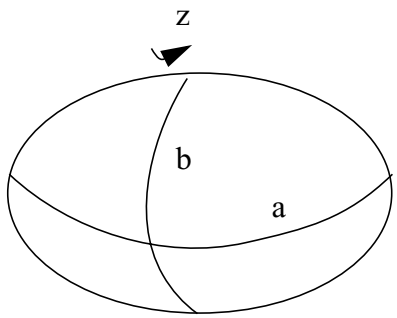

Abb.6.5: Skizze des Gauß-Ktüger Koordinatensystems

Das Koordinatensystem wird durch die große Halbachse a und die kleine Halbachse b beschrieben. Die große Halbachse verläuft vom Erdmittelpunkt zum Äquator, die kleine Halbachse vom Erdmittelpunkt zum Nordpol. Für Baden-Württemberg ist a = 63377397,155 in Metern und b = 1 geteilt durch f = 299,15281282. Gauß-Krüger ist ein ebenes Koordinatensystem und besitzt einen Rechtswert und einen Linkswert. Der Rechtswert setzt sich aus der Kennziffer des Bezugsmeridian und einer loklalen Verschiebung bezüglich des Bezugsmeridian, nach Ost in Metern (Addition) oder nach West in Metern (Subtraktion), zusammen. Auf diesen Wert wird anschließend noch die rechnerische Größe von 500 000 addiert. Die Kennziffer des Bezugsmeridian ist 3 und steht für das Ergebnis aus Bezugsmeridian dividiert durch 3. Der Meridian ist bei 30 Grad Einteilung für die (ehemalige) BRD-West 9 Grad. Der Hochwert besteht aus dem Weg vom Äquator nach Norden (Addition) oder Süden (Subtraktion) auf der Ellipse in Metern. Die Höhe kann als die Anzahl der Meter in Richtung Zenith definiert werden.

6.5 Konvertierung von WGS84 nach Gauß-Krüger

Für die Umrechnung benötigt man ein Bezugssytem mit geographischer Länge (Breite) und Höhe. Das Bezugsellipsoid (Bessel) bleibt gleich. Die geographosche Länge beschreibt eine Rotation um die z-Achse bis zur betreffenden Überlappung der Verbindungslinie Nordpol - gewünschter Punkt mit dem Greenwhich-Meridian. Die geographische Höhe ergibt sich aus dem Winkel der Rotation um die y-Achse (nach entsprechender Rotation um die z-Achse) bis zur Deckung des Koordinatenpunktes. Das geographische System ist wie das Gauß-Krüger Koordinatensystem ein ebenes Koordinatensystem. Die Höhe kann wie bei Gauß-Krüger als die Anzahl der Meter in Richtung Zenith betrachtet werden.

Zu bemerken ist, daß die Mittelpunkte sowie die Achsenorientierungen von WGS84 bzw dem lokalen Bezugselipsoid nicht exakt diesselben sind, sondern leicht verschieden sind. Das macht eine Translation erforderlich.

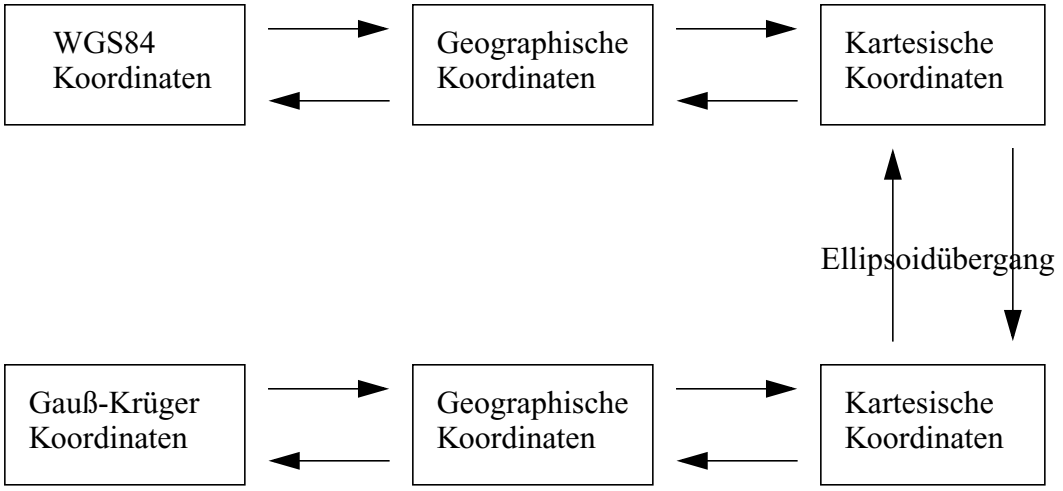

Abb.6.6: Schema der Koordiantentransformationen

Die Abbildung 6.5 zeigt den Zusammenhang der Koordinatentransformationen. Eine Konvertierung von WGS84 nach Gauß-Krüger verläuft in folgenden drei Schritten:

1. Schritt:

Transformation von WGS84 in die kartesische Darstellung bezüglich dem lokalen Rotationsellipsoid. Realisiert wird die Umformung durch eine Translation, verbunden mit einer Rotation und einer Skalierung.

2. Schritt:

Transformation der kartesischen Darstellung bezüglich lokalem Rotations-Ellipsoid in geographische Koordinaten.

3. Schritt:

Transformation geographischer Koordinaten nach Gauß-Krüger.

6.6 Realisierungsmöglichkeiten der Anwendung Bildgenerierung

Grundsätzlich könnte die Anwendung als Applikation, Applet oder als Bean realisiert werden. Beans benötigen eine umfangreiche Umgebung. Für diese einzelne Anwendung steht der damit verbundene Aufwand in keinem Verhältnis zum erzielten Gewinn. Applets bieten keine Möglichkeit Ergebnisse lokal zu speichern. Es könnte der Mechanismus der Signaturen dazu verwendet werden, Schreiben zu erlauben, allerdings ist der Mehraufwand nicht unerheblich. Für die Unterzeichnung kann das Programm javakey benutzt werden. Es verwaltet für den Benutzer eine Menge von bekannten öffentlichen und den eigenen privaten Schlüssel. Findet der appletviewer in der Menge der öffentlichen Schlüssel einen, mit dem ein Applet unterzeichnet ist, so hat dieses Applet die gleichen Rechte wie ein lokales Applet. Für die Unterzeichnung eines Applets sind folgenden Schritte notwendig: Identifizierung, Schlüsselerzeugung, Zertifizierung, Archiverzeugung und Unterzeichnung. Die Realisierung der Anwendung als Applikation bietet im direkten Vergleich zum Applet zwei Vorteile. Als eigenständig laufende Programme benötigen sie keine Container. Desweiteren haben sie weitergehende Möglichkeiten auf dem lokalen System, was Speichervorgänge vereinfacht.

Zusammenfassend läßt sich feststellen, Applikation, Applet und Beans eignen sich prinzipiell zur Realisierung. Stellt man jeweils Vor- und Nachteile gegenüber und zieht noch als weiteres Kriterium den Implementierungsaufwand hinzu, ist die Applikation gegenüber den anderen Möglichkeiten vorzuziehen.

6.7 Entwurf der Kommunikation

Die Bildberechnung soll nicht lokal erfolgen, sondern wird entfernt von einem Dienst ausgeführt. Dazu ist ein entfernter Methodenaufruf notwendig. In Frage kommen dafür prinzipiell Java/Rmi, SOAP RPC, Corba und Applet-Servlet. Der Applet-Servlet Mechanismus kommt aufgrund des damit verbundenen Aufwands nicht in Betracht. Der Dienst ist in C++ implementiert. Die Möglichkeiten werden mit dieser Einschränkung auf Corba, Java/RMI und SOAP RPC begrenzt.

6.7.1 Corba

Corba basiert auf dem Internet Inter-Orb Protokoll (IIOP) für entfernte Objekte. Alles in einer Corba-Architektur hängt von einem Object Request Broker (ORB) ab. Der ORB funktioniert als zentraler Objekt Bus, über die jedes Corba Objekt mit anderen Objekten lokal oder entfernt interagiert. Für eine Anfrage an einen

Service wird ein Corba Client benötigt, der eine Objektreferenz zu einem Corba Server-Objekt besitzt. Der Client kann Methoden aufrufen und über die Objektreferenz ausführen, als ob die Corba Server Objektreferenz im Adreßraum des Clients vorhanden wäre. Der ORB ist verantwortlich zum Auffinden des Service und die Vorbereitung der Kommunikation. Corba ist nur eine Spezifikation und kann damit auf verschiedenen Plattformen, wie z.B. UNIX, Windows, eingesetzt werden.

Der Aufbau einer Corba-Architektur bedingt gleichzeitig einen ORB, der eine zentrale Rolle spielt. Innerhalb des Nexus-Systems sind keine Möglichkeiten vorgesehen, eine derartige Architektur aufzusetzen. Corba ist im Rahmen dieser Arbeit damit nicht weiter von Interesse.

6.7.2 Java/RMI

Java/RMI baut auf einem Protolkoll auf, das Java Remote Method (RMI) genannt wird. Java ist stark mit der Serialisierung von Objekten verbunden. Dies ermöglicht, daß Objekte als Strom versendet werden können. Die Serialisierung der Objekte ist charakteristisch für Java. Das bedeutet jedoch, daß sowohl der Client als auch das Java/RMI Server-Objekt in Java implementiert werden müssen. Jedes Java/RMI Server-Objekt definiert ein Interface, um das Server-Objekt außerhalb der Java Virtual Machine (JVM) erreichbar zu machen. Ein Mechanismus für die Namensgebung wird benötigt, um Informationen der erreichbaren Server-Objekte bereitzustellen. Dieser wird als RMI Registry bezeichnet und läuft auf auf dem Server. Ein Java/RMI Client muß zuerst ein Lookup nach einer Objektreferenz ausführen, um auf diese zugreifen zu können. Anschließend kann die entfernte Methode auf dem Server-Objekt so aufgerufen werden, als ob sie im Adreßraum des Clients vorhanden wäre.

Java/RMI baut auf Java auf und kann auf verschieden Plattformen eingesetzt werden. Dazu gehören UNIX, Windos und PDAs. Es ist auch möglich über das Java Native Interface C Funktionen auszuführen. Diese haben ihren Ursprung nicht in Java. Funktionen, die in C++ implementiert sind, werden nicht unterstützt. Die Bildgenerierung ist in C++ geschrieben, kann also über Java/RMI nicht aufgerufen werden.

6.7.3 SOAP

Das Simple Object Access Protokoll (SOAP) dient der Etablierung eines einfachen, plattformübergreifenden und erweiterbaren Mechanismus für den Datentransport. Web-Standards, wie HTTP und XML werden unterstützt. SOAP-Nachrichten bestehen aus XML. Es können auf diese Weise strukturierte und typisierte Nachrichten ausgetauscht werden. Das Spektrum reicht von einfachen Nachrichtensystemen bis zu Remote Procedure Calls.

Unter den genannten Möglichkeiten stellt SOAP für diese Anwendung aus mehreren Gründen die bessere Alternative dar: Erstens werden Datenformate einfacher Art übertragen. Dazu gehören Strings für eine URL, oder Bilder im jpeg-Format. Zweitens wird C++ unterstützt. Zweitens wird die Kommunikation mit SOAP im Vergleich zu Corba nicht durch Firewalls beeinträchtigt. Zuletzt ist noch von Bedeutung, daß Services, wie die Bildgnerierung und SOAP miteinander kommunizieren können.

6.7.4 RPC via SOAP

Der Service Bildgenerierung ist ein eigenständiger, externer Service, der über das Netz auch im Nexus-System angeboten werden soll. Um das Serviceangebot in Nexus zu nutzen, ist ein entfernter Funktionsaufruf (RPC) notwendig.

Die Verwendung von SOAP bietet sich im Rahmen des entfernten Aufrufs an. Es wird neben einem standardisierten Weg für die Serilisation von Daten ein einfacher Mechanismus zur Verfügung gestellt, um auf entfernten Rechnern Funktionen aufzurufen [Apache SOAP 2001]. Dazu wurde eine einheitliche Darstellung von RPCs gewählt. Das Format ist für den Austausch und das Kapseln von RPCs im XML-Datenstrom notwendig. Existierende Standards, z.B. XML und HTTP wurden verwendet, um auf diesen ein erweiterbares, einheitliches Protokoll aufzusetzen.

6.7.5 Ablauf des SOAP-Aufrufs

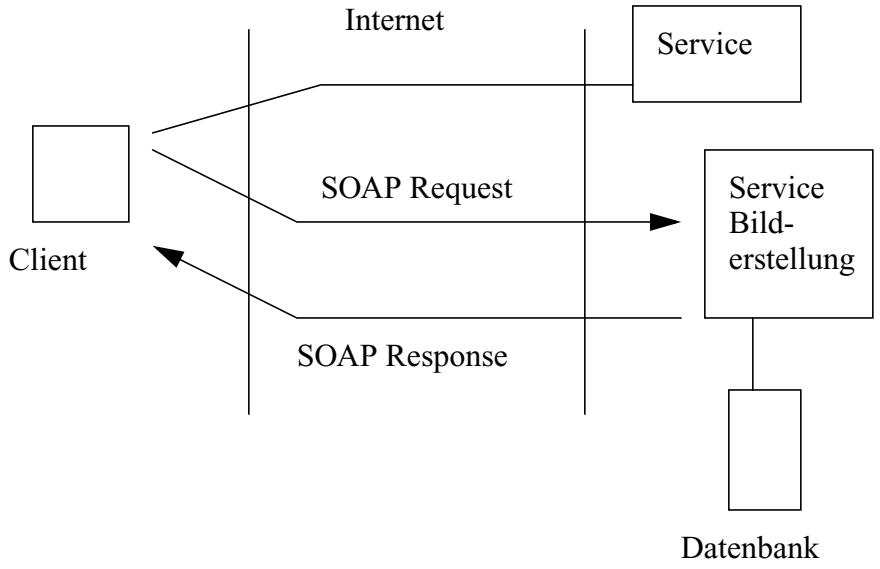

Abb.6.7: Darstellung des SOAP Aufrufs

SOAP-Request

Der Client innerhalb des Nexus-Systems ruft den Service Bildgenerierung auf, der auf einem entfernten Server vorhanden ist. Dabei werden in der Anfrage die aktuellen Prameterwerte an die Bildgenerierung übergeben. Für den Aufruf ist die Programmiersprache, in der die Bildgenerierung implementiert wurde, irrelevant.

Anfragebearbeitung

Die Anfrage trifft beim Service als SOAP-Nachricht ein. Der Webserver (Apache) prüft, ob eine Anfrage an die Bildgenerierung gestellt wurde. Der Funktion "Bild erstellen" werden beim Aufruf die Anfrageparameter Position, Blickrichtung sowie Bildparameter (Größe, Format, Farbe) übergeben und das zugehörige Objekt gestartet.

SOAP-Response

Die SOAP-Response enthält das Ergebnis der Verarbeitung. Der Client bekommt mit dieser Nachricht das Ergebnis zurückgesendet.

Schema Architekturbild

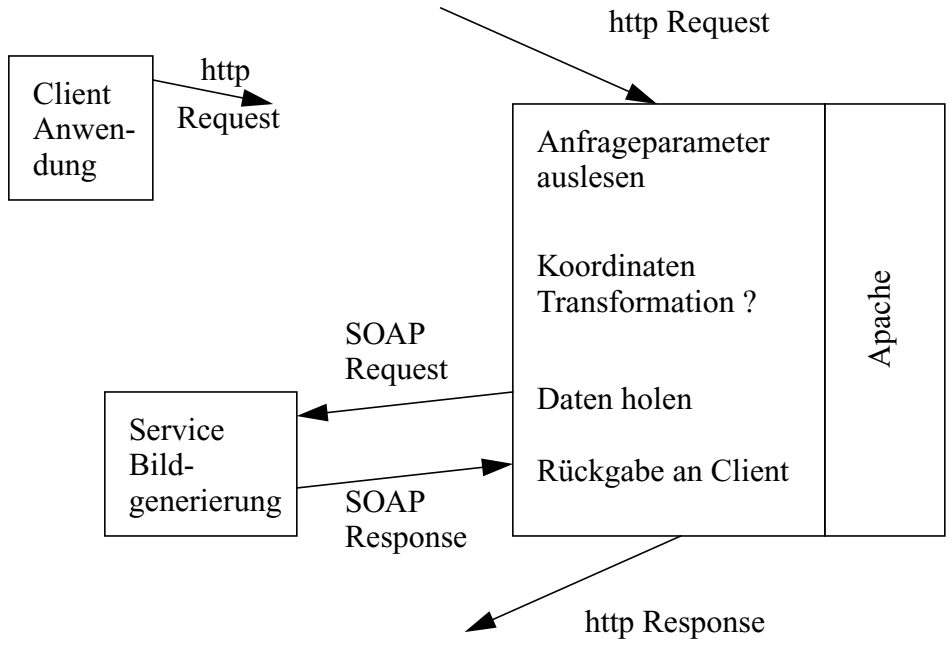

Abb.6.8: Schema zeigt den architektonischen Zusammenhang

Die Abbildung 6.8 zeigt den Client in Zusammenhang mit Apache als Webserver. Dieser horcht, ob ein neuer Request gestartet wurde. Ist dies der Fall, wird an die Bildgenerierung ein SOAP Request gesendet. Die Anfrageparameter werden an diesen übergeben Hat die Bildgenerierung die Erstellung des Bildes beendet, wird eine SOAP Response an den Webserver zurüchgegeben. Dieser gibt das Bild an die Anwendung zurück.

7 Zusammenfassung und Ausblick

In diesem Kapitel wird die Arbeit zusammengefasst und ein Ausblick auf künftige Entwicklungsmöglichkeiten gegeben.

7.1 Zusammenfassung

Das Ziel der vorliegenden Diplomarbeit bestand darin, dreidimensionale Modelldaten in die Nexus-Plattform zu integrieren. Dafür kamen Wrapper oder Mehrwertdienst in Betracht. Der Wrapper bietet die Möglichkeit, bestehende Systeme als Spatial Model Server anzusprechen. Durch den Mehrwertdienst können bestehende Systeme für die Speicherung von 3D-Daten eingebunden werden.

Im ersten Teil der Arbeit wurden zunächst die Anforderungen analysiert. Anschließend sind der Wrapper und der Mehrwertdienst als Möglichkeiten für die Einbindung dreidimensionaler Daten untersucht worden. Den Anforderungen wurden die Ansätze gegenübergestellt Das Ergebnis des Vergleichs bestand darin, einen Mehrwertdienst prototypisch umzusetzen.

Im Entwurf ist der Ablauf der Verarbeitung beschrieben worden, wie aus einer Anfrage an den externen Bilddienst ein generiertes Bild als Ergebnis zurückgesendet wird. Die dabei getroffenen Entscheidungen sind entsprechend dokumentiert worden.

Die Implementierung des Entwurfs weist folgende Merkmale auf: ein RPC via SOAP realisiert die Kommunikationsstruktur, Anfragen und Ergebnisse werden über das standardisierte Transportprotokoll SOAP zwischen Client und Dienst versendet, die Daten werden aus einem XML-File durch SAX ausgelesen, die Koordinatentransformation von WGS84 nach Gauß-Krüger wird durch die Transformationsfunktion realisiert, die Nexus anbietet.

7.2 Ausblick

Eine sich dynamisch ändernde Welt führt in vielen Bereichen zu immer neuen Anwendungsgebieten, um den aktuellen Anforderungen gerecht zu werden. Es entstehen Anwendungen, um komplexere Strukturen und Informationen zu verarbeiten. Die Folge besteht in umfangreicheren Applikationen und Diensten, um die Zusammenhänge der Informationen zu erfassen.

Die Untersuchung der Use-Cases hat die Notwendigkeit offengelegt, dreidimensionale Datenstrukturen einzubinden. Damit können neue Anwendungen und Dienste entwickelt und angeboten werden. Dienste, die den lokalen Gegebenheiten angepaßt sind, können spezielle Funktionen wahrnehmen. Viele spezielle Anwendungen können zu einer komplexeren aufgebaut werden, um komplizierte Sachverhalte zu bearbeiten.

Die Einbindung externer Daten und Anwendungen entlastet die Nexus-Plattform. Dies ist von Vorteil, da die Daten immer umfangreicher und vor allem vielfältiger werden. Die Wahl der Datenstruktur beeinflusst weiterhin die Verarbeitung. Eine weitere Entwicklungsstufe kann die Einbindung ganzer Portale sein. Mobil-Portale, Portale für mobile Geräte, stellt auch eine neue Anwendungsklasse für Nexus dar. Die Vielfältigkeit des Angebots wird gesteigert, da Portale mehrere eigene Anwendungen anbieten können.

Die Nexus-Plattform wird mit jedem weiteren Dienst und Datenart das Spektrum der Anwendungsmöglichkeiten ausweiten und das Potential erhöhen.

Die Bearbeitung der Aufgabe, hat meinen persönlicher Eindruck verstärkt, daß ortsbezogene Anwendungen und Dienste erst in einem Anfangsstadium sind. Die ersten Anwendungen in diesem Bereich finden bereits großen Anklang, z.B. bei der Navigation. Deshalb ist meines Erachtens auch hier ein Schwerpunkt zu setzen. Diese Dienste können mit anderen Diensten erweitert werden, um ein breiteres Anwendungsspektrum anzubieten. Spannend bleibt im Rahmen möglicher Anwendungen weiterhin der Konflikt zwischen der Wahrung der Privatsphäre auf der einen Seite und ortsbasierten Anwendungen auf der anderen Seite.

Literaturverzeichnis

[Apache SOAP 2001]
Dokumentation über Apache SOAP Version 2.0
HTTP://xml.apache.org/soap
Apache XML Project, June2 10, 2002

[DOM]
http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/
http://www.w3.org/TR/2000/REC-DOM-Level-2-Traversal-Range-20001113/

[Engelen & Gallivan]
http://www.cs.fsu.edu/~engelen/soap.html
Robert A. van Engelen, Kyle A. Gallivan
Department of Computer Science and School of Computational Science and Information Technology, Florida State University,
Tallahassee, FL 32306-4530, USA
The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks
In: Proceedings of IEEE CCGrid Conference 2002

[Gendt, Oheim, Gehrke]
Gendt, Michael; Oheim, Frank; Gehrke, Stephan
Geodätisches Seminar zur Koordinatentransformation mittels einer analytischen Lösung der Gaußschen Abbildung
Technische Universität Berlin, 21 Dezember 2000

[GI-Regionalgruppe Hannover]
"dynamic e-business"
GI-Regionalgruppe Hannover, 17.01.2001

[GML 2.0]
Siman Cox, Adrian Cuthbert, Ron Lake, Richard Martell
Geography Markup Language (GML)2.0:
http://www.opengis.net/gml/01-029/GML2.html
OGC Document Number: 01-029
OpenGIS Implementation Specification, 20 February 2001

[Hinke]
Implementing C++ Servlet Containers - Harnessing the power of the Java Servlet API
Dr. Dobb's Journal April 2002

[Hohl et al 1999]
Hohl, Fritz; Kubach, Uwe; Leonhardi, Alexander; Rothermel, Kurt; Schwehm, Markus
Next Century Challenges: Nexus - An Open Global Infrastructure for Spatial Aware Applications
In: Proceedings of the Fith Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, Washington, USA

[Kreger]
Kreger, Heather
Web Services Conceptual Architecture, May 2001
IBM Software Group

[Messmer 2001]
Messmer, Jens
Modellierung der Augmented World in Nexus
Diplomarbeit Nr. 1870, Fakultät Informatik, Universität Stuttgart, 2001

[Möller & Haines]
Tomas Möller; Eric Haines
Real Time Rendering
A K Peters Natick, Massachusetts, ISBN 1-56881-101-2
Chapter 3

[Nexus Vision]
Häusser, Tobias
Nexus - Die Vision
http://www.informatik.uni-stuttgart.de/ipvr/as/lehre/edudoc/WebTut/1_NeXus_DieFision/seite10.html
das Seminar zum Studienprojekt Spatial Model Server für Nexus

[Nicklas 2001]
Nicklas, Daniela; Großmann, Matthias; Schwarz, Thomas; Volz, Stffen; Mitschang, Bernhard
A Model-Based, Open Architekture for Mobile, Spatially Aware Applications

[Nicklas & Mitschang 2001]
Nicklas, Daniela; Mitschang, Bernhard
The Nexus Augmented World Model: An Extensible Approach for Mobile, Saptially Aware Applications

[Reuter]
Reuter, Andreas
Konvergenz der Endgeräte - Integration der Medien-Multimodalität
Baden-Badener Sommerakademie, 11 März 2000

[SAX]
SIMPLE API for XML
http://sax.sourceforge.net/
sax 2.0

[Scribner & Stiver 2001]
Kennard Scribner; Mark C. Stiver
Understanding SOAP
Sams Publishing USA, ISBN 0-672-31922-5

[Schwarz 2001]
Schwarz, Thomas; Nicklas, Daniela; Großmann, Matthias; Volz, Steffen
Information Management and Exchaneg in Nexus

[Tomcat 4.1]
http://jakarta.apache.org/tomcat/tomcat-4.1-doc/index.html

[ViLis 2000]
Virtuelle Litfaßsäule
http://www.informatik.uni-stuttgart.de/ipvr/vs/lehre/ws99000/studienprojekt/Studenten/index.html
Studienprojekt 1999/2000 ViLis

[W3C]
http://w3.org

Erklärung

Ich versichere, daß ich die Arbeit selbstständig verfasst und nur die angegebenen Hilfsmittel verwendet habe.

(Dietrich Müller)