INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Fortgeschrittenenpraktikum

Eine Benutzeroberfläche zur Darstellung von Virtuellen Organisationen und Grid-Diensten

Bojidar Panov

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Fortgeschrittenenpraktikum

Eine Benutzeroberfläche zur Darstellung von Virtuellen Organisationen und Grid-Diensten

Bojidar Panov

Aufgabensteller:	Prof. Dr. Heinz-Gerd Hegering
Betreuer:	Michael Schiffers
	Timo Baur
Abgabetermin:	8. Oktober 2008

Hiermit versichere ich, dass ich das vorliegende Fortgeschrittenenpraktikum selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 8. Oktober 2008

. . . . . .

(Unterschrift des Kandidaten)

Abstract

Diese Arbeit hat als Hauptziel die Entwicklung einer Benutzerschnittstelle für die geographische Darstellung von vorhandenen virtuellen Organisationen (VO's) und Ressourcen in dem D-DGRID. Dabei muss gewährleistet werden, dass jede VO einzeln oder in Verbindung mit anderen VO's darstellbar ist. Die Daten für die VO's werden aus vorhandenen .xml Files und Datenbanken gewonnen, verarbeitet und für die Generierung der einzelnen VO Karten zur Verfügung gestellt. Der Vorteil der Lösung, die in dieser Arbeit vorgestellt ist, besteht in der Möglichkeit des Schichtenaufbaus der Oberfläche bis die gewünschte Information dargestellt wird. In der Tat sind diese Schichten durch eine geographische Darstellung des Regions in dem sich der D-GRID befindet, der einzelnen virtuellen Organisationen, und der vorhandenen Ressourcen, realisiert. So wird dem Benutzer die Entscheidung überlassen, was aus allen Informationen zu einem gegebenen Zeitpunkt auf der Schnittstelle dargestellt wird.

Inhaltsverzeichnis

1 Einleitung

Nachdem die überwiegend statische Bereitstellung von Daten im Internet für viele wissenschaftliche und unternehmerische Aufgaben nicht mehr ausreicht, und immer mehr der Wunsch von einem globalen direkten Zugriff auf die Ressourcen selbst, wie Rechner, Speicher, Anwendungen, Daten, wissenschaftliche Instrumente etc., geäußert wird, nimmt das Grid-Computing immer mehr an Bedeutung zu. Man kann sogar sagen, daß das die nächst bedeutendste Stufe der Internet-Entwicklung sein wird.

Grid-Infrastrukturen bieten den Wissenschaftlern eine Vielzahl von Vorteilen, wie zum Beispiel den transparenten Zugriff und die bessere Nutzung der Ressourcen, nahezu unendlich große Rechen- und Speicherkapazität, Flexibilität und automatische Anpassung von komplexen Rechenprozessen durch dynamischen und konzertrierten Betrieb der vernetzten Ressourcen, höhere Qualität der Ergebnisse durch gridunterstützte Entwicklung, und schließlich Einsparungen durch eine verbrauchsorientierte Abrechnung. Daneben bietet die Grid-Technologie auch Vorteile für die Industrie, insbesondere für solche Unternehmen, die in einer Welt wachsender Dynamik, schrumpfender Entfernungen und globalem Wettbewerb operieren müssen. Ingenieure sind dadurch in der Lage, jegliche Ressource (wie Computer, Anwendung, Daten und Software-Werkzeuge) sozusagen auf Knopfdruck zu nutzen, um Prozesse und Produkte virtuell zu simulieren, bevor sie an deren reale Entwicklung gehen. Dies führt zu höherer Qualität, mehr Funktionalität und Reduktion von Kosten und Risiko. Grid-Technologien helfen die Unternehmens-IT an die realen Geschäftsbedingungen anzupassen (und nicht wie bisher umgekehrt).

In einer Grid-Infrastruktur sind virtuelle Organisationen die zentrale Einheit für die Benutzung der Ressourcen, die das Grid anbietet. Diese Organisationen stellen Informationen über ihren momentanen Zustand. Das heisst: welche Dienste laufen und welche nicht, was sind die Ergebnisse, wer ist der Benutzer, wie Lange es noch bis zum Beenden des Dienstes dauert etc. Das Ziel dieser Arbeit ist die Entwicklung einer kompakten und intuitiven Benutzeroberfläche für die Darstellung dieser Organisationen und Dienste in dem D-Grid. Die Hauptanforderung ist die Möglichkeit der Darstellung einzelner bzw. mehrerer VO's mit deren Diensten und den zugehörigen Informationen. Ausserdem soll der Nutzer VO's dynamisch hinnehmen bzw. wegnehmen können.

1.1 Motivation

Um die Notwendigkeit einer solchen Benutzeroberfläche zu verdeutlichen, stelle man sich eine VO vor, die sehr viel verschiedene Aufgaben zu einem bestimmten Zeitpunkt erledigen soll. Also wird diese VO mehrere Dienste (Ressourcen) in Anspruch nehmen. So wird es klar, dass man den Überblick über diese VO und die Ressourcen, die sie belegt, schnell verlieren kann, insbesondere weil sich alles dementsprechend schnell ändern kann. Umso schwieriger wird es dann neue Aufgaben für diese VO zu planen, ohne, dass man eine einfache Darstellung des Zustands der VO und ihre Ressourcen hat. Eine andere Nutzungsmöglichkeit ist die

gemeinsame Darstellung aller VO's. So kann man sich ein Bild davon machen welche Dienste in dem D-Grid von welchen VO's benutzt werden und welche gerade frei sind.

1.2 Scenario

Um so eine Benutzeroberfläche anbieten zu können, muss man die VO's und die Dienste geographisch darstellen können. Dafür benötigt man eine Karte des Gebiets, auf dem sich das Grid erstreckt. Für dieses FoPra wird die Bundesrepublik Deutschland als Grundlage genommen. Die Technologien, die man dafür benutzen wird, machen es aber möglich das Gebiet beliebig auszudehnen. Das heißt, dass es ein weltweit abdeckendes Grid realisierbar ist. Damit man die Darstellung einzelner, oder aller VO's realisieren kann, wird die Karte Schicht für Schicht aufgebaut. Die erste Schicht wurde gerade angesprochen, nämlich die Deutschland Karte. Sie ist auch die Grundschicht, die immer vorhanden sein wird. Die nächsten Schichten werden dann optional zuschaltbar sein, und werden aus der Darstellung einzelner VO's bestehen. So wird dem Benutzer die Möglichkeit gegeben nach Wunsch VO's hinzunehmen oder zu entfernen. Durch die „Überlappung"der verschiedenen Schichten erzielt man die gewünschte individuelle Visualisierung. Folglich muss die Benutzeroberfläche, die Möglichkeit anbieten diese Schichten dynamisch hinzuzunehmen oder zu entfernen. Außerdem muss auch eine Benutzerverwaltung angeboten werden, mit derer Hilfe man die Rechte einzelner Benutzer festlegen kann. Das heißt, dass jeder Benutzer nur das sehen können muß, was er auch sehen darf.

1.3 Grundlegende Begriffe

Grid Bei einem Grid handelt es sich um eine Infrastruktur, die eine integrierte, gemeinschaftliche Verwendung von meist geographisch auseinander liegenden, autonomen Ressourcen erlaubt.

Grid-Computing Die gemeinsame Nutzung von Ressourcen, mit der wir uns hier beschäftigen, ist nicht primär der Austausch von Dateien, sondern vielmehr der direkte Zugriff auf Computer, Software, Daten und andere Ressourcen, wie sie bei einer Reihe von kollaborativen, problemlösenden und Ressourcen vermittelnden Strategien benötigt werden, die zur Zeit in Industrie, Wissenschaft und im Ingenieurwesen auftauchen. Diese gemeinsame Nutzung von Ressourcen ist, notwendigerweise, in einem Höchstmaß kontrolliert, wobei die Anbieter und Konsumenten der Ressourcen klar und eindeutig festlegen, welche Ressourcen geteilt werden, wem die gemeinsame Nutzung erlaubt ist, und unter welchen Bedingungen die gemeinsame Nutzung erfolgt. - *I. Foster and C. Kesselman, „The Grid: Blueprint for a new computing infrastructure,"2. Auflage, Morgan Kaufmann, Tech. Rep., 2003. ISBN 978-1-55860-933-4*

Grid Dienst Kann verschiedene Natur haben und steht in einem Grid zur Verfügung. Jeden Dienst werden Ressourcen zugeordnet, wobei er ein Teil oder alle belegen kann. Ein Benutzer kann sich für ein Dienst registrieren und ihn in Anspruch nehmen. Für die Zeit, die der Dienst dem Benutzer zugeordnet ist, kann im Regelfall kein anderer Benutzer darauf zugreifen. Beispiel ist ein Jobdienst, der die Berechnung des Luftwiderstandes von drei dimensionalen Formen durchführt.

Virtuelle Organisation (VO) Ist eine Menge von Individuen und/oder Institutionen, die durch Richtlinien zur gemeinsamen Nutzung von Ressourcen in einem Grid verbunden sind. Die Notwendigkeit solcher Organisationen liegt in der Tatsache, dass es Aufgaben gibt, die so komplex sind, dass sie von einer einzigen realen Organisation alleine nicht gelöst werden können.

Geospatial Information System (GIS) Ist ein rechnergestütztes Informationssystem, das aus Hardware, Software, Daten und den Anwendungen besteht. Mit ihm können raumbezogene Daten digital erfasst, gespeichert, reorganisiert, modelliert, analysiert sowie alphanumerisch und grafisch präsentiert werden. Geoinformationssysteme sind normalerweise verteilt und bieten drei verschiedene Dienste an: Web Feature Service (WFS), Web Coverage Service (WCS), Web Map Service (WMS). Ob diese Dienste alle angeboten werden hängt von dem Typ des Geoinformationsystems an.

Web Coverage Service (WCS) Ist ein vom Open GIS Consortium (OGC) entwickelter Geoservice-Standard. Der Web Coverage Service 1.0.0 (WCS; OGC-Dokument Nr. 03-065r6) normiert den Zugriff auf große, multidimensionale Rasterarchive. (Prinzipiell ist der Begriff „Coverage“weiter gefasst, jedoch fokussiert WCS derzeit auf Raster).

Web Feature Service (WFS) Darunter versteht man den internetgestützten Zugriff auf Geodaten innerhalb eines verteilten GIS. Der WFS beschränkt sich dabei ausschließlich auf Vektordaten, wie sie in Datenbanken abgelegt werden können.

Web Map Service (WMS) Ist ein Web Service, der auf den WMS Implementation Specifications beruht, einer Schnittstelle, um Karten zu generieren. Diese Spezifikation beschreibt eine Schnittstelle des Open Geospatial Consortium (OGC) zur Kartengenerierung anhand spezifischer Parameter.

2 Anforderungen und Systemdesign

In diesem Kapitel wird darauf eingegangen, wie die vorgenommene Aufgabe zu realisieren ist und welche Anforderungen letztendlich erfüllt werden müssen.

2.1 Anforderungen

Die weiter unten aufgelisteten Anforderungen entstehen durch vertiefte Analyse der im ersten Kapitel beschriebenen Aufgabe. Ein Teil davon sind direkt aus dem User Szenario entstanden, ein anderer Teil wird durch den Einsatz eines Geoinformationsystems diktiert.

1. WFS Funktionalität für den Zugriff auf die Features eines VO's Jede Virtuelle Organisation wird durch ihre Eigenschaften (features) beschrieben. Diese Eigenschaften sind verschiedener Natur. Einige beschreiben die VO, andere geben Auskunft über die Ressourcen in der VO, und andere stellen wiederrum die räumlichen Koordinaten von den Ressourcen. Der momentane Zustand (snapshot) der VO wird auch mit Hilfe der Eigenschaften represäntiert. Aus diesem Grund ist ein **W**eb **F**eature **S**ervice (**WFS**) unerlässlich, wenn es darum geht, gegebene Eigenschften der VO's zu verändern.

2. WMS Funktionalität für das Kreieren der Karten der einzelnen VO's Auch die am vollständigsten beschriebenen Virtuellen Organisationen nutzen einem wenig, wenn es keine einfache Visualisierung der Information vorliegt. Da wir von einem Geoinformationssystem sprechen, wird hier diese Visualisierung mit Hilfe von Karten (maps) realisiert. Für den Aufbau und die Bereitstellung solcher Karten ist in einem GIS der **W**eb **M**ap **S**ervice (**WMS**) zuständig. So wird er auch unverzichtbarer Bestandteil der Lösung.

3. UI mit der Möglichkeit der Hinzunähme von zusätzlichen Karten Weil es hauptsächlich darum geht, eine VO mit all seinen Diensten und zusätzlichen Informationen darzustellen, ist das Konzept dieser Arbeit eine VO mit nur einer Karte darzustellen. Weil aber es Benutzer Szenarien gibt, in denen ein Benutzer den Überblick über mehr als eine VO gleichzeitig behalten muß, wobei diese VO's eventuell dynamisch kreiert oder gelöscht werden, ist es auch notwendig, dass die Benutzerschnittstelle die zusätzliche Karten auch dynamisch hinzunehmen oder entfernen kann.

4. Zusammenfassen verschiedener Karten zu einer Karte. Eine andere Situation, die abgedeckt sein muß, ist die Darstellung der Informationen von mehr als eine bzw. allen VO's. Ein solches Szenario ist besonders für Benutzer mit administrativen Rechten, die den Überblick und das Geschehen über einen Teil oder über den gesamten Grid, mit all seinen VO's und Dienste, behalten muß. Da jede einzelne VO auch ihre eigene Karte hat, muß auch eine Möglichkeit bestehen, diese Sichten zu Einer zusammenzufassen. Im Prinzip ist

diese Funktionalität auch dann gegeben, wenn alle Karten aufeinander “gestappelt„werden können. So entspricht eine Schicht der gesamten Sicht, die Sicht auf die einzelne VO.

5. Benutzerverwaltung Allein die Existenz verschiedener Rollen, wie “Administrator „, “Gast „, usw., die ein Benutzer annehmen muß, reicht aus um eine Benutzerverwaltung unverzichtbar zu machen. In dieser Arbeit kommt noch verstärkt hinzu, dass ein Benutzer vielleicht nicht alle Informationen zu einer gegebenen VO-Karte sehen darf, sondern nur ein Teil davon.

6. Hinzufügen von zusätzliche Markierungen auf der Karte Um die ganze Darstellung benutzerfreundlich zu gestalten, werden zusätzliche Markierungen, die nicht auf Geo-Daten basieren, benötigt.

2.2 Design

Um die gestellten Ziele, die in der Einleitung beschrieben worden sind, realisieren zu können wird ein Geospatial Information System (GIS) aufgebaut. Mit Hilfe dieses Geoinfirmationssystems wird man in der Lage sein die gewünschte Funktionalitä der Benutzerschnittstelle, anzubieten. Außerdem wird eine Schnittstelle zwischen dem GIS und den eigentlichen D-GRID Daten, die in verschiedener Form vorliegen, benötigt. Ein GIS besteht aus mehreren verschiedenen Diensten. Das sind unter anderem:

Web coverage service (WCS)
Web feature service (WFS)
Web map service (WMS)

Die Anzahl dieser Dienste kann variieren und verteilt sein. Das heißt, dass man zum Beispiel mehrere WMS Dienste, ein WFS Dienst, und gar kein WCS Dienst haben kann. Alle Dienste können dann an verschiedenen Orten sein und auf verschiedene Rechner laufen. Außerdem verfügt das GIS über eine Datenbank, die alle benötigten Informationen für die Darstellung der Karten beinhaltet. Für die Datenbank gilt auch die Ort- bzw. Anzahlunabhängigkeit. Das gesamte GIS dient nur als Background für die Lieferung der für die Benutzeroberfläche benötigten Daten. Um die Realisierung des FoPras abzurunden, wird auch eine Implementierung dieser Oberfläche benötigt.

3 Implementierung

In diesem Kapitel wird ausführlich beschrieben, welche Software für die Realisierung des im Kapitel 1 besprochenen Szenarios, programmiert werden muß. Außerdem wird erklärt, wie das gemacht worden ist.

3.1 Optionen und Entscheidungen

Es existieren für alle GIS Komponente schon fertige Lösungen. Das gilt nicht nur für die Dienste, sondern auch für die Datenbank und die eigentliche Benutzeroberfläche, die die einzelnen Karten darstellt. Ein wichtiges Aspekt bei so einem komplexen System sind die Kosten, die mit der Softwarebeschaffung verbunden sind. Um diese Kosten praktisch zu eliminieren wird in dieses FoPra nur freeware Software benutzt. Da es aber trotzdem viele verschiedene Implementierungen solcher Software gibt, muss man sich vorher Gedanken machen was überhaupt die einzelne Komponente leisten müssen und welche Software das auch abdeckt
Für den WFS bzw. WMS wird ein Server gebraucht. Die meist verbreiteten und erprobten Server (nach eigener Recherche) sind UMN MapServer und GeoServer. Wie man in der folgenden Tabelle sieht, bieten beide die gewünschten Dienste an.

GIS Server Vergleich			
Server Typ	**Web Map Service**	**Web Feature Service**	**WFS Transaktional**
UMN MapServer	ja	ja	nein
GeoServer	ja	ja	ja

Weil der **GeoServer** ein transaktionaler Zugriff auf die Features ermöglicht wird er als GIS Server für diese FoPra gewählt.

Für die Benutzeroberfläche (Anforderungen 3., 4., 5.) ist der **Mapbender**, mit seinem „multi layer “Konzept, zur Zeit die bestmögliche Auswahl. Dieses Konzept erlaubt es die Hauptfunktionalität dieser FoPra abzudecken. Man generiert für jede Virtuale Organisation ein eigener Layer und ermöglicht es dem Nutzer, nur diese Layer darzustellen, die er will oder darf. Ausserdem verfügt der Mapbender über eine ausgereifte Benutzerverwaltung.

Für die zusätzliche Markierungen der VO's auf der Karte (Anforderung 6.) wird der **Style Language Descriptor (SLD)** benutzt.

Als Datenbank nimmt man **postgreSQL** mit **Proj4**, **GEOS** und **postGIS** Erweiterung für die Geodatenunterstützung.

Zusätzliche Software, die noch gebraucht wird ist ein apache Server und PHP. Jede dieser Software ist für die meist verbreiteten Betriebssystemen verfügbar (Windows, Mac, Linux).

Für die FoPra wird alles auf ein Linux Rechner installiert und betrieben. Folgende Liste der für die Realisierung der FoPra benutzte Software ergibt sich:
•PostgreSQL

•Proj4, GEOS und die postGIS-Erweiterungen

•Mapbender mit apache2 und PHP5

•GeoServer

Achtung: Software die kein direkter Zusammenhang mit der GIS Funktionalität hat, aber benötigt wird, um die GIS-Software zu installieren, wird hier nicht erwähnt. Beispiel für solche Software ist: (g)make und andere benötigte Pakete.

D-GRID GIS
Bildlich vorgestellt sieht das Geoinformationssystem wie auf dem folgenden Bild gezeigt.

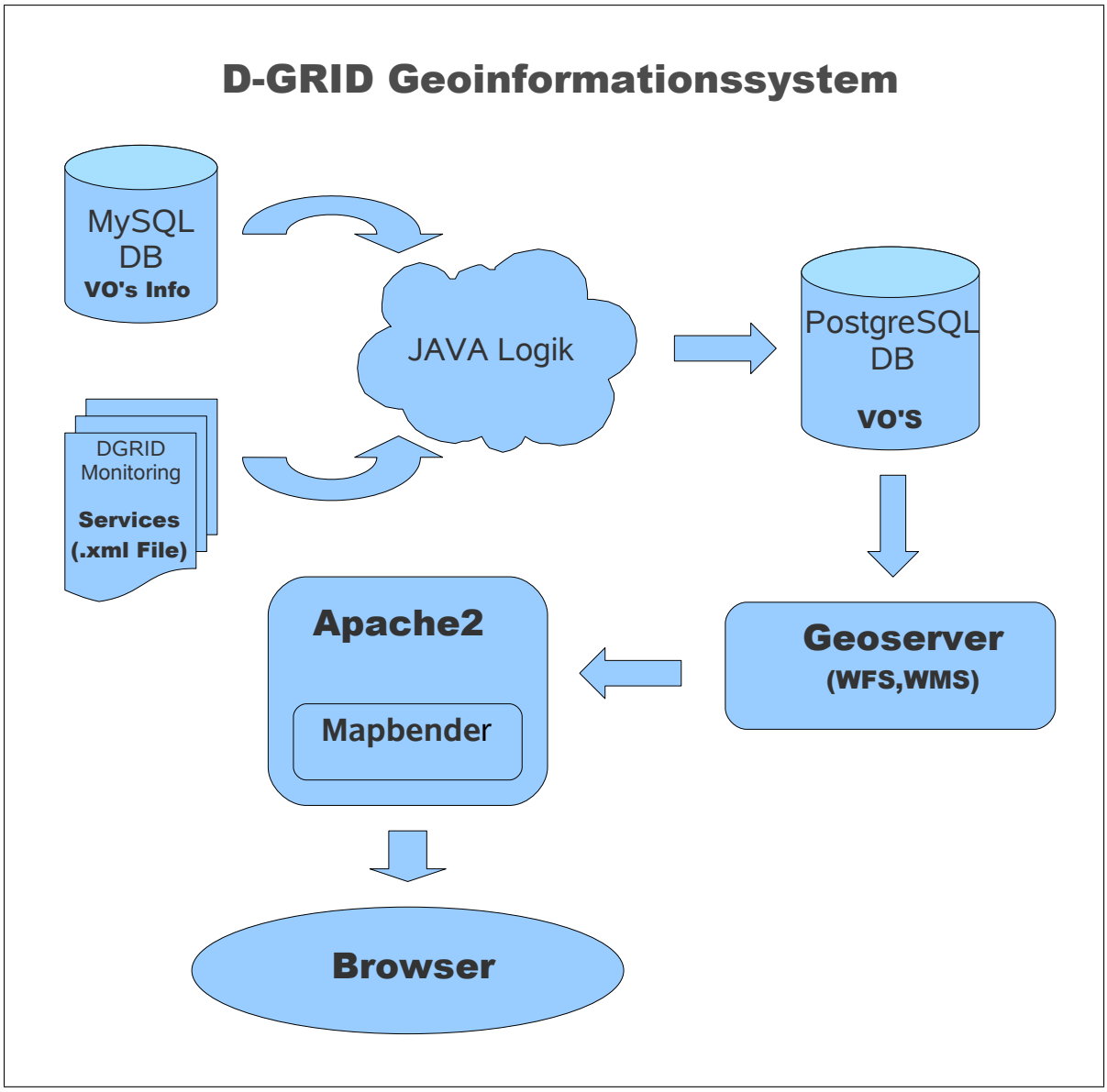

Abbildung 3.1: D-GRID GIS

3.2 Ziele

Um die Implementierung modular zu gestalten, sind folgende vier Teilaufgaben als Endziele zu erreichen.

1. **Implementierung eines XML Parsers für die Daten der Services, die in dem D-GRID bereitstehen.**

   Die Daten, die alle Services in dem D-GRID darstellen, sind jederzeit mit Hilfe eines Web Services abrufbar und stehen in xml Format zur Verfügung. Diese Daten beinhalten unter anderem folgende Informationen: Name, Beschreibung, URL, Ort und für ein Geosystem die relevanteste Information, nähmlich geographische Länge und Breite.

2. **Implementierung einer MySQL Datenbankanbindung für weitere Daten der virtualen Organistionen.**

   Da aus dem xml File keine Daten gewonnen werden können, die Aussagen welcher Dienst gerade von welcher VO benutzt wird, braucht man eine zusätzliche Verbindung zu einer MySQL Datenbank, wo diese Informationen zu finden sind. Diese Anbindung muß eine Logik in Form von MySQL Anweisungen beinhalten, um die Daten zu holen.

3. **Implementierung einer Logik für die Auswertung der aus dem xml File und MySQL Datenbank gewonnenen Daten.**

   Nachdem alle Daten, die für die VO's nötig sind, gewonnen sind, müssen noch diese Daten verarbeitet werden. Im Grunde werden aus den Daten die eigentlichen virtuellen Organisationen kreiert.

4. **Implementierung einer PostgreSQL Datenbankanbindung für GeoServer.**

   Nachdem die VO's nun zur Verfügug stehen, braucht man eine Verbindung zu der Datenbank des Geoservers, um dort die entsprechende Tabellen für die Kartengenerierung einzupflegen. Um dieses Vorhaben umsetzen zu können, braucht man außer der Verbindung zu der Datenbank, noch eine Logik in der Form von PostgreSQL Anweisungen, die die Daten tatsächlich einfügt.

Die Implementierung aller Modulle erfolgt ausschliesslich unter JAVA 1.5. Als zusätzliche Bibliotheken sind MySQL und PostgreSQL Bibliotheken, die die Datenbanktreiber zur Verfügung stellen, benutzt worden.

3.3 Realisierung

In diesem Abschnitt wird beschrieben wie die Ziele aus 5.1 realisiert worden sind und welche Technologien dabei genutzt worden sind.

3.3.1 XML Parser

Bei der Implementierung eines jeden Parsers ist die grundsätzliche Frage, die sich stellt, „Was will man eigentlich? Soll der Text, der geparst wird, auf irgendeine Weise verändert werden, oder reicht es, wenn man bestimmte Textelemente findet? ". Bei dem XML Parser für diese FoPra reicht es, wenn man bestimmte XML Tags und deren Inhalt findet und extrahiert.

Aus diesem Grund wurde hier ein Hauptspeicher schonendes SAX Parser gewählt und implementiert. Der größte Vorteil von SAX gegenüber DOM Parser ist, dass man keine komplette Baumdarstellung des XML Dokuments in dem Hauptspeicher halten muss, um parsen zu können. Der XML Parser wird am Anfang des Dokuments „losgelassen “und wirft jedes Mal ein Ergebnis aus, wenn er auf dem gesuchten Tag trifft. Mit einem DOM Parser wäre das zwar auch möglich, aber aus Speicher- und Komplexitätsgründe (bei der Implementierung) wurde hier SAX Parser gewählt. Die Daten die aus dem Parser gewonnen werden sind ausschliesslich Daten über den einzelnen Dienste, die in dem D-GRID zur Verfügug stehen. Es werden folgende Informationen für jeden Dienst extrahiert:

1. Name (der Name des Dienstes)
2. URL (URL unter die der Dienst zu finden ist)
3. Beschreibung (Was der Dienst leistet)
4. Zusätzliche Information (Beliebige Information)
5. Lokalität (Beschreibung wo sich der Dienst befindet)
6. geographische Breite (Koordinate)
7. geographische Länge (Koordinate)

Aus dem XML File können noch mehr Informationen gewonnen werden. In dieser Arbeit wurde aber darauf verzichtet, weil es parallel eine Datenbank entwickelt wird, die diese Informationen beinhalten wird. Bei Bedarf kann in der Zukunft dann alles aus der Datenbank geholt werden. Der XML Parser ist nur eine „vorübergehende “Lösung. Die zwei wichtigsten Informationen für die Darstellung der virtuellen Organisationen sind die geographische Breite und Länge, die später bei der Generierung der Karten der VO's zentralle Rolle spielen.

Mehr Information über die Implementierung kann man aus den am Ende dieser Arbeit hinzugefügten Sourcen bekommen.

3.3.2 Datenbanken

Wie schon bei den Zielen aufgelistet, benötigt man zwei Datenbankanbindungen. Eine für die zusätzliche Information der virtuellen Organisationen und eine für die Einpflegung der Enddaten in dem Geoserver.

MySQL Datenbankanbindung für Services und VO Daten.

Wie schon bei dem XML Parser erwähnt wurde, existiert außer dem .xml File noch eine andere Datenquelle, die nötig, ist um die VO's komplett zu gestallten. Diese Datenquelle ist eine MySQL Datenbank, die unter anderem die Zuordnung Dienst zu VO beinhaltet. Aus dem .xml File ist diese Zuordnung nähmlich nicht zu holen. Implementierungstechnisch besteht dieser Teil der Arbeit aus einer Datenbankverbindung, die mit Hilfe eines MySQL Treibers für JAVA realisiert ist, und ein Paar SQL Anweisungen, um die Daten zu holen. Die Logik, die bei der Datenverarbeitung gefolgt wird, kann man folgender Maßen beschreiben: Für jeden Service, den man aus dem .xml File gewonnen hat, finde aus der MySQL Datenbank alle virtuellen Organisationen, die diesen Service gerade beanspruchen, und aktualisiere die eigentlichen VO Daten.

Mehr Information über der Implementierung kann man aus den am Ende dieser Arbeit hinzugefügten Sourcen bekommen.

PostgreSQL Datenbankanbindung für GeoServer.

Nachdem alle Daten, die nötig sind, aus dem .xml File und der externen Datenbank gewonnen sind, und alle virtuelle Organisationen als Objekte generiert sind, muss dafür gesorgt werden, dass sie auch in der Datenbank von dem Geoserver eingepflegt werden. So wird gewährleistet, dass der Web Map Service des Geoservers in der Lage ist, für diese VO's einzelne Karten zu generieren, und zur Verfügung zu stellen. Erst dann kann der Mapbender mit einer „getFeature"Anfrage die Karten holen und entsprechend seiner Konfiguration, sie anzuzeigen. Implementierungstechnisch besteht dieser Teil der Arbeit aus einer Datenbankverbindung, die mit Hilfe eines PostgreSQL Treibers für JAVA realisiert ist, und ein Paar SQL Anweisungen, um die Tabellen für die VO's anzulegen.

Mehr Information über der Implementierung kann man aus den am Ende dieser Arbeit hinzugefügten Sourcen bekommen.

3.3.3 Logik für die Auswertung und Verarbeitung der gewonnenen Daten

Als letzter Puzzlestein der Implementierung dient der auf der nächsten Seite bildlich dargestellte Ablauf der gesamten Software. Folgende drei Schritte sind dann der Kern der Datenaufbereitung für die Benutzerschnittstelle.

1. Schritt: Hole aus dem .xml File mit Hilfe des Parsers alle verfügbaren Ressourcen in dem D-GRID.

2. Schritt: Für jede Ressource ermittle aus der MySQL Datenbank, welche VO's gerade diese Ressource in Anspruch nehmen. Kreiere oder update dabei die einzelnen VO-Objekte.

3. Schritt: Generiere für jedes VO-Objekt eine PostgreSQL Tabelle in der PostgreSQL Datenbank des Geoservers. Diese Tabellen dienen später als Grundlage für die Karten der einzelnen VO's.

Es wird noch ein zusätzlicher Schritt, der auf dem Bild nicht sichtbar ist und für die Darstellung der einzelnen VO's nicht notwendig ist, durchgeführt. Dieser Schritt besteht darin, einzelne Tabellen für jede Ressource in der PostgreSQL Datenbank anzulegen. Diese Tabellen werden dann benutzt, um die Ressourcen als Positionen auf der Karte anzeigen zu können.

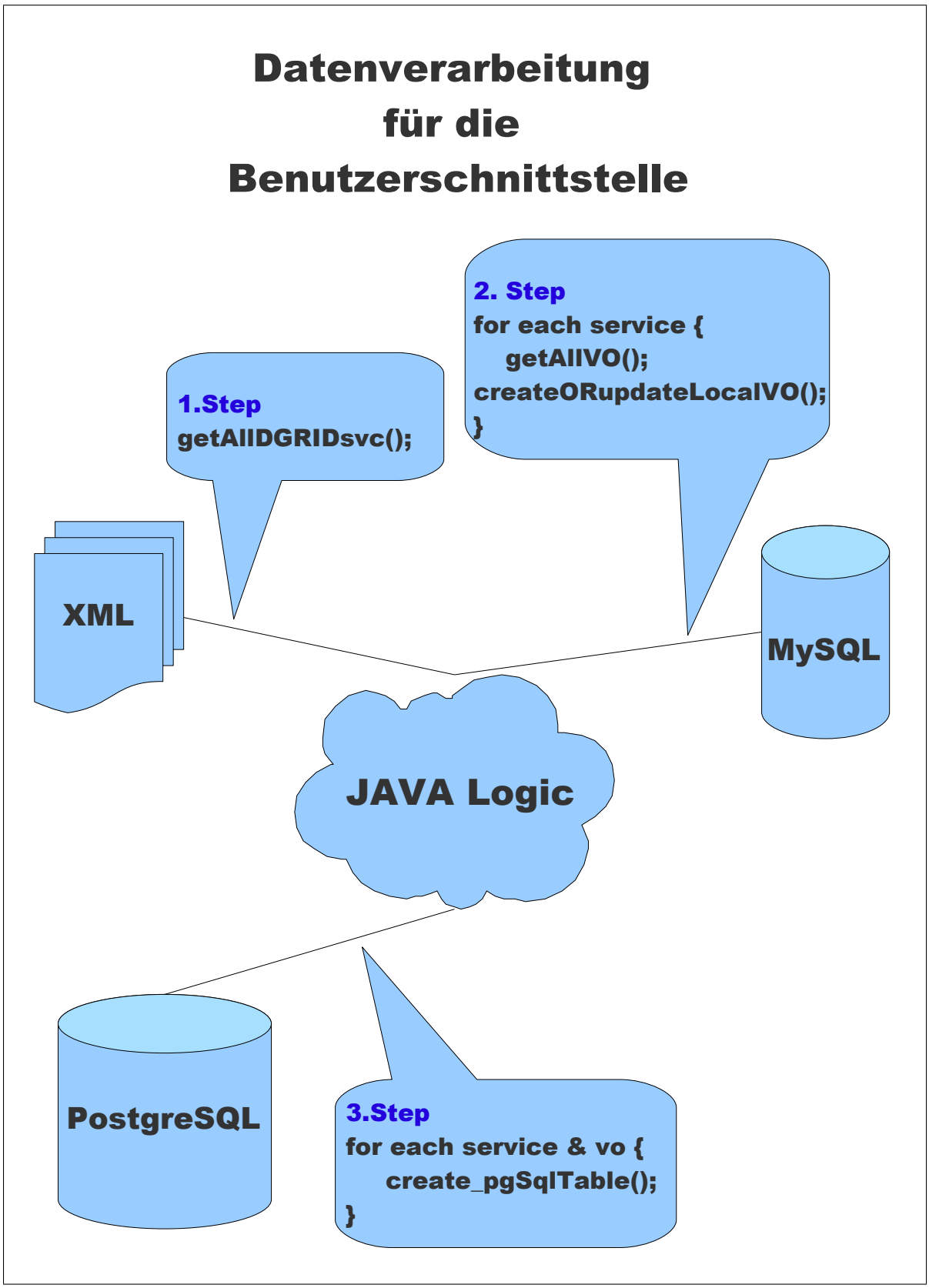

Abbildung 3.2: Datenverarbeitung

4 Deployment der Java-Komponente

Um die realen Daten der virtuellen Organisationen letztendlich anzeigen zu können, muß das GIS mit der im Kapitel 4 beschriebenen Komponente ergänzt werden. Im Folgenden wird erläutert, wie das Deployment dieser Komponente durchgeführt werden muß.

4.1 Voraussetzungen

Die Java-Komponente ist im Prinzip von der Sortware des gesamten GIS unabhängig. Das einzige, was sie davon benutzt, ist die PostgreSQL Datenbank, wobei diese Datenbank über die PostGIS Erweiterungen verfügen muß. Wie man die Installation der Datenbank vornimmt, kann man aus dem in Kapitel 4.2.1 bis 4.2.3 angegebenen Installationsquellen erfahren.

4.2 Installation der GIS-Komponente

Dieses Kapitel gibt eine Übersicht darüber, wie die Software installiert werden soll und woher sie zu kriegen ist.

4.2.1 Installation von PostgreSQL

Download: Unter dem Link `http://www.postgresql.org/ftp/binary/v8.3.1/` kann man sich die neuesten Versionen der Datenbank binaries für Linux, Windows und Solaris herunterladen.

Installation: An diese Stelle wird auf die Seite `http://www.postgres.de/install.html` hingewiesen, wo es eine Schritt für Schritt Anweisung auf Deutsch gibt, wie man eine postgreSQL Datenbank auf Linux installiert. Wer der englischen Sprache mächtig ist kann natürlich auch die offiziellen Installationsanweisungen von postgreSQL unter dem Link `http://www.postgresql.org/docs/8.3/static/installation.html` benutzen.

4.2.2 Installation von Proj4 und GEOS

Download: Proj4 steht unter `http://proj.maptools.org/` zum Download bereit. GEOS steht unter `http://trac.osgeo.org/geos/` auch zum Download bereit.

Installation: Unter `http://proj.maptools.org/` steht eine Installationsanweisung sowie FAQ bereit.

4.2.3 Installation von PostGis Erweiterungen

Download: Die neuesten binaries stehen zum Download bereit unter `http://postgis.refractions.net/download/`.

Installation: Eine kurze Installationsanweisung (die aber meistens ausreicht) ist unter `http://postgis.refractions.net/documentation/` zu finden. Für Lösungen auf die meisten Probleme steht auch ein Wiki unter `http://postgis.refractions.net/support/wiki/` zur Verfügung.

4.2.4 Installation von GeoServer

Download: Die neuste Geoserver Version steht unter `http://geoserver.org/display/GEOS/Download` zum Download bereit.

Installation: Unter `http://geoserver.org/display/GEOSDOC/1.1+Install+GeoServer` ist eine detaillierte Schritt für Schritt Anweisung wie man GeoServer installiert und konfiguriert.

4.2.5 Installation von apache2 und PHP5

Download: Die neueste Version von apache2 findet man unter `http://httpd.apache.org/download.cgi` zum Download bereit. PHP5 kann man unter `http://www.php.net/downloads.phpherunterladen`.

Installation: Eine detaillierte Anweisung wie man apache2 zusammen mit PHP5 installiert und konfiguriert ist unter `http://de.php.net/manual/de/install.unix.apache2.php` zu finden.

4.2.6 Installation von Mapbender

Download: Unter `http://www.mapbender.org/download/` kann man die neueste Version von Mapbender herunterladen.

Installation: Unter `http://www.mapbender.org/Installation` findet man Installationsanweisungen auf Deutsch und Englisch. Für weitere Fragen steht unter `http://www.mapbender.org/Main_Page` ein Wiki zur Verfügung.

4.3 Ausführung auf mabtest.lrz-muenchen.de

Um die Software zu starten, müssen folgende Befehle in der Shell-Konsolle (linux) oder CMD-Eingabe (Windows) ausgeführt werden:
Hier wird als Beispiel Shell verwendet.

1. Schritt: Einloggen per ssh
   ssh -l panov mabtest.lrz-muenchen.de
   password: ge0p4s

2. Schritt: In dem FoPra Verzeichnis wechseln
   cd FoPra

3. Schritt: Falls das Verzeichnis GeoDataParser schon existiert überspringen Sie diesen Schritt und machen Sie mit dem nächsten weiter. Das mitgelieferte GeoDataParser.tar Archiv auspacken
   tar -xvzf GeoDataParser.tar.gz

4. Schritt: In dem Verzeichnis bin wechseln
   cd GeoDataParser/bin

5. Schritt: XML Daten holen. Eine webmds.xml Datei vom 31. August 2008 ist schon auf dem rechner unter /home/panov/FoPra abgelegt. Falls Sie eine neuere benutzen wollen, laden Sie sie herunter indem Sie in dem Sie folgende URL aufrufen: `http://webmds.lrz-muenchen.de:8080/webmds/webmds?info=dgridinfo`. Speichern sie dann die Datei als webmds.xml und ersetzen Sie die alte Datei unter FoPra mit der neuen. Überspringen Sie diesen Schritt falls Sie keine neuere webmds.xml Datei benutzen wollen.

6. Schritt: Programm starten. Führen Sie folgende zeile aus:
   java -cp .:../lib/mysql-connector-java-5.1.6-bin.jar:../lib/postgresql-8.3-603.jdbc3.jar GeoParser ../../webmds.xml

   Es folgt eine Ausgabe, wobei es ziemlich am Ende zwei java.sql.BatchUpdateException und zwei org.postgresql.util.PSQLException ausgegeben werden. Diese kann man ignorieren. Alles hat trotzdem richtig geklappt.

4.4 Ausführung auf einem anderen Rechner

Die Java-Komponente kann auf jeden beliebigen Rechner, auf dem Java 1.5 installiert ist, ausgeführt werden. Dafür muß sie aber auch entsprechend konfiguriert werden.

4.4.1 Datenbank Konfiguration

Um die Software auf einem anderen Rechner ausführen zu lassen, müssen zuerst die drei Datenbankanbindungen in dem source code angepasst werden. Um das zu erleichtern wird hier ein .tar Ball mitgeleifert, der das gesamte Software als eclipse project beinhaltet. Dazu müssen Sie den File GeoDataParser.tar.gz in einem Verzeichniss entpacken und das darin enthaltene Projekt in eclipse importieren.

1. **PostgreSql**
   In der Klasse LocalJDBCConnector in der Zeile 31-32 finden Sie die Stelle wo Sie die URL, Benutzer und Passwort angeben müssen. Bitte beachten Sie, dass die URL am Ende der Name der Datebank enthalten muss.

2. **MySQL**
   In der Klasse ExternalJDBCConnector können auch die GLUE20 und External Datenbanken umkonfiguriert werden. In der Regel ist das aber nicht nötig, ausser Sie wollen mit einem anderen Benutzer darauf zugreifen.

Nachdem die Konfiguration der Datenbankanbindungen abgeschlossen ist, machen Sie ab Schritt 4 in Abschnitt 4.3 weiter. Sie können auch alternativ das Ganze in eclipse als java Programm starten, wobei in der run Umgebung noch ein File Prompt für den webmds.xml File konfiguriert werden muß.

5 Status Quo

Nachdem hier beschrieben wurde, was das gesamte System leistet, welche Vorteiele es bringt, und wie man es zum Laufen bekommt, ist hier die Stelle für eine visuelle Darstellung. Anhand dieser Darstellung können nochmal die grundlegende Funktionen nachvollzogen werden. Folgendes Bild ist ein Snapshot, des zur Zeit verfügbaren Systems.

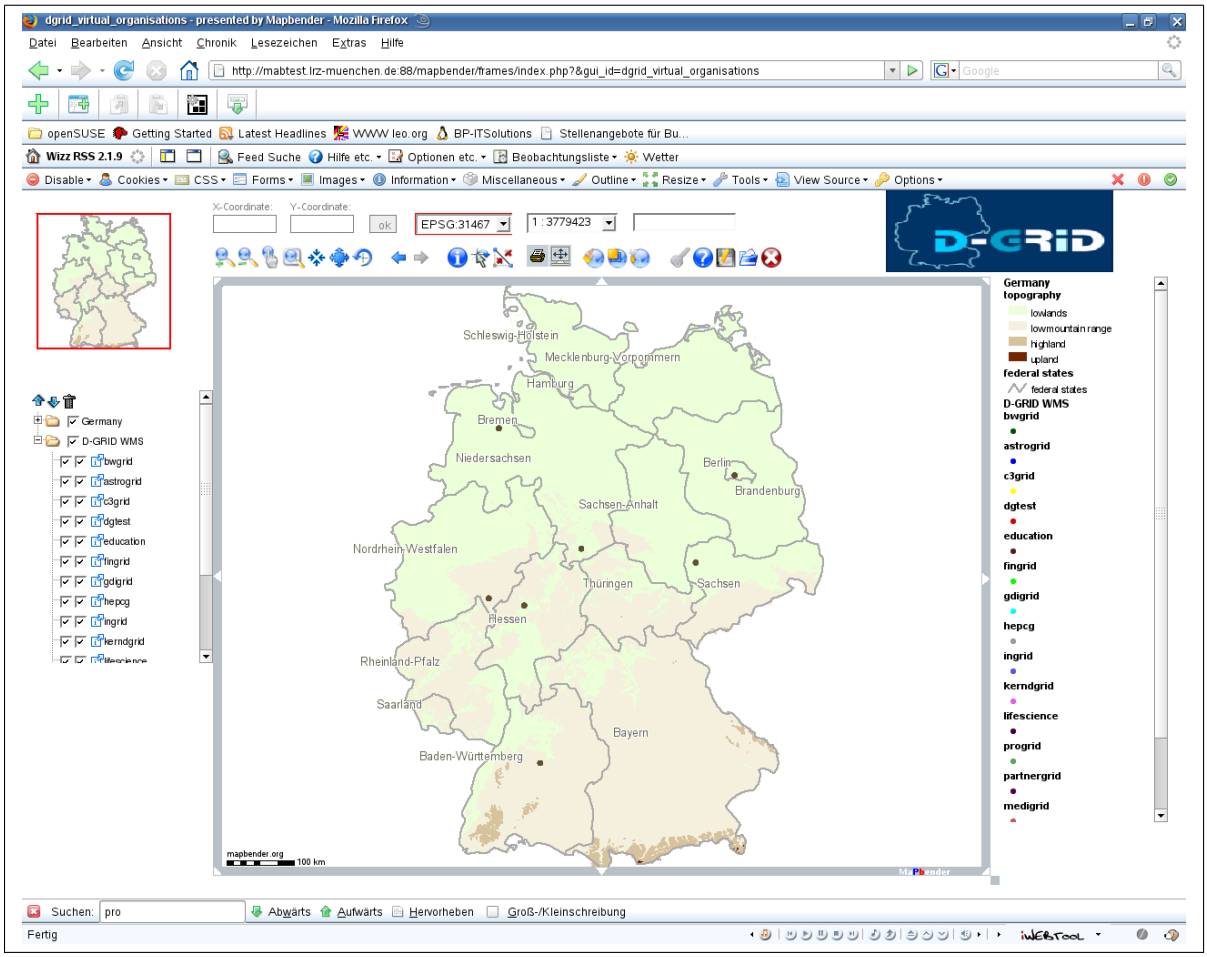

Abbildung 5.1: D-GRID Benutzerschnittstelle

Auf diesem Bild sind alle verfügbare virtuelle Organisationen ausgewählt, wobei die einzelnen Karten sich überlappen und die Information aller VO's anbieten. Auf Grund der Locationgleichheit von den einzelnen VO's überlappen sich hier alle Punkte und sind grau dargestellt. Wobei fairweise gesagt werden muss, dass zu diesem Zeitpunkt jede VO seine eigene Farbe hat, was der besseren Trennung dient. Sobald aber eine Location in mehreren VO's vorkommt, ist dieser Ansatz nicht mehr zu gebrauchen, weil es nicht mehr entschieden

werden kann, welche Farbe der Punkt annehmen muss. Wenn in der Zukunft auch die anderen Informationen der VO's darstellbar werden (siehe Kapitel 6 Punkt 5), wird die Farbe keine Rolle mehr, für die Trennung der einzelnen VO's spielen. Die Information, welcher Punkt zu welchen VO's gehört, wird auf eine andere Art und Weise dargestellt.

6 Weiterentwicklung

Diese Arbeit deckt nur die ersten Schritte bei der Visualisiering der virtuellen Organisationen in dem D-GRID mit Hilfe eines Geoinformationssystems. Hier sind nur einige der grundlegenden Eigenschaften eines solchen Systems vorgestellt. Für die Weiterentwicklung entstehen noch mehrere Aufgaben, die gelöst werden müssen. Einige der wichtigsten sind:

1. **VO Daten Quellen**
   In der Zukunft werden alle verfügbaren Daten einer VO in einer bzw. mehreren Datenbanken zur Verfügung stehen. Das bedeutet, dass der Service, der die Benutzerschnittstelle anbieten wird, die Daten, die er braucht, nicht mehr aus einem .xml File beziehen wird, sondern mehrere komplexe Datenbank-Transaktionen durchführen wird.

2. **GIS Daten Quellen**
   Zum Zeitpunkt dieser Arbeit besteht die GIS Datenbank nur aus einer Liste der locations und der virtuellen Organisationen. Wobei die VO's noch ein Paar zusätzliche Informationen beinhalten. Für die Zukunft wird das keines Wegs ausreichen. Es wird eine komplexe umfangreiche Datenbank nötig sein, um alle Informationen der VO's zu verwalten.

3. **Freundlichkeit und Vollständigkeit der Benutzerschnittstelle**
   Das Design für die Darstellung der Daten auf der Benutzerschnittstelle ist noch zu entwickeln.

4. **Effizienz**
   Weil die VO Informationen sich dynamisch ändern können, ist eine sich ständig wiederholende Abfrage von allen relevanten Datenbanken nötig. Ausserdem muss die Benutzerschnittstelle jedes Mal upgedated werden. Die früher erwähnten Transaktionen, die der GeoServer unterstützt, spielen dabei eine wichtige Rolle.

5. **Informationsdarstellung**
   In dieser Arbeit geht es um die Informationen der einzelne VO's. Zur Zeit werden aber nur der Name und die Locations der einzelnen bzw. alle VO's dargestellt. Die Erweiterung um die restlichen Informationen steht noch an.

Im Allgemeinen sind in einem GIS alle diesen Fragestellungen lösbar. Der Grund ist, dass das gesamte System modular aufgebaut ist. Das garantiert die Freiheit bestimmte Module auszutauschen, zu erweitern oder hinzunehmen. Also man ist nicht an eine bestimmte Funktionalität des Systems gebunden, sondern hat die freie Wahl welche Komponente benutzen werden und welche man selbs implementieren wird.

Abbildungsverzeichnis

Literaturverzeichnis