

OpenSource für Geodateninfrastrukturen – eine echte Alternative?

Martin MAY & Stephan MERTEN & Johannes BRAUNER

Martin May & Stephan Merten & Johannes Brauner: Institut für Geoinformatik der Westf. Wilhelms-Universität Münster, Robert-Koch-Str. 26-28, D-48149 Münster, Email: :mmay@uni-muenster.de, smerten@uni-muenster.de, joejoe@uni-muenster.de

1 EINLEITUNG

Geodateninfrastrukturen (GDI) lassen sich meist nur unter Aufwendung umfangreicher Ressourcen erstellen. Die schlechte finanzielle Situation der öffentlichen Haushalte erfordert die Suche nach kostengünstigen Lösungen. Die praktischen Erfahrungen bei der Umsetzung von Geodateninfrastrukturen im Umfeld von GDI-NRW zeigen, dass kommerzielle Produkte nicht immer das leisten, was ihr Preis erwarten lässt. Beim Aufbau einer Geodateninfrastruktur kann der Einsatz von freier oder OpenSource-Software helfen, Kosten zu sparen. OpenSource bietet außerdem die Möglichkeit, ein Produkt stärker an die individuellen Anforderungen anzupassen.

Im vorliegenden Beitrag wird an Hand des Beispiels FLUMAGIS aus dem Planungsumfeld aufgezeigt, wie sich bestehende weitgehend auf kommerziellen und nicht offenen Produkten basierende GDI-Architekturen auf bestimmte verfügbare Lösungen aus dem OpenSource-Umfeld übertragen lassen. Der Fokus liegt hierbei auf standardisierten Web-Diensten (WMS, WFS, etc.) für Geobasisdaten. Die Umstellung von GDI-Komponenten kann entscheidend davon abhängen, wie interoperabel diese auch auf den verschiedenen System-Ebenen sind. Wichtigste Grundannahme für diesen Beitrag ist, nicht nur einzelne GDI-Komponenten auszuwechseln, sondern eine vollständig OpenSource-basierte Geodateninfrastruktur aufzubauen.

Zunächst werden die verschiedenen Begriffe rund um Geodateninfrastrukturen und OpenSource erläutert. Danach werden die verschiedenen System-Ebenen einer GDI näher beleuchtet und auf den Anwendungsfall des Projektes FLUMAGIS bezogen. Daraufhin werden ausgewählte OpenSource Produkte vorgestellt, die zum Aufbau von Geodateninfrastrukturen eingesetzt werden können. Die praktischen Erfahrungen beim Einsatz und Zusammenspiel dieser OpenSource Software werden im letzten Teil dieses Beitrags beschrieben und im Fazit abschließend bewertet.

2 BEGRIFFE

2.1 Geodateninfrastruktur

Geoinformationen haben eine gesellschaftlich und wirtschaftlich hohe Bedeutung. Typische Einsatzbereiche sind die Entscheidungsunterstützung bei räumlichen Planungen und Entwicklungsmaßnahmen, die Optimierung beim Einsatz natürlicher und ökonomischer Ressourcen sowie die Modellierung komplexer räumlicher Zusammenhänge z.B. bei Hochwasserprognosen (IMAGI 2003). Geoinformationen müssen nicht nur umfassend und aktuell vorliegen, sondern auch für den öffentlichen und privaten Sektor leicht zugänglich sein.

Geodateninfrastrukturen verfolgen das Ziel, den Zugang zu Geoinformationen zu erleichtern. Eine Geodateninfrastruktur bietet „a basis for spatial data discovery, evaluation, and application for users and providers within all levels of government, the commercial sector, the non-profit sector, academia and by citizens in general.“ (NEBERT 2001, S.8)

In verschiedenen nationalen und internationalen Initiativen haben sich Firmen, öffentliche Institutionen und Universitäten zusammengeschlossen, um Geodateninfrastrukturen aufzubauen. Beispiel für eine nationale GDI ist GDI-NRW (<http://www.gdi-nrw.org/>). INSPIRE (Infrastructure for Spatial Information in Europe) ist eine Initiative der europäischen Kommission um eine europäische Geodateninfrastruktur zu fördern (<http://www.ec-gis.org/inspire/>).

Geodateninfrastrukturen bestehen aus verschiedenen Komponenten: Zum einen aus der Datenbasis, die aus den Geobasisdaten und Geofachdaten und den dazugehörigen Metadaten besteht. Zum anderen bilden GI-Dienste und das sie miteinander verbindende Netzwerk die Kernkomponenten einer GDI. Die Dienste ermöglichen beispielsweise den schreibenden und lesenden Zugriff auf die Geodaten. Auch die Suchdienste nach Geodaten oder anderen Diensten kommen typischerweise in einer Geodateninfrastruktur zum Einsatz.

Um eine umfassende kooperative Nutzung der verteilten GI-Dienste zu erreichen, ist der Einsatz von Standards beim Aufbau einer GDI unerlässlich. Wichtigste Grundlage sind Standardisierungen der internationalen Standardisierungs-Organisation (ISO), wie beispielsweise von Metadaten für Geodaten (ISO 19115) und die weltweit anerkannten Schnittstellenspezifikationen für Webdienste des OpenGIS Konsortiums (OGC). Dieses internationale Konsortium aus Unternehmen, Behörden, Verbänden und Forschungsinstituten hat bereits einige Schnittstellenspezifikationen webbasierter Dienste verabschiedet, zu denen bereits Referenzimplementierungen existieren (vgl. Kap 5): Für den Zugriff auf vektorielle Geodaten wurden die Schnittstellen von Web Feature Services (WFS) spezifiziert. Die Schnittstellen von Web Coverage Services (WCS) wurden für den Zugriff auf gerasterte Geodaten festgelegt. Web Map Services (WMS) dienen der kartographischen Visualisierung von Geoinformationen. Weitere OGC Webdienste sind Web Catalog Services, die der Recherche nach Geodaten und GI-Diensten dienen und Web Coordinate Transformation Services durch die eine Transformation zwischen unterschiedlichen Koordinatensystemen durchgeführt werden kann. Außerdem wurde vom OGC die Geography Markup Language (GML) als XML-basierte Beschreibungssprache für Geoobjekte spezifiziert (<http://www.opengis.org/>).

Durch die Standardisierung können diese GI-Dienste zu Dienstketten kombiniert werden, bei denen beispielsweise ein WMS eine Anfrage an weitere Web Map Services weiterschickt und die Ergebniskarte somit aus verschiedenen Quellen durch unterschiedliche Dienste zusammengestellt wird.

Die Vorteile, die eine Geodateninfrastruktur bietet, sind vielfältig. Aus der dezentralen Organisation von Geodaten und GI-Diensten resultiert eine Qualitätssteigerung, da die Daten dort wo sie erfasst wurden gepflegt und Dienste weiterentwickelt werden. Der

erleichterte Zugang bietet eine Effizienzsteigerung im Umgang mit Geoinformationen (BERNARD & STREIT 2002). Die kooperative Nutzung verteilter Geoinformationsdienste ermöglicht die Erschließung des Geodatenmarktes bzw. des Wirtschaftsgutes Geodaten für einen breiten Markt.

2.2 OpenSource – Freie Software

Die Grundidee von OpenSource bzw. freier Software geht auf den Ursprung der Wissenschaft zurück. Traditionell basiert Wissenschaft auf freiem Austausch von Wissen und Gedanken. Angeregt durch diese Idee wurde 1984 die Free Software Foundation (FSF) gegründet, die die Vision von freiem Austausch von Wissen in Form von programmierter Software verfolgt.

Nach Definition der FSF sind vier Basisfreiheiten für freie Software zu unterscheiden (<http://www.fsf.org/philosophy/free-sw.html>):

- Freiheit 0: Die Freiheit, das Programm für jeden Zweck zu nutzen.
- Freiheit 1: Die Freiheit, zu verstehen, wie das Programm funktioniert und wie man es für seine Ansprüche anpassen kann. Dazu ist der Zugang zum Quellcode Voraussetzung.
- Freiheit 2: Die Freiheit, Kopien weiterzuverbreiten, so dass man seinem Nächsten weiterhelfen kann.
- Freiheit 3: Die Freiheit, das Programm zu verbessern und die Verbesserungen der Öffentlichkeit zur Verfügung zu stellen, damit die ganze Gemeinschaft davon profitieren kann. Auch hier ist der Zugang zum Quellcode Voraussetzung.

Der Begriff ‚OpenSource‘ war der Versuch, der OpenSource Initiative (OSI), freier Software einen Namen bzw. einen geschützten Begriff, ein Marketing-Programm zu geben. Die Definition der OSI für OpenSource folgt nahezu deckungsgleich der Definition der FSF. Die Bedeutung von OpenSource geht weit über die Offenlegung des Quelltextes hinaus, dies ist aber essentielle Voraussetzung dafür. (http://www.opensource.org/docs/definition_plain.php)

Um diese Rechte und Freiheiten zu schützen, gibt es eine Vielzahl von Lizenzmodellen, auf die in ihrer Vielgestalt an dieser Stelle nicht eingegangen werden kann. Beispielhaft soll die GNU General Public License (GNU GPL) kurz skizziert werden. GNU ist ein rekursives Akronym für „GNU is Not Unix“. Die GNU GPL gewährt nicht nur die Basisfreiheiten, darüber hinaus schützt sie sie auch. Das bedeutet, dass Software, die anhand von GNU GPL-lizenzierten Komponenten erstellt wird, auch wieder unter der GNU GPL veröffentlicht werden muss. Andere Lizenzmodelle (z.B. die GNU Lesser General Public License (GNU LGPL)) gewähren die Freiheiten, schützen sie aber nicht. (<http://www.fsf.org/licenses/license-list.html>)

Das „frei“ in freier Software bedeutet nicht, dass das Programm kostenlos verbreitet werden muss. Entgegen gebräuchlicher Vorurteile ist es legitim, OpenSource bzw. freie Software zu verkaufen. Der Unterschied zu „closed-source“-Software ist lediglich, dass die Nutzungsbedingungen im Sinne der vier Freiheiten nicht eingeschränkt sind. (<http://www.fsf.org/philosophy/selling.html>)

Freie bzw. offene Schnittstellen und Datenformate müssen unabhängig von freier Software betrachtet werden. Sie dienen nur dem Austausch von Informationen beispielsweise zwischen den verschiedenen Komponenten einer GDI. Auch „closed-source“-Software versteht freie Datenformate und kann mit offenen Schnittstellen kommunizieren. Eine Standardisierung von Schnittstellen und Datenformaten ist nicht direkt Bestandteil der Grundidee von freier Software, unterstützt sie aber maßgeblich. Folgendes Diagramm versucht, bestehende Software (Kreise) anhand der verschiedenen Begrifflichkeiten in den verschiedenen Richtungen zu kategorisieren.

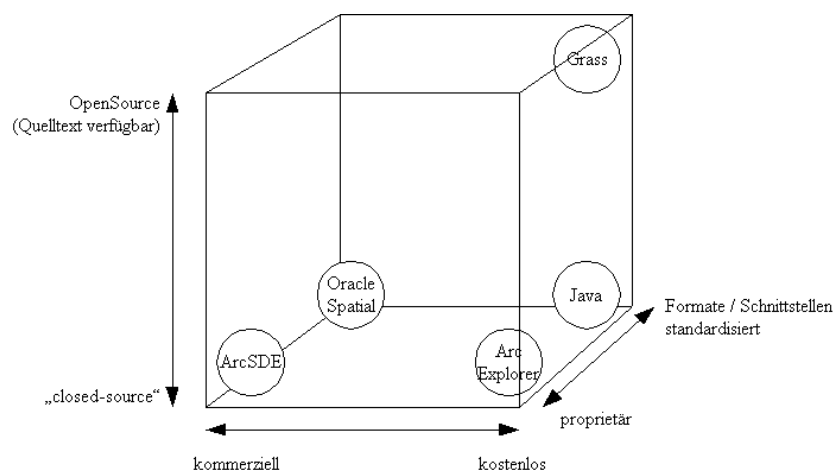


Abb. 13: Software und ihre Zuordnung zu häufig verwendeten Begrifflichkeiten (Quelle: Eigene Darstellung)

Der Vorteil der Verwendung von OpenSource bzw. freier Software liegt in der größeren Unabhängigkeit von einem Softwarehersteller und dessen Produkt-Spektrum, zum Beispiel wenn dieser seine Produkte nur so verkauft, dass aufgrund von Nichtoffenlegung von Schnittstellen und Quelltext eine freie Kombination von verschiedenen verfügbaren Komponenten nicht möglich ist, sondern der Kunde auf seine Produkte angewiesen ist. Ein Nachteil der Verwendung von OpenSource bzw. freier Software kann sein, dass der Benutzer, da er nicht bezahlt hat, und so wenn nicht selbst an der Entwicklung beteiligt ist, keine einklagbaren Rechte bezüglich erwarteter Funktionalität oder Entwicklungsgeschwindigkeit hat. Dies wird jedoch inzwischen in den meisten Projekten im Vorhinein festgelegt und dokumentiert.

3 SYSTEM-EBENEN EINER GDI

Für das allgemeine Verständnis sei an dieser Stelle erklärt, dass mit dem Begriff des Dienstes in diesem Artikel nicht die spezialisierten Web- oder GI-Dienste gemeint sind, sondern zunächst einmal funktionale Einheiten, die anhand bestimmter Eingabedaten und Anweisungen ein erwünschtes Ergebnis erstellen können. Wie dies geschieht, ist zunächst nicht näher festgelegt – im Zweifelsfall wäre beispielsweise auch die manuelle Aufbereitung von Eingangsdaten in das gewünschte Ergebnis ein Dienst – der jedoch gewiss mit Verzögerung arbeiten würde.

Die Aufgabe von Geodateninfrastrukturen ist, Geodaten verschiedener Art und Herkunft in Verbindung mit bestimmten Funktionalitäten verfügbar zu machen. Dabei können auch Eigentum und Nutzungsrechte eine große Bedeutung haben. Die Verteilung solcher Daten kann die Wartung vereinfachen, Aktualität erhöhen, Redundanzen verhindern und kooperatives Arbeiten erleichtern. Damit dies funktioniert, müssen die jeweiligen Teildatenbestände in den Teilen der GDI syntaktisch und semantisch korrekt interpretiert werden – wozu die Spezifikation anhand von Schnittstellen dient. Größere kommerzielle Anbieter von entsprechender Software integrieren diese häufig in ihre Software und liefern „application programming interfaces“ (APIs) mit, die dem Software-Entwickler die Verwendung der Schnittstellen so einfach wie möglich machen soll. Die Schnittstelle selbst tritt dabei u.U. sehr in den Hintergrund. Allen Web-Diensten gemein ist wohl, dass sie auf grundlegende Protokolle zur Übermittlung von Daten-Paketen zurückgreifen, wie z.B. TCP/IP. Darüber hinaus ist in einigen Fällen wenig Information zu erhalten, wie die Kodierung erfolgt (z.B. Klartext oder binär) und welche Syntax verwendet wird – d.h. es handelt sich um eine nicht offen gelegte Schnittstelle. Einige Hersteller legen zwar ihre Schnittstellen offen, allerdings wird die Wartbarkeit von darauf basierenden Komponenten nicht in jedem Falle erhöht, da Veränderungen der Schnittstelle im jeweiligen Anwendungsfall ungeahnte Veränderungen bei den abhängigen zusätzlichen Komponenten der Benutzer nach sich ziehen können, jedoch ausgehend vom Hersteller erfolgen.

„Hinter“ solchen Schnittstellen können sich beliebige Strukturen befinden, um den GI-Dienst zu realisieren. Im klassischen Falle ist dies eine mehrschichtige Web-Applikation, die wiederum verteilt arbeiten kann und selbst aus Diensten zusammengesetzt ist. So wird man hier eine Datenbank ggf. mit einer Erweiterung um räumliche Operatoren finden, die die Daten und eingeschränkte GIS-Funktionalität vorhält. Ein Applikations-Server sorgt für den Zugriff auf die Datenbank, integriert Benutzerrechte und stellt u.U. mehrere Dienste in seiner Umgebung zur Verfügung und sorgt für ihre Verbindung.

Um den Begriff des Geo-Dienstes zu fassen, wird die Beschreibung aus einer bestehenden größeren GDI herangezogen: GDI-NRW. Im Testbed 1 wurde dort der folgende Rahmen angeführt:

„Allen GI-Services in GDI ist gemein, daß sie

- über einen definierten Raumbezug verfügen
- über einen Mindestsatz an Metadaten beschrieben werden
- über das Internet (HTTP) aufgerufen werden
- geringe Anforderungen an die technische Ausstattung der GI-Nutzer stellen
- geringe Einstiegshürde für Anbieter und Nutzer von GI-Services

GI-Services implementieren die ihrem ServiceTyp entsprechende und im Testbed spezifizierte Service-Schnittstelle. Ein GI-Client nutzt einen oder mehrere GI-Services. Eine GI-Applikation steht dem Endnutzer zur Verfügung und ist ein (komplexer) GI-Client am Ende einer Kette von GI-Services. Der Zugriff auf eine GI-Applikation erfolgt via einer URL als Schnittstelle.“ (Testbed1).

Eine wichtige Eigenschaft von GI-Diensten ist darüber hinaus ihre Beschreibbarkeit in Form von Metadaten, wie es die folgende Abbildung verdeutlicht:

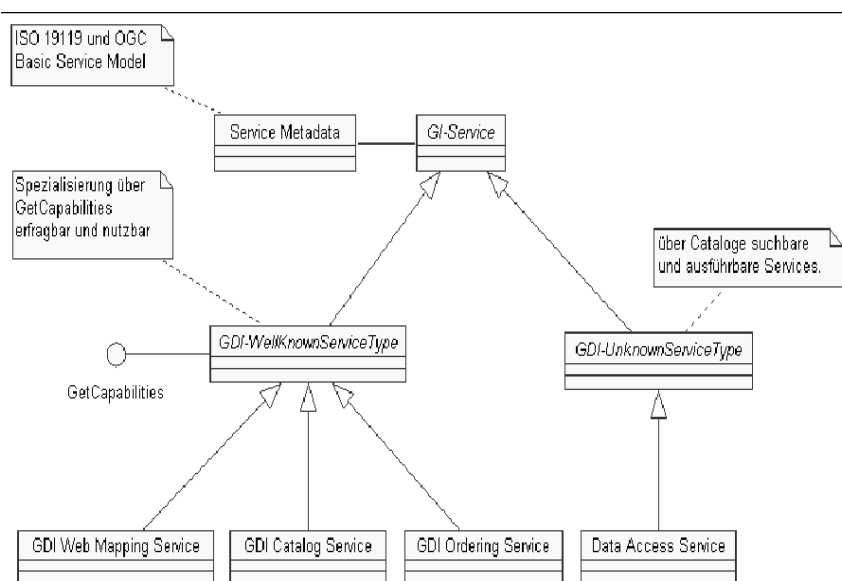


Abb. 2: Klassendiagramm GI-Dienste (Quelle: TestbedI)

In diesem Beispiel wird angedeutet, dass ein GI-Dienst anhand einer Menge von Eigenschaften und Methoden beschrieben werden kann. Die Zusammenhänge zwischen verschiedenen Diensten müssen nicht notwendiger Weise in einer Klassen-Hierarchie

abgebildet werden. Ein Dienst kann im Allgemeinen beliebige Aktionen ausführen, um eine zu ihm passende Anfrage entsprechend seiner Spezifikation zu erfüllen.

Die Anfrage an einen Dienst wird von einem Client gestellt. Die Antwort muss nicht notwendiger Weise auch wieder an diesen zurückgeliefert werden. Ein Dienst kann selbst gleichzeitig ein Client sein, wenn er für die Bearbeitung von Anfragen auf andere Dienste zugreifen kann.

Der klassische Fall einer Webmapping-Applikation wird in etwa wie folgt zusammengesetzt sein:

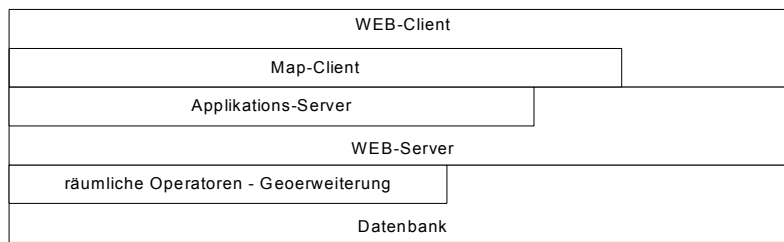


Abb. 3: Zusammensetzung einer Web-Applikation (Quelle: Eigene Darstellung)

4 MACHBARKEIT EINER GDI ANHAND EINES BEISPIELS AUS DEM PLANUNGSUMFELD (FLUSSEINZUGSGEBIETS-MANAGEMENT)

FLUMAGIS ist ein Forschungsprojekt des BMBF, das ein spezialisiertes planungs-unterstützendes GIS entwickelt. Es handelt sich um ein „spatial decision support system“ (SDSS), das anhand einer Wissensbasis Expertenwissen zum experimentellen Planen interaktiv nutzbar macht. Dies orientiert sich weitgehend an den Vorgaben der EG-Wasserrahmenrichtlinie (WRRL) der Europäischen Union.

FLUMAGIS stellt den klassischen Fall eines Systems dar, das im planerischen Umfeld Entscheidungen unterstützen soll. Es basiert auf Geodaten, die anhand ihrer Geometrie, Topologie und Semantik hinsichtlich spezifischer Fragestellungen untersucht werden. Insbesondere werden spezialisierte Ursache-Wirkungs-Zusammenhänge in einer Wissensbasis modelliert, die dazu auf bestimmte weitgehend im Vorhinein als Funktionskomplexe verkettete GIS-Funktionen zurückgreift, die von einer zentralen Stelle des Systems (Controller) zur Ausführung gelangen.

Diese System-Architektur enthält somit Komponenten, die über eine klassische GDI hinaus gehen – die Funktion einer Wissensbasis gehört nicht direkt dieser Software-Domäne an. Wir betrachten also hier die Komponenten Datenbank, Geo-Erweiterung der Datenbank, Geo-Services bzw. Schnittstellen und GIS als GDI, auf die durch den Controller bzw. dahinter liegende Komponenten zugegriffen wird.

Da es sich bei FLUMAGIS um einen Prototypen handelt, sind bestimmte Fragestellungen bei der Ausführung von Abfragen selbst Gegenstand der Forschung. Dies gilt beispielsweise für die Koppelung mit der Wissensbasis und die Einbindung diverser Modelle. Allerdings sind grundlegende Funktionen zu unterscheiden, die erforscht und praktisch direkt einsetzbar sind.

Beispielsweise ist ein Funktionskomplex die Selektion des Untersuchungsgebietes, bei der ein bestimmter Bereich als Arbeitsgrundlage ausgewählt wird. Fachliche Zusammenhänge werden hierbei darüber entscheiden, wie dies für verschiedene Geodatentypen vollzogen werden muss. Beispielsweise kann für linienhafte oder flächige Geoobjekte teilweise eine Verschneidung notwendig sein – teilweise jedoch auch die Auswahl des vollständigen Objektes. Am Ende dieses Vorgangs steht eine Sammlung von Geoobjekten, die im Zustand ‚ausgewählt‘ befindlich sind. Dieser kann entweder durch eine Arbeits-Kopie oder über zusätzliche Felder in den Original-Daten abgebildet werden. Beide Varianten verursachen Probleme: Die Abbildung über zusätzliche Felder kann beliebig komplex werden und die Verarbeitungsgeschwindigkeit drastisch senken. Die Schaffung einer Arbeitskopie schafft ein Redundanz-Problem, das bei der späteren Abspeicherung des Arbeitsstandes berücksichtigt werden muss.

Unabhängig davon wird jedoch deutlich, dass der Transfer der Geodaten durchaus bidirektional vollzogen werden muss. Proprietäre kommerzielle GIS- bzw. GDI-Systeme realisieren dies über transaktionale Dienste, die meist direkt Bestandteil der Kommunikations-Schicht einer Applikation sind. Es ist zu berücksichtigen, dass diese meist nicht oder nur beschränkt offenen Schnittstellen-Spezifikationen folgen. Die Ab- oder Rückspeicherung der Daten schließt die Arbeiten des Benutzers ab und ist wie die im vorherigen Absatz beschriebene Selektion zu Beginn als vorhandene Funktionalität anzusehen, die in geeigneter Weise eingebunden werden muss.

Bestimmte weitere Funktionen lassen sich ohne Weiteres mit den ausgewählten Daten aus einem bestimmten Bereich durchführen, die jedoch in FLUMAGIS einer direkten Plausibilitätsüberprüfung durch die Wissensbasis unterliegen. Beispielsweise kleinräumige Abfragen, wie: Gibt es einen Nutzungskonflikt zwischen zwei benachbarten Nutzungen, etc. sind so bis auf wenige Ausnahmen am Rande des ausgewählten Gebietes durchführbar. Andere Abfragen, wie die Ursachen-Suche entlang dem Flussverlauf aufwärts werden unabhängig von der Größe des Auswahlgebietes immer dessen Grenze erreichen – es sei denn, man wählt das vollständige Gewässer aus, was jedoch aufgrund der Vorgaben durch die Wasser-Rahmen-Richtlinie nur die Arbeit mit der kleinmaßstäbigen Variante darstellt und somit nicht generell möglich ist. Welche Untersuchungen aus Experten-Sicht dazu notwendig sind, ist in der Wissensbasis modelliert und zunächst einmal nicht Gegenstand dieses Dokuments.

Betrachten wir die benötigten GIS-Funktionen darauf hin, welche auf dem Client und welche auf dem Server benötigt werden, so ergibt sich die folgende Aufteilung:

Client	Server
Kartendarstellung Auswahlbox Attributänderung Objekt geometrisch ändern Feature-Klasse neues Objekt hinzu fügen Analyse-Funktionen	Erzeugung eines Vorschlagslayers Erzeugung eines Ergebnislayers Datenhaltung Ausführung von komplexen Abfragen Erzeugung eines Projekt-Datensatzes Versionierung

Tab. 1: Aufteilung von GIS-Funktionalitäten auf Client- und Serverseite (Quelle: Eigene Darstellung)

Die System-Architektur von FLUMAGIS ist auf die folgenden vereinfacht dargestellten Datenflüsse zugeschnitten.

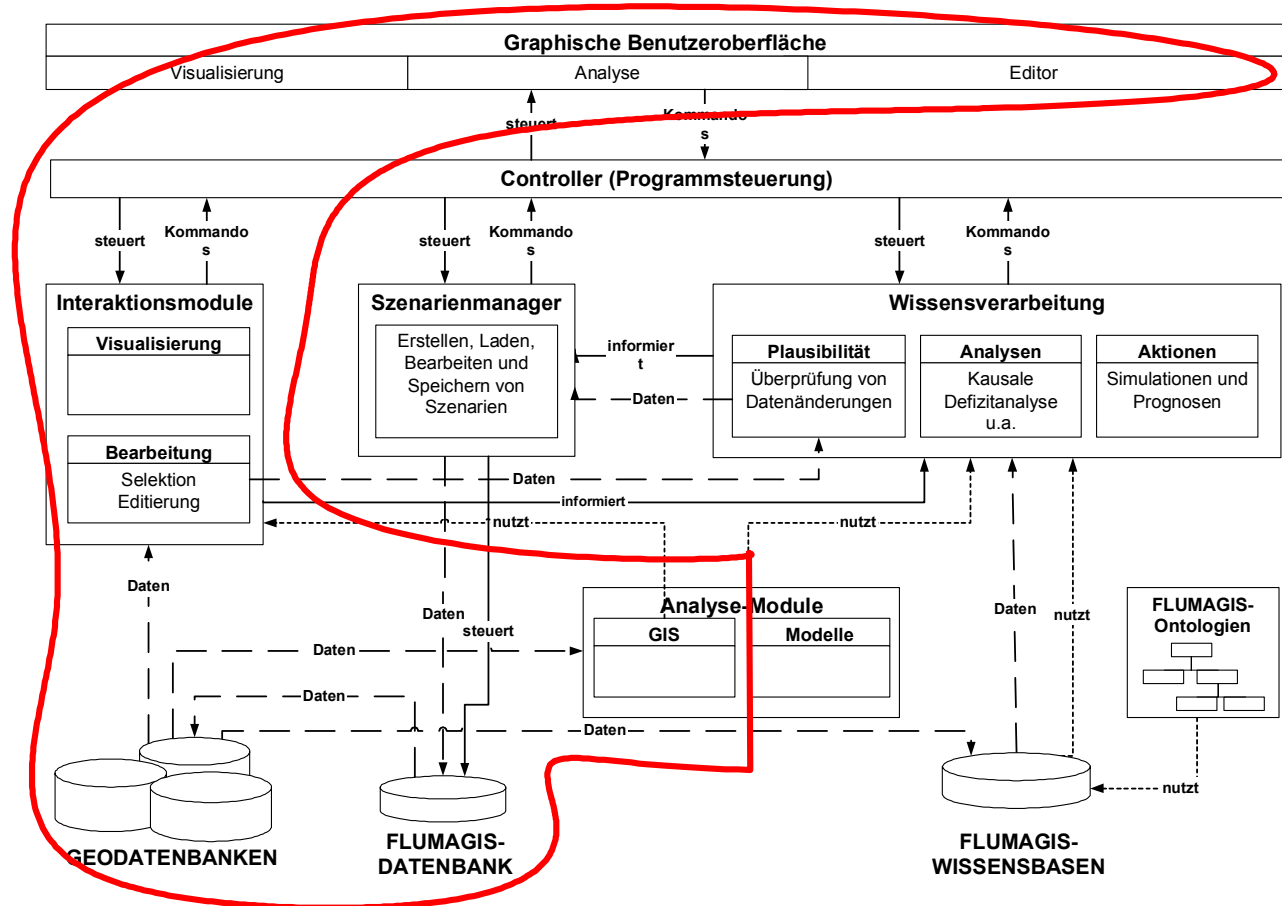


Abb. 4: Systemarchitektur von FLUMAGIS (Quelle: Eigene Darstellung)

Die aufgelegte Linie verdeutlicht, welche Teile des Systems für die hier angestellten Betrachtungen relevant sind und im weitesten Sinne mit den Systemebenen aus dem vorhergehenden Kapitel zur Deckung zu bringen sind. Dies bedeutet, dass die Anbindung der weiteren Komponenten zu einem späteren Zeitpunkt separat betrachtet werden muss.

5 AUSGEWÄHLTE OPENSOURCE PRODUKTE FÜR GEODATENINFRASTRUKTUREN

Der Aufbau einer Geodateninfrastruktur erfordert den Einsatz unterschiedlicher Softwarekomponenten. Zur Speicherung und Verwaltung der Geodaten wird eine Datenbanksoftware benötigt, die räumliche Daten verarbeiten kann. Für den Zugriff auf die Daten werden Applikationen oder Webdienste benötigt. Für letzte ist zusätzlich der Einsatz eines Webservers notwendig. Diese Softwarekomponenten erfordern eine leistungsfähige Serverhardware mit einem leistungsfähigen Betriebssystem.

Nachfolgend soll eine Übersicht über ausgewählte OpenSource Produkte gegeben werden, durch die die oben genannten Anforderungen erfüllt werden können:

OpenSource-Software	Version	Kategorie	Internet-Ressourcen
PostgreSQL	7.4	Datenbank	http://www.postgresql.org/
PostGIS	0.8	räumliche Erweiterung für PostgreSQL	http://postgis.refractions.net/
UMN MapServer	4.0	OpenGIS Map Service	http://mapserver.gis.umn.edu/
Deegree		OpenGIS Web Services	http://deegree.sourceforge.net/
Geoserver	1.0.1	OpenGIS Web Feature Service	http://geoserver.sourceforge.net
Grass	5.0.3	Geoinformationssystem	http://grass.itc.it/

Tab. 2: ausgewählte OpenSource Produkte, Stand: Dez. 2003 (Quelle: Eigene Darstellung)

5.1 PostgreSQL / PostGIS

PostgreSQL ist ein objekt-relationales Datenbankmanagementsystem (ORDBMS). Es basiert auf POSTGRES, das im Rahmen eines universitären Projektes an der University of California am Berkeley Computer Science Department entstanden ist. PostgreSQL wird als OpenSource Projekt von einer Vielzahl von Entwicklern in der ganzen Welt ständig weiterentwickelt und kann kostenlos genutzt werden.

PostgreSQL ist eine der am weitesten entwickelten OpenSource Datenbanken. Es werden die Standards SQL92 und SQL99 unterstützt und fortgeschrittene Operationen angeboten wie Subselects, Mengenoperationen, Joins, Views, Trigger, Regeln sowie ein Transaktionsmanagement. PostgreSQL lässt sich um eigene Datentypen, Operationen und Funktionen erweitern und bietet eine beachtliche Menge von Schnittstellen an. Durch die objektrelationalen Eigenschaften lassen sich nichtatomare Datentypen in Arrays oder in geometrischen Datentypen speichern. (<http://www.postgresql.de/>)

Für die Verarbeitung räumlicher Daten in Form von geographischen Objekten dient die PostgreSQL-Erweiterung PostGIS (<http://postgis.refractions.net/>). PostGIS wird von der Firma Refractions Research als ein Forschungsprojekt im Bereich OpenSource Datenbanktechnologie entwickelt. In der Version 0.8 unterstützt es alle Funktionen der OpenGIS "Simple Features for SQL" Spezifikationen (OGC 1999). PostGIS bietet die Möglichkeit PostgreSQL an Geoinformationssysteme anzubinden und räumliche Daten in PostgreSQL zu speichern.

5.2 OpenSource Web Dienste

5.2.1 UMN MapServer

Der MapServer wird im Rahmen von Forschungsprojekten der University of Minnesota (UMN) in Kooperation mit der NASA und anderen Partnern entwickelt und ist eine OpenSource-Paket für Internetanwendungen mit GIS Funktionalitäten (<http://mapserver.gis.umn.edu/>). Der MapServer ist für den Einsatz unter UNIX oder UNIX-Derivaten in Kombination mit dem Apache Web Server konzipiert, er läuft aber auch unter Windows. Auf die Funktionalität kann über verschiedene Sprachen wie Java, Perl, PHP, Python etc. zugegriffen werden.

Neben den Entwicklungsmöglichkeiten kann der MapServer auch in Form einer CGI-Anwendung genutzt werden, die einige GIS-Funktionalitäten für das Internet zur Verfügung stellt. Dazu gehört die Unterstützung verschiedener Vektor- und Rasterformate, Selektionsmöglichkeiten, automatisches Generieren der Legende und des Maßstabsbalkens, thematische Kartengenerierung mit Klassifikation durch logische und reguläre Ausdrücke, Beschriftung mit automatischer Ausrichtung und on-the-fly Projektion. Der MapServer unterstützt auch die WMS und WFS Schnittstellenspezifikationen des OGC.

Für den UMN MapServer gibt es eine Reihe weiterer OpenSource Werkzeuge, wie beispielsweise MapLabs (<http://www.maptools.org/maplab/>). Diese Webanwendung ist eine Sammlung von Werkzeugen durch die das Erstellen und Konfigurieren von MapServer Webmapping Anwendungen vereinfacht wird.

5.2.2 Deegree

Im OpenSource Projekt Deegree der Firma Lat/Lon werden wesentliche Webdienste für den Aufbau einer Geodateninfrastruktur entwickelt. Dabei werden konsequent die Standards vom OpenGIS Konsortium und von ISO/TC 211 umgesetzt. Deegree beinhaltet folgenden OGC konforme Webdienste: Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), Web Catalog Service, Web Gazetteer Service, Web Terrain Service (WTS), Web Coordinate Transformation Service (WCTS).

Alle Webdienste werden komplett in Java implementiert und sind damit plattformunabhängig auf diversen Betriebssystemen einsetzbar. Es werden für den lesenden und schreibenden Zugriff Schnittstellen zu Datenbanken wie ORACLE Spatial, PostgreSQL/PostGIS, allgemein JDBC-fähige Datenbanken, ESRI Shapefiles und Rasterdatenformate unterstützt. Andere OGC konforme Dienste wie beispielsweise andere Web Map Server lassen sich auch über Dienstketten einbinden (<http://deegree.sourceforge.net/>).

5.2.3 Geoserver

Geoserver ist ein transaktionaler Web Feature Service der nach den OpenGIS Web Feature Server Spezifikationen 1.0 entwickelt wird (<http://geoserver.sourceforge.net/>). Er ermöglicht den standardisierten Zugriff und die Manipulation von geographischen Daten. Als Datengrundlage kann der Geoserver in der Version 1.0.1 auf PostgreSQL in Kombination mit PostGIS, auf Oracle Datenbanken

sowie auf ESRI Shape-Dateien zugreifen und liefert geographische Features in GML 2.1.2. Geoserver nutzt die OpenSource Bibliothek GeoTools (<http://geotools.org/>) und wird in Java entwickelt. Er kann somit plattformunabhängig eingesetzt werden.

5.3 GRASS GIS

GRASS (Geographic Resources Analysis Support System) ist ein OpenSource Geoinformationssystem. Es bietet umfangreiche GIS-Funktionalitäten für die Verarbeitung von Raster-, Vektor- und Volumendaten (3D-Voxel). Der Funktionsumfang reicht von Raster- und Vektoranalysen über Bildverarbeitung und Visualisierung bis hin zu Simulationen. GRASS verfügt über eine Datenbank-Anbindung an Oracle und PostgreSQL und kann falls eine Lizenz für die ESRI-SDE-Client-API vorliegt, auch auf die SDE von ESRI zugreifen (frühes Stadium der Implementierung). Neben dem Einsatz auf Unix und Linux ist GRASS auch für Windows und PDAs mit Linuxbetriebssystem verfügbar. GRASS ist modular aufgebaut, so dass beliebige Entwickler unterschiedliche Module entwickeln und integrieren können.

6 UMSETZUNG DER GDI

6.1 Installation der Software

Als Hardware-Plattform wurde eine SUN Ultra-Sparc 1 mit zwei 200 MHz 64-Bit Prozessoren verwendet. Das System verfügt über 512 MB Hauptspeicher und 20 GB SCSI-II Festplatten und ist mit 100 Mbit in das lokale Netzwerk eingebunden.

Als Betriebssystem wurde die Linux-Meta-Distribution Gentoo 1.4 verwendet. Gentoo erlaubt ein vollständiges vollautomatisches Kompilieren, sodass sämtliche System-Komponenten an die Hardware-Fähigkeiten angepasst werden können. Dies ist für eine Vielzahl von Plattformen möglich und beliebig konfigurierbar (<http://www.gentoo.org/>). Als Kernel kommt der 64-Bit-fähige Linux-Kernel 2.4.21 zum Einsatz. Da Gentoo zur Installation hochgradig aktuelle Source-Versionen direkt von Gentoo-Servern verwenden kann und installierte ältere Versionen auf Wunsch automatisch aktualisiert, gestaltet sich die weitere Wartung des Systems äußerst komfortabel und die Installation der weiteren benötigten Software muss nicht durch mühseliges und zeitaufwändiges manuelles Einspielen von gerade benötigten Versions-Nummern vorbereitet werden.

Auch alle weiteren Software-Komponenten wurden bei der Installation kompiliert – bis auf die virtuelle Maschine für Java, die zwar kostenlos ist, jedoch in den wesentlichen Teilen nicht im Quelltext vorliegt. Dies gilt für die von SUN entwickelten Versionen und für die von Blackdown für Linux.

Als Datenbank kommt die objekt-relationale Datenbank PostgreSQL in der Version 7.3.4 zum Einsatz, die auf einige standardmäßig vorhandene System-Bibliotheken angewiesen und um Schnittstellen zu Perl, Java und verschlüsselte Datenübertragung optional erweiterbar ist.

Die Datenbank wurde mit der Geo-Erweiterung PostGIS 0.8.0 ausgestattet, die die Datenbank PostgreSQL und die Bibliothek *proj* erfordert. *proj* macht kartographische Projektionen und zugehörige Werkzeuge verfügbar (<http://www.remotesensing.org/proj/>).

Web- und Applikationsserver ist Tomcat 4.1.4-r1 (<http://jakarta.apache.org/>), der das Vorhandensein einer Java virtuellen Maschine erfordert. Das Blackdown Java Development Kit 1.4.1 wurde bereits mit dem System installiert.

Als Web Feature Service kommt Geoserver 1.0.1 zum Einsatz. Geoserver ist eine java-basierte Serverapplikation, die als Web-Applikation in entsprechenden Applikations-Server-Umgebungen verwendet werden kann (hier Tomcat).

Für die Installation des Web Map Services UMN MapServer sind folgende Bibliotheken nachzuinstallieren:

- *xerces-c* (XML-Parser)
- *gdal* (Bibliothek zur Rasterdatenverarbeitung)
- *ogr* (Bibliothek zur Vektordatenverarbeitung)
- *proj* (s.o.)

Tomcat wurde für den UMN Mapserver zusätzlich als cgi-Umgebung konfiguriert.

6.2 Funktionstest/Probleme

Die Installation anhand mitgelieferter bzw. online verfügbarer Anleitungen der verwendeten Komponenten verlief weitestgehend problemlos. Bei der Installation der Bibliothek *gdal* (s.o.) war es nicht möglich, die neueste Version 2.4.0 des XML-Parsers *xerces-c* zu verwenden. Es musste die ältere Version 1.6.0 installiert werden, was der Mailing-Liste zu entnehmen war. Die Verbindung der Layer des Geoservers auf die Datenbanklayer war aufgrund knapper Dokumentation aufwändig und wurde erst nach längerem Ausprobieren abgeschlossen. Kleinere administrative Probleme ergaben sich bei der Verteilung der verschiedenen Nutzerrechte.

Als Beispieldatensätze wurden verschiedene Punkt-, Linien- und Polygonlayer verwendet. Mit dem PostGIS-Skript *shp2pgsql* wurden die Layerdaten aus dem ESRI-Shapefile Format (offenes Format) in die Datenbank eingepflegt. Vorsicht war bei der Wahl des entsprechenden Referenzsystems geboten. Die PostGIS-internen Referenzsystemcodierungen nach der European Petroleum Survey Group (EPSG) waren nicht auf dem neusten Stand (EPSG 2003) und führten zu falscher Referenzierung beim Auslesen durch den Geoserver. Da die Codierungen Teil der Datenbank sind, konnten sie ohne Probleme angepasst werden, da die entsprechenden Referenzparameter bekannt sind.

Das Auslesen der Layer aus der Datenbank durch die WFS-Komponente (Geoserver) funktionierte ansonsten einwandfrei. Die Konformität der Anfrage und des Datenformats nach OGC WFS Spezifikation (OGC 2002) war gegeben. Auch räumlich begrenzte Abfragen waren möglich. Die bisher selten implementierte Transaktionalität wurde durch manuelles Einfügen eines Layers durch den

Geoserver in die Datenbank gezeigt. Die Kommunikation von WFS und WMS (Mapserver) über die durch WFS standardisierte Schnittstelle verlief problemlos. Die Konformität beider Komponenten zur Simple Feature Specification (OGC 1999) ist gegeben.

7 FAZIT

Mit Linux-Kenntnissen und Erfahrungen im Geo-Software Umfeld, beispielsweise dem Gespann ArcSDE/OracleSpatial/etc., und mit geringem Zeitaufwand (ca. drei Tage) wurde eine lauffähige und in ihren Ansprüchen als GDI voll funktionale Systemumgebung geschaffen. Ein Leistungs- bzw. Lasttest steht noch aus. Das System machte trotz der veralteten Hardware subjektiv einen durchaus performanten Eindruck.

Bei der Zusammenstellung der Komponenten wurde aus einer Vielfalt von Produkten gewählt. Für die Zukunft wäre ein vergleichender Test verschiedener Lösungen für die einzelnen Komponenten hilfreich. Darüber hinaus wurde festgestellt, dass die Entwickler dieser Programme sehr aktiv sind (im Durchschnitt einmal pro Monat ein abwärtskompatibles Update) und gute Dokumentationen und viel benutzte Mailinglisten als Recherche- und Anfragemöglichkeit bei Problemen und Supportwünschen vorhanden sind. Speziell für die PostGIS Komponente muss gesagt werden, dass nur ein Bruchteil der Funktionalität untersucht wurde und diese Geodatenbankerweiterung Gegenstand weiterer Forschungsarbeiten sein wird.

Abschließend ist festzustellen, dass eine aus OpenSource bzw. aus freier Software zusammengestellte GDI im beschriebenen Umfang und weitgehend unter Einhaltung bzw. Verwendung von Standards machbar ist. Eine „Alltagstauglichkeit“ kann allerdings nur durch Last- und Leistungstest jeder einzelnen Komponente und durch eine Vielzahl von Anwendern und Zugriffen unter Beweis gestellt werden. Die Standard-Konformität wurde nur stichproben-artig untersucht, was nicht zuletzt durch die Vielzahl der Standards im Rahmen dieser Arbeit nicht anders möglich war.

Die Autoren waren angesichts der Vielzahl der entwickelten und aufeinander aufbauenden Programme und Bibliotheken aus den verschiedensten Hersteller-Kreisen überrascht, dass eine so reibungslose Inbetriebnahme möglich war. Es war sehr schnell möglich, einen ungewöhnlich detaillierten Eindruck vom Zustand einzelner Komponenten zu gewinnen. Die teilweise schon sehr langlebigen Software-Produkte belegen zusätzlich die Verlässlichkeit, die trotz sehr heterogener und räumlich weit verstreuter Entwickler-Gruppen erreicht wird.

8 LITERATUR

BERNARD, L., STREIT, U. (2002): Geodateninfrastrukturen und Geoinformationsdienste: Aktueller Stand und Forschungsprobleme. In:

Publikationen der Deutschen Gesellschaft für Photogrammetrie und Fernerkundung (Bd 11), S. 11-20

EUROPÄISCHES PARLAMENT UND RAT DER EUROPÄISCHEN UNION (2000): EG-WRRL

EPSG (2003): EPSG Geodesy Parameters Version 6.4 , 15 October 2003 (<http://www.epsg.org/>)

GDI-NRW (2002): TestbedI, (<http://www.gdi-nrw.org/>)

IMAGI (2003): Geoinformation und moderner Staat. Frankfurt am Main

NEBERT, D. (Hrsg.) (2001): Developing Spatial Data Infrastructures: The SDI Cookbook. Version 1.1, GSDI

OGC (1999): OpenGIS Simple Feature Specification for SQL, Revision 1.1, May 1999 (<http://www.opengis.org/docs/99-049.pdf>)

OGC (2002): OpenGIS Web Feature Service Implementation Specification, Revision 1.0, May 2002 (<http://www.opengis.org/docs/02-058.pdf>)