

**Wissenschaftliche Berichte
FZKA 6000**

UIS

Baden-Württemberg

Projekt GLOBUS

Umsetzung der neuen

Systemarchitektur und Entwicklung

weiterer Produktionssysteme für

globale Umweltsachdaten im

Umweltinformationssystem

Baden-Württemberg

Phase IV 1997

R. Mayer-Föll, A. Jaeschke (Hrsg.)

Ministerium für Umwelt und Verkehr Baden-Württemberg

Forschungszentrum Karlsruhe

Institut für Angewandte Informatik

Dezember 1997

Forschungszentrum Karlsruhe

Technik und Umwelt

Wissenschaftliche Berichte

FZKA 6000

Projekt GLOBUS

**Umsetzung der neuen Systemarchitektur und Entwicklung
weiterer Produktionssysteme für globale Umweltsachdaten im
Umweltinformationssystem Baden-Württemberg
Phase IV 1997**

R. Mayer-Föll, A. Jaeschke (Hrsg.)

Ministerium für Umwelt und Verkehr Baden-Württemberg

Forschungszentrum Karlsruhe
Institut für Angewandte Informatik

Forschungszentrum Karlsruhe GmbH, Karlsruhe

1997

Für diesen Bericht behalten sich
das Ministerium für Umwelt und Verkehr Baden-Württemberg
Postfach 103439, 70029 Stuttgart
und
das Forschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe
alle Rechte vor.

Druck und Vertrieb
Forschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe
ISSN

Projekt GLOBUS

**Umsetzung der neuen Systemarchitektur und Entwicklung
weiterer Produktionssysteme für globale Umweltsachdaten
im Umweltinformationssystem Baden-Württemberg**

Phase IV - 1997

Projektträger:

**Ministerium für Umwelt und Verkehr Baden-Württemberg (UVM);
I. Henning, A. Koch, R. Mayer-Föll**

**Landesanstalt für Umweltschutz Baden-Württemberg (LfU);
A. Keitel, M. Müller, E. Schmid, H. Spandl**

Projektpartner:

**Forschungszentrum Informatik
an der Universität Karlsruhe (FZI);
A. Koschel, R. Kramer, R. Nikolai, C. Rolker**

**Forschungsinstitut für anwendungsorientierte Wissensverarbeitung
an der Universität Ulm (FAW);
W. F. Riekert, G. Wiest**

**Forschungszentrum Karlsruhe - Technik und Umwelt (FZK);
W. Geiger, A. Jaeschke, R. Weidemann**

**Institut für Kernenergetik und Energiesysteme
der Universität Stuttgart (IKE);
F. Schmidt**

**Institut für Photogrammetrie und Fernerkundung
der Universität Karlsruhe (IPF);
J. Wiesel**

Vorwort

Das Umweltinformationssystem (UIS) Baden-Württemberg ist der Rahmen für die Bereitstellung von Umweltdaten und die Bearbeitung von fachbezogenen und fachübergreifenden Aufgaben im Umweltbereich. Das Ministerium für Umwelt und Verkehr Baden-Württemberg baut gemeinsam mit anderen Ministerien und der Stabsstelle für Verwaltungsreform im Innenministerium dieses ressortübergreifende Umweltinformationssystem als Teil des Landessystemkonzepts Baden-Württemberg bedarfsorientiert weiter aus. Die Landesanstalt für Umweltschutz (LfU) entwickelt und betreibt wesentliche UIS-Komponenten.

Auch die Umweltinformatik muß ihren Beitrag auf dem Weg in die Informationsgesellschaft leisten. Multimedia und Internet haben das Experimentierstadium bereits weit hinter sich gelassen. Neue IuK-Standards, neue Softwaretechnologie und das Dienstekonzept bedürfen der Erprobung in der Praxis. Insbesondere in Zeiten knapper Ressourcen ist der zeitnahe Technologietransfer von Forschungsergebnissen in den täglichen Betrieb sehr wichtig.

Aus diesem Grund führte das Ministerium für Umwelt und Verkehr das Forschungs- und Entwicklungsprojekt GLOBUS - Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg, Phase IV 1997 - durch. Es beauftragte hiermit das bewährte Konsortium folgender Institute:

- Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW) an der Universität Ulm,
- Forschungszentrum Informatik (FZI) an der Universität Karlsruhe,
- Forschungszentrum Karlsruhe - Technik und Umwelt - Institut für Angewandte Informatik (FZK/IAI)
- Institut für Kernenergetik und Energiesysteme (IKE) der Universität Stuttgart,
- Institut für Photogrammetrie und Fernerkundung (IPF) der Universität Karlsruhe.

Im Rahmen von GLOBUS IV wurden auf der Basis der neuen, im Verlauf der vorhergehenden Projektphasen erarbeiteten Systemarchitektur weitere Lösungen entwickelt und in der Landesanstalt für Umweltschutz und dem Ministerium für Umwelt und Verkehr eingeführt.

UVM und LfU brachten Ergebnisse aus GLOBUS IV unter anderem ein in:

- den IuK-Verbund Land/Kommunen am Beispiel des Informationssystems Wasser, Abfall, Altlasten, Boden (WAABIS),
- die Erstellung eines neuen IuK-Modells zur Fortschreibung des Landessystemkonzepts Baden-Württemberg,
- das gemeinsame F+E-Projekt Hypermediatechnik für Umweltdaten (HUDA) mit Umweltbundesamt und Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit,
- den Bund/Länder-Arbeitskreis Umweltinformationssysteme (BLAK UIS) der Umweltministerkonferenz.

Einen Teil der Arbeiten stellten UVM, LfU und die GLOBUS-Partner 1997 im Rahmen der Fachmesse Multi-Media Market in Stuttgart, beim Symposium Umweltinformatik in Straßburg und bei der Intergeo in Karlsruhe mit beachtlicher Resonanz bereits vor.

Wir möchten an dieser Stelle den am F&E-Projekt beteiligten Instituten für Ihr großes Engagement und die ausgezeichneten Leistungen herzlich danken und hoffen, daß die Ergebnisse von GLOBUS IV zu einer weiteren Effizienzsteigerung der Verwaltung und zu einer verbesserten Information der Öffentlichkeit beitragen.

Unser Dank gilt auch der Stabsstelle für Verwaltungsreform im Innenministerium für deren fachliche Beratung und finanzielle Unterstützung.

Besonders bedanken wir uns beim Institut für Angewandte Informatik des Forschungszentrums Karlsruhe, das zusätzlich das Projektmanagement und die Erstellung des Abschlußberichts übernommen hat.

*Inge Henning, Armin Koch, Roland Mayer-Föll
Ministerium für Umwelt und Verkehr Baden-Württemberg*

*Andree Keitel, Manfred Müller, Ernst Schmid, Horst Spandl
Landesanstalt für Umweltschutz Baden-Württemberg*

Inhalt

Vorwort

Inhaltsverzeichnis

Das Projekt GLOBUS 1

Die Ergebnisse des Projekts GLOBUS Phase IV als Beitrag für die Fortschreibung und Weiterentwicklung der Rahmenkonzeption des Umweltinformationssystems Baden-Württemberg 9

UVIS Umwelt- und Verkehrs-Informationsservice, Weiterentwicklung des Umwelt-Führungs-Informations-Systems..... 23

Die Anpassung des Personal Data Node an UIS-Nutzungen, Bericht des IKE über die Arbeiten im Teilprojekt PDN-UIS 57

WWW-UDK, Erweiterungen der Software zum komfortablen Recherchieren und Retrieval und Entwicklung von Komponenten zur Erfassung und Pflege von UDK-Daten via WWW 81

WWW-Szenario..... 95

Pflege der installierten Software aus vorhergehenden Projekten und Unterstützung der LfU beim Betrieb der WWW-Server (GIS-Dienste) 145

Vom Pilotprojekt zur Anwendung, Das Altlasten-Fachinformationssystem AlfaWeb 153

Ausblick auf die F+E-Arbeiten im Rahmen von GLOBUS Phase V (1998)..... 175

Das Projekt GLOBUS

Einführung

Aufgaben des Umweltinformationssystems des Landes Baden-Württemberg (UIS BW; UIS) sind insbesondere:

- die informationstechnische Unterstützung für die Fach- und Führungskräfte
 - im Ministerium für Umwelt und Verkehr (UVM)
 - in der Landesanstalt für Umweltschutz (LfU)
 - in weiteren Dienststellen des Landes und
 - in unter der Fachaufsicht des UVM stehenden Einrichtungen bei ihrer Aufgabenerledigung;
- die Bereitstellung von Umweltinformationen und Verkehrsinformationen;
- die Schaffung des freien Zugangs des Bürgers zu Informationen über die Umwelt.

Die Aufgabe des UIS in der Verwaltung umfaßt dabei die Bereiche:

- Einsatz der Informations- und Kommunikationstechnik (IuK) in Planung und Vollzug zur effizienten Erledigung der fachbezogenen und fachübergreifenden Verwaltungsaufgaben;
- Erhebung, Analyse und Prognose der punktuellen und landesweiten Situation in Umwelt und Verkehr;
- Informationsaufbereitung zur Entscheidungsunterstützung in der politischen und administrativen Führung in Landtag, Regierung und Verwaltung;
- Unterstützung der Bewältigung von Not-, Stör- und Vorsorgefällen, insbesondere durch schnellere und umfassendere Nachrichtenübermittlung und -verarbeitung.

Mit der Konzeption des Umweltinformationssystems Baden-Württemberg wurde bereits 1984 durch das damalige Ministerium für Ernährung, Landwirtschaft, Umwelt und Forsten begonnen. Die Konzeption wurde in den Jahren 1987 bis 1990 unter Führung des Umweltministeriums weiter ausgearbeitet und funktional erweitert. Parallel zu den Konzeptionsarbeiten begann in diesen Jahren die Realisierung von Basis- und Teilsystemen des UIS BW /1/, /2/, /3/. Die Verantwortung für das UIS hat seit 1996 das neu gebildete Ministerium für Umwelt und Verkehr.

Bedingt durch den technischen Fortschritt im IuK-Bereich wie etwa

- Dezentralisierung, Ausbau der Netzverbindungen,
- preisgünstige Hardware,
- standardisierte Betriebssysteme wie WINDOWS und UNIX,
- Standardisierung der Zugriffe auf relationale Datenbanken,

- einfach nutzbare Web-Technologie,
- mächtige Entwicklungswerkzeuge wie JAVA, C++ etc.
- verteilbare Softwarearchitekturen durch Middleware wie CORBA, DCOM
- Objektorientierung, Dienstkonzepte etc.

sowie durch ein verändertes Anwendungsprofil, z.B. verursacht durch

- Umorganisation in der Verwaltung,
- erweiterten Nutzerkreis (Kommunen, Öffentlichkeit etc.),
- neue gesetzliche Auflagen,

ist eine kontinuierliche Fortschreibung des systemtechnischen Rahmenkonzepts des Umweltinformationssystems erforderlich, an der sich alle Neuentwicklungen und die Weiterentwicklung existierender UIS-Komponenten orientieren müssen.

Mit dem Ziel einer kontinuierlichen funktionalen und systemtechnischen Fortentwicklung des UIS werden vom zuständigen Ministerium, begleitend zu Weiterentwicklung und Betrieb des Systems, Forschungs- und Entwicklungsprojekte /4/, /5/, /6/ zu unterschiedlichen Aspekten der Fortschreibung des UIS-Konzepts durchgeführt.

Ein zentrales Forschungs- und Entwicklungs(F+E)-Projekt in diesem Rahmen ist das Projekt **GLOBUS** /7/, /8/, /9/, das in Ein-Jahres-Teilprojekten seit 1994 durchgeführt wird.

An dem F+E-Entwicklungsprogramm dieses Projekts sind die folgenden wissenschaftlichen Institutionen beteiligt:

- das Forschungsinstitut für anwendungsorientierte Wissensverarbeitung an der Universität Ulm (FAW),
- das Forschungszentrum Informatik an der Universität Karlsruhe (FZI),
- das Institut für Angewandte Informatik des Forschungszentrums Karlsruhe (FZK/IAI),
- das Institut für Kernenergetik und Energiesysteme der Universität Stuttgart (IKE) und
- das Institut für Photogrammetrie und Fernerkundung der Universität Karlsruhe (IPF).

Das Ziel dieses Projekts ist es, für die UIS-Nutzer im UVM, in der LfU und gegebenenfalls weiteren Behörden eine aktive, benutzerfreundliche Auskunftskomponente im UIS BW zu entwickeln, die die unterschiedlichsten Informationsbestände auch über den engeren Rahmen des UIS hinaus erschließt und auf diese Weise die angestrebte Effizienzsteigerung in Sachbearbeitungs- und Entscheidungsvorgängen erreicht. Parallel hierzu sollen im Projekt GLOBUS Methoden, Konzepte und Werkzeuge zur Abgabe von Umweltinformationen an Ingenieurbüros und die Öffentlichkeit erarbeitet werden.

Die erste Phase des Projekts (GLOBUS I - 1994) hatte den Charakter einer Vorstudie /7/. Hierbei wurde untersucht, inwieweit und wie Internet-Techniken und -Werkzeuge zur Zeiteinsparung und Qualitätsverbesserung bei der Informationssuche im UIS nutzbar gemacht werden können. Es wurde das Konzept für eine Auskunftskomponente auf Basis des World- Wide Web (WWW) zur Erschließung der Umweltdatenberichte entworfen und in einigen seiner zentralen Funktionen prototypisch realisiert. Dabei wurde gezeigt, daß mit dem WWW und den zugehörigen, frei verfügbaren Werkzeugen für das UIS auf den gängigen Rechnerplattformen

(VMS, UNIX, MS-Windows u.a.) einfach zu bedienende, einheitliche Nutzerschnittstellen realisierbar sind und insbesondere auch der Zugang zu relationalen Datenbanken und GIS möglich ist.

Ziel der zweiten Phase des Projekts (GLOBUS II - 1995) war es, das Konzept einer übergreifenden Rahmenarchitektur für die Auskunftskomponente detaillierter auszuarbeiten und es an beispielhaften Anwendungen zum Einsatz zu bringen /8/. Die Arbeiten umfaßten die Konzeption einer diensteorientierten, auf WWW und CORBA basierten Architektur für das UIS. Dies wurde ergänzt durch eine Evaluierung von CORBA-Implementierungen im Hinblick auf den Einsatz im UIS BW und durch grundlegende Arbeiten zur Definition, Nutzung und Implementierung von Diensten im UIS. Als Element dieser neuen Architektur wurde für das Management verteilter Informationsquellen im UIS BW ein generelles Konzept entwickelt und in wesentlichen Teilen am Beispiel verschiedener UIS-Komponenten prototypisch realisiert.

Weiter in dieser GLOBUS-Phase enthalten waren Untersuchungen und Entwicklungen zum clientseitigen Einsatz von Werkzeugen zur Visualisierung von Umweltdaten. Hierzu wurden marktverfügbare Softwaretools bezüglich Funktionalität, Leistung, Integrierbarkeit und Kosten untersucht und in eine einfache Auskunftsumgebung integriert.

Als wichtige Anwendung im UIS wurde mit der Entwicklung des WWW-basierten Altlasten-Fachinformationssystems AlfaWeb begonnen. Dieses System soll dem Anwender alle relevanten Berichte und rechnergestützten Werkzeuge für die Altlasten-Bearbeitung über WWW zur Verfügung stellen und für die lokale Installation auch auf CD-ROM angeboten werden.

Im Rahmen der dritten Phase des GLOBUS-Projekts (GLOBUS III - 1996) wurden folgende F+E-Themen bearbeitet /9/:

- Weiterentwicklung des Umwelt-Führungs-Informationssystems UFIS unter Beachtung der neuen Systemarchitektur zu einem Umwelt- und Verkehrs-Informationsservice UVIS auf der Basis von Internet- und PC-Technologie. Dies umfaßte für 1996 die Erstellung einer UFIS-Version mit allen Funktionen des persönlichen Arbeitsplatzes unter Integration von ArcView, Word, Excel und einer komfortablen Bedienung auf der Basis von Metadaten.
- Erweiterung der UDK (Umwelt-Datenkatalog)-basierten Auskunftsdienste im UIS mit verbesserten Suchfunktionen, umfassendem Hilfesystem und GIS-Anbindung.
- Prototypische Realisierung von Diensten im UIS unter Einsatz von CORBA als Middleware (Beispiel: Simulationssystem CRAYSIM zur Berechnung der luftgetragenen Ausbreitung von Spurenstoffen).
- Inhaltlicher Ausbau des Altlasten-Fachinformationssystems AlfaWeb und Weiterentwicklung des Zugangssystems (u.a. Anbindung an den UDK) sowie Fortentwicklung und Integration der Werkzeuge für die Berichtsaufbereitung und -konvertierung.
- Konsolidierung des Dienstkonzepts für das UIS.
- Evaluierung von Java speziell im Hinblick auf die Integration von Applets in HTML.
- Umstellung der unter WWW im UIS realisierten Kartendienste von einem batch-orientierten serverseitigen Konzept auf eine client-orientierte Lösung zur Verbesserung der Interaktionsmöglichkeiten und Reduzierung der Netzbelastung.

Zeitlich und thematisch auf dem GLOBUS-Vorhaben aufbauend wurde 1996 das Projekt Hypermediatechnik für Umweltdaten **HUDA** /10/ begonnen.

Dieses Forschungs- und Entwicklungsvorhaben wurde durch das Umweltbundesamt und das Ministerium für Umwelt und Verkehr Baden-Württemberg auf den Weg gebracht. Es wird in Kooperation vom FAW Ulm (Leitung), dem Forschungszentrum Informatik in Karlsruhe, dem Forschungszentrum Karlsruhe und dem Lehrstuhl für Software-Technologie an der Universität Dortmund durchgeführt.

Ziel des Vorhabens ist die Unterstützung der Behörden der Umweltverwaltung bei der Erstellung und Verbreitung von Berichten und Informationen für Regierung, Verwaltung und Öffentlichkeit durch Einsatz moderner Informations- und Medientechniken. Dies betrifft in erster Linie die Präsentation von Umweltberichten, wie sie vom Umweltbundesamt in Form des Berichts 'Daten zur Umwelt' sowie von der Landesanstalt für Umweltschutz und dem Ministerium für Umwelt und Verkehr Baden-Württemberg in Form der 'Umweltdaten' herausgegeben werden.

Aufgabe des Vorhabens HUDA ist die Konzeption und Entwicklung eines Werkzeugkastens, mit dem die Erstellung von Umweltberichten auf der Grundlage von Basisinformationen aus den jeweiligen Fachgebieten mit möglichst wenig manuellem Aufwand ermöglicht wird. Die hierbei erarbeiteten Berichte sollen sowohl in gedruckter Form, wie auch als Hypermediadokumente im Internet / WWW und auf CD-ROM bereitgestellt werden.

Nach einer Machbarkeitsstudie (1996) wurden in einer zweiten Projektphase (1997) die Anforderungen und Arbeitsumgebung der Endanwender durch intensive Benutzerbefragungen eruiert und analysiert. Es wurde für das HUDA-System eine flexible und offene Systemarchitektur konzipiert, die es erlaubt, existierende Werkzeuge aus der verwendeten Bürokommunikationsumgebung auf einfache Weise einzubinden. Im Zentrum dieser Architektur steht ein Datenverwaltungssystem, in dem alle Berichtsteile abgelegt werden. Dieses Verwaltungssystem bietet Schnittstellen zu den anderen Systemkomponenten wie Datenaufbereitungssystem, Redaktionssystem, Konfigurations- und Layoutsystem. Im Rahmen dieser zweiten Phase wurde in 1997 bereits ein Prototyp des Systems erstellt.

Gliederung des GLOBUS IV - Berichts

Die vierte Phase des GLOBUS-Projekts baut voll inhaltlich auf F+E-Arbeiten und Ergebnisse der Vorphasen auf und stellt in den wesentlichen Punkten eine direkte Fortsetzung der dort begonnenen Arbeiten dar. Die Ergebnisse sind im vorliegenden Abschlußbericht GLOBUS - Phase IV in sieben in sich abgeschlossenen Kapiteln zusammengefaßt.

Das erste Kapitel '**Die Ergebnisse des Projekts GLOBUS Phase IV als Beitrag für die Fortschreibung und Weiterentwicklung der Rahmenkonzeption des Umweltinformationssystems Baden-Württemberg**' beschreibt die Konsequenzen und Perspektiven für die Weiterentwicklung des UIS BW, die sich aus den bis jetzt erarbeiteten F+E-Ergebnissen des GLOBUS-Projekts ergeben.

Im Kapitel 2 '**UVIS Umwelt- und Verkehrs-Informationsservice**' werden die vom FAW in 1997 durchgeführten Arbeiten zum Projekt Umwelt-Führungs-Informationssystem (UFIS II) beschrieben, das als Zwei-Jahres-Projekt in GLOBUS Phase III - 1996 und Phase IV - 1997 eingebettet war. Die Arbeiten umfassen das Redesign des ersten UFIS-Systems auf der Basis der neuen UIS-Systemarchitektur und seine funktionale Erweiterung zum Umwelt- und Verkehrs-Informationsservice UVIS.

Das Kapitel '**Die Anpassung des Personal Data Node an UIS-Nutzungen**' berichtet über die F+E-Arbeiten des IKE zur Weiterentwicklung des Personal Data Nodes (PDN) und seinen Einsatz als Metadaten-basierte Strategiekomponente zur Vermittlung von Diensten am UIS-Arbeitsplatz (UFIS II / UVIS).

Im Kapitel '**WWW UDK**' werden die Arbeiten und Resultate des FZI im Anwendungsumfeld des Umwelt-Datenkatalogs (UDK) dargestellt. Dies umfaßt Erweiterungen in den Recherche- und Retrieval-Funktionen sowie die Entwicklung von Komponenten zur Erfassung und Pflege von UDK-Daten über WWW.

Im Kapitel '**WWW-Szenario**' wird für die Integration von Metainformation, Sachdateninformation und Geoinformation das Konzept eines über WWW zugreifbaren Gesamtsystems im UIS entworfen und über die prototypische Realisierung auf der Basis von Java berichtet.

Das nächste Kapitel berichtet über die Arbeiten des IPF zur Unterstützung der LfU bei Pflege und Betrieb der installierten graphischen Dienste auf den WWW-Servern.

Im letzten Fachkapitel des GLOBUS-Berichts '**Das Altlasten-Fachinformationssystem AlfaWeb**' werden die Arbeiten von FZK/IAI zur Konsolidierung von AlfaWeb, zur Weiterentwicklung der Komponenten der Systemverwaltung und -pflege und speziell die Arbeiten zur Erstellung eines über CD-ROM vertreibbaren Systems beschrieben.

Einordnung von GLOBUS

GLOBUS ist ein Forschungs- und Entwicklungsvorhaben des Umweltinformationssystems im Rahmen des **Landessystemkonzepts Baden-Württemberg (LSK)**. Das LSK wurde 1997 durch ein neues IuK-Modell fortgeschrieben /11/ /12/, wozu GLOBUS wesentliche Beiträge leistete.

Literatur

- /1/ R. Mayer-Föll, P. Schilling, D. Weigert et al;
Konzeption für das Umweltinformationssystem Baden-Württemberg.
Ministerium für Ernährung, Landwirtschaft, Umwelt und Forsten Baden-Württemberg,
Mai 1986
- /2/ Umweltministerium Baden-Württemberg / McKinsey and Company, Inc.;
Konzeption des ressortübergreifenden Umweltinformationssystems
im Rahmen des Landessystemkonzepts Baden-Württemberg
(Phase I: Bestandsaufnahme und inhaltliche Konzeption;
Phasen II/III: Systemkonzeption und Umsetzungsplanung;
Phase IV: Weiterentwicklung der Rahmenkonzeption;
Phase V: Umsetzung der Rahmenkonzeption).
1987-1990
- /3/ Arbeitsgemeinschaft Diebold - Dornier - Ikoss;
Erstellung eines Landessystemkonzepts für einen rationellen und
wirtschaftlichen Einsatz der Informations- und Kommunikationstechniken
in der öffentlichen Verwaltung des Landes Baden-Württemberg,
Dezember 1984
- /4/ K. Blank, W.-F. Riekert, F. Schmidt, M. Tischendorf;
Projekt Integral: Integration heterogener Komponenten
des Umweltinformationssystems (UIS) Baden-Württemberg.
Abschlußbericht der Phase I.
FAW Ulm und IKE Universität Stuttgart, 1994
- K. Blank, W.-F. Riekert, Th. Krist, R. Münster, F. Schmidt, M. Tischendorf, K. Kübler;
Projekt Integral: Integration heterogener Komponenten
des Umweltinformationssystems (UIS) Baden-Württemberg.
Abschlußbericht der Phase II.
FAW Ulm und IKE Universität Stuttgart, 1994
- M. Gaul, L. Kadric, G. Klügl, Th. Krist, W.-F. Riekert, G. Wiest, I. Henning, F. Schmidt, A.
Schweizer, M. Tischendorf
Projekt Integral: Integration heterogener Komponenten
des Umweltinformationssystems (UIS) Baden-Württemberg.
Abschlußbericht der Phase III.
FAW Ulm, Umweltministerium Stuttgart und IKE Universität Stuttgart, 1995
- /5/ R. Kramer, R. Nikolai;
WWW-UIS Anwenderhandbuch.
Forschungszentrum Informatik (FZI), Karlsruhe, 1995
- R. Kramer, R. Nikolai;
WWW-UIS Administrationshandbuch.
Forschungszentrum Informatik (FZI), Karlsruhe, 1995

- R. Kramer, R. Nikolai;
 WWW-UIS: Auskunfts-, Administrations- und Verwaltungsdienste.
 Forschungszentrum Informatik (FZI), Karlsruhe, 1995
- /6/ R. Mayer-Föll, J. Pätzold et. al.
 Ministerium für Umwelt und Verkehr, Landesanstalt für Umweltschutz Baden-Württemberg,
 FAW Ulm;
 Umweltinformationssystem Baden-Württemberg -
 Fortschreibung und Weiterentwicklung der UIS Rahmenkonzeption
 als Teil des Landessystemkonzepts.
 Entwurf 1997.
- /7/ R. Mayer-Föll, A. Jaeschke;
 Projekt GLOBUS - Vorstudie zur Konzeption und prototypischen Realisierung
 einer aktiven Auskunfts-komponente für globale Umwelt-Sachdaten.
 Phase I 1994
 Wissenschaftliche Berichte FZKA 5863
- /8/ R. Mayer-Föll, A. Jaeschke;
 Projekt GLOBUS - Konzeption und prototypische Realisierung
 einer aktiven Auskunfts-komponente für globale Umwelt-Sachdaten
 im Umweltinformationssystem Baden-Württemberg
 Phase II 1995
 Wissenschaftliche Berichte FZKA 5700
- /9/ R. Mayer-Föll, A. Jaeschke;
 Projekt GLOBUS - Konsolidierung der neuen Systemarchitektur und Entwicklung erster Produk-
 tionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg
 Phase III 1996
 Wissenschaftliche Berichte FZKA 5900
- /10/ Wolf-Fritz Riekert, Andree Keitel, Thomas Schütz und Jürgen Strohm;
 Anforderungen und Lösungsansätze für ein hypermediales System zur
 Umweltberichterstattung
 Umweltinformatik '97 Tagungsband
 Metropolis-Verlag, Marburg, 1997
- /11/ Innenministerium Baden-Württemberg (1997): Landessystemkonzept Baden-Württemberg - Lei-
 stungsbilanz '96, Band 18 der Schriftenreihe Verwaltung 2000 der Stabsstelle für Verwaltungsre-
 form des Innenministeriums Baden-Württemberg, Stuttgart
- /12/ Schäfer, G. (1997): Neues IuK-Modell zur Fortschreibung des Landessystemkonzepts Baden-
 Württemberg, Stabsstelle für Verwaltungsreform des Innenministeriums Baden-Württemberg,
 Stuttgart

**Die Ergebnisse des Projekts
GLOBUS Phase IV
als Beitrag für die
Fortschreibung und Weiterentwicklung
der Rahmenkonzeption des
Umweltinformationssystems
Baden-Württemberg**

*R. Mayer-Föll,
Ministerium für Umwelt und Verkehr, Baden-Württemberg,
Kernerplatz 9,
70182 Stuttgart*

*J. Strohm, A. Schultze
Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW)
an der Universität Ulm,
Helmholtzstr. 16,
89081 Ulm/Donau*

1. EINFÜHRUNG.....	11
2. DIE GENERATIONEN DES UIS.....	11
3. DER STRUKTURELLE AUFBAU DES UIS.....	12
4. DIE VERÄNDERUNG DER RAHMENBEDINGUNGEN.....	13
5. LEITLINIEN FÜR DIE KONZEPTION.....	14
6. DIE DIENSTEARCHITEKTUR DES UIS.....	15
7. NEUE SYSTEMSICHT	16
8. UMSETZUNG DER RAHMENKONZEPTION.....	17
9. RESÜME UND AUSBLICK	18
10. LITERATUR:.....	20

1. Einführung

Zur Bewältigung der aus den Erfordernissen eines vorsorgenden und nachhaltigen Umweltschutzes resultierenden Aufgaben sind Politik und Verwaltung auf eine Vielzahl von möglichst umfassenden, problembezogenen und aktuellen Informationen angewiesen. Das UIS dient zum einen der effizienten Erledigung der Fachaufgaben durch die koordinierte Entwicklung und Bereitstellung leistungsfähiger Fachanwendungen, zum anderen werden vielfältige Informationen aus unterschiedlichen Quellen systematisch aufbereitet und an verschiedene Interessenten schnell und anwenderfreundlich weitergegeben. Die Entwicklung des UIS wurde im Jahr 1984 vom damaligen Ministerium für Ernährung, Landwirtschaft, Umwelt und Forsten (EM) begonnen. Im Jahr 1987 ging dann die Aufgabe der Weiterentwicklung des UIS an das damals neugebildete Umweltministerium (UM) über. Seit Juni 1996 ist das Ministerium für Umwelt und Verkehr (UVM) für das UIS verantwortlich. Von Anfang an wurde mit den berührten Ressorts und dem kommunalen Bereich innerhalb Baden-Württembergs sowie im Bund-/Länder-Arbeitskreis Umweltinformationssysteme (BLAK-UIS) bundesweit zusammengearbeitet.

Die Ziele des UIS waren frühzeitig umfassend formuliert: In der Verwaltung soll das UIS Planungs- und Vollzugsaufgaben im gesamten Umweltbereich effizient unterstützen. Das UIS soll Methoden zur Verfügung stellen, um Daten zur Umweltsituation zu erheben, zu analysieren und zukünftige Entwicklungen zu prognostizieren. Im Sinne eines Investitionsschutzes soll das UIS vorhandene IuK-Verfahren zusammenführen und in ein gemeinsames Konzept integrieren. Auch die Unterstützung bei der Bewältigung von Notfallsituationen sowie die rechtzeitige und umfassende Information von Politik, Verwaltung und Öffentlichkeit sind wichtige Ziele des UIS.

Das UIS ist ein Szenario des Landessystemkonzepts (LSK) Baden-Württemberg, ohne dessen Infrastruktur und Regelwerk es nicht wirtschaftlich aufgebaut und betrieben werden könnte.

2. Die Generationen des UIS

Die Entwicklung des UIS vollzog sich in mehreren Stufen (siehe Abbildung 1). Die erste UIS-Generation wurde als ressortinterne Konzeption des damaligen Ernährungsministeriums in den Jahren 1983-86 entwickelt. In dieser ersten Realisierungsstufe stand insbesondere der Ausbau von Fachsystemen im Vordergrund. In den Jahren von 1987-94 erfolgte ein ressortübergreifender Ausbau im UIS der 2. Generation. Neben der Fortführung der Fachsysteme wurde in dieser Phase mit der Entwicklung der übergreifenden Komponenten begonnen.

Seit 1995 wird die Weiterentwicklung zum dienstorientierten UIS der 3. Generation betrieben. Schwerpunkte sind hier die Vernetzung mit dem kommunalen Bereich sowie mit Behörden in anderen Bundesländern, Behörden des Bundes, anderer Nachbarstaaten und der EU. Wichtig ist auch die Schaffung eines freien Zugangs zu Umweltinformation für den Bürger. Auf der Grundlage der Arbeiten im Forschungs- und Entwicklungsvorhaben GLOBUS (Globale Umweltsachdaten) wurde mittlerweile die Rahmenkonzeption für dieses UIS der dritten Generation entworfen.

1. UIS-Generation	1983-86	<ul style="list-style-type: none"> - Ressortinterne Konzeption (UIS-EM) - Realisierung von Fachsystemen
2. UIS-Generation	1987-94	<ul style="list-style-type: none"> - Ressortübergreifende Konzeption (UIS-UM) - Ausbau der Fachsysteme - Konzeption und Ausbau übergreifen UIS-Komponenten - Landesverwaltungsnetz (SNA/DCA/DIA)
3. UIS-Generation	1995-99	<ul style="list-style-type: none"> - Dienstorientierte Konzeption (UIS-UVM) - Intranet/Internet/WWW - Landesverwaltungsnetz (multiprotokollfähig) - Verbindung zu anderen Ländern, Bund, EU - Direkte Zugriffsmöglichkeit des Bürgers

Abbildung 1: Konzeption, Aufbau, Betrieb und Ausbau des UIS in Generationen

3. Der strukturelle Aufbau des UIS

Das UIS wird konzeptionell von zwei Modellen geprägt, dem Ökologie-Modell und dem Modell einer allgemeinen Aufgabenstruktur, die die hohe informatorische Komplexität des Umweltbereiches sowie Folgen und Auswirkungen von Maßnahmen und Eingriffen strukturieren sollen.

Das „Ökologie-Modell“ trennt die Umwelt in Schutzgüter und Technosphäre. Während zu den Schutzgütern die klassischen Umweltmedien und -prozesse zählen, umfaßt die Technosphäre die Gesamtheit aller technischen Anlagen, Einrichtungen, Infrastrukturelemente, Abläufe und Prozesse einschließlich der Land- und Forstwirtschaft. Die Analyse der Beziehungen zwischen Umweltschutzgütern und Technosphäre und deren Beeinflussung wird als Steuerungsmöglichkeit für eine nachhaltige Verbesserung der Schutzgüter und damit der Umweltqualität betrachtet. Eine „allgemeine Aufgabenstruktur“ bestimmt Art und Umfang der Unterstützung durch das UIS. Hierdurch soll sichergestellt werden, daß die weitgehende Unabhängigkeit des UIS von Aufgabenverteilung und Organisation der Verwaltung gewährleistet ist. Gleichzeitig soll damit die Verknüpfung des UIS mit Informationssystemen der Kommunen, anderer Bundesländer, des Bundes, internationaler Institutionen und sonstiger Einrichtungen im Umweltbereich erleichtert werden.

Die zahlreichen inzwischen entwickelten und eingesetzten Systeme des Landes lassen sich in drei Kategorien einteilen: Basissysteme, Grundkomponenten und übergreifende Komponenten (siehe Abbildung 2). Ihr Zusammenspiel wird in der UIS-Rahmenkonzeption geregelt. Leitmotiv sind dabei die Durchgängigkeit der Daten durch die Systeme des UIS, die Verknüpfbarkeit dieser Daten und die berichts- und führungsorientierte Aufbereitung der Informationen. Die Mehrfachnutzung der Daten, Informationen und Systeme für unterschiedliche Anforderungen wird konsequent angestrebt.

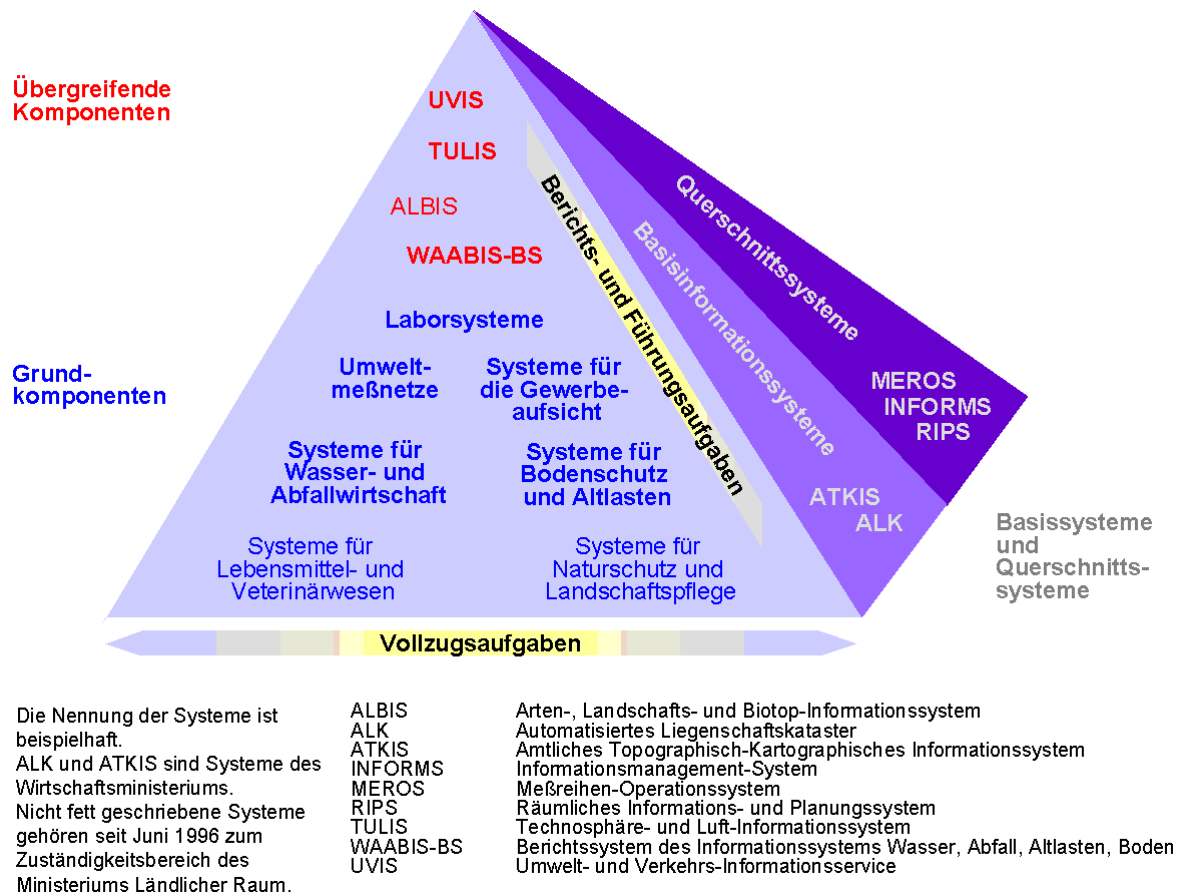


Abbildung 2: Systemkategorien des UIS mit Beispielen

4. Die Veränderung der Rahmenbedingungen

Die technische Konzeption des UIS erfolgt im Rahmen der Standards und Regeln sowie der Empfehlungen für offene Systemarchitekturen des Landessystemkonzepts Baden-Württemberg (LSK). Aufgrund der rasanten Entwicklung der Computertechnologie haben sich die Rahmenbedingungen seit der Konzeption des UIS der 2. Generation deutlich verschoben. Einschneidend war die Entwicklung von zentralen Mainframe-Architekturen hin zu verteilten PC-Netzen und Client-Server-Architekturen. Die zunehmende Verbreitung des Internet mit seinen Diensten wie dem WWW und das Entstehen neuer Standards im Middleware-Bereich wie der Common Object Request Broker Architecture (CORBA) ermöglichen heute im Rahmen einer Client-Server-Architektur die weltweite Nutzung von verteilten Ressourcen. Diese Ansätze im UIS nutzbar zu machen, erfordert zum Teil völlig neue Systemkonzeptionen. Dabei muß der Betrieb der bestehenden UIS-Komponenten bis zum Abschluß der Neuentwicklungen stets gewährleistet werden.

Zusätzlich zu diesen technischen Veränderungen ergaben sich auch Veränderungen in den Aufgabenstellungen, die sich beispielsweise aus dem gesetzlich festgeschriebenen Recht des Bürgers auf freien Zugang zu Informationen über die Umwelt ergeben, und strukturelle Veränderungen der Verwaltung, die von Umressortierungen, Verwaltungsreform, dem IuK-Verbund zwischen Land und sowie der Einführung von betriebswirtschaftlichen Strukturen geprägt sind.

5. Leitlinien für die Konzeption

Das Landessystemkonzept (LSK) wird derzeit durch ein Neues IuK-Modell fortgeschrieben. Das LSK ist nach wie vor zentrale Vorgabe, Rahmen und Leitlinie für das UIS.

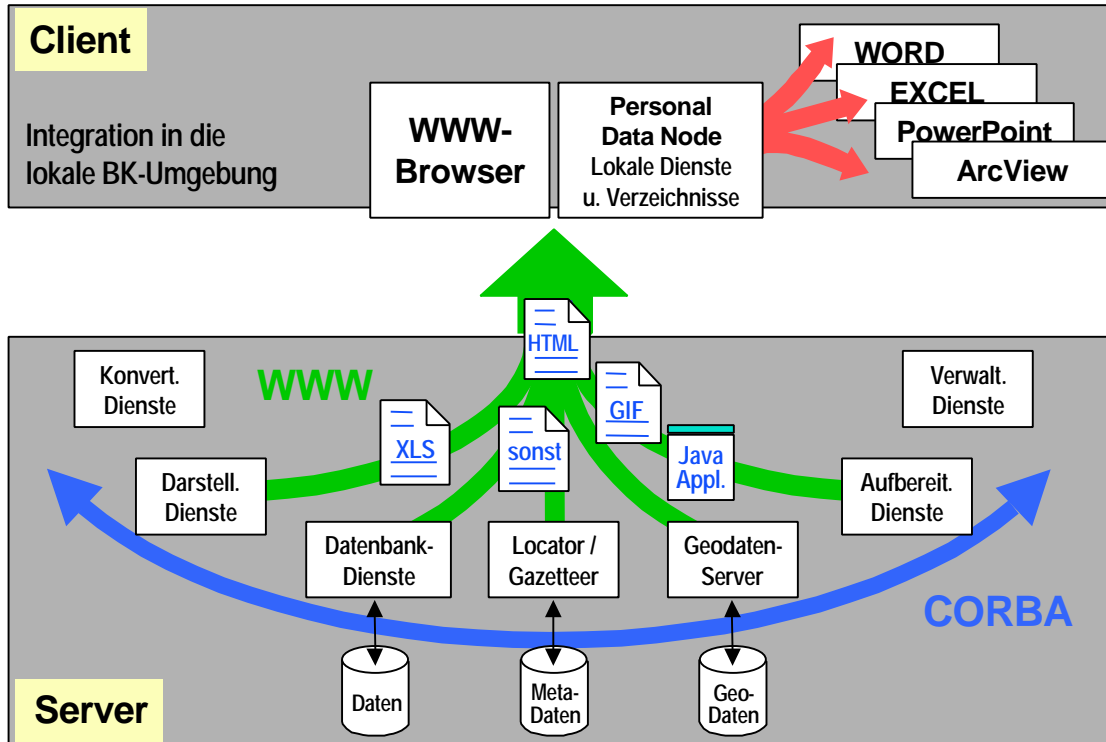


Abbildung 3: Technische Konzeption des UIS der 3. Generation

Eine neue, weitere Leitlinie für die Entwicklung des UIS stellen marktorientierte Mechanismen dar. Der grundsätzliche Gedanke hierbei ist es, Anbietern und Nutzern von Diensten oder Daten eine gemeinsame Plattform zur Verfügung zu stellen, über die eine Informationsweitergabe erfolgen kann. In einem solchen verteilten System stellt die zentrale UIS-Verwaltung Mechanismen zur Verfügung, mit deren Hilfe die Interoperabilität der Einzeldienste, die Konsistenz der Daten und die Aktualität der Metainformationen gewährleistet werden kann. Modulare Anwendungssysteme sind sehr flexibel, so daß Daten und Dienste verschiedener Anbieter kombiniert und zu komplexen Systemen zusammengefügt werden können. Voraussetzung hierfür ist die Schaffung eines effizienten Metainformationssystems. Als Kern dieses Systems wird im UIS der Umweltdatenkatalog (UDK) der Bund-/Länder-Kooperation UDK eingesetzt. Er stellt ein einheitliches Datenmodell zur Verfügung, das als Grundlage für die Bereitstellung von Filter- und Brokermechanismen sowie von Suchfunktionalitäten wie Repositories und Thesauri dient. Weitere Voraussetzungen sind wirkungsvolle Benutzer- und Sicherheitsmanagementsysteme, Verwaltungskomponenten und Verfahrensregeln.

6. Die Dienstarchitektur des UIS

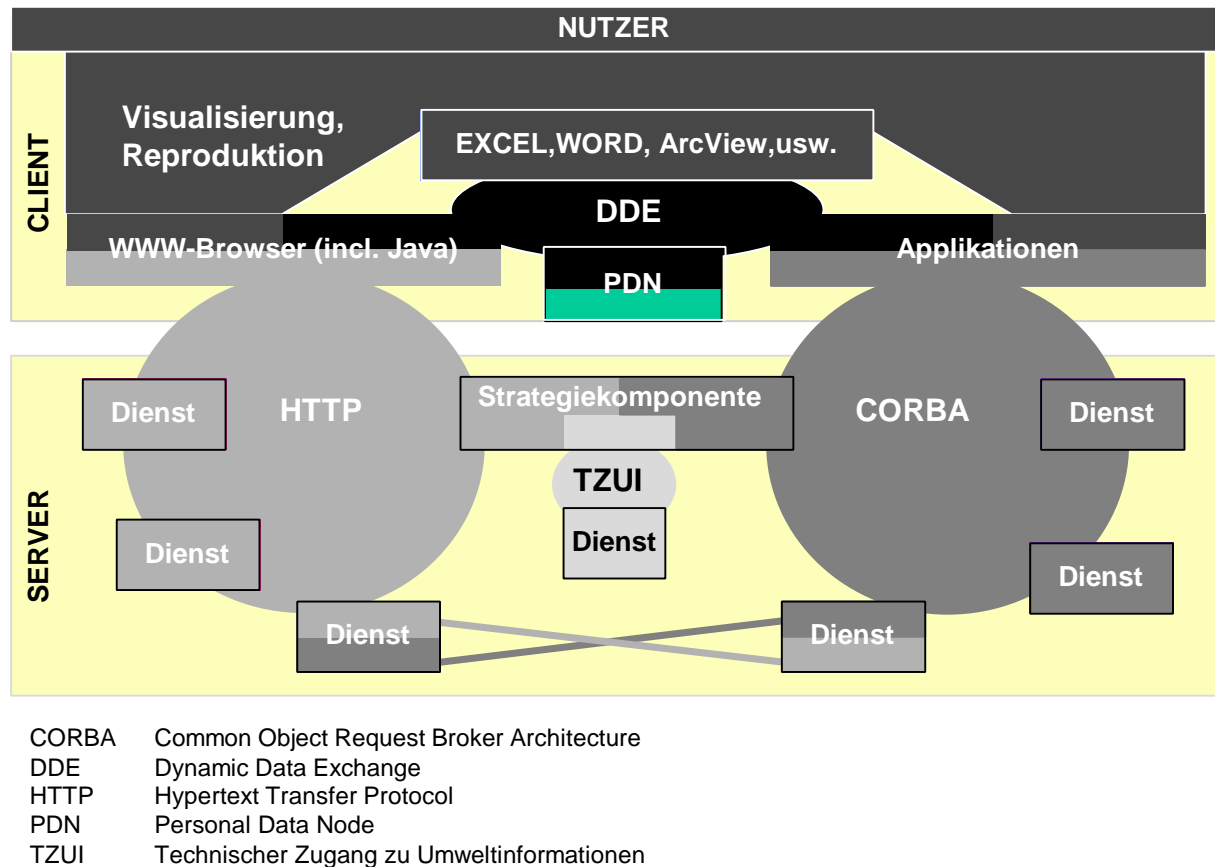


Abbildung 4: Dienstkonzeption des UIS (vereinfachte Darstellung)

Aus technischer Sicht bildet die in GLOBUS erarbeitete und in Abbildung 4 vereinfacht dargestellte Dienstkonzeption die Grundlage für die Entwicklung des UIS. Als Dienst im Sinne des UIS ist eine abgeschlossene Programmeinheit zu verstehen, die eine spezielle Aufgabe erfüllt, Methoden von Objekten aufrufen und an einer Rechnerschnittstelle anbieten kann, von einem Dienstanbieter angeboten und von einem Dienstanutzer verwendet werden kann. Die Kommunikation des Benutzers mit einzelnen Diensten erfolgt dabei im wesentlichen über die Mechanismen des WWW. Durch den WWW-Browser wird eine weitgehende Plattformunabhängigkeit erzielt und die Einbringung externer Dienstleistungen und Daten in die Bürokommunikationsumgebung optimal unterstützt. Erste Erfolge mit dem Einsatz von Java-Applets lassen weitere Verbesserungen der Integration von Serverdiensten in die lokale Arbeitsumgebung des UIS-Nutzers erwarten. Die Kommunikation zwischen serverseitigen Diensten sowie zwischen client-seitigen Java-Applets und serverseitigen Basisdiensten kann mit Hilfe von CORBA erfolgen. Im UIS wird gegenwärtig daran gearbeitet, ein Verzeichnis von Diensten aus den verschiedensten thematischen Bereichen in einer einheitlichen Beschreibungsform mittels CORBA-IDL (Interface Definition Language) bereitzustellen.

Das Nutzenpotential dieses Konzeptes ergibt sich aus der Möglichkeit, solche Dienste zu komplexeren Anwendungen zusammenzufassen und plattformübergreifend einzusetzen. Im Gegensatz zu bisherigen, eher monolithischen Anwendungsstrukturen lassen sich einzelne Dienste

dabei mehrfach verwenden und somit auf sehr effiziente Weise entwickeln und pflegen, was insbesondere unter Wirtschaftlichkeitsaspekten sinnvoll ist.

7. Neue Systemsicht

Das Dienstekonzept führt zu einer deutlichen Veränderung der Systemsicht im UIS. Wurden Systeme und Komponenten des UIS bislang als eher geschlossene Systeme verstanden, die durch die Paradigmen der Durchgängigkeit und Verknüfbarkeit der Daten verbunden waren, entsteht durch die Möglichkeit, Anwendungen technisch zu verknüpfen und auf gemeinsam genutzte, universelle Dienste aufzubauen, ein gänzlich anderes Bild.

Im Rahmen des Dienstekonzeptes ist eine Kategorisierung der Systeme entsprechend der vermuteten, dominanten Nutzergruppe nicht mehr notwendig und nicht mehr sinnvoll. Eine Anpassung an den speziellen Bedarf einzelner Nutzergruppen muß statt dessen durch die Bildung spezieller Sichten erfolgen, die durch die Kombination verschiedener Dienste zu größeren Anwendungsdiensten entstehen können. Die Zuordnung zwischen Nutzergruppen aus bestimmten Ebenen der Verwaltung und Systemkategorien in der bisherigen Rahmenkonzeption, die ihre Ausprägung in der UIS-Pyramide findet, ist daher heute teilweise überholt.

Abbildung 5 stellt die Sicht auf die UIS-Systeme entsprechend dem Dienstekonzept dar, in dessen Mittelpunkt der Nutzer steht. Die Basis des UIS bilden, wie auch bisher, die Daten. Sie werden durch Dienste erschlossen und verarbeitet. Dabei kann unterschieden werden zwischen systemnahen Diensten, die beispielsweise die Datenbestände verfügbar machen, und Anwendungsdiensten, die die Basisdienste zu komplexeren Systemen zusammenschließen.

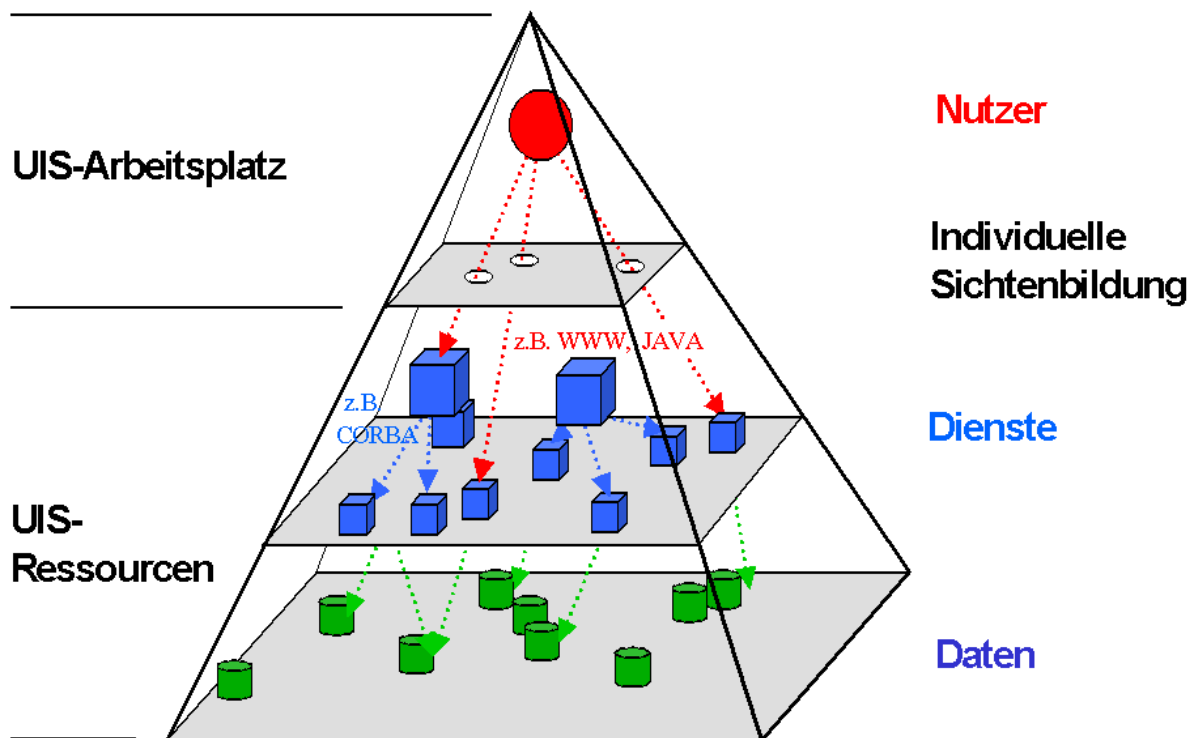


Abbildung 5: Neue Sicht auf die Systeme im UIS

Die Dienste stellen zusammen mit den Daten die Ressourcen des UIS dar. Auf die Dienste kann durch die Nutzer des UIS in der Verwaltung und der Öffentlichkeit zugegriffen werden. Die jeweilige Ausprägung, in der sich das UIS dem individuellen Nutzer gegenüber darstellt, ist durch die Bildung spezifischer Sichten bestimmt. Die Sichtenbildung kann dabei lokal auf dem Arbeitsplatz des UIS-Nutzers erfolgen oder nach erfolgter Authentifizierung des Nutzers durch die Dienste des UIS selbst vorgenommen werden.

8. Umsetzung der Rahmenkonzeption

Die Konsequenzen des neuen technischen UIS-Architekturmodells für die Rahmenkonzeption des UIS sind vielfältig. Eine herausragende Rolle kommt in Zukunft dem Informationsmanagement zu, das umfangreiche Metainformation über die vorhandenen Daten und Dienste im UIS bereithalten muß, um eine möglichst umfassende Unterstützung des Nutzers an seinem Arbeitsplatz zu ermöglichen.

Notwendig sind auch technische und organisatorische Konzepte, um die Anwendungsentwicklung umfassend zu unterstützen, die Verfügbarkeit von Daten und Diensten zu gewährleisten, sicherzustellen, daß Informationen über Gültigkeitsbereiche und Randbedingungen von Daten und Diensten beachtet werden, und zu garantieren, daß diejenigen Informationen des UIS, die nur einem begrenzten Personenkreis zugänglich sein sollen, auf sichere und zuverlässige Weise genutzt werden können. Letzteres erfordert mittelfristig die Weiterentwicklung bestehender Konzepte zu einem umfassenden, rollenbasierten Sicherheitskonzept, das allerdings nur in einem ressortübergreifend abgestimmten Verfahren angegangen werden kann.

Für die Umsetzung der UIS-Architekturvorschläge werden in der Rahmenkonzeption eine Reihe von Maßnahmen vorgeschlagen: Im Hard- und Softwarebereich muß eine weitere Migration in Richtung der aktuellen technologischen Erfordernisse erfolgen. Zur Umsetzung des Dienstekonzeptes werden Anleitungen und Leitlinien für die Systementwickler sowie Beschreibungen von Schnittstellen, Konzepten und vorhandenen Diensten benötigt.

Eine Reihe spezieller Entwicklungsaufgaben sollte vorrangig verfolgt werden. Dies betrifft den weiteren Ausbau der Werkzeuge, die den UIS-Arbeitsplatz bilden, die prototypische Integration von Sicherheitstechniken in das UIS sowie den weiteren Ausbau der WWW-Server des Ministeriums für Umwelt und Verkehr und der Landesanstalt für Umweltschutz.

Konzepte, die bereits angedacht und in Grundzügen erstellt worden sind, müssen weiter fortgeführt und in verständlicher Form den Entwicklern vorgelegt werden können. Dies betrifft insbesondere die Bereitstellung von ausreichender, beschreibender Information über Daten und Dienste sowie die ressortübergreifend abgestimmte Ausarbeitung eines umfassenden Sicherheitskonzeptes. Die Grundlage des Informationsmanagementsystems im UIS wird hierzu geeignet erweitert und fortentwickelt werden müssen.

In vielen Fällen ist eine kontinuierliche Beobachtung der informationstechnischen Entwicklung notwendig, so daß gegebenenfalls prototypische Umsetzungen neuer, vielversprechender Technologien erfolgen können. Solche prototypischen Umsetzungen sind notwendig, um konkrete Aussagen über die Verwendbarkeit neuer Technologien im UIS ableiten zu können. In

diesem Zusammenhang bietet sich an, die Arbeiten im Rahmen des Forschungs- und Entwicklungsvorhabens GLOBUS unter dieser Themenstellung fortzuführen.

Die Rahmenkonzeption für das UIS der 3. Generation soll die weitere Entwicklung und Pflege der UIS-Komponenten in den nächsten Jahren fördern und begleiten. Für die Konzeption gilt es dabei die Waage zu halten zwischen neuen, innovativen und vielversprechenden Technologien einerseits und andererseits den – sehr praktischen – Anforderungen, mit denen das UIS in der Unterstützung der täglichen Arbeit in der Umweltverwaltung konfrontiert wird. Als Orientierungshilfe über den Zeitraum von fünf Jahren hinaus, sind jedoch auch weiterführende Konzepte sinnvoll. Für das UIS bietet sich hierbei das Konzept eines Informationsmarktes an. Im Gegensatz zur aktuellen Situation, in der die Inhalte des UIS noch häufig von zentraler Stelle eingebracht und gepflegt werden, sieht ein Informationsmarkt die Betätigung von Anbietern und Nachfragern von Daten und Diensten vor. Die Aufgabe der zentralen UIS-Verwaltung bestünde dann einerseits darin, eine Plattform mit entsprechenden Basisdiensten bereitzustellen, auf der Anbieter und Nutzer von Daten und Diensten arbeiten könnten, und andererseits darin, das Regelwerk dafür zu definieren und die Qualitätssicherung zu betreiben.

9. Resüme und Ausblick

Die Bestrebungen, das UIS in Richtung eines verteilten Systems auf Basis des WWW weiterzuentwickeln, dienen auch der Kooperation mit anderen Ländern, Bund und Europäischer Union. Die Intensivierung der Zusammenarbeit mit dem kommunalen Bereich hat Priorität. Aufgrund der vielfältigen Kompetenz- und Zuständigkeitsverflechtungen im Umweltbereich zwischen dem Land und dem kommunalen Bereich ist dies eine wichtige und komplexe Aufgabe.

Ergebnisse aus der Fortschreibung und Weiterentwicklung der RK UIS wurden unter anderem bereits eingebracht in:

- den IuK-Verbund Land/Kommunen am Beispiel des Informationssystems Wasser, Abfall, Altlasten, Boden (WAABIS),
- die Spezifikation der Teilerneuerung der Kernreaktor-Fernüberwachung Baden-Württemberg
- die Erstellung eines Neuen IuK-Modells zur Fortschreibung des Landessystemkonzepts Baden-Württemberg,
- den Bund/Länder-Arbeitskreis Umweltinformationssysteme (BLAK UIS) der Umweltministerkonferenz.

Im UIS der 3. Generation kann dem Anspruch des Bürgers auf freien Zugang zu Informationen über die Umwelt besser als bisher entsprochen werden. Durch die Entwicklung verteilter UIS-Komponenten auf Basis des WWW und aufgrund der starken Verbreitung, die dieses Medium in den weltweiten Netzwerken findet, wird sich der Informationsbedarf der Öffentlichkeit in Zukunft effizienter decken lassen, als dies seither der Fall war.

Der wirtschaftliche Nutzen des UIS ergibt sich neben der wichtigen Unterstützung der Fachaufgaben durch übergreifende Ansätze. Mehrfachentwicklungen im Softwarebereich werden ebenso vermieden, wie redundante und inkonsistente Datenhaltung bzw. -fortführung.

Umständliche und langwierige Rechercheverfahren werden durch Informationstechnik schneller und qualitativ hochwertiger. Die kooperative Nutzung verschiedenster Informationsquellen im Umweltbereich rückt durch die Aktivitäten im UIS in greifbare Nähe. Zwar sind wichtige Probleme wie die Erarbeitung eines Dienstmanagements oder die Einrichtung leistungsfähiger Sicherheitsmechanismen erst in Ansätzen gelöst, entscheidende Schritte wurden jedoch mit der Entwicklung eines umfassenden Informationsmanagementsystems und der sukzessiven Anbindung verschiedener Fachdatenbanken bereits unternommen.

Als eines der wichtigsten Kriterien für den Nutzen der eingesetzten Systeme im UIS gilt weiterhin die Bereitstellung qualitativ hochwertiger Daten in erforderlichem Umfang zur Erledigung der anstehenden Dienstaufgaben.

Zusammenfassend läßt sich feststellen, daß sich der Aufbau des UIS bewährt hat. Die Rahmenkonzeption der früheren Generationen des UIS ist in vielen wichtigen Punkten auch heute noch aktuell und tragfähig. Die Unzulänglichkeiten, die sich zwischenzeitlich in einigen Bereichen ergeben haben, sollen durch die aktuell vorliegende Rahmenkonzeption für das UIS der 3. Generation nunmehr beseitigt werden.

Hervorzuheben ist der über die Jahre hinweg kontinuierliche Ausbau des Informationsmanagements. Nur den fortgesetzten Bestrebungen, im Rahmen der jeweils vorhandenen technischen Möglichkeiten sukzessive Komponenten des Informationsmanagements zu entwickeln und die dem UIS zugrunde liegende Datenbasis zu vereinheitlichen, ist es zu verdanken, daß für den heutigen Um- und Ausbau des Informationsmanagements bei ganz anderen technischen Möglichkeiten eine bemerkenswert gute Ausgangsbasis besteht. Letztlich ist dies auch eine Konsequenz der Erkenntnis, daß die Datenbasis den größten Wert des UIS darstellt. Als Eckwert für die Priorisierung des Aufwands gilt auch im UIS die bekannte Formel:

$$\text{Daten:Software:Hardware} = 100:10:1.$$

Bewährt hat sich auch die Organisationsstruktur des UIS. Die Bildung des Informationstechnischen Zentrums in der Landesanstalt für Umweltschutz war ein wichtiger Schritt, mit dem die informationstechnischen Ressourcen innerhalb des Geschäftsbereichs des Ministeriums und darüber hinaus gebündelt und im Sinne einer effizienten Unterstützung der Nutzer sowie der Systementwicklung zum Einsatz gebracht werden konnten.

Auch die Art und Weise, in der im UIS die Weiterentwicklung von Komponenten in Zusammenarbeit zwischen Verwaltung, Wirtschaft und Wissenschaft betrieben wird, hat sich als ausgesprochen befruchtend und erfolgreich erwiesen, auch und gerade für die Fortschreibung und Weiterentwicklung der Rahmenkonzeption des UIS.

Wichtig erscheint an dieser Stelle eine grundlegende Feststellung: Die Veränderung der Randbedingungen des UIS und die neuen technischen Möglichkeiten sind in erster Linie als Chance zu begreifen, aus der sich für das UIS neue Möglichkeiten einer Weiterentwicklung ergeben, die vor allem auch im Hinblick auf die Unterstützung der Aufgabenerledigung in der Verwaltung neue, wirtschaftliche Perspektiven bietet.

Mit den hier dargelegten Konzepten, Überlegungen und Maßnahmenvorschlägen wird das UIS modernisiert und ertüchtigt. Es trägt damit zu einer effizienteren Aufgabenerledigung in der Verwaltung sowie zu einer verbesserten Information der Öffentlichkeit wesentlich bei.

10. Literatur

- /1/ Birn, H.; Kaufhold, G.; Keitel, A.; Mayer-Föll, R. (1994): Umweltinformationssystem Baden-Württemberg - Die Konzeption, in: ONLINE Nr 10-1994, DATACOM-Verlag, Stuttgart.
- /2/ Birn, H.; Radermacher, F.J.; Schmidt, F.(1993): Das Umweltinformationssystem Baden-Württemberg (UIS) als kooperatives und integrierendes System: Stand und Ausblick, in: Jaeschke A. et al. (Hrsg.), Informatik für den Umweltschutz, Springer-Verlag.
- /3/ Bußmann, M.; Heißler, W.; Henning, I.; Müller, M. (1994): Umweltinformationssystem Baden-Württemberg - Die Konzeption, in: ONLINE Nr 11-1994, DATACOM-Verlag, Stuttgart.
- /4/ Ebbinghaus, J.; Grünreich, D. et al. (1996): Fachdatenintegration in ATKIS für das Umweltinformationssystem Baden-Württemberg - Abschlußbericht, Ministerium für Umwelt und Verkehr Baden-Württemberg.
- /5/ Geiger, W.; Jaeschke, A.; Rentz, O.; Simon, E.; Spengler, Th.; Zillox, L.; Zündel, T. (Hrsg.) 1997: Umweltinformatik '97, 11. Internationales Symposium der Gesellschaft für Informatik (GI), 2 Bände, Straßburg 1997.
- /6/ Günther, O. (1995): Gutachten zur Entwicklung des Umwelt-Datenkatalogs (UDK), Umweltministerium Niedersachsen, Hannover.
- /7/ Henning, I.; Wiest, G.; Schmidt, F.; (1997): UFIS II im Rahmen des UIS Baden-Württemberg, Informationsrecherche auf neuen Wegen, in: Geiger, W. et al (Hrsg.), Umweltinformatik '97, Straßburg, S. 356 ff.
- /8/ Hess, G.; Schultze A. (1995): Dokumentation des UIS-Workshops am 15.5.1995 im Umweltministerium Baden-Württemberg, Umweltministerium Baden-Württemberg, FAW Ulm.
- /9/ Innenministerium Baden-Württemberg (1997): Landessystemkonzept Baden-Württemberg - Leistungsbilanz '96, Band 18 der Schriftenreihe Verwaltung 2000 der Stabsstelle für Information und Kommunikation, Stuttgart.
- /10/ Innenministerium/Umweltministerium Baden-Württemberg (1995): Umweltinformationssystem Baden-Württemberg, Band 6 der Schriftenreihe Verwaltung 2000 der Stabsstelle für Information und Kommunikation, Stuttgart.
- /11/ Keitel, A.; Müller, M. (1995): Die Integration von Sachdaten, Geodaten und Metadaten im Umweltinformationssystem Baden-Württemberg, Landesanstalt für Umweltschutz, Karlsruhe.
- /12/ Mack, J.; Page, B. (1996): Zum Stand der Umweltinformationssystem-Entwicklung auf Landes- und Bundesebene, in: Lessing, H.; Lipeck U.W. (Hrsg.), Informatik für den Umweltschutz, 10. Symposium Hannover 1996, Metropolis Verlag, Marburg.
- /13/ Mayer-Föll, R.; Strohm, J.; Schultze, A. (1996): Das Umweltinformationssystem Baden-Württemberg - Überblick Rahmenkonzeption, in: Lessing, H.; Lipeck U.W. (Hrsg.), Informatik für den Umweltschutz, 10. Symposium Hannover 1996, Metropolis Verlag, Marburg.

- /14/ Mayer-Föll, R.; Jaeschke, A. (Hrsg.) (1994): Projekt GLOBUS: Vorstudie zur Konzeption und prototypischen Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten Phase I - 1994, Forschungszentrum Karlsruhe, Wissenschaftliche Berichte, FZKA 5863.
- /15/ Mayer-Föll, R.; Jaeschke, A. (Hrsg.) (1995): Projekt GLOBUS: Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten im Umweltinformationssystem Baden-Württemberg Phase II - 1995, Forschungszentrum Karlsruhe, Wissenschaftliche Berichte, FZKA 5700.
- /16/ Mayer-Föll, R.; Jaeschke, A. (Hrsg.) (1996): Projekt GLOBUS: Konsolidierung der neuen Systemarchitektur und Entwicklung erster Produktionssysteme für globale Umwelt-Sachdaten im Umweltinformationssystem Baden-Württemberg Phase III - 1996, Forschungszentrum Karlsruhe, Wissenschaftliche Berichte, FZKA 5900, 1996.
- /17/ Mayer-Föll, R.; Müller, M.; Schultze, A. (1997): Umweltinformationssystem Baden-Württemberg, in: DVW (Deutscher Verein für Vermessungswesen e.V.) Schriftenreihe 27/1997, 81. Deutscher Geodätentag 1997 in Karlsruhe - Geodäsie - Technik für Raum und Recht, Verlag Konrad Wittwer, Stuttgart.
- /18/ Mayer-Föll, R.; Schilling, P.; Weigert, D. et al. (1986): Konzeption für das Umweltinformationssystem des Landes Baden-Württemberg, Ministerium für Ernährung, Landwirtschaft, Umwelt und Forsten Baden-Württemberg.
- /19/ Ministerium für Umwelt und Verkehr Baden-Württemberg (1997): Rahmenkonzeption WAABIS (Informationssystem Wasser, Abfall, Altlasten, Boden) als Teil des Umweltinformationssystems Baden-Württemberg, Stuttgart.
- /20/ Ministerium für Umwelt und Verkehr Baden-Württemberg/debis Systemhaus (1997): Spezifikation der Teilerneuerung der Kernreaktor-Fernüberwachung Baden-Württemberg.
- /21/ Page, B.; Schickore, E.; Mack, J. (1996): Dokumentation der Umweltinformationssysteme des Bundes und der Länder, erstellt im Auftrag des Bund/Länder-Arbeitskreises Umweltinformationssysteme, Hamburg.
- /22/ Riekert, W.-F. (1995): Cooperative Management of Data and Services for Environmental Applications, GIS 95, Informatik aktuell, Springer.
- /23/ Riekert, W.-F.; Mayer-Föll, R.; Wiest, G. (1997): Management of Data and Services in the Environmental Information System (UIS) of Baden-Württemberg, in: SIGMOD RECORD Nr. 1, March 1997.
- /24/ Schäfer, G. (1997): Neues IuK-Modell zur Fortschreibung des Landessystemkonzepts Baden-Württemberg, Innenministerium Baden-Württemberg, Stuttgart.
- /25/ Seggelke, J.; Lessing, H. (1996): Globales Umweltinformationsnetz: Eckpunkte, Chancen und Gefahren, in: Lessing, H.; Lipeck U.W. (Hrsg.), Informatik für den Umweltschutz, 10. Symposium Hannover 1996, Metropolis Verlag, Marburg.
- /26/ Umwelt- und Verkehrsinformationen Baden-Württemberg, WWW-Server:
<http://www.uis-extern.um.bwl.de/>
- /27/ Umweltministerium Baden-Württemberg; McKinsey & Company (1988-1990): Konzeption des ressortübergreifenden Umweltinformationssystems (UIS) im Rahmen des Landessystemkonzeptes Baden-Württemberg, 12 Bände, Stuttgart.

UVIS Umwelt- und Verkehrs- Informationsservice

Weiterentwicklung des Umwelt-Führungs- Informations-Systems

*G. Wiest, M. Gaul, R. Meggle, B. Müntz, E. Rainbold, W.-F. Riekert,
Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW)
an der Universität Ulm,
Helmholtzstraße 16, 89081 Ulm,
Postfach 2060, 89010 Ulm*

*I. Henning,
Ministerium für Umwelt und Verkehr, Baden-Württemberg,
Kernerplatz 9,
70182 Stuttgart*

1 EINLEITUNG	25
2 DIE SYSTEMARCHITEKTUR	26
2.1 DER METADATENSERVEN (UVIS-RECHERCHE).....	27
2.2 DIE INFORMATIONSSERVEN (UVIS-SELEKTOR).....	27
2.3 DER PC-ARBEITSPLATZ (UVIS-CLIENT)	28
3 DIE UVIS-RECHERCHE	28
3.1 SCHLAGWORTE	28
3.2 RAUMBEZUG	29
3.3 ZEITBEZUG	30
3.4 DATENHALTENDE STELLE	30
3.5 INFORMATIONSTYP	31
4 DIE UVIS-SELEKTOREN	31
4.1 DIE DATENABFRAGE	31
4.2 METADATENGENERIERUNG	33
4.2.1 Grundsätzlicher Ablauf der Metadatengenerierung	33
4.2.2 Beispiel	34
5 DER UVIS-CLIENT	38
5.1 DER PERSONAL DATA NODE (PDN)	38
5.2 DER DECISION-MANAGER (DM)	39
5.2.1 Basisdaten für den Entscheidungsmechanismus	39
5.2.2 ADF (ApplicationDefinitionFile)	41
5.2.3 Parametertabelle	43
5.2.4 Kartentabelle	43
5.2.5 Das Regelwerk	44
6 DIE ARCVIEW-AUFBEREITUNG	47
6.1 DIE OBERFLÄCHE	47
6.2 DAS DARSTELLUNGSFENSTER	49
6.3 DIAGRAMMERSTELLUNG	49
6.3.1 Balkendiagramme.....	49
6.3.2 Flächendarstellung.....	52
6.4 BEENDEN DER ARCVIEW-AUFBEREITUNG	54
7 AUSBLICK.....	54
8 LITERATUR.....	56

1 Einleitung

Im Umweltinformationssystem (UIS) Baden-Württemberg wurden seit den Anfängen im Jahr 1984 inhaltlich und funktional weit fortgeschrittene Informationssysteme entwickelt, die inzwischen die Tauglichkeit für den praktischen Einsatz bewiesen haben. In den einzelnen Dienststellen wurde ein umfangreicher Bestand an Informationen und Wissen aufgebaut, der in Form von Daten, Methoden und multimedialen Dokumenten auf verschiedenen Computersystemen bereitgehalten wird.

Dieses wachsende Angebot von umweltrelevanten Informationen bringt mit sich, daß einzelne Informationsangebote oft nur sehr schwer auffindbar sind. Erforderlich ist deshalb ein Werkzeug, das es Mitarbeitern in der Umweltverwaltung ermöglicht, auf der Basis von Metadaten gesuchte Informationen zu finden, ohne durch ungewünschte Informationen gestört zu werden.

Zunehmende Bedeutung gewinnt neben der Recherche nach Umweltinformationen die Aufgabe, den Schatz an Informationen und Wissen an die (PC-)Arbeitsplätze der Mitarbeiter in der Umweltverwaltung zu bringen und dort mit bereits vorhandenen Applikationen aufzubereiten und zu nutzen. Besonders wichtig ist es in diesem Zusammenhang, einen sogenannten Umweltgeneralisten zu unterstützen, der sich einen fachübergreifenden Überblick über bestimmte Umweltfragen verschaffen möchte, um beispielsweise ganzheitliche Bewertungen der Umweltsituation vorzunehmen oder für Planungen Entscheidungsgrundlagen zusammenzustellen. Zu diesem Zweck entwickelt das FAW seit Anfang 1996 im Auftrag des Ministeriums für Umwelt und Verkehr Baden-Württemberg auf der Basis von World-Wide Web (WWW) und der neuen Systemarchitektur des UIS als Teil des Landessystemkonzeptes (LSK) eine grundlegend überarbeitete Betriebsversion des Umwelt-Führungs-Informationssystems. Als Weiterentwicklung des bisherigen Systems erhielt dieses Projekt zunächst den Namen UFIS II. Bereits in der Konzeptionsphase des Projekts UFIS II stellte sich heraus, daß die neue Betriebsversion nicht nur einer Weiterentwicklung sondern einer Neukonzeption des bisherigen Systems mit zusätzlichen Funktionalitäten entsprach. Es fanden nicht nur Veränderungen in der Systemkonzeption statt, die das bisher monolithische Gesamtsystem in einzelne verteilte Dienste (Services) aufbrachen (s. /1/, Informationsrecherche auf neuen Wegen). Auch der Nutzerkreis hat sich von der Führungsebene in die Ebene des Umweltgeneralisten verlagert. Daher wurde Ende 1997 das ehemalige UFIS, auf Grund des service- d.h. diensteorientierten Architektur, der geänderten Nutzergruppe und des, bedingt durch den Zusammenschluß des Umwelt- und Verkehrsministeriums, erweiterten Datenumfangs von reinen Umweltdaten auf Umwelt- und Verkehrsdaten, in UVIS (Umwelt- und Verkehrs-Informationsservice) umbenannt.

Das neue System basiert auf einer verteilten Client- / Serverarchitektur, die moderne Telekommunikationstechniken wie WWW und Internet bzw. Intranet nutzt. Eine mittels WWW konzipierte Systemarchitektur bietet den Vorteil, daß sie auf allen gängigen Rechnernetzen lauffähig ist und einen ortsunabhängigen Zugang zu Daten und Diensten, die mit einer entsprechenden Schnittstelle ausgestattet sind, ermöglicht. Server- und Clientsoftware für einen aktiven und passiven Zugang zum Internet stehen derzeit für fast alle Plattformen zur Verfügung und garantieren somit einen plattformunabhängigen Durchgriff auf alle im Netz zugänglichen Informationen.

Die Client- / Serverarchitektur des Systems stellt einen Verbund aus einem Server zur Verwaltung der Metadaten für die Recherche, verteilten Informationsservern zur Datenbereitstellung und den UVIS-Clients als Arbeitsplätzen dar. Sowohl die Verwaltung der Metadaten als auch die der Datenbestände erfolgt in einer relationalen Datenbank. Um einen Online-Zugang zu Informationen in solchen Datenbanken über WWW zu ermöglichen, wurde ein generisches Datenbankgateway entwickelt, das für den Zugriff auf die Metadaten und auf ausgewählte Informationsbestände eingesetzt wird.

Der Schwerpunkt der Entwicklung des Systems UVIS liegt auf dem UVIS-Client, auf den die Daten nach der Selektion transferiert werden und wo eine Integration und Aufbereitung abhängig von der Semantik der Daten erfolgt. Zur Aufbereitung werden bereits vorhandene und mittels Makros und Skripten erweiterte MS-Office-Applikationen und ArcView eingesetzt. Um eine Vorauswahl der möglichen Aufbereitungsarten abhängig von den Selektionsdaten und der Arbeitsplatzkonfiguration zu treffen, wird serverseitig eine Modellierung und Generierung von Beschreibungsinformationen zu den selektierten Daten vorgenommen, die von Entscheidungsmechanismen auf dem UVIS-Client ausgewertet werden.

Die verteilte Systemarchitektur von UVIS ermöglicht eine dezentrale Datenhaltung und Pflege der Informationsquellen und vermindert somit den administrativen Aufwand ebenso, wie eine redundante Datenhaltung verhindert wird. Das auf WWW basierende Vernetzungskonzept gestattet zum einen die wirtschaftliche Nutzung der bei den einzelnen Dienststellen bereits vorhandenen Hard- und Softwareressourcen und erlaubt zum anderen den Einsatz preisgünstiger und leicht wartbarer PC-Systeme als Rechnerarbeitsplätze.

Die Verlagerung der Aufbereitung von selektierten Umweltdaten auf den PC-Client unter Einsatz bereits existierender Applikationen minimiert Entwicklungskosten und ermöglicht es, Weiterentwicklungen von Standard-Applikationen ohne großen Entwicklungsaufwand in die Anwendung einfließen zu lassen. Für die Mitarbeiter der Umweltverwaltung ermöglicht dies einen Zugang zu ausgewählten Umweltinformationen und deren Aufbereitung ohne Einarbeitung in neue Applikationen und gewährleistet somit eine problemlose Integration in die gewohnte Arbeitsumgebung.

Die dienstorientierte und verteilte Systemarchitektur von UVIS ermöglicht so einen wirtschaftlichen und bequemen Zugang zu den angebotenen Daten, Funktionalitäten und Informationen und stellt einen weiteren Schritt auf dem Weg hin zu einer computerunterstützten kooperativen Arbeitsweise in der Umweltverwaltung dar.

2 Die Systemarchitektur

Die neue Systemarchitektur von UVIS basiert auf WWW und entspricht einer verteilten Client-Server-Architektur. Prinzipiell unterteilt sich UVIS physikalisch in drei Komponenten, die auch den drei logischen Schichten aus Benutzersicht entsprechen.

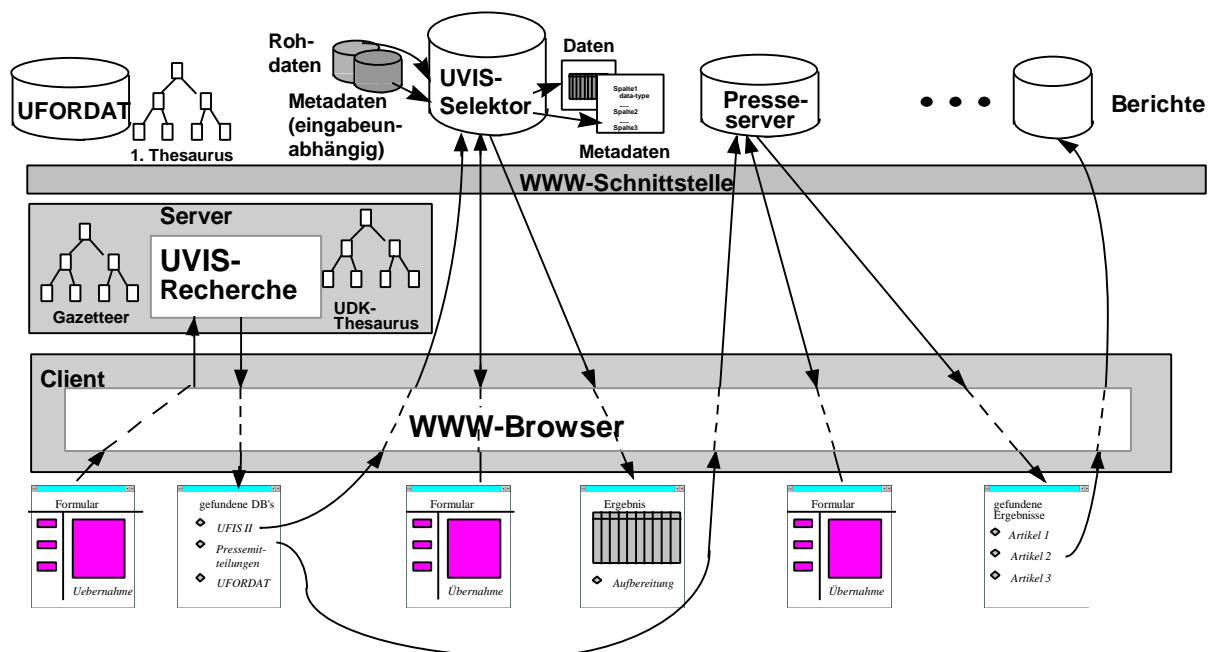


Abbildung 1: Die Systemarchitektur von UVIS

2.1 Der Metadatenserver (UVIS-Recherche)

Umwelt- und Verkehrsinformationen jeglicher Klassifizierung können durch einen gemeinsamen Satz an Attributen beschrieben werden. Die für UVIS relevanten Umwelt- und Verkehrsinformationen werden erfaßt, indem sie mittels dieser spezifischen Attribute (Metadaten) beschrieben werden. Der Benutzer kann dann mit Hilfe dieser Beschreibungsmerkmale spezifisch nach den eingetragenen bzw. erfaßten Informationen recherchieren, indem er eine Anfrage an den Metadatenserver stellt. Die Informationen selbst werden jedoch nicht an dieser zentralen Komponente redundant archiviert, sie verbleiben bei den verteilten datenhaltenden Stellen.

2.2 Die Informationsserver (UVIS-Selektor)

Die Informationen, deren Metadaten zentral archiviert werden, verbleiben bei den datenhaltenden Stellen. Somit existieren neben dem Metadatenserver verschiedene verteilte Server, die Informationen für den Benutzer anbieten. Hat der Benutzer beim Metainformationsserver eine Recherche gestartet und Ergebnisse erhalten, die ihn über Inhalt und Standort der Information informieren, so kann er auf die Informationen zugreifen. Dabei werden die Informationsserver vom Benutzer direkt kontaktiert, d.h. der Benutzerclient baut seine Verbindung direkt mit dem

entsprechenden Server auf und kann dort die gewünschten Daten abfragen. Hierbei wurden die UVIS-Daten für einen solchen Durchgriff für WWW erschlossen und selektierbar gemacht. Für UVIS spielt vor allem auch die Aufbereitung und Integration dieser Meßdaten eine große Rolle. Daher werden bei einer Abfrage von UVIS-Daten gleichzeitig beschreibende Informationen über Inhalt und Kennzeichen der selektierten Daten generiert. Diese können dann für eine Aufbereitung ausgewertet werden.

2.3 Der PC-Arbeitsplatz (UVIS-Client)

Der Arbeitsplatz von UVIS dient in erster Linie dazu, mittels eines WWW-Browsers Zugang zu den mit WWW-Technologien realisierten Servern zu erhalten. Dies ist ausreichend für die Informationsrecherche und den Zugang zu den entsprechenden Daten. Speziell für den Durchgriff auf UVIS-Daten-Selektoren sind jedoch erweiterte Funktionalitäten für den Client vorgesehen. UVIS-Daten sind Meßdaten, die mittels graphischer Aufbereitung auf dem Client weiterverarbeitet werden sollen. Hierzu werden von Seiten des Informationsservers bei der Selektion Metadaten (Informationen, die die selektierten Ergebnisdaten beschreiben) generiert, die Auskunft darüber geben, wie diese Selektionsergebnisse aufbereitet werden können. Der UVIS-Client besitzt daher zusätzlich zu den MS-Office-Applikationen und ArcView einen Decision-Manager, der diese Auswertung vornimmt und dem Benutzer gemäß der Konfiguration seines PCs mitteilen kann, welche Aufbereitungsarten für diese Daten möglich und sinnvoll sind.

3 Die UVIS-Recherche

Die Metadatenkomponente von UVIS wurde bereits in diversen Berichten ausführlich dargestellt. Hier wird insbesondere auf die detaillierte Beschreibung im Bericht zu GLOBUS III (siehe /2/) verwiesen. An dieser Stelle soll nur ein Überblick gegeben werden, welche Attribute für die Recherche zur Verfügung stehen und wie die wichtigsten dieser Attribute mit den zu recherchierenden Informationen verknüpft sind.

Für die Beschreibung von und Recherche nach Umwelt- und Verkehrsinformationen wurden, angelehnt an die bestehende Struktur des UDK, die unten beschriebenen Attribute herauskristallisiert. Anhand dieser Beschreibungselemente lassen sich in sehr allgemeiner Form Informationen sämtlicher Klassifizierung beschreiben und recherchieren.

3.1 Schlagworte

Jede Information wird mittels Schlagworten beschrieben, die ihren thematischen Inhalt beschreiben. Diese Eingabe der Schlagworte kann bei der Erfassung nicht der freien Eingabe des Benutzers überlassen werden. Ebenso kann die Spezifikation der Schlagworte bei der Recherche nicht frei eingegeben werden. Bei der Recherche wären sonst die anfragespezifischen Schlagworte und die bei den Metadaten erfaßten, auf Grund der Problematik der Synonyme und verwandten Begriffe und der unterschiedlichen Schreibweisen, wenn überhaupt nur mit aufwendigen Filtern vergleichbar. Die Schlagworte werden daher aus dem polyhierarchischen UDK-Thesaurus gewählt und bei der Erfassung die Informationen (Ressourcen) mit dem je-

weiligen Thesaurusbegriff (Deskriptor) verknüpft. Durch Spezifikation eines Schlagwortes aus dem Thesaurus für die Recherche werden dann die dem Begriff zugeordneten Ressourcen selektiert. Darüber hinaus werden sämtliche Unterbegriffe berücksichtigt, und die entsprechend zugeordneten Ressourcen werden ebenfalls in die Ergebnismenge aufgenommen.

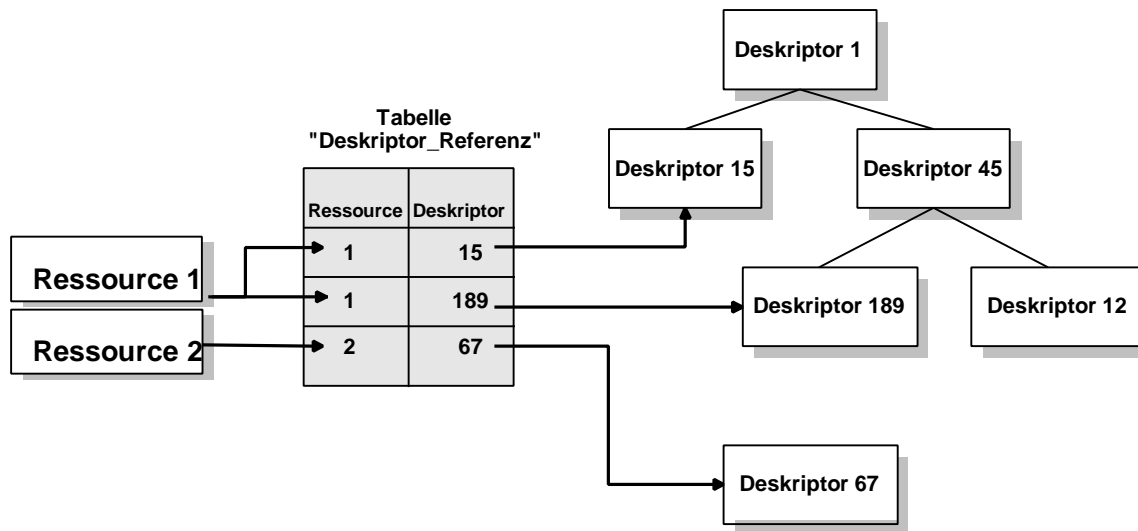


Abbildung 2: Zuordnung Schlagworte zu Informationen

3.2 Raumbezug

Ähnlich wie bei den Schlagworten werden Georeferenzen den Informationen (Ressourcen) zugeordnet. Georeferenzen werden durch Gebietseinheiten repräsentiert, die einen Namen tragen und in (teilweise hierarchischer) Beziehung zueinander stehen. Jede Gebietseinheit wiederum ist, zur Festlegung ihrer geometrischen Struktur, einer Gruppe Rastern zugeordnet, die sich innerhalb dieser Gebietseinheit befinden. Die Ressource wird wie bei der Verschlagwortung Gebietseinheiten zugeordnet. Für die Recherche kann der Benutzer zusätzlich zur Festlegung der Gebietseinheit über Auswahl des Gebietsnamens Rechteckbereiche auf den gerasterten Bereichen festlegen. Mittels der Zuordnung sämtlicher selektierter Raster zu den Gebietseinheiten kann dann die Ressource ermittelt werden, die den Gebieten und somit der Rasterfläche zugeordnet ist.

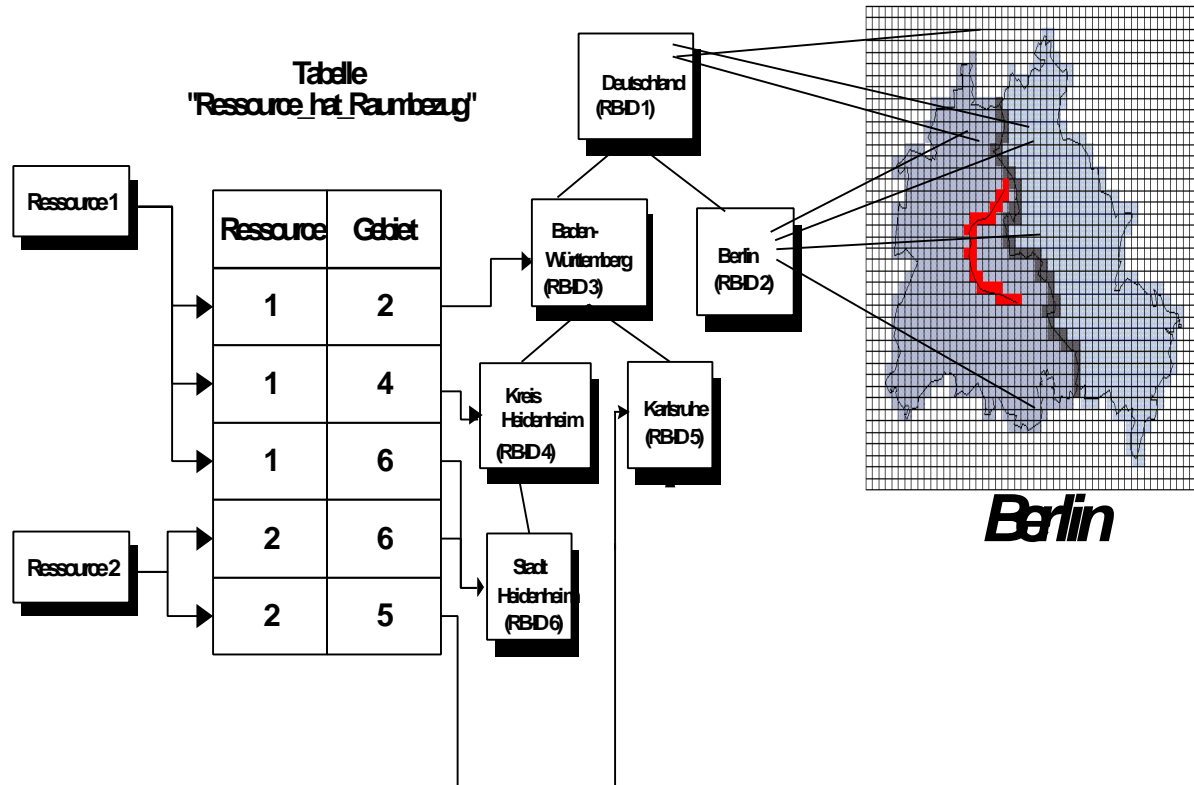


Abbildung 3: Zuordnung Rasterdaten, Gebietseinheiten zu Informationen

3.3 Zeitbezug

Für jede Information wird festgelegt, zu welchen Zeitpunkten sie zu den festgelegten Themen Inhalte trägt. Hierbei werden Zeitintervalle oder Zeitpunkte für die Informationen spezifiziert. Der Anfragende kann somit in seiner Suchanfrage einen Zeitraum/-punkt festlegen, für den er Informationen benötigt. Für die Recherche werden dann die vom Benutzer spezifizierten Zeitintervalle in einzelne Zeitpunkte aufgelöst, und die in den Metadaten diesen Zeitpunkten zugeordneten Ressourcen werden selektiert.

3.4 Datenhaltende Stelle

Die verteilte Serverarchitektur sieht vor, die Informationen bei ihren datenhaltenden Stellen zu belassen. Somit werden bei den Metadaten die datenhaltenden Stellen mit erfasst. Zum einen, um den Verantwortlichen für die Inhalte der recherchierbaren Informationen für den Benutzer mit auszugeben. Zum anderen kann der Benutzer gezielt seine Recherche auf bestimmte Datenherren beschränken. Darüber hinaus setzt sich die Adresse für einen anschließenden Durchgriff (URL) aus der Serveradresse der entsprechenden datenhaltenden Stelle zusammen. Daher muß dann nicht für jede Information diese Serveradresse archiviert werden, sondern nur für die datenhaltende Stelle.

3.5 Informationstyp

Informationen lassen sich klassifizieren. Hier wird bei der Erfassung und Recherche für alle Klassen ein gemeinsamer Satz an Attributen spezifiziert. Es wird jedoch festgehalten, welcher Klasse die Informationen angehören, und der Benutzer kann sich bei der Recherche auf eine Klasse spezialisieren.

Diese fünf oben beschriebenen Attribute dienen zur Beschreibung der Informationen und stehen auch für die Recherche in einem Suchformular als Kann-Felder zur Verfügung. Somit kann durch Spezifikation dieser Attribute eine Anfrage an das zentrale MIS gestellt werden und sämtliche, diesen spezifizierten Attributen zugeordneten Ressourcen, werden als Suchergebnis mit ihrer URL für den Durchgriff aufgelistet. Die UVIS-Selektoren, die die Datenabfrage der UVIS-Meßdaten ermöglichen, werden somit mit zu den Kenngrößen adäquaten Begriffen aus dem UDK-Thesaurus verschlagwortet; gemäß den Gebieten, zu denen Meßdaten vorhanden sind, werden die Raumbezugszuordnungen von den einzutragenden Selektoren zu den entsprechenden Gazetteer-Gebietseinheiten vorgenommen, und es werden die Zeitpunkte zugewiesen, zu denen Meßdaten existieren.

4 Die UVIS-Selektoren

Da die Selektoren sowohl aus Benutzer- als auch aus programmtechnischer Sicht bereits in /2/ beschrieben sind, soll hier nur ein kurzer Überblick über die Datenabfragemöglichkeit mittels Selektoren gegeben werden, bevor ausführlicher auf die Metadatengenerierung eingegangen wird.

4.1 Die Datenabfrage

Jede Datenabfrage über einen UVIS-Selektor basiert letztendlich auf einem parametrisierten Select-Statement. Die Parameter für eine konkrete Abfrage können vom Benutzer über eine framebasierte Oberfläche festgelegt werden (s. Abb. 4 als Beispiel). Sind alle gewünschten Parameter eingetragen, kann im Funktionsframe „Selektor starten“ aufgerufen und so die UVIS-Abfrage mit den im Formularframe angezeigten Parametereinstellungen durchgeführt werden. Mit der Datenabfrage wird gleichzeitig eine Datei erstellt, die beschreibende Daten zu den Ergebnisdaten enthält.

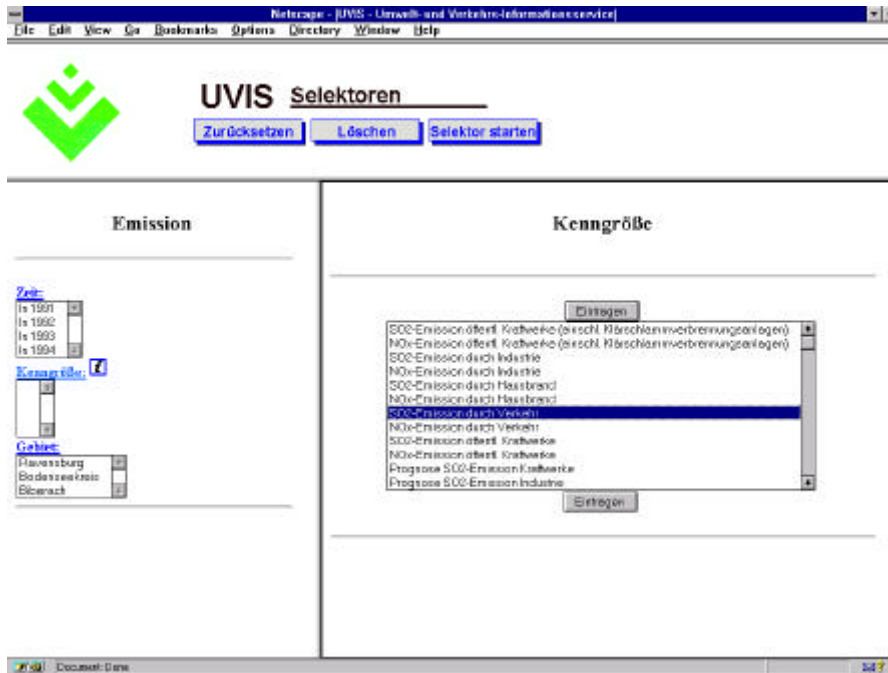


Abbildung 4: UVIS-Selektor „Emission“

Für die Darstellung des Ergebnisses existieren verschiedene Möglichkeiten (s. Abb. 5):

- Für einen ersten Überblick können die Ergebnisdaten im Browserfenster als HTML-Tabelle angezeigt werden.
- Die Ergebnisdaten können in einen Tabellenbrowser geladen werden. Dieser zeigt die Daten in einem separaten (Java-)Fenster an mit der Möglichkeit, Spalten ein- und auszublenzen oder auch die Ergebnissätze nach beliebigen Spalten zu sortieren. D.h. die Ergebnisdaten können (in eingeschränktem Umfang) weiterverarbeitet werden, auch wenn clientseitig keine sonstigen Tools zur Weiterverarbeitung vorliegen.
- Die Ergebnisdaten können über den PDN auf den Client heruntergeladen und dort weiter aufbereitet werden.

The screenshot shows a Netscape browser window with the title "Netscape - [UVIS - Umwelt- und Verkehrs-Informationsservice]". The main content area displays the "UVIS Selektoren" interface. On the left, there are filters for "Zeit" (years 1991-1994), "Kenngröße" (checkbox), and "Gebiet" (dropdown menu with options: Ravensburg, Bodenseekreis, Biberach). The main table has columns: MER_RO_X, MER_RO_Y, WERT, KURZNAME, ZEIT. The table contains 10 rows of data. To the right of the table is a "Tabelle einblenden" button. Below the table is a "Weiter zur Aufbereitung (PDF)" button and another "Tabelle einblenden" button. The bottom of the browser window shows a status bar with "SO2-Emission durch Verkehr" and "Unsigned Java Applet Window".

MER_RO_X	MER_RO_Y	WERT	KURZNAME	ZEIT
3588674	5353107	189	+	1991
3588402	5321069	236	+	1991
3588455	5306596	128	+	1991
3588674	5353107	194	+	1992
3588455	5306596	132	+	1992
3588402	5321069	249	+	1992
3588674	5353107	197	+	1993
3588455	5306596	134	+	1993
3588402	5321069	265	+	1993
3588674	5353107	202	+	1994

Abbildung 5: Darstellung des Abfrageergebnisses im Browser- und Java-Fenster

4.2 Metadatengenerierung

4.2.1 Grundsätzlicher Ablauf der Metadatengenerierung

Bei jeder Datenabfrage über einen UVIS-Selektor wird neben der eigentlichen Ergebnistabelle eine Datei erstellt, die die Ergebnisdaten in MDDL¹ beschreibt. Hierbei existiert für jeden Selektor ein Template mit spezifischen statischen Beschreibungselementen, das bei der Datenselektion um die für die konkrete Abfrage spezifischen Elemente ergänzt wird. Die daraus resultierende MDDL-Datei bildet die Eingabe für den PDN, der auf dieser Grundlage die Entscheidung über mögliche Aufbereitungen trifft.

Im einzelnen sieht der Programmablauf bei der Metadatengenerierung so aus, daß zunächst das Template geparkt wird, um festzuhalten, an welchen Stellen noch welche Informationen ergänzt werden müssen. Anschließend wird die Ergebnisdatei eingelesen, wobei alle notwendigen Berechnungen (z.B. Minimalwerte, Anzahl unterschiedlicher Werte pro Spalte, Gesamtzeilenanzahl, ...) durchgeführt werden. Anschließend wird noch einmal das Template eingelesen und mit den Ergänzungen in die MDDL-Datei geschrieben.

¹ MDDL: Meta Data Description Language (Teilmenge von SGML)

4.2.2 Beispiel

Im folgenden soll die Metadatengenerierung am Beispiel des UVIS-Selektors „Emission“ beschrieben werden; abgefragt werden verschiedene Kenngrößen nach 1990 für die Gebiete Bodenseekreis, Ravensburg und Biberach.

Select-Statement

Das Starten des UVIS-Selektors in diesem Beispiel führt zu folgendem Select-Statement:

```

NAME emission;
SELECT lfu_ve_kenngroessen.ve_kenngroessen_nr,
       lfu_ve_kenngroessen.ve_kenngroessen_name,
       lfu_ve_kenngroessen.dimensions_nr, lfu_ve_kenngroessen.zeittyp,
       lfu_ve_kenngroessen.messobjekt, lfu_ve_kenngroessen.messdichte,
       lfu_ve_kenngroessen.ansprechpartner,
       lfu_ve_kenngroessen_werte.ve_kennz,
       lfu_ve_kenngroessen_werte.rips_oac,
       lfu_rips_oac.name as gebietsebene, lfu_ve_verzeichnis.name,
       lfu_ve_verzeichnis.rechtswert, lfu_ve_verzeichnis.hochwert,
       lfu_ve_verzeichnis.mer_lu_x, lfu_ve_verzeichnis.mer_lu_y,
       lfu_ve_verzeichnis.mer_ro_x, lfu_ve_verzeichnis.mer_ro_y,
       lfu_ve_kenngroessen_werte.wert, lfu_dimension.kurzname,
       lfu_ve_kenngroessen_werte.zeit
FROM   lfu_ve_kenngroessen, lfu_ve_kenngroessen_werte, lfu_ve_verzeichnis,
       lfu_dimension, lfu_ve_kenn_begr_zuord, lfu_rips_oac
WHERE  ((ZEIT > '1990')) AND
       ((lfu_ve_kenngroessen.ve_kenngroessen_nr = 1 ) or
        (lfu_ve_kenngroessen.ve_kenngroessen_nr = 3 ) or
        (lfu_ve_kenngroessen.ve_kenngroessen_nr = 5 ) or
        (lfu_ve_kenngroessen.ve_kenngroessen_nr = 7 )) AND
       ((lfu_ve_kenngroessen.ve_kenngroessen_nr =
         lfu_ve_kenn_begr_zuord.ve_kenngroessen_nr) AND
        (begriffs_nr = 27)) AND
       ((lfu_ve_kenngroessen_werte.ve_kennz = 8426000 AND
         lfu_ve_kenngroessen_werte.rips_oac = 107010400) or
        (lfu_ve_kenngroessen_werte.ve_kennz = 8435000 AND
         lfu_ve_kenngroessen_werte.rips_oac = 107010400) or
        (lfu_ve_kenngroessen_werte.ve_kennz = 8436000 AND
         lfu_ve_kenngroessen_werte.rips_oac = 107010400)) AND
       (lfu_ve_kenngroessen.ve_kenngroessen_nr =
        lfu_ve_kenngroessen_werte.ve_kenngroessen_nr) AND
       (lfu_ve_kenngroessen_werte.rips_oac =
        lfu_ve_verzeichnis.rips_oac) AND
       (lfu_ve_kenngroessen_werte.ve_kennz =
        lfu_ve_verzeichnis.ve_kennz) and
       (lfu_ve_kenngroessen_werte.rips_oac = lfu_rips_oac.rips_oac) AND
       (lfu_ve_kenngroessen.dimensions_nr = lfu_dimension.dimensions_nr)
ORDER BY ZEIT;
```

Ergebnistabelle des Selektors in Ausschnitten

Zur Vereinfachung soll ein Ausschnitt aus der Ergebnistabelle des Selektors herausgegriffen werden:

<i>Ve_kenngroessen_name</i>	<i>Gebietsebene</i>	<i>Gebiet</i>	<i>Wert</i>	<i>Zeit</i>
SO2-Emission durch Verkehr	Kreis	Biberach	189	1991
SO2-Emission durch Verkehr	Kreis	Bodenseekreis	128	1991
SO2-Emission durch Verkehr	Kreis	Ravensburg	236	1991
...	...			

Tabelle 1: Ausschnitt einer Ergebnistabelle (UVIS-Selektor-Emission)

Beschreibung der Ergebnisdaten (natürlichsprachlich und in MDDL)

Im folgenden werden die Daten zunächst natürlichsprachlich beschrieben, um die sich anschließende MDDL-Beschreibung verständlich zu machen:

Prinzipielle, die Ergebnisdatei betreffende Informationen

- Die URL der Ergebnisdatei ist
http://www.faw.uni-ulm.de:9876/integral_bin/SRV_DATA/TMP/878197800-13348
- Die Dateigröße der Archiv-Datei beträgt 651 Byte
- Erstellungsdatum des Archivs ist der 30. Oktober 1997, 10:32
- Zum Entpacken der Archiv-Datei wird „unzip“ verwendet

Informationen zum Kartenausschnitt

- Der Kartenausschnitt der Ergebnisdaten hat die Koordinaten xmin=..., ymin=..., xmax=..., ymax=...
- Die Daten liegen auf Kreisebene vor

Prinzipielle, die Tabelle betreffende Informationen

- Die Tabelle hat als Spaltentrenner TAB
- Zeilentrenner ist CR
- Dateiendekennung ist EOF
- Die Ergebnistabelle hat 12 Datensätze und 20 Spalten

Beschreibung der Tabellenspalten

- Spalte 2, „VE_KENNGROESSEN_NAME“
Spaltentyp: String
Anzahl unterschiedlicher Werte: 1
Die Spalte stellt für alle Ebenen einen Bezugswert (x-Achsen-Wert) dar.

- Spalte 10, „GEBIETSEBENE“
Spaltentyp: String
Anzahl unterschiedlicher Werte: 1
Die Spalte beinhaltet die Ebene und stellt weder Bezugs- noch Ergebniswert dar.
- Spalte 11, „NAME“
Spaltentyp: String
Anzahl unterschiedlicher Werte: 3
Die Spalte stellt für alle Ebenen eine Georeferenz dar, die die geographische Einheit mit einem Namen kennzeichnet; sie kann für alle Ebenen als Bezugswert verwendet werden.
- Spalte 18, „WERT“
Spaltentyp: Numerische Werte
Anzahl unterschiedlicher Werte: 12
Kleinster Wert: 128
Größter Wert: 273
Die Spalte beinhaltet für alle Ebenen einen Ergebniswert (y-Achsen-Wert).
- Spalte 20, „ZEIT“
Spaltentyp: Numerische Werte
Anzahl unterschiedliche Werte: 4
Kleinster Wert: 1991
Größter Wert: 1994
Die Spalte enthält für alle Ebenen einen Zeitbezug, der als Bezugswert (für diese Ergebnisdaten: Bezugsintervall) verwendet werden kann.

MDDL-Beschreibung

```
<!DOCTYPE meta-dat SYSTEM "meta-dat.dtd">

<meta-dat>
  <transfer>
    <transfer-typ>HTTP
    <content-typ>application/pdn-data
    <data-server>www.faw.uni-ulm.de
    <server-port>9876
    <data-path>/integral_bin/SRV_DATA/TMP/878197800-13348
  </transfer>
  <archiv>
    <archiv-file>emission.txt.apz
    <archiv-size>651
    <archiv-comment>UVIS-Daten
    <archiv-time>Thu Oct 30 10:32:31 1997
    <archiv-typ>UVIS!TabTabelle!Emission
    <archiv-label>UVIS-Daten Emission
    <packer-typ>UNZIP
  </archiv>
  <data filename = "emission.txt">
    <data-typ>UVIS!TabTabelle!Emission
    <data-comment>Emission
    <data-size>7287
  </data>

  <chart-info>
    <topic>Luft
    <mapsection>
      <xmin>3502301
      <ymin>5269229
```

```

        <xmax>3588674
        <ymax>5353107
    </mapsection>
    <layer>Kreis
</chart-info>

<data-structure>
<table>
    <column-sep>TAB
    <eol-char>CR
    <eof-char>EOF
    <num-datasets>12
    <num-columns>20

    <column columnname="VE_KENNGROESSEN_NAME" list-number=2>
        <printed-name>KENNNAME
        <value-set value-type=String>
            <num-of-dif-values>1
        </value-set>
        <semantics layer = ALLE>
            <presentation type=Bezugswert>
            <field-ref datatype=String>
        </semantics>
    </column>
    <column columnname="GEBIETSEBENE" list-number=10>
        <printed-name>GEBIETSEBENE
        <value-set value-type=String>
            <num-of-dif-values>1
        <value-set>
        <semantics layer = ALLE>
            <presentation type=NONE>
            <layer-ref>
        </semantics>
    </column>
    <column columnname="NAME" list-number=11>
        <printed-name>GEBIETSNAME
        <value-set value-type=String>
            <num-of-dif-values>3
        </value-set>
        <semantics layer = ALLE>
            <presentation type=Bezugswert>
            <geo-ref>
                <area type=name>
            </geo-ref>
        </semantics>
    </column>
    <column columnname="WERT" list-number=18>
        <printed-name>WERT
        <value-set value-type=Number>
            <num-of-dif-values>12
            <min-value>128
            <max-value>273
        </value-set>
        <semantics layer = ALLE>
            <presentation type=Ergebniswert>
            <value>
        </semantics>
    </column>
    <column columnname="ZEIT" list-number=20>
        <printed-name>JAHR
        <value-set value-type=Number>
            <num-of-dif-values>4
            <min-value>1991
            <max-value>1994
        </value-set>
        <semantics layer = ALLE>

```

```

                <presentation type=Bezugsintervall>
                <temp-ref>
                </semantics>
            </column>
        </table>
    </data-structure>
</meta-dat>

```

Aufgrund der Kennzeichnung einer Spalte z. B. als Georeferenz kann der PDN (d. h. der Decision-Manager) später entscheiden, daß die Daten räumlich dargestellt werden können. Dazu benötigt er dann natürlich weitere Angaben, welche Spalten (z. B. für eine Balkendiagrammdarstellung) als Bezugswerte in Frage kommen (in diesem Fall die Spalte „ZEIT“ und die Spalte „VE_KENNGROESSEN_NAME“) und welche Spalte die Ergebniswerte beinhaltet (im Beispiel die Spalte „WERT“).

Die Auswertung dieser beschreibenden Daten durch den PDN ist im folgenden Abschnitt detaillierter beschrieben.

5 Der UVIS-Client

Der UVIS-Client hat die prinzipielle Aufgabe, mittels eines WWW-Browsers Zugang zur UVIS-Recherche und den Informationen, insbesondere den UVIS-Selektoren, zu ermöglichen. Nach erfolgter Recherche kann der Benutzer auf die UVIS-Datenbestände mittels Selektoren zugreifen. Um die mittels dieser Datenbankabfragekomponente von UVIS selektierten Daten aufzubereiten, sind weitere Anwendungen auf dem UVIS-Client notwendig. Zum einen ist der PDN (Personal-Data-Node) dafür verantwortlich, die Konfiguration des Arbeitsrechners zu erfassen und das „Download“ der Daten und der diese beschreibenden Metadaten zu steuern, ebenso wie den Aufruf der letztendlich für die Aufbereitung notwendigen Applikationen mit entsprechenden Skripten und den darzustellenden Daten zu organisieren. Der PDN beinhaltet einen Decision-Manager, der die Auswertung der Metadaten vornimmt und gemäß der PC-Konfiguration mittels Regeln entscheidet, welche Aufbereitungen für die selektierten Daten möglich sind.

5.1 Der Personal Data Node (PDN)

Der Personal Data Node (PDN) ist die zentrale Komponente der Client-Server-Architektur auf Anwender-Seite. Er verknüpft die einzelnen Bereiche des persönlichen Arbeitsplatzes und führt, falls notwendig, die Schritte Datentransfer, Datenentschlüsselung und Datenkonvertierung durch. Er verfügt über ein Daten-Repository zur persistenten Speicherung von Datenobjekten und Methoden. Dabei werden mit Datenobjekten die auf Datenstrukturen abgebildeten Informationen bezeichnet, Methoden sind Skript-Dateien, die das Wissen um die Verarbeitung bestimmter Informationsstrukturen repräsentieren. Eine ausführliche Beschreibung des PDN findet sich in dem Bericht des IKE über die Anpassung des PDN an UIS-Nutzungen (s. /4/). Die Untersuchungen im Rahmen des UIS haben gezeigt, daß die Zuordnung von Datenobjekten und Methoden zu Werkzeugen und Hilfsmodulen über Entscheidungstabellen unzureichend ist (siehe /2/). Daraufhin wurde am FAW der Decision-Manager als Teilkomponente des PDN entwickelt.

5.2 Der Decision-Manager (DM)

Der Decision-Manager als Bestandteil des PDN erfaßt und bearbeitet komplexe Zusammenhänge zwischen Ein- und Ausgaberräumen von Diensten, Aufbereitungsmöglichkeiten und Werkzeugen. Durch geeignete Entscheidungsmechanismen wird dem Benutzer eine Auswahl der Aufbereitungsmöglichkeiten (z.B. Balkendiagramme auf Karte) mit den zugehörigen Applikationen (z.B. ArcView) visualisiert und zur späteren Bearbeitung zur Verfügung gestellt.

Ein weiterer Schwerpunkt des Decision-Managers ist, neben dem Regelmechanismus, den aufgerufenen Applikationen eindeutige Angaben über Pfade, Karten, Ergebnisdaten usw. weiterzuleiten. Dies wird mit Hilfe zweier ASCII-Dateien realisiert. Es handelt sich hierbei um eine Parameter- und eine Kartenauswahltabelle.

5.2.1 Basisdaten für den Entscheidungsmechanismus

Ein Geoinformationssystem, wie z.B. ArcView, benötigt für eine sinnvolle Aufbereitungsmöglichkeit folgende Parameter:

- Raumbezug,
- minimale und maximale Anzahl von Bezugs- bzw. Ergebniswerten und
- Gebietsebenen.

Ein Raumbezug ermöglicht die Aufbereitung von Diagrammen auf einer Hintergrundkarte. Dieser wird durch einen Gebietsschlüssel oder Koordinaten festgelegt; fehlen diese Angaben, so ist keine Darstellung auf einer Hintergrundkarte möglich.

Die Anzahl der Bezugs- bzw. Ergebniswerte beeinflußt die Aufbereitung insofern, daß z.B. bei zu großen Werten die Oberfläche sehr schnell unübersichtlich werden kann.

Gebietsebenen wie Kreis- Gemeinde oder Regierungsbezirkebene bestimmen zum einen, welche Hintergrundkarte geladen werden, zum anderen ist eine Unterscheidung der Gebietsebenen wichtig, da manche Aufbereitungsmöglichkeiten nur auf bestimmten Ebenen möglich sind.

Damit das Regelwerk sowohl eindimensionale als auch mehrdimensionale Tabellen bearbeiten und auswerten kann, ist es notwendig, die Metadaten sehr genau und präzise zu beschreiben (siehe auch Abschnitt 4.2). Es ist hier nicht mehr ausreichend, nur Angaben über die Struktur, Dateieigenschaften, etc. zu machen. Man muß hier vor allem näher auf die einzelnen Spalten der Ergebnistabelle, insbesondere auf die unterschiedlichen Gebietsebenen und deren Bedeutungen, eingehen. Der folgende Ausschnitt aus einem Metadaten-File zeigt eine solche Spaltenbeschreibung.

```
<column columnname="GEFAEHRDUNGSGRAD" list-number=13>
  <printed-name>GEFAEHRDUNGSGRAD
  <value-set value-type=String>
    <num-of-dif-values>3
  </value-set>
  <semantics layer = BIOTOP>
    <presentation type = Ergebniswert>
```

```

    <field-ref>
  </semantics>

  <semantics layer = GEMEINDE>
    <presentation type=Bezugswert>
      <field-ref datatype = String>
    </semantics>
  </column>

```

In dieser Spaltenbeschreibung ist gut zu erkennen, wie den einzelnen Ebenen (Layer) unterschiedliche Bedeutung zukommt. Obwohl die Spalte auf Biotop-Ebene einen Ergebniswertcharakter aufweist, hat sie auf Gemeinde-Ebene eine andere Bedeutung.

Das liegt daran, daß der Gefährdungsgrad auf Biotop-Ebene als mögliche Ordinatenachsenbezeichnung verwendet werden kann, auf Gemeinde-Ebene würde dies aber keinen Sinn ergeben.

<i>Emission</i>	<i>Gebiet</i>	<i>Jahr</i>	<i>Wert</i>
CO ₂	Bodenseegebiet	1990	95
CO ₂	Ravensburg	1990	102
CO ₂	Biberach	1990	231
CO ₂	Ulm	1990	312
SO ₂	Bodenseegebiet	1990	21
SO ₂	Ravensburg	1990	43
SO ₂	Biberach	1990	21
SO ₂	Ulm	1990	55

Tabelle 2: Auszug aus einer Ergebnistabelle eines UVIS-Selektors

Tabelle 2 zeigt einen kleinen Ausschnitt einer Ergebnistabelle eines UVIS-Selektors. In der ersten Spalte ist die Kenngröße *Emission* und in der dritten die *Jahr*-Spalte aufgetragen, beide sind mögliche Bezugswerte.

Die *Wert*-Spalte beinhaltet die Meßwerte für die vorangegangene Kenngrößenspalte und ist somit die Ergebniswertspalte.

Der Raumbezug wird durch die *Gebiet*-Spalte beschrieben, für diesen gibt es aber auch noch weitere relevante Spalten, die aber nicht immer zusammen auftreten müssen. Tabelle 3 zeigt alle für den Raumbezug relevanten Spalten.

Der Raumbezug ist für alle Applikationen notwendig, die geographische Informationen darstellen können. Zu diesen Applikationen gehört im Moment das Programm ArcView von der Firma ESRI.

<i>Gebietsschlüssel</i>	<i>Gebiet</i>	<i>Hochwert</i>	<i>Rechtswert</i>
8435	Bodenseegebiet	5288211	3530499
8426	Biberach	5331765	3552377
8436	Ravensburg	5296329	3556940

Tabelle 3: Raumbezugsspalten in einer Datentabelle

Bei den Spalten *Hoch-* bzw. *Rechtswert* handelt es sich um Koordinatenangaben aus dem Gauß-Krüger-Koordinatensystem, der Gebietsschlüssel ist eine eindeutige Kennung eines bestimmten Gebietes.

Es wäre wenig sinnvoll, einen Entscheidungsmechanismus zu realisieren, der alle möglichen Aufbereitungsmöglichkeiten der verschiedenen Applikationen prüft, wenn eine oder mehrere dieser Applikationen nicht auf dem PC-Arbeitsplatz installiert sind.

Der PDN führt deshalb eine solche Arbeitsplatzkonfiguration durch. Die gefundenen Komponenten, hier speziell die Werkzeuge zur Darstellung der verschiedenen Diagramme, werden dem Decision-Manager zur Verfügung gestellt, die dieser dann im Entscheidungsmechanismus mit berücksichtigt.

Folgende Komponenten sind Bestandteil des Decision Managers:

5.2.2 ADF (ApplicationDefinitionFile)

Das ADF ist eine bedeutende Komponente des Entscheidungsmechanismus, hier werden Beschreibungen für die jeweilige Aufbereitungsmöglichkeit verwaltet.

Zum einen sind dies Einschränkungen der verschiedenen Applikationen, und zum anderen kann der Benutzer seine Präferenzen in diesem File eintragen.

Einschränkungen sind z.B., daß Excel 7.0 keinen Raumbezug hat, d.h. mit dieser Applikation können keine Werte innerhalb einer Hintergrundkarte visualisiert werden.

Unter Präferenzen sind hier vor allem die Einstellungen für die verschiedenen Darstellungsmöglichkeiten der Diagramme zu verstehen, so kann der Benutzer Einfluß auf die Anzahl der Diagramme pro Karte oder der Balken pro Diagramm nehmen.

Die folgenden Zeilen zeigen einen Ausschnitt aus diesem File:

```
[***** Application Definition File *****]
[ Zeichen innerhalb dieser Klammern sind als Bemerkungen zu verstehen ]

[ Beschreibung der Darstellungsmöglichkeit "Balkendiagramme in ArcView" ]
```

```

<ArcViewBarCharts>
{
    Bigger : Bezugswerte = 2;
    Bigger : Bezugswertspalten = 1;
    Smaller : Ergebniswerte = 3;
    Equal : Raumbezug = 1;
}

[Beschreibung der Darstellungsmöglichkeit "Tortendiagramme in ArcView" ]
<ExcelCakeCharts>
{
    Bigger : Bezugswerte = 2;
    Bigger : Bezugswertspalten = 1;
    Smaller : Ergebniswerte = 3;
}

[Beschreibung der Darstellungsmöglichkeit "Flächendarstellung in ArcView" ]
<ExcelCakeCharts>
{
    Equal : Bezugswerte = 1;
    Bigger : Bezugswertspalten = 2;
    Smaller : Ergebniswerte = 3;
}
.

```

Die Ausdrücke in den eckigen Klammern sind als Kommentar zu verstehen, es können hier alle beliebigen Zeichen vorkommen außer [,], < und >.

Der Ausdruck zwischen den spitzen Klammern beinhaltet die Aufbereitungsmöglichkeit mit der entsprechenden Applikation. Die genaue Beschreibung für diese Aufbereitungs-möglichkeit ist als Block innerhalb der geschweiften Klammern zu finden.

Das Semikolon kennzeichnet das Ende einer Zeile, der Punkt das Ende des Files.

Die verwendeten Begriffe sind selbstsprechend, unter Bezugswert versteht man die Anzahl der möglichen x-Werte und unter Ergebniswert die Anzahl der möglichen y-Werte. Kann die Applikation keine Hintergrundkarten visualisieren, so darf keine Raumbezug-Referenz in der zu beschreibenden Aufbereitungsmöglichkeit vorkommen.

Die Regeln und ihre Verwendung:

Der Block innerhalb der geschweiften Klammern wird im Decision Manager eingelesen und dort nach Regeln sortiert.

Insgesamt sind vier Grundregeln implementiert :

- **Bigger** entspricht dem größer gleich „>=“
- **Smaller** entspricht dem kleiner gleich „<=“
- **Between** entspricht der Formel „... <= x <= ...“

- **Equal** entspricht dem gleich „=“

Die Zeile

```
Bigger : Bezugswerte = 1;
```

in dem Block „Balkendiagramme in ArcView“ sagt somit folgendes aus:

Die zu prüfende Aufbereitungsmöglichkeit kann mehr als einen Bezugswert pro Geo-Referenz darstellen, dies trifft z.B. für die Balkendarstellung zu, jedoch nicht für die Flächendarstellung, da hier pro Gebiet immer nur ein Wert dargestellt werden kann.

5.2.3 Parametertabelle

Die mittels DDE aufgerufene Applikation benötigt, damit sie sinnvolle Diagramme darstellen kann, einige Angaben über Kartenpfad, Kartenausschnitt, Spaltenbezeichnungen, etc.

Da die Applikationen meistens Tabellen schneller und sicherer auswerten können als z.B. ASCII-Files, werden diese Parameter sinnvollerweise vom Decision-Manager in einer Tabelle zusammengefaßt.

5.2.4 Kartentabelle

Applikationen wie ArcView können Diagramme zu bestimmten Gebieten auf einer Hintergrundkarte darstellen. Macht der Benutzer eine Abfrage über mehrere Ebenen, beispielsweise Gemeinden und Regierungsbezirke, so müssen der Applikation die zugehörigen Hintergrundkarten, Bezeichnung und Pfad bekannt sein. Um dieser Anforderung gerecht zu werden, ist eine variable Bestimmung der Karten und Themes notwendig. Hierfür wird eine ASCII-Datei erstellt, die die oben genannten Bestimmungen der einzelnen Karten, Themes und Legenden beinhaltet. Der Aufbau ist dem des ADF (s.o.) sehr stark angeglichen, der Benutzer hat hier ebenfalls die Möglichkeit, selbst Änderungen vorzunehmen, wenn z.B. neue Karten oder Namensänderungen auftreten. Die folgenden Zeilen zeigen wiederum einen Ausschnitt aus diesem File.

```
[Beschreibung der Baden-Württembergkarte ]
<BW>
{

  Pfad ;;
  Name: Baden-Württemberg;
  Kennzifferspalte : LAND_NR;
  Legende ;;
  Typ:Shape;

}
```

```
[ Beschreibung der Kreiskarte ]
<Kreis>
{

  Pfad ;;
  Name: Landkreise_in_BW;
  Kennzifferspalte : KRS_INDEX;
  Legende :;
  Typ:Shape;

}
```

Pro Karte existieren jeweils fünf Elemente, die beschrieben werden müssen, im einzelnen sind das

- der Pfad, in dem die Karte liegt; wird hier nichts angegeben, so greift der Decision-Manager auf das Standard-Kartenverzeichnis PDNHOME/Map/ zu,
- der Name der Karte, dieser kann nach eigenen Wünschen gewählt werden,
- die Kennzifferspalte der Kartentabelle, diese Bezeichnung muß mit der Kennzifferspalte der Kartentabelle (*.dbf) übereinstimmen,
- die Legende, hier muß ebenfalls der vollständige Pfad für das Legenden-File angegeben werde, ansonsten wird auf das Standard-Kartenverzeichnis verwiesen und
- der Typ der Karte, der bei ArcView-Karten meistens vom Typ SHAPE ist.

5.2.5 Das Regelwerk

Beim Regelwerk handelt es sich um das eigentliche Herzstück des Decision-Managers. Hier finden Vergleiche zwischen den Werten des SGML-Moduls und dem ADF-Modul statt. Basierend auf einfachen Regeln wie <, >, =, ..., werden die gesamten Aufbereitungsmöglichkeiten auf ihre Brauchbarkeit bezüglich der graphischen Darstellungsmöglichkeit mit bestimmten Applikationen hin überprüft.

Das folgende Beispiel soll dieses Regelwerk verdeutlichen:

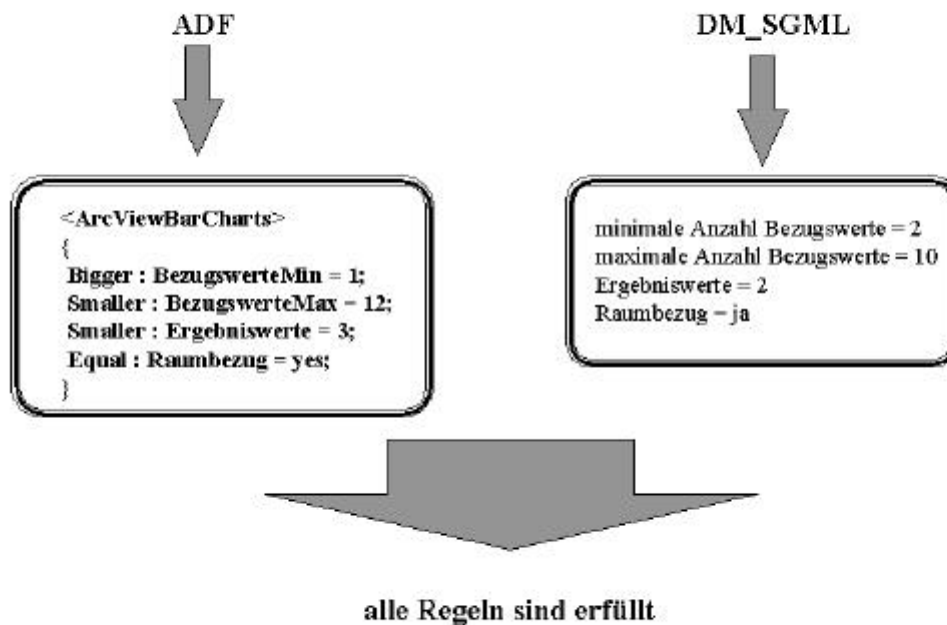


Abbildung 6: Funktionsweise des Regelwerkes

Zuerst wird aus der ADF-Tabelle eine Darstellungsmöglichkeit extrahiert und die einzelnen Schlüsselworte mit den entsprechenden Werten (z.B. Bezugswerte = 1) nach Regeln sortiert.

Daraufhin werden die Einträge der beiden Module miteinander nach den vorgegebenen Regeln verglichen.

So wird z.B. der Eintrag *Bezugswerte* aus der ADF-Table mit dem namentlich gleichen Eintrag der SGML-Struktur mit der Regel *Bigger* verglichen, mathematisch sieht dieser Vergleich dann so aus:

```

if (SGML-BezugswerteMin >= ADF-Table-BezugswerteMin) TRUE;

else FALSE;

```

In diesem Beispiel werden somit alle Regeln erfüllt, und damit kann diese Darstellungsmöglichkeit mit in die Liste aller möglichen Darstellungsmöglichkeiten übernommen werden. Wird aber eine der Regeln nicht erfüllt, so wird diese Darstellungsmöglichkeit aus der Liste „gestrichen“.

Aus Benutzersicht hat der DM nach Analyse der Metadaten und der Konfigurationen des Arbeitsplatzrechners die möglichen und sinnvollen Aufbereitungsarten selektiert. Diese Ergebnismenge der Darstellungsarten hängt stark von der auf dem Client vorhandenen Konfiguration ab, d.h. nur wenn die erforderlichen Applikationen und Zusatzskripte vorhanden sind, wird die entsprechende Darstellungsart aufgelistet und dem Benutzer zur Auswahl gestellt. Abbildung 7 zeigt das dazu gehörige Fenster mit der Auswahlliste der verschiedenen Aufbereitungsmöglichkeiten, wenn z.B. ArcView mit den für UVIS realisierten Skripten installiert ist.

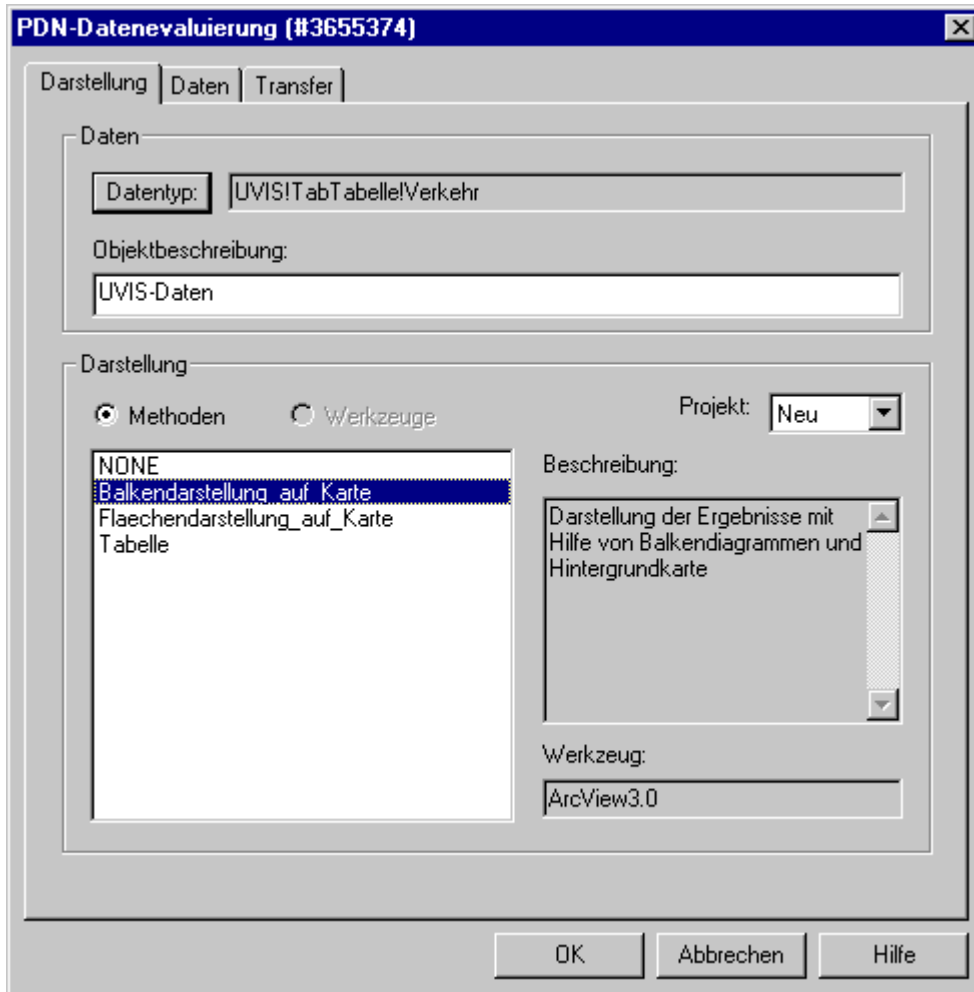


Abbildung 7: Auswahlliste der Aufbereitungsmöglichkeiten

Wählt der Benutzer eine Aufbereitungsart aus, so veranlaßt dies den PDN, die Daten auf den Client zu transferieren und die entsprechende Applikation mit den Erweiterungsmakros und -skripten aufzurufen.

6 Die ArcView-Aufbereitung

Die ArcView-Aufbereitung wird im Anschluß an den PDN-Manager gestartet. Durch den PDN-Manager werden die abgerufenen Daten aufgearbeitet und in eine darstellungsgerechte Form gebracht.

Bei dieser Aufbereitung wird das Geoinformationssystem ArcView in einer für UVIS gefertigten Konstellation aufgerufen. Diese Konstellation ist in der ArcView-Datei „ufis_gda.apr“ abgelegt, die Informationen zur Aufbereitung der Daten in einer Parametertabelle „paramet.txt“. Auch die Daten selbst sind in einer ASCII-Tabelle abgelegt. Zuerst werden die Parameter ausgewertet, dann die Daten eingelesen und schließlich die entsprechende Hintergrundkarte geladen.

6.1 Die Oberfläche

Die UVIS-ArcView-Oberfläche enthält oben drei Funktionsleisten und ein Hauptfenster, das mit UVIS betitelt ist. Das Hauptfenster besteht im wesentlichen aus der Zeichenfläche, in der die Karten dargestellt werden, und der Legendenspalte links davon (Abbildung 8).

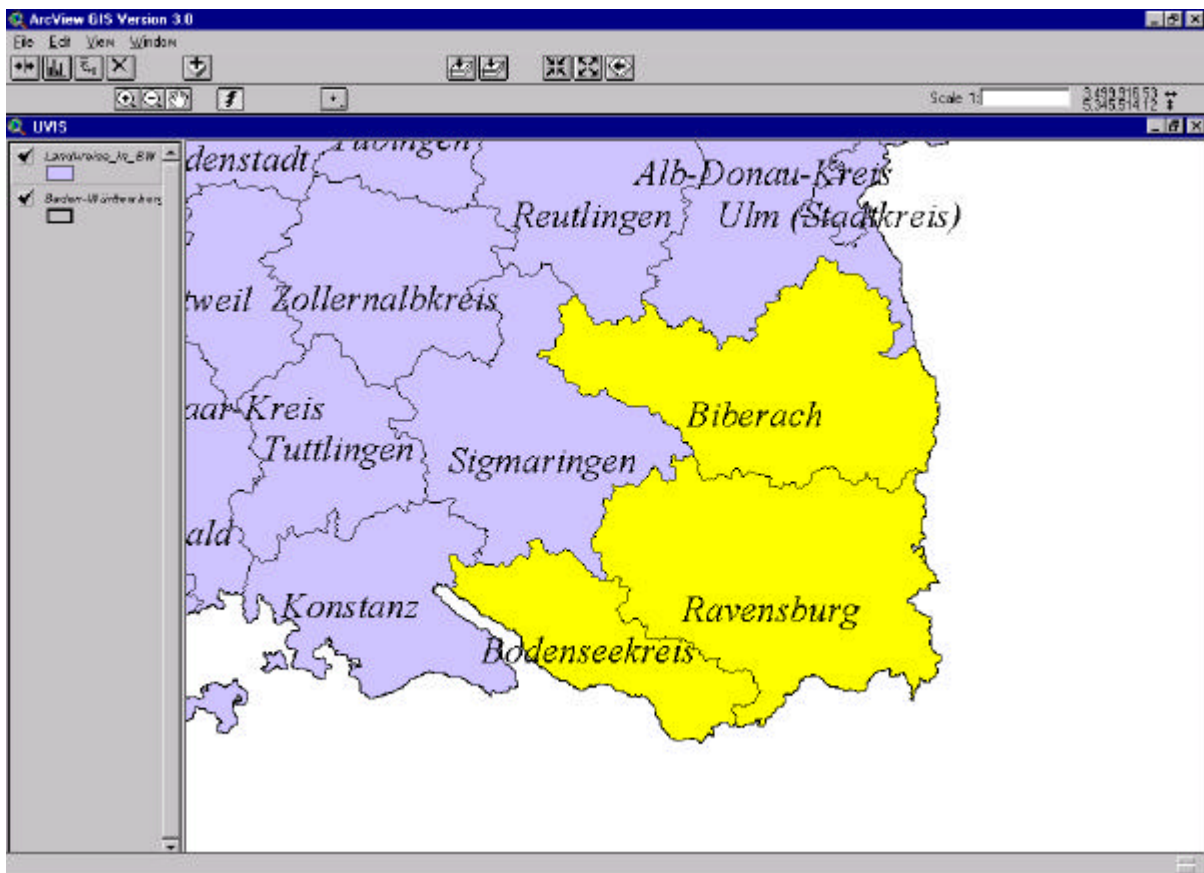


Abbildung 8: ArcView-Aufbereitung nach dem Start (die eingetragenen Karten können sich dabei unterscheiden)

Die Funktionsleisten

Die drei Funktionsleisten sind im einzelnen die Menüleiste, die Buttons und die Werkzeugleiste.

Die Menüs

Die Menüleiste ist eine reduzierte Standard-Menüleiste. Sie enthält keine zusätzlichen Funktionen.

Die Buttons

Mit den Buttons werden bestimmte Aktionen gestartet. Im einzelnen haben sie folgende Bedeutung:



Wechsel zwischen der UVIS-Oberfläche und der Standard-ArcView-Oberfläche.



Erzeugen einer detaillierten Grafik aus einer aktiven Grafik in einem View.



Auswahl aller möglichen Hauptmerkmale, bzw. Kenngrößen.



Setzen des Ausschnitts so, daß alle Karten vollständig gezeichnet werden.



Zoomen in die Karte (bei gleichem Mittelpunkt).



Zoomen aus der Karte (bei gleichem Mittelpunkt).

Die Werkzeuge

Im Gegensatz zu den Buttons ist mit der Aktivierung eines Werkzeuges keine Aktion verbunden. Es wird lediglich eingestellt, was geschehen soll, wenn im Kartenfenster etwas angeklickt wird. Die Werkzeuge bedeuten:



Kleineren Ausschnitt wählen. Ein in die Karte gezogenes Rechteck bildet den neuen Ausschnitt



Größeren Ausschnitt wählen. Ein in die Karte gezogenes Rechteck bildet den neuen Mittelpunkt.



Zeichenobjekt verschieben. Ein in der Karte angeklicktes und damit selektiertes Objekt kann mit der Maus verschoben werden.



Hotlink ausführen. Für ein in der Karte angeklicktes Gebiet wird ein Programm ausgeführt.



Zeichnen. Beim Anklicken dieses Werkzeuges erhält man eine Liste der zu Verfügung stehenden Zeichenelemente.

6.2 Das Darstellungsfenster

In das eigentliche Darstellungsfenster für die Datenaufbereitung (View UVIS) werden nach dem Start die Karten geladen, die zur Auswertung der Daten benötigt werden. Wenn die Daten auf Kreisebene vorliegen, wird also zumindest die Kreiskarte geladen sein. Wenn es sich beispielsweise um Biotop-Daten handelt, wird auch eine Biotop-Karte dabei sein.


Jede Karte ist in der Legende aufgeführt. Vor dem Namen der Karte ist ein Feld mit einem Häkchen geklickt, verschwindet es zusammen mit der Darstellung der Karte. So kann eine Karte ein- bzw. ausgeblendet werden.

Um eine Karte (Theme) zu aktivieren, wird auf den Kartennamen oder auf die Legendenbeschreibung geklickt. Die gesamte Legendenbeschriftung der Karte ist dann hervorgehoben, wie beispielsweise die Karte „Landkreise in BW“. Sämtliche Werkzeuge und Buttons wie die Kartenerstellung und der Hotlink (Blitz) beziehen sich auf die aktive Karte.

6.3 Diagrammerstellung

Die wichtigste Funktion, die zur Verfügung steht, ist die graphische Darstellung der Daten zu einem Gebiet, also z.B. zu einem Landkreis, falls die Daten landkreisspezifisch sind.

Dazu muß zuerst sichergestellt werden, daß die Karte, auf die sich die Daten beziehen, aktiv ist.

Zu Beginn ist dies der Fall. Das Hotlink-Werkzeug ist standardmäßig zum Start von ArcView ebenfalls aktiviert. 

6.3.1 Balkendiagramme

Einzeichnen von Diagrammen

Nun wird in das Gebiet geklickt, für das die Daten aufbereitet werden sollen. Dadurch werden die Berechnungen für die Aufbereitung gestartet und schließlich das Darstellungsfenster neu gezeichnet. Ein Beispiel für diese Darstellung ist in Abbildung 9 zu sehen.

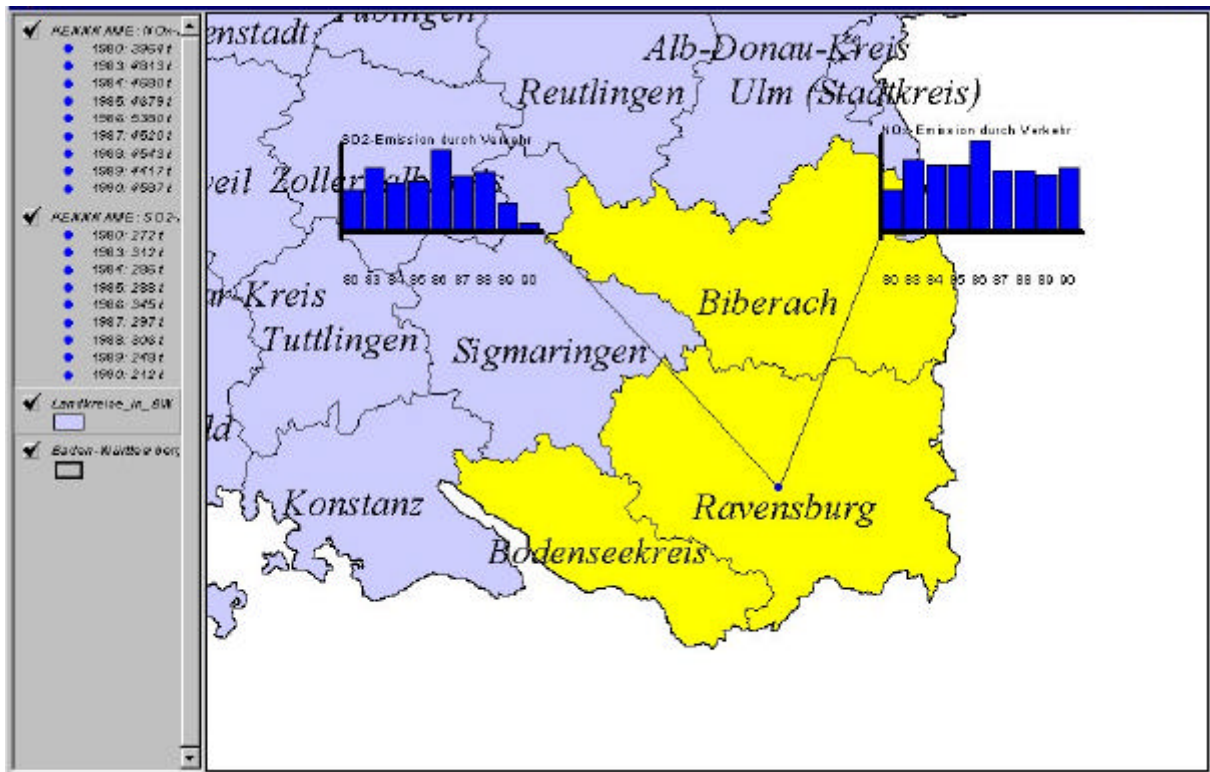


Abbildung 9: Diagramme im Kartenfenster mit selektiertem Kreis Ravensburg

Die Diagramme werden um den Rand des Kartenfensters herum plaziert und mit dem selektierten Gebiet verbunden. Das selektierte Gebiet ist zusätzlich farbig (gelb) hervorgehoben. Jedes Diagramm wird in der Legende erklärt.

Welches Diagramm zu welcher Legende gehört, kann man an den gleichen Titeln für Legende und Diagramm erkennen.

Wird ein anderes Gebiet ausgewählt², werden die Diagramme des alten Gebietes gelöscht und die Diagramme für das neue Gebiet erstellt.

Wie die Karten können auch die Diagramme mit den Häkchen an der Legende ein- bzw. ausgeblendet werden.

Zusammenstellen der Diagramme

Die von der Parametertabelle übergebenen Felder sind in diesem Beispiel die fünf Felder: Gebiet, Kenngröße, Zeit, Wert und Einheit. Diese Felder sind für die Aufbereitungsmöglichkeit ausschlaggebend.

² Der Mauszeiger erscheint immer noch als „Blitz“. Das Hotlink-Werkzeug ist immer noch aktiv.

<i>Gebiet</i>	<i>Kenngröße</i>	<i>Zeit</i>	<i>Wert</i>	<i>Einheit</i>
Biberach	SO ₂ -Emissionen durch Verkehr	1983	266	t
Biberach	SO ₂ -Emissionen durch Verkehr	1984	247	t
Biberach	SO ₂ -Emissionen durch Verkehr	1985	249	t
Ravensburg	SO ₂ -Emissionen durch Verkehr	1983	312	t
Ravensburg	SO ₂ -Emissionen durch Verkehr	1984	286	t

Tabelle 4: Auszug aus einer Datentabelle

Wie werden nun die Daten in ein oder sogar mehrere Balkendiagramme umgesetzt? Dadurch, daß ein Gebiet in der Karte ausgewählt wird, können aus der Tabelle die zugehörigen Daten herausgelesen werden. Dann wird durch einen weiteren Vorgang der zugehörige Wert zu einem bestimmten Zeitpunkt bestimmt. Die verschiedenen Werte werden nun in Balkenhöhen umgesetzt. Die Einheit spielt bei der Balkenerstellung keine Rolle, sie wird der Legende beigefügt.

Mit dem Feld *Kenngröße* kann nun der Bezug auf Daten bestimmt werden. Eine andere Kenngröße bedeutet ein neues Diagramm.

Interpretation der Diagramme

In einem Diagramm werden in der Regel vergleichbare Werte dargestellt, wie z.B. die SO₂-Emissionen durch Verkehr von mehreren Jahren. Das Diagramm vermittelt dann einen guten Eindruck über die Entwicklung dieser Meßwerte.

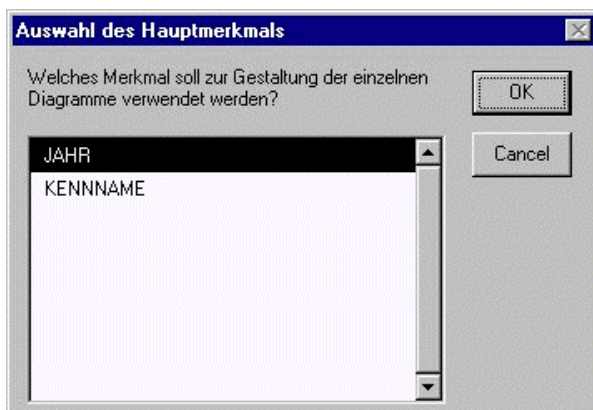




Abbildung 10: Auswahl des Hauptmerkmals

Eine Diagrammerstellung kann auch dann sinnvoll sein, wenn die Verwendung von Kenngröße und Zeit vertauscht wird. Dazu klickt man auf den Button . Es erscheint ein Menü mit den möglichen Merkmalen (Abbildung 10). Das erste in der Liste ist das bisherige Hauptmerkmal.

Das Hauptmerkmal wird gewechselt, indem man auf das neue Merkmal und dann auf OK klickt.

Wenn dann erneut die Diagrammerzeugung aktiviert wird, werden die Diagramme nach Jahren geordnet mit Balken pro Kenngröße dargestellt.

Aufbereitung eines Diagramms

Ein einzelnes Diagramm kann separat weiterverarbeitet werden. Dazu muß das Diagramm zum aktiven Theme gemacht werden, indem es in der Legende angeklickt wird. Die Legende ist dann hervorgehoben, und man kann auf den Button  klicken. Dann erscheint das Diagramm in einem neuen Fenster (s. Abbildung 11). Das Fenster zeigt einen anderen Dokumententyp in ArcView, einen Chart. Deshalb gibt es auch andere Funktionsleisten. Diese wurden für die UVIS-Aufbereitung nicht verändert.

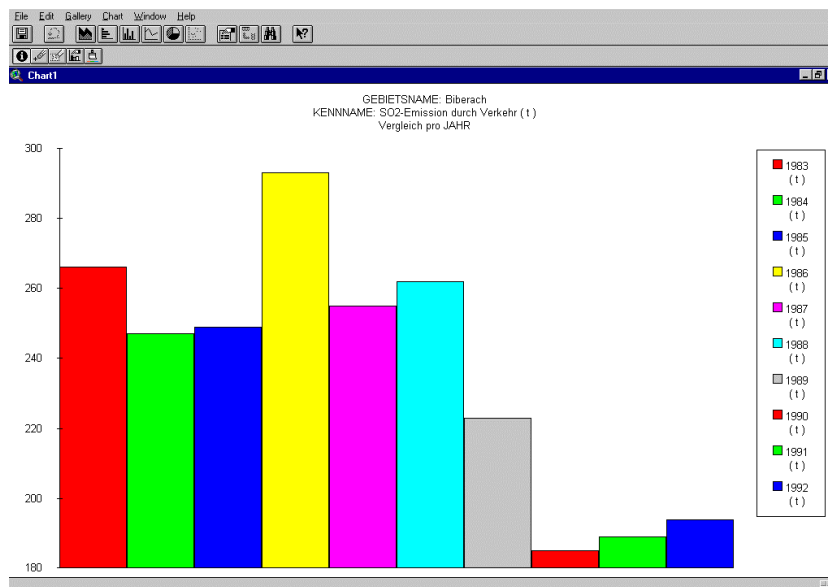



Abbildung 11: Erstellung einer Chart aus einem aktiven Diagramm

Man kommt zum UVIS-View zurück, indem man das Chart-Fenster schließt.

6.3.2 Flächendarstellung

Erstellung einer Flächendarstellung

Da bei einer Flächendarstellung nur eine Kenngröße pro Bezugswert dargestellt werden kann, z.B. die SO2-Emissionen durch Verkehr für das Jahr 1993, wird von ArcView die erste mögliche Konstellation gewählt. Mit dem Button  kann nun zwischen allen Kenngrößen gewechselt werden. Dazu erscheinen folgende Menüpunkte:

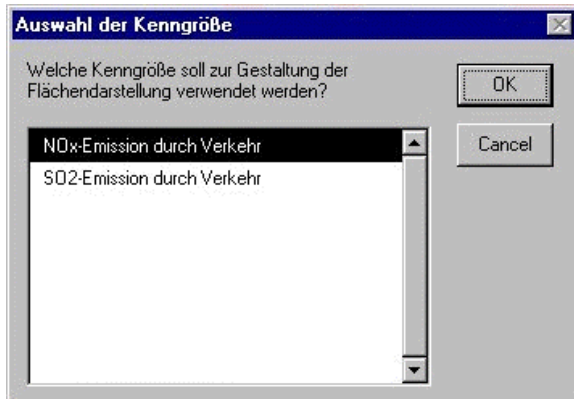


Abbildung 12: Auswahl der Kenngröße

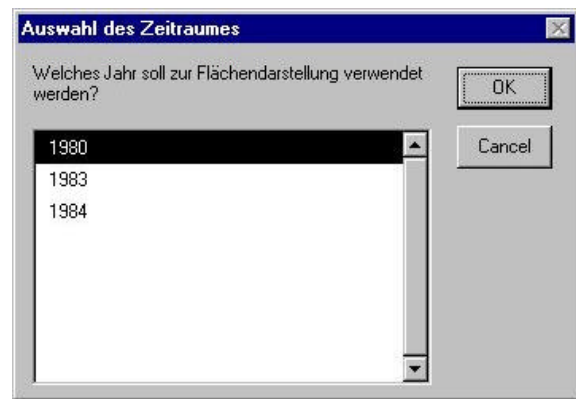
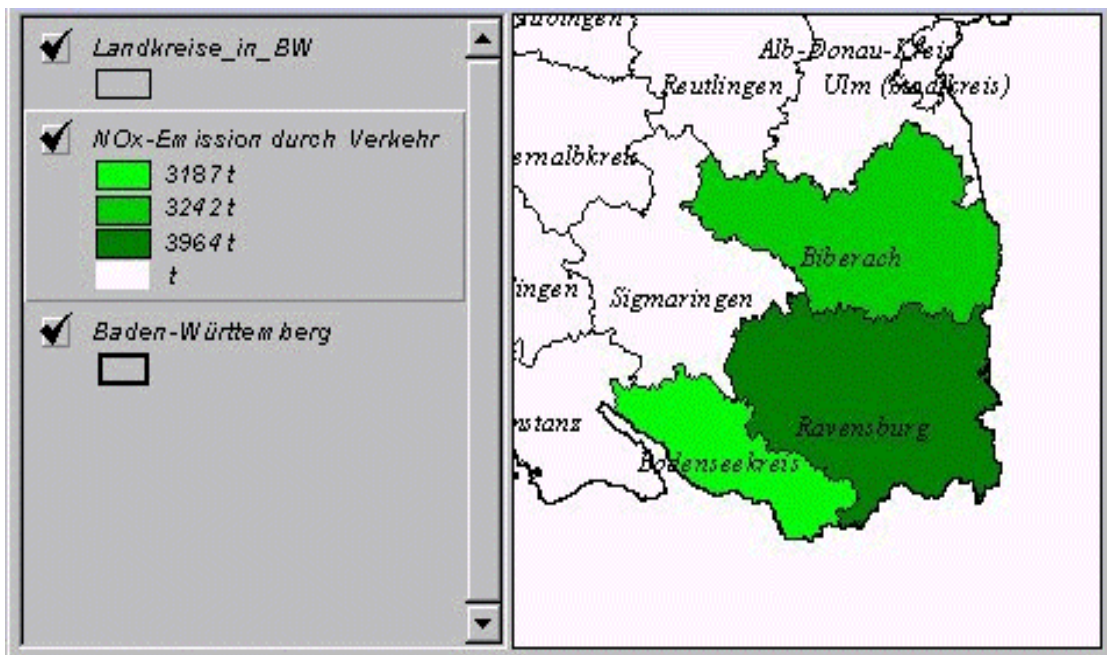


Abbildung 13: Auswahl eines Zeitraumes

Nach Auswahl der neuen Kenngröße wird nun eine neue Flächendarstellung erstellt. Gebiete ohne Daten erscheinen als weiße Flächen. Sonstige werden in farbigen Abstufungen gestaffelt gegliedert (Abbildung 14).

Abbildung 14: Flächendarstellung von NO_x-Emissionen

Zusammenstellen der Flächendarstellung

Prinzipiell ist die Aufarbeitung der erhaltenen Daten die gleiche wie in der Balkendarstellung. Es wird lediglich die Datentabelle nach anderen Kriterien untersucht. Wie schon erwähnt, ist die Bedingung zur Erstellung einer Flächendarstellung die, daß nur eine Kenngröße zu einem Bezugswert aufgetragen werden kann. Im Gegensatz zu der Balkendarstellung ist es jedoch hier ausgeschlossen, daß Werte verglichen werden, die zueinander keinen Bezug haben. Es erfolgt immer eine Selektion der Datentabelle nach einer Kenngröße zu einem bestimmten Zeitpunkt.

Interpretation der Diagramme

Es ist auf einem Blick möglich, sich eine Übersicht über alle vergleichbaren Kreise zu machen und sich z.B. das Ausmaß einer Umweltbelastung zu veranschaulichen. Nun kann eine bestimmte Kenngröße, z.B. SO₂-Emissionen durch Verkehr, auf einer Karte dargestellt werden und mit benachbarten Kreisen verglichen werden, d.h. die Belastung der einzelnen Kreise kann nun im Verhältnis zu anderen Kreisen betrachtet werden.


Die genauen Daten, die für einen Landkreis herangezogen wurden, können mit dem Hotlink-Werkzeug  und dem Anklicken eines der farbigen Kreises eingesehen werden (Abbildung 15).



Abbildung 15: Datensatz für ein Gebiet

6.4 Beenden der ArcView-Aufbereitung

Die ArcView-Aufbereitung wird mit „Exit“ aus dem Menü „File“ beendet. Falls ein Projekt gespeichert werden soll, ist folgendes zu tun::

- Aus dem Menü „Window“ ist das Fenster „ufis_gda.apr“ auszuwählen. In diesem Fenster wird das Projekt verwaltet.
- Dann wählt man aus dem Menü „File“ den Punkt „Save Project As...“; dort kann ein anderer Namen angegeben werden (das Projekt „ufis_gda.apr“ selbst ist schreibgeschützt, was auch nicht geändert werden sollte).

7 Ausblick

Die erste Version des Systems UFIS war in erster Linie für die Führungsebene als Benutzerschicht konzipiert, woraus der hohe Anspruch an eine aussagekräftige visuelle Aufbereitung der selektierten UVIS-Daten resultiert. Durch die Neukonzeption des UIS mit neuen, auf WWW-Technologie basierenden Systemarchitekturen (siehe z. B. /3/, /5/), ist es nun bereits mit minimal ausgestatteten Rechnern, die Internet-Anschluß besitzen, möglich, Zugang zu bereits nach diesen Konzepten realisierten UIS-Komponenten zu erhalten. Auch das einst mono-

lithische System UFIS wurde in der neuen Version UVIS diesen Systemkonzepten angepaßt. Dadurch wurde es möglich, längst bekannten, bislang mit dem alten System noch nicht in Verbindung gebrachten Anforderungen gerecht zu werden. Durch die Öffnung des Systems für die Mitarbeiter in den Behörden konzentrierte sich die Konzeption der Aufbereitungskomponente auf die Integration der UVIS-Daten in bereits im Einsatz befindliche Applikationen. Somit haben die Mitarbeiter die Möglichkeit, aus ihrer gewohnten Arbeitsumgebung auf UVIS-Daten zuzugreifen, diese komfortabel visuell aufzubereiten und in ihnen bereits bekannte Applikationen zu integrieren. Hierfür sind zum einen zusätzliche Skripte (z.B. für ArcView) erforderlich, die eine UVIS-spezifische Darstellung realisieren, und zum anderen intelligente Komponenten wie PDN und Decision-Manager, die den Ablauf bei der Aufbereitung automatisieren. Durch weitere Installationen dieser Komponenten des UVIS-Clients könnte der Nutzerkreis innerhalb der Verwaltung weiter ausgedehnt werden.

Die neue Systemarchitektur kommt aufgrund der weiten Verbreitung der verwendeten WWW-Technologien ohne besonderes Zutun dem Anspruch der Öffnung von Informationen für den Bürger entgegen. Grundsätzlich sind ausgewählte Daten des UVIS auch für die Bürger relevant, jedoch müßten spezielle Sichten auf die Gesamtdatenbestände gebildet oder auch Auszüge des Gesamtbestandes gespiegelt werden. Durch die weite Verbreitung von Internetanbindungen auch in privaten Haushalten wäre ein breiter Zugang der Bürger zu reduzierten UVIS-Daten problemlos möglich; der Nutzerkreis könnte so weiter ausgedehnt werden. Da jedoch nicht davon ausgegangen werden kann, daß die für UVIS spezifischen Zusatzapplikationen und Skripte auf jedem Zugangssystem installiert werden können, müßten serverseitige Aufbereitungsmöglichkeiten bereitgestellt werden. Außerdem sollte eine Aufbereitung über serverseitig abrufbare Module, die clientseitig aktiv werden, vorgesehen werden. Hierbei werden folgende Realisierungen empfohlen:

- Einsatz von **MS-ActiveX**-Komponenten, die eine clientseitige Aufbereitung ohne Zusatzapplikationen ermöglichen.
- **Java-Applets** (z. B. GisTerm), die die UVIS-Daten über serverseitig abrufbare Applets clientseitig aufbereiten.
- Konzeption **Bildarchiv**; fertig aufbereitete Bilder werden mit den zugehörigen Parametern abgespeichert und können dann vom Client abgerufen werden.

Für den Bürger könnte somit ein externer Server realisiert werden, der einen reduzierten, für die Öffentlichkeit zugänglichen Datenbestand verwaltet. In einer ersten Version könnten regelmäßig durch Abfragen generierte Bilder in ein Bildarchiv auf den öffentlichen Server übertragen werden. In einer Weiterentwicklung sollten dann abrufbare Komponenten wie GisTerm und ActivX-Komponenten eingesetzt werden, die eine komfortable Aufbereitung ermöglichen.

8 Literatur

- /1/ Henning I., Wiest G., Schmidt F. (1996): UFIS II im Rahmen des UIS Baden-Württemberg, Informationsrecherche auf neuen Wegen, 11. Symposium, Tagungsband, Metropolis-Verlag, Marburg.
- /2/ Mayer-Föll, R. / Jaeschke, A. (Hrsg.) (1996): Projekt GLOBUS – Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten im Umweltinformationssystem Baden-Württemberg – Phase III – 1996, Abschlußbericht, Bericht FZKA 5700. Umweltministerium Baden-Württemberg, Forschungszentrum Karlsruhe.
- /3/ Mayer-Föll, R. / Strohm, J. / Schultze A. (1996): Umweltinformationssystem Baden-Württemberg – Überblick Rahmenkonzeption, in: Informatik für den Umweltschutz, 10. Symposium, Tagungsband, Metropolis-Verlag, Marburg.
- /4/ Kopetzky, R. Schmidt F. (1996): Entwurf und Implementierung eines Klienten zur Verarbeitung von Informationen aus verteilten Systemen, IKE Institut für Kernenergetik und Energiesysteme, Universität Stuttgart
- /5/ Riekert W.-F., Gaul M., Klügl G., Wiest G., Henning I., Schmidt F. (1996): Management verteilter und heterogener Informationsquellen für das UIS-Baden-Württemberg, 10. Symposium, Tagungsband, Metropolis-Verlag, Marburg.

Die Anpassung des Personal Data Node an UIS-Nutzungen

Bericht des IKE über die Arbeiten im Teilprojekt PDN-UIS

*F. Schmidt, A. Lurk, R. Kopetzky, J.-E. Schulz,
Universität Stuttgart,
Institut für Kernenergetik und Energiesysteme (IKE),
Pfaffenwaldring 31,
70550 Stuttgart*

1. EINLEITUNG	59
2. DER PDN ALS REFERENTEN-ARBEITSPLATZ IM UIS.....	61
2.1 NUTZUNG DES PDN IM RAHMEN VON UFIS-II.....	62
2.2 NUTZUNG DES PDN IM RAHMEN DES KFÜ	64
2.3 WEITERENTWICKLUNG DER LOGISCHEN ARCHITEKTUR	64
2.4 DIE STRATEGIEKOMPONENTE	66
2.4 DIE REPOSITORY KOMPONENTE.....	68
2.6 DIE KOOPERATIONSKOMPONENTE.....	70
2.6.1 <i>Das Referenzmodell Kooperation und Kommunikation</i>	70
2.6.2 <i>Soft- und Hardware Anforderungen für den Einsatz von CSCW</i>	72
2.6.3 <i>Teamorientierte Erstellung hypermedialer Umweltberichte</i>	72
2.6.4 <i>Dokumentation einer Ausbreitungsrechnung</i>	74
2.7 DIE WORKFLOW-KOMPONENTE	75
3. SICHERHEITSKONZEPTE FÜR DAS UIS.....	76
3.1 ANFORDERUNGEN AN EINE FIREWALL BEI NUTZUNG VON COMPUTER SUPORTED KOOPERATIVE WORKING TECHNIKEN	77
3.2 DATENÜBERTRAGUNG UNTER NUTZUNG DES VERSCHLÜSSELUNGSPROTOKOLLS SSL	78
4. LITERATUR.....	79

1. Einleitung

Der Personal Data Node wurde in den vergangenen Jahren als Framework für einen Klienten im Rahmen des UIS entwickelt. Dabei lag dem Framework vor allem das Bild eines erfahrenen Nutzers (Referentenebene) zugrunde. Dieser Nutzer ist eingebettet in ein sehr heterogenes Umfeld, in dem er Aufgaben hoher Komplexität zu bearbeiten hat. Dazu benötigt er zum einen fachspezifisches Wissen und dieses Wissen umsetzende Werkzeuge und Methoden und zum anderen den Zugang zu einer Vielzahl von Dienstleistungen, die entweder von Kolleginnen und Kollegen oder aber von externen Anbietern erbracht werden.

Ein wichtiger Bestandteil dieses Nutzerbildes ist die Fähigkeit, zur Bearbeitung von Aufgaben Dienste in Anspruch zu nehmen. Dienste sind in diesem Zusammenhang abgeschlossene Einheiten, die eine spezielle Aufgabe erfüllen. Sie werden von Diensteanbietern erbracht und von Dienstenutzern verwendet. Der Diensteanbieter ist in der Regel ein Spezialist, der in der Lage ist, den Dienst sachgerecht einzusetzen, zu warten und entsprechend dem Stand der Wissenschaft, der Technik und der Gesetzgebung weiter zu entwickeln.

Kann ein Nutzer verschiedene Dienste zur Lösung von Aufgaben in Anspruch nehmen, so ist er Teil eines dienstebasierten Systems. Grundsätzlich gilt, daß ein dienstebasiertes System ein offenes und verteiltes System sein sollte. Ein offenes System ermöglicht den dynamischen Ausbau des Dienstangebots. Ein verteiltes System bildet die Grundlage, Dienste lokal von Diensteanbietern pflegen zu lassen und zugleich in Form von Dienstangeboten zur Verfügung zu stellen. Ein dienstebasiertes System, das aus einer Menge von Diensten und Agenten /1/ besteht, die Informationen erzeugen und/oder benutzen und die durch Kommunikationsbeziehungen und Kooperationsbeziehungen miteinander verbunden sind, heißt Informationssystem. Das Umwelt-Informationssystem Baden-Württemberg ist in diesem Sinne ein Beispiel für dienstorientiertes System. Die von den Diensten eines Informationssystems zu erfüllenden Aufgaben bestehen dabei in der Aufnahme, Verarbeitung und Aufbereitung von Informationen. Aus dieser Auflistung lassen sich vier Klassen von Diensten identifizieren:

- Dienste zur *Aufnahme von Informationen* (z.B. Datenbanken)
- Dienste zur *Verarbeitung von Informationen* (z.B. Konverter, Simulationsrechnungen, ...)
- Dienste zur *Aufbereitung von Informationen* (z.B. Kartendienste, Dienste zur Erstellung von Geschäftsgraphiken, ...)
- Unterstützende *Systemdienste* zur Umsetzung der Struktur des Informationssystems (z.B. Strategie-Service, Workflow-Komponente, ...).

Dem vorgestellten Bild eines möglichen UIS-Nutzers und der daraus abgeleiteten Klassen von Diensten wurde durch zwei Architekturentscheidungen im UIS Rechnung getragen. Das erste ist die Aufspaltung des ursprünglich monolithischen Systems in ein verteiltes System, mit einer klaren Client-Server-Struktur. Als Middleware wurden in den letzten Jahren vor allem das World Wide Web und CORBA untersucht. Beide haben Stärken und Schwächen, was dazu führt, daß es für die hier angedachten Nutzungen mittelfristig wünschenswert erscheint, Elementen beider Ansätze zur Verfügung zu haben. Darauf wird an anderen Stellen dieses GLOBUS-Berichtes detaillierter eingegangen.

Die zweite Entscheidung betrifft die Architektur des Klienten und schlägt sich nieder in den Merkmalen, die den PDN auszeichnen. Sie wurde von zwei Forderungen bestimmt:

- Offenheit des Zugangs zu Informationen und Diensten verschiedenster Quellen,
- Anpassungsfähigkeit des Systems an individuelle Nutzungsbedürfnisse.

Der Realisierung lagen zwei Basisideen zugrunde:

- Die Verwendung einer komponentenorientierten Struktur und
- die klare Trennung zwischen Komponenten zur Steuerung von Arbeitsabläufen zur Verwaltung von Daten, zur Ankopplung von Werkzeugen und den für eine bestimmte Tätigkeit charakteristischen Applikationen.

Eine Arbeitsumgebung, die diese Komponenten zur Verfügung stellt, nennen wir einen logischen Klienten. In den Phasen II und III des Projektes GLOBUS wurde mit dem Personal Data Node (PDN) ein Framework für einen logischen Klienten zum Arbeiten in verteilten Systemen entwickelt, mit dessen Hilfe die prinzipiellen Anforderungen an solch einen Klienten ermittelt und im prototypischen Einsatz getestet werden sollten. Dabei wurde im ersten Ansatz der logische Klient mit dem physikalischen Klienten gleichgesetzt.

In der Phase IV des Projektes GLOBUS galt es, nun den physikalischen Klienten an aktuelle UIS-Nutzungen anzupassen und gleichzeitig das Framework und ausgewählte Komponenten so weiter zu entwickeln, daß sie auch für weitergehende Aufgaben eingesetzt werden können. Das geschah in 4 Arbeitspaketen:

Arbeitspaket 1:

Anpassung des Personal Data Node / Dienstvermittler an UIS-Nutzungen

Arbeitspaket 2:

Bereitstellung eines Repositorydienstes unter Verwendung von Erfahrungen mit der Kommunikations-Auftragsverwaltung (KAV) aus TZUI.

Arbeitspaket 3:

Einführung von Techniken kooperativen Arbeitens in das UIS.

Arbeitspaket 4:

Entwicklung eines Frameworks für Sicherheitskonzepte im UIS zusammen mit FAW, LfU/ITZ und UVM/IKS-UVM.

Während der Entwicklung des Frameworks und beim Arbeiten mit dem PDN stellte sich heraus, daß der logische Klient keineswegs nur für Anwendungen im Rahmen des UIS BW von Nutzen war. Vielmehr läßt sich das geschilderte Nutzerbild in einer Vielzahl von anderen Bereichen wiederfinden. Entsprechend dieser Erkenntnis haben wir versucht, den logischen Klienten auch in anderen Forschungsvorhaben einzusetzen und dafür Ergebnisse aus diesem Vorhaben im Projekt GLOBUS mit zu nutzen. Dies ist uns bei den ersten drei Arbeitspunkten in vielfältiger Weise gelungen und hat sowohl zur Bereitstellung von verbesserten Komponenten als auch zur Entwicklung einer allgemeinen Formulierung des Frameworks beigetragen. Vor allem folgende Projekte sind hierbei zu nennen:

- Das Graduiertenkolleg Parallele und Verteilte Systeme mit den Beiträgen von R. Kopezky zur Entwicklung eines Framework für einen flexiblen, agentenunterstützten Klienten zum Arbeiten in offenen, verteilten Systemen.
- Das Softwarelabor der Universität Stuttgart mit den Beiträgen von J-E. Schulz zur Entwicklung eines Repositorydienstes und der Einführung des Workflow-Gedankens.
- Das DFN- (Deutscher Forschungs-Netz-Verein) Projekt B-WiN4M Baden-Württemberg mit den Beiträgen von A. Lurk zum kooperativen Arbeiten und zur Verfügbarmachung dieser Techniken bei den Partnern des Projektes GLOBUS.

2. Der PDN als Referenten-Arbeitsplatz im UIS

Abb. 1 zeigt die Architektur des Klienten, wie sie im Rahmen von GLOBUS III umgesetzt wurde.

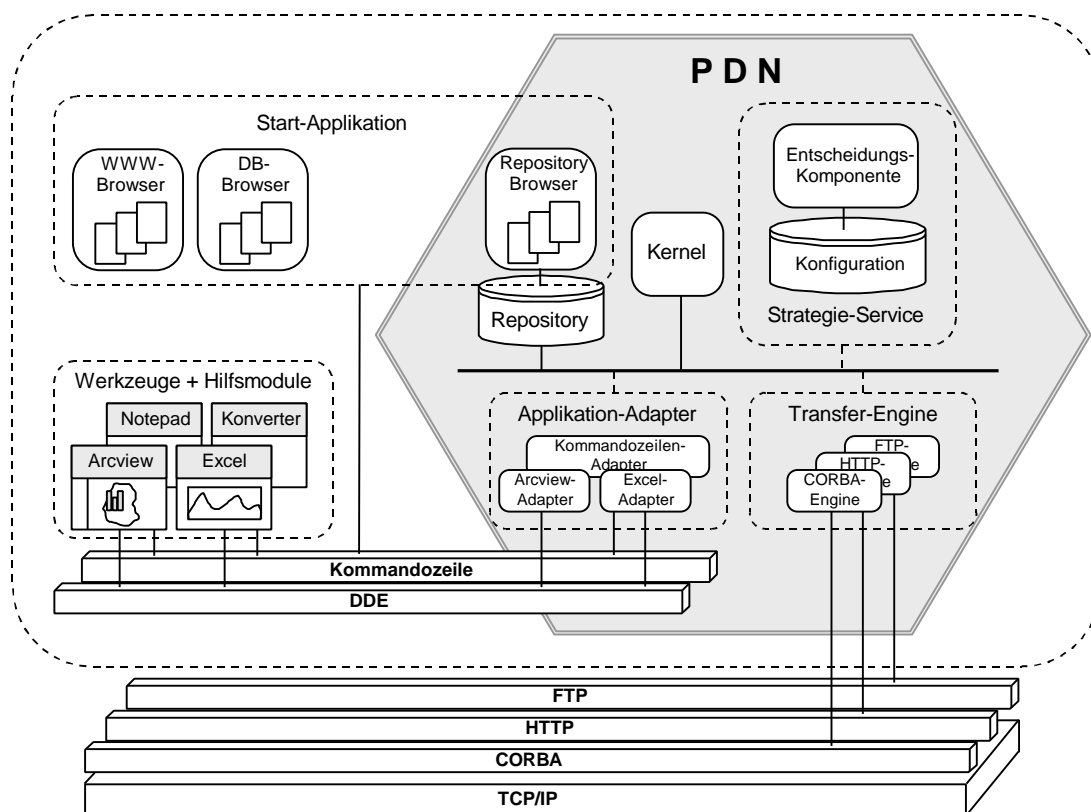


Abb. 1: Architektur des Personal Data Node (PDN)

Die Kommunikation erfolgt über Dynamik Data Exchange (DDE) und Kommandozeilenparameter. Die Einbindung in die jeweilige Gesamtarchitektur erfolgt über Transfer-Module, die die Übertragung von Datenobjekten über FTP, HTTP oder CORBA ermöglichen. Die Komponenten des Klienten lassen sich zu drei Bereichen zusammenfassen:

- Start-Applikationen können nahezu beliebige Informations-Kataloge sein, über die der Anwender Zugriff auf Informationen erhält. Aufgabe eines Informations-Katalogs ist das Bereitstellen der korrekten und vollständigen Metadaten für die Entscheidungsprozesse des Strategie-Service.

- Hilfsmodule und Werkzeuge sind die auf dem Klienten verfügbaren Dienste zur Verarbeitung und Aufbereitung von Informationen. Hilfsmodule sind dabei beispielsweise Konverter zur Umwandlung von Datentypen.
- Der PDN als Kernstück der Architektur integriert die unterstützenden Systemdienste. Dazu zählen ein Strategie-Service zur Modellierung der Arbeitsabläufe, der Kernel zur Umsetzung der modellierten Arbeitsabläufe, ein Repository zur persistenten Speicherung von Datenobjekten und Methoden. Des Weiteren stellt er eine Reihe von Transfer-Engine für die Übertragung von Datenobjekten zur Verfügung sowie die Applikation-Adapter als Abstraktionsschicht zwischen Kernel und lokalen Diensten.

Die zentrale Komponente des PDN ist der Strategie-Service. Er besteht aus der Konfigurationsdatenbank und der Entscheidungs-Komponente. In der Konfigurationsdatenbank werden die Informationen über die lokal vorhandenen Dienste verwaltet. Aus diesen Informationen sowie aus den Metadaten, die der PDN von der Start-Applikation erhält, modelliert der Strategie-Service auf der Grundlage von Entscheidungstabellen den Arbeitsablauf, der in einem Datenbegleitschein festgelegt wird. Ein Arbeitsablauf besteht im allgemeinen aus den Schritten Übertragung, Dekomprimierung, Konvertierung und Aufbereitung. Während der Phase der Modellierung kann der Dienstenutzer Einfluß auf die Festlegung des Dienstes zur Informationsaufbereitung nehmen. Die übrigen Schritte, insbesondere eventuell notwendige Konvertierungen, sind für ihn vollständig transparent.

Der PDN-Kernel stellt die Workflow-Komponente des Klienten dar. Wie in /2/ definiert, versteht man unter „Workflow“ die Automatisierung von Geschäftsvorgängen, in ihrer Gesamtheit oder als Teilabschnitte, während denen Dokumente, Informationen oder Prozesse entsprechend prozeduraler Regeln von einem Beteiligten an den nächsten zur Be- bzw. Verarbeitung weitergegeben werden. Dementsprechend agiert der Kernel auf Basis des im Datenbegleitschein modellierten Arbeitsablaufs und führt die darin festgelegten Arbeitsschritte in sequentieller Folge aus. Nebenläufigkeiten in der Abarbeitung bleiben bisher unberücksichtigt.

Diese Version des PDN setzt bereits einige Aspekte des logischen Klienten um, die im einleitenden Abschnitt angesprochen wurden. Dazu gehören beispielsweise die Trennung in den Strategie-Service und den PDN-Kernel und die Beschreibung der Dienste in Form von Metadaten. Darauf aufbauend wurden vor allem für die UIS-Anwendungen UFIS-II (siehe Beitrag des FAW zu diesem Bericht) und KFÜ-Baden-Württemberg /12/ Anpassungen vorgenommen.

2.1 Nutzung des PDN im Rahmen von UFIS-II

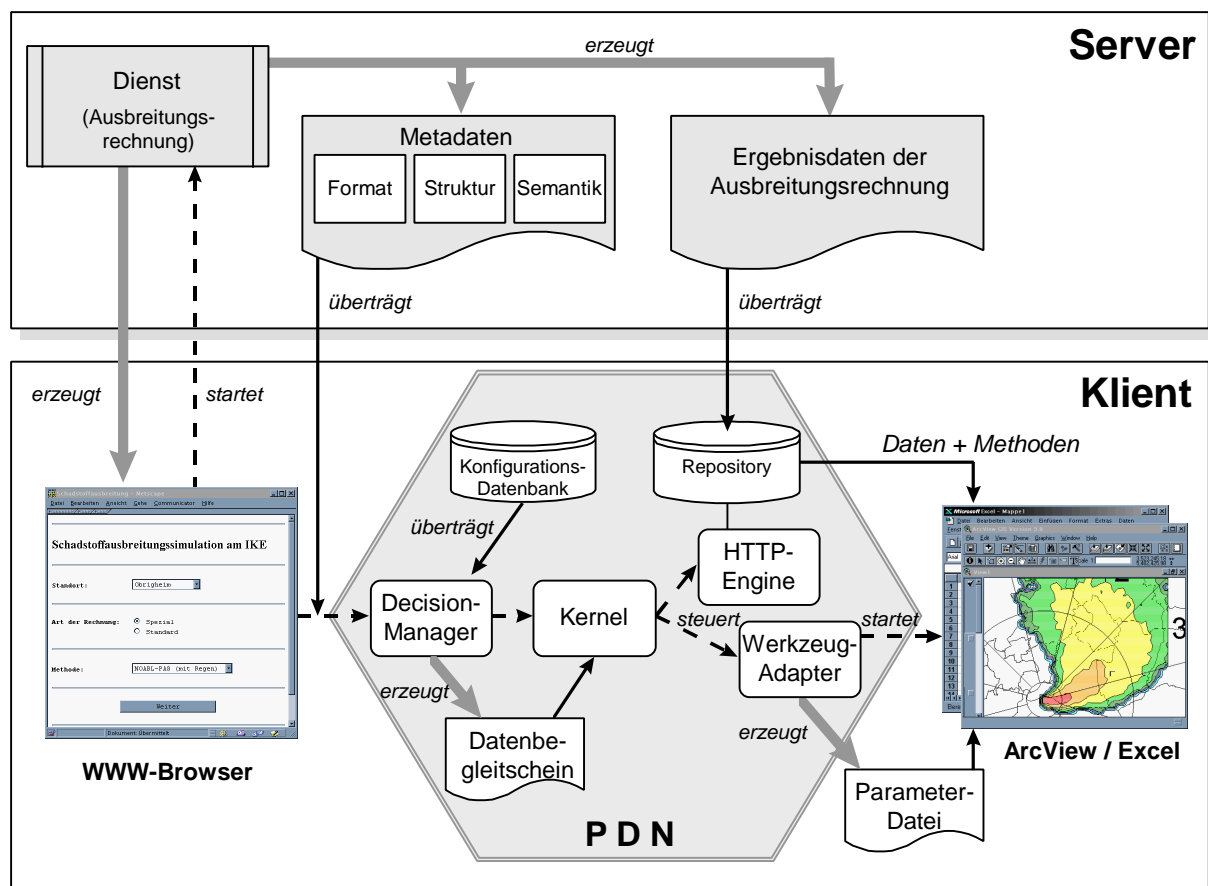
Die Architektur von UFIS-II sieht einen Informations-Katalog auf Basis eines Web-Browsers vor, über den der Anwender die benötigten Informationen (z.B. Schadstoffaufkommen eines bestimmten Stoffes zu einem bestimmten Zeitpunkt in einem bestimmten Gebiet) auswählen kann. Ein Datenbank-Zugriffsdienst formuliert die Abfrage an den Datenbank-Server und stellt die Ergebnisdaten als Datenobjekt auf dem Server bereit. Als unmittelbares Ergebnis liefert er dem Klienten Informationen über das Datenobjekt in Form von Metadaten. Anhand dieser Informationen bietet der PDN dem Anwender eine Auswahl möglicher Darstellungsarten (Methoden) auf Basis der lokal verfügbaren Aufbereitungs-Dienste. Nach der interaktiven Festlegung des Arbeitsablaufs wird das Datenobjekt auf den Klienten transferiert, entpackt und für die Erfordernisse des jeweiligen Werkzeugs über Hilfsmodule aufbereitet. Als letzter Schritt

folgt die automatische Generierung der vom Anwender gewählten Darstellung mit Hilfe eines geeigneten Werkzeugs und lokal verfügbarer Methoden. Im Rahmen dieses Prototyps werden neben den im vorigen Abschnitt beschriebenen unterstützenden Systemdienste folgende Dienste eingesetzt:

Als *Dienst zur Aufnahme von Informationen* wird das relationale Datenbanksystem Oracle verwendet. Der Zugriff auf die Datenbank erfolgt über einen Auswahlkatalog auf Basis des WWW-Browsers Netscape.

Dienste zur Verarbeitung von Informationen sind ein Konvertierungsmodul zur Umwandlung der Daten in ein für Microsoft Excel verwendbares Format sowie Module zur Komprimierung und Dekomprimierung der Datenobjekte.

Die *Dienste zur Aufbereitung* gestatten die Darstellung mit folgenden Applikationen: Tabellen und Geschäftsgraphiken mit Microsoft Excel, einfache Textdarstellung mit dem Microsoft Notepad sowie Darstellungen mit geographischem Raumbezug mit dem GIS-Viewer ArcView von ESRI. Eine nähere Beschreibung der Aufbereitungsdienste findet sich in /4/. Die Einbettung des PDN in UFIS-II zeigt Abb. 2.



Dipl.-Ing. R. Kopetzky, Universität Stuttgart, IKE 1997

Abb. 2: Einbettung des PDN in UFIS-II

2.2 Nutzung des PDN im Rahmen des KFÜ

Das Kernreaktor-Fernüberwachungssystem Baden-Württemberg ist Bestandteil des UFIS. Insbesondere der Dienst CRAYSIM /13/ wurde auch im Hinblick auf seine Nutzung im KFÜ entwickelt. Innerhalb des KFÜ steht daher ein Server zur Verfügung, auf dem CRAYSIM Rechnungen für Notfallschutzübungen durchgeführt werden können. Der PDN dient dabei als Klient, mit dem CRAYSIM als Dienst in Anspruch genommen werden kann und mit dem Ergebnisse von CRAYSIM-Rechnungen visualisiert und interpretiert werden können. Im Rahmen des GLOBUS-Projektes wurden für diese Anwendung spezielle Methoden und Skripte entwickelt, mit deren Hilfe es möglich ist, Isolinien darstellungen aus Werten an Punkten mit beliebigen Koordinaten zu erstellen und sie in ArcView in den im KFÜ gewohnten Sektorendarstellungen wiederzugeben.

In Abb. 3 ist ein Beispiel solch einer Sektorendarstellung gezeigt. Die Methoden werden auch Bestandteil des teilerneuterten KFÜ's /12/ werden.

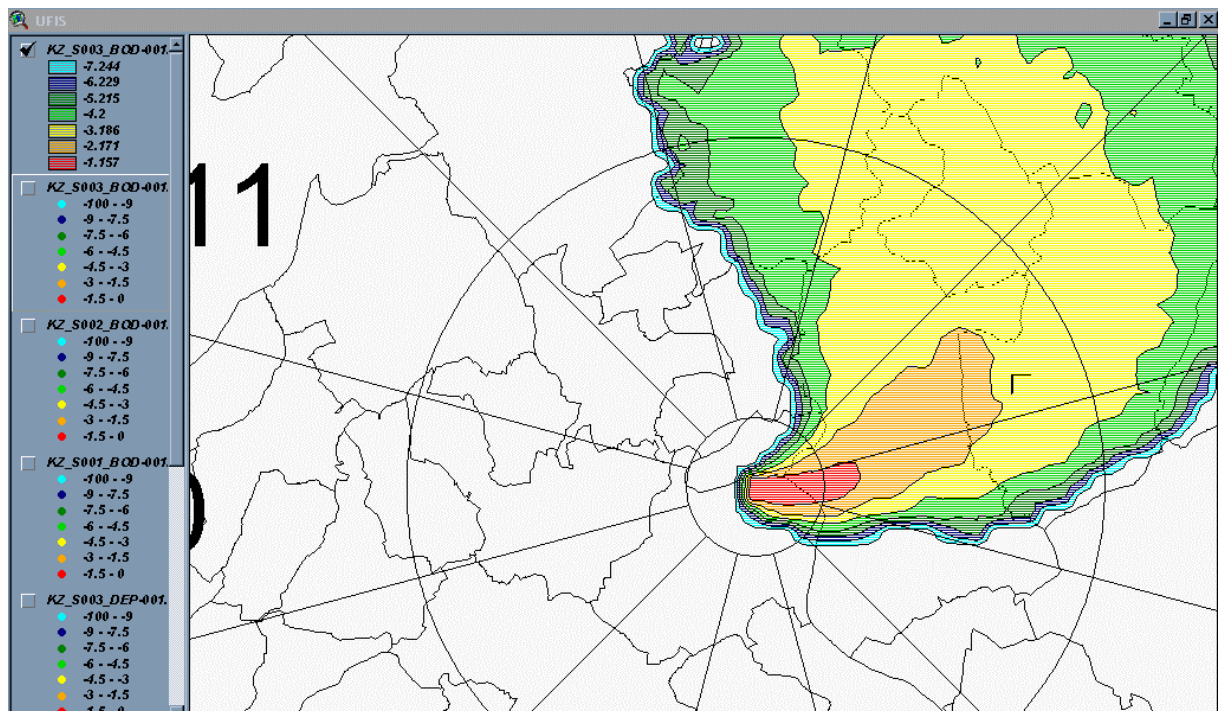


Abb. 3: Sektorendarstellung der bodennahen Konzentration

2.3 Weiterentwicklung der logischen Architektur

Als Ergebnisse aus den Arbeiten im Rahmen von GLOBUS IV und aus weiteren Projekten in den Bereichen „Umweltmanagement“ und „Umweltinformationssysteme“ lassen sich folgende Punkte feststellen: Das Grundkonzept hat sich bewährt und erlaubt es, Fragestellungen der beschriebenen Komplexität zu bearbeiten. Insbesondere der modulare Aufbau gestattet die schnelle Anpassung des Frameworks an die Erfordernisse des jeweiligen Informationssystems. Auch hat sich die Meta-Data Description Language (MDDL) /5/, die im Rahmen der PDN-

Entwicklung auf Basis der Standard Generalized Markup Language (SGML) für die Beschreibung der Metadaten definiert wurde, als ausreichend erwiesen.

Allerdings ergaben sich auch eine Reihe von Bereichen, in denen verbesserte Lösungen notwendig sind. Bereits bei der Umsetzung von Systemen mittlerer Komplexität ist abzusehen, daß die Verknüpfungslogik über Entscheidungstabellen für komplexe Problemstellungen, wie sie ein offenes, dynamisches System mit sich bringt, nicht ausreichend ist. Ein weiterer Punkt ist die bisherige konzeptionelle Trennung der Applikationen in Start-Applikation, Werkzeuge und Hilfsmodule. Diese Trennung führt zu zwei unterschiedlichen Schnittstellen und verhindert beispielsweise, Werkzeuge als Start-Applikationen zu verwenden. Dies kann aber erforderlich sein, wenn zu einer bestimmten Darstellung ergänzende Informationen notwendig werden. Des Weiteren erlaubt die derzeitige Realisierung des PDN-Kernel nur sequentielle Arbeitsabläufe. Insbesondere bei der Informationssammlung aus mehreren Datenquellen ist eine parallele Abarbeitung wünschenswert. Schließlich führen die Überlegungen, in zukünftigen Systemen auch „Thin Clients“ einzusetzen; zu der Notwendigkeit einer Trennung von physikalischem und logischen Klienten. Daraus folgt, daß DDE als Middleware gegen eine plattformunabhängige Architektur (z.B. CORBA) ausgetauscht werden sollte.

Abb. 4 zeigt die daraus abgeleitete Architektur des logischen Klienten, wobei die einzelnen Komponenten entsprechend der CORBA-Spezifikation den Bereichen

- Common Facilities (allgemein nützliche, höherwertige Klassen und Objekte),
- Application Objects (Dienste) und
- Object Services (grundlegende Operationen für die logische Modellierung)

zugerechnet werden.

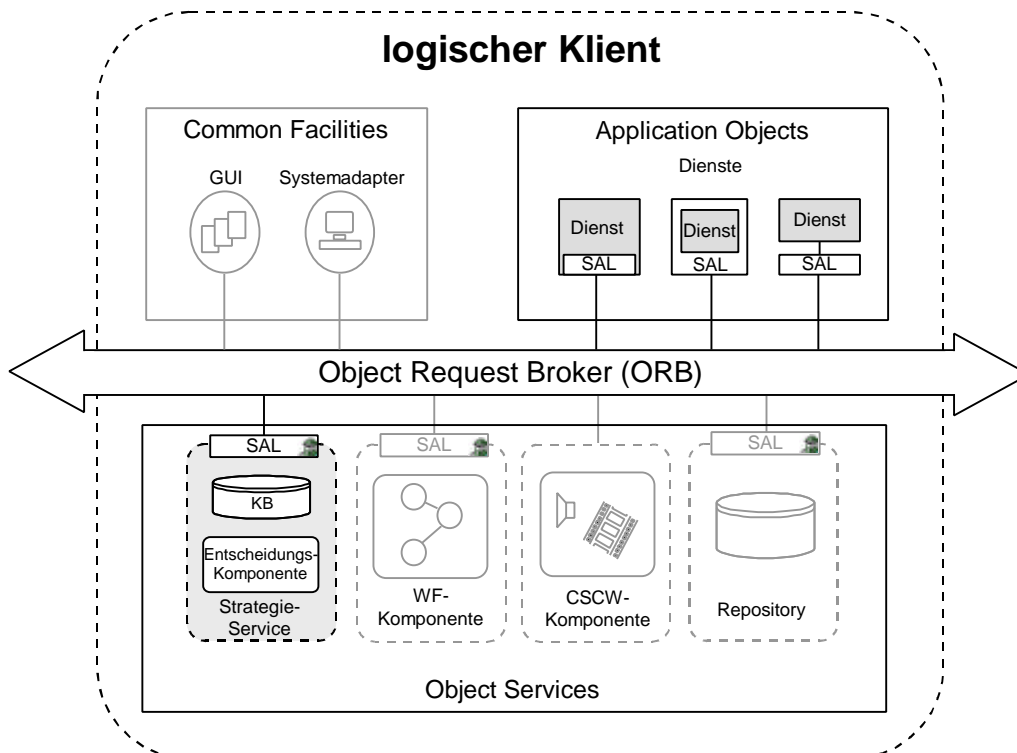


Abb. 4: Konzeption der Architektur des logischen Klienten

In diesem Bild sieht man neben den Diensten zur Aufnahme, Verarbeitung und Aufbereitung von Informationen eine Reihe unterstützender Systemdienste.

- *CSCW-Komponente*: Diese Komponente beinhaltet Computer Supported Cooperative Work (CSCW) - Techniken zur Unterstützung der Kommunikation und Kooperation zwischen Dienstenutzern und Experten. Sie erlaubt schnelle multimediale Rückfragen, das gemeinsame Bearbeiten von Vorlagen und Protokollen und die partnerschaftliche Durchführung komplexer Aufgaben einschließlich der Diskussion von Ergebnissen.
- *Strategie-Service*: Der Strategie-Service ist die zentrale Komponente der Arbeitsumgebung. Er besitzt eine Konfigurationsdatenbank, in der die Beschreibungen der lokal verfügbaren Dienste abgelegt sind. Der Klient wird dadurch entsprechend den Anforderungen seines Nutzers konfigurierbar und kann daher optimal angepaßt werden. Die Anpassung erfolgt über ein Nutzerprofil, das ebenfalls in der Konfigurationsdatenbank abgelegt ist. Auf der Grundlage eines regelbasierten Systems modelliert der Strategie-Service den Arbeitsablauf für den jeweils auszuführenden Auftrag. Dabei wird neben den Daten aus der Konfigurationsdatenbank auch der aktuelle Systemzustand berücksichtigt, zu dem der Strategie-Service über Systemadapter Zugang hat. Der Definition in /1/ folgend, bezeichnen wir ein System, das als Teil einer virtuellen Umgebung Informationen aus dieser Umgebung aufnimmt und diese eigenständig und zielgerichtet beeinflusst, als Agenten. Demzufolge ist der Strategie-Service ein Agent zur Ausführung von Benutzeraufträgen beliebiger Komplexität. Weitere Eigenschaften von Agenten können z.B. Mobilität, Lernfähigkeit oder Glaubwürdigkeit sein.
- *WF-Komponente*: Die Aufgabe der Workflow (WF) - Komponente ist die Ausführung des von dem Strategie-Service modellierten Arbeitsablaufs. Ein zentrales Merkmal und zugleich großer Kritikpunkt traditioneller Workflow-Anwendungen ist die zentrale, ex-ante definierte und damit unflexible Steuerung von Vorgängen. In einem heterogenen, verteilten System mit einer großen Anzahl von Diensten und hohem Datenaufkommen wird ein zentraler Workflow-Server zum „Flaschenhals“ des Systems, da er über alle Abläufe Bescheid wissen muß. Das Ziel im vorgestellten Szenario ist deshalb ein System verteilter, kooperierender WF-Komponenten.
- *Verteiltes Repository*: Zur Beherrschung der Probleme, die aufgrund der Verteilung in heterogenen und verteilten Systemen bei der Bereitstellung von Informationen auftreten, ist das Modell eines gemeinsamen virtuellen Datenspeichers sinnvoll. In einem realen System wird durch ein Repository den Anwendungen ein Behälter für die Ablage der anfallenden Daten bereitgestellt. In diesem Sinne ist das Repository ein passiver Basismechanismus, der einen plattformunabhängigen, strukturierten Zugriff auf die im Repository enthaltenen Daten realisiert. Dabei ist die physikalische Lokation der Daten im System transparent.

2.4 Die Strategiekomponente

Mit dem Strategie-Service soll ein Agent /1/ konzipiert und umgesetzt werden, der in der Lage ist, unter Berücksichtigung von verfügbaren Diensten, Systemzustand und Nutzerprofil einen Arbeitsablauf für das Zusammenwirken kooperierender Dienste festzulegen.

Für die Konzeption des Strategie-Service sind drei Teilbereiche von Bedeutung:

- *Schnittstelle* für Kommunikation und Inanspruchnahme (Service Agent Layer - SAL)
- *Entscheidungskomponente* für die Modellierung des Arbeitsablaufs

- *Konfigurationsdatenbank (KB)* zur Aufnahme der Fakten und Regeln

Die Konzepte für diese Teilbereiche sollen im Folgenden näher erläutert werden:

Die Schnittstelle

Die offene und flexible Systemarchitektur des Informationssystems sowie die Absicht, an geeigneten Stellen auf etablierte Standards aufzusetzen, war Ausgangspunkt für die Idee, die Knowledge Query and Manipulation Language (KQML) als Grundlage für die Kommunikation aller Komponenten des verteilten, heterogenen Systems zu verwenden. Bei KQML handelt es sich um die Spezifikation eines Nachrichtenformats und eines Protokolls für den Wissensaustausch zwischen kooperierenden Agenten. Die Agenten kommunizieren über einen standardisierten Satz von Befehlen, der unter anderem aus grundlegenden Befehlen wie „ask-all“ und informierenden Befehlen wie „tell“ besteht /9/. KQML ist nicht an eine bestimmte Sprache gebunden, sondern kann als einheitlicher Rahmen für Nachrichten und Daten in unterschiedlichen Beschreibungssprachen verwendet werden.

Für die Definition der Schnittstellen-Syntax wird die CORBA-IDL verwendet. Dabei besteht die Schnittstelle aus einem für alle Komponenten des Systems standardisierten Teil für die Kommunikation und Kooperation und einem Komponenten-spezifischen Teil für die Inanspruchnahme des eigentlichen Dienstes.

Die Entscheidungskomponente

Die Modellierung von Arbeitsabläufen in dynamischen, offenen Systemen stellt ein komplexes Problem dar, für das klassische Programmieransätze wie die prozedurale Programmierung nicht geeignet erscheinen. Insbesondere die Notwendigkeit, für die Einbeziehung der Nutzerbedürfnisse qualitative und einschätzende Informationen zu berücksichtigen, legt den Einsatz eines regelbasierten Systems für die Entscheidungskomponente des Strategie-Service nahe. /10/

Die Entscheidungskomponente soll sowohl in der Lage sein, die für Informationen geeigneten Aufbereitungen mit deren Zwischenschritten zu identifizieren (Forward-Chaining) als auch die für eine bestimmte Aufbereitung notwendigen Dienste festzulegen (Backward-Chaining). Grundlage für die Modellierung dieser Arbeitsabläufe sind zum einen Fakten und Regeln, die in der Konfigurationsdatenbank gespeichert sind, als auch dynamische Informationen, die den aktuellen Systemzustand widerspiegeln. Der Modellierungsansatz soll sowohl kausal voneinander abhängige als auch nebenläufige Arbeitsschritte unterstützen. Diesen Anforderungen wird der Petri-Netz-Ansatz gerecht, da dieser die Beschreibung von ereignisorientierten, verteilten Systemen ermöglicht und dabei auch nebenläufige Arbeitsschritte zulässt. /11/

Die Konfigurationsdatenbank

Die Konfigurationsdatenbank dient der Speicherung der Fakten und Regeln, auf deren Basis die Entscheidungskomponente die Arbeitsabläufe modelliert. Fakten sind statische oder dynamische Informationen über den Systemzustand, das persönliche Nutzerprofil des jeweiligen Dienstenutzers sowie die Beschreibung der im Arbeitsbereich verfügbaren Dienste. Diese erfolgt durch die Beschreibung der Ein- und Ausgaberräume der Dienste in Form von Metadaten. Für die Beschreibung der Metadaten wird die Meta-Data Description Language (MDDL), eine Beschreibungssprache auf Basis der Standard Generalized Markup Language (SGML), ver-

wendet /4/. Die MDDL stellt grundlegende Konstrukte wie Elemente und Attribute für die Beschreibung hierarchisch strukturierter Metadaten zur Verfügung. Das Konzept sieht eine Grobstrukturierung in drei Ebenen vor: Die syntaktische, die semantische und die technische Ebene. Regeln speichern das Wissen über Bearbeitungsabläufe und Abhängigkeiten und können dynamisch erweitert werden.

Es ist zu überprüfen, ob die oben beschriebenen Lösungsansätze geeignet sind, die komplexen Anforderungen an einen logischen Klienten zu bewältigen. Insbesondere soll das Konzept weiter verfeinert und in das Konzept für ein Gesamtsystem integriert werden. Mit der praktischen Umsetzung in dem Bereich „Umweltinformationssysteme“ soll die Flexibilität und Skalierbarkeit des Ansatzes untersucht werden.

2.4 Die Repository Komponente

Das Repository ist ein Behälter für die Ablage der während der Anwendung eines Dienstes anfallenden Daten. In diesem Sinne ist das Repository ein passiver Basismechanismus, der einen plattformunabhängigen, strukturierten Zugriff auf die im Repository enthaltenen Daten realisiert.

Um die Wiederverwendbarkeit von Repositories in den unterschiedlichsten Umgebungen zu ermöglichen, dürfen keine anwendungsbezogene oder systemspezifische Spezialfälle in einer Repository-Variante berücksichtigt werden. Derartige Anforderungen müssen durch externe Erweiterungen (z.B. Agenten- oder Workflow-Systeme) befriedigt werden.

Wegen der zu erwartenden großen Anzahl von Diensten, die auf unterschiedlichen Rechner-Architekturen und Betriebssystemrevisionen verfügbar sind, ist ein einfaches Konzept für das Repository, basierend auf wenigen Basismethoden, erforderlich. Die wesentlichen Operationen auf einem Repository sind das Ablegen, Suchen, Holen und Löschen von Daten. Um anwendungsgerecht zu arbeiten, ist die Einbeziehung folgender externer Mechanismen (möglichst in Form von Diensten) notwendig:

- Session-Verwaltung
- Benutzerverwaltung
- Datenverwaltung
- Zeitverwaltung

Repositories arbeiten auf der Grundlage von Datenobjekten. Dies sind Daten (z.B. in Dateien) mit einer beliebigen internen Struktur sowie diesen Daten zugeordnete Attribute. Die Attribute enthalten alle benötigten Verwaltungsinformationen.

Alle Operationen eines Repositories müssen garantieren, daß der interne Zustand der Daten eines Datenobjekts nicht verändert wird. Anders verhält sich die Beziehung des Repositories zu dem Attribut-Teil des Datenobjekts. Diese repository-interne Datenstruktur wird nur durch Repositories angelegt, geändert und zusammen mit dem Datenteil des Datenobjekts wieder gelöscht. Zusammenstellungen bestimmter Attribute eines Datenobjekts können durch den Nutzer (Anwender oder Programm) vom Repository erfragt werden.

Weitere Eigenschaften des Repositories sind:

1. **Passivität**

Das Repository führt Operationen nur auf Anforderung durch. Es ist somit nicht in der Lage, selbständig Operationen zu veranlassen.

2. **Information Hiding**

Operationen auf Datenobjekten werden nur nach Authentifizierung (durch passende Session-Kennung oder Benutzername) durchgeführt.

3. **Identifikation, Kennung**

Alle Datenobjekte in Repositories verfügen über eine bidirektional eindeutige Adresse, ihre Kennung. Der Zugriff auf Daten erfolgt ausschließlich über diese Kennung. Die Abfrage von Attributen kann über diese Kennung oder mittels anderer Methoden (z.B. SQL-Queries) erfolgen.

4. **Unveränderbarkeit**

Daten in einem geschlossenen Datenobjekt können nicht mehr verändert (sondern nur noch kopiert oder gelöscht) werden.

5. **Alterung**

Jedes Datenobjekt verfügt über eine ihm zugeordnete Lebensdauer. Ist diese überschritten, kann das Datenobjekt (z.B. aus Speicherplatzgründen) von der Zeitverwaltung gelöscht werden.

6. **Unabhängigkeit von Implementierungen**

Jede Inkarnation eines Repositories ist ausschließlich für die Verwaltung und Speicherung der ihm unterstellten Datenobjekte zuständig. Es ist nicht zulässig, daß sich zwei Inkarnationen von Repositories einen Speicherbereich teilen.

7. **Beliebiges Speicherformat**

Die Art und Weise, wie ein Repository die ihm unterstellten Datenobjekte speichert, ist nicht festgelegt. Somit sind „sichere“ (durch Verschlüsselung) und „kompakte“ (durch Komprimierung) Repositories realisierbar.

Die Funktionalitäten der Repositories **R** entstehen aus den sinnvollen Kombinationen der Grundoperationen $\mathbf{G} = \{\text{erzeugen, abschließen, einfügen, herausholen, löschen}\}$ mit den Bestandteilen der Datenobjekte $\mathbf{DO} = \{\text{Daten, Attribute}\}$. Zum Auffinden von Datenobjekten ist zusätzlich ein Auskunftsbzw. Abfrage-Mechanismus **Q** (Query) notwendig, welcher Session-**S**, Auftrags- **A** und Nutzer- **U** (User) Attribute zur Bestimmung der Kennung **K** von Datenobjekten verwendet.

Zur Integration von bestehenden Anwendungssystemen darf die bisherige Methodik zur Speicherung von Daten nicht unberücksichtigt bleiben. Bestehende Anwendungen speichern ihre Daten in der Regel in einem Dateisystem **DS**. Der Speicherort im Dateisystem wird durch einen betriebssystemabhängigen **Pfad** spezifiziert. Aus diesem Grund müssen die Grundoperationen *einfügen* und *herausholen* ebenfalls das Dateisystem als Quelle oder Ziel unterstützen. Tab. 1 definiert den Funktionsumfang des Repositories.

<i>Funktion</i>	<i>Beschreibung</i>
R.erzeugen $\rightarrow \mathbf{K}$	Neues, leeres Datenobjekt im Repository R anlegen, liefert Kennung K für das Datenobjekt.
R.einfügen (R'.K) $\rightarrow \mathbf{K}'$	Kopieren eines Datenobjekts mit Kennung R'.K aus dem Repository R' in das Repository R (incl. aller Attribute), liefert Kennung K' der Kopie zurück.
R.herausholen (K, Pfad) $\rightarrow \mathbf{DS}$	Kopieren des Datenobjekts K in das Verzeichnis Pfad im Dateisystem DS
R.löschen (K)	Entfernt ein Datenobjekt K aus dem Repository R
R.erzeugen (K)	nicht verfügbar (entspricht: erzeugen eines Datums ohne Inhalt in einem Datenobjekt)
R.einfügen (K, Pfad)	Kopiert eine Datei mit Pfad aus dem Dateisystem in das Datenobjekt K . Die Originaldatei bleibt erhalten.
R.einfügen (K, K')	Fügt ein Datenobjekt K' in das Datenobjekt K ein, das Repository muß K' nicht unbedingt kopieren.
R.abschließen (K)	Ändern des Zustands eines geöffneten Datenobjekts mit Kennung K zu einem geschlossenen Datenobjekt, damit sind auf K keine weiteren einfüge Operationen zulässig.
R.attribut (K, einfügen)	Einfügen, Ändern und Löschen von beschreibenden Attributen zu einem Datenobjekt. Diese Operationen sind auch auf abgeschlossene Datenobjekte anwendbar.
R.attribut (K, ändern)	
R.attribut (K, löschen)	
Q.K (R, {S, A, U}*) $\rightarrow \{ \mathbf{K} \}^*$	Kennungsanfrage Abfrage-Mechanismus Q liefert anhand einer beliebigen Folge von Session- S Auftrags- A oder Nutzer- U Attributen eine Menge von Kennungen $\{ \mathbf{K} \}^*$ für Datenobjekte in dem Repository R
Q.Attrib (R, { S A U }^*) $\rightarrow \{ S A U \}^*$	Attributanfrage, Abfrage-Mechanismus Q liefert anhand von Attributfolge eine Menge von Ergebnisattributen.

Tab. 1: Funktionsumfang des Repositories

2.6 Die Kooperationskomponente

2.6.1 Das Referenzmodell Kooperation und Kommunikation

Die Inanspruchnahme von Diensten kann automatisch (etwa über Middleware wie CORBA) oder im Kontakt mit den Diensteanbietern geschehen. Dies erfordert neben einem Modell für Kommunikation auch ein Modell für Kooperation. Beide sind Bestandteil unseres Referenzmodells Kooperation und Kommunikation, welches Abb. 5 zeigt.

Das Modell enthält die Komponenten Dienstenutzer und Diensteanbieter, die als Personen beispielsweise über Applikation Sharing Prozesse miteinander kooperieren und jeweils für die ih-

nen zugeordneten Prozesse auf Seiten des Klienten und ihres Servers zuständig sind. Die Prozesse kommunizieren durch Tausch von Daten mit Hilfe der dem System zugeordneten Middleware, die im Rahmen des UIS mittels WWW oder CORBA Techniken realisiert sind.

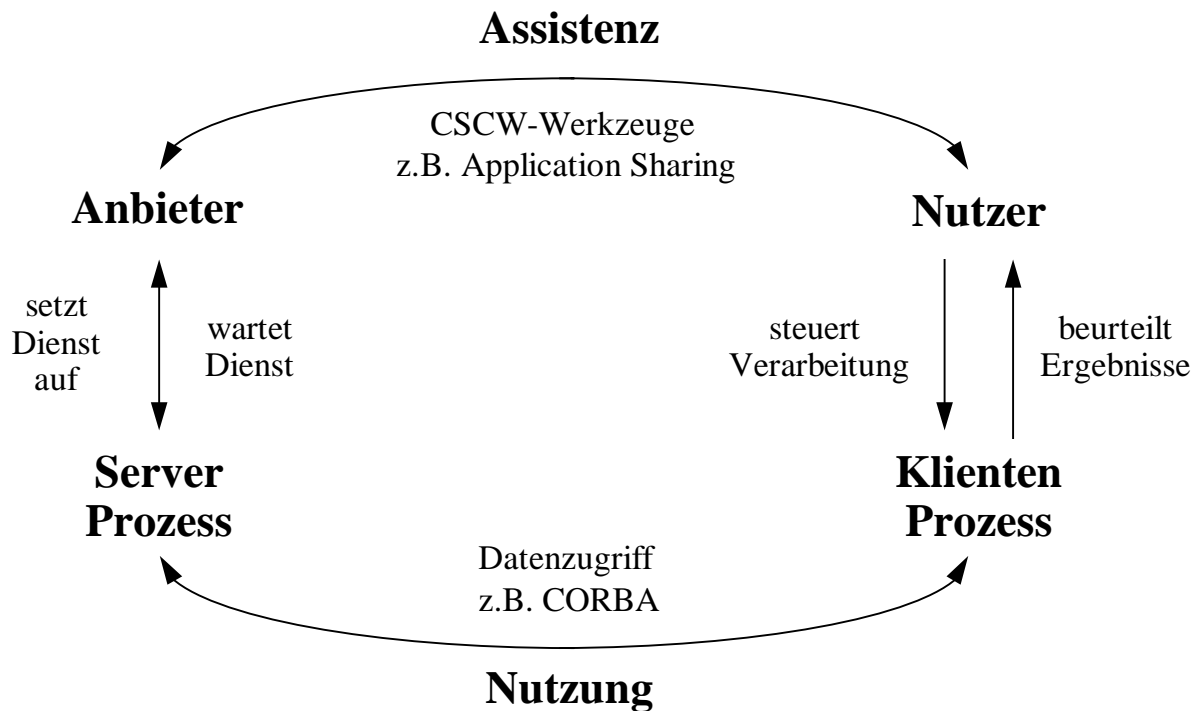


Abb.5: Referenzmodell Kooperation und Kommunikation

Im Laufe des Berichtsjahres wurde gezeigt, daß auch die Kooperation mit Mitteln der Informationstechnik unterstützt werden kann. Dies geschah am Beispiel zweier typischer Anwendungen.

Die eine ist die Erstellung eines multimedialen Querschnittsberichtes. Die zweite ist die Durchführung und Auswertung einer komplexen Ausbreitungsrechnung. Beide Tätigkeiten erfordern:

- Die Einbindung von Informationen aus unterschiedlichen Bereichen in eine Anwendung,
- die Diskussion der Bereiche, in denen die Informationen verknüpft werden können,
- die Verbindung der Informationen (in der Regel über Programme) zur Ableitung neuer Informationen,
- die Diskussion der Ergebnisse mit Hilfe von virtuellen Hilfsmittel,
- die Untersuchung von alternativen Lösungen.

Über Möglichkeiten der Unterstützung durch kooperative Techniken bei der Inanspruchnahme des Dienstes Ausbreitungsrechnung (CRAYSIM) wurde schon im letzten Jahr berichtet. Hier soll deswegen der Schwerpunkt bei der Unterstützung der Erstellung multimedialer Berichte etwa zur Dokumentation solcher Rechnungen liegen.

2.6.2 Soft- und Hardware Anforderungen für den Einsatz von CSCW

Basis unserer Arbeiten zum kooperativen Arbeiten ist das CSCW-System NetMeeting von Microsoft, das im weiteren Verlauf des Projektes eingesetzt werden soll. NetMeeting besitzt alle Funktionalitäten, die benötigt werden, um die Zusammenarbeit lokal getrennter Personen zu unterstützen und somit effektiver zu gestalten. NetMeeting steht im Oktober in der Version 2.0 zur Verfügung und unterstützt die Betriebssysteme Windows 95 und Windows NT 4.0. NetMeeting wird von Microsoft als Freeware angeboten und soll in der Zukunft Teil des Betriebssystems werden. Somit fallen im Rahmen der Software keine zusätzlichen Kosten an.

Für den Betrieb ist es empfehlenswert, als Basis einer CSCW-Station einen Pentium PC mit 166 MHz (64 MB RAM) oder höher einzusetzen. Dies gilt insbesondere dann, wenn Videodaten ausgetauscht werden sollen.

Als zusätzliche Hardware benötigt man eine Soundkarte, ein Mikrofon, externe Lautsprecher, eine Videokarte und eine Videokamera. Wenn möglich, sollte ein Headset eingesetzt werden.

Tab. 2 zeigt die Kosten für die zusätzliche Hardware, die benötigt wird, um einen PC-Arbeitsplatz zu einer CSCW-Station (Audio- und Video) auszubauen.

Video- und Soundkarte	470.- DM
Videokamera mit Mikrofon	400.- DM
Externe Lautsprecher	130.- DM
Headset	100.- DM
Gesamtkosten (inkl. 15% MwSt.)	1100.- DM

Tab. 2: Kosten für die Aufrüstung eines PCs (Audio- und Video), Stand Oktober 1997

2.6.3 Teamorientierte Erstellung hypermedialer Umweltberichte

Das folgende Szenario für die Erstellung eines hypermedialen Umweltberichtes orientiert sich weitgehend an den im Projekt HUDA entwickelten Workflow. Unsere Verallgemeinerung ist in Abb. 6 dargestellt.

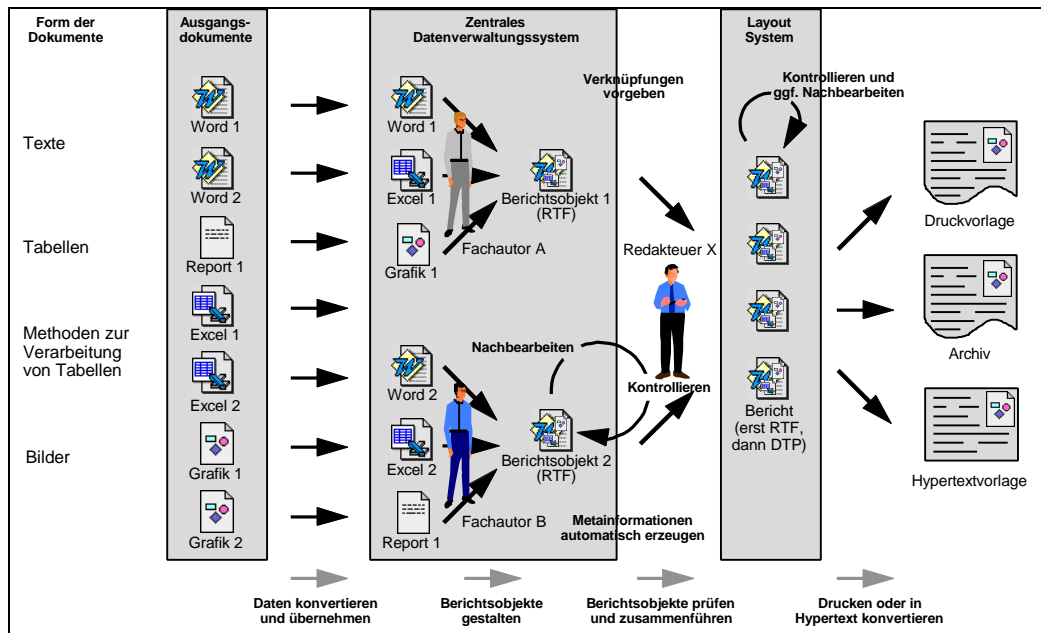


Abb. 6: Allgemeiner Arbeitsablauf zur Dokumentenerstellung /15/

Die Berichtserstellung beginnt mit der Bereitstellung der Ausgangsdaten. Diese sind hier Texte, Tabellen, Methoden zur Verarbeitung von Tabellen sowie Grafiken. Beim Einbringen in das zentrale Datenverwaltungssystem werden diese in definierte Zielformate konvertiert.

In einem nächsten Schritt verteilt das zentrale Datenverwaltungssystem nach seinen Vorgaben die einzelnen Dokumente an die für die Erstellung von Teilberichten zuständigen Autoren, im Folgenden Fachautoren genannt. Diese fügen die entsprechenden Text- oder Grafikelemente sowie ggf. Tabellen zu Berichtobjekten zusammen.

Nach der Fertigstellung platziert das zentrale Datenverwaltungssystem die Berichtobjekte automatisch in die Postfächer der für das Korrekturlesen und Freigeben zuständigen Redakteure. Außerdem kann das Berichtobjekt auch anderen Mitarbeitern zugänglich gemacht werden, um ggf. von ihnen Anregungen oder Kritik zu erhalten.

Der Redakteur entnimmt das Berichtobjekt seinem Postfach und kontrolliert den Inhalt. Sollte eine Korrektur durch den Fachautor notwendig werden, reicht der Redakteur das Dokument zurück. Dabei markiert er im günstigsten Falle die zu ändernden Passagen direkt in dem Dokument. Der Fachautor fertigt nach Erhalt zum Korrigieren der Fehler eine neue Version von dem Dokument an und legt sie anschließend wieder vor.

Sobald alle Teilberichte eines Gesamtberichtes durch die Redaktion freigegeben sind, kann ein Redakteur mit dem Zusammenfügen beginnen. Die Berichtobjekte werden dazu sequentiell geordnet und in einem DTP-System miteinander verknüpft. Schon in diesem Arbeitsschritt lassen sich die für den Export von Hypertext notwendigen Hyperlinks in den Text einfügen.

Die Druckvorlage sowie das Archivdokument entstehen direkt durch die Ausgabe aus dem DTP-System. Der Hypertext wird durch Konvertierung aus der Druckvorlage erzeugt. Es ist allerdings auch denkbar, den Hypertext ohne Umwege über die Druckvorlage direkt aus der vorgegebenen Reihenfolge der Berichtobjekte auszugeben.

CSCW-Techniken können diesen Vorgang in vielerlei Weise unterstützen. Offensichtlich ist dies vor allem bei der Kooperation von Fachautoren untereinander und bei der Abstimmung der Beiträge in Zusammenarbeit zwischen Redakteur und Fachautoren. Insbesondere wenn es dabei um die Klärung fachlicher Fragen geht, ist es vorteilhaft, gemeinsam örtlich getrennt an einem Berichtsbjekt arbeiten zu können und dabei Beziehungen zu anderen Berichtsteilen mit in Betracht zu ziehen. In /15/ wurden dazu Überlegungen zur Organisationsstruktur und den aus dieser Struktur abgeleiteten Kommunikationskanälen aufgezeigt.

2.6.4 Dokumentation einer Ausbreitungsrechnung

Ausbreitungsrechnungen werden unter anderem im Rahmen von Notfallschutzübungen eingesetzt. Ein mögliches Szenario könnte dabei wie folgt aussehen:

Im Rahmen der Übung wird der Dienst Ausbreitungsrechnung in Anspruch genommen. Solch eine Dienstinanspruchnahme gliedert sich in drei Schritte: Eingabeerstellung, Durchführung der Berechnung sowie Auswertung und Dokumentation der Ergebnisse.

Die Eingabe der Ausgangsdaten einer Ausbreitungsrechnung erfolgt über HTML-Seiten. Nachdem alle Daten eingegeben und die Art der Rechnung ausgewählt wurde, kann der Dienst gestartet werden. Die Ergebnisse der Rechnung (Tabellen und Matrizen) werden in einem Repository abgelegt und verwaltet. Mit entsprechenden Visualisierungs-Tools werden anschließend Ergebnisbilder zur besseren Interpretation erstellt. Mittels einem vorgefertigten Berichtsgerüst und den Ergebnissen der Rechnung kann nun der Bericht unter Einsatz eines DTP-Systems erstellt werden.

Bei der Auswahl geeigneter Rechenszenarien und der zugehörigen Eingabedaten sowie bei der Interpretation der Ergebnisse und der Ableitung von Entscheidungen, ist es jedoch häufig erforderlich, Fachberater mit einzubeziehen, um mit diesem gemeinsam die benötigte Interpretation und Abschätzung der Folgen mit in den Bericht aufzunehmen. Steht der Fachberater nicht am Arbeitsplatz zur Verfügung, so läßt sich mittels CSCW-Techniken (z.B. NetMeeting) eine Konferenz aufbauen. Unter Einsatz von Audio und Application Sharing kann der Fachberater den Bericht entfernt einsehen, darüber diskutieren und seine Interpretation direkt in das DTP-System online eingeben.

Dieses Szenario wurde in Form einer Vorführung im Rahmen einer GLOBUS-Sitzung am 26.09.1997 am IKE experimentiert und ist in ähnlicher Weise Bestandteil der Abschlußdemo.

Entsprechend dem Vorgehen bei der Nutzung des Dienstes Ausbreitungsrechnung kann die Dokumentation der Rechnung in einem Bericht erfolgen, der sich aus drei Kapiteln zusammensetzt.

Kapitel 1 besteht aus reinen statischen Textteilen mit allgemeinen Beschreibungen zur Ausbreitungsrechnung, z.B. die Ziele der Ausbreitungsrechnung oder kurze Erläuterungen zu den einzelnen im Rahmen der Berechnung eingesetzten Modulen. Dieser Berichtsteil ist allgemein gehalten und kann vorab erzeugt werden. Er ist Teil des Berichtsgerüsts.

Kapitel 2 besteht aus sogenannten statischen Textteilen und dynamischen Elementen. Letztere sind sowohl Eingabedaten (Tabelle) für die Berechnung als auch Berechnungsergebnisse in

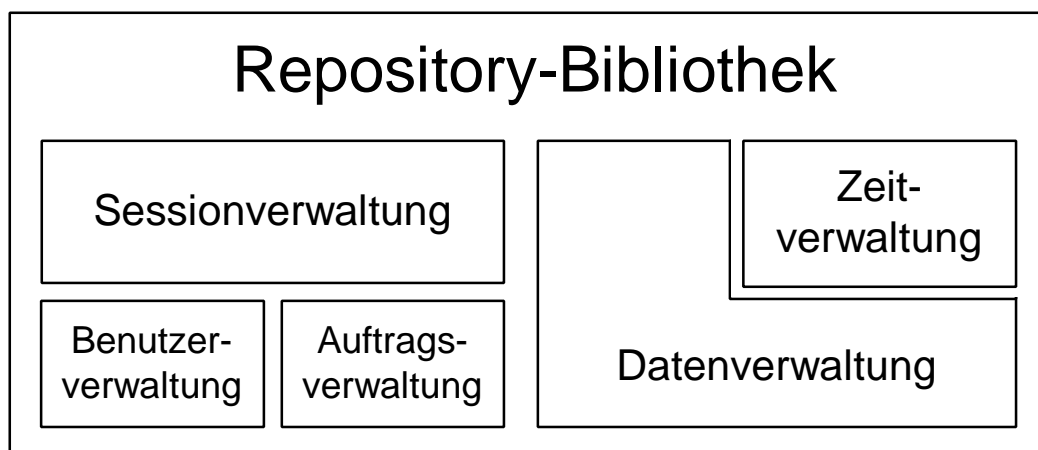
Form von Grafiken (ArcView, Excel). Diese dynamischen Elemente werden zum Zeitpunkt der Erstellung des Berichtes über sogenannte Links aus dem Repository des Dienstes geladen, in das sie kurz zuvor im Rahmen der Berechnung und der Visualisierung abgelegt wurden. Dadurch wird es möglich, die dynamischen Teile dadurch zu aktualisieren, daß lediglich die Identifikationsnummer einer Berechnung eingegeben wird. Auf Basis der Informationen im Repository können jetzt die entsprechenden Werte und Graphiken in den Bericht mit eingebunden werden.

Das letzte Kapitel des Berichts dient zur Interpretation der Ergebnisse. Es besteht zum Zeitpunkt vor der Berichtserstellung nur aus einer Überschrift, da der Interpretationstext erst im Rahmen der Berichtserstellung vom Fachberater erzeugt wird.

Wie schon anfangs beschrieben wurde, wird der Fachberater nach Erstellung des 2. Kapitels mittels NetMeeting in den Berichtserstellungsprozeß mit einbezogen. Hierzu wird zuerst Kapitel 2 mit dem Fachberater geshart, damit dieser die Ergebnisse einsehen kann. Danach wird zusätzlich Kapitel 3 geshart und der Fachberater kann seine Bewertung selbst eingeben, als lief das DTP-System auf seinem eigenen Rechner.

2.7 Die Workflow-Komponente

Die Darlegungen im vorigen Abschnitt lassen schon ahnen, warum es sinnvoll sein könnte, kompliziertere Abläufe über einen Workflow-Service absteuern zu lassen. Während es die Aufgabe der Strategiekomponente ist, unter Berücksichtigung von verfügbaren Diensten, Systemzustand und Nutzerprofil einen Arbeitsablauf für das Zusammenwirken kooperierender Dienste festzulegen, kommt es der Workflow-Komponente zu, Unterstützung bei der Formulierung vorgegebener Arbeitsabläufe zu geben (Buildtime) und deren sach- und termingerechte Durchführung zu überwachen (Run-time). Die Rolle, die Workflow-Managementsysteme spielen können, ist in Abb. 7 gezeigt.



Abschnitt I Bibliothek 1_0.ppt (11.02.97)

Abb. 7: Stellung von Workflowkomponenten im Rahmen einer dienstorientierten Arbeitsumgebung

Wir unterscheiden dazu 4 Kontrollebenen. Auf der untersten Ebene sind die Dienste angesiedelt. Dazu gehören etwa der Dienst CRAYSIM oder der Dienst RIMPUFF, mit dem Schadstoffausbreitungen nach dem deutsch-französischen Modell berechnet werden können.

Auf der 2. Ebene sind die lokal verfügbaren Anwendungen angesiedelt. Dazu gehören etwa die Klienten von UFIS oder dem KFÜ aber auch der PDN oder ArcView, wenn es als eigene Anwendung gefahren wird. Zusätzlich ist auch dieser Ebene die Applikation Arbeitsliste angesiedelt. Diese Arbeitsliste ist einem bestimmten Anwender zugeordnet und enthält die von ihm durchzuführenden Aufgaben. Durch die Arbeitsliste kommunizieren Anwender und Workflowkomponente.

Dies geschieht im wesentlichen auf der 3. Ebene, der lokale Steuerungs- und Durchführungsfunktionen zugeordnet sind. Diese können entweder durch Eingriffe des Anwenders oder aber durch Workflow Management Engine (lokaler Repräsentant des Workflow-Management-Systems - WF-Komponente) vorgenommen werden. Die Workflow Management Engine hat dabei die Aufgabe, die Arbeitsliste zu verwalten, d.h. neue Aufträge einzutragen und die Erledigung existierender Aufträge zu vermerken und die daraus folgenden Aktionen (z.B. Ablage eines Dokumentes im zentralen Repository) zu initiieren.

Auf der obersten Ebene befindet sich das Workflow-Management-System, das für die globale Steuerung von Arbeitsabläufen zuständig ist. Das Auffinden und die Verarbeitung von Umweltinformationen, wie sie mit dem UIS durchgeführt werden können, ist ein solcher Ablauf, der durch das Workflow-Management-System mit anderen Abläufen (z.B. aus den Bereichen Bürokommunikation oder Desktop Publishing) verbunden werden kann. Die in Abb. 7 gezeigte Architektur ist damit gleichzeitig ein Vorschlag dafür, wie die in der Fortschreibung der Rahmenkonzeption des Umweltinformationssystems des Landes Baden-Württemberg ausgesprochene Empfehlung der Integration von Daten und Diensten des UIS in die Bürokommunikation umgesetzt werden kann.

3. Sicherheitskonzepte für das UIS

In der Rahmenkonzeption für das UIS werden die Sicherheitsaspekte ausführlich behandelt. Dabei wird der Gesamtkomplex „Sicherheit“ in drei Themenschwerpunkte aufgeteilt:

- Sicherheit bei der Übertragung von Daten und Informationen,
- Sicherheit beim Zugriff auf Daten und Systeme,
- Sicherheit bei der Qualität der Daten.

Die Sicherstellung der Qualität von Daten im Rahmen einer speziellen Verwendung ist normalerweise nicht Bestandteil eines Sicherheitskonzeptes. Das UIS ist aber soweit entwickelt, daß gerade diesem Punkt besondere Bedeutung zukommt. Im Rahmen des Projektes GLOBUS wird dem durch die Entwicklung des Metainformationskonzeptes Rechnung getragen. Die Arbeiten an der Strategiekomponente und an der Workflowkomponente tragen ebenfalls dazu bei, die Mächtigkeit des UIS-Metainformationskonzeptes zu erweitern.

Die Sicherheit bei der Übertragung von Daten und Informationen wird dadurch erreicht, daß als Basis dieser Übertragungen das Landes-Verwaltungsnetz, das als ein vertrauenswürdige Netzwerk einzustufen ist, eingesetzt wird.

Zur Erreichung der Sicherheit beim Zugriff auf Daten und Systeme werden im Landessystemkonzept eine Reihe von Maßnahmen vorgeschlagen, die im Rahmen des Einsatzes von Diensten im UIS behördenübergreifend zu treffen sind. Die Beiträge des IKE beschränkten sich daher darauf, nachzuweisen, daß bei der Realisierung solcher Maßnahmen die Funktionalitäten des UIS nicht wesentlich verschlechtert werden. Darüber wird im Folgenden berichtet.

Wichtige Komponenten eines Sicherheitskonzeptes in diesem Schwerpunkt sind:

- Firewall-Rechner zur Kopplung von Intranet und Internet.
- Ein Zugangskontrollsystem auf Basis eines Rollenmodells und unter Einbeziehung einer behördenübergreifenden Zertifizierungsstelle.
- Ein Verschlüsselungskonzept auf Basis Secure Socket Layer Konzeptes (SSL), auf das sowohl NETSCAPE als auch Microsoft aufbauen.

Zugangskontrollen werden schon jetzt auf einzelnen Servern praktiziert. Mit der Einrichtung von Zertifizierungsstellen beschäftigen sich verschiedene Stellen wie etwa auch das Regionale Rechenzentrum der Universität Stuttgart (siehe etwa <http://rtca.rus.uni-stuttgart.de/POLICY/>).

Die Erfahrungen aus diesen Versuchen sind abzuwarten und zu übernehmen.

Damit bleiben zwei Untersuchungspunkte:

- Der Einfluß von Firewalls auf die Inanspruchnahme von Diensten.
- Der Einfluß von Verschlüsselungstechniken auf die Datenübertragung zum PDN.

3.1 Anforderungen an eine Firewall bei Nutzung von Computer Supported Cooperative Working Techniken

Bei einer Konferenz mit NetMeeting können Daten über Audio, Video und als Datenströme ausgetauscht werden. Im Bereich Audio und Video unterstützt NetMeeting den Standard H.323, im Bereich Daten-Ströme den Standard T.120.

Um eine Daten-Konferenz (Application Sharing, Whiteboard, Chat-Tool und File Transfer) über eine Firewall zu betreiben, muß die Firewall „Primary“ (Initial) TCP Verbindungen mit statischen Ports unterstützen. Bei Audio und Video (Videokonferenzen) ist die Unterstützung von „Secondary“ (Subsequent) TCP und UDP Verbindungen mit dynamischen Ports erforderlich.

Firewall-Komponenten können in Abhängigkeit der Sicherheitspolitik der jeweiligen Organisation auf unterschiedliche Art und Weise konfiguriert werden. Während es die meisten Firewalls ermöglichen, „Primary“ und „Secondary“ TCP und UDP-Verbindungen zu steuern, werden in der Regel nur bestimmte (benötigte) Verbindungen konfiguriert. Einige Firewalls unterstützen nur „Primary“ TCP Verbindungen. Diese Firewalls werden als sehr sicher eingestuft. Als

Nachteil kann sich aber die Einschränkung der zu nutzenden Dienste einer Applikation erweisen, sofern diese Dienste „Secondary“ TCP- und UDP-Verbindungen benötigen.

Um mit NetMeeting Daten-Konferenzen durchzuführen, muß die Firewall „Primary“ TCP Verbindungen über genau definierte Ports ermöglichen. Für Audio- und Video-Konferenzen muß die Firewall „Secondary“ TCP und UDP Verbindungen über sich ändernde (dynamische) Ports unterstützen.

Sind über eine Firewall nur „Primary“- und keine „Secondary“-Verbindungen möglich, so kann bei NetMeeting kein Audio und Video genutzt werden.

Soll eine NetMeeting-Konferenz mit allen Funktionalitäten von NetMeeting über eine Firewall geführt werden, so muß man die Firewall so konfigurieren, daß die in Tab. 3 aufgelisteten Ports nicht blockiert werden.

Port	wird benötigt für	Protokoll
389	Internet Locator Server	TCP
522	User Location Service	TCP
1503	T.120	TCP
1720	H.323 call setup	TCP
1731	Audio call control	TCP
dynamisch	H.323 call control	TCP
dynamisch	H.323 streaming	RTP über UDP

Tab. 3: Zuordnung der NetMeeting-Dienste zu entsprechenden Ports

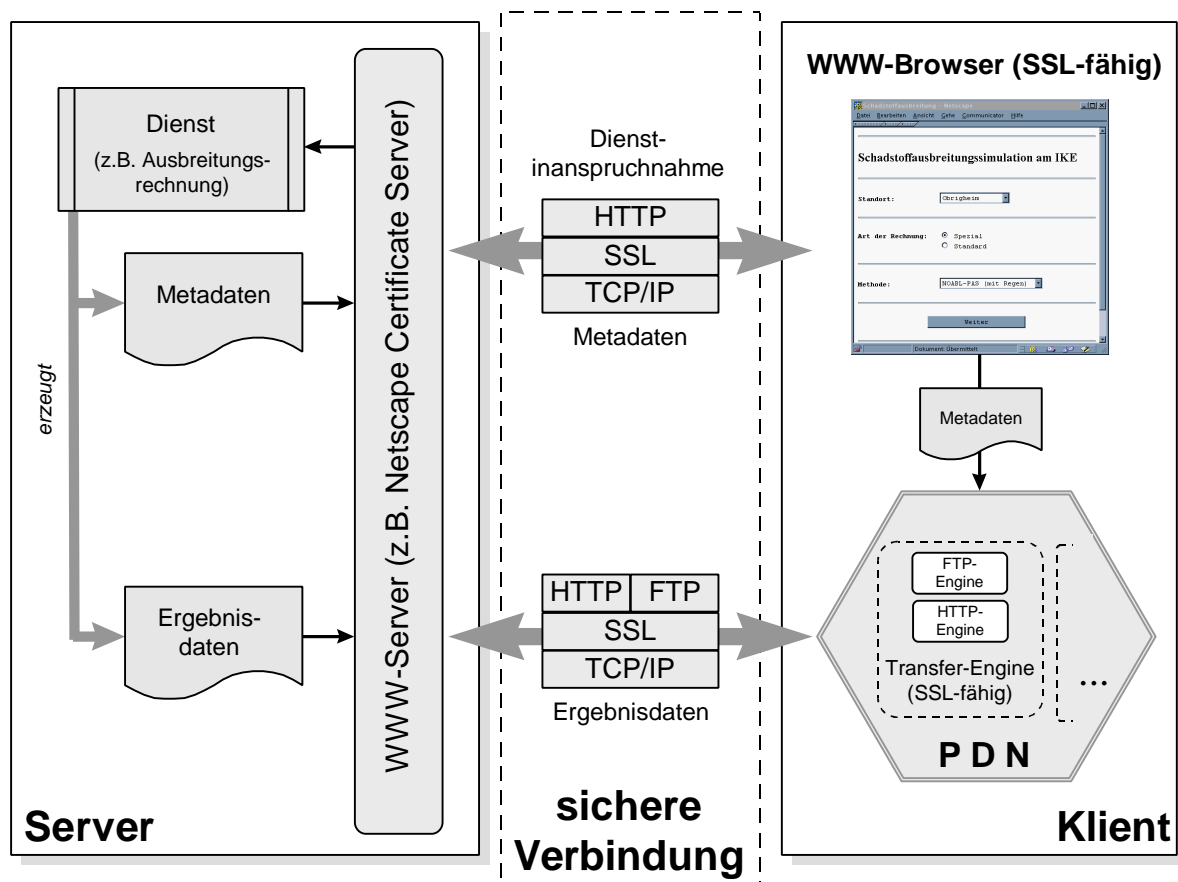
3.2 Datenübertragung unter Nutzung des Verschlüsselungsprotokolls SSL

Die im Internet verwendeten Anwendungsprotokolle wie das Hypertext Transfer Protocol (HTTP), das File Transfer Protocol (FTP) oder das Telnet verfügen über keine Verschlüsselungsmechanismen und verwenden zur Authentifizierung lediglich Paßwörter. Secure Socket Layer (SSL) ist ein von Netscape entwickelter offener Protokollstandard, der diese Lücke schließen soll, indem er Verschlüsselung und zertifikatbasierte Authentifizierung ermöglicht.

SSL fügt zwischen Anwendungs- und TCP-Schicht eine zusätzliche Kryptoschicht ein, die für die benachbarten Schichten weitgehend transparent ist. Das Protokoll, das von SSL abgearbeitet wird, sieht vor dem Austausch der Nutzdaten einen Handshake zur Authentifizierung mittels Zertifikaten vor. Mit den darin enthaltenen öffentlichen RSA-Schlüsseln wird ein Sitzungsschlüssel ausgetauscht, mit dem die anschließende Kommunikation codiert wird. Hierzu dienen symmetrische Standardverfahren wie DES oder RC4. Die Zertifikate gemäß dem X.509-Standard werden von einer Certification Authority (CA) ausgestellt, einer unabhängigen, vertrauenswürdigen Instanz.

Um SSL zu nutzen, müssen sowohl Server als auch Klient das zugrundeliegende Protokoll kennen, es sind aber in der Regel keine Modifikation auf der Anwendungsebene nötig. Vom

IKE wurde der PDN so erweitert, daß er nach SSL verschlüsselte HTTP-Files verstehen kann. Die dazu nötigen Modifikationen sind in Abb. 8 gezeigt.



Dipl.-Ing. R. Kopetzky, Universität Stuttgart, IKE 1997

Abb. 8.: Einschluß einer Kryptoschicht in den PDN

Auf Serverseite wurde vom FAW ein WWW-Server aufgesetzt, auf dem ebenfalls SSL implementiert war. Der Tausch von Daten konnte wie gewohnt durchgeführt werden. Die durch die Verschlüsselung auftretenden Verzögerungen waren klein gegenüber den Zeiten, die für Datenbereitstellung und Datenübertragung benötigt wurden.

4. Literatur

- /1/ Franklin, S., Graesser, A.: Is it an Agent, or just a Program ?, in: Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag, 1996.
- /2/ Workflow Management Coalition: The Workflow Management Coalition Specification, Juni 1996.
- /3/ Mayer-Föll, R., Jaeschke, A. (Hrsg.): UIS Baden-Württemberg, Projekt GLOBUS. Phase III, Forschungszentrum Karlsruhe GmbH, Karlsruhe, 1996.
- /4/ Kopetzky, R., Schmidt, F.: Entwurf und Implementierung eines Klienten zur Verarbeitung von Informationen aus verteilten Systemen. IKE Universität Stuttgart, 1996 (IKE 4-145).

- /5/ Szillat, H.: SGML - Eine praktische Einführung. Thomson Publishing, Bonn, 1995.
- /6/ Schulz, J.-E.: Integration von Daten, Methoden und Metainformationen in Umweltinformationssysteme, in: Das Software-Labor (Zweiter Bericht), Stuttgart, 1996.
- /7/ Lurk, A.: Bewertung von Multimedia-Software im Hinblick auf ihre Nutzung in einem Umweltinformationssystem. IKE Universität Stuttgart, 1997 (IKE 4-FB-DFN-7).
- /8/ Schulz, J.-E.: Repository Übersicht, Interner IKE-Bericht, Stuttgart, 1996.
- /9/ Labrou, Y., Finin, T.: A Proposal for a new KQML Specification, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, 1997.
- /10/ Jackson, P.: Introduction To Expert Systems (Second Edition), Addison-Wesley, publishing Company Inc., Wokingham, 1990.
- /11/ Scheer, A.-W. (Hrsg.): Metamodelle des Workflow-Managements, in: Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 121, Berlin, 1995.
- /12/ debis Systemhaus: Spezifikation der Teilerneuerung der Kernreaktor-Fernüberwachung Baden-Württemberg. Reviewfassung vom 16.09.1997.
- /13/ Sohn, A., et al.: Dosismaxima im Notfallschutz bei Regen. IKE Universität Stuttgart, Sept. 1996 (IKE 6-UVM 1).
- /14/ <http://www.microsoft.com/netmeeting/>
- /15/ Alber, C., Lurk, A.: Ausarbeitung und Erprobung eines Konzeptes zur teamorientierten Erstellung hypermedialer Berichte. IKE Universität Stuttgart, 1997 (IKE 4 D-214).

WWW-UDK

Erweiterungen der Software zum komfortablen Recherchieren und Retrieval und Entwicklung von Komponenten zur Erfassung und Pflege von UDK-Daten via WWW

*R. Nikolai,
Forschungszentrum Informatik (FZI),
Haid-und-Neu-Str. 10-14,
76131 Karlsruhe*

1. EINLEITUNG	83
2. ERWEITERUNGEN DES HTML-BASIERTEN RECHERCHE-MODULS.....	83
2.1 GDK-THEMENSUCHE.....	83
2.1.1 <i>Motivation</i>	83
2.1.2 <i>Benutzung der GDK-Themensuche</i>	84
2.2 ÜBERSICHTLICHERE GESTALTUNG DER ERGEBNISLISTEN.....	84
2.2.1 <i>Seitenweise Ausgabe der Ergebnisliste</i>	84
2.2.2 <i>Verweise von Ergebnislisten auf detaillierte Informationen</i>	85
2.3 VERBESSERUNG DER DARSTELLUNG VON UDK-OBJEKTEN	85
3. UNTERSTÜTZUNG DES RETRIEVALS MIT JAVA-APPLETS.....	85
3.1 KOMBINATION EINER HTML/CGI-ANWENDUNG MIT JAVA-APPLETS: DER APPLLETVERWALTER	85
3.1.1 <i>Problemstellung und generelle Vorgehensweise</i>	85
3.1.2 <i>Entscheidung Applet läuft / Applet muß gestartet werden</i>	86
3.1.3 <i>Kommunikation zwischen CGI-Programm und Applet</i>	86
3.1.4 <i>Weg/Zeit-Diagramm</i>	87
3.2 NAVIGATION IM STRUKTURBAUM: DER UDK-NAVIGATOR	87
3.2.1 <i>Problemstellung</i>	87
3.2.2 <i>Anforderungen</i>	88
3.2.3 <i>Benutzung des UDK-Navigators</i>	89
4. ERFASSEN VON UDK-DATEN ÜBER DAS WWW	90
4.1 MOTIVATION	90
4.2 ERFASSEN UND PFLEGEN VON VERWEISEN (URLS): LINKEDIT	90
4.2.1 <i>Motivation</i>	90
4.2.2 <i>Anforderungen</i>	91
4.2.3 <i>Benutzung von LinkEdit</i>	91
5. ZUSAMMENFASSUNG UND AUSBLICK.....	93
6. LITERATUR.....	94

1. Einleitung

Der Umwelt-Datenkatalog (UDK) ist ein Metainformationssystem /4/, /5/, /6/, /7/, speziell für Umweltinformationen, das ursprünglich auf eine Initiative des Niedersächsischen Umweltministeriums hin entwickelt wurde. Im Laufe der Zeit hat sich eine internationale Länderkooperation gebildet, die für seine Einführung sorgte und dessen Weiterentwicklung fördert. Inzwischen wird er in fast allen Bundesländern sowie in Österreich eingesetzt und hat sich als Standard für Metainformationen über Umweltinformationen etabliert. Im UDK werden beschreibende Informationen und Zugangswege zu Datenbeständen des Umweltinformationssystems (UIS) bereitgestellt.

Der UDK im UIS Baden-Württemberg wird den übergreifenden UIS-Komponenten zugeordnet. Das sind Systeme, die der Zusammenführung und fachübergreifenden Nutzung von Informationen aus verschiedenen Umwelt- oder Zuständigkeitsbereichen dienen /9/. Damit spielt der UDK im UIS Baden-Württemberg eine zentrale Rolle.

Bereits seit 1995 werden die Daten des UDK durch den vom FZI entwickelten WWW-UDK /12/ sowohl behördenintern als auch - seit 1996 - einer breiten Öffentlichkeit über das WWW zugänglich gemacht. Ursprünglich war WWW-UDK ein Werkzeug, das ausschließlich die Recherche in den Daten des UDK ermöglichte. Dieses Recherchewerkzeug wurde nach und nach erweitert (insbesondere auch dank der Kooperation mit dem österreichischen Umweltbundesamt) und die Benutzungsfreundlichkeit wurde mit Voranschreiten der technischen Entwicklungen verbessert. Auch 1997 fanden solche Erweiterungen statt, die u.a. aufgrund den Ergebnissen einer WWW-Fragebogenaktion durchgeführt wurden. Diese Erweiterungen werden in Kapitel 2 und – insofern sie als Java-Applets realisiert wurden – in Kapitel 3 vorgestellt. Desweiteren wurde WWW-UDK aber auch um eine Reihe von Administrationswerkzeugen ergänzt, die dem Administrator das Erfassen und Pflegen von UDK-Daten erleichtern sollen. Eines dieser Werkzeuge, LinkEdit, wird in Kapitel 4 vorgestellt. Unter Berücksichtigung der positiven Erfahrungen mit LinkEdit erscheint eine Pflege der gesamten Datenbestände des UDK möglich und wurde für UDK-Objekte bereits vom österreichischen Umweltbundesamt beauftragt.

2. Erweiterungen des HTML-basierten Recherche-Moduls

Aufgrund den Erfahrungen und Rückmeldungen der Benutzer des WWW-UDK wurden eine Reihe von Erweiterungen und Verbesserungen am Recherchemodul vorgenommen. Die wesentlichen werden im folgenden kurz vorgestellt.

2.1 GDK-Themensuche

2.1.1 Motivation

Im bisherigen WWW-UDK fehlte eine Suchmöglichkeit, in der der Benutzer nicht von vorneherein einen Suchbegriff eingibt, sondern weniger zielgerichtet in den Daten „herumstö-

bern“ kann. Hierzu bietet sich die Hierarchie der Begriffe des Grunddatenkatalogs (GDK), von denen jeweils einer einem UDK-Objekt zugeordnet werden kann, geradezu an. Der im UDK verwendete GDK ist eine Hierarchie von Umweltthemen. Auf oberster Ebene sind dies: „Sozioökonomische Daten“, „Energie“, „Flächennutzung“, „Landwirtschaft“, „Nahrung“, „Boden“, „Natur und Landschaft“, „Wald“, „Abfall“, „Luft“, „Lärm“ und „Wasser“. Entlang dieser sowie der entsprechend untergeordneten Umweltthemen (z.B. ist der Begriff „Geschützte Landschaftsteile“ Unterbegriff von „Natur und Landschaft“) sollte dem Benutzer eine thematische Suche im WWW-UDK ermöglicht werden.

2.1.2 Benutzung der GDK-Themensuche

Der Einstieg in die GDK-Themensuche erfolgt über die oberste Ebene der GDK-Hierarchie. Die 12 Umweltthemen werden aufgelistet und es wird jeweils angezeigt, ob zu diesem Thema Daten im UDK vorhanden sind oder nicht. Nun können entweder die entsprechenden UDK-Objekte angezeigt werden oder aber es kann zur nächsten Ebene eines der GDK-Themen gewechselt werden. Dem Benutzer wird bei den GDK-Themen jeweils angezeigt, ob die unterste Ebene der Begriffe erreicht ist oder nicht. Ohne einen einzigen Suchbegriff eingegeben zu haben, gelangt er so zu den thematisch geordneten UDK-Objekten und kann sich rasch über die im UDK erfaßten Datenbestände einen Überblick verschaffen.

2.2 Übersichtlichere Gestaltung der Ergebnislisten

2.2.1 Seitenweise Ausgabe der Ergebnisliste

Inzwischen enthält der baden-württembergische UDK ca. 1.000 UDK-Objekte. Bei relativ unspezifischen Suchen (wie z.B. eine allgemeine Suche nach Umweltdaten mit dem Suchbegriff „Wasser“) können daher sehr lange Ergebnislisten zurückgeliefert werden. Um die daraus resultierenden langen Wartezeiten zum vollständigen Laden einer Ergebnisliste in den Web-Browser sowie ein unhandliches Scrollen in diesen Listen zu vermeiden, werden die Ergebnisse seitenweise und nach UDK-Klasse sortiert ausgegeben. Zwischen den verschiedenen Seiten sollte beliebig gewechselt werden können. Die Anzahl der Ergebnisse pro HTML-Seite kann vom Benutzer in den Suchformularen konfiguriert werden.

Um auf eine bestimmte Anzahl von Ergebnissen einer Datenbankfrage zurückgreifen zu können (z.B. auf die 10. bis 20. Ergebnisdatensätze), müssen sie entsprechend vorgehalten werden. Hierzu ist ein serverseitiger Cache erforderlich. Der serverseitige Cache wurde realisiert, indem die Ergebnisse einer UDK-Anfrage in Datenbanktabellen abgelegt werden. Da bei der Suche nach Umweltdaten andere Ergebnisstrukturen entstehen als bei der Suche nach Adressen, wurden zwei neue Datenbanktabellen angelegt, PAGINATE für Ergebnisse der Suche nach Umweltdaten und PAGINATEA für Ergebnisse der Suche nach Adressen. Diese Tabellen werden bei der ersten Anfrage gefüllt. Folgeanfragen greifen darauf zurück, um die gewünschte Ergebnisdatenmenge zu holen, die dem Benutzer präsentiert wird.

2.2.2 Verweise von Ergebnislisten auf detaillierte Informationen

Um schnell von einer in der Ergebnisliste dargestellten Adresse bzw. einem UDK-Objekt zu den detaillierten Information zu gelangen, werden die Adressen bzw. UDK-Objekte in der Ergebnisliste als Verweis (Link) dargestellt. Per einfachem Mausclick können so die detaillierten Informationen abgerufen werden.

2.3 Verbesserung der Darstellung von UDK-Objekten

Bei der detaillierten Darstellung von UDK-Objekten fanden eine ganze Reihe kleinerer Umstellungen statt, um die ausführlichen Informationen, die zu einem UDK-Objekt gespeichert sind, dem Benutzer noch übersichtlicher zu präsentieren. So wurden die Felder umsortiert, damit die Adresse der Datenauskunft und der datenhaltenden Stelle schneller gefunden werden. Außerdem wird der entsprechende Verweis (Link) auf die vollständige Adreßinformation nun im Klartext ausgegeben statt nur das Adreßkennzeichen anzuzeigen. Zusätzlich wird das Datum der letzten Änderung eingeblendet, um dem Benutzer die Aktualität der Daten anzuzeigen.

3. Unterstützung des Retrievals mit Java-Applets

WWW-UDK ist als HTML/CGI-Lösung realisiert. Um benutzungsfreundlichere, interaktivere Benutzungsoberflächen für komplexe Interaktionen (z.B. die Navigation im Strukturbaum des UDK) zu realisieren, deren Komfort deutlich über den einer HTML-Oberfläche hinausgeht, wurden in die bestehende Lösung nahtlos verschiedene Java-Applets (z.B. der UDK-Navigator, vgl. Abschnitt 3.2) integriert. Dabei sollte sowohl ein Aufruf des WWW-UDK durch die in Java realisierten Komponenten als auch ein Aufruf von den in Java realisierten Komponenten durch den WWW-UDK möglich sein.

3.1 Kombination einer HTML/CGI-Anwendung mit Java-Applets: Der Appletverwalter

Während ein Aufruf von HTML-/CGI-basierten Systemen wie dem WWW-UDK durch in Java realisierten Komponenten relativ einfach zu realisieren ist (entweder durch Aufruf der entsprechenden URL oder durch Aufruf des Datenzugriffsdienstes), bereitete insbesondere der wiederholte Aufruf von Java-Applets aus HTML-CGI-Anwendungen wie dem WWW-UDK heraus Schwierigkeiten. Im Rahmen von GLOBUS wurde diese neue Herausforderung angegangen und durch Entwicklung des Appletverwalters gelöst.

3.1.1 Problemstellung und generelle Vorgehensweise

Die generelle Vorgehensweise ist, aus WWW-UDK heraus ein CGI-Programm oder ein Servlet (serverseitiges Java-Programm, das Bestandteil des WWW-Servers ist) zu starten, das als Eingabe den Namen und die Parameter für das Applet erhält. Dieses CGI-Programm bzw. Servlet überprüft, ob das entsprechende Java-Applet bereits auf dem Client-Rechner läuft. Ist

dies nicht der Fall, generiert es eine HTML-Seite, die einen Applet-Tag beinhaltet, mit dem das Applet (parametrisiert) gestartet wird. Läuft das Applet bereits, müssen ihm nur die entsprechenden Parameter mitgeteilt werden (z.B. die dekadische Notation von einem UDK-Objekt, das dargestellt werden soll). Da dennoch eine HTML-Seite generiert werden muß, wird eine bereits im Web-Browser angezeigte Seite erneut generiert (derzeit das in einem eigenen Frame angezeigte UIS-Logo).

Als problematisch erweisen sich insbesondere zwei Punkte:

1. Das CGI-Programm muß entscheiden, ob das entsprechende Applet bereits auf dem Client läuft.
2. Das CGI-Programm muß, falls das Applet bereits läuft, mit diesem kommunizieren, um ihm die entsprechenden Parameter mitzuteilen.

Auf diese beiden Punkte soll im folgenden eingegangen werden. Wichtige Komponente, die in beiden Fällen benötigt wird, ist der Appletverwalter.

3.1.2 Entscheidung Applet läuft / Applet muß gestartet werden

Wenn das entsprechende Applet bereits auf dem Client läuft, darf es nicht erneut gestartet werden, da es dort ansonsten mehrfach auf dem Bildschirm vorhanden ist. Aus diesem Grund wird ein server-seitiger Appletverwalter realisiert. Bei diesem Appletverwalter melden sich die Applets an, sobald sie gestartet werden und ab, wenn sie beendet werden. Der Appletverwalter kann somit jederzeit Auskunft darüber erteilen, ob ein bestimmtes Applet auf einem bestimmten Client läuft oder nicht. Dazu bietet der Appletverwalter eine Methode `applet_running` an, die über RMI aufgerufen werden kann und „wahr“ oder „falsch“ zurück liefert.

Voraussetzung für die eindeutige Identifikation eines Applets durch den Appletverwalter ist, daß dieses sich mit einer eindeutigen ID identifiziert. Diese ID muß bei Anfragen an den Appletverwalter angegeben werden, also auch dem Anfrager bekannt sein. Daher wird bei erstmaligem Aufruf von WWW-UDK eine „Session-ID“ erzeugt, die auf allen folgenden HTML-Seiten des WWW-UDK und allen CGI-Programmen bekannt ist. Diese Session-ID kann zusammen mit dem Applet-Namen (z.B. UDK-Navigator) als eindeutige ID verwendet werden.

3.1.3 Kommunikation zwischen CGI-Programm und Applet

Ein in Java realisiertes CGI-Programm bzw. ein Servlet kann prinzipiell über RMI (Remote Method Invocation) mit entfernten Java-Objekten kommunizieren. Allerdings kann das CGI-Programm/Servlet nicht direkt das Applet über RMI kontaktieren, da die Sicherheitsmechanismen für Java-Applets nur einen Aufruf vom Applet heraus erlauben und nicht umgekehrt. Die Kommunikation zwischen CGI-Programm/Servlet und Java-Applet findet somit indirekt statt. Der Appletverwalter tritt dabei als Vermittler auf. Dazu baut das Applet beim Start über RMI eine Kommunikationsverbindung zum Appletverwalter auf. Das CGI-Programm teilt dann dem Appletverwalter mit (per RMI-Aufruf), welchem Applet er welche Informationen senden möchte. Diese werden dann vom Appletverwalter über Callbacks an das Applet weitergereicht.

3.1.4 Weg/Zeit-Diagramm

Die folgende Abbildung 3.1 zeigt ein vereinfachtes Weg/Zeit-Diagramm für den Aufruf eines Java-Applets durch WWW-UDK.

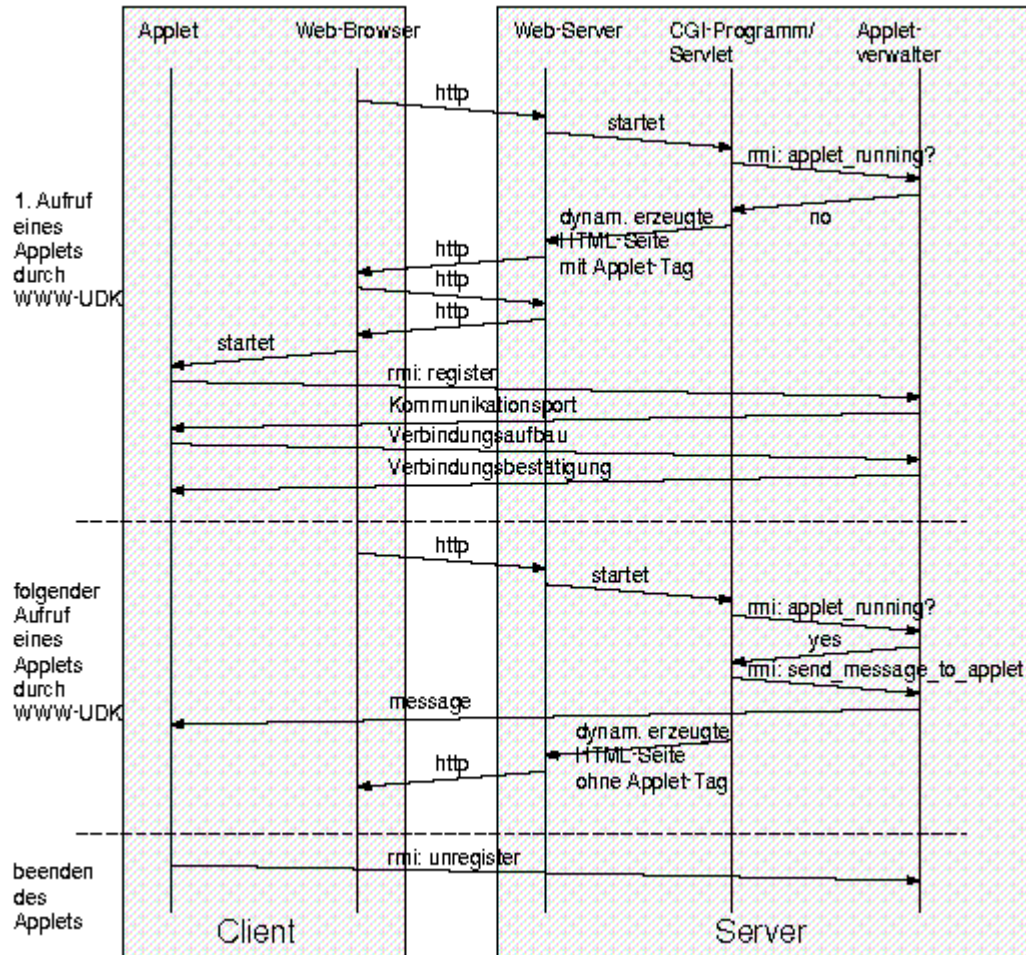


Abb. 3.1: Weg/Zeit-Diagramm: Aufruf eines Applets durch WWW-UDK

3.2 Navigation im Strukturbaum: Der UDK-Navigator

3.2.1 Problemstellung

UDK-Objekte sind in sog. Primärkatalogen hierarchisch geordnet. Diese Hierarchie sollte - ergänzend zu der bisherigen Vorgänger-/Nachfolgerdarstellung innerhalb der Detaildarstellung - graphisch dargestellt werden. Eine Navigation entlang der Hierarchie sollte durch einfaches Anklicken der Objekte ermöglicht werden. Im Auftrag des niedersächsischen Umweltministeriums wurde dazu der Java-basierte UDK-Navigator entwickelt, der auch in den WWW-UDK Baden-Württembergs integriert wurde. Eine Verbindung mit dem bisherigen WWW-UDK sollte in beide Richtungen möglich sein: Im UDK-Navigator ausgewählte Objekte können vom WWW-UDK dargestellt werden und umgekehrt. Die Kommunikation von WWW-UDK mit dem UDK-Navigator findet über den im vorangegangenen Abschnitt 3.1 vorgestellten Appletverwalter statt.

3.2.2 Anforderungen

Anforderungen an die Darstellung:

- Grafische Darstellung der Hierarchie der UDK-Objekte, gegeben durch den UDK-Primärkatalog, vergleichbar einem Filebrowser unter einer grafischen Benutzeroberfläche wie WindowsNT, OS/2 oder X-Windows.
- Ansprechendes Design der Oberfläche, möglichst selbsterklärend. Dies kann z.B. durch die Verwendung grafischer Knöpfe erreicht werden.
- Anzeige der Objektklasse durch ein entsprechendes Icon im Strukturbaum.
- Darstellung des Objektnamens und (optional) der dekadischen Notation im Strukturbaum.

Funktionale Anforderungen:

- Navigation durch den Strukturbaum: Nachfolgerobjekte durch Anklicken eines Eintrages sichtbar/unsichtbar machen, Scrollen durch gesamten Baum mittels Scrollbars.
- Selektieren eines Objektes durch Anklicken und darauffolgende Darstellung weiterer Detailinformationen (Attribute) in einem gesonderten Anzeigebereich. Geforderte Attribute sind: Name, Klasse (Klartext), dekadische Notation, letzte Änderung, Beschreibung, Suchbegriffe.
- Sortierung des Strukturbaumes nach Namen der Knoten/nach dekadischer Notation (wählbar durch Benutzer).
- Knotendarstellung im Strukturbaum mit/ohne Anzeige der dekadischen Notation (wählbar durch Benutzer).
- Aufruf der entsprechenden dynamisch generierten HTML-Seite des WWW-UDK, um sämtliche Informationen zu einem Objekt anzuzeigen (Realisierung durch Aufruf eines CGI-Skriptes).
- Schließen des Baumes bis auf den Wurzelknoten (hier: des Wurzelkataloges).
- Suche eines Objektes mittels seiner dekadischen Notation und Darstellung im Strukturbaum.
- Aufruf des UDK-Navigators mit einer dekadischen Notation und darauffolgende Darstellung des entsprechenden Objektes im Strukturbaum.
- Performante Datenbankanfragen, um unnötige Wartezeiten zu vermeiden.

3.2.3 Benutzung des UDK-Navigators

Nach dem Aufruf präsentiert sich der UDK-Navigator dem Benutzer wie in Abbildung 3.2 dargestellt.

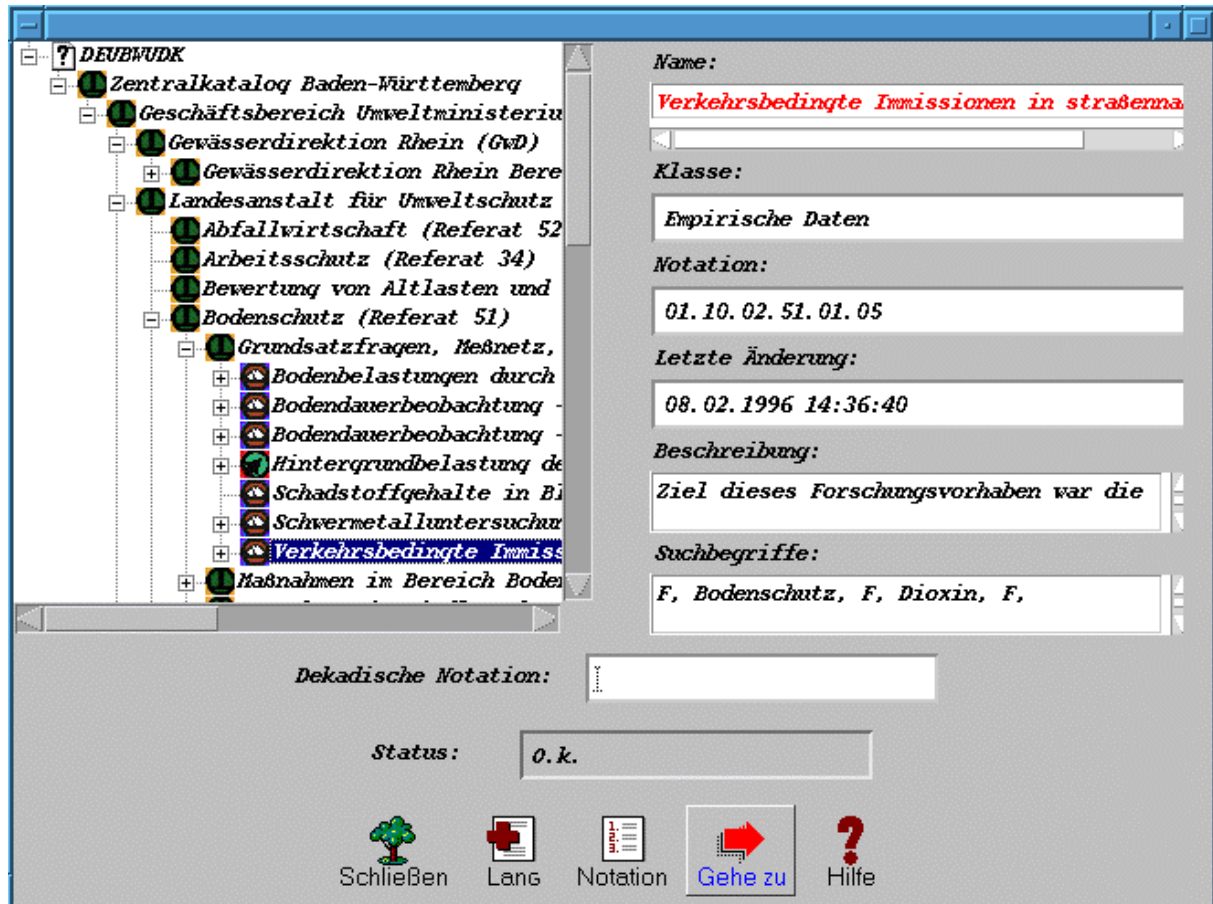


Abb. 3.2: Darstellung des UDK-Objektes “Verkehrsbedingte Immissionen in straßennahen Böden” im UDK-Navigator. In der Strukturbaumdarstellung sind die UDK-Objekte innerhalb der einzelnen Hierarchieebenen alphabetisch sortiert, da Kurzdarstellung gewählt wurde, ist die dekadische Notation nicht eingeblendet.

In der linken Hälfte wird der Strukturbaum des UDK angezeigt (bei erstmaligem Öffnen ohne Angabe einer initialen dekadischen Notation nur der "Wurzel"-Katalog). Das Öffnen eines Eintrags und somit die Anzeige der dem ausgewählten UDK-Objekt hierarchisch nachgeordneten Objekte ist über das „+“-Symbol möglich. Geschlossen wird ein angezeigter Teilbaum durch Anklicken des „-“-Symbols. Wird der Name eines im Strukturbaum angezeigten UDK-Objektes angeklickt, erscheinen im rechten Teil des UDK-Navigators zusätzliche Informationen über das Objekt: der Name des Objektes, seine Klasse, die dekadische Notation, die letzte Änderung, die Beschreibung und die Suchbegriffe.

Im unteren Bereich befinden sich mehrere Knöpfe und Textfelder:

- Schließen: „Schließen“ schließt den gesamten Strukturbaum und zeigt nur noch die Wurzel an, sämtliche offene Unterbäume werden ebenfalls geschlossen.

- Kurz / Lang: „Kurz“ zeigt nur die Namen der einzelnen Objekte im Strukturbaum an, „Lang“ gibt zusätzlich die dekadische Notation aus.
- Notation / Name: Hier kann die Sortierung der Einträge im Strukturbaum ausgewählt werden: entweder nach Name oder nach dekadischer Notation.
- Gehe zu: Über diesen Button wird die HTML-Seite des aktuellen Eintrags im WWW-UDK aufgerufen.
- Hilfe: Darstellung der Hilfe-Informationen.
- Dekadische Notation: Die Eingabe einer dekadischen Notation in dieses Textfeld erlaubt die Suche nach einem UDK-Objekt, dessen Notation bekannt ist. Das gefundene Element wird im Strukturbaum dargestellt und die Zusatzinformationen werden angezeigt.

Die direkte Eingabe einer dekadischen Notation wird wohl nur vom Experten verwendet werden. Die einfachere Möglichkeit, ein bestimmtes UDK-Objekt im UDK-Navigator anzuzeigen ist, dieses im WWW-UDK zu suchen, die Detaildarstellung zu wählen und dann den Knopf „Darstellung im UDK-Navigator“ zu drücken. WWW-UDK benachrichtigt daraufhin über den Appletverwalter den UDK-Navigator, der dann das entsprechende Objekt darstellt.

4. Erfassen von UDK-Daten über das WWW

4.1 Motivation

WWW-UDK war bisher ein reines Recherchewerkzeug. Inzwischen erlauben die Techniken und Werkzeuge des WWW allerdings auch die Erfassung und Pflege von Datenbankinhalten über das WWW. Insbesondere bietet sich hier Java an: Zum einen ist durch die Plattformunabhängigkeit garantiert, daß der Code überall ausgeführt werden kann, zum anderen werden die Schwächen des klassischen HTML/CGI-Ansatzes vermieden (z.B. zustandsloses HTTP-Protokoll, mangelnde Interaktivität). So kann etwa die dezentrale Pflege einer zentralen UDK-Datenbank durch das Bereitstellen von Java-Applets ermöglicht werden, ohne daß dazu entsprechende Software auf dem Client installiert werden muß. In einem ersten Schritt wurde dazu vom FZI ein Applet zum Erfassen und Pflegen der URLs realisiert.

4.2 Erfassen und Pflegen von Verweisen (URLs): LinkEdit

4.2.1 Motivation

Das für die PC-Version des UDK entwickelte Datenmodell bietet keinerlei Möglichkeiten, um auf die von den Metadaten im UDK beschriebenen (Basis-)Daten, also die Umweltdatenobjekte wie etwa Umweltberichte und Umweltdatenbanken on-line zuzugreifen. Um als Informationssuchender zu den eigentlichen Informationen zu gelangen, ist der angegebene Ansprechpartner zu kontaktieren. Da für das UIS Baden-Württemberg aber bereits verschiedene (Basis-) Datenbestände speziell für das WWW aufbereitet wurden und dieses Angebot von umweltrelevanten Informationen im WWW weiter wächst, wurde das Datenmodell des UDK bereits innerhalb des Projektes GLOBUS III durch das FZI erweitert /3/. Durch diese Erweiterung, die voll kompatibel zum Datenmodell der PC-Version des UDK ist, erlaubt WWW-

UDK die Verknüpfung von bis zu 99 URLs je UDK-Objekt sowie zusätzlich Typinformationen, die Hinweise auf Format und Umfang der Informationen, auf die verwiesen wird, geben. Die Funktionalität des WWW-UDK als Auskunftssystem und Nachweissystem wird somit um die Funktionalität eines Verweissystems ergänzt.

Da die PC-Version des UDK weiterhin die Erfassung und Pflege solcher Verweisinformationen nicht unterstützt, wurden diese Informationen bisher ohne Werkzeugunterstützung direkt in die Datenbank des UDK eingetragen. Dazu waren sowohl Kenntnisse über Datenbanken allgemein als auch über das Datenmodell des UDK erforderlich. Auch mit diesen Kenntnissen bleibt der Vorgang mühsam. An diesem Punkt setzt das Applet LinkEdit an: Es stellt ein komfortables Werkzeug für den Administrator dar, mit dem diese Verweise angelegt, bearbeitet und eventuell wieder gelöscht werden können. Es ist dazu in die Administrations-Version des WWW-UDK nahtlos eingebettet.

4.2.2 Anforderungen

Die wesentlichen funktionalen Anforderungen an LinkEdit waren:

- LinkEdit sollte sämtliche Verweisinformationen eines UDK-Objektes anzeigen können. Diese Informationen sollen modifiziert werden können.
- LinkEdit soll das Einfügen neuer Verweise unterstützen. Wird zu einem UDK-Objekt erstmals ein Verweis angelegt, soll durch Aufruf des entsprechenden CGI-Skripts eine Objekt-ID erzeugt und der UDK-Administrator benachrichtigt werden (vgl. Abschnitt 1.4.4.1 in /3/).
- Beim Einfügen bzw. Modifizieren sollen die Eingaben des Benutzers auf Vollständigkeit überprüft werden. Fehlen Angaben, so soll der Benutzer zur Eingabe dieser Angaben aufgefordert werden
- Typ und Untertyp sollen durch Auswahl von den in der Datenbank vorhandenen Typen bzw. Untertypen eingegeben werden können.
- Das Einfügen neuer Verweise soll durch eine frei definierbare Vorlage unterstützt werden. Diese Vorlage kann entweder ein bereits in der Datenbank gespeicherter Verweis sein oder aber explizit angegeben werden. Beim Editieren von Verweisen kann die aktuell gültige Vorlage übernommen werden. Somit wird das wiederholte Eintippen ähnlicher Verweisinformationen vermieden.
- LinkEdit soll das Löschen von Verweisen unterstützen.

4.2.3 Benutzung von LinkEdit

LinkEdit wird gestartet, wenn man die Administrations-Version des WWW-UDK, die zusätzlich im behördeninternen Intranet eingesetzt wird, benutzt und nach der Auswahl eines UDK-Objektes in der detaillierten Darstellung den Knopf „Editieren der Verweise auf Basisdaten“ mit der Maus anklickt. Nach dem Laden des Applets präsentiert es sich dem Benutzer wie in Abb. 4.1 dargestellt.

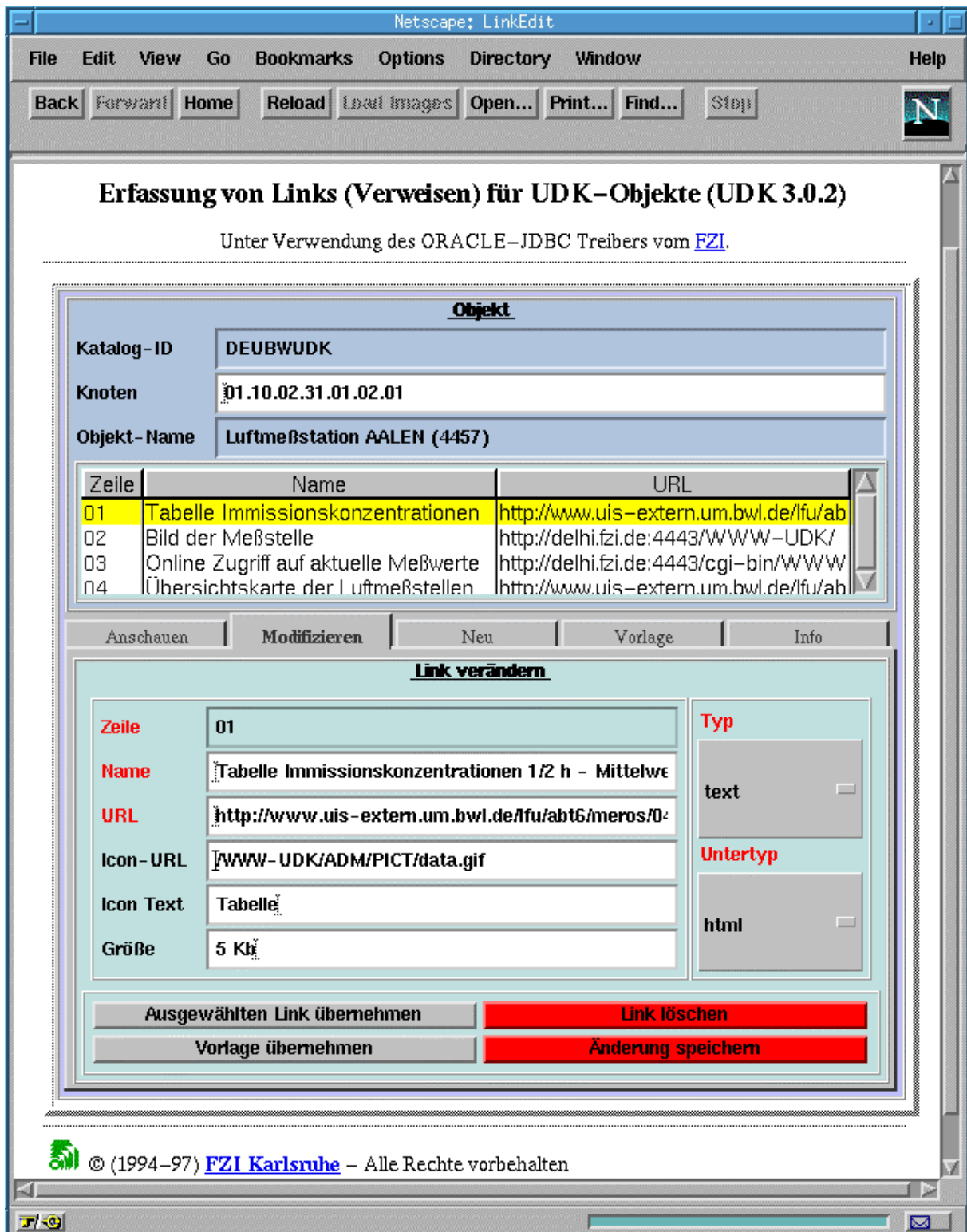


Abb. 4.1: Modifizieren eines Verweises (typisierte URL) eines UDK-Objektes mit LinkEdit.

LinkEdit zeigt im oberen Drittel das zur Bearbeitung ausgewählte UDK-Objekt. Das UDK-Objekt wird durch eine Katalognummer und seine dekadische Notation ausgewählt. Um ein schnelles Wechseln der Objekte eines Katalogs zu ermöglichen, ist das Feld mit der dekadischen Notation (Knoten) editierbar. Wenn der Inhalt dieses Feldes durch den Administrator verändert wird, wird nach einem Objekt mit dieser dekadischen Notation in der Datenbank

gesucht und nach erfolgreicher Suche können die Verweise auf Basisdaten zu diesem Objekt geändert werden.

Im zweiten Drittel der Anzeige befindet sich eine mehrspaltige Liste, in der die Verweise (Links) zu Basisdaten von diesem Objekt in Kurzform aufgeführt sind. Aus dieser Liste kann ein Verweis zur Bearbeitung ausgewählt werden.

Der untere Teil von LinkEdit besteht aus einer Art Register, mit dem die Programmmodi „Anschauen“, „Modifizieren“, „Neu“, „Vorlage“ und „Info“ ausgewählt werden. Diese Modi sind im wesentlichen selbsterklärend. Auf eine detailliertere Beschreibung dieser Modi muß hier aus Platzgründen verzichtet werden, es sei auf die entsprechende on-line Dokumentation verwiesen.

5. Zusammenfassung und Ausblick

Mit WWW-UDK steht inzwischen in einer Produktionsversion ein Werkzeug zur Verfügung, das den komfortablen Zugriff auf UDK-Daten über das WWW ermöglicht. WWW-UDK wird derzeit außer in Baden-Württemberg auch in Österreich, Niedersachsen, Sachsen-Anhalt und Brandenburg (voraussichtlich ab Januar 1998) eingesetzt. Das Recherchewerkzeug WWW-UDK konnte durch eine Reihe von Detailverbesserungen, aber insbesondere durch die Integration des UDK-Navigators, der erstmals die Hierarchie des Strukturbaumes grafisch veranschaulicht, weiter verbessert werden. Aufgrund der bereits vorliegenden positiven Erfahrungen bei Einsatz des UDK-Navigators liegt es nahe, in einer weiteren Projektphase auch einzelne Suchmasken des WWW-UDK durch Java-Applets bereitzustellen. So kann der Benutzer bei der interaktiven Formulierung seines Informationsbedarfes noch besser unterstützt werden. Der entwickelte Appletverwalter bietet für die nahtlose Einbindung solcher und weiterer Applets beste Voraussetzungen.

Da der Informationsbedarf des Recherchierenden in der Regel nicht nach Erhalt der Metadaten befriedigt ist, sind Verweise (URLs) wichtiger Bestandteil eines Metainformationssystems im WWW. Mit LinkEdit wird erstmals ein Werkzeug zur Verfügung gestellt, mit dem diese Verweise komfortable erfaßt und gepflegt werden und bedeutet somit eine wichtige Erleichterung für UDK-Administratoren.

Basierend auf den Erfahrungen mit LinkEdit bei der Erfassung von UDK-Daten über das WWW entstand die Idee, den gesamten UDK-Objektbestand über das WWW dezentral pflegbar zu machen. Diese Idee wird vom FZI für das österreichische Umweltbundesamt bereits prototypisch realisiert. Insgesamt wird WWW-UDK somit vom Auskunfts- und Nachweissystem zum Verweissystem und in einem ersten realisierten Schritt auch zum Metadatenmanagementsystem.

6. Literatur

- /1/ Mayer-Föll, R.; Jaeschke, A. (1995): Projekt GLOBUS, Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten im Umweltinformationssystem Baden-Württemberg, Phase II 1995. Number FZKA 5700, in Wissenschaftliche Berichte, Forschungszentrum Karlsruhe Technik und Umwelt, Karlsruhe
- /2/ Mayer-Föll, R.; Jaeschke, A. (1996): Projekt GLOBUS, Konsolidierung der neuen Systemarchitektur und Entwicklung erster Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg, Phase III 1996. Number FZKA 5900, in Wissenschaftliche Berichte, Forschungszentrum Karlsruhe Technik und Umwelt, Karlsruhe
- /3/ Kramer, R.; Nikolai, R. (1996): WWW-UDK/Metadaten, in /2/
- /4/ Lessing, H.; Schütz, T. (1995): Der Umwelt-Datenkatalog als Instrument zur Steuerung von Informationsflüssen, in: /10/, S.159-167
- /5/ Lessing, H.; Günther, O.; Swoboda, W. (1995): Ein objektorientiertes Klassenkonzept für den Umweltdatenkatalog (UDK), in: Kremers, H.; Pillmann, W., Raum und Zeit in Umweltinformationssystemen, 9th International Symposium Computer Science for Environmental Protection CSEP'95, in: Umwelt-Informatik Aktuell, Metropolis Verlag Marburg, S.391-399
- /6/ Lessing, H.; Swoboda, W.; Günther, O. (1996): UDK, A European Environmental Data Catalogue, in Third International Conference/Workshop Integrating GIS and Environmental Modeling, Santa Fe
- /7/ Swoboda, W.; Lessing, H.; Grolimund, P.; Günther, O.; Haas, U.; Legat, R.; Vogler, M.; Zirm, K. (1995): Metadatenklassen im Umweltdatenkatalog (UDK), in /11/, S601-609
- /8/ Lessing, H.; Lipeck, U.W. (1996): Informatik für den Umweltschutz, 10 Symposium, Hannover 1996, Metropolis Verlag, Marburg
- /9/ Mayer-Föll, R.; Strohm, J.; Schultze, A. (1996): Das Umweltinformationssystem Baden-Württemberg, Überblick Rahmenkonzeption, in /8/
- /10/ Hilty, L.M.; Jaeschke, A.; Page, B.; Schwabl, A. (1994): Informatik für den Umweltschutz, 8 Symposium 1994, Metropolis Verlag, Marburg
- /11/ Huber-Wäschle, F.; Schauer, H.; Widmayer, P. (1995): Herausforderungen eines globalen Informationsverbundes für die Informatik, 25 GI-Jahrestagung und 13 Schweizer Informatikertag, Zürich
- /12/ Kramer, R.; Nikolai R.; Koschel A.; Rolker C.; Lockemann P.; Keitel A.; Legat R.; Zirm K.: WWW-UDK: A Web-based Environmental Metainformation System, ACM SIGMOD Record, März 1997

WWW-Szenario

*C. Rolker, R. Nikolai, R. Kramer, R. Schmidt,
Forschungszentrum Informatik (FZI),
Haid-und-Neu-Str. 10-14,
D-76131 Karlsruhe*

*C. Hofmann, Zs. Veszelka, J. Wiesel,
Institut für Photogrammetrie und Fernerkundung (IPF),
Universität Karlsruhe,
Englerstr. 7,
D-76128 Karlsruhe*

1. EINLEITUNG	98
2. ANFORDERUNGEN	99
2.1 ALLGEMEINE ANFORDERUNGEN	99
2.2 FUNKTIONALE ANFORDERUNGEN AN DIE WWW-UDK-SOFTWARE.....	100
2.3 FUNKTIONALE ANFORDERUNGEN AN DIE SACHDATENKOMPONENTE	100
2.4 FUNKTIONALE ANFORDERUNGEN AN DIE GIS-/GRAFIKKOMPONENTE.....	102
2.5 ZUSAMMENSPIEL DER KOMPONENTEN.....	103
2.6 STATUS DER KOMPONENTEN	103
3. GRUNDLAGEN	104
3.1 PROBLEMSTELLUNG.....	104
3.2 AUSGANGSBASIS.....	104
3.3 KOMMUNIKATION AUF CLIENTSEITE: JAVABEANS	105
3.4 KOMMUNIKATION INNERHALB DER VERTEILTEN UMGEBUNG.....	106
3.5 VERGLEICH RMI UND IIOP	107
3.6 AUFTEILUNG DER SOFTWARE-KOMPONENTEN AUF CLIENT UND SERVER	107
3.7 ZUGRIFF AUF DIE DATENBANKEN	107
4. SACHDATENKOMPONENTE	108
4.1 ÜBERSICHT	108
4.2 STEUERUNG	108
4.3 SELEKTOREN.....	109
4.4 UDK-ADAPTER	111
4.5 STATISTISCHE AUFBEREITUNG	111
4.6 DARSTELLUNGS-AUSWAHL	112
4.7 ERGEBNISVERWALTUNG	113
4.8 TABELLARISCHE DARSTELLUNG.....	113
4.9 HISTORIE	114
4.10 ARCHIV.....	114
5. UMWELTDATENKATALOG	115
5.1 AUFBAU DES WWW-UDK	115
5.2 ZUGRIFF AUF DEN WWW-UDK	116
5.2.1 Zugriffsmöglichkeiten auf die UDK-Datenbank mittels WWW-UDK.....	116
5.2.2 Vergleich der Zugriffsmöglichkeiten auf den WWW-UDK.....	117
5.3 AUFRUF VON ANDEREN SOFTWAREKOMPONENTEN AUS DEM WWW-UDK	118
6. GIS/GRAPHIK-KOMPONENTE	119
6.1 GESAMTARCHITEKTUR GIS-TERMINAL	119
6.1.1 Die Abstraktionsschicht.....	121
6.1.2 Die Managementschicht.....	122
6.1.3 Die Darstellungsschicht	123
6.2 MODULE UND KOMPONENTEN DES GISTERM	124
6.2.1 Grundlegende Module/Subsysteme	124
6.2.1.1 Schemaverwaltung.....	124
6.2.1.2 Ergebnisinstanzverwaltung (serverseitig)	124
6.2.1.3 Einheitliche Geometriedarstellung	124
6.2.1.4 Systembeschreibung	125
6.2.1.5 Darstellungsabbildung	125
6.2.1.6 Geometriereduktion	125
6.2.1.7 Rasterdatenmanagement	125
6.2.1.8 ShapeURLService.....	126
6.2.1.9 Shapefile Adapter.....	126
6.2.1.10 RIPS Adapter	126
6.2.2 Allgemeine graphische Komponenten	126
6.2.2.1 Diagrammdarstellung.....	127

6.2.2.2 Graphikkern	127
6.2.2.3 DisplayToolbar	127
6.2.2.4 Koordinatenanzeige	127
6.2.2.5 Layerverwaltung	128
6.2.2.6 Locator View	128
6.2.3 Anwendungsbezogene graphische Komponenten	129
6.2.3.1 Feature View	129
6.2.3.2 URL-Dialog	129
6.2.3.3 Spatial URL-Browser	130
6.2.3.4 Bereichsanfrage	131
6.3 MANAGEMENT HETEROGENER DATENQUELLEN	131
6.3.1 Arten von Geoinformationen	132
6.3.2 Das Datenmodell des GIS-Terminals	132
6.3.3 Abfragebearbeitung	134
6.3.4 Realisierung der Adapter	136
6.3.4.1 Kommunikation zwischen GIServer und den Adaptern	136
6.3.4.2 Der RIPS-Adapter	136
6.3.4.3 Der Shapefile-Adapter	138
6.4 KLIENTENSEITIGE UIS-ANWENDUNGSSCHNITTSTELLE DES GIS-TERMINALS	139
6.4.1 Darstellungsdienste	139
6.4.2 Auswahldienst	141
7. PROTOTYP	142
7.1 WWW-UDK	142
7.2 SACHDATENKOMPONENTE	142
7.3 GIS-/GRAFIKKOMPONENTE	143
7.4 KOMMUNIKATION	144
8. ZUSAMMENFASSUNG	144
9. LITERATUR	144

1. Einleitung

An der Landesanstalt für Umweltschutz (LfU) werden verschiedene spezialisierte Fachsysteme betrieben, die Umweltinformationen für die Landesverwaltung zugreifbar machen. Techniken und Werkzeuge des WWW erlauben neuerdings, Anwendungen zu entwickeln, mittels derer Umweltinformationen nicht nur ausschließlich verwaltungsintern - im sogenannten Verwaltungs-Intranet - sondern auch im WWW einer breiten Öffentlichkeit zur Verfügung gestellt werden können.

In verschiedenen Projekten wurden inzwischen Prototypen und Produkte entwickelt, die auf Basis der "klassischen" WWW-Technologien HTML und CGI (Common Gateway Interface) Umweltinformation behördenintern und für die Öffentlichkeit bereitstellen. Komplexe Anwendungen, die eine sehr interaktive Benutzungsschnittstelle erfordern, stoßen bei Einsatz dieser Technologien allerdings bald an die Grenzen des Möglichen. Java, eine von Sun unter besonderer Berücksichtigung der Randbedingungen im WWW (heterogene Plattformen, erhöhte Sicherheitsanforderungen, etc.) entwickelte Programmiersprache, die seit Anfang 1996 eine rasante Verbreitung findet, verspricht, hier Abhilfe zu schaffen.

Im Rahmen des Projektes GLOBUS III untersuchte die AG Java im Verlaufe des Jahres 1996 den möglichen Einsatz von Java für das Umweltinformationssystem (UIS) Baden-Württemberg /1/. Die Vorteile von Java gerade auch im Vergleich mit "klassischen" Internet-Technologien wie HTML und CGI sowie die breite Unterstützung von Java durch verschiedenste Hersteller trugen schließlich dazu bei, daß das Bild der AG Java über Java sehr positiv ausfiel. Kinderkrankheiten und andere Probleme waren weniger der sehr neuen Sprache Java selbst als mehr den Java-fähigen Browsern anzulasten.

Um die vielfältigen neuen Möglichkeiten für WWW-Anwendungen zu demonstrieren, die durch Java möglich werden, wurde von den Partnern der AG Java ein komplexeres Integrationsbeispiel realisiert /2/. Über ein Java-Applet wurde ein Zugang zu (ausgewählten) Luftmeßwerten geschaffen und die Meßreihen konnten als Business-Grafiken zusammen mit Vektorkarten im GIS-Viewer präsentiert werden. Das Beispiel zeigte zum einen, wie Java-Anwendungen der gewohnten Bedienungsfreundlichkeit von PC-Anwendungen nahe kommen, zum anderen die Tragfähigkeit der neuen Technologie auch für komplexere Anwendungen. Die seit Anfang 1997 verfügbare Version Java 1.1 beseitigt nicht nur einige der bisherigen Schwächen von Java 1.0, sondern verdoppelt beinahe die Geschwindigkeit des Java-Interpreters und enthält viele neue Features wie ein objektorientiertes Komponentenmodell, entfernte Methodenaufrufe, Druckmöglichkeiten und Verschlüsselung. Java wird somit zu einer idealen Programmiersprache in der heutigen Welt heterogener, vernetzter Systeme.

Aufbauend auf den Erfahrungen in GLOBUS III (insbesondere bei der Entwicklung des Java-basierten GIS-Viewers und der innerhalb der AG Java prototypisch realisierten Intranet-Anwendung für MEROS-Datenzugriffe) und den im Rahmen des Projekts Merlin gewonnenen Erkenntnissen (prototypischen Realisierung in Java für den Zugriff auf Meßnetze) wurde innerhalb dieses Projektes die Zusammenführung von Metainformationssystem, Sachdateninformationssystem und Geoinformationssystem zu einem über das WWW zugreifbaren Gesamtsystem entworfen und realisiert. Die erwarteten Vorteile dieser Entwicklung und der Integration bereits existierender Komponenten sind z.B. die zur Verfügungstellung von Umweltinformation unabhängig von Fachsystemen, eine bei Einsatz von HTML und Java erzielte

Plattformunabhängigkeit sowie eine durch komponentenbasierte Softwareentwicklung zu erreichende Wiederverwendbarkeit von Systemteilen. Zusätzlich können Kosten, die für die Systemadministration anfallen, deutlich reduziert werden, da Java-Anwendungen nicht auf den lokalen Clients installiert werden müssen, sondern bei Bedarf in der aktuellsten Version vom Anwendungsserver geladen werden. Die bei der Landesanstalt für Umweltschutz im jahrelangen tagtäglichen Einsatz der Fachsysteme gewonnenen Erfahrungen sollen genutzt werden, um bisher im wesentlichen dem geschulten Spezialisten zugängliche Umweltinformation auf einfache und leicht verständliche Art und Weise der Öffentlichkeit zugänglich zu machen.

Das vorliegende Text ist wie folgt aufgebaut: Im folgenden Kapitel 2 werden die funktionalen Anforderungen an die verschiedenen Komponenten des WWW-Szenarios (Sachdatenkomponente, Umweltdatenkatalog, GIS/Graphikkomponente) sowie deren Zusammenspiel vorgestellt. In Kapitel 3 werden die Möglichkeiten der Interaktion der verschiedenen Komponenten diskutiert, bevor in den Kapiteln 4, 5 und 6 die drei Komponenten einzeln vorgestellt werden. Welche Funktionalität schließlich Bestandteil des Prototyps ist, der bis Ende 1997 realisiert wurde, und welche Funktionalität erst in einer späteren Phase realisiert wird, wird in Kapitel 7 aufgeführt. Das Text schließt in Kapitel 8 mit einer Zusammenfassung.

2. Anforderungen

2.1 Allgemeine Anforderungen

Das Ziel des Projekts ist die Zusammenführung von Metainformationssystem, Sachdateninformationssystem und Geoinformationssystem zu einem über das WWW zugreifbaren Gesamtsystem, wobei die Oberfläche von Sachdaten- und Geoinformationssystem mit Java realisiert sein sollen.

Die drei Komponenten sind:

- Metadatenkomponente WWW-UDK: Umweltdatenkatalog, der Metadaten zu Umweltdaten hält, sowie WWW-Zugriff auf die Metadaten des UDK.
- Sachdatenkomponente: In einer Oracle-Datenbank sind Umweltdaten aus verschiedenen Fachsystemen abgelegt. Sogenannte Selektoren (vorstrukturierte, komfortable Anfragemöglichkeiten) sollen den Zugriff auf diese Daten realisieren. Es gibt z.Z. etwa 40 bekannte Selektoren, bei denen sowohl Eingabekriterien (Selektorkriterien) und Attribute der Ergebnisdatsätze festgelegt sind. Neben dem Auslesen der Datenbank soll die Sachdatenkomponente auch eine statistische Aufbereitung der Ergebnisdatsätze durchführen und die Weitergabe der Ergebnisse zur GIS-/Graphikkomponente und zu Spread-Sheets organisieren.
- GIS-/Graphikkomponente (im folgenden auch GIS-Komponente, Graphikkomponente oder GIS-term genannt): Die GIS-/Graphikkomponente ermöglicht die Anfrage und Visualisierung geographischer raumbezogener Daten. Numerische Wertereihen können in Business-Diagrammen präsentiert werden.

Die drei Komponenten sollen nahtlos zusammenspielen können, trotzdem soll jede auch unabhängig von den anderen Komponenten benutzt werden können.

Ein Schwerpunkt des Projekts ist die Navigation, Selektion, Aufbereitung und Darstellung von Meßwerten aus den Meßnetzen des Landes Baden-Württemberg. Bei der Systemkonzeption und bei der Realisierung war stets auf eine weitestgehende Allgemeingültigkeit bzw. auf eine generelle Wiederverwendbarkeit der zu entwickelnden Komponenten zu achten.

2.2 Funktionale Anforderungen an die WWW-UDK-Software

Bisherige Funktionalität (unter anderem):

- Der UDK liefert detaillierte Informationen über Karten- bzw. Sachdatenzugriffsdienste.

Vom UDK zum Selektor (Sachdatenkomponente oder GIS-/Grafikkomponente):

- Der UDK soll die navigatorische und deskriptive Suche nach Sachdaten-Selektoren und nach GIS-Diensten unterstützen.
- Der UDK soll die navigatorische und deskriptive Suche nach Karten unterstützen.

Aufruf des UDKs von der Sachdatenkomponente oder von der GIS-/Grafikkomponente:

- Der WWW-UDK soll parametrisierbar von den anderen Komponenten aufrufbar sein.
- Von der GIS-/Grafikkomponente soll nach allen Metainformationen (UDK-Objekten), die in einer Bounding Box oder in einer Verwaltungseinheit liegen, im UDK gesucht werden können (WWW-UDK: Geosuche).
- Es soll möglich sein, zu einem gesuchten Begriff die im UDK vorhandenen Metainformationen (UDK-Objekten) zu erhalten (WWW-UDK: Allgemeine Objektsuche)
- Der Aufruf des WWW-UDK soll sowohl mit WWW-UDK- als auch ohne WWW-UDK-Dialog durchgeführt werden können.

2.3 Funktionale Anforderungen an die Sachdatenkomponente

Funktionalität der Sachdatenkomponente:

- In einer Oracle-Datenbank sind Umweltdaten aus verschiedenen Fachsystemen abgelegt. Die Selektoren (vorstrukturierte, komfortable Anfragemöglichkeit) realisieren den Zugriff auf diese Daten. Es gibt z.Z. etwa 40 bekannte Selektoren, bei denen sowohl Eingabekriterien (Selektorkriterien) als auch die Attribute der Ergebnisdatensätze festgelegt sind.
- Ein Selektor soll in Abhängigkeit von gefundenen Ergebnissen auch weitere Selektoren aufrufen können.
- Selektionen können wahlweise kontextsensitiv oder -insensitiv durchgeführt werden
- Sowohl Selektoren als auch die Selektorkriterien sind - mit und ohne Dialog - parametrisiert aufrufbar (Vorbelegung).
- Die Selektorkriterien sind in beliebiger Reihenfolge aufrufbar.
- Zu den Selektorkriterien sollen als Hilfe für den Benutzer auch weitere Informationen (aus ORACLE-DB oder Textdateien) abgerufen werden können.

- Ein nicht näher bestimmtes Kriterium wird nicht in die Selektion aufgenommen.
- Abhängig von den gewählten Werten von bestimmten Kriterien sind u.U. noch weitere fachspezifische Kriterien notwendig. Sind weitere Kriterien hinzugekommen, dann sind die bereits ausgefüllten Kriterien nicht mehr änderbar. Sollte der Benutzer doch Änderungen vornehmen wollen, dann muß er einen neuen Selektor starten.
- Die Darstellung der möglichen Werte von Kriterien sind i.d.R. immer Klartexte (u.U. verknüpft mit Nummern bzw. mehreren DB-Attributen).
- Die Auswahllisten eines Kriteriums sollen Suche-Funktionalität anbieten (Einschränkung der Auswahlliste durch Eingabe von Teilstrings mit und ohne Platzhaltersymbol).
- Abschätzung der erwarteten Anzahl der Ergebnisse: Nach dem Auftrag der Selektion soll eine Abschätzung der erwarteten Anzahl der Ergebnisdatensätze angeboten werden; bei Überschreitung einer noch festzulegenden Anzahl soll der Selektor nicht ausgeführt werden.
- Projektion: Der Benutzer hat die Möglichkeit, die Ergebnisdatensätze auf bestimmte Attribute einzuschränken.
- Die Aufbereitung soll es ermöglichen, die schon selektierte Ergebnismenge statistisch aufzubereiten, z.B. durch Summen- oder Mittelwertbildung
- Das Ergebnis einer Selektion kann immer als Tabelle dargestellt und abhängig von der Ergebnismenge grafisch aufbereitet werden (Übergabe an GIS-/Grafikkomponente). Zusätzlich soll eine Schnittstelle zu EXCEL existieren. Eine Archivierung von Ergebnissen und Anfragen soll einen Zugriff auf diese Daten zu einem späteren Zeitpunkt ermöglichen.

Vom UDK oder von der GIS-/Grafikkomponente zur Sachdatenkomponente:

- Sowohl die Selektoren als auch die Selektorkriterien sind - mit und ohne Dialog - parametrisiert aufrufbar. Die betroffenen Kriterien werden dann voreingestellt.

Von der Sachdatenkomponente zur GIS-/Grafikkomponente:

- Da alle Sachdaten eine räumliche Komponente haben, muß zur Suche von Sachdaten auch ein Kartendienst genutzt werden können, der von der GIS-/Grafikkomponente zur Verfügung gestellt werden muß. Der Kartendienst muß die graphische Auswahl von Verwaltungseinheiten (Kreisen, Regierungsbezirken, Landkreisen, etc.) und Bounding Boxen ermöglichen, die dann der Sachdatenkomponente mitgeteilt werden. Diese füllt dann entsprechend die Kriterien aus. (Die bereits im Selektor ausgefüllten Eingaben werden durchgereicht.)
- Die Ergebnisse eines Selektors sollen grafisch dargestellt werden können. Diese Aufgabe übernimmt die GIS-/Grafikkomponente.

2.4 Funktionale Anforderungen an die GIS-/Grafikkomponente

Der RIPS-Pool enthält geographische Daten mit Umweltbezug für Baden-Württemberg in verschiedenen Maßstäben. Zum Einsatz kommen verschiedene GIS-Systeme, u.a. Small-world-GIS und Arc/Info. Die geographischen Daten müssen deshalb teilweise aus unterschiedlichen Datenquellen entnommen werden. Die Suchen auf diesen Datenbeständen mit räumlichen Daten (z.B. Shapefiles) werden Geoselektoren genannt.

Funktionalität der GIS-/Grafikkomponente

- Als Basis für den Einstieg soll die Karte Baden-Württembergs dienen.
- Ein Übersichtsfenster zeigt Baden-Württemberg und den im Kartenfenster dargestellten aktuellen Kartenausschnitt.
- Die GIS-/Grafikkomponente kann auf Daten in RIPS und in Shapefiles zugreifen.
- Der Benutzer kann in bestimmte Bereiche herein- und herauszoomen.
- Objekte sollen mit Hilfe einer Bounding Box selektiert werden können.
- Im Kartenfenster sollen mehrere Layer gleichzeitig dargestellt werden können. Dies ermöglicht etwa das Übereinanderlegen verschiedener Karten.
- Eine Einfärbung von Objekten soll entsprechend einer Klassifikation von Sachdaten/Attributen erfolgen können.
- Die Diagrammdarstellung soll in Karte und ohne Karte möglich sein.
- Ein Plot des GIS-Grafikkomponenten-Fensters (z. B. als Postscript) soll möglich sein.

Von der Sachdatenkomponente oder dem UDK zur GIS-/Grafikkomponente

- Objekte aus der Sachdatenkomponente sollen übernommen und dargestellt werden können. Es soll sowohl der Raumbezug visualisiert werden als auch eine grafische Darstellung von Ergebnissen, z.B. Meßwerten, in der Form von Business-Grafiken möglich sein.
- Der Raumbezug von UDK-Objekte soll visualisiert werden können.
- Karten, die im UDK beschrieben werden, sollen dargestellt werden können, insofern es sich um Shapefiles oder Karten im RIPS-Pool handelt.

Von der GIS-/Grafikkomponente zur der Sachdatenkomponente oder zum UDK

- Festlegung eines Gebiets: Anzeige aller in diesem Gebiet vorhandenen Objekte (UDK-Objekte oder Objekte eines Selektors), danach direkte Möglichkeit, zu einem Objekt (z.B. Meßstelle) unter anderem Sachdaten (z.B. Aufruf des Meßstellenselektors), zugeordnete Bilder, UDK-Informationen etc. zu erhalten.
- Metainformationen über selektierte Objekte sollen aus dem UDK abgefragt werden können.
- Selektionsdienste der Sachdatenkomponente sollen aus der Karte heraus gestartet werden können.

2.5 Zusammenspiel der Komponenten

In der Abb. 2.1 ist das Zusammenspiel der einzelnen Komponenten dargestellt.

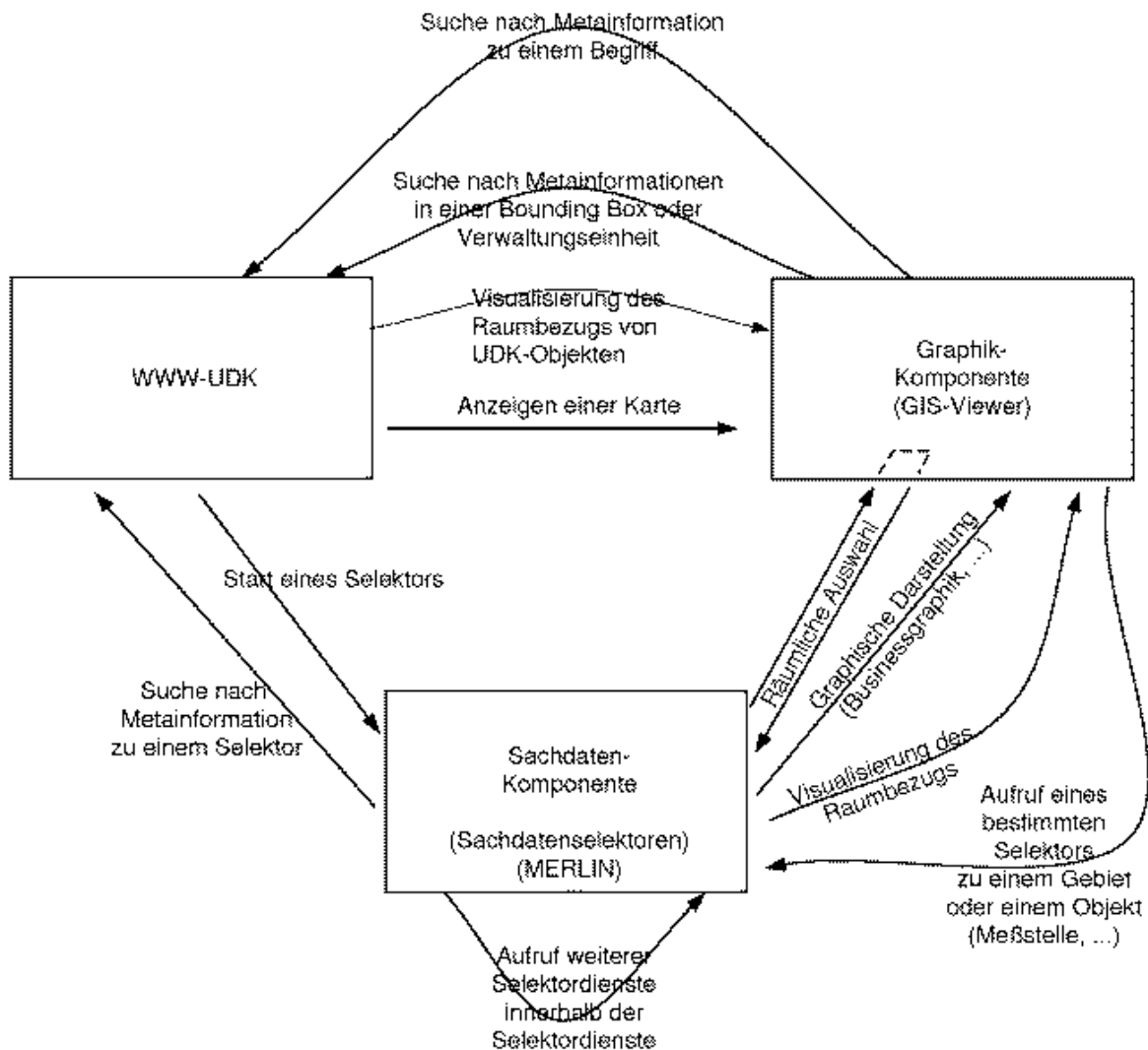


Abb. 2.1: Zusammenspiel von Metadaten-, Sachdaten und GIS-/Grafikkomponente

2.6 Status der Komponenten

Die drei Komponenten wiesen unterschiedliche Entwicklungsstadien zu Beginn des Projekts auf. Dies hatte natürlich großen Einfluß auf die Entwicklung des Gesamtsystems, denn Komponenten in fortgeschrittenem Entwicklungsstadium können nicht mehr beliebig verändert werden. Die Integration dieser Komponenten stellt eine zusätzliche Herausforderung dar.

Im einzelnen lagen folgende Entwicklungsstadien bei den Komponenten vor:

- Der WWW-UDK ist vollständig realisiert und hat, wie die Einsätze in mehreren deutschen Bundesländern und Österreich zeigen, Produktionscharakter.
- Sachdatenkomponente: Es gibt eine prototypische Realisierung für Meßnetze aus der vorherigen Merlinphase. Dieser in Java geschriebenen Prototyp wurde untersucht. Er

stellte sich heraus, daß er für die gegebene umfassendere Aufgabenstellung nicht als Ausgangsbasis geeignet ist. Konzepte und Lösungsansätze wurden übernommen.

- GIS-/Grafikkomponente: Der Graphikkern, der als Basis für die graphische Darstellung und Interaktionsverwaltung dient, war in einer prototypischen Form bereits realisiert. Dieser mußte um eine Layerverwaltung ergänzt werden. Datenselektoren und Anwendungssemantik mußten neu implementiert werden. Serverseitig war die Konzeption und Basisimplementierung der Abstraktionsschicht und Adaptoren für einzelne GIS-Systeme zu realisieren.

3. Grundlagen

3.1 Problemstellung

Um ein komplexes System wie dieses zu realisieren, werden die unterschiedlichen Aufgaben verschiedenen Komponenten zugeordnet. Für die Realisierung des Gesamtsystems ist die Interaktion dieser Komponenten wesentliche Voraussetzung. Innerhalb dieses Kapitels sollen die verschiedenen Techniken, mit denen eine Kommunikation zwischen Komponenten,

1. die innerhalb einer Java Virtual Machine (JVM) eines WWW-Browser ausgeführt werden,
2. die auf einem oder mehreren Servern ausgeführt werden oder
3. die auf Client und Server verteilt sind

realisiert werden kann, vorgestellt und bewertet werden.

3.2 Ausgangsbasis

Bereits existierendes System ist der WWW-UDK. Er ist auf Serverseite als CGI-Skripten in der Programmiersprache C mit eingebetteten SQL-Anweisungen realisiert. Diese C-Programme generieren HTML-Seiten, die vom WWW-Server an den WWW-Browser geschickt werden, der sie dem Benutzer präsentiert.

Weitere Systeme sind die Graphikkomponente sowie die Sachdatenkomponente. Beide werden in Java realisiert und bestehen sowohl aus clientseitigen als auch aus serverseitigen Komponenten.

Prinzipiell müssen all diese Komponenten miteinander kommunizieren können. Die Kommunikationsmöglichkeiten mit WWW-UDK werden in Kapitel 6 beschrieben. Deshalb werden im folgenden nur die möglichen Schnittstellen zwischen Java-Komponenten erläutert.

3.3 Kommunikation auf Clientseite: JavaBeans

JavaBeans sind Teil von JDK1.1. Sie stellen das Komponentenmodell für die Programmiersprache Java dar und ermöglichen es, wiederverwendbare, allgemeine Softwarekomponenten zu erstellen, die je nach Einsatzgebiet entsprechend angepaßt (spezialisiert) werden können. JavaBeans können graphische Oberflächenelemente sein, aber auch nicht-sichtbare Elemente oder auch komplette Anwendungen. Die Modularisierung der Anwendung in JavaBeans (also die Granularität der JavaBeans) ist dem Entwickler überlassen.

Aus der Sicht des Entwicklers, sind JavaBeans gewöhnliche Java-Klassen, die sich allerdings an gewisse Konventionen der Ereignisbehandlung halten müssen. Eine JavaBean muß ihre Attribute (Properties), ihre Methoden und ihre Events der Öffentlichkeit zur Verfügung stellen können.

Die wichtigsten Konzepte von JavaBeans sind

- Introspektion: Erlaubt, herauszufinden, welche Eigenschaften, Ereignisse und Methoden von einer JavaBean unterstützt werden.
- Spezialisierung: Anpassung einer JavaBean an das Einsatzgebiet.
- Packaging: Um Beans in einer Beanbox benutzen zu können, muß es zusammen mit benötigten Klassen und Ressourcen in eine JAR-Datei gepackt und darin als Bean gekennzeichnet werden.
- Aggregation: Bildung neuer JavaBeans durch Zusammensetzung.

JavaBeans kommunizieren über Ereignisse. JavaBeans, die Ereignisinteressenten darstellen, müssen sich bei den als Ereignisquellen fungierenden JavaBeans zunächst registrieren und ihr Interesse für ein oder mehrere Ereignisse bekunden. Tritt ein Ereignis ein, dann wird eine entsprechende Methode der Ereignisinteressenten aufgerufen.

Ereignisse können unicast (d.h. ein Interessent eines Ereignisses) oder multicast (d.h. mehrere Interessenten eines Ereignisses) sein und sind immer streng typisiert.

Die Kommunikation zwischen verschiedenen JavaBeans erfolgt über dieselbe JVM. Das Prinzip der Ereignisübermittlung ist dazu da, bestimmte Zustandsänderungen der JavaBean dem Zielobjekt mitzuteilen. Ereignisse sind allerdings nicht an ein Attribut gebunden, sondern es kann z.B. auch ein Timer ein Ereignis auslösen.

Ereignisse werden durch den Aufruf von Methoden des Ereignisinteressenten übermittelt. Für jedes einzelne Ereignis sollte in der Regel deshalb auch genau eine Methode im Zielobjekt implementiert werden. Besteht ein Ereignis allerdings aus mehreren zusammengehörigen und nicht trennbaren Unterereignissen, können diese auch in Ereignisgruppen zusammengefaßt werden.

Eine Erweiterung der JavaBeans Spezifikation stellt das InfoBus API dar, der noch nicht Bestandteil von JDK1.1 ist. Es ermöglicht das Teilen und Austauschen von dynamischen Daten. Dabei werden die an der Kommunikation beteiligten JavaBeans nach Datengeber und Datennehmer unterschieden. Datengeber sind JavaBeans, dessen primäre Funktion darin besteht, Daten aus einem Datenspeicher zu holen und sie auf dem InfoBus zur Verfügung zu stellen.

Datennehmer zeichnen sich dadurch aus, daß sie Daten auf dem InfoBus suchen und diese analysieren oder visualisieren. Der InfoBus ermöglicht also eine Kommunikation von Applets oder Beans mit dynamischem Datenaustausch. Die Kommunikation ist asynchron und abhängig vom Inhalt der Daten. Auch hier findet die Kommunikation innerhalb einer JVM statt.

In der nächsten Version der Spezifikation des JavaBeans Komponentenmodells (auch Glasgow Specification genannt) wird die aktuelle Spezifikation unter anderem ergänzt durch ein Aggregationsmodell zum Schachteln von JavaBeans und durch ein Protokoll zum Zusammenstellen von Komponenten und Erfragen der einzelnen Dienste zur Laufzeit.

Durch JavaBeans ist es bereits derzeit möglich, modulare Java-Anwendungen zu erstellen, dessen Komponenten wiederverwendbar sind. Über Ereignisse können JavaBeans innerhalb einer JVM kommunizieren. Obwohl JavaBeans ein sehr neues Konzept darstellen und es noch wenig Programmiererfahrungen gibt, stellen sie ein vielversprechendes Komponentenmodell für Java dar und ermöglichen eine einfache Realisierung der Kommunikation innerhalb einer JVM eines Browsers.

Weiterer Vorteil von JavaBeans-Architekturen ist, daß damit andere Komponenten-Architekturen angepasst werden, etwa ActiveX, Live Connect und OpenDoc. JavaBeans können also auch in diesen und anderen Komponenten-Architekturen verwendet werden.

3.4 Kommunikation innerhalb der verteilten Umgebung

Da nicht nur auf Clientseite parallele Prozesse laufen sollen, sondern auch auf Serverseite Prozesse mit unterschiedlichen Adreßräumen auf evtl. unterschiedlichen Rechnern miteinander kommunizieren sollen, wird auch hier ein Kommunikationsmechanismus benötigt.

Folgende zwei Mechanismen sind in engerer Wahl. Es ist möglich, beide Mechanismen zu kombinieren. Prinzipiell sind beide sowohl für die Kommunikation auf dem Server als auch zwischen Client und Server geeignet.

- RMI (Remote Method Invocation) wurde speziell für verteilte Java-Anwendungen entwickelt. Es ist Teil von JDK1.1 und erlaubt die Kommunikation über verschiedene JVMs hinweg. Unter Verwendung von RMI können JavaApplets (auf Clientseite) mit Java Applikations (auf Serverseite) kommunizieren.
- CORBA (Common Object Request Broker Architecture) ist ein Produkt der Object Management Group (OMG) und die Middleware für die Kommunikation zwischen verteilten Objekten. Die an der Kommunikation beteiligten Systeme können in C, C++, Java oder Ada geschrieben sein. Seit CORBA 2.0 ist auch ein Protokoll für die Kommunikation zwischen ORBs unterschiedlicher Hersteller spezifiziert, das IIOP (Internet Inter-ORB Protocol). Ein ORB auf Clientseite, der Bestandteil einiger WWW-Browser oder des JavaApplets ist, kann unter Verwendung von IIOP mit einem ORB auf Serverseite kommunizieren. Serverseitig agieren also JavaApplikations oder Programme in anderen Sprachen als CORBA-Objekte.

Für genauere Ausführungen der beiden Technologien sei auf das Grobkonzept des WWW-Szenarios verwiesen.

3.5 Vergleich RMI und IIOP

Einer der wichtigen Unterschiede für dieses Projekt zwischen RMI und IIOP ist die Tatsache, daß der Einsatz von RMI Java auch auf Serverseite (oder zumindest Java Wrapper, die über sogenannte native calls C -oder C++-Funktionen aufrufen) voraussetzt, währenddessen IIOP und CORBA Serverprogramme in Java, C, C++ oder Ada erlaubt. Durch CORBA und IIOP ist es möglich, bestehende Systeme in diese Projektumgebung aufzunehmen und sie zu benutzen.

Umgekehrt gilt dabei auch, daß andere Programme (z.B. aus WAABIS) über denselben ORB oder von einem anderen ORB aus, auf die in diesem Projekt realisierte Software zugreifen können. Es wird also eine höhere Flexibilität durch CORBA erreicht und die Programme sind auf Serverseite nicht auf die Programmiersprache Java angewiesen.

Desweiteren erreicht man durch CORBA Ortstransparenz, die RMI nicht vorweisen kann.

Ausführlich werden CORBA und RMI in /3/ verglichen.

3.6 Aufteilung der Software-Komponenten auf Client und Server

Die Aufteilung der Software-Komponenten auf Client und Server muß unter den folgenden zwei Gesichtspunkten betrachtet werden:

- Einerseits sollte der Client nicht zu sehr belastet werden.
- Andererseits sollte nicht zu viel Kommunikation zwischen dem Client und dem Server (Stichwort Ping-Pong Kommunikation) nötig sein, da die Bandbreite des Internets sonst zu große Verzögerungen bewirken würde und die gesamte Anwendung für einen Benutzer nicht zumutbar wäre.

Allgemein läßt sich deshalb folgende Aufteilung machen:

- auf dem Client: aktuelle Ergebnismenge, Berechnungen
- auf dem Server: Anwendungen auf Datenbasen, große Ergebnismengen

3.7 Zugriff auf die Datenbanken

Aus den JavaApplets und JavaApplikationen wird über JDBC auf die Datenbanken zugegriffen. JDBC ermöglicht den Zugriff auf relationale Datenbanken aus Java-Programmen heraus. Es ist ein Datenbankspezifischer JDBC-Treiber oder eine JDBC-ODBC-Bridge und ein ODBC-Treiber erforderlich. Seit JDK 1.1 ist die JDBC-Schnittstelle integraler Bestandteil der Basisklassenbibliothek.

4. Sachdatenkomponente

4.1 Übersicht

Die Hauptmodule der Sachdatenkomponente in Verbindung mit den Aufrufbeziehungen (auch zu den anderen beiden Systemen) sind in Abb. 4.1 dargestellt. Umgesetzt wurde das Konzept mit Hilfe der Technik der JavaBeans (siehe Abschnitt 3.3).

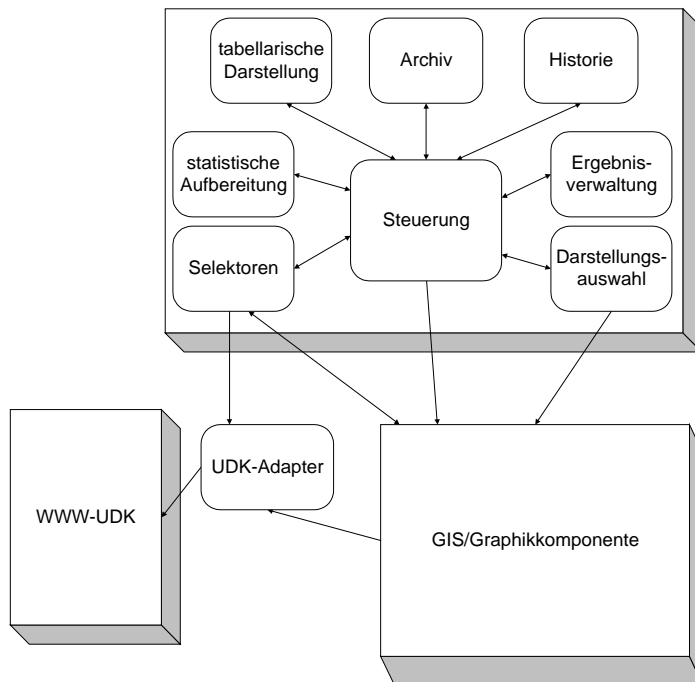


Abbildung 4.1 : Modulübersicht mit Schwerpunkt Sachdatenkomponente

Im folgenden werden die einzelnen Module genauer beschrieben.

4.2 Steuerung

Die *Steuerung* ist das Kernmodul der Sachdatenkomponente. Sie kennt nicht nur alle anderen Module, sondern regelt hauptsächlich den Kontroll- und den Datenfluß.

An der Oberfläche präsentiert sich die *Steuerung* durch die Anzeige des aktuellen Datenflusses, die nach jedem Verarbeitungsschritt (Datensuche, statistische Aufbereitung, graphische Darstellung) den aktuellen Stand und weitere mögliche Schritte anzeigt bzw. anbietet.

Die Steuerung ist auf Clientseite positioniert.

4.3 Selektoren

Wie in Abbildung 4.2 dargestellt, besteht das *Selektorenmodul* aus mehreren Untermodulen, die auf Client und Server aufgeteilt sind.

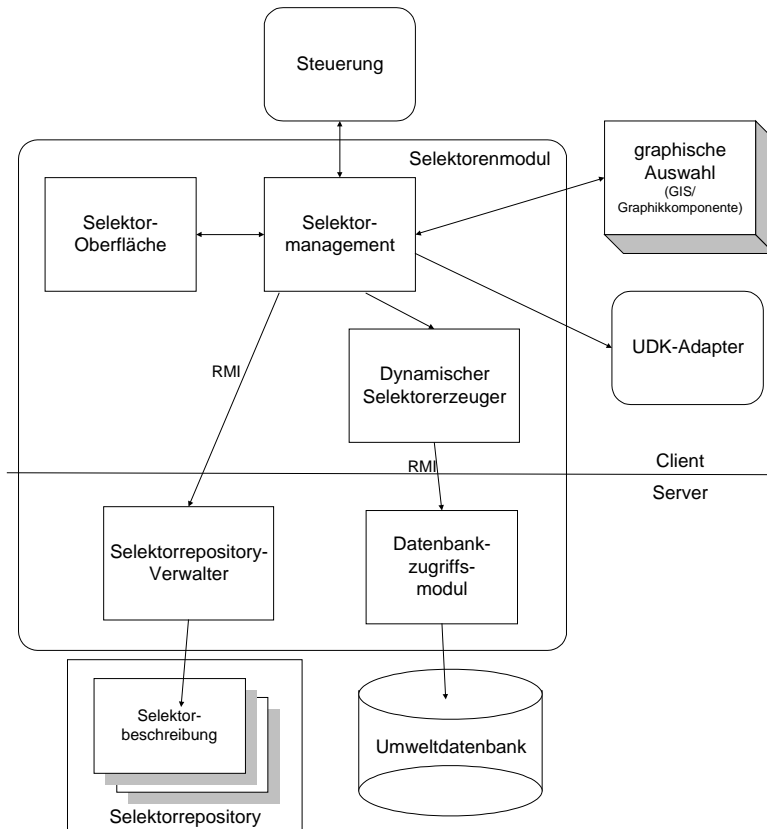


Abbildung 4.2: Architektur des Selektorenmoduls

Im folgenden werden die einzelnen Module genauer erklärt:

- Selektorrepository* und *Selektorrepository-Verwalter*
 Im *Selektorrepository* sind alle Informationen über die verschiedenen Selektoren abgelegt. Dies beginnt mit den Eingabe- und Ausgabeparametern, enthält aber auch die Abhängigkeiten zwischen den Selektoren zwecks Selektorchaining (Verknüpfung von Selektoren zur aufeinanderfolgenden Ausführung) und auch Informationen über dynamische Kriterien, die je nach Eingabe für die anderen Kriterien automatisch zusätzlich parametrisiert werden können. Neben den direkt für die Datenbankselektion benötigten Informationen sind dort weitere selektorspezifische Daten abgelegt, die für die Auswahl sinnvoller Darstellungsarten vom *Darstellungsauswahlmodul* (siehe Abschnitt 4.6) gebraucht werden. Beispielsweise ist hierunter eine Klassifikation der Ergebnisattribute eines Selektors zu verstehen, die der Algorithmus zur Berechnung der möglichen Darstellungsarten benötigt, um die richtigen Entscheidungen treffen zu können.
 Das *Selektorrepository* ist auf dem Server zu finden. Die eigentlichen Daten sind in Dateien abgelegt, so daß der Administrator der gesamten Software neue Selektoren hinzufügen kann, ohne den Programmcode ändern zu müssen. Der *Selektorrepository-Verwalter* ist über RMI ansprechbar. Er liest bei entsprechendem Aufruf die In-

formationen zu einem Selektor aus den Dateien oder aus seinem Cache, wenn sie schon gelesen wurden.

- Dynamischer Selektorerzeuger und Datenbankzugriffsmodul*
 Mit den Angaben aus dem *Selektorrepository* werden dynamisch die SQL-Anfragen für die Selektoren zusammengestellt. Wenn eine kontextinsensitive Anfrage durchgeführt wird, dann werden einmal zu Beginn alle Auswahllisten für die verschiedenen Kriterien gefüllt. Bei kontextsensitiven Anfragen werden die zur Auswahl stehenden Werte dynamisch in Abhängigkeit von den bereits parametrisierten Kriterien erstellt. Wenn die Selektoreingabe fertig ist, wird die SQL-Anfrage des Selektors ermittelt und mit Hilfe des *Datenbankzugriffsmoduls* durchgeführt. Dieses ist über RMI ansprechbar. Technisch gesehen könnte der dynamische Selektorerzeuger auch direkt über JDBC (Java Database Connectivity) die Datenbankanfrage an das DBMS mit Umweltdaten stellen. Aus sicherheitstechnischen Gründen werden aber die Paßworte für die Datenbanken nicht über das Netz übertragen, sondern werden im *Datenbankzugriffsmodul* zusammen mit dem Datenbanknamen gehalten, um sie zu einer übergebenen SQL-Anfrage mit Datenbankname hinzufügen zu können. Das Ergebnis der SQL-Anfrage wird dem *dynamischen Selektorerzeuger* übertragen, der zusammen mit einigen Informationen aus dem *Selektorrepository* die Daten in eine weiterverarbeitbare Struktur bringt und Metainformationen dazu erzeugt. Diese Daten werden an das *Selektormanagement* übergeben, der dies dann an das allgemeine *Steuerungsmodul* (siehe Abschnitt 4.2) weitergibt.
- Selektormanagement und Selektoroberfläche*
 Die *Selektoroberfläche* ist für die Interaktion mit dem Benutzer zuständig und leitet alle Anfragen an das *Selektormanagement* weiter, das dann entsprechende Aktionen veranlaßt. Die Oberfläche besteht im wesentlichen nur aus einer Benutzermaske, an der die entsprechenden Einstellungen für die Selektoren durchführbar und inhaltliche Informationen abrufbar sind. Das *Selektormanagement* ist für die Durchführung von Selektoren verantwortlich. Es nimmt den Selektornamen und evtl. parameterisierte Kriterien entgegen, holt dann die entsprechenden Informationen aus dem *Selektorrepository* und leitet diese dann an den *dynamischen Selektorerzeuger* weiter. Dieser holt die Auswahllisten und am Ende auch das Selektorergebnis aus der Datenbank und übergibt es dem *Selektormanagement*, welches die Auswahllisten mit Hilfe der *Selektoroberfläche* darstellt und die Auswahl des Benutzers an den dynamisch erzeugten Selektor zurückgibt. Die Selektorergebnisse übergibt es an die allgemeine *Steuerung*. Will der Benutzer das Ergebnis eines Selektors als Eingabe für einen weiteren Selektor benutzen, dann wendet sich das *Selektormanagement* an die *Selektorrepository-Verwaltung*, um die möglichen Anschlußselektoren zu erfragen. Es ist also eine intensive Kommunikation mit dem *Selektorrepository-Verwalter* und dem *dynamischen Selektorerzeuger* notwendig. Wenn der Benutzer (innerhalb der Selektoroberfläche) veranlaßt, daß Informationen zum Selektor zu Selektorkriterien oder zu Selektorkriterienwerten aus dem UDK geholt werden sollen, dann wendet sich das *Selektormanagement* an den UDK-Adapter, der die entsprechende Anfrage weiterleitet. Für den Fall, daß ein Benutzer für räumliche Kriterien eines Selektors die Werte mit Hilfe der GIS/Graphikkomponente festlegt, werden die Daten von der GIS/Graphikkomponente asynchron an das *Selektormanagement* weitergeleitet. Umgekehrt ruft die GIS/Graphikkomponente auch das *Selektormanagement* auf, wenn es mit der Eingabe eines ausgewählten Gebiets oder einer anderen räumlichen Einheit (z.B. Meßstelle) einen Selektor initiieren will.

4.4 UDK-Adapter

Der *UDK-Adapter* stellt allgemeine Primitive zum Aufruf der WWW-UDK Software zur Verfügung. Er unterstützt sowohl den Aufruf mit WWW-UDK- als auch ohne WWW-UDK-Dialog. Im zweiten Fall werden auch die Ergebnisse der UDK-Suchen entsprechend an den Aufrufer eines UDK-Adapter Primitivs übergeben.

Für genauere Ausführungen bzgl. der Realisierung dazu sei auf das Kapitel 5 verwiesen.

4.5 Statistische Aufbereitung

Das *Aufbereitungsmanagement* steht im Zentrum des Moduls zur statistischen Aufbereitung. Es organisiert den Datenfluß und Kontrollfluß zwischen dem Modul mit den Aufbereitungsregeln und der Berechnung der statistischen Aufbereitung. Abhängig vom Selektor und dem erzielten Ergebnis des Selektors werden vom *Aufbereitungsregelmodul* die möglichen Aufbereitungen bestimmt und dem Benutzer zur Auswahl gestellt. Dieser wählt davon eine aus und läßt diese durch das Modul zur *Aufbereitungsberechnung* berechnen.

Das Aufbereitungsregelmodul trifft die Entscheidung über sinnvolle Aufbereitungen für ein Selektorergebnis, indem es die Aufbereitungsregeln evaluiert. Je nach Selektor und erzieltm Ergebnis sind nur bestimmte Aufbereitungarten sinnvoll. Das Wissen darüber ist in den Aufbereitungsregeln abgelegt.

Das Modul zur Aufbereitungsberechnung führt die eigentlichen Berechnungen durch. Als Eingabe benötigt sie die Aufbereitungsart, die ausgewählten Aufbereitungsparameter, das Selektorergebnis und die dazugehörigen Metadaten. Das Ergebnis der statistischen Aufbereitung besteht aus den Ergebnisdaten und entsprechenden Metadaten.

4.6 Darstellungsauswahl

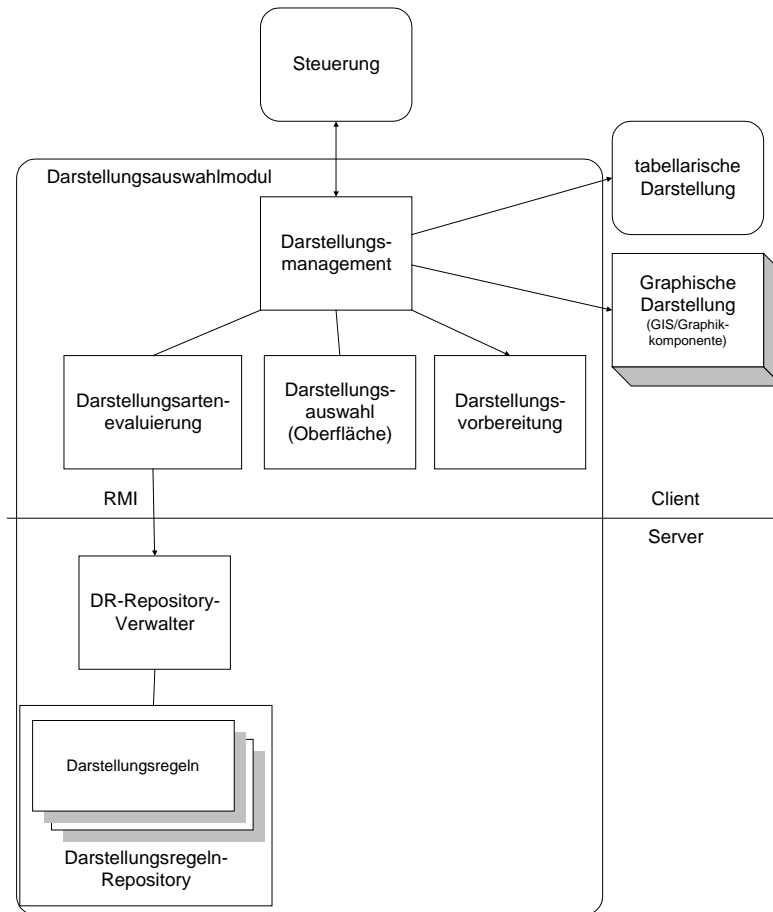


Abbildung 4.3: Architektur des Darstellungsauswahlmoduls

Auch in diesem Modul gibt es ein Untermodul, das für den Kontroll- und Datenfluß zwischen den Untermodulen zuständig ist: das Darstellungsmanagement. Es kommuniziert mit der Darstellungsartenevaluierung, der Benutzeroberfläche, der Darstellungsvorbereitung und dem Modul für die eigentliche Darstellung als Tabelle oder in Form einer Graphik (siehe auch Abb. 4.3). Aufgerufen wird es durch die allgemeine Steuerung, die es danach auch regelmäßig über den aktuellen Status informiert.

Soll ein Ergebnis eines Selektors oder einer statistischen Aufbereitung dargestellt werden, dann wendet sich das Darstellungsmanagement an die Darstellungsevaluierung, die das erhaltene Ergebnis mit Hilfe der vom DR-Repository-Verwalter erhaltenen Darstellungsregeln auf mögliche Darstellungsarten überprüft. Die Regeln werden über RMI vom Server geholt, wo sie in Dateien abgelegt sind, um eine einfache Wartbarkeit zu erzielen. Die Regeln beschreiben, welche Voraussetzungen für welche Darstellungsarten gegeben sein müssen.

Das Ergebnis der Darstellungsartenevaluierung wird an das Oberflächenmodul weitergegeben, das nun dem Benutzer alle möglichen Darstellungsarten abhängig vom Darstellungsziel zur Auswahl stellt. Hat sich dieser für eine Darstellung entschieden, dann werden die anzu-

zeigenden Daten für die ausgewählte Darstellungsart mit Hilfe der Darstellungsvorbereitung in das erwartete Format umgewandelt.

Danach ruft das Darstellungsmanagement je nach Benutzerwahl die tabellarische Darstellung oder die graphische Darstellung mit den entsprechenden Eingaben auf.

Die meisten Module sind auf Clientseite angesiedelt. Nur das Repository und der Repositoryverwalter sind naturgemäß auf Serverseite.

4.7 Ergebnisverwaltung

Die Ergebnisverwaltung stellt Primitive zum Ablegen und Suchen von Selektorergebnissen zur Verfügung. Es werden sowohl die Sachdaten (Selektorergebnisse) als auch Metadaten zu den Ergebnissen gespeichert. Im Gegensatz zum Archiv werden hier die Ergebnisse nicht persistent gehalten, sondern nur bis zum Ende der Sitzung, in der sie erzeugt wurden, aufbewahrt.

4.8 Tabellarische Darstellung

Das Modul für die tabellarische Darstellung ermöglicht einerseits die Darstellung von Sachdatenergebnissen (Selektorergebnisse oder Ergebnisse einer statistischen Aufbereitung) in einer zum Gesamtsystem gehörigen Tabellendarstellung, an der sich der Benutzer die Ergebnisse nur anschauen aber nicht nach seinem Belieben weiterverarbeiten kann, andererseits aber auch den Export der Ergebnisse in ein Dateiformat, das von Excel gelesen werden kann.

Beide Arten der tabellarischen Darstellung werden in unterschiedlichen Modulen abgearbeitet. Die 1. Variante läuft auf ein clientseitiges Tabellendarstellungsmodul hinaus. Für die Realisierung der 2. Variante gerät man aus sicherheitstechnischen Gründen an die Grenzen der Realisierung von Anwendungen für das Internet mit Java.

In Java 1.1 ist zwar die Möglichkeit hinzugekommen, mittels Cut-and-Paste-Metapher Daten zwischen Anwendungen auszutauschen. Prinzipiell können mit diesem Mechanismus Daten sowohl innerhalb einer Java-Anwendung, zwischen Java-Anwendungen und zwischen einer Java-Anwendung und anderen Anwendungen über ein Clipboard ausgetauscht werden. Nicht-zertifizierte Applets gelten allerdings als "nicht vertrauenswürdig" und dürfen von der Möglichkeit des Zugriffs auf die Systemressource Clipboard aus Sicherheitsgründen nicht Gebrauch machen.

Solange nur eingeschränkt die Möglichkeit besteht, über Client-seitige Dateien oder direkt über das Clipboard mit Anwendungen zu kommunizieren, können nur auf dem Server Dateien in dem entsprechenden Format erzeugt werden. Diese Dateien können anschließend von einem Web-Browser geladen werden, der, wenn er entsprechend konfiguriert ist, die Daten automatisch an das zugehörige Werkzeug - z.B. Excel - weiterreicht. Der Aufruf einer URL, die auf die Datei im Excel-Format verweist, kann vom Java-Applet (auf dem Client) geschehen. Nachdem die Datei dann auf den Client geladen worden ist, kann sie serverseitig wieder gelöscht werden.

4.9 Historie

Das Historienmodul hat den Zweck, auf alle Aktionen (Selektion, statistische Aufbereitung, Darstellung), die seit Programmaufruf durchgeführt wurden, wieder zurückzugreifen, die alten Einstellungen anschauen und die Aktionen nochmals durchführen zu können. Alle Aktionen werden dokumentiert und in sogenannten Historienelementen gehalten. Auch ihre Verknüpfungen, die den Ablauf zusammenhängender Aktionen verkörpern, werden festgehalten.

Die *Historienverwaltung* stellt Primitive zum Erzeugen, Verknüpfen und Aufrufen von Historienelementen (Darstellungsobjekten, Selektorobjekten, Aufbereitungsobjekten) zur Verfügung. Sie präsentiert sich auch dem Benutzer in Form einer Oberfläche (Extra-Fenster).

4.10 Archiv

Im Archiv werden auf Verlangen eines Benutzers zusammenhängende Aktionen und auch Ergebnisse persistent abgelegt.

Das Archivmanagement stellt Primitive zum Aktivieren, Erzeugen und Verknüpfen von Archivelementen zur Verfügung. Es präsentiert sich dem Benutzer in Form einer Oberfläche (Extra-Fenster).

Zur Realisierung des Archivs gibt es verschiedene Möglichkeiten:

- **Clientseitiges Archiv**
Die Daten werden in einem Archiv auf Clientseite abgelegt. Dadurch entstehen keine großen Datenvolumen auf dem Server und jeder Benutzer ist für sein Archiv selbst verantwortlich.
Da der Zugriff auf Client-Ressourcen wie das Dateisystem für nicht-zertifizierte Java Applets aus Sicherheitsgründen nicht möglich ist, wird diese Alternative vorerst nicht weiterverfolgt.
- **Serverseitiges Archiv**
Die Daten werden vom Server (in Dateien oder einer Datenbank) verwaltet und können nach einer Authentifizierung dem entsprechenden Benutzer zugeordnet werden. Nachteilig wirkt sich aus, daß evtl. große Datenmengen anfallen, die auf einem zentralen Server verwaltet werden müssen. Zur Zeit stellt sich mit der serverseitigen Archivierung allerdings der einzig gangbare Weg dar. Mittelfristig sollte eine Option für eine Migration hin zu einer Client-seitigen Lösung offengehalten werden.

5. Umweltdatenkatalog

5.1 Aufbau des WWW-UDK

Die Basis für den Zugriff auf die Daten des UDK soll der WWW-UDK und seine Software sein. Der Aufbau des WWW-UDK ist in Abb. 5.1 zu sehen.

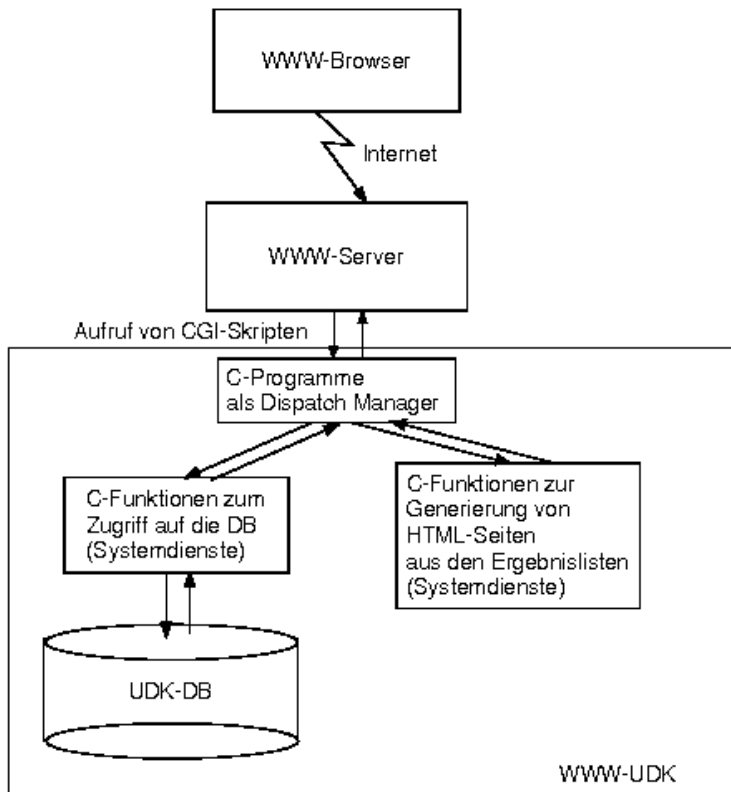


Abb. 5.1: Architektur des WWW-UDK

Der WWW-UDK ermöglicht verschiedene Suchen, die im folgenden aufgezählt werden.

Suchen nach Umweltobjekten:

- Allgemeine Suche (Eingabe eines Begriffs, nach dem in den verschiedensten Feldern der UDK-Objekte gesucht wird)
- Detailsuche (Eingabe von mehreren Suchattributen, die auf bestimmte Felder der UDK-Objekte abgebildet werden können)
- Schlagwortsuche (Eingabe mehrerer Schlagworte (evtl. aus Thesaurus), die im Feld für Schlagworte gesucht werden)
- Geosuche (Eingabe von Bounding-Box-Koordinaten)
- GDK-Themensuche (Hinabsteigen entlang der GDK-Hierarchie)

Suchen nach Adressen:

- Allgemeine Suche (Eingabe eines Begriffs -> Suche in den verschiedensten Feldern der Adreßrelation im UDK)

5.2 Zugriff auf den WWW-UDK

5.2.1 Zugriffsmöglichkeiten auf die UDK-Datenbank mittels WWW-UDK

Der WWW-UDK ist sehr modular aufgebaut. Da die Softwarekomponenten zur Datensuche in der UDK-Datenbank unabhängig von den Softwarekomponenten zur Generierung der HTML-Seiten aus den Ergebnissen sind, können sie auch separat aus anderen Systemen aufgerufen werden (z.B. aus der Sachdatenkomponente oder der GIS/Graphikkomponente).

Insgesamt existieren die folgenden Möglichkeiten, um von anderen Systemen aus mit Hilfe des WWW-UDK in der UDK-Datenbank zu suchen.

1. Möglichkeit (Starten einer URL)
 - Öffnen/Anspringen eines Browserfensters (Frames), darin Aufruf der URL des entsprechenden CGI-Programms mit den URL-codierten Parametern als Eingabe
 - Präsentation des Suchergebnis in diesem Browser-Fenster
2. Möglichkeit: (Serverprogramme in C, RMI)
 - Aufruf des C-Programms des WWW-UDKs direkt,
 - Präsentation des Suchergebnis z.B. im JavaApplet, das die Suche angestoßen hat.
3. Möglichkeit (Aufruf über CORBA-Schnittstelle, IIOP)
 - Aufruf des mit CORBA gewrappten C-Programms des WWW-UDKs
 - Präsentation des Suchergebnis z.B. im JavaApplet, das die Suche angestoßen hat

Bei der 1. Möglichkeit wird also die (HTML-) Oberfläche mit benutzt, während bei der 2. Möglichkeit und bei der 3. Möglichkeit intern die Suche der UDK-Module auf Serverseite durchgeführt wird. Die Unterschiedlichkeit der drei Möglichkeiten zeigt auch Abb. 6.2 .

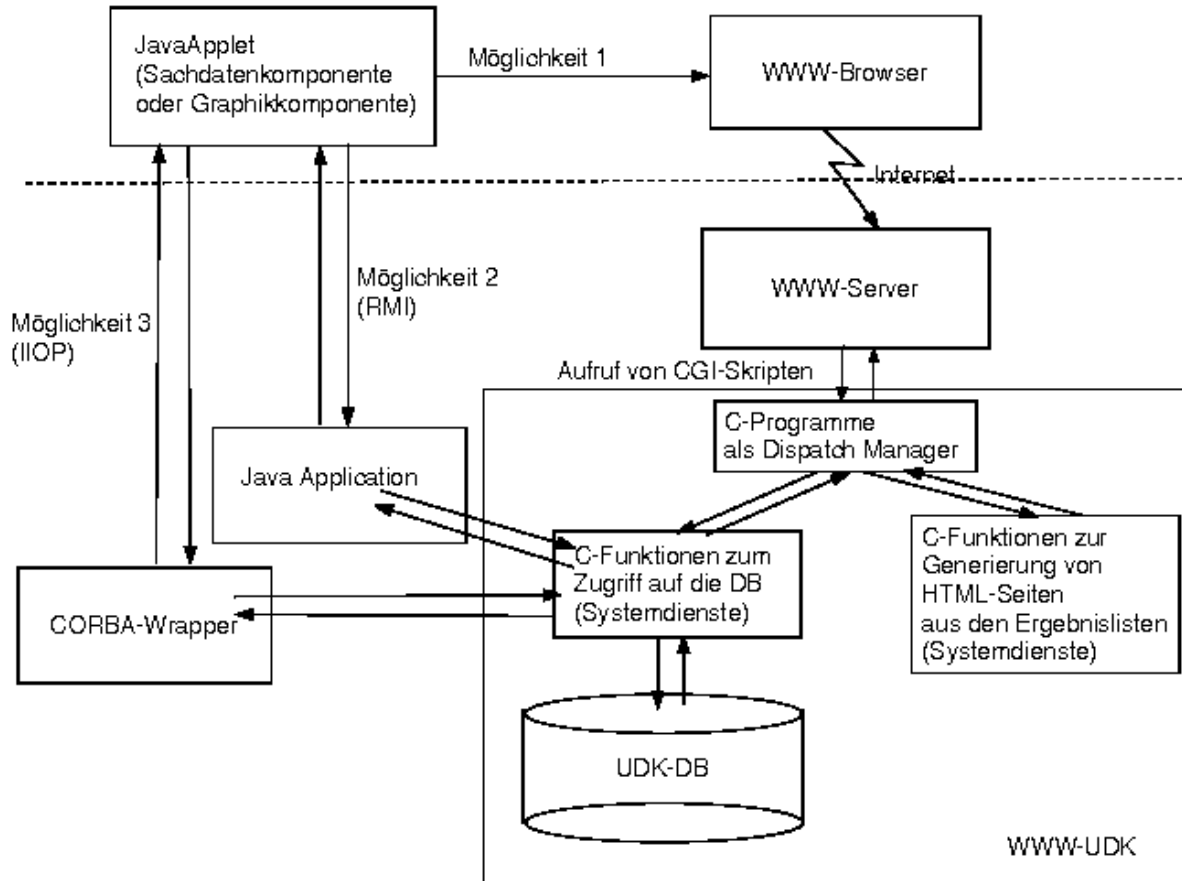


Abb. 5.2: Möglichkeiten zum Zugriff auf Daten der UDK-Datenbank mittels des WWW-UDK

5.2.2 Vergleich der Zugriffsmöglichkeiten auf den WWW-UDK

Die drei Zugriffsmöglichkeiten auf den WWW-UDK werden nun anhand von sechs Kriterien verglichen:

- **Realisierungsaufwand:** Der Realisierungsaufwand für die erste Lösung ist geringer als der Realisierungsaufwand für die zweite und dritte Lösung, da bei diesen weitere Software entwickelt werden muß. Diese ist entweder ein CORBA-Wrapper oder eine Java-Applikation, welche auf dem Rechner des C-Programms laufen muß. In jeder der beiden letzten Möglichkeiten wird das C-Programm zur Suche in der UDK-Datenbank aufgerufen und die gefundenen Ergebnisse dem JavaApplet übergeben, welches diese dann auch noch darstellen muß. Bei der ersten Lösung wird die Suche und die Präsentation der Ergebnisse an den WWW-UDK abgetreten. Es muß nur ein neues Browserfenster mit einer entsprechend parameterisierten URL zum Aufruf des CGI-Skripts gestartet werden. Vergleicht man die zweite und die dritte Lösung bzgl. des Realisierungsaufwands, dann unterscheiden sich die zu entwickelnden Softwarelösungen nur in der eingesetzten Programmiersprache und der Tatsache, daß für die CORBA-Lösung ein ORB installiert sein muß.
- **Verarbeitung der Ergebnisse:** Bei der zweiten und bei der dritten Möglichkeit können die Ergebnisse einer Suche im WWW-UDK noch weiter verarbeitet werden, dagegen ist bei der ersten Möglichkeit die Verbindung zum UDK nach dessen Aufruf beendet.

- **Benutzerfreundlichkeit:** Die Benutzerfreundlichkeit ist nicht objektiv wertbar. Die Bedienbarkeit einer Software wird von verschiedenen Benutzern oft unterschiedlich beurteilt. Bei der ersten Möglichkeit öffnet sich ein neues Fenster. Je nachdem wieviele andere Fenster schon offen sind, kann es leicht zu Unübersichtlichkeit führen. Auf der anderen Seite ist für den Benutzer klarer erkenntlich, daß der WWW-UDK ein unabhängiges System ist. Er kann nach der Ergebnisausgabe den WWW-UDK weiterbenutzen, während bei der zweiten und dritten Möglichkeit dem Benutzer nicht die Gelegenheit dazu gegeben wird.
- **Einheit der Gesamtanwendung:** Die Gesamtanwendung wirkt sicherlich einheitlicher, wenn auf den UDK mittels der Möglichkeit 2 oder Möglichkeit 3 zugegriffen wird.
- **Wartungsaufwand:** Sollte sich die URL für den Aufruf des WWW-UDK verändern, dann müssen in allen Anwendungen, die diese aufrufen (Möglichkeit 1), einzeln die Veränderungen nachgezogen werden. Bei den beiden weiteren Möglichkeiten müssen nicht nur Änderungen der Eingabeparameter nachgeführt werden, sondern auch Änderungen der Rückgabeparameter. Diese sind bei Änderungen des Datenmodells zu erwarten, insbesondere etwa nach der Umstellung auf das derzeit in Entwicklung befindliche Datenmodell UDK 4.0. Insgesamt ist also der Wartungsaufwand für Möglichkeit 1 am geringsten. Vergleicht man Möglichkeit 2 und 3 bzgl. des Wartungsaufwands, dann unterscheiden diese sich nur darin, daß bei der dritten Möglichkeit noch ein ORB gewartet werden muß. Dieser zusätzliche Aufwand ist allerdings sehr gering.
- **Wiederverwendbarkeit:** Die dritte Möglichkeit öffnet sicherlich die meisten Wiederverwendungsmöglichkeiten, da über CORBA jedes CORBA-fähige Programm (unabhängig von der Programmiersprache) den UDK-CORBA-Wrapper aufrufen kann. Die zweite Möglichkeit kann von allen Java-Programmen wiederverwendet werden.
- **Geschwindigkeit:** Große Geschwindigkeitsunterschiede sind nicht zu erwarten.

5.3 Aufruf von anderen Softwarekomponenten aus dem WWW-UDK

Im WWW-UDK sind die Sachdatenselektoren unter der Klasse Produkt abgelegt. Um vom WWW-UDK zu den Sachdatenselektoren zu navigieren, müssen entsprechende Suchbegriffe bei einer der Objektsuchen eingegeben werden. In der Ergebnisliste sind unter anderem dann die Sachdatenselektoren aufgelistet, die durch das Verfolgen eines Links (Verweises) direkt aufrufbar sind.

Für weitere Ausführungen bzgl. der Realisierung des Aufrufs von Softwarekomponenten aus dem WWW-UDK sei auf den Artikel „WWW-UDK“ in diesem Band verwiesen.

6. GIS/Graphik-Komponente

6.1 Gesamtarchitektur GIS-Terminal

In diesem Kapitel wird zunächst die Gesamtarchitektur des GIS-Terminal vorgestellt. Im Anschluß daran werden die einzelnen Schichten des Entwurfs nacheinander in detaillierter Form erläutert.

Das Konzept des GIS-Termial steht in der Tradition der Client-Server Architekturen, auch wenn der Begriff Terminal mit dem zentralistischen „Terminal-Host-Schemata“ assoziiert werden kann. Die Architektur verfolgt das Ziel, alle interaktiven und visuellen Teile eines Geografischen Informationssystem (GIS) zu vereinheitlichen und auf einen intelligenten Klienten zu verlagern. Zudem soll serverseitig eine Abstraktion und Integration unterschiedlicher Systeme verwirklicht werden. Dazu muß eine systemabstrahierende Schnittstelle definiert werden, die den vereinheitlichten Zugriff auf vorhandene Informationssysteme ermöglicht. Diese Schnittstelle muß gegenüber den darunter liegenden Systemen als Dienstnehmer auftreten und Differenzen zwischen den Systemarchitekturen, sofern diese für die Visualisierung und Aufbereitung der Daten nötig ist, ausgleichen. Ziel ist es dabei, möglichst viel Ressourcen der vorhandenen Systeme zu nutzen. Soll ein neues System zu der Gesamtarchitektur hinzugefügt werden, muß für das System ein Adapter, in Form eines Treibers implementiert werden. Dieser Treiber hat die Aufgabe, die Funktionalitäten, die in der abstrakten Schnittstelle definiert sind, auf das reale System abzubilden. An dieser Stelle soll nicht verschwiegen werden, daß diese Schnittstelle zunächst allgemeine und gebräuchliche Eigenschaften vereinheitlichen will. Spezielle Fähigkeiten eines bestimmten Systems können dabei im ersten Ansatz nicht abstrahiert und den oberen Schichten als Dienst zur Verfügung gestellt werden. Dies ist jedoch im allgemeinen Fall nicht als Nachteil zu werten, da die Zielgruppe des Systems im Allgemeinen kein Interesse an spezialisierten Funktionen besitzt. Der konzeptionelle Schwerpunkt liegt auf der breiten Verteilung von Geodaten. Der Benutzer soll in einer einheitlichen, transparenten Form das virtuelle GIS benutzen können, ohne Kenntnis über die darunterliegende Systeme und deren Struktur zu besitzen. Weiterhin sollen diese Dienste allen Informationsinteressenten flexibel und dynamisch über das Netzwerk zur Verfügung stehen.

Abbildung 6.1 verdeutlicht das Konzept und gliedert den Entwurf. Die Architektur ist Modular aufgebaut und besteht aus zwei Softwarekomponenten, die auf dem Client-Server Prinzip beruhen. Der Server (GIStermServer) besitzt gegenüber dem Klient (GISterm) die Serverrolle, gleichzeitig verhält er sich jedoch gegenüber den darunterliegenden Informationssystemen als Klient. Man kann in der Abbildung erkennen, daß das Konzept zwei Rechnergrenzen berücksichtigt. Alle beteiligten Systeme können räumlich verteilt, auf verschiedenen Rechner und Rechnerplattformen beheimatet sein. Der eigentliche Klient (GISterm) kann über das Inter-, bzw. Intranet an alle Rechner im Netzwerk dynamisch verteilt werden. Dazu muß der Klientrechner lediglich einen Netzanschluß sowie einen Java-fähigen WWW-Browser (z.B. Netscape) besitzen.

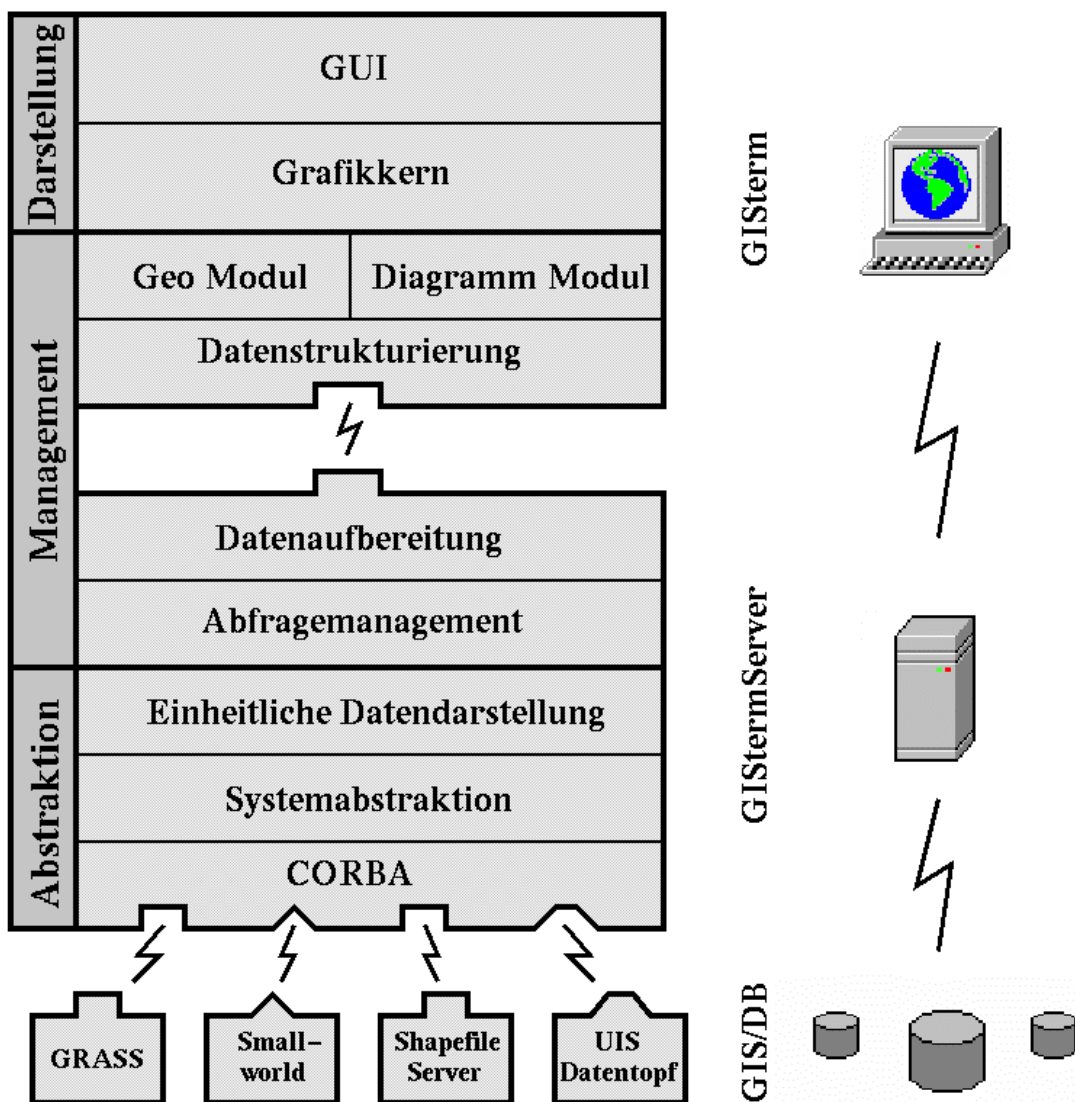


Abbildung 6.1: Gesamtarchitektur

Der Server (GIStermServer) wird auf dem gleichen Rechnersystem installiert, auf dem der für das Netz zuständige WWW-Server residiert. Dieser Server kann jetzt wiederum auf beliebige im Netz verteilte Informationssysteme zugreifen. Diese Flexibilität wird durch die Basistechnologien Java, Java/RMI und CORBA ermöglicht. Insgesamt wird durch dieses Konzept erreicht, daß die Information, die meist räumlich und organisatorisch verteilt ist, orthogonal und flexibel im Netz zur Verfügung gestellt werden kann.

Das Gesamtsystem des GIS-Terminal läßt sich in drei Hauptschichten zerlegen.

1. Darstellungsschicht
2. Managementschicht
3. Abstraktionsschicht

Dabei besteht die Serverkomponente (GIStermServer) aus der Abstraktionsschicht, sowie aus einem Teil der darauf aufbauenden Managementschicht. Der Klient (GISterm) setzt sich aus dem zweiten Teil der Managementschicht sowie der Darstellungsschicht zusammen. Die Managementschicht ist aus Realisierungsgründen aufgeteilt. Um unnötige Netzbelastungen zu

vermeiden, sowie die Performanz zu steigern, werden Daten in dieser Schicht auf dem Server für die Darstellung aufbereitet und strukturiert. Ziel ist es den Datenstrom so zu modellieren, daß die Netzwerkbelastung minimiert wird. Gleichzeitig soll der Klient die übertragenen Daten eigenverantwortlich halten und manipulieren können. Beispielsweise soll die Aufbereitung der Daten zu einer Präsentation fast ausschließlich auf dem Klienten ablaufen, was zu einer deutlichen Reduzierung der Netzbelastung und Serverauslastung führt.

In den folgenden Abschnitten sollen die einzelnen Schichten nacheinander näher vorgestellt werden. Dabei wird mit der untersten Schicht begonnen.

6.1.1 Die Abstraktionsschicht

Die unterste Schicht hat die Aufgabe, den darauf aufbauenden Schichten, den Dienst eines einheitlichen, virtuellen Informationssystem anzubieten. Als Basistechnologie dieser Schicht ist an dieser Stelle vor allem Java/RMI oder CORBA zu nennen. Da diese Technologien grundlegend in den wissenschaftlichen Berichten des GLOBUS Projekts zur Phase II erläutert wird, wird hier auf eine Einführung verzichtet. An dieser Stelle sollen nur die prinzipiellen Zusammenhänge verdeutlicht werden.

Die Abstraktionsschicht kann in drei Subschichten untergliedert werden, welche folgende Aufgaben besitzen.

1. Einheitliche Datendarstellung
2. Systemabstraktion
3. GIS-Operatoren unter CORBA oder Java/RMI

Die dritte und unterste Schicht realisiert den Zugriff auf im Netz verteilte Informationssysteme. Dazu werden GIS-Operatoren mittels CORBA oder Java/RMI implementiert, welche auf die realen Geoinformationssysteme zugreifen und Daten aus den Systemen selektieren und übermitteln. Weiterhin soll an dieser Stelle der Zugriff auf Datenfiles Mithilfe eines Fileserver ermöglicht werden. Dadurch ist es möglich einzelne Datendateien, die in der Praxis vor allem für den Datenaustausch benutzt werden, in das System zu integrieren. Im einzelnen soll der Zugriff auf

- Arc Shapefile-Dateien
- DXF-Dateien

realisiert werden. Weiterhin könnten alle Datenquellen die innerhalb des WWW-UIS realisiert wurden an dieser Stelle in das Gesamtkonzept aufgenommen werden. Im Rahmen der Java-AG wurden erste Planungen zu einem allgemeinen Datentopf bereits diskutiert.

Die auf der CORBA-Schicht aufbauende Systemabstraktionsschicht bietet die Operatoren der untersten Schicht in einer definierten Schnittstelle an. In dieser Schicht wird auch eine Zugriffsverteilung und Systemselektion realisiert. Ein weiteres Modul ist für die Migration der Geometrieformate der einzelnen Systeme in ein einfaches, topologiloses Format zuständig. Dieses generalisierte Geometrieformat soll im Hinblick auf die Aufbereitung und Darstellung der Geodaten definiert werden. Topologische und semantische Aspekte werden zunächst nicht unterstützt. Sobald sich eine einheitliche Geodatenbeschreibung etabliert hat, soll diese sich

an dieser Stelle integrieren lassen. Neben dem Geometrieformat muß diese Schicht ein einheitliches Format zur Darstellung von Tabellen (Relationen) definieren.

6.1.2 Die Managementschicht

Innerhalb der Managementschicht, sollen die von der Abstraktionsschicht zur Verfügung gestellten Daten verwaltet, für die graphische Darstellung aufbereitet und strukturiert werden. Wie bereits im Rahmen der Gesamtarchitektur angesprochen, ist diese Schicht auf den Klient (GISterm) als auch auf den Server (GIStermServer) aufgeteilt. Diese verteilte Architektur begründet sich vor allem durch das Bestreben die Netzwerkbelastung sowie den Speicherverbrauch des Klienten zu reduzieren. Der Umgang mit ressourcenintensiven Geodaten erfordert besondere Verwaltungsmechanismen, weshalb eine rein Klientenbasierende Verwaltung nicht praktikabel ist.

Geographische Daten lassen sich im allgemeinen in zwei Teilaspekte zerlegen, einen geometrischen Teil und einen Sachdatenteil. Diese Schicht verwaltet diese Datenteile getrennt, um die beiden genannten Ressourcen (Netzlast, Speicherbedarf) zu optimieren.

1. Geometrische Daten sollen auf dem Serverteil dieser Schicht für die Darstellung vorbereitet und in dieser optimierten Form in den Kliententeil übertragen werden. Der Klient (GISterm) kann dann ohne weitere Netzzugriffe die Daten für die Darstellung verwenden. Der Benutzer hat dadurch die Möglichkeit die Daten auf seinem Rechner interaktiv aufzubereiten und in eine präsentierbare Form zu bringen. Bei der Darstellungsvorbereitung auf dem Serverteil wird die Geometrie für den aktuellen Darstellungsmaßstab ausgedünnt und somit das zu übertragende Datenvolumen reduziert. Jedes Darstellungsobjekt, das zu dem Klienten übertragen wird, besitzt jedoch eine Referenz zu dem originalen Geometrieobjekt auf dem Serverteil. Über diese Referenz kann es jederzeit, etwa wenn der Darstellungsmaßstab geändert wurde, eine verbesserte Geometrie anfordern. Der Ausdünnungsalgorithmus kann über eine Deltatechnik noch nicht übertragene Teile aufbereiten, die zu dem Darstellungsobjekt übertragen werden.
2. Der Sachdatenteil wird zunächst nur auf dem Server (GIStermServer) vorgehalten, wobei die Verbindung zum Darstellungsobjekt ebenfalls erhalten bleibt. Erst wenn der Benutzer Sachattribute direkt anfordert, werden diese an den Klienten übermittelt und dort angezeigt.

Durch diese dynamische Aufteilung der Objekte wird sowohl die Netzbelastung, als auch die Speicherbelastung des Klienten minimiert. Damit können auch ressourcenschwache Klienten das GIS-Terminal nutzen.

Die Module der Managementschicht sind wie folgt unterteilt:

1. Das Geo-Modul sowie das Diagramm-Modul. Diese Module haben die Aufgabe Datenvisualisierungen interaktiv, oder Mithilfe von Vorgabewerten, zu erstellen. Dazu müssen Algorithmen entwickelt werden, die aus Eingabedaten generativ Graphiken erzeugen. Beispiele für die Aufgaben dieser Module sind Business-Diagramme oder Kartenlegenden.
2. Die Datenstrukturierung/Datenmanagement. Dieses Modul strukturiert und verwaltet die zu visualisierenden Daten. Dazu werden Mechanismen benötigt, mit denen die Daten zu semantischen Einheiten gruppiert werden können. Weiterhin ist dieses Modul für die Kommunikation zu dem Serverteil (GIStermServer) verantwortlich,

- wenn ein visualisiertes Datenobjekt auf seine Sachattribute zugreift, oder seine Geometrie aktualisieren will.
3. Die Datenaufbereitung. Dieses Modul hat die Aufgabe Ergebnisdaten (von Anfragen) zur Darstellung aufzubereiten. Dieses Modul ist für die Abbildung der Rohgeometrie, die aus einem GIS-System kommt, in einer Darstellungsgeometrie verantwortlich. Diese Abbildung ist nicht nur von der Geometrie abhängig, sondern die Anwendung und die Datensemantik spielt dabei eine große Rolle. Weshalb je nach Anwendungsfall mehrere Module implementiert werden müssen. Eine weitere Aufgabe des Moduls ist die Ausdünnung der erzeugten Darstellungsgeometrie zur Datenreduktion.
 4. Das Abfragemanagement. An dieser Stelle wird die Benutzerabfrage, die der Anwender über Eingabemasken interaktiv spezifiziert hat, in eine formale Abfrage umgewandelt und an das darunterliegende virtuelle Informationssystem übergeben. Wenn das System die gewünschten Daten bereit gestellt hat, werden diese an die übergeordnete Datenaufbereitung weitergeleitet.

6.1.3 Die Darstellungsschicht

Diese Schicht ist für die Grafikausgabe sowie die Benutzerinteraktion zuständig und gliedert sich in zwei Module.

1. GUI
2. Grafikkern

Beide basieren auf dem Abstract Window Toolkit (AWT), ein abstraktes, grafisches Fenstersystem das innerhalb der standardisierten Java-Klassenbibliothek definiert ist. Eine detaillierte Beschreibung der GUI-Komponenten würde den Rahmen dieses Berichtes sprengen. Für das GIS-Terminal werden eine Vielzahl von Dialogkomponenten benötigt, einige davon bietet das Toolkit direkt an, andere müssen durch Spezialisierung und Gruppierung von vorhandenen Komponenten erstellt werden. Das abstrakte Fenstersystem AWT bietet derzeit, aufgrund der Komplexität und der unterschiedlichen Architektur der abstrahierten Fenstersysteme, nur eine Untermenge der Möglichkeiten dieser darunterliegenden realen Systeme an. Der Leistungsumfang ist für die Implementierung einer interaktiven Benutzeroberfläche mehr als ausreichend. Für die Ausgabe von Grafik definiert die abstrakte Klasse Graphics die Grafikprimitive Linie, Rechteck, Kreis, Oval, Polygon, Bilder und Text zusammen mit einer einfachen Attributierung. Eine Implementierung dieser Grafikkategorie wird für die GUI-Komponente Canvas (eine gedächtnis- und aktivitätslose Zeichenfläche) angeboten. Für das Neuzeichnen von Bildteilen, sowie die Skalierung der Grafik ist das Anwenderprogramm selbst verantwortlich. Eine Möglichkeit der vom Ausgabegerät unabhängigen Grafikausgabe, etwa eine PostScript- oder HPGL-Ausgabedatei existiert standardmäßig bislang nicht.

Für das GIS-Terminal ist es unerlässlich, aufbauend auf die von Java gebotene Grundfunktionalität einen grafischen Kern zu implementieren. Erst durch einen solchen allgemeinen Grafikkern ist es möglich, komplexere Grafiken darzustellen, zu verwalten und zu modifizieren. Der hierzu zu entwickelnde Kern soll, angelehnt an grafische Standards wie GKS (Graphics Kernel System, ISO1985) und PHIGS (Programmer's Hierarchical Interactive Graphics System, ISO/IEC 9592:1989), Funktionalitäten wie

- Weltkoordinaten/Modellierungskordinaten,
- abstrakte Ausgabegeräte (PostScript, CGM, ...),

- semantische Eingabeoperationen (Pick, Select),
- Graphikprimitive als eigenverantwortliche Instanzen,
- Primitivhierarchie als Baum,
- keine Unterscheidung zwischen Primitiven und Aggregaten,
- Interaktion auf jeder Ebene der Primitivhierarchie,
- flexible Attributierung (lokal, vererbt, orthogonal),
- Layerverwaltung (Prioritätenverwaltung)

bieten. Das Design verfolgt das Ziel, konform zu dem objektorientierten Paradigma zu sein. Der Grafikkern soll ein hohes Abstraktionsniveau bieten, um komplexe Strukturen einfach handhaben zu können. Weiterhin soll die Integration von Semantik in die Grafikszenen ermöglicht werden. Innerhalb des Pakets `java.java2d` sind Erweiterung der Grafikfähigkeiten von Java angekündigt, deshalb setzt der Entwurf seinen Schwerpunkt vor allem in die Realisierung von höheren, abstrakten Funktionalitäten und umgeht derzeit bewußt eine Erweiterung der bislang schwach ausgeprägten Attributierung von Java. Sobald die Grafikerweiterungen vorliegen, können diese in die unteren Schichten des Kerns integriert werden.

6.2 Module und Komponenten des GISterm

Die nachfolgende Auflistung gibt eine Übersicht über die Module und Komponenten des GISterm die sich derzeit in der Realisierung befinden. Darüber hinaus müssen zur Erweiterung der Funktionalität und zur Gewährleistung einer vielseitigen Nutzbarkeit des GISterm weitere Komponenten entworfen und realisiert werden.

6.2.1 Grundlegende Module/Subsysteme

Die folgenden Module stellen Basisfunktionalitäten für das GISterm bereit:

6.2.1.1 Schemaverwaltung

Die Datenschemata (*FeatureSchema*) der angeschlossenen Systeme werden einheitlich in dem Subsystem Schemaverwaltung vorgehalten. Diese Schemabeschreibung wird für die Anfragegenerierung sowie für alle typisierten Operationen auf Ergebnisobjekten benötigt.

6.2.1.2 Ergebnisinstanzverwaltung (serverseitig)

Alle Ergebnisobjekte (*Feature*) einer Anfrage werden mit ihren Geometrie- und Sachattributen im Serverteil repräsentiert. Ausgehend von diesen Instanzen werden Darstellungsgeometrien erzeugt und an den Klienten übertragen. Der Zugriff auf Sachattribute vom Klienten erfolgt nach Bedarf. Das Subsystem Ergebnisinstanzverwaltung (*FeatureCollection*) verwaltet die Ergebnisobjekte und bietet auch Funktionen zum Zugriff auf die Instanzen.

6.2.1.3 Einheitliche Geometriedarstellung

Die Geometrie (*GeometricPrimitive*) der Ergebnisobjekte (*Feature*) einer Anfrage werden für alle angeschlossenen GIS-Systeme einheitlich im Serverteil repräsentiert. Das Subsystem

Geometriedarstellung stellt hierfür die nötigen Klassen bereit. Die Geometrie der einzelnen Feature-Objekte wird für die Abbildung auf eine Darstellungsgeometrie benötigt. Weiterhin dient sie als Basis für die Geometriereduktion/Generalisierung.

6.2.1.4 Systembeschreibung

Aufgabe dieses Subsystems ist die Metadatenbeschreibung der an das GIS-System angeschlossenen GIS. Die Modellierung dieses Subsystems ist noch nicht abgeschlossen.

6.2.1.5 Darstellungsabbildung

Aufgabe dieses Subsystems ist die Abbildung der Rohgeometrie, wie sie vom GIS-System geliefert wird, in eine Darstellungsgeometrie. Diese Abbildung ist anwendungsabhängig, so daß hier Konkretisierungen für Anwendungsgebiete realisiert werden müssen.

6.2.1.6 Geometriereduktion

Hauptaufgabe dieses Subsystems ist die dynamische Geometriedatenreduktion der Ergebnisobjekte (*Feature*) entsprechend des Darstellungsbereichs bzw. Darstellungsmaßstabs in der klientenseitigen Visualisierung.

6.2.1.7 Rasterdatenmanagement

Rasterdaten liegen häufig in sehr hoch aufgelöster Form vor. Das Modul Rasterdatenmanagement soll Verwaltungsmechanismen für eine gezielte Zerlegung der Rasterdaten bieten. Es bildet die Grundlage für eine klientenseitige Visualisierung von datenintensiven Rasterdaten.

Abfragemanagement

Dieses Subsystem (*QueryHandler*) realisiert die Anfragebearbeitung aufbauend auf die GIS-Adapter.

Featuremanagement (klientenseitig)

Subsystem zur Verwaltung von Feature-Instanzen in der graphischen Darstellung. Aufgaben dieses Systems sind Zugriffe auf Feature-Attribute im Serverteil sowie die Steuerung der dynamischen Geometriereduktion der Feature.

Shapeverwaltung (klientenseitig)

Dieses Subsystem ermöglicht das Einlesen von ESRI-Shapefiles über URLs und deren Repräsentation. Weiterhin bietet das System die Funktionalität die Shape-Objekte in eine interne Darstellung (*ShapeFeatureProxy*) abzubilden.

6.2.1.8 ShapeURLService

Auf einem WWW-Server werden Shapefiles (+ dbase-Files) in einem Verzeichnisbaum abgelegt. Der Verzeichnisbaum (Verzeichnisnamen) repräsentiert eine thematische Hierarchie in welche die jeweiligen Shapes eingeordnet werden. Dieses Modul besitzt die Aufgabe, Verzeichnisbäume auf einem WWW-Server dynamisch auszulesen und in eine interne Struktur zu überführen. Die Wurzel eines Verzeichnisbaums ist durch eine URL gegeben. In einem solchen (Shape-) Wurzelverzeichnis kann es eine Datei mit einem ausgezeichneten Namen (z.B. SpatialURLServiceList) geben in der weitere (Shape-) Wurzelverzeichnisse auf anderen WWW-Servern eingetragen sind. Aufgabe des Moduls ist die Baumhierarchie eines Wurzelverzeichnis auszulesen. Dabei sollen Verzeichnisnamen als Themen erkannt werden und Verzeichnisinhalte (nur Shapes) als Shape-URL-Ressourcen zusammen mit der Dateigröße ausgelesen werden. Falls das Wurzelverzeichnis eine „SpatialURLServiceList“ besitzt sollen diese Server ebenfalls rekursiv ausgelesen werden.

6.2.1.9 Shapefile Adapter

Das auch als Shapefile-Server betrachtbare Subsystem liest die Metadatenbeschreibung (Datenschemata) einer Menge von ESRI-Shapefiles ein und kann Bereichsabfragen, die sich auf die in der Datenschema definierten Themen beziehen, beantworten. Die sich aus der Abfrage ergebenden Objekte werden in die einheitliche Feature- bzw. Geometriedarstellung abgebildet und damit für andere Subsysteme für eine weitere Verarbeitung zur Verfügung gestellt. Die Sichtbarkeit der aus den den Shapefiles gehörenden dBase-Dateien ermittelten Sachattributen kann eingestellt werden.

6.2.1.10 RIPS Adapter

Dieses Subsystem dient als komplexe Schnittstelle zwischen GIS-System und dem Räumlichen Informations- und Planungssystem (RIPS). Die Datenschemata von RIPS werden in die interne systemunabhängige Metadaten-Schema abgebildet und als Basis für Objektanfragen verwendet. Die sich aus den Bereichsanfragen ergebenden Objekte werden in die einheitliche Feature- bzw. Geometriedarstellung transformiert und damit für andere Subsysteme für eine weitere Verarbeitung zur Verfügung gestellt. Sachattribute werden ebenfalls durch Objektanfragen ermittelt. Die Sichtbarkeit jedes Sachattributs kann aus- bzw. eingeschaltet werden.

6.2.2 Allgemeine graphische Komponenten

Diese Subsysteme realisieren graphische Komponenten auf der Basis der zuvor aufgeführten internen Subsysteme. Diese Komponenten sind konzeptionell für die JavaBeans-Komponentenarchitektur vorbereitet und werden ab Q1 '98 in die JavaBean-Technologie überführt.

6.2.2.1 Diagrammdarstellung

Mithilfe dieses Subsystems werden Business-Diagramme generiert und visualisiert.

6.2.2.2 Graphikkern

Subsystem zur Darstellung interaktiver Graphikszene. Dieses System realisiert eine allgemein anwendbare, objektorientierte 2D-Graphikbibliothek. Der Graphikkern ermöglicht die deklarative Repräsentation von Graphikdaten auf einem hohen Abstraktionsniveau. Dadurch werden komplexe Strukturen wie sie im GIS- und CAD Bereich auftreten, für den Anwendungsprogrammierer beherrschbar. Die gesamte graphische Funktionalität des GISterms leitet sich aus diesem System ab. Teil dieses Systems ist auch die grundlegende Layermanagement. Über eine Anzeigekomponente *Display* wird der Graphikkern in die

6.2.2.3 DisplayToolbar

Mit dieser Komponente kann der Benutzer den Interaktions-Modus der *Display*-Komponente bestimmen. Es ist möglich die Funktionalität dieser Komponente parametrisierbar einzuschränken und dadurch indirekt die interaktiven Möglichkeiten der Anzeigekomponente festzulegen.



Abbildung 6.2: Werkzeugleisten-Widget

6.2.2.4 Koordinatenanzeige

Die Koordinatenanzeige realisiert eine Komponente, welche die aktuelle Mausposition in der Anzeigekomponente (*Display*) in Anwendungs koordinaten anzeigen kann.

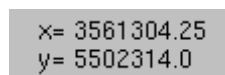


Abbildung 6.3: Koordinatenanzeige

6.2.2.5 Layerverwaltung

Durch diese Komponente erlangt der Benutzer Kontrolle über die Layer die innerhalb einer Anzeigekomponente verwaltet werden.

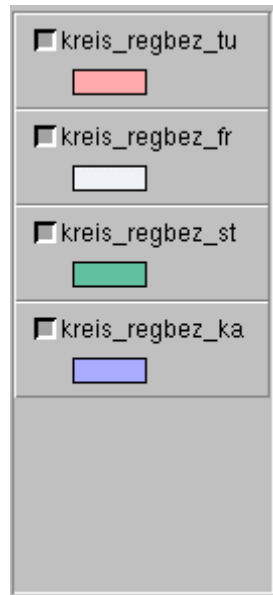


Abbildung 6.4: Layerverwaltung

6.2.2.6 Locator View

Die Komponente *LocatorView* gibt dem Benutzer Orientierungshilfe, indem sie den aktuell in der Anzeigekomponente sichtbaren Ausschnitt im Gesamtkontext anzeigt.

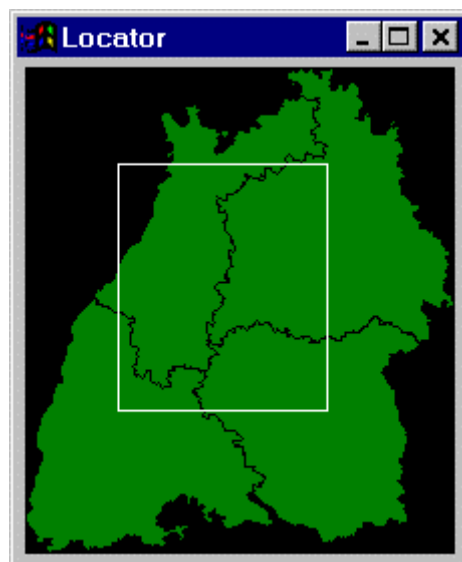


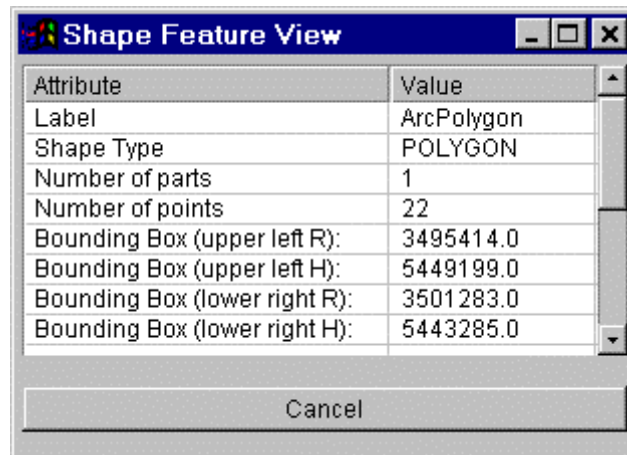
Abbildung 6.5: Auswahlbox

6.2.3 Anwendungsbezogene graphische Komponenten

Diese Komponenten realisieren graphische Anwendungskomponenten auf der Basis der zuvor aufgeführten internen Subsysteme.

6.2.3.1 Feature View

Diese Komponente dient der Anzeige aller Sachattribute eines Geoobjekts (*Feature*).



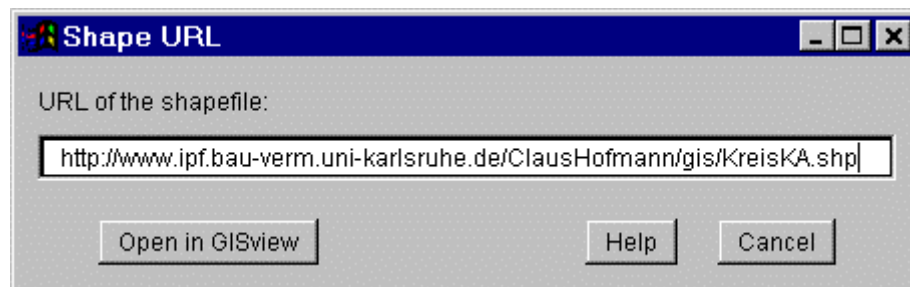
Attribute	Value
Label	ArcPolygon
Shape Type	POLYGON
Number of parts	1
Number of points	22
Bounding Box (upper left R):	3495414.0
Bounding Box (upper left H):	5449199.0
Bounding Box (lower right R):	3501283.0
Bounding Box (lower right H):	5443285.0

Cancel

Abbildung 6.6: Attributsanzeige

6.2.3.2 URL-Dialog

Ermöglicht den Zugriff auf beliebige URL-Ressourcen.



Shape URL

URL of the shapefile:

Open in GISview Help Cancel

Abbildung 6.7: URL-Dialog

6.2.3.3 Spatial URL-Browser

Sie dient dem Browsen und Aktivieren von räumlichen URL-Ressourcen.

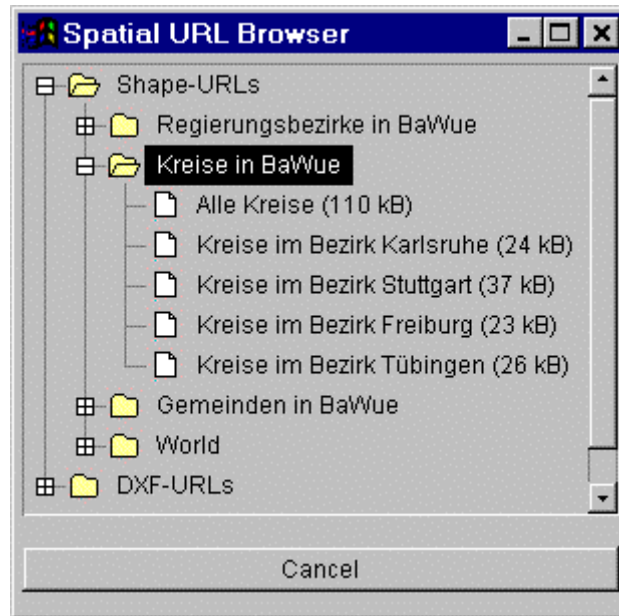


Abbildung 6.8: Themahierarchie

6.2.3.4 Bereichsanfrage

Diese Komponente ermöglicht zusammen mit der Anzeigekomponente die interaktive Definition einer Bereichsanfrage.

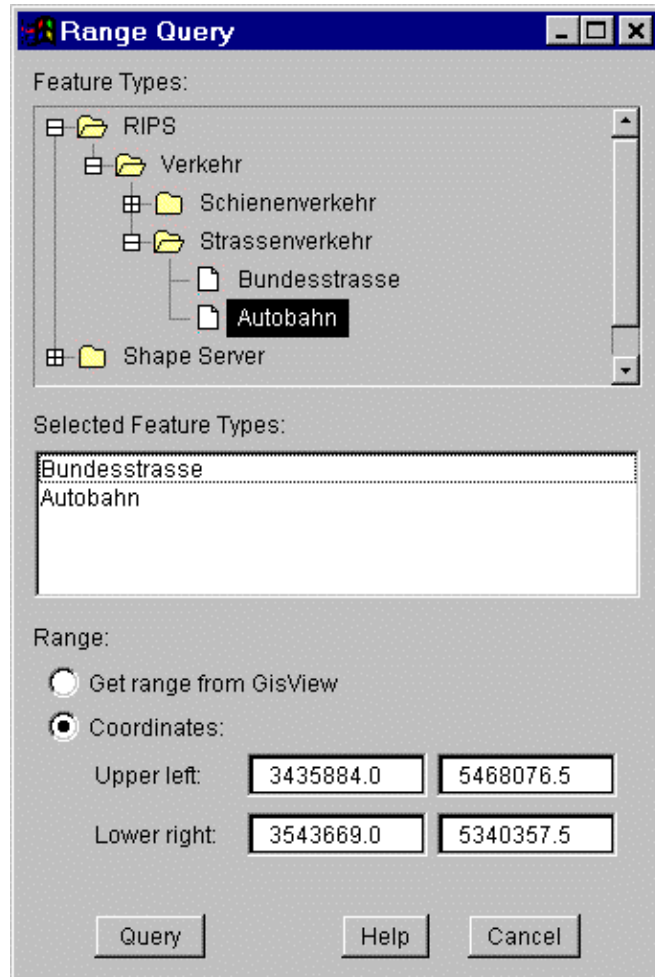


Abbildung 6.9: Bereichsanfrage

6.3 Management heterogener Datenquellen

Der Benutzer des GIS-System sieht dank der Abstraktionsschicht ein einheitliches virtuelles GIS und muß die Eigenschaften der einzelnen, darunterliegenden Geoinformationssysteme nicht kennen. Die transparente Arbeit mit Daten aus mehreren Informationsquellen wird durch die Abstraktion der Informationssysteme ermöglicht. Hierunter ist vor allem die Abbildung des jeweiligen Datenmodells in ein einheitliches Modell, sowie die Abstraktion der von den Geoinformationssystemen gebotenen Funktionalität zu verstehen. Diese finden im Serverteil bzw. mittels den als Treiber für die Informationssysteme funktionierenden Adaptionen statt.

6.3.1 Arten von Geoinformationen

Die in Geoinformationssystemen vorgehaltenen Informationen kann man in zwei Kategorien zerlegen:

- geographische Daten,
- Metadaten.

Digitale geographische Daten sind meistens nach einem GIS-spezifischen Datenmodell organisiert. Hinsichtlich dem Datenmodell kann man zwischen zwei Teilmodellen unterscheiden: einem Semantik- und einem Geometriemodell. Das Geometriemodell beschreibt die Raumgeometrie sowie die Topologie der betrachteten Geoobjekte in der Natur. Ein Semantikmodell beschreibt dagegen die zugrundeliegende Thematik eines räumlichen Objekts. Daten dieses Modells werden oft auch als Sachdaten bzw. Sachattribute bezeichnet. Das Semantikmodell ist stets von der Anwendung abhängig, jedoch finden sich grundlegende Konzeptionen und damit Gemeinsamkeiten in vielen, voneinander unterschiedlichen Problemstellungen.

Metadaten sind Daten beschreibende Daten oder einfach formuliert: Daten über Daten. Metadaten beinhalten wichtige Informationen, die dem Benutzer über den Inhalt, Struktur, Verfügbarkeit und Genauigkeit der Daten Auskunft geben. Sie können als Beschreibung des Datenmodells betrachtet werden. Metadaten sind bei Systemen, die Daten aus verschiedenen Quellen zusammenführen, von besonders großer Bedeutung.

6.3.2 Das Datenmodell des GIS-Terminals

Das Datenmodell in der Abstraktionsschicht wurde auf internationale Bemühungen zur Standardisierung von Geodaten (OGIS, CEN) basierend definiert. Es verfolgt dem Objektklassenprinzip und ist allgemein genug, um das Datenmodell verschiedener auf dem Markt befindlicher Geoinformationssysteme abzubilden. Dieses einheitliche Modell dient als Basis für Operationen in den oberen Schichten.

Das Semantikmodell eines GIS ist hochgradig von der Anwendung für die es konzipiert wurde abhängig. Gleiche physikalische Gebilde können einer Vielzahl von Interpretationsmöglichkeiten unterzogen werden. Deshalb wurde ein dynamisches, objektorientiertes Semantik- bzw. Geometriemodell im Serverteil des GIS-Terminals implementiert (Abbildung 6.10). Objekttypen, Assoziationen und Hierarchieebenen können dynamisch, d.h. zur Laufzeit, gebildet werden. Systeme, deren Datenmodell nicht dem Objektklassenprinzip unterliegen, lassen sich in das universelle Modell der Abstraktionsschicht transformieren.

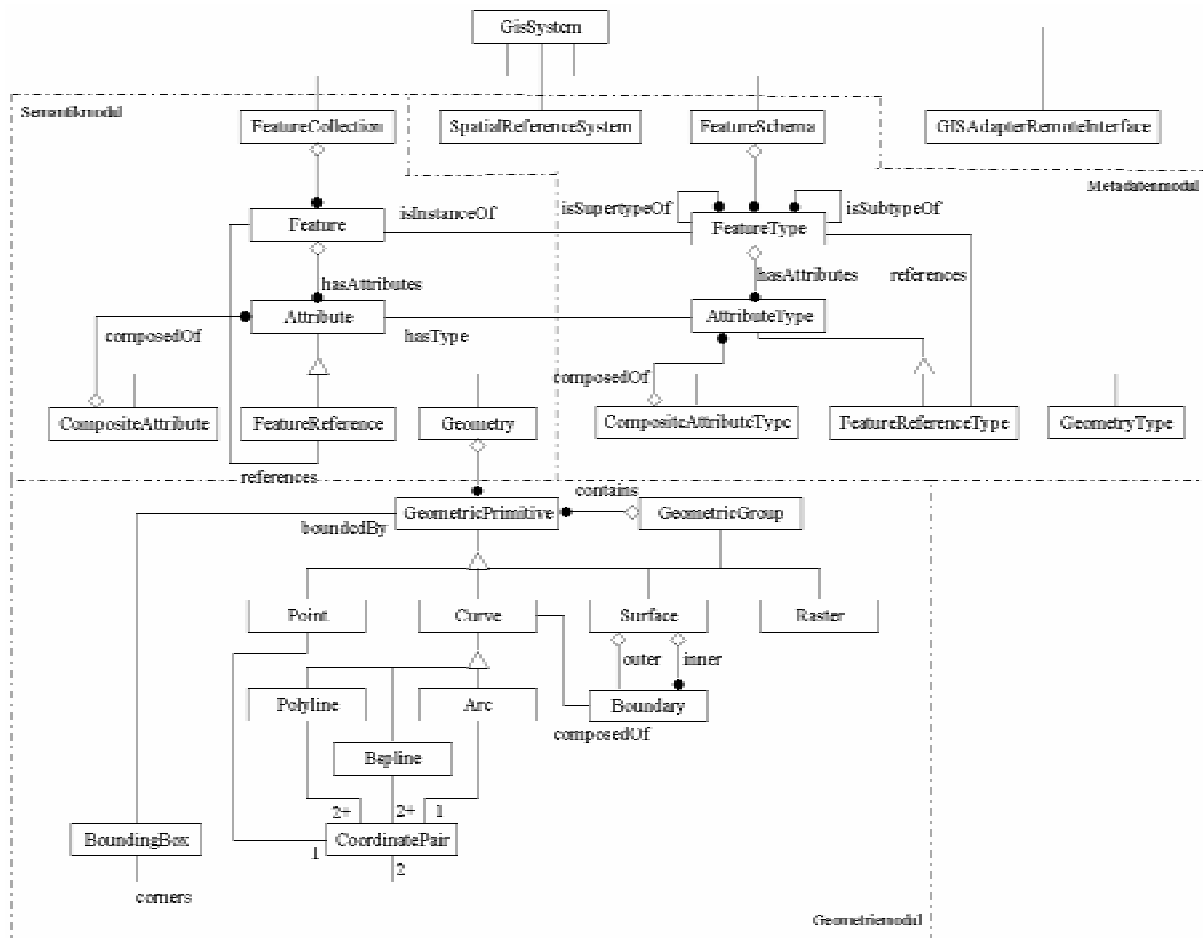


Abbildung 6.10: Datenmodell der Abstraktionsschicht

Metadaten werden intensiv für die Generierung und Bearbeitung von Abfragen benutzt. Die Objekttypen (*FeatureType*) kennen ihre Ober- bzw. Untertypen sowie ihre mögliche Attributierung (*AttributeType*). Durch einen speziellen Attributtyp (*FeatureReferenceType*) werden Assoziationen zwischen Objekttypen beschrieben.

Die Ergebnisobjekte der Abfragen werden in einer Instanz der Klasse *FeatureCollection* vorgehalten. Für jedes Geoobjekt wird eine *Feature*-Instanz gebildet, die das Ergebnisobjekt repräsentiert. Diese Instanz aggregiert sowohl die Geometriebeschreibung (*Geometry*) als auch seine Sachattribute (Attribut). Das spezialisierte Attribut *FeatureReference* dient der Navigation zwischen Geoobjekten. Das Geometriemodell ist so konstruiert, daß die Geometriemodelle der angeschlossenen Geoinformationssysteme darin – möglichst ohne Informationsverlust – konvertiert werden können. Die Geometrie eines Objekts besteht aus geometrische Primitiveelemente (*GeometricPrimitive*), aus denen durch ein Aggregationsobjekt (*GeometricGroup*) komplexere Strukturen aufgebaut werden können. Die Grundprimitive der Geometrie sind folgendermaßen kategorisiert:

- Punktgeometrie (*Point*),
- Liniengeometrie (*Polyline*, *Arc*, *Bspline*),
- Flächengeometrie (*Surface*),
- Rastergeometrie (*Raster*).

Bei jedem Grundprimitiv wird das achsenparallele minimale umschließende Rechteck (BoundingBox) mitverwaltet.

6.3.3 Abfragebearbeitung

Auf der abstrakten Datendarstellung aufbauend wurde das Abfragemanagement in der Managementschicht des GIS-Terminals realisiert. Die heterogene Menge der unterstützten Geoinformationssysteme machte erforderlich, daß ein allgemeiner Abfragemechanismus ausgearbeitet wurde. Dieser Mechanismus kann nicht direkt auf systemspezifische GIS-Datenbanken bauen, weil diese nach sehr unterschiedlichen Konzepten arbeiten. Die Vielfältigkeit der proprietären Lösungen einzelner Geoinformationssysteme wird durch das Adapterkonzept, das zwischen der einheitlichen abstrakten Abfragedarstellung und den konkreten GIS-Abfragen konvertiert, verdeckt.

Die Metainformationen der einzelnen Geoinformationssysteme, also Objekttypenbeschreibungen, Attribute und mögliche Assoziationen, werden für die Generierung und Bearbeitung von Abfragen benutzt. Dies führt dazu, daß die Abstraktionsschicht und die darauf stützende Managementschicht sehr eng zusammenarbeiten müssen.

Abfragen lassen sich in drei Arten unterscheiden (Abbildung 6.11.). Mit Hilfe von Metadatenabfragen (MetadataQuery) wird das Datenmodell eines angeschlossenen Geoinformationssystems in eine Objekttyphierarchie abgebildet. Geoobjekte können mittels Bereichsanfragen (SpatialObjectQuery) aus einem GIS gewonnen und instanziiert werden. Außerdem können Attribute von Objekten desselben Typs durch Attributabfragen in Tabellenform (AttributeQuery) ermittelt werden. Attributbedingungen, denen die Ergebnisobjekte genügen müssen, werden durch Condition-Instanzen angegeben. Die Einschränkungen können aufgrund der Raumgeometrie der Objekte, Attributbedingungen oder einer Kombination der beiden zusammengestellt werden.

Das Ergebnis einer Bereichsanfrage wird als eine Menge von Geoobjekten in einer Instanz der Klasse ResultSet zurückgegeben. Die Attributwerte, welche sich durch eine Attributabfrage ergeben, werden in einer AttributTable vorgehalten. Bei all diesen Ergebnisobjekten werden Verknüpfungen zu den Metadaten hergestellt, so können die Ergebnisdaten sofort in darauffolgenden Abfragen verwendet werden.

Für jede Abfrage erzeugt der GisTermServer ein QueryHandler-Objekt, das als eigener Thread realisiert ist. Dadurch ist die Bedienung des GIS-Terminals von der Abfragebearbeitung entkoppelt und der Benutzer kann die Wartezeit für weiteres Explorieren in bereits vorhandenen Daten nutzen. Der QueryHandler analysiert die Anfrage und sortiert diese nach FeatureTypes mit Hilfe des FeatureSchema nach angeschlossenen GIS-Systemen. Daraufhin wird für jedes GIS eine eigene Subabfrage erzeugt, die nur Geoobjektclassen des entsprechenden Systems besitzt. Diese Subabfragen werden über die Adapterobjekte der einzelnen GIS-Systeme erfragt.

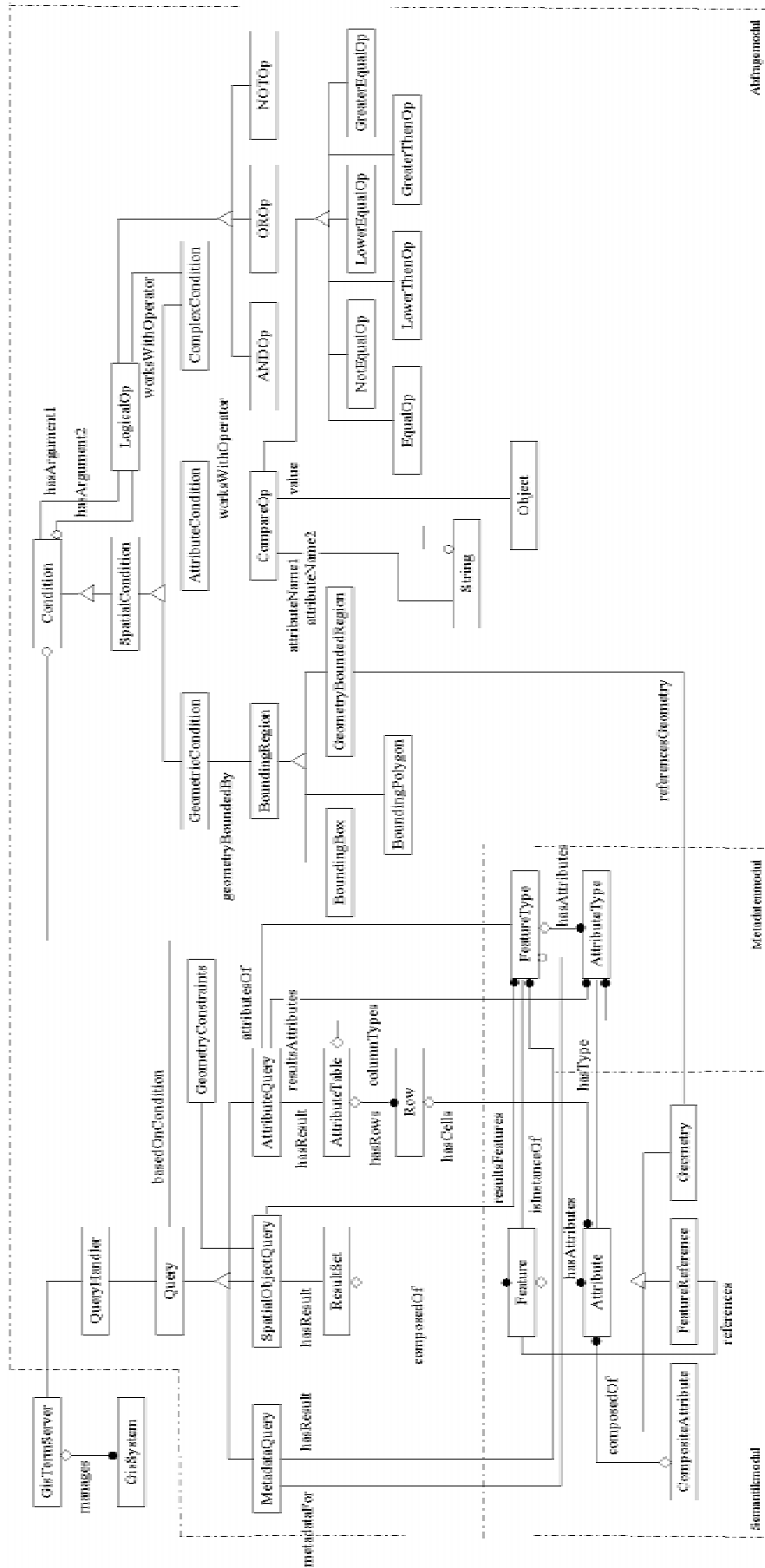


Abbildung 6.11: Das Abfragemodul

6.3.4 Realisierung der Adapter

Das GIS-Terminal – zusammen mit den angeschlossenen Geoinformationssystemen – stellt ein für den Benutzer transparentes, verteiltes Informationssystem, dar. Die Transparenz wird durch die Abstraktionsschicht, die auf den Adaptern basiert, gewährleistet. Da sowohl der Zugriff auf unterschiedliche Systeme, als auch das vorhandene Datenmodell der Systeme sehr unterschiedlich sein kann, wird für jedes unterstützte System ein Adapter implementiert. Diese systemspezifische Adapter bieten der Abstraktionsschicht eine einheitliche Schnittstelle. Es werden einheitliche Zugriffsoperatoren geboten, des weiteren findet hier die Transformation des jeweiligen Datenmodells sowie des Metamodells in das Datenformat der Abstraktionsschicht statt.

6.3.4.1 Kommunikation zwischen GIS-Server und den Adaptern

Weil die Adapter den Zugriff auf Geoinformationssysteme, deren Aufbau und externe Schnittstellen unveränderbar vorgegeben sind, realisieren, kann keine reine Java-basierte Implementierung verwirklicht werden, es sei denn, daß das System ein Java-Interface besitzt (was derzeit bei keinem auf dem Markt befindlichen System der Fall ist). Deshalb muß man nach anderen Möglichkeiten suchen. Es bieten sich grundsätzlich zwei Möglichkeiten an, um auf die Daten eines zu integrierenden Informationssystems zuzugreifen:

1. eine CORBA-basierte Lösung,
2. eine auf JNI- und RMI-basierte Lösung.

Bei der CORBA-basierten Lösung muß das Geoinformationssystem über eine externe C oder C++ Programmierschnittstelle verfügen, damit der Adapter durch diese Schnittstelle Operationen im GIS anstoßen bzw. Daten aus dem GIS gewinnen kann. Der C/C++-ORB (Object Request Broker) leitet die Daten mittels IIOP (Internet Inter-ORB Protocol) an den Java-ORB im GIS-Server weiter, wo daraus Java-Objekte instanziiert werden. Der große Vorteil dieser Lösung ist es, daß der Server nicht genau wissen muß, an welchem Rechner der Adapter residiert (Ortstransparenz).

In dem anderen Fall, wenn kein CORBA-Einsatz möglich ist, werden die Abfragen mittels RMI an den Adapter weitergeleitet. Der Adapter benutzt die externe Programmierschnittstelle des GIS, um darüber Anfragen zu stellen. Die Ergebnisdaten werden aus der verwendeten Programmiersprache mit Hilfe des Java Native Interface (JNI) – noch im Adapter – nach Java-Objekte umgewandelt. Der Adapter setzt wiederum RMI ein, um die Java-Objekte an den Serverteil des GIS-Terminals zu übertragen.

Da in dieser Projektphase der CORBA-Einsatz nicht beschlossen wurde, benutzt die derzeitige Implementierung der Adapter das JNI- und RMI-basierte Verfahren.

6.3.4.2 Der RIPS-Adapter

Der zu dem Smallworld-GIS bzw. dem darauf basierenden Räumlichen Informations- und Planungssystem (RIPS) gefertigte Adapter hat die Aufgabe, auf die in RIPS abgelegten Geoinformationen zuzugreifen.

Die Geometriedaten werden vom Smallworld-GIS verwaltet, die Sachdaten verwaltet dagegen eine separate Oracle-Datenbank. Smallworld-GIS kann mit seiner eigenen, objektorientierten Sprache Magik, die auch SQL-Abfragen versteht und diese sogar zu externen Datenbanken weiterleiten kann, programmiert werden. Durch eine zu dem System gehörende C-Bibliothek wird es ermöglicht, daß externe C-Programme – mittels TCP/IP – mit Magik-Programmen kommunizieren können. Da aus Magik nicht direkt auf das JNI, sondern nur auf C zugegriffen werden kann, wurde eine mehrschichtige Architektur für den RIPS-Adapter entworfen (Abbildung 6.12). Die von dem GIS-Terminal zusammengestellten Abfragen werden von Schicht zu Schicht in weniger abstrakte Formen transformiert und weitergereicht, bis letztendlich das GIS die benötigten Operationen ausführt. Die Ergebnisdaten werden dann auf dem umgekehrten Weg nach oben geleitet. Mit Hilfe des JNI werden schließlich Java-Objekte

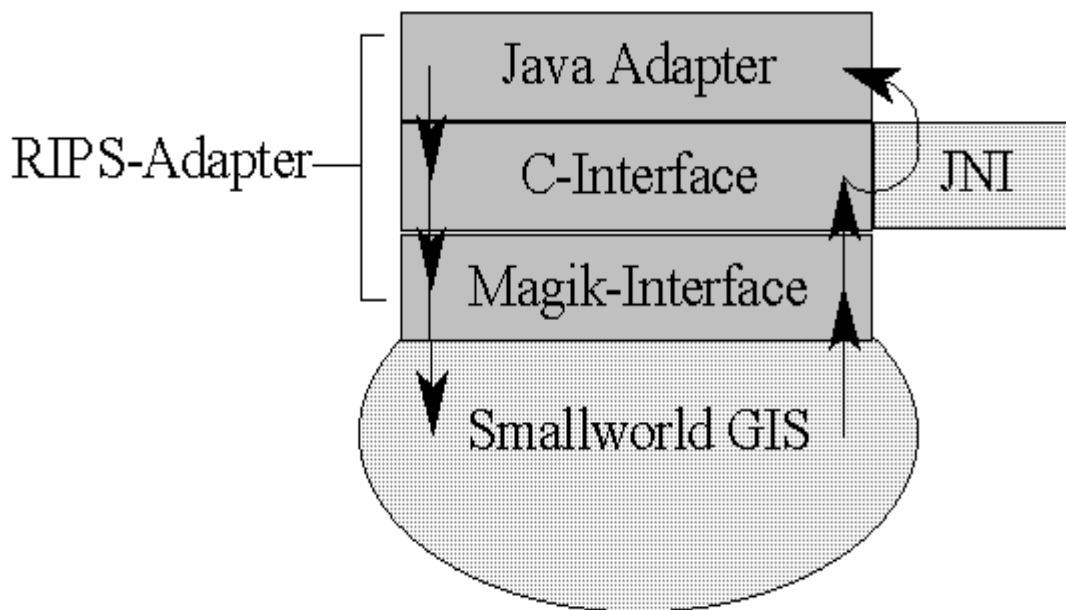


Abbildung 6.12: Architektur des RIPS-Adapters

Es wurde eine Schnittstelle zum Smallworld-GIS entworfen und in Zusammenarbeit mit der Firma isys realisiert. Die definierte (zunächst Basis-) Funktionalität, die von dem Magik-Interface und einem Teil des C-Interfaces abgedeckt wird, beinhaltet das Beantworten von Metadatenabfragen, einfachen Bereichsabfragen und Attributabfragen. Unter Bereichsabfrage wird hier lediglich die Suche nach Geobjekten, die sich mit einem angegebenen rechteckigen Bereich (Bounding Box) überlappen, verstanden. Das Problem, daß die Geometrie einiger Ergebnisobjekte extrem große Mengen von Vektor- bzw. Rasterdaten aufweisen kann, könnte durch Ausdünnung der Geometriedaten beseitigt werden, wurde aber in dieser prototypischen Form des RIPS-Adapters nicht implementiert.

Bei der Implementierung des Adapters entstanden Probleme durch das nicht vollständig realisierte OO-Datenmodell des RIPS. Weiterhin wird das ATKIS-Modell in dem derzeitigen RIPS nicht unterstützt. Aus diesen Gründen mußte der Adapter zusätzlich einen Teil von der fehlenden objektorientierten Funktionalität implementieren. Ein Problem stellt die Tatsache dar, daß kein allgemeiner Objektidentifikator in RIPS vorhanden ist. Obwohl eine eindeutige Objekt-ID (bestehend aus dem RIPS-Objektartencode, einem Rechtswert und einem Hochwert) in dem RIPS-Entwurf definiert wurde, ist diese ID – oder gar ihr wichtigster Bestandteil, der RIPS-Objektartencode – zur Zeit nicht in der RIPS-Datenbank aufzufinden. Für die

Zusammenarbeit verschiedener Systemkomponenten (z.B. für die Kommunikation zwischen dem GIS-Terminal und der Sachdatenkomponente), die auf denselben Daten arbeiten müssen, ist aber die Existenz solcher allgemeiner und eindeutiger Objekt-IDs unerlässlich.

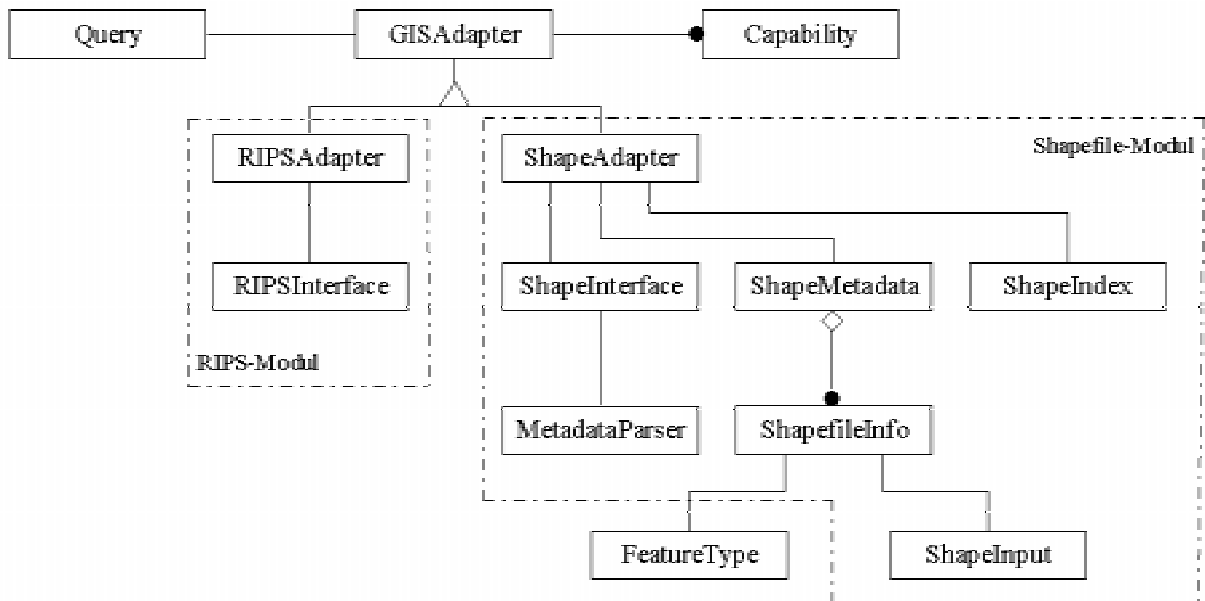


Abbildung 6.13: Die realisierten Adaptermodule

6.3.4.3 Der Shapefile-Adapter

Eine Menge von ArcView Shapefiles stellt kein Geoinformationssystem dar, vielmehr werden sie als Dateien für den Datenaustausch benutzt. Da es sich nur um Dateien handelt, mußte der Shapefile-Adapter einige einfache GIS-Funktionen selbst implementieren.

Hinter dem Shapefile-Format steckt eine sehr einfache Datenstruktur. Die Beschreibung eines Themas wird in drei separaten Dateien abgelegt: das Shapefile selbst beinhaltet die Geometriedaten (ohne topologische Beziehungen) und zu diesem Shapefile gehört ein Indexfile. Die Attributdaten werden in einer dBase-Datei vorgehalten.

Das Format speichert keine Metadaten über geographische Objekte, lediglich die Typen der einzelnen Attribute werden in dem dBase-File definiert. Weil das GIS-Terminal Metadaten über die von ihm verwalteten Geoobjekte benötigt, wurde eine einfach generierbare, leicht wartbare Metadatenbeschreibung definiert, die zusammen mit den Shapedateien gespeichert wird. Die in Shapefiles abgelegten Informationen sind nach dem Layerprinzip¹ geordnet. Eine der Aufgaben der Metadatenbeschreibung besteht darin, die Transformation der Daten in Objekte (dem Objektklassenprinzip nach) zu erleichtern. Des weiteren dienen diese zusätzliche Metadaten der besseren Identifizierung der Themen (Objektklassen) und gewähren dem Benutzer weitere Informationen über die Attribute von Geoobjekten.

Neben dem Versehen des GIS-Terminals mit Metadaten umfaßt die realisierte Funktionalität des Shapefile-Adapters das Beantworten von Objekt- und Attributabfragen. Wie erwähnt, soll

¹ Dieses Prinzip separiert die Geometriedaten verschiedener thematischer Bedeutung streng durch die Abspeicherung in verschiedenen Ebenen, die als Layer bezeichnet werden.

der Adapter dabei einige GIS-Aufgaben selbst erledigen, etwa die Suche nach Objekten in einem angegebenen geographischen Bereich (SpatialObjectQuery).

Der Shape-Adapter wurde vollständig in Java implementiert, d.h. das JNI wurde nicht verwendet. Um dies zu ermöglichen, mußte die vorhandene C-Bibliothek für Einlesen von Shape-Dateien in Java umgeschrieben werden. Der Zugriff auf die Shapefiles wurde durch die Klasse ShapeInput realisiert (s. Abbildung 6.13).

6.4 Klientseitige UIS-Anwendungsschnittstelle des GIS- Sterm

Die nachfolgend erläuterte Schnittstelle beschreibt allgemeine Anwendungsdienste, die von der GIS-*Sterm*-Komponente für das UIS-Szenario angeboten werden. Die Schnittstelle wurde klientseitig realisiert, damit Daten die in der Sachdatenkomponente schon auf dem lokalen Rechner vorliegen direkt übernommen werden können. Derzeit lassen sich zwei grundlegende UIS-Anwendungsdienste unterscheiden.

6.4.1 Darstellungsdienste

Das GIS-*Sterm* bietet Darstellungsdienste für die Präsentation von numerischen Wertereihen in Business-Diagrammen sowie für die Visualisierung von räumlichen Objekten in Karten an. Die zu präsentierenden Daten werden in einer objektorientierten Form an der Anwendungsschnittstelle übergeben. Die OMT-Modellierung der Schnittstellenobjekte ist in Abbildung 6.14 zu sehen.

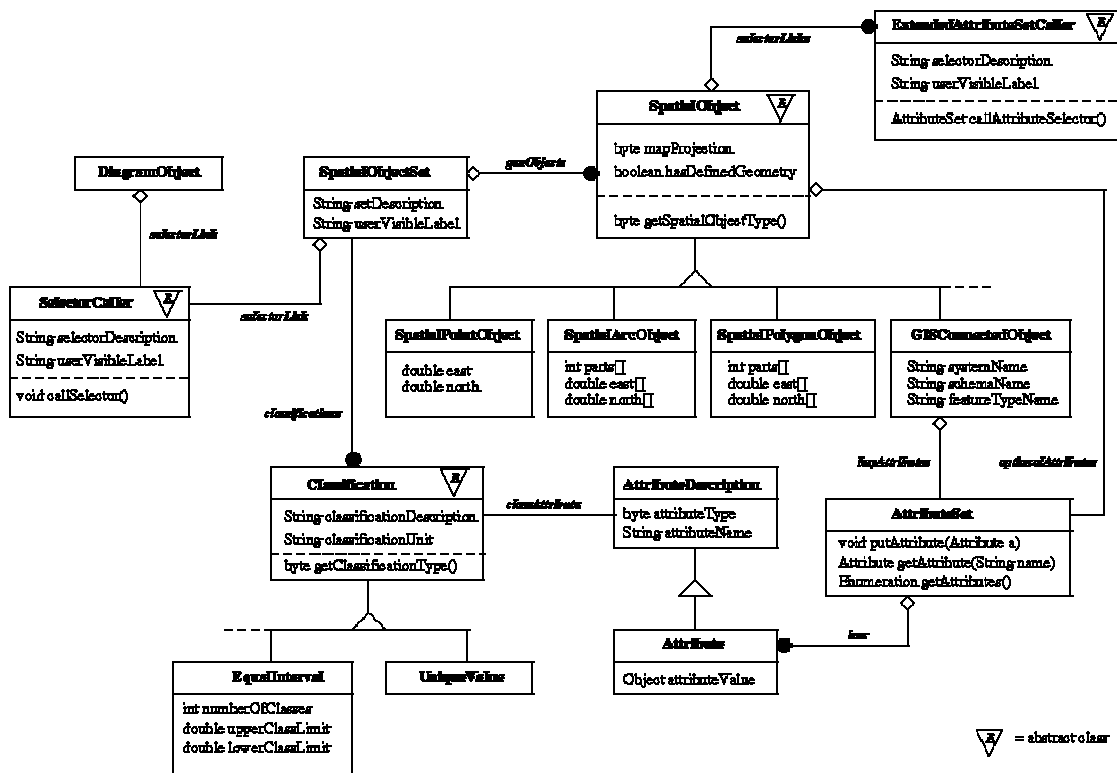


Abbildung 6.14: OMT-Modellierung der Schnittstellenobjekte

Die Anwendungsschnittstelle wurde im Hinblick auf eine saubere Trennung zwischen Daten- definition und Datenpräsentation erstellt. Die Schnittstellenobjekte beschreiben die zu prä- sentierenden Daten, ohne selbst Vorgaben zur Darstellung zu machen. Dadurch wird eine allgemeine Anwendbarkeit erreicht. Weiterhin können die einzelnen Komponenten dadurch unabhängig voneinander weiterentwickelt werden. Zwei Schnittstellenklassen sind für die Darstellungsdienste zentral. Mithilfe der Klasse `DiagramObject` wird die Definition von meh- reren Wertereihen die in einem Business-Diagramm dargestellt werden sollen realisiert. Die Klasse beinhaltet Metainformationen zu den Wertereihen, die es ermöglichen dem Benutzer für den Datensatz geeignete Darstellungsziele (Präsentationsarten) anzubieten. Weitere Informa- tionen die das `DiagramObject` näher beschreiben sind eventuell vorhandene Grenzwerte und geographische Bezüge der einzelnen numerischen Wertereihen. Diese Bezüge erlauben die Präsentation des Gesamtdiagramms, aber auch einzelner Wertereihen, innerhalb einer Karte. Die zweite zentrale Schnittstellenklasse `SpatialObjectSet` aggregiert eine heterogene Kollekti- on von räumlichen Objekten. Räumliche Objekte können dabei unterschiedlicher Natur sein. Es stehen Klassen für die Beschreibung von Punkt-, Linien- und Flächenobjekten (`Spatial- PointObject`, `SpatialArcObject`, `SpatialPolygonObject`) zur Verfügung. Die Geometriebe- schreibung dieser Objekte lehnt sich an das einfache und vielfach für den Datenaustausch eingesetzte ESRI-ArcShape Format an. Mit der Klasse `GISConnectedObject` können räumli- che Objekte referenziert werden, die das GIS eigenständig in einem der angeschlossenen geographischen Informationssystemen erfragen kann. In diesem Fall muß also die Geometrie nicht an der Anwendungsschnittstelle angegeben werden. Die Klasse `GISConnectedObject` besitzt Informationen, mit der die Zuordnung zu Geoobjektklassen der Abstraktionsschicht des GIS erfolgen kann. Zudem kennt ein solches Objekt die Schlüsselattribute, welche das Objekt im GIS eindeutig identifiziert. Alle hier vorgestellten räumlichen Objekte können eine optionale Sachattributierung beinhalten. Diese Attribute werden dem Benutzer zur nähe- ren Identifikation des jeweiligen Geoobjekts angeboten. Bei der Übergabe der eben vorge- stellten räumlichen Objektmenge an das GIS können durch die Klasse `Classification` mehrere Klassifikationen definiert werden. Räumliche Objekte, die der gleichen Klasse zuge- ordnet werden, werden mit der gleichen Darstellungsattributierung visualisiert. Auch an die- ser Stelle trennt die Schnittstelle zwischen der Datendefinition und der eigentlichen Präsen- tationsart der Daten im GIS. Ein Beispiel einer klassifizierten Punktmenge wird in Abbil- dung 6.15 gezeigt. Die Klassifikation wird über die optionale Sachdatenattributierung reali- siert, indem der Name und der Typ des Klassifikationsattributs festgelegt wird. Alle räumli- chen Objekte der übergebenen heterogenen Kollektion, welche das in der Klassifikation defi- nierte Sachattribut besitzen, werden einer Klasse zugeordnet und entsprechend visualisiert. In der ersten Version dieser Schnittstelle sind nur einfache Klassifikationsarten spezifiziert (`EqualInterval`, `UniqueValue`).

Um dem Benutzer sowohl den Rücksprung zum datenerzeugenden Selektor, als auch die weitere Navigation auf vorhandenen Datenobjekten zu ermöglichen, definiert die Schnittstelle Kommunikationsobjekte. Die Objekte `DiagramObject` und `SpatialObjectSet` besitzen beide einen `SelectorCaller`, mit dem der Rücksprung zu dem Datenobjekt mit dem Datenmenge erfragt wurde ermöglicht wird. Einzelne räumliche Objekte können Navigationsobjekte (`Ex- tendedAttributSetCaller`) besitzen, über die zusätzliche Informationen zu diesem Objekt er- fragt und angezeigt werden können. Beide Caller-Klassen sind in der Schnittstelle als ab- strakte Klassen definiert. Die Implementierung bleibt den jeweiligen Selektoren vorbehalten, wodurch eine vom GIS unabhängige Entwicklung von Selektoren ermöglicht wird.

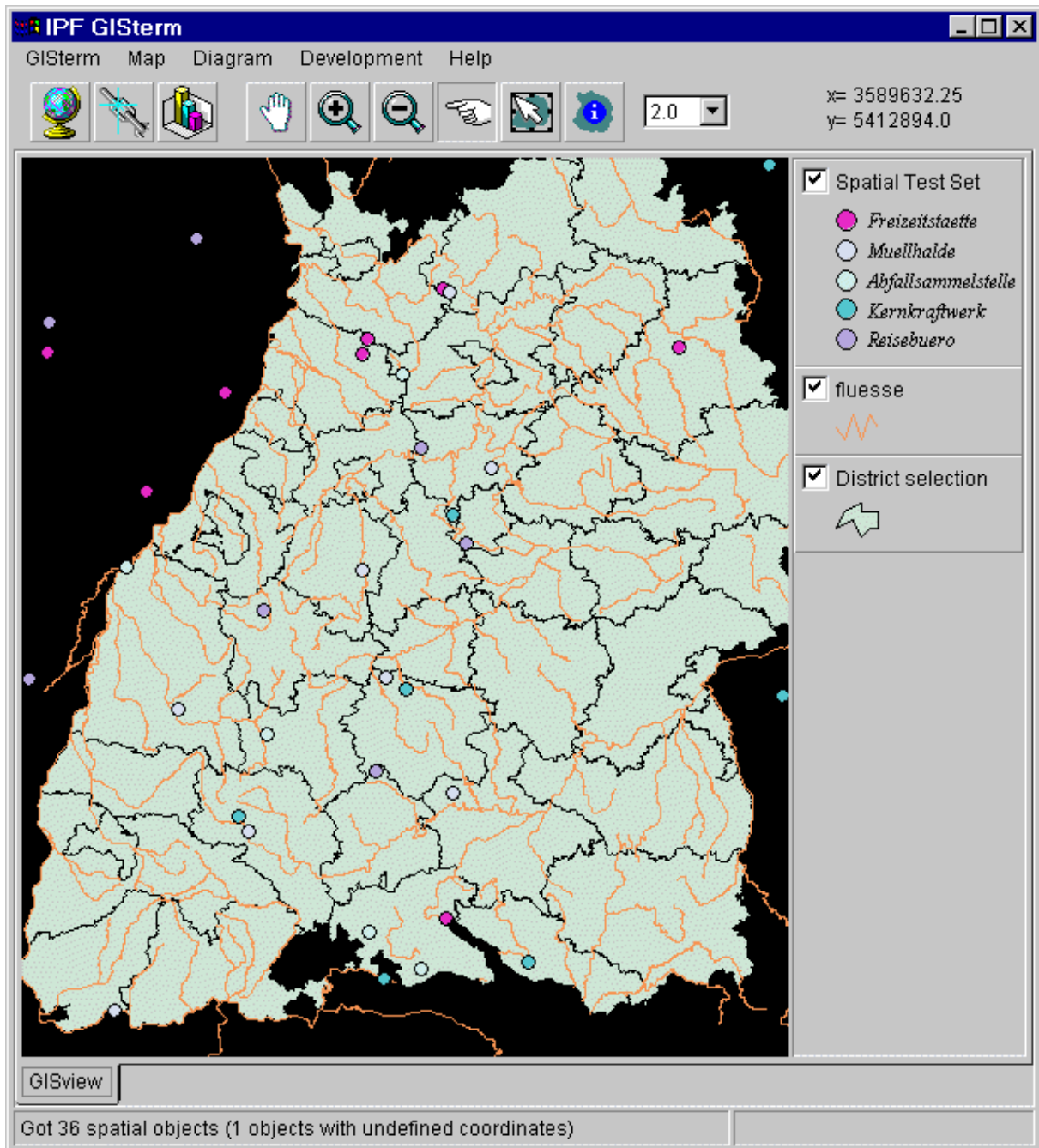


Abbildung 6.15: Beispiel einer klassifizierten Objektmenge

6.4.2 Auswahldienst

Die Auswahl von Verwaltungseinheiten (z.B. Kreise, Gemeinden, usw.) kann sowohl in der Sachdatenkomponente erfolgen, als auch graphisch im GISterm. Hat der Benutzer in einer Komponente eine Verwaltungseinheit ausgewählt, muß der jeweilig anderen Komponente dies mitgeteilt werden. Damit die Sachdatenkomponente die Selektionsereignisse dem GISterm melden kann, stellt diese Anwendungsschnittstelle die nötigen Kommunikationsmechanismen bereit.

7. Prototyp

Im Rahmen der Projekt-Phase sollte ein Prototyp konzeptioniert und realisiert werden, der aus den Komponenten WWW-UDK, Sachdatenkomponente und GIS-/Grafikkomponente besteht. Dieser Prototyp soll in einer weiteren Projektphase zum Produkt mit vollem Funktionalitätsumfang ausgebaut werden. Der Prototyp wurde bis Ende des Jahres 1997 realisiert und umfaßt einen Teil der Gesamtfunktionalität. Welche Funktionalitäten von den verschiedenen Komponenten im Umfang des Prototypen enthalten bzw. nicht enthalten sind, soll im folgenden beschrieben werden.

7.1 WWW-UDK

Der WWW-UDK wird an der LfU bereits seit 1996 als Produkt mit bekannter Funktionalität betrieben. Erforderlich für den Prototypen ist insbesondere:

- das Eintragen der Selektoren in den UDK
- eine Ankopplung der Sachdatenkomponente und der GIS-/Grafikkomponente

Nicht im Prototypen realisiert:

- das Liefern von Meta-Informationen des UDK an die Sachdaten- bzw. GIS /Grafikkomponente zur Darstellung in diesen Komponenten.

7.2 Sachdatenkomponente

Im Rahmen des Prototyps wurden folgende Funktionalitäten der Sachdatenkomponente realisiert:

- Steuerung
- Selektion
 - Selektorverkettung
 - Kontextsensitivität von Selektoreingaben
 - Beispielhaft für: allg. Meßstellenselektor, Meßreihenselektor (Laie), Meßreihenselektor (Spezialist)
- Ergebnisverwaltung
- Darstellungsauswahl (automatische Auswahl sinnvoller Darstellungen in Abhängigkeit vom Selektorergebnis)
- Tabellendarstellung

Nicht Bestandteil des Prototypen werden sein:

- Archiv
- Hierarchie
- Statistische Aufbereitung

Abbildung 7.1 zeigt einen Screendump des Selektorfensters. Hier wurde der Entsorgungsanlagenselektor ausgefüllt.



Abbildung 7.1 Selektorfenster am Beispiel des Entsorgungsanlagenselektors

7.3 GIS-/Grafikkomponente

Das auf dem Client-Server Prinzip aufgebaute GIS-Terminal dient zur Darstellung von Geodaten und Businessgraphiken. Die Fähigkeiten im Rahmen des Prototypen sind folgende:

- Darstellung und Verwaltung von Geobjekten (mit Attributen und Metainformationen, die über GIS-Adapter ermittelt werden)
- räumliche Auswahl von Geobjekten durch Bounding-Box
- mehrfache Selektion von Geobjekten aus dem Viewer
- Businessgraphik
- Abfrage von Metainformationen aus dem WWW-UDK

Im Rahmen des Prototypen sind folgende Einschränkungen vorgesehen:

- keine allgemeine Selektion von Verwaltungseinheiten (für Gebietsauswahl in Sachdatenteil), die Selektion wird nur auf eine bestimmte Verwaltungsebene eingeschränkt (z. B. auf Kreise)
- keine Zerlegung des GIS-Terminals in JavaBeans-Komponenten
- keine Selektion von Objekten mit Bounding Box

- Objekttyp Meßstelle und Objekttyp einer bestimmten Verwaltungseinheitsebene (z.B. Kreis) können nur aus der Sachdatenkomponente übernommen werden.
- Diagrammdarstellung erfolgt zunächst separat von Karten
- kein Plot der dargestellten Karten.

7.4 Kommunikation

Die Kommunikation zwischen Sachdaten- und GIS-/Grafikkomponente findet im Prototyp nur zwischen clientseitigen Komponenten statt.

8. Zusammenfassung

Das Ziel des WWW-Szenarios ist die Zusammenführung von Metainformationssystem, Sachdateninformationssystem und Geoinformationssystem zu einem über das WWW zugreifbaren Gesamtsystem.

Die Komponenten wiesen zu Beginn des Projekts unterschiedliche Entwicklungsstadien auf, was die Integration der Komponenten zusätzlich erschwerte. Der bestehende WWW-UDK war bereits in der Produktionsversion und ermöglichte mit dem bekannten CGI-Ansatz die Suche nach Metadaten. Er sollte nicht für die Integration ins WWW-Szenario verändert werden. Die beiden anderen Komponenten Sachdatenkomponente und Geoinformationssystem wurden neu mit Java bzw. JavaBeans, der Komponentenarchitektur von Java, entwickelt. Durch diese neue Technologie ist benutzerfreundlichere WWW-Software realisierbar.

Obwohl im WWW-Szenario unterschiedliche WWW-Technologien eingesetzt werden, ist es gelungen, diese zusammenzuführen und auch Datenaustausch zu ermöglichen. In einem ersten Prototyp wurde gezeigt, daß die erarbeiteten Konzepte und Lösungen sinnvoll umsetzbar sind.

Das vollständig realisierte Gesamtsystem wird für die interessierten Öffentlichkeit und für die Mitarbeiter der Umweltbehörden ein umfangreiches Informationssystem darstellen, das er/sie sich über das Intranet bzw. Internet zu einem beliebigen Zeitpunkt ohne Installationsvoraussetzungen (nur ein Javafähiger Browser wird gebraucht) an seinen Arbeitsplatz etc. laden kann.

9. Literatur

- /1/ Kramer, R.; Nikolai, R.; Rolker, C. (1996): Java-Evaluierung für das UIS Baden-Württemberg - Phase 1. Technical Report 5/96, Forschungszentrum Informatik (FZI), Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, May 1996
- /2/ Mayer-Föll, R.; Jäschke, A. (1996): Projekt GLOBUS, Konsolidierung der neuen Systemarchitektur und Entwicklung erster Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg, Phase III 1996, Number FZKA 5900, in Wissenschaftliche Berichte, Forschungszentrum Karlsruhe Technik und Umwelt, Karlsruhe
- /3/ Orfali, R.; Harkey, D. (1997): Client/Server Programming with Java and CORBA, Wiley

Pflege der installierten Software aus vorhergehenden Projekten und Unterstützung der LfU beim Betrieb der WWW-Server (GIS-Dienste)

*W. Weisbrich, J. Wiesel,
Institut für Photogrammetrie und Fernerkundung (IPF),
Universität Karlsruhe,
Englerstr. 7,
76128 Karlsruhe*

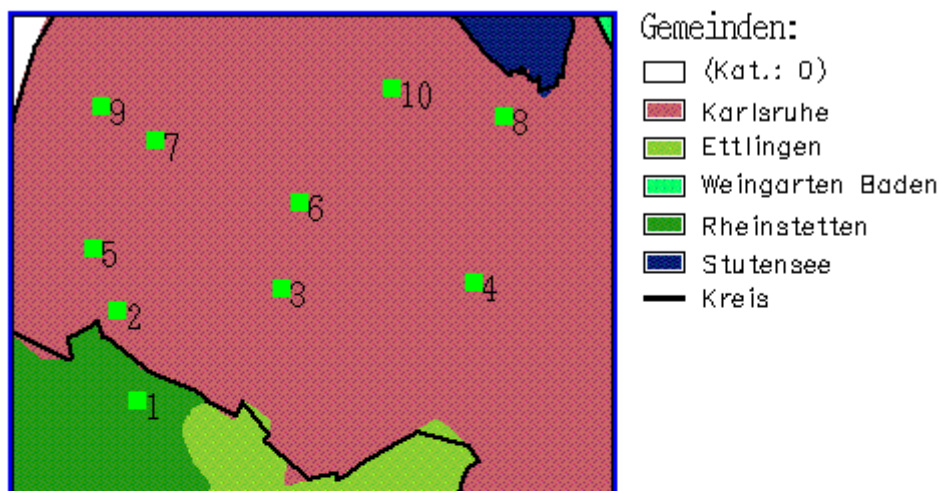
1. EINLEITUNG	147
2. ERWEITERUNGEN DER GRAPHISCHEN DIENSTE	147
2.1 EIPUNKT-DARSTELLUNG	147
2.2 HINTERGRUNDKARTE	148
2.3 DER STOPDÄMON.....	148
2.4 MODULARISIERUNG DER GRAPHISCHEN DIENSTE.....	149
3. LITERATUR.....	151

1. Einleitung

Eine der wesentlichsten Aufgaben des Projektes bestand in der Erweiterung, Stabilisierung und Modularisierung der Graphischen Dienste. Als Grundlagen dienten die Wünsche und Fehlerprotokolle, die uns von den bei der Entwicklung Verantwortlichen des Landesamt für Umweltschutzes (LfU) an getragen wurden. Hierzu gehörte auch die Dokumentation über die Schnittstelle zwischen TULIS/MEROS und den Graphischen Diensten, die in Form eines technischen Berichtes der LfU ausgehändigt wurde /1/.

2. Erweiterungen der Graphischen Dienste

2.1 Einpunkt-Darstellung



■ geographische Bezüge

Geobezüge: Betriebsstätte

1: 8116334 SEVAR Entsorgungsanlagen
2: 3294102 LINDER
3: 2737604 GWK Wertstoffrecycling Ka
4: 2831457 HOFHEINZ HOCK
5: 2831201 KOST

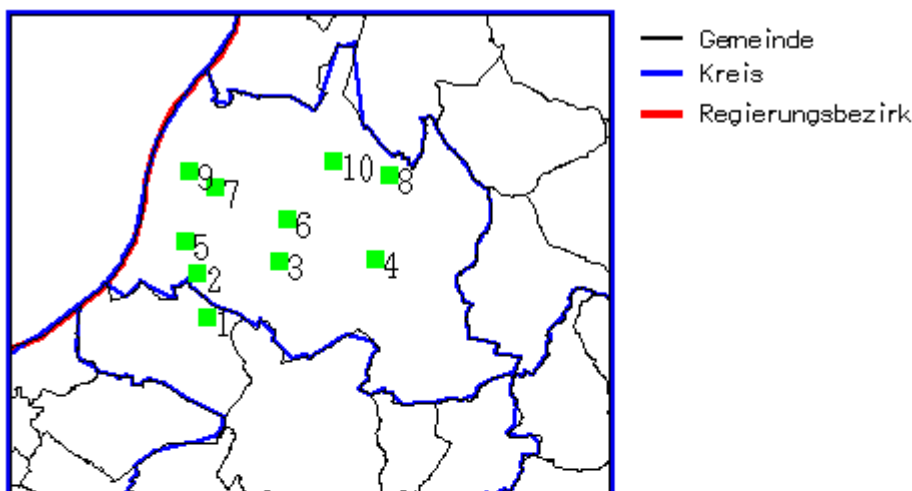
Abbildung 1: Beispiel einer Darstellungsseite mit Einpunkt-Darstellung (Ausschnitt)

Die meisten Anwendungen zur Visualisierung von umweltbezogenen Daten bestehen in der geokodierten Darstellung von Objekten, wie z. B. Betriebs-, Arbeitsstätten, Standorte, bei denen keinerlei Meßwerte selektiert wurden. Die übergebenen Geobezüge werden als Punkte mit Nummern in der Karte dargestellt. Zusätzlich wird eine Listbox angezeigt, in der zu den

Punktnummern der Name sowie zusätzliche Informationen zu den dargestellten Geobezügen angezeigt werden (siehe Abb. 1).

2.2 Hintergrundkarte

Bei der Kartendarstellung auf der Darstellungsseite der Graphischen Dienste war es bisher nur möglich entweder je nach Verwaltungsebene entsprechend eingefärbte Verwaltungseinheiten darzustellen oder eine Übersichtskarte (ÜK 500) mit verschiedenen Layern darzustellen. Diese Art der Kartendarstellung führte zu einer großen Belastung des Servers. Da eine Vielzahl von Anwendungen nicht unbedingt eine eingefärbte Hintergrundkarte benötigen, wurde eine leere Hintergrundkarte in die Karten-Auswahl der Graphischen Dienste integriert (siehe Abbildung 2). Hiermit wird die Belastung des Server bei der Generierung der Hintergrundkarte reduziert.



■ geographische Bezüge

Geobezüge: Betriebsstätte

1:	8116334 SEVAR Entsorgungsanlagen
2:	3294102 LINDER
3:	2737604 GWK Wertstoffrecycling Ka
4:	2831457 HOFHEINZ HOCK
5:	2831201 KOST

Abbildung 2: Beispiel einer Darstellungsseite ohne Hintergrundkarte (Ausschnitt)

2.3 Der Stopdämon

Das Hypertext-Transfer-Protocol (HTTP) ist ein zustandsloses Protokoll. Durch diese Tatsache bekommt der Server den Abbruch (STOP-Taste oder BACK-Taste des Browsers) einer gerade laufenden Interaktion des Client mit dem Server erst mit, wenn die Daten vom Server an den Client übertragen werden. In der Zwischenzeit ist die Sitzung für weitere Interaktion

mit dem Server gesperrt. Diese Sperrung der Sitzung während einer Interaktion ist notwendig, damit keine weiteren Anfragen der gleichen Sitzung an den Server geleitet werden können.

Leider ist es nicht möglich, die Verbindung zwischen Server und Client zu belasten, da zum Teil aufwendige Prozesse wie z. B. GRASS, Gnuplot, Gifdraw aufgerufen werden. Daher ist eine zusätzliche Aktivität notwendig, die in dem Zeitraum, in dem vom CGI-Script keine Ausgaben kommen, die die Verbindung belastet. Dieser Prozeß, genannt Stopdämon, sendet vom Server dem Client pro Sekunde ein Leerzeichen. Sobald der Stopdämon den Abriß der Verbindung zwischen Server und Client (Browser) bemerkt, schickt er der ganzen Prozeßgruppe ein Signal. Daraufhin reagiert der Hauptprozeß der Graphischen Dienste mit Entsperren der Datenbasis und Beendigung seiner Aktivität.

Da es bis zu einer Sekunde dauern kann, bis der Stopdämon die Betätigung der Stoptaste am Browser erkennt, und auch das Beenden der Prozesse sowie das Entsperren der Datenbasis eine gewisse Zeit beanspruchen, wurde die Prozedur, die die Datenbasis sperrt, bzw. testet, ob eine Aktivität innerhalb dieser Sitzung stattfindet, geändert. Die Reaktion auf eine schon gesperrte Sitzung wird nun um ca. 3 Sekunden verzögert, um dem Stopdämon die Möglichkeit zu geben, die laufenden Prozesse und die Lock-Datei zu entfernen. Dieses ist ein gutes Maß für UNIX-Betriebssysteme auf herunterfahrende Prozesse zu warten.

2.4 Modularisierung der Graphischen Dienste

Um eine unproblematische Installation der Graphischen Dienste zu erzielen, wurden die rechnerabhängigen Pfadnamen in der Software der Graphischen Dienste so weit als möglich in einer Datei als Umgebungsvariablen abgelegt. In dieser Datei wurden folgende Umgebungsvariablen definiert:

Umgebungsvariable	Beschreibung
WWWGRAPHIK_GISENV	Name dieser Datei
ORACLE_HOME ORACLE_SID TWO_TASK NLS_LANG	Umgebungsvariable für Datenbank für folgende Tabellen: lfu_ve_verzeichnis, lfu_kartenblatt
WWWGRAPHIK_DBUSER	Datenbank-User für obige Datenbank-Tabellen
WWWGRAPHIK_DBPASSWD	Datenbank-Passwort für WWWGRAPHIK_USER
WWWGRAPHIK_CLIENTCGI	URL des Client-CGIs (z.B. WWW-UDK), welches von Graph. Dienste gestartet werden kann, muß als Interface die Aufrufe implementieren, die dieses versteht. Start des Klienten mit Bounding Box.

WWWGRAPHIK_CLIENTNAME	ausführlicher Name des Client-CGI
WWWGRAPHIK_CLIENTSNAME	Kurzname des Client-CGI
TULISTRENNDEF	Trennzeichen, mit der die Spalten der Daten-datei getrennt sind (Default="DOLLAR")
WWWGRAPHIK_HOME	Pfad des Basisverzeichnisses der Graph. Dienste
WWWGRAPHIK_URL	URL auf das Basisverzeichnis der Graph. Dienste
WWWGRAPHIK_SESSDIR	Verzeichnis für temporäre Dokumente, Bilder Gifs, dieses Verzeichnis muß Lese-, Schreib- und Zugriffsrechte für alle besitzen und über URL zugreifbar sein.
WWWGRAPHIK_SESSURL	URL von WWWGRAPHIK_SESSDIR
WWWGRAPHIK_TCLDIR	Pfadnamen des Verzeichnisses der TCL-Prozeduren der Graph. Dienste
WWWGRAPHIK_ICONURL	URL für Icons und Gifs von WWW-UIS
WWWGRAPHIK_GDICONURL	URL für Icons und Gifs der Graph. Dienste
WWWGRAPHIK_SAVESESS	Pfadname des Verzeichnisses um Sitzungen zu sichern und zu laden; wenn nicht gesetzt besteht keine Möglichkeit Sitzungen zu sichern.
WWWGRAPHIK_ERRDIR	Pfadname des Verzeichnisses für Fehlerbehandlungen
GRASS_DIR	Pfadname der GRASS-Installation
GRASS_DATA	Pfadname der GRASS-Daten
GRASS_LOCATION	Location der GRASS-Daten
GRASS_MAPSETS	Mapsets der GRASS-Daten
WWWGRAPHIK_TOOLS DIR	Verzeichnis, in sich sonstige Programme befinden, die die Graph. Dienste noch benötigen

Tabelle 1: Beschreibung der Umgebungsvariablen

Durch diese Umgebungsvariablen ist es leicht möglich die CGI-Scripten und Programme, die die Graphischen Dienste benötigen, in beliebige Verzeichnisse zu installieren, wobei die in der Liste aufgeführten Voraussetzungen erfüllt sein müssen.

3. Literatur

- /1/ Nikolai,R.; Otterstätter, A.; Weisbrich, W.; Wiesel, J. (1997): Schnittstelle TULIS/MEROS Dienst - Graphische Dienste, (Technischer Bericht)

Vom Pilotprojekt zur Anwendung

Das Altlasten-Fachinformationssystem AlfaWeb

*R. Weidemann, F. Ehrmann, W. Geiger, M. Reißfelder,
Forschungszentrum Karlsruhe GmbH,
Postfach 3640,
76021 Karlsruhe*

*E. Schmid, J. Witt-Hock,
Landesanstalt für Umweltschutz
Baden-Württemberg,
Griesbachstr. 1,
76185 Karlsruhe*

1. EINLEITUNG	155
2. AKTIVITÄTEN ZUR VORBEREITUNG DES EINSATZES	156
2.1 PROJEKTABLAUF	156
2.2 RAHMENKONZEPT	157
3. AUSBAU DES SYSTEMUMFANGS	158
3.1 INHALTLICHER AUSBAU	158
3.1.1 <i>Berichte</i>	158
3.1.2 <i>Programme</i>	159
3.1.2.1 RefAS	159
3.1.2.2 BOBAN	160
3.2 AUSBAU DES ZUGANGSSYSTEMS	161
3.2.1 <i>Benutzerschnittstelle</i>	161
3.2.2 <i>Schlagwortsuche</i>	162
3.2.3 <i>Fachzugang</i>	163
4. CD-ROM-ENTWICKLUNG.....	163
4.1 DYNAMISCHE SEITEN	164
4.1.1 <i>Motivation</i>	164
4.1.2 <i>Installation eines lokalen Servers</i>	164
4.1.3 <i>Einsatz kommerzieller Offline-Browser</i>	164
4.1.4 <i>Erweitern eines Standardbrowsers durch ActiveX-Scripting</i>	165
4.1.5 <i>Erweitern eines Standardbrowsers durch ein Plug-in bzw. ActiveX</i>	165
5. KOMPONENTEN ZUR PFLEGE UND WARTUNG.....	168
5.1 SYSTEMVERWALTUNGSKOMPONENTE	168
5.2 KONVERTIERUNGSPROGRAMM	171
6. AUSBLICK.....	173
7. LITERATUR.....	174

1. Einleitung

Ziel des Vorhabens AlfaWeb (**Altlasten-Fachinformationen im World-Wide Web**) ist es, von der Landesanstalt für Umweltschutz Baden-Württemberg erstellte Arbeitshilfen für eine landeseinheitliche, systematische Altlastenbearbeitung mit den Mitteln moderner Informations- und Kommunikationstechnologien zu erschließen und den Altlasten-Sachbearbeitern über rechnergestützte Navigations- und Zugangshilfen eine effektive Informationsbeschaffung zu ermöglichen.

In den vorhergehenden Projektphasen (/1/, /2/) wurde dazu ein Prototyp entwickelt und im Internet auf dem WWW-Server des Ministeriums für Umwelt und Verkehr Baden-Württemberg bereitgestellt (<http://www.uis-extern.um.bwl.de/lfu/abt5/altlasten/>). Die weiteren, in diesem Beitrag beschriebenen Arbeiten im Rahmen GLOBUS IV sollten den Übergang vom Pilotprojekt in die Anwendung vorbereiten. Dazu wurden 3 Aufgabenbereiche definiert:

1. **Konsolidierung von AlfaWeb**

Der Ausbau und die Abrundung vorhandener Inhalte und Funktionen unter Einbeziehung der künftigen Nutzer des Systems (siehe Kap. 2.1) soll sicherstellen, daß das Arbeiten mit AlfaWeb für den Altlasten-Sachbearbeiter attraktiv und lohnend erscheint. Schwerpunktthemen sind hier:

- Überarbeitung der Benutzerschnittstelle (siehe Kap. 3.2.1)
- Neukonzeption der Schlagwortsuche (siehe Kap. 3.2.2)
- Inhaltlicher Ausbau durch Integration weiterer Berichte und Programme (siehe Kap. 3.1)

2. **Komponenten zur Verwaltung und Pflege von AlfaWeb**

Diese Software-Komponenten sollen den AlfaWeb-Systemverwalter in die Lage versetzen, ohne Programmierkenntnisse neu erscheinende Berichte in die Gesamtstruktur des Systems zu integrieren. Voraussetzung dafür sind ein geeignetes Ablaufkonzept (siehe Kap. 2.2) und die Bereitstellung entsprechender Werkzeuge wie z.B.

- die Systemverwaltungskomponente (siehe Kap. 5.1)
- das Konvertierungsprogramm (siehe Kap. 5.2)

3. **CD-ROM Version**

Besonders wegen der bisher nicht flächendeckenden Rechner-Vernetzung in der Verwaltung, aber auch wegen oft unzureichender Übertragungsraten im Internet ist kurzfristig als primäres Medium für den Einsatz von AlfaWeb die CD-ROM vorgesehen. Um später einen gleitenden Übergang ins Internet zu ermöglichen (siehe Kap. 2.2 und 6) und Doppelarbeit zu vermeiden, sollen Netz- und CD-ROM-Version von AlfaWeb möglichst wenig von einander abweichen. Die Veröffentlichung eines WWW-Informationssystems auf CD-ROM bereitet jedoch bzgl. der dynamischen Komponenten besondere Probleme (siehe Kap. 4).

2. Aktivitäten zur Vorbereitung des Einsatzes

2.1 Projektablauf

Die AlfaWeb-Entwicklung war von Anfang an durch den starken Anwendungsbezug geprägt, der durch die unmittelbare Beteiligung der zuständigen Fachabteilung der Landesanstalt für Umweltschutz Baden-Württemberg (LfU) gewährleistet wurde. Um Fehlentwicklungen zu vermeiden und einen nutzbringenden Einsatz von AlfaWeb bei der Altlastenbearbeitung zu ermöglichen, wurden im zurückliegenden Jahr die zukünftigen Nutzer noch stärker in die Entwicklung einbezogen.

In einem ersten Schritt wurde im Januar 1997 eine Test-CD-ROM produziert. Diese enthielt alle bis zu diesem Zeitpunkt in AlfaWeb integrierten Berichte, dasselbe Zugangssystem wie die Netzversion, jedoch keine dynamischen Zugriffsmöglichkeiten (siehe Kap. 4). Die Test-CD-ROMs wurden gezielt an Personen innerhalb zweier Nutzergruppen verteilt:

1. Verwaltung (Regierungspräsidien, Landratsämter, Kommunen) und Forschung (VEGAS)
2. Ingenieurbüros

Absicht war, die potentiellen Nutzer auf die Möglichkeiten der neuen Medien aufmerksam zu machen und Anstöße zur Formulierung eigener Anforderungen zu liefern. In einem Workshop am 27.02.97 wurden die Erfahrungen und Ideen der Test-Nutzer zusammengetragen. Auf dieser Basis wurden von der LfU vier Ingenieurbüros aufgefordert, einen Vorschlag für den weiteren fachlichen Ausbau von AlfaWeb zu erarbeiten und ein entsprechendes Angebot abzugeben. Die Nutzergruppe 1 hingegen wurde gebeten, in einer Projektsteuerungsgruppe mitzuarbeiten, welche die weitere Entwicklung aus Anwendersicht begleitet.

In der ersten Sitzung der Projektsteuerungsgruppe am 17.03.97 präsentierten die Ingenieurbüros ihre Angebote. Die Vergabe auf Empfehlung der Projektsteuerungsgruppe erfolgte an die Arbeitsgemeinschaft der Firmen IUT und AHK. Diese identifizierten als vorrangiges Projektziel die Entwicklung einer CD-ROM-Version mit voller Funktionalität und als zusätzlichen Inhalt Druckvorlagen für alle Berichte. Als wichtigste Arbeitspunkte wurden grob die folgenden, nach Prioritäten geordneten Themen genannt:

- fachspezifische Verschlagwortung der Berichte
- Erweiterung und teilweise Neukonzeption des Fachzugangs
- Aufnahme weiterer Inhalte
- Maßnahmen zur Erhöhung der Benutzerfreundlichkeit
- Hotline, Nutzerbetreuung

In der Folge wurde das Systemkonzept von AlfaWeb überarbeitet (siehe Kap. 2.2) und eine detailliertere Spezifikation der Entwicklungsziele und der Aufgabenverteilung vorgenommen. In der AlfaWeb-Weiterentwicklung ist IUT für die fachliche Aufarbeitung der Altlasten-Informationen zuständig, während AHK die Pflege und Wartung des Systems und die Produktion der CD-ROM übernimmt. Das Forschungszentrum Karlsruhe ist für die informationstechnische Weiterentwicklung verantwortlich. Die Projektsteuerungsgruppe überwacht die

Arbeiten und berät die LfU, welche als Auftraggeber die Gesamtverantwortung und die Koordination inne hat.

2.2 Rahmenkonzept

Die Entwicklung von AlfaWeb geht davon aus, daß in den nächsten Jahren mehrere Medien gleichzeitig, aber mit sich verschiebender Gewichtung unterstützt werden müssen:

- Die CD-ROM Version ist kurzfristig das wichtigste Verbreitungsmedium zu den Altlastenbearbeitern in Verwaltung und Ingenieurbüros.
- Für die LfU und die Verwaltungseinheiten, die Zugang zum Intranet der LfU haben, wird eine vollständige und jeweils aktuelle Version von AlfaWeb auf einem LfU-Server bereitgestellt.
- Im Internet wird mit dem Erscheinen der kostenpflichtigen CD-ROM-Version der AlfaWeb-Prototyp (s. Kap. 1) durch ein "Schaufenster-System" abgelöst, das sich auf Übersichtsinformationen (z.B. Kurzfassungen der Berichte) beschränkt.

Mittelfristig soll auch das vollständige System im Internet zur Verfügung stehen. Dies erfordert jedoch die Einrichtung eines Abrechnungssystems, da die LfU gehalten ist, für Ihre Dienstleistungen Gebühren zu erheben. Mit zunehmender Vernetzung sollte die CD-ROM-Version durch die Netz-Versionen im Internet bzw. Intranet abgelöst werden.

Um den Aufwand zur Entwicklung, Pflege und Wartung des Systems in Bezug auf die unterstützten Medien zu minimieren, wurden verschiedene Maßnahmen ergriffen. Es wurde ein Vorgehensmodell entwickelt, das zum einen die Anforderungen an die Berichtsersteller - die Lieferanten der Fachberichte und Handbücher - gering hält, zum anderen jedoch die technischen Randbedingungen für die Veröffentlichung mittels verschiedener Medien und für die Integration in das Gesamtsystem erfüllt. Die Vorgehensweise wurde bereits an anderer Stelle detailliert beschrieben (/1/, /2/, /3/). Abbildung 1 zeigt in vereinfachter Form¹ das Grundprinzip, einschließlich der Modifikationen, die sich im Projektverlauf (siehe Kap. 2.1) ergeben haben.

Die Berichtserstellung erfolgt mit einem Standard-Textverarbeitungssystem (Microsoft Word) unter Zuhilfenahme einer Dokumentenvorlage und einer Gestaltungsrichtlinie. Querverweise innerhalb des Dokuments, aber auch auf andere in AlfaWeb enthaltene Berichte werden im WinWord-Dokument eingetragen. Neu ist, daß die Schlagworte entsprechend der fachspezifischen Verschlagwortung (s. Kap. 3.2.2) ebenso direkt ins WinWord-Dokument aufgenommen werden. Das WinWord-Dokument (im RTF-Format) ist Basis der weiteren Bearbeitung. Dabei werden zum einen über ein Konvertierungsprogramm die WWW-Seiten generiert, zum anderen - und dies ist wiederum neu - mittels WinWord und Adobe Acrobat Druckvorlagen im PDF-Format erzeugt. Die PDF-Dateien sind für die CD-ROM-Version bestimmt und sollen es dem Nutzer ermöglichen, bei Bedarf bestimmte Berichte in beliebiger Anzahl auszudrucken² und damit Aufwand und Kosten der LfU für den Druck von Berichten reduzieren zu helfen. Gleichzeitig gewährleistet das gewählte Format konsistente Gliederung und Aussehen der gedruckten Berichte.

¹ Nicht in der Abbildung wiedergegeben ist z.B. die Bereitstellung und Einbindung von Abbildungen oder die Extraktion und Weiterverarbeitung der Schlagworte.

² Dazu wird das kostenlos erhältliche Softwareprodukt Adobe Acrobat Reader benötigt.

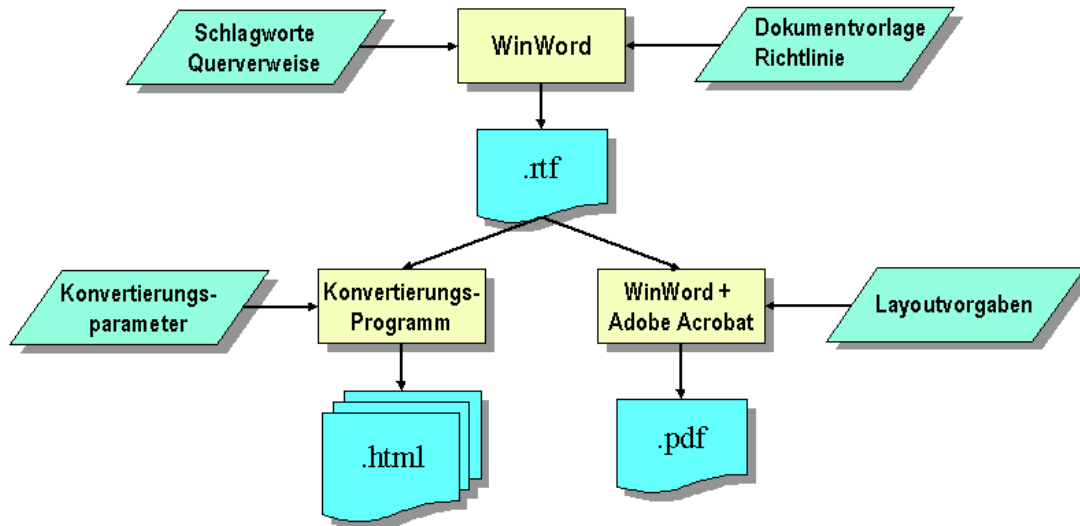


Abbildung 1: Erstellung von Berichten für AlfaWeb (Prinzipialskizze)

Eine weitere Maßnahme zur Minimierung des Pflege- und Wartungsaufwands bestand in der Vereinheitlichung der Rechnerumgebung. Während beim letztjährigen Prototypen von AlfaWeb zwar die Berichtserstellung am PC erfolgte, lief die eigentliche WWW-Anwendung, ebenso wie das Konvertierungsprogramm unter Unix. Da die CD-ROM-Version entsprechend der typischen Rechnerausstattung des Nutzers eine Windows-Anwendung werden muß, lag es nahe, AlfaWeb insgesamt zu portieren. AlfaWeb läuft inzwischen vollständig unter Windows NT. Auf der Basis eines Pentium PC mit Windows NT 4.0 und dem Internet Information Server von Microsoft wurde im Intranet der LfU ein WWW-Server für AlfaWeb aufgesetzt.

3. Ausbau des Systemumfangs

3.1 Inhaltlicher Ausbau

Der inhaltliche Ausbau von AlfaWeb wurde sowohl im Bereich der enthaltenen Fachberichte und Handbücher als auch der Programme fortgesetzt.

3.1.1 Berichte

Neu erschienene Berichte in den Berichtsreihen "Materialien zur Altlastenbearbeitung" und "Texte und Berichte zur Altlastenbearbeitung" der LfU wurden unmittelbar in AlfaWeb aufgenommen und noch vorhandene Lücken in den Berichtsreihen geschlossen. Außer 5 veralteten Berichten, die nicht für die Aufnahme in AlfaWeb vorgesehen sind, sind die Berichtsreihen inzwischen vollständig enthalten. Daneben wurden eine Reihe sonstiger, für die Altlastenbearbeitung relevanter Berichte aufgenommen. Die folgende Tabelle zeigt den Ausbaustand von AlfaWeb.

Berichtsreihen	Ist-Stand 23.11.95	Ist-Stand 07.11.96	Ist-Stand 31.10.97
Materialien zur Altlastenbearbeitung	4	21	25
Texte und Berichte zur Altlastenbearbeitung	4	18	29
Sonstige Berichte	2	3	9
Summe:	10	42	63

Tabelle 1: Altlastenberichte der LfU in AlfaWeb

Die Verschiebung der wichtigsten Nutzungsmöglichkeit von der Netz- zur CD-ROM-Version und durch die neu hinzugekommene Anforderung, alle Berichte auch zusätzlich als Druckvorlagen bereitzustellen, erforderten eine Überarbeitung auch der vorhandenen Berichte:

1. **Qualität der eingescannten Abbildungen**
Die überwiegende Zahl der z.Zt. 2400 Abbildungen in AlfaWeb wurden von Papiervorlagen eingescannt. In Abwägung zwischen Dateigröße (Übertragungszeit) und Qualität der Abbildung wurden ursprünglich eher Abstriche an der Qualität akzeptiert, um die Übertragungszeiten im Netz zu minimieren. Im Hinblick auf die CD-ROM ist die Dateigröße jedoch nicht das entscheidende Argument. Daher wurde damit begonnen, die Qualität der Abbildungen durch Neueinscannen zu verbessern.
2. **Aufbereitung der Berichte**
Um die WinWord-Dokumente auch für die Erzeugung der Druckvorlagen verwenden zu können, mußten diese in einheitlicher Weise aufbereitet und vervollständigt werden.

Die zuletzt genannten Arbeiten werden derzeit von studentischen Hilfskräften erledigt.

3.1.2 Programme

Neben Handbüchern und Fachberichten gewinnen zunehmend PC-Programme als Arbeitshilfen in der Altlastenbearbeitung an Bedeutung. Es ist in jedem Einzelfall zu prüfen, ob eine Integration in AlfaWeb aus fachlicher und informationstechnischer Sicht sinnvoll und vom Aufwand her gerechtfertigt ist. Derzeit sind die beiden nachfolgend beschriebenen Programme in AlfaWeb integriert, d.h. die zugehörigen Daten(-blätter) können sowohl über eine spezielle Suchmaske ermittelt werden, sind aber prinzipiell auch über die anderen AlfaWeb-Zugangswege auffindbar, wobei derzeit der Zugang über die Volltextsuche realisiert ist.

3.1.2.1 RefAS

Der **Referenzkatalog Altlasten / Schadensfallsanierung (RefAS)** enthält Literaturangaben zu durchgeführten Sanierungsmaßnahmen. Das System wird von der LfU als eigenständiges PC-Programm (Basis: Microsoft Access) vertrieben und ist auch mit vollständigem Inhalt in AlfaWeb enthalten. Version 1.0 wurde bereits im letzten Jahr in AlfaWeb integriert /2/. Die aktuelle Version 1.1, die wesentlich mehr Fälle enthält, wurde in 1997 integriert. Wegen struktureller Änderungen an der Datenbank des PC-Programms mußte AlfaWeb entsprechend angepaßt werden.

3.1.2.2 BOBAN

Die Datenbank **Bodenbehandlungsanlagen (BOBAN)** enthält Datenblätter zu stationären und mobilen Bodenbehandlungsanlagen. Neben Standort und Betreiber werden Informationen über die Kapazität der Anlage, die behandelbaren Verunreinigungen, Kosten und Verbleib des Materials aufgeführt. Über ein Formular können Datenblätter gezielt gesucht werden.

Abbildung 2: Suchformular BOBAN

Das System basiert auf dem Bericht "Bodenbehandlungsanlagen in der Bundesrepublik Deutschland" (Texte und Berichte zur Altlastenbearbeitung, Band 32/97) und ist exklusiv in AlfaWeb enthalten, wird also nicht als separates PC-Programm vertrieben.

Der Datenbestand wird in einer Microsoft Access Datenbank verwaltet und gepflegt. Über ein Generierungsprogramm werden für AlfaWeb die Datenblätter als statische HTML-Seiten erzeugt, Indexfiles für die Suche angelegt und die Auswahlmenüs des Suchformulars aktualisiert. Abfragen (Formular siehe Abbildung 2, rechte Seite) werten über ein Perl-Programm die Indexfiles aus und erzeugen Verweise auf die statischen HTML-Seiten (siehe Abbildung 2, linke Seite). Die statischen Seiten sind ohne zusätzlichen Aufwand in Volltextsuchen eingebunden, es können aber auch direkte Verweise z.B. aus dem Fachzugang (siehe Kap. 3.2.3) angelegt werden.

Die Pflege der Datenbasis soll mittelfristig durch die Betreiber von Bodenbehandlungsanlagen selbst erfolgen. Dazu wird derzeit ein Erfassungsprogramm entwickelt. Aus der Access-Datenbank heraus werden dabei Runtime-Module generiert, die jeweils die Daten genau eines

Betreibers enthalten. Diese Runtime-Module sollen dem jeweiligen Betreiber zugeschickt werden (Email oder Diskette). Nach Änderung der Daten und Rücklieferung an die LfU kann die gemeinsame Datenbasis aktualisiert werden.

3.2 Ausbau des Zugangssystems

Die Benutzeroberfläche von AlfaWeb und die Zugangsmöglichkeiten (hierarchischer Zugang, Volltextsuche, Schlagwortsuche, Fachzugang) wurden entsprechend den sich wandelnden Möglichkeiten der WWW-Sprache HTML und den Anforderungen und Ideen von Seiten der künftigen Nutzer (siehe Kap. 2.1) weiterentwickelt.

3.2.1 Benutzerschnittstelle

Um beim Navigieren innerhalb der Berichte jederzeit die verschiedenen Zugangswege im direkten Zugriff zu haben, wurde die Benutzeroberfläche mittels Frames zweigeteilt (siehe Abbildung 3).



Abbildung 3: Startseite von AlfaWeb

Die linke Menüleiste bleibt während des Arbeitens mit AlfaWeb fest, die wechselnden Inhalte werden im größeren rechten Teilfenster angezeigt (siehe z.B. Abbildung 9). Lediglich beim Arbeiten mit den in AlfaWeb enthaltenen Programmen wird die Menüleiste durch eine programm-spezifische Leiste ersetzt (siehe Abbildung 2).

3.2.2 Schlagwortsuche

Im Anwendungstest hatte es sich erwiesen, daß der ursprünglich verwendete Thesaurus umweltrelevanter Begriffe des Umweltbundesamts für den Altlastenbereich zum einen zu breit angelegt, zum anderen im engeren Fachbereich nicht tief genug ist. Außerdem wird von Seiten der Anwender eine automatische Verschlagwortung, wie sie anfangs für AlfaWeb durchgeführt wurde, als nicht zweckmäßig angesehen.

Da auf absehbare Zeit kein altlastenspezifischer Thesaurus als Basis für die Schlagwortsuche zur Verfügung steht, muß die Schlagwortliste für AlfaWeb von Grund auf neu aufgebaut werden. Dazu werden von Fachexperten die einzelnen Berichte separat, aber soweit möglich in konsistenter Weise verschlagwortet, indem die Berichte inhaltlich analysiert, relevante Textstellen identifiziert und treffende Fachtermini zugewiesen werden. Die Verschlagwortung erfolgt durch das Ingenieurbüro IUT (siehe Kap. 2.1). Die Fachbegriffe werden als Indexeinträge in das WinWord-Dokument eingetragen. Für das Papierdokument (Druckvorlage) kann daraus ein Indexverzeichnis generiert werden. Nach der Konvertierung des Berichts in WWW-Seiten (siehe Kap. 5.2) wird auch hier ein Indexverzeichnis zu dem einzelnen Bericht erzeugt, und außerdem ein Gesamtschlagwortverzeichnis³ generiert. Abbildung 4 zeigt die Benutzerschnittstelle der Schlagwortsuche.

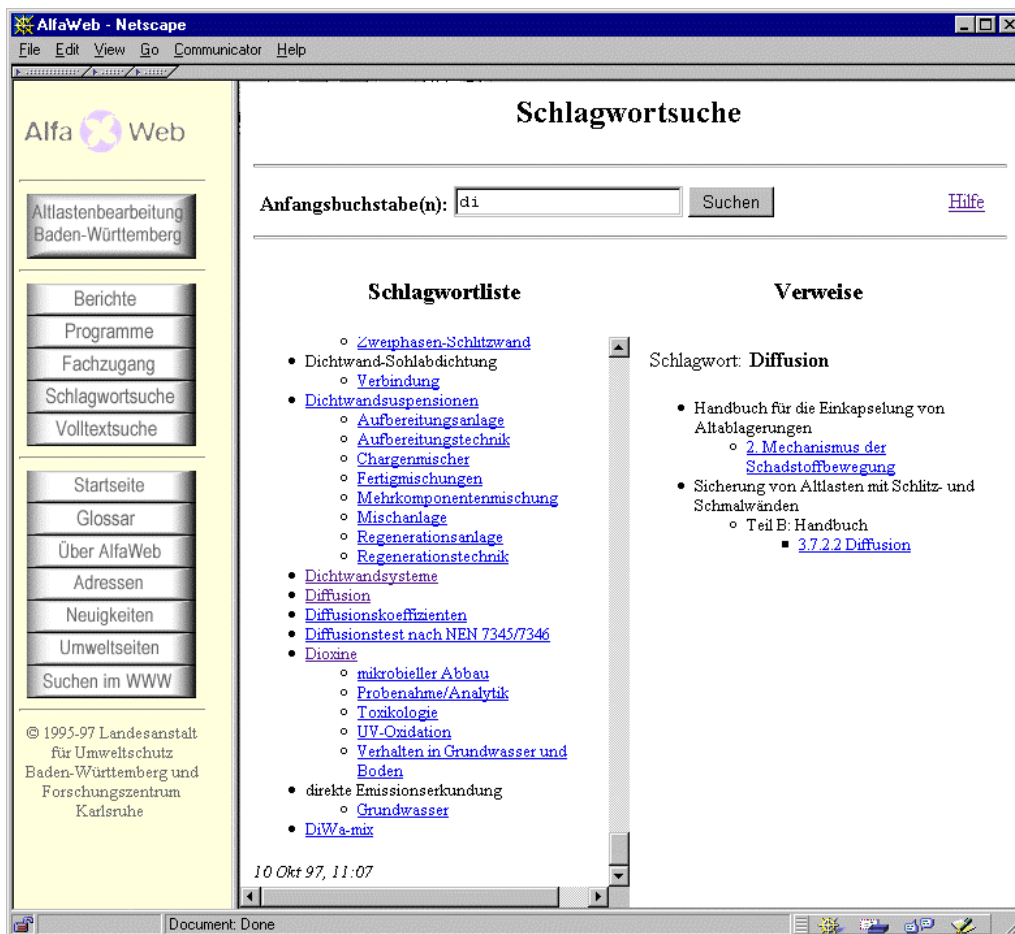


Abbildung 4: Beispiel für Schlagwortsuche

³ Es ist angedacht, das Gesamtverzeichnis als Basis für die Entwicklung eines altlastenspezifischen Thesaurus zu verwenden.

Die Schlagwortsuche erfolgt in drei Schritten:

1. Durch Angabe des gewünschten Wortanfangs wird eine Teilmenge der Gesamtschlagwortliste ausgewählt und auf der linken Seite im Suchfenster angezeigt. In Abbildung 4 ist erkennbar, daß für AlfaWeb eine zweistufige Begriffshierarchie gewählt wurde, d.h. man unterscheidet zwischen Ober- und Unterbegriffen.
2. Nach Auswahl eines Ober- bzw. Unterbegriffs werden die zugehörigen Verweise auf der rechten Seite aufgeführt. Pro Verweis ist angegeben:
 - Titel des Berichts
 - Berichtsteil, falls vorhanden
 - Kapitel, innerhalb dessen das Schlagwort vergeben wurde.
3. Die Kapitelbezeichnung ist maus-sensitiv und blättert die entsprechende Berichtsstelle auf. Die relevante Textstelle ist i.d.R. besonders hervorgehoben (siehe Kap. 5.2, Abbildung 9).

3.2.3 Fachzugang

Der Fachzugang geht von den typischen Vorgängen und Fragestellungen bei der Altlastenbearbeitung aus und stellt dafür geeignete Gliederungsstrukturen bereit, die letztendlich die in AlfaWeb enthaltenen Informationen erschließen. Für ausgewählte Teilgebiete der Altlastenbearbeitung wurde im Rahmen von GLOBUS III /2/ eine Strukturierung vorgenommen und implementiert. Die Erprobung im Zusammenhang mit der Test-CD-ROM (siehe Kap. 2.1) bestätigte den gewählten Ansatz, zeigte aber auch, daß ein rein Menü-orientierter Ansatz bei tieferen Gliederungsstrukturen an die Grenzen der Übersichtlichkeit stoßen.

Als Konsequenz daraus wird der Fachzugang derzeit durch das Ingenieurbüro IUT fachlich überarbeitet und in Breite ausgebaut. Dabei werden die drei Themen Altlastenbearbeitung, Recht und Verfahren als separate Zugangsstrukturen bearbeitet. Zur übersichtlicheren Strukturierung ist insbesondere im ersten Fall der Einsatz von maus-sensitiven Flußdiagrammen (image maps) vorgesehen. Die Realisierung in AlfaWeb erfolgt unmittelbar nach Ende der fachlichen Arbeiten.

4. CD-ROM-Entwicklung

Die Verbreitung von Informationsangeboten mittels CD-ROM hat in den letzten Jahren stark zugenommen. Entsprechend sind eine Vielzahl leistungsfähiger Werkzeuge auf dem Markt, welche die Entwicklung dedizierter CD-ROM-Anwendungen in speziellen Dateiformaten unterstützen. AlfaWeb geht jedoch von einem anderen Ansatz aus. Hier soll ein WWW-Informationssystem, d.h. eine Client-Server-Anwendung, mit möglichst wenigen Anpassungen auf CD-ROM übertragen werden. Dies ist problemlos, wenn server-seitig lediglich ein Dateisystem zur Verfügung gestellt wird (statische HTML-Seite), jedoch schwierig, wenn der Server Informationen erst auf Nachfrage zusammenstellt (dynamische Seiten). Im folgenden werden Alternativen untersucht, wie diese Funktionalität mit auf CD-ROM gebracht werden kann.

4.1 Dynamische Seiten

4.1.1 Motivation

In der WWW-Version von AlfaWeb sind einige dynamische Komponenten enthalten. Diese benötigen einen WWW-Server, der auf Grund von Formulareingaben Ergebnisse ermittelt, welche als HTML-Seite aufbereitet dann an den Browser des Anwenders zurückgegeben werden. Dieser WWW-Server kann auf einer CD-ROM, die auch Anwender ohne Internetanschluß bedienen soll, nicht vorausgesetzt werden. Dynamische Komponenten sind in der jetzigen Version von AlfaWeb:

- Volltextsuche
- Schlagwortsuche
- RefAS
- BOBAN

Da weitere dynamische Komponenten hinzukommen können, wurde nach einer universellen Lösung gesucht, um die CGI-Schnittstelle (Common Gateway Interface) des Servers durch eine lokale Schnittstelle (LocalCGI) zu ersetzen. Hierzu wurden mehrere Ansätze näher untersucht:

- Installation eines lokalen Servers auf jedem Zielrechner
- Einsatz kommerzieller Offline-Browser
- Erweitern eines Standardbrowsers durch ActiveX-Scripting
- Erweitern eines Standardbrowsers durch ein Plug-in bzw. ActiveX-Steuerelement

4.1.2 Installation eines lokalen Servers

Diese Lösung würde ein vollständig identisches Abbild der WWW-Version erlauben, indem jeder Anwender einen eigenen Web-Server installiert, der natürlich alle benötigten Servereigenschaften der auf dem Netz verfügbaren AlfaWeb-Version unterstützen könnte. Es sind jedoch erhebliche Probleme bei der Wartung und Pflege zu erwarten, da es Überschneidungen mit eventuell existierenden Web-Servern bei den Anwendern geben kann. Mit der Installation eines eigenen Web-Servers wäre jeder Anwender auch gleichzeitig Anbieter von Informationen auf seinem Rechner, die eventuell für unbefugte Dritte zu erreichen wären. Wegen der Wartungs- und Sicherheitsbedenken wurde dieser Lösungsweg nicht mehr weiter verfolgt /4/.

4.1.3 Einsatz kommerzieller Offline-Browser

Als zweiter Lösungsweg wurde recherchiert, welche Browser geeignet sind, lokale HTML-Seiten darzustellen sowie lokale CGI-Programme auszuführen. Bei der weltweiten Suche wurden neben Offline-Browsern, die neben der Darstellungsmöglichkeit von HTML-Seiten auch Volltextsuche über das lokale Filesystem besitzen, auch zwei Browser mit lokaler CGI-Schnittstelle gefunden. Diese sind die Produkte

- SPIS Webview

- I-View Professional

Beide Browser sind in der Lage, die Formulareingaben in einem temporären File zwischenspeichern und ein beliebiges Programm, das als Aktion im HTML-Formular angegeben wird, aufzurufen. Das Ergebnis, das die Programme erzeugen, wird wiederum in den Browser geladen und als HTML-Dokument dargestellt. Diese Lösung fordert geringe Änderungen im HTML-Formular gegenüber der Web-Version, sowie geringe Anpassungen bei der Übergabe der Eingaben und bei der Ausgabe des Ergebnisses, die vorher über Standard-Input bzw. -Output funktionierten. Leider stellen diese Browser keinen Standard dar und besitzen gegenüber den herkömmlichen Internetbrowsern einige Einschränkungen in den Möglichkeiten und in der Bedienung. Daher wurde auch diese Möglichkeit als wenig akzeptabel erachtet /4/.

4.1.4 Erweitern eines Standardbrowsers durch ActiveX-Scripting

Ein Standardbrowser für das Internet ist in der Grundausstattung bezüglich lokaler Zugriffe nur in der Lage, Inhalte lokaler Dateien darzustellen, nicht jedoch, auf das Filesystem zu schreiben und lokale Ressourcen direkt zu nutzen. Mit Erweiterungen, die auf dem Markt verfügbar sind, existiert die Möglichkeit, dieses Problem in gewissem Rahmen zu umgehen. So wird für den Microsoft Internet Explorer eine Erweiterung angeboten, die es zuläßt, ein Dokument, das Anweisungen in der Programmiersprache Perl enthält, zu laden und diese Anweisungen clientseitig, d.h. innerhalb des Browsers, auszuführen. Diese PerlScript-Technik wird beim Internet Explorer ActiveX-Scripting genannt. Mit dieser Erweiterung können somit alle Perl-Programme, die bisher auf der Serverseite ausgeführt wurden, unterstützt werden. Lediglich an der Schnittstelle zwischen dem HTML-Formular und der Ausgabe für den Browser müssen Anpassungen vorgenommen werden. Alle Programme, die nicht in Perl realisiert sind, müssen entweder nach Perl portiert werden oder aber durch eine Schnittstelle in PerlScript allgemein verfügbar gemacht werden. Dieser Lösungsweg unterstützt jedoch nicht den Netscape Navigator. Ein wesentlich größerer Nachteil, der zum Ausschluß dieser Möglichkeit führte, ist die Tatsache, daß ein Anwender, der einen Internetzugang besitzt, auch unbekannte Perl-Dokumente laden kann, die vollen Zugriff auf die lokalen Ressourcen haben. Dies stellt eine Sicherheitslücke dar.

4.1.5 Erweitern eines Standardbrowsers durch ein Plug-in bzw. ActiveX

Von Netscape wurde eine Schnittstelle definiert, die es erlaubt, eigene Ergänzungen zum Web-Browser zu implementieren. Dieses Netscape API (Application Programming Interface) bietet Templates an, auf die der Entwickler aufbauen kann. Weiterhin werden API-Funktionen angeboten, die innerhalb dieser Plug-ins verwendet werden können, um mit dem Browser Informationen austauschen zu können. Diese Plug-in-Architektur wird von beiden führenden Herstellern unterstützt, d.h. vom Netscape Navigator seit der Version 2.0 und vom Microsoft Internet Explorer ab der Version 3.0.

Seit der Version 3.0 des Netscape Navigators wurde eine weitere Architektur von Netscape entwickelt, die es ermöglicht, daß alle drei zur Verfügung stehenden Komponenten HTML-Dokument inkl JavaScript, Java-Applet und Plug-in untereinander kommunizieren können, was im herkömmlichen Plug-in-Modell nicht möglich war. Eine weiterer untersuchter Ansatz

ist der Weg über ein ActiveX-Steuerelement. Mit ActiveX können Benutzeraktionen für Web-Anwendungen vereinfacht werden.

Architekturmodell

Trotz der unterschiedlichen Ansätze wird allen vorgeschlagenen Ansätzen ein einheitliches Architekturmodell zu Grunde gelegt. Das Modell sieht vor, daß beim Selektieren eines Knopfes innerhalb eines Formulars der Auftrag nicht, wie im WWW üblich, an einen Server übergeben wird, sondern daß die Anforderung lokal behandelt wird. Hierzu wird grundsätzlich ein Plug-in oder ein ActiveX-Steuerelement (ActiveX-Control) benötigt, dem die notwendigen Informationen zu den Eingaben übergeben werden.

Die Aufgabe des Plug-ins besteht in folgenden Punkten:

1. Ermitteln, welche Aktion tatsächlich ausgeführt werden soll; dadurch ist ein flexibles Reagieren auf unterschiedliche Installationen von Programmen der Anwender möglich.
2. Übergeben/Halten der Eingaben
3. Aufruf des CGI-Scripts/Programms
4. Laden des Ergebnisses in den Browser

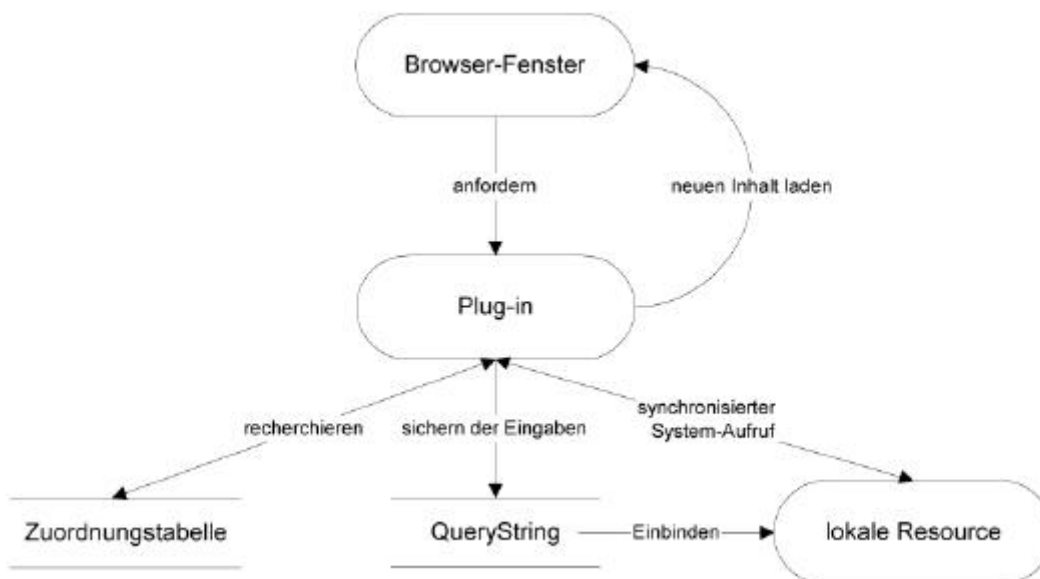


Abbildung 5: Architekturmodell einer lokalen CGI-Schnittstelle mittels Plug-In

Der Unterschied zwischen den verschiedenen Ansätzen besteht nur im Aufruf und in der Ermittlung bzw. Übergabe der Daten vom HTML-Formular hin zum Plug-in bzw. zum ActiveX-Steuerelement und umgekehrt. Im Folgenden werden die verschiedenen Ansätze getrennt betrachtet:

Full-Page Plug-in

Bei einem Full-Page Plug-in wird der CGI-Aufruf über das Netz ersetzt durch den Plug-in-Aufruf. Wird die GET-Methode verwendet, so werden alle Parameter durch den Browser

selbst ermittelt und mit der URL an das Plug-in übertragen. Das Plug-in wird in eine neue Seite geladen, führt an Hand der übergebenen Parameter die geforderte Aktion aus und lädt das Ergebnis als neue Seite in den Browser. Für die POST-Methode wurde keine geeignete Lösung gefunden.

Embedded Plug-in

Bei diesem Lösungsweg wird eine HTML-Seite dargestellt, die um JavaScript-Funktionen erweitert ist. Diese Funktionen dienen dazu, die Eingaben zusammenzufassen und eine HTML-Seite mit einem eingebetteten Plug-in vorzubereiten und über einen Stream an ein quasi verstecktes Frame zu senden. Dieses erzeugte Dokument initialisiert beim Laden das Plug-in und führt dabei den Auftrag - das CGI-Programm - aus und stellt das Ergebnis in einem spezifizierten Frame dar.

LiveConnect

Die LiveConnect-Architektur erlaubt die Kommunikation zwischen einem HTML-Dokument über JavaScript und einem Plug-in. Die Idee hierbei ist, daß in JavaScript der Auftrag an ein eingebettetes Plug-in gegeben wird und alle notwendigen Parameter mit dem Funktionsaufruf übergeben werden. Das eingebettete Plug-in ist zuständig für die Ausführung des Auftrags und für die Weitergabe des Ergebnisses an den Browser, wie es bei den anderen bereits besprochenen Plug-in-Varianten bereits vorgestellt wurde.

ActiveX-Steuerelement

Der Weg über ein ActiveX-Steuerelement ist vergleichbar mit der LiveConnect-Variante, die mit dem Netscape Navigator möglich ist. Hier werden über eine Scriptsprache alle Werte der Eingabefelder ermittelt und der Auftrag an das ActiveX-Steuerelement gegeben. Das Steuerelement ist zuständig für die Ausführung der Aktion und ruft zur Darstellung des Ergebnisses in der auftraggebenden Seite eine Scripting-Funktion auf, die für die Darstellung des Ergebnisses verantwortlich ist.

Die folgende Tabelle stellt die wichtigsten Eigenschaften der beschriebenen Ansätze gegenüber:

	Full-Page Plug-in	Embedded Plug-in	LiveConnect	ActiveX-Steuerelement
einsetzbar mit Netscape Navigator	Ab Version 2.0	Ab Version 2.0	Ab Version 3.0	Mit zusätzlichem ActiveX-Plug-in ab Version 2.0
einsetzbar mit Microsoft Internet Explorer	--	Ab Version 3.0	--	Ab Version 3.0
benötigt Java	Nein	Nein	Ja	Nein
benötigt JavaScript/VBScript	Nein	Ja	Ja	Ja
Anmerkungen	- Probleme mit Back-Button - Netscape-Bug	- Verstecktes Frame notwendig	--	--

Sicherheit

Bezüglich der Sicherheit ist im hier beschriebenen Zusammenhang die Frage relevant, ob durch die vorgestellte Architektur und Technik der Anwender beim Laden eines HTML-Dokuments ein zusätzliches Risiko eingeht, daß auf seinem Rechner Daten zerstört oder ausgespäht werden. Dies ist grundsätzlich möglich, wenn ein HTML-Dokument ein "verseuchtes" CGI-Programm ausführt oder wenn eine Sprache oder Technik eingesetzt wird, die selbst Sicherheitsrisiken birgt.

Die vorgestellte Technik wird nur auf der CD-ROM-Version von AlfaWeb zum Einsatz kommen, d.h. ist nur für den lokalen Einsatz vorgesehen und somit einem Angriff von außen (über das Netz) nicht unmittelbar ausgesetzt. Dies schließt jedoch nicht aus, daß der Anwender doch einen Internetanschluß besitzt und sich potentiellen Gefahren aussetzt, wenn er sich nach einer AlfaWeb-Sitzung mit demselben Browser und unveränderten Voreinstellungen auf unbekannte Seiten im Netz begibt.

Der Aufruf eines unbekanntes, eventuell böartigen CGI-Programms wird jedoch dadurch ausgeschlossen, daß nur in einer Zuordnungsdatei aufgeführte, d.h. registrierte Programme erkannt und ausgeführt werden. Dies wird dadurch gewährleistet, daß Programme im Dokument nicht direkt aufgerufen werden - wie es in einem Offline-Browser geschieht - sondern lediglich Identifier (Bezeichner) angegeben sind. Diese Identifier werden erst in einer Zuordnungsdatei Programmaufrufen zugeordnet.

Die Sicherheit des Anwenders hängt im weiteren davon ab, welche Einstellungen er in seinem Browser vornimmt. Beim Internet Explorer ist es ab Version 4 möglich, ActiveX-Steuerelemente nur dann zuzulassen, wenn diese vom lokalen System oder vom Intranet stammen.

5. Komponenten zur Pflege und Wartung

5.1 Systemverwaltungskomponente

Mit zunehmender Anzahl von Berichten in AlfaWeb steigt auch der Arbeitsaufwand, diese zu verwalten. Um den notwendigen Arbeitsaufwand bei der Verwaltung der Berichte und Berichtsdaten in Grenzen zu halten, bietet sich die Verwendung einer Datenbank an. Hier setzt die Systemverwaltungskomponente an.

Mit Hilfe einer benutzerfreundlichen Oberfläche soll dem Systemverwalter die Möglichkeit gegeben werden, die Berichte, insbesondere die Berichtsdaten (bibliographische Angaben, Konvertierungsparameter, Querverweise, Strukturinformationen), auf bequeme Weise zu verwalten. Dabei hat die Systemverwaltungskomponente folgende Aufgaben:

- Formularbasierte Ein- und Ausgabe von Berichtsdaten
- Überwachung der Datenkonsistenz

- Extrahieren von Konvertierungsparametern und Ausführen der Konvertierungsprogramme
- Herauslösen der Metadaten und automatisches Versenden der Daten an den Umweltdatenkatalog (UDK)
- Erzeugen des Literaturverzeichnisses als WWW-Seite. (Diese dient als Zugang zu AlfaWeb über Berichtsreihe und Titel).

Die Realisierung der Systemverwaltungskomponente erfolgt mittels Microsoft Access (Office 97). Access wird in der Fachabteilung der LfU standardmäßig für Datenbankanwendungen im Bürobereich eingesetzt und erlaubt die Erstellung benutzerfreundlicher Oberflächen im gewohnten Windows "Look and Feel".

Auf der Basis der vorhandenen Berichte bzw. der Berichtsdaten wurde ein Datenbankentwurf erstellt und anschließend in Access implementiert.

Abbildung 6 stellt das Entity-Relationship-Modell der Systemverwaltungskomponente dar.

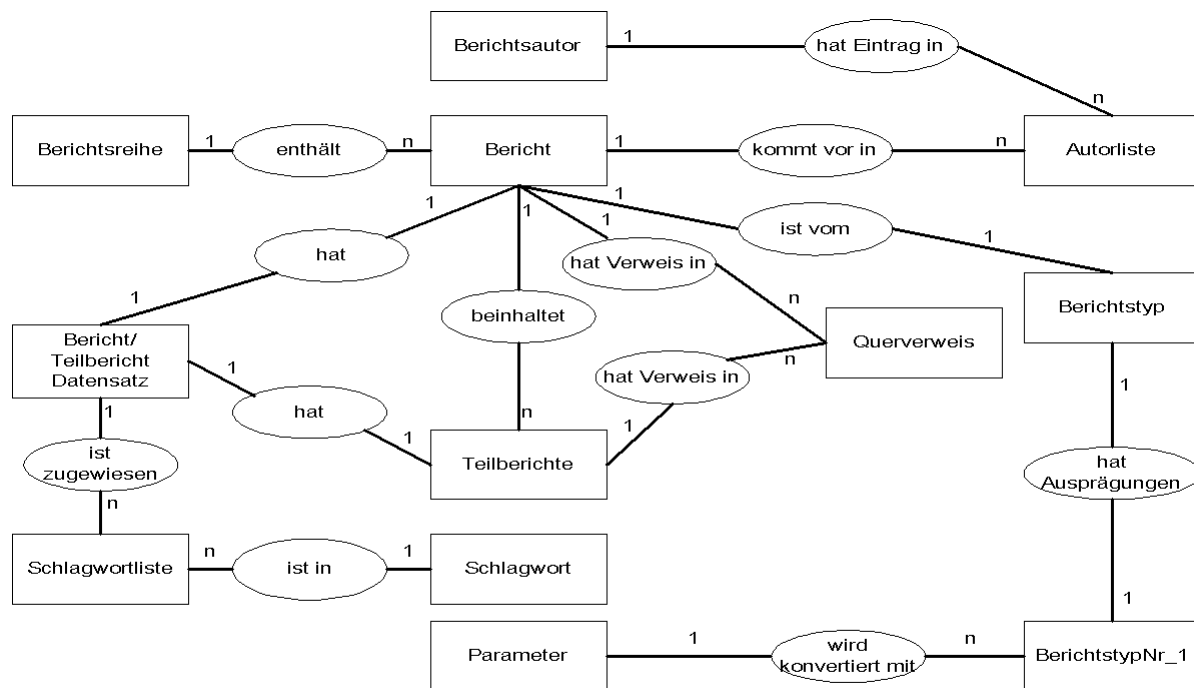


Abbildung 6: Normalisiertes ER-Modell der Systemverwaltungskomponente

Der AlfaWeb-Systemverwalter benötigt keine Datenbankankenntnisse sondern kann die Berichtsdaten über Formulare bearbeiten. Die Prüfung der Datenkonsistenz erfolgt in der Datenbank selbst, wobei als wichtigste Hilfsmittel der Transaktionsmanager und die Prüfung auf referentielle Integrität verwendet werden. Datenbankinterne Fehlermeldungen werden aufgefangen und benutzerfreundlich dargestellt.

Nach erfolgter Dateneingabe können die Berichte in das HTML-Format konvertiert werden. Dazu extrahiert die Systemverwaltungskomponente die erforderlichen Konvertierungsparameter und startet mit diesen die Konvertierungsprogramme. Das Konvertierungsprogramm zerlegt mit Hilfe der Parameter die Berichte in HTML-Dokumente (/1/, /2/).

Abbildung 7 zeigt Ablauf der Berichtskonvertierung.

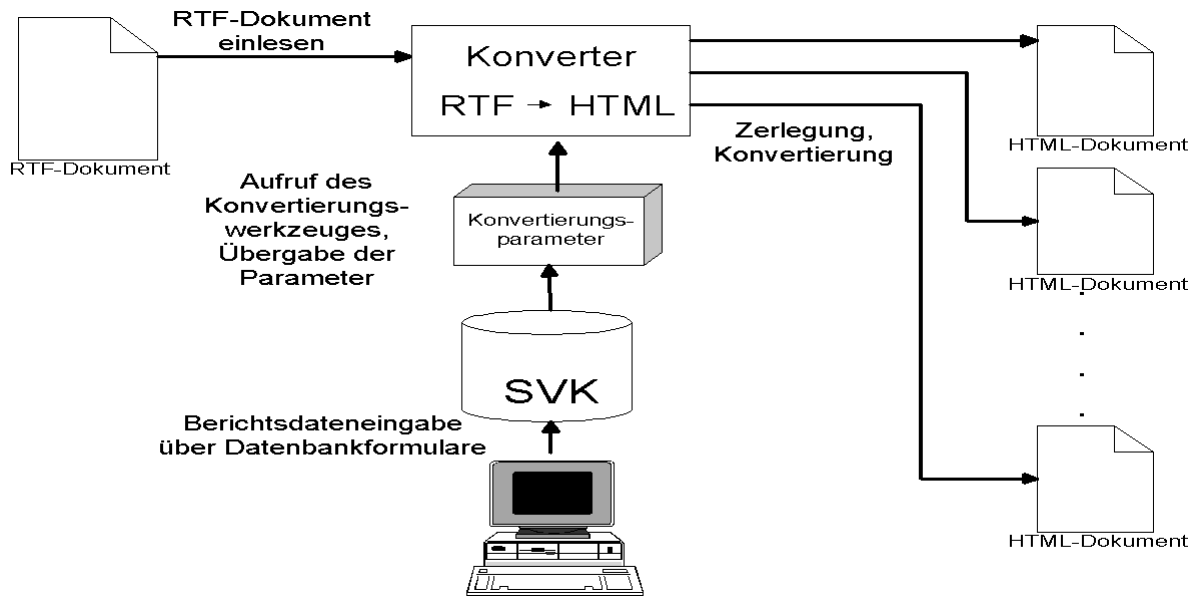


Abbildung 7: Berichtskonvertierung mit Hilfe der Systemverwaltungskomponente (SVK)

Weiter prüft die Systemverwaltungskomponente die Konsistenz der Verweise zwischen den in der Datenbank enthaltenen Berichten. Dadurch werden sogenannte „tote Links“⁴ aufgedeckt und beseitigt.

Aufbauend auf den Tabellen der Access-Datenbank wurden Formulare entwickelt, die zur Aus- und Eingabe der Berichtsdaten dienen. Dabei wurde folgende Vorgehensweise gewählt:

Von einem Zentralformular aus kann der Systemverwalter entscheiden, ob er:

1. einen neuen Berichtsdatensatz eingeben,
2. einen vorhandenen Berichtsdatensatz bearbeiten,
3. einen Bericht konvertieren oder
4. einen Berichtsdatensatz löschen will

In den Punkten 1 und 2 werden nacheinander Formular aufgerufen, die auf den Datentabellen basieren und in die der Systemverwalter die Berichtsdaten eingeben bzw. in denen er die Berichtsdaten verändern kann. Nach erfolgtem Durchlauf erscheint das Zentralformular wieder auf dem Bildschirm. Vom Zentralformular aus kann der Bericht dann z.B. konvertiert werden.

In der Abbildung 8 ist das Zentralformular zu sehen.

⁴ Verweise, die auf kein existierendes HTML-Dokument zeigen.

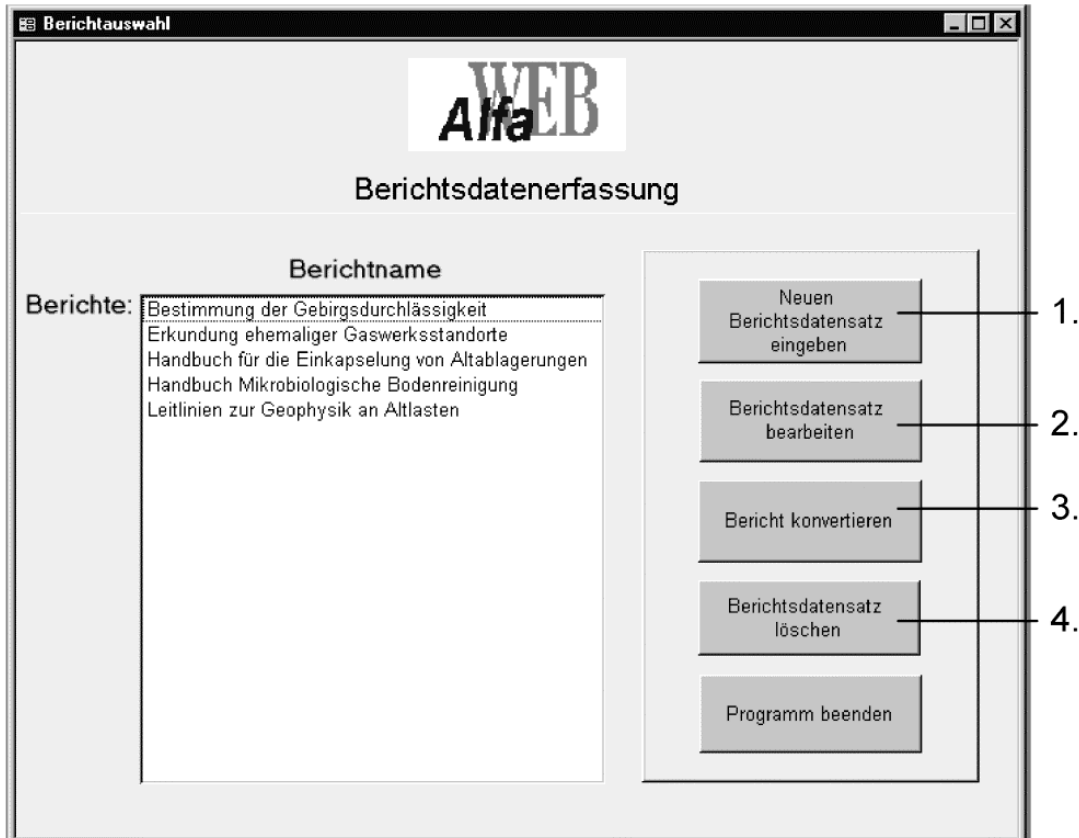


Abbildung 8: Zentralformular mit einigen Berichten und Auswahloptionen

Die Ein- und Ausgabeformulare für die meisten Datentabellen wurden inzwischen erstellt und befinden sich in der Testphase. Im weiteren Verlauf der Entwicklung der Systemverwaltungskomponente sollen die Konvertierungsprogramme an die Datenbankapplikation angeschlossen werden.

5.2 Konvertierungsprogramm

Aufgabe des Konvertierungsprogramms ist die Zerlegung eines im RTF-Format gespeicherten Berichts entsprechend dessen Gliederungsstruktur, die Umwandlung in das HTML-Format und die Vernetzung der erzeugten Berichtsseiten. Strukturelle Verweise werden am Anfang einer Berichtseite in einer Steuerleiste zusammengefaßt. Inhaltliche Querverweise können an jeder Stelle innerhalb des Berichts erscheinen. Abbildung 9 zeigt ein Beispiel für eine Berichtseite und die möglichen Arten von Verweisen.

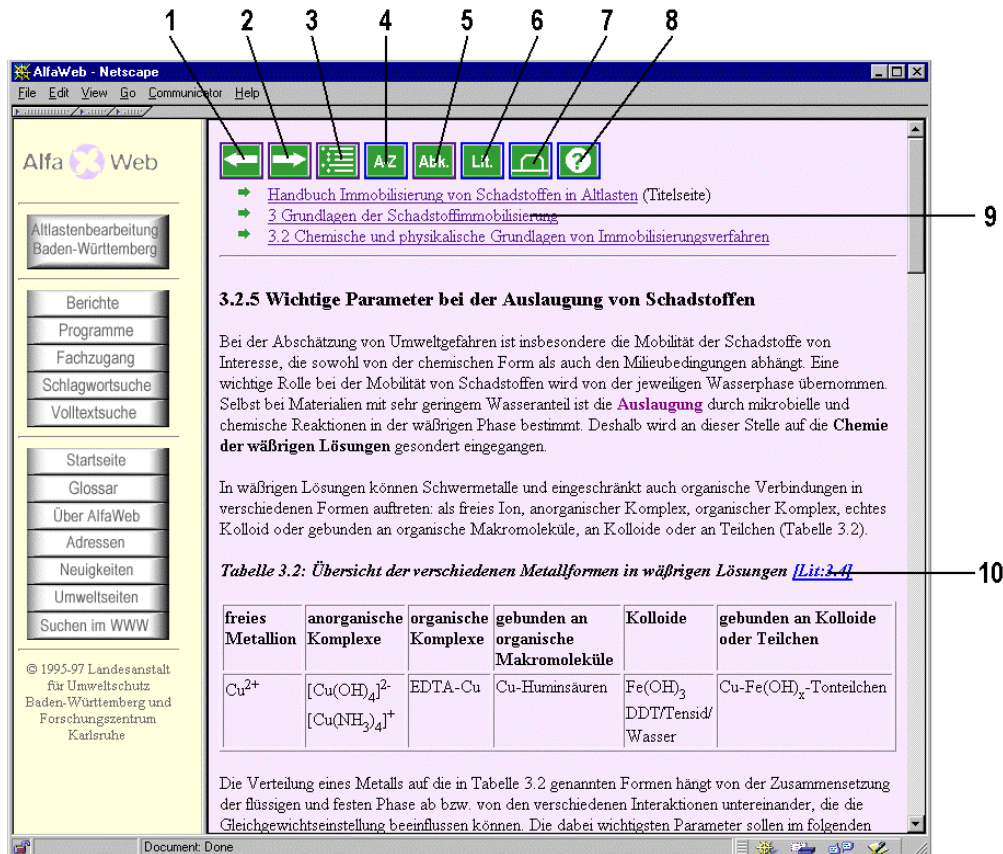


Abbildung 9: Beispiel für die Vernetzung von Berichtsseiten

- 1 = Verweis zur vorhergehenden Seite
- 2 = Verweis zur nachfolgenden Seite
- 3 = Verweis zum Inhaltsverzeichnis
- 4 = Verweis zum Indexverzeichnis
- 5 = Verweis zum Abkürzungsverzeichnis
- 6 = Verweis zum Literaturverzeichnis
- 7 = Lesezeichen setzen
- 8 = Hilfe
- 9 = Verweise zum Anfang übergeordneter Kapitel
- 10 = Verweis auf eine andere Berichtsseite

Im Rahmen von Globus IV wurden neue Versionen des Sharewareprogramms RTFtoHTML /5/, welches Kern des Konvertierungsprogramms ist, in Betrieb genommen, und das Gesamtsystem von Unix nach Windows NT portiert. An funktionalen Erweiterungen wurden die Unterstützung von farbigen Textstellen, Querverweise auf das Abkürzungsverzeichnis (siehe Abbildung 9) und die Möglichkeit zum Setzen von Lesezeichen implementiert.

Die farbige Hervorhebung von Texten wird in AlfaWeb für die Schlagwortsuche verwendet. Wenn Schlagworte nicht unmittelbar auffallen, weil diese z.B. mitten in einem längeren Fließtext stehen, werden diese in den AlfaWeb-Berichten farbig markiert. Wird eine Seite über die Schlagwortsuche aufgeblättert, findet man so leichter die relevanten Informationen.

Das Setzen von Lesezeichen (Bookmarks, Favoriten) ist zwar eine Standardfunktion der gängigen Browser, die jedoch nicht in gewünschter Weise funktioniert, wenn mit Frames gearbeitet wird. Dort kann ein Lesezeichen nur auf die Konfiguration gesetzt werden, die beim

Aufruf der Framestruktur als erstes angezeigt wird. Um auch Lesezeichen auf beliebige Berichtsseiten setzen zu können, wurde für AlfaWeb eine spezielle Funktion implementiert, die über die in Abbildung 9, Nr. 7 angezeigt Schaltfläche aktiviert wird.

Weitere Änderungen hinsichtlich Aufbau und Aussehen der Berichte wie die Hinterlegung von Berichtsseiten mit farbigem Hintergrund, die Angabe zum Stand eines Berichts (Datum) am Ende jeder Berichtsseite und die Einbettung von Abbildungen direkt in die WWW-Seiten erfolgten über eine geänderte Parametrisierung beim Konvertieren.

6. Ausblick

Ein entscheidender Meilenstein in der AlfaWeb-Entwicklung wird die Veröffentlichung der ersten vollständigen CD-ROM-Version sein. Die derzeitige Projektplanung geht davon aus, daß bis Ende des ersten Quartals 1998 die vorbereitenden Arbeiten abgeschlossen werden können.

Die Einsatzerfahrungen mit dieser CD-ROM werden von den an AlfaWeb beteiligten Ingenieurbüros erfaßt und ausgewertet. Es ist beabsichtigt, die zweite Version der CD-ROM in nicht allzu langem Abstand zur ersten Version fertigzustellen (ca. Ende 1998). Neben Verbesserungen, die als Reaktion auf die Einsatzerfahrungen vorgenommen werden sollen und der Aufnahme neu-erschienener Berichte sind auch funktionale Weiterentwicklungen vorgesehen, die aus zeitlichen Gründen für die erste Version nicht mehr zum Tragen kommen können. Beispiele sind ein "persönliche Assistent", der browser-seitig den Anwender bei der Zusammenstellung individueller Informationsbestände unterstützt oder ein Navigationstool, das die Orientierung im System durch jederzeit aktuelle Positions- und Umgebungsinformationen erleichtert.

Um mittelfristig auch im Internet wieder die vollständige AlfaWeb-Version anbieten zu können, sollen die im Internet existierenden Abrechnungssysteme evaluiert und die Verwaltung bei der Auswahl eines geeigneten Verfahrens unterstützt werden. In diesem Zusammenhang soll auch die Einrichtung von Benutzergruppen mit unterschiedlichen Profilen (Zugriffsrechte, Informationsangebote) untersucht werden.

Schließlich sind nach Fertigstellung der Komponenten zur Wartung und Pflege von AlfaWeb (siehe Kap. 5) die Voraussetzungen gegeben, um diese Werkzeuge in der Anwendungsumgebung zu installieren. Die weitere Pflege der AlfaWeb-Datenbasis kann dann direkt durch die Fachabteilung der LfU oder in deren Auftrag durch ein Ingenieurbüro erfolgen.

7. Literatur

- /1/ Weidemann, R.; Geiger, W.; Jaeschke, A.; Reißfelder, M. (1995): Entwicklung eines WWW-basierten Altlasten-Informationssystems, in Mayer-Föll, R.; Jaeschke, A. (Hrsg.): Projekt GLOBUS - Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg - Phase II 1995, Forschungszentrum Karlsruhe, wissenschaftliche Berichte, FZKA 5700, S. 271 - 297.
- /2/ Weidemann, R.; Geiger, W.; Reißfelder, R.; Schmid, E.; Reichert, U (1996): Inhaltlicher Ausbau und Weiterentwicklung des Altlasten-Fachinformationssystems AlfaWeb, in R. Mayer-Föll, A. Jaeschke (Hrsg.): Projekt GLOBUS, Konsolidierung der neuen Systemarchitektur und Entwicklung erster Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg - Phase III 1996, Forschungszentrum Karlsruhe, wissenschaftliche Berichte, FZKA 5900, S. 201 - 223.
- /3/ Geiger, W.; Reißfelder, M.; Weidemann, R. (1997): Distribution of Expert Information on Contaminated Soil via Internet, Intranet, CD-ROM and Print Media, in Proc. ECO-INFORMA '97, p. 72 -77, Eco-Infirma Press.
- /4/ Reißfelder, M.; Geiger, W.; Weidemann, R. (1997): WWW-Daten auf CDs abbilden: bewegter Abzug, iX, Heise Verlag Hannover, Heft 10, S. 102 - 107.
- /5/ Hector, C.: RTFtoHTML, <http://www.sunpack.com/RTF/>.

Ausblick auf die F+E-Arbeiten im Rahmen von GLOBUS Phase V (1998)

Die fünfte Phase des Vorhabens GLOBUS soll schwerpunktmäßig ausgerichtet werden auf die Umsetzung der bisher in GLOBUS erarbeiteten Ergebnisse und Konzepte im Rahmen der Weiter- und Neuentwicklung von praxis-relevanten Teilsystemen und Komponenten des UIS BW. Neben den bisher schon bearbeiteten Anwendungen bieten sich hier die UIS-Komponenten Informationssystem Wasser, Abfall, Altlasten, Boden (WAABIS), Kernreaktor-Fernüberwachungs-System (KFÜ-System) und der Fachdienst Naturschutz der LfU Baden-Württemberg (NaFaWeb) für entsprechende Realisierungen an. Weiter hat das Niedersächsische Umweltministerium seine Beteiligung an GLOBUS V in Aussicht gestellt.

Im folgenden sind die für 1998 geplanten bzw. angedachten F+E-Arbeiten der am Vorhaben GLOBUS Phase V beteiligten Institute zusammengefaßt:

1. F+E-Arbeiten des FAW

Vom FAW sollen in 1998 die F+E-Arbeiten zur Fortschreibung der Rahmenkonzeption des UIS Baden-Württemberg abgeschlossen, eine Erweiterung des Umwelt- und Verkehrs-Informationsservice (UVIS) untersucht und das Projekt Hypermediatechnik für Umweltdaten (HUDA) fortgesetzt werden.

Arbeitspunkt 1: RK-UIS

Ziel der Arbeiten ist die Weiterentwicklung der Rahmenkonzeption des UIS unter Berücksichtigung der geänderten Nutzeranforderungen, des aktuellen Stands der Technik und des neuen IuK-Modells der Landesverwaltung.

Einen Schwerpunkt der für 1998 geplanten Arbeiten bildet die ressortinterne Konsolidierung der erarbeiteten Konzepte im Ministerium für Umwelt und Verkehr (UVM), in der Landesanstalt für Umweltschutz und teilweise im Landesamt für Straßenwesen. Mit der Stabsstelle für Verwaltungsreform im Innenministerium, den betroffenen Ministerien und der Datenzentrale erfolgt eine ressortübergreifende Abstimmung. Weiter werden Leitlinien für die Konzeption von UIS-Fachsystemen erstellt. Die bereits im Laufe des Jahres 1997 begonnene Bestandserhebung und -analyse wird ausgearbeitet. Die Ergebnisse und Erkenntnisse der Fortschreibung und Weiterentwicklung der Rahmenkonzeption des UIS BW sollen dem Bund/Länder-Arbeitskreis UIS und ggf. der Öffentlichkeit vorgestellt werden.

Arbeitspunkt 2: UVIS

Die neue Systemarchitektur des Umwelt- und Verkehrs-Informationsservices (UVIS) kommt aufgrund der weiten Verbreitung der verwendeten WWW-Technologien dem Anspruch der Bürger auf aktuelle Informationen entgegen. Damit wäre auch in privaten Haushalten ein breiter Zugang zu reduzierten UVIS-Daten problemlos möglich; der Nutzerkreis könnte so weiter ausgedehnt werden. Da clientseitig allerdings nicht alle benötigten Funktionen dezentral bereitgestellt werden können, müßten zusätzliche Aufbereitungen im Server erfolgen.

Hierbei wird z.B. an folgendes gedacht:

- Einsatz von MS-ActiveX-Komponenten, die eine clientseitige Aufbereitung ohne Zusatzapplikationen ermöglichen.
- Java-Applets (z.B. GISTerm), die die UVIS-Daten über serverseitig abrufbare Applets clientseitig aufbereiten.
- Konzeption Bildarchiv; fertig aufbereitete Bilder werden mit den zugehörigen Parametern abgespeichert und können dann vom Client abgerufen werden.

Für den Bürger könnte somit ein externer Server realisiert werden, der einen reduzierten, für die Öffentlichkeit zugänglichen Datenbestand verwaltet. In einer ersten Version könnten regelmäßig durch Abfragen generierte Bilder in ein Bildarchiv auf den öffentlichen Server übertragen werden. In einer Weiterentwicklung sollten dann abrufbare Komponenten wie GISTerm und ActivX-Komponenten eingesetzt werden, die eine komfortable Aufbereitung ermöglichen.

Eine weitere Zusammenarbeit mit dem Umweltbundesamt in den künftigen Projektphasen des Verweis- und Kommunikationsservices Umwelt (VKS) und des Umweltinformationsnetzes Deutschland (GEIN) verspricht Synergieeffekte bei der Realisierung der hier skizzierten Entwicklungsziele.

Arbeitspunkt 3: Abstimmung mit HUDA

Es ist vorgesehen, das F+E-Vorhaben Hypermediatechnik für Umweltdaten (HUDA) im Auftrag des Umweltbundesamtes und des UVM 1998 fortzuführen. In der Projektphase III soll die Betriebsversion erstellt und erprobt sowie dokumentiert werden. Diese Arbeiten für HUDA sind mit GLOBUS abzustimmen.

2. F+E-Arbeiten des FZI

Die in GLOBUS V vom FZI für 1998 geplanten F+E-Arbeiten haben die Umsetzung der GLOBUS-Konzepte am Beispiel des Berichtssystems für das UIS-Informationssystem Wasser, Abfall, Altlasten, Boden (WAABIS) zum Ziel. Dies umfaßt die folgenden Arbeitspunkte:

Arbeitspunkt 1: Entwurf des WAABIS-Berichtssystems

Diese Aufgabe beinhaltet:

- Anforderungsanalyse relevanter Teile des WAABIS-Berichtssystems:
Zur Sicherstellung der Integration bzw. Zusammenführung der im F+E-Vorhaben GLOBUS, dem Web-Szenario (WWW-UIS) und dem WWW-UDK erarbeiteten Lösungen sind erforderlich:
 - Analyse der technischen Randbedingungen
 - Analyse der inhaltlichen Anforderungen.
- Software-Entwurf/Konzeption:
Das zu realisierende System ist zu konzipieren, der Entwurf ist zu dokumentieren. Dies umfaßt die folgenden Punkte:
 - Festlegung der Benutzerführung
 - Layout-Gestaltung (GUI)
 - Integration WWW-UDK (einschließlich der Berücksichtigung der Umstellung auf das Datenmodell UDK 4.0. Der Umweltdatenkatalog wird parallel von einer Bund-/Länderkooperation weiterentwickelt. Die Überarbeitung des Datenmodells, deren Abschluß für das Frühjahr 1998 geplant ist, und Weiterentwicklungen im Bereich WWW-UDK sind zu berücksichtigen.),
 - Architekturkonzept einschl. schriftlicher Dokumentation.

Arbeitspunkt 2: Realisierung

Die Umsetzung des in Arbeitspunkt 1 erarbeiteten Entwurfs umfaßt:

- Realisierung weiterer Selektoren,
- Integration des WWW-UDK in das WAABIS-Berichtssystem,
- Schnittstelle zum GISTerm des IPF,
- Einarbeitung von Verbesserungen basierend auf User-Feedback.

Arbeitspunkt 3: Betriebsunterstützung

Die Replikation der Datenbanken auf den öffentlichen Server ist Teil des INFORMS-Projektes WWW-UIS. Ziel ist es hierbei, die Replikation der internen Datenbestände auf die für die Öffentlichkeit zugänglichen externen Datenbestände zu automatisieren. Die zu replizierenden Datenbanken sind TULIS, MEROS und UDK. Neben der reinen Vervielfältigung oder Kopie der Daten stehen hierbei folgende Teilaufgaben im Vordergrund:

- Anonymisierung der Daten,
- Substitution von Attributwerten,

- Berücksichtigung der Sicherheitsaspekte der unterschiedlichen Vervielfältigungsmöglichkeiten.

Aufbauend auf den bisherigen Arbeiten wird die Realisierung der Konzepte im Vordergrund stehen. Hierzu gehören die Erstellung der Replikationsskripte sowie eine evtl. Anpassung der Skripte an das Datenmodell UDK 4.0.

Arbeitspunkt 4: Zusammenarbeit mit IITB / FhG

Im Rahmen der Realisierung des Informationssystems Wasser, Abfall, Altlasten, Boden (WAABIS) werden in 1998 F+E-Arbeiten vom IITB / FhG (Fraunhofer-Institut Informations- und Datenverarbeitung, Karlsruhe) durchgeführt. Diese Arbeiten umfassen

- die IuK-technische Untersuchung der vorhandenen WAABIS-Module bezüglich ihrer Struktur, Funktionalität und Semantik,
- die Erstellung einer verteilten bzw. verteilbaren Dienststruktur für WAABIS,
- die prototypische Umsetzung von WAABIS-Modulen (insbesondere: 'Berichtssystem', 'WAABIS-GEO', 'Grundwasser') in einer verteilten Dienststruktur auf einer geeigneten Plattform.

Die oben aufgeführten Entwicklungen müssen eng zwischen FZI und IITB / FhG abgestimmt werden.

3. F+E-Arbeiten des FZK / IAI

Die für 1998 im Rahmen von GLOBUS V geplanten Arbeiten des FZK / IAI sind die direkte Weiterführung der Entwicklung des Altlasten-Fachinformationssystem AlfaWeb. Im Einzelnen beinhaltet dies die folgenden Aufgaben:

Arbeitspunkt 1: AlfaWeb auf CD-ROM

Bis März 1998 wird die erste vollständige und kommerziell verbreitbare CD-ROM-Version von AlfaWeb erstellt.

Auf Basis der Nutzer-Erfahrungen mit dieser ersten CD-ROM-Version sollen qualitative Verbesserungen durch funktionale Weiterentwicklungen erreicht werden, die zusammen mit inhaltlichen Erweiterungen (neue Berichte) eine weitere Verbesserung der Nutzbarkeit der für Ende 1998 geplanten zweiten Auflage der CD-ROM-Version ergeben sollen.

Arbeitspunkt 2: AlfaWeb im Internet

Derzeit ist AlfaWeb als Pilotversion kostenfrei im Internet verfügbar. AlfaWeb mit dem kompletten Umfang und allen Funktionen - wie sie auf der CD-ROM zur Verfügung stehen werden - kostenfrei ins Internet einzustellen, ist u.a. aus wirtschaftlichen Gründen nicht möglich. Während verwaltungsintern (Intranet) die vollständige Version verfügbar bleibt, ist mit dem Erscheinen der CD-ROM für das Internet eine spezielle Version zu erstellen, die sich auf ausgewählte Informationen beschränkt. Hierzu ist ein Programm zu entwickeln, das diese Internet-Version aus der vollständigen Version generiert.

Mittelfristig soll auch im Internet wieder die vollständige Version angeboten werden, wobei bestimmte Teile dann allerdings nur kostenpflichtig abgerufen werden können. Die im Internet existierenden Abrechnungssysteme sollen 1998 evaluiert und die Verwaltung bei der Auswahl eines geeigneten Verfahrens unterstützt werden. In diesem Zusammenhang soll auch die Einrichtung von Benutzergruppen mit unterschiedlichen Profilen (Zugriffsrechten, Informationsangeboten) untersucht werden.

Arbeitspunkt 3: Intelligente Zugangssysteme

Derzeit sind als vollimplementierte Zugangsmöglichkeiten der hierarchische Zugang und die Volltextsuche vorhanden, Fachzugang und Schlagwortsuche werden momentan für die CD-ROM-Version fertiggestellt. Von Anwenderseite wurden Anforderungen an das Zugangssystem formuliert, deren Umsetzung nur durch die Realisierung neuartiger Systemkomponenten möglich ist. Lösungsmöglichkeiten sind hier ein "persönlicher Assistent", der browser-seitig den Anwender bei der Zusammenstellung individueller Informationsbestände unterstützt oder ein Navigationstool, das die Orientierung im System durch jederzeit aktuelle Positions- und Umgebungsinformationen erleichtert. Auf der Basis der genannten Anforderungen sollen Realisierungsvorschläge erarbeitet, mit Anwendern diskutiert und anschließend implementiert und in AlfaWeb integriert werden.

Arbeitspunkt 4: Systemverwaltungskomponente

Eine erste Version der Systemverwaltungskomponente wird bereits bis Ende 1997 entwickelt. Die Komponente muß in AlfaWeb integriert und die LfU bei der Einführung und dem ersten Einsatz unterstützt werden. Außerdem ist geplant, die Komponente entsprechend den Einsatzverfahren und Weiterentwicklungen von AlfaWeb zu erweitern und zu ergänzen.

Arbeitspunkt 5: Fachinformationssystem NaFaWeb

Aufbauend auf den Konzepten und Werkzeugen, die im F+E-Vorhaben Altlasten-Fachinformationssystem AlfaWeb erarbeitet wurden, ist geplant, für den bei der LfU installierten Fachdienst Naturschutz das Fachinformationssystem NaFaWeb im Auftrag des Ministeriums Ländlicher Raum (MLR) zu entwickeln.

Der Fachdienst Naturschutz soll fachliche und rechtliche Arbeitsgrundlagen des Naturschutzes in Form praxisorientiert aufbereiteter Informationen und Materialien herausgeben, hierzu beraten und so der Verbesserung des Vollzugs, der Steigerung der Qualität der Arbeit der Naturschutzbehörden und der Beschleunigung von Verfahren sowie einer gleichmäßigen Anwendung des Naturschutzrechts dienen. Dies erfolgt durch die Erstellung von Arbeitshilfen in Form von Handbüchern, Fachberichten, Datenbanken und Fachprogrammen.

Ziel des Projekts ist es, diese Arbeitshilfen sowie andere für die Bearbeitung relevante Informationsquellen den Beteiligten in digitaler Form zur Verfügung zu stellen und durch rechnergestützte Zugangshilfen ein rasches Auffinden der gesuchten Informationen zu ermöglichen.

In einem ersten Schritt sollen durch ein Ingenieurbüro eine Bestandsaufnahme der vorhandenen Arbeitshilfen, Checklisten, Leitfäden, Handbücher etc. sowie eine Konzeption für ein auf CD-ROM basierendes Fachinformationssystem erstellt werden.

4. F+E-Arbeiten des IKE

Die vom IKE für 1998 in GLOBUS V geplanten F+E-Arbeiten umfassen zum einen den Ausbau der in den vorangegangenen Projektphasen entwickelten und erprobten Komponente Personal Data Node PDN und zum anderen den Transfer der Konzepte und Komponenten aus GLOBUS in die Teilerneuerung des Kernreaktor-Fernüberwachungs-Systems (KFÜ-Systems). Dies beinhaltet die Aufgaben:

Arbeitspunkt 1: Entwurf der Architektur des Dienstes Ausbreitungsrechnung

Nachdem für den PDN zunächst die Architektur des logischen Klienten entwickelt und in verschiedenen Realisierungen erprobt wurde, soll nun auf Serverseite entsprechend vorgegangen werden. Dazu wird im ersten Schritt die Architektur des Dienstes für Ausbreitungsrechnungen entworfen und im zweiten Schritt in Teilen prototypisch implementiert.

Durch die bisherigen Arbeiten im F+E-Vorhaben GLOBUS - und hier insbesondere im Rahmen der Dienste AG - wurden die folgenden Komponenten solch einer Architektur herausgearbeitet:

- die Erbringung des Dienstes (Rechnungsdurchführung),
- die Verwaltung der Dienst-Inanspruchnahmen (Session Manager),
- die Verwaltung der Ergebnisse der Dienst-Inanspruchnahmen (Ergebnis Manager),
- die Auswahl einer konsistenten Strategie für die Erbringung eines Dienstes und die Bereitstellung der dazu nötigen Daten (Strategiekomponente oder Arbeitsdaten Manager).

Die Spezifikation dieser Komponenten wird auf Basis objektorientierter Methoden mit dem Case Tool Rational Rose erstellt und dokumentiert. Die prototypische Umsetzung ausgewählter Teile wird 1998 begonnen.

Arbeitspunkt 2: Unterstützung des Arbeitens mit dem PDN durch aktive Objekte

Aktive Objekte sind Agenten, die in der Lage sind, einfache Aufgaben des UIS lokal oder remote durchzuführen. Dazu gehören etwa die Auswahl des für die Lösung einer Aufgabe geeigneten Dienstes unter Berücksichtigung von Verfügbarkeit, Kosten und Systemzustand (Trading) oder die dynamische Verbindung verfügbarer Dienste zur Lösung einer Aufgabe im Rahmen eines vorgegebenen Kontextes (Strategiekomponente).

Arbeitspunkt 3: Realisierung des KFÜ-Systems gemäß der Rahmenkonzeption UIS

Das UVM beauftragte die Firma debis Systemhaus (debis SH) mit der Teilerneuerung der Kernreaktor-Fernüberwachung. Das IKE erhielt von debis SH einen Unterauftrag zur Realisierung von Teilen dieses Projekts. Ziel der Mitarbeit ist es, zum einen die bei der Spezifikation des KFÜ-Systems gemachten Erfahrungen in GLOBUS zur Verfügung zu stellen und zum anderen dazu beizutragen, daß GLOBUS-Ergebnisse dort sofort mitbenutzt werden können.

Konkrete Aufgaben sind:

- Verfügbarmachung des Isolinien Algorithmus von GLOBUS im KFÜ-System, d.h. Integration des GIS-Viewers des KFÜ in den PDN und
- Verfügbarmachung des GLOBUS-Dienstekonzepts für Ausbreitungsrechnungen im KFÜ-System, d.h.
 - Integration der KFÜ-Ausbreitungsdienste,
 - Verfügbarmachung des GLOBUS-Repositorydienstes für den Ausbreitungs-Server des KFÜ-Systems.

Arbeitspunkt 4: Ausbau der Kommunikationskomponente des PDN; Kooperatives Arbeiten

Die Informationstechnik kann heute zeit- und aufwandsparende Unterstützung im Bereich der behördlichen Kommunikation und Kooperation leisten. Beispiele, die schon jetzt intensiv genutzt werden, sind e-mail oder die Bereitstellung von Berichten im Netz. Computer Supported Cooperative Work Techniken unterstützen die Kommunikation, indem sie schnelle multimediale Rückfragen erlauben, das gemeinsame Bearbeiten von Vorlagen und Protokollen unterstützen und die partnerschaftliche Durchführung komplexer Aufgaben einschließlich der Diskussion von Ergebnissen ermöglichen.

Ziel des IKE in diesem Arbeitspunkt ist die Untersuchung und Bereitstellung von aktuellen Techniken des rechnergestützten kooperativen Arbeitens und die Entwicklung von Konzepten und Prototypen für den Einsatz in einer UIS-Umgebung.

5. F+E-Arbeiten des IPF

Die für 1998 vom IPF geplanten F+E-Arbeiten konzentrieren sich auf die Weiterentwicklung des JAVA-basiertes GIS-Terminal für das UIS im Rahmen der IuK-Vorhaben GLOBUS und WAABIS (WAABIS-GEO) und für das Niedersächsische Umweltministerium.

Arbeitspunkt 1:

Ziel dieser Teilaufgabe ist die Weiterentwicklung des GISTerm für die grafischen Aufgaben in GLOBUS und WAABIS-GEO mit den Arbeitsschritten:

- Visualisierung von Punkt-, Linien-, Flächenshapes,
- Visualisierung von Raster-Karten (Images) als gif oder jpeg-Formate,
- Integration von Diagrammen in die Karte,
- Schnittstelle zum Geodatenserver Smallworld (CORBA) und zu Shapefile-Servern,
- Online-Dokumentation,
- Online-Hilfefunktionen.

Arbeitspunkt 2:

Die Teilaufgabe besteht in der Ergänzung der Grundfunktionen von GISTerm um wichtige Anwendungsfunktionen mit den Entwicklungsschritten:

- einfache Rasteroperationen (Reklassifizieren von Rasterinformationen anhand der Zuordnung von Farbtabelle zu Klassifizierungsmerkmalen),
- Anzeigen von Klassifizierungsmerkmalen einer Raster-Karte (Pickfunktion für Raster),
- einfache Erfassungsfunktionen für Vektoren (Manuelle On-Screen Digitalisierung auf der Basis von Rasterdaten und Speicherung als Shape-Dateien),
- Auswahl von Gebieten/Linien/Punkten anhand der Vorgaben für die Präsentation-aufbereitung,
- Verbinden (join) von Sachdaten (z.B. dbase-Tabellen) mit den Attributtabelle einzelner Layer,
- Visualisierung von Klassifikationen anhand der Attributierung.

Die mit dem GISTerm entwickelten Geo-Funktionen sollen in skalierbarem Umfang (als Dienste) auch von anderen UIS-Anwendungen genutzt werden können. Ziel ist es daher, die Software lizenzfrei und mit standardisierten Schnittstellen im Client-Umfeld und für den Datenzugriff auf der Serverseite bereitzustellen.

6. Schlußbemerkung:

Die dargestellten Arbeitspunkte der Institute sind Vorschläge. Sie müssen teilweise noch verifiziert und priorisiert werden. Bei Bedarf können weitere Partner hinzugezogen werden.

Es ist vorgesehen, die F+E-Arbeiten im Rahmen von GLOBUS Phase V im November 1998 abzuschließen.

Roland Mayer-Föll

Andreas Jaeschke