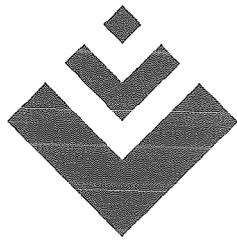


F
F
Forschungszentrum Karlsruhe
Technik und Umwelt

Wissenschaftliche Berichte
FZKA 6250



UIS

Baden-Württemberg

Projekt GLOBUS
Multimediales Recherchieren
und Verarbeiten von globalen
Umweltsachdaten im
Umweltinformationssystem
Baden-Württemberg
Phase V 1998

R. Mayer-Föll, A. Jaeschke (Hrsg.)

Ministerium für Umwelt und Verkehr Baden-Württemberg

Forschungszentrum Karlsruhe
Institut für Angewandte Informatik

Dezember 1998

Forschungszentrum Karlsruhe
Technik und Umwelt

Wissenschaftliche Berichte
FZKA 6250

Projekt GLOBUS

Multimediales Recherchieren und Verarbeiten
von globalen Umweltsachdaten
im Umweltinformationssystem Baden-Württemberg
Phase V 1998

R. Mayer-Föll, A. Jaeschke (Hrsg.)

Ministerium für Umwelt und Verkehr Baden-Württemberg

Forschungszentrum Karlsruhe
Institut für Angewandte Informatik

Forschungszentrum Karlsruhe GmbH, Karlsruhe

1998

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.

Projekt GLOBUS

**Multimediales Recherchieren und Verarbeiten
von globalen Umweltsachdaten
im Umweltinformationssystem Baden-Württemberg**

Phase V - 1998

Projektträger:

**Ministerium für Umwelt und Verkehr Baden-Württemberg (UVM);
R. Mayer-Föll**

**Landesanstalt für Umweltschutz Baden-Württemberg (LfU);
A. Keitel**

Weitere beteiligte Stellen:

Bundesanstalt für Wasserbau Karlsruhe (BAW)

Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit (BMU)

Fraunhofer-Institut für Informations- und Datenverarbeitung Karlsruhe (FhG/IITB)

Ministerium Ländlicher Raum Baden-Württemberg (MLR)

Niedersächsisches Umweltministerium (MU NI)

Stabsstelle für Verwaltungsreform im Innenministerium Baden-Württemberg (StaV/IM)

Umweltbundesamt Berlin (UBA)

Projektpartner:

**Forschungszentrum Karlsruhe - Technik und Umwelt -
Institut für Angewandte Informatik (FZK/IAI);
A. Jaeschke (Federführung)**

**Forschungsinstitut für anwendungsorientierte Wissensverarbeitung
an der Universität Ulm (FAW);
K. Tochtermann**

**Forschungszentrum Informatik
an der Universität Karlsruhe (FZI);
R. Kramer**

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

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

Vorwort

Im Umweltbereich benötigt die Verwaltung zur Erledigung ihrer Aufgaben Daten und Informationen. Für deren Erhebung, Erfassung, Auswertung, Darstellung und Weitergabe sind Informations- und Kommunikationstechniken (IuK) inzwischen unverzichtbar geworden. Die gesamten IuK-Aktivitäten des Landes mit Umweltbezug werden im Umweltinformationssystem Baden-Württemberg zusammengefaßt. Umweltprobleme sind komplex. In verstärktem Maße müssen die Zusammenhänge beachtet werden, die zwischen den Umweltbereichen wie beispielsweise Wasser, Abfall, Altlasten und Boden bestehen. Dem Bürger ist der freie Zugang zu Informationen über die Umwelt zu ermöglichen.

Das Umweltinformationssystem (UIS) Baden-Württemberg ist der Rahmen für die Bereitstellung von Umweltdaten und die Bearbeitung von fachbezogenen und fachübergreifenden Aufgaben im Umweltbereich. 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 (UVM) das Forschungs- und Entwicklungsprojekt GLOBUS - Multimediales Recherchieren und Verarbeiten von globalen Umweltsachdaten im Umweltinformationssystem Baden-Württemberg, Phase V 1998 - 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.

Während des Projekts kam

- das Fraunhofer-Institut für Informations- und Datenverarbeitung (FhG/IITB) Karlsruhe neu hinzu.

Wie in den vorangegangenen Phasen hat das Institut für angewandte Informatik des Forschungszentrums Karlsruhe das Projektmanagement und die Erstellung des Abschlußberichts übernommen.

Im Rahmen von GLOBUS V wurden auf der Basis der neuen, im Verlauf der vorhergehenden Projektphasen erreichten Ergebnisse weitere Lösungen entwickelt und eingeführt. Multimediales Recherchieren und Verarbeiten von globalen Umweltsachdaten ist in der Landesanstalt für Umweltschutz, im Ministerium für Umwelt und Verkehr und weiteren Dienststellen möglich geworden. Anwender nutzen diese verbesserten Angebote für die Aufgabenerledigung bereits intensiv.

Einen Teil der Arbeiten stellten UVM, Landesanstalt für Umweltschutz (LfU) und die GLOBUS-Partner 1998 beim Symposium Umweltinformatik in Bremen, dem GI-Workshop Hypermedia im Umweltschutz im FAW Ulm und auf Sitzungen des Bund/Länder-

Arbeitskreises Umweltinformationssysteme (BLAK UIS) mit beachtlicher Resonanz bereits vor.

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

Unser Dank gilt insbesondere auch

- der Landesanstalt für Umweltschutz Baden-Württemberg,
- der Stabsstelle für Verwaltungsreform im Innenministerium,
- dem Ministerium Ländlicher Raum Baden-Württemberg,
- dem Niedersächsischen Umweltministerium,
- der Bundesanstalt für Wasserbau Karlsruhe,
- dem Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit,
- dem Umweltbundesamt Berlin und
- der Datenzentrale Baden-Württemberg

für deren fachliche Beratung und wesentliche Unterstützung.

Die Erkenntnisse von GLOBUS V sollen im vorliegenden Bericht der Fachöffentlichkeit bekanntgegeben werden.

Jürgen Pätzold, Roland Mayer-Föll
Ministerium für Umwelt und Verkehr Baden-Württemberg

Inhalt

Vorwort

Inhaltsverzeichnis

Das F+E-Vorhaben GLOBUS.....	1
GISterm, Weiterentwicklung von GISterm zu einer allgemeinen Komponente für den Zugriff und die Visualisierung von räumlichen Daten.....	9
Das Sachdatensystem, Das komponentenorientierte System zum komfortablen Recherchieren nach Umweltdaten im UIS Baden-Württemberg.....	31
Zertifizierung und Signierung von Java Applets für deren Nutzung in Internet-basierten Umweltinformationssystemen	63
Von AlfaWeb zur XfaWeb-Systemfamilie	71
Anpassungen des Personal Data Node an UIS-Nutzungen, insbesondere für die Kernreaktor-Fernüberwachung (KFÜ)	105
Bestandsaufnahme und Bewertung von ausgewählten UIS-Systemen zur Fortschreibung und Weiterentwicklung der Rahmenkonzeption des Umweltinformationssystems Baden-Württemberg.....	129
Dokumentation des Umwelt- und Verkehrs-Informationsservices Baden-Württemberg	141
HUDA, Hypermediatechnik für Umweltdaten.....	149
Ausblick auf die F+E-Arbeiten im Rahmen von GLOBUS Phase VI 1998/1999.....	159

Das F+E-Vorhaben GLOBUS

Einführung

Projektrahmen

Das Forschungs- und Entwicklungsvorhaben GLOBUS (Globale Umweltsachdaten) wurde 1994 vom Umweltministerium Baden-Württemberg (UM) begonnen und wird seit 1996 vom Ministerium für Umwelt und Verkehr Baden-Württemberg (UVM) als Projektträger fortgeführt. Dabei handelt es sich um jeweils einjährige Projektphasen (GLOBUS I bis V).

An dem Vorhaben sind die Landesanstalt für Umweltschutz Baden-Württemberg (LfU) als Projektentwicklungsstelle, die Stabsstelle für Verwaltungsreform im Innenministerium BW (StaV) und die folgenden Forschungsinstitute beteiligt:

1. das Forschungsinstitut für anwendungsorientierte Wissenverarbeitung an der Universität Ulm (FAW),
2. das Forschungszentrum Informatik an der Universität Karlsruhe (FZI),
3. das Institut für Angewandte Informatik des Forschungszentrums Karlsruhe (FZK/IAI),
4. das Institut für Kernenergetik und Energiesysteme der Universität Stuttgart (IKE),
5. das Institut für Photogrammetrie und Fernerkundung der Universität Karlsruhe (IPF)
6. und seit Mitte 1998 das Fraunhofer-Institut für Informations- und Datenverarbeitung (FhG/IITB).

Im Laufe des Vorhabens traten das Ministerium Ländlicher Raum BW, das Niedersächsische Umweltministerium und die Bundesanstalt für Wasserbau, Karlsruhe, der GLOBUS-Kooperation bei. Eine intensive Austauschbeziehung entstand mit dem Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit sowie dem Umweltbundesamt, insbesondere im Rahmen des F+E-Vorhabens HUDA (Hypermediatechnik für Umweltdaten). Mit der Datenzentrale Baden-Württemberg wurde im Zusammenhang mit dem Land/Kommunen-Vorhaben WAABIS (Informationssystem Wasser, Abfall, Altlasten, Boden) ein Entwicklungsverbund gebildet.

Ein Lenkungsausschuß (Leitung UVM, Geschäftsführung FZK/IAI) begleitet die verschiedenen F+E-Aktivitäten in GLOBUS steuernd.

Ziele, Aufgaben und Anforderungen

Ziel des Vorhabens GLOBUS ist die konzeptionelle Fortschreibung und funktionale Weiterentwicklung des Umweltinformationssystems Baden Württemberg (UIS BW / UIS). Dies umfaßt - unter Wahrung des vorhandenen Informationskapitals - die Anpassung des Systems an innovative Hard- und Softwarekonzepte, die Integration und Nutzung moderner Informa-

tik-Werkzeuge und die Erweiterung des UIS entsprechend den Anforderungen eines gewandelten Anwendungsprofils.

Die Aufgabe des UIS ist die informationstechnische Unterstützung der Fach- und Führungskräfte im Ministerium für Umwelt und Verkehr und in allen Behörden des Landes mit Umweltbezug, ausgerichtet auf eine Steigerung von Effizienz und Qualität in der Durchführung der Verwaltungsaufgaben. Tangiert sind damit die Bereiche Planung, Verwaltungsvollzug, Umweltbeobachtung, Notfallmanagement, Umweltberichterstattung und generelle Informationsbereitstellung. Die Versorgung der Öffentlichkeit mit Umweltinformationen gewinnt dabei zunehmend an Bedeutung.

Die Entwicklung des UIS begann bereits 1983 mit organisatorischen und konzeptionellen Vorarbeiten. Ab 1984 entstand der Systementwurf /1/, /2/, /3/. Entsprechend den zu dieser Zeit aktuellen Konzepten wurde ein System mit zentralistischer Hardwarearchitektur (zentrale Rechner / unintelligente Peripherie) entworfen. In entsprechender Softwarestruktur wurden in einer ersten Entwicklungsphase die ersten ressortspezifischen Fachsysteme als monolithische Bausteine realisiert. In der zweiten Entwicklungsphase (ab 1987) standen dann - im wesentlichen unter Fortentwicklung der Systemarchitektur - fach- und ressortübergreifende UIS-Fachsysteme und -Komponenten und die Nutzung des Landesverwaltungsnetzes im Vordergrund.

1995 begann die dritte Entwicklungsphase des UIS. Diese wurde angestoßen zum einen - von DV-technischer Seite - durch den Fortschritt im Bereich der Hardware/Software-Konzepte, -Werkzeuge und -Systeme und zum anderen - von Seiten der Anwendung - durch neue Anforderungen an das UIS.

Auf IuK-technischer Seite waren mittlerweile zentralorientierte Architekturen durch dezentrale, verteilte Strukturen abgelöst worden. Datennetze und entsprechende Kommunikationstools fanden immer breitere Anwendung. Parallel und damit Hand in Hand hatten PCs und intelligente Peripherie enorm an Bedeutung gewonnen und wurden steigend in der Verwaltung eingesetzt. Für die PCs waren bereits mächtige Werkzeuge für den Bereich der Bürokommunikation marktverfügbar und z.T. zum Standard geworden.

Um die Hardware- und Software-technische Pflege des Systems sicherzustellen und um die technischen und kostenmäßigen Vorteile der Innovationen zu nutzen, war es zwingend, diese neuen Gegebenheiten und Trends in die Fortschreibung und Weiterentwicklung des UIS zu integrieren.

Parallel zu den veränderten IuK-technischen Gegebenheiten führten geänderte Prioritätensetzung in den Aufgabenbereichen zu neuen Anforderungen an das UIS von Seiten der Anwender. Diese ergaben sich aus ständigen Veränderungen in der Verwaltungsstruktur, der Neugliederung der Ressorts auf Ministeriums- und Behördenebene und einer gleichzeitigen, engeren Verzahnung der Verwaltungsabläufe von Land und Kommunen. Hinzu kam eine wachsende Vernetzung der Aktivitäten auf Länder-, Bundes- und europäischer Ebene und Anforderungen, die unter anderem aus der Verpflichtung zur Bereitstellung von Umweltinformationen für die Öffentlichkeit resultieren.

Um Akzeptanz und Effizienz des Systems sicherzustellen, waren die resultierenden Anforderungen in der Weiterentwicklung des UIS vorrangig zu beachten.

Projektverlauf

Die Berücksichtigung der genannten, neuen IuK-technischen und anwendungsbezogenen Aspekte führte im Rahmen der F+E-Arbeiten des Vorhabens GLOBUS zu einer neuen Architektur des UIS als dezentrales, verteiltes System, in dem die bisherigen monolithischen Fachsysteme in Dienste aufgesplittet werden und von Diensteanbietern den Nutzern/Anwendern zur Verfügung gestellt werden (Informationsmarkt).

In den ersten zwei Phasen des GLOBUS-Vorhabens standen Untersuchungen von aktuellen IuK-Konzepten und -Werkzeugen bezüglich ihrer Eignung zum Einsatz im UIS und darauf aufbauende konzeptionelle Arbeiten im Vordergrund. In den Folgephasen dominierte zunehmend die Realisierung von UIS-Anwendungskomponenten, in denen die erarbeiteten, neuen Konzepte umgesetzt und erprobt wurden. Zu diesen Entwicklungen gehört das System 'Umwelt- und Verkehrs-Informationsservice' UVIS, Realisierungen im Umfeld des Umweltdatenkatalogs, der Geo-Arbeitsplatz GISterm und das Altlasten-Fachinformationssystem AlfaWeb als erstes Element der Fachinformationssystem-Familie XfaWeb.

In der ersten Phase des F+E-Vorhabens GLOBUS /4/ wurde 1994 untersucht, in welcher Form sich die neuen Techniken des Internet zur Effizienz- und Qualitätsverbesserung bei der Informationsbereitstellung im UIS einsetzen lassen. Mit einer prototypisch realisierten Auskunftskomponente auf Basis des World-Wide Web (WWW) wurden Lösungswege und Möglichkeiten aufgezeigt und Machbarkeit und Effizienz nachgewiesen.

Die zweite Phase des Projekts 1995 /5/ diente der weiteren Detaillierung und Ausarbeitung einer übergreifenden Rahmenarchitektur für die UIS-integrierte Auskunftskomponente. Dies führte zu einer diensteorientierten, auf WWW und CORBA (Common Object Request Broker Architecture) basierten Architektur für das UIS. Es wurden eine Evaluierung der verfügbaren CORBA-Implementierungen und Untersuchungen zum Einsatz von Diensten im UIS durchgeführt.

Weiter wurden marktverfügbare Visualisierungstools bezüglich ihrer Eignung zum clientseitigen Einsatz in einem Geo-Arbeitsplatz untersucht und mit der Entwicklung des Altlasten-Fachinformationssystems AlfaWeb begonnen.

In der dritten Phase des Projekts wurde 1996 /6/ das vom FAW, Ulm, für das UIS entwickelte Umwelt-Führungs-Informationssystem UFIS in der neuen Systemarchitektur zum Umwelt- und Verkehrs-Informationsservice UVIS erweitert, die UDK-basierten Auskunftsdienste im UIS funktional ausgebaut und Inhalt, Zugangssystem sowie Konvertierungstools des Altlasten-Fachinformationssystems AlfaWeb weiterentwickelt.

Das Dienstekonzept wurde konsolidiert und durch die prototypische Realisierung von UIS-Komponenten (Spurengasausbreitung) untermauert. Aufbauend auf die Untersuchungen in Phase II wurde die Entwicklung eines Geo-Clients fortgeführt.

Die vierte Phase von GLOBUS /7/ umfaßte die direkte Fortführung der in der vorangegangenen Phase begonnenen Entwicklungen.

Die Arbeiten am Umwelt- und Verkehrs-Informationsservice UVIS wurden fortgesetzt und abgeschlossen. Hierzu wurde eine Komponente zur metadaten-basierten Vermittlung von

Diensten erstellt. Die Entwicklungen im Anwendungsumfeld des UDK auf WWW wurden komplettiert. Für das Fachinformationssystem AlfaWeb standen die Entwicklung einer CD-ROM-Version und von Systemverwaltungskomponenten im Vordergrund. Weiter wurde der Geo-Arbeitsplatz GISterm funktional wesentlich erweitert.

Parallel zu den F+E-Arbeiten von GLOBUS ergaben sich im Umfeld des Vorhabens Projekte mit Zielsetzungen, die auf Resultate von GLOBUS aufbauen oder in die Ergebnisse aus GLOBUS Eingang gefunden haben. Zu nennen sind hier insbesondere:

- Das Projekt RK-UIS (Fortschreibung und Weiterentwicklung der Rahmenkonzeption des UIS) /8/, /9/, das im Auftrag des UVM BW vom FAW, Ulm, bearbeitet und in 1998 abgeschlossen wurde. Es enthält systemanalytische Untersuchungen zum Systemeinsatz und erarbeitet Vorschläge und Konzepte zur Weiterentwicklung des UIS BW.
- Das Projekt HUDA (Hypermediatechnik für Umweltdaten) /10/, /11/ wurde aus UFOPLAN-Mitteln des UBA vom BMU und UVM Baden-Württemberg beauftragt, wobei die Projektträgerschaft beim UVM und die Projektleitung bei der LfU liegt. Die Entwicklungsarbeiten werden von einem Konsortium aus Instituten des GLOBUS-Entwicklerverbundes unter Federführung des FAW durchgeführt. Aufgabe des Projekts ist die Erstellung eines Werkzeugkastens für die weitgehend automatisierte Erstellung von Umweltberichten aus den verfügbaren Basis-Fachinformationen. Das Projekt begann 1996 mit einer Machbarkeitsstudie; 1997 erfolgte die Erstellung des Systemkonzepts und eines Prototyps; 1998/99 soll das Projekt mit der Erstellung der Betriebsversion und der Dokumentation von HUDA abgeschlossen werden.
- Das Projekt zur Entwicklung der UIS-Komponente WAABIS (Informationssystem Wasser, Abfall, Altlasten, Boden) /12/, /13/ wird vom UVM in enger Abstimmung mit dem kommunalen Bereich seit 1996 betrieben. Das System ist ein Ersatz und eine Erweiterung des aufgrund der Zuständigkeitsverlagerung im Zuge des Sonderbehörden-Eingliederungsgesetzes von 1995 (SoBEG) nicht weiter einsetzbaren Informationssystems KIWI (kommunikativ integriertes wasser- und abfallwirtschaftliches Informationssystem). In die Konzeption von WAABIS flossen von Beginn an die Ergebnisse aus GLOBUS ein.
- Das Projekt zur Teilerneuerung der Kernreaktor-Fernüberwachung Baden-Württemberg (KFÜ) wird vom UVM im Benehmen mit den Betreibern der Kernkraftwerke durchgeführt. Generalunternehmer ist die Firma debis Systemhaus. Über weitere Aufträge des UVM ist das IKE als GLOBUS-Partner in die Teilerneuerung des KFÜ eingebunden, wodurch der wechselseitige Wissenstransfer gewährleistet ist.

GLOBUS Phase V - 1998

Aufgabenstellung für Phase V

In der fünften Phase des F+E-Vorhabens GLOBUS wurden - der Gesamtplanung des Vorhabens entsprechend - vorrangig die laufenden anwendungsnahen Implementierungen weitergeführt, in der Praxis erprobt und/oder eingeführt. Im Wesentlichen handelt es sich hier um den Aufbau einer Fachinformationssystem-Familie XfaWeb - auf der Basis der Konzepte und Werkzeuge des Altlasten-Fachinformationssystems AlfaWeb - und weiter um den funktionalen Ausbau des Geo-Arbeitsplatzes GISterm durch das IPF und das Sachdatensystem durch das FZI. Vom IKE wurde der Personal Data Node PDN entsprechend den Anforderungen aus dem UIS-Bereich komplettiert. Inhaltliche Schnittstellen zu GLOBUS hatten die Arbeiten zur Rahmenkonzeption UIS (RK UIS '98 / FAW) und die Arbeiten der GLOBUS-Partner zum Vorhaben HUDA.

Gliederung des GLOBUS V - Berichts

In dem vorliegenden Abschlußbericht der Phase V folgen auf das Vorwort und die Einführung folgende Fachkapitel:

Im ersten Kapitel des Berichts sind die in 1998 vom IPF durchgeführten Arbeiten zur **'Weiterentwicklung des GISterm zu einer allgemeinen Komponente für den Zugriff und die Visualisierung von räumlichen Daten'** beschrieben. Im Wesentlichen handelt es sich hier um funktionale Erweiterungen. Die Basis hierfür - Systemarchitektur und ein funktionsfähiger Prototyp - wurde bereits in den vorangegangenen Projektphasen entwickelt.

Unter dem Titel **'Das Sachdatensystem'** werden im zweiten Kapitel die in 1998 vom FZI erzielten Ergebnisse bei der Entwicklung des Sachdatensystems dargestellt, mit dem im UIS eine komfortable Möglichkeit geschaffen wird, über vorstrukturierte Abfragen (Selektoren) auf Datenbanken zuzugreifen.

Das dritte Kapitel **'Zertifizierung und Signierung von Java Applets für deren Nutzung in Internet-basierten Umweltinformationssystemen'** gibt den Stand der Überlegungen einer GLOBUS-AG wieder, die Lösungsmöglichkeiten für einen risikolosen Einsatz von Java Applets im Rahmen der GLOBUS-Entwicklungen erarbeiten soll.

Im vierten Kapitel **'Von AlfaWeb zur XfaWeb-Systemfamilie'** berichtet das FZK über den Stand der Entwicklung der Fachinformationssysteme AlfaWeb (Altlasten), NafaWeb (Naturschutz) und BofaWeb (Bodenschutz). Die drei Systeme bilden derzeit die Grundlage der Fachinformations-Systemfamilie XfaWeb.

Das fünfte Kapitel stellt die Arbeiten des IKE zur **'Anpassung des Personal Data Node an UIS-Nutzungen, insbesondere an das KFÜ-System'** vor und setzt damit direkt das entsprechende Kapitel des Abschlußberichts GLOBUS IV fort.

Das sechste Kapitel enthält - in Ergänzung der Darstellung des Vorhabens RK UIS in den Abschlußberichten der GLOBUS-Phasen III und IV - eine Zusammenfassung der **'Bestandsaufnahme und Bewertung von ausgewählten UIS-Systemen zur Fortschreibung und Weiterentwicklung der Rahmenkonzeption des Umweltinformationssystems Baden-Württemberg'**.

Im siebten Kapitel berichtet das FAW über die abschließende **'Dokumentation des Umwelt- und Verkehrs-Informationsservices Baden-Württemberg'**. Das System UVIS wurde vom FAW in den Phasen III und IV aus dem Umwelt-Führungs-Informationssystem UFIS entwickelt

Das achte Kapitel stellt die in der Phase II des Vorhabens HUDA erzielten Ergebnisse dar. In dieser Phase wurden für **'HUDA - Hypermediatechnik für Umweltdaten'** auf der Basis der 1996 erstellten Machbarkeitsstudie das Konzept und ein Funktionsprototyp erstellt.

Das letzte Kapitel enthält einen Ausblick auf die für 1999 im Rahmen von GLOBUS geplanten F+E-Arbeiten.

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/ R. Mayer-Föll, A. Jaeschke (Hrsg.);
Projekt GLOBUS - Vorstudie zur Konzeption und prototypischen Realisierung
einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten.
Phase I 1994
Wissenschaftliche Berichte FZKA 5863
- /5/ R. Mayer-Föll, A. Jaeschke (Hrsg.);
Projekt GLOBUS - Konzeption und prototypische Realisierung
einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten
im Umweltinformationssystem Baden-Württemberg
Phase II 1995
Wissenschaftliche Berichte FZKA 5700

- /6/ 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
Wissenschaftliche Berichte FZKA 5900
- /7/ R. Mayer-Föll, A. Jaeschke (Hrsg.);
Projekt GLOBUS - Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg
Phase IV 1997
Wissenschaftliche Berichte FZKA 6000
- /8/ R. Mayer-Föll, J. Pätzold (Hrsg.)
Umweltinformationssystem Baden-Württemberg - Rahmenkonzeption 1998
Ministerium für Umwelt und Verkehr Baden-Württemberg / Universitätsverlag Ulm
1998
- /9/ R. Mayer-Föll, J. Pätzold (Hrsg.)
Umweltinformationssystem Baden-Württemberg - Fortschreibung und Weiterentwicklung der Rahmenkonzeption - Hauptuntersuchung
Ministerium für Umwelt und Verkehr Baden-Württemberg, Stuttgart / Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW) Ulm
1998
- /10/ R. Mayer-Föll, A. Keitel, W.-F. Riekert, K. Tochtermann (Hrsg.)
Hypermediatechnik für Umweltdaten - Phase I Machbarkeitsstudie
Forschungsbericht 101 05 060 des Umweltbundesamtes, Berlin
1997
- /11/ A. Keitel, R. Mayer-Föll, K. Tochtermann, W.-F. Riekert (Hrsg.)
Hypermediatechnik für Umweltdaten - Phase II
Konzeption und Erstellung eines Funktionsprototyps
Forschungsbericht 209 01 222 des Umweltbundesamtes, Berlin
1997
- /12/ Ministerium für Umwelt und Verkehr Baden-Württemberg (Hrsg.);
Rahmenkonzeption für das Informationssystem Wasser, Abfall, Altlasten, Boden (WAABIS)
als Teil des Umweltinformationssystems Baden-Württemberg (UIS)
Stuttgart
1997.
- /13/ Ministerium für Umwelt und Verkehr Baden-Württemberg (Hrsg.);
Hauptuntersuchung WAABIS Informationssystem Wasser, Abfall, Altlasten, Boden,
Abschlußbericht
Stuttgart
1997.
- /14/ Ministerium für Umwelt und Verkehr, debis Systemhaus:
Spezifikation der Teilerneuerung der Kernreaktor-Ferüberwachung Baden-Württemberg
Stuttgart
1997.

GISterm

**Weiterentwicklung von *GISterm* zu einer
allgemeinen Komponente für den Zugriff und
die Visualisierung von räumlichen Daten.**

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

1. EINLEITUNG	11
2. NUTZUNGSARTEN UND DATENQUELLEN DES GISTERM.....	12
3. AUSBAU DES FUNKTIONSUMFANGS.....	14
3.1 BESCHRIFTUNG VON KARTENOBJEKTEN.....	16
3.2 KLASSIFIKATION (STATISTISCHE AUFBEREITUNG)	17
3.3 SPATIAL URL BROWSER.....	19
3.4 ONLINE-HILFE	20
4. GEODATENHALTUNG.....	21
4.1 GEODATENHALTUNG IN TRADITIONELLEN GEOINFORMATIONSSYSTEMEN.....	21
4.2 DATENBANKBASIERTE KOMPLETTLÖSUNGEN	22
4.3 DATENBANKBASIERTE GEODATENHALTUNG IN GISTERM.....	23
4.3.1 <i>Der Shapefile-Adapter</i>	23
4.3.2 <i>Der JDBC GIS-Adapter</i>	23
4.3.2.1 JDBC-Anbindung	23
4.3.3 <i>Verwaltete Daten</i>	24
4.3.3.1 Metadaten	24
4.3.3.2 Attributdaten.....	25
4.3.3.3 Vektordaten.....	25
4.3.3.4 Rasterdaten	25
4.4 DIE FACHKOMponente: DER JDBC RIPS-ADAPTER	27
4.4.1 <i>Architektur</i>	27
4.4.2 <i>Funktionalität</i>	28
4.4.2.1 Schemaabfrage.....	28
4.4.2.2 Abfrage der Bounding Box von Geobjekten	28
4.4.2.3 Geobjektabfrage	28
4.4.2.4 Veränderung von Geobjekten	29
4.4.2.5 Datenimport	29
5. LITERATUR.....	30

1. Einleitung

Ziel des *GISterm* (Produktbezeichnung und Akronym für GIS-Terminal) ist die Entwicklung einer allgemeinen und flexiblen Abfrage-, Visualisierungs- und Eingabekomponente für räumliche Daten. In einem Umweltinformationssystem wie dem UIS Baden-Württemberg werden eine Vielzahl von räumlichen Daten erhoben und verwaltet. Diese Raumdaten, oft auch als Geodaten bezeichnet, nehmen einen hohen Stellenwert in einem solchen Umweltinformationssystem (UIS) ein, da der überwiegende Teil der Daten einen Raumbezug (sei er direkt oder indirekt) besitzt. Um einem Nutzer des UIS diese Rauminformation zur Verfügung zu stellen, benötigt man an vielen Stellen des Informationssystems eine GIS-Komponente, welche Geodaten erfragen, visualisieren oder erzeugen kann. An eine solche GIS-Komponente werden, bedingt durch ihren Einsatz in vielen auch von ihrer Art unterschiedlichen Nutzungsszenarien, eine Reihe von Anforderungen gestellt. Eine solche Komponente muß interoperabel mit vorhandenen Datenquellen sowie weiteren Komponenten des Informationssystem sein. Sie sollte betriebssystemneutral einsetzbar sein und sich ebenfalls gut in Web-Umgebungen integrieren lassen. Nicht zuletzt ist die Skalierbarkeit der Komponente hinsichtlich der Komponentengröße und der Nutzung unterschiedlich granulierter Datenquellen (Files, Datenbanken) eine wichtige Anforderung.

In den vorausgegangenen Projektphasen wurde die grundlegende Softwarearchitektur sowie ein funktionsfähiger Prototyp des *GISterm* entwickelt und in /1/, /2/, /3/ ausführlich beschrieben. Die Architektur berücksichtigt durch eine Vielzahl von modernen Softwarekonzepten die sehr allgemeinen Anforderungen. So wird beispielsweise durch das Konzept der Datenquellen-Adaptoren allgemein die Interoperabilität zu vorhandenen, als auch zu zukünftigen Datenquellen gewährleistet. Durch die moderne und objektorientierte Programmiersprache Java, in dem das *GISterm* implementiert wird, ergibt sich der betriebssystemunabhängige Einsatz der Komponente. Weiterhin wird durch die moderne dreischichtige Architektur (multi-tier architecture) der Einsatz sowohl im Intra- bzw. Intranet als auch als Applikation ermöglicht. Zunehmend wichtiger wird auch der Einsatz in Softwarekomponentenszenarien, welche im *GISterm* durch die Realisierung der Teilmodule als Java/Beans /5/ unterstützt werden. Durch die beiden zuletzt genannten Softwarekonzepte, also die Mehrschichtenarchitektur und Softwarekomponententechnik, ist auch die Anforderung der Skalierbarkeit hinsichtlich Einsatzszenario und anzubindender Datenquellen erfüllbar.

Die in diesem Beitrag beschriebenen Funktionalitäten und Konzepte schildern die Arbeiten die in dieser Projektphase geleistet wurden, um den vorhandenen Prototyp so weiterzuentwickeln, daß er flexibel und allgemein einsetzbar ist. Der erste Abschnitt erläutert, in welchen technischen Anwendungsumgebungen und den jeweils dafür zur Verfügung stehenden Datenquellen das *GISterm* bzw. dessen Subkomponenten eingesetzt werden kann. Danach werden einige in dieser Projektphase neu hinzugekommenen Funktionalitäten zum Explorieren und Präsentieren von Raumdaten geschildert. Eine detaillierte Darstellung aller Funktionalitätserweiterungen würde den Rahmen dieses Berichts sprengen. Im letzten Abschnitt wird die neu hinzugekommene Basisfunktionalität der Geodatenhaltung in der relationalen Datenbank Oracle im Detail erläutert.

2. Nutzungsarten und Datenquellen des *GISterm*

Dieser Abschnitt gibt eine Übersicht über die verschiedenen technischen Anwendungsumgebungen in denen die *GISterm*-Komponenten und die damit verbundenen Datenquellen genutzt werden können. Bedingt durch die Mehrschichtenarchitektur sowie die Komponentenstruktur innerhalb der einzelnen Schichten ergeben sich eine Reihe von Einsatzmöglichkeiten und Einsatzkombinationen die nacheinander vorgestellt werden. Abbildung 2.1 zeigt ein Schaubild, in dem die Systemkomponenten des *GISterm* und ihre Datenquellen dargestellt werden.

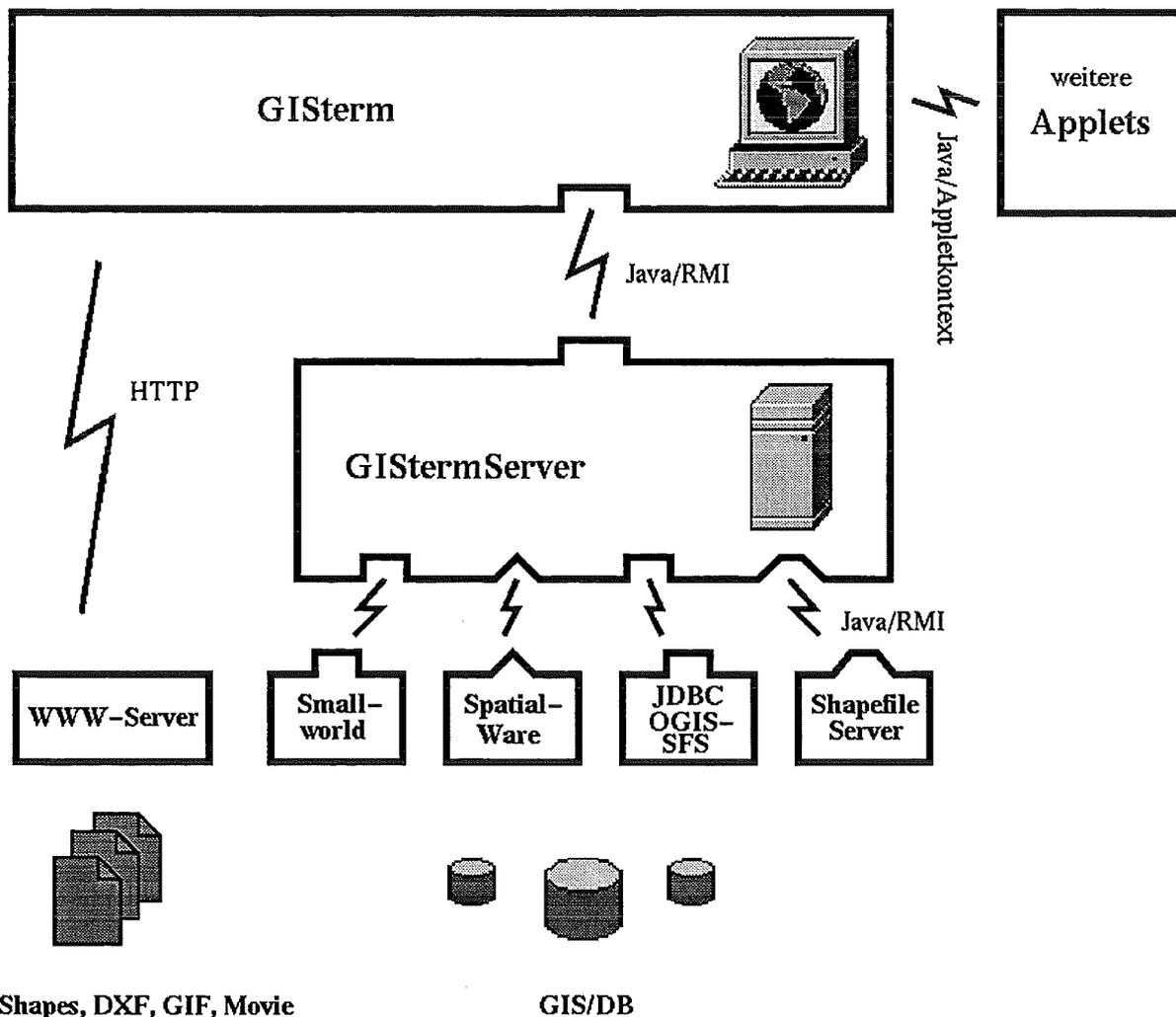


Abbildung 2.1: Systemkomponenten und Datenquellen des *GISterm*

Zunächst ist der Klient des *GISterm* (oberste Schicht) unabhängig von dem Serverteil (mittlere Schicht, *GIStermServer*) nutzbar, sofern in der Anwendung kein Onlinezugang zu GIS-Datenbanken erforderlich ist. Dieser Klient kann dann Geodaten über Dateien (URL-Ressourcen) die im Filesystem oder auf einem oder mehreren Web-Server vorgehalten werden zur Kartenpräsentation nutzen. Zum Lesen der Daten wird in diesem Fall das HTTP-Protokoll benutzt. Der Vorteil dieser Nutzungsart ist, daß auf dem Datenserver (der auch mit dem Klientenrechner identisch sein kann) keine weiteren Komponenten nötig sind. Auch ser-

verseitige Sicherheitsmechanismen wie Firewalls bereiten in diesem Fall keine Probleme. Somit eignet sich diese Form uneingeschränkt für das Intra- wie auch das Internet. Dieser Kliententeil ist sowohl als Java-Applet als auch als Java-Applikation realisiert. Dadurch kann er einfach in eine Web-Umgebung eingebunden oder auch direkt auf dem Arbeitsplatzrechner des Benutzers installiert werden. Der Kliententeil besitzt weiterhin eine klientenseitige Datenschnittstelle (siehe UIS-Anwendungsschnittstelle in /2/), mittels der er mit anderen Komponenten direkt Daten austauschen kann. Über diese Datenquellen können Anwendungsentwickler anwendungsspezifische Daten an das *GISterm* übergeben oder einfache Funktionserweiterungen in Form von selbst entwickelten Java-Applets vornehmen. Ein Beispiel für die Nutzung dieser Einsatzform ist der Einsatz von *GISterm* zur hypermedialen Präsentation von Umweltberichten im Projekt HUDA. Hier stehen derzeit keine Online-Datenquellen zur Verfügung. Der Anwendungsentwickler stellt eine Kollektion von Karten- und Meßdaten für den bestimmten Einsatzfall zusammen und konfiguriert das *GISterm* so, daß der Endbenutzer direkt und nur mit dieser Kollektion arbeitet.

Wird in einem Anwendungsfall der Zugriff auf Online-Daten aus GIS-Datenbanken benötigt, kommen sowohl der eben beschriebene Kliententeil als auch der Serverteil *GIStermServer* des *GISterm* zum Einsatz. Dieser Serverteil fungiert als Applikationsserver und realisiert somit eine Schicht zwischen den Datenbanken und dem *GISterm*-Klienten. Die Netzchnittstelle zwischen dem *GISterm*-Klienten und dem *GIStermServer* ist über die Middleware-Technologie Java/RMI /4/ realisiert. Der Serverteil kommuniziert mit der Datenbank, abhängig vom angesprochenen Datenbanksystem, über JDBC oder Java/RMI. Dabei wird jedes Datenbanksystem über einen dedizierten Adapter, der Teil des *GIStermServer* ist, angesprochen. Dieser Adapter abstrahiert von dem realen System und bietet eine einheitliche GIS-Schnittstelle an. In einem Anwendungsszenario kann der *GIStermServer* so konfiguriert werden, daß nur ausgewählte Datenbanksysteme angesprochen werden. Die vom Kliententeil angesprochenen Datenquellen (URL-Ressourcen und Klientenschnittstelle) stehen in dieser Nutzungsart weiterhin zur Verfügung. Diese Nutzungsart ist ebenso als Java-Applikation als auch als Java-Applet nutzbar und kann dadurch ebenso in eine Web-Umgebung integriert werden. Im Falle einer Applikation kann die Netzchnittstelle zwischen Klienten- und Serverteil entfernt werden, wodurch das System an Performanz gewinnt. Die derzeitige Einschränkungen dieser Technologie sind lediglich Server die hinter einem Firewall platziert sind sowie die zur Verfügung stehende Netzbandbreite.

Bislang wurden nur die allgemeinen Komponenten zum Erstellen und Explorieren von Karten vorgestellt und direkt sowie unverändert benutzt. In einem Entwicklungsszenario, in dem neue Software von einem Entwickler erstellt wird, kann das *GISterm* ebenso zum Einsatz gelangen. Hier stehen zwei Varianten zur Verfügung, die sich vor allem in der Granularität der Softwareentwicklung unterscheiden.

Zum einen können die einzelnen Softwarekomponenten (Java/Beans, /5/), aus denen das *GISterm* besteht, direkt in andere Applikation eingebaut werden. Entsprechend der Anwendungslogik, die innerhalb einer Komponente implementiert ist, unterscheidet das *GISterm* dabei zwei Typen von Softwarekomponenten:

- *Basiskomponenten*: Diese bieten eine Funktionalität, die in vielen Anwendungsfällen benötigt wird, realisieren jedoch selbst keine Anwendungslogik. Ein Beispiel wäre die View-Komponente zur Visualisierung von Vektor- oder Rastergeometrien.
- *Fachkomponenten*: Das sind Anwendungskomponenten, die meist basierend auf Basiskomponenten implementiert werden, die jedoch eine Anwendungsbezogene Schnittstelle nach außen bieten und dementsprechend auch eine Anwendungslogik innerhalb der Komponente implementieren. Ein Beispiel für diesen Komponententyp ist eine Komponente zur Visualisierung von Geometrien eines bestimmten GIS-Datenschemas. Für das UIS Baden-Württemberg wird hierfür eine RIPS-Bean implementiert, welche auf die Besonderheiten der Datenmodelle eingeht. Diese Komponente kann dann direkt mit Objekten umgehen, welche durch einen UIS-OAC oder eine UIS-ID (vormals RIPS-ID) definiert sind.

Es ist auch möglich, aufbauend auf den allgemeinen Komponenten das *GISterm* als Framework für spezifische Anwendungen zu nutzen. Unter einem Framework versteht man eine Menge von Klassen, welche die Elemente eines Wiederverwendbaren Entwurfs für eine bestimmte Softwareart darstellen. Ein Anwendungsentwickler paßt das Framework für eine bestimmte Anwendung an, indem er Unterklassen der Frameworkklassen bildet und ihre Objekte zusammensetzt /6/. Dadurch kann der Entwickler direkt auf der Standardfunktionalität des *GISterm* aufsetzen und nur spezifische Anwendungsfunktionalitäten hinzufügen, womit eine drastische Produktivitätssteigerung in der Softwareentwicklung verbunden ist.

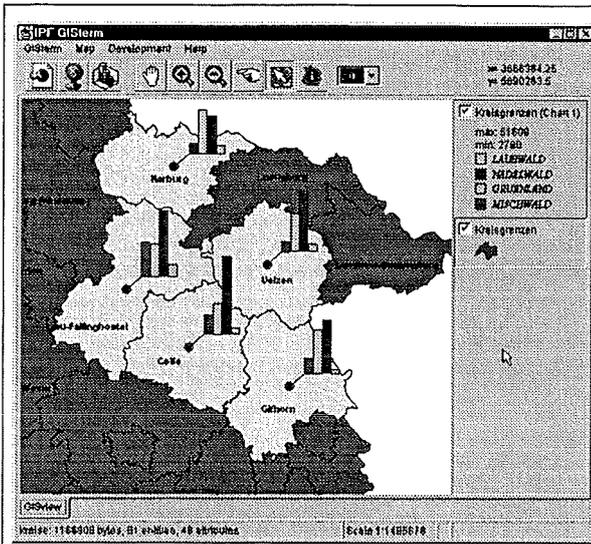
3. Ausbau des Funktionsumfangs

In dieser Projektphase wurde die allgemeine Funktionalität des *GISterm* soweit entwickelt, daß es bereits in vielen Anwendungsszenarien nützlich eingesetzt werden kann. Primäres Ziel dabei war es, die Grundfunktionalität zum Kreieren, Präsentieren, Explorieren und Ausgeben von Karten zu entwickeln. Die wichtigsten dazu nötigen Funktionalitäten, sofern sie in dieser Projektphase entwickelt wurden, werden in diesem Abschnitt beschrieben. Die Erweiterung um die Kernfunktionalität der Geodatenhaltung in Oracle wird im nächsten Abschnitt gesondert und detailliert erläutert.

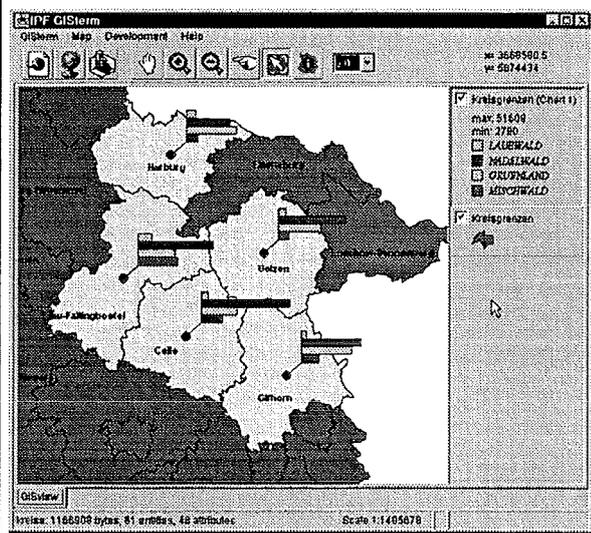
Kartogramme zur Visualisierung von numerischen Meßreihen mit geographischen Bezügen

Das *GISterm* wurde um ein Softwaremodul erweitert, daß die graphische Ausgabe von Diagrammen mit geographischen Bezügen innerhalb der Karte erlaubt. Diese Diagramme können von anderen Komponenten (z.B. Sachdatenkomponente) an der Klientenschnittstelle übergeben oder aber auch direkt aus Geodaten abgeleitet werden. Kartogramme werden gleichberechtigt mit Geodaten in einem eigenen Datenlayer pro Darstellungsthema verwaltet. Ein Benutzer kann interaktiv in die Diagrammpositionierung eingreifen, so daß eine ansprechende Datenpräsentation gewährleistet ist. Weiterhin kann ein Benutzer einfach zwischen verschiedenen Präsentationsarten (Balken-, Liniendiagramme usw.) wechseln. Es ist ebenfalls möglich, Daten die in einem oder mehreren Kartogrammen angezeigt werden nur als Business-Diagramm (mit dem in der vorausgehenden Projektphase entwickelten Business-Diagramm Modul), also ohne Geobezüge, zu visualisieren und beliebig zwischen den Präsentationsarten zu wechseln. Abbildung 3.1 zeigt die Möglichkeiten an einem Beispiel in dem

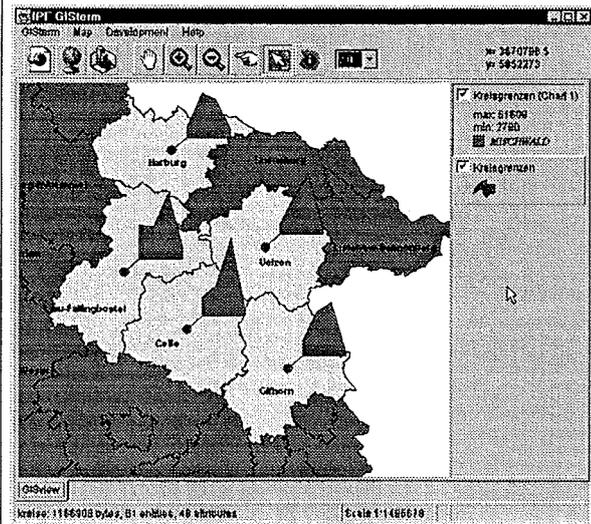
die Wald- und Grünflächenanteile in bestimmten Kreisen von Niedersachsen als Kartogramme visualisiert werden.



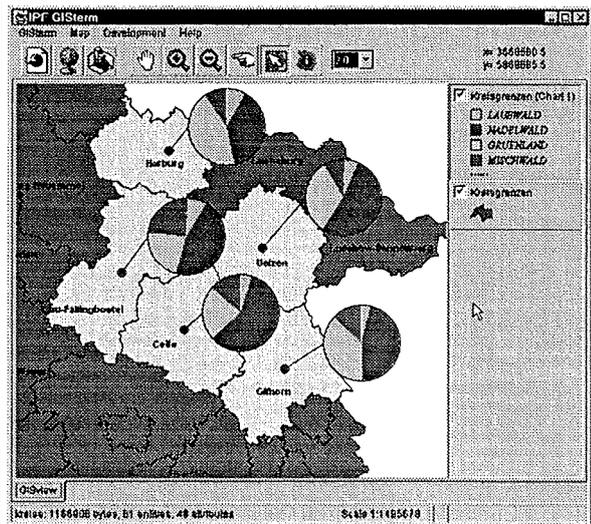
Horizontale Balkendiagramme



Vertikale Balkendiagramme



Liniendiagramme



Kuchendiagramme

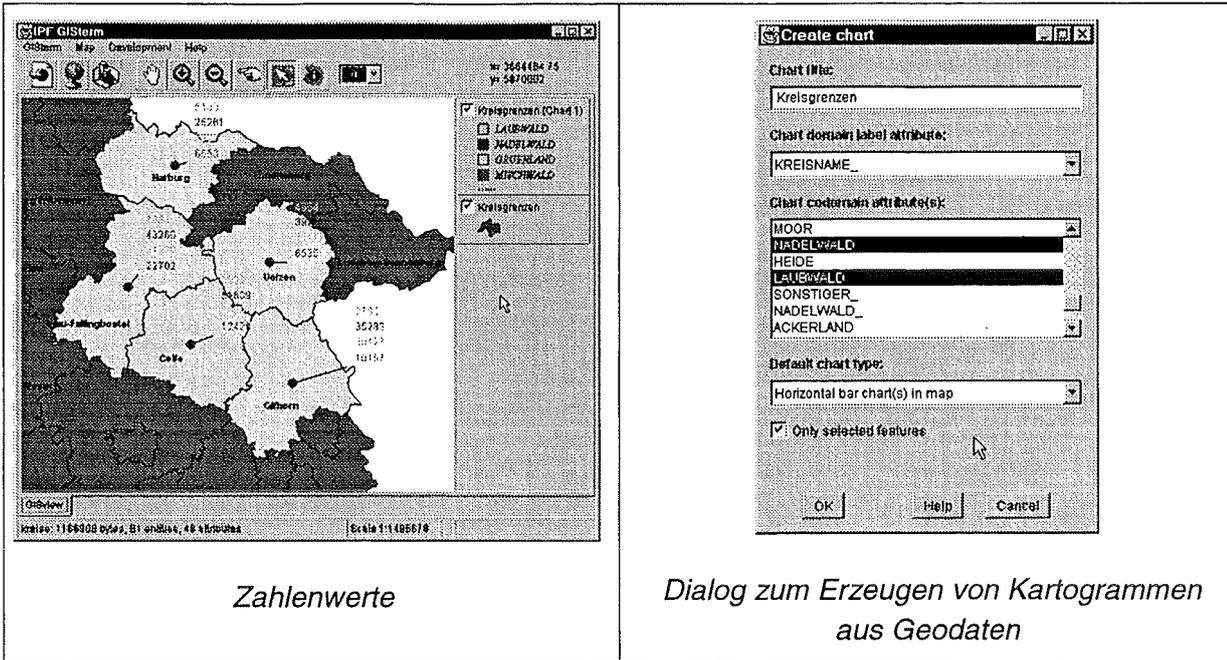


Abbildung 3.1: Visualisierungsarten für Kartogramme

3.1 Beschriftung von Kartenobjekten

Es wurde ein Softwaremodul entwickelt, das die automatische Beschriftung von beliebigen Kartenobjekten ermöglicht. Dabei können alle Objekte oder nur selektierte Objekte mit einem zuvor bestimmten Sachattribut beschriftet werden. Die folgende Abbildung 3.2 zeigt eine Karte in der von der Beschriftungsmöglichkeit in vielen Datenlayern Gebrauch gemacht wurde.

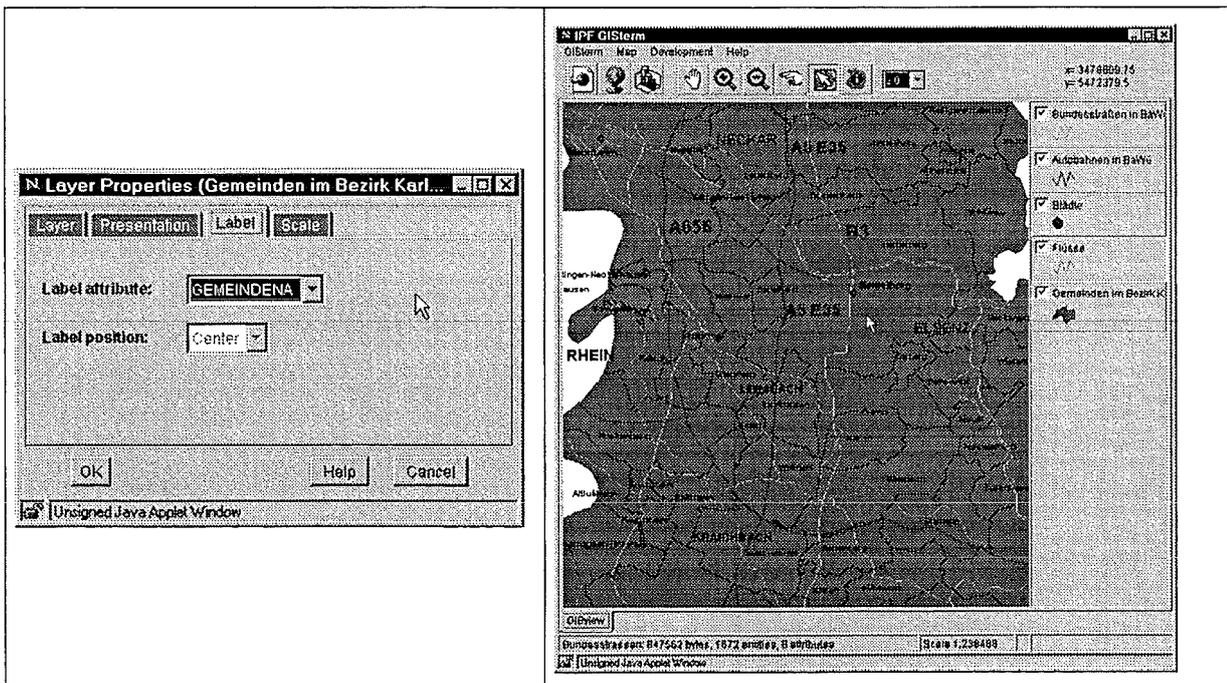


Abbildung 3.2: Beschriften von Kartenobjekten

3.2 Klassifikation (Statistische Aufbereitung)

Mit dieser Funktionalität können Geoobjekte eines Datenlayer nach Attributwerten klassifiziert werden. Ein Benutzer besitzt dadurch die Möglichkeit thematische Karten zu erstellen. Thematische Karten sind die Grundlage für zweckmäßige Analysen der mit einer Karte verbundenen Geodaten. Die Kartenobjekte können dabei entsprechend der Datenwerte schattiert dargestellt werden. In einer Klassifizierung werden die Objekte anhand eines Attributes untersucht und entsprechend der Attributwerte dieses Attributes in Klassen eingeteilt. Derzeit stehen folgende Klassifikationen bereit:

"*unique value*": Mittels dieser Klassifizierungsmethode werden die einzelnen Werte eines Attributs durch jeweils andere Farben dargestellt. In einer Landnutzungskarte beispielsweise würde daher jeder Nutzungstyp durch eine andere Farbe dargestellt, z.B. blau für Wohngebiete, gelb für Industriegebiete und grün für öffentliches Land. Im allgemeinen eignet sich die Klassifikation nach Einzelwerten für die kartographische Darstellung der folgenden drei Attributtypen: Attribute, aus denen Name, Typ, Zustand oder Kategorie eines Objekts hervorgeht, Attribute, die Messungen oder Mengen enthalten, die bereits klassifiziert sind (z.B. "0-99", "100-199") und Attribute, die ein Objekt eindeutig identifizieren (z.B. Verwendung des Attributes Regionenname, um eine Karte zu erstellen, in der jede Region in einer anderen Farbe dargestellt ist). Die folgende Abbildung 3.3 zeigt ein Beispiel in dem Kreise von Niedersachsen nach Regionen klassifiziert wurden.

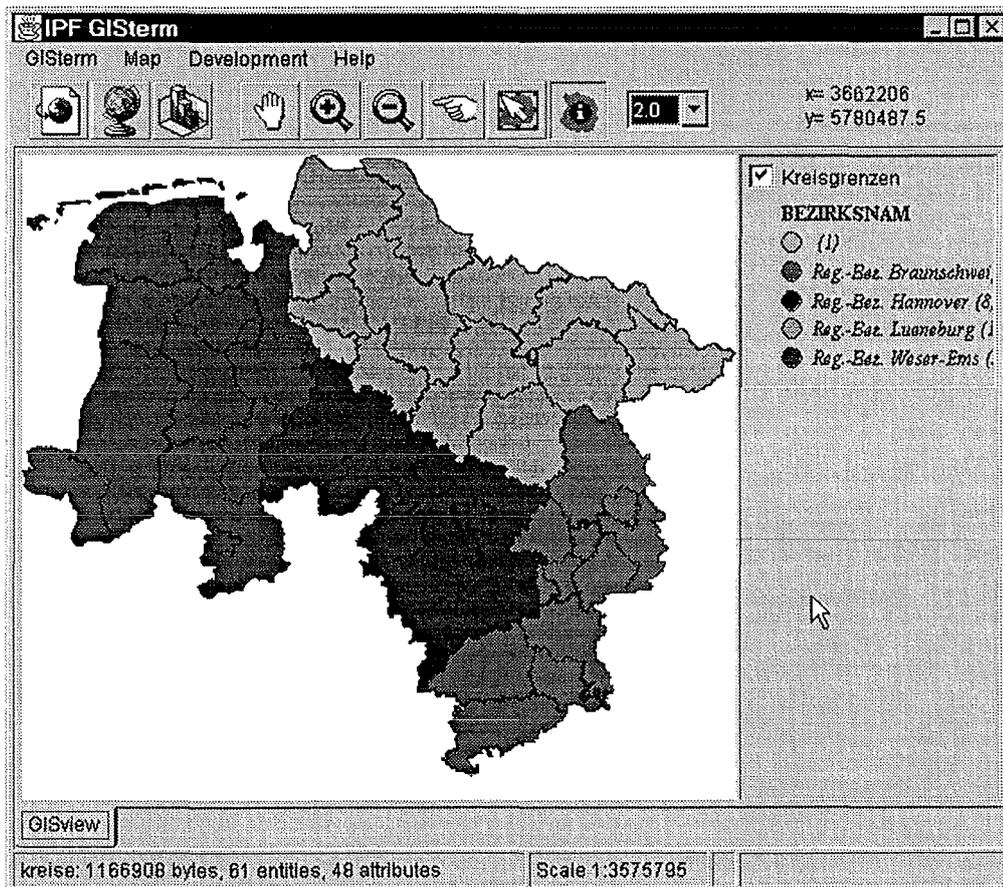


Abbildung 3.3: Klassifikation nach gleichen Attributwerten

"*equal interval*": Bei der Klassifizierung nach gleichem Intervall wird der Bereich der numerischen Attributwerte in gleich große Unterbereiche aufgeteilt. Wenn die Objekte beispielsweise Werte im Bereich von 12 bis 351 aufweisen, ist der Gesamtbereich dieser Werte 339. Wenn man diese Objekte dann nach gleichem Intervall in drei Klassen aufteilt, repräsentiert jede Klasse einen Bereich von 113, und die Klassenwertbereiche lauten daher: 12-125, 126-238 und 239-351. Abbildung 3.4 zeigt den Anteil von Ackerland in den Kreisflächen von Niedersachsen eingeteilt in 6 Klassen.

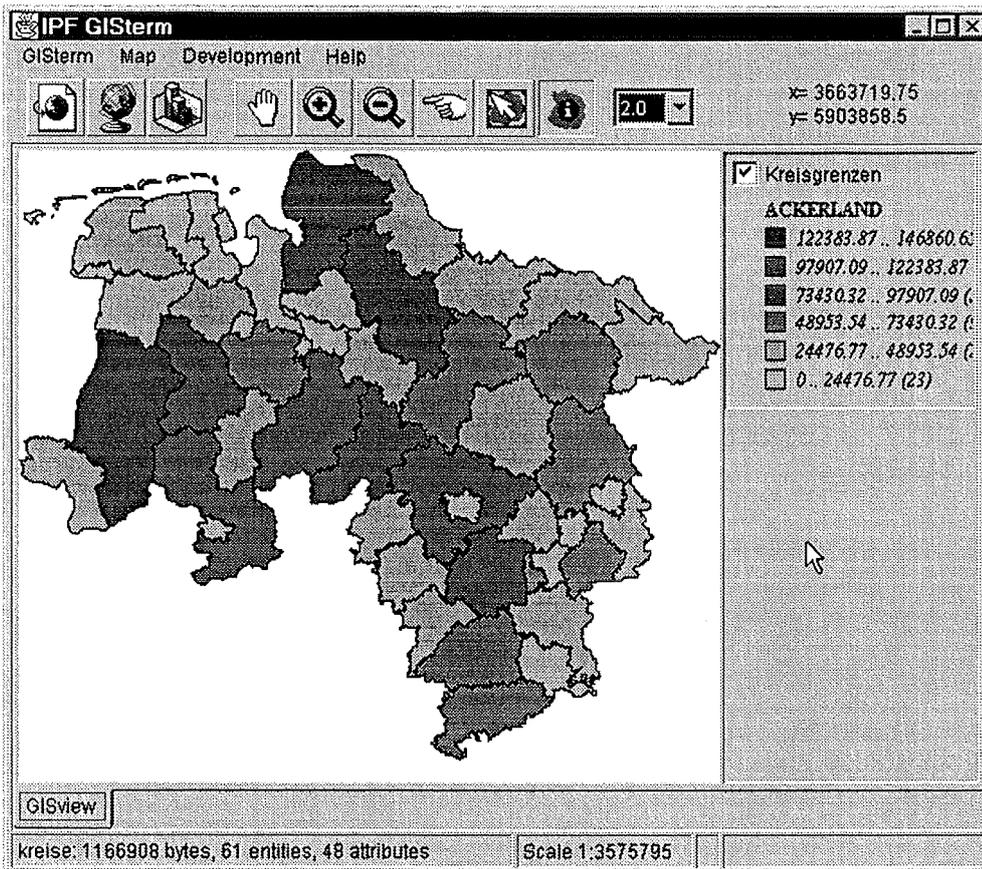


Abbildung 3.4: Klassifikation nach numerischen Intervallen

"*standard deviation*": Die Klassifizierungsmethode zeigt, wie stark die Werte eines Attributes vom Mittel aller Werte abweichen. Wenn Daten nach der Standardabweichung klassifiziert werden, ermittelt *GIStern* den Mittelwert und trennt die Klassen ober- und unterhalb des Mittelwerts. Abbildung 3.5 zeigt dieses Verfahren wieder am Beispiel der Ackerlandfläche in den Kreisen von Niedersachsen.

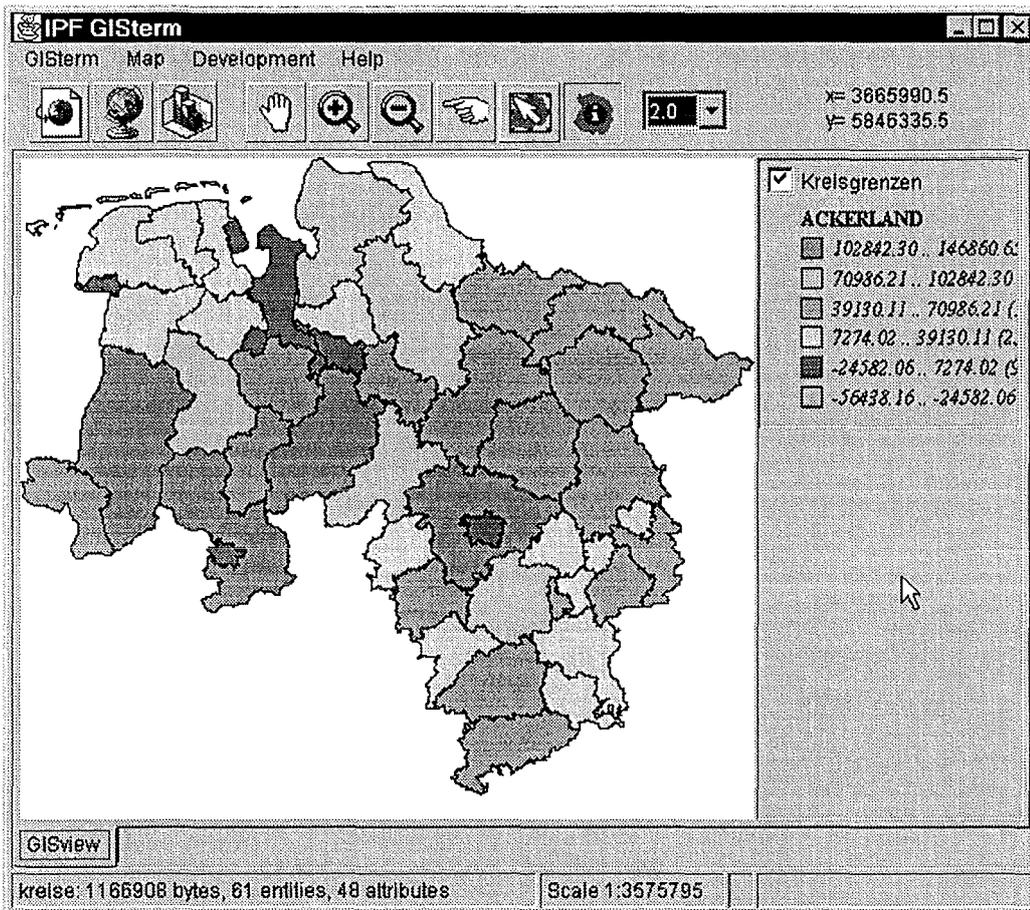


Abbildung 3.5: Klassifikation nach Mittelwert und Standardabweichung

3.3 Spatial URL Browser

Um dateiartige Datenquellen an das *GISterm* anzuschließen, wurde der Spatial-URL-Browser entwickelt. Dieser Browser zeigt einem Benutzer die zur Verfügung stehenden Datenressourcen in einer baumförmigen Struktur an. Abbildung 3.7 zeigt das zugehörige Dialogfenster.

In dem URL-Browser werden alle Dateiarten, mit denen das *GISterm* arbeiten kann, angezeigt und zur Auswahl angeboten. Der URL-Browser kann dazu entweder lokale Dateisysteme und/oder auch über das HTTP-Protokoll verteilte Datenbestände anzeigen. Die Datenbestände werden dabei beginnend von einer oder mehreren zu konfigurierenden Startadresse(n) aus dynamisch ausgelesen. D.h. immer wenn der Benutzer tiefer in die Verzeichnishierarchie, die gleichzeitig eine Themenhierarchie realisiert, einsteigt, wird der Inhalt des entsprechenden Knotens dynamisch ausgelesen. Zeigt der Knoten auf ein Verzeichnis in einem lokalen Filesystem, wird dieses ausgelesen. Zeigt der Knoten jedoch auf eine URL so wird der entsprechende Web-Server über das HTTP-Protokoll ausgelesen. Dies erfolgt durch Parsen (Interpretieren) der vom Web-Server zurückgegebenen Informationsseiten. Dadurch ist es möglich, sehr einfach und flexibel einen möglicherweise über mehrere Organisationseinheiten verteilten Datenbestand aufzubauen und dem *GISterm*-Anwender zur Verfügung zu stellen. Durch das dynamische Auslesen der Hierarchien kann der Anwender immer auf den aktuellen Datenbestand zugreifen ohne daß dafür administrative Maßnahmen nötig sind.

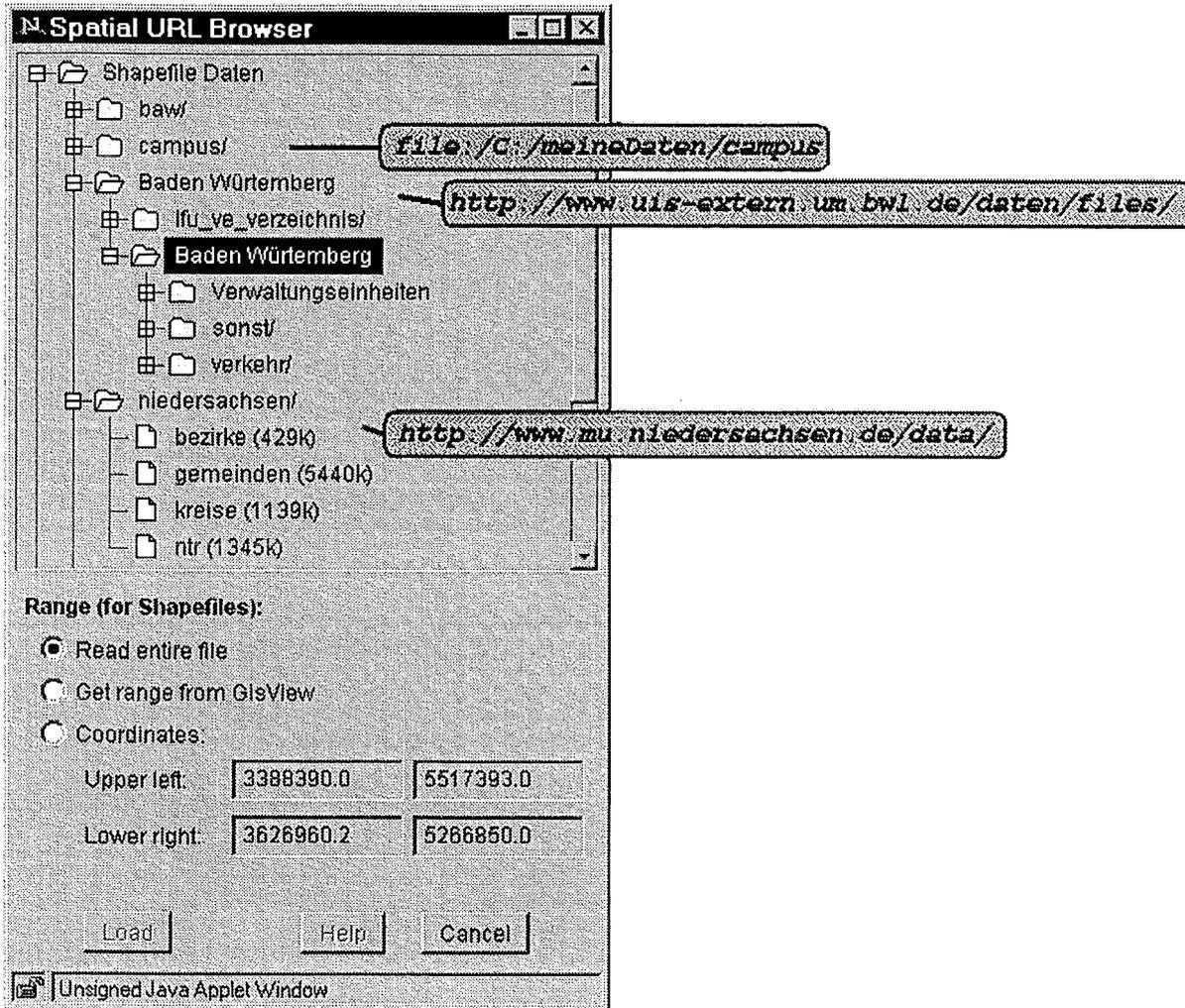


Abbildung 3.7: Spatial URL Browser

Weiterhin ist es möglich parallel zu den Datenressourcen Metadateien zu hinterlegen. Diese Metadateien werden, sofern sie vorhanden sind, eingelesen und zur Präsentation der Daten benutzt. Darüber ist es möglich, für eine neue URL-Ressource eine Defaultpräsentationsart (Farbe, Beschriftungsattribut und -position, Klassifikation usw.) zu definieren und sofort allen *GISterm* Nutzern zur Verfügung zu stellen.

3.4 Online-Hilfe

Das *GISterm* verfügt über eine Online-Hilfe. Diese kann sowohl als Funktionsreferenz als auch als direkte Hilfe zu Dialogelementen genutzt werden. In jedem Dialog wurde hierzu eine Hilfefunktion integriert, so daß auch ungeübte Benutzer mit dem *GISterm* umgehen können. Ein kleines Online-Tutorial gibt neuen Nutzern eine kurze Einführung in die Bedienung von *GISterm*. Abbildung 3.6 zeigt Ausschnitte aus der Hilfefunktionalität.

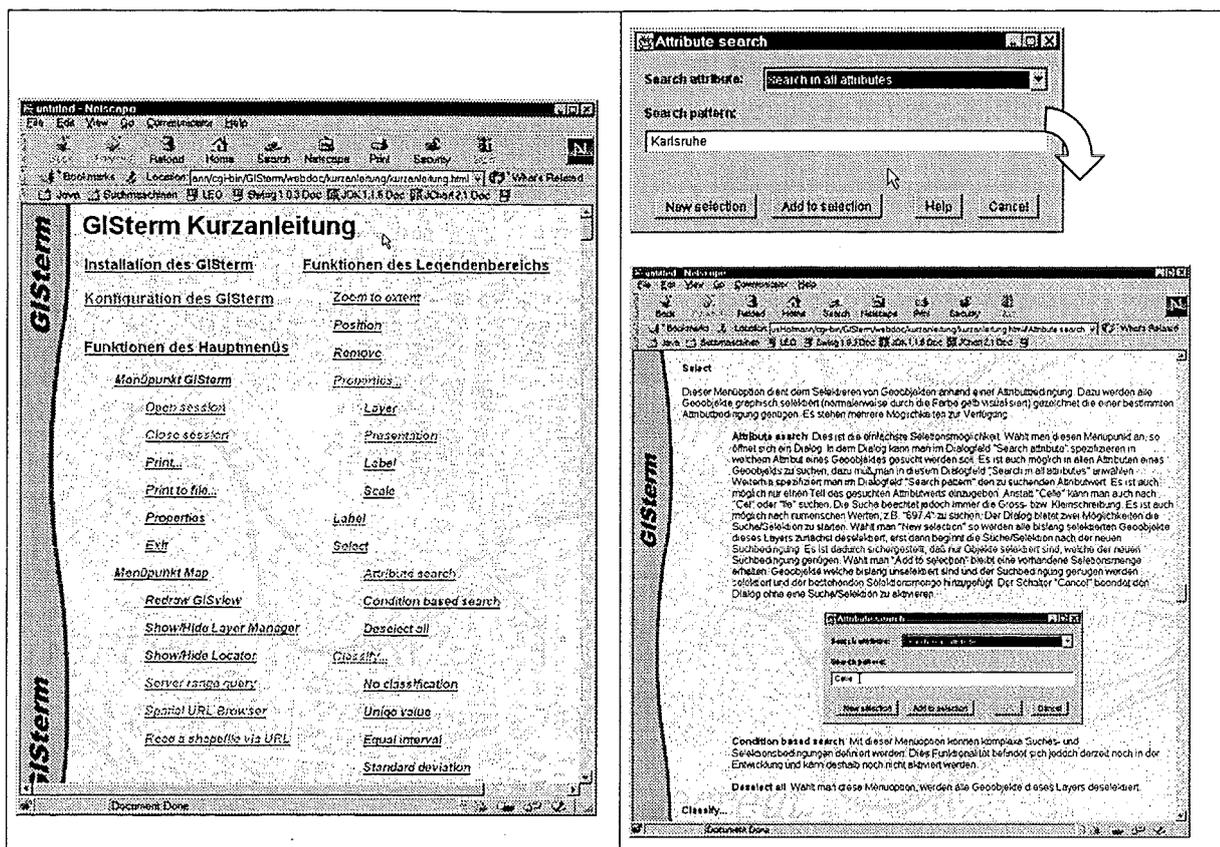


Abbildung 3.6: Hilfefunktion des GISterm

4. Geodatenhaltung

4.1 Geodatenhaltung in traditionellen Geoinformationssystemen

Die Mehrheit traditioneller Geoinformationssysteme hält ihre Geometrie- und Attributdaten (Sachdaten) separat voneinander vor, d.h. Geometrie wird z.B. im Filesystem, die dazu gehörenden Attribute in einer kommerziellen Datenbank gespeichert. Für diese Aufspaltung der Daten gibt es mehrere Gründe: sowohl historische als auch technische. Frühere Systeme wurden auf existierenden graphischen Dateiformaten (meistens aus dem CAD-Bereich) aufgebaut. CAD-Systeme zum Beispiel bieten eine ausgereifte graphische Funktionalität, eine gewohnte Benutzerschnittstelle und oft auch eine Programmierschnittstelle (API) für Entwickler. Die Zuordnung zwischen Graphik und Attribute erfolgt durch proprietäre Indexe, was einerseits sehr einfach ist und elegant zu sein scheint, andererseits aber zu schweren Problemen bezüglich Datenkonsistenz führen kann, denn der Update der Attribut- bzw. Geometriedaten findet getrennt voneinander statt.

Ein wesentliches Merkmal von Geoinformationssystemen ist, daß sie häufig mit enormen Datenmengen arbeiten müssen. Der Performanz kommt daher überaus große Bedeutung zu, denn die Wartezeiten müssen – trotz der großen Menge an Daten – in einem für den Benutzer akzeptablen Bereich gehalten werden. In diesem Zusammenhang ergeben sich Vorteile aus der getrennten Datenhaltung. Weil die Verwaltung der Attribut- und Geometriedaten von

separaten Systemen verwirklicht wird und diese auf ihre jeweiligen – aus dem Gesichtspunkt des GIS beschränkte – Aufgaben optimiert sind, kann die Performanz solcher Systeme bei einfachen Aufgaben deutlich über der modernerer Geoinformationssysteme liegen. Genau hier liegt aber auch ein weiterer Nachteil: die Zuordnung zwischen den Datenkomponenten muß hergestellt werden und das kostet Zeit (und reduziert letztlich die Performanz). Außerdem können bei komplexeren Aufgaben erhebliche Schwierigkeiten wegen des gespaltenen Datenmodells auftreten.

Solange die Daten ausschließlich durch ein einziges System zugänglich und änderbar sind, kann die Konsistenz vom System gewährleistet werden. Im Mehrbenutzerbetrieb soll aber das System darauf achten, daß schreibende Operationen an einer Dateneinheit nicht gleichzeitig von mehreren Benutzer durchgeführt werden. Systeme, welche die Sachdaten auch in separaten Files verwalten (z.B. Lösungen, die Shapefiles verwenden), müssen sich mit noch schwereren Problemen bezüglich der Konsistenz auseinandersetzen. Dürfen mehrere Systeme an demselben Datensatz parallel arbeiten, wird die Gewährleistung der Datenkonsistenz ohne gemeinsamen Transaktionsrahmen unmöglich.

4.2 Datenbankbasierte Komplettlösungen

Um die bisher aufgezählten Probleme zu vermeiden, werden verschiedene Lösungen herangezogen. Die offensichtlichste von denen ist es, wenn man alle Komponenten der Geodaten – also Attribute und Geometrien – zusammen in einer Datenbank ablegt. Dies bietet den Vorteil, daß alle Daten und alle auszuführende Operationen auf diesen Daten vom Datenbanksystem überwacht werden. Dadurch können sämtliche von dem Datenbanksystem angebotene Techniken, wie Parallelisierung, konkurrente Operationsausführung, verteilte Datenhaltung und Sicherheitssystem transparent ausgenutzt werden. Insbesondere relationale Datenbanksysteme erfreuen sich in letzter Zeit immer größerer Popularität im Bereich der Geodatenhaltung.

Relationale Datenbanksysteme sind im Grunde eher für Sachdatenmanagement geeignet. Wenn man Geometrien mitverwalten möchte, benötigt man effiziente Speicherungs- bzw. Anfragemechanismen für diese Art von Daten. Geometriedaten haben keine feste Größe, sind zweidimensional und strukturiert; sie können also nicht wie einfache Datentypen behandelt werden. Sie werden üblicherweise entweder als BLOB (binary large object) oder wie normale Typen aber in mehreren Tabellen verteilt (falls die Datenbank auch Arrays von variabler Länge unterstützt, nur in einer Tabelle) abgelegt. Des weiteren muß der implementierte Anfragemechanismus die neuen Datenstrukturen performant behandeln können. Eine bessere Lösung ist es, wenn die Anfragesprache (SQL) um Geoooperatoren ergänzt wird, um alle Datenarten gleichberechtigt in einer Anfrage benutzen zu können.

Beispiele für solche auf dem Markt erhältliche Komplettlösungen sind die SDE von der Firma ESRI, die Spatial Cartridge von ORACLE und SpatialWare von MapInfo. Sie sind meistens zwar leistungsfähig, aber dementsprechend auch sehr teuer in der Anschaffung.

4.3 Datenbankbasierte Geodatenhaltung in GIStern

4.3.1 Der Shapefile-Adapter

Die Adapter in der Architektur des *GIStern* haben die Aufgabe, durch Abstraktion einen einheitlichen Zugriff auf existierende Geoinformationssysteme zu gewährleisten. Überdies kann auch auf Fileserver zugegriffen werden, die ebenfalls mit der Schnittstelle des Adapters versehen sind und sich daher ähnlich wie alle – wirkliche Systeme abstrahierende – Adapter verhalten.

Ein Beispiel für einen solchen Fileserver ist der Shapefile-Adapter, welcher in der vorausgehenden Projektphase /2/ implementiert wurde. Der Shapeadapter benutzt das Shapefile-Format von ESRI. Bei diesem Format werden Geodaten eines Themas jeweils in drei Dateien gespeichert: die Geometrie in der ersten, die Sachattribute in der zweiten und der die Zuordnung zwischen den beiden Geodatenkomponenten herstellende Index in der dritten Datei. Neben den Vorteilen, die sich aus der Einfachheit des Formats ergeben, besitzt der Adapter alle Nachteile der vorher erwähnten filebasierten Systeme. Er kann deshalb keine schreibende Operationen durchführen, ohne die Konsistenz der Daten zu gefährden.

4.3.2 Der JDBC GIS-Adapter

Die Notwendigkeit von schreibenden Zugriffen, die parallel auf demselben Datenbestand operieren können, hat zur Entwicklung eines datenbankbasierten Adapters geführt. Es kann auch so betrachtet werden, daß der filebasierte Shapefileserver, der im Rahmen des Projektes Web-Szenario /2/ entwickelt wurde, auf eine datenbankbasierte Lösung umgestellt wurde. Geometrien von Geoobjekten werden bei dieser Lösung in Tabellen in der Datenbank abgelegt. Die Geometriedaten können dadurch zusammen mit den Sachdaten in einer Datenbank gehalten, was u.a. das Datenmanagement erleichtert. Der Transaktionsrahmen wird vom Datenbanksystem zur Verfügung gestellt. Beliebige Transaktionen von gleichzeitig mit dem System arbeitenden Benutzern können quasi parallel abgehandelt werden.

Selbstverständlich soll dieser Datenserver – im folgenden JDBC GIS-Adapter genannt – auch für das Importieren von Geodaten (Shapefiles, z.B. aus RIPS exportiert) einen geeigneten Mechanismus anbieten.

4.3.2.1 JDBC-Anbindung

Durch die Java Database Connection (JDBC) als integraler Teil der Java Klassenbibliotheken ist die Möglichkeit gegeben, auf Daten in einer relationalen Datenbank aus Java heraus zuzugreifen. Die JDBC API ist eine Java Schnittstelle für die Ausführung von SQL-Anweisungen in relationalen Datenbanken. Lediglich ein JDBC-Treiber vom Datenbankhersteller oder eine JDBC-ODBC Bridge wird dazu zusätzlich benötigt.

Weil das JDBC API fast alle datenbankspezifischen Eigenschaften verdeckt, kann der JDBC GIS-Adapter theoretisch unabhängig von der darunterliegenden Datenbank realisiert werden.

4.3.3 Verwaltete Daten

In der Datenbank werden alle Geodaten zusammen abgelegt, d.h. jeweils Geometrie und Attribute für die relevanten Objekte der abgebildeten Wirklichkeit. Obwohl – streng genommen – keine Objekte in einer rein relationalen Datenbank abzuspeichern sind, versucht der Adapter mittels geeigneter Strukturierung eine gewisse objektorientierte Sicht auf die Tabellen in der Datenbank zu gewähren.

Geoobjekte eines Themas werden in derselben Tabelle als Zeilen gespeichert, Geoobjekte aus verschiedenen Themenbereichen werden aber wegen deren unterschiedlichen Attributierungen in separaten Tabellen vorgehalten. Um die Arbeit mit mehreren Geometrien, die jeweils zu einem Geoobjekt gehören, zu ermöglichen, werden die Geometrien wiederum getrennt von den Attributen gehalten.

Unter Geometrie werden hier sowohl Vektor- als auch Rasterdaten verstanden. Rasterdaten werden in verschiedenen Auflösungen (Rasterpyramide) für verschiedene Maßstäbe sowie gekachelt gespeichert. Dies ermöglicht einen effizienten Umgang mit den üblicherweise sehr großen Datenmengen. Das Ergebnis einer Abfrage an den Adapter enthält nur die Kacheln, die tatsächlich benötigt werden, das gesamte Rasterbild braucht nicht übertragen zu werden. Des Weiteren wird die Auflösung einer Kachel entsprechend dem angegebenen Maßstab gewählt, was zu schnellerer Beantwortung der Abfrage und anschließend zu schnellerer Darstellung führt.

Um Geoobjekte, die gegebenen räumlichen Bedingungen entsprechen, effizient auffinden zu können, sind Indexstrukturen auch für Geometriedaten von großer Bedeutung. Weil Geometriedaten – im Gegensatz zu einfachen eindimensionalen Attributdaten – von ihrer Natur aus zweidimensional sind, müssen auch räumliche Indizes diese Tatsache widerspiegeln. D.h. die eindimensionalen Datenbankindizes können nicht direkt benutzt werden. Der JDBC GIS-Adapter des *GISterm* verwendet vorerst keine explizite räumliche Indizes; es wird lediglich das minimale umschließende Rechteck (bounding box) der jeweiligen Geometrie als „Miniindex“ in die entsprechende Tabelle eingetragen. Es wird aber eine leistungsfähigere Indexstruktur entworfen, die die Beantwortung komplexerer Anfragen ermöglicht.

4.3.3.1 Metadaten

Metadaten fassen Daten über die relevanten Themen, Attribute der Themen, sowie Eigenschaften der Geometrie zusammen.

Metadaten zu Themen informieren über solche Eigenschaften wie Name des Themas, geben ausführliche Beschreibung an, organisieren die Struktur der zum Thema gehörenden Tabellen usw. Ein wichtiger Teil der Metadaten definiert die Themenhierarchie. Die einzelnen Themen werden in Themengruppen verwaltet, wodurch sich der Benutzer viel leichter einen Überblick über thematischen Zusammenhänge verschaffen kann.

GISterm kann einem Geoobjekt mehrere Geometrien zuordnen. Die Metabeschreibung zu den Geometrien ist nicht nur aus Informationen zusammengesetzt wie Name und Typ der Geometrie, sondern auch wie Sichtbarkeitsangaben für die Darstellung auf verschiedenen Maßstabsebenen.

4.3.3.2 Attributdaten

Zu Geoobjekten gehörende Sachattribute werden in ganz normalen Tabellenspalten abgelegt. Außer den eigenen Attributen können Geoobjekte – mittels Joins – mit weiteren Sachdaten aus beliebigen anderen Tabellen verknüpft werden. Die möglichen Verknüpfungen werden durch Metadaten angegeben.

4.3.3.3 Vektordaten

Vektorgeometrien werden intern in einer der OGIS Simple Features Specification kompatiblen Struktur vorgehalten. In der Datenbank kommt jedoch aus praktischen Gründen das ESRI Shapeformat zur Verwendung. Eine Geometriehaltung im OGIS-Format wäre aber denkbar und für einen allgemeinen Datenaustausch sogar geeigneter. Die in der Datenbank speicherbaren Vektorgeometrien sind:

- Punkt,
- Punktmenge,
- Linienzug,
- Polygon.

Die „measured“ und die dreidimensionale Varianten dieser Shapegeometrien werden vorerst nur teilweise vom Adapter unterstützt: sie werden nicht geschrieben und beim Lesen werden die den grundlegenden Geometrietypen hinzukommenden Informationen nicht behandelt.

Geometrien von Geoobjekten eines Themas werden in derselben Tabelle im Binärformat (als BLOB) gespeichert. Einige Elemente der Geometrie werden zusätzlich in numerischen Spalten untergebracht (z.B. die Eckkoordinaten der Bounding Box), damit die räumlichen Operatoren ihre Aufgabe effizient erledigen können, ohne die tatsächliche Geometrie unbedingt auslesen zu müssen.

4.3.3.4 Rasterdaten

GISterm verwaltet Vektor- und Rastergeometrien gleichberechtigt und integriert. Jedes Geoobjekt kann beliebig viele Vektor- und Rastergeometrien besitzen und diese sogar gemischt.

Trotzdem soll man die Unterschiede zwischen den beiden Geometriemodellen im Auge behalten: bei dem Vektormodell stellt der Punkt, bei dem Rastermodell das Pixel die Informationseinheit dar. Rasterdaten sind – im Gegensatz zu Vektordaten – flächendeckend und in ihrer Auflösung stärker eingeschränkt. Wenn man eine gute Auflösung benötigt, kann man den Umgang mit besonders großen Datenmengen nicht umgehen. Bildkompression hilft nur beim Speichern bzw. beim Übertragen; wenn Rasterdaten angezeigt werden, müssen die Daten wieder dekomprimiert werden.

Die Handhabung großer Rasterdaten kann wesentlich erleichtert werden, wenn das Bild in kleinere Teile zerlegt wird. Gewöhnlich wird nicht das gesamte Bild zum Anzeigen verwendet, sondern lediglich ein Ausschnitt wird dem Benutzer präsentiert. Fällt das Bild gänzlich in den Anzeigebereich, so benötigt man keine feine Auflösung mehr. Diese Überlegungen führen zu der Idee, Rasterdaten in Teile aufgeschnitten und mit verschiedenen Auflösungen

vorzuhalten, um die Präsentation zu beschleunigen sowie mit den zur Verfügung stehenden Rechnerressourcen sparsamer zu umgehen. Eine Zerlegung – auch Kachelung genannt, – erzeugt Teile (Kacheln), die auf einer Maßstabsebene abgelegt werden. Sie werden im nächsten Schritt dann ausgedünnt (die Zahl der Reihen bzw. Spalten wird reduziert) und wiederum dem Kachelungsprozeß unterworfen. So entsteht eine weitere Maßstabsebene, welche durch fortgesetzte Verwendung der Kachelung und Ausdünnung um weitere Maßstäbe ergänzt werden kann. Am Ende dieses Prozesses bekommt man eine Reihe von Ebenen, die Kacheln immer niedriger werdender Auflösung enthalten und zusammen eine sogenannte Rasterpyramide bilden (Abbildung 4.1). Abhängig vom aktuellen Maßstab werden Kacheln von der entsprechenden Pyramidenebene genommen und zwar nur jene Kacheln, die tatsächlich für die Präsentation oder für andere Operationen benötigt werden.

Die Rasterdatenhaltung in *GISterm* basiert auf solchen Rasterpyramiden. Informationen über die verschiedenen Ebenen einer Pyramide (Metadaten zur Rasterdatenverwaltung) werden in Datenbanktabellen gespeichert. Sie geben Auskunft über den Maßstabsbereich, die Kachelgröße, die Ausdünnungsparameter usw. Die Kacheln selbst werden als BLOBs abgelegt; die Bounding Box wird – ähnlich wie bei den Vektordaten – in normalen numerischen Spalten vorgehalten.

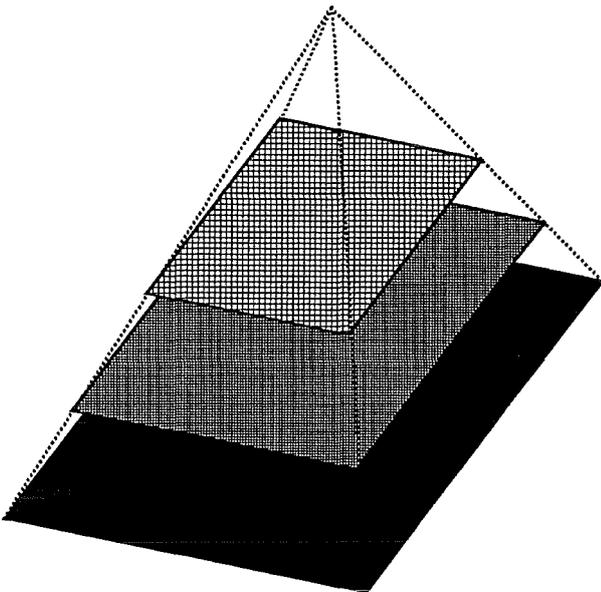


Abbildung 4.1: Rasterpyramide

Eine Frage wäre auf den ersten Blick berechtigt: Warum werden die Rasterdaten auch mit in die Datenbank eingeladen? Rasterdaten werden hauptsächlich nur gelesen, so könnten sie theoretisch auch im Filesystem gespeichert und aus der Datenbank heraus referenziert werden. Mit dieser Lösung gibt es aber mehrere Probleme:

1. Es ist schwer zu gewährleisten, daß die Rasterdaten im Filesystem nicht gelöscht oder an eine andere Stelle verschoben werden und so die Referenzen in der Datenbank immer auf die richtigen Raster zeigen.
2. Ein weiteres Problem stellt der Unterschied im Umgang mit der Datenbank und dem Filesystem dar. Die Datenbank kann mittels des JDBC-Treibers auch entfernt benutzt werden, auf das Filesystem dagegen kann nur lokal zugegriffen werden.

3. Zudem wird die einheitliche Vektor- und Rasterdatenverwaltung erschwert, wenn nicht gleich unmöglich gemacht: Rasterdaten müßten separat gehandelt werden.

4.4 Die Fachkomponente: der JDBC RIPS-Adapter

4.4.1 Architektur

Der wegen ihrer Datenhaltung ausführlich behandelte JDBC GIS-Adapter stellt eine allgemeine Funktionalität zur Verfügung, womit er eine Grundlage für komplexere Geodatenhaltung in relationalen Datenbanken bietet. Er kann also als Basiskomponente gesehen werden, welche mit verschiedenen anwendungsspezifischen Schnittstellen versehen und mit der jeweils benötigten Anwendungslogik erweitert die Basis verschiedener Fachkomponenten bilden kann.

Eine solche Fachkomponente stellt der neu entwickelte JDBC RIPS-Adapter dar. Dieser Adapter kann mit Geobjekten umgehen, die durch UIS-ID identifiziert bzw. durch UIS-OAC klassifiziert werden. Dabei baut er auf die vom JDBC GIS-Adapter realisierte Grundfunktionalität auf und bietet eine Sicht auf Objekte, welche an der des Räumlichen Informations- und Planungssystems (RIPS) ähnelt. Der JDBC GIS-Adapter kann mittels eines Treibers über JDBC auf die relationale Datenbank (in diesem Fall: ORACLE) zugreifen, Meta- und Geodaten lesen oder schreiben. Die Fachschale (JDBC RIPS-Adapter) instanziiert Themen mit den korrekten UIS-OACs aus den Metadaten und erzeugt eine Themenhierarchie. Einzelne Geobjekte mit UIS-ID werden als Java-Objekte im Adapter instanziiert, anschließend an den *GISternServer* weitergegeben.

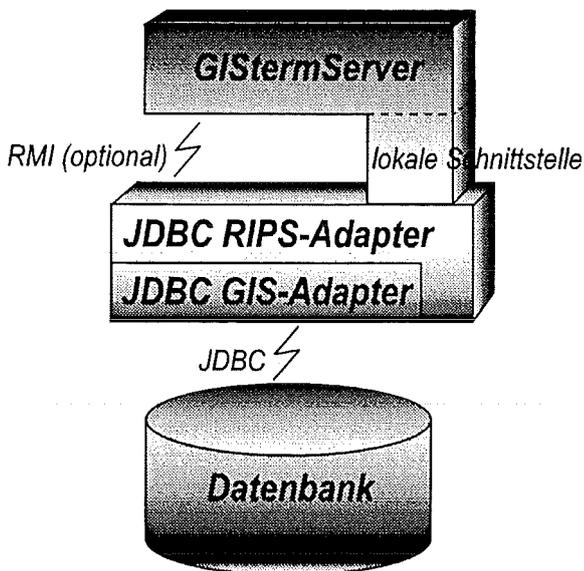


Abbildung 4.2: Architektur des JDBC RIPS-Adapters

Die Weitergabe findet entweder innerhalb der Java Virtuellen Maschine als lokale Operation statt, oder über den Remote Method Invocation (RMI) /4/ – wie es bei allen Adaptern vorgesehen ist. Die letztere Variante hat aber den wesentlichen Nachteil, daß die Objekte vor Übertragung serialisiert und danach deserialisiert werden müssen. Das ist im Standardfall eine überflüssige zusätzliche und daher performanzreduzierende Last, da die JDBC-

Verbindung allein ausreicht, Daten über das Netzwerk zu transferieren. Der aus dem Gesichtspunkt des *GIStermServers* lokal instanziierte JDBC RIPS-Adapter ist also in der Lage, Geobjekte schnell zwischen *GIStermServer* und Datenablage (Datenbank) bewegen zu können.

Abbildung 4.2 zeigt die Architektur und die Kommunikation zwischen ihren Komponenten.

4.4.2 Funktionalität

Der JDBC RIPS-Adapter arbeitet mit einem in der relationalen Datenbank definierten Schema. Er kann Themen anhand ihrer UIS-OACs identifizieren und verwalten. Ähnlich sollen Geobjekte mit UIS-IDs versehen werden, so kann der Adapter diese Objekte eindeutig von anderen unterscheiden.

Der Adapter ist in der Lage, auf diesen Objekten sowohl lesende als auch schreibende Operationen zu durchführen. Wie schon früher erwähnt, um die Arbeit von mehreren Benutzern/Adaptoren zu ermöglichen, muß der JDBC GIS-Adapter sicherstellen, daß die Daten immer konsistent bleiben.

Es werden 5 Funktionsgruppen unterschieden:

1. Schemaabfrage
2. Abfrage der Bounding Box von Geobjekten
3. Geobjektabfrage
4. Veränderung von Geobjekten
5. Datenimport

4.4.2.1 Schemaabfrage

Auf Anfrage vom *GIStermServer* wird das Datenschema aus der Datenbank gelesen. Zu jedem Thema wird – soweit vorhanden – der zugehörige RIPS-Objektartencode mit abgefragt. Aufgrund dieses Schemas kann dann der Server weitere Operationen initiieren, sowie dem Benutzer über die vorhandenen Themen und deren Eigenschaften Auskunft geben.

4.4.2.2 Abfrage der Bounding Box von Geobjekten

Die exakte Geometrie eines Objektes wird nicht immer benötigt; zu manchen Operationen genügt es, lediglich die grobe Approximation der Geometrie – die Bounding Box – zu kennen. In solchen Fällen kann man mit dieser Funktion die Bounding Boxen von Geobjekten, die durch UIS-IDs identifiziert werden, erfragen. Objekte, deren UIS-ID nicht bekannt ist (oder fehlt), können über eine interne Objekt-ID angesprochen werden.

4.4.2.3 Geobjektabfrage

Neben den vom JDBC GIS-Adapter angebotenen Möglichkeiten, sollen auch Abfragen mit Angabe von UIS-IDs oder UIS-OACs gestellt werden können. Diese Abfragekriterien können weiter mit Angabe einer Bounding Box verfeinert werden. So ergeben sich folgende Fälle:

- Abfrage von mittels Objektidentifikatoren angegebenen Objekten (Features). Dieser Anfragetyp benutzt eine interne Objekt-ID anstatt der UIS-ID. Bei Themen, in denen die Features keine UIS-ID besitzen, ist dies die einzige Möglichkeit, bereits identifizierte Features zu manipulieren.
- Abfrage von Objekten eines bestimmten Themas (FeatureType). Die Ergebnismenge kann durch Angabe einer Bounding Box eingeschränkt werden.
- Abfrage von Objekten, die mittels UIS-ID spezifiziert werden.
- Abfrage von Objekten eines bestimmten Themas (FeatureType). Hier wird das Thema durch den UIS-OAC angegeben. Eine Bounding Box kann auch hier das Interessengebiet bestimmen.

4.4.2.4 Veränderung von Geoobjekten

Zu dieser Funktionsgruppe gehören Methoden, welche die Existenz von Features bestimmen oder Änderungen an einem existierenden Feature vornehmen.

Anlegen neuer Features. Hier werden neue Geoobjekte mit Geometrie und Attributierung in die Datenbank geschrieben. Der Feature muß dabei zu einem im Datenschema definierten Thema gehören. Des weiteren muß eine UIS-ID zum neuen Objekt mit angegeben werden.

Löschen von Features. Die mittels UIS-ID angegebenen Geoobjekte werden in der Datenbank gelöscht. Objekte, die keine UIS-ID besitzen, werden durch ihre interne Objekt-IDs identifiziert.

Modifizieren der Geometrie und Attribute existierender Features. Durch diese Methode können sowohl räumliche Eigenschaften als auch andere Attribute eines Geoobjektes verändert werden.

4.4.2.5 Datenimport

Im wesentlichen handelt es sich bei dieser Funktionsgruppe um die automatisierte und konsistente Auffüllung der Datenbank mit Geodaten.

Zuerst muß selbstverständlich ein Schema definiert und in die entsprechende Metadatatabelle eingetragen werden. Dies kann entweder durch gezielte manuelle Manipulation der Tabelleneinträge mittels SQL-Anweisungen oder – besser – automatisiert erfolgen. Der automatisch ablaufende Prozeß (derzeit noch nicht implementiert) kann sich die für den Shapefile-Adapter erarbeitete Metadatenbeschreibung zunutze machen.

In der Phase des Geodatenimportes werden einerseits Daten von Shapefiles ausgelesen und in der Datenbank gespeichert. Damit die neue Geoobjekte in der Datenbank mit UIS-ID identifiziert werden können, müssen die zu den Shapes gehörenden dBase-Dateien auch UIS-ID als Attribut (Spalte) deklariert haben. Andererseits wird das Importieren von Rasterdaten gewährleistet. Die Kachelung und Ausdünnung der Daten erfolgt als Teil dieses Prozesses (Abbildung 4.3).

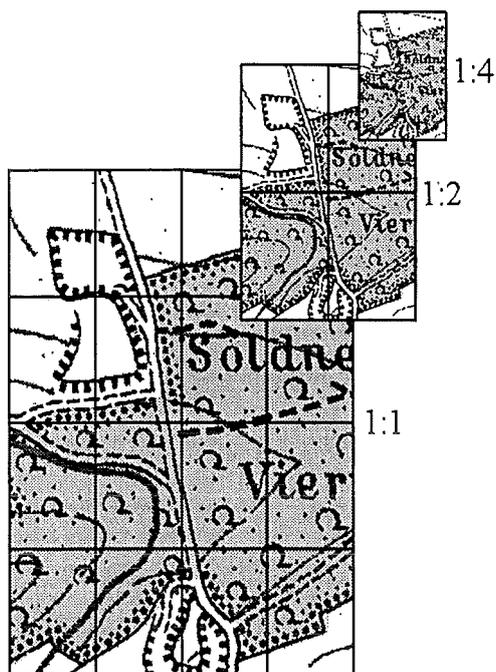


Abbildung 4.3: Kacheln in einer vom JDBC GIS-Adapter erstellten Rasterpyramide

Wie früher erwähnt, wird für die Vektorgeometriehaltung das ESRI Shapeformat eingesetzt. Als eine ernst zu nehmende Alternative dazu kann das in der Simple Features Specification von OGIS beschriebene Binärformat betrachtet werden. Wenn dieses Format vom Adapter unterstützt wird, soll dessen Import auch implementiert werden.

5. Literatur

- /1/ 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
- /2/ Mayer-Föll, R.; Jaeschke, A. (1997): Projekt GLOBUS, Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg, Phase IV 1997. Number FZKA 6000, in Wissenschaftliche Berichte, Forschungszentrum Karlsruhe Technik und Umwelt, Karlsruhe
- /3/ Claus Hofmann (1998): Architektur eines GIS-Terminal - Eine auf Web-Technologien basierende Zugriffsarchitektur auf verteilte, heterogene Geodaten. In W.-F. Riekert, K. Tochtermann (Hrsg.): Hypermedia im Umweltschutz, 1. Workshop, Ulm 1998; Marburg: Metropolis-Verl., 1998 (Umwelt-Informatik aktuell; Bd. 17), ISBN 3-89518-191-1
- /4/ Orfali, R.; Harkey, D. (1997): Client/Server Programming with Java and CORBA, Wiley
- /5/ Englander R. (1997): Developing Java Beans, O'Reilly
- /6/ Gamma et.al. (1995): Entwurfsmuster, Addison-Wesley.

Das Sachdatensystem

Das komponentenorientierte System zum komfortablen Recherchieren nach Umweltdaten im UIS Baden-Württemberg

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

1. EINLEITUNG	33
2. FUNKTIONALITÄT: ANFORDERUNGEN UND REALISIERUNG	34
2.1 ARCHITEKTUR.....	34
2.2 ANFORDERUNGEN	34
2.3 SQL-STATEMENTGENERIERUNG.....	36
2.3.1 Erzeugung der Endabfrage des Selektors.....	36
2.3.4 Erzeugung der SQL-Statements für tabellarische Kriterien.....	37
3. BENUTZUNG DES SACHDATENSYSTEMS.....	38
3.1 SELEKTOR.....	38
3.2 EINGABE	39
3.3 SELEKTORKRITERIEN	40
3.3.1 Selektorkriterium: Table	41
3.3.2 Selektorkriterium: Area	42
3.3.3 Selektorkriterium: Value.....	43
3.3.4 Selektorkriterium: Time	44
3.3.5 Selektorkriterium: Duration	45
3.3.6 Selektorkriterium: BoundingBox.....	46
3.4 OPTIONEN	47
3.5 DATENFLUßFENSTER	48
3.6 DARSTELLUNGS AUSWAHL	49
4. WIEDERVERWENDBARE JAVABEANS UND DAS SELEKTIONSFRAMEWORK	51
4.1 EINGABE- UND VISUALISIERUNGS-BEANS.....	51
4.1.1 <i>FilterableTable</i>	51
4.1.2 <i>HierarchieBrowser</i>	53
4.1.3 <i>DateInput</i>	54
4.2 SELEKTIONSFRAMEWORK.....	56
4.2.1 <i>Funktionalität des Frameworks</i>	56
4.2.2 <i>Initialisierung des Frameworks</i>	57
4.2.3 <i>Schnittstellen</i>	57
4.2.4 <i>Erweiterbarkeit und Abdeckungsbereich</i>	58
5. ADMINISTRATION: ERSTELLUNG NEUER SELEKTOREN	59
6. ZUSAMMENFASSUNG UND AUSBLICK.....	60
7. LITERATUR.....	61

1. Einleitung

Im Umweltinformationssystem (UIS) Baden-Württemberg wurde bereits früh auf den Einsatz von Web-Technologien auch zur Erschließung der Inhalte von Datenbanken für das Internet sowie im verwaltungsinternen Intranet gesetzt. Zunächst wurden ausschließlich "klassische" Techniken wie das Common Gateway Interface (CGI) eingesetzt. Insbesondere für Anwendungen im verwaltungsinternen Intranet kristallisierte sich jedoch schnell die Anforderung einer komfortableren Benutzungsoberfläche mit größerem Funktionsumfang heraus. In GLOBUS IV wurden diese Anforderungen im WWW-Szenario aufgegriffen. Es wurde ein kooperierendes System, bestehend aus dem Metadaten- oder Katalogsystem WWW-UDK, dem Sachdatensystem SDS sowie dem GIS/Graphiksystem GISterm, geschaffen. Gegenstand des vorliegenden Beitrags ist die Weiterentwicklung einer dieser Komponenten, nämlich des Sachdatensystems (SDS) in GLOBUS V, das für Endanwender den Zugriff auf Daten, die in einem relationalen Datenbanksystem gehalten werden, ermöglicht.

Das Sachdatensystem des UIS Baden-Württemberg ist eine Client/Server-Anwendung, mit der Datenbanken auf komfortable Weise mit vorstrukturierten Anfragemöglichkeiten, sogenannte Selektoren, abgefragt werden können. Dazu wird die sich hinter einem Selektor verborgene Struktur der Datenbankanfrage in einem serverseitigen Repository abgelegt. Zur Laufzeit des Programms wird dann diese Struktur ausgewertet und ein dazu passendes Eingabeformular auf Clientseite generiert. In diesen Formularen wählt der Anwender bestimmte Eigenschaften der Datensätze aus. Daraus wird dann dynamisch eine Datenbankanfrage generiert, mit der die vom Benutzer selektierten Datensätze aus der Datenbank vom Server in die Anwendung geladen werden. Hier können die Datensätze in Tabellen aufgelistet, visualisiert und für weitere Anfragen benutzt werden.

Die Funktionalität des Sachdatensystems ist so allgemein, daß sich seine Anwendung für eine Reihe weiterer Aufgabenstellungen im UIS Baden-Württemberg empfiehlt. Dies sind zum Beispiel fachspezifische Berichtssysteme. Deshalb wurde das SDS als erweiterbares Framework entwickelt.

Um diese Wiederverwendbarkeit darüber hinaus zu fördern, wurden Teilfunktionen der Anwendung in Komponenten gekapselt. In Java ist JavaBeans das Synonym für eine Komponentenarchitektur, die diese Wiederverwendbarkeit über die Grenzen unterschiedlicher Computerarchitekturen hinweg gewährleistet. So können z.B. Komponenten des SDS auch in Erfassungssystemen wiederverwendet werden.

Der vorliegende Text ist wie folgt aufgebaut: Kapitel 2 gibt einen Überblick über die Funktionalität des Sachdatensystems. Danach wird in Kapitel 3 das Sachdatensystem aus der Sicht eines Benutzers präsentiert. In Kapitel 4 werden kleinere JavaBeans beschrieben, die im SDS selbst als Elemente in den Eingabeformularen verwendet werden, aber auch außerhalb dieser Anwendung sehr hilfreich sein können. Desweiteren wird in Kapitel 4 das Selektionsframework vorgestellt, worauf andere Systementwicklungen zurückgreifen können. Kapitel 5 beschreibt, wie neue Selektoren von Administratoren erstellt und in das SDS aufgenommen werden können. Der Text schließt in Kapitel 6 mit einer Zusammenfassung.

2. Funktionalität: Anforderungen und Realisierung

2.1 Architektur

Die Architektur des Sachdatensystems präsentiert sich wie in Abbildung 2.1 dargestellt. Auf Serverseite befindet sich der zentrale, in Java geschriebene Applikationsserver des SDS, der als Schnittstelle zu den verschiedenen Umweltdatenbanken und dem Selektor-Repository fungiert. Jede SDS-Clientsoftware entspricht einem Java-Applet, das sich direkt nach seinem Start mit dem SDS-Applikationsserver verbindet.

Der Applikationsserver hat unter anderem die Aufgabe, Datenbankverbindungen zu allen Datenbanken, die im Repository aufgelistet sind, zu verwalten und auch selbständig wieder eine Verbindung zu einer Datenbank herzustellen, sollte die Verbindung unterbrochen worden sein. Durch diese Technik wird der zeitaufwendige Verbindungsaufbau bei der Durchführung eines Selektors vermieden und eine bessere Performanz erzielt. Alle Benutzer greifen über dieselben - bereits vor ihrem Aufruf des Applets bestehenden - Datenbankverbindungen auf die Datenbank zu.

Das Repository hält alle Beschreibungen der Selektoren in ASCII-Dateien. Der Applikationsserver parst diese Beschreibungen und stellt sie der Clientsoftware in einem lesbaren Format zur Verfügung. Die Clientsoftware kann aus diesen Beschreibungen die GUI und alle benötigten SQL-Statements (natürlich unter der Berücksichtigung der Benutzereingaben) generieren.

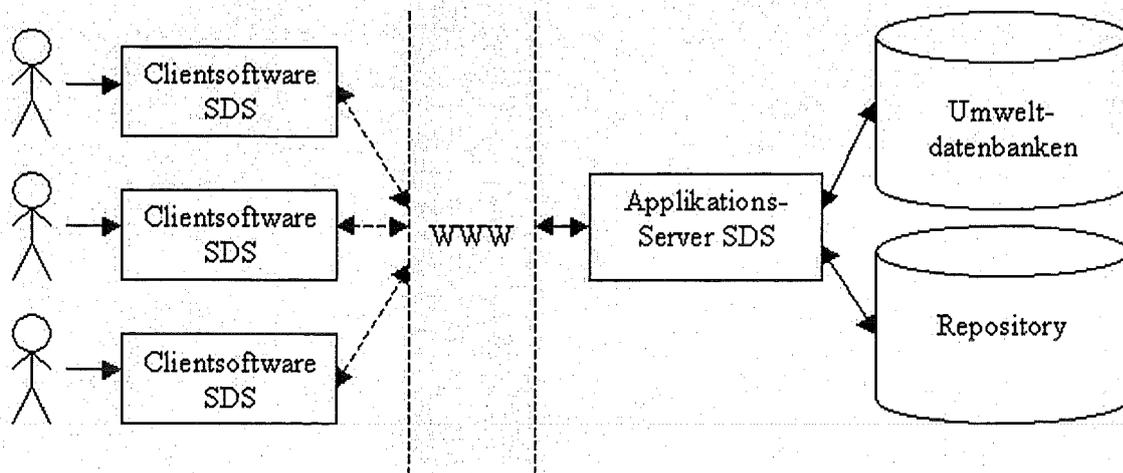


Abbildung 2.1: Architektur des SDS

2.2 Anforderungen

Im Rahmen von GLOBUS V wurde das in GLOBUS IV begonnene Sachdatensystem, das dort Teil des WWW-Szenarios war, weiterentwickelt (siehe auch /1/). Die Basis des Sachdatensystems stellen die sogenannten Selektoren (vorstrukturierte Anfragemöglichkeiten) dar, die zur Laufzeit aus dem serverseitigen Repository generiert werden. Neben Schnitt-

stellen zum GIStern zur Visualisierung von Selektionsergebnissen gibt es auch eine Schnittstelle zum WWW-UDK.

Im Rahmen von GLOBUS V sollten eine Reihe von Verbesserungen und Erweiterungen am SDS vorgenommen werden:

- Überarbeitung der GUI: Die Oberfläche des SDS sollte mit den neu herausgekommenen SWING-Klassen von Java entwickelt werden, mit denen sich im Gegensatz zur AWT des JDK viel ansprechendere GUIs entwickeln lassen. Das Ergebnis dieser Überarbeitung ist auf große Zustimmung gestoßen und wird anhand von Screendumps in Abschnitt 3 vorgestellt.
- Performanz der Datenbankabfragen: Dem SDS liegen große Datenmengen zugrunde. Daher sind Optimierungen der SQL-Anweisungen wesentliche Grundlage für Performanzverbesserungen. Es wurde eine Analyse der SQL-Anweisungen durchgeführt. Das Ergebnis führte zu einem neuen Algorithmus zur Erzeugung der SQL-Anweisungen aus den Repositorydaten und den Benutzereingaben (siehe hierzu auch Abschnitt 2.3) sowie zu einer Veränderung der Angaben von existierenden Selektoren. Insondere die Ermittlung der möglichen Werte bei tabellarischen Kriterien im kontextsensitiven Modus, bei dem alle bisher gemachten Eingaben des Benutzers berücksichtigt werden, hat zu hohen Performanzverbesserungen geführt.
- Anpassung der Darstellungsauswahl an die neu vom GIStern gebotenen Darstellungsarten (siehe hierzu auch den GLOBUS V Berichtsteil des IPF über das GIStern), die in GLOBUS V entwickelt wurden.
- Vereinfachung der Erstellung und Verwaltung der Selektoren im Repository: Die Selektoren im Repository werden in einer eigenen Syntax definiert. Der Administrator, der die Dateien in diese Syntax erstellt, braucht keine Kenntnisse über den Code des SDS oder irgendwelche Programmiersprachen zu besitzen. Die im Repository verwendete Syntax zur Definition von neuen Selektoren wurde neu entwickelt. Das Erstellen der Selektoren des Repositories basierend auf einer LISP-ähnlichen Syntax und ist nun einfacher. Der Nachweis der Praktikabilität dieser Syntax wurde in den letzten Monaten von LfU-Mitarbeitern gebracht, die selbständig und problemlos neue Selektoren nur unter Zuhilfenahme des Handbuchs erstellten. (siehe auch Abschnitt 5).
- Globale Parameterisierung der Selektion durch Einschränkung der Anfragezeit und der Ergebnismenge.
- Selektorchaining: Unter Selektorchaining wird das Hintereinanderausführen von Selektoren verstanden, wobei das Ergebnis des einen Selektors als Vorbelegung eines Kriteriums des darauffolgenden Selektors übertragen werden soll. Dazu mußte unter anderem das Repository so abgeändert werden, daß dynamisch zur Laufzeit entschieden kann, welche Selektoren auf dem gerade durchgeführten Selektor aufbauen und deshalb dem Benutzer angeboten werden können (siehe auch Abschnitt 5).

Neben diesen funktionalen Verbesserungen und Erweiterungen sollten wiederverwendbare Komponenten aus dem SDS identifiziert und anderen Softwareentwicklern verfügbar gemacht werden. Das Ergebnis sind unter anderem die drei GUI-Beans, die in Abschnitt 4.1 vorgestellt werden. Desweiteren sollte ein Framework bereitgestellt werden, auf dessen Basis neue Recherche-Tools aufbauen können und das die Basisfunktionalität eines solchen Systems liefert. Dieses wird in Abschnitt 4.2 erläutert.

2.3 SQL-Statementgenerierung

Die Erzeugung der SQL-Statements ist eine der zentralen Aufgaben des SDS-Applets. Wird ein Selektor vom Benutzer ausgefüllt, so wird der SQL-Abfragebaum aus der Repositorybeschreibung und den Eingaben generiert. Abbildung 2.2 zeigt einen solchen Abfragebaum. Innerhalb des Abfragebaums werden dabei nur die ausgewählten Kriterien (ConditionTable.Key), die vom Benutzer gemachten Kriterieneingaben sowie die zu verknüpfenden Schlüsselwerte (SelectorTable.ForeignKey) des Selektors gespeichert. Alle ausgewählten Kriterien werden mit UND (AND-Knoten) verknüpft. Indirekte Fremdschlüsselbeziehungen zwischen der ConditionTable und der Selektortable laufen über eine oder mehrere Zwischentabellen (ZT).

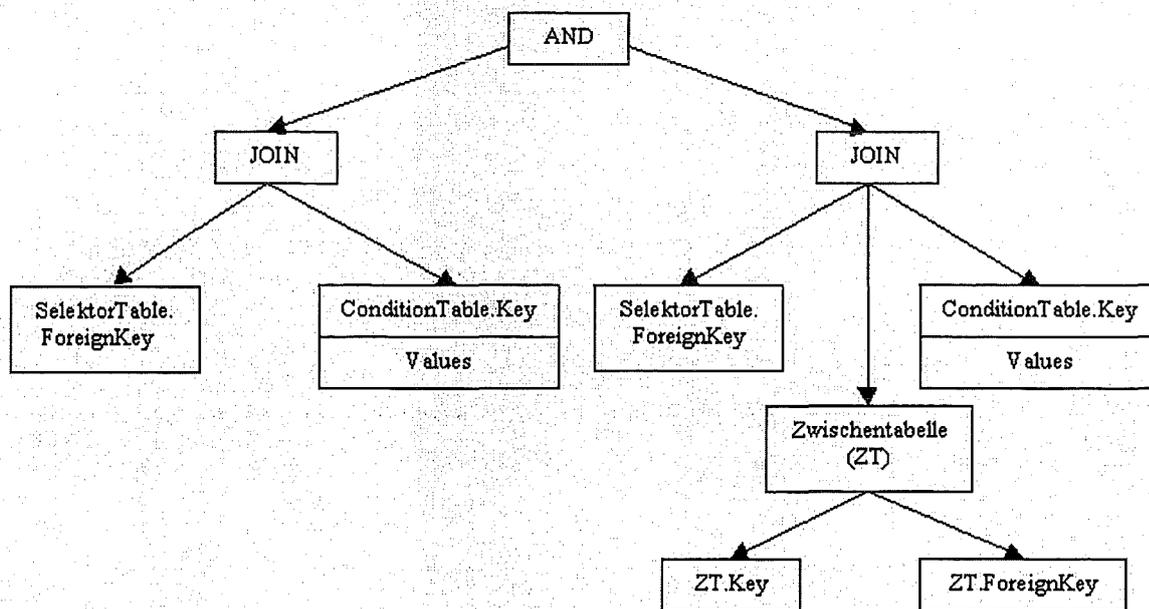


Abbildung 2.2: Abfragebaum

2.3.1 Erzeugung der Endabfrage des Selektors

Um den Vorgang der Statementerzeugung zu verdeutlichen, werden die einzelnen Schritte aufgeführt.

1. Schritt: Die globalen Abfrageelemente des Selektors werden zusammengestellt. Im einzelnen bedeutet dies, daß die SELECT-Klausel mit allen zu selektieren Feldern erzeugt wird. Danach wird die FROM-Klausel mit den verwendeten Tabellen erstellt.

2. Schritt: In diesem Schritt werden alle ausgefüllten Kriterien bearbeitet. Dieser Schritt liefert die einzelnen Wertebedingungen der WHERE-Klausel. Dabei gibt es entsprechend der einzelnen Kriterien eine unterschiedliche Erzeugung:

- Wert:
Wurde nur ein Wert ausgewählt, dann wird eine Gleichheitsoperation verwendet.
SelectorTable.ForeignKey = Value

(Bei manchen Kriterien können auch andere Operatoren (<, >, =) gewählt werden)

- Menge:
Wurden mehrere Werte ausgewählt, dann wird eine Mengenoperation verwendet.
SelectorTable.ForeignKey IN (Value1, Value2,...)
- Bereich:
Ein spezieller Bereich wird wie folgt selektiert.
SelectorTable.ForeignKey BETWEEN Value1 AND Value2

Gibt es keine direkte Verbindung zwischen *SelectorTable* und *ConditionTable*, sondern nur eine indirekte über eine Zwischentabelle, dann steht in den obigen drei Fällen jeweils *ZT.ForeignKey* anstelle von *SelectorTable.ForeignKey* und es kommt folgender Ausdruck in der WHERE-Klausel hinzu: *SelectorTable.ForeignKey = ZT.Key*. Desweiteren muß in der FROM-Klausel natürlich die Zwischentabelle ergänzt werden.

2.3.4 Erzeugung der SQL-Statements für tabellarische Kriterien

Bei der Ermittlung der Wertelisten für tabellarische Kriterien muß man zwischen kontextsensitivem und kontextinsensitivem Ablauf unterscheiden:

Kontextsensitive Durchführung

Bei kontextsensitiver Erzeugung des SQL-Statements für die Selektorkriterien wird die *SelectorTable* verwendet. Dabei wird mit allen zuvor ausgewählten Conditions ein Join auf die *SelectorTable* durchgeführt. Dies führt zu folgendem Statement:

```
SELECT      ConditionTable.Key, ConditionTable.Text
FROM        Condition, SelectorTable
WHERE ConditionTable.Key = SelectorTable.Key
AND        SelectorTable.ForeignKey1 = ConditionTable1.Key
AND        SelectorTable.ForeignKey2 = ConditionTable2.Key
```

Innerhalb dieses Statements wurden *Condition1* und *Condition2* bereits vom Benutzer eingeschränkt. Gibt es für *Condition1* zum Beispiel keine direkte Verbindung zwischen *SelectorTable* und *ConditionTable*, dann setzt sich das SQL-Statement wie folgt zusammen:

```
SELECT      ConditionTable.Key, ConditionTable.Text
FROM        Condition, SelectorTable, ZT
WHERE ConditionTable.Key = SelectorTable.Key
AND        SelectorTable.ForeignKey1 = ZT.Key
AND        ZT.ForeignKey = ConditionTable1.Key
AND        SelectorTable.ForeignKey2 = ConditionTable2.Key
```

Kontextinsensitive Durchführung:

Bei kontextinsensitiver Erzeugung des SQL-Statements für die Selektorkriterien wird die *SelectorTable* nicht verwendet. Dies führt zu folgendem Statement:

```
SELECT    ConditionTable.Key, ConditionTable.Text
FROM      ConditionTable
```

3. Benutzung des Sachdatensystems

Dieses Kapitel befaßt sich mit der Benutzerschnittstelle. Dazu wird anhand von Screenshots die Benutzung des SDS erklärt.

3.1 Selektor

Nachdem ein Selektor durch den Benutzer ausgewählt wurde, wird dieser aktiviert und im zentralen Fenster angezeigt.

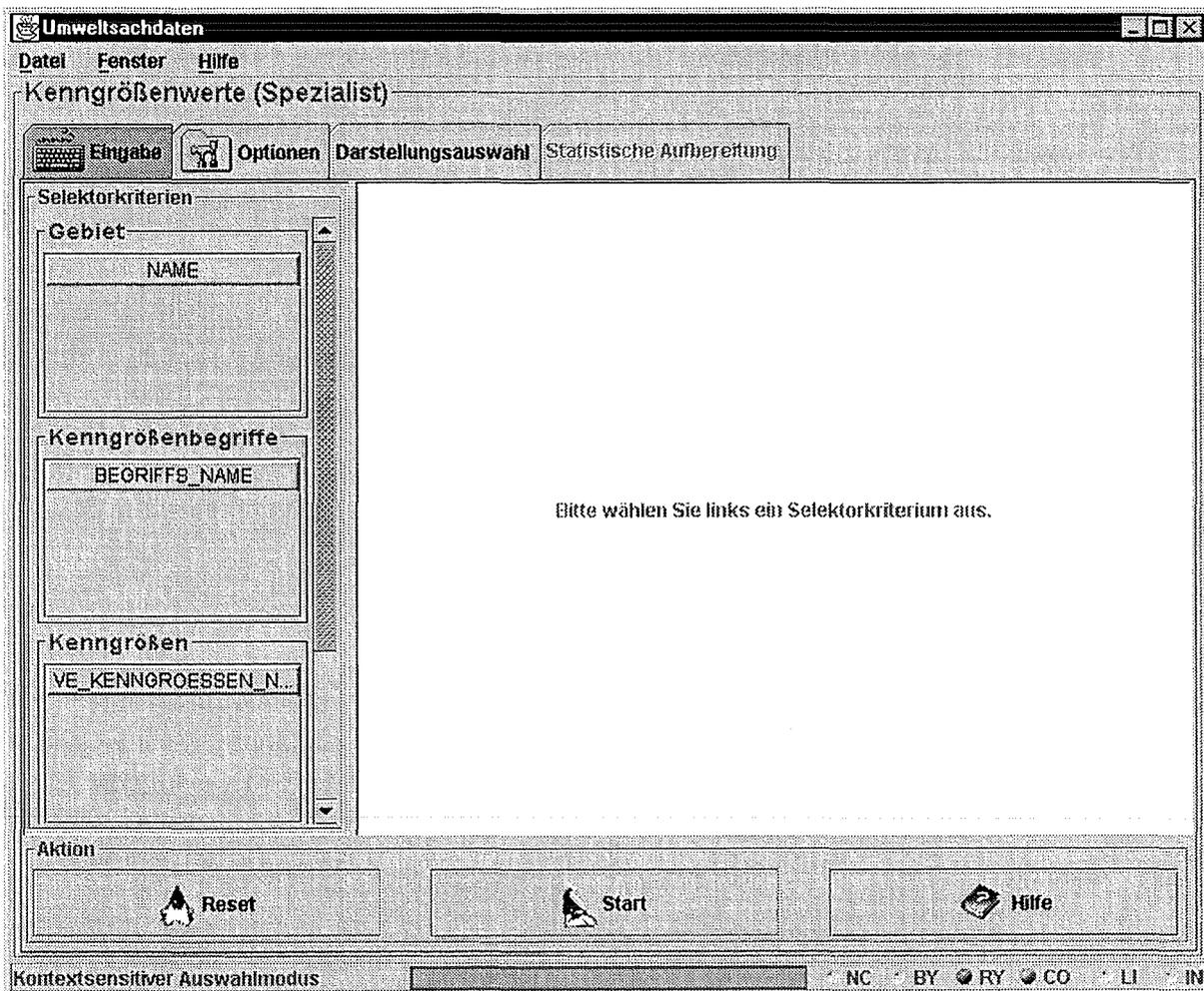


Abbildung 3.1: Selektoreingabe (hier Kenngrößenselektor)

Die Benutzerschnittstelle besteht dabei wie in Abbildung 3.1 dargestellt, aus folgenden vier Karteikarten:

- Eingabe: Parametrisierung des Selektors durch seine aggregierten Kriterien.
- Optionen: Einstellung von Selektionsparametern.

- Darstellungsauswahl: Darstellung von Selektorergebnissen
- Statistik: Noch nicht implementiert.

Die Karteikarten sind einzeln anwählbar und führen zur entsprechenden Programmfunktion.

3.2 Eingabe

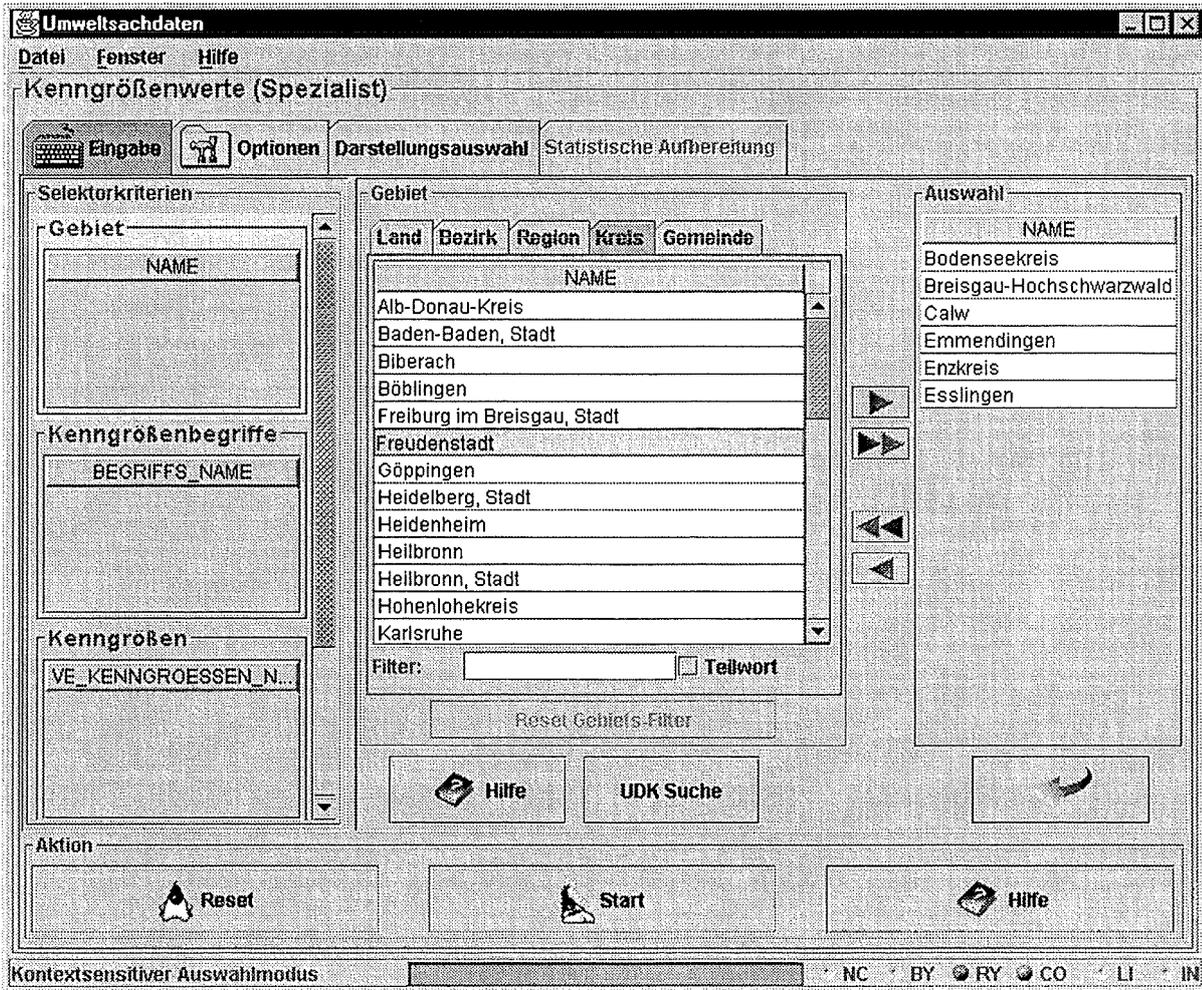


Abbildung 3.2: Eingabe

Selektorkriterien

Die Eingabe des Selektors präsentiert sich wie in Abbildung 3.2 dargestellt. In der linken Hälfte sind die zu diesem Selektor gehörenden Selektorkriterien aufgelistet. Jedes Selektorkriterium ist mit seinem Namen bezeichnet, und es wird seine aktuelle (und zu Anfang leere) Werteauswahl in einer Tabelle angezeigt. Die Selektorkriterien sind je nach Zustand unterschiedlich farblich hinterlegt.

Eingabemaske der Selektorkriterien

Ein Selektorkriterium wird durch einen Klick auf die Tabelle zur Bearbeitung ausgewählt. Das Selektorkriterium wird daraufhin links mit einem leuchtendem Gelb markiert und im rechten

Teil des Eingabe-Panels dargestellt. Jedes Selektorkriterium hat einen bestimmten Typ, der die Eingabemaske und damit die Möglichkeiten der Parameterisierung festlegt. Allen Eingabemasken gemein ist eine rechts angeordnete und gelb umrandete Zwischentabelle, die der Zwischenspeicherung der Parameterisierung eines Kriteriums dient.

Selektorkriterium Aktionsleiste

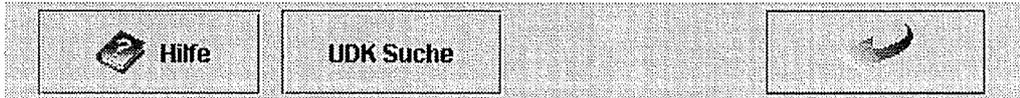


Abbildung 3.3: Aktionsleiste der Kriterien

Unabhängig von der Art des Kriteriums ist eine einheitliche Aktionsleiste (Abbildung 3.3), die einen Button zum Aufruf kriteriumsspezifischer Hilfe, einen Button zum Starten der Suche im Umweltdatenkatalog (UDK) und einen Button zur Übernahme der Auswahl aus der Zwischentabelle in die Werteauswahl besitzt.

Beim Betätigen der UDK-Suche wird im Metainformationssystem **Umweltdatenkatalog (UDK)** die allgemeine Suche angestoßen, um ausführliche Informationen zu einem Begriff zu erhalten, der in der aktiven Auswahlliste selektiert ist. Wurde kein spezieller Begriff ausgewählt, wird nach dem Namen des aktuell ausgewählten Kriteriums im UDK gesucht.

Selektor-Aktionsleiste



Abbildung 3.4: Aktionsleiste des Selektors

Der Selektor selbst besitzt auch eine Aktionsleiste (Abbildung 3.4). Darin enthalten ist ein Reset-Button, der die Werteauswahl aller Selektorkriterien auf deren (leeren) Anfangswert zurücksetzt. Mit dem Start-Button wird die Selektoranfrage mit der aktuellen Wertebelegung gestartet. Mit dem Hilfe-Button wird die selektorspezifische Hilfe aufgerufen.

3.3 Selektorkriterien

Jedem Selektorkriterium wird im Repository ein Typ zugewiesen, der die Darstellung für die Eingabe und die Möglichkeiten der Parameterisierung festlegt. Nach der Auswahl eines Selektorkriteriums im linken Teil des Selektor-Panels wird rechts eine zum Typ des Selektorkriteriums passende Eingabemaske präsentiert. Im einzelnen stehen folgenden Selektorkriterien zur Auswahl.

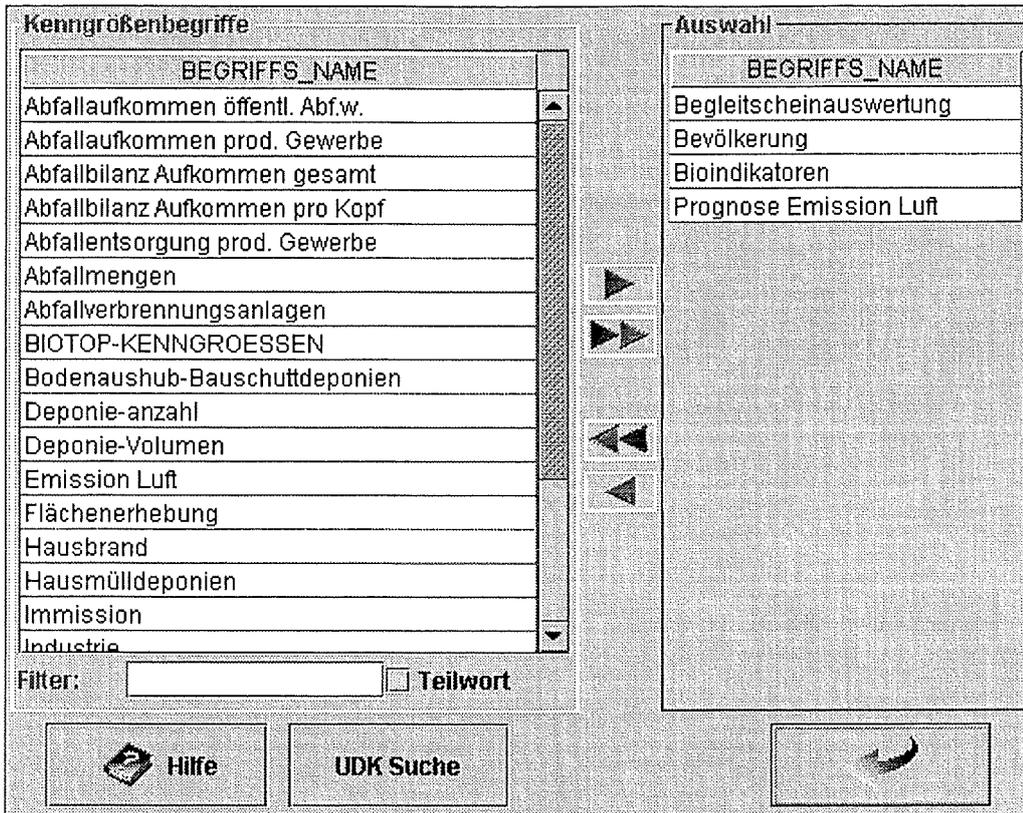


Abbildung 3.5: Kriterium Table

3.3.1 Selektorkriterium: Table

Ein Selektorkriterium vom Typ Table (Abbildung 3.5) lädt eine Relation aus einer relationalen Datenbank und stellt eine Auswahl der Spalten dieser Relation in Form einer Daten-Tabelle dar.

Die Anzahl der dargestellten Zeilen in der Daten-Tabelle kann durch den Einsatz eines Filters reduziert werden. Der Filter wirkt auf die erste Spalte der Tabelle. Das Vergleichsmuster für den Filter wird im Textfeld eingegeben und ignoriert die Groß- oder Kleinschreibung. Der Filter aktualisiert nach jedem eingegebenen Zeichen die Datentabelle und filtert dort je nach Modus gleiche Wortanfänge oder Teilworte heraus.

Rechts neben der Datentabelle befindet sich die Auswahltabelle, in der die ausgewählten Datensätze angezeigt werden. Mit Hilfe einer Selektionsleiste können markierte Einträge zwischen der Daten- und der Auswahltabelle verschoben werden. Um eine Auswahl zu bestätigen, werden die Einträge mit dem Werteauswahl Button aus der Aktionsleiste des Selektors in die Darstellung der Werteauswahl übernommen. Mehrere ausgewählte Werte werden in der Datenbankabfrage ODER-verknüpft.

3.3.2 Selektorkriterium: Area

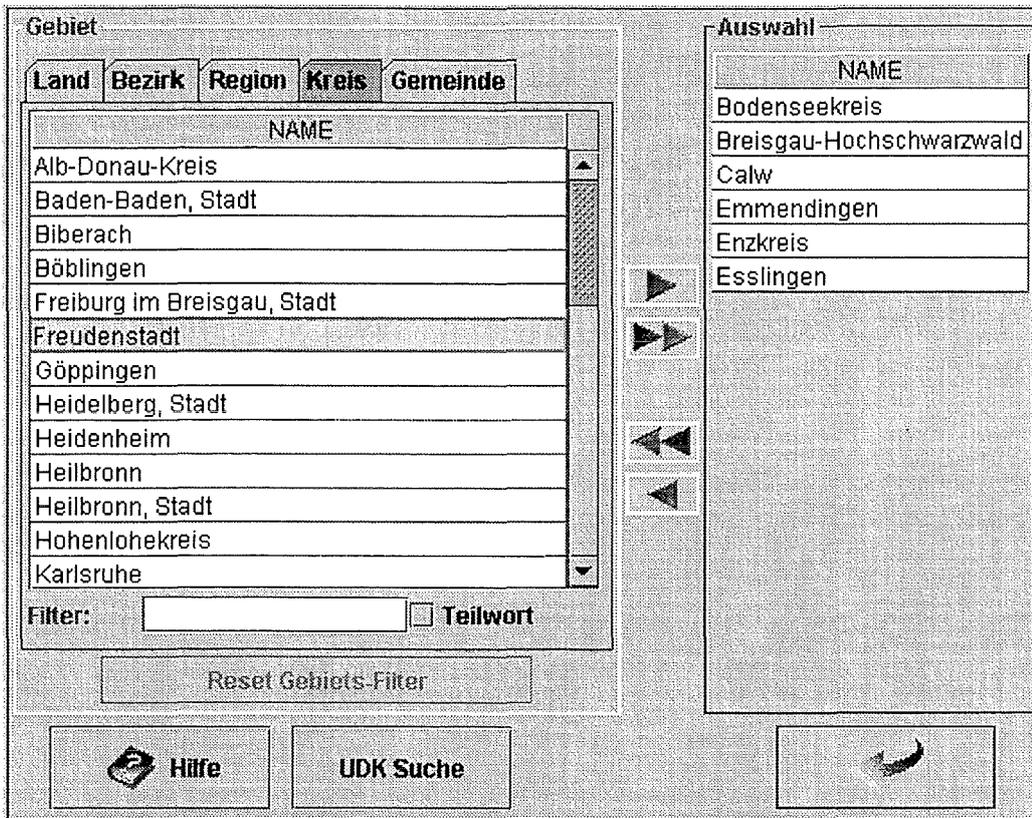


Abbildung 3.6: Kriterium Area

Ein Selektorkriterium vom Typ Area (Abbildung 3.6) ist im Grunde ein spezialisierter Table-Typ. Wie dieser stellt auch der Area-Typ eine Relation aus einer Datenbank dar, und bietet die Auswahl einer Untermenge an. Das besondere an diesem Typ ist, daß immer die gleiche Relation dargestellt wird, und daß auf die Besonderheiten dieser Relation im besonderen Maße eingegangen wird. Die Relation, die von diesem Selektorkriterium ausgelesen wird, enthält Namen von Ländern, Regionen, Regierungsbezirken, Kreisen und Gemeinden und ordnet diesen Identifikatoren zu, anhand derer eine Einordnung in die Hierarchie vorgenommen wird. Die einzelnen Gebiete sind dadurch eindeutig einer Datentabelle zugeordnet, die in einer der fünf Karteikarten untergebracht ist.

Wie beim Selektorkriterium Table kann auch hier die Anzahl der Zeilen durch die Eingabe eines Filters eingeschränkt werden. Zusätzlich gibt es hier einen weiteren Filter, der die hierarchische Anordnung der Gebiete ausnutzt und es dadurch ermöglicht, sich beispielsweise alle Gemeinden oder Kreise in einer bestimmten Region auflisten zu lassen. Dazu ist lediglich eine Region auszuwählen, und anschließend in die gewünschte Hierarchie zu gehen. Hier werden dann nur die passenden unter- oder übergeordneten Gebiete angezeigt. Um wieder alle Gebiete zu sehen, ist der Knopf „Reset Gebiets-Filter“ zu drücken.

Rechts neben der aus fünf Teillisten bestehenden Gebietsdatentabelle befindet sich wieder die Auswahltabelle, in der die ausgewählten Datensätze angezeigt werden, und in die mit Hilfe einer Selektionsleiste markierte Einträge verschoben werden können. Beim Auswählen

von Gebietsdaten, die sich in der Hierarchie „Kreise“ befinden, wird automatisch im GIStern Applet der entsprechende Kreis farblich markiert. Beim Zurücksetzen werden die Einträge wieder in die richtige Hierarchieebene eingeordnet und die farbliche Markierung im GIStern Applet zurückgesetzt.

3.3.3 Selektorkriterium: Value

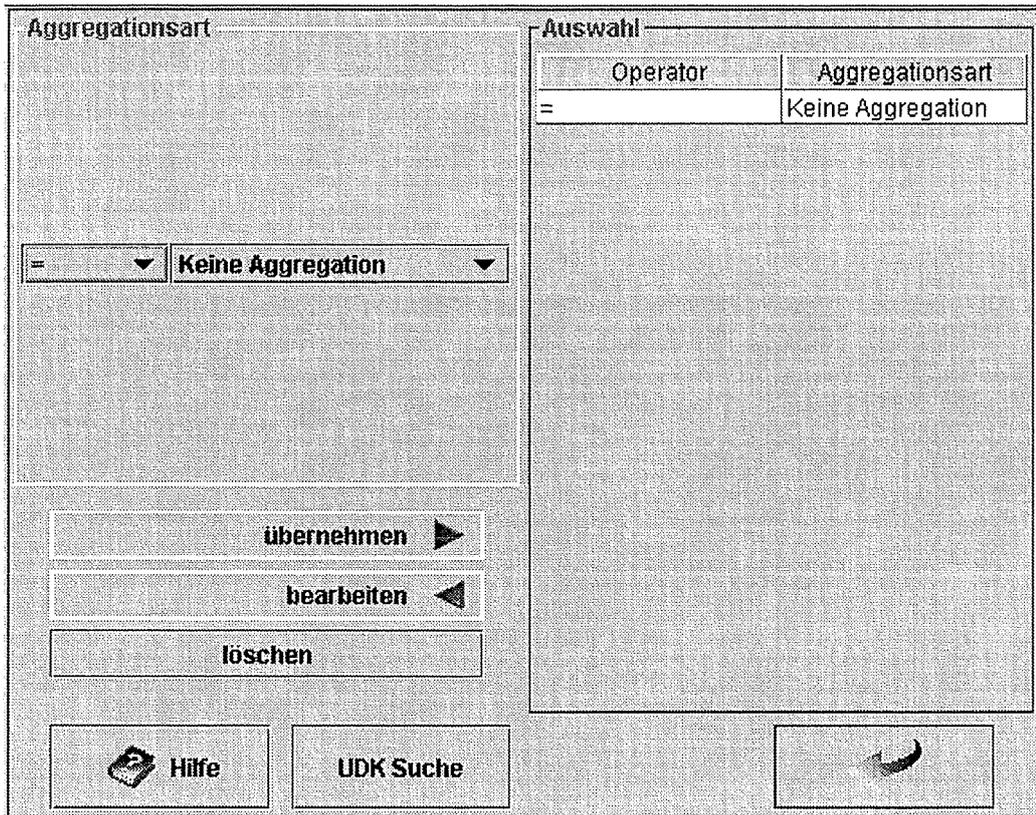


Abbildung 3.7: Kriterium Value

Beim Selektorkriterium Value (Abbildung 3.7) trägt der Benutzer die Werte selbst ein. Dazu steht ein Textfeld zur Verfügung, in dem die Texte oder Zahlenwerte einzutragen sind. Jedem Wert kann mit einer Auswahlbox ein Vergleichsoperator zugeordnet werden, der bei der Selektoranfrage benutzt werden soll. Die einzelnen Operatoren sind:

- = : gleich
- < : kleiner
- <= : kleiner oder gleich
- > : größer
- >= : größer oder gleich
- <> : ungleich
- **like**: Stringvergleich mit Muster
- **between**: Wert in einem Intervall (Syntax: BETWEEN x AND y)

3.3.4 Selektorkriterium: Time

Beim Selektorkriterium Time (Abbildung 3.8) kann der Benutzer eine genaue Zeitangabe machen, um diese dann in der Abfrage zu verwenden.

The screenshot shows a software interface with two main panels. The left panel, titled 'AggregationsZR', contains input fields for 'Tage' (Days), 'Monate' (Months), and 'Jahre' (Years), each with a small box containing the number '0'. Below these are fields for 'Stunde' (Hours) and 'Minuten' (Minutes), also with '0' in their respective boxes. The right panel, titled 'Auswahl', has a header row with five columns: 'JAHR', 'MO...', 'TAG', 'STU...', and 'MIN...'. Below this header is a large empty rectangular area. Between the two panels are two vertical buttons with right-pointing and left-pointing arrows. At the bottom of the interface are three buttons: 'Hilfe' (Help) with a book icon, 'UDK Suche' (UDK Search), and a button with a curved arrow icon.

Abbildung 3.8: Kriterium Time

3.3.5 Selektorkriterium: Duration

The screenshot shows a user interface for selecting a duration. On the left, there are two dropdown menus. The top one is labeled 'Von' and has '1974' selected. The bottom one is labeled 'Bis' and has '1982' selected. In the center, there are two arrow buttons pointing right and left. On the right, there is a table titled 'Auswahl' with two columns: 'Von' and 'Bis'. The 'Von' column contains '1974' and the 'Bis' column contains '1982'. At the bottom, there are three buttons: 'Hilfe' with a book icon, 'UDK Suche', and a search icon.

Abbildung 3.9: Kriterium Duration

Beim Selektorkriterium Duration (Abbildung 3.9) wählt der Benutzer die Anfangs- und Endzeit für den gewünschten Zeitraum aus. Dabei können die zur Auswahl stehenden und erfaßbaren Zeitwerte sich je nach Selektor unterscheiden. Zum Beispiel kann in einem Selektor nur nach einem jährlichen Zeitraum gefiltert werden, während es in einem anderen Selektor wiederum möglich ist, Zeitangaben bis auf den Tag oder sogar die Minute genau einzugeben.

3.3.6 Selektorkriterium: BoundingBox

Beim Selektorkriterium Bounding Box (Abbildung 3.10) trägt der Benutzer die Koordinatengrenzen für die gewünschte BoundingBox in die vier entsprechenden Zahlenfelder. In nächsten Versionen des SDS wird diese Eingabe auch über das GIS-Tool möglich sein.

Auswahl			
XMin	XMax	YMin	YMax

Abbildung 3.10: Kriterium Bounding Box

3.4 Optionen

In der Karteikarte Optionen (Abbildung 3.11) können allgemeine Einstellungen für die Selektoren vorgenommen werden.

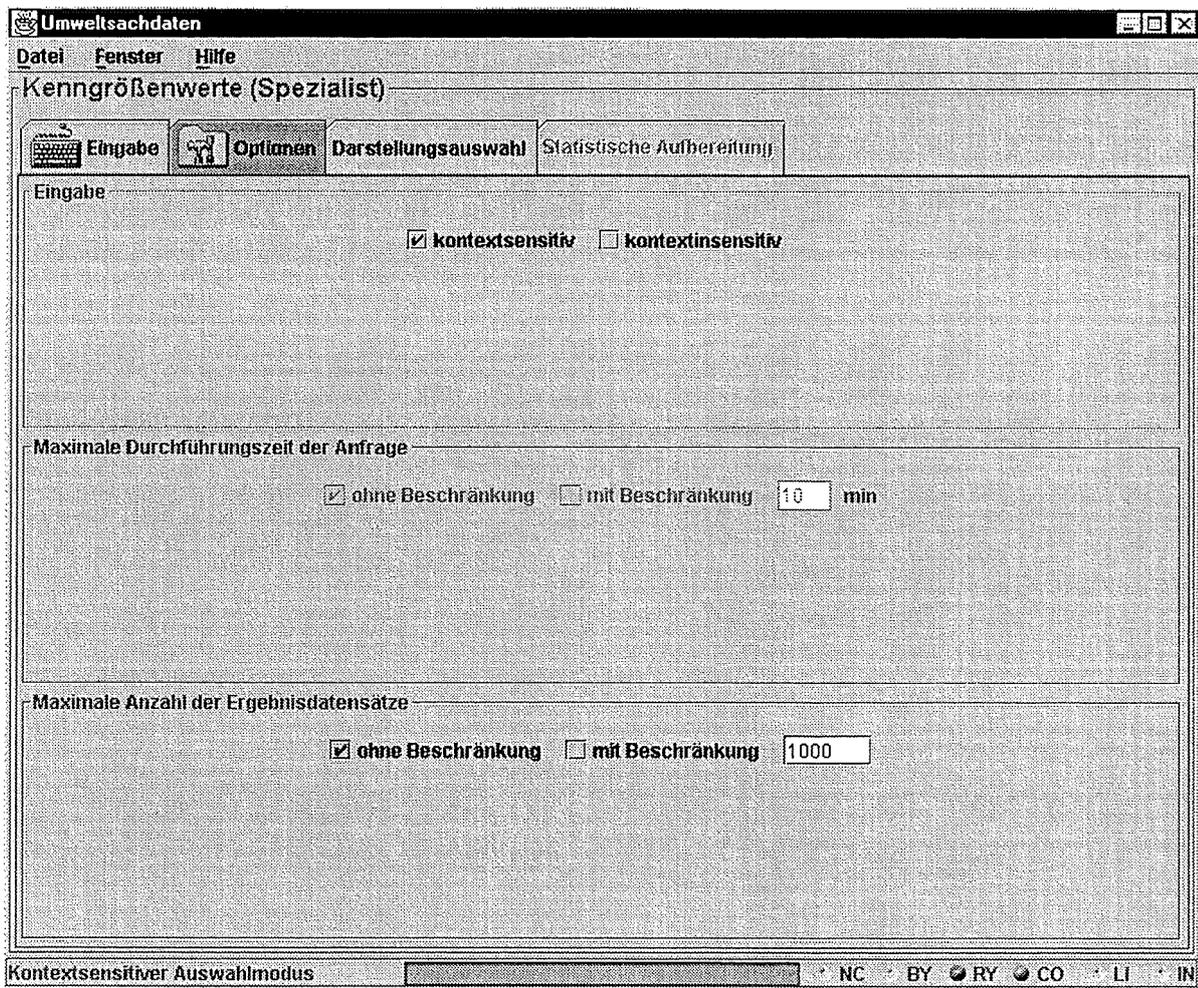


Abbildung 3.11: Optionen

Im Feld **Eingabe** kann zwischen kontextsensitivem Modus und kontextinsensitivem Modus gewählt werden. Im kontextsensitiven Modus werden dem Benutzer nur die Werte in tabellarischen Kriterien zur Auswahl gegeben, die auch zu Selektionsergebnissen mit den bisher eingegebenen Werten anderer Selektorkriterien führen. Im kontextinsensitiven Modus werden alle in der Datenbank abgelegten Werte für ein tabellarisches Kriterien dem Benutzer angeboten, unabhängig von den anderen, bereits gemachten Selektoreingaben.

Im Feld **Maximale Durchführungszeit der Anfrage** kann eingestellt werden, ob die Selektoranfrage zeitlich begrenzt sein soll. Diese Option ist derzeit allerdings nicht möglich (programmiert und auskommentiert), da der Oracle OCI-Datenbanktreiber in diesem Modus noch instabil ist.

Im Feld **Maximale Anzahl der Ergebnisdatensätze** kann eingestellt werden, ob die Ergebnismenge der Selektoranfrage begrenzt werden soll. Bei einer Begrenzung kann im Eingabefeld die maximale Anzahl der Ergebnisse eingegeben werden.

Die Einstellungen für Maximale Durchführungszeit der Anfrage und Maximale Anzahl der Ergebnisdatensätze beziehen sich ausschließlich auf die Selektoranfrage und nicht auf Anfragen, die Werte für Conditions aus der Datenbank holen.

3.5 Datenflußfenster

Im Fenster Datenfluß (Abbildung 3.12) werden allgemeine Daten zu einer Selektoranfrage visualisiert, nachdem eine Selektion durchgeführt wurde bzw. im Selektionsfenster alle Eingaben gemacht wurden und dann auf Start gedrückt wurde (siehe Kapitel 3.2).

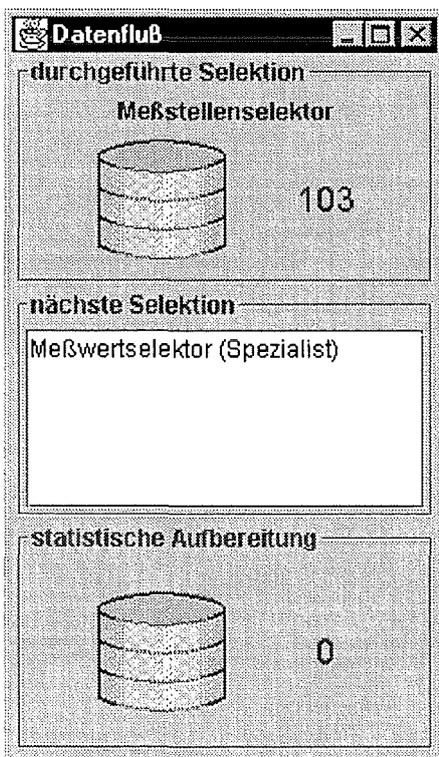


Abbildung 3.12: Datenflußfenster

Im Feld **durchgeführte Selektion** wird der Name des Selektors und die Anzahl der gefundenen Ergebnisse angezeigt.

Im Feld **nächste Selektion** wird eine Liste von möglichen Selektoren angeboten, die mit dem aktuellen Selektor verknüpft werden können (Selektorchaining). Die Auswahl für den nächsten Selektor erfolgt über Doppelklick auf den Selektornamen.

Beim Selektorchaining (Selektorverknüpfung) dient das Ergebnis einer Selektoranfrage als Vorbelegung für einen nachfolgenden Selektor. Zwei Selektoren sind grundsätzlich dann verknüpfbar, falls die Schlüsselattribute der Ergebnismenge des ersten Selektors gleich den Schlüsselattributen eines Selektorkriteriums des zweiten Selektors sind.

Beispiel: Der Meßstellenselektor liefert als Ergebnis eine Menge von Meßstellen. Der Meßwerteselektor besitzt als Selektorkriterium ein Meßstellenkriterium und kann somit im Selektorverknüpfungsfall die Ergebnismenge des Meßstellenselektors als Vorbelegung für das Meßstellenkriterium benutzen.

Zusätzlich zum Selektorergebnis werden auch alle gleichen Selektorkriterien der zwei zu verknüpfenden Selektoren übernommen. Alle vorbelegten Selektorkriterien werden gesperrt und können nicht mehr abgeändert werden.

Im Feld **statistische Aufbereitung** wird in der nächsten Version des SDS die Anzahl der Ergebnisse nach der statistischen Aufbereitung angezeigt.

3.6 Darstellungsauswahl

Die Darstellungsauswahl (Abbildung 3.13, Abbildung 3.14) ermöglicht die tabellarische Ausgabe eines Selektorergebnisses sowie die Auswahl einer Darstellungsart, in der die Selektorergebnisse dann mit Hilfe des GISterm dargestellt werden.

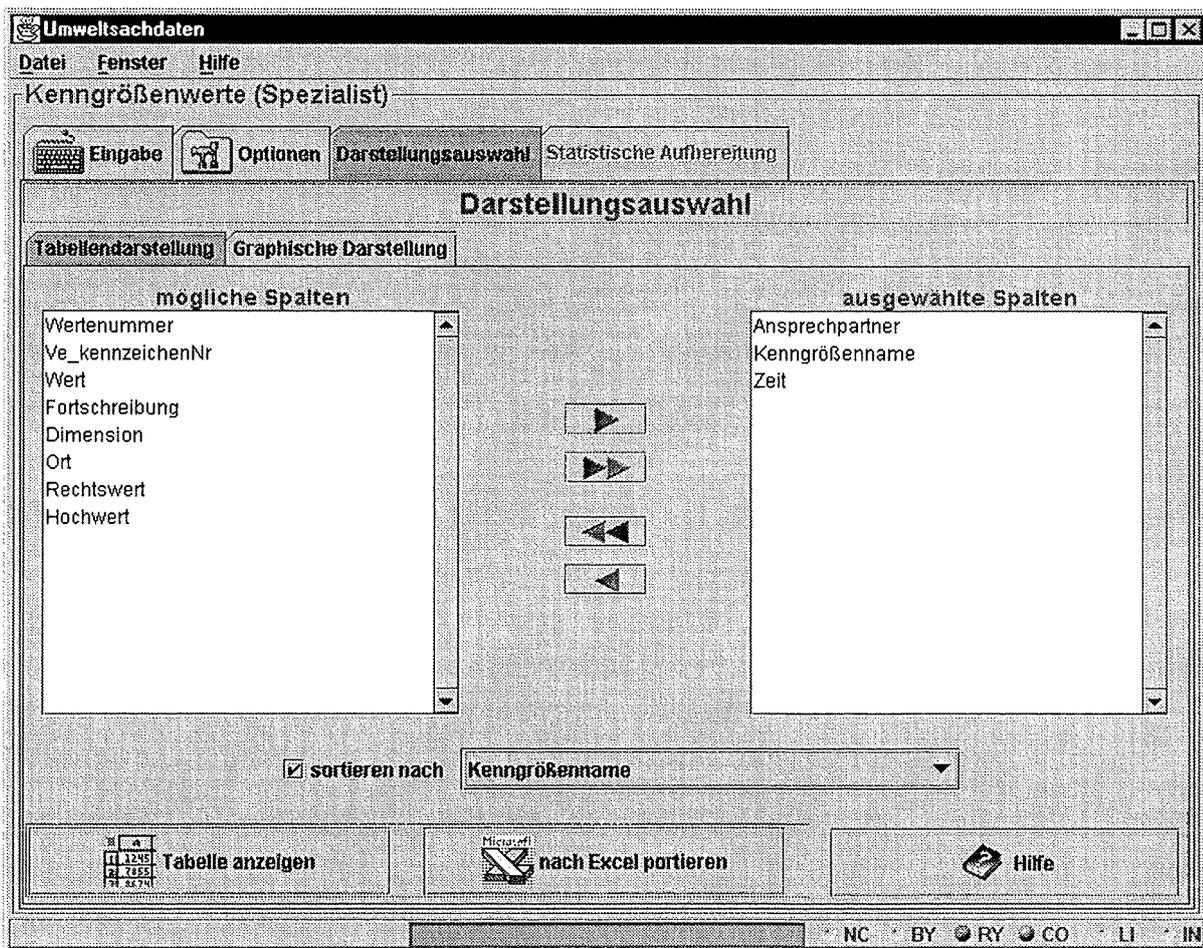


Abbildung 3.13: Auswahl der Tabellendarstellung

Die Karteikarte „Tabellendarstellung“ (Abbildung 3.13) ermöglicht es, das Selektionsergebnis als Tabelle am Bildschirm anzuzeigen oder an das Tabellenkalkulationsprogramm Excel zu übergeben. In der linken Liste sind alle Spalten der Selektionstabelle aufgelistet. Durch Betätigen der Knöpfe in der Mitte des Fensters können die Spalten einzeln oder alle in die rechte Liste übernommen werden. Dadurch wird festgelegt, welche Spalten angezeigt werden sollen. Zusätzlich kann angegeben werden, ob nach einer Spalte sortiert werden soll oder nicht. Die Spalte, nach der sortiert werden soll, muß nicht in der Liste der anzuzeigenden Spalten auftauchen.

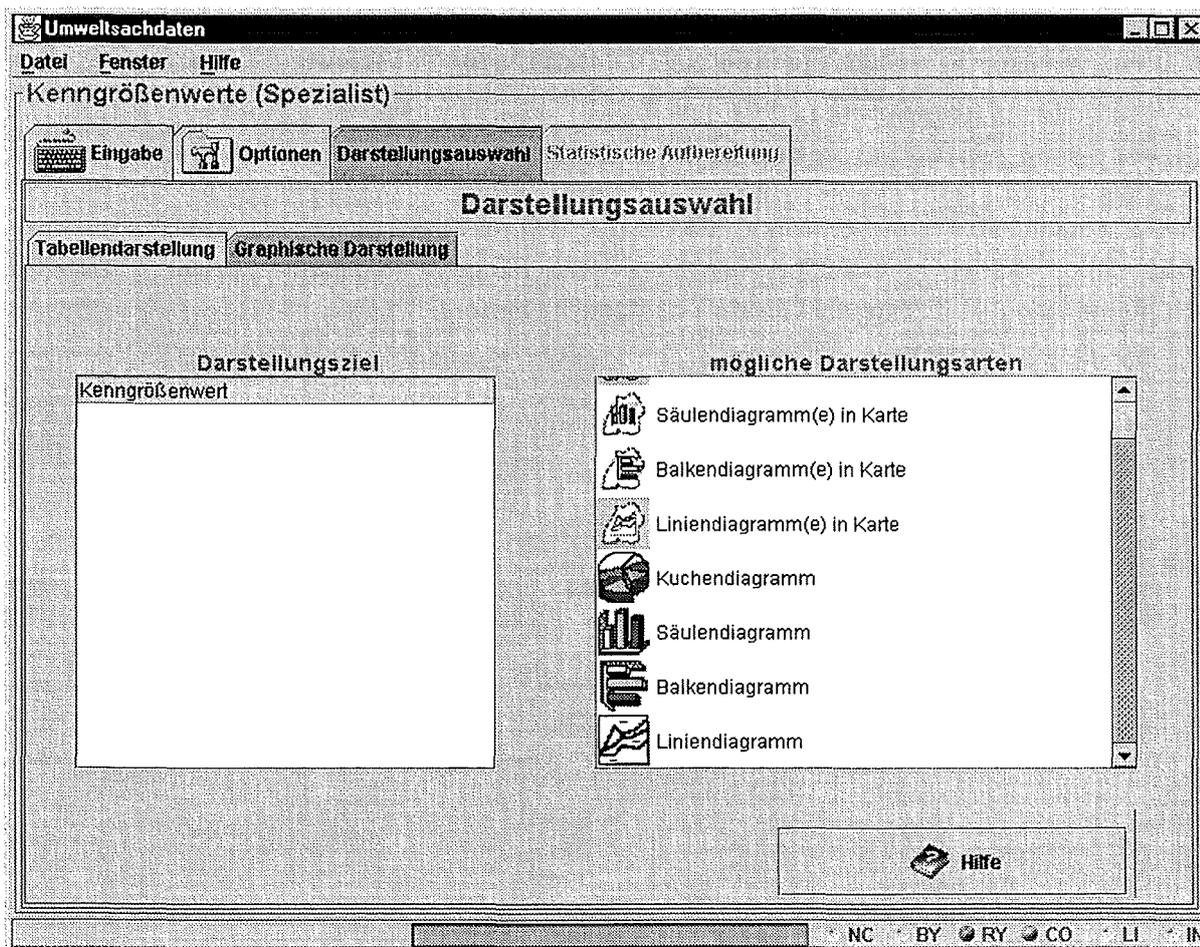


Abbildung 3.14: Darstellungsauswahl

In der Karteikarte „Graphische Darstellung“ (Abbildung 3.14) muß zunächst eines der möglichen Darstellungsziele aus der linken Liste angeklickt werden. In Abhängigkeit davon sind verschiedene Darstellungsarten möglich, die in der rechten Liste angewählt werden können. Wenn dann eines der nun anklickbaren Darstellungsarten ausgewählt wird, wird das Selektorergebnis in der gewünschten Darstellungsart vom GIStern visualisiert. Die Darstellungsziele sowie die mit den Darstellungszielen möglichen Darstellungsarten werden vom System auf Grundlage von sogenannten Graphikregeln für jedes Selektorergebnis einzeln ermittelt.

4. Wiederverwendbare JavaBeans und das Selektionsframework

Neben funktionalen Verbesserungen und Erweiterungen sollten wiederverwendbare Komponenten aus dem SDS identifiziert und anderen Softwareentwicklern verfügbar gemacht werden. Das Ergebnis sind unter anderem die drei GUI-Beans, die in Abschnitt 4.1 vorgestellt werden. Desweiteren sollte ein Framework bereitgestellt werden, auf dessen Basis neue Recherche-Tools aufbauen können und das die Basisfunktionalität eines solchen Systems liefert. Dieses wird in Abschnitt 4.2 erläutert.

4.1 Eingabe- und Visualisierungs-Beans

4.1.1 FilterableTable

Beschreibung:

Die FilterableTable (siehe Abbildung 4.1) erweitert die Swing Bean JTable um die Möglichkeit, die Zeilenausgabe der Tabelle auf eine Untermenge aller Zeilen zu beschränken. Dazu filtert ein mittels eines Vergleichsmusters zu konfigurierender Filter passende Zeilen einer frei wählbaren Spalte heraus.

MESSTELLEN_NAME_1	MESSTELLEN...
DONAUESCHINGEN	BRIGACH
KARLSRUHE NORDWE...	
MUELHEIM POLIZEIREVI	

Filter Spalte #1 Teilwort

Abbildung 4.1: FilterableTable

Das Vergleichsmuster für den Filter einer mittels einer JComboBox frei wählbaren Spalte (oben wurde „Filter Spalte #1“ ausgewählt) wird in einem Textfeld eingeben und ignoriert die Groß- oder Kleinschreibung. Mittels eines Checkbuttons kann zwischen den beiden Filtermodi "Wortanfang" und "Teilwort" gewechselt werden. Der Filter aktualisiert nach jedem eingegebenen Zeichen die Datentabelle und filtert dort je nach Modus gleiche Wortanfänge oder Teilworte heraus.

Desweiteren können die Zeilen einer frei wählbaren Spalte alphanumerisch sortiert werden. Dies geschieht entweder durch einen Methodenaufruf, kann aber auch interaktiv vom Benutzer zur Laufzeit geschehen. Dazu ist auf die Überschrift der zu sortierenden Spalte zu klicken. Bei gedrückter Shift-Taste wird absteigend, ansonsten aufsteigend sortiert.

Methoden, Properties, Events:

Die FilterableTable wurde mittels Swing entwickelt und setzt wie diese Klassenbibliothek die Trennung zwischen GUI (View) und Anwendungslogik (Model) um.

GUI:

Die FilterableTable erbt alle Bean-Features von ihrer Oberklasse JPanel. In diesem sind die JTable, das JTextField und der JCheckBox eingebettet, von denen die JTable mit getJTable zur direkten Bearbeitung erfragt werden kann.

Model:

Die eigentliche Funktionalität wird durch eine Verkettung spezieller neuer TableModels für die JTable erzielt, die aber für den Anwender oder Programmierer bis auf eine Ausnahme nicht in Erscheinung treten.

Die FilterableTable wird mit einem DefaultTableModel initialisiert, welches mit den neuen TableModels intern verkettet wird. Änderungen an diesem Original wirken sich direkt auf die JTable aus, und dieses Original sollte vom Anwender zur Bearbeitung benutzt werden. Die internen TableModels wirken sich nur beim Erfragen selektierter Zeilen der JTable aus. Hierfür gibt es eine Hilfsmethode (getSelectedRowInSrc()), die die Zeilenindices für das Original TableModel umrechnet.

Die genauen Schnittstellen sind der API-Dokumentation zu entnehmen.

4.1.2 HierarchieBrowser

Beschreibung:

Der HierarchieBrowser (siehe Abbildung 4.2) bietet eine zu einem TreeView alternative Ansicht für eine hierarchische Datenstruktur. Statt sich wie im TreeView durch die einzelnen Ebenen einer Hierarchie hangeln zu müssen, können die Ebenen direkt angesprungen werden. Dadurch bietet der HierarchieBrowser eine kompakte Ansicht auf Elemente gleicher Hierarchieebenen.

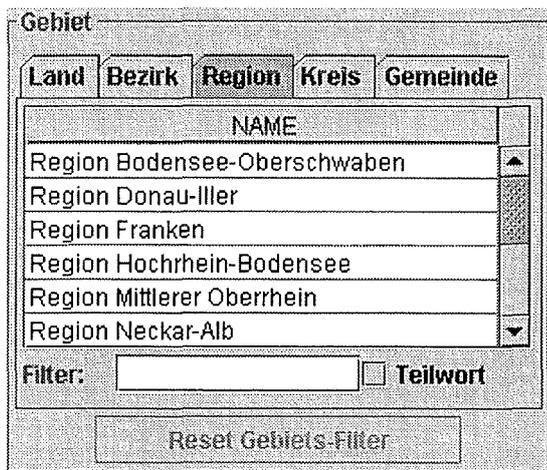


Abbildung 4.2: Der HierarchieBrowser

Die verschiedenen Ebenen werden dabei in je einer FilterableTable dargestellt, die nach dem Karteikarten-Prinzip aufgedeckt werden kann. In jeder Ebene kann mittels des Filters nach bestimmten Zeilen gesucht werden. So ist es sehr einfach, ein sehr verschachteltes Element der Hierarchie zu finden, ohne alle seine Vorgänger zu kennen.

Wird ein Element einer Ebene ausgewählt, zeigt ein Umschalten auf eine andere Ebene sofort alle seine Vorgänger oder Nachfolger, so daß die Hierarchiebeziehung einfach nachzuvollziehen bleibt. Auch ist es so mit nur einem Schritt möglich, zu einem Element alle "entfernten" Verwandten einer Ebene auszuwählen.

Schließlich ist es in der Tabellendarstellung möglich, tupelwertige Elemente darzustellen.

Methoden, Properties, Events:

Der HierarchieBrowser trennt wieder zwischen GUI und Model.

GUI:

Der HierarchieBrowser erbt von JPanel alle Bean-Features. In das JPanel ist ein JTabbedPanel eingebettet, das für jede Hierarchieebene eine eigene FilterableTable enthält. Die einzelnen Tabellen können mit `getTable(int level)` erfragt und bearbeitet werden. Eine Me-

thode `getModel()` liefert das Model zu dieser Bean, in der alle weiteren Funktionen und Properties zur Verfügung stehen.

Model:

Um die Beziehungen der Elemente unterschiedlicher Ebenen herzustellen, wird ein `HierarchieBrowserModel` benötigt. Dieses Model ist nach dem Entwurfsmuster `Template` realisiert und muß für jede hierarchische Struktur abgeleitet und angepaßt werden. Die Anpassung besteht dabei in der Implementierung folgender Methoden:

- `isChildren(Object parent, Object element, Level)` : prüft ob cand entferntes Kind in Ebene Level ist.
- `isParent(Object children, Object element, Level)` : prüft ob cand entfernter Vorfahre in Ebene Level ist.
- `getLevel(Object element)` : gibt die Hierarchieebene des Elements zurück.

Mit diesem spezialisierten Methodensatz in der Unterklasse ist die `Template`-Klasse `HierarchieBrowserModel` in der Lage, sowohl die anfängliche Datenstruktur aufzubauen, als auch dynamisch die vom Benutzer ausgewählten verwandten Elemente zu ermitteln.

Das `HierarchieBrowserModel` bietet ferner folgende

Properties:

- `LevelNames`: Überschriften für die einzelnen Hierarchieebenen

Events:

- `SelectedLevelChanged`: Event, das den Wechsel einer Hierarchieebene meldet.

Methoden:

- `-addRow`: Hinzufügen einer Reihe, die automatisch in die richtige Ebene eingeordnet wird.
- `-removeRow`: Entfernen einer Reihe.

4.1.3 DateInput

Beschreibung

Die Bean `DateInput` (siehe Abbildung 4.3) ist ein Formularelement zur Eingabe von Zeitpunkten, Zeitintervallen und relativen Zeitangaben. Hierzu wird je nach Eingabeformat und Genauigkeit eine Maske angeboten, die die Benutzereingaben soweit möglich auf ihre Konsistenz überprüft, um falsche Zeiteingaben, wie den 32. Januar zu verhindern.

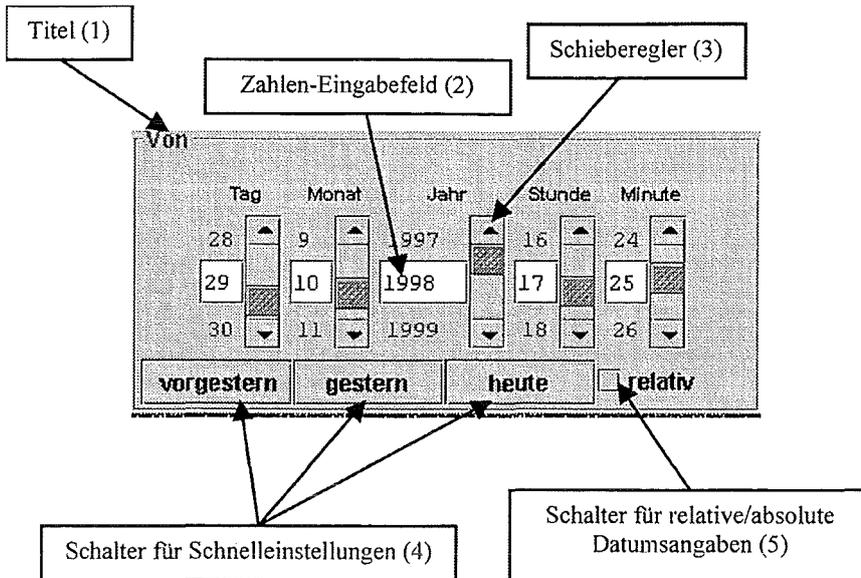


Abbildung 4.3: Zeitkriterium

Die Eingabe kann dabei auf drei Arten abgefragt werden:

- Als Instanz eines Objektes der Klasse `java.util.Calendar`, zur direkten Weiterverwendung in Java.
- Als String der gemäß einer vorher frei wählbaren Formatvorgabe formatiert ist, z.B. "DDMMYYYYhhmmss", und so z.B. in eine Datenbankabfrage eingebunden werden kann.
- Als anonymes Objekt für relative Datumsangaben, so daß ein relatives Datum abhängig vom aktuellen Datum ausgewertet werden kann.

Im Falle eines Zeitintervalls besteht die Eingabe und demnach auch die Ausgabe aus zwei Zeitpunkten. Eine relative Zeitangabe bezieht sich auf das aktuelle Datum, wobei eine vom Benutzer angegebene Anzahl von Jahren, Monaten, Tagen, Stunden, Minuten und Sekunden addiert bzw. subtrahiert wird. So ist es z.B. möglich, eine Zeitangabe wie „gestern“ oder „nächstes Jahr“ anzugeben.

Wie in Abbildung 4.3 visualisiert, kann die Eingabe eines Datums über eine Zahleneingabe (2) in den entsprechenden Feldern oder über die zu den Eingabefeldern gehörigen Schieberegler (3) erfolgen. Die blauen Ziehknöpfe in den Schiebereglern rasten an der jeweils erste und letzten Position ein, so kann bequem und schnell beispielsweise zum letzten Tag des Monats oder zum ersten Monat des Jahres gesprungen werden. Über die Pfeilknöpfe an den Schiebereglern kann die Einstellung für das Feld frei gewählt werden: Bei Stunde 23 würde ein Druck auf den Pfeil nach unten beim Stundenfeld wieder Stunde 0 des nächsten Tages einstellen. Das vorgestrige, gestrige und heutige Datum kann über die Schnelleinstellungsknöpfe (4) mit einem Klick eingestellt werden. Der Schalter (5) dient für die Auswahl zwischen relativen und absoluten Datumseingaben. Ist dieser Schalter mit einem Häkchen versehen, so werden für die einzelnen Felder Eingaben zwischen -50 und 50 zugelassen. Die eingestellten Werte werden bei einer Abfrage des Datums dann zu den jeweiligen Feldern addiert. Eine Einstellung von -1 im Feld für den Tag und eine 0 in den übrigen Feldern repräsentiert dann das Datum „gestern“.

4.2 Selektionsframework

Unter einem Framework versteht man eine Menge kooperierender Klassen, welche die Elemente eines wiederverwendbaren Entwurfs für eine bestimmte Art von Software darstellen. Ein Framework bietet eine Architekturhilfe beim Aufteilen des Entwurfs in abstrakte Klassen und beim Definieren ihrer Zuständigkeiten und Interaktionen. Ein Entwickler paßt das Framework für eine bestimmte Anwendung an, indem er Unterklassen der Frameworkklassen bildet und ihre Objekte zusammensetzt /2/.

4.2.1 Funktionalität des Frameworks

Das Ziel des Selektionsframework ist es, inklusive der in Kapitel 4.1 vorgestellten Beans eine komfortable DB-Suchanfrage zu erstellen, und eine erweiterbare Grundlage für die Zusammenarbeit mit weiteren Diensten anzubieten. Beliebige Dienste können zur Verarbeitung und graphischen Darstellung der Ergebnisse registriert werden. Intern greift das System auf ein Repository zu, um Beschreibungen für Selektoren, Selektorkriterien und Datenquellen zu erhalten. Aus diesen Beschreibungen wird die GUI generiert. Nach einer Benutzerinteraktion wird eine entsprechende DB-Anfrage generiert, und das Ergebnis inklusive der Metadaten (Darstellungsregeln) wird an registrierte Dienste übermittelt.

Die Abbildung 4.4 zeigt die durch das Framework bereitgestellte GUI.

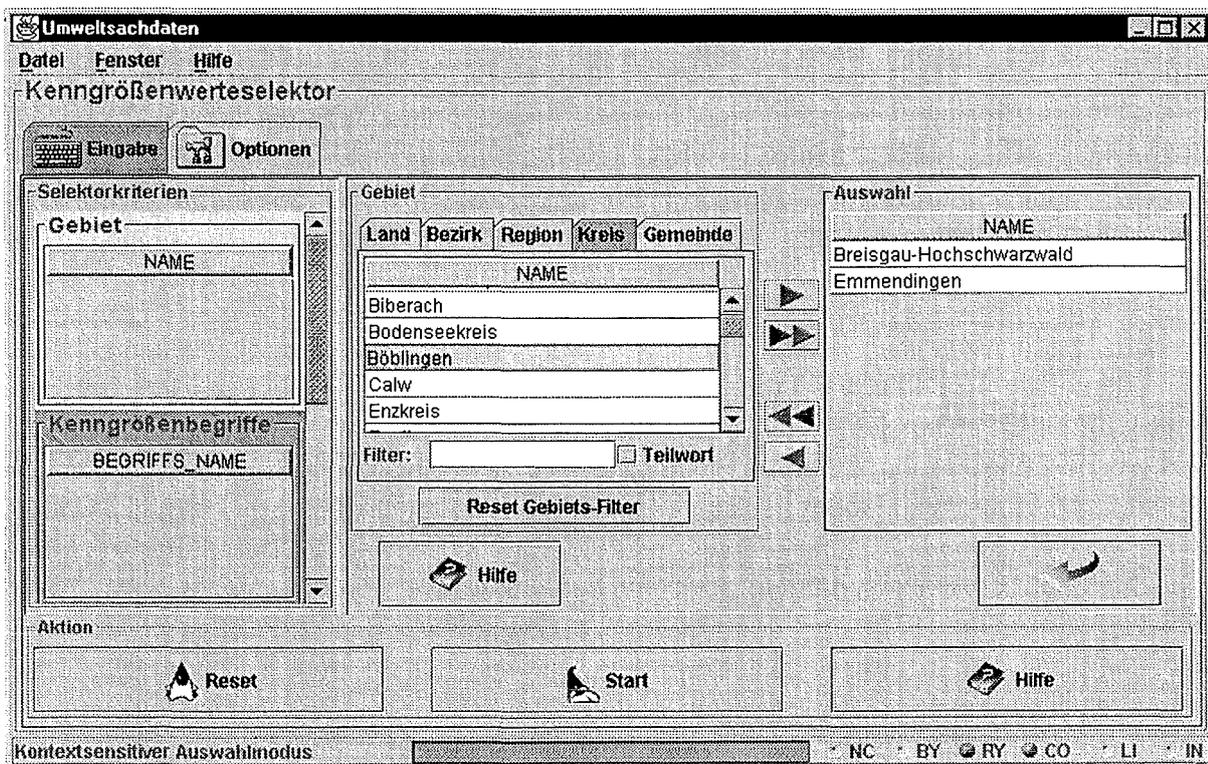


Abbildung 4.4: Selektionsframework

4.2.2 Initialisierung des Frameworks

Zur Initialisierung des Frameworks wird vom Repository ein Objekt der Klasse `RepositoryDescription` erhalten. Die Repositorybeschreibung enthält verfügbare Selektoren und die dazugehörigen `SelectorDescription`-Objekte. Die aus den `SelectorDescription`-Objekten entnommenen `ConditionDescription`- und `DataSourceDescription`-Objekte dienen als Vorlage für die zu erzeugenden Selektorkriterien und Datenquellen. Aus den `SelectorDescription`-Objekten werden die Selektoren erzeugt.

In einem optionalen Schritt können `Service`-Objekte (Dienste) in den erzeugten Selektor eingebunden werden (siehe Kapitel 3.3.3). Diese Dienste können als GUI-Komponenten im Selektorfenster dargestellt werden und auf Änderungen des Selektionsergebnisses mittels eines Ereignismechanismus reagieren. Sämtliche interne Anwendungslogik und GUI-Komponenten sind ebenfalls von der konkreten Anwendung zur Verfügung zu stellen.

4.2.3 Schnittstellen

Das Framework besteht aus einer Menge von Schnittstellen (interface) und abstrakten Klassen (abstract class) (siehe Abbildung 4.5). Diese Schnittstellen und Klassen definieren Methoden zur Erzeugung und Manipulation von Objekten, die diese Schnittstellen und Klassen implementieren und erweitern.

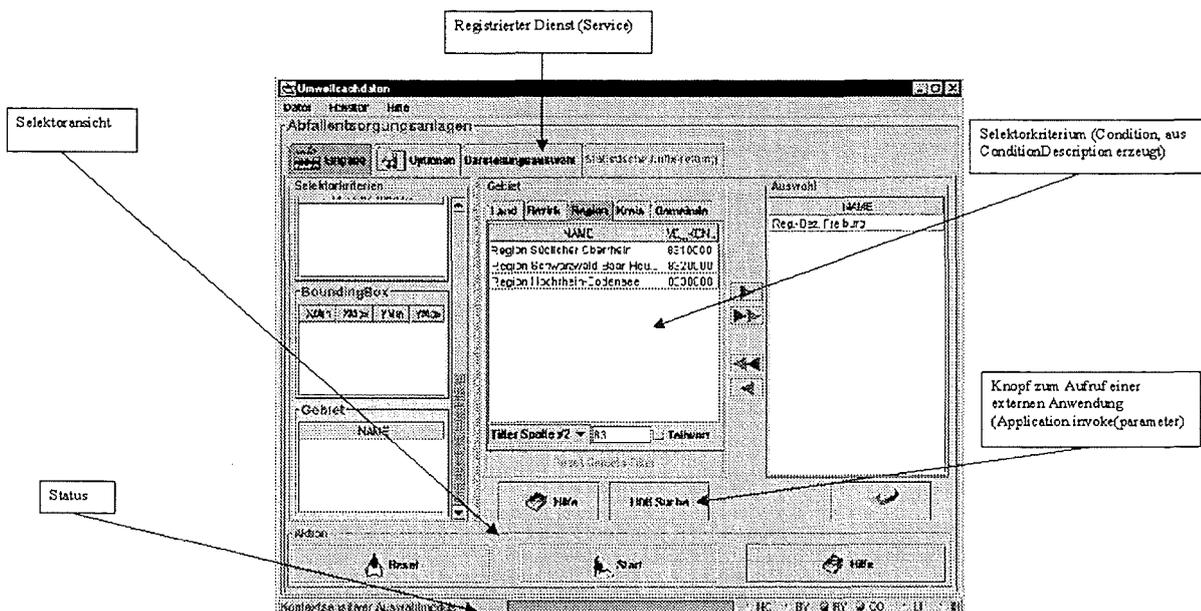


Abbildung 4.5: Erweiterbarkeit des Selektionsframeworks

4.2.3.1 Abfrage von Stausinformationen

Der Status des Selektors kann jederzeit durch die Methode `getStatus()` abgefragt werden. Außerdem können dem Selektor Komponenten der Klasse `StatusListener` hinzugefügt werden, welche auf Statusänderungsereignisse reagieren können und die eine grafische Ausgabe der Statusinformationen realisieren.

4.2.3.2 Dynamische Conditiontypen

Das Framework sieht eine Methode zur Konstruktion von Conditions über den Aufruf von `createCondition()` der `ConditionDescription`-Schnittstelle vor. Sollten weitere Conditions neuer Typen erzeugt werden, so kann ein Objekt mit der Schnittstelle `ConditionFactory` zur Erzeugung von neuen Conditions eingesetzt werden. Bei neuen Conditiontypen müssen also zusätzlich zu Änderungen im Repository neue Java-Klassen entworfen, implementiert und der SDS-Anwendung oder dem SDS-Applet zugänglich gemacht werden.

4.2.3.3 Einbindung von Benutzerdiensten (Teildialogen)

Jedes Objekt, dessen Klasse die Schnittstelle `Service` implementiert, kann in ein Objekt mit der Schnittstelle `Selektor` eingebunden werden. Beim Einfügen eines `Service`-Objekt in ein `Selektor`-Objekt kann sich das `Service`-Objekt als `ResultChangeListener` beim `Selektor`-Objekt anmelden, dann kann ein solches `Service`-Objekt Ergebnisse einer Selektion durch das Ereignis `resultChanged(ResultChangeEvent evt)` erhalten. Das Ergebnis einer Selektion ist ein Objekt mit der Schnittstelle `Result`. Von einem `Result`-Objekt kann das eigentliche Ergebnis in Tabellenform mit der Methode `getResultTable()` erhalten werden, die dem Ergebnis zugehörigen Metadaten (z.B. graphische Darstellungsregeln) können mit der Methode `getResultMetadata()` abgefragt werden.

4.2.3.4 Aufruf von externen Anwendungen

Jede externe Java-Anwendung, die die Schnittstelle `Application` implementiert, kann über die Methode eines `Selektors` `invokeApplication(Application target, String parameter)` aufgerufen werden. Der Parameter des Aufrufs ist im allgemeinen ein Suchbegriff. Die externe Anwendung sollte über den Aufruf `invoke(String parameter)` auf diesen Aufruf reagieren.

4.2.4 Erweiterbarkeit und Abdeckungsbereich

Das Framework kann durch Bildung neuer Schnittstellen oder Klassen erweitert werden. Zur Erweiterung kann Klassenvererbung eingesetzt werden, oder dem Framework können neue Klassen- und Schnittstellen-Hierarchien hinzugefügt werden. Das Framework bietet ein Konzept auf einer hohen Abstraktionsebene. Das momentane Interaktionsmodell des bestehenden SDS ist in dem Framework enthalten. Es werden weiterhin Schnittstellen zur Integration von neuen Conditiontypen, neuen Diensten und externen Applikationen angeboten.

5. Administration: Erstellung neuer Selektoren

Um den Benutzern einen neuen Selektor zur Verfügung zu stellen, legt der Administrator Repository-Dateien nach einem bestimmten Format an. Programmiersprachenkenntnisse oder Code-Kenntnisse sind nicht erforderlich. Bevor er den Selektor dann frei gibt, kann er seine Beschreibung mit einem Parser auf syntaktische Korrektheit testen. Der Parser gibt zeilengenau an, wenn eine Angabe nicht der von ihm lesbaren Syntax entspricht.

Haben die neuen Dateien diesen Syntaxtest bestanden, dann kann der Administrator diese den neuen Selektor beschreibenden Dateien in das Repository-Verzeichnis kopieren, das vom SDS-Server ausgelesen wird.

Der SDS-Server parst beim Start automatisch das Repository. Clients (SDS-Java Applets) gibt der SDS-Server beim Start der Clients die bei seinem Start geparsten Selektordaten. Um nun auch während des laufenden Server- und Client-Betriebs neue, gerade definierte Selektoren ausführen zu können, kein jeder beliebige Client veranlassen, daß der Server das Repository neu parst. Damit erhalten alle danach gestarteten Clients oder solche, die die Selektorinformationen nochmals vom Server lesen möchten, das aktuelle, um einen oder mehrere neue Selektoren ergänzte Repository.

Teil des SDS Servers ist also ein Parser, der die Aufgabe hat, die Dateien im Repository zu parsen und entsprechend der darin enthaltenen Strukturen die Beschreibungen zum Aufbau des Selektoren zu erstellen. Der Parser wurde dabei aus einer Grammatik mit der für das Repository definierten Tags erstellt. Er basiert auf einer LISP ähnlichen Syntax. Genauere Beschreibungen der Syntax seinem dem Repository-Handbuch entnommen /3/.

Abbildung 5.1 verdeutlicht den Aufbau des Repositories. Für das gesamte Repository gibt es eine *Konfigurationsdatei*, die alle globalen Parameter beinhaltet, wie z.B. die Datenquellenbeschreibung etc. Hier können auch die Selektoren namentlich genannt werden, die Teil des Repositories sind, obwohl mehr Dateien mit Selektorbeschreibungen innerhalb des Repositories existieren. Alternativ kann hier auch ein alle-Tag angegeben werden, der bestimmt, daß alle im Repository befindlichen Selektoren geparst werden.

Für jeden Selektor gibt es eine *Selektordatei*, die alle Angaben zu dem speziellen Selektor und Referenzen auf die innerhalb dieses Selektors verwendeten Kriterien enthält. Jedes Kriterium wird in einer separaten Datei (*Kriteriumsdatei*) beschrieben. Durch die Methode der Referenzen können dieselben Kriterien in mehreren Selektoren verwendet werden.

Die Selektordatei enthält neben den Verweisen auf die Kriterien außerdem die Auflistung aller in der Endabfrage verwendeten Felder und die zu selektierende Zieltabelle. Für das Selektorchaining wird in der Selektordatei auch ein Kriterium angegeben, das in anderen Selektoren benutzt wird und das das Ergebnis dieses Selektors beim Chaining übernimmt. D.h. wenn Selektor 2 auf den Ergebnissen von Selektor 1 aufbauen soll, dann muß Selektor 2 das Kriterium besitzen, das in Selektor 1 als Chainingkriterium angegeben wurde.

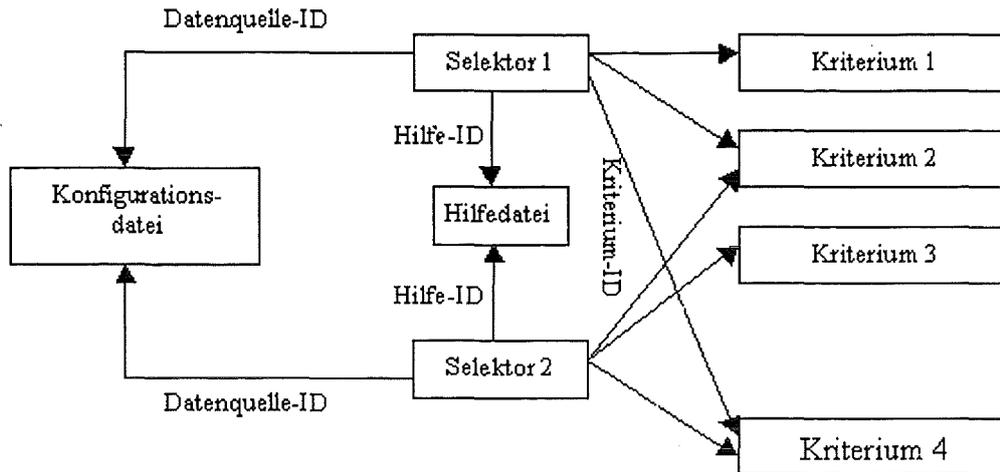


Abbildung 5.1 Aufbau des Repositories

In der Selektordatei wird außerdem noch ein Verweis auf einen Eintrag in der *Hilfdatei* gegeben. Über die Hilfe-ID ist eine URL innerhalb der Hilfdatei identifizierbar, die aufgerufen wird, wenn in der graphischen Oberfläche auf Hilfe gedrückt wird.

6. Zusammenfassung und Ausblick

Das Sachdatensystem (SDS) des UIS Baden-Württemberg stellt nun ein breit einsetzbares, leicht bedienbares Tool zur komfortablen Selektion von Umweltdaten dar: Benutzer füllen fachspezifische Formulare - sogenannte Selektoren - aus, um zu den gewünschten Daten zu gelangen. Mit Hilfe des GIS-Terms können diese Daten dann direkt visualisiert werden. Die Verwendung von Web-Technologien ermöglicht einen Einsatz sowohl im Landes-Intranet als auch im Internet. Umweltsachdaten können somit einem großen Benutzerkreis zur Verfügung gestellt werden.

Auch das Hinzufügen neuer Selektoren zum SDS wird den Administratoren nun sehr leicht gemacht: Sie müssen speziell formatierte Dateien erstellen. Diese Dateien beschreiben die Selektoren und bilden das Selektorrepository. Eingriffe in den Code oder Kenntnisse einer Programmiersprache sind für das Erstellen der Selektoren nicht notwendig. Der Nachweis der Praktikabilität dieses Ansatzes wurde durch LfU-Mitarbeiter bereits in den letzten Monaten erbracht. Sie erstellten selbständig und problemlos neue Selektoren nur unter Zuhilfenahme des Handbuchs.

Im Rahmen von GLOBUS V wurden eine Reihe von Verbesserungen des in GLOBUS IV entwickelten Systems vorgenommen. Hierzu zählt die Überarbeitung der graphischen Oberfläche und des Selektorrepositorys. Desweiteren wurde aufgrund einer Analyse der SQL-Anweisungen auch die Performanz des Systems erheblich verbessert, insbesondere hinsichtlich der Durchführung von kontextsensitiven Selektoren. Es kamen auch neue Features hinzu, wie z.B. das sogenannte Selektor chaining, das die Verknüpfung und Hintereinanderführung von Selektoren ermöglicht.

Ein wesentliches Ergebnis von GLOBUS V ist auch die Bereitstellung von drei GUI-Beans sowie dem Selektionsframework. Die Wiederverwendung hat bereits begonnen, da Code und Dokumentation bereits den Entwicklern des Biotoperfassungssystem zur Verfügung gestellt wurden.

Die Recherche in Biotopdaten wird in Zukunft auch direkt über das SDS stattfinden und somit wird die Basis an über eine einheitliche Schnittstelle erreichbaren Umweltdaten weiter verbreitert.

Potentielle Erweiterungen des SDS liegen in der Realisierung von Benutzerarchiven, der statistischen Aufbereitung von Selektionsergebnissen und der Berichterstellung. Benutzerarchive geben den Nutzern die Möglichkeit sowohl ausgefüllte Selektoren als auch erzielte Ergebnisse unter Angabe von Useraccount und Paßwort persistent abzulegen, um zu einem späteren Zeitpunkt in einer anderen Sitzung diese wieder zu laden. Mit Hilfe einer statistischen Aufbereitung können Benutzer statistische Berechnungen auf den Selektionsergebnissen durchführen und dann diese Statistiken mit dem GIS-Tool visualisieren. Die Berichterstellung ermöglicht das Ausdrucken von Erfassungs- und Datenbögen entsprechend vorgegebener Layouts.

7. Literatur

- /1/ Mayer-Föll, R.; Jaeschke, A. (1997): Projekt GLOBUS: Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg; Phase IV 1997, Wissenschaftliche Berichte des Forschungszentrum Karlsruhe, FZKA 6000, 1997
- /2/ Gamma et.al.(1995): Entwurfsmuster, Addison-Wesley, 1995
- /3/ C. Rolker, S. Behrens, T. Koepke (1998): Das Repository-Handbuch des Sachdatensystems, Forschungszentrum Informatik (FZI) und Disy Informationssysteme GmbH, 1998

Zertifizierung und Signierung von Java Applets für deren Nutzung in Internet-basierten Umweltinformationssystemen

*K. Pursche, C. Fuchs, K. Tochtermann,
Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW)
an der Universität Ulm,
Helmholtzstr. 16,
89081 Ulm*

*A. Koch,
Landesanstalt für Umweltschutz,
Informationstechnisches Zentrum Stuttgart,
Postfach 13 09 10,
70067 Stuttgart*

*H. Spandl,
Landesanstalt für Umweltschutz,
Postfach 210 752,
76157 Karlsruhe*

*J. Arnold,
Innenministerium Baden-Württemberg,
Stabsstelle für Verwaltungsreform,
Dorotheenstrasse 6,
70173 Stuttgart*

1. EINLEITUNG	65
2. TECHNISCHE LÖSUNGSMÖGLICHKEITEN ZUM SIGNIEREN VON JAVA APPLETS	65
2.1 BESCHAFFUNG EINES ENTWICKLER-ZERTIFIKATES	66
2.2 SIGNIEREN VON JAVA APPLETS	67
3. SIGNIERTE APPLETS IM GLOBUS-UMFELD	69
4. AUSBLICK	69
5. LITERATUR.....	70

1. Einleitung

Mit der Programmiersprache Java der Firma Sun steht heute eine Technologie zur Verfügung, Softwarelösungen weitgehend plattformunabhängig über das WWW zur direkten Anwendung zur Verfügung zu stellen. Dafür bietet Java die Möglichkeit der Entwicklung sogenannter Java Applets. Ein solches Applet kann mit Ausnahme einiger Sicherheitsrestriktionen beliebige Funktionalitäten erfüllen. Der Unterschied zu herkömmlichen Anwendungen besteht darin, daß Java Applets nicht auf jedem Rechner installiert sein müssen, sondern bei Bedarf über das WWW geladen und ausgeführt werden können.

Im Umweltbereich werden bereits eine Reihe auf Java Applets basierender Softwarelösungen genutzt. So wurde im Rahmen des Forschungs- und Entwicklungsprojektes GLOBUS des Ministeriums für Umwelt und Verkehr (UVM) Baden-Württemberg /1/ der Geodaten-Viewer GIStern am Institut für Photogrammetrie und Fernerkundung (IPF) der Universität Karlsruhe /2/ entwickelt. Zukünftig ist aufgrund der hohen Funktionalität eine starke Zunahme des Einsatzes von Java Applets zu erwarten.

Als problematisch bei der Nutzung von Java Applets stellen sich allerdings die Sicherheitsrestriktionen in Java dar, die auch als Sandbox-Prinzip bezeichnet werden. Aufgrund dieses Prinzips sind Java Applets u.a. folgende Funktionen auf dem Rechner des Nutzers nicht gestattet:

- lokale Dateien lesen oder schreiben
- drucken
- Aufbau von Netzwerkverbindungen
- andere Anwendungen starten
- anderen Anwendungen Nachrichten senden
- Benutzereigenschaften abfragen

Einen Ausweg aus dieser Situation bietet das Signieren der Java Applets. Im Rahmen des Forschungs- und Entwicklungsprojektes GLOBUS des UVM Baden-Württemberg wurde für Entwickler und Nutzer nach technischen Lösungen zu dieser Thematik gesucht.

2. Technische Lösungsmöglichkeiten zum Signieren von Java Applets

Die Signierung von Java Applets basiert auf Verschlüsselungsverfahren der Kryptographie. Zwei grundsätzliche Ansätze, die symmetrische und die asymmetrische Verschlüsselung, sind hier in der Literatur zu finden /3/ /4/. Die Signierung von Java Applets verwendet asymmetrische Verschlüsselungsverfahren.

Einmalig wird im lokalen Internet-Browser des Softwareentwicklers ein Schlüsselpaar generiert. Dieses Paar besteht aus einem privaten und einem öffentlichen Schlüssel. Der private Schlüssel dient der Signierung von Java Applets durch den Entwickler. Um Mißbrauch vorzubeugen, sollte dieser Schlüssel sicher verwahrt werden /5/ /6/.

Der öffentliche Schlüssel wird Teil des Zertifikates. Die sogenannte Zertifizierung des öffentlichen Schlüssels kann durch ein öffentliches Trust-Center, einen Certificate Server bzw. durch Selbst-Signierung vorgenommen werden (siehe Abbildung 1). Der Entwickler muß nun sein Zertifikat öffentlich den Nutzern zur Verfügung stellen. Erst nachdem der Nutzer das Zertifikat importiert und somit in seinem Browser bekannt gemacht hat, kann er mit diesem Zertifikat signierte Java Applets ausführen.

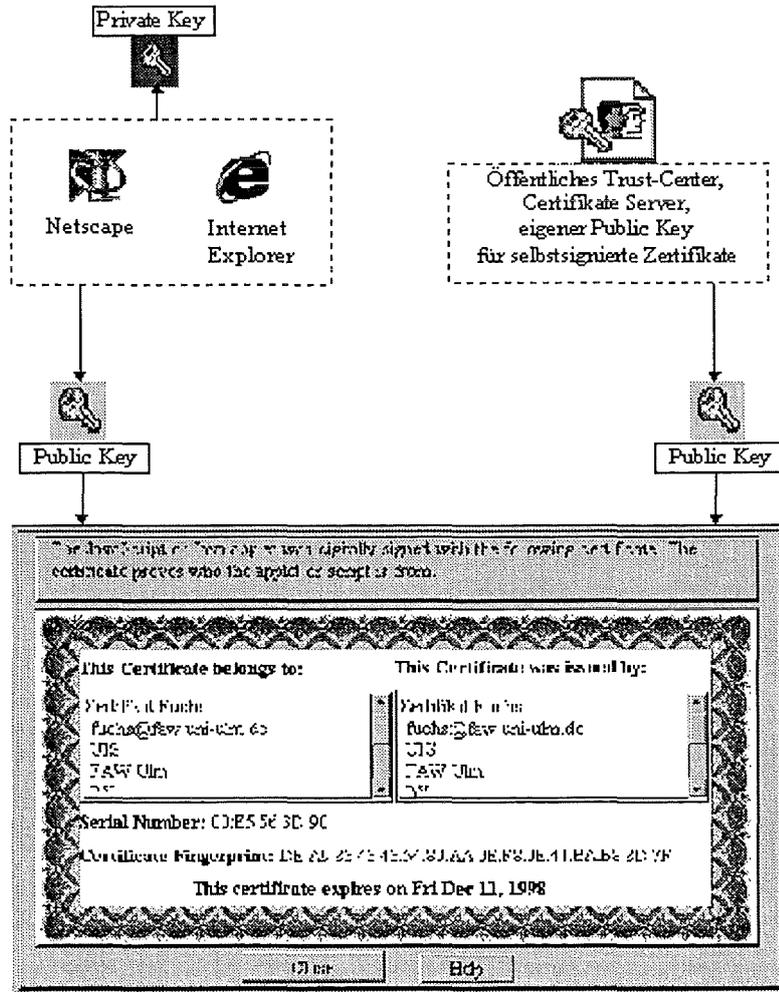


Abbildung1: Generierung und Aufbau eines Zertifikates

2.1 Beschaffung eines Entwickler-Zertifikates

Ein Entwickler-Zertifikat kann man von einem öffentlichen Trust-Center, einem Certificate Server bzw. durch Selbst-Signierung erhalten. Vor- und Nachteile sowie die jeweiligen Kosten werden im folgenden beschrieben.

Öffentliche Trust-Center

Bei öffentlichen Trust-Centern handelt es sich um Einrichtungen, die aufgrund der Überprüfung der Identität des Kunden Zertifikate ausstellen. Sie garantieren nicht für die Schadlosigkeit der Software, da sie diese nicht prüfen. Ein großer Vorteil der Verwendung von Zertifikaten öffentlicher Trust-Center liegt darin, daß sie in den Standard-Browsern als Zertifizie-

rungsstelle bereits eingetragen sind. Somit muß der Nutzer in diesem Fall kein zusätzliches Zertifikat importieren, um das signierte Java Applet ausführen zu können. Zusätzlich kann der Nutzer auf eine sogenannte Revocation List des Trust-Centers zugreifen, um die Gültigkeit von Zertifikaten zu überprüfen. Im Moment bieten nur zwei der bekannten Trust-Center Entwickler-Zertifikate an. Es handelt sich um VeriSign (400\$/Jahr und Zertifikat /7/) und Thawte (200\$/erstes Jahr und Zertifikat, 100\$/jedes weitere Jahr und Zertifikat /8/). Dabei ist zu beachten, daß jeweils ein Zertifikat für den Netscape Navigator bzw. MS Internet Explorer benötigt werden, wenn man Java Applets für beide Welten zur Verfügung stellen möchte.

Certificate Server

Ein Certificate Server stellt eine Art privates Trust-Center dar, dessen Anwendung in erster Linie in einem Intranet geeignet ist. Es dient dem Generieren und der Verwaltung von Zertifikaten. Certificate Server werden z.Z. von Netscape und Microsoft angeboten.

Der Microsoft Certificate Server ist bis zur aktuellen Version 1.0 nicht in der Lage, Entwickler-Zertifikate zu generieren. Da somit keine Software signiert werden kann, kommt dieser Certificate Server zur Signierung von Java Applets nicht in Frage.

Der Netscape Certificate Server ab V. 1.0 kann Entwickler-Zertifikate erstellen. Diese lassen sich allerdings nur für die Signierung von Java Applets für den Netscape Navigator verwenden. Der Netscape Certificate Server bietet alle notwendigen Funktionen zur Verwaltung von Zertifikaten aufgrund von Datenbankfunktionen. Wie bei öffentlichen Trust-Centern wird eine Revocation List zur Überprüfung der Gültigkeit von Zertifikaten durch den Nutzer erzeugt. Vorteilhaft für den Nutzer ist, daß er nur das Zertifikat des Netscape Certificate Servers laden muß, um alle damit signierten Zertifikate erkennen zu können. Der Netscape Certificate Server kostet z.Z. 525\$ /9/ für 100 Nutzerlizenzen.

Selbst-Signierung

Alle drei im Anschluß beschriebenen Signierwerkzeuge bieten die Möglichkeit der Generierung von selbstsignierten Zertifikaten. Das bedeutet, daß der öffentliche Schlüssel des Entwicklers nicht von einer zweiten Instanz signiert wird, sondern durch den eigenen privaten Schlüssel. Im Zertifikat tritt der Entwickler also sowohl als die Person auf, für die das Zertifikat ausgestellt wurde als auch die Person, die das Zertifikat ausgestellt hat. Der Nutzer hat bei dieser Vorgehensweise den Nachteil, daß er das Zertifikat jedes einzelnen Entwicklers explizit importieren muß, wenn er deren signierte Java Applets ausführen möchte. Die Signierwerkzeuge stehen kostenlos zur Verfügung.

2.2 Signieren von Java Applets

In der vorliegenden Arbeit steht das Signieren von Java Applets für die beiden Browser Netscape Navigator und MS Internet Explorer im Vordergrund. Für jeden der beiden Browser wird von den jeweiligen Herstellern ein Signierwerkzeug angeboten. Für den Netscape Navigator handelt es sich um Signtool /10/, für den MS Internet Explorer ist es Bestandteil des Software Developer's Kit (SDK) /12/. Eine weitere Möglichkeit besteht im Einsatz des Signierwerkzeugs Javakey /13/ im Zusammenhang mit dem Java Plug-In von Sun. Die hier genannte Software steht kostenlos auf den entsprechenden WWW-Seiten zur Verfügung.

Signtool (Netscape)

Signtool stellt das Signierwerkzeug für den Netscape Navigator dar. Eine eingehende Dokumentation der Vorgehensweise zum Signieren ist in /10/ zu finden.

Der Vorteil von Signtool liegt vor allem darin, daß die Sicherheitsstrategie des Netscape Navigator bei der Ausführung der so signierten Applets eingehalten wird. D.h. der Nutzer kann für jedes Entwickler-Zertifikat sehr feingranular die einzelnen Rechte festlegen. Für den Nutzer fällt außer dem Import des Entwickler-Zertifikates kein weiterer Aufwand an.

Als nachteilig muß generell eingeschätzt werden, daß mit Signtool signierte Java Applets nur im Netscape Navigator ausgeführt werden können. Der Netscape Navigator unterstützt bis zur V.4.05 noch nicht den JDK 1.1-Standard. Dadurch kann es zu Problemen bei der Ausführung von Java Applets kommen, die neue JDK 1.1-Klassen nutzen (z.B. beim Drucken). Zur Behebung dieser Einschränkung kann entweder der JDK 1.1-Patch von Netscape ab der V.4.03 des Netscape Navigator installiert bzw. direkt mit dem Netscape Navigator V.4.06 und höher gearbeitet werden.

Bei der Verwendung von Signtool stellt sich für den Entwickler das Problem, das er seinen Quellcode an die Klassen des Netscape Capabilities API /11/ anpassen muß. Erst der Einsatz dieser Klassen erlaubt die Anforderung der benötigten Zugriffsrechte an den Nutzer.

SDK (Microsoft)

Dieses Werkzeug stellt bzgl. des Signierens von Java Applets das Gegenstück zu Signtool dar. Eine Dokumentation zur Handhabung des Signierens mit SDK ist unter /12/ zu finden.

Vorteilhaft ist, wie auch bei Signtool, daß der Nutzer nur das Entwickler-Zertifikat importieren muß. Bei der Einstellung der Zugriffsrechte unterscheidet sich die Anwendung von SDK gegenüber Signtool aufgrund der unterschiedlichen Sicherheitskonzepte der Browser. Hier muß der Entwickler beim Signieren eines Java Applets die von ihm geforderte Sicherheitsklasse angeben. Diese Anforderung kann dann vom Nutzer nur akzeptiert oder abgewiesen werden.

Nachteilig ist wiederum, daß mit SDK signierte Java Applets nur im MS Internet Explorer ausgeführt werden können. Weiterhin unterstützt der MS Internet Explorer V.4 bisher nicht den JDK 1.1-Standard. Es wird auch kein entsprechendes Plug-In angeboten. Laut Ankündigung von Microsoft soll der JDK 1.1-Standard in einer der nächsten Versionen unterstützt werden.

Javakey (Sun)

Javakey wird als Werkzeug zum Signieren von Java Applets von Sun angeboten. Eine ausführliche Dokumentation ist z.B. unter /13/ zu finden.

Mit Javakey signierte Java Applets können nur beim Vorhandensein des Java Plug-In von Sun /14/ sowohl im Internet Explorer als auch im Netscape Browser ausgeführt werden. Das Plug-In muß auf jedem Rechner, der mit so signierten Applets arbeiten will, installiert werden.

Vorteilhaft an dieser Variante ist der unproblematische Ablauf signierter Java Applets sowohl unter dem MS Internet Explorer als auch unter dem Netscape Navigator. Die Applets müssen nur einmal signiert werden und es existiert nur ein Zertifikat je Entwickler. Ebenfalls von Vorteil ist, daß der Quelltext für die Verwendung im Netscape Navigator nicht verändert werden muß. Das Java Plug-In unterstützt vollständig den JDK 1.1-Standard.

Nachteilig ist die notwendige Installation des Plug-In zusätzlich zum Import der Zertifikate. Durch den Einsatz des Java Plug-In werden auch die Sicherheitsmechanismen der Browser übersprungen. Somit besitzen signierte Java Applets, deren Zertifikat importiert wurde, in jedem Fall alle Zugriffsrechte. Der Nutzer hat keinerlei Einfluß mehr auf die Zuteilung von Rechten.

3. Signierte Applets im GLOBUS-Umfeld

Im letzten Abschnitt wurden die verschiedenen technischen Varianten zur Signierung von Java Applets beschrieben. In Absprache mit den anderen Entwicklungsinstituten (FZI, FZK, IPF) und den betroffenen Behörden (UVM, LfU, IM/StaV) wird aufgrund des geringeren Aufwandes für den Nutzer die parallele Verwendung von Signtool und SDK favorisiert. Somit werden die Browser Netscape Navigator und MS Internet Explorer bedient.

Die Beschaffung der Zertifikate erfolgt über die Selbst-Signierung mit den Werkzeugen Signtool und SDK, da diese Lösung kostengünstig ist und kein Sicherheitsverlust entsteht. Da selbstsignierte Zertifikate nicht nur beim Entwickler, sondern auch an zentraler Stelle generiert werden können, ist die sichere Herkunft der Zertifikate durchaus gewährleistet. Diese Zertifikate müssen dann den Nutzern möglichst zentral (quasi als privates Trust-Center) bekannt gemacht und mit dem privaten Schlüssel dem jeweiligen Entwickler übergeben werden. Diese Bekanntmachung kann mittels Versendung der Zertifikate per E-Mail, Bereitstellung auf einer HTML-Seite etc. durchgeführt werden. Es ist ebenso möglich, daß nur die zentrale Stelle ein Entwickler-Zertifikat besitzt, welches den Nutzern bekannt gemacht wird. Nur diese Stelle ist dann privilegiert, die Java Applets der zur Organisation gehörenden Entwickler zu signieren.

Für die Anwendung der beiden Werkzeuge Signtool und SDK wurde eine Anleitung erstellt, die weiterhin Vorschläge für die Organisation eines privaten Trust-Centers enthält.

4. Ausblick

Beim bisherigen Stand der Untersuchungen muß festgestellt werden, daß keine der gefundenen technischen Lösungen ohne Einschränkungen in der Praxis eingesetzt werden kann. Besondere Probleme bereiten die unterschiedlichen Vorgehensweisen der Standardbrowser Netscape Navigator und MS Internet Explorer. Da die Entwicklung der Techniken des Signierens noch am Anfang steht, ist die weitere Beobachtung des Marktes von hoher Bedeutung.

Im Rahmen des GLOBUS-Umfeldes wird die Variante der parallelen Verwendung von SDK und Signtool für das Signieren von Java Applets in der Praxis eingesetzt. Da der Entwickler in diesem Fall die Klassen des Netscape Capabilities API verwenden muß, steht die Ent-

wicklung integrativer Java-Klassen zur Diskussion. Diese Klassen sollen die Anforderungen von Zugriffsrechten auf lokale Ressourcen browserunabhängig ermöglichen und so den Entwickler von zusätzlichem Programmieraufwand entlasten.

Neben der Signierung von Java Applets werden in Zukunft die Möglichkeiten für den Electronic Commerce im Umweltbereich einen Kernpunkt einnehmen. Dabei geht es u.a. um die Entwicklung von Konzepten für die Verbuchung von Kleinstbeträgen für angebotene Informationsdienste von Behörden. Einen weiteren Punkt stellt in diesem Zusammenhang die Behandlung von Abonnements von Umweltinformationen dar. Hier sind sichere und praktische Lösungen für Anbieter und Kunden gefragt.

5. Literatur

- /1/ Mayer-Föll, R.; Jaeschke, A. (Hrsg.); 1997: UIS Baden-Württemberg, Projekt GLOBUS - Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg, Phase IV, Forschungszentrum Karlsruhe, Wissenschaftliche Berichte, FZKA 6000.
- /2/ Wiesel, J.; Hofmann, C.; 1996: Architektur eines GIS-Terminal zur Visualisierung von Geodaten, <http://www-ipf.bau-verm.uni-karlsruhe.de> 1996
- /3/ Schneier, B.; 1996: Angewandte Kryptographie, Addison-Wesley Company.
- /4/ Garfinkel, S.; Spafford, G.; 1997: Web Security & Commerce, O'Reilly & Associates, Inc..
- /5/ von Appen, C.; 1998: Der kleine Java Sicherheits-Workshop, <http://www.abim.net/jsw/index.htm#Inhalt>.
- /6/ Griscom, D.; 1998: Code Signing for Java Applets, http://www.suitable.com/Doc_CodeSigning.shtml.
- /7/ <http://www.verisign.com/>.
- /8/ <http://www.thawte.com/>.
- /9/ <http://home.netscape.com/servers/pricing/index.html>.
- /10/ <http://developer1.netscape.com:80/software/signedobj/jarpack.html#signtool1.1>.
- /11/ <http://developer.netscape.com/docs/manuals/signedobj/capabilities/contents.html>.
- /12/ <http://www.eu.microsoft.com/workshop/security/authcode/signing.htm>.
- /13/ <http://www.java.sun.com/products/jdk/1.1/docs/tooldocs/solaris/javakey.html>.
- /14/ <http://java.sun.com/products/plugin/1.1.1/docs/signed.html>.

Von AlfaWeb zur XfaWeb-Systemfamilie

*R. Weidemann, W. Geiger, N. Peter, M. Reißfelder, G. Zilly,
Forschungszentrum Karlsruhe GmbH,
Institut für Angewandte Informatik (IAI),
Postfach 3640,
76021 Karlsruhe*

1. EINLEITUNG	73
2. DIE XFAWEB-SYSTEMFAMILIE	73
2.1 ALFAWEB.....	74
2.1.1 <i>Inhaltlicher und funktionaler Ausbau</i>	74
2.1.2 <i>CD-ROM-Version</i>	78
2.1.2.1 Motivation.....	78
2.1.2.2 Anforderungen an das Steuerelement	78
2.1.2.3 Analyse	78
2.1.2.3.1 Use Case Diagramm	78
2.1.2.3.2 Klassendiagramm.....	79
2.1.2.3.3 Kollaborationsdiagramm.....	81
2.1.2.3.4 Aktivitätsdiagramm	82
2.1.2.4 Sicherheitsaspekt.....	83
2.1.2.5 Realisierung	83
2.1.2.6 Auswertung und Ausblick.....	85
2.1.2.6.1 Aktueller Stand	85
2.1.2.6.2 Zukünftige Erweiterungen	85
2.1.3 <i>Schaufenster-Version</i>	86
2.2 NAFAWEB	87
2.3 BOFAWEB	89
3. WERKZEUG-ENTWICKLUNG.....	90
3.1 SYSTEMVERWALTUNGSKOMPONENTE	90
3.1.1 <i>Datenbankaufbau</i>	91
3.1.2 <i>Realisierung</i>	91
3.1.3 <i>Benutzerschnittstelle</i>	92
3.1.4 <i>Fazit</i>	95
3.2 ELEKTRONISCHE ZAHLUNGS- UND ABRECHNUNGSSYSTEME	96
3.2.1 <i>Systemeigenschaften eines Elektronischen Zahlungssystems</i>	96
3.2.2 <i>Übersicht derzeit verfügbarer Systeme</i>	97
3.2.2.1 Zahlungssysteme.....	97
3.2.2.1.2 Kreditkartenzahlung über das Internet	97
3.2.2.1.3 Digitales Bargeld	97
3.2.2.1.3 Chipkarten-Zahlungssystem.....	98
3.2.2.2 Abrechnungssysteme.....	98
3.2.2.2.1 Abrechnung über Kundenkonten.....	98
3.2.2.2.2 Jahresabonnement.....	99
3.2.3 <i>Vorauswahl für XfaWeb</i>	99
3.2.3.1 Anforderungsanalyse	99
3.2.3.2 Realisierungsvorschläge	100
3.2.3.2.1 Jahresabonnement.....	100
3.2.3.2.2 Kundenkonto	100
3.2.3.2.3 Chipkarten	100
3.2.3.2.4 Werbung.....	101
3.3 SONSTIGE KOMPONENTEN	101
3.3.1 <i>Hilfsmittel zur Berichterstellung</i>	101
3.3.2 <i>Konvertierungsprogramm</i>	101
3.3.3 <i>Volltextsuche</i>	102
4. AUSBLICK	103
5. LITERATUR.....	104

1. Einleitung

Im Rahmen der früheren GLOBUS-Entwicklungsphasen wurde in mehreren Stufen das Altlasten-Fachinformationssystem AlfaWeb entwickelt und inhaltlich ausgebaut /1/, /2/, /3/. Ziel des Vorhabens 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.

Anfang 1998 konnte der inhaltliche Ausbau abgeschlossen werden, d.h. alle vorhandenen relevanten Berichte waren in AlfaWeb eingestellt, sodaß eine Aktualisierung nur beim Erscheinen neuer bzw. überarbeiteter Berichte erforderlich wird. Das vollständige System steht jetzt in Form einer CD-ROM zur Verfügung (siehe Kap. 2.1.2). Die Internet-Version von AlfaWeb wurde inhaltlich reduziert (Schaufenster-Version, siehe Kap. 2.1.3), um den Absatz der CD-ROM nicht zu beeinträchtigen. Um mittelfristig die vollständige Version wieder - dann aber kostenpflichtig - im Internet anbieten zu können, wurde eine Studie angefertigt, welche die derzeit im Internet einsetzbaren Abrechnungs- und Zahlungssysteme vergleicht und auf Ihre Verwendbarkeit bei AlfaWeb untersucht (s. Kap. 3.2).

Parallel dazu wurden die zur Verwaltung und Pflege der Datenbasis (Berichtstexte, Schlagworte, Querverweise etc.) notwendigen Software-Werkzeuge, insbesondere die Systemverwaltungskomponente (s. Kap. 3.2), weiterentwickelt. Funktionale Ausgestaltung und formularorientierte Benutzeroberfläche erlauben es nun, daß die Pflege der Datenbasis direkt durch den Betreiber vorgenommen werden kann. In Anlehnung an AlfaWeb wurde mit dem Aufbau einer Systemfamilie (**XfaWeb**) begonnen. Ziel ist die Nutzung der vorhandenen Werkzeuge und Erfahrungen in anderen Anwendungsgebieten. Die ersten Neuentwicklungen sind **NafaWeb** (s. Kap. 2.2), das im Bereich Naturschutz, und **BofaWeb** (s. Kap. 2.3), das im Bereich Bodenschutz angesiedelt ist. In beiden Fällen wurde ein erster Prototyp entwickelt.

2. Die XfaWeb-Systemfamilie

Die Bezeichnung "XfaWeb-Systemfamilie"¹ steht bisher als begriffliche Klammer, um die Zusammengehörigkeit und die gemeinsame, synergetische Weiterentwicklung der Fachinformationssysteme zu dokumentieren. Die Mitglieder der Familie verfügen über eine einheitlich gestaltete Benutzeroberfläche und basieren auf einem gemeinsamen Satz von Werkzeugen und Funktionen. Eine weitergehende Integration der Einzelsysteme ist vorgesehen (siehe Kapitel 4).

¹ Das "X" steht als Platzhalter (Wildcard) für das Anwendungsgebiet, "fa" ist die Abkürzung für Fachinformationssystem oder Fachinformationen.

2.1 AlfaWeb

AlfaWeb ist ein Projekt des Ministeriums für Umwelt und Verkehr des Landes Baden-Württemberg. Projektentwicklungsstelle ist das Referat 44 (Altlasten) der Landesanstalt für Umweltschutz. Das System wird entwickelt vom Forschungszentrum Karlsruhe, Institut für Angewandte Informatik, der IUT Ingenieurgesellschaft für Umwelttechnik mbH (Kirchzarten) und der AHG Gesellschaft für Angewandte Hydrologie und Kartographie (Freiburg).

2.1.1 Inhaltlicher und funktionaler Ausbau

Während zum Ende der letzten GLOBUS-Phase (1997) bereits alle relevanten Materialien der Landesanstalt für Umweltschutz (LfU) in AlfaWeb aufgenommen waren /3/, konzentrierte sich der inhaltliche Ausbau anschließend darauf, weitere bei der Altlastenbearbeitung benötigte Texte, insbesondere Gesetze, Verordnungen und andere rechtliche Regelungen in das System einzubringen. Tabelle 1 zeigt summarisch den derzeitigen Umfang von AlfaWeb.

Berichtsreihen	Ist-Stand 23.11.95	Ist-Stand 07.11.96	Ist-Stand 31.10.97	Ist-Stand 31.10.98
Materialien zur Altlastenbearbeitung	4	21	25	25
Texte und Berichte zur Altlastenbearbeitung	4	18	29	31
Sonstige Berichte	2	3	9	29
Summe:	10	42	63	85

Tab.1: Fachberichte und andere Dokumente in AlfaWeb

Neu in AlfaWeb aufgenommen wurde ein Firmenverzeichnis. Dieses wurde u.a. auch deshalb realisiert, um eine zusätzliche Finanzierungsquelle für die CD-ROM-Version (s. Kap. 2.1.2) zu erschließen. Das Verzeichnis enthält Adressen von Firmen, die im Altlastenbereich tätig sind (Bau-, Bohr-, Entsorgungs- und Sanierungsfirmen, Labors und Ingenieurbüros). Einträge in die Datenbank sind kostenpflichtig. Bis zum Erscheinen der CD-ROM konnten ca. 150 Firmeneinträge verkauft werden.

Zugriffe auf das Firmenverzeichnis erfolgen über ein einfaches Formular, indem der Benutzer die gewünschte Branche und/oder einen Postleitzahlenbereich auswählt. Abbildung 1 zeigt eine Abfrage nach Sanierungsfirmen im Postleitzahlenbereich 76.



Abb.1: Beispiel einer Abfrage im Firmenverzeichnis

Als weitere inhaltliche Ergänzungen wurde folgendes zur Verfügung gestellt:

- Excel-Programm Immissions-/Emissionsbetrachtung:
Mit Hilfe des Programms kann überprüft werden, ob der aufgrund von Erkundungen vorgefundene Zustand eines Schadensfalles die einzelfallbezogene Mindestanforderung der "Verwaltungsvorschrift über Orientierungswerte für die Bearbeitung von Altlasten und Schadensfällen" einhält.
- Formulare als Microsoft Word Dokumentvorlagen:
Neben einem Probenahmeprotokoll sind Förderantrag-, Zuwendungsbescheid- und Verwendungsnachweis-Formulare vorhanden.
- Druckvorlagen (PDF-Dateien):
Damit die in AlfaWeb enthaltenen Berichte (nicht die Gesetzestexte) auch in vorgegebener Form ausgedruckt werden können, sind diese auch als PDF-Dateien verfügbar.

Nachdem der inhaltliche Ausbau weitgehend abgeschlossen war, konnte der Fachzugang realisiert werden. Der Fachzugang zu AlfaWeb geht im Gegensatz zu den anderen Zugängen, die mehr formale Eigenschaften auswerten (Berichtestruktur, alphabetischer Index) von inhaltlichen Gesichtspunkten, d.h. typischen Vorgängen und Fragestellungen bei der Altla-

stenbearbeitung aus. Von Fachexperten wurden für die einzelnen Tätigkeiten die wichtigsten Informationen und Hilfsmittel zusammengestellt. Es werden parallel drei Gliederungsstrukturen angeboten:

- Systematische Vorgehensweise (gegliedert nach den Vorgängen bei der Altlastenbearbeitung)
- Recht (rechtliche Fragestellungen)
- Methoden und Techniken (bei der Altlastenbearbeitung eingesetzte Verfahren)

Alle drei Gliederungsstrukturen besitzen einen hierarchischen Aufbau. Beginnend mit der jeweiligen Startseite werden auf jeder Fachzugangs-Seite Menüpunkte zur Auswahl angeboten. Die Auswahl eines Menüpunktes blättert die nächste Detaillierungsstufe zu diesem Punkt auf (falls vorhanden) oder führt direkt zu einer Informationsseite (i.d.R. ein Kapitel eines Berichts aus AlfaWeb). Die einzelnen Fachzugangs-Seiten enthalten entweder ein Auswahlmenü oder ein Flußdiagramm. Die nächste Detaillierungsstufe erreicht man jeweils über die Auswahl eines Menüpunktes im Auswahlmenü bzw. die Auswahl eines maus-sensitiven Bereichs im Diagramm. Abb.2 zeigt als Beispiel die Startseite des Zweigs "Systematische Vorgehensweise".

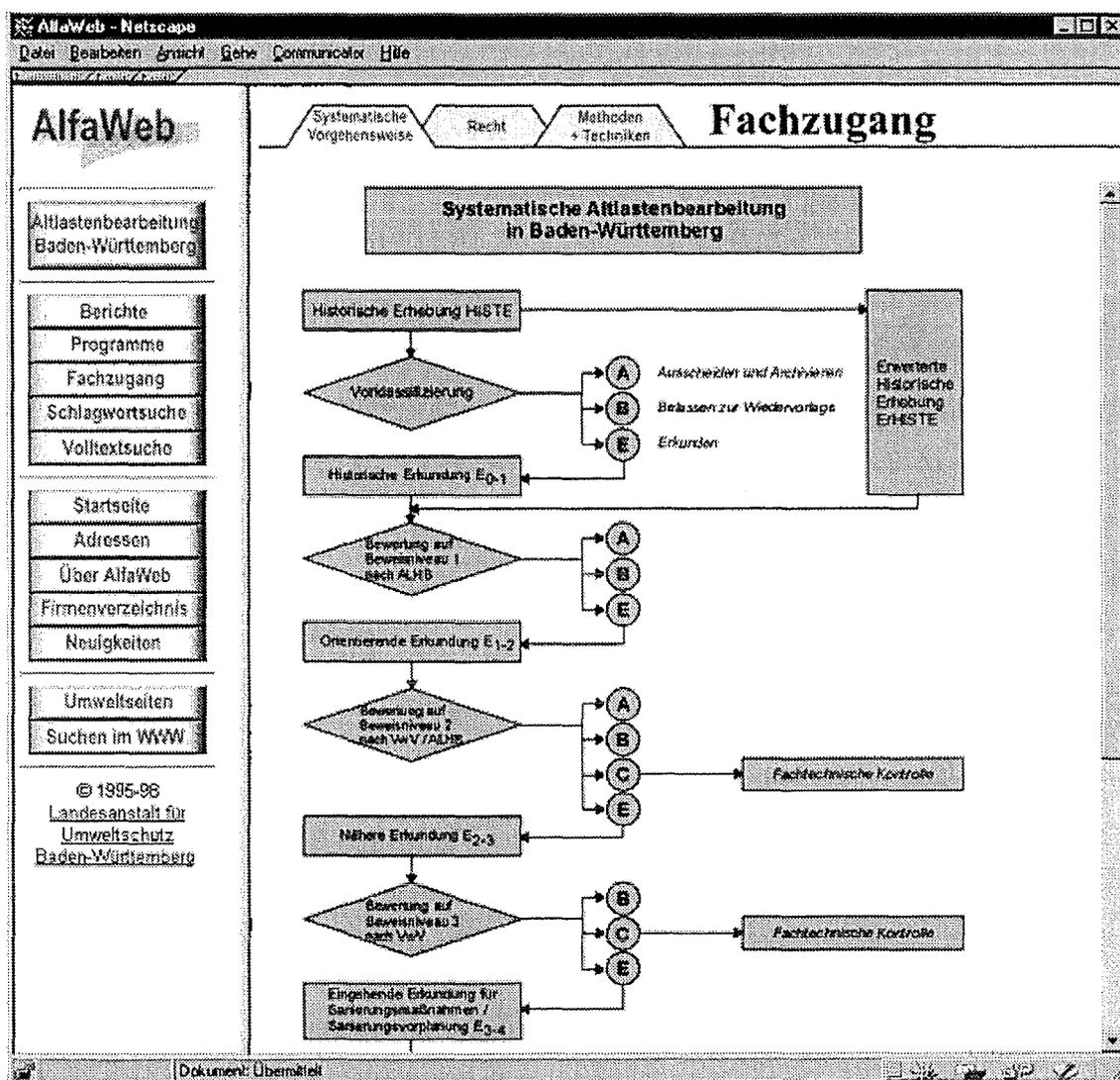


Abb.2: Fachzugang in AlfaWeb (Beispielseite)

Wie bereits in der Einleitung erwähnt, wurde eine Diversifizierung von AlfaWeb in verschiedene Versionen vorgenommen. Unterschiede gibt es:

- in Inhalt und Funktionalität (Vollversion <-> Schaufensterversion)
- in der Realisierung (WWW-Version <-> CD-ROM-Version)
- in der Erreichbarkeit (Internet <-> Intranet LfU <-> lokal (CD-ROM)) und
- in der Laufzeitumgebung (Windows <-> Unix)

Die Aufspaltung in verschiedene Versionen hat ihren Grund zum einen in der Tatsache, daß nicht alle potentiellen Nutzer auf einheitliche Weise erreicht werden können (daher WWW und CD-ROM) und zum anderen in dem Zwang der Verwaltung, für ihre Dienstleistungen Gebühren zu erheben (daher Vollversion und Schaufenster-Version). Die weiteren Unterschiede sind technischer Natur und folgen aus den zuerst genannten Gründe.

Basis aller Versionen ist das vollständige, WWW-basierte System, das unter Windows NT 4.0 (Server) und dem Microsoft Internet Information Server 4.0 läuft. Soweit möglich wurden Hilfswerkzeuge geschaffen bzw. Anleitungen erstellt, welche es erlauben, die anderen Versionen aus der Basis-Version zu erzeugen. Durch die nicht automatisch abgleichbaren Teile verbleibt jedoch ein nicht unerheblicher Aufwand zur Pflege der verschiedenen Versionen.

Die folgende Tabelle gibt einen Überblick über die derzeit eingesetzten Versionen:

Version	Vollversion	Schaufensterv.	Internet	Intranet (LfU)	CD-ROM	Windows	Unix	Bemerkungen
Entwicklungs-Version 1	X		X			X		nur für Entwickler und Projektsteuerungsgruppe
Entwicklungs-Version 2		X	X				X	
Intranet-Version 1	X			X		X		
Intranet-Version 2		X		X			X	
Offizielle Version (UVM)		X	X				X	
CD-ROM-Version	X				X	X		

Tab.2: Übersicht derzeitige AlfaWeb-Versionen

Im Abschlußbericht zu GLOBUS Phase IV /3/ wurden Alternativen untersucht, ein WWW-basiertes System einschließlich dynamischer Seiten auf CD-ROM anzubieten. Nachdem nun seit Juli 1998 die fertige CD-ROM-Version ausgeliefert wird, kann hier über die letztlich ge-

wählte Realisierungsmethode berichtet werden (s. Kap. 2.1.2). Kapitel 2.1.3 stellt anschließend die Schaufenster-Version von AlfaWeb kurz vor.

2.1.2 CD-ROM-Version

2.1.2.1 Motivation

Im Rahmen von GLOBUS Phase IV /3/ wurde in zwei Studien untersucht, welche Möglichkeiten für eine Offline-Nutzung durch HTML-Browser und im besonderen durch die bekannten WWW-Browser existieren. Problematisch war hierbei nicht die Darstellung der HTML-Dokumente, sondern die Unterstützung der dynamischen Komponenten, wie z.B. die Auswertung von Formularen, da der Web-Server im Offline-Betrieb nicht zugänglich ist. Ergebnis dieser Untersuchungen war, daß die beiden Standardbrowser Netscape und Microsoft Internet Explorer zwar unterstützt werden können, jedoch waren zum Zeitpunkt der Erstellung der Studie die Lösungsmöglichkeiten nicht kompatibel, so daß sich die Projektsteuerungsgruppe AlfaWeb für die Unterstützung des damals einzigen kostenfrei kopierbaren Web-Browsers des Internet Explorer von Microsoft entschied /3/, /8/.

2.1.2.2 Anforderungen an das Steuerelement

Da der zentrale Web-Server bei der CD-ROM-Lösung nicht verwendet werden kann, muß die bisherige CGI-Funktionalität auch lokal bereitgestellt werden. Hierzu müssen folgende Punkte gewährleistet werden:

- Unterstützung der GET -Methode von HTTP
- Unterstützung des MIME-Typen TEXT/HTML
- Darstellung der Ergebnisse in beliebigen Frames oder Browser-Fenstern durch Unterstützung des TARGET-Attributs
- Ergebnisse sollen vorübergehend nachgehalten werden, so daß die History-Liste des Browsers für den Verlauf der Sitzung unterstützt wird.
- Vermeidung eines "Datenfriedhofs" mit der möglichen Folge eines Plattenüberlaufs.
- Befriedigende Zugriffszeiten, jedenfalls nicht schlechter als Netzversion.
- Erhalt der Sicherheit
- Geringstmögliche Unterschiede zur Web-Version, so daß die CD-ROM-Version jeweils aus der Web-Version möglichst automatisch erzeugt werden kann.
- Die CD soll mindestens für Windows 95 / NT 4.0 verfügbar sein.

2.1.2.3 Analyse

2.1.2.3.1 Use Case Diagramm

Die beiden Diagramme in Abb.3 zeigen die denkbaren Anwendungsfälle bei der Nutzung der Web-Variante und bei der Nutzung der CD-ROM. Hierbei soll die Gesamtanwendung aus Benutzersicht identische Ergebnisse liefern. Für statische Seiten ist der Anwendungsfall ebenso wie dessen Realisierung identisch. Bei den dynamischen Seiten nutzt der Anwender mit Web-Anschluß die herkömmliche Client-Server-Architektur der Web-Technologie. Der Anwender der CD-ROM benötigt jedoch einen alternativen Ansatz. Die dynamischen Seiten

können in der CD-ROM-Version nur durch JavaScript-Funktionen sowie durch Erweiterungen des Web-Browsers, die über JavaScript mit dem Browser kommunizieren, erzeugt werden. Hier ist stellvertretend die ActiveX-Technologie zu erwähnen, die bei der Realisierung der AlfaWeb-CD-ROM verwendet wurde. Alternativen sind die Netscape Plug-in-Technologie und die LiveConnect-Architektur.

Als Sonderfall muß berücksichtigt werden, daß der Anwender der CD-ROM einen Internetzugang besitzen kann, was vor allem Auswirkungen auf die Überlegungen zur Sicherheit der Lösungen zur Folge hat.

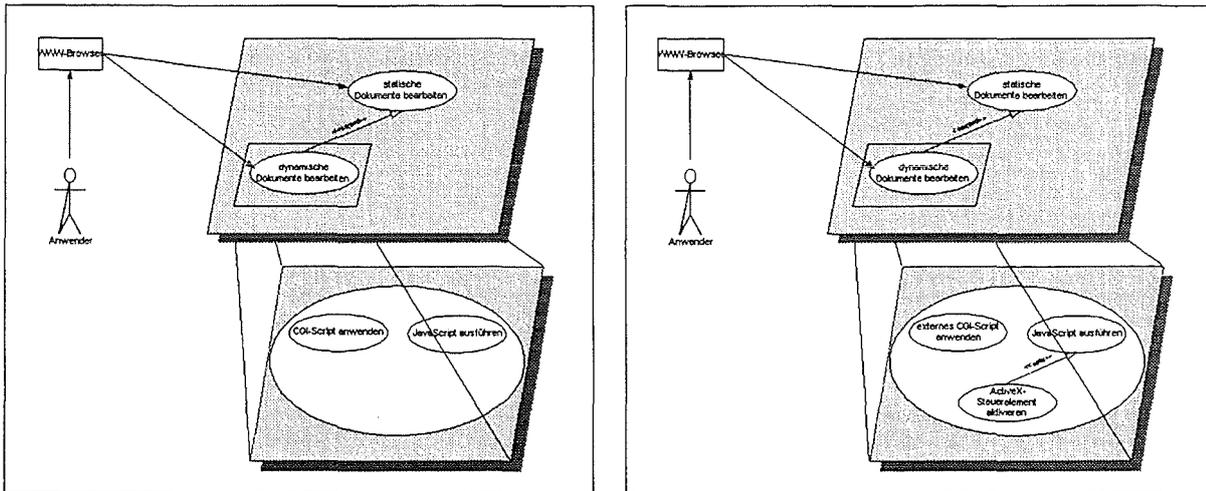


Abb.3: Gegenüberstellung der Anwendungsfälle bei der Web- (links) und der CD-ROM-Lösung (rechts)

2.1.2.3.2 Klassendiagramm

Das im Klassendiagramm aufgezeigte ideale Modell zeigt auf, wie das ActiveX-Steuerelement als zentrale Komponente der lokalen CGI-Schnittstelle arbeitet. Dieses Steuerelement wird in ein HTML-Dokument eingebettet. Es besitzt Methoden zur Prüfung der Lizenz und zur Ermittlung des angeforderten Skripts. Das Steuerelement muß dieses Skript synchron aufrufen und das durch dieses Skript erzeugte Ergebnis an den Browser mitteilen. Zusätzlich ist dieses Steuerelement in der Lage, bei Fehlern einen entsprechenden Fehler-text zu ermitteln und diesen an den Browser mitzuteilen.

Im idealen Fall kann dieses Steuerelement mehrere Anwendungen unterstützen, wobei in jeder Anwendung jedes Skript nur einmal registriert werden kann. Jedoch ist es denkbar, daß ein Skript in mehreren Anwendungen enthalten sein kann.

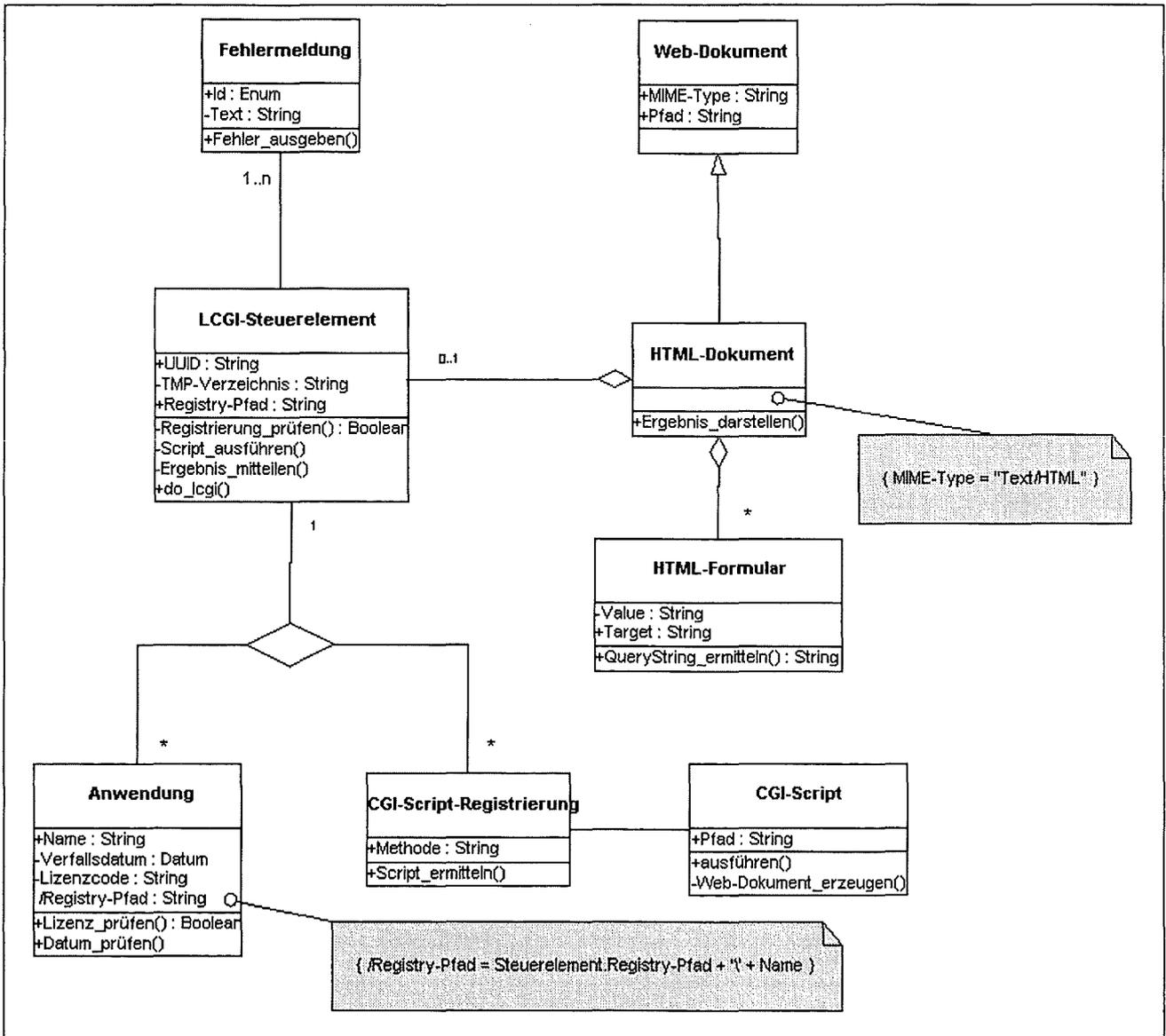


Abb.4: logische Sicht auf die lokale CGI-Schnittstelle (UML²)

² Unified Modeling Language, siehe z.B. <http://www.oose.de/uml/>

2.1.2.3.3 Kollaborationsdiagramm

Das nachfolgende Diagramm zeigt die Objekte und ihre Zusammenarbeit untereinander.

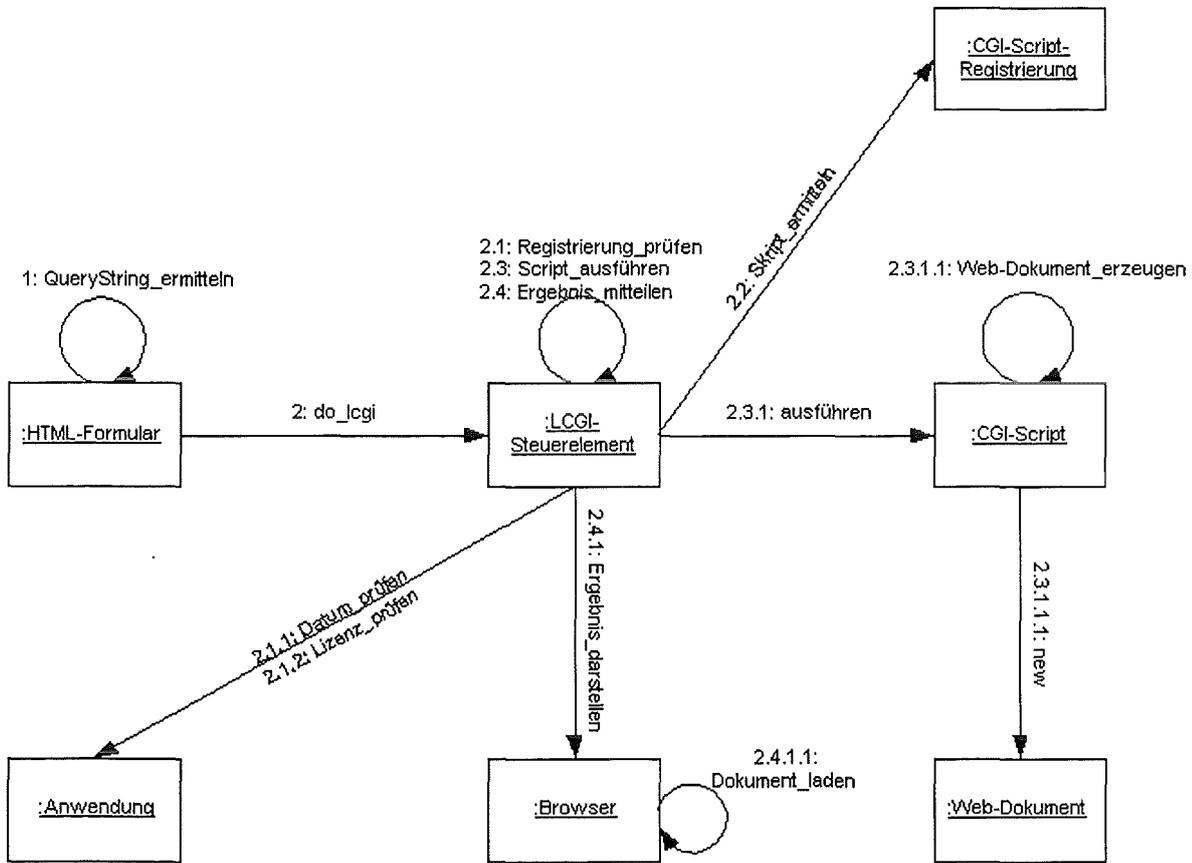


Abb.5: Kollaborationsdiagramm für die lokale CGI-Schnittstelle (UML)

2.1.2.3.4 Aktivitätsdiagramm

In dem nachfolgenden speziellen Zustandsdiagramm sollen die Aktivitäten und die davon ausgehenden Transitionen veranschaulicht werden.

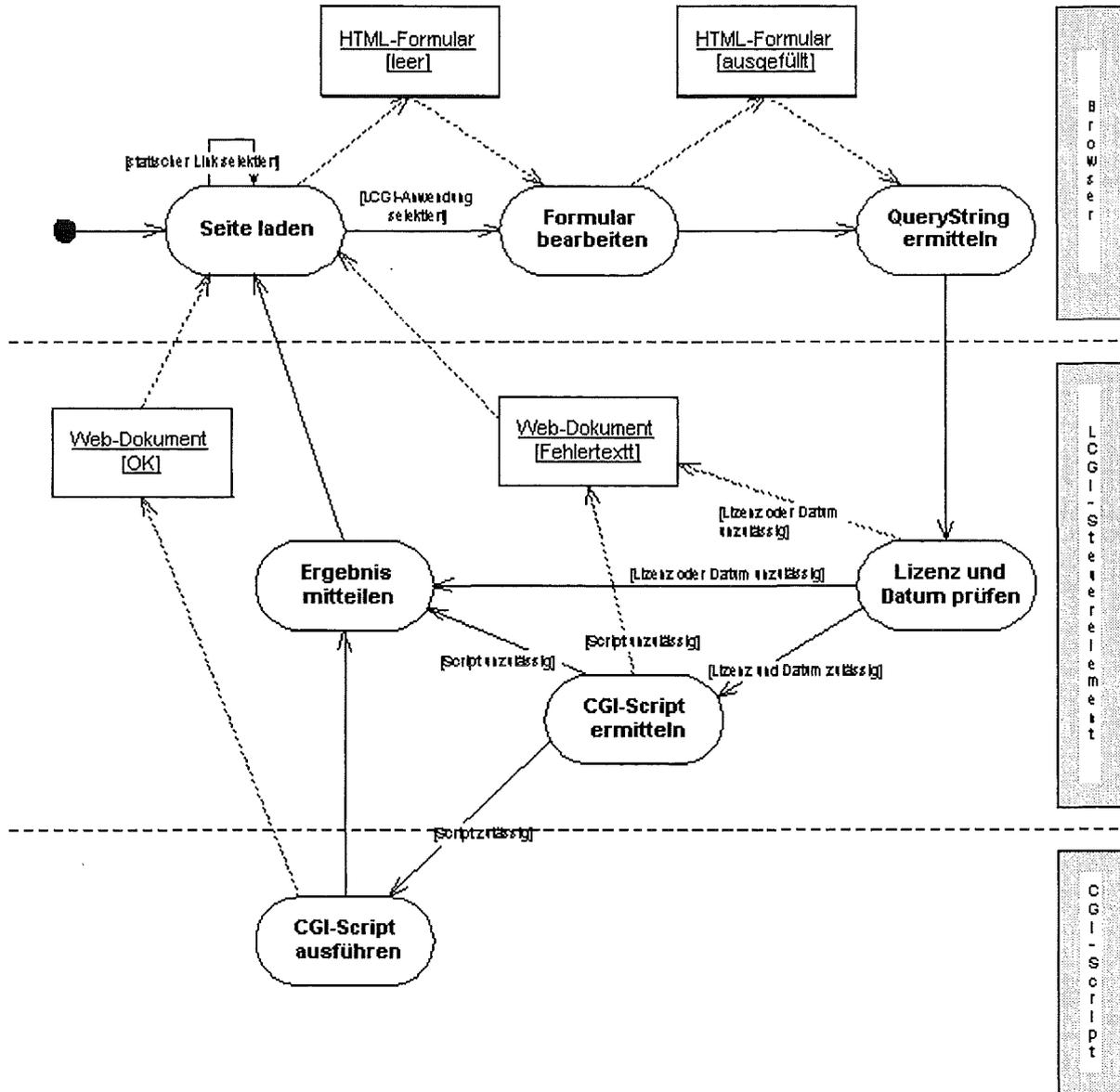


Abb.6: Aktivitätsdiagramm für die lokale CGI-Schnittstelle (UML)

2.1.2.4 Sicherheitsaspekt

Ein nicht zu vernachlässigender Aspekt bei der Entwicklung der CD-ROM ist die Sicherheit. Hier ist einerseits die Sicherheit der Lösung gegenüber Manipulationen in der ausgelieferten Software zu verstehen, andererseits aber auch die Gewährleistung, daß der Anwender keine Sicherheitsmechanismen wegen der ausgelieferten Anwendung aufgeben muß. Die Sicherheit der Implementierung beruht auf folgendem Ansatz:

- Verwendung eines Lizenzcodes, der vor Manipulationen, aber auch vor unberechtigter Verwendung in fremden Anwendungen schützt
- Registrierung aller für die Anwendung verwendeten CGI-Skripts. Diese sind im Lizenzcode verschlüsselt, so daß Manipulationen auf dieser Ebene ausgeschlossen werden können.
- Verwendung eines Verfallsdatums, um zu gewährleisten, daß mit gültigen Versionen gearbeitet wird (v.a. für die erfolgreich abgeschlossene, vorgeschaltete Testphase von Bedeutung).
- Vermeidung der Einrichtung eines Web-Servers, der über winsock.dll einen TCP/IP-Port belegen würde und weitere Sicherheitsvorkehrungen erfordern würde.

Sicherheitslücken werden dadurch verhindert, daß in der ausgelieferten Version die im Microsoft Internet Explorer fest vorgegebene Sicherheitseinstellung für lokale Dateien verwendet wird. Dadurch bleibt gewährleistet, daß der Anwender keine Sicherheitsoptionen in den vier Internet-Zonen freischalten muß. Der Anwender kann daher den Browser mit der höchsten Sicherheitsstufe weiterhin betreiben /8/.

2.1.2.5 Realisierung

Die CD-ROM basiert weitestgehend auf dem Web-Angebot. Einzig die Anpassung der dynamischen Seiten ist problematisch, da hier die Unterstützung der lokalen CGI-Schnittstelle durch ein ActiveX-Steuerelement vorgenommen wurde. Diesem ActiveX-Steuerelement liegt eine modular aufgebaute Funktionsbibliothek zu Grunde, die aus Basisfunktionen besteht, die von der eigentlichen Realisierung mit ActiveX unabhängig ist. Dadurch ist es bei Bedarf möglich, die ActiveX-Schnittstelle dieser Bibliothek durch ein Interface für die Plug-in-Technologie zu ersetzen. Die Lizenzierung geschieht in der Registrierungsdatenbank, die Registrierung der Anwendungen wird zur Zeit noch in einer Initialisierungsdatei vorgenommen.

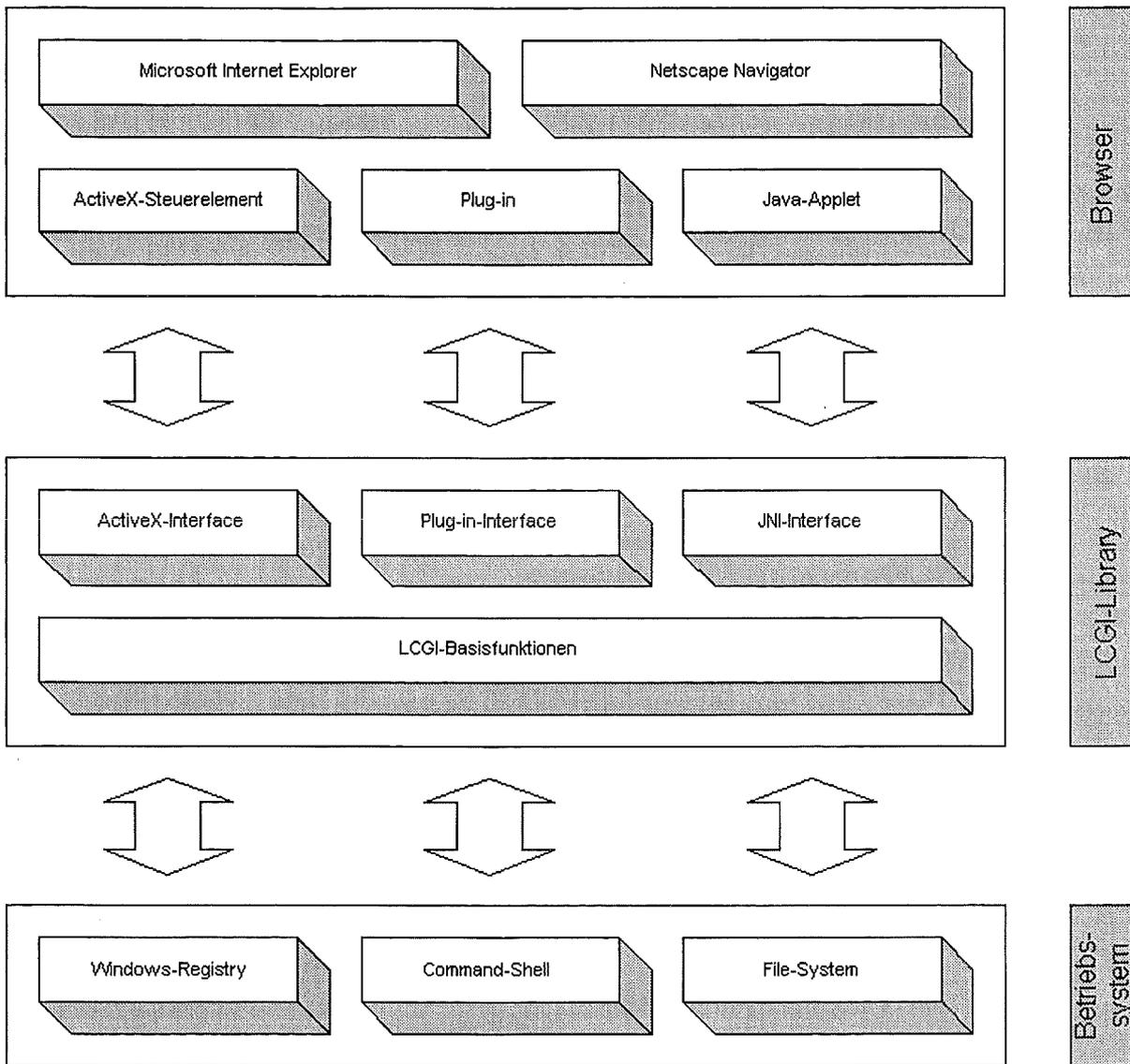


Abb.7: Schichtenstruktur der lokalen CGI-Schnittstelle

Um einen Datenfriedhof zu verhindern, werden nur 50 dynamisch erzeugte Ergebnisdokumente im Filesystem aufbewahrt. Werden weitere Dokumente erzeugt, so wird jeweils das älteste überschrieben. Im Verlauf einer Sitzung ist nicht zu erwarten, daß der Anwender mehr Aufrufe nacheinander vornimmt, auf die er über die History-Liste des Browsers wieder zurückgreifen will.

Die ausgelieferte Version läuft auf älteren Heimrechnern unter Windows 95 (z.B. 486DX4, 16 MB, CD-ROM-Laufwerk) ebenso wie unter Windows NT mit aktueller Standardausstattung, so daß ein sehr großer Personenkreis die technische Voraussetzung besitzt.

Die Anpassungen der Web-Variante für die CD-ROM-Anwendung wird durch zwei Skripts vorgenommen, die eine teilweise Automatisierung der Anpassungen erlauben:

- Skript zur Anpassung der meisten HTML-Formulare an die CD-ROM-Lösung
- Skript zur Anpassung statischer Links in den HTML-Seiten zur Erzeugung von Bookmark-Einträgen, die die jeweils aktuelle Konstellation eines Framesets sichern.

Eine automatische Anpassung der CGI-Skripts ist mit vertretbarem Aufwand nicht möglich, so daß der Entwickler die notwendigen Unterschiede der CD-ROM-Variante gegenüber der Web-Variante von Hand einarbeiten muß.

2.1.2.6 Auswertung und Ausblick

2.1.2.6.1 Aktueller Stand

Für die bereits in AlfaWeb in der Version 1.0 ausgelieferten CD-ROM wurde für die Realisierung des vorgestellten Modells ein ActiveX-Steuerelement implementiert. Dieses unterstützt zur Zeit genau die eingangs spezifizierten Anforderungen. Die Antwortzeiten erfüllten alle Anforderungen. Hingegen wurden bei der Entwicklung des Steuerelements einige Vereinfachungen gegenüber der im Klassenmodell vorgestellten Ideallösung vorgenommen. So unterstützt die Realisierung derzeit genau eine Anwendung, so daß sich der Lizenzcode und die Registrierung der Anwendungen als Problempunkt bei Tests für eine weitere Anwendung herausstellte. Weiterhin wurde die Unterstützung der POST-Methode ebenso wie die unterschiedlicher MIME-Types nicht berücksichtigt. In der ausgelieferten Version werden v.a. Perl-Skripts verwendet, die zur Zeit eine Installation des mitgelieferten Perl-Interpreters erfordern. Seit September 1998 ist mit dem neuesten Perl-Release eine Möglichkeit geboten, auch Runtime-Versionen der Skripts zu erzeugen, die diesen Skript-Interpreter nicht mehr benötigen und somit die Installation vereinfachen.

2.1.2.6.2 Zukünftige Erweiterungen

Aus den Erfahrungen der erfolgreichen Entwicklung und Einführung der ersten Version der AlfaWeb-CD konnten die Anforderungen an die nächste Version der CD-ROM-Lösung erarbeitet werden.

Anforderungen der Anwender:

- Unterstützung von CD-ROM-Servern
- Unterstützung von Windows 98 und dem kommenden Windows NT 5.0

Anforderungen für eine vereinfachte Umsetzung der Web-Variante:

- Bessere Unterstützung der Verarbeitung der Umlaute
- Direkte Unterstützung von "rekursiven", d.h. verschachtelten CGI-Aufrufen, z.B. in dynamisch erzeugten Framesets mit dynamisch erzeugten Inhalten der einzelnen Frames.
- Verwendbarkeit in anderen Anwendungen; hier muß die Lizenzierung flexibler gehandhabt werden.

Anforderungen, die mit der technischen Erweiterung der Web-Variante Hand in Hand gehen müssen:

- Zusätzliche Unterstützung der POST-Methode
- Unterstützung beliebiger MIME-Types

2.1.3 Schaufenster-Version

Bis Ende 1997 war die bis dahin noch prototypische Vollversion über den UVM-Server frei im Internet verfügbar. Mit der inhaltlichen Vervollständigung des Systems und dem Erscheinen der AlfaWeb-CD-ROM wurde diese Version durch eine Schaufenster-Version ersetzt³.

Die Schaufenster-Version (Demo-Version) von AlfaWeb bietet einen Überblick zu Inhalt, Funktionalität und Aussehen der Vollversion. Vollständig enthalten ist eine Einführung in die Altlastenbearbeitung in Baden-Württemberg. Aus der Veröffentlichungsliste sind Inhaltsverzeichnis und (meist) Kurzbeschreibungen der Handbücher und Fachberichte zur Altlastenbearbeitung abrufbar. Als Beispiel für ein integriertes Anwendungsprogramm wurde BOBAN (Datenbank Bodenbehandlungsanlagen) in die Schaufenster-Version aufgenommen. Bezugsinformationen zur CD-ROM-Version komplettieren das Angebot.

Der Systemrahmen (Abb.8 zeigt die Startseite des Systems) wurde aus der Vollversion abgeleitet und um spezifische Teile ergänzt (z.B. CD-ROM Bestellformular). Die sich ändernden Teile (Literaturliste, Inhaltsverzeichnisse, Kurzbeschreibungen, BOBAN-Datenblätter) können aus der Vollversion generiert werden. Das dazu entwickelte Programm wurde in die Systemverwaltungskomponente (s. Kap. 3.1) integriert.

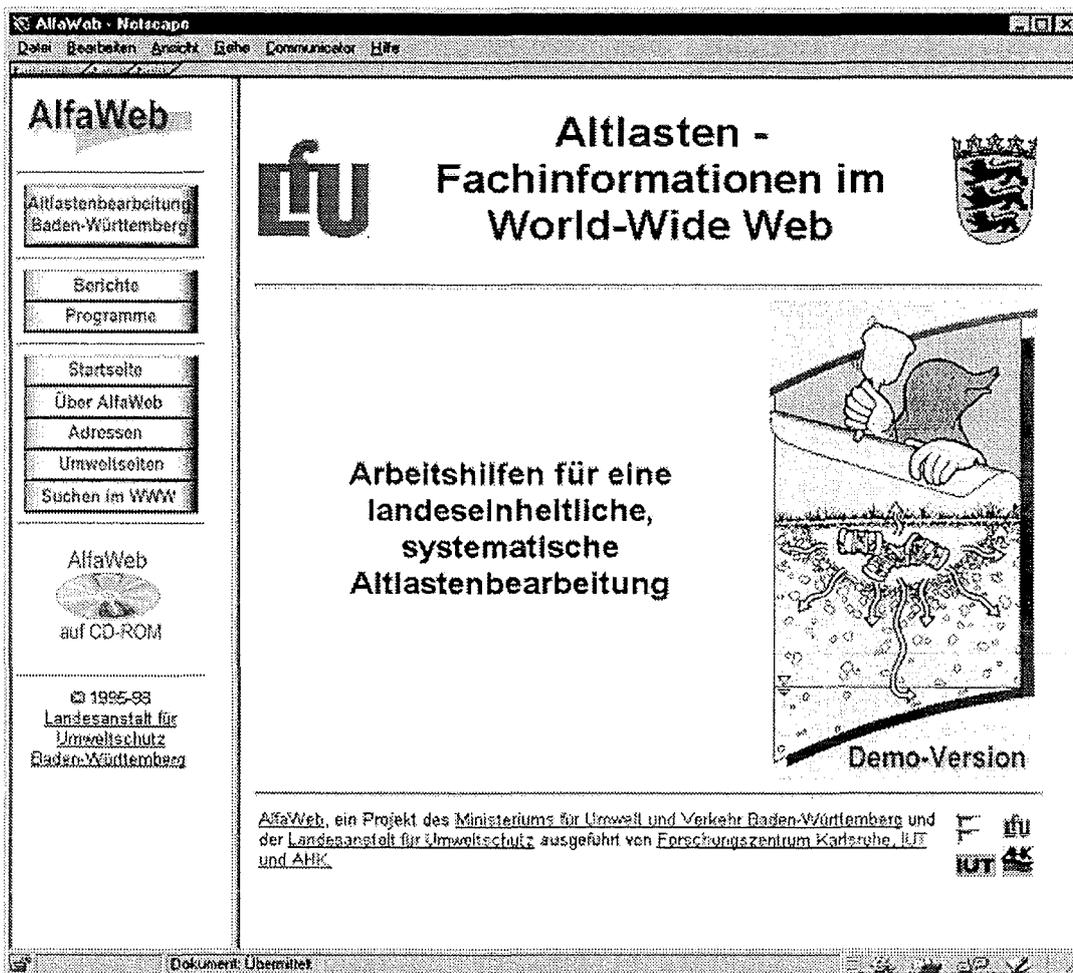


Abb.8: Startseite der AlfaWeb-Schaufrenster-Version

³ Als Folge davon gab es eine Reihe von Anfragen und Protesten von bisherigen Nutzern insbesondere aus dem universitären Bereich.

2.2 NafaWeb

Ziel des Naturschutz-Fachinformationssystems NafaWeb ist es, Arbeitshilfen für die Aufgaben in der Naturschutzpraxis (Handbücher, Fachberichte, Datenbanken und Programme) bereitzustellen, sowie durch geeignete Suchhilfen einen schnellen Zugang zu den gewünschten Informationen zu ermöglichen. Das System dient dazu, einen effektiven Vollzug in Naturschutz und Landschaftspflege zu unterstützen, insbesondere bei den unteren Naturschutzbehörden und den Naturschutzbeauftragten.

NafaWeb ist ein Projekt des Ministeriums Ländlicher Raum sowie des Ministeriums für Umwelt und Verkehr des Landes Baden-Württemberg. Projektentwicklungsstelle ist der Fachdienst Naturschutz an der Landesanstalt für Umweltschutz (Sachgebiet 24.2, Leitung: M. Theis). Das System wird entwickelt vom Forschungszentrum Karlsruhe, Institut für Angewandte Informatik, und der IUT Ingenieurgesellschaft für Umwelttechnik mbH in Kirchzarten. Zur fachlichen Begleitung des Projekts und frühzeitigen Einbeziehung potentieller Nutzer wurde von der LfU eine Projektsteuerungsgruppe mit Vertretern von verschiedenen späteren Anwendergruppen eingerichtet. Die Entwicklungsarbeiten wurden im Frühjahr 1998 begonnen, Konzept und Inhalt des Systems wurden auf drei Workshops eingehend diskutiert.

In der ersten Phase des Projekts bis November 1998 wurde ein Prototyp des Systems entwickelt, siehe Abbildung 9. Der Prototyp umfaßt allgemeine Informationen zum Naturschutz in Baden-Württemberg wie z.B. eine Übersicht über die Aufgaben des Fachdienstes Naturschutz und ein Verzeichnis von Fachstellen und der Beauftragten für Naturschutz. Er enthält sechs Naturschutz-Fachberichte, u.a. die Kartieranleitung für die §24a-Kartierung Baden-Württemberg und den Bericht "Arten, Biotope, Landschaft – Schlüssel zum Erfassen, Beschreiben, Bewerten". Daneben umfaßt das System weitere Dokumente wie die Roten Listen der Pflanzen und Tiere Baden-Württembergs sowie ein Glossar mit Begriffen aus der Landschaftsplanung. Außerdem stellt der Prototyp den Schlagwortkatalog der Literaturdatenbank ABSLIT der LfU mit über 9000 Schlagworten aus dem Arten- und Biotopschutz bereit.

An Such- und Navigationshilfen bietet das System den Benutzern zum einen einen Zugang über die Berichtsreihen im Naturschutz und die hierarchische Struktur der Berichte sowie eine Volltextsuche über sämtliche Dokumente des Systems. Weiter wurden die im System enthaltenen Berichte verschlagwortet und kann der Benutzer über eine Schlagwortsuche auf die Inhalte zugreifen. Außerdem wurde ein Prototyp des Fachzugangs entwickelt, über den der Benutzer entsprechend der fachlichen Gliederung des Anwendungsgebiets Naturschutz auf die Inhalte des Systems zugreifen können soll. Im Schlagwortkatalog der Datenbank ABSLIT können über Formulare Fachbegriffe gesucht und die ausgewählten Schlagworte für eine Literaturrecherche in der Volldatenbank direkt über Email an die LfU gesandt werden.



Abb. 9: Startseite von NafaWeb

Das Naturschutz-Fachinformationssystem NafaWeb wurde auf der Basis des Vorgehensmodells für die Dokumentenerstellung und der Werkzeuge des Altlasten-Fachinformationssystems AlfaWeb realisiert, insbesondere wurde auf dessen Konvertierungs- und Informationsaufbereitungstools sowie dessen Systemverwaltungskomponente aufgebaut. Diese Werkzeuge wurden verallgemeinert, um sie für verschiedene Fachinformationssysteme einsetzen zu können. Daneben wurden sie so weiterentwickelt, daß auch die zusätzlichen Anforderungen von Seiten NafaWebs erfüllt werden konnten; z.B. wurden hierarchische Berichtsreihen, das Umhängen von Berichten zwischen den Berichtsreihen und die Konvertierung einzelner Inhaltsverzeichnisse ermöglicht. Die Werkzeuge wurden vom Forschungszentrum Karlsruhe der IUT zur Verfügung gestellt und von dieser zur Aufbereitung der Dokumente gemäß XfaWeb-Richtlinie und Konvertierung in WWW-Formate genutzt.

Zum Testen des Systems wurde den Mitgliedern der Projektsteuerungsgruppe während der Entwicklungsphase der jeweilige Stand des Systems über ein WWW-Server des Forschungszentrums paßwortgeschützt zur Verfügung gestellt. Außerdem wurde am Ende der Projektlaufzeit eine erste CD-ROM-Version des Systems erstellt und an den genannten Kreis sowie einige weitere Interessierte zur Erprobung verteilt. Auf einem abschließenden Workshop wurden die Nutzererfahrungen im Hinblick auf die Weiterentwicklung des Systems gesammelt und ausgewertet.

2.3 BofaWeb

Das Projekt BofaWeb (Bodenschutz-Fachinformationen) wurde Mitte 1998 neu begonnen. Projektentwicklungsstelle ist das Referat 22 (Bodenschutz) in der Abteilung 2 (Ökologie, Boden- und Naturschutz) der Landesanstalt für Umweltschutz mit Beteiligung des Informationstechnischen Zentrums (ITZ) und unter der Projektträgerschaft des Ministeriums für Umwelt und Verkehr Baden-Württemberg. Die Projektdurchführung erfolgt durch das Forschungszentrum Karlsruhe, Institut für Angewandte Informatik (FZK-IAI), das Ingenieurbüro IUT und Clemens Ritter. Wie auch in den übrigen Projekten der XfaWeb-Familie wird die Entwicklung durch eine Projektsteuerungsgruppe begleitet, welche dafür sorgt, daß frühzeitig die Anforderungen aus der Praxis mit einbezogen werden können. In dieser Gruppe sind Vertreter aus Forschung, Landratsämtern, Regierungspräsidien und Umweltämtern der Städte.

Ziel der ersten Stufe war, unter Nutzung der in AlfaWeb entwickelten Verfahren und Werkzeuge einen Prototypen zu implementieren, der es erlaubt, in einem breiteren Kreis von Fachpublikum die Einsatz- und Entwicklungsmöglichkeiten dieses Ansatzes zu diskutieren. Zu diesem Zweck wird nach Abschluß einer Testphase Ende Januar 1999 ein entsprechender Workshop veranstaltet werden.

Als Inhalt des Prototypen (Abb.10 zeigt die Startseite) sind vorgesehen:

- die vollständige Liste der Berichte von LfU und UVM zum Bodenschutz inkl. jeweils einer Kurzbeschreibung,
- exemplarisch zwei vollständige Berichte,
- vollständig ausgearbeitete Übersichtsinformationen zum Bodenschutz in Baden-Württemberg,
- einen Verweis auf das existierende Fachinformationssystem Bodenschutz (FIS-Bo),
- Bezugsquellen für Karten, wie sie im Bodenschutz eingesetzt werden, und beispielhaft einige thematische Karten,
- eine Übersicht über laufende und abgeschlossene Projekte (Datenblätter),
- ein Vorschlag zu einem Fachzugang im Bodenschutz,
- Schlagwortsuche,
- Volltextsuche,
- Informationen zum Projekt BofaWeb,
- Verweise auf bodenschutz-relevante Informationsangebote im WWW,
- eine Hilfe zur Suche im WWW.

Der Prototyp ist auf einem Server des FZK-IAI installiert und über das Internet jederzeit für das Entwicklungsteam inkl. der Projektsteuerungsgruppe erreichbar. Dadurch sind auch zwischen den Projektsitzungen kurzfristig inhaltliche und gestalterische Absprachen per Email möglich.

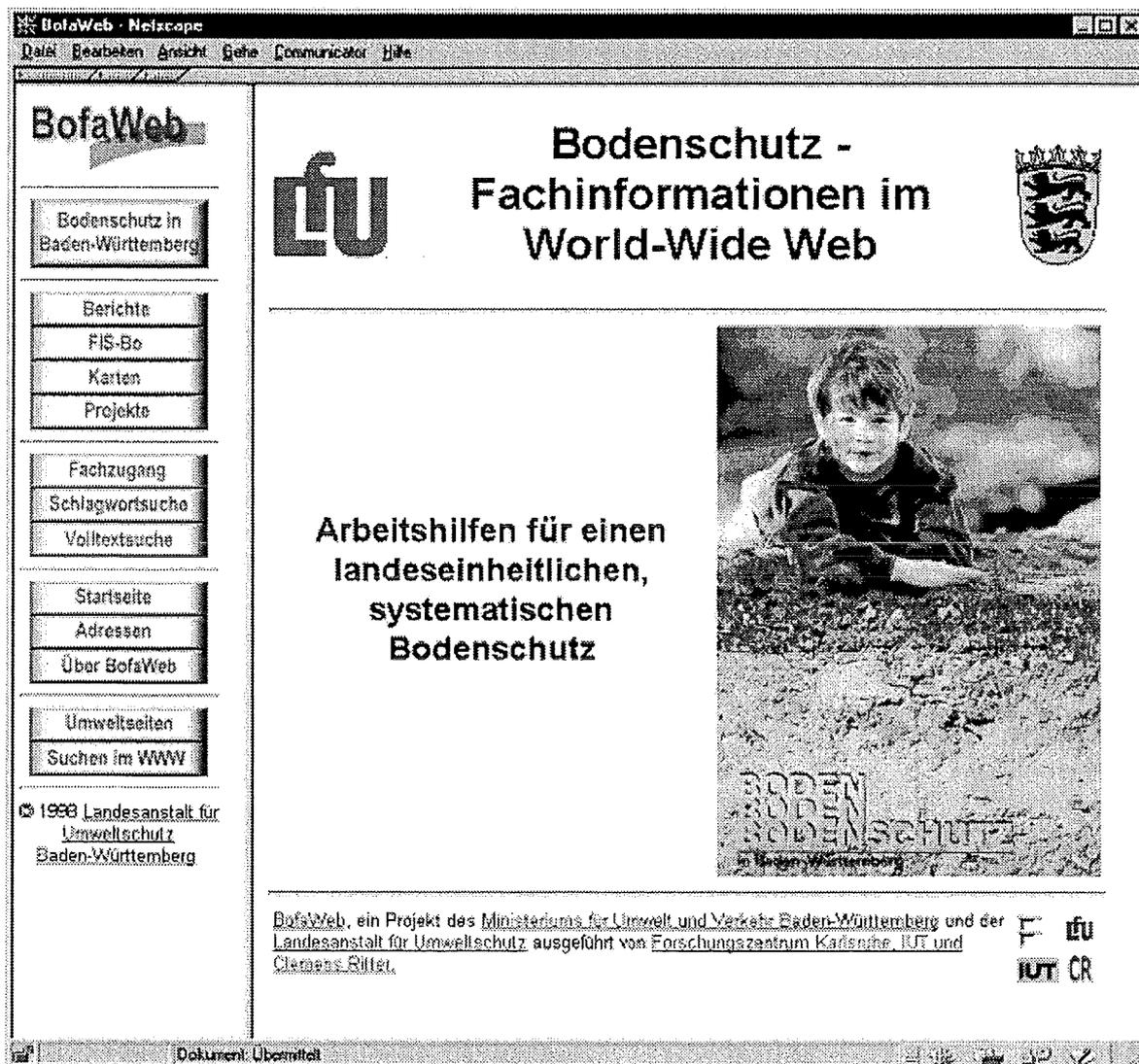


Abb.10: Startseite BofaWeb

3. Werkzeug-Entwicklung

Ziel der Werkzeug-Entwicklung ist es, Software-Komponenten und weitere Hilfsmittel zur Verfügung zu stellen, die für die gesamte XfaWeb-Familie und möglichst auch für andere Applikationen verwendbar sind. Im Rahmen von GLOBUS V wurden i.w. die Systemverwaltungskomponente (s. Kap. 3.1) zur Einsatzreife weiterentwickelt und als Vorbereitung auf den geplanten kostenpflichtigen Einsatz der Vollversionen im Internet die möglichen Zahlungs- und Abrechnungssysteme untersucht (s. Kap. 3.2). In Kap. 3.3 sind die an den übrigen Komponenten vorgenommenen Erweiterungen beschrieben.

3.1 Systemverwaltungskomponente

Die Systemverwaltungskomponente (SVK) ist ein Programmpaket, mit dem die Berichtsdaten der XfaWeb-Fachinformationssysteme der LfU verwaltet und für das WWW aufbereitet werden können.

Sie enthält Daten von Berichtsreihen, Berichten und Teilberichten der einzelnen Fachinformationssysteme. Diese Daten bestehen aus bibliografischen Angaben (Titel, Autor, Erschein-Datum, Preis etc.) und Angaben über Dateinamen, Verzeichnisse und WWW-Eigenschaften. Das Wissen über die letztgenannten ist zur automatischen Erstellung der Webseiten notwendig.

Das Konzept der Berichtsverwaltung und der Bereitstellung im WWW erfordert zwei Benutzer-Arbeitsbereiche. Im ersten befindet sich die Entwicklungsumgebung mit der Rohfassung der Berichte (RTF-Dateien), im zweiten die Server-Dateien (HTML-, GIF- und JPG-Dateien). Auf der Entwicklungsseite werden die Berichte erfasst, konvertiert und anschließend auf die Serverseite kopiert.

3.1.1 Datenbankaufbau

Die Systemverwaltungskomponente eines Fachinformationssystems besteht aus zwei Datenbank-Dateien. Eine Datei enthält die fachinformationssystem-spezifischen Daten ('Daten'-Datenbank), die andere die SVK-Programme, -Formulare etc. ('SVK'-Datenbank). Dies hat den Vorteil, daß Programmänderungen unabhängig von den Daten gemacht werden können. Zudem ist es nur dadurch möglich, mehrere Fachinformationssysteme mit einer SVK zu betreiben, da zwar die Datenstruktur gleich, die Dateninhalte aber verschieden sind.

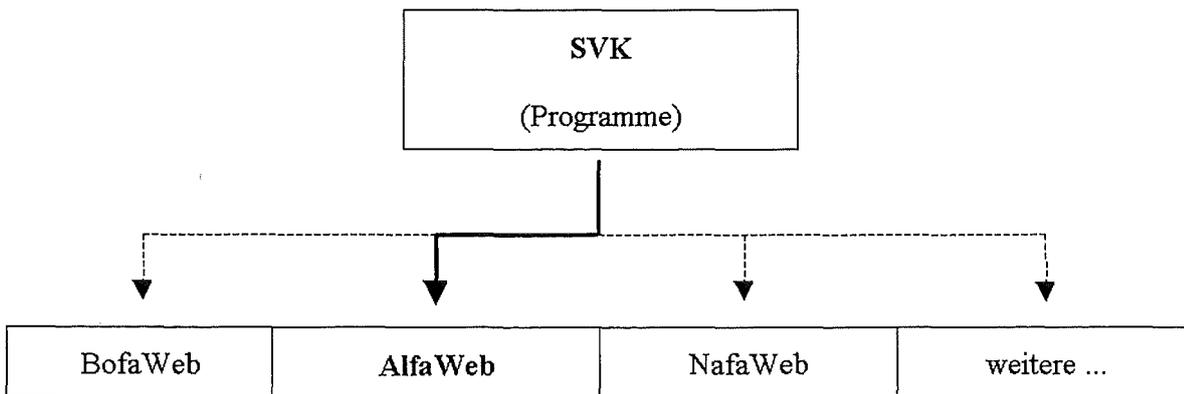


Abb.11: Struktur der Systemverwaltungskomponente

3.1.2 Realisierung

Die Systemverwaltungskomponente ist anwendungsunabhängig für alle Fachinformationssysteme mit gleicher Datenstruktur einsetzbar (AlfaWeb, NafaWeb, BofaWeb, ...).

Sie basiert auf der Datenbank Access 97 von Microsoft unter Verwendung der Programmiersprache Visual Basic for Applications (VBA).

Durch die Einbeziehung zusätzlicher Fachinformationssysteme (ursprünglich nur AlfaWeb) wurde eine Erweiterung der Systemverwaltungskomponente notwendig. Um dem Systemverwalter die Möglichkeit zu geben, mehrere Fachinformationssysteme zu bearbeiten, wurde ein fachinformationssystem-übergreifender Einstieg implementiert. Damit kann der Systemverwalter innerhalb der SVK das gewünschte Fachinformationssystem zur Bearbeitung aus-

wählen, ohne die SVK erneut starten zu müssen. In Abb. 12 ist das Formular für die Auswahl der Fachinformationssysteme, in Abb.13 das Startformular eines Fachinformationssystems wiedergegeben.

Die Systemverwaltungskomponente wurde entsprechend den aktuellen Anforderungen fertiggestellt und bei der Firma IUT installiert, wo inzwischen die Fachinformationssysteme BofaWeb und NafaWeb betreut werden.

XfaWeb - Auswahl Fachinformationssystem

Bitte Fachinformationssystem (FIS) auswählen:

- AlfaWeb
- BofaWeb
- NafaWeb

Lokale Angaben:

Projektname: AlfaWeb

Laufwerk: D:

Pfadangaben: \\Daten\FZK\LIU\AlfaWeb\mdb

Datenbank: alfawebTab.mdb

FIS neu FIS löschen FIS starten Schließen

21

Abb.12: Startformular zur Auswahl eines Fachinformationssystems

3.1.3 Benutzerschnittstelle

Die SVK kann in drei Blöcke untergliedert werden:

1. Bearbeitung von Berichtsreihen, Berichten, Teilberichten, Autoren und UDK-Schlagworten
2. Allgemeine Konvertierungs- und Kopier Routinen (Verwaltung)
3. Systemfunktionen

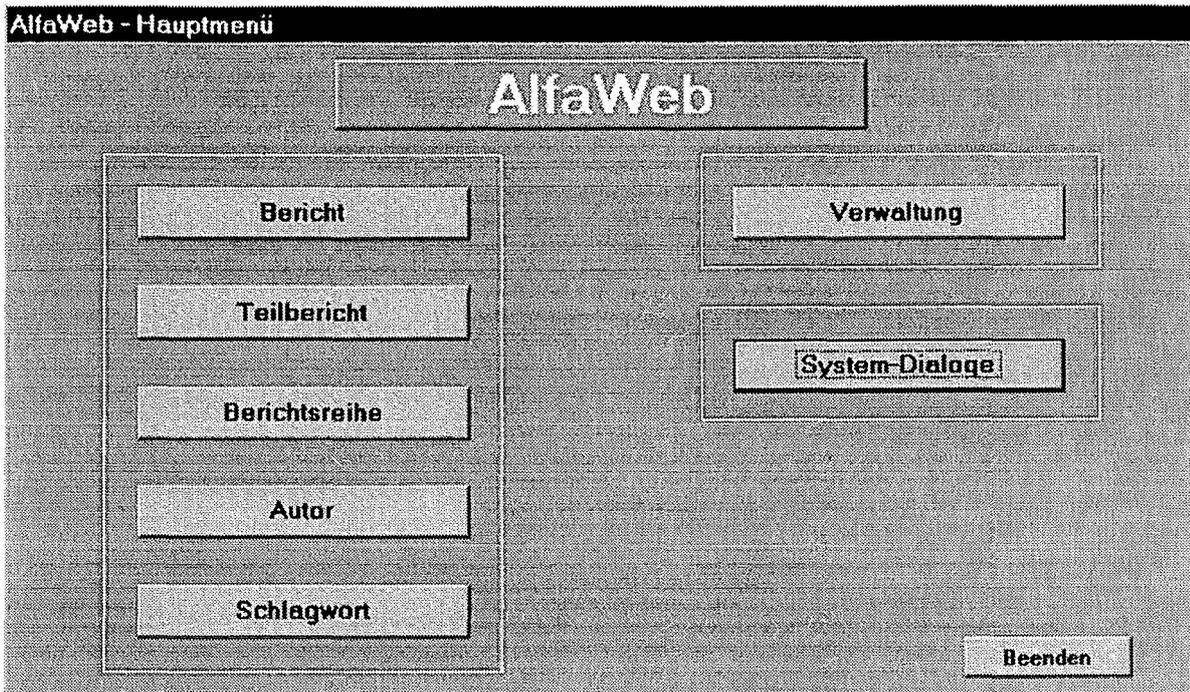


Abb.13: Startformular der SVK für das Fachinformationssystem AlfaWeb

Im ersten Block werden in erster Linie Daten verwaltet, die in der WWW-Version in der Berichteliste zu sehen sind (Reihename, Berichtstitel, Autoren etc.). Zusätzlich werden bei den Berichten Typdaten und Parameter erfasst, die zur Konvertierung benötigt werden. Dazu gehören das Verzeichnis, in dem der Bericht zu finden ist, der Name des zu konvertierenden RTF-Files, Gliederungstiefe usw.. Außerdem werden hier die für den Bericht verwendeten GIF-Files für Buttons u. ä. definiert. Aus diesen Daten werden die URLs für den Bericht, die Kurzbeschreibung, das Inhaltsverzeichnis und das PDF-File generiert. Auch kann von hier aus die Konvertierung eines Berichts angestoßen und das Kopieren auf den Server vorgenommen werden.

Solange das Konvertierungsprogramm als eigenständige Applikation benutzt wurde, musste für jeden Bericht ein eigenes Konfigurationsfile erstellt werden, in dem die Konvertierungsparameter berichtsspezifisch angepasst werden mussten. Parallel dazu musste auch die Literaturliste von Hand aktualisiert werden.

Durch die Integration des Konvertierungsprogramms in die SVK wird dieses Konfigurationsfile nun vor der Konvertierung von der SVK generiert und ist somit immer auf dem aktuellen Stand. Damit wird eine Inkonsistenz zwischen Konfigurationsfile und WWW-Literaturliste vermieden.

Desweiteren können die Querverweise (Links), die von anderen Berichten auf diesen Bericht existieren, angezeigt und ausgedruckt werden. Dazu wird zuerst eine Gesamt-Querverweisliste erstellt und anschließend werden die berichtsrelevanten Links herausgefiltert und in der Datenbank abgelegt. Eine zusätzliche Funktion der SVK ist das Extrahieren von Berichtsdaten, die zum Eintrag in den Umweltdatenkatalog (UDK) bestimmt sind.

AlfaWeb - Bericht bearbeiten

Reihe: **Materialien zur Altlastenbearbeitung**

Bericht: **Erkundung ehemaliger Gaswerksstandorte** Teilbericht(e): **0**

Untertitel: _____ Seitenzahl: **91**

Band-Nr. **Band 01** Erschein-Datum: **Februar** 1990

Preis: **15,00** DM Status: **vollständig (integriert)**

Anmerkung: _____

Beschreibung: **Die Vorgehensweise bei der historischen Erkundung wird in dieser Schrift am ausführlichsten beschrieben. Zu den daran anschließenden Erkundungsstufen werden erste Erfahrungen und**

Autor		Schlagwort	
▶	Landesanstalt für Umweltschutz Baden-Württemberg	▶	
*			

UrlKurz: **berichte/abstracts/band01.html**

UrlInhalt: **berichte/mza01/eg_c.htm**

UrlBericht: **berichte/mza01/eg.htm**

UrlDruckvorl: **print/mza01.pdf**

Typdaten ...

Konvertieren ... **03.08.98 13:56:31**

Querverweise ... **26.06.98 15:14:36**

UDK-Mail

Datensatz neu **Speichern** **Löschen** **Schließen**

Abb.14: Dialog zur Berichtsbearbeitung

Der zweite Block der SVK (Verwaltung) enthält die Erstellung der Berichteliste, sowie berichtsübergreifende Konvertier- und Generierfunktionen (s. Abb. 15).

AlfaWeb - Verwaltung

Datei 'Berichte.htm' erstellen **Schauenster-Version erstellen**

Kurzbeschreibungen konvertieren **Inhaltsverzeichnisse konvertieren**

Gesamt-Querverweisliste erstellen **Volltextsuche**

Schlagwortliste erstellen

Schließen

Abb.15: Menü der Verwaltungsdialoge

Die Berichterliste (Button "Berichte" im Fachinformationssystem bzw. die Datei "Berichte.htm") wurde bisher von Hand erstellt (HTML-File) und musste bei Aufnahme eines neuen Berichts manuell aktualisiert werden. Dazu mussten für jeden neuen Bericht alle bibliografischen Daten, die in der Berichterliste erscheinen sollten, in diese Datei eingetragen werden. Hierbei war die Einhaltung von HTML-Konventionen notwendig.

Diese Funktion wird jetzt von der SVK übernommen. Dazu wird eine Textdatei als Eingabeschablone verwendet, die, zusätzlich zu den HTML-Befehlen, Platzhalter für die Reihen- und Berichtsdaten enthält. Diese werden durch die Werte in der Datenbank ersetzt und daraus das HTML-File generiert.

Die Konvertier- und Generierfunktionen ermöglichen z. Zt. die Erzeugung der Gesamtquerverweisliste, der XfaWeb-Schlagwortliste, der Schaufensterversion und des Index-Files der Volltextsuche. Außerdem können die Kurzbeschreibungen und die Inhaltsverzeichnisse konvertiert werden.

Die Systemfunktionen (Block drei der SVK) dienen zur Pflege fachinformationssystem-spezifischer (lokaler) Daten, wie

- **Pfadangaben zu den Berichten:**
Bei den Konvertierungen ist die Verzeichnisstruktur unterhalb eines Einstiegsverzeichnisses festgelegt. Der Pfad des Einstiegsverzeichnisses, der hier erfaßt wird, kann aber beliebig gesetzt werden.
- **Default-Werte für die Berichtsparameter:**
Viele Angaben zur Struktur eines WWW-Berichts sind für alle Berichte eines Fachinformationssystems gleich. Damit der Benutzer diese Angaben nicht immer neu eingeben muss, besteht hier die Möglichkeit, diese als Default-Werte zu definieren. Diese werden bei einem neuen Bericht als Vorschlagswerte angezeigt.
- **Hintergrundfarbe:**
Damit kann die Hintergrundfarbe der fachinformationssystem-spezifischen Dialoge bestimmt werden.

3.1.4 Fazit

Durch die Konzentrierung der verschiedenen Arbeitsschritte in einem Programm bringt die Systemverwaltungskomponente für den Systemverwalter eine wesentliche Vereinfachung seiner Tätigkeit. Alle Daten eines Berichts sind nur einmal gespeichert und die benötigten Parameter zur Konvertierung von Berichten werden daraus abgeleitet. Zudem wird die Versionsführung der einzelnen Bearbeitungsschritte durch Plausibilitäts- und Konsistenzprüfungen unterstützt.

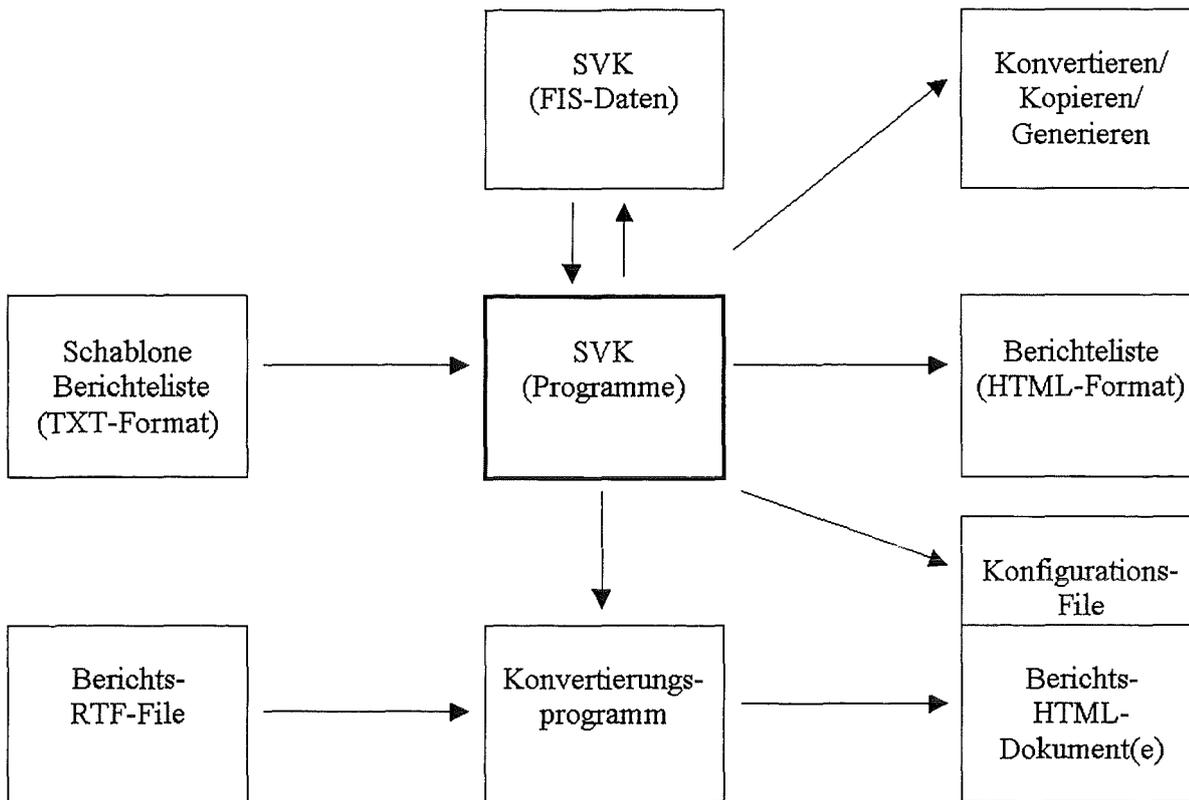


Abb.16: Die SVK als zentrales Element der Berichtsdatenverwaltung (vereinfacht)

3.2 Elektronische Zahlungs- und Abrechnungssysteme

Mittelfristig sollen vollständige Versionen der XfaWeb-Systeme im Internet angeboten werden. Bestimmte Teile sollen dann allerdings nur kostenpflichtig abgerufen werden können. Aus diesem Anlaß wurde eine Studie über Elektronische Zahlungs- und Abrechnungssysteme im Internet erstellt. Solche Systeme fallen unter den Begriff 'Electronic Commerce', worunter man das Kaufen und Verkaufen von Produkten, Informationen und Dienstleistungen über elektronische Netze versteht. Es wird im wesentlichen unterschieden zwischen produktbezogener Ware, von der meistens eine bestimmte Stückzahl geordert wird, und informationseinheitbezogener Ware, was für XfaWeb zutrifft. Das Hauptaugenmerk der Studie wurde deshalb auf letzteres gelegt. Die Studie gibt einen Überblick über die derzeit im Internet existierenden Zahlungs- und Abrechnungssysteme.

3.2.1 Systemeigenschaften eines Elektronischen Zahlungssystems

Elektronische Zahlungssysteme sind nicht zuletzt durch prinzipielle Systemeigenschaften gekennzeichnet /9/. Von Bedeutung sind hier:

- **Sicherheit:** Schutz vor jeglichem Mißbrauch. Dieser wird in der Regel durch Datenverschlüsselung (Kryptographie) erreicht. Sogenannte "Trust-Center" und "Certification Authorities" /10/ beginnen derzeit damit, elektronische Pässe auszustellen. Mit diesem Zertifikat wird die Identität einer Person untrennbar mit seiner digitalen Signatur verknüpft. Damit soll sichergestellt werden, daß es sich bei einem Kommunikationspartner auch um die angegebene Person handelt.

- **Anonymität:** Elektronische Zahlungsvorgänge werden anonym abgewickelt, gleich der Barbezahlung in einem Geschäft.
- **Skalierbarkeit:** Ein Zahlungssystem, das skaliert, kann problemlos auf einen großen Nutzerkreis erweitert werden.
- **Mikrozahlungen:** Fallen Mikrozahlungen an, muß bei der Auswahl eines Zahlungssystems auf dessen Wirtschaftlichkeit geachtet werden, d.h. Kleinbeträge (weniger als DM 5) müssen wirtschaftlich abgerechnet werden können.

Bei der Auswahl eines Zahlungs- oder Abrechnungssystems muß die Wichtigkeit der einzelnen Systemeigenschaften berücksichtigt werden.

3.2.2 Übersicht derzeit verfügbarer Systeme

3.2.2.1 Zahlungssysteme

Die hier beschriebenen Systeme zeigen Möglichkeiten auf, wie elektronisch bezahlt werden kann, d. h. wie der Geldfluß stattfinden kann.

3.2.2.1.2 Kreditkartenzahlung über das Internet

Bei der Kreditkartenzahlung über das Internet wird ein bereits bestehendes Zahlungssystem verwendet. Lediglich das Medium, über das der Kunde seine Transaktionsinformationen dem Händler übermittelt, wird geändert. Der Händler kann über sein normales Vertragsverhältnis mit dem entsprechenden Karteninstitut abrechnen. Internationale Zahlungen sind problemlos möglich, da Kreditkarten weltweit verbreitet sind.

Bei unverschlüsselter Übertragung von Kreditkarteninformationen besteht die Gefahr, daß diese von Dritten eingesehen werden. Aus diesem Grund wurden Protokolle entwickelt, die eine gesicherte Datenübertragung zwischen dem Kunden und dem Händler ermöglichen sollen. In erster Linie sind hier die Protokolle **SSL** (Secure Socket Layer) von Netscape Communications und **S-HTTP** (Secure HyperText Transfer Protocol) von Terisa Systems zu erwähnen. Diese beiden Protokolle ermöglichen einen sicheren Transport beliebiger Daten. VISA und MasterCard entwickelten gemeinsam für eine sichere Kreditkartenzahlung den offenen Standard **SET** (Secure Electronic Transaction). SET erfaßt im Unterschied zu den Übertragungsprotokollen SSL und S-HTTP die sichere Abwicklung von kompletten Kaufvorgängen von der Bestellung bis zur Quittung. Desweiteren legt es Anmeldeprozeduren für das Ausstellen von Zertifikaten sowie die Struktur von Zertifizierungstellen fest. Komplette Zahlungsdienste für das Internet entwickelt die Firma **CyberCash**. Sie vermarktet insbesondere das System zur gesicherten Übertragung von Kreditkarteninformationen (Secure Internet Payment System) erfolgreich.

3.2.2.1.3 Digitales Bargeld

Mit digitalem Bargeld wird die Eigenschaft und die Funktionalität von realem Bargeld in die digitale Welt abgebildet. Dabei wird reales Geld bei einer Bank in digitales Geld getauscht. Das Problem dabei ist, daß digitales Bargeld, da es nur durch eine Reihe von Bits dargestellt wird, beliebig oft kopiert werden kann. Es muß mit Mechanismen ausgestattet sein, die entweder eine Kopie erkennen oder das Kopieren verhindern. Man unterscheidet bei digitalem

Bargeld zwischen anonymem Bargeld und identitätsbezogenem Bargeld. Bei anonymem Bargeld kann die Bank nur die Gültigkeit des Geldes prüfen. Sie kann nur im Betrugsfall eine Verbindung zum Kunden herstellen (ein kryptographisches Verfahren namens 'secret sharing' stellt sicher, daß die Bank diese Möglichkeit nicht mißbrauchen kann). Bei identitätsbezogenem Bargeld kann die Bank den Kunden identifizieren und den Geldfluß genau verfolgen. Desweiteren wird unterschieden zwischen Online-Zahlung und Offline-Zahlung. Bei einer Online-Zahlung wird die Münze sofort von der Bank auf Echtheit geprüft. Dazu muß die Bank sämtliche im Umlauf befindlichen Münzen oder sämtliche bereits eingereichten Münzen kennen. Eine Münze ist nur für eine Transaktion gültig. Wird offline bezahlt, gelangt eine Münze erst nach einiger Zeit wieder zur Bank. Ein Betrug wird also erst im nachhinein erkannt. Die Münze muß daher Informationen über den oder die Besitzer enthalten.

3.2.2.1.3 Chipkarten-Zahlungssystem

Zukünftig könnte das Chipkarten-Zahlungssystem eine große Rolle spielen. Eine solche elektronische Geldbörse hätte die gleichen Eigenschaften wie eine normale Geldbörse. Die Chipkarte muß vorbezahlt sein. Sie läßt sich beliebig oft mit Geldeinheiten beladen und wieder entladen. Benutzt wird sie wie eine Telefonkarte. Die Bezahlung erfolgt schnell und unkompliziert. Alle Bezahlungen finden anonym statt. Wann wo was gekauft wird kann nicht zurückverfolgt werden. Auf der Chipkarte werden lediglich betriebsinterne Zustandsinformationen protokolliert. Bei einer Bezahlung via Internet müßte der Kunde über ein Chipkarten-Leseteil verfügen, das beispielsweise in sein Keyboard integriert sein könnte (derartige Keyboards sind bereits im Handel). Der Händler dagegen muß über ein entsprechendes Terminal verfügen. Die ihm übertragenen Geldeinheiten kann er bei der Bank in reales Geld umtauschen. Herkömmliche Sicherheitsmechanismen müßten für eine Internet-Lösung mit Chipkarten noch angepaßt werden.

3.2.2.2 Abrechnungssysteme

Bei den folgenden Systemen steht das "wofür wird bezahlt" im Vordergrund. Die Bezahlung selbst kann dann wieder mit Hilfe eines Zahlungssystems oder durch konventionelle Zahlungsweise erfolgen.

3.2.2.2.1 Abrechnung über Kundenkonten

Die Kundenkonten werden hierbei von einer dritten Partei (Systembetreiber, z. B. Online-Dienste) geführt. Diese wickelt auch die Bestellungen bzw. Zahlungen mit dem Händler ab. Der Händler wie auch der Kunde stehen in einem festen Vertragsverhältnis mit dem Systembetreiber. Ein wesentlicher Vorteil dieses Abrechnungssystems ist, daß mehrere Transaktionen vom Systembetreiber gesammelt abgerechnet werden können. Teure Einzeltransaktion entfallen dadurch, weshalb auch kleine Beträge wirtschaftlich abgerechnet werden können. Denkbar wäre auch, daß der Händler selbst die Kundenkonten einrichtet und die Abrechnungen vornimmt, was mit nicht unerheblichem Aufwand verbunden ist. Vertragsverhältnisse mit Systembetreibern wären dann überflüssig.

3.2.2.2 Jahresabonnement

Beim Jahresabonnement handelt es sich um ein einfaches, verwaltungsaufwandarmes Abrechnungssystem. Einmal jährlich wird eine Pauschalzahlung geleistet und ein Paßwort mit Zugangsberechtigung vergeben. Um eventuellen Mißbrauch zu unterbinden, könnte man eine Ausweisung mit Hilfe eines elektronischen Passes, ausgestellt von einem Trust-Center /10/, fordern.

3.2.3 Vorauswahl für XfaWeb

3.2.3.1 Anforderungsanalyse

Die Nutzer der Informationen der XfaWeb-Systeme sind in der Hauptsache:

- Sachbearbeiter in den Behörden: Abruf von Informationen kostenlos
- Mitarbeiter von Ingenieurbüros: Abruf von Informationen kostenpflichtig

Hinzukommen könnten:

- Angehörige von Universitäten: Abruf von Informationen kostenpflichtig (eventuell Sondertarife)
- Interessierte Öffentlichkeit: Abruf von Informationen kostenpflichtig

Das Angebot der XfaWeb-Systeme gliedert sich in:

- Zugangssystem
- Fachinformationen

Das Zugangssystem (Literaturliste, Volltextsuche, Schlagwortsuche etc.) sollte für jedermann kostenfrei sein. Ebenso ist nicht beabsichtigt, Kosten zu erheben für Übersichtsinformationen, wie Inhaltsverzeichnisse, Kurzbeschreibungen oder eine Einführung in das Fachgebiet.

Erst das Abrufen gewünschter Fachinformationen sollte kostenpflichtig sein. Die Beschreibungen müßten auch detaillierte Preisangaben enthalten (beim Jahresabonnement ist das nicht nötig), damit der Nutzer weiß, welche Kosten bei einem Informationsabruf auf ihn zukommen. Nur so kann sich ein Interessent für oder gegen einen kostenpflichtigen Informationsabruf entscheiden.

Die Fachinformationen bestehen aus:

- html-Dateien
- gif-Bildern
- pdf-Dateien (Druck-Dateien)

Der Preis für eine html-Datei sollte unabhängig von deren Inhalt einheitlich sein, obwohl inhaltlich (Länge) erhebliche Unterschiede bestehen und manche html-Dateien nur Verweise auf Unterkapitel enthalten. Jedoch erscheint aufgrund der großen Anzahl von html-Dateien eine dateibezogene Preisgestaltung nicht praktikabel. Abgerufene gif-Bilder könnten auch nach Anzahl abgerechnet werden. Die pdf-Dateien unterscheiden sich stark in ihrer Größe. Da ihre Anzahl nicht übermäßig groß ist, könnte man sie preislich unterscheiden.

3.2.3.2 Realisierungsvorschläge

Bei verschiedenen Diskussionen sind drei Möglichkeiten für die Abrechnung und Bezahlung eines Xfa-Web-Internetzugriffes in die engere Wahl gekommen.

3.2.3.2.1 Jahresabonnement

Geringe Kosten für Anschaffung und Verwaltung entstehen bei einem Jahresabonnement. Man muß bei dieser Wahl nur einmal jährlich eine Zugangsberechtigung vergeben sowie die Jahrespauschale einziehen. Dadurch muß der Betreiber den einzelnen Kundenzugriff und dessen Informationsabruf nicht detailliert nachweisen. Der Kunde allerdings muß warten, bis er eine Zugangsberechtigung erhalten hat. Beachten muß man hierbei, daß Paßwörter einfach weitergegeben werden können und eine Rechneridentifikation nicht immer möglich ist, da beispielsweise Online-Dienste die TCP/IP-Adresse dynamisch vergeben. Um ein Gefühl für eventuellen Mißbrauch zu bekommen, könnte man dieses Zahlungssystem zunächst zeitlich begrenzt testen. Mißbrauch könnte man erschweren, indem man fordert, daß sich der Kunde mit einer digitalen Signatur, die von einem Trust-Center ausgestellt wurde, ausweist. Da der Kunde pauschal bezahlt, sollte gewährleistet sein, daß der Server möglichst rund um die Uhr verfügbar ist.

3.2.3.2.2 Kundenkonto

Ebenfalls in die engere Auswahl kommt das selbstgeführte Kundenkonto. Der Vorteil hierbei ist, daß jeder Zugriff gemäß dem damit verbundenen Informationsabruf abgerechnet werden kann. Dadurch können die Zugriffe preislich unterschieden werden. Da nur abgerufene Informationen bezahlt werden müssen, ist anders als beim Jahresabonnement ein zeitweiser Ausfall des Servers unproblematischer. Von Nachteil ist der nicht unerhebliche Aufwand für die Erstellung eines solchen Abrechnungssystems sowie für die spätere Verwaltung desselben. Die Zugriffe müssen lückenlos nachgewiesen werden. Da hier auch Kleinbeträge anfallen können, muß man in jedem Fall bei Nutzung des Systems einen Mindestbetrag fordern, damit diese das System nicht unrentabel machen. Ebenso wie beim Jahresabonnement kann gefordert werden, daß sich der Kunde mit einer digitalen Signatur, die von einem Trust-Center ausgestellt wurde, ausweist. Spontaner Informationsabruf ist wie beim Jahresabonnement auch hier nicht möglich, da zuerst eine Zugangsberechtigung beantragt werden muß. Bei dieser Auswahl müssen eventuell entstehende Rechtsstreits einkalkuliert werden, falls ein Kunde nach Erhalt einer detaillierten Rechnung Zugriffe in Frage stellt.

3.2.3.2.3 Chipkarten

Zukünftig könnte man auch die Chipkarte in Erwägung ziehen. Wie beim Kundenkonto kann damit jeder Zugriff gemäß dem damit verbundenen Informationsabruf abgerechnet werden. Ein Verwaltungsaufwand in Form von Kontoführung entsteht hierbei nicht. Jeder Zugriff wird sofort verrechnet, weshalb ein späteres Infragestellen desselben i. allg. nicht zu erwarten ist. Mikrozahlungen sind möglich. Ein Interessent kann sofort Informationen abrufen. Er muß keine Zugangsberechtigung beantragen und den Erhalt dieser abwarten. Ein Ausfall des Servers hat keine rechtlichen Konsequenzen. Derzeit ist dieses Zahlungssystem wegen fehlender Hardware-Ausstattung nicht möglich.

3.2.3.2.4 Werbung

Eine einfache Alternative zu den vorgeschlagenen Zahlungs- bzw. Abrechnungssystemen wäre, daß man den Zugriff kostenfrei gestattet und statt dessen die Finanzierung über Werbung gewährleistet. Der große Vorteil dieser oft gewählten Alternative (anderer Anbieter) ist ohne Zweifel, daß keine Kosten für die Anschaffung und später auch kein Verwaltungsaufwand für ein Zahlungssystem entsteht. Notwendig für diese Alternative ist allerdings Marketing. Werbe-Kunden müssen gefunden werden. Normalerweise fordert ein Werbekunde den Nachweis über Anzahl und Dauer des Zugriffes (Werbezeiten), um sicherzustellen, daß seine Werbung auch die Öffentlichkeit erreicht.

3.3 Sonstige Komponenten

3.3.1 Hilfsmittel zur Berichtserstellung

Die Erstellung von Berichten für die XfaWeb-Systemfamilie erfolgt mittels Microsoft Word unter Verwendung einer speziellen Dokumentvorlage, die in einer Richtlinie /11/ näher erläutert wird. Gegenüber dem in den zurückliegenden GLOBUS-Phasen berichteten Projektstand wurden nur kleinere Erweiterungen vorgenommen.

In die Dokumentvorlage wurden einige neue Formatvorlagen aufgenommen, die sich bei der Bearbeitung von Berichten für AlfaWeb und NafaWeb als wünschenswert erwiesen (Fließtext mit größerem und kleinerem Font, Tabellentext). Um die Bearbeitung von Berichten nach Richtlinie zu erleichtern, erweitert die Dokumentvorlage die WinWord-Menüleiste um ein Pull-Down-Menü, das bereits vorher vorhandene und einige neue Makros auf diese Weise zur Verfügung stellt:

- Gleichzeitiges Speichern als .doc und .rtf
- Formular zur Erfassung von Schlagworten
- Markieren von Verweisen, Schlagworten etc.

3.3.2 Konvertierungsprogramm

Die nach Richtlinie aufbereiteten Berichte (s. Kap. 3.3.1) werden über ein Konvertierungsprogramm in HTML-Seiten zerlegt und können in das jeweilige Fachsystem aufgenommen werden. Das Konvertierungsprogramm inkl. seiner Parametrisierung wurde inzwischen in die Systemverwaltungskomponente (s. Kap. 3.1) integriert.

Funktionale Erweiterungen wurden in GLOBUS V nur in geringem Umfang vorgenommen:

Bei der Konvertierung eines Berichts, der aus Teilberichten aufgebaut ist, werden jetzt im Inhaltsverzeichnis des Berichts automatisch Verweise auf die Inhaltsverzeichnisse der Teilberichte eingefügt.

Um leichter Änderungen am Layout existierender Fachsysteme vornehmen zu können, wurde eine eingeschränkte Unterstützung von Cascading Style Sheets (CSS) implementiert. Bei der Konvertierung eines Berichts kann jetzt eine Referenz auf ein externes Style Sheet (CSS-Datei) angegeben werden. Diese Referenz wird in jeden erzeugten HTML-File einge-

baut. Das Editieren der Style Sheet Datei, z.B. die Definition einer anderen Hintergrundfarbe, verändert gleichzeitig das Erscheinungsbild aller HTML-Dateien, die auf diese Datei verweisen. Das Anpassen jeder einzelnen HTML-Dateien entfällt.

3.3.3 Volltextsuche

Zur Volltextsuche wurden bisher unterschiedliche Werkzeuge für die WWW-Version und die CD-ROM-Version benutzt. Während im WWW der Microsoft Index Server als Erweiterung des Internet Information Server eingesetzt wurde, basierte die Volltextsuche der CD-ROM auf dem inzwischen nicht mehr unterstützten Shareware-Produkt SWISH (Kevin Hughes). Neben dem Aufwand zur Pflege zweier verschiedener Systeme ergaben sich auch Probleme mit den eingesetzten Werkzeugen. SWISH wurde für Unix entwickelt, eine befriedigend funktionierende Portierung für den PC wurde nicht gefunden. Eine Index-Erzeugung mit SWISH auf dem PC war nur für eine begrenzte Anzahl Seiten möglich, AlfaWeb konnte nur unter Unix indiziert werden. Beim Microsoft Index Server hingegen traten in einigen Fällen (z.B. gleichzeitige Aktualisierung vieler Dateien) Stabilitätsprobleme auf, Suchanfragen mit Wildcards lieferten teils nur schwer nachvollziehbare Suchergebnisse.

Mit SWISH-E /12/, einer Weiterentwicklung von SWISH, wurde inzwischen ein Werkzeug gefunden, das sowohl für die WWW- als auch für die CD-ROM-Version genutzt werden kann und unter Unix und Windows verfügbar ist. SWISH-E wird gemäß der GNU Public License /13/ vertrieben, kann also kostenlos verwendet werden.

Die Volltextsuche mit SWISH-E ist ein zweistufiger Prozeß. Im ersten Schritt wird eine Index-Datei erzeugt, wobei dieser Vorgang über eine Parameter-Datei gesteuert werden kann. Die eigentliche Suchanfrage verwendet im zweiten Schritt diesen Index-File zur Ermittlung der passenden Dokumente. Falls sich Änderungen in dem über die Volltextsuche abgedeckten Dateien-Pool ergeben, muß die Index-Datei neu erzeugt werden⁴, ein bei größeren Datenmengen länger dauernder Vorgang. Die Erzeugung der Index-Datei für AlfaWeb (ca. 9000 HTML-Files) benötigt knapp 30 Minuten⁵, die Index-Datei umfaßt mit 11.3 MB ca. 25% des Speicherplatzes der HTML-Files. Die Bearbeitung der Suchanfragen über die Index-Datei zeigt ein gutes Antwortzeitverhalten, das (subjektiv) besser ist als bei beiden vorher verwendeten Werkzeugen.

⁴ Dies erfolgt über die Systemverwaltungskomponente.

⁵ PC 266MHz, Windows NT 4.0, 64 MB Hauptspeicher

4. Ausblick

Die beiden Fachsysteme NafaWeb und BofaWeb wurden in 1998 als Prototypen realisiert. Bei erfolgreicher Abstimmung mit den jeweiligen Nutzergruppen steht für diese Systeme anschließend der inhaltliche Ausbau im Vordergrund. Ziel ist die Aufnahme einer "kritischen Masse" an Unterlagen, d.h. es muß genügend Inhalt eingebracht werden, um einen nutzbringenden Einsatz im Routinebetrieb zu ermöglichen. Erst dann ist die Produktion einer allgemein verfügbaren CD-ROM bzw. die Freigabe einer Netzversion für einen breiten Nutzerkreis sinnvoll.

Die Freigabe im Internet erfordert jedoch die vorherige Installation eines Abrechnungssystems. Basierend auf der in 1998 angefertigten Studie soll ein Verfahren ausgewählt und praktisch erprobt werden. Kostenlos im Internet zur Verfügung gestellt werden, wie bei AlfaWeb, auch für die anderen XfaWeb-Systeme Schaufenster-Versionen.

Stärker als AlfaWeb benötigen NafaWeb und BofaWeb einen Kartenteil, d.h. die Möglichkeit thematische Karten in das Informationsangebot mit aufzunehmen. Nach einer entsprechenden Anforderungsanalyse soll eine für die gesamte XfaWeb-Familie nutzbare Komponente zur Kartendarstellung unter Verwendung von in GLOBUS bereits vorliegenden Lösungen entwickelt werden.

Bisher sind die Mitglieder der XfaWeb-Familie völlig eigenständige Systeme, die lediglich auf einem gemeinsamen Satz an Werkzeugen und Design-Richtlinien aufsetzen. Im weiteren Ausbau ist jedoch auch ein eigenes XfaWeb-Zugriffssystem vorgesehen, das den einzelsystem-übergreifenden Zugriff erlaubt. Außerdem sollten Berichte und andere Daten, welche in mehreren Fachsystemen benötigt werden, unmittelbar in die Verwaltung einer übergeordneten Komponente übergehen. Mit zunehmendem Umfang und wachsender Komplexität der Systeme wird auch der Bedarf an intelligenteren Zugriffsmethoden immer größer. Hier soll untersucht werden, welche weiteren Möglichkeiten dem Benutzer geboten werden können, schnell und effizient an benötigte Informationen heran zu kommen.

Daueraufgabe in der XfaWeb-Entwicklung bleibt die Marktbeobachtung und das Aktualhalten der Werkzeuge. Die rasend-schnelle Weiterentwicklung der in XfaWeb verwendeten Basistechnologien erfordert eine ständige Überprüfung der Entwurfsentscheidungen und Realisierungsmethoden und erzwingt die Anpassung an neue Sprach-Standards und Software-Versionen. Andererseits bieten sich aber auch immer wieder neue Chancen, eine noch bessere Unterstützung der Anwender zu erreichen.

5. 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 Mayer-Föll, R.; Jaeschke, A. (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/ Weidemann, R.; Ehrmann, F.; Geiger, W.; Reißfelder, M.; Schmid, E.; Witt-Hock, J. (1997): Vom Pilotprojekt zur Anwendung, in Mayer-Föll, R.; Jaeschke, A. (Hrsg.): Projekt GLOBUS, Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg - Phase IV 1997, Forschungszentrum Karlsruhe, wissenschaftliche Berichte, FZKA 6000, S. 153 - 174.
- /4/ Geiger, W.; Reißfelder, M.; Weidemann, R.: Sharing technical information on contaminated sites by means of modern information and communication technologies, ASME Focus on Environmental Communication, 2-3 December 1997, Miami, Florida.
- /5/ Weidemann, R.; Geiger, W.; Jaeschke, A.; Reißfelder, M.: Moderne Informationsbereitstellung durch AlfaWeb, in Landesanstalt für Umweltschutz Baden-Württemberg (Hrsg.): Statusbericht Altlasten, 10 Jahre Altlastenbearbeitung in Baden-Württemberg. Materialien zur Altlastenbearbeitung, Bd. 27, S. 307-312, Karlsruhe, 1997.
- /6/ Geiger, W.; Reißfelder, M.; Weidemann, R.: Support of the Efficient Handling of Contaminated Sites by the AlfaWeb Information System, in Contaminated Soil '98 (ConSoil '98), Proc of the Sixth Int. FZK/TNO Conf. on Contaminated Soil, 17-21 May 1998, Edinburgh, UK. Thomas Telford Publishing, London 1998, pp. 1163-1164.
- /7/ Lautner, P.; Weidemann, R.; Witt-Hock, J.: Anforderungen an ein hypermediales Umwelt-Fachinformationssystem, in Riekert, W.-F.; Tochtermann, K. (Hrsg.): Hypermedia im Umweltschutz, 1. WS, Ulm 1998, Metropolis Verlag, Marburg, S. 169 - 173.
- /8/ Reißfelder, M.; Geiger, W.; Weidemann, R.: Dynamische Seiten für ein Altlastenfachinformationssystem auf CD-ROM auf Basis der WWW-Technologie, in Riekert, W.-F.; Tochtermann, K.: (Hrsg.): Hypermedia im Umweltschutz, 1. WS, Ulm 1998, Metropolis Verlag, Marburg, S. 181 - 184.
- /9/ Schuster, R., Färber, J., Eberl, M., Digital Cash, Springer-Verlag, 1997.
- /10/ Trust-Center bilden die Basis für einen sicheren E-Commerce, Computerwoche 28/98, S. 64-65.
- /11/ Weidemann, R.; Geiger, W.; Peter, N.; Reißfelder, M.; Zilly, G.: Richtlinie zur Erstellung WWW-verfügbarer Berichte, Version 4 (Stand: 02.10.1998), <http://www.iai.fzk.de/~weideman/doc/richtl/richtl.htm>
- /12/ SWISH-E, Simple Web Indexing System for Humans - Enhanced, <http://sunsite.berkeley.edu/SWISH-E/>.
- /13/ GNU Public License, Free Software Foundation Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA, <http://www.fsf.org/copyleft/gpl.html>

Anpassung des Personal Data Node an UIS-Nutzungen, insbesondere für die Kernreaktor-Fernüberwachung (KFÜ)

*F. Schmidt, A. Lurk, R. Kopetzky, K. De Marco, Grohmann, J.-E. Schulz, M. Weigele,
Institut für Kernenergetik und Energiesysteme (IKE),
Universität Stuttgart,
Pfaffenwaldring 31,
70569 Stuttgart*

Aktuelle Informationen zu den Arbeiten des Projektes

PDN: http://www.ike.uni-stuttgart.de/~www_wn/projects/PDN/index.html

LCA: http://www.ike.uni-stuttgart.de/~www_wn/projects/STRAT-SV/index.html

KFÜ: <http://www.uis-extern.um.bwl.de/lfu/abt3/kfue/kfue.htm>

1. DER PERSONAL DATA NODE.....	107
2. UNTERSTÜTZUNG DES PDN DURCH AKTIVE OBJEKTE	108
2.1 AKTIVE OBJEKTE – AGENTEN	108
2.2 MIKROARCHITEKTUR EINES AGENTEN	109
2.3 MAKROARCHITEKTUR EINES AGENTENBASIERTEN SYSTEMS	111
2.4 DAS FRAMEWORK FÜR DIE KOMMUNIKATION.....	112
2.4.1 CORBA als Middleware zum Botschaftsaustausch.....	113
2.4.2 KQML als einheitliches Botschaftsformat	113
2.5 DER STRATEGIE SERVICE.....	114
2.5.1 Komponenten des Strategy Service	115
2.5.2 Ontologie und Dienstbeschreibung	115
2.5.3 Beispiel einer Kommunikationssequenz.....	116
2.6 DER REPOSITORY SERVICE	117
2.6.1 Der Workflow-Service	118
3. KOOPERATIVES ARBEITEN IM RAHMEN DER LCA.....	119
3.1 ANBINDUNG VON NETMEETING MITTELS DER SAL	119
3.2 VORTEILE DURCH DIE INTEGRATION VON NETMEETING IN DEN LOGISCHEN CLIENTEN	120
3.3 KONFERENZEN MIT NETMEETING	121
4. DER DIENST ABR IM TEILERNEUERTEN KFÜ	122
4.1 STAND DER ARBEITEN IM PROJEKT TEILERNEUERUNG DES KFÜ BADEN-WÜRTTEMBERG	122
4.2 DER DIENST AUSBREITUNGSRECHNUNG IM KFÜ	123
5. ONTOLOGIE UND WORKFLOW IN DER ABR	125
5.1 ONTOLOGIE	125
5.1.1 Workflow	126
5.1.2 Vorteile der Kombination von Workflow-Ansatz und ontologischer Beschreibung.....	127
5.2 WECHSELWIRKUNGEN ZWISCHEN PROJEKTEN TEILERNEUERUNG KFÜ UND GLOBUS.....	127
6. LITERATUR.....	127

1. Der Personal Data Node

Der *Personal Data Node* (PDN) wurde in den Projekten GLOBUS II bis GLOBUS IV /1/ als zentrale Komponente der Klient-Server-Architektur auf Anwender-Seite entwickelt. Er verknüpft die einzelnen Bereiche des persönlichen Arbeitsplatzes und führt, falls notwendig, die Schritte *Datentransfer*, *Datenentschlüsselung* und *Datenkonvertierung* durch. Er verfügt über eine Komponente zur persistenten Speicherung von Datenobjekten und Methoden, die als *Repository* bezeichnet wurde. Seine *Datenobjekte* sind die auf Datenstrukturen abgebildeten Informationen. *Methoden* sind Skript-Dateien, die das Wissen um die Verarbeitung bestimmter Informationsstrukturen repräsentieren. Die Zuordnung von Datenobjekten und Methoden zu Werkzeugen und Hilfsmodulen wurde durch Konfigurationsdateien auf der Basis von für den Arbeitsplatz gültigen Entscheidungstabellen unterstützt.

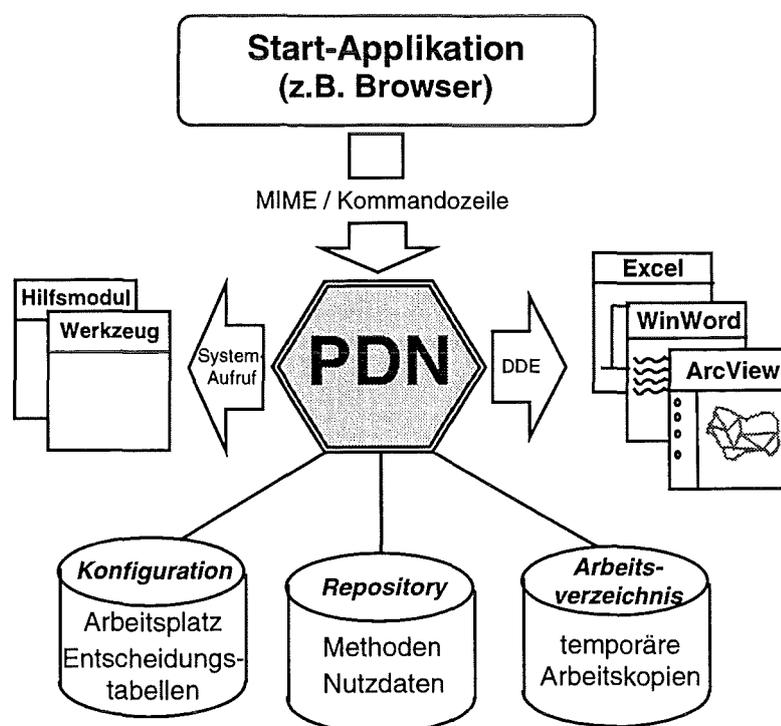


Abb. 1: Der „Personal Data Node“ (PDN) und seine Komponenten

Im Rahmen des Projektes GLOBUS IV wurde mit dem Re-Design des PDN begonnen. Das führte zur Architektur des Logischen Klienten. Basis der LCA (Logische Klienten Architektur) ist das Dienstekonzept. Dienste tauschen dabei Botschaften über eine Middlewarekomponente aus. Dies erlaubt es, die in Abb. 1 gezeigte Architektur so zu verallgemeinern, daß sie unabhängig davon wird, ob Dienste lokal oder remote zur Verfügung stehen. Dies wird im nächsten Abschnitt detaillierter beschrieben. Als Konsequenz dieses Vorgehens wird die LCA nicht nur die Architektur auf Seite des Klienten, sondern kann als Makroarchitektur verteilter, dienstorientierter Systeme angesehen werden.

Eine weitere wichtige Konsequenz aus diesem Ansatz ist, daß der Entwickler auf die alleinige Kontrolle über die von ihm verwendeten Dienste verzichtet. Dienste sind dann gekapselte

Einheiten, die dem Dienstenutzer Funktionalitäten anbieten. Um diese nutzen zu können, benötigt man eine funktionale Beschreibung der Dienste und ihrer Schnittstellen.

Werden Dienste im Rahmen von Simulationen eingesetzt, benötigen sie Daten, die die Verbindung zwischen dem zu simulierenden Weltausschnitt und dem der vom Dienst angebotenen Funktionalitäten zugrundeliegenden Modell herstellen. Um dies zu gewährleisten, müssen die Daten sowohl syntaktisch als auch semantisch beschrieben werden und in sich stimmig sein.

Müssen schließlich mehrere Dienste kombiniert werden, um eine Aufgabe zu erledigen, so muß auch sichergestellt sein, daß die dabei verwendeten Datensätze im Rahmen der gewünschten Aussagegenauigkeit konsistent sind.

All diese Überlegungen zum Einsatz von Diensten sind Konkretisierungen der in der Phase II des Projektes GLOBUS gemachten Empfehlungen. Die Entwicklung des PDN zur LCA und die in der Phase V erfolgte Konkretisierung durch den Einsatz aktiver Objekte (Agenten) setzt daher die in der Rahmenkonzeption 1998 /22/ des UIS-BW gemachten Vorgaben für den Bereich der Simulationsdienste konsequent um. (Anmerkung: Die Rahmenkonzeption 1998 definiert: Dienste sind funktionale Komponenten, die eine charakteristische Aufgabe erfüllen. Sie unterscheidet zwischen systemnahen Diensten, die häufig auftretende Aufgaben erfüllen und Simulationsdiensten, die Daten verschiedenen Ursprungs mit Hilfe von Modellen zu Informationen verarbeiten. Die Rahmenkonzeption empfiehlt Dienste als CORBA-Objekte zu implementieren und schlägt vor, daß ein Strategiedienst die Vermittlung von Diensten und Dienstefolgen übernehmen sollte).

2. Unterstützung des PDN durch aktive Objekte

2.1 Aktive Objekte – Agenten

Objekte im klassischen Sinn kapseln einen internen Zustand, "kommunizieren" über Message Passing und haben Methoden, die Operationen auf ihren Zustand ermöglichen. Als Erweiterung dazu ist ein Agent ein System, das - als Teil einer virtuellen Umgebung - Informationen aus dieser Umgebung aufnimmt und diese zielgerichtet beeinflusst /2/. Ein wesentliches Kriterium für einen Agenten ist seine Autonomie, d.h. er hat die Fähigkeit zu agieren, ist eigenständig und hat die Kontrolle über seinen internen Status. Ein intelligenter Agent ist ein Computersystem, das in der Lage ist, flexibel autonome Aktionen in einer bestimmten Umgebung auszuführen. "Flexibilität" ist dabei charakterisiert durch *reaktives*, *proaktives* und *soziales* Verhalten.

Ein reaktives System paßt sein Verhalten an geänderte Umgebungsbedingungen zur Laufzeit an. Im Gegensatz dazu werden klassische Programme unter Annahme einer garantierten Umgebung "blind" ausgeführt, was in offenen Systemen zu großen Schwierigkeiten führen kann.

Der Bereich des proaktiven Verhaltens beinhaltet das planorientierte Vorgehen unter Berücksichtigung vorhandener Randbedingungen. Dazu zählen das Vorhandensein von Sy-

stemressourcen oder die Verfügbarkeit anderer Dienste für die Durchführung komplexerer Aufgaben.

Soziales Verhalten beinhaltet die Fähigkeit des Systems, sich mit anderen Systemen über eine "Agent Communication Language" (ACL) zu verständigen und mit diesen gegebenenfalls zu kooperieren. Um die Kommunikation zwischen Agenten zu ermöglichen, sind eine Reihe von Rahmenbedingungen zu erfüllen, die in drei Ebenen zusammengefaßt werden können:

- Ein Verteilungssystem für Nachrichten, um den systemweiten Austausch von Botschaften zu ermöglichen (z.B. ein ORB nach CORBA /9/),
- eine Syntax und ein Protokoll für den Nachrichtenaustausch (z.B. Knowledge Query und Manipulation Language KQML /10/),
- ein gemeinsames Verständnis des verwendeten Vokabulars, in dem die eigentlichen Informationen formuliert sind (z.B. über Ontologie).

Bei der Modellierung eines Agentensystems können die "Mikroarchitektur" und die "Makroarchitektur" unterschieden werden. Die Mikroarchitektur beschäftigt sich mit der Modellierung eines einzelnen Agenten, während die Makroarchitektur den Aufbau der Gesellschaftsstruktur der Agenten beschreibt.

2.2 Mikroarchitektur eines Agenten

Das Hauptproblem der Mikroarchitektur liegt in der Umsetzung des reaktiven und proaktiven (planorientierten) Verhaltens. Abb. 2 zeigt die Schichtenarchitektur eines Agenten, in der die Ebenen dargestellt sind, die das Verhalten eines Agenten bestimmen. Die Modellierungsebene (modelling layer) repräsentiert das Wissen über die Makroarchitektur und damit über das Vorhandensein (bzw. dem Nicht-Vorhandensein) anderer Agenten. Sie ist für das soziale Verhalten des Agenten zuständig. Die Planungsebene (planning layer) modelliert das proaktive Verhalten auf Basis vorgegebener Regeln und Fakten aus der Modellierungsebene und externer Informationen, die über Nachrichten oder Sensoren (perception) aufgenommen werden können. Die reaktive Ebene (reactive layer) schließlich behandelt Informationen aus der Umgebung, auf die unmittelbar reagiert werden muß. Das Kontrollsystem (control subsystem) entscheidet, welche Prioritäten den drei Ebenen bei der Ermittlung der geeigneten Aktion (action subsystem) zugeordnet werden.

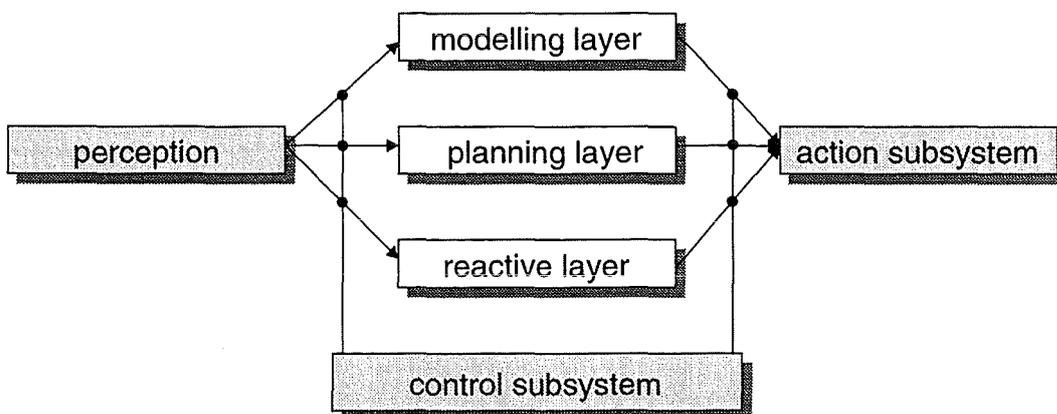


Abb. 2: Mikroarchitektur eines intelligenten Agenten

Die Komponenten dieser Mikroarchitektur sind in Service Agent Layer (SAL) zusammengefaßt.

Die Funktionsweise der SAL läßt sich anhand der Abb. 3 und der Betrachtung der Verarbeitung einer eingehenden Botschaft erläutern. Zunächst sei dazu die Bedeutung des Begriffs „Session“ erklärt. Eine Session ist für die Abarbeitung von zusammengehörenden Nachrichten und daraus resultierenden Aktionen zuständig. Im allgemeinen können in der SAL eines Agenten nach Bedarf Sessionen erzeugt und wieder gelöscht werden.

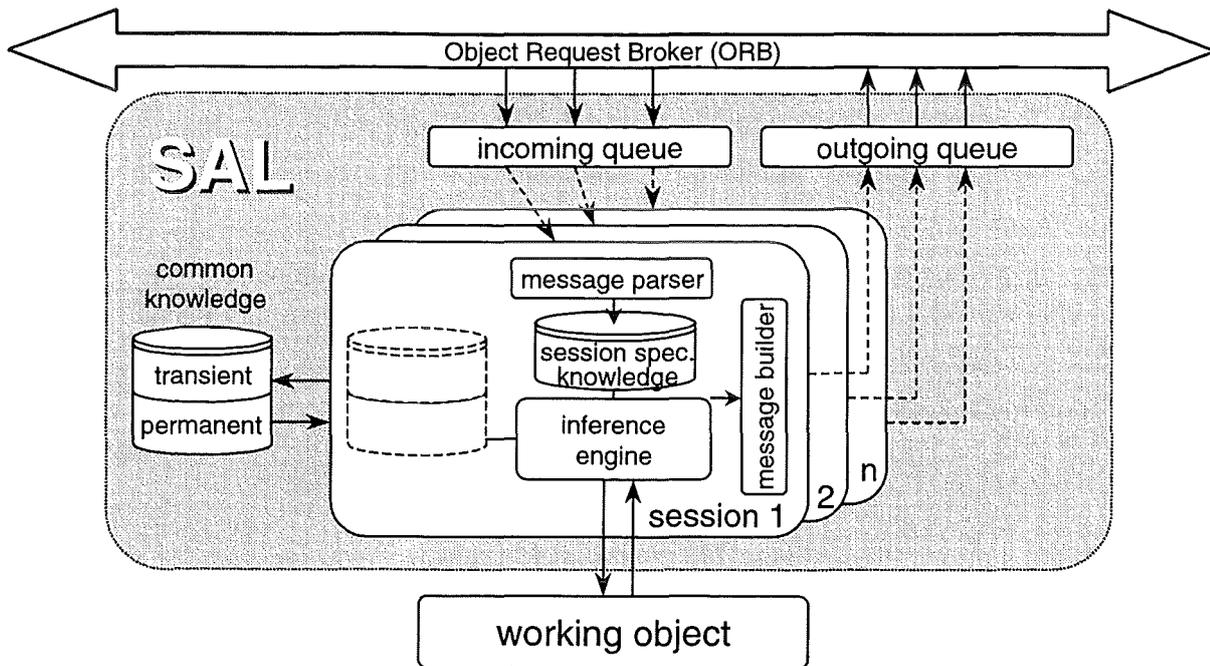


Abb. 3: Architektur des Service Agent Layer (SAL)

Beabsichtigt nun ein anderer Agent im System, eine Nachricht zu schicken, dann ruft er eine Methode des Adressaten auf. Die Übergabewerte der Nachricht werden beim Adressaten in ein Objekt zusammengefaßt und zunächst in eine Warteschlange gestellt. Eine einstellbare Anzahl von Threads bearbeitet die Nachrichten in der Warteschlange. Es wird festgestellt, zu welcher Session die Nachricht gehört (falls die Nachricht keiner Session zugeordnet werden kann, wird eine neue Session angelegt). Diese Session erhält die Nachricht und arbeitet sie ab. In einfachen Fällen (falls die Kommunikationsfähigkeit aufgrund des geringen Leistungsspektrums des Agenten gering ist) reicht dazu eine Entscheidung im if-then-else-Stil aus, normalerweise wird die SAL jedoch über ein Expertensystem verfügen, das aufgrund der gespeicherten Regeln die Nachricht bearbeitet. Mögliche Reaktionen sind etwa das Erzeugen eines „working objects“ (Dienstanspruchnahme), das Starten eines Programms, der Verbindungsaufbau zu einem laufenden Programm, das Absetzen eines Funktionsaufrufs, oder aber das Versenden weiterer Nachrichten, beispielsweise um nach genaueren Angaben zu fragen. Ausgehende Informationen werden in das im System verwendete Format für den Nachrichtenaustausch umgewandelt und in eine Warteschlange gestellt, von wo aus Threads die Zustellung an die Adressaten übernehmen.

Die SAL wurde nach diversen Designpatterns entwickelt, wobei in erster Linie die Trennung von Interface und Implementierung /3/ (auch Bridge-Pattern /4/ oder Letter-Envelope-Pattern

/5/ genannt) zu nennen wäre. Es wird auf einige frei verfügbare Softwarebibliotheken zurückgegriffen (omniORB /6/ als CORBA-Implementierung, OSE /7/ für einige allgemeine Klassen und ACE /8/ für einige spezielle Klassen), wobei Wert darauf gelegt wurde, nur solche Bibliotheken zu verwenden, die plattformübergreifend sind.

2.3 Makroarchitektur eines agentenbasierten Systems

Abb. 4 zeigt die Makroarchitektur des Logischen Clienten unter Verwendung des Agentenansatzes. Sein Kommunikations-Backbone bildet die "Common Object Request Broker Architecture" (CORBA) zusammen mit dem "Service Agent Layer" (SAL). Die Systemkomponenten sind entsprechend der CORBA-Spezifikation /9/ in "common facilities", "application objects" und "object services" eingeteilt. (Durch die SAL werden alle Dienste des Systems zu aktiven Objekten oder Agenten. Im Sinne der Rahmenkonzeption UIS 98 sprechen wir aber im Folgenden weiter von Diensten und unterscheiden nicht, ob diese aktiv oder passiv sind).

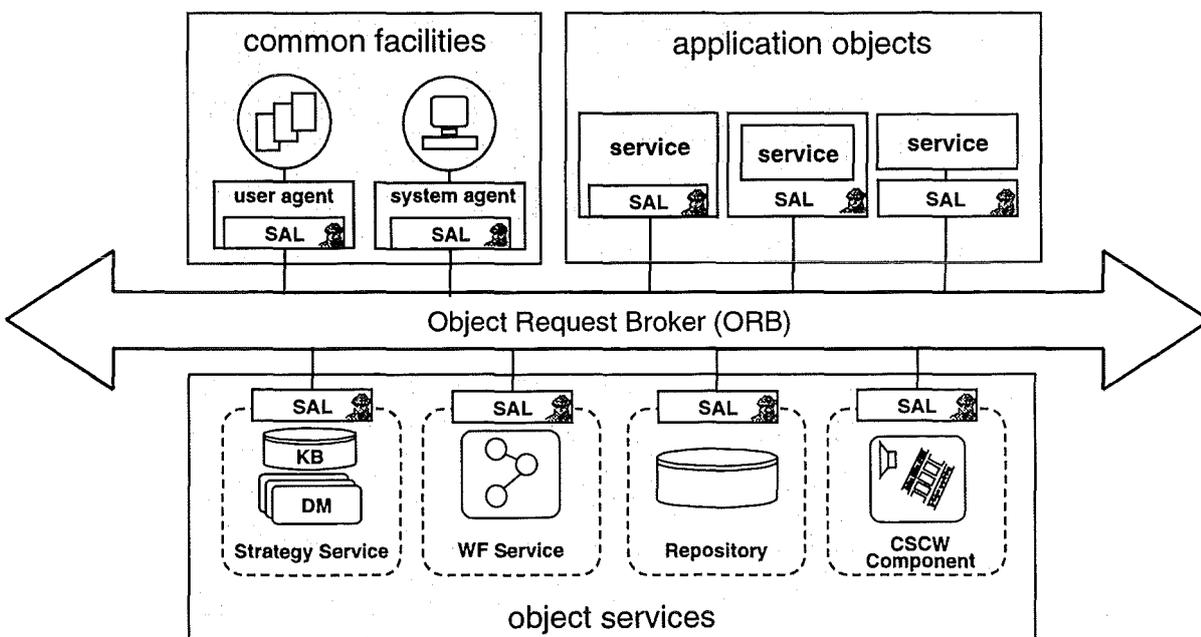


Abb. 4: Architektur des Logischen Clienten unter Verwendung der SAL

Zu den "common facilities" gehören Dienste wie der Systemagent oder der Anwenderagent, die den Zugriff zu Systemressourcen verwalten und die Schnittstelle zum Anwender bilden. "Application objects" sind die im System verfügbaren Dienste (Service-Agent). Die Art, Anzahl und Komplexität der einzelnen Dienste variieren entsprechend dem jeweiligen Anwendungsfall und den Bedürfnissen der Anwender. Zu den "object services" zählen die sogenannten Systemdienste, die in jedem Logischen Clienten vorhanden sind und eine Reihe grundlegender Aufgaben wahrnehmen:

- Die *CSCW-Komponente* unterstützt den Anwender mit Techniken des Kooperativen Arbeitens (Video, Audio, "Application Sharing", ...) und ermöglicht so die Kommunikation und Kooperation zwischen Anwendern bzw. zwischen Anwender und Experte. Sie wird im Abschnitt über kooperatives Arbeiten beschrieben.

- Das (verteilte) *Repository* ist ein allgemeiner, virtueller Speicher für alle Arten von Daten. Das Repository kapselt die Problematik des Transports großer Datenmengen zwischen den einzelnen Diensten. Dieser Systemdienst wird extra erläutert.
- Die Aufgabe des *Workflow (WF) Service* ist den Aufgaben eines Workflow-Engines in klassischen Workflow Management Systemen (WFMS) vergleichbar. Der Unterschied liegt darin, dass der WF Service dynamisch (zur Laufzeit) modellierte Prozeßketten verarbeitet, die ihm in einem standardisierten Austauschformat zur Ausführung übergeben werden. Eine kurze Beschreibung findet man bei der Beschreibung wichtiger Dienste.
- Der *Strategy Service* ist ein Agent für die dynamische Modellierung von Prozeßketten. Er kombiniert vorhandene Dienste zu geeigneten Prozeßketten, über die komplexe Aufgaben durchgeführt werden können. Dabei berücksichtigt er die Anforderungen des jeweiligen Anwenders. Der Strategy Service wird in einem eigenen Abschnitt ausführlich erläutert.

2.4 Das Framework für die Kommunikation

Der Kommunikation zwischen den Agenten kommt eine entscheidende Bedeutung zu, weil sie erst die Grundlage für "soziales" Verhalten bildet. Zu ihrer Vereinfachung gibt es im Logischen Clienten ein Kommunikationsframework, das die Funktionalitäten zur Verfügung stellt, die zur Kommunikation zwischen den Agenten benötigt werden. In Analogie zum Briefverkehr lassen sich drei Ebenen aufstellen, die abgedeckt werden müssen:

- So wie es ein System zur Verteilung von Briefen gibt (Briefkästen, Postboten, ...), gibt es im Framework eine Ebene, die für den *physikalischen Transfer* der Information zuständig ist. Dafür wird CORBA /9/ eingesetzt.
- Im Briefverkehr gibt es feste Regeln, wie Briefe auszusehen haben. Dazu gehört ein Feld mit der Adresse des Empfängers und des Senders. Auf dieser Ebene der *Syntax* und des *Protokolls* sind darüber hinaus Vereinbarungen anzusiedeln, wie auf bestimmte Briefsendungen reagiert werden soll (senden eines Antwortbriefes, ...). Hier wird auf die Knowledge Query and Manipulation Language (KQML) /10/ zurückgegriffen.
- Eine für erfolgreiche Kommunikation unerläßliche Voraussetzung ist die Einigung auf der *semantischen Ebene*. Die verwendeten Vokabeln müssen auf allen Seiten gleich verstanden werden.

In einem Multi-Agenten-System reicht es nicht aus, die verwendeten Datenobjekte wie in einem Glossar zu definieren. Es ist notwendig, die Datenobjekte, ihre Zusammenhänge, die Regeln und Abhängigkeiten zwischen ihnen formal und maschinenlesbar zu beschreiben (Ontologie). Die Erstellung von Ontologien ist aufwendig, bringt aber erhebliche Vorteile mit sich. Dies wird im nachfolgenden Kapitel über den Strategie-Service noch genauer besprochen werden. Der Strategie-Service kann auf der Basis einer Ontologie ermitteln, welche Dienste wie zusammenarbeiten müssen, um komplexe Aufgaben zu lösen.

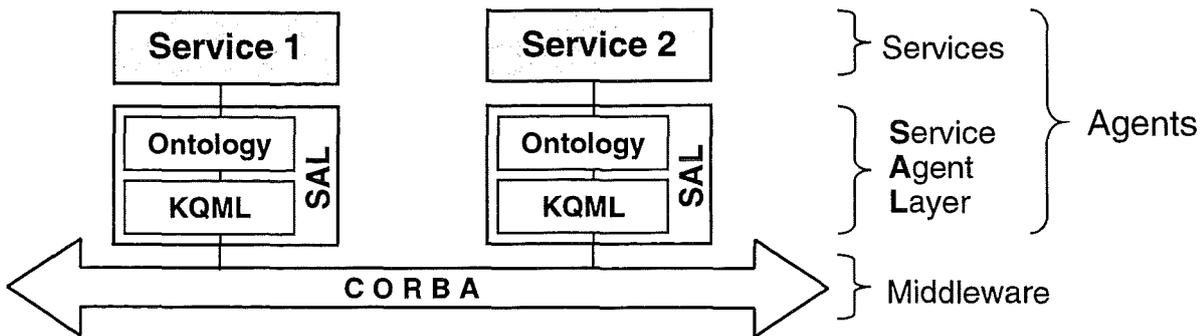


Abb. 5: Verschiedene Ebenen der Kommunikation

In Abb. 5 sind diese drei Ebenen schematisch dargestellt. Die Dienste benötigen zu ihrer Kommunikation eine Einigung bezüglich der Ontologie, sie verwenden Botschaften auf eine einheitliche Art gemäß der Syntax und dem Protokoll, das durch KQML vorgeschlagen wird, und der physikalische Austausch erfolgt mit Hilfe von CORBA als Middleware. Die Ebene der Ontologie und der KQML werden zu einer allen Agenten im System gemeinsamen Schicht zusammengefaßt, der Service Agent Layer (SAL). Diese hat ihren Namen daher, daß sie Services mit der nötigen Funktionalität versieht, um aus ihnen Agenten zu machen.

Im Folgenden wird näher auf CORBA und KQML eingegangen, während das daran anschließende Kapitel Einzelheiten zur Architektur und Implementierung der SAL enthält.

2.4.1 CORBA als Middleware zum Botschaftsaustausch

CORBA (Common Object Request Broker Architecture) ist ein Standard, der von der Open Management Group /9/ erarbeitet und von verschiedenen Seiten implementiert wurde. Er erlaubt es, Methoden von nicht lokalen Objekten aufzurufen und kapselt sowohl die Datenübertragung als auch die Verbindung zu Prozessen und darin instanziierten Objekten. CORBA wurde für alle gängigen Plattformen und viele Programmiersprachen entwickelt, und es möglich, beispielsweise von einem C++-Programm über CORBA auf ein entferntes Java-Objekt zuzugreifen. Dies alles macht CORBA für verteilte Agentensysteme sehr geeignet. Jeder Agent hat eine CORBA-Methode, über die andere Agenten Botschaften absetzen können.

2.4.2 KQML als einheitliches Botschaftsformat

Bezüglich der Syntax und des Protokolls beim Austausch von Botschaften ist bereits viel Arbeit geleistet worden. Es gibt Vorschläge für Agentensprachen, die vom Inhalt der übertragenen Information unabhängig sind /11/, KQML (Knowledge Query and Manipulation Language /10/ ist ein für unsere Anwendungszwecke geeigneter Standard zum Austausch von Botschaften. Eine typische KQML-Botschaft in C++ könnte etwa so aussehen:

```
askIf(sender, receiver, inReplyTo, replyWith, language, ontology, content);
```

Das bedeutet, daß *sender* eine Frage an den *receiver* stellt, auf die er eine unmittelbare Antwort erwartet. Die Antwort des *receiver* enthält den Wert von *replyWith* der Frage im Feld *inReplyTo*. Das ermöglicht eine Zuordnung von eingehenden Antworten auf Fragen; *langua-*

ge spezifiziert die Sprache, in der der *content* geschrieben ist (etwa C++), während *ontology* angibt, in welchem Zusammenhang der *content* steht.

Der Name der o.a. Funktion, *askIf*, ist eine sog. Performative. KQML bietet ein System von Performativen für unterschiedliche Sprechakte (weitere Beispiele sind *tell*, *askOne*, *askAll*, *subscribe*, *delete*, *advertise*, *deny*, ...) und das zugehörige Protokoll, d.h. welche Botschaften auf welche Performativen erwartet werden. Das System der Performativen stützt sich auf die Sprechakttheorie und scheint für den logischen Clienten ausreichend zu sein.

Darüber hinaus gibt es Implementierungen von KQML, die auch den physikalischen Transfer der Botschaften durchführen. Anstelle der KQML-Implementierungen verwenden wir jedoch CORBA, während wir uns bezüglich der Syntax und des Protokolls an KQML orientieren.

2.5 Der Strategie Service

In diesem Abschnitt werden die Konzepte für den Strategy Service als Planungsagent der Architektur beschrieben.

Wissen und Planung

Für die Planung komplexer Vorgänge müssen im wesentlichen zwei Voraussetzungen gegeben sein: Eine Komponente, die die Planung durchführt (i.A. regelbasiertes System) und das Wissen über die für die Planung notwendigen Faktoren (verfügbare Dienste, Ressourcen, ...). Es lassen sich prinzipiell drei Fälle unterscheiden:

Fall I: Die Planungskomponente und das Wissen liegen bei den einzelnen Agenten. Das führt in einem offenen und dynamischen System zu einem erheblichen Kommunikationsaufwand, da jeder Agent von jeder Änderung in Kenntnis gesetzt werden müßte. Auch erscheint die redundante Speicherung der umfangreichen Informationen nicht zweckmäßig.

Fall II: Für die Verwaltung des Wissens wird ein zentraler "Yellow-Page Service" modelliert, der einzelnen Agenten Auskunft über Verfügbarkeit und Eigenschaften anderer Agenten geben kann. Die Planung erfolgt bei dem jeweiligen Agenten. Das entspricht der Vorgehensweise bei Einsatz eines CORBA-Traders oder die Verwendung von "communication facilitators" /13/. Für Anwendungsbereiche, in denen jeweils ein bestimmter Dienst vermittelt werden soll, scheint dieser Ansatz geeignet. Im Bereich von Informationssystemen, in denen oft 10 und mehr Dienste für die Ausführung komplexer Aufgaben zusammenwirken müssen, ist wie bei Fall I der Kommunikationsaufwand zwischen Agent und "Yellow-Page Service" sehr hoch.

Fall III: In dem in der Architektur des Logischen Clienten konzipierten Strategy Service sind das Wissen über die verfügbaren Dienste und die Planungskomponente für die Prozeßmodellierung zusammen gefaßt. Dabei ist jeweils ein Strategy Service für einen Logischen Clienten "zuständig". Um Dienste nutzen zu können, die in anderen Logischen Clienten angeboten werden, kann er mit den jeweiligen Strategy Services kooperieren. Die Strategien, die hierbei verwendet werden, entsprechen im wesentlichen den Trading-Strategien /9/ in verteilten Systemen und sind nicht Gegenstand dieser Arbeit.

2.5.1 Komponenten des Strategy Service

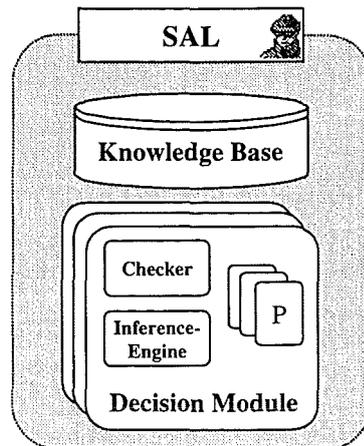


Abb. 6: Bestandteile des Strategie Service: SAL, Knowledge Base und Decision Module

Abb. 6 zeigt die Komponenten eines Strategy Service. Neben der SAL für die Kommunikation sind die wesentlichen Bestandteile eine "Knowledge Base" (KB) und sogenannte "Decision Modules" (DM) /14/. Die Knowledge Base speichert Informationen zu verfügbaren Diensten, Benutzerprofile zu den jeweiligen Anwendern und "gelernte", also bereits modellierte und validierte Prozeßketten. Ein Decision Module ist die eigentliche Planungskomponente und besteht aus einem "Inference Engine", einem "Checker Module" und der Liste mit modellierten Prozeßketten. Auf Basis der Informationen aus der Knowledge Base modelliert der Inference Engine geeignete Prozeßketten, deren Gültigkeit von dem Checker Module überprüft wird. Nach dieser Validierung werden die Prozeßketten an den Auftraggeber gesendet, der dann den WF Service mit der Abarbeitung beauftragt.

2.5.2 Ontologie und Dienstbeschreibung

Ein detailliertes Wissen über ein System, seine Schnittstellen und Komponenten ist von entscheidender Bedeutung für das Verständnis der Zusammenhänge und die korrekte Interpretation der gelieferten Informationen. Verwendet man den Begriff "Ontologie" aus dem Bereich der Künstlichen Intelligenz (AI), so bezeichnet man damit ein Modell eines Teils der Welt. Der hier favorisierte Ansatz ist die Beschreibung über eine "first-order logic" (FOL) Sprache (z.B. KIF /15/ oder CLIPS /16/). In FOL wird die Welt in Objekten (objects) beschrieben, die über eine eigene Identität verfügen und Eigenschaften (properties) haben, die sie charakterisieren. Zwischen diesen Objekten existieren eine Reihe von Beziehungen (relations), die funktionale Beziehungen (functions) sein können, also Beziehungen, in denen zu einem Eingabewert genau ein Ausgabewert gehört /17/.

Für die Auswahl geeigneter Dienste ist die Beschreibung der Dienste und ihrer Schnittstellen von entscheidender Bedeutung. Durch die Beschreibung

- ist es dem Strategy Service möglich, aus den vorhandenen Diensten eine geeignete Prozeßkette zu modellieren,
- ist es dem WF Service möglich, Dienstaufträge mit den geeigneten Eingabewerten zu parametrisieren und

- dem Visualisierungsdienst bzw. dem Anwender möglich, die Ausgabedaten korrekt zu interpretieren.

Die Beschreibung erfolgt durch Metainformationen, die auf Basis der Systemontologie den Eingaberaum, den Ausgaberaum und die Funktionalität des Dienstes beschreiben.

Um geeignete Prozeßketten zu identifizieren, wird in Anlehnung an Russell et al /17/ die Nützlichkeit (utility) eingeführt. Für die Berechnung der Nützlichkeit werden drei Parameter herangezogen. Die Zeit, die ein Prozeß benötigt, die Kosten eines Prozesses und die Qualität der Informationen. Zu Prozeßzeit und Prozeßkosten zählen zum einen die Zeit und die Kosten, die von den Diensten einer Prozeßkette benötigt bzw. verursacht werden und Zeit und Kosten für den Transport und die Speicherung der gewonnenen Informationen. Bei der Qualität ist der Dienst mit dem geringsten Qualitätsfaktor ausschlaggebend für die Qualität der Gesamtinformationen einer Prozeßkette. Beispielsweise sind Schätzwerte qualitativ geringer einzustufen als gemessene Werte.

Für die Beschreibung der Prozeßketten wird das von Lee et al /18/ entwickelte Process Interchange Format (PIF) verwendet, eine anwendungsunabhängige Prozeßbeschreibung, deren Kern aus den Elementen "Aktivität", "Objekt", "Zeitpunkt" und "Beziehung" aufgebaut ist.

2.5.3 Beispiel einer Kommunikationssequenz

In diesem Abschnitt soll an einem praktischen Beispiel die grundsätzliche Vorgehensweise bei Einsatz des Strategy Service erläutert werden. Abb. 7 zeigt in fünf Schritten den Ablauf der Kommunikation zwischen den beteiligten Agenten. Dabei sind A, B und C Dienste, S ist der Strategy Service und D repräsentiert zur Vereinfachung den Anwender und den WF Service.

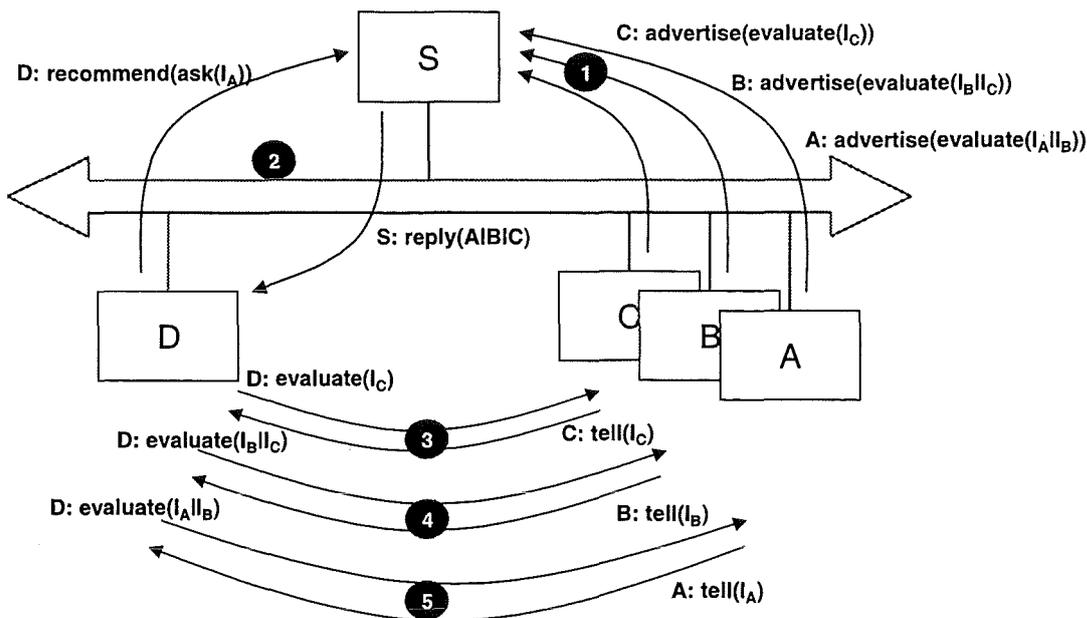


Abb. 7: Kommunikationsschema des Strategie Service

Alternativ zu dem diskutierten Ablauf könnte Service B dem Repository-Service bekanntgeben, daß er auf bestimmte Dateien wartet. Sobald diese in das Repository eingechekkt worden sind, benachrichtigt der Repository-Service B davon.

Die Variante setzt allerdings voraus, daß sichergestellt ist, daß die entsprechenden Dateien auch tatsächlich erzeugt werden. Damit kommen wir in den Bereich der Organisation von Arbeitsabläufen (Workflows). Im folgenden Abschnitt wird der Workflow-Service beschrieben, der für die Bearbeitung von Workflows, der Koordination von Diensten und dem Weiterreichen von Repositories und anderen Daten zuständig ist.

2.6.1 Der Workflow-Service

Der Workflow-Service erhält vom Strategie-Service eine Workflow-Beschreibung, in der festgelegt ist, welche Aktivitäten (Inanspruchnahme von Diensten) in welcher Reihenfolge zu starten sind, welche Daten und Dateien dabei erzeugt und von welchen Aktivitäten welche Daten benötigt werden. Aufgrund dieser Workflow-Beschreibung ermittelt der Workflow-Service, für welche Aktivitäten bereits alle erforderlichen Eingangsdaten vorliegen und stößt diese mit einer SAL-Botschaft an. Sobald der Dienst seine Arbeit getan hat, meldet er den Erfolg (oder auch Mißerfolg) seiner Tätigkeit gemeinsam mit den erzeugten Daten zurück. Diese Daten stehen nun weiteren Aktivitäten zur Verfügung. Wenn alle notwendigen Aktivitäten durchgeführt sind, meldet der Workflow-Service den Erfolg (oder auch Mißerfolg) an seinen Auftraggeber zurück.

Die Verwendung von Workflow-Beschreibungen und dem Workflow-Service hat den entscheidenden Vorteil, daß die einzelnen Dienste voneinander nichts zu wissen brauchen. Es ist nicht notwendig, daß die einzelnen Dienste von sich aus entsprechende Nachfolger im Workflow ermitteln und kontaktieren. So können die Dienste hinsichtlich ihres Wissens um andere Dienste relativ einfach gehalten werden. Zusätzlich hat die Verwendung von Workflow-Beschreibungen den Vorteil, daß neuen Anforderungen, dem Vorhandensein neuer Dienste oder dem Wegfall bestehender Dienste einfach dadurch entsprochen werden kann, daß die entsprechenden Workflow-Beschreibungen geändert werden. Der Strategie-Service hat in diesem Zusammenhang die Aufgabe, zur Laufzeit Workflow-Beschreibungen auf Basis der gerade verfügbaren Dienste zu generieren. So ist gewährleistet, daß komplexe Aufgaben in einer sich ständig verändernden Umgebung durchgeführt werden können, und gleichzeitig können durch die zentrale Steuerung und Kontrolle durch den Workflow-Service auch Probleme in einem Arbeitsablauf schnell erkannt und mit Hilfe des Strategie-Service behoben werden.

Abb. 9 enthält die Illustration eines Workflows, dessen Aktivität A1 aus einem Subworkflow besteht. Die Verwendung von Subworkflows ermöglicht es, komplexe Workflows hierarchisch und übersichtlich zu erstellen. Gleichzeitig kann im Fehlerfall folgendermaßen vorgegangen werden: Wenn die Aktivität A2 fehlschlägt, dann wird beim Strategie-Service nach einem Ersatzworkflow für A2 gefragt (also nach einem Subsubworkflow). Gelingt das, wird der Unterworkflow zu Ende geführt. Schlägt jedoch auch dieser Ersatzworkflow fehl (oder wird keiner gefunden), dann scheitert der Unterworkflow und damit die Aktivität A1 des oberen Workflows. Nun wiederholt sich das Spiel auf der höheren Ebene: Für A1 wird ein Ersatzworkflow gesucht. Wenn einer gefunden wird und funktioniert, kann der Workflow abgeschlossen werden. Andernfalls scheitert auch der Workflow der oberen Ebene.

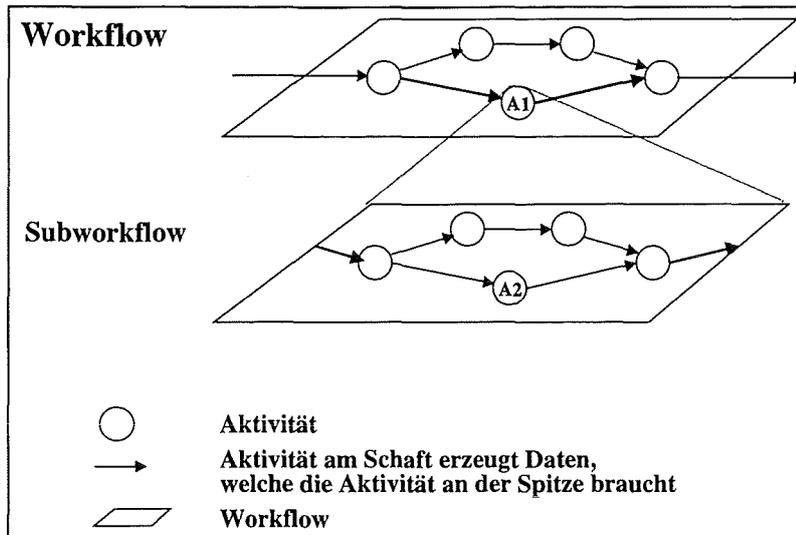


Abb. 9: Eine Aktivität kann ein Subworkflow sein. Komplexe Workflows lassen sich so hierarchisch gliedern.

3. Kooperatives Arbeiten im Rahmen der LCA

Um den logischen Clienten um Funktionalitäten des Computer Supported Cooperative Work (CSCW) zu erweitern, wurde das System NetMeeting /19/ der Firma Microsoft als Systemdienst (CSCW-Dienst) in die Architektur des logischen Clienten integriert. Abb. 10 zeigt schematisch diese Integration, auf die im Folgenden eingegangen wird.

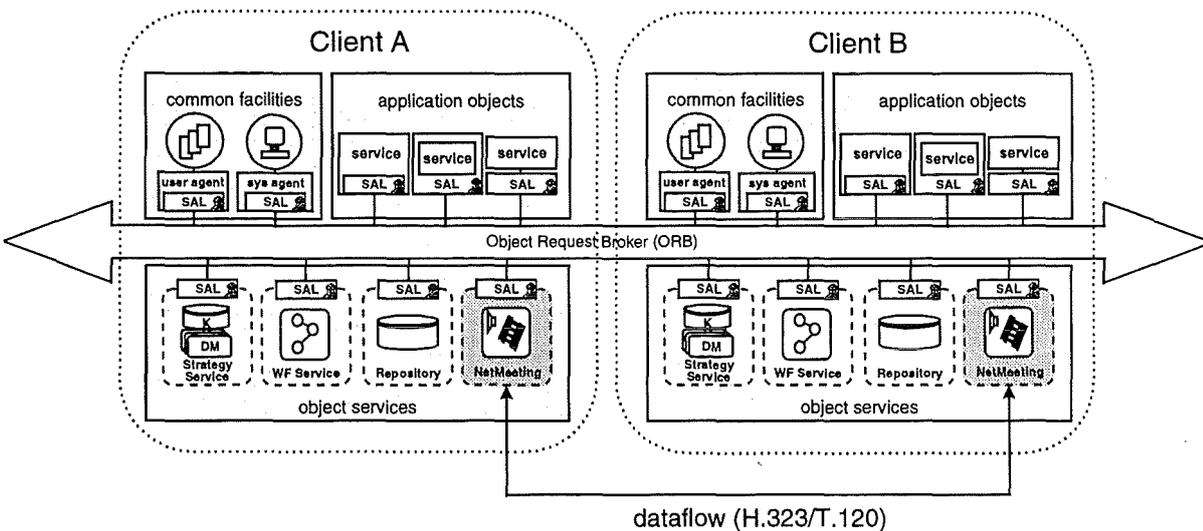


Abb. 10: Integration von NetMeeting in die Architektur des logischen Clienten

3.1 Anbindung von NetMeeting mittels der SAL

Für die Schnittstelle zwischen NetMeeting und der Middleware wird, wie beschrieben, die SAL eingesetzt. Um Informationen zwischen der SAL und NetMeeting auszutauschen, wird

der Software Developer's Kit (SDK) von NetMeeting verwendet, der somit als Schnittstelle zwischen der SAL und NetMeeting fungiert.

Ähnlich wie beim Repository-Dienst muß man zwischen dem Informations- und dem Datenfluß unterscheiden. Der Informationsfluß verläuft über CORBA und beinhaltet die Kommunikation sowohl innerhalb des logischen Clienten als auch zwischen mehreren logischen Clienten eines Netzes (z.B. Client A und B aus Abbildung 10). Beim Datenfluß im Rahmen von NetMeeting handelt es sich zum einen um den Informationsaustausch von Audio- und Videodaten (H.323) und zum anderen um den Datenaustausch über Application Sharing, Whiteboard und Filetransfer (T.120). Wie in Abbildung 10 zu sehen ist, verläuft der Datenfluß wie üblich direkt zwischen den beteiligten Konferenzteilnehmern und nicht wie beim Informationsfluß über CORBA ab.

3.2 Vorteile durch die Integration von NetMeeting in den logischen Clienten

Durch die Aufnahme von NetMeeting /19/ in den logischen Clienten wird NetMeeting mit allen benötigten Attributen im lokalen Strategie-Service bekanntgemacht. Hierzu wird die Beschreibung des Systemdienstes NetMeeting in der Konfigurationsdatenbank des Strategie-Service abgelegt. Zu diesen Attributen gehören der Name des Teilnehmers (Vor- und Nachname) und die IP-Adresse des Clientrechners.

Wenn nun beispielsweise Client A Kontakt mit Client B aufnehmen möchte, muß er sich keine Gedanken machen, unter welcher IP-Adresse er zu erreichen ist. Es reicht aus, wenn er ihn über das Attribut Name anspricht. Der lokale Strategie-Service hat dann die Aufgabe, anhand des Namens die zum Aufbau einer Konferenz benötigten Informationen zu beschaffen. Hierzu kommunizieren die Strategie-Services der logischen Clienten untereinander über die Middleware CORBA (Informationsfluß).

Wenn alle Informationen zur Verfügung stehen, kann die Konferenz gestartet werden, wobei der Datenfluß, beispielsweise Audio- und Videodaten, nicht mehr über die Middleware CORBA, sondern direkt zwischen den beteiligten Konferenzteilnehmern ausgetauscht werden. Zu diesem Datenfluß zählen sowohl Audio und Video (H.232) als auch der Datenaustausch mittels Application Sharing, Chat-Tool, Whiteboard und Filetransfer (T.120).

Um über NetMeeting Kontakt mit einem Partner aufzunehmen, muß beim Aufruf NetMeeting beim Empfänger laufen. Durch die Integration von NetMeeting in den logischen Clienten ist dies nicht mehr erforderlich. Sobald ein Anruf ankommt, wird NetMeeting automatisch durch die SAL gestartet. Durch entsprechende Abfragen kann der Empfänger dieses automatische Starten unterbinden. Befindet sich der Empfänger nicht an seinem Arbeitsplatz bzw. reagiert er nicht auf den Anruf, so wird der Anruf nach einer gewissen Wartezeit abgebrochen und dem Sender eine entsprechende Nachricht, beispielsweise „Der Konferenzpartner ist zur Zeit nicht zu erreichen“, zurückgeschickt.

3.3 Konferenzen mit NetMeeting

Parallel zur Einführung der CSCW-Arbeitsplätze beider GLOBUS-Partner haben wir auch andere Projektpartner bei der Installation und Einführung der CSCW-Techniken unterstützt. Dadurch ist es gelungen, ein Netzwerk für kooperatives Arbeiten aufzubauen, das unsere täglichen Arbeiten immer wieder erleichtert (Abb. 11).

Die Basis des Netzes bildet der Konferenzserver MeetingPoint aus dem Hause White Pine /20/. MeetingPoint ist ein kommerzielles Produkt und wurde von der Firma Innovative Software Solutions GmbH (ISS) /21/ aus Mainz zur Verfügung gestellt. Mittels MeetingPoint können Konferenzen zwischen mehreren (auch unterschiedlichen) CSCW-Systemen durchgeführt werden, vorausgesetzt, die CSCW-Systeme unterstützen den H.323-Standard. Die Anzahl der Konferenzteilnehmer hängt ab von der Anzahl der Lizenzen, die man mit dem Server erwirbt. Uns stehen 10 Lizenzen von ISS zur Verfügung, d.h. man kann eine Konferenz mit 10 Teilnehmern führen oder auch zwei Konferenzen gleichzeitig mit jeweils 5 Teilnehmern. MeetingPoint wird von White Pine für die beiden Plattformen Windows NT 4.0 und Sun Solaris 2.5 zur Verfügung gestellt.

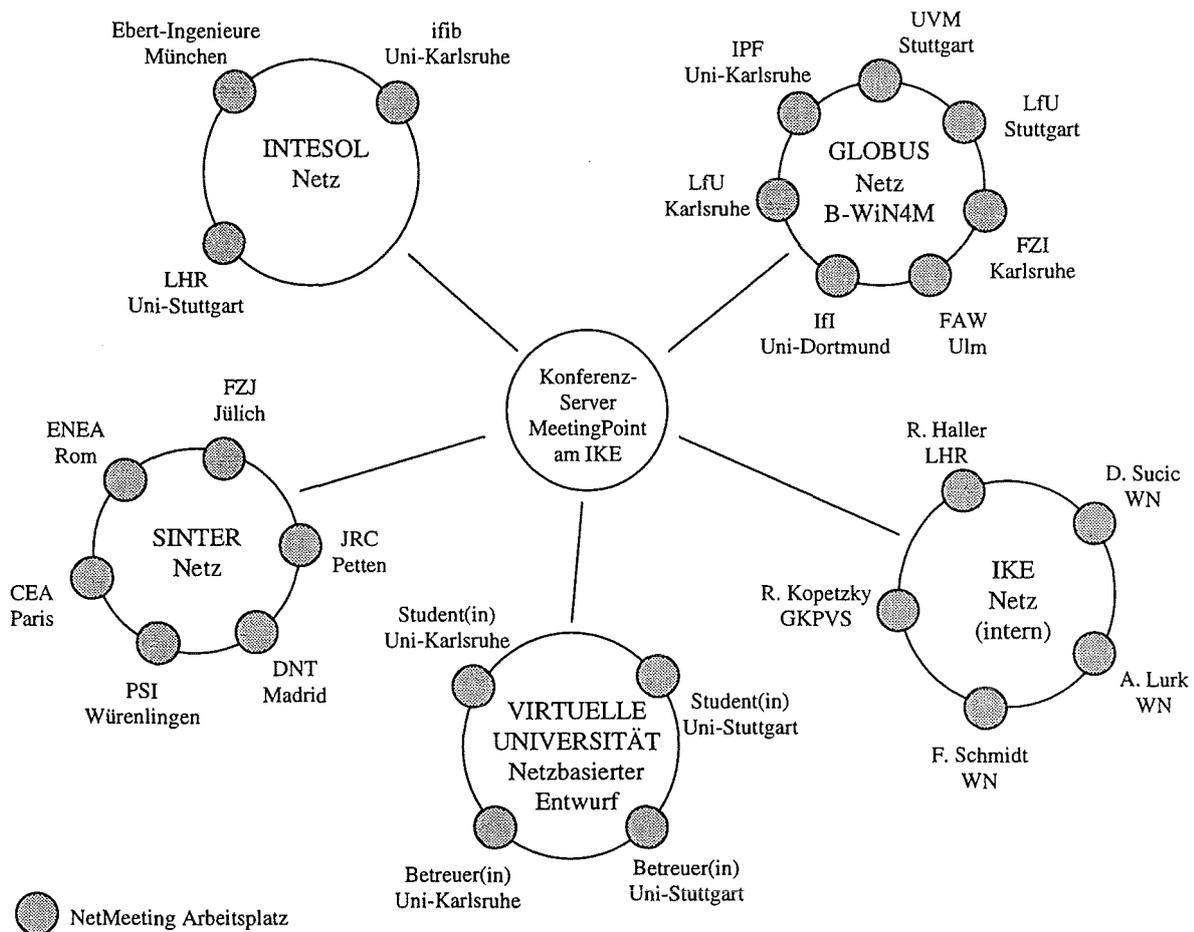


Abb. 11: Das Konferenznetz des IKE (CSCW-Netz).

4. Der Dienst ABR im teilerneuerten KFÜ

4.1 Stand der Arbeiten im Projekt Teilerneuerung des KFÜ Baden-Württemberg

Wachsende Anforderungen an die Kernreaktor-Fernüberwachung (KFÜ) führten zur Notwendigkeit einer Erneuerung des bestehenden Systems. Dabei wurde besonderen Wert auf eine zukunftsichere, wartbare und erweiterbare Architektur mit klar abgegrenzten Komponenten und entsprechenden Schnittstellen zwischen diesen Komponenten gelegt. Dies führte zu einem Ansatz wie er in Abbildung 12 dargestellt ist

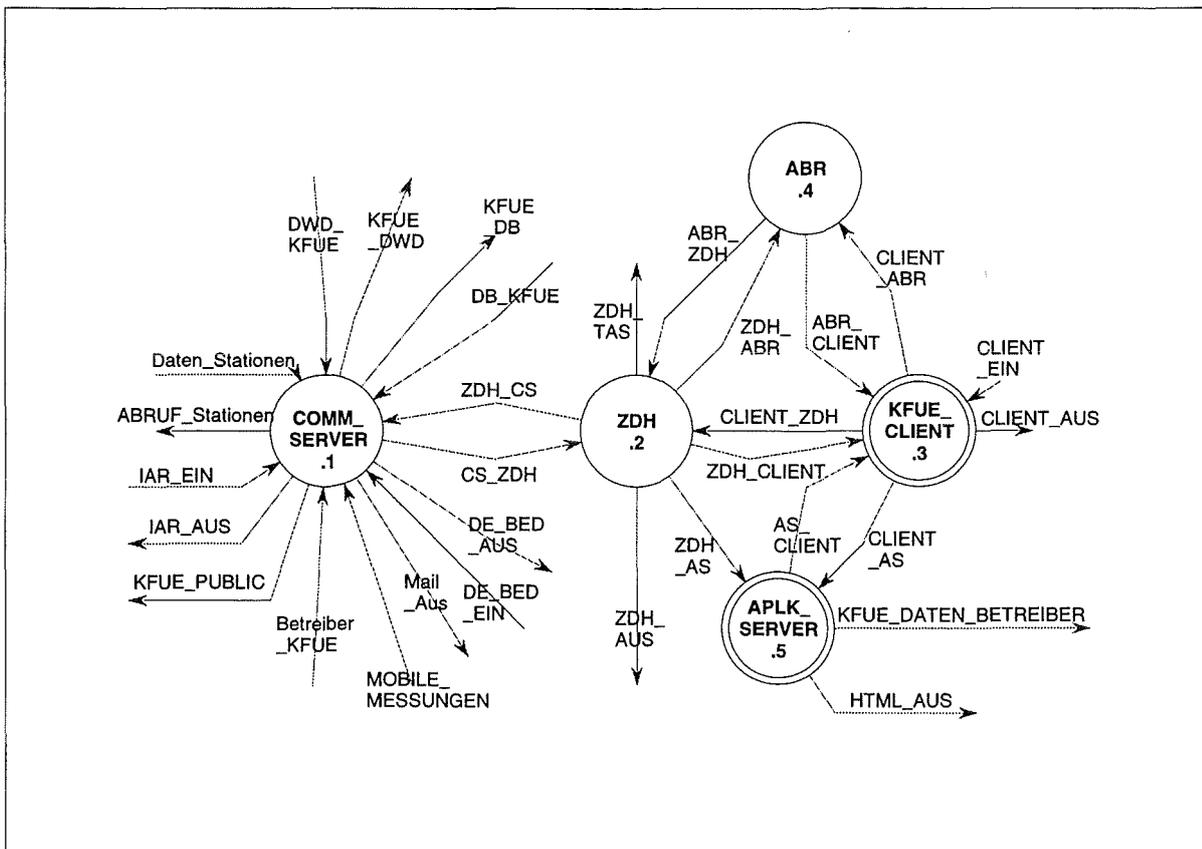


Abb. 12: Komponenten des teilerneuertem KFÜ Baden-Württemberg

Das System besteht dabei aus einem Kommunikationsserver, welcher für den Datenaustausch mit externen Datenquellen und Senken zuständig ist, einer zentralen Datenhaltung, den KFÜ-Clients, welche die Benutzeroberfläche für die Bediener darstellen, dem Ausbreitungsserver und den Applikationsservern für die lokale Datenbereitstellung an verteilten Standorten. Diese Komponenten kommunizieren über eindeutig definierte Schnittstellen.

Im Rahmen der Projektarbeiten wurden die benötigten Funktionalitäten identifiziert und die Schnittstellen spezifiziert. Auf der Basis dieser Spezifikation erfolgte die Definition der Schnittstellen und der Entwurf von Kommunikationsserver, zentraler Datenhaltung (ZDH), Applikationsserver und KFÜ-Clients. Dies erfolgte nach den Vorgaben des Landessystem-

konzeptes und der UIS-Rahmenkonzeption 1998 unter Verwendung von Methoden der Objektorientierung und Techniken aus dem Bereich relationaler Datenbanken. Der Entwurf des Ausbreitungsservers ABR ist im Entstehen. Der bisher erreichte Stand wird im nächsten Abschnitt beschrieben. Bei den Komponenten Kommunikationsserver, zentraler Datenhaltung, Applikationsserver und KFÜ-Clients wurde inzwischen mit der Implementierung begonnen.

4.2 Der Dienst Ausbreitungsrechnung im KFÜ

Der Dienst Ausbreitungsrechnung (ABR) nimmt im KFÜ-BW eine zentrale Stellung ein. Ursprünglich mit der Methode der strukturierten Analyse konzipiert, hat sich im Berichtsjahr gezeigt, daß eine Umsetzung mit den in der Spezifikation geforderten Eigenschaften unter der LCA besonders effektiv erfolgen kann. Deshalb wurde ein Re-Design des Dienstes ABR mit Hilfe des Case-Tools Rational Rose und unter Verwendung der LCA durchgeführt. Abb. 13 zeigt das Ergebnis.

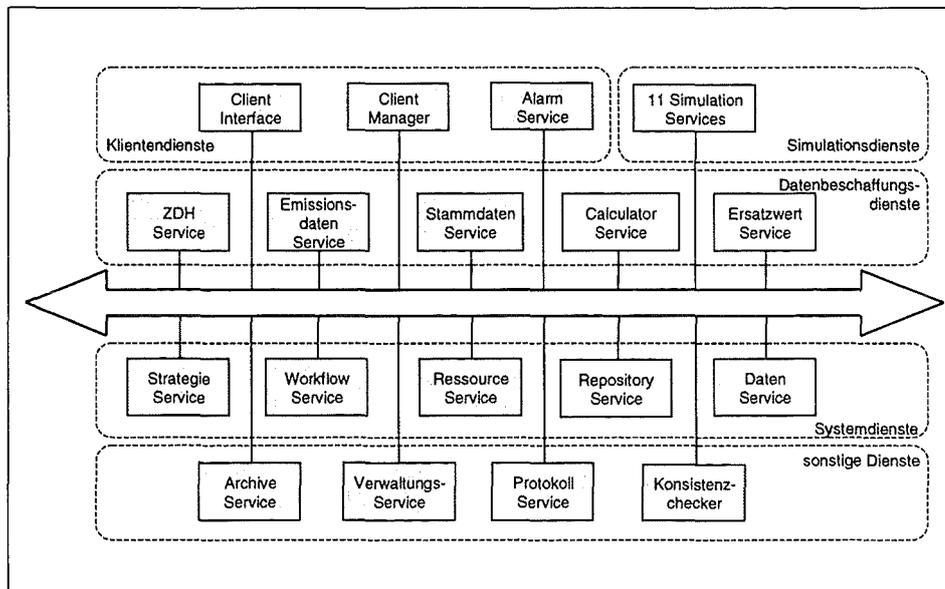


Abb. 13: Architektur der ABR

Die Dienste stellen die Funktionalitäten der in früheren Projektphasen spezifizierten Komponenten Rechnungsdurchführung, Arbeitsdaten-, Sessions- und Ergebnismanagement zur Verfügung. Im Berichtszeitraum wurden folgende Dienste auf der Basis objektorientierter sowie regelbasierter Methoden mit dem Case Tool Rational Rose entworfen.

Systemdienste:

- *Strategie-Service* ist die Verwaltung der im System vorhandenen Dienste. Auf der Basis der Dienstebeschreibung der einzelnen Dienste erstellt er die benötigten Arbeitsabläufe zur Erbringung einer Leistung.
- *Workflow-Service* ist die Steuerung und Kontrolle der vom Strategie-Service erstellten Arbeitsabläufe.
- *Ressource-Service* ist für die Verwaltung der Systemressourcen zuständig. Bei ihm müssen die einzelnen Dienste jeweils benötigte Systemressourcen reservieren.

- *Repository-Service* ermöglicht einen plattformunabhängigen Austausch von Dateien zwischen den einzelnen Diensten.
- *Daten-Service* ist für die konsistente Aufbewahrung von Daten, welche die Simulationsdienste benötigen, zuständig. Er stellt z.B. Funktionalitäten einer Datenbank zur Verfügung.

Allgemeine Dienste:

- *Archiv-Service* ist für die Archivierung und ggf. Wiedereinspielung von Eingabedaten und Ergebnissen zuständig. Die Archivierung kann dabei prinzipiell auf ein beliebiges Archiv-Medium erfolgen. Im KFÜ dient die ZDH als Archiv-Medium.
- *Verwaltungs-Service* ist für die Verwaltung von Simulationssessions und Benutzer zuständig.
- *Protokoll-Service* generiert aus den Einzelprotokollen der verschiedenen Dienste aufgabenbezogen aufbereitete Protokolle.
- *Konsistenzchecker* prüft die Datenkonsistenz aller ins System eingehenden Daten. Er muß besonders darauf achten, daß die Daten der Ontologie entsprechen.

Clientendienste:

- *ClientInterface* kapselt die Schnittstelle zur Benutzeroberfläche. Somit ist es auch möglich, fremdentwickelte Benutzeroberflächen in das System zu integrieren, ohne daß andere Dienste modifiziert werden müssen.
- *ClientManager* repräsentiert den Clienten (Benutzer) im System und verbindet eine Simulationssession (aufgabenbezogene Tätigkeiten) mit einer Kommunikations-session (Kommunikationsverbindung zwischen Benutzer und System). Der *ClientManager* sorgt dafür, daß Nachrichten und Anfragen an den entsprechenden Benutzer weitergeleitet werden. Durch die Aufteilung in *ClientManager* und *ClientInterface* ist es möglich, dies auch sicherzustellen, wenn der Benutzer in dem Moment der Anfrage nicht im System angemeldet ist.
- *Alarm-Service* stellt eine besondere Art von Clienten dar. Im Falle der Ausbreitungsrechnung ist er der Auftraggeber von, in einem Störfall, automatisch gestarteten Ausbreitungsrechnungen.

Datenbeschaffungsdienste:

- *ZDH-Service* kapselt die Schnittstelle zu externen Meß- und Stammdaten. Im KFÜ sind dies die KFÜ-Meßdaten, KFÜ-Stammdaten und Benutzerdefaults.
- *Emissions-Service* bereitet die Emissionsdaten für die Simulationsrechnung auf.
- *Stammdaten-Service* verwaltet die ABR-spezifischen Stammdaten.
- *Calculator-Service* stellt eine Reihe von einfachen Berechnungen und Umrechnungen zur Verfügung. Er ist somit eine Art spezieller Taschenrechner.
- *Ersatzwert-Service* verwaltet Ersatzwerte, auf die im Falle eines Ausfalls von externen Daten zurückgegriffen werden kann.

Simulationsdienste:

- Je ein Dienst entspricht einem Simulationsdienst. Diese Dienste kapseln die Simulationsmodule des alten Dienstes CRAYSIM /1/ so, daß die Simulationsmodule unabhängig von diesem System weiterentwickelt und jeweils die aktuellsten Versionen integriert werden können. Im KFÜ wurden 11 Dienste für die einzelnen Simulationsaufgaben ausgewählt.

Die 11 Simulationsdienste bilden das Kernstück der ABR. Sie ermöglichen etwa

- die Berechnung von Windfeldern,
- die Bestimmung von Quelltermen,
- die Simulation der Ausbreitung oder
- die Abschätzung von Folgen.

Aus diesen Diensten sind für eine Simulation geeignete Dienstketten zu bestimmen, die mit konsistenten Daten zu versorgen sind und deren Ergebnisse korrekt verwaltet werden müssen. Dem dienen

- die Datenbeschreibungsdienste,
- die Systemdienste und
- die sonstigen Dienste.

Schließlich muß das System mit dem Systemnutzer (Client) kommunizieren, das geschieht über die Klientdienste.

Alle Dienste gemeinsam kommunizieren über eine CORBA-Schiene. Sie können daher sowohl auf dem Klient als auch auf CORBA-fähigen Servern implementiert werden.

Die Implementierung aller Dienste erfolgt auf der Basis des Entwurfes mit Hilfe von Visual C++.

Wie schon ausgeführt, muß es zum flexiblen Einsatz dieser Dienste möglich sein, konsistente Datensätze bereitzustellen und Dienstketten zu bilden, die unter den Bedingungen einer konkreten Datenlage geeignet sind, das gewünschte Ergebnis zu erzielen. Deswegen wurden in die ABR das Konzept der Ontologie als Basis für die Verständigung und das Konzept des Workflows zur Prozeßsteuerung eingeführt.

Ein weiterer Vorteil dieser Formalisierungen ist die Möglichkeit, die Konsistenz der in der Ontologie beschriebenen Informationen zu prüfen. Außerdem können die Dienste Teile der Ontologie automatisch verarbeiten und gleichzeitig ihre Übereinstimmung mit dieser sicherstellen.

5. Ontologie und Workflow in der ABR

5.1 Ontologie

Im gegenwärtigen System der ABR ist das Wissen um die Informationen und ihre Abhängigkeiten primär in den Köpfen der Entwickler vorhanden. Erweiterungen des Systems, Modifikationen bestehender Systemkomponenten (Dienste) und die Integration neuer Dienste müssen in Übereinstimmung mit diesen impliziten Zusammenhängen sein. Die systeminterne Kommunikation und die gemeinsame Verwendung von Informationen setzt eine Übereinstimmung mit diesen impliziten Konventionen voraus.

Wenn Mitarbeiter das Team verlassen oder neue Mitarbeiter zu einem Team hinzustoßen, gehen implizite Informationen verloren bzw. müssen neu erarbeitet und vermittelt werden.

Über lange Zeiträume laufende Systeme sind dadurch schwer zu warten, zu modernisieren und zu erweitern. Die Wahrscheinlichkeit einer grundlegenden, aber vermeidbaren Neuimplementierung steigt, weil wichtige Informationen im Laufe der Zeit verloren gehen.

Die Architektur des „logischen Klienten“ stützt sich auf eine explizite formale Beschreibung der im System vorhandenen Informationen und Abhängigkeiten. Basis einer solchen Ontologie ist die Beschreibung der Informationen durch Daten und ihre Metadaten. Dies wird ergänzt durch kontextabhängige Beziehungen zwischen den Informationen. Durch den Einsatz einer Ontologie soll das bestehende Wissen gesichert und ein gemeinsames Verständnis bezüglich aller Informationen und deren Verwendung erreicht werden. Eine formale Beschreibung ist insbesondere notwendig, um eine einheitliche, allgemein akzeptierte Basis für die Kommunikation zwischen den Diensten und zwischen Dienst und Anwender zu schaffen. Der Aufwand für die Einordnung und Integration neuer Dienste in das System wird mit diesem allgemeinen, klaren und expliziten Verständnis von Informationen und ihrer Verwendbarkeit im Vergleich zu herkömmlichen Ansätzen deutlich reduziert.

Ein weiterer Vorteil der formalen ontologischen Beschreibung ist die Möglichkeit, die Konsistenz der in der Ontologie beschriebenen Informationen zu prüfen. Außerdem können die Dienste Teile der Ontologie automatisch verarbeiten und gleichzeitig ihre Übereinstimmung mit dieser sicherstellen.

5.1.1 Workflow

Komplexe Arbeitsabläufe treten immer stärker in den Mittelpunkt der ABR. Geänderte Randbedingungen, neue Anforderungen und die Verwendung neuer bzw. modifizierter Komponenten führen zu großen Herausforderungen bezüglich der Modellierung und Durchführung von Arbeitsabläufen. Je flexibler die Arbeitsabläufe beschrieben und durchgeführt werden können, desto flexibler ist das Gesamtsystem. Hierzu ist es notwendig, daß die Information über den Arbeitsablauf nicht mehr bei den einzelnen Diensten sondern zentral gespeichert und verwaltet werden. Ein Dienst soll von seinen „Vorgängen“ und „Nachfolgern“ in der Prozeßkette unabhängig sein, um in neue Arbeitsabläufe mit anderen Vorgängern und Nachfolgern integrierbar zu werden.

Aus diesem Grund wurde im Konzept des logischen Klienten für die Modellierung und Durchführung der Arbeitsabläufe der Workflow-Ansatz gewählt. Die Workflows werden nicht mehr fest verdrahtet, sondern können flexibel und zur Laufzeit eingelesen und durchgeführt werden. Dazu benötigt man eine Beschreibungssprache für Workflows. Das Wissen um die Arbeitsabläufe und die logische Abfolge der Einzelaktivitäten ist in der Workflow-Beschreibung festgehalten.

Wenn aufgrund von neuen bzw. geänderten Bedingungen neue Arbeitsabläufe notwendig werden, dann brauchen nur neue Workflow-Beschreibungen erstellt bzw. bestehende Workflow-Beschreibungen modifiziert werden. Diese können dann sofort ausgeführt und getestet werden.

5.1.2 Vorteile der Kombination von Workflow-Ansatz und ontologischer Beschreibung

Die in einer Workflow-Beschreibung festgehaltenen Abläufe müssen den logischen Zusammenhängen zwischen den erzeugten und benötigten Informationen entsprechen. Die formale Beschreibung von Workflows und die formale Beschreibung der Zusammenhänge erlaubt prinzipiell das automatische Überprüfen von Workflows anhand der Ontologie hinsichtlich ihrer Vollständigkeit, Sinnhaftigkeit und Durchführbarkeit. Damit scheint ein Ansatz dafür gefunden, Qualitätssicherung in dienstebasierten Systemen schon beim Entwurf mit einzubringen. Die Durchführbarkeit dieser Idee soll im Rahmen von GLOBUS IV nachgewiesen werden.

5.2 Wechselwirkungen zwischen Projekten Teilerneuerung KFÜ und GLOBUS

Anhand der ABR wird auch deutlich, wie stark sich die Projekte GLOBUS und Teilerneuerung KFÜ beeinflussen.

Wesentliche Architekturentscheidungen aus dem GLOBUS-Projekt wurden beim Entwurf von den Partnern dSH und IKE/KE übernommen und auf die spezielle Situation im KFÜ angepaßt. Man kann schon heute sagen, daß sich dies bewährt hat. Aufgrund dieses Vorgehens wird das KFÜ nicht nur ein Web-basiertes Informationssystem sein, in das Teile aus anderen Systemen – etwa in Form von Klassen – mit übernommen wurden. Vielmehr wird es das erste Informationssystem sein, das den im Rahmen der Fortschreibung und Weiterentwicklung der Rahmenkonzeption 1998 /22/ verfolgten Ansatz der dienstebasierten Systeme umsetzt. Dies gilt sowohl für die im KFÜ verwendeten Architekturen, als auch für die Art und Weise wie bei der Entwicklung vorgegangen wird. Dabei kommt auf Seite des IKE/KE das Vorgehensmodell des Bundes (V-Modell /13/) in seiner objektorientierten Fassung zum Einsatz.

Das Projekt Teilerneuerung KFÜ kann damit im Sinne der Rahmenkonzeption auch als Leitprojekt für andere ähnlich anspruchsvolle Entwicklungen angesehen werden.

6. Literatur

/1/ Mayer-Föll, R., Jaeschke, A.: Projekt GLOBUS – Vorstudie zur Konzeption und prototypischen Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten. Phase I 1994. Wissenschaftliche Berichte FZKA 5863.

Mayer-Föll, R., Jaeschke, A.: Projekt Globus: Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten im Umweltinformationssystem Baden-Württemberg, Phase II 1995. FZKA 5700, Dez. 1995.

Mayer-Föll, R., Jaeschke, A.: Projekt GLOBUS – Konsolidierung der neuen Systemarchitektur und Entwicklung erster Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg. Phase III 1996: Wissenschaftliche Berichte FZKA 5900.

Mayer-Föll, R., Jaeschke, A.: Projekt GLOBUS – Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umweltsachdaten im UIS-BW. Phase IV 1997. Wissenschaftliche Berichte FZKA 6000.

- /2/ 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
- /3/ Meyers, Scott: *Effective C++*, Addison-Wesley, Reading et al., 1997.
- /4/ Gamma, E., Helm, R. et al.: *Entwurfsmuster*, Addison-Wesley, Bonn et al., 1992.
- /5/ Coplien, J.O.: *Advanced C++*, Addison-Wesley, Reading et al., 19
- /6/ Olivetti Research Laboratories: omniORB, URL <http://www.orl.co.k/omniORB/omniORB.html> 1998
- /7/ Dumbleton, G.: *The Home of OSE*, URL <http://www.dspl.com.au/>, 1998.
- /8/ Schmidt, D.: *The Adaptive Communication Environment*, URL <http://www.cs.wustl.edu/~schmidt/ACE.html>, 1998.
- /9/ Object Management Group: CORBA/IIOP Index, URL <http://www.omg.org/corba/c2indx.htm>, 1998
- /10/ Finin, T., Fritzson R., et al : *KQML as an Agent Communication Language*, in: Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), ACM Press, November 1994
- /11/ Labrou in <http://kleinstweich.com/billyboy.html> 1998
- /12/ Siegel, J.: *CORBA – Fundamentals and Programming*, Wiley Computer Publishing Group, New York, 1996
- /13/ Dröschel, W., Heuser, W., Midderhoff, W. (Hrsg.): *Inkrementelle und objektorientierte Vorgehensweisen mit dem V-Modell 97*. R. Oldenbourg Verlag, München, Wien, 1998.
- /14/ Grohmann, A., Kopetzky, R.: *An Agent-Enhanced Architecture for the Logical Client in an Environment Information System - To appear in the Proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agents (PAAM'98)*, London, March 1998.
- /15/ Gensereth, M., Fikes, R.: *Knowledge Interchange Format*, in: Stanford Logic Report Logic-92-1, Stanford University, <http://logic.stanford.edu/kif/kif.html>, 1992.
- /16/ Giarratano, J., Riley, G.: *Expert Systems – Principles and Programming*, PWS Publishing Company, Boston, 1994.
- /17/ Russell, S., Norvig, P.: *Artificial Intelligence – A Modern Approach*, Prentice-Hall, Inc., Englewoods Cliffs, New Jersey, 1995.
- /18/ Lee, J., PIF Working Group: *The PIF Process Interchange Format and Framework*. <http://soa.cba.hawaii.edu/pif> (1996).
- /19/ <http://www.microsoft.com/netmeeting/>
- /20/ <http://www.wpine.com/>
- /21/ <http://www.iss-mainz.de/>
- /22/ Mayer-Föll, R., Pätzold, J., et al.: *Ministerium für Umwelt und Verkehr, Landesanstalt für Umweltschutz Baden-Württemberg, FAW Ulm; Umweltinformationssystem Baden-Württemberg. Rahmenkonzeption 1998 als Teil des Landessystemkonzepts, Entwurf 1998.*

Bestandsaufnahme und Bewertung von ausgewählten UIS-Systemen zur Fortschreibung und Weiterentwicklung der Rahmenkonzeption des Umweltinformationssystems Baden-Württemberg

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

*A. Keitel,
Landesanstalt für Umweltschutz Baden-Württemberg,
Bannwaldallee 24,
76185 Karlsruhe*

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

1. EINFÜHRUNG	131
2. DIE RK UIS '98.....	131
3. ZWECK DER BESTANDSAUFNAHME DES UIS	132
4. BESCHREIBUNGSSYSTEMATIK.....	133
5. SYSTEME DES UIS	133
6. DATENFLÜSSE IM UIS	134
7. RESÜMEE	138
8. LITERATUR.....	139

1. Einführung

Das Umweltinformationssystem Baden-Württemberg (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. Von Anfang an wurde mit den berührten Ressorts und dem kommunalen Bereich innerhalb Baden-Württembergs sowie bundesweit im Bund/Länder-Arbeitskreis Umweltinformationssysteme (BLAK UIS) zusammengearbeitet.

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.

2. Die RK UIS '98

Die Fortschreibung und Weiterentwicklung der Rahmenkonzeption des UIS aus dem Jahr 1992 zur Rahmenkonzeption 1998 (RK UIS '98) wurde aufgrund durchgreifender Veränderungen der Anforderungen und Möglichkeiten notwendig.

Die Fachaufgaben im Umweltbereich wurden ausgeweitet. Auf organisatorischer Ebene erfolgten Änderungen infolge von Umressortierungen, der Verwaltungsreform und des IuK-Verbands zwischen Land und Kommunen. Weiter sind eine Abkehr von der Kameralistik und eine Zuwendung zu betriebswirtschaftlichen Strukturen zu verzeichnen. Die Berichtspflichten gegenüber dem Bund und der EU sind aufgrund der Fortentwicklung des europäischen Umweltrechts umfangreicher geworden. Hinzu kommt das Recht des Bürgers auf freien Zugang zu Informationen über die Umwelt.

Die rasante informationstechnische Weiterentwicklung, insbesondere in den Bereichen Hardware, Betriebssysteme und Netzwerke/Internet, sowie eine zunehmende Objektorientierung in allen Bereichen der Informationstechnik macht Fortschreibung unumgänglich. Dieser Entwicklung wurde in Baden-Württemberg auch durch das neue IuK-Modell zur Fortschreibung des Landessystemkonzepts Rechnung getragen.

Die Inhalte der RK UIS '98 entstanden in einem mehrjährigen, umfassenden Diskussionsprozeß. An diesem Diskurs waren - neben dem Ministerium für Umwelt und Verkehr Baden-Württemberg als Projektträger und der Landesanstalt für Umweltschutz als Projektentwicklungsstelle - die Stabsstelle für Verwaltungsreform im Innenministerium (StaV) sowie die im Koordinierungsausschuß UIS vertretenen Dienststellen beteiligt. Außer dem FAW als der mit der Projektdurchführung beauftragten Einrichtung waren auch andere Forschungsinstitute und Firmen involviert. Von besonderer Bedeutung für die RK UIS '98 hatten das Forschungs- und Entwicklungsvorhaben GLOBUS, das Land/Kommunen-Verbundvorhaben WAABIS, das neue IuK-Modell zur Fortschreibung des Landessystemkonzepts und die Beratungen im Bund/Länder-Arbeitskreis Umweltinformationssysteme (BLAK UIS).

Die Ergebnisse dieser Untersuchung wurden vom FAW in enger Zusammenarbeit mit dem Ministerium für Umwelt und Verkehr und der Landesanstalt für Umweltschutz zur Fortschreibung und Weiterentwicklung der Rahmenkonzeption des Umweltinformationssystems Baden-Württemberg in Form einer detaillierten Hauptuntersuchung in vier Bänden dokumentiert. Auf der Grundlage dieser Hauptuntersuchung wurde die RK UIS '98 erstellt.

Die in der RK UIS '98 enthaltenen Konzepte und Handlungsvorschläge wurden in den Abschlußberichten der GLOBUS-Phasen III und IV bereits detailliert dargelegt und sollen hier nicht wiederholt werden. Im folgenden soll vielmehr die Bestandsaufnahme des UIS, der ein Band der Hauptuntersuchung gewidmet ist und die eine wesentliche Basis der RK UIS '98 darstellt, diskutiert werden.

3. Zweck der Bestandsaufnahme des UIS

Die Grundlage jeder Fortentwicklung der Rahmenkonzeption eines informationstechnischen Systems besteht in einer Analyse des bisher erreichten Entwicklungsstandes und in einer kritischen Diskussion dieses Entwicklungsstandes im Hinblick auf die aktuell formulierten Anforderungen an das System. Daher wurde im Zuge der Fortschreibung der Rahmenkonzeption des UIS eine Bestandsaufnahme ausgewählter Einzelsysteme vorgenommen. Die im Auftrag des BLAK UIS durchgeführte Dokumentation der Umweltinformationssysteme des Bundes und der Länder wurde für Baden-Württemberg vertieft.

Die detaillierte Beschreibung der Einzelsysteme ermöglicht insbesondere:

- eine aktuelle Darstellung einzelner UIS-Systeme,
- einen besseren Überblick über das Gesamtsystem UIS,
- die Visualisierung der Komplexität des Gesamtsystems,
- die Aufdeckung von Synergiepotentialen zwischen den einzelnen Systemen,
- den Vergleich technischer Konzeptionen und Realisierungen,
- die Analyse thematischer Verknüpfungen zwischen den Systemen,
- die Überprüfung der Ansätze für die Modularisierung des Systems,
- die Analyse der Anforderungen an die Netzinfrastruktur,
- die Gewährleistung der Konsistenz im weiteren Ausbau des UIS,
- die bessere Einschätzbarkeit der Nutzungsintensität und des Entwicklungsstands.

Die Analyse bezieht sich somit auf technische Aspekte, auf organisatorische Fragestellungen sowie auf die praktische Nutzung der Systeme einschließlich der Verfügbarkeit der Daten. Im folgenden werden zusammenfassend wichtige Punkte der Analysen vorgestellt sowie eine Übersicht über ausgewählte - größere - Systeme gegeben, die sich im UIS im Einsatz befinden.

4. Beschreibungssystematik

Im UIS werden generell drei Systemkategorien unterschieden:

Übergreifende UIS-Komponenten sind Systeme, die der Zusammenführung und fachübergreifenden Nutzung von Informationen aus verschiedenen Umweltbereichen oder verschiedenen Zuständigkeitsbereichen dienen. Unterkategorien der Übergreifenden UIS-Komponenten sind: Auskunfts- und Nachweissysteme, Berichtssysteme sowie Datenhaltungs- und Verteilungs-Systeme.

Grundkomponenten des UIS sind fachspezifische, nach Umweltmedien oder sektoral nach Zuständigkeiten gegliederte Systeme zur Erfassung und Bearbeitung von Fachdaten im Umweltbereich. Sie werden typischerweise von Facheinheiten der Verwaltung im Umweltbereich im weiteren Sinne betrieben, um dienststellenbezogene Fachaufgaben zu unterstützen. Unterkategorien der Grundkomponenten des UIS sind: Dienststellensysteme, Fachliche Vorgangsbearbeitungs-Systeme, Fachspezifische Auswertesysteme, Fachdatenbanken sowie Meß- und Erfassungssysteme.

Basissysteme des UIS sind Informationssysteme und Infrastruktur-Elemente, die nicht speziell für die Bearbeitung von Umweltinformationen geschaffen wurden, deren Daten, Methoden oder Infrastruktur aber auch hierzu herangezogen werden. Unterkategorien sind: Hintergrund-Verfahren und -Datenbanken sowie interne und externe Infrastruktur-Systeme.

5. Systeme des UIS

Innerhalb des Umweltinformationssystem Baden-Württemberg wurden 49 größere Systeme im informationstechnischen Sinn identifiziert. Die nachfolgende Aufzählung enthält UIS-Systeme und Komponenten des UIS, die sich überwiegend im Geschäftsbereich des Ministeriums für Umwelt und Verkehr, aber auch im Geschäftsbereich des Ministeriums Ländlicher Raum und des Wirtschaftsministeriums befinden. Systeme in der Zuständigkeit anderer Ressorts und kleinere UIS-Komponenten werden nicht dargestellt. Die Aufstellung ist daher, was das gesamte UIS Baden-Württemberg betrifft, nicht vollständig.

Kurzbezeichnung	Ausgewählte Systeme des UIS
WWW-UIS	WWW- Service UIS
UDK	Umweltdatenkatalog
HUDA	Hypermedia-System zur Umweltberichterstattung
UVIS	Umwelt- und Verkehrs-Informationsservice
GR-DB	Grenz- und Richtwerte-Datenbank
MEROS	Meßreihen-Operationssystem
RIPS	Räumliches Informations- und Planungssystem
IS-GAA	Informationssystem der Gewerbeaufsicht
EFÜ	Emissionsdatenfernübertragung
EKAT11	Emissionkataster zur 11. BlmschVO
START	Datenbank Stand der Abgasreinigungstechnik
AnlaKat	Anlagenkataster zur Störfallverordnung
SIB	Straßeninformationsbank

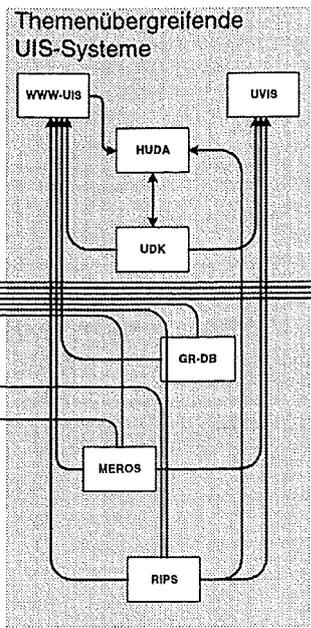
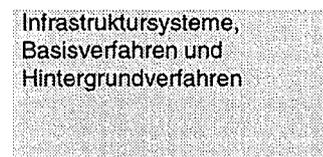
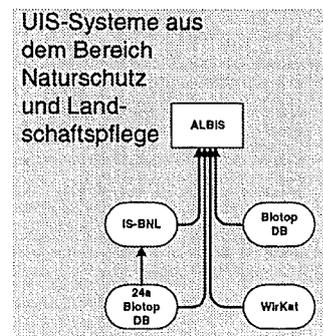
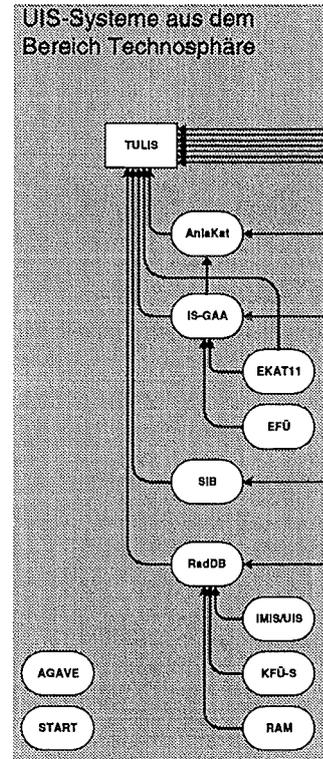
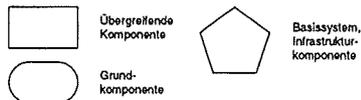
AGAVE	Atomrechtliches Genehmigungs- und Aufsichtsverfahren
KFÜ-S	System der Kernreaktor-Fernüberwachung
IMIS/UIS	Integriertes Meß- und Informationssystem zum Strahlenschutzvorsorgegesetz in BW
RAM	Radioaktivitätsmeßnetz
RadDB	Radioaktivitätsdatenbank
TULIS	Technosphäre- und Luft-Informationssystem
ZFD	Zentraler Fachdienst Wasser, Boden, Abfall, Altlasten
KIWI	Kommunikativ-Integriertes Wasser- und Abfallwirtschaftliches Info-System
VAwS	Verfahren zur Überwachung von Anlagen zum Umgang mit wassergefährdenden Stoffen
WEE	Verfahren zum Wasserentnahmeentgelt
MAWAG	Managementsystem Abwasserabgabe
FÖWASS	Verfahren Förderwesen Wasserwirtschaft
GWDB	Grundwasserdatenbank
DK-FIGewB	Datenkatalog Fließgewässerbeschaffenheit
LABDÜS	Labordaten-Übertragungssystem
HyAP	Hydrologischer Arbeitsplatz
HVZ	Hochwasservorhersage-Zentrale
GewIS	Gewässerinformationssystem
DB-Bodensee	Datenbank Bodensee
ArcWaWiBo	Geosystem Wasserwirtschaft/Bodenschutz
ARÜS	System zur Abfall- und Reststoffüberwachung
AWIDAS	Abfallwirtschaftliche Datensammlung
XUMA	Expertensystem Altlasten
FIS-Bo	Fachinformationssystem Bodenschutz
WAABIS-BS	WAABIS-Berichtssystem
WirKat	Ökologisches Wirkungskataster
BiotopDB	Biotopdatenbank
24a-BiotopDB	24a-Biotopdatenbank
IS-BNL	Informationssystem der BNL
ALBIS	Arten-, Landschafts-, Biotop-Informationssystem
LIMS-LfU	Labor-Informations- und Management-System der LfU
RZ-KA	Rechenzentrum/Servicezentrum Karlsruhe
SZ-TU	Servicezentrum Technische Unterstützung Stuttgart
IKS-UVM	Informations- und Kommunikationssystem des UVM
ALK	Automatisierte Liegenschaftskarte
ATKIS	Amtliches Topographisch-Kartographisches Informationssystem

Tabelle 1 : Ausgewählte Systeme des UIS

6. Datenflüsse im UIS

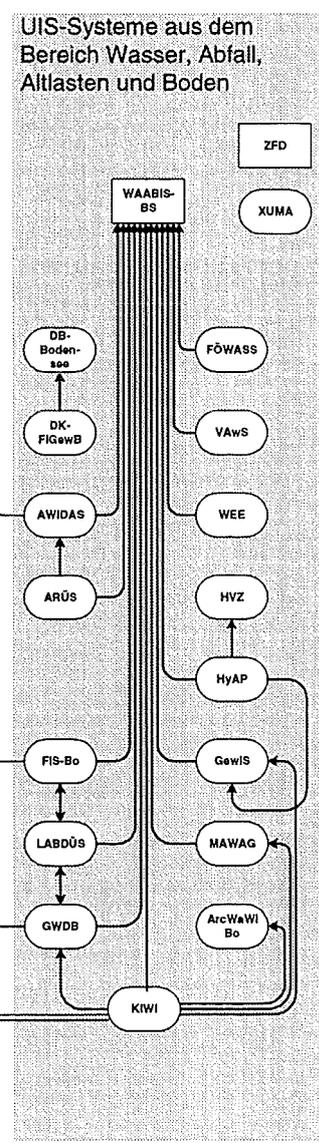
Auf der Basis der Systembeschreibungen wurde eine Analyse der Systemlandschaft in Bezug auf den Datenaustausch zwischen den ausgewählten UIS-Systemen vorgenommen. Ziel war, exemplarisch einen Überblick über die Systemlandschaft zu gewinnen und einen Eindruck von den Möglichkeiten und der Methodik der Untersuchung der UIS-Systemlandschaft zu vermitteln. Das Resultat der Analyse ist in den Abbildungen der folgenden Seiten wiedergegeben.

Datenflüsse im UIS



Schwerpunkt Betrachtung der UIS-Systeme im Bereich Technosphäre

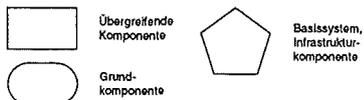
Datenflüsse in die Systeme



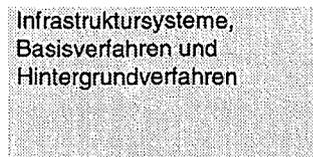
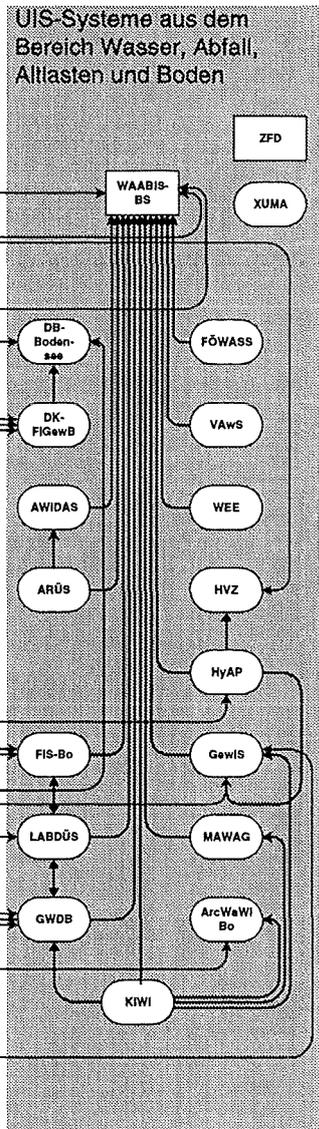
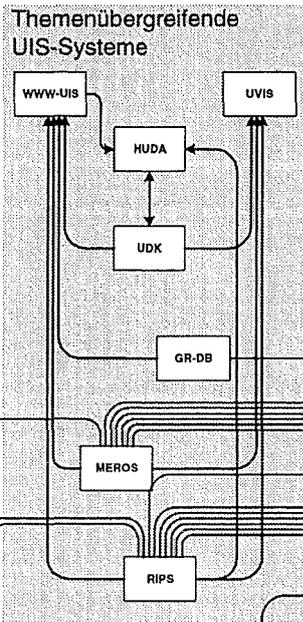
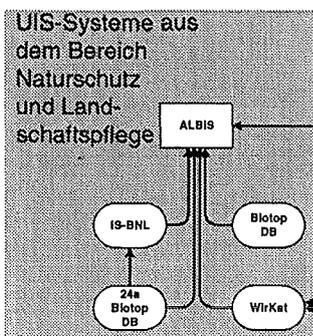
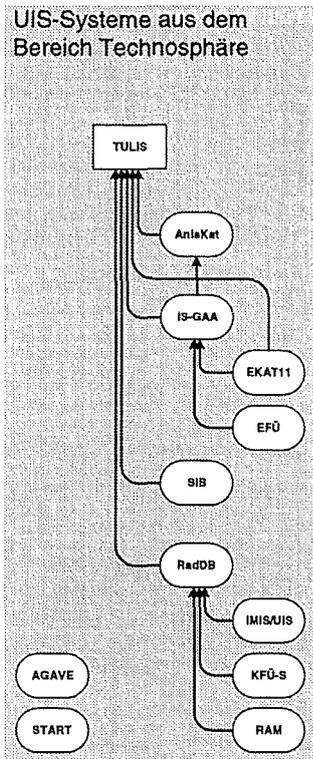
Ein Datenaustausch zwischen den verschiedenen UIS-Systemen und den Infrastrukturkomponenten finden im Zuge der Systemadministration statt. Da das Hauptaugenmerk dieser Auswertung jedoch auf dem Austausch von Fachdaten liegt, werden diese Relationen in diesem Zusammenhang nicht berücksichtigt.

Abbildung 1: Datenflüsse in ausgewählte Systeme des UIS im Bereich der Technosphäre

Datenflüsse im UIS



Schwerpunkt Betrachtung der UIS-Systeme im Bereich Wasser, Abfall, Altlasten, Boden sowie Naturschutz und Landschaftspflege
Datenflüsse in die Systeme

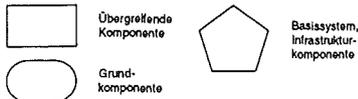


Ein Datenaustausch zwischen den verschiedenen UIS-Systemen und den Infrastrukturkomponenten findet im Zuge der Systemadministration statt. Da das Hauptaugenmerk dieser Auswertung jedoch auf dem Austausch von Fachdaten liegt, werden diese Relationen in diesem Zusammenhang nicht berücksichtigt.



Abbildung 2: Datenflüsse in ausgewählte Systeme des UIS der Bereiche Wasser, Abfall, Altlasten, Boden, Naturschutz und Landschaftspflege

Datenflüsse im UIS



Schwerpunkt Betrachtung der themenübergreifenden UIS-Systeme
Datenflüsse in die Systeme

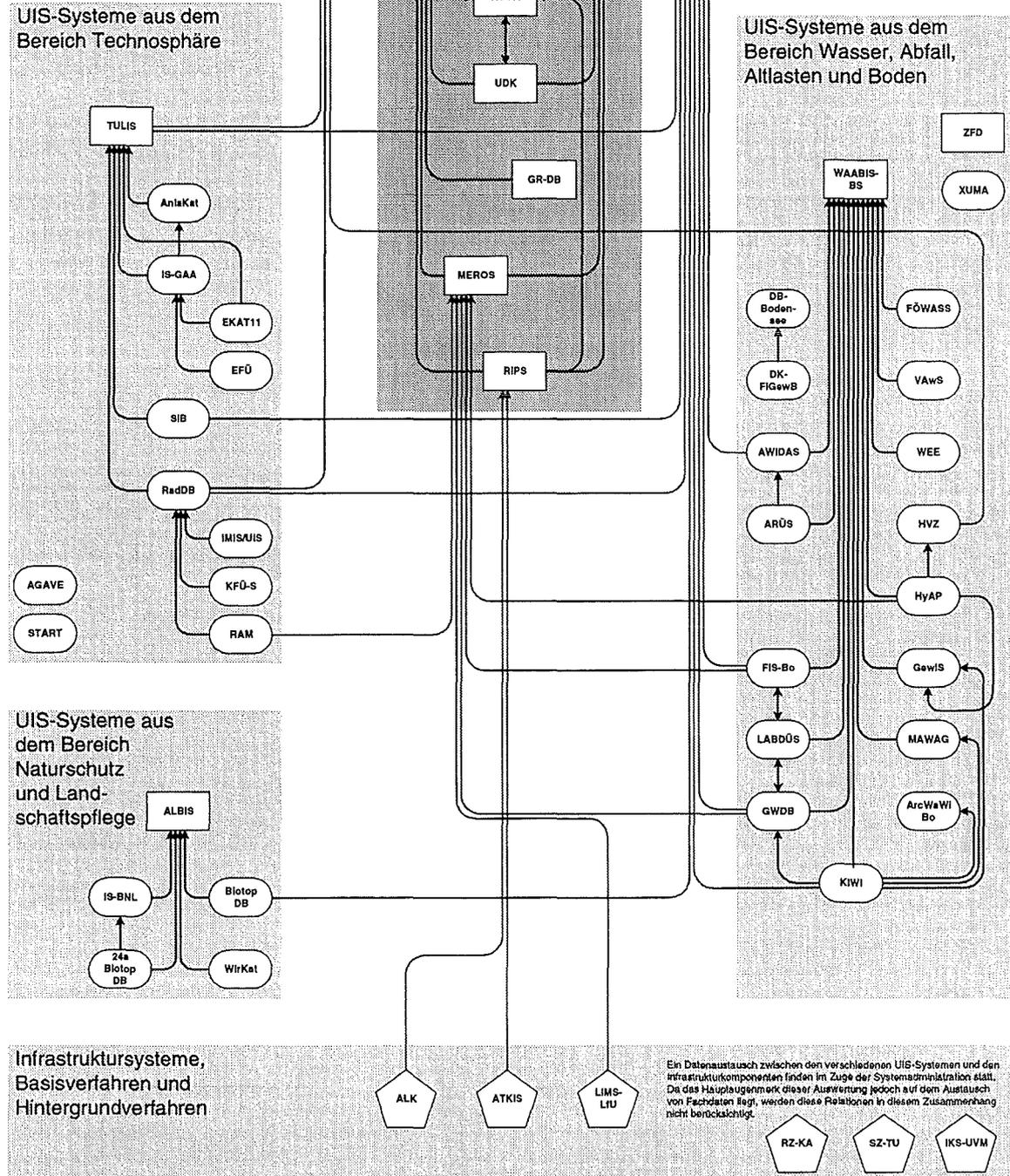


Abbildung 3: Datenflüsse in ausgewählte themenübergreifende Systeme des UIS

Aus den in den Abbildungen 1, 2 und 3 dargestellten Schwerpunktbetrachtungen lassen sich einige interessante Feststellungen zum Datenaustausch zwischen den ausgewählten UIS-Systemen ableiten:

- Der Datenfluß findet überwiegend von den Basissystemen und Grundkomponenten zu den übergreifenden Komponenten statt. Dies entspricht dem Grundsatz der Durchgängigkeit der Daten, dem bereits in der Konzeption des UIS der 1. und 2. Generation großes Gewicht zugemessen wurde. Die Datenerhebung erfolgt primär durch die Grundkomponenten.
- Erkennbar ist die besondere Bedeutung der Berichtssysteme TULIS, WAABIS-BS und ALBIS in ihren jeweiligen Themenbereichen. Vor allem WAABIS-BS greift auf eine Vielzahl von Grundkomponenten des UIS zu. Ersichtlich wird hieraus unmittelbar die große thematische Breite des WAABIS-Gesamtsystems, das einen erheblichen Teil aller Umweltthemenbereiche des UIS abdeckt.
- Deutlich werden die Rollen des themenübergreifenden Berichtssystems UVIS und der Online-Bereitstellung von Informationen im World-Wide Web mit dem System WWW-UIS.
- Bei der Datenhaltung und Datenbereitstellung nehmen die Systeme MEROS und RIPS eine zentrale Rolle ein. Bei deren Konzeption wurde versucht, die plattform- und systemübergreifende Verfügbarkeit einer ausgewählten Datenbasis des UIS sicherzustellen. Wie den umfangreichen Datenflußbeziehungen der beiden Systeme zu entnehmen ist, ist dies gelungen.
- Die Bedeutung der Georeferenzierung für die Nutzung von Umweltinformationen wird aus den Datenflußbeziehungen der Basissysteme ATKIS und ALK deutlich. Eine Vielzahl von Grundkomponenten und übergreifenden Komponenten des UIS nutzen unmittelbar diese Daten der Vermessungsverwaltung. Die Bereitstellung von ATKIS-Daten erfolgt über RIPS.
- Datenflüsse sind auch in horizontaler Richtung, d.h. zwischen Grundkomponenten oder zwischen übergreifenden Komponenten, zu beobachten. Hieran ist zu erkennen, daß die Verknüpfung von Daten unterschiedlicher Umweltthemenbereiche für viele Dienststellen der Umweltverwaltung - auch und gerade auf der Fachebene - wichtig ist.
- Die Datenflußbeziehungen zwischen den Systemen sind allgemein recht komplex. Daraus resultieren auf der technischen Ebene Anforderungen im Hinblick auf die Kopplung der Systeme.

7. Resümee

Mit der Bestandsaufnahme des UIS wurde eine wesentliche Grundlage für die Fortschreibung und Weiterentwicklung der Rahmenkonzeption des UIS geschaffen. Es steht darüber hinaus eine aktuelle Dokumentation zur Verfügung, die die wesentlichen informationstechnischen Systeme des UIS und ihr Zusammenspiel benutzerfreundlich darstellt.

Gezeigt hat sich bei dieser Analyse das enge Zusammenwirken der verschiedenen Komponenten in den Dienststellen und Behörden der Verwaltung Baden-Württembergs. Hinzu kommen die Verflechtungen im Rahmen des IuK-Verbunds Land/Kommunen sowie der Austauschbeziehungen Land/Länder/Bund/Nachbarstaaten/Europäische Union.

Die Analyse der Datenströme bestätigt auch die Inhalte früherer UIS-Konzeptionen bezüglich der Hierarchie des UIS und der Unterscheidung der Systemkategorien "Übergreifende UIS-Komponenten", "Grundkomponenten" und "Basissysteme".

Die Komplexität der entstandenen Systemlandschaft zeigt darüber hinaus, welche Bedeutung leistungsfähigen Organisationsstrukturen und einer stetigen Fortschreibung und Weiterentwicklung der Rahmenkonzeption für den zukünftigen Ausbau des UIS zukommt.

8. Literatur

- /1/ Beuerle, B.; Grünreich, D. et al. (1998): Fachdatenintegration in ATKIS für das Umweltinformationssystem Baden-Württemberg - Phase II- Abschlußbericht. Ministerium für Umwelt und Verkehr Baden-Württemberg, Stuttgart.
- /2/ Henning, I., Wiest, G., Gaul, M. et al. (1998): Weiterentwicklung des Umwelt-Führungs-Informationssystems (UFIS) zum Umwelt- und Verkehrs-Informationsservice (UVIS) im Rahmen des Umweltinformationssystems Baden-Württemberg. Ministerium für Umwelt und Verkehr, Stuttgart.
- /3/ Mayer-Föll, R.; Jaeschke, A. (Hrsg.) (1997): Projekt GLOBUS: Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umwelt-Sachdaten im Umweltinformationssystem Baden-Württemberg Phase VI - 1997, Forschungszentrum Karlsruhe, Wissenschaftliche Berichte, FZKA 6000.
- /4/ 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.
- /5/ Mayer-Föll, R.; Pätzold, J. (Hrsg) (1998): Umweltinformationssystem Baden-Württemberg - Rahmenkonzeption 1998. Ministerium für Umwelt und Verkehr / Universitätsverlag Ulm.
- /6/ Mayer-Föll, R.; Pätzold, J. (Hrsg) (1998): Umweltinformationssystem Baden-Württemberg - Fortschreibung und Weiterentwicklung der Rahmenkonzeption - Hauptuntersuchung. Ministerium für Umwelt und Verkehr / FAW Ulm.
- /7/ Ministerium für Umwelt und Verkehr Baden-Württemberg (1997): Rahmenkonzeption WAABIS (Informationssystem Wasser, Abfall, Altlasten, Boden) als Teil des Umweltinformationssystems Baden-Württemberg. Stuttgart.
- /8/ Ministerium für Umwelt und Verkehr Baden-Württemberg/debis Systemhaus (1997): Spezifikation der Teilerneuerung der Kernreaktor-Fernüberwachung Baden-Württemberg. Stuttgart.
- /9/ Page, B.; Schickore, E.; Mack, J. (1996): Dokumentation der Umweltinformationssysteme des Bundes und der Länder, erstellt im Auftrag des Bund/Länder-Arbeitskreises Umweltinformationssysteme, Hamburg.
- /10/ Schäfer, G. (1997): Neues IuK-Modell zur Fortschreibung des Landessystemkonzepts Baden-Württemberg, Innenministerium Baden-Württemberg, Stuttgart.
- /11/ Umwelt- und Verkehrsinformationen Baden-Württemberg, WWW-Server:
<http://www.uvm.baden-wuerttemberg.de/>



Dokumentation des Umwelt- und Verkehrs-Informationsservices Baden-Württemberg

*W.-F. Riekert, A. Schultze,
Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW),
Helmholtzstr. 16,
89081 Ulm/Donau*

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

1. AUSGANGSPUNKT.....	143
2. REALISIERUNG.....	143
3. NUTZEN	144
4. BILDSCHIRMABZÜGE	145
5. LITERATUR.....	148

1. Ausgangspunkt

Das fach- und ressortübergreifende Umweltinformationssystem (UIS) Baden-Württemberg verfügt über inhaltlich und funktional weit fortgeschrittene Informationssysteme. In den einzelnen Dienststellen, insbesondere in der Landesanstalt für Umweltschutz (LfU) und im Ministerium für Umwelt und Verkehr (UVM), wurde ein umfangreicher Bestand an Informationen und Wissen aufgebaut, der in Form von Daten, Dienstprogrammen und multimedialen Dokumenten auf verschiedenen Computersystemen bereitgehalten wird. Zunehmende wirtschaftliche Bedeutung hat die Aufgabe, dieses große Angebot an Informationen und Wissen in einer generalisierten Sicht einem möglichst breiten Nutzerkreis in der Verwaltung sowie teilweise für die Öffentlichkeit verfügbar zu machen. Insbesondere wird es immer wichtiger, eine Gesamtsicht über bestimmte Umwelt- oder Verkehrsfragen fachübergreifend herzustellen. Mit IuK-Unterstützung können beispielsweise ganzheitliche Bewertungen der Umweltsituation leichter vorgenommen oder Entscheidungsgrundlagen für Planungen einfacher zusammengestellt werden.

Das UVM beauftragte das FAW, das Umwelt-Führungs-Informationssystem (UFIS) zum *Umwelt- und Verkehrs-Informationsservice* (UVIS) weiterzuentwickeln. UFIS war ursprünglich nur auf Anforderungen von Führungskräften zugeschnitten. Der Nutzerkreis wurde auf Fach- und Führungskräfte bei UVM, LfU und anderen Dienststellen ausgeweitet. Wesentliche Ziele von UVIS sind die effiziente Unterstützung dieses breiteren Nutzerkreises innerhalb der Verwaltung, insbesondere im Geschäftsbereich des UVM, sowie die dienstorientierte Realisierung des Systems als netzbasiertes Serviceangebot. UVIS war ein Projekt im Rahmen der Phasen III und IV des Forschungs- und Entwicklungsvorhabens Globale Umweltsachdaten (GLOBUS). Das Projekt wurde in GLOBUS V 1998 mit der Dokumentation abgeschlossen.

2. Realisierung

UVIS wurde auf der Basis der neuen Systemarchitektur des UIS als Client-/Server-Lösung realisiert und folgt dem neuen Informations- und Kommunikationsmodell des Landessystemkonzeptes Baden-Württemberg. Es werden die Internet/WWW-Technologie sowie PC-Standardsoftwarekomponenten genutzt, wodurch ein wirtschaftlicher Betrieb ermöglicht wird.

UVIS stellt einen Verbund aus einem Metainformationsserver für die Recherche und verteilten Informationsservern für die Bereitstellung von fachbezogenen Daten, Diensten und Dokumenten dar. Dabei kann eine Vielzahl von UVIS-Clients als Arbeitsplätze betrieben werden. Sowohl die Metadaten als auch die fachbezogenen Daten werden in einer relationalen Datenbank verwaltet. Um einen Online-Zugang zu Informationen in solchen Datenbanken über WWW zu ermöglichen, wurde das generische Datenbankgateway WebQuery entwickelt, das für den Zugriff auf die Metadaten und auf die fachbezogenen Daten eingesetzt wird.

Eine besondere Rolle spielt hierbei der UVIS-Client, auf den die Daten nach der Selektion transferiert werden und auf dem eine Aufbereitung abhängig von der Semantik der Daten erfolgt. Zur Aufbereitung werden (ggf. mittels Makros und Skripts erweiterte) PC-Standardsoftwarekomponenten sowie ein Desktop-Geoinformationssystem eingesetzt. Um die Nutzer

bei der Auswahl der möglichen Aufbereitungsarten in Abhängigkeit von den selektierten Daten und der aktuellen Arbeitsplatzkonfiguration unterstützen zu können, werden serverseitig Beschreibungsinformationen zu den selektierten Daten generiert, die vom UVIS-Client ausgewertet werden. Durch die alternative Verwendung von WWW- und Java-basierten Aufbereitungstechniken kann UVIS darüber hinaus grundsätzlich auch Informationssuchende (z.B. aus der Verwaltung und aus der Öffentlichkeit) bedienen, die über keine Installation des UVIS-Clients verfügen.

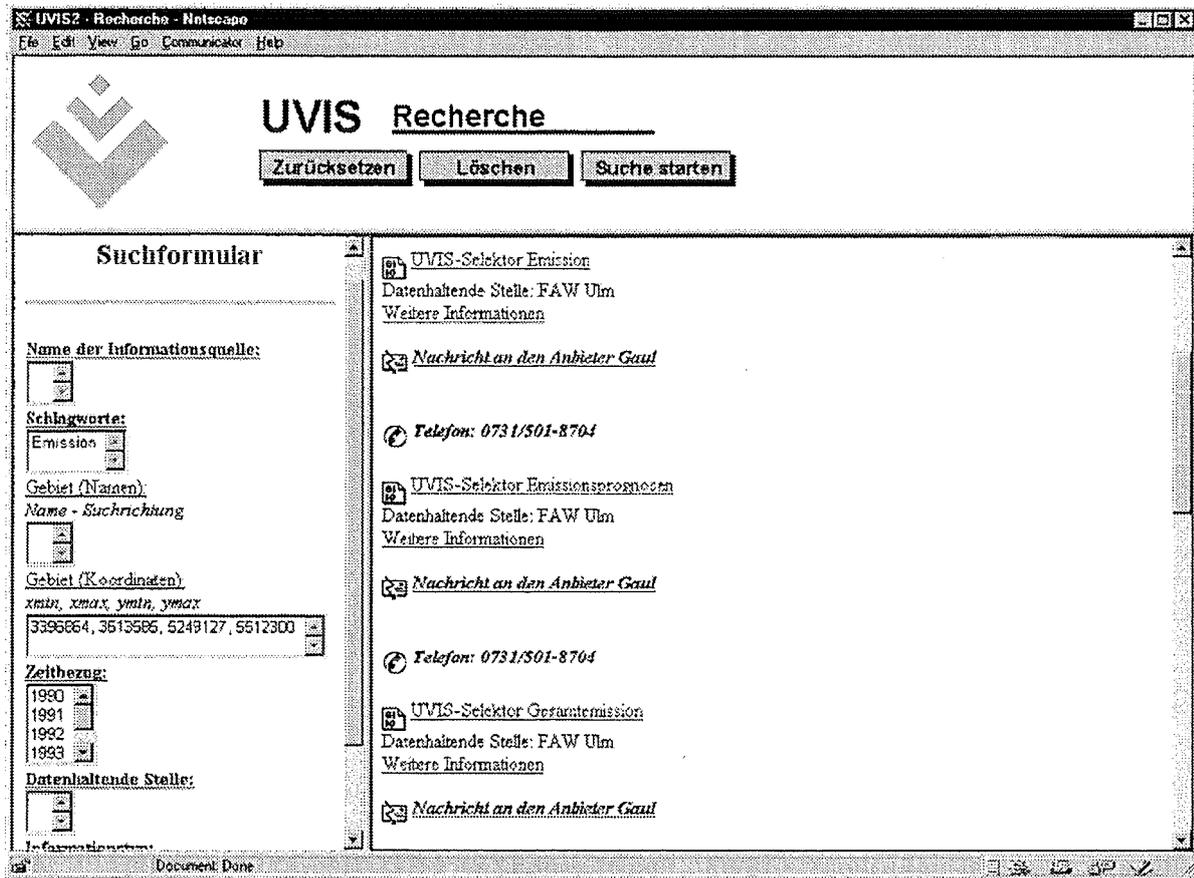
Die Entwicklung von UVIS wurde qualifiziert dokumentiert. Es wurden ausführliche Handbücher für die Anwender erstellt. Vorschläge für den weiteren Ausbau sind erarbeitet worden.

3. Nutzen

Durch die Realisierung von UVIS als verteilte Client-/ Server-Konfiguration können die Verantwortlichkeit und der Pflegeaufwand für die bereitgestellten Daten, Dienste und Dokumente bei den zuständigen Dienststellen belassen werden, wodurch eine weitgehend redundanzfreie Datenhaltung ermöglicht und administrativer Aufwand verringert wird. Das auf der Internet/WWW-Technologie basierende Vernetzungskonzept erlaubt die wirtschaftliche Nutzung der bei den einzelnen Dienststellen vorhandenen Informationsquellen mit Hilfe eines gängigen Internet-Browsers. Optional bietet der UVIS-Client zusätzliche Unterstützung bei der Aufbereitung der Informationen, wobei preisgünstige und leicht wartbare PC-Technik zum Einsatz kommt. Die Aufbereitung der Umweltinformationen auf dem UVIS-Client mit Hilfe marktgängiger PC-Applikationen minimiert Entwicklungskosten und ermöglicht es, am technischen Fortschritt in der PC-Welt ohne zusätzlichen Entwicklungsaufwand teilzuhaben. UVIS eröffnet so einen wirtschaftlichen und effektiven Weg hin zu einer computerunterstützten kooperativen Arbeitsweise in der Umweltverwaltung.

Über den Einsatz in der baden-württembergischen Verwaltung hinaus stellte UVIS eine Grundlage für die Entwicklung des Prototyps für das Umweltinformationsnetz Deutschland GEIN (German Environmental Information Network) des Bundesministeriums für Umwelt, Naturschutz und Reaktorsicherheit sowie für den Verweis- und Kommunikationsservice Umwelt des Umweltbundesamtes (VKS-Umwelt) dar. Synergien zwischen den Projekten haben UVIS, GEIN und VKS-Umwelt gegenseitig bereits in der Vergangenheit wesentlich vorangebracht und eine effiziente Systementwicklung ermöglicht. Es ist zu hoffen, daß die Synergieeffekte auch in der Zukunft fort dauern und zu einer Erweiterung des Nutzerkreises dieser Systeme beitragen werden.

4. Bildschirmabzüge



Recherche mit dem UVIS-Metainformationsserver

UVIS Selektoren

Zurücksetzen Löschen Selektor starten

Emission

Zeit: []

Kenngröße: [f]

- SO₂-Emission durch Industrie
- NO_x-Emission durch Industrie
- SO₂-Emission durch Hausbrand
- NO_x-Emission durch Hausbrand

Gebiet:

- Stuttgart, Landeshauptstadt
- Tübingen
- Tuttlingen
- Ulm, Universitätsstadt

NO _x -Emission durch Industrie	UM, Ref.43, Hr. Maxion	8416000	Tübingen
SO ₂ -Emission durch Industrie	UM, Ref.43, Hr. Maxion	8327000	Tuttlingen
SO ₂ -Emission durch Industrie	UM, Ref.43, Hr. Maxion	8421000	Ulm, Universitätsstadt
NO _x -Emission durch Industrie	UM, Ref.43, Hr. Maxion	8111000	Stuttgart, Landeshauptstadt
NO _x -Emission durch Industrie	UM, Ref.43, Hr. Maxion	8327000	Tuttlingen
NO _x -Emission durch Industrie	UM, Ref.43, Hr. Maxion	8421000	Ulm, Universitätsstadt
SO ₂ -Emission durch Industrie	UM, Ref.43, Hr. Maxion	8416000	Tübingen
SO ₂ -Emission durch Industrie	UM, Ref.43, Hr. Maxion	8337000	Waldshut

Weiter zur Aufbereitung (FNN)

Labels einblenden

Selektion von Daten mit einem UVIS-Informationsserver

5. Literatur

- /1/ I. Henning, G. Wiest, M. Gaul et al.: „Weiterentwicklung des Umwelt-Führungs-Informationssystems (UFIS) zum Umwelt- und Verkehrs-Informationsservice (UVIS) im Rahmen des Umweltinformationssystems Baden-Württemberg“. Ministerium für Umwelt und Verkehr, Stuttgart, 1998.
- /2/ I. Henning, G. Wiest, M. Gaul, E. Rainbold, A. Schultze, W.-F. Riekert: „UVIS: Umwelt- und Verkehrs-Informationsservice Baden-Württemberg“. In: W.-F. Riekert und K. Tochtermann (Hrsg.): *Hypermedia im Umweltschutz*. 1. Workshop 1998. Metropolis-Verlag, Marburg, 1998.
- /3/ I. Henning, G. Wiest und F. Schmidt: „Informationsrecherche auf neuen Wegen“. In: W. Geiger, A. Jaeschke, D. Rentz, E. Simon, T. Spengler, L. Zilliox und T. Zundel (Hrsg.): *Umweltinformatik '97*. 11. Internationales Symposium, Straßburg, September 1997, Tagungsband. Metropolis-Verlag, Marburg, 1997.
- /4/ R. Mayer-Föll, J. Pätzold (Hrsg): „Umweltinformationssystem Baden-Württemberg - Rahmenkonzeption 1998“. Ministerium für Umwelt und Verkehr / Universitätsverlag Ulm, 1998.
- /5/ W.-F. Riekert, G. Wiest, M. Gaul, B. Münst, and I. Henning: „Management of Distributed and Heterogeneous Information Resources for Environmental Administrations.“ In: R. Denzer, D.A. Swayne, and G. Schimak (eds.): *Environmental Software Systems*. Volume 2, Chapman & Hall, London, 1997.
- /6/ W.-F. Riekert, M. Gaul, G. Klügl, G. Wiest, I. Henning und F. Schmidt: „Management verteilter und heterogener Informationsquellen für das UIS Baden-Württemberg“. In: H. Lessing und U.W. Lipeck (Hrsg.): *Informatik für dem Umweltschutz*. 10. Symposium, Tagungsband. Metropolis-Verlag, Marburg, 1996.

HUDA

Hypermediatechnik für Umweltdaten

*K. Tochtermann; L. Kadric, E. Rainbold, K. Pursche,
Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW)
an der Universität Ulm,
Helmholtzstr. 16,
89081 Ulm*

*A. Keitel,
Landesanstalt für Umweltschutz,
Postfach 210 752,
76157 Karlsruhe*

1. EINLEITUNG	151
2. HINTERGRUND UND ANFORDERUNGEN	151
2.1 HINTERGRUND DER UMWELTBERICHTERSTATTUNG	151
2.2 ANFORDERUNGEN DER AUFTRAGGEBER	153
3. REALISIERUNG DES HUDA-WERKZEUGKASTENS	154
3.1 SOFTWAREARCHITEKTUR DES HUDA-WERKZEUGKASTENS	154
3.2 WERKZEUGE INNERHALB DES HUDA-WERKZEUGKASTENS	155
3.3 WERKZEUGE DES HUDA-WERKZEUGKASTENS.....	156
3.3 ERFÜLLUNG DER ANFORDERUNGEN DER AUFTRAGGEBER	157
4. LITERATUR.....	158

1. Einleitung

In den Behörden der Umweltverwaltung nimmt die Berichterstattung für Regierung und Verwaltung sowie die gesetzlich verankerte Informationspflicht gegenüber der Öffentlichkeit einen breiten Raum ein. Derzeit gibt es auf Bundesebene den Bericht „Daten zur Umwelt“ und auf Landesebene in Baden-Württemberg den Bericht „Umweltdaten“. Beide erscheinen derzeit nur in gedruckter Form, so daß für die Öffentlichkeit lediglich die für Druckwerke möglichen Zugangsformen existieren. Zudem ist die Datentransparenz auf die Möglichkeiten des Mediums Papier beschränkt. Aus diesen Gründen wird zunehmend die Erschließung dieser Informationen über neue Medien gefordert. Moderne Techniken zum Informationsmanagement versprechen darüber hinaus große Vorteile bei der Fortschreibung der Berichte.

Ziel des Vorhabens Hypermediatechnik für Umweltdaten (HUDA) ist die Konzeption und Entwicklung eines Werkzeugkastens, der es ermöglicht, insbesondere die Umweltberichte des Umweltbundesamtes (UBA) und der Landesanstalt für Umweltschutz Baden-Württemberg (LfU) in der traditionellen gedruckten Form zu erstellen. Darüber hinaus soll der Werkzeugkasten Möglichkeiten anbieten, um gleichzeitig elektronische und multimedial aufbereitete Umweltberichte auf CD-ROM und im World Wide Web (WWW) verfügbar zu machen.

HUDA wird aus Mitteln des Umweltforschungsplans des UBA finanziert. Das Projekt ist in drei Phasen aufgeteilt: In der Phase I wurde 1996 eine Machbarkeitsstudie erstellt /1/, /2/. Die Konzeption und Implementierung eines Funktionsprototyps für HUDA war Ergebnis der Phase II /3/. In der Phase III wird der Funktionsprototyp zu einer Betriebsversion ausgebaut.

Gegenstand dieses Beitrages ist die Präsentation der in Phase II erzielten Ergebnisse. Für die Durchführung der Phase II beauftragte das Ministerium für Umwelt und Verkehr Baden-Württemberg (UVM) in Abstimmung mit dem UBA auf Grundlage der Machbarkeitsstudie und den Ergebnissen des Vorhabens Globale Umweltsachdaten (GLOBUS) im Umweltinformationssystem Baden-Württemberg /4/ ein Entwicklungskonsortium aus Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW) Ulm, Forschungszentrum Informatik Karlsruhe (FZI), Forschungszentrum Karlsruhe (FZK) und dem Lehrstuhl für Softwaretechnologie an der Universität Dortmund. Die Projektleitung lag bei der LfU.

2. Hintergrund und Anforderungen

Um die Anforderungen der Auftraggeber an den HUDA-Werkzeugkasten verständlich zu machen, ist es notwendig, auf einige Randbedingungen der öffentlichen Umweltberichterstattung bei den beauftragenden Institutionen hinzuweisen. Diese werden zunächst beschrieben. Anschließend werden die zentralen Anforderungen der Auftraggeber aufgelistet.

2.1 Hintergrund der Umweltberichterstattung

Sowohl beim UBA als auch bei der LfU wird die Erstellung medienübergreifender Umweltberichte von einem eigenen Redaktionsfachgebiet wahrgenommen, das im wesentlichen Beiträge Dritter, sowohl institutionseigener als auch externer Facheinheiten, übernimmt und

redaktionell zu einem Gesamtbericht zusammenführt. Bei allen erforderlichen Änderungen bezüglich der Überarbeitung von Texten, der Aggregation von Daten und der Herstellung eines einheitlichen Layouts darf die Verantwortung der datenliefernden Stellen für den fachlichen Inhalt nicht in Frage gestellt werden. Hieraus ergibt sich die Notwendigkeit, jeden relevanten Bearbeitungsschritt von der ursprünglichen Beitragslieferung bis zum fertigen, in den Bericht integrierten Kapitel, zu dokumentieren.

Die Herausgabe eines umfassenden Umweltberichts steht in der Führungsverantwortung der höchsten Umweltbehörden des Bundes bzw. des Landes. Bisher beruht die Freigabe des Berichts zur Veröffentlichung auf der Prüfung des vollständigen Berichtsentwurfs als gedrucktem Werk. Der Prüfprozeß kann erhebliche Zeit in Anspruch nehmen, da oft eine inhaltliche Abstimmung zwischen mehreren teilverantwortlichen Stellen herbeigeführt werden muß. Entscheidend ist, daß die Freigabe sich im Grundsatz genau auf die Darstellung bezieht, die zur Prüfung vorgelegt wurde.

Hieraus ergeben sich folgende Konsequenzen: Um den Prüfprozeß und die ebenfalls zeit- aufwendigen Vorbereitungen zum Druck zeitlich zu entflechten, sind die Redaktionsfachgebiete beim UBA und bei der LfU dazu übergegangen, bereits den Berichtsentwurf vollständig in einer Form aufzubereiten, die bei kurzfristig notwendigen Änderungen keine Nachbearbeitung seitens der zu beauftragenden Druckereien mehr erforderlich macht. Hierzu wird bei den Redaktionsfachgebieten das in der Druckindustrie verbreitete Desktop-Publishing-Werkzeug QuarkXPress /5/ eingesetzt. Da das Prüfexemplar zur Freigabe der Veröffentlichung weitestgehend dem künftigen Druckwerk entsprechen muß, muß der erste vollständige Berichtsentwurf ein QuarkXPress-Dokument sein. Derzeit existiert kein geeignetes Verfahren, ein digitales, nicht-lineares Dokument, also einen Hypermedia-Bericht, den Dienststellenleitungen zur Prüfung auf Freigabe vorzulegen. Bis einschlägige Erfahrungen mit den verschiedenen Publikationsmedien vorliegen, sind die Verantwortlichen darauf angewiesen, daß sich die Hypermedia-Version inhaltlich nicht vom gedruckten Bericht unterscheidet. Dies bedeutet, daß Darstellungen in digitaler, nicht-linearer Form, die ein Berichtsergebnis hervorragend illustrieren, aber auf Papier praktisch nicht prüfbar sind, auch für das Hypermedia-Angebot auf CD-ROM oder WWW zunächst nicht verwendet werden können. Ein großer Vorteil der Hypermediatechnik kann somit, bis auf speziell aufbereitete Einzelfälle, für die Umweltberichterstattung der Behörden nicht genutzt werden - zumindest so lange nicht, bis die verantwortlichen Stellen hinreichend Erfahrungen mit diesen neuen Medien gesammelt haben.

Abgesehen davon, daß diese Erfahrungen von den Redaktionsfachgebieten und den Dienststellenleitungen nunmehr in der Praxis erworben werden müssen, gibt es dennoch gute Gründe, so bald wie möglich die Umweltberichterstattung auf die Medien Internet und CD-ROM auszuweiten: Entscheidend ist, daß die bisher in zweijährigem Abstand erscheinenden Umweltberichte nur einen winzigen, hochaggregierten Ausschnitt der in den Umweltbehörden vorliegenden Informationen darstellen, daß also ein großes Informationsangebot über den Bericht hinaus, inhaltlich aber mit ihm verknüpft, für die Öffentlichkeit bereitgestellt werden kann und soll. Beispielsweise zeigt die Erfahrung, daß die Fachbeiträge der datenliefernden Stellen oft viel mehr Detailinformationen enthalten als aus Gründen der Druckkosten und des Gesamtlayouts in einem gedruckten Bericht veröffentlicht werden können. Wo diese, oftmals methodisch wichtigen Details bisher herausgekürzt und verworfen werden mußten, sind sie in einer Hypermedia-Version ohne weiteres mit einem Verweis vom

Berichtskern aus zugänglich zu machen. Ein weiterer Vorteil der Hypermedia-Berichterstattung liegt darin, daß die Publikation von Teilen des Gesamtberichts, die bereits abgestimmt und freigegeben sind, in der WWW-Version vorgezogen und damit das Informationsangebot wesentlich aktueller gestaltet werden kann.

2.2 Anforderungen der Auftraggeber

Aus den im vorherigen Abschnitt genannten Gründen ergaben sich für die Entwicklung des HUDA-Werkzeugkastens konkrete Vorgaben, die in der Entwicklung des vorliegenden Prototyps umzusetzen waren. Ohne hier auf die etwas unterschiedlichen Randbedingungen beim UBA und bei der LfU einzugehen, werden im folgenden die wesentlichen Anforderungen an HUDA aufgeführt:

1. Integration in die Arbeitsumgebung der Endanwender: Generell muß HUDA in die bestehenden Arbeitsumgebungen der Redaktionsfachgebiete beim UBA und bei der LfU vollständig zu integrieren sein. Höchste Priorität hat die Nutzbarkeit durch diese auf Umweltberichterstattung spezialisierten Mitarbeiter. Darüber hinaus soll die Anforderung berücksichtigt werden, den Werkzeugkasten so zu gestalten, daß er möglichst breit eingesetzt werden kann. Dies hat aber hinter der ersten Vorgabe zurückzustehen. Da die erste Version des Gesamtberichts immer als QuarkXPress-Dokument vorliegt, ist es sinnvoll, die WWW- und CD-Version auf der Basis dieses QuarkXPress-Dokuments zu erstellen. Die in der Machbarkeitsstudie dargestellte Lösungsmöglichkeit, die Druckversion vollständig parallel zur Hypermedia-Version zu erstellen, hat sich wegen der beschriebenen Restriktionen im Erstellungs- und Prüfprozeß als ungeeignet herausgestellt.

2. Offenheit und Flexibilität: Eine wichtige Anforderung, die im Zusammenhang der Integration in die Arbeitsumgebung der Endanwender zu nennen ist, ist die offene und flexible Gestaltung des HUDA-Systems. Offenheit ist wichtig, um weitere Werkzeuge zu einem späteren Zeitpunkt mit möglichst geringem Implementierungsaufwand in das HUDA-System einbetten zu können. Flexibilität ist wichtig, um derzeit verwendete Werkzeuge innerhalb des Werkzeugkastens zu einem späteren Zeitpunkt mit möglichst wenig Aufwand durch andere ersetzen zu können.

3. Verwaltungssystem: Für die Ablage und das Handling der Berichtsbestandteile wird ein Verwaltungssystem benötigt. In diesem Verwaltungssystem sollen die im Bearbeitungsprozeß aufeinander folgenden Arbeitsschritte und die hierbei entstehenden Versionen einschließlich der zur Bearbeitung benötigten Zusatz- bzw. Rahmeninformationen dokumentiert werden können. Die Bedeutung eines solchen zentralen Verwaltungssystems und eine beispielhafte technische Realisierung zeigt die Entwicklung des Digitalen Umweltatlas Hamburg /6/. Wie im Abschnitt 3 dargestellt, wird in HUDA hierzu eine erweiterte Version der im UBA eingesetzten Datenbank „Grunddaten zur Umweltberichterstattung“ (DB Grunddaten) verwendet /7/. Wichtig ist aber dabei, daß diese Berichtsdatenbank auch außerhalb des HUDA-Arbeitsplatzes wie bisher im UBA weiter genutzt werden kann. Mit dem Verwaltungssystem ist sicherzustellen, daß bei einem Personalwechsel der Bearbeitungsstand gut nachvollziehbar ist und daß bei der Herausgabe von Folgeberichten nach oftmals langen Zeiträumen auf den Versionen des Vorläufer-Berichts leicht aufgesetzt werden kann.

4. Einbindung vorhandener Informationsangebote: HUDA muß die inhaltliche Einbindung des Berichts und seiner Bestandteile in das umfassende Informationsangebot des UBA und der LfU unterstützen. Hierzu wird eine Verweisverwaltung benötigt, mit der nicht nur Verweise innerhalb des Berichts, sondern vor allem zu den weitergehenden WWW-verfügbaren Daten und Dokumenten kontrolliert und bearbeitet werden können.

Eine weitere wichtige Verbindung zu den Informationen außerhalb des Berichts stellt die Einbindung des Umweltdatenkatalogs (UDK) in den HUDA-Werkzeugkasten dar. Der UDK ist das von einer Bund/Länder-Kooperation betriebene Metadatensystem für den Umweltbereich, in dem Beschreibungen der umweltrelevanten Datenbestände und Dokumente des Bundes und der Länder mit Zugangsinformationen bereitgestellt werden. Es ist sicherzustellen, daß hierfür geeignete Berichtsteile aus HUDA heraus als UDK-Objekte beschrieben werden können, so daß beispielsweise bei einer Schlagwortsuche im UDK auf das entsprechende Kapitel im Umweltbericht verwiesen wird.

3. Realisierung des HUDA-Werkzeugkastens

Dieser Abschnitt stellt zunächst die Architektur des HUDA-Werkzeugkastens und wichtige darin vorkommenden Werkzeuge vor. Abschließend wird darauf eingegangen, inwieweit die Anforderungen der Auftraggeber erfüllt wurden.

3.1 Softwarearchitektur des HUDA-Werkzeugkastens

Die HUDA-Softwarearchitektur ist im Hinblick auf die Anforderung entwickelt worden, einen flexiblen und erweiterbaren Werkzeugkasten zu schaffen, dessen Bestandteile nicht nur von QuarkXPress, sondern auch von anderen Werkzeugen aus genutzt werden können. Während die Forderungen nach Erweiterbarkeit und die Verwendbarkeit durch externe Anwendungen durch den Einsatz einer komponentenbasierten Technologie erfüllt wird, ist zur Erzielung einer größtmöglichen Flexibilität eine geeignete Architektur zu schaffen. Dies erreicht die HUDA-Softwarearchitektur durch die Trennung der drei Schichten Präsentation, Verwaltung und Datenzugriff.

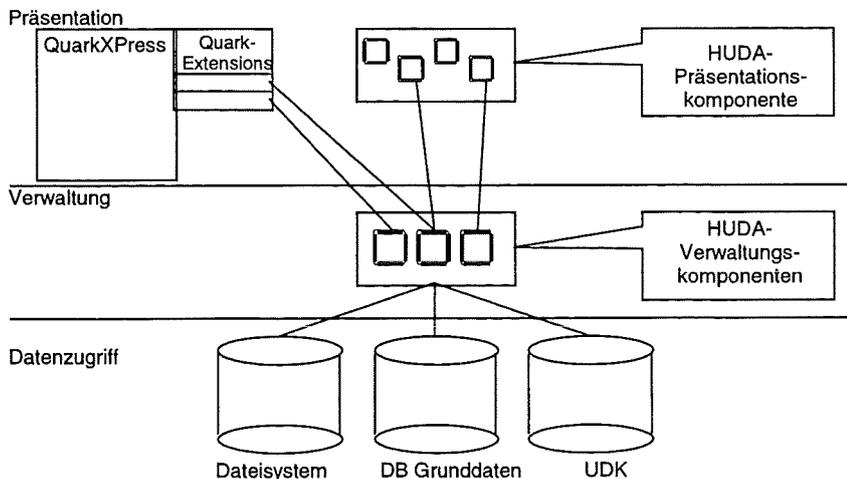


Abbildung 1: Softwarearchitektur von HUDA

Die Präsentationsschicht dient zur Kommunikation mit dem Benutzer. Dies geschieht in Form von Präsentationskomponenten, die als ActiveX-Controls ausgeführt sind, und sich so in den meisten Werkzeugen für die Anwendungsentwicklung verwenden lassen. Zur Präsentationsschicht werden etwa QuarkXPress und QuarkXTensions gezählt. Für beide gilt jedoch, daß sie keine Verwaltungsfunktionalität beinhalten. Diese befindet sich in der Verwaltungsschicht von der HUDA-Softwarearchitektur, die aus einzelnen HUDA-Verwaltungskomponenten besteht. Durch die Trennung von Präsentations- und Verwaltungsschicht kann die Duplizierung von Funktionalität in QuarkXTensions und extern zu nutzenden Komponenten vermieden werden. Die HUDA-Verwaltungskomponenten sind ihrerseits von der Datenhaltung durch die Datenzugriffsschicht entkoppelt. Aufgabe der Datenzugriffsschicht ist es, den transparenten Zugriff auf unterschiedliche Datenquellen wie die Datenbank Grunddaten oder den UDK zu ermöglichen. Dabei kommen die sogenannten Data Access Objects (DAOs) zum Einsatz, die den Zugriff auf unterschiedliche Datenbanken kapseln. Zum Zugriff auf den in einer ORACLE-Datenbank befindlichen UDK kann z.B. ODBC Treiber (Open Database Connectivity) verwendet werden.

3.2 Werkzeuge innerhalb des HUDA-Werkzeugkastens

Die Machbarkeitsstudie und am Anfang der Phase II durchgeführte Befragungen zu HUDA hatten ergeben, daß der HUDA-Werkzeugkasten ein zentrales Datenverwaltungssystem, ein Datenaufbereitungssystem, ein Redaktionssystem, ein Metainformationssystem sowie ein Layout- und Konfigurationssystem enthalten sollte.

Zentrales Datenverwaltungssystem: Den Mittelpunkt des HUDA-Werkzeugkastens bildet das zentrale Datenverwaltungssystem, in dem alle Berichtsbestandteile eines Umweltberichtes möglichst redundanzfrei abgelegt werden. Dieses Verwaltungssystem bietet Schnittstellen zu den anderen Systemkomponenten, die Zugriff auf die Berichtsteile haben müssen. Wie die Befragungen ergaben, liegt beim UBA mit der DB Grunddaten bereits ein System vor, das als Arbeitsinstrumentarium des Berichterstatters zur Verwaltung hochaggrierter Berichtsbestandteile eingesetzt wird. Eine genaue Analyse der DB Grunddaten ergab, daß diese als Grundlage für das zentrale Datenverwaltungssystem im Rahmen von HUDA verwendet werden kann.

Datenaufbereitungssystem: Die Berichtsbestandteile kommen üblicherweise von verschiedenen datenliefernden Stellen. Elektronisch abgelieferte Berichtsbestandteile liegen in der Regel in Formaten vor, die kompatibel mit der Microsoft-Office Umgebung sind. Da sie von verschiedenen Stellen erstellt und abgeliefert werden, muß ihr Layout zunächst durch das Datenaufbereitungssystem vereinheitlicht werden, bevor sie nach inhaltlichen Abstimmungsprozessen für die Gesamtintegration in den Bericht aufgenommen werden können.

Redaktionssystem: Die Gesamtintegration von Berichtsbestandteilen zu einem Bericht erfolgt durch ein Redaktionssystem. Die Endanwender setzen für die Bürokommunikation zum Teil Microsoft-Office-Komponenten ein. Die zentrale Rolle nimmt jedoch das Desktop Publishing Werkzeug QuarkXPress ein.

Metainformationssystem: Das Metainformationssystem dient dazu, Berichtsbestandteile über das WWW recherchierbar zu machen. Als Metainformationssystem wird auf den Umweltdatenkatalog (UDK) zurückgegriffen.

Konfigurations- und Layoutsystem: Die Aufbereitung des Gesamtberichtes entsprechend der drei Medien Druck, CD-ROM und WWW wird schließlich in dem Konfigurations- und Layoutsystem vorgenommen. In HUDA wurde für die CD-ROM- und WWW-Aufbereitung ein auf HexWeb /8/ basierendes Werkzeug für die Konvertierung von QuarkXPress nach HTML entwickelt.

3.3 Werkzeuge des HUDA-Werkzeugkastens

Dieser Abschnitt stellt einige der Werkzeuge des HUDA-Werkzeugkastens vor. Während der Berichtserstellung kann über den HUDA-Explorer die Struktur von Berichten eingesehen und einfach verändert werden. Dies ist von großer Bedeutung, da in der Regel die Berichtserstellung mit einer Grobstruktur des Berichtes beginnt, die sich aber im Verlaufe der Erstellung häufig ändert. Der HUDA-Explorer bietet zur Unterstützung bei der Umstrukturierung von Berichten die Möglichkeit, Berichtselemente einfach zu kopieren, zu verschieben und zu löschen

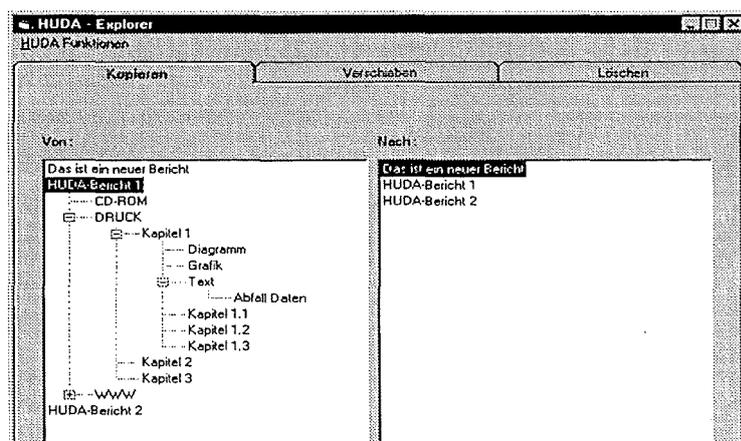


Abbildung 2: Der HUDA-Explorer

Um mit HUDA erstellte Berichte und darin enthaltene Kapitel im Umweltdatenkatalog (UDK) nachweisen zu können, wurde eine Schnittstelle zwischen dem UDK und der DB Grunddaten entwickelt. Diese Schnittstelle ermöglicht es, bereits in der DB Grunddaten vorhandene Metainformationen in den UDK zu übernehmen. Zusätzlich können weitere UDK-spezifische Metainformationen erhoben werden.

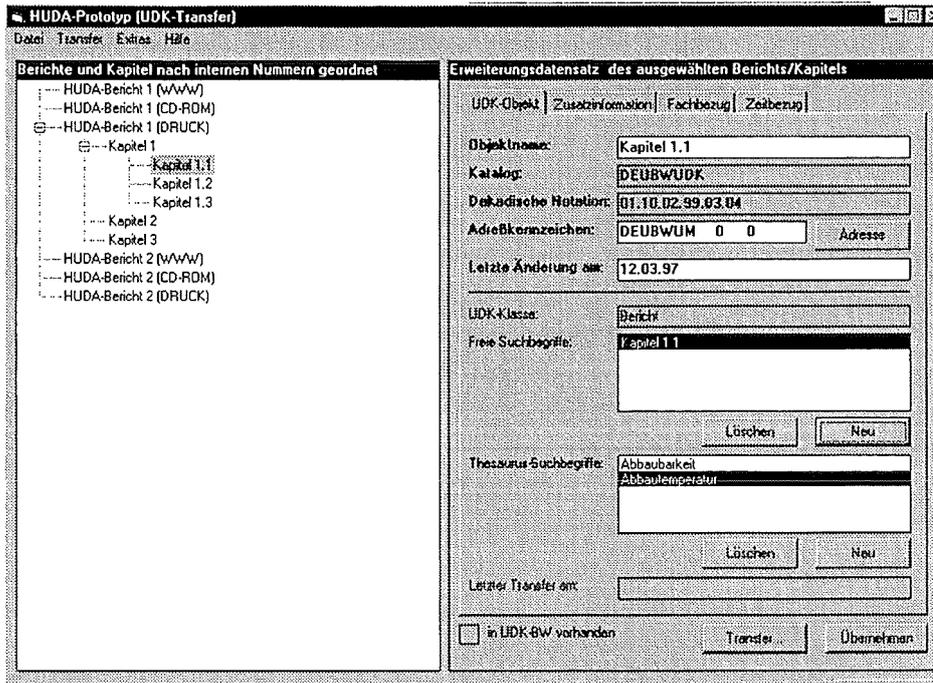


Abbildung 3: Erfassung von Metainformationen für den UDK

Die nachfolgende Abbildung zeigt, wie aus QuarkXPress heraus das Konfigurationssystem zur Erzeugung der HTML-Dokumenten aufgerufen werden kann. Damit ist es möglich, unter Verwendung von HexWeb, ein geöffnetes Quark-Dokument einfach in ein HTML-Dokument zu konvertieren.

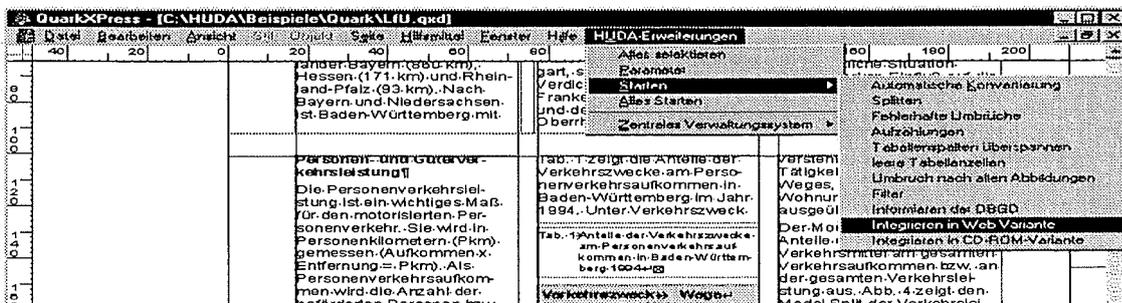


Abbildung 4: HTML-Konvertierung aus QuarkXPress heraus

3.3 Erfüllung der Anforderungen der Auftraggeber

Dieser Abschnitt diskutiert, inwieweit durch die oben genannten Werkzeuge die in Abschnitt 2 genannten Anforderungen erfüllt werden können.

Integration in die Arbeitsumgebung der Endanwender: Die Datenaufbereitung berücksichtigt die Arbeitsumgebung der Endanwender. Hier sind im wesentlichen Programme der MS Office Suite integriert worden, um z.B. die Generierung einer Businessgrafik mit Excel aus einer in Word vorliegenden Tabelle schnell zu ermöglichen. Schließlich ist der HUDA-Werkzeugkasten auf QuarkXPress als zentralem Bestandteil des Redaktionssystem zugeschnitten. Die Anforderung der Integration des HUDA-Werkzeugkastens in die Arbeitsumgebung der Endanwender wurde also in hohem Maße erfüllt.

Offenheit und Flexibilität: Offenheit und Flexibilität des HUDA-Werkzeugkastens wurde durch die oben beschriebene Softwarearchitektur erfüllt.

Verwaltungssystem: Mit der DB Grunddaten basiert das zentrale Datenverwaltungssystem des HUDA-Werkzeugkastens auf einem bei den Endanwendern bereits eingesetzten System. Die DB Grunddaten wurde lediglich um speziell auf die Arbeitsvorgänge in HUDA zugeschnittene Funktionalität erweitert. Im Datenbankdesign wurde eine klare Trennung von Daten und der darauf operierenden Anwendungslogik vorgenommen. Damit ist es möglich, auf ein und denselben Daten mit verschiedenen Anwendungslogiken zu operieren. Aufgrund dieser Tatsache ist es nun möglich, daß die DB Grunddaten auch außerhalb des HUDA-Arbeitsplatzes wie bisher eingesetzt werden kann.

Einbindung vorhandener Informationsangebote: In dem vorliegenden Funktionsprototypen ist der UDK als ein vorhandenes Informationsangebot bereits an den HUDA-Werkzeugkasten angebunden. Im Rahmen der Entwicklungen für die Betriebsversion sind jedoch weitaus mehr existierende Informationsangebote an HUDA anzuschließen.

4. Literatur

- /1/ Mayer-Föll, R.; Keitel, A.; Riekert, W.-F.; Tochtermann, K., 1997: Hypermediatechnik für Umweltdaten - Phase I Machbarkeitsstudie, Forschungsbericht 101 05 060 des Umweltbundesamtes, Berlin.
- /2/ Riekert, W.-F.; Keitel, A.; Schütz, Th.; Strohm J. (1997): Anforderungen und Lösungsansätze für ein hypermediales System zur Umweltberichterstattung; W. Geiger, A. Jaeschke, D. Rentz, E. Simon, T. Spengler, L. Zilliox, T. Zundel (Hrsg.): Umweltinformatik '97 - 11. Internationales Symposium. Straßburg, 10.-12. Sept. 1997. Tagungsband, Metropolis-Verlag, Marburg.
- /3/ Keitel, A.; Mayer-Föll, R.; Tochtermann, K.; Riekert, W.-F. (1997): Hypermediatechnik für Umweltdaten - Phase II Konzeption und Erstellung eines Funktionsprototyps, noch unveröffentlicht.
- /4/ Mayer-Föll, R.; Jaeschke, A. (1997): UIS Baden-Württemberg. Projekt GLOBUS. Umsetzung der neuen Systemarchitektur und Entwicklung weiterer Produktionssysteme für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg. Phase IV 1997. Wissenschaftliche Berichte, FZKA-6000. <http://www.iai.fzk.de/~weideman/doc/globus4/>.
- /5/ <http://www.quark.com/>.
- /6/ Blasius K.; Maier K., 1997: Digitaler Umweltatlas (DUA); K. Greve, A. Keitel, J. Seggelke, Th. Schütz, H.-G. Sobottka (Hrsg.): Workshop Hypermediatechnik für Umweltdaten, Internes Memorandum (Nr. 95) des Lehrstuhls Software-Technologie der Universität Dortmund, S. 45-54.
- /7/ Dokumentation der Datenbank Grunddaten Version 2. Dr. Lippke und Dr. Wagner GmbH, Berlin.
- /8/ <http://www.hexmac.com>.

Ausblick auf die F+E-Arbeiten im Rahmen von GLOBUS Phase VI 1998/99

Neben den Projektträgern Ministerium für Umwelt und Verkehr Baden-Württemberg (UVM) und Landesanstalt für Umweltschutz Baden-Württemberg (LfU) haben für die Phase VI des Vorhabens GLOBUS weitere Stellen ihre Beteiligung als Auftraggeber zugesagt:

- Bundesanstalt für Wasserbau Karlsruhe (BAW),
- Ministerium Ländlicher Raum Baden-Württemberg (MLR),
- Niedersächsisches Umweltministerium (MU NI) und die
- Stabsstelle für Verwaltungsreform im Innenministerium Baden-Württemberg (StaV/IM).

Die F+E-Arbeiten im Vorhaben GLOBUS - Phase VI führen die folgenden wissenschaftlichen Institutionen durch:

- Forschungsinstitut für anwendungsorientierte Wissensforschung FAW, Ulm
- Forschungszentrum Informatik FZI, Karlsruhe
- Forschungszentrum Karlsruhe - Institut für angewandte Informatik FZK/IAI
- Fraunhofer-Institut für Informations- und Datenverarbeitung FhG/IITB
- Institut für Kernenergetik und Energiesysteme - Universität Stuttgart IKE und
- Institut für Photogrammetrie und Fernerkundung - Universität Karlsruhe IPF.

In Fortsetzung der bisherigen Arbeiten sollen 1998/99 - untereinander abgestimmt - von den Projektpartnern die folgenden Aufgaben gelöst werden.

1. F+E-Arbeiten des FAW

Aufbau, Pflege und Nutzung multimedialer Bestände von Umweltdokumenten in Bürokommunikationssystemen

In den bisherigen GLOBUS-Projekten wurden Internettechniken im wesentlichen eingesetzt, um Umwelt-Fachdokumente für eine große Zahl von Nutzern leicht recherchierbar und zugreifbar zu machen. Die geplanten Arbeiten des FAW konzentrieren sich nun darauf, wie Nutzer über ihre normale Bürokommunikationsumgebung Intranet / Internet nutzen können, um Dokumente aus ihrer täglichen Arbeitsumgebung einfach in einen multimedialen Umweltdatenbestand einstellen und damit anderen Nutzern zugänglich machen können.

Im Gegensatz zu den Systemen der XfaWeb-Familie und HUDA ist die geplante Entwicklung kein Redaktionssystem zur Erstellung von durchgängigen, in sich geschlossenen Berichten. Vielmehr wird ein multimediales Dokumentenarchiv aufgebaut, das die Wiederverwendbarkeit

und damit die Mehrfachnutzung von 'leichtgewichtigen' Arbeitsergebnissen in der Umweltverwaltung ermöglicht.

Ziel der FAW-Entwicklungen in 1999 ist es daher, unter Einsatz moderner Internettechnologien, ein Werkzeugsystem zu entwickeln, das den Aufbau, die Pflege und die Nutzung von multimedialen Umweltdatenbeständen durch ein breites Spektrum von Normal-Nutzern aus deren Standard-Arbeitsplatzumgebung heraus ermöglicht.

Über die Autorenkomponente des Systems können Nutzer - als Autoren - die von ihnen angefertigten Umweltdokumente auf einem Indexserver einstellen. Die Recherchekomponente ermöglicht dem Nutzer des Systems die komfortable Suche (Umweltthesaurus, Volltext etc.) und den Zugriff auf den Dokumentbestand über gängige Web-Browser. Über die Administratorkomponente werden die Dokumentbestände auf dem Dokumentenserver verwaltet und Nutzerrechte für Autoren und Informationssuche vergeben.

Die für 1999 geplanten Arbeiten beinhalten eine Analyse der aktuellen Marktsituation im WWW / BK - Bereich, eine Analyse der bestehenden Anforderungen der BK-Nutzer und darauf aufbauend die Erstellung eines Konzepts zur Bereitstellung von Dokumenten im Intranet/Internet für die bestehenden BK- und Intranet-Umgebungen sowie die Umsetzung bei UVM und LfU.

Das FAW ist darüberhinaus beteiligt an den Arbeiten zur Auswahl eines UVM/LfU-einheitlichen E-Commerce-Abrechnungssystems (s. Abschnitt 7).

2. F+E-Arbeiten des FZI

Die steigende Nachfrage nach querschnittsorientierter Information bei den UIS-Nutzern, hat den Bedarf für ein querschnittsorientiertes, netzbasiertes Werkzeug zur Datenrecherche und Auswertung gezeigt. Ein zentraler Punkt in den F+E-Arbeiten des FZI ist daher die Pflege und Weiterentwicklung des Sachdatensystems SDS. Weiter werden vom FZI in 1999 Arbeiten zur Integration des WWW-UDK 4 in das UIS Baden-Württemberg und zur Nutzung neuer Internet-Technologien und Werkzeuge durchgeführt.

Weiterentwicklung des Sachdatensystems SDS

Als neue Komponente für das Sachdatensystem wird vom FZI ein Selektor-Generator entwickelt. Dieser Modul unterstützt den Anwendungsbetreuer beim Anlegen von Selektoren und den dazugehörigen Conditions sowie deren Verwaltung. Für die Erstellung und Verarbeitung von Reports ist die Realisierung eines Report-Generators geplant.

In Rahmen des Ausbaus des Sachdatensystems wird außerdem der Java-basierte UDK-Navigator als "Meta-Selektor" erweitert, an das Sachdatensystem angepaßt und integriert und weiterhin ein Aufbereitungsdienst realisiert, der die statistische Aufbereitung von Selektorergebnissen ermöglicht, die dann wie ein normales Selektionsergebnis in tabellarischer Form ausgegeben, mit Hilfe des GISterms visualisiert oder durch MS-Office Produkte weiterverarbeitet werden können.

Ferner wird der Projektionsdienst im aktuellen SDS erweitert und verbessert und eine Historienverwaltung eingeführt, die es dem Benutzer ermöglicht, auf innerhalb einer Session durchgeführte Selektoren zurückzugreifen und bereits durchgeführte Selektionen wieder aufzurufen.

Integration WWW-UDK 4 in das UIS Baden-Württemberg

Im Auftrag der Koordinierungsstelle Umweltdatenkatalog beim Umweltministerium Niedersachsen hat das FZI die Version WWW-UDK 4 entwickelt. Ziel ist es, WWW-UDK 4 in das UIS Baden-Württemberg zu integrieren. Dies betrifft das Zusammenspiel mit unterlagerten Datenquellen, die Katalogdaten für den Umweltdatenkatalog liefern, sowie mit den UIS-Komponenten, die auf den UDK und auf im UDK beschriebene Daten des UIS zugreifen.

Nutzung neuer Internet-Technologien und Werkzeuge

In diesem Rahmen sind die neuen Internet-Technologien und Werkzeuge bezüglich ihres Einsatzpotentials im UIS Baden-Württemberg zu untersuchen und an praktischen Beispielen aus den obigen Anwendungsszenarien prototypisch zu erproben.

Hierzu gehören auch die unten (Abschnitt 7) definierten Arbeiten des FZI zur Auswahl eines UVM/LfU-einheitlichen Abrechnungsverfahrens bzw. -systems für die Informationsverbreitung über Netz.

3. F+E-Arbeiten des FZK

Die F+E-Arbeiten 1999 an den Systemen der Fachinformationssystem-Familie XfaWeb beinhalten

- zum einen die Neu- und Weiterentwicklung allgemeiner Komponenten und Werkzeuge der Familie, wobei hier die Integration von Karten im Vordergrund steht und
- zum anderen die Weiterentwicklung der Einzelsysteme AlfaWeb, BofaWeb und NafaWeb.

Allgemeine Komponenten und Werkzeuge der XfaWeb-Familie

Für die Verwendung von Karten in den XfaWeb-Systemen sollen für NafaWeb, BofaWeb und AlfaWeb die Anforderungen von fachlicher Seite und informations-technischer Seite ermittelt und die bereits vorliegenden Karten-Datenbestände hinsichtlich ihrer Verwendbarkeit analysiert werden. Darauf aufbauend soll ein Konzept für die XfaWeb-Kartenkomponente inkl. Geodaten- und Sachdaten-Teil, Schnittstellen und Zugangsstruktur erarbeitet werden.

In diesem Zusammenhang ist auch ein informationstechnisches Konzept für die Verbreitung der Informationen über mehrere, untereinander vernetzte CD-ROMs auf der Basis von WWW-Werkzeugen zu entwickeln.

Die vorhandenen Werkzeuge (Konvertierungsprogramm, Systemverwaltungskomponente, ..) sind basierend auf den Einsatzerfahrungen und unter Berücksichtigung neuer technischer Möglichkeiten weiterzuentwickeln. Die XfaWeb-CD-ROM-Software soll so erweitert werden, daß eine Nutzung der dynamischen Seiten der CD-ROM auch bei einem Betrieb der CD über einen CD-ROM-Server möglich wird. In diesem Rahmen sollen auch weitere intelligente Zugangswege konzipiert und implementiert werden.

Mit dem Ziel einer Integration der verschiedenen Systeme der XfaWeb-Familie soll ein übergreifendes Zugangssystem konzipiert und für gemeinsame Informationen (z.B. Gesetzestexte) Möglichkeiten zur redundanzfreien Verwaltung geschaffen werden.

FZK/IAI ist bei der Auswahl eines für UVM / LfU einheitlichen Abrechnungssystems für E-Commerce (s. Abschnitt 7) beteiligt. Das selektierte System wird von FZK für die XfaWeb-Fachinformationssysteme eingesetzt, um damit die Vollversionen der Fachsysteme über Netz verfügbar machen zu können.

AlfaWeb

Der inhaltliche Ausbau von AlfaWeb ist bereits weitgehend erfolgt und wird jeweils durch neu erscheinende Berichte auf den aktuellen Stand gebracht. Bei Bedarf werden eine neue Intranet-Version, eine neue Schaufenster-Version und eine neue CD-ROM (Master) von AlfaWeb erstellt.

Im Rahmen der Integration von Karten in AlfaWeb werden die im Altlastenbereich vorliegenden Sachdaten und Karten in Abstimmung mit der AlfaWeb-Steuerungsgruppe ausgewählt, aufbereitet und die Zugriffsstrukturen auf Geodaten und Sachdaten realisiert.

BofaWeb

Basierend auf dem in 1998 entwickelten Prototyp ist BofaWeb weiterzuentwickeln. Weitere Berichte und Gesetzestexte (ca. 28) sind in Abstimmung mit der Projektsteuerungsgruppe BofaWeb aufzuarbeiten (Verschlagwortung und inhaltliche Querverweise) und in die BofaWeb-Systemverwaltungskomponente zu übernehmen (Metadaten-Erfassung, Konvertierung).

Für die Implementierung eines Fachzugangs müssen von fachlicher Seite eine Gliederung der relevanten Inhalte erfolgen und die Zugangsstrukturen festgelegt und realisiert werden.

BofaWeb soll in folgende inhaltlich und funktional verschiedene Versionen diversifiziert werden:

- Schaufenster-Version für Internet und Intranet
- vollständige Version für Internet und Intranet
- vollständige CD-ROM-Version.

Auch zu BofaWeb sollen erste Karten aufbereitet und unter Verwendung der XfaWeb-Kartenkomponente integriert werden.

NafaWeb

Der Schwerpunkt der Arbeiten an NafaWeb liegt in der Integration von Karten. Gemeinsam mit der NafaWeb-Projektsteuerungsgruppe wird das zu integrierende Kartenangebot, die Sachdaten zu den Karten (Datenbanken) sowie der Umfang der anzubietenden Verknüpfungen ("links") zwischen Geodaten und Sachdaten festgelegt. Das ausgewählte Datenmaterial wird aufbereitet und in geeignete Formate umgesetzt; die Zugriffsstrukturen für Sach- und Geodaten werden konzipiert und realisiert und Such- und Navigationshilfen zu diesen Daten erstellt.

Daneben werden weitere Berichte aus dem Bereich Naturschutz und Landschaftsschutz für das System aufbereitet und in die verschiedenen Zugänge von NafaWeb integriert.

4. F+E-Arbeiten von FhG / IITB

Die im Rahmen von GLOBUS VI für 1999 geplanten Arbeiten des IITB konzentrieren sich auf Aspekte in der Realisierung des Informationssystems Wasser, Abfall, Altlasten, Boden WAABIS und umfassen Arbeiten zum Qualitätsmanagement, Datenmanagement und WAABIS-Framework.

Qualitätsmanagement

Das Vorhaben WAABIS als großes IuK-Verfahren mit vielen Teilkomponenten birgt Risiken für eine termin- und sachgerechte Realisierung. Zum gesteuerten Umgang mit den Entwicklungsrisiken ist die Erstellung und Umsetzung eines angepaßten Konzepts zur Qualitätssicherung erforderlich. Dies umfaßt die Unterstützung der LfU bei der Erstellung eines Prüfplans, bei der Prüfungsvorbereitung und Durchführung.

Datenmanagement

Eine zentraler Aspekt bei der Realisierung von WAABIS ist die Verwaltung der auf unterschiedliche Behörden und Dienststellen verteilten Daten. Die WAABIS-Kerndaten sind in den einzelnen Dienststellen zu halten und im Rahmen des WAABIS-Verbunds zwischen den beteiligten Behörden auszutauschen. Es ist die Konzeption zum Datenmanagement zu konkretisieren und die LfU bei der Erstellung und Umsetzung von Regelwerken für das lokale Datenmanagement und den Datenaustausch sowie beim Aufbau und Betrieb der WAABIS-Referenzdatenbank zu unterstützen.

Das Gesamtsystem WAABIS löst verschiedene Altssysteme ab. Zur weiteren Nutzung der Datenbestände ist eine Konzeption für die Übernahme und Integration von bereits bestehenden Datenbeständen in die Datenbanktabellen der neuen WAABIS-Module zu erstellen, zu implementieren und zu testen.

Weiterentwicklung WAABIS-Framework

Für WAABIS wird ein eigenes Entwicklungs-Framework aufgebaut. In der ersten Phase wurde bereits eine Basisschicht entwickelt, die eine Datenbankzugriffsebene für die Realisierung der Objektpersistenz enthält. Im geplanten weiteren Ausbau des WAABIS-Systems werden jetzt gemeinsam nutzbare Fachfunktionalität innerhalb der Anwendungsmodule identifiziert und in Form von Diensten in das Framework zur allgemeinen Nutzung einbezogen.

Die Arbeiten des IITB umfassen die Unterstützung der LfU bei der Integration bestehender Dienste in das Framework und bei der Konzeption und Realisierung neuer Dienste für die WAABIS-Module.

5. F+E-Arbeiten des IKE

Beschreibung des System-Umfeldes für den Einsatz von Diensten über Ontologien

Ziel dieser Arbeiten des IKE ist die Entwicklung eines Konzeptes für die ontologische Beschreibung von Informationen in Umweltinformationssystemen und der Nachweis der Brauchbarkeit dieser Vorgehensweise in realen Systemen.

Eine formale Beschreibung der im System vorhandenen Informationen und Abhängigkeiten ist notwendig, um eine allgemeine, einheitliche Basis für die Kommunikation zwischen den Diensten und zwischen Dienst und Anwender zu schaffen. Ein weiterer Vorteil der formalen ontologischen Beschreibung ist die Möglichkeit, die Konsistenz der in der Ontologie beschriebenen Informationen zu prüfen.

Basis einer solchen Ontologie ist die Beschreibung der Informationen durch Daten und ihre Metadaten. Dies wird ergänzt durch kontextabhängige Beziehungen zwischen den Informationen.

Beschreibung von Dienstketten über Workflows

Ziel ist die Entwicklung einer Workflow-Beschreibungssprache für Anwendungen, deren Ablauf aus der sequentiellen Ausführung von Computerprogrammen (Diensten) bestehen. Dabei soll ein Workflow-Ansatz erprobt und der Nutzen bei der Anwendung in realen Systemen nachgewiesen werden.

Informationen über den Ablauf sollen nicht mehr bei den einzelnen Diensten, sondern zentral gespeichert und verwaltet werden. Auf der Basis der Beschreibungssprache muß die Durchführung der Arbeitsabläufe nicht mehr fest verdrahtet, sondern kann flexibel und zur Laufzeit eingelesen und abgearbeitet werden.

Entwicklung von Systemdiensten für das UIS und die KFÜ

In GLOBUS V wurden bereits für das UIS und das KFÜ-System verschiedene, allgemeine Systemdienste entworfen. Diese Dienste werden in 1999 vom IKE mit Hilfe von Visual C++ implementiert. Es handelt sich dabei um folgende Dienste:

- Strategie-Service,
- Workflow-Service,
- Resource-Service,
- Repository-Service und
- Daten-Service.

6. F+E-Arbeiten des IPF

Weiterentwicklung und Konsolidierung des JAVA-basierten GIS-Terminals für das UIS

Aufbauend auf dem vorhandenen Prototyp GISTerm soll ein für das UIS im Betrieb einsetzbarer GIS-Viewer mit flexibler Einbindung in Sachdatensysteme realisiert werden. Dazu wird das GISTerm im Rahmen der Vorhaben GLOBUS und WAABIS-Geo funktional weiterentwickelt. Dies umfaßt u.a. die Erweiterung der Funktionen für die kartographische Darstellung (Symbolverwaltung, Füllmuster, transparente Raster, ...), die Erstellung eines Präsentationsmoduls (Nutzereinstellungen, Standardpräsentationen, Plotverwaltung), die Weiterentwicklung des Klassifikationsmoduls (statistische Aufbereitung) und die Integration in die Office-Umgebung mit den Schnittstellen zu Clipboard und Standard-Vektorformaten (WMF, Postscript).

Diese funktionalen Erweiterungen werden ergänzt durch Arbeiten zur Konsolidierung der Betriebsversion GISTerm/WAABIS-Geo.

Für die Einbindung der WAABIS-Geo-Komponenten in die WAABIS-Fachmodule und das Berichtssystem werden ein Konzept entwickelt, die Programm-Schnittstellen mit den WAABIS-Projektpartnern abgestimmt und die WAABIS-Geokomponenten in das WAABIS-Modul Grundwasser integriert.

7. Gemeinsame F+E-Arbeiten von FAW, FZI und FZK/IAI

Electronic Commerce - Abrechnungssysteme (FZI, FAW, FZK/IAI)

Im Rahmen dieser Aufgabe, die zusammen von FZI, FAW und FZK/IAI bearbeitet wird, soll für UVM/LfU für die Verbreitung von Informationen über Netz ein einheitliches Abrechnungsverfahren

ren bzw. -system identifiziert, für den praktischen Einsatz vorbereitet und für die XfaWeb-Fachinformationssysteme eingesetzt werden. Hierzu sind folgende Arbeiten durchzuführen:

- Definition von Produkten und Dienstleistungen, die von UVM/LfU über Netz angeboten werden können; Identifikation der Nutzergruppen und ihrer Anforderungen; Anforderungen seitens Behörden und Verwaltung (FZI, FAW)
- Untersuchung und Bewertung der marktverfügbaren Abrechnungssysteme für E-Commerce; praktische Evaluierung ausgewählter Produkte (FZI, FAW)
- Auswahl eines geeigneten Produkts (FZK/IAI, FZI, FAW)
- Anpassung der XfaWeb-Systeme für die Integration mit dem Abrechnungssystem (FZK).

Die Ergebnisse sollen auf andere Vorhaben übertragbar sein.

8. Schlußbemerkung

Die dargestellten F+E-Arbeiten sind Vorschläge, die teilweise noch detailliert, verifiziert und priorisiert werden müssen. Weitere Partner können sowohl auf Auftraggeber- wie auch auf Entwicklerseite jetzt noch beitreten und ihre Anforderungen, Ressourcen und Lösungsvorschläge einbringen.

Es ist vorgesehen, das F+E-Vorhaben GLOBUS mit der Projektphase VI im November 1999 abzuschließen.

Roland Mayer Föll

Andreas Jaeschke