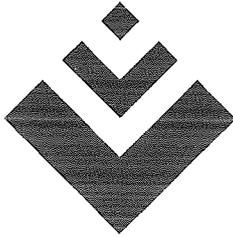


Forschungszentrum Karlsruhe
Technik und Umwelt

Wissenschaftliche Berichte
FZKA 5700



UIS

Baden-Württemberg

Projekt GLOBUS
Konzeption und prototypische
Realisierung einer aktiven
Auskunfts-komponente für
globale Umwelt-Sachdaten
im Umweltinformationssystem
Baden-Württemberg
Phase II 1995

R. Mayer-Föll, A. Jaeschke (Hrsg.)

Umweltministerium Baden-Württemberg

Forschungszentrum Karlsruhe
Institut für Angewandte Informatik

Dezember 1995

Forschungszentrum Karlsruhe

Technik und Umwelt

Wissenschaftliche Berichte

FZKA 5700

Projekt GLOBUS

**Konzeption und prototypische Realisierung einer aktiven
Auskunftskomponente für globale Umwelt-Sachdaten im
Umweltinformationssystem Baden-Württemberg**

Phase II 1995

R. Mayer-Föll, A. Jaeschke (Hrsg.)

Umweltministerium Baden-Württemberg

**Forschungszentrum Karlsruhe
Institut für Angewandte Informatik**

Forschungszentrum Karlsruhe GmbH, Karlsruhe

1995

**Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor**

**Forschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe**

ISSN 0947-8620

Projekt GLOBUS

**Konzeption und prototypische Realisierung
einer aktiven Auskunfts Komponente
für globale Umwelt-Sachdaten
im Umweltinformationssystem Baden-Württemberg**

Phase II - 1995

Projektträger:

**Umweltministerium Baden-Württemberg;
I. Henning, R. Mayer-Föll**

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

Projektpartner:

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

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

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

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

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

Vorwort

Das Umweltministerium Baden-Württemberg baut gemeinsam mit anderen Ministerien und der Stabsstelle Verwaltungsstruktur, Information und Kommunikation ein ressortübergreifendes Umweltinformationssystem (UIS) auf. Dieses UIS ist Teil des Landessystemkonzepts der Verwaltung Baden-Württemberg. Das UIS ist der Rahmen für die Bereitstellung von Umweltdaten und die Bearbeitung von fachbezogenen und fachübergreifenden Aufgaben im Umweltbereich. Die große Bedeutung der Informations- und Kommunikationstechnik (IuK) für eine wirkungsvolle Unterstützung von Umweltaufgaben wurde bereits 1984 erkannt. Art, Vielfalt, Menge und Verteilung der Daten erfordern den Einsatz von Informations- und Kommunikationstechnik (IuK) für den Umweltschutz. Große Teile der benötigten Software mußten bisher individuell entwickelt werden. Dies bedeutet mächtige proprietäre Systeme mit erheblichem Wartungs- und Pflegeaufwand. Der aktuelle Stand der Softwaretechnologie bietet modulare Werkzeuge, neue Standards und stellt die Verbindung zur Microsoft-Welt der Personalcomputer als Arbeitsumgebung her. Zusätzlich bringt der Einsatz von freiverfügbarer Software wirtschaftliche Vorteile.

Das Umweltministerium hat mit dem

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

eine längerfristige Zusammenarbeit zur Realisierung des UIS und seiner zahlreichen Komponenten vereinbart.

Mit dem Forschungsprojekt GLOBUS - Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umweltsachdaten im Umweltinformationssystem Baden-Württemberg - Phase II 1995 beauftragte es ein Konsortium dieser 5 Institute, wobei das FZI die Federführung und das FZK/IAI das Projektmanagement sowie die Erstellung des Abschlußberichts übernahmen.

Neue IuK-Standards und neue Softwaretechnologie bedürfen der Evaluierung und Überprüfung in bezug auf Einsatzmöglichkeiten, Nutzen und Wirtschaftlichkeit. In GLOBUS II wurden notwendige Grundlagen für die Fortschreibung der UIS-Rahmenkonzeption erarbeitet. Darin sollen Migrationspfade aufgezeigt und der Forderung, die richtige Information zur richtigen Zeit in der geeigneten Form schnell und leicht zur Verfügung zu stellen, entsprochen werden.

Wir möchten an dieser Stelle den am Projekt beteiligten Instituten für Ihre gute und engagierte Arbeit herzlich danken und hoffen, daß die Ergebnisse von GLOBUS II zur Unterstützung der Umweltverwaltung und zu einem verbesserten Informationsservice für den Bürger beitragen.

Inge Henning, Roland Mayer-Föll, Umweltministerium Baden-Württemberg
Manfred Müller, Ernst Schmid, Horst Spandl, Landesanstalt für Umweltschutz

Inhalt

Vorwort

Inhaltsverzeichnis

Überblick zum Projekt GLOBUS.....	1
Dienste im UIS	15
Architektur des WWW- und CORBA-basierten UIS	39
CORBA - Evaluierung für das UIS Baden-Württemberg.....	47
Management verteilter Umweltinformationsquellen	93
Nutzung von Diensten im WWW mit dem Common Client Interface (CCI)	193
Visualisierung von Umweltdaten.....	239
Entwicklung eines WWW-basierten Altlasten-Informationssystems	271
Ausblick - Fortführung des Projekts GLOBUS.....	299
Anhang	303
Anlage zu: Architektur des WWW- und CORBA-basierten UIS	305
Anlage zu: Nutzung von Diensten im WWW mit dem Common Client Interface (CCI)	317
Anlage zu: Visualisierung von Umweltdaten.....	339
Anlage: Richtlinie zur Erstellung WWW-verfügbarer Berichte.....	361

Überblick zum Projekt GLOBUS

1. Einleitung

Ziel des Umweltinformationssystems des Landes Baden-Württemberg (UIS) ist die Bereitstellung von Umweltdaten und die Bearbeitung von fachbezogenen und fachübergreifenden Aufgaben im Umweltbereich. Es basiert auf einem aufgabenorientierten, informationstechnischen und organisatorischen Rahmenkonzept. Die Aufgaben des UIS sind:

- Planung und Verwaltungsvollzug - Einsatz der Informations- und Kommunikationstechnik (IuK) zur effizienten Erledigung der Verwaltungsaufgaben mit Umweltbezug;
- Umweltbeobachtung - Erhebung, Analyse und Prognose der punktuellen und landesweiten Umweltsituation;
- Integration und Investitionsschutz - Koordination und Integration der vorhandenen Verfahren zur Informationsverarbeitung im Umweltbereich;
- Notfallmanagement - Unterstützung der Bewältigung von Not-, Stör- und Vorsorgefällen, insbesondere durch schnellere und umfassendere Nachrichtenübermittlung und -verarbeitung;
- Information - Information der politischen und administrativen Führung in Landtag, Regierung und Verwaltung sowie der Öffentlichkeit und Schaffung des freien Zugangs des Bürgers zu Informationen über die Umwelt.

Das UIS gilt als eine der ersten umfassenden Ansätze in Deutschland. Es wurde seit dem Jahre 1984 als Teil des Landessystemkonzeptes federführend vom Ministerium für Ernährung, Landwirtschaft, Umwelt und Forsten [EM] und ab 1987 bis 1990 vom neu gebildeten Umweltministerium für die Landesverwaltung Baden-Württemberg gemeinsam mit McKinsey und anderen Ministerien konzipiert und wird seitdem umgesetzt [UM/Mck].

Mit dem übergeordneten Landessystemkonzept (LSK) Baden-Württemberg werden ökonomische und strategische Zielsetzungen verfolgt. Bezüglich der Wirtschaftlichkeit sollen das Landessystemkonzept und seine Teile, wie das Umweltinformationssystem Baden-Württemberg, zu einer Leistungsverbesserung und zur Rationalisierung der betreffenden Ressorts führen. Strategische Zielsetzungen des LSK waren 1984 zum einen die Bereitstellung von Informationssystemen als Führungsinstrument der entsprechenden Ressorts, zum anderen eine gezielte Technologieförderung der Wirtschaft [AG LSK].

Der technische Fortschritt im IuK-Bereich bringt ständig verbesserte Konzepte und Werkzeuge, die effizientere und wirtschaftlichere Problemlösungen ermöglichen. Dies macht eine kontinuierliche Fortentwicklung des systemtechnischen Rahmenkonzepts des UIS erforderlich, an der sich Neuentwicklungen im UIS und die Weiterentwicklung existierender UIS-Komponenten orientieren müssen.

In diesem Aufgabenumfeld werden im Auftrag des Umweltministeriums Baden-Württemberg und der Landesanstalt für Umweltschutz (LfU) von verschiedenen Forschungsinstituten in Kooperation folgende Forschungs- und Entwicklungs- (F+E-)Projekte durchgeführt:

- Projekt **INTEGRAL** - Integration von heterogenen Komponenten des Umweltinformationssystems (UIS) Baden-Württemberg durch das Forschungsinstitut für anwendungsorientierte Wissensverarbeitung an der Universität Ulm (FAW) und das Institut für Kernenergetik und Energiesystem der Universität Stuttgart (IKE) [INTEGR],
- Projekt **WWW-UIS** - Anwendungsintegration der Grenz-/Richtwert-Datenbank durch das Forschungszentrum Informatik an der Universität Karlsruhe (FZI) und das Institut für Photogrammetrie und Fernerkundung der Universität Karlsruhe (IPF) [WWWUIS],
- Projekt **UIS-RK-Fortschreibung** - Fortschreibung der UIS-Rahmenkonzeption durch das FAW [UISRK] und
- Projekt **GLOBUS** - Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten durch FZI, FAW, IKE, IPF sowie das Institut für Angewandte Informatik des Forschungszentrums Karlsruhe (FZK/IAI) [GLOBUS].

Das Gesamtziel des Projekts **GLOBUS** ist vorrangig für Referenten und Sachbearbeiter im Umweltministerium und in der LfU eine aktive, benutzerfreundliche Auskunftskomponente zu entwickeln, die unterschiedlichste Informationsbestände auch über den Rahmen des UIS hinaus erschließt. Damit soll eine wesentliche Effizienzverbesserung bei Referenten und Sachbearbeitern erreicht werden.

In der ersten Phase des Projekts (GLOBUS I - 1994) wurde mit Erfolg ein Prototyp für die Erschließung der Umweltdatenberichte der LfU und des Umweltbundesamts erstellt.

Ziel der zweiten Phase des Projekts (GLOBUS II - 1995) war es, eine übergreifende Rahmenarchitektur für die Auskunftskomponente zu konzipieren, die prototypische Arbeiten aus GLOBUS I weiterzuführen und sie an beispielhaften Anwendungen zum Einsatz zu bringen.

Im vorliegenden Bericht sind die Aufgaben, Arbeiten und Ergebnisse von GLOBUS II zusammengestellt.

Im Kap. 2 ist das aktuelle Umfeld und die Ausrichtung der Fortschreibung der UIS-Rahmenkonzeption skizziert, wie diese - ausgehend von den Ergebnissen des Workshops vom 15.5.95 im Umweltministerium [UISW95] - in den Sitzungen des GLOBUS-Lenkungsausschusses und in den zugehörigen technischen Gesprächen diskutiert wurden. Darauf basierend sind in Kap. 3 die Aufgabenstellungen für die Teilprojekte der Projektpartner FZI, FAW, IKE, IPF und FZK dargestellt.

Es folgen die Ergebnisberichte:

- Dienste im UIS
(Zwischenergebnis des GLOBUS-Arbeitskreises 'UIS-Dienstdefinition')
- Architektur des WWW- und CORBA-basierten UIS
- CORBA - Evaluierung für das UIS Baden-Württemberg
(Ergebnisberichte des FZI)
- Management verteilter Umweltinformationsquellen
(Ergebnisbericht des FAW)
- Nutzung von Diensten im WWW mit dem Common Client Interface (CCI)
(Ergebnisbericht des IKE)
- Visualisierung von Umweltdaten
(Ergebnisbericht des IPF)
- Entwicklung eines WWW-basierten Altlasten-Informationssystems
(Ergebnisbericht des FZK/IAI)

Im Schlußkapitel folgt ein Ausblick mit Empfehlungen für die Fortführung der Arbeiten im Rahmen des Projekts GLOBUS.

Bemerkung:

Die Ergebnisberichte der Forschungsinstitute sind inhaltlich als eigenständige Berichte strukturiert und in dieser Form auch in den Gesamtbericht GLOBUS II eingefügt. Dies kann in den Texten zu gelegentlichen Wiederholungen führen.

2. Umfeld der GLOBUS-Entwicklungen

2.1 Neue Randbedingungen aufgrund technischer Weiterentwicklung

Die technische Fortentwicklung der letzten Jahre hat im IuK-Bereich zu Konzepten und Werkzeugen geführt, die deutliche Auswirkungen auf die Fortschreibung des UIS-Rahmenkonzepts haben.

Eine wesentliche Rolle spielen hier die hard- und softwaremäßigen Neuentwicklungen, die neue Möglichkeiten erschließen aber auch neue Anforderungen an die künftige Architektur des Systems stellen. Die Entwicklung erfordert hier die Ausrichtung auf Client/Server-Konzepte. Clientseitig bekommen PCs eine dominierende Rolle, serverseitig Workstations in applikationsspezifischem Ausbau.

Speziell für das Anforderungsprofil des UIS werden die Aspekte offene und verteilte Systeme relevant. Hier eröffnen CORBA-Architekturen neue Möglichkeiten für die Verknüpfung in komplexen Dienststrukturen.

Daneben setzen sich neue Standards durch:

Bei Betriebssystemen dominiert in der PC-Welt Windows; bei höheren Leistungsanforderungen (serverseitig) UNIX. Im Datenbanksektor sind inzwischen weitgehend relationale DBMS

Standard (ORACLE...). Objektorientierte Verfahren, Konzepte und Tools gewinnen zunehmend an Bedeutung.

Die Kosten für Softwareerstellung und -pflege zwingen dazu, proprietäre Lösungen, wo nur möglich, zu vermeiden und kommerziell verfügbare Standard-Software einzusetzen. In der PC-Welt sind einige Microsoft-Produkte zum Quasi-Standard geworden.

2.2 Neue Randbedingungen aufgrund verändertem Nutzerprofil

Gegenüber den Gegebenheiten, auf die das ursprüngliche Rahmenkonzept des UIS aufsetzte, sind einige aktuelle Entwicklungen und Trends zu berücksichtigen. Der Nutzerkreis des UIS hat sich erweitert und ist in diesem Zusammenhang deutlich heterogener geworden. Durch organisatorische Maßnahmen in der Umweltverwaltung sind die Kommunen stärker als bisher eingebunden. Die Ausgliederung einiger Bereiche in privatwirtschaftlich arbeitenden Einrichtungen schafft neue Schnittstellen. Weitere Anforderungen ergeben sich durch das gesetzlich verbrieftete Recht der Öffentlichkeit auf freien Zugang zu Umweltinformationen.

Steigende Bedeutung erhalten Anforderungen speziell bezüglich Standardisierung und Harmonisierung von Daten, Metadaten, Schnittstellen etc., die aus einer Intensivierung der Kooperation auf Länder-, Bundes- und EU-Ebene resultieren.

2.3 Weiterentwicklung der UIS-Rahmenkonzeption

Die systemtechnische Fortschreibung der UIS-Rahmenkonzeption unter Beachtung der oben aufgelisteten neuen Randbedingungen und Trends wurden u. a. im UIS-Workshop vom 15.5.95 [UISW95] diskutiert und werden in dem Vorhaben UIS-RK-Fortschreibung des FAW detailliert ausgearbeitet. Berücksichtigt werden auch die Vorgaben des LSK. Im folgenden sind einige Grundzüge des (sich abzeichnenden) Lösungskonzepts skizziert, soweit sie für die Aufgabenstellung des GLOBUS II-Projekts relevant sind.

Die systemtechnische Weiterentwicklung des UIS ist ausgerichtet auf eine konsequente Migration des Systems in Richtung auf ein offenes, verteiltes Client-Server-System mit einer Dienste-orientierten Struktur (siehe hierzu 'Architektur des WWW-basierten UIS', Seite 39ff und hier speziell Abb. 2).

In diesem Konzept werden System- und Anwenderdienste unterschieden:

Dienste sind in sich abgeschlossene Programmbausteine, die eine definierte eindeutige Funktion durchführen und über eine Schnittstelle angesprochen werden. Aus Sicht der Funktionalität sind Systemdienste daten- bzw. systemnahe Komponenten mit beschränkter, aber systemweit vielfach benötigter Funktionalität.

Anwenderdienste implementieren komplexere Funktionen, wobei die angebotene Funktionalität einer beschränkten, in der Anwendung sinnvoll abgrenzbaren Aufgabe entspricht (siehe hierzu 'Dienste im UIS', Seite 15 ff).

Zur Kommunikation zwischen den Diensten in einer offenen, verteilten Umgebung werden diese in eine Architektur entsprechend den Standards von Common Object Broker Architecture CORBA eingehängt. Diese Standards und die zugehörigen Implementierungen bieten eine geeignete Architektur für die Verteilung und Kommunikation objektorientierter Softwarebausteine in heterogenen vernetzten Systemen unabhängig davon, in welcher Programmiersprache sie realisiert und auf welcher Plattform sie installiert sind.

Für die Kommunikation zwischen Anwender und Anwenderdiensten wird World-Wide-Web (WWW) ein Hypertext-System auf Basis des Internet eingesetzt. Über WWW kann jeder Rechner am Internet als Server eigene Informationen anbieten und/oder als Client auf die angebotenen Informationen zugreifen.

Ein Zugangssystem bietet client- und serverseitige Funktionen zur Unterstützung der Benutzer bei der Informationssuche (Dienste, Dokumente, Berichte ...), bei der Navigation im System und beim Zugriff auf die Informationen. Es beinhaltet dafür unterschiedliche Suchverfahren (über den Umweltdatenkatalog (UDK), über Verschlagwortung, Volltextsuche...).

Für die schrittweise Umsetzung des umrissenen Konzepts ergaben sich in den bisherigen Diskussionen als wesentliche Randbedingungen:

- Weiterentwicklung des UIS stets unter der Gesamtsicht: Erhalt, Pflege und Qualitätsverbesserung der im UIS bereitgestellten Informationsgehalte (Daten, Funktionen, Wissen),
- Enge Einbindung des Anwenders und Orientierung am Bedarf der Anwender und an den Nutzeranforderungen sowohl bei Neuentwicklungen im Rahmen des UIS als auch bei der Migration des laufenden Systems in Richtung auf das Zielkonzept,
- Vermeidung proprietärer Softwarelösungen; möglichst breiter Einsatz von kommerziell verfügbarer oder (wenn Qualität und Pflege gesichert ist) von Public Domain Software,
- derzeit noch nicht ausschließliche Festlegung auf aktuelle Trends wie WWW und CORBA,
- Einhaltung der Standards des LSK,
- Berücksichtigung der LSK-Empfehlungen für offene Systemarchitekturen,
- Nutzung von Synergieeffekten durch abgestimmte oder gemeinsame Entwicklungen mit Ländern, Bund, Europäischer Union, kommunalem Bereich, Wirtschaft und Wissenschaft.

2.4 Forschungs- und Entwicklungsaufgaben

In der schrittweisen Umsetzung dieser Konzeption für die Fortentwicklung des UIS sind im Vorfeld bis hin zur praktischen Realisierung eine Reihe von systemtechnischen F+E-Arbeiten durchzuführen.

Diese umfassen:

- Exploratorische Projekte; das sind Untersuchungen innovativer Konzepte, Verfahren und Tools auf der Basis von Literaturstudien und Testinstallationen. Ergebnis ist hier in der Regel ein Gutachten, das die Eignung des Testobjekts für eine Problemlösung im UIS qualifiziert.
- Prototypische Implementierungen; dies umfaßt die Erstellung einer Systemkomponente mit ausgewählter Funktionalität in geeigneter Testumgebung zur Prüfung der Machbarkeit und Tauglichkeit einer Problemlösung. Ergebnis ist ein Demonstrationssystem inkl. einer Bewertung.
- Realisierungsprojekte; dies beinhaltet die Entwicklungen eines Systems für den praktischen Einsatz. Ergebnis ist das System mit Dokumentation und Handbuch.

Inhaltlich liegen die Schwerpunkte der anstehenden F+E-Projekte im Rahmen der systemtechnischen Fortentwicklung des UIS in folgenden Bereichen:

Im Bereich **Systemaufbau / Server**

- Konzeption einer offenen, verteilten Server-Architektur auf der Basis von CORBA oder ähnlicher aktueller Konzepte. Dies beinhaltet die Evaluierung entsprechender Implementierungen und Tools und die Auswahl eines unter UIS-Aspekten optimalen Produkts, sowie weiter
- Installation der ausgewählten CORBA-Implementierung auf UIS-Systemplattformen und prototypische Integration von ausgewählten Applikationen (alte/neue),
- Einbindung externer Dienste, das sind hier Bausteine außerhalb des UIS oder UIS-Bausteine, die in ihrer derzeitigen monolithischen Struktur eingebunden werden müssen und letztlich
- (schrittweise) Überführung der Resultate in Produktionssysteme.

Im Bereich **Dienststrukturen**

- Definition der Begriffe System- und Anwenderdienste mit Vorgaben für deren sinnvolle Abgrenzung im UIS-Rahmen unter den Aspekten Anwenderfunktionalität, Mehrfachverwendbarkeit, Reintegrierbarkeit etc., dann
- Aufgliederung der monolithischen Komponenten des aktuellen UIS in einzelne Dienste und Definition einer Dienststruktur unter Architektur- und Anwenderaspekten (Reintegrierbarkeit entsprechend den Anforderungen der Vorgangsbearbeitung am Arbeitsplatz, Zuständigkeit der Fachleute bzgl. Entwicklung und Pflege, hardwaremäßige Verteilung im Netz etc.)
und entsprechend
- Definition von UIS Diensten in neuen UIS Applikationen unter den gleichen eben genannten Aspekten
sowie
- Integration in die neue Architektur zu prototypischen und anwendungsreifen Komponenten.

Im Bereich **Metainformationen für das Informations-Retrieval**

- Konzeption eines integrierten Zugangssystems zu den Informationen des UIS (Daten, Dienste, multimediale Dokumente etc.), das die Anforderungen lokaler als auch globaler Suche erfüllt und den Anforderungen von gelegentlichen als auch von Routine-Nutzern gerecht wird. Das System muß umfassen:
 - Zugriffsstrategien über Metadatenkataloge - hier Verfahren auf der Basis des UDK und/oder auf der Basis spezieller Informationsquellenkataloge
 - thesaurusgestützte Verfahren,
 - Verschlagwortungstechniken,
 - Volltextsuche,
 - erweiterte Indexierungstechniken (räumliche Indexierung/Georeferencing) etc.

Im Bereich **benutzergerechte Arbeitsplatzumgebung**

- Aufbau eines flexiblen UIS-PC-Arbeitsplatzes, an dem entsprechend den individuellen Anforderungen eine aufgabenspezifische, softwaremäßige Arbeitsplatzumgebung generiert werden kann. Dies umfaßt:
 - die Entwicklung flexibler Sessionkonzepte,
 - die Bereitstellung von Schnittstellen für den Zugriff aus Anwender-Applikationen auf Informationen und Funktionalitäten aus dem UIS,
 - die Integration von Tools für Visualisierung (geographische Viewer), Text- und Tabellenverarbeitung, Grafik, Desktop-Publishing / -Mapping etc.,
 - die Bereitstellung der Schnittstellen für diese Tools zu entfernten Informationsquellen,
 - u.ä.

Weitere F+E-Arbeiten stehen in folgenden Bereichen an:

- Maßnahmen zu Datensicherheit und Datenschutz - Unter beiden Aspekten ist ein UIS-durchgängiges Modell zu entwickeln, das die Einhaltung der Sicherheitsstandards garantiert.
- Integration aktiver Mechanismen im UIS - Damit wird die automatische Reaktion des Systems auf vom Nutzer vordefinierte Ereignisse möglich, was zu hoher Zuverlässigkeit in der Reaktion und zur Entlastung des Nutzers beitragen kann.
- Entwicklung von Konzepten zur Erhöhung der Datenqualität - Dies umfaßt z. B. die Nutzung semantischer Metainformationen, die den Benutzer bei der Interpretation der gesuchten Informationen unterstützen,
- Techniken für das kooperative Arbeiten - Dies beinhaltet die Bereitstellung von Funktionen im UIS zur Unterstützung bei der arbeitsplatzübergreifenden Erledigung von Aufgaben, die die Kooperation mehrerer Spezialisten erfordern.

3. Aufgabenstellungen im Rahmen von GLOBUS II

3.1 Teilprojekt 1: Konzept und Komponentenevaluierung Ziele und Aufgaben des FZI

Ziele

Die Arbeiten des FZI konzentrieren sich im Projekt GLOBUS II auf Untersuchungen von Konzepten offener verteilter, dienstorientierter Architekturen im Hinblick auf ihre Eignung für den serverseitigen Einsatz im UIS und auf die Evaluierung von Basissoftware, die diese Architekturen unterstützen.

Das Hypertext Transfer Protokoll (HTTP) sowie die WWW-Viewer ermöglichen den Zugriff auf dynamische HTML-Dokumente, die von einem Server zu vollständigen Dokumenten aufgearbeitet und im Netz verfügbar gemacht werden. Bei der Generierung dynamischer HTML-Seite können dabei Zugriffe auf unterschiedliche Programme und Datenquellen erforderlich sein, die wiederum in unterschiedlicher Realisierung auf verschiedene Rechner verteilt sind. Zur Lösung derartiger Probleme wurde durch die Object Management Group (OMG) CORBA definiert und standardisiert. Auf der Basis von CORBA soll für den Server-Bereich ein durchgängiges Architekturkonzept für das UIS entwickelt werden.

Aufgaben in GLOBUS II

Im Rahmen dieser Zielsetzung sollten in GLOBUS II die derzeit auf dem Markt verfügbaren CORBA-Implementierungen bezüglich Funktionsumfang, unterstützte Plattformen, Einhaltung der Standards etc. evaluiert werden.

In einer Produktvorauswahl aufgrund allgemeiner und UIS-spezifischer Kriterien auf der Basis der vom Hersteller oder Distributor gelieferten Produktinformationen sollte dabei in einem ersten Schritt der Evaluierungsrahmen auf 3 Implementierungen eingeschränkt werden. UIS-spezifische Kriterien sind hierbei die Kompatibilität mit den Betriebssystemumgebungen DEC-Unix, OpenVMS, MS-Windows, Windows NT und den Netzprotokollen TCP/IP und DECNet.

Die drei ausgewählten Produkte sollten in geeigneter Testumgebung installiert und anhand UIS-anwendungsnaher Dienste ausgetestet werden.

3.2 Teilprojekt 2: Thesaurusgestütztes Retrieval von Umweltinformationen

Ziele und Aufgaben des FAW

Ziele

Die F+E-Arbeiten des FAW im Projekt GLOBUS II sind ausgerichtet auf den Problembereich Zugang zu Informationen (Navigation, Retrieval...) im UIS.

Die Suche nach Informationen in komplexen, vernetzt/verteilten Systemen, die Dokumente, Daten und Dienste anbieten, gestalten sich für die Praxis oft schwierig (hoher Aufwand, unadäquate/unvollständige Ergebnisse). Es ist daher erforderlich, Thesauri und andere an den Anforderungen der Anwendung orientierte Indexmechanismen für eine effiziente Suche zu integrieren. Im ersten Teil des Projekts GLOBUS wurde prototypisch eine thesaurusgestützte Retrievaltechnik auf Basis des Thesaurus der UBA am Beispiel des Berichts „Umweltdaten“ der LfU demonstriert. Die aktuelle Version 2.0 des Umweltdatenkatalogs (UDK) umfaßt einen Thesaurusmodul, in dem der UBA-Thesaurus in einer relationalen Datenbank zur Verfügung steht.

Aufgaben in GLOBUS II

Unter Berücksichtigung der weiteren Datenmodellierung des UDK ist im Rahmen des Projekts GLOBUS II ein Konzept mit prototypischer Realisierung zur Nutzung des UDK-Thesaurus zum Zugriff auf entsprechend verschlagwortete Dokumente (Umweltberichte) zu erstellen.

In diesem Rahmen waren die folgenden Anforderungen zu erfüllen:

- Thesaurus-unterstützter Zugriff auf Umweltinformation, die von unterschiedlichen Servern zur Verfügung gestellt werden,
- Ausweitung des Thesaurus-unterstützten Zugriffs auf allgemeine Dokumente, Daten und Dienste,
- Entwicklung und Integration weiterer Indizierungstechniken, z. B. unter den Aspekten Raumbezug, Zeitbezug, etc.,
- Zugriff mit logischer Verknüpfung von Diskriptoren.

3.3 Teilprojekt 3: Einführung externer Dienste Ziele und Aufgaben des IKE

Ziele

Die Arbeiten in diesem Teilprojekt untersuchen die Möglichkeit, clientseitig Anwender-Applikationen und serverseitig externe Dienste ins WWW einzubinden. Der Zugriff von Anwender-Applikationen auf in WWW verfügbare Informationen und Dienste wird von den verfügbaren WWW-Browsern nicht unterstützt. Lösungsansätze werden hier ermöglicht auf der Basis

- des Common Gateway Interface CGI, das es einem HTTP-Server gestattet, über Gateway-Programme auf Informationen zuzugreifen und
- des für Mosaic verfügbaren Common Client Interface CCI, das es einer Applikation ermöglicht, das Funktionsangebot von Mosaic zu nutzen.

Aufgaben in GLOBUS II

- Es sollte untersucht werden, wie auf der Basis der CCI-Schnittstelle clientseitig/ lokale (Microsoft-) Tools mit Mosaic gekoppelt werden können. Dabei sollen vor allem auch Untersuchungen zur Sicherheit der Kommunikation durchgeführt werden und Aussagen über notwendige Erweiterungen im Hinblick auf eine spätere Einführung externer Dienste erfolgen.
- Weiter sollte geprüft werden, ob und wie eine Verbindung des im Projekt TZUI entwickelten Kommunikationsinterpreters (KIP) und WWW möglich ist. Eine solche Verbindung sollte beispielhaft am Zugriff auf MEROS-Daten der LfU erprobt und demonstriert werden.

3.4 Teilprojekt 4: Visualisierung von Umweltdaten Ziele und Aufgaben des IPF

Ziele

Die Aufgabenstellung des IPF im Projekt GLOBUS II umfaßt F+E-Arbeiten zum clientseitigen Einsatz von Werkzeugen zur Präsentation von Umweltdaten. Für den Aufbau von optimalen Arbeitsumgebungen auf dem Client, die eine anwendungsspezifische Bearbeitung der vom Server übernommenen Informationen entsprechend den Anforderungen des Nutzers ermöglichen, sind moderne Werkzeuge der Bürokommunikation, Tabellenkalkulation und Textverarbeitung zu integrieren. Aufgrund des häufigen Raumbezugs von Umweltinformationen ist in

der Arbeitsumgebung eine breite Funktionalität zur Verknüpfung von Fachdaten und Karten und zur Visualisierung der Informationen durch entsprechende Geodaten-Viewer gefordert.

Aufgaben in GLOBUS II

Ausgerichtet auf diese Anforderungen sollten von IPF in 1995 folgende Aktivitäten durchgeführt werden:

- Evaluierung von marktverfügbaren (kommerziell und Public Domain) Softwaretools zur Generierung von Geschäftsgrafiken und Darstellung von Meßreihen. Die Systeme waren unter den Aspekten Funktionalität, Leistung, Kosten und bezüglich ihrer Integrierbarkeit ins WWW zu bewerten.
- Erstellung einer Schnittstelle zum Räumlichen Informations- und Planungssystems RIPS der LfU. Das Konzept sieht einen Zwischenspeicher (Cache) für gerasterte Karten vor, in dem häufig benutzte Karten in der Arbeitsumgebung vorgehalten werden.
- Realisierung einer einfachen clientseitigen Auskunftsfunktion für den Zugriff auf RIPS-Daten, basierend auf dem RIPS-Auskunftssystem.

Weiter sollten die Möglichkeiten einer clientseitigen Nutzung von Arc-View untersucht werden. Das Konzept sieht hier eine Datenextraktion aus RIPS über Smallworld, die serverseitige Zwischenablage in ARC-Generate-Format und der Transfer mittels HTTP vor.

3.5 Teilprojekt 5: Fachanwendung Altlasten Ziele und Aufgaben des FZK/IAI

Ziele

Ziel der Arbeiten des FZK/IAI ist die Entwicklung eines Altlasten-Informationssystems im Rahmen des UIS. In der LfU und der Wasserwirtschaftsverwaltung werden ständig Berichte (Dokumente) und rechnergestützte Werkzeuge (Fachsysteme) für die Altlasten-Bearbeitung erstellt. In einem Altlasten-Informationssystem soll für den Sachbearbeiter in Behörden und außerhalb ein einfacher Zugang zu diesen Berichten und Werkzeugen und zu weiteren externen Altlasten-relevanten Informationen und Tools geschaffen werden. Das Informationssystem soll dem Anwender über WWW zur Verfügung stehen und auch in einer CD-ROM-Version zur lokalen Installation angeboten werden.

Schwerpunkt der Arbeiten ist die Analyse der Anforderungen an den Server und Browser und an die im Informationssystem benutzten Konvertierungs- und Integrationswerkzeuge sowie die Verbesserung/Erweiterung bestehender bzw. die Implementierung fehlender Werkzeuge.

Aufgaben in GLOBUS II

Im Rahmen dieser Zielstellung sollten in 1995 im Bereich Altlastenberatung der LfU beispielhaft drei Berichte ausgewählt, konvertiert und unter einer geeigneten Zugriffsstruktur in das WWW eingebunden werden.

Hierzu waren 1995 im einzelnen folgende Arbeiten durchzuführen:

- Analyse der Anforderungen an die Architektur des Altlasten-Informationssystems (WWW und CD-ROM) und die benötigten Werkzeuge,
- Untersuchung der in Frage kommenden Berichte im Altlastenbereich im Hinblick auf ihre Umsetzbarkeit ins WWW,
- Umwandlung der Dokumente in WWW-Dokumente,
- Entwurf und Implementierung verschiedener Zugriffspfade,
- Unterstützung der LfU bei der Entwicklung neuer Altlasten-Berichte im Hinblick auf deren problemlose Einbindung ins WWW; Erstellen einer entsprechenden Richtlinie.

Die in den Arbeiten entwickelten Konzepte bauen auf die Ergebnisse von GLOBUS I auf. Die bei den Partnern in GLOBUS I und verwandten UIS-Projekten entwickelten und verfügbaren Werkzeuge werden übernommen.

Literatur

- [EM] 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
- [UM/Mck] 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
- [AG LSK] 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
- [INTEGR] K. Blank, W.-F. Riekert, F. Schmidt, M. Tischendorf;
Projekt Integral: Integration heterogener Komponenten
des Umweltinformationssystems (UIS) Baden-Württemberg.
Abschlußbericht der Phase I.
FAW Ulm und IKE Universität Stuttgart, 1994
- K. Blank, W.-F. Riekert, Th. Kirst, R. Münster, F. Schmidt, M. Tischendorf, K.
Kübler;
Projekt Integral: Integration heterogener Komponenten
des Umweltinformationssystems (UIS) Baden-Württemberg.
Abschlußbericht der Phase II.
FAW Ulm und IKE Universität Stuttgart, 1994
- FAW Ulm, IKE Universität Stuttgart
Projekt Integral: Integration heterogener Komponenten
des Umweltinformationssystems (UIS) Baden-Württemberg.
Abschlußbericht der Phase III.
in Vorbereitung

- [WWWUIS] R. Kramer, R. Nikolai;
WWW-UIS Anwenderhandbuch.
Forschungszentrum Informatik (FZI), Karlsruhe, 1995
- R. Kramer, R. Nikolai;
WWW-UIS Administrationshandbuch.
Forschungszentrum Informatik (FZI), Karlsruhe, 1995
- R. Kramer, R. Nikolai;
WWW-UIS: Auskunfts-, Administrations- und Verwaltungsdienste.
Forschungszentrum Informatik (FZI), Karlsruhe, 1995
- [UISRK] Umweltministerium Baden-Württemberg, FAW Ulm;
Umweltinformationssystem Baden-Württemberg -
Fortschreibung der UIS Rahmenkonzeption. Abschlußbericht der Phase I.
in Vorbereitung
- [GLOBUS] J. Wiesel, F. Schmidt, W.-F. Riekert, R. Kramer;
GLOBUS - Konzeption und prototypische Realisierung
einer aktiven Auskunfts-komponente für globale Umwelt-Sachdaten.
Abschlußbericht Phase I.
Umweltministerium Baden-Württemberg, 1994
- [UISW95] Umweltinformationssystem Baden-Württemberg (UIS).
Dokumentation des UIS-Workshops am 15.5.1995
im Umweltministerium Baden-Württemberg.
erstellt durch: G. Hess, A. Schultze; FAW Ulm, 1995

Dienste im UIS

*F. Schmidt; M. Tischendorf
Universität Stuttgart,
Institut für Kernenergetik und Energiesysteme (IKE)
Pfaffenwaldring 31
D-70569 Stuttgart*

1 EINLEITUNG	17
2 DEFINITION DES BEGRIFFES DIENST	17
3 EIGENSCHAFTEN VON DIENSTEN	19
3.1 STRUKTUR VON DIENSTEN IN HETEROGENEN UMGEBUNGEN	19
3.2 AUFBAU EINES DIENSTES.....	21
3.2.1 <i>Beschreibung der Funktionalitäten</i>	21
3.2.2 <i>Beschreibung der Schnittstellen</i>	22
3.2.3 <i>Beschreibung von Zusatzattributen</i>	22
3.3 KLASSIFIZIERUNG VON DIENSTEN - DIENSTETYPEN	22
3.4 VORSCHLAG FÜR SCHNITTSTELLEN IN VERBINDUNG MIT DIENSTEN	23
3.5 WIE FINDET MAN DIENSTE.....	26
3.6 DIENSTE UNTER WWW	27
4 DIENSTE IM RAHMEN DES UIS.....	27
5 INTEGRATION VON DIENSTEN IN UIS-ANWENDUNGEN	31
5.1 DIENSTE IN UIS-ANWENDUNGEN	31
5.2 EXTERNE DIENSTE.....	31
5.3 INTEGRATION VON DIENSTEN IN ANWENDUNGSSYSTEMEN	32
5.4 NUTZUNG EXTERNER DIENSTE.....	32
GLOSSAR.....	34
LITERATUR.....	38

1 Einleitung

Informationen sind geordnete Daten, die im Kontext vorhandenen Wissens interpretiert werden können. Die Interpretation erfolgt über Methoden und Regeln für die Anwendung dieser Methoden auf Daten. Beauftragt man mit der Bereitstellung interpretierbarer Daten einen Dritten (Diensteanbieter), so nehmen wir seinen Dienst in Anspruch.

Die Inanspruchnahme von Dienstleistungen ist eine wichtige Voraussetzung einer arbeitsteiligen Gesellschaft. Die Nutzung von Diensten ist entsprechend zentrales Element verteilter, kooperativer Anwendungssysteme, wie sie im Rahmen der UIS-Konzeption entwickelt werden sollen.

Im folgenden wird versucht, einen Vorschlag für eine Definition des Begriffes Dienst für die UIS-Entwickler zu machen und daraus einige Konsequenzen für Diensteanbieter abzuleiten. Dabei werden Ergebnisse wiedergegeben, die im Rahmen der UIS-Projekte INSERV /5/, INTEGRAL /7/ und WWW-UIS /8/ entwickelt wurden. Die Grundideen des Dienstekonzeptes sind wesentlicher Bestandteil der Promotionsarbeit von Herrn Tischendorf und in diesem Kontext von ihm im Umfeld IKE entwickelt worden.

2 Definition des Begriffes Dienst

Ein Dienst (englisch: service) ist eine abgeschlossene (Programm-)Einheit, die eine spezielle Aufgabe erfüllt. Ein Dienst kann eine Menge an Funktionen besitzen, die sich gegenseitig nicht beeinflussen. Ein Dienst wird von einem Diensteanbieter erbracht und von einem Dienstenutzer verwendet. Ein Dienst kann Methoden von Objekten aufrufen und an einer Rechnerschnittstelle anbieten.

Diese Dienstedefinition ist weitreichender als die meistens in der Literatur verfügbare Definition, nach welcher ein Dienst eine Funktion ist, die von einem Objekt an einer Schnittstelle angeboten wird /4/. Sie ist andererseits sehr allgemein und umfaßt sowohl Aufgaben, die durch einfache Operationen (siehe Glossar) als auch Aufgaben, die durch komplexe Programmsysteme erfüllt werden. Von daher bietet es sich an, die Dienste nach ihrer Komplexität zu klassifizieren. Andere Klassifikationen sind möglich. Eine formale Klassifikation aus Sicht der auszulösenden Aktionen zur Verwaltung von Diensteaufrufen und Diensteergebnissen findet sich in /8/. Eine funktionale Klassifizierung wird im Abschn. 3.4 eingeführt. In diesem Abschnitt soll die Klassifizierung nach Gesichtspunkten der Komplexität erfolgen.

In diesem Kontext unterscheiden wir drei Klassen oder Typen von Diensten:

- Systemdienste implementieren Basisfunktionalitäten, wie beispielsweise den Zugriff auf Datenbanksysteme, die Aufbereitung von HTML-Seiten oder die Objekte, die im Projekt TZUI zur Erledigung der Kommunikation entwickelt werden. Systemdienste können als Methoden von Objekten oder Dienste angeboten werden. Sie benötigen in der Regel keinen Benutzerdialog und haben typischerweise eine eindeutige Funktion, die über eine Schnittstelle mit eindeutiger Signatur (siehe Glossar) angesprochen werden.
- Anwenderdienste repräsentieren ein System gegenüber den Endanwendern. Anwenderdienste können in ihrer einfachsten Form als eine statische Abfolge von Systemdiensten realisiert werden. Ein einfaches Beispiel (aus /8/) für einen Anwenderdienst ist der Aufruf eines Systemdienstes Datenzugriff sowie daran anschließend der Aufruf eines Systemdienstes HTML-Seite, der das Ergebnis der Inanspruchnahme des Datendienstes als HTML-Seite darstellt. Wieder gilt, daß Anwenderdienste eine eindeutige Funktion und entsprechend eine einzige Signatur haben.
- Externe Dienste sind Dienste, die auf vorhandenen, in der Regel nicht beeinflussbaren Programmen (z.B. legacy Problematik) beruhen. Sie werden von Spezialisten gewartet und entwickeln sich entsprechend dem Stand der Wissenschaft und der Gesetzgebung. Diese Dienste ändern sich also. Bei der Diensteananspruchnahme muß es möglich sein, anzugeben, welche Modifikationen berücksichtigt werden sollen. Das hat zur Folge, daß ein Dienst mehrere Varianten seiner speziellen Aufgaben (Funktionen) anbieten kann, die dann wenigstens zum Teil über verschiedene Interfaces angesprochen werden müssen. Die Interfaces sollten es daher erlauben, syntaktische und semantische Informationen getrennt zu übertragen. Dies wird ausführlich in 3.2.2 erläutert.

Die Zuordnung von bestimmten Diensten zu diesen Klassen ist nicht immer eindeutig und sollte daher pragmatisch erfolgen. Dienste unterscheiden sich in einigen wichtigen Punkten von Objekten. Dazu gehört vor allem

- daß Dienste keine Eigenschaften von anderen Diensten erben,
- daß ihre Funktionen nicht durch Vererben überladen werden können,
- daß die Ausführung von Funktionen von Diensten sequentiell und seiteneffektfrei erfolgen muß und
- daß die Funktionen der Dienste nicht mehr von beliebigen Stellen aufgerufen werden können.

Dagegen sollen Dienste, ähnlich den Objekten, Informationen selbst besorgen können und nicht wie Module darauf angewiesen sein, daß alle erforderlichen Daten beim Aufruf übergeben werden. Daraus folgt, daß Objekte, die keinen Gebrauch von Vererbung und Überladung machen, auch als Dienste bezeichnet werden können. Das trifft vor allem für Systemdienste und die daraus aufgebauten Anwenderdiensten zu. Sind solche Objekte nach den Vorgaben der Objekt Management Group (OMG) entwickelt, das heißt vor allem, daß ihr Interface mit der Interface Definition Language (IDL) beschrieben wird, so heißen wir sie verteilte oder IDL-Objekte. Solche Objekte können später unter CORBA verwaltet und verteilt werden.

3 Eigenschaften von Diensten

Die folgenden Texte beziehen sich vor allem auf Anwenderdienste und externe Dienste.

3.1 Struktur von Diensten in heterogenen Umgebungen

Dienste lassen sich auf verschiedene Arten realisieren (Methoden eines Objektes, Methodensequenzen, Programme). Im Gegensatz zu Funktionalitäten, die ganz in eine Applikation eingebunden sind, müssen Funktionalitäten, die durch Dienste angeboten werden, mit einem eigenen Interface versehen und durch eine Reihe von zusätzlichen Informationen gekennzeichnet werden. Das Interface dient dazu, Botschaften zu empfangen oder zu versenden. Wie schon im letzten Kapitel ausgeführt, sollte dieses Interface, wo immer möglich, mittels der IDL beschrieben werden.

Stehen Varianten oder gar alternative Dienste zur Verfügung, so müssen vom Diensteanbieter Kriterien angegeben werden, nach denen ein Dienst ausgewählt werden kann. Dazu gehören Informationen über

- den Zugang,
- die Kosten,
- die Verfügbarkeit,
- die Methoden,
- die Einschränkungen,
- die Qualität der Ergebnisse.

Diese Informationen müssen vom Diensteanbieter bereitgestellt und in einem Diensterepository allen Diensteanforderern zur Verfügung stehen. Damit ergibt sich die in Abb. 3.1 gezeigte Struktur eines Dienstes.

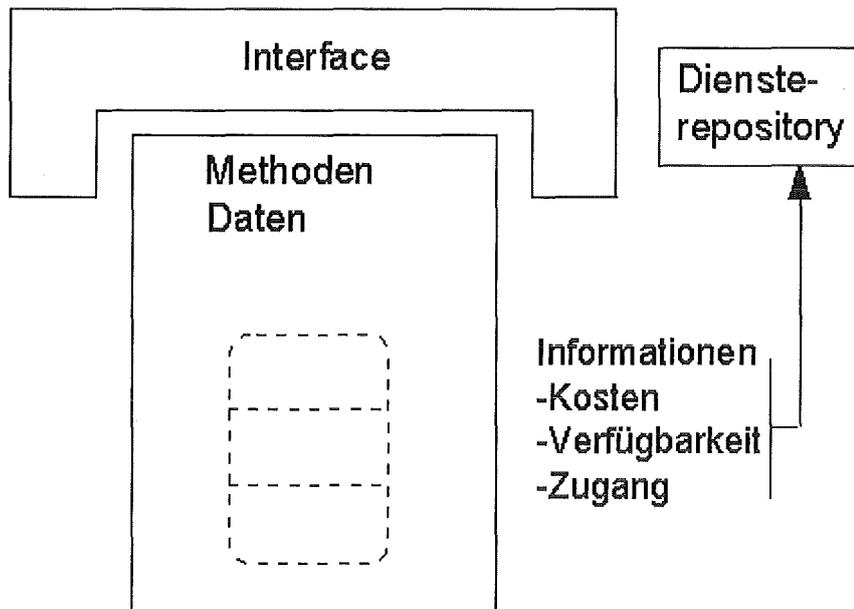


Abb. 3.1: Struktur eines Dienstes

Wie gesagt, kann die Implementierung dieser Struktur auf verschiedene Weise geschehen. Solange gilt, daß ein Dienst ein eigenständiger, abgeschlossener Prozeß ist, auf den nur über wohldefinierte Schnittstellen zugegriffen werden kann. Wichtig sind dabei

- die Kapselung von Daten, Methoden und Informationen, durch die der Dienstenutzer nichts über Implementierungs-Details wissen muß;
- die Unabhängigkeit von Seiteneffekten, die es dem Dienstenutzer erlauben, den Dienst allein über die Schnittstelle anzusprechen;
- die Bereitstellung aller Informationen, die zur Bewertung des Dienstes durch ein Dienstemanagementsystem nötig sind.

Will man Dienste verwalten, so wird es notwendig, weitere Zustandsvariablen einzuführen, die etwa den aktuellen Stand einer Diensteausführung (Bearbeitungsstand) angeben.

3.2 Aufbau eines Dienstes

Um einen Dienst auf seine Anwendung für eine bestimmte Aufgabe einschätzen zu können, genügt es nicht nur, seine Funktionalität zu kennen. Wichtig ist eine Beschreibung eines Dienstes so, daß seine Verwendung (Funktionalität), seine Anwendung (Schnittstelle) und seine Identifikation auch automatisch zu ermitteln sind. Für einen Dienst im UIS sollten folgende Informationen bereitgestellt werden:

- eindeutiger Identifier,
- Beschreibung der Funktionalität einschließlich ihres Gültigkeitsbereiches, Beschreibung der Schnittstellen,
- Angabe von Zusatzattributen zur Auswahl eines Dienstes bei Alternativmöglichkeiten,
- Metainformation zur Bewertung des Dienstes durch das Dienstemanagement.

Die Angabe eines eindeutigen Identifiers wird zur Lokalisierung des Dienstes verwendet. Für das Auffinden des Dienstes können Trader oder Broker eingesetzt werden, die Kommunikation kann etwa der Kommunikationsinterpret (KIP) aus dem Projekt TZUI übernehmen.

3.2.1 Beschreibung der Funktionalitäten

Wir gehen zunächst davon aus, daß der Diensteanbieter die zuverlässigsten Informationen über einen Dienst besitzt. Es wird also Aufgabe des Anbieters sein, seinen Dienst in das bestehende Angebot einzuordnen, d.h. Metainformationen bereitzustellen, anhand dessen die Dienstemanagement-Komponente die Diensteauswahl unterstützen kann. Dazu muß zunächst ein Dienstemanagement-Konzept entwickelt werden, aus dem heraus eine Dienstebeschreibung bereitgestellt wird, die der Diensteanbieter vervollständigt. Dieses Formular kann später dazu dienen, neue Dienste automatisch bekannt zu machen. Ähnlich einem Datenbankadministrator wird es einen Dienstadministrator geben, der für das Eintragen der Dienste verantwortlich ist. Neben bekannten Auswahlkriterien wie funktionaler Beschreibung und Zugangsbeschreibung müssen Zugriffsberechtigungen erteilt werden. Diese Aufgaben werden kaum vollständig automatisierbar sein, da nicht jeder Diensteanbieter alle Benutzer oder Benutzergruppen kennen kann und er somit die Entscheidung, für welche Gruppen er welche Zugangsprivilegien erteilen muß, nicht treffen kann.

Die Beschreibung der Funktionalität wirft aber noch weitere Fragen auf. Berechnungsdienste können über ein Gedächtnis verfügen. Damit unterscheiden sich die Ergebnisse mehrerer gleichlautender Aufrufe eines Dienstes. Bei einer Ausbreitungsrechnung müssen mehrere Varianten möglich sein, sowohl die Berechnung der neuen Situation mit Berücksichtigung der bisher erstellten Berechnungen (Aufsummieren der Bodendepositionen), als auch die gleiche Berechnung ohne Aufsummieren von einem anderen Arbeitsplatz aus.

3.2.2 Beschreibung der Schnittstellen

Ein Dienst beschränkt sich nicht nur auf eine Funktionalität. Der Zugang zu den verschiedenen Funktionalitäten muß bei der Klassifizierung beachtet werden. Der Zugang zu einer Funktionalität eines Dienstes geschieht über die Schnittstelle. Ein Dienst kann somit verschiedene Schnittstellen haben. Jede Schnittstelle kann mehrere Operationen enthalten. Jede Operation wird über eine Signatur syntaktisch beschrieben. Die Signatur ist vom Aufbau her mit einer Parameterklammer beim Aufruf einer Routine in einer höheren Programmiersprache vergleichbar.

Bei der Beschreibung der Schnittstelle wird ein Ansatz verfolgt, der Änderungen transparent machen soll. Wichtig ist vor allem, daß existierende Anwendungen weiter funktionieren, wenn die Schnittstelle verändert wird. Dies soll erreicht werden, indem Attribut-Wert-Paare für die Signatur eingesetzt werden. Eine Anwendung muß dann nur überprüfen, ob alle Attribute, die sie für die Ausführung des Dienstes benötigt, mit Werten belegt sind. Erweiterungen der Signatur bleiben damit für alle bestehenden Anwendungen ohne Bedeutung. Dieser Ansatz erfordert allerdings auf der Diensteanbieterseite zusätzlichen Aufwand. Bei Diensten, die auf validierten Modellen und Programmen beruhen, müssen Frontends implementiert werden, die die Umsetzung von der Schnittstellenspezifikation auf die beim Dienst vorhandene Schnittstelle leisten.

3.2.3 Beschreibung von Zusatzattributen

Zusatzattribute ermöglichen es, die Auswahl von Diensten durch das Dienstemanagement zu unterstützen. Möglich sind beispielsweise Angaben über den Ausführungszeitpunkt. Der Anforderer kann angeben, ob eine sofortige Ausführung erwünscht ist oder ob er einen kostengünstigeren Tarif zu einem späteren Zeitpunkt vorzieht. Ebenso können anfallende Kosten für einen Dienst festgesetzt werden. Neben statischen Zusatzattributen können noch eine Reihe von dynamischen Zusatzattributen existieren, die die Verfügbarkeit des Dienstes (Zugang über Netz, Verfügbarkeit der Hardware) und die das Antwortzeitverhalten eines Dienstes beschreiben (Auslastung der Hardware).

3.3 Klassifizierung von Diensten - Dienstetypen

Viele Dienste sind in ihrer Funktionalität in ähnlich, werden von anderen abgeleitet oder aus ihnen entwickelt. Es kann somit Dienste geben, die in bestimmten Bereichen zueinander kompatibel sind. So können die Funktionalitäten zweier Dienste eine gemeinsame Teilmenge

besitzen, d. h., sie können bestimmte gleichartige Funktionen durchführen. Dies gestattet, auf einen ähnlichen Dienst auszuweichen, wenn der bevorzugte Dienst nicht verfügbar ist. Bei der Auswahl möglicher Dienste müssen Kriterien zur Beurteilung von Ähnlichkeiten zwischen Diensten herangezogen werden. Wir unterscheiden dazu vier Ebenen jeweils in Syntax und Semantik

- Datentypeebene
- Operationsebene
- Interfaceebene
- Diensteebene.

Für technisch-wissenschaftliche Dienste muß folgendes beachtet werden:

- Die Gleichsetzung von Semantik und Syntax der Datentypen ist nicht ohne weiteres möglich. Eine Änderung der Einheit eines Parameters (Übergang von m in cm) verändert die Semantik, nicht aber die Syntax.
- Die Gleichsetzung von Syntax und Semantik würde bei Operationen voraussetzen, daß eine Änderung (anderer Lösungsalgorithmus) auch eine Änderung der Syntax bedeuten würde. Es lassen sich aber sehr wohl unterschiedliche Simulationsalgorithmen über die gleiche Syntax ansprechen.
- Im Gegensatz zu Informatikansätzen kann ein Dienst auch dann geeignet sein, wenn er ein anderes Interface bereitstellt. Damit läßt sich die Zuordnung zur gleichen Klasse nicht dadurch bestimmen, daß er als Untermenge die gleichen gewünschten Operationen bietet wie ein in der Klasse enthaltener Dienst.

Die Zusammenfassung von Diensten zu Dienstetypen geschieht aufgrund semantischer und nicht syntaktischer Kriterien. Wie ausgeführt, sind sie nicht ohne weiteres gleichzusetzen.

3.4 Vorschlag für Schnittstellen in Verbindung mit Diensten

Zur Beschreibung und zur Verwendung von Diensten werden drei Schnittstellen vorgeschlagen:

- Die erste Schnittstelle wird verwendet, um Dienste innerhalb des Systems bekanntzumachen. Diese Schnittstelle verwenden Diensteanbieter, um ihren Dienst zu exportieren.
- Die zweite Schnittstelle nehmen Diensteanforderer in Anspruch, um ihren Dienstewunsch zu äußern und die Ergebnisse eines Diensteaufufes zu bekommen.
- Die dritte Schnittstelle stellt die Schnittstelle von der Strategiekomponente zum aufzurufenden Dienst dar.

Die beiden ersten Schnittstellen können durch den Einsatz von Formularen realisiert werden. Sie sind neu zu entwickeln. Für sie wird im folgenden ein Vorschlag unterbreitet.

Die dritte Schnittstelle ist die Schnittstelle zwischen Diensteanbieter und Strategiekomponente zum Aufruf des Dienstes. Hier werden die Aufrufmöglichkeiten in der Regel durch den Diensteanbieter vorgegeben. Die Festlegung auf eine Schnittstellendefinition führt zwar zu einer Standardisierung im Umweltinformationssystem, fraglich ist nur, ob jeder Diensteanbieter, der für das UIS interessante Dienste exportieren möchte, auf eine solche Schnittstelle festzulegen ist. An dieser Stelle sollte deshalb möglichst nach pragmatischen Lösungen gesucht werden. Im Rahmen des INTEGRAL-Projektes wird an dieser Problematik weiter gearbeitet.

Die Formulare für Dienstexport und Diensteanforderung bestehen aus Attributen und den zugehörigen Werten. Für die Beschreibung der Attribute werden folgende Komponenten vorgeschlagen:

Wert:	der eigentliche Wert eines Attributes
Name:	Bezeichnung des Attributes
Datentyp:	Datentyp des Elements
Format:	Das Format für das Datenelement
Wertebereichstyp:	Aufzählungstyp, Beschreibungstyp, Bereichstyp, Einwerttyp
Wertebereich:	Angabe des Wertebereiches in der Syntax entsprechend dem Typ
Wertebereichszusatz:	Mittelwert, etc.
Einheit-Typ:	UIS-Notation, SI-System
Einheit:	physikalische Einheit
Ortsangabe:	Ort der Erhebung, Probenahme, etc.
Zeitangabe:	Zeit der Messung, Probenahme, etc.
Meßverfahren:	Art der Datenerhebung.

Ein Diensteanbieter generiert aus den für seine Anwendung erforderlichen Eingabedaten und den gelieferten Ausgabedaten eine Beschreibung, in der für jedes Attribut die oben beschriebenen Komponenten angegeben sind. Darüber hinaus hat der Diensteanbieter weitere Angaben zu seinem Dienst zu liefern. Diese Beschreibung erfolgt in einem Formular, für dessen Aufbau wir folgende Strukturen vorschlagen:

Formular Dienstexport

Dienstexporteur

- Name:

- Zugang:

Dienstbeschreibung:

Zugangsberechtigung

- Benutzergruppe:

- Liste der Nutzerkennungen für Einzelnutzer:

Verfügbarkeit

-Wochentag:

-Wochenende:

-Feiertag:

Kosten

- Aufruf (einmalige Fixkosten):

- Kosten/Nutzungsdauer:

- Wahrung:
Antwortzeit:
Eingabeparameter
E_Attribut 1
E_Attribut 2
:
:
Ausgabeparameter
A_Attribut 1
A_Attribut 2
:
:

Ende der Dienstbeschreibung fur den Export.

Die Beschreibung der Attribute sollte dabei fur alle relevanten Komponenten Angaben enthalten.

Der Diensteanforderer wendet sich mit einem ahnlichen Formular an die Strategiekomponente. Auch hier sind Angaben uber gelieferte Daten mit ihrer Beschreibung und uber die erwarteten Ergebnisse erforderlich. Diese Angaben sollten sich aber auf ein Minimum beschranken und werden in der Regel sogar entfallen, da sie durch den Dienstetyp weitgehend festgelegt sind. Fur den Aufbau des Formulars wird folgende Struktur vorgeschlagen:

Formular Diensteanforderung

Identifikation Diensteanforderer:

Dienstetyp:

gewunschte Antwortzeit:

Kosten:

Verfugbare Daten

E_Attribut 1

E_Attribut 2

:

:

Erwartete Ergebnisse

A_Attribut 1

A_Attribut 2

:

Ende der Beschreibung des Formulars Diensteanforderung.

3.5 Wie findet man Dienste

Bei der Identifikation von Diensten können zwei Vorgehensweisen, die wir in diesem Zusammenhang als Topdown- oder Bottomup-Ansatz bezeichnen, verwendet werden. Der Topdown-Ansatz erlaubt die Analyse des Aggregationsprozesses in UIS-Komponenten. Dabei wird, ausgehend vom Informationsbedarf, eine Umsetzung in eine Informationsanfrage vorgenommen. Diese Anfrage wird in Teilschritte aufgespalten und solange verfeinert, bis sich jeder Teilschritt als Dienst formulieren läßt. Der Zugriff auf diesen Dienst führt zu der Information oder erzeugt diese Information. Der Bottomup-Ansatz identifiziert Datenbestände und Methoden in den einzelnen Komponenten und faßt diese Einzeldienste in Klassen aus ähnlichen Diensten zusammen.

Beim Bottomup-Ansatz ist die Semantik der Daten zweitrangig. Wichtig ist festzustellen, welche Methode auf die Daten angewandt werden und welche dieser Methoden von generellem Interesse sind. Es können drei Bereiche identifiziert werden:

- Datenverwaltung
- Bearbeitung
- Datenausgabe, Darstellung

Die Datenverwaltung schließt dabei Datenimport und Datenexport ein. Schreiben und Lesen von Daten können solange gemeinsam behandelt werden, wie sie aus Sicht der Dienste keine Unterschiede aufweisen. Damit führt der Bottomup-Ansatz eher zu Dienstklassen, die sich auf die Primärdaten und die Infrastruktur des UIS (Druckerdienste, Anzeigendienste) beziehen.

Der Topdown-Ansatz hilft dagegen, komplexe Dienste im UIS zu beschreiben. Beispiele für komplexe Dienste, die von allen Komponenten genutzt werden können, sind:

- Aufbereitung von Daten
- Bildung von Zeitreihen
- Bereitstellung von Hintergrunddaten und raumbezogenen Basisdaten
- Simulationen
- Verschneidungen von Grafikobjekten.

Werden mit Hilfe der beiden Verfahren Applikationen in Dienste aufgebrochen, so ergibt sich ein gewisser Widerspruch in der Klassifizierung. Werden die Maßstäbe für eine Klassenzugehörigkeit zu weit gefaßt, so erfüllen Dienste die Anforderungen einer Klasse, obwohl sie für die dort geforderten Anforderungen nicht geeignet sind. Werden die Parameter zu eng gefaßt, so sind einelementige Klassen die Folge. Dies würde der bisherigen Vorgehensweise entsprechen, indem für jede Aufgabe im UIS ein Weg realisiert wird. Die Aufteilung in Klassen muß diesen Widerspruch auflösen. Dabei wird der Zusammenfassung ähnlicher Dienste in eine Klasse der Vorrang vor einer eindeutigen Zuordnung gegeben. Das auch deshalb, weil Dienste mit unterschiedlichen Funktionalitäten auch in mehreren Klassen auftreten können.

3.6 Dienste unter WWW

Das WWW hat drei Eigenschaften, die eine Nutzung von Diensten in der beschriebenen Art behindern:

- es ist transaktionsorientiert, d.h. sobald die Ergebnisseiten versandt werden, ist die Transaktion abgeschlossen und daher kein weiterer Zugriff auf Ergebnisse möglich,
- es anonymisiert die Benutzer, d.h. alle Dienstenutzer greifen auf dieselbe Prozeßumgebung zu,
- es arbeitet blockierend, d.h. Dienste, die längere Zeit in Anspruch nehmen, können praktisch nicht genutzt werden.

Will man trotzdem Dienste nutzen, die von anderen angeboten werden, so sind diese für eine Nutzung unter WWW derart zu kapseln, daß sie

- jedem Dienstenutzer einen eindeutigen Schlüssel zuordnen, den sie dem Nutzer auch mitteilen,
- ihre Ergebnisse in einem Repository unter einer eindeutigen Struktur, die mit dem Schlüssel verbunden ist, ablegen,
- dem Benutzer über Statusmeldungen über ihren Zustand informieren.

Zur Zeit ist die Übermittlung von Statusmeldungen nur an bestimmte Browser möglich. Verwendet ein Dienst solche spezifische Eigenschaften (etwa CCI), so muß er in der Lage sein, den Typ des Browsers zu erkennen, um evtl. auf die Nutzung dieser Eigenschaften verrichten zu können.

Wird ein Dienst, etwa über TZUI, nicht blockierend in Anspruch genommen, so benötigt die Clientseite Methoden zur Auftragsverwaltung. Sie sollten analog denen, die im TZUI angeboten werden, gestaltet werden.

4 Dienste im Rahmen des UIS

Wichtige Funktionalitäten, die im Rahmen eines UIS vom Anspruch des UIS-BW zur Verfügung gestellt werden sollten, sind:

Informationsquellen

Zweck der Informationsquellen ist das Auffinden von Informationen über Ergebnisse (vorhandene Informationen), über reale Daten, über Methoden zur Erzeugung von Daten

(virtuelle Daten) und über Methoden zur Verarbeitung und Darstellung. Das Informationsquellenmanagement muß dazu mit abstrakten Informationen (Metainformationen) umgehen können. Informationen, reale Daten und Methoden zur Erzeugung von Daten haben meist nur eine zeitlich sehr begrenzte Gültigkeit. Es ist daher essentiell, das Informationsquellenmanagement möglichst offen zu gestalten, zu erlauben, daß insbesondere externe Informationen, Daten und Methoden eingeschlossen werden können und daß Anbieter solcher Ressourcen die Beschreibung ihrer Angebote eigenverantwortlich in einen Informationsquellenkatalog einbringen können. Wichtige Komponenten des Informationsquellenmanagement sind daher

- ein Verwaltungsdienst, der in der Lage ist, Metainformationen über Informationen, Daten und Dienste bereitzustellen,
- ein Strategiedienst, der in der Lage ist, aus den Wünschen des Nutzers, den Angaben eines Diensteanforderers und den Metainformationen Strategien für die Informationsbeschaffung vorzuschlagen und geeignete Dienste zu identifizieren und anzusprechen.
- ein Verknüpfungsdienst, mit dessen Hilfe Ketten von Verarbeitungsschritten aufgebaut werden können.

Das Informationsquellenmanagement verlangt die Verarbeitung syntaktischer und semantischer Informationen in Form dynamischer und statischer Attribute. Anregungen, wie ein Informationsquellenmanagement gestaltet werden kann, bieten Repositories von Case Tools und die Traderkonzepte. Unter CORBA verfügbare Object Request Broker können Objekte, die auf Basis von CORBA angeboten werden (IDL-Objekte), verwalten. Die IDL gibt an, wie Objektinterfaces standardisiert werden können.

Informationsbeschaffung

Informationen können auf drei Arten beschafft werden:

- Der Auskunftsdienst
Ist eine Information schon vorhanden, so muß sie in die lokale Arbeitsumgebung transferiert und für die weitere Verarbeitung bereitgestellt werden. Information kann dabei als Bericht vorliegen (Beispiel Berichte zur Umwelt der LfU), in Datenbanken abgelegt sein (Beispiel Umweltdatenkatalog) oder aus vorangegangenen Arbeiten resultieren (Beispiel Archiv in UFIS/TULIS). Im Rahmen des GLOBUS-Projektes wird solch ein Dienst, der einen einheitlichen Zugang zu Katalogen und Berichten erlaubt, entwickelt.
- Der Datenbeschaffungsdienst
Ist die Information noch nicht verfügbar, so muß sie erstellt werden. Dabei gibt es zwei Möglichkeiten. Die erste ist die, daß die für die Information benötigten Daten verfügbar sind. Die Informationsbeschaffung hat dann die Aufgabe, einen Selektor anzubieten, den Zugriff auf die Datenbank zu ermöglichen und das Ergebnis der Query in die lokale

Arbeitsumgebung zu transferieren. Dies kann etwa durch einen Datenbeschaffungsdienst geschehen, der weiß, wie auf die gewünschten Daten zugegriffen werden kann. Der eigentliche Zugriff erfolgt dann über einen produktunabhängigen Datenzugriffsdienst.

- Der externe Dienst
Sind die zur Generierung von Informationen benötigten Daten nicht verfügbar (z.B. weil Prognosen erstellt oder Ursachen ermittelt werden sollen), so müssen sie mittels Algorithmen erstellt werden (virtuelle Daten). Algorithmen zur Erstellung virtueller Daten erfordern in der Regel Spezialisten und sollten deswegen als eigene Dienste verfügbar gemacht werden. Die Komponente Informationsbeschaffung sollte daher eine Schnittstelle anbieten, über die externe Dienste aktiviert und die Ergebnisse ihrer Inanspruchnahme in die lokale Arbeitsumgebung transferiert werden können. Die Diensteananspruchnahme kann über Agenten geschehen, die insbesondere wissen, wie die Transformationen von und zur lokalen Arbeitsumgebung erfolgen müssen.

Informationsspeicherung

Informationen erlangen in der Regel erst dadurch an Bedeutung, daß sie im Kontext der eigenen Erfahrungen weiterverarbeitet und mit anderen Informationen verknüpft werden können. Wichtiger Bestandteil eines Umweltinformationssystems sind daher Repositories, in denen die Ergebnisse der Inanspruchnahme von Diensten zwischengespeichert und von dem aus sie weiterverarbeitet werden können. Es wäre wünschenswert, wenn für ein Repository die gleichen Datenbanken verwendet werden könnten, in denen Sachdaten abgelegt sind (in der Regel eine relationale Datenbank). Beim gegenwärtigen Stand der Entwicklung ist das noch nicht möglich. Die Verwendung von Repositories bietet die Chance, wenigstens innerhalb eines Anwendungssystems Datenhomogenität zu schaffen. Dazu sind Filter und Browser bereitzustellen, die das Wissen darüber enthalten, wie verschiedene Datenmodelle aufeinander abgebildet werden können. Wichtige Dienste im Umfeld eines Repositories sind also

- Archivierungsdienst
- Konvertierungsdienst
- Datentransferdienst
- Browsingdienst.

Informationsverarbeitung

- Der Visualisierungsdienst
Wichtigstes Instrument der Informationsverarbeitung ist die interpretierende Darstellung. Die Darstellung kann mit Hilfe von statistischen Verfahren (z.B. Mittelungen), mit Geschäftsgraphiken (z.B. Balkendiagrammen) und über die Einbettung in den Raumbezug (Lagekarten) geschehen. Die Einbeziehung des Raumbezugs erfolgt über die Visualisierungskomponente (Viewer) geographischer Informationssysteme. Dabei ist es wichtig, Funktionalitäten verschiedener solcher Systeme alternativ nutzen zu können und Wege aufzuzeigen, wie solche GIS-Funktionalitäten als Dienste in Umgebungen eingebunden werden können, die etwa

Benutzern von PC-Arbeitsplätzen zur Vorgangsbearbeitung zur Verfügung stehen. Dienste, die von GIS-Servern angeboten werden, stellen besondere Anforderungen an die Interoperabilität, an Filter und Browser.

- Der Berechnungsdienst
Häufig müssen Informationen mit einfachen Methoden weiterverarbeitet werden. Dies ist die Aufgabe des Berechnungsdienstes. Im Rahmen von Umweltinformationssystemen bietet sich an, Berechnungen, beispielsweise mittels Tabellenkalkulation, durchzuführen.
- Der Reportdienst
Ergebnisse, die mit Hilfe von Umweltinformationssystemen erzeugt werden, müssen dokumentiert und in Berichten verbreitet werden. Zunehmend wird es erforderlich, diese Berichte als Hardcopies, in elektronischer Form und als HTML-Seiten zur Verfügung zu stellen. Dies zu unterstützen ist Aufgabe des Reportdienstes.
- Der Ausgabedienst
hat die Aufgabe, Informationsobjekte aus der eigenen Arbeitsumgebung zu versenden. Dies kann sowohl in andere Umgebungen als auch zu Druckern und Plottern erfolgen.

Kooperatives Arbeiten

Das Arbeiten mit Systemen, die stark diensteorientiert sind, erfordert neue Wege der Kommunikation zwischen Dienstenutzern und Diensteanbietern und die Entwicklung von Techniken kooperativen Arbeitens. Das gilt zunächst für das gemeinsame Erstellen von Berichten (Authoring), mittelfristig aber auch für die gemeinsame Interpretation von Ergebnissen (Multi Media Conferencing - MMC) oder gar die gemeinsame Erstellung neuer Informationen (kooperatives Arbeiten). Daraus ergibt sich eine Notwendigkeit für folgende Dienste:

- Informationsdienst (blackboard)
- Konferenzdienst (MMC)
- Botschaftsdienst (Multi Media Mail - MMC)
- Berichtsdienst (Authoring System).

Systemdienst

Der Umgang mit Diensten erfordert die Systemunterstützung in verschiedenen Bereichen. Dazu gehören:

- ein Sitzungsverwaltungsdienst,
- ein Kommunikationsverwaltungsdienst,
- ein Oberflächen-Generatordienst.

Diese Dienste sind für viele Anwendungen relevant. Es wird erwartet, daß sie als Standardprodukte verfügbar und evtl. in Betriebssysteme integriert werden.

Im Rahmen des UIS-BW steht als Kommunikationsverwaltungsdienst der Kommunikationsinterpret KIP /10/ zur Verfügung.

5 Integration von Diensten in UIS-Anwendungen

Die in Kapitel 2 gegebene Definition von Dienst berücksichtigt die Besonderheiten im UIS. Durch die Vorgabe, monolithische Blöcke solange als Dienst weiterzuverwenden, bis sie im Laufe der Weiterentwicklung in feine granulare Dienste aufgebrochen werden, können Ansätze, die von Neuentwicklungen ausgehen, nur berücksichtigt werden, wenn sie die Integration des Vorhandenen gewährleisten. Im INTEGRAL-Ansatz sind zwei unterschiedliche Arten der Diensteintegration vorgesehen. Einerseits müssen die Dienste innerhalb von UIS-Anwendungen für andere Anwendungen bereitgestellt werden, andererseits müssen Dienste, die außerhalb der UIS-Umgebung angeboten werden, integrierbar sein.

5.1 Dienste in UIS-Anwendungen

Im ersten Schritt wurden Anwendungsteile innerhalb einer UIS-Anwendung identifiziert, die als Dienste angeboten werden sollen. Diese Dienste sollen weiterhin über die Benutzeroberfläche einer Anwendung in Anspruch genommen werden. Dies geschieht durch den Benutzer, der interaktiv mit einer UIS-Anwendung arbeitet. Dem Benutzer soll sich dabei das Erscheinungsbild gegenüber der monolithischen Version nicht unterscheiden. Zusätzlich sollen die Dienste auch von anderen Diensten aufgerufen werden können. Das bedeutet, daß Dienste aus der Oberfläche und damit mit im Dialog erfaßten Parametern ebenso aktivierbar sein müssen wie über den Aufruf von einem anderen Dienst aus. Es genügt jedoch nicht, nur den Zugang zu diesen Diensten sicherzustellen. Ein Dienst muß sich selbst wieder an INTEGRAL wenden können, um Dienste anzufordern, die eine Teilaufgabe erledigen können. Auch hier muß die feste Zuordnung von Problemstellung und Realisierung aufgehoben werden.

5.2 Externe Dienste

Bei externen Diensten stellt sich das Problem, daß der Dienstbereitsteller einen Dienst zwar anbieten möchte, daß er aber die Schnittstelle nicht zwingend nach UIS-Richtlinien aufsetzen muß. Damit solche Dienste dennoch integriert werden können, ist ein pragmatischer Ansatz erforderlich. Es muß ein Frontend implementiert werden, der das Verhalten eines UIS-Dienstes simuliert. Damit wird es möglich, auch externe Dienste alternativ zu UIS-Diensten anzufordern. Der Frontend muß dabei sicherstellen, daß alle Informationen, die der externe Dienst benötigt, bereitgestellt werden.

5.3 Integration von Diensten in Anwendungssystemen

Ausgang dieser Strategie ist ein Nutzermodell, bei dem angenommen wird, daß der Nutzer die Funktion eines Informationsmanagers erfüllt. Er ist Spezialist und gewohnt, mit einer Vielzahl von Funktionalitäten eines Umweltinformationssystems gekonnt umzugehen. Er benötigt eine Oberfläche, von der aus er möglichst viele der Funktionalitäten ansprechen und in Form von Diensten in Anspruch nehmen kann. Die Ergebnisse seiner Arbeiten sollten in Form von Berichten und Multimediadokumenten zur Verfügung stehen. Das WWW und seine Browser (etwa Mosaic) bieten dazu eine Vielzahl von Werkzeugen an, mit denen Oberflächen gestaltet und auf entfernte Dienste zugegriffen werden kann. Diese Werkzeuge sind leicht verfügbar und einsetzbar und ermöglichen eine Arbeitsumgebung aufzubauen, die auch von Außenstehenden akzeptiert wird und bei Befolgung der WWW-Regeln in ihrer Funktionalität leicht verstehbar ist. Kern des darüber ansprechbaren integralen Systems ist ein Repository, über das die einzelnen Komponenten Informationen tauschen können. Solch ein Repository muß zunehmend Elemente einer objektorientierten Datenbank anbieten.

5.4 Nutzung externer Dienste

Abb. 5.1 zeigt den generellen Ablauf einer Diensteananspruchnahme und der nachfolgenden Verarbeitung ihre Ergebnisse in WWW. Der Diensteananspruchnahme geht das Auffinden des geeigneten Dienstes voraus. Bei der Diensteananspruchnahme selber muß dann eine eindeutige Auftragsnummer generiert werden, über die Statusabfragen durchführbar sind, Ergebnisse identifiziert werden können und die Auftragsverwaltung erfolgt.

Das in Abb. 5.1 gezeigte Repository sollte beim Diensteanbieter lokalisiert sein. Es sollte nach einer standardisierten Art mindestens die Funktionalitäten Store, Get, Transfer und Delete für alle im Repository gemanagten abstrakten Datentypen anbieten.

Die Vorteile dieses Ansatzes liegen auf der Hand. Sie umfassen

- das einheitliche look and feel der WWW Multi-Media-Browser und der durch sie möglichen Darstellungen von Informationen,
- die Möglichkeiten, Ressourcen, die im Internet angeboten werden, mit nutzen zu können,
- die Nutzung der Kommunikationsdienste im Internet und die relative Unabhängigkeit von Hardware.

Es ist möglich, PC's oder Workstations als Endgeräte zu verwenden.

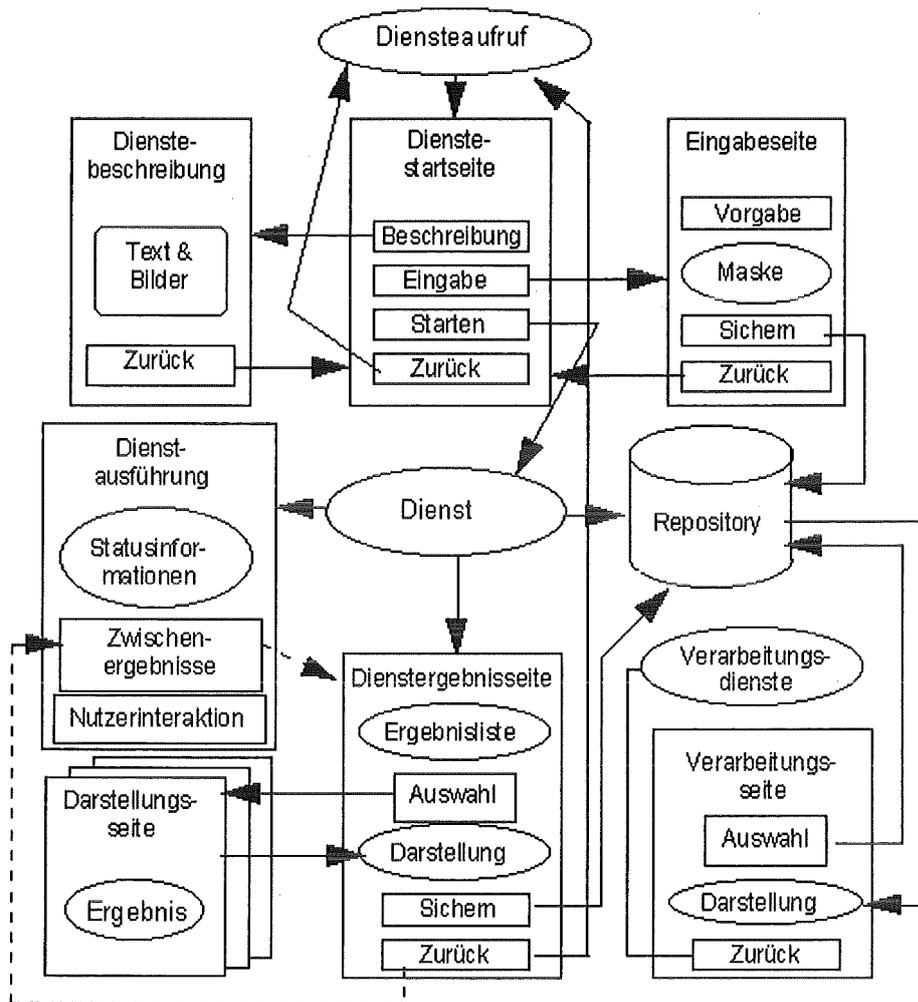


Abb. 5.1: Dienste und die Verarbeitung von Ergebnissen einer Diensteanforderung

Die Grenzen dieser Lösung werden offensichtlich, wenn kein Internet-Anschluß verfügbar ist, wenn Daten und Informationen nicht nur als HTML-Seiten übertragen werden und wenn die Zahl der Objekte des Systems wächst. Dann ist es sinnvoll, den Ansatz um Konstrukte zu erweitern, wie sie von CORBA-Implementierungen angeboten werden (CORBA als Middleware). Nach [9] bieten heute verfügbare CORBA-Implementierungen folgende Unterstützung:

- Einheitliche Syntax von Schnittstellenbeschreibungen (IDL) zu Objektimplementierungen.
- Prinzipielle Unabhängigkeit von: Programmiersprachen, Betriebssystemen und Netzwerkprotokollen, also Verdeckung der technischen Heterogenität in verteilten Systemen.
- Ortstransparenter Zugriff auf verteilte Objekte.
- Statischer und dynamischer Zugriff auf Objekte.
- Beschreibung aller vorhandenen Objektimplementierungen im Interface Repository.

Voraussetzung, dies für Dienste zu nutzen, ist, daß die Dienste als Objekte im Sinne von CORBA implementiert sind, d.h. insbesondere, daß die Schnittstellen zu den Objektimplementierungen in der Interface-Definition Language beschrieben werden.

Zur Zeit werden existierende Dienste, komplexe Dienste, die ein nicht vorhersehbares Antwortverhalten haben und komplexere Managementprozeduren, wie etwa das Trading, nicht oder nur unzureichend unterstützt. Außerdem existieren noch wenig Erfahrungen mit dem Zusammenspiel von Objekt Request Broker verschiedener Hersteller. Trotzdem sollte die CORBA-Entwicklung sehr intensiv verfolgt werden.

Glossar

Abstrakter Datentyp (ADT):

Ein abstrakter Datentyp ist eine Sammlung von Daten und den zur Bearbeitung darauf definierten Operationen. Eine Spezifikation eines abstrakten Datentyps konzentriert sich auf die Beschreibung von Daten, nicht durch ihre Struktur, sondern durch die Sammlung von Operationen, die auf ihnen ausgeführt wird [Inc92].

Abstraktes Datenobjekt (ADO):

Ein abstraktes Datenobjekt ist die Instanzierung eines abstrakten Datentyps. Berechnungsschnittstelle (Computational Interface):
siehe Schnittstelle.

Client:

siehe Dienstenutzer

Client-Server-Computing:

Client-Server-Computing sollte zunächst als ein Verteilungskonzept von betrieblichen Funktionen und Daten verstanden werden, auf dem das Softwarekonzept aufbaut und dem das Hardwarekonzept folgt.

Dienst (engl. Service):

Ein Dienst ist eine abgeschlossene (Programm-)Einheit, die eine spezielle Aufgabe erfüllt. Ein Dienst kann eine Menge an Funktionen besitzen, die sich gegenseitig nicht beeinflussen. Ein Dienst wird von einem Diensteanbieter erbracht und von einem Dienstenutzer verwendet. Ein Dienst kann Methoden von Objekten aufrufen und an einer Rechnerschnittstelle anbieten. Damit ist diese Dienstedefinition weitreichender als die meistens in der Literatur verfügbare Definition, nach welcher ein Dienst eine Funktion ist, die von einem Objekt an einer Schnittstelle angeboten wird [Spa94].

Dienste im Umfeld von Umweltinformationssystemen beruhen häufig auf vorhandenen, in der Regel nicht beeinflussbaren Programmen (legacy Problematik). Sie werden von Spezialisten gewartet und entwickeln sich entsprechend dem Stand der Wissenschaft und der Gesetzgebung. Dienste ändern sich also. Bei der Diensteananspruchnahme muß es möglich sein, anzugeben, welche Modifikationen berücksichtigt werden sollen. Das hat zur Folge, daß ein Dienst mehrere Varianten seiner speziellen Aufgabe anbieten kann, die dann wenigstens zum Teil über verschiedene Interfaces angesprochen werden müssen. Diese Varianten nennen wir Funktionen eines Dienstes.

Diensteanbieter (engl. Server):

Ein Diensteanbieter bietet Dienste in einem Rechnernetz an.

Dienstebeschreibung:

Die Dienstebeschreibung beschreibt die Syntax des Dienstes und dessen Diensteeigenschaften.

Diensteeigenschaften:

Die Diensteeigenschaften beschreiben syntaktisch die Features eines Dienstes.

Dienstenutzer (engl. Client):

Ein Dienstenutzer verwendet Dienste eines oder mehrerer Diensteanbieter für die Durchführung seiner Aufgaben. Die Dienstenutzer steuern den Diensteanbieter.

Dienstetyp:

Ein Dienstetyp ist eine abstrakte Beschreibung von ähnlichen oder gleichen Diensten. Ein Dienstetyp wird durch eine Menge von Diensteeigenschaften charakterisiert. Die Zusammenfassung von Diensten zu Dienstetypen geschieht aufgrund der Features der Dienste, also aufgrund semantischer und nicht syntaktischer Kriterien. Diese Definition ist wie die Dienstedefinition weitreichender als die in der Literatur verwendete, wonach Dienste des gleichen Dienstetyps die gleiche Syntax besitzen [Spa94].

Dienstevermittler:

Prozess, bei dem die Auswahl eines Dienstes mit Hilfe von Such- und Selektionsfunktionen gemäß den Forderungen eines Dienstenutzers durchgeführt wird [Spa94].

Exporter:

Ein Exporter ist ein Server, der seine Leistungen innerhalb eines verteilten Systems über einen Vermittler anbietet [Spa94].

Funktion:

Eine Funktion ist eine von einem Dienst angebotene Operation.

Wir unterscheiden Basisfunktionen und Anwenderfunktionen. In den Basisfunktionen gehören die Methoden, die Abstrakte Datentyp Module anbieten, während die Anwenderfunktionen eher den Funktionsmodulen entsprechen.

Importer:

Ein Importer ist ein Client, der Leistungen eines Server über einen Vermittler in Anspruch nimmt [Spa94].

Leistungsmerkmal (Feature):

Ein Feature definiert eine vom Diensteanbieter festgelegte Fähigkeit eines Dienstes [Beu92]. Es stellt somit eine semantische Beschreibung des Dienstes dar.

Klasse:

Ein Klasse ist eine Beschreibung von einem oder mehreren Objekten mit einem einheitlichen Satz von Attributen und Methoden, inklusive der Beschreibung, wie neue Objekte der Klasse erzeugt werden [Coa91]. Eine Klasse unterscheidet sich von einem abstrakten Datentyp durch die Möglichkeiten der Vererbung, Polymorphie und dynamisches Binden.

Methode:

Eine Methode ist eine von einem Objekt angebotene Operation.

Modul:

Module sind Programmeinheiten, die eine spezielle Aufgabe erfüllen. Man unterscheidet Abstrakte Datentyp Module und Funktionsmodule. ADT-Module kapseln einen abstrakten Datentyp. Ihre Schnittstelle enthält Methoden. Ihr Ergebnis ist von der Vorgeschichte des ADO abhängig, sie haben also ein Gedächtnis. Funktionsmodule kapseln die Umsetzung von Methoden. Ihre Schnittstelle enthält ADT's. Ihr Ergebnis hängt allein vom Aufruf ab. Module sind in der Regel Bestandteile von Programmen.

Objekt:

Ein Objekt ist die Instanzierung einer Klasse.

Operation:

Operationen führen wir aus pragmatischen Gründen ein. Eine Operation ist eine abgeschlossene Programmeinheit, die syntaktisch durch ihre Signatur beschrieben wird. Operationen, die von einem Client verwendet werden können, werden bei Diensten als Funktionen und bei Objekten als Methoden bezeichnet.

Prozess-Definition:

Ein Prozess ist eine geordnete (parallel und/oder seriell) Menge von Prozessaktivitäten, die verbunden werden, um ein gemeinsames Ziel zu erreichen. Wird ein Prozess von einem anderen Prozess aufgerufen, so heißt er Teilprozess.

Prozess-Aktivität:

Teilschritt, der zur Erreichung des Zieles eines Prozesses nötig ist. [WfMC95].

Prozess-Beispiele:

Geschäftsprozess:

Prozess, der im Geschäftsleben eine Rolle spielt.

Workflow:

Prozesseteil, der auf Rechnern ausgeführt wird. Besteht in der Regel aus mehreren auf dem Rechner ausführbaren Workflow-Aktivitäten. Der Workflow wird in der Workflow-Prozess-Definition modelliert und kann in beliebig vielen Instanzen existieren. Die Instanzen eines Workflow-Prozesses werden vom Workflow-Management-System erzeugt und verwaltet.

Informationsbeschaffung:

Prozesse zur Beschaffung, Erzeugung oder Verarbeitung von Informationen entsprechen Diensten im Sinne dieser Definition.

Schnittstelle (engl. Interface):

Eine Schnittstelle ist eine Beschreibung einer Menge an Signaturen von möglichen Operationen, die ein Client von einem Server anfordern kann [OMG91]. Eine solche Schnittstelle wird auch als Berechnungsschnittstelle (Computational Interface) bezeichnet [Spa94]. Die OMG bietet zur Formulierung von Schnittstellen die Interface Definition Language (IDL) an. Objekte, deren Schnittstelle mit der IDL erzeugt wurde, heißen verteilte IDL-Objekte.

Schnittstellentyp (Interfacetyp):

Der Schnittstellentyp ist eine abstrakte Beschreibung gleicher Schnittstellen. Die Zusammenfassung erfolgt aufgrund von syntaktischen Kriterien.

Server:

siehe Diensteanbieter

Signatur:

Die Signatur beschreibt die Syntax der Operationsschnittstelle. Sie besteht aus dem Namen der Operation, sowie den Datentypen der Parameter und Ergebnisse der Operation.

Trading:

Vermittlung von Leistungen eines Servers in einem verteilten System.

Literatur

- /1/ Hamhaber, M.: Realisierung der Basisfunktionen eines Traders. Institut für Parallele und Verteilte Höchstleistungsrechner, Universität Stuttgart, 1992.
- /2/ Lurk, A.: Anforderungen an die Vermittlung von Dienstangeboten im Rahmen technisch-wissenschaftlicher Anwendungen. IKE 4-D-190, 1993.
- /3/ Schuch, A., Bußmann, M.: Technischer Zugang zu Umweltinformationen. 3. Workshop Integration von Umweltdaten, Schloß Dagstuhl, April 1995.
- /4/ Spaniol, O., Popien, C., Meyer, B.: Dienste und Dienstevermittlung in Client/Server-Systemen. Thomson's Aktuelle Tutorien, 1994.
- /5/ Tischendorf, M., Schmidt, F.: INSERV - Spezifikation von Diensten im UIS, Institut für Kernenergetik und Energiesysteme, Universität Stuttgart, IKE 4-D-133-2, 1994.
- /6/ Workflow Management Coalition. Glossary - A Workflow Management Coalition Spezifikation Nov. 1994. <http://WWW.aiai.ed.ac.uk/WfMC/>.
- /7/ Blank, K., et.al.: Projekt INTEGRAL: Integration von heterogenen Komponenten des UIS Baden-Württemberg. FAW Ulm, Okt. 1994.
- /8/ Kramer, R., Nikolai, R. (Hrsg.): WWW-UIS: Auskunfts-, Administrations- und Verwaltungsdienste. FZI-Bericht 10/95.
- /9/ Koschel, A.: Private Mitteilung im Rahmen des GLOBUS-Projektes.

Architektur des WWW- und CORBA-basierten UIS

*Arne Koschel, Ralf Kramer, Ralf Nikolai
Forschungszentrum Informatik (FZI)
Haid-und-Neu-Str. 10-14
D-76131 Karlsruhe*

Unter Mitwirkung von:

*Sven Behrens, Uwe Beutler, Corinna Habeck, Philipp Kirsch, Bettina Mayer, Peter Palmer,
Claudia Rolker, Oliver Scheffczyk, Martin Schoel, Manfred Walz*

1 EINLEITUNG	41
2 DIE ARCHITEKTUR VON WWW-UIS.....	41
3 CORBA ALS MIDDLEWARE IM UIS.....	43
4 ZUSAMMENFASSUNG	45
LITERATUR.....	45

1 Einleitung

Die Funktionalität der übergreifenden Komponenten des UIS Baden-Württembergs wird aktuell weitestgehend in Form monolithischer Applikationen zur Verfügung gestellt. Ein weiteres Problem ist, daß Systeme wie UFIS und TULIS nur auf proprietären Hardwareplattformen und Betriebssystemen zur Verfügung stehen. Diese Systeme sind zudem jeweils vor Ort, d. h., am Ort ihrer Benutzung zu installieren. Zielsetzung der in diesem Abschnitt vorgestellten Architektur ist es, die oben genannten Nachteile weitestgehend zu umgehen. Kosteneinsparungen können u. a. durch die Nutzung qualitativ hochwertiger frei verfügbarer Software (hier ist insbesondere die Software des World-Wide Webs zu nennen), die Anbindung an PC-Standardwerkzeuge wie Excel sowie eine gewisse Zentralisierung der Installation durch eine strikte Client-Server-Architektur erzielt werden. Die entwickelte Architektur ist modular und dienstebasiert. Sie ist nicht ausschließlich auf das WWW fixiert und berücksichtigt auch sich etablierende Industriestandards für offene verteilte Systeme.

In diesem Abschnitt wird die Architektur des UIS in zwei Schritten entwickelt. Zunächst wird die im Projekt WWW-UIS am Forschungszentrum Informatik (FZI), Karlsruhe, in Zusammenarbeit mit dem Institut für Photogrammetrie und Fernerkundung (IPF), Universität Karlsruhe, entwickelte und prototypisch realisierte Architektur vorgestellt. Zielsetzung dieser Architektur ist es, die Dienste des UIS in einer rein WWW-basierten Umgebung zur Verfügung zu stellen. Anzumerken ist, daß die Nutzung der WWW-Techniken und -Tools nicht die Nutzung des Internets impliziert. Vielmehr können diese Techniken auch in privaten Netzen, beispielsweise Verwaltungsnetzen, eingesetzt werden; vorausgesetzt wird lediglich die Verwendung von TCP/IP als Kommunikationsprotokoll. Darauf aufbauend wird im folgenden Abschnitt eine Architektur entwickelt, die durch Nutzung von CORBA (Common Object Request Broker Architecture der Object Management Group (OMG)) die einige der Defizite einer ausschließlich am WWW orientierten Architektur vermeidet.

2 Die Architektur von WWW-UIS

Abbildung 1 veranschaulicht die von FZI und IPF entwickelte und z.Zt. realisierte Architektur. World-Wide-Web-Clients (Browser wie beispielsweise Mosaic oder Netscape) greifen über einen WWW-Server auf die Dienste des WWW-UIS zu. Horizontal kann zwischen Anwender- und Systemdiensten unterschieden werden.

- *Systemdienste* implementieren die Basisfunktionalität wie beispielsweise den Zugriff auf Datenbanksysteme und die Aufbereitung von HTML-Seiten.
- *Anwenderdienste* repräsentieren das System gegenüber den Endanwendern. In einer WWW-Umgebung handelt es sich dabei um eine Folge von HTML-Seiten.

In ihrer einfachsten Form werden Anwenderdienste als eine weitestgehend statische Abfolge von Systemdiensten realisiert. Ein einfaches Beispiel für einen Anwenderdienst ist der Aufruf eines Datendienstes, der den Zugriff auf eine Datenbank realisiert, sowie die sich daran anschließende Aufbereitung des Ergebnisses durch einen Darstellungsdienst zu einer Darstellung als HTML-Seiten.

Vertikal können zunächst Daten- und Hilfsdienste unterschieden werden.

- *Datendienste* greifen auf unterschiedliche Datenquellen zu. Sie können weiter untergliedert werden in Basis- und Auskunftsdienste.
 - *Basisdienste* realisieren den Zugriff auf unterschiedliche Datenbestände, die die Anwender unmittelbar interessieren. Diese können sich in unterschiedlichen Datenbanksystemen befinden. Die in Abbildung 1 aufgeführten Datendienste beziehen sich auf die Datenbestände der übergreifenden Komponenten des UIS Baden-Württembergs. Diese liegen im relationalen Datenbanksystem Oracle, in ADABAS unter DEC/VMS sowie in einem Geoinformationssystem vor.

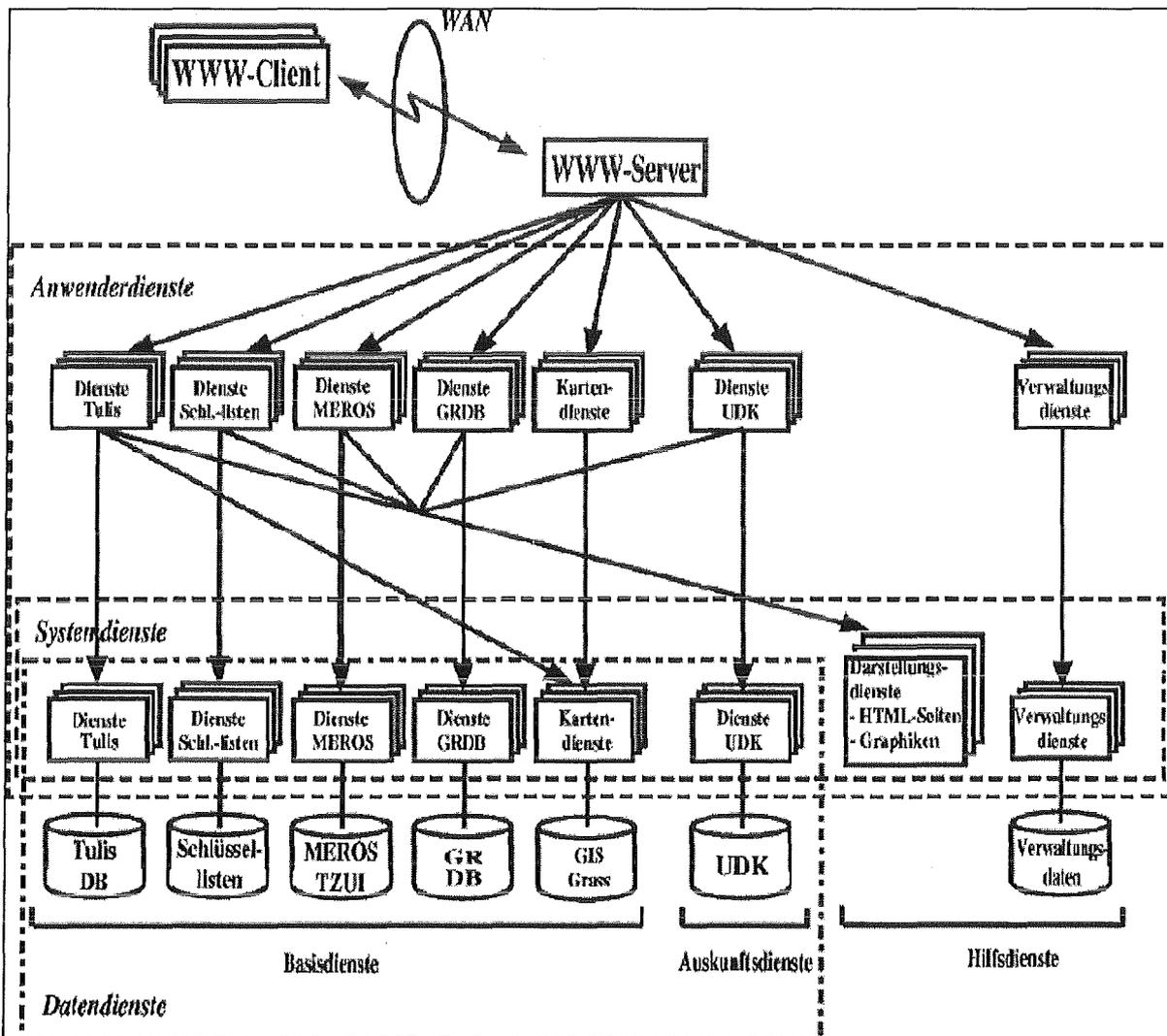


Abbildung 1: Architektur von WWW-UIS

- Auskunftsdienste basieren auf Metadatenbeständen, also vereinfacht auf „Daten über Daten“. Metadaten ihrerseits verweisen beispielsweise auf die datenhaltende bzw. auskunftsgabende Stelle, auf beliebige andere Anwenderdienste oder andere Dokumente wie Berichte.
Grundlage der WWW-UIS-Auskunftsdienste ist das Datenmodell des Umweltdatenkatalogs (UDK). Dieses Datenmodell wurde - voll UDK-kompatibel - so

erweitert, daß der Zugriff auf beliebige HTML-Dokumente, insb. die Dienststartseiten der Basisdienste auf der Ebene der Anwenderdienste und statische Berichtsseiten im World-Wide Web ermöglicht wird. Zur besseren Übersichtlichkeit werden statische HTML-Seiten in Abbildung 1 nicht dargestellt. Datenerfassung- und Pflege dieser erweiterten UDK-Datenbestände erfolgen zum einen mittels entsprechender WWW-Masken, zum anderen mittels des UDK-Anwendungsprogramms.

- Hilfsdienste greifen hingegen nicht auf Datenquellen mit Nutzdaten zu. Sie gliedern sich in Darstellung- und Verwaltungsdienste.
 - Die *Darstellungsdienste* bereiten die Ergebnisse von Datenbankanfragen zu HTML-Seiten auf. Ferner ermöglichen sie die graphische Aufbereitung der Ergebnisse von Datenbankanfragen.
 - Die *Verwaltungsdienste* unterstützen u.a. die Verwaltung von Dienstaufrufparametern und Dienstaufufrergebnissen. Insbesondere für Dienste, deren Aufruf sehr zeitintensiv sein kann (z.B. Aufrufe des MEROS-Dienstprogramms) oder deren Ergebnisse auf unterschiedliche Arten graphisch dargestellt werden können (z.B. Dienste, die auf die Oracle-Datenbestände des derzeitigen Systems TULIS - Dienste TULIS in Abbildung 1- zugreifen), ist eine Speicherung der Dienstaufrufparameter und -ergebnisse auf dem Rechnerknoten des WWW-Servers sinnvoll bzw. erforderlich.
- Abbildung 1 zeigt die für Anwender zugänglichen Dienste. Nicht in Abbildung 1 dargestellt sind die *Administrationsdienste* und -werkzeuge, die es dem Systemadministrator gestatten, die Installation von WWW-UIS zu verwalten. Hierzu zählen zum einen die bereits genannten Dienste zur Verwaltung von Dienstaufrufparametern und -ergebnissen, zum anderen eine separate Benutzerverwaltung (einschl. Paßwort-Schutzmechanismen).

Zur besseren Übersichtlichkeit in Abbildung 1 nicht gezeigt wird die clientseitige Anbindung von Werkzeugen wie z. B. PC-Standardtools wie EXCEL, aber auch Geoinformationssysteme wie ArcView. Anzumerken ist, daß die angestrebte Hardware- und Software-Unabhängigkeit bei der clientseitigen Nutzung von Werkzeugen nicht mehr sichergestellt ist. Ebenfalls nicht in Abbildung 1 dargestellt werden der Zugriff auf statische Berichte bzw. Berichtsteile sowie die Zugriffsmöglichkeit auf weitere Dienste wie beispielsweise die im Projekt Integral mittels WWW zur Verfügung gestellten UFIS-Selektoren. Diese können ebenso wie beliebige weitere Dienste, die als HTML-Seiten vorliegen, über die WWW-UIS-Auskunftsdienste zugänglich gemacht werden.

3 CORBA als Middleware im UIS

Das WWW und seine Werkzeuge bieten zahlreiche Vorteile wie beispielsweise die einfache und weitestgehend intuitive Benutzung. Bei der Realisierung entsprechender Dienste vermag es jedoch nicht alle Probleme vollständig zu lösen. So ist es beispielsweise nicht möglich, auf verteilte Datenquellen gleichzeitig zuzugreifen und Teilergebnisse inhaltlich zu verknüpfen. Dies ist beispielsweise dann erforderlich, wenn Datenkataloge verteilt vorliegen. Eine Integration technisch heterogener Komponenten wird ebenfalls nicht unmittelbar unterstützt. Ist beispielsweise ein WWW-Server unter Unix (DEC Unix) installiert, und soll auf die MEROS-

Datenbank unter ADABAS auf Vax/VMS zugegriffen werden, so ist dies nicht ohne weiteres möglich.

Um diese Defizite zu umgehen, wurde bei WWW-UIS eine Unterscheidung zwischen Anwenderdiensten und Systemdiensten vorgenommen. Diese Unterscheidung spiegelt sich auch in der Implementierung wieder. Sie ermöglicht es, die Systemdienste durch sog. Wrapper zu kapseln. Die Schnittstellen dieser Wrapper sind in der standardisierten Interface Definition Language (IDL) definiert. Hierdurch wird es möglich, die in WWW-UIS entwickelten und realisierten Systemdienste auch in CORBA-Umgebungen wiederverwenden zu können. Da es sich bei CORBA um eine Middleware handelt, also um eine mittlere Ebene der Architektur, können als graphische Benutzerschnittstelle wiederum das WWW bzw. seine Werkzeuge eingesetzt werden; prinzipiell sind aber auch andere Benutzeroberflächen möglich.

Abbildung 2 zeigt die entsprechende Architektur auf der Grundlage von WWW-UIS und CORBA.

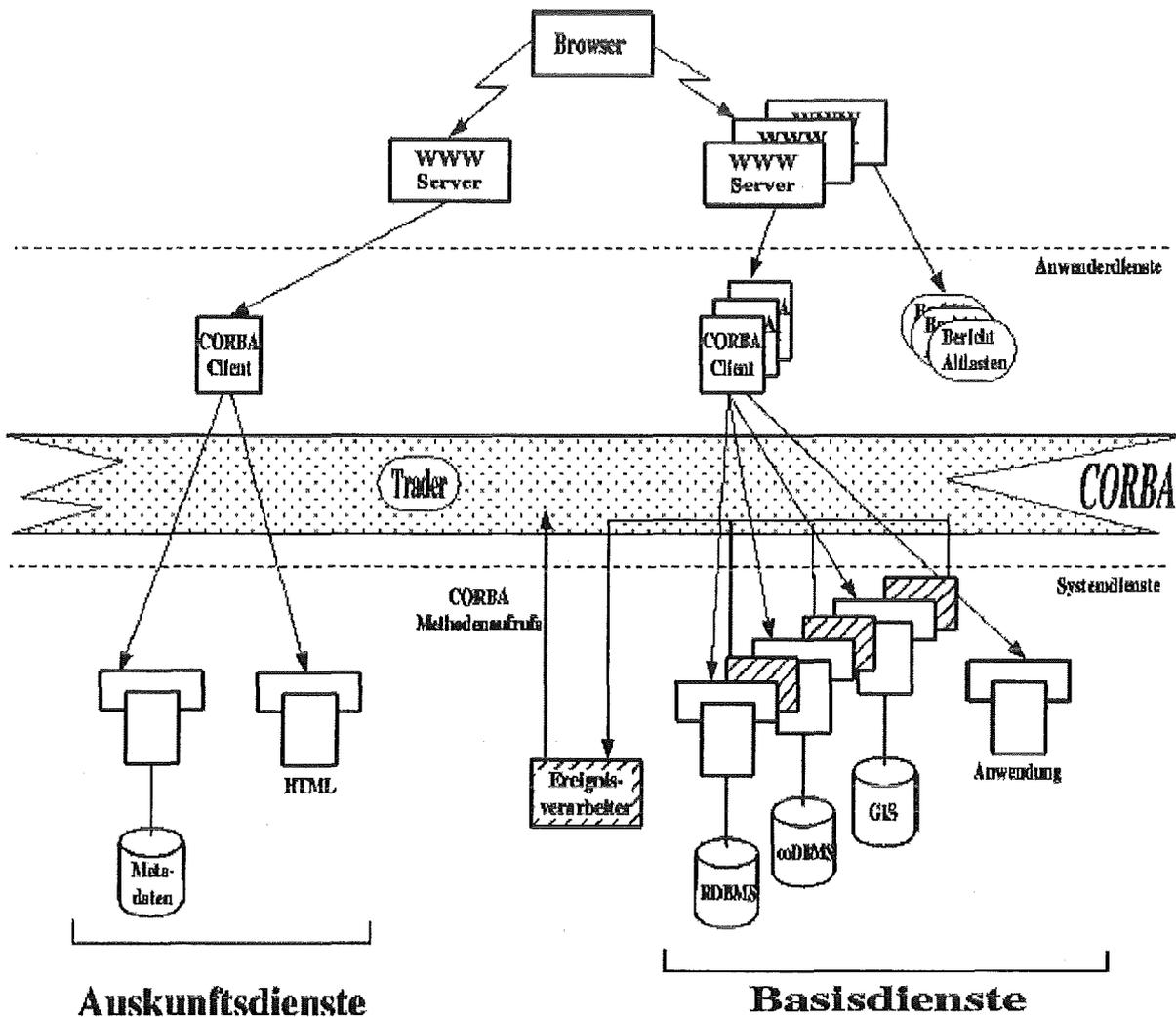


Abbildung 2: UIS-Architektur basierend auf WWW-UIS und CORBA

4 Zusammenfassung

Mit WWW-UIS wurde am FZI eine Architektur entwickelt, die über ein vollständig UDK-kompatibles Auskunftssystem den Zugriff auf beliebige Informationsquellen, insbesondere Dienste und Berichte, ermöglicht. Die prototypische Realisierung erfolgte so, daß in zukünftigen Ausbaustufen durch Einsatz von CORBA als Middleware einige der Defizite der aktuellen WWW-Umgebung umgangen werden können. Hierbei können die Systemdienste von WWW-UIS unmittelbar wiederverwendet werden. Dies wurde bereits an ersten Beispielen, auch im Rahmen der CORBA-Evaluierung des FZI für das UIS Baden-Württemberg, erfolgreich erprobt.

Literatur

- [KoKN95] Arne Koschel, Ralf Kramer, Ralf Nikolai. *WWW-UIS: Überblick und Einführung*. in [KrNi95]. Forschungszentrum Informatik (FZI). Karlsruhe. Oktober 1995.
- [KrNi95] Ralf Kramer, Ralf Nikolai. *WWW-UIS: Auskunfts-, Administrations- und Verwaltungsdienste*. FZI-Bericht 10/95. Forschungszentrum Informatik (FZI). Karlsruhe. Oktober 1995.
- [KrQu95] Ralf Kramer, Tom Quellenberg. *Global Access to Environmental Information*. In Environmental Software Systems; Proc. of the International Symposium on Environmental Software Systems, ISESS'95. S. 209 - 218. R. Denzer, D. Russel, G. Schimak (Hrsg.). Serie: International Federation for Information Processing (IFIP). Chapman and Hall. London. 1995.
- [KrSp95] Ralf Kramer, Horst Spandl. *Metadatenzugriff in Weitverkehrsnetzen: Eine Realisierung am Beispiel des Umweltdatenkatalogs UDK*. In Herausforderungen eines globalen Informationsverbundes für die Informatik; Gemeinsame Jahrestagung von GI und SI, GISI'95. S. 610-617. F. Huber -Wäschle, H. Schauer, P. Widmayer (Hrsg.). Serie: Informatik Aktuell. Springer. Zürich. September 1995.



CORBA - Evaluierung für das UIS Baden-Württemberg

*Arne Koschel, Ralf Kramer, Dietmar Theobald
Forschungszentrum Informatik (FZI)
Haid-und-Neu-Str. 10-14
D-76131 Karlsruhe*

*Unter Mitwirkung von:
Günter von Bültzingsloewen, Thomas Bleibel, Philipp Leibfried, Martin Schoel,
Manfred Walz, Christian Weinand*

1 EINLEITUNG	49
1.1 MOTIVATION	49
1.2 KURZEINFÜHRUNG - CORBA, DIE IDEE -	49
1.3 ÜBERBLICK.....	50
2 PRODUKTÜBERBLICK UND -VORAUSWAHL.....	51
2.1 VORGEHENSWEISE.....	51
2.2 UIS-SPEZIFISCHE KRITERIEN ZUR VORAUSWAHL.....	51
2.3 PRODUKTÜBERBLICK.....	53
2.4 PRODUKTVORAUSWAHL	55
2.5 WEITERES VORGEHEN	57
3 PRAKTISCHE ERPROBUNG CORBA-KERNSYSTEME.....	57
3.1 EINFÜHRUNG, ÜBERBLICK UND ANWENDUNGSBEISPIEL	57
3.2 OBJEKTMODELL.....	60
3.3 CORBA-KLIENT	62
3.4 CORBA-SERVER.....	65
3.5 REPOSITORIES.....	68
3.6 CORBA-SYSTEMKERN.....	70
3.7 CORBASERVICES UND CORBAFACILITIES.....	70
3.8 PRAKTISCHE ERFAHRUNGEN	72
3.8.1 <i>ObjectBroker</i>	72
3.8.1.1 Grundsätzliches.....	72
3.8.1.2 Installation.....	73
3.8.1.3 Dokumentation.....	73
3.8.1.4 Handhabung bei der Programmentwicklung.....	73
3.8.1.5 Bedienung	74
3.8.2 <i>Orbix</i>	74
3.8.2.1 Grundsätzliches.....	74
3.8.2.2 Installation.....	74
3.8.2.3 Dokumentation.....	75
3.8.2.4 Handhabung bei der Programmentwicklung.....	75
3.8.2.5 Bedienung	75
3.8.2.6 Support.....	75
3.9 RESÜMEE.....	75
4 PRAKTISCHE ERPROBUNG EINER FUNKTIONALITÄT FÜR DAS TRADING	76
4.1 EINFÜHRUNG TRADING.....	76
4.2 EINSATZBEISPIELE DES TRADING IM UIS BADEN-WÜRTTEMBERG.....	77
4.3 DER DAIS-TRADER.....	77
4.4 PRAKTISCHE ERPROBUNG TRADING	80
4.4.1 <i>Szenario 1: statisches Trading</i>	81
4.4.2 <i>Szenario 2: fehlertolerantes Trading</i>	82
4.4.3 <i>Szenario 3: dynamisches Trading</i>	82
4.5 RESÜMEE.....	83
5 LEISTUNGSMESSUNGEN.....	83
5.1 ÜBERBLICK.....	83
5.2 MESSUNGEN ELEMENTAROPERATIONEN	84
5.3 MESSUNGEN ANWENDUNGSSZENARIO UMWELTINFORMATIONSSYSTEM AUS WWW-UIS.....	86
5.4 RESÜMEE.....	88
6 ZUSAMMENFASSUNG UND AUSBLICK	89
6.1 WESENTLICHE ERGEBNISSE.....	89
6.2 EMPFEHLUNGEN FÜR DAS WEITERE VORGEHEN	89
LITERATUR.....	91

1 Einleitung

1.1 Motivation

Die übergreifenden Komponenten des Umweltinformationssystems (UIS) Baden-Württembergs bestehen z. Zt. noch aus einer Reihe weitestgehend monolithisch aufgebauter Applikationen. In mehreren Projekten wird die Funktionalität dieser Applikationen in einer diensteorientierten Architektur zur Verfügung gestellt. Eingesetzt werden die Techniken und Werkzeuge des World-Wide Webs. Das World-Wide Web löst jedoch nicht alle Probleme. So ist beispielsweise eine inhaltliche Verknüpfung von Diensten, die auf unterschiedlichen Knoten residieren, nicht vorgesehen; diese Dienste können im WWW nur unabhängig voneinander genutzt werden.

Die Common Object Request Broker Architecture (CORBA) der Object Management Group (OMG) etabliert sich momentan als Industriestandard für offene verteilte Systeme. Mitglied der OMG sind mehr als 500 Firmen und Institutionen, die CORBA-Implementierungen anbieten oder nutzen. Bei CORBA handelt es sich um eine sogenannte Middleware (also eine mittlere Architekturschicht), in die mittels einer syntaktisch einheitlichen Beschreibung (Interface Definition Language, IDL) die Schnittstellen zu beliebigen Software-Komponenten programmiersprachenunabhängig definiert und diese damit eingebunden werden können. Als Benutzeroberflächen kann darauf basierend beispielsweise auch das World-Wide Web eingesetzt werden.

Zielsetzung der in diesem Abschnitt geschilderten am FZI, Karlsruhe durchgeführten Arbeiten von GLOBUS II ist es, zu untersuchen, inwieweit CORBA-Implementierungen bereits heute in der aktuell in Entwicklung und Realisierung befindlichen Architektur der übergreifenden Komponenten des UIS Baden-Württembergs eingesetzt werden können. Zu diesem Zweck soll ein Marktüberblick über die derzeit verfügbaren Implementierungen des CORBA-Standards erstellt werden. Ferner sollen mehrere Implementierungen möglichst anhand realistischer und für das UIS relevanter Szenarien evaluiert werden. Auf dieser Grundlage soll eine Empfehlung für das weitere Vorgehen gegeben werden.

1.2 Kurzeinführung - CORBA, die Idee -

Die Kernidee von CORBA ist es, für verteilte Umgebungen eine syntaktisch einheitliche Schnittstellendefinition anzubieten, die es ermöglicht, die Schnittstellen sowohl zu bestehenden als auch zu künftig entstehenden Informationsquellen einheitlich zu beschreiben. Insbesondere wird durch CORBA ein über Herstellergrenzen hinweg einheitlicher Kern festgelegt, der auch schon (recht weitgehend) kommerziell verfügbar ist. Beschrieben werden hierbei in einer eigenen Interface Definition Language (IDL) Schnittstellen zu Objektimplementierungen (Server) (ähnlich zu C++ Klassendefinitionen, aber beschrieben werden ausschließlich die Schnittstellen), bestehend aus Daten (Instanzvariablen) und Methoden. Durch die Verwendung einer eigenen IDL wird eine Programmiersprachenunabhängigkeit erreicht, indem die

IDL durch entsprechende Pre-Compiler auf eine Programmiersprache, z.B. C, C++, Smalltalk u.a.m., abgebildet wird.

Soll nun durch Klienten auf derartig beschriebene Objekte (Instanzen) zugegriffen werden, so ist dies durch den Object Request Broker (ORB) von CORBA ortstransparent möglich, d.h. ein Klient muß nicht wissen, wo sich ein Objekt befindet. CORBA selbst ist plattformunabhängig definiert, d.h. prinzipiell kann z.B. unter SunOS auf Objekte aus einer OpenVMS oder gar MS-Windows Umgebung zugegriffen werden (und umgekehrt). Dort würden jeweils systemspezifische ORBs „laufen“. Gleiches gilt für die Netzwerkprotokolle zwischen den Systemen und für die Programmiersprachen der Objektimplementierungen. CORBA bietet also prinzipiell eine gute Möglichkeit, die technische Heterogenität der Systeme zu verdecken. Zur Flexibilität von CORBA trägt ferner bei, daß ein Zugriff auf Objekte sowohl statisch über die entsprechende IDL-Schnittstelle (IDL-Stubs) als auch dynamisch möglich ist. Für den dynamischen Zugriff existiert unterstützend ein Interface Repository, das alle verfügbaren Schnittstellenbeschreibungen von Objektimplementierungen enthält. Ferner ist eine Fehler- und Ausnahmebehandlung für Objektzugriffe definiert. Im Rahmen der CORBA 2.0 Spezifikation wurde ferner die Interoperabilität zwischen ORBs verschiedener Hersteller standardisiert. Wenn auch mit dem aktuellen Entwicklungsstand der ORB-Produkte (noch) keine Quellcodeportabilität erreicht wird, so wird bereits zum jetzigen Zeitpunkt Portierbarkeit und Stabilität von Systemdesign und -architektur über ORB-Produkte hinweg erreicht. Allein die durch den Normierungsdruck der OMG-Standardisierung vereinheitlichte Begriffsbildung bringt entscheidende Vorteile.

Zusammengefaßt noch einmal die wesentlichen Eigenschaften, die CORBA bereits heute (insbesondere als verfügbare Produkte) mindestens auszeichnen:

- Einheitliche Syntax von Schnittstellenbeschreibungen (IDL) zu Objektimplementierungen.
- Prinzipielle Unabhängigkeit von: Programmiersprachen, Betriebssystemen und Netzwerkprotokollen, also Verdeckung der technischen Heterogenität in verteilten Systemen.
- Ortstransparenter Zugriff auf verteilte Objekte.
- Statischer und dynamischer Zugriff auf Objekte.
- Beschreibung aller vorhandenen Objektimplementierungen im Interface Repository.
- Kleine und klar abgegrenzte Entwicklungseinheiten. Jeder ist im wesentlichen für „seine“ Objektimplementierungen verantwortlich.
- Systemübergreifenden einheitliche Begriffsbildungen und Portabilität von Systemdesign und Systemarchitektur über verschiedene ORB-Produkte hinweg.

1.3 Überblick

Eine erste Recherche des FZI ergab, daß aktuell ca. 15 CORBA-Implementierungen am Markt verfügbar sind. Aufgrund des vorgegebenen Kosten- und Zeitrahmens macht diese Anzahl eine Vorauswahl für die praktische Evaluierung erforderlich. Im folgenden Abschnitt 2 wird ein Überblick über CORBA-Implementierungen gegeben. Eine Vorauswahl fand zum einen anhand UIS-spezifischer harter Kriterien, zum anderen aufgrund der in einer dienstebasierten UIS-Architektur sinnvoll einsetzbaren Zusatzfunktionalität statt. Basierend auf dieser Vor-

auswahl befaßt sich Abschnitt 3 mit der Evaluierung von CORBA-Kernsystemimplementierungen. Eine für die dienstbasierte Architektur der übergreifenden Komponenten des UIS besonders relevante Funktionalität ist das Trading, d. h. der Vermittlung von Diensten. In Abschnitt 4 wird die Einsetzbarkeit des Trading-Mechanismus im UIS anhand einiger typischer Szenarien untersucht. Neben der angebotenen Funktionalität ist auch die Leistung der Implementierungen für den praktischen Einsatz relevant. In Abschnitt 5 werden daher (grobe) Leistungsmessungen zum einen von Elementaroperationen, zum anderen von Anwendungsszenarien aus dem UIS vorgestellt. Die wesentlichen Ergebnisse und praktischen Erfahrungen werden im abschließenden Abschnitt 6 zusammengefaßt. Ferner wird eine Empfehlung für das weitere Vorgehen gegeben.

2 Produktüberblick und -vorauswahl

2.1 Vorgehensweise

Als Vorgehen zur zügigen Auswahl detailliert zu evaluierender CORBA-Implementierungen wurde beschlossen, zunächst eine Reihe wichtiger UIS-Kriterien zu finden, um die eigentliche Evaluierung von ca. 20 CORBA-Implementierungen im vorgegebenen Zeit- und Kostenrahmen deutlich einschränken zu können. Eine Reihe dieser Kriterien und deren Gewichtung wurden vom FZI vorgeschlagen und in Abstimmung mit den weiteren Projektbeteiligten sowie dem Auftraggeber festgelegt. Ferner wurde beschlossen, anhand „harter“ Kriterien, die insbesondere die Produktverfügbarkeit für bestimmte Plattformen betreffen, zwei Produkte auszuwählen und ein weiteres, das eine möglichst große bzw. besonders UIS-relevante Funktionalität enthalten soll. Letzteres soll einen Überblick über bereits vorhandene Möglichkeiten von CORBA liefern.

Zunächst wurden Produktinformationen durch die Hersteller und Distributoren von CORBA-Produkten, aber auch durch Quellen wie das World-Wide Web eingeholt. Hierbei stellte das FZI fest, daß einige der Hersteller nicht mehr existierten bzw. sie ihre CORBA-Implementierung nicht mehr unterstützen. Andere Produkte stellten keine echten CORBA-Implementierungen dar. Ferner lieferten einige Hersteller trotz mehrfacher Aufforderung bis zum festgelegten Ausschlußtermin kein Informationsmaterial. Anhand dieser Informationen wurde gemäß der UIS-spezifischen Kriterien mit höchster Gewichtung (siehe Abschnitt 2.2) eine negative Vorauswahl getroffen. Die verbleibenden Produkte wurden noch einmal tabellarisch verglichen, wobei UIS-spezifisch niedriger gewichtete Kriterien einfließen (siehe Abschnitt 2.3). Hieraus wurden die endgültig zu evaluierenden Produkte ausgewählt. Zu beachten ist, daß unsere Auswahl anhand der im August 1995 vorliegenden Produktstände und Materialien erfolgte. Die Entwicklung von CORBA-Implementierungen ist jedoch gerade wieder sehr stark im Fluß, bedingt durch neu verabschiedete Standards der OMG.

2.2 UIS-spezifische Kriterien zur Vorauswahl

Als für das UIS relevante Kriterien wurden festgelegt:

1. Unterstützung möglichst vieler System-Plattformen, insbesondere aber: Betriebssysteme
 - 1a. A - Priorität
DEC-Unix (OSF/1), OpenVMS/Axp & OpenVMS/Vax, Windows NT
 - 1b. A - Priorität
Windows 3.* / 95
oder
COM/OLE Integration
(bisher nur als Zusatzfunktionalität, bald im ORB-Kern)
 - 1c. C - Priorität
SunOS/Solaris
2. Netzprotokolle
 - 2a. A - Priorität
TCP/IP
 - 2b. A - Priorität
DecNet
3. CORBA ORB-Kern
A - Priorität
Implementierung aller Komponenten des ORB - Kerns gemäß der CORBA 1.2 oder gar der CORBA 2.0 Spezifikation.
4. Hersteller (generell B - Priorität)
 - 4a. Absehbare künftige Bedeutung / Größe
 - 4b. Kooperationsbereitschaft
 - 4c. Support in Deutschland
 - 4d. Abhängigkeit von bestimmter Hardware
 - 4e. Referenzkunden
5. Interface Repository
B - Priorität
Möglichst gutes Interface Repository (Versionsverwaltung, Abgleichmechanismen, Anpaßbarkeit)
6. CORBAServices
B - Priorität
CORBAServices, besonders als Grundlage für Entwicklung eines einfachen Traders, also LifeCycle, Naming, Properties, (Events)
7. Zusatzfunktionalität
 - 7a. B - Priorität
Spezielle Security- oder alternativ Filtermechanismen zur Unterstützung kontrollierter Dienstzugriffe
 - 7b. B - Priorität
Zusatzfunktionalität für Trading (gemäß Dienste-AG)
 - 7c. C - Priorität
Allgemein zusätzliche Schnittstellen bzw. Bibliotheken zur Integration weiterer Softwarekomponenten (Datenbanksysteme, allgemeine Executables, ...)

2.3 Produktüberblick

Dieser Abschnitt faßt Angaben über Produkte der dem FZI bekannten Hersteller von CORBA-Implementierungen tabellarisch zusammen. In der Spalte Status von Tabelle 0-1 ist festgehalten, wenn nur ungenügende Informationen zu erhalten waren oder der Hersteller nicht mehr existiert oder auch das Produkt nicht mehr unterstützt.

	Hersteller	Produkt	Status
1	AT&T / NCR	NCR Cooperative Frameworks	Produkt im deutschen Vertrieb nicht/nicht mehr bekannt. Anfragen in Amerika lieferten kein Antwort. Aussortiert. Keine Kurzbeschreibung.
2	Digital (DEC)	ObjectBroker	Im 2. Auswahlschritt wegen Verfügbarkeit für UIS-Plattformen. Kurzbeschreibung in [FZI95].
3	Expersoft	XShell / PowerBroker	Im 2. Auswahlschritt wegen großer Funktionalität. Kurzbeschreibung in [FZI95].
4	HP	DST Distributed Smalltalk, ORB Plus	Telefonisch durch den deutschen Vertrieb versprochenes Material wurde nicht geliefert. Anfragen in Amerika lieferten keine Antwort. Aussortiert. Keine Kurzbeschreibung.
5	IBM	DSOM Distr. System Object Model	Nicht im 2. Auswahlschritt Kurzbeschreibung in [FZI95].
6	ICL	DAIS Distributed Application Integration System	Im 2. Auswahlschritt wegen Verfügbarkeit für UIS-Plattformen und Funktionalität. Kurzbeschreibung in [FZI95].
7	IONA	Orbix	Im 2. Auswahlschritt wegen Verfügbarkeit für UIS-Plattformen und Funktionalität Kurzbeschreibung in [FZI95].
8	(Isis)	RDO Reliable Distributed Objects; gemeinsam mit IONA	Direkte Kooperation mit IONA (Orbix) für zusätzliche Fehlertoleranz und Management verteilter Umgebungen. Keine eigene Beschreibung.
9	Lohara Software Systems	ODDESYY++, CSEM, APPBLDR VEIS, EPMS	Nur Entwicklungsumgebung mit CORBA Anbindung. Kein eigener ORB. Aussortiert. Keine Kurzbeschreibung.
10	NEC	NEC-ORB, RTTS	Produkt im deutschen Vertrieb nicht mehr bekannt. Anfragen in Amerika lieferten kein Antwort. Aussortiert. Keine Kurzbeschreibung.
11	NetSmiths	DataEngine, ObjectKit NetManager, CORBAkit	out of business
12	Next	Portable Distributed Objects	Nur CORBA-konforme Business Objects. Kein eigener ORB. Aussortiert. Keine Kurzbeschreibung.
13	Oberon Software	SynchroWorks	Nur ORB-Aufsatz zur Entwicklung von "Business Applications". Kein eigener ORB. Aussortiert. Keine Kurzbeschreibung.
14	Object Oriented Technologies	Distributed Object Management Environment (D.O.M.E)	Im 2. Auswahlschritt wegen Verfügbarkeit für UIS-Plattformen. Kurzbeschreibung in [FZI95].
15	PostModern Computing	ORBeline	Im 2. Auswahlschritt wegen geplanter Verfügbarkeit für UIS-Plattformen und Funktionalität. Kurzbeschreibung in [FZI95].

16	Prism Technologies	OpenBase Software Integration Platform (SIP) Applications Development Environment (ADE)	Nicht im 2. Auswahlschritt Kurzbeschreibung in [FZI95].
17	Rölsch Consult.	RC - ORB	Ein ORB lediglich für Windows 3.1. Aussortiert. Keine Kurzbeschreibung.
18	Software AG	Entire BROKER	Ein herstellerspezifischer Object Broker, der erst zukünftig (keine Zeitangaben) CORBA-konform werden soll. Aussortiert. Keine Kurzbeschreibung.
19	SunSoft	Distributed Objects Everywhere (DOE)	Im 2. Auswahlschritt wegen Funktionalität Kurzbeschreibung in [FZI95].
20	Tivoli Systems	Management Platform (TMP) Advanced Development Environment (ADE) Application Extension Facility (AEF)	Allgemeine Werkzeuge für Management verteilter Umgebungen. Ein eigenständig nutzbare ORB ist zwar vorhanden, jedoch ist herstellerseitig diese Nutzung - nach Aussage des deutschen Vertriebs - eigentlich nicht vorgesehen und unterstützt. Aussortiert. Keine Kurzbeschreibung.

Tabelle 0-1: Überblick CORBA-Implementierungen

Anhand der Kriterien aus Abschnitt 2.2 und des uns vorliegenden Materials (siehe Tabelle 0-1) wurde zunächst eine negative Vorauswahl getroffen, d.h. für das UIS nicht sinnvolle Systeme wurden aussortiert. Gründe für das Aussortieren waren:

- Ein Produkt stellt keine wirkliche CORBA-Implementierung dar, sondern nur eine CORBA konforme Zusatzbibliothek oder einen CORBA-Aufsatz, bzw. will CORBA erst künftig unterstützen.
Dies gilt für die 5 Produkte von Lohara, Next, Oberon, Software AG und Tivoli.
- Trotz mehrfacher Aufforderung (schriftlich und telefonisch) wurde herstellerseitig versprochenes Material nicht geliefert.
Dies gilt für die 3 Produkte von AT&T/NCR (Cooperative Frameworks), HP (ORBplus) und NEC (NEC-ORB).
Anm.: Uns vorliegende Produktkurzbeschreibungen der OMG [OMG94a], [OMG95c] lassen im Vergleich keines der Produkte für das UIS relevant erscheinen.
- Der Hersteller ist „out of business“.
Dies gilt für NetSmiths.
- Ein Produkt ist weder für alle UIS-relevanten Plattformen verfügbar, noch bietet es im Vergleich besonders hohe Funktionalität.
Dies gilt für die 3 Produkte von IBM (DSOM), Prism (OpenBase) und Rölsch (RC-ORB).
- Ein weiteres Produkt von Isis wurde nicht aussortiert, sondern direkt mit dem Produkt von Iona (Orbix) zusammengefaßt, da die beiden Hersteller ein gemeinsames Produktpaket anbieten.

2.4 Produktvorauswahl

Somit blieben für die Endauswahl der zu evaluierenden Systeme noch 7 Produkte übrig. Diese sind in der folgenden Tabelle nach weiteren Kriterien, insbesondere denen aus Abschnitt 2.2 gegenübergestellt.

	Digital Object-Broker	Expersoft XShell	ICL DAIS	IONA Orbix	Obj.Or.-Technol. DOME	PostModern Computing ORBeline	SunSoft DOE
Plattformen							
- Betriebssysteme	Alle UIS-relevanten	nicht VMS	Alle UIS-relevanten	nicht OpenVMS / Vax	Alle UIS-relevanten	VMS und OpenVMS geplant	nur Solaris
- Netzprotokolle	Alle relev.	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP
CORBA 1.2 Vollständigkeit	ja	ja	nein DII fehlt	ja	nein DII fehlt	ja	ja
CORBA 2.0 Ankündigung	geplant	Ende '95	geplant	Sept. '95	Ende '95	August '95	Ende '95 ?
CORBA services (a)							
- Concurrency	-	-	-	-	-	angekünd.	zukünftig
- Events (T)	-	Ende '95	Ende '95	-	Ende '95	4. Q. 95	zukünftig
- Externalization	-	Ende '95	-	-	-	angekünd.	zukünftig
- Lifecycle (T)	-	ja	-	-	Ende '95	4. Q. 95	ja
- Naming (T)	-	Ende '95	Ende '95	-	-	3. Q. 95	ja
- Persistence	-	ja	-	mit Isis	- (eigene)	- (eigene)	ja
- Relationship	-	-	-	-	-	angekünd.	ja
- Transactions	-	-	Ende '95	mit Isis	Ende '95	angekünd.	zukünftig
Zusatzfunktionen							
- COM/OLE	ja	ja	-	ja	Ende '95	-	ja, via Orbix
- Filter / Security	Security	-	Sec. angek.	Filter	-	Filter	Filt. einfach
- Trading	minimal	-	ja	-	-	-	-
- Ankopplung an:	Skripts		-	ooDBMS	-	-	RDBMS
- Weitere		viele	-	Fehlertol.	allg. Bib.	-	OpenStep
Hersteller							
- Vertrieb: Deutsch	ja	-	bald	ja	-	-	ja
- Kooperation FZI	befriedig.	befriedig.	befriedig. (b)	sehr gut (c)	befriedig.	(sehr) gut	ausreichend
- Koop. andere Hersteller	Microsoft	Rogü Wave		Sun, Isis Obj.Des.		angekündigt	Next, Orbix kompatibel
Bemerkungen					Klassenbibliothek	frei für Forschung	nur Beta-Version

Tabelle 0-2: Gegenüberstellung der CORBA-Implementierungen der Vorauswahl

In Tabelle 0-2 bedeuten:

- (a) T relevant für einfachen Trader;
- (b) Deutscher Service schlecht, englischer Service gut;
- (c) Deutscher Distributor.

Werden alle Punkte aus Tabelle 0-1 gegenübergestellt, so zeigt sich, daß die „harten“ Plattformkriterien nur 3 Produkte (ObjectBroker, DAIS, DOME) und zwei weitere mit Einschränkungen bzw. Ankündigungen (Orbix: Nur OpenVMS/Axp), ORBeline (VMS und OpenVMS nur geplant) erfüllen. Insbesondere erfüllt nur der ObjectBroker das Netzwerkprotokollkriteri-

um DecNet. Allerdings soll der Parallelbetrieb von TCP/IP und DecNet problemlos sein, so daß hier TCP/IP ausreichend scheint.

Betrachtet man nun die weiteren Punkte, so kann DOME zum jetzigen Zeitpunkt nicht einmal (einigermaßen) vollständige CORBA 1.2 Konformität bieten, da essentielle Elemente wie das Dynamic Invocation Interface und das Interface Repository fehlen. Auch ist keine besonders herausragende Funktionalität vorhanden. Erst mit der für Ende '95 angekündigten CORBA 2.0 Unterstützung sollen die fehlenden Elemente vorhanden sein. Dann könnte dieses Produkt wieder interessanter werden.

Dem Produkt DAIS fehlen, ähnlich DOME, derzeit Dynamic Invocation Interface und Interface Repository. Es ist also nicht CORBA 1.2 konform. Jedoch bietet es als Besonderheit einen flexiblen Trader-Mechanismus, der sehr interessant scheint.

Die Auswahl zwischen Orbix und ORBeline ist relativ schwierig, da sich die Produkte zum jetzigen Zeitpunkt sehr ähneln. Beide Implementierungen gelten als stabil und (Orbix nur mit ISIS-Zusätzen) fehlertolerant.

Für ORBeline spricht, daß mehrere der CORBAServices schneller als bei Orbix vorhanden sein werden. Interessant sind auch die frei verfügbaren Forschungslizenzen. OpenVMS und VMS Unterstützung ist zumindest geplant.

Auf der anderen Seite gilt IONA mit Orbix derzeit (neben Expersoft) nicht ohne Grund als Marktführer in ORB-Implementierungen. Es besteht eine enge Kooperation mit SunSoft - SunSoft besitzt eine größere Beteiligung an IONA - und anderen Firmen wie ObjectDesign (Hersteller des ooDBMS ObjectStore) und ISIS (bekannt für Netzwerkprodukte mit sehr hoher Ausfallsicherheit). Des weiteren zeigte sich der deutsche Distributor (Interactive Objects) von IONA gegenüber dem FZI als ausgesprochen kompetent und hilfsbereit.

Die noch nicht genannten beiden Produkte DOE und XShell (jetzt PowerBroker) sind besonders wegen ihrer Funktionalität interessant. So bietet DOE viele CORBAServices und die Anbindung an die Next-Umgebung OpenStep/PDO, ist und wird allerdings rein auf SUN/Solaris Plattformen beschränkt sein.

XShell bietet interessante Zusatzfunktionalität, wie Regelverarbeitung, Fuzzy-Objekte, Administrationswerkzeuge und eine Reihe OMG ähnlicher Services. Ferner sollen eine Reihe der CORBAServices Ende'95 vorhanden sein.

Als Fazit der Endauswahl wurden daher zur gründlichen Evaluierung gemäß der harten Kriterien zunächst das Produkt ObjectBroker gewählt, da es alle Plattformkriterien erfüllt. DOME scheidet aus, da es derzeit nicht CORBA 1.2 konform ist und keine besondere Zusatzfunktionalität bietet. Zwischen Orbix und ORBeline haben wir uns für Orbix entschieden, besonders wegen der vorhandenen Unterstützung für OpenVMS/Axp und der sehr guten Distributorkooperation. Als besonders interessante Zusatzfunktionalität wurde, gerade auch im Hinblick der Diskussionen der Dienste-AG und im INSERV 2.3 Dokument des IKE [TS94], vielfach ein Trader-Mechanismus genannt. Anhand des Softwarepaketes DAIS, welches eine Traderfunktion enthält, wird der Trading-Mechanismus genauer untersucht.

2.5 Weiteres Vorgehen

Im folgenden Abschnitt 3 werden die beiden Systeme, die aufgrund der Erfüllung der harten UIS-spezifischen Kriterien gewählt wurden, also ObjectBroker und Orbix, hinsichtlich des CORBA-Kernsystems evaluiert. Abschnitt 4 umfaßt die Evaluierung der Trading-Funktionalität, anhand des im DAIS-System enthaltenen Traders. Leistungsmessungen sind Gegenstand von Abschnitt 5.

3 Praktische Erprobung CORBA-Kernsysteme

3.1 Einführung, Überblick und Anwendungsbeispiel

Gegenstand dieses Abschnitts ist der *Object Request Broker* (ORB) selbst, in seinem Aufbau und in seiner Einordnung in die Standardisierungslandschaft der OMG, sowie praktische Erfahrungen bei der Nutzung der ausgewählten Systeme ObjectBroker und Orbix. Der Aufbau des Abschnitts orientiert sich an der Sicht eines Nutzers: ausgehend vom konzeptuellen Modell entlang des „Wegs“ eines Aufrufs durch einen ORB und abschließend Aufbau und Administration der ORB-Systeme. Ein vereinfachtes Anwendungsbeispiel dient dabei zur Illustration und praktischen Erprobung der vorgestellten Konzepte.

Die Rolle eines ORB erklärt sich aus der den OMG-Standards zugrundeliegenden konzeptuellen Infrastruktur, der sogenannten *Object Management Architecture* (OMA) [OMG90]: Diese umfaßt zunächst ein allgemeines Objektmodell und des weiteren eine Referenzarchitektur, das einen ORB als zentrale Komponente einer verteilten, heterogenen Umgebung vorsieht (siehe Abbildung 1). Der ORB sichert dabei die Interoperabilität von Objekten über ein Netzwerk heterogener Systeme durch die Realisierung des grundlegenden Interaktionsmechanismus in diesen Systemen: *Objektaufrufe*, genauer Aufrufe von Operationen für Objekte (*Request*).

Es lassen sich somit zwei Rollen unterscheiden: die des Aufrufers (*Klient*) sowie die des aufgerufenen Objekts, das einen *Dienst* (*Service*) für den Aufrufer erbringt. Das aufgerufene Objekt wird daher häufig auch als *Server* bezeichnet. Objekte nehmen dabei dynamisch häufig beide Rollen ein; ein dienstgebendes Objekt kann so zur Erfüllung einer Anforderung auf die Dienste anderer Objekte zurückgreifen, also selbst zum Klient werden. Beide - Klient und Dienstleister - sind dabei über wohldefinierte Schnittstellen isoliert.

Zentrales Ziel der OMG ist wiederverwendbare, portable und interoperable objektbasierte Software für verteilte, heterogene Umgebungen [OMG95a; §0]. Der durch die Richtlinien und Spezifikationen der OMG vorgegebene Rahmen soll auf den wichtigsten Hardware- und Softwareplattformen kommerziell verfügbar sein. Dies bildet dann die Basis für den übergreifenden Einsatz von Objekttechnologie und erlaubt, die Komplexität der Anwendungsentwicklung für verteilte, heterogene Systeme zu beherrschen.

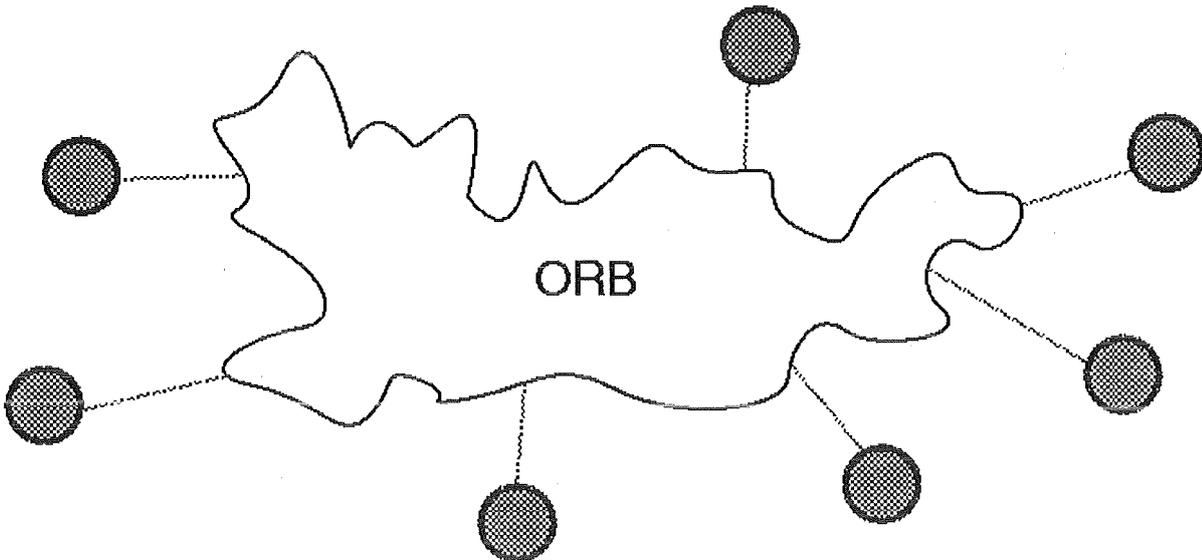


Abbildung 1: ORB als Kommunikationskern

Zentrales Ziel der OMA ist die Standardisierung einer *Common Object Reference Architecture (CORBA)* [OMG95a]. Hierbei handelt es sich zum einen um die Konkretisierung des OMA- Objektmodells wie zum anderen auch um eine ORB-Rahmenarchitektur. Das CORBA-Objektmodell gestattet dabei die Definition von Objektschnittstellen, die Rahmenarchitektur legt Schnittstellen und Verhalten eines ORB so weit fest, daß zum einen Klienten und Dienste implementiert werden können, aber dennoch Freiheit für ein breites Spektrum an CORBA-Implementierungen unterschiedlichster Qualitäts-/Leistungseigenschaften verbleibt.

Abbildung 2 zeigt die wesentlichen Elemente der Architektur eines Object Request Brokers nach CORBA [OMG95a]. Sichtbar sind unterschiedliche Schnittstellen, über die ein Klient / Server mit dem ORB-Kern interagiert. Hinzu kommen Repositories, in denen Informationen über das durch den ORB verwaltete Objektsystem liegen. Diese Architekturelemente werden in den folgenden Abschnitten vorgestellt.

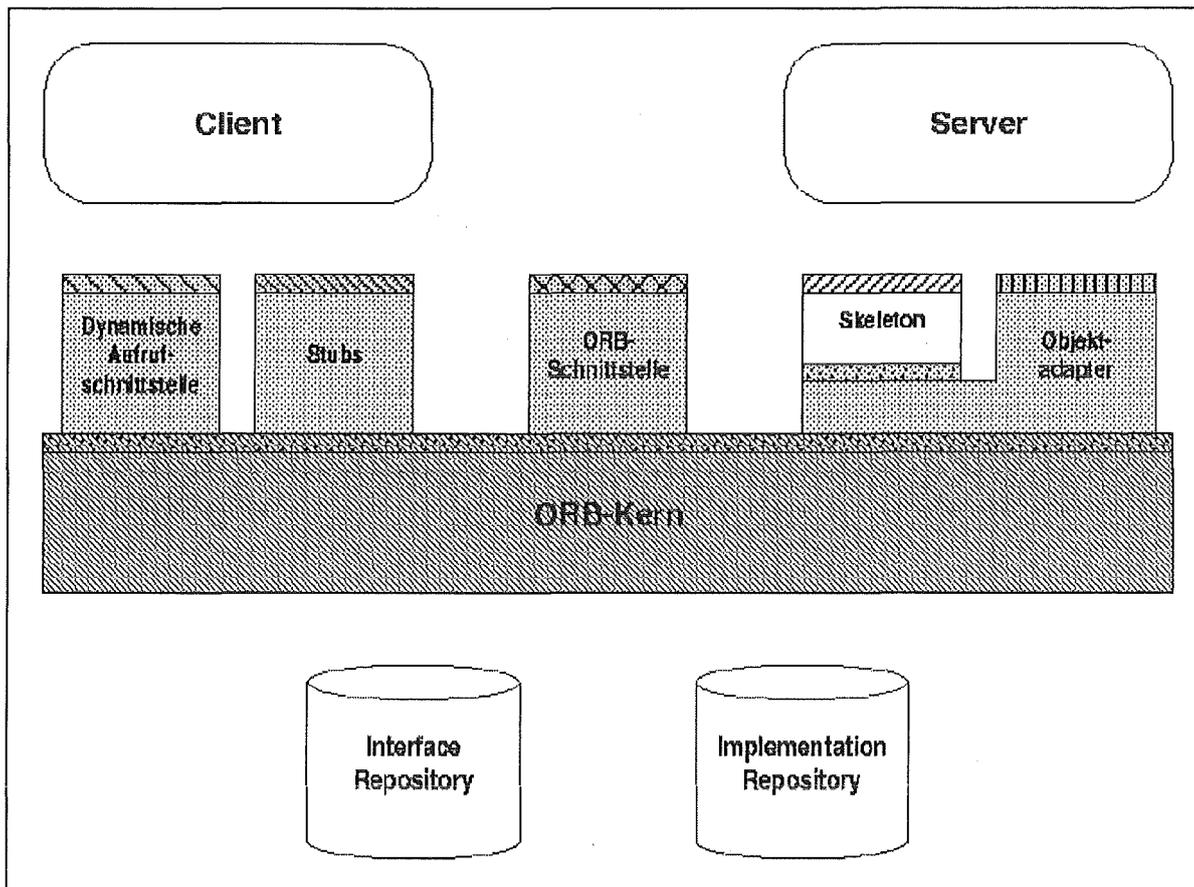


Abbildung 2: CORBA - Architekturschema

Als Anwendungsbeispiel zur Illustration und praktischen Erprobung der betrachteten Konzepte nutzen wir ein vereinfachtes Modell eines verteilten Systems zur Verwaltung und Auswertung von Meßdaten:

- Meßwerte werden in einer Datenbank abgelegt, die als Dienstobjekt im Netz zugänglich ist. Über Suchspezifikationen kann auf die interessierenden Daten zugegriffen werden.
- Teil der Auswertung von Meßdaten sind Berechnungen, die auf dedizierten Rechnern durchgeführt werden, welche die erforderliche Leistungskapazität aufweisen.
- Eine Anwendung, entnimmt der Datenbank Meßwerte, führt auf diesen einige Berechnungen durch und liefert als Ergebnis einen Bericht.

Abbildung 3 skizziert obiges Szenario in seiner Abbildung auf ein verteiltes Objektsystem: Sowohl Datenbank wie auch Berechnungsfunktionen sind als Dienste für die Anwendung zugreifbar, welche die Rolle eines Klienten hat.

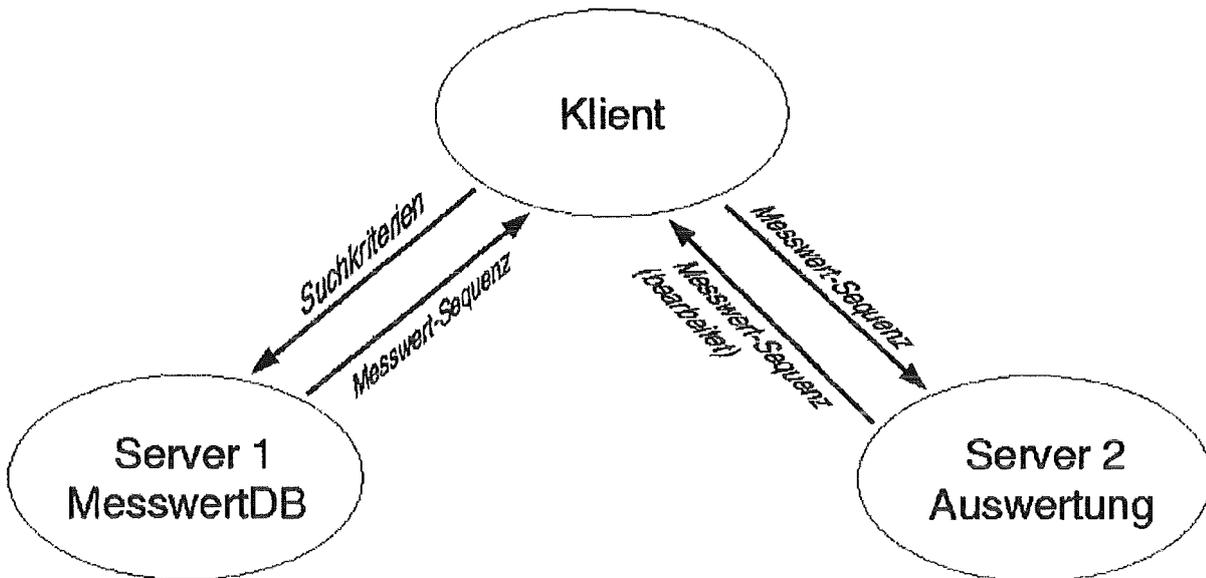


Abbildung 3: Anwendungsszenario

3.2 Objektmodell

Das CORBA-Objektmodell umfaßt Regeln/Vorgaben für die Objekte, die in einer CORBA-Umgebung auftreten können, sowie insbesondere auch eine konkrete Sprache, in der die extern beobachtbaren Eigenschaften der Objekte in standardisierter Form beschrieben werden können: die CORBA - *Interface Definition Language*, kurz *IDL* [OMG95a; §3]. IDL leistet dies durch Deklaration von Typen und hierauf anwendbaren Operationen. Beschrieben werden dabei die Schnittstellen der Typen und Operationen, d.h. die „Syntax“ der Objektinteraktionen - nicht deren Semantik, wie dies etwa mit Vor-/Nachbedingungen möglich wäre.

Das CORBA-Objektmodell kennt zwei Arten an Werten: Objektbezeichner, genannt *Objektreferenzen* oder auch *OID*, und „Nicht-Objekte“. Hierzu korrespondierend wird zwischen Objekttypen und Typen für Nicht-Objekte unterschieden. Abbildung 4 gibt einen Überblick über die im Objektmodell auftretenden Werte bzw. Typen.¹

¹ Abbildung 4 weicht zur besseren Verständlichkeit von der typischen Darstellung der Wertehierarchie ab, wie etwa in [OMG95a] abgebildet.

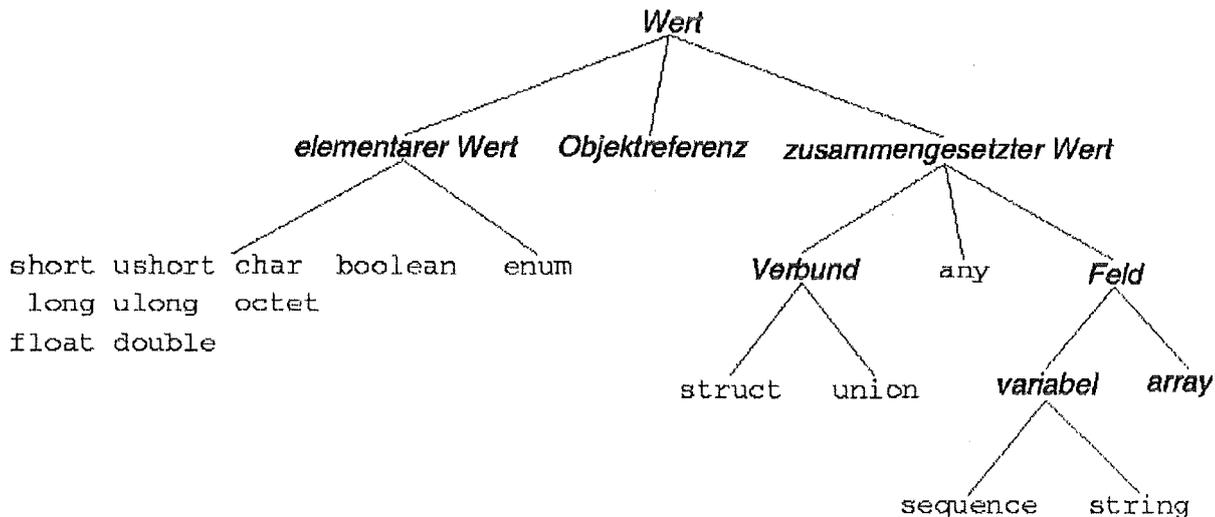


Abbildung 4: Taxonomie CORBA-Werte

Elementare Typen umfassen die wichtigsten skalaren Typen einer typischen Programmiersprache: Ganz- und Gleitkommazahlen, Zeichen, Byte, boolesche Werte und anwendungsdefinierte Aufzählungen. Zusammengesetzte Typen sind Verbund und varianter Verbund, mehrdimensionale Felder fester Länge und eindimensionale Felder mit fester Obergrenze aber variabler aktueller Länge. Schließlich kann ein Wert des Typs any alle sonstigen Werte enthalten.

Objekte haben eine über ihre Lebensdauer eindeutige Identität und werden über Objektreferenzen repräsentiert und manipuliert. Es kann dabei mehrere unterschiedliche Referenzen geben, die auf dasselbe Objekt verweisen. Bei Aufruf einer Operation auf einem Objekt wird auf Serverseite eine passende Implementierung (*Methode*) zugeordnet und ausgeführt. Je Operationsaufruf wird dabei durch den ORB maximal eine Methode aktiviert (*at-most-once*). Operationsaufrufe können sowohl synchron oder auch asynchron sein. Der erste Fall ähnelt einem Prozeduraufruf, d.h. der Aufrufer wartet auf das Operationsende und ist danach sicher, ob die Operation ausgeführt wurde (*exactly-once*), oder ob ein Fehler auftrat. Im asynchronen Fall kann dies i.a. aufgrund der fehlenden Rückmeldung nicht entschieden werden (*best-effort*).

Objektypen stehen in Vererbungsbeziehungen ausgehend vom allgemeinsten Typ `Object`, Mehrfachvererbung wird unterstützt. Die Deklaration umfaßt Attribute und Operationen, d.h. Zustand und Verhalten von Objekten. Eine Attributdeklaration ist dabei im wesentlichen äquivalent zur Deklaration zweier Operationen zum Lesen und Setzen des Attributwertes. Operationsdeklarationen umfassen zunächst Name und Typ des Ergebnisses und der Parameter, sowie analog C++ [ES90] die in der Operation möglicherweise auftretenden Ausnahmen. Hinzu kommen weitere Informationen, die aus der Anwendung für verteilte Umgebungen motiviert sind:

- Parameter sind entweder reine Eingabe- (*in*) oder reine Ausgabeparameter (*out*), oder Ein- & Ausgabeparameter (*inout*).
- Eine optionale Kontextdeklaration gibt an, welcher Teil des Kontexts eines Aufrufers beim Aufruf mitübergibt wird. Analogon für diesen Kontextbegriff sind die von Unix her bekannten Environmentvariablen und ihre implizite Weitergabe an aufgerufene Subprozesse.

- Wie oben beschrieben kann eine Operation sowohl synchron wie auch asynchron aufgerufen werden. Asynchrone Operationen werden als solche deklariert und unterliegen einigen Einschränkungen: so gibt es dann etwa keine Ausgabeparameter.

Neben Typen kann eine IDL-Spezifikation noch weitere Deklarationen enthalten: Konstanten, die bei Operationen möglicherweise auftretenden Ausnahmen und die Deklaration von Typnamen (`typedef`). Schließlich kennt IDL Module in Form statisch schachtelbarer Namensräume (`module`).

Die Syntax von IDL ist eng mit C++ [ES90] verwandt, insbesondere wird auch der von C/C++ her bekannte Präprozessor verwendet, der Zeichensatz und die Regeln für Literale und Kommentare sind dieselben. IDL-Quelltexte werden in Dateien mit Endung `„.idl“` plaziert.

Als Ausschnitt aus dem in Abschnitt 3.1 skizzierten Anwendungsbeispiel wird im folgenden Block die Modellierung einer Datenbank von Meßwerten mit einer rudimentären Anfrage-schnittstelle gezeigt:

```
#include „Messungen.idl“

module MesswertDB {
    interface MessDB {
        exception DB_NICHT_BEREIT {};

        Messungen::MesswertListe          // Ein Typ aus Messungen.idl
        suche (in string mst_name, in string bezeichnung)
            raises (DB_NICHT_BEREIT);
    };
};
```

IDL wurde mit Version 1.2 des CORBA-Standards spezifiziert [OMG93], diese Spezifikation wurde dann in die aktuelle Version 2.0 des Standards übernommen [OMG95a; §3].

Orbix unterstützt den vollen Sprachumfang, ObjectBroker hat zumindest zwei Spracheinschränkungen: die Typen von Feldelementen sind eingeschränkt [DEC94d; §1.1.18] und `short`-Konstanten erlauben nicht den vollen Wertebereich laut [OMG93]. Beides schränkt die Ausdrucksmächtigkeit von IDL nicht wirklich ein und wird sich vornehmlich bei Portierungen und beim Zusammenwirken von ORBs bemerkbar machen.

3.3 CORBA-Klient

Konzepte

Dieser Abschnitt behandelt die Sicht eines Klienten auf ein CORBA-System anhand der in Abbildung 5 gezeigten Schnittstellen. Hierbei handelt es sich zum einen um Schnittstellen, über die Operationen auf Objekten aufgerufen werden. Zum anderen kann auf ORB-Funktionen zugegriffen werden. Beide Arten von Schnittstellen werden jeweils im Kontext einer konkreten Programmiersprache bereitgestellt.

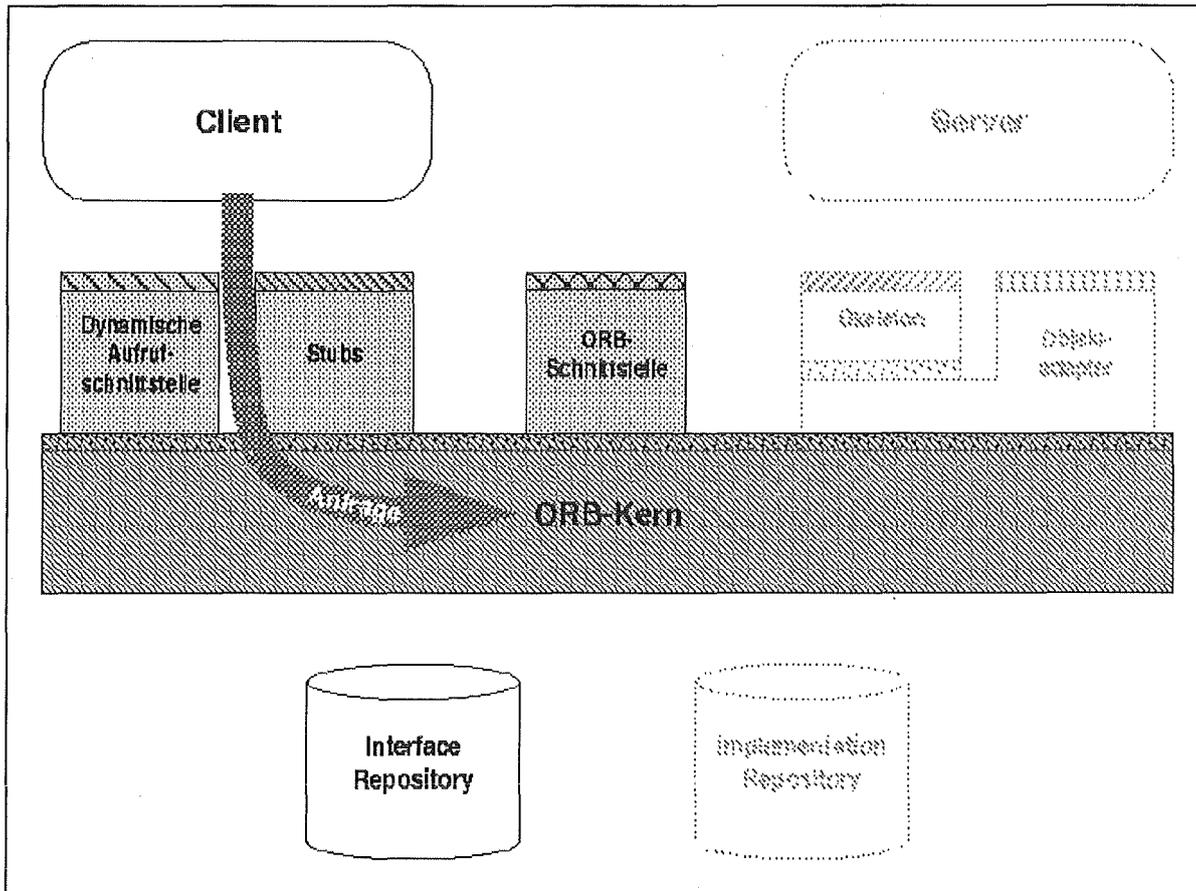


Abbildung 5: Klientenschnittstelle CORBA

In Sinne eines konsistenten Benutzungsmodells ist auch die ORB-Schnittstelle im Standard in Form von IDL-Definitionen beschrieben und wird Anwendungen durch ORB-Implementierungen i.d.R. auch konsistent mit der Schnittstelle zu anwendungsdefinierten Objekten präsentiert. Für diese Art an IDL-Definitionen findet sich im Standard die Bezeichnung *Pseudo-IDL*.

Operationsaufrufe sind zum einen über eine statische wie auch über eine dynamische Aufruf-schnittstelle möglich. Die statische Aufruf-schnittstelle besteht aus operations- bzw. objekttyp-spezifischem Programmtext, der durch den IDL-Übersetzer einer CORBA-Implementierung erzeugt wird. Hauptaufgaben dieser Aufruf-schnittstelle sind:

- Eine möglichst natürliche, transparente Einbettung in die jeweilige Programmiersprache. Dies umfaßt eine Repräsentation für Werte und dabei insbesondere für Objektreferenzen.
- Speicherverwaltung von Parametern und Operationsergebnissen: je nach Sprachanbindung gibt der Standard [OMG95a; §14.19, §16.16] an, ob Klient oder Server für die Speicherfreigabe verantwortlich ist und welche Konsequenzen sich hieraus für den Gegenpart ergeben.
- Handhabung von Objektaufrufen, Ausnahmen und Kontexten. Hierzu enthalten die Methoden der statischen Schnittstelle den Code zur Kommunikation mit dem ORB-Kern bzw. dem Server.

Die dynamische Aufruf-schnittstelle kommt dagegen ohne generierten typspezifischen Programmtext aus. Modelliert wird die dynamische Schnittstelle durch ein Request-Objekt, das

die Parameterliste sowie das Operationsergebnis verwaltet und zudem eine Schnittstelle zur Steuerung der Operationsbearbeitung bietet. Hierüber ist es möglich, erst zur Laufzeit festzulegen, welche Operation auf welchem Objekt aufgerufen wird. Dies macht es dann z.B. möglich, auch von dynamisch typisierten, interaktiven Sprachumgebungen aus auf CORBA-Objekte im vollen Umfang zuzugreifen.

Das zweite Hauptmotiv für die Anwendung der dynamischen Aufrufschnittstelle liegt darin, daß diese es gestattet, beliebige Operationen asynchron aufzurufen und dann zum für die Anwendung passenden Zeitpunkt auf deren Ergebnis zuzugreifen. Hierdurch kann der Parallelitätsgrad eines Anwendungssystems erhöht werden.

Produktspezifika

Die C++-Schnittstelle wurde ab CORBA 2.0 standardisiert [OMG95a; §16] und schlägt sich daher noch nicht im vollen Umfang im betrachteten Produkt Orbix nieder. Allerdings arbeitete IONA aktiv an der Standardisierung der C++-Schnittstelle mit und hat wesentliche Bestandteile bereits in der jetzigen Produktversion berücksichtigt.

Die C++-Schnittstelle von Orbix [IO95a; §3] stellt für jeden IDL-Objektyp eine entsprechende C++-Klasse zur Verfügung. Objektreferenzen werden als Zeiger auf solche Klassen repräsentiert. Der Aufruf von Operationen entspricht dann dem Aufruf virtueller Methoden in C++, wie das nachfolgende Beispiel zeigt. Elementare IDL-Typen (s. Abschnitt 3.2) werden auf entsprechende C++-Typen abgebildet, für zusammengesetzte IDL-Typen stehen entsprechende C++-Klassen zur Verfügung, die eine Reihe von Hilfs- und Bearbeitungsoperationen anbieten. Die Ausnahmebehandlung wird ähnlich der C-Schnittstelle direkt ohne Rückgriff auf C++-Ausnahmen mit Makros und separatem Operationsparameter realisiert. Orbix gibt einen leeren Aufrufkontext vor, wobei beim statischen Operationsaufruf noch auf Unix-Environmentvariablen zugegriffen werden kann.

ObjectBroker bietet die in [OMG93; §5] standardisierte C-Schnittstelle mit einer Reihe spezifischer Erweiterungen. Objekte werden auch in C über Objektreferenzen bearbeitet, in der C-Schnittstelle haben jedoch alle Objektreferenzen denselben Typ. Operationen werden auf globale Funktionen abgebildet, deren erster Parameter jeweils das Objekt ist, auf dem eine Operation aufgerufen wird. Die Abbildung für sonstige Typen ähnelt derjenigen der C++-Schnittstelle. Mit zusammengesetzten Typen wie z.B. einer Sequenz sind jedoch nicht wie in C++ Bearbeitungsoperationen etwa zur Speicherverwaltung direkt assoziiert, sondern separat davon in einer Funktionsbibliothek zusammengefaßt. ObjectBroker bietet eine ausgearbeitete Verwaltung von Kontextobjekten, über die auch der Broker selbst konfiguriert werden kann.

Der nachfolgende Block zeigt als einfaches Beispiel die Suche nach Meßwerten in einem Datenbankobjekt, implementiert mit der C++-Schnittstelle von Orbix:

```
MesswertDB::MessDB*      messDB = <...>;
Messungen::MesswertListe messwerte(20);

TRY {
    messwerte = messDB->suche (mst_name, bezeichnung, IT_X);
}
CATCH (MesswertDB::MessDB::DB_NICHT_BEREIT, db) {
    cerr << "Datenbank nicht bereit !" << IT_X << endl;
    exit(1);
}
```

```
}  
CATCHANY {  
    cerr << "*** Fehler bei Datenbankzugriff!" << endl  
        << "      " << IT_X << endl;  
    exit(1);  
}  
ENDTRY
```

Der Standard [OMG95a] sieht für die dynamische Aufrufchnittstelle Operationen vor, die einen schrittweisen Aufbau der Argumentliste sowie die Zuweisung / Extraktion von Werten erlauben. Diese Schnittstelle wurde von ObjectBroker realisiert [DEC94a; §6]. Orbix geht hier einen Schritt weiter und bietet zusätzlich eine einfacher nutzbare Schnittstelle an, die typspezifische Operationen wie von C++ Streams her bekannt anbietet. Der folgende Block zeigt einen kleinen Ausschnitt des obigen Programmfragments unter Nutzung der dynamischen Aufrufchnittstelle.

```
CORBA::Environment env;  
CORBA::Request r(messDB);  
r.setOperation(„suche“);  
r << mst_name;  
r << bezeichnung;  
r.invoke(env);
```

3.4 CORBA-Server

Konzepte

Die Serverschnittstelle eines ORB betrachtet die Interaktion zwischen ORB und einer Objekt- bzw. Operationsimplementierung. Betroffen sind die in Abbildung 6 hervorgehobenen Bestandteile. Wie die Abbildung zeigt, wird der Kern der Schnittstelle durch einen Objektadapter gebildet, der wiederum über statisch generierte Programmfragmente auf die eigentlichen Objekt-/Operationsimplementierungen zugreift. Mit CORBA 2.0 wurde eine dynamische Serverschnittstelle als Analogon zu der in Abschnitt 3.3 eingeführten dynamischen Aufrufchnittstelle spezifiziert. Da diese zur Zeit nicht verfügbar ist, wird sie hier nicht näher betrachtet.

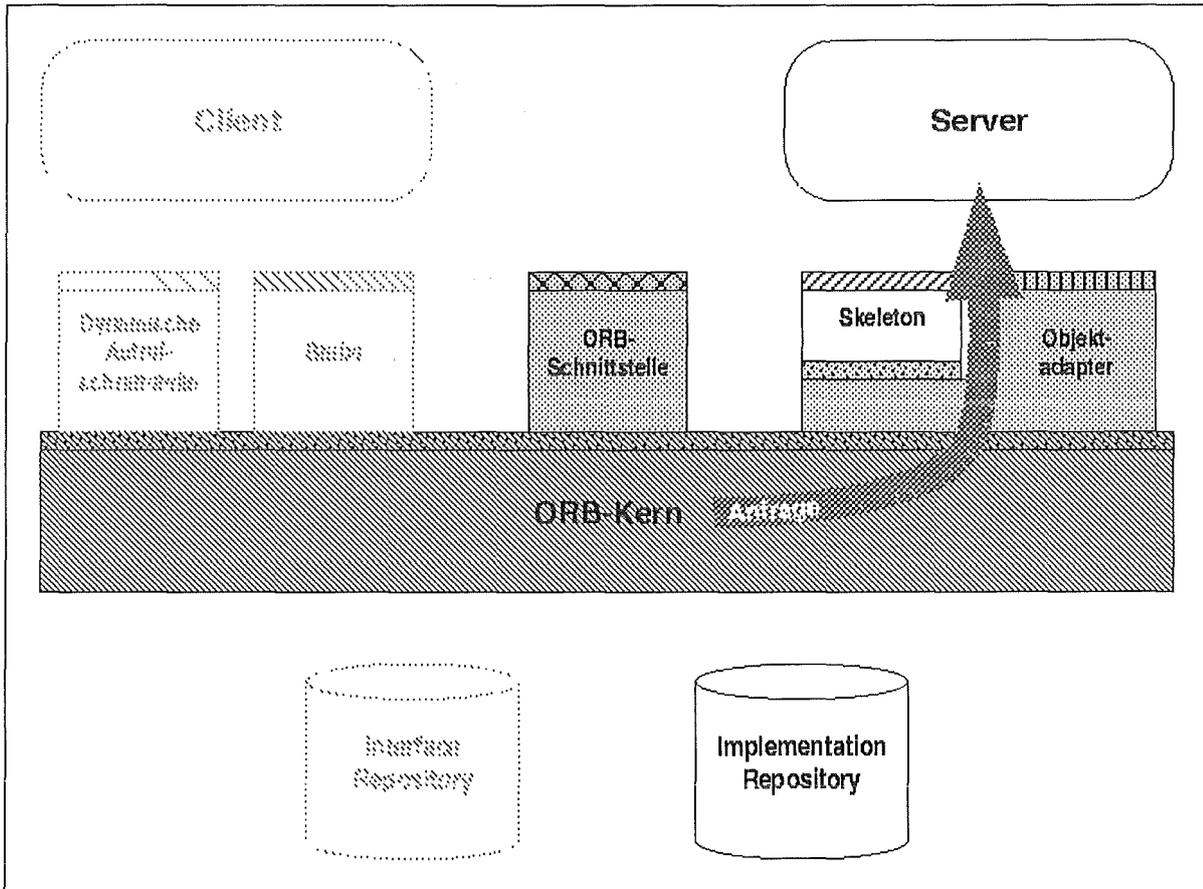


Abbildung 6: Serverschnittstelle CORBA

Minimal im CORBA-Standard gefordert ist der sogenannte *Basic Object Adapter (BOA)*, von dem erwartet wird, daß er ein breites Anwendungsspektrum abdeckt. Es finden sich denn auch mit Ausnahme von SOM [KL95][KL96] keine Produkte, die eine wesentlich abweichende Schnittstelle aufweisen. Vielfältige Variationen finden sich dagegen in Administrations- und Eingriffsmöglichkeiten.

Grundaufgaben eines Objektadapters sind:

- Interpretation und Verwaltung von Objektreferenzen, einschließlich der persistenten Speicherung einer Objektreferenz über die Lebensdauer eines Objekts.
- Authentifizierung und Autorisierung von Aufrufen bzw. Aufrufern.
- Aktivierung und Deaktivierung von Objekten und Methoden, d.h. deren Bereitstellung für die Requestbearbeitung. Hierzu zählt auch deren Auswahl.
- Aufruf der ausgewählten Methode.

Der Standard unterscheidet bei der Aktivierung von Implementierungen und der Zuordnung von Objekten/Methoden zu Prozessen 4 Aktivierungsmodi:

- *unshared / shared server*: Der Server enthält maximal ein / potentiell mehrere aktive Objekte einer Implementierung.
- *persistent server*: Der Server wird von außerhalb und nicht durch den BOA (de)aktiviert. Davon abgesehen handelt es sich zumeist um einen Shared Server.
- *Server-per-Method*: BOA startet und beendet einen Serverprozeß je Methodenaufruf und es können mehrere solcher Prozesse zur gleichen Zeit aktiv sein.

Der für einen Anwender sichtbarste Teil des BOA sind die Vorgaben zur Anbindung eigener Implementierungen an die vom IDL-Übersetzer erzeugten Programmrümpfe (*skeleton*). Die mit CORBA 2.0 standardisierte C++-Sprachanbindung läßt hier mehrere Möglichkeiten zu. Vorgegeben wird zwar die eigentliche Implementierungsklasse, die aus der IDL-Definition abgeleitete Methoden aufweisen muß und ansonsten beliebig benannt sein kann. Der Standard trifft dagegen keine Aussagen über den Mechanismus, mit denen die anwendungsdefinierte Implementierungsklasse an die für den BOA generierten Fragmente angebunden wird. Des weiteren läßt der Standard Freiraum für C++-Sprachdialekte, insbesondere hinsichtlich der Unterstützung von Namensräumen und der Nutzung von C++-Ausnahmen zur Realisierung des CORBA-Ausnahmemechanismus.

Produktspezifika

In Erweiterung der obigen Aktivierungsmodi bietet Orbix die Möglichkeit, einen Serverprozeß jeweils je Nutzer oder auch je Klientprozeß zu starten. Orbix unterstützt zwei Mechanismen zur Anbindung von Implementierungsklassen an generierte Skeletons, die zugleich auch die beiden wichtigsten Realisierungsalternativen für C++ darstellen. Beide Lösungen werden durch Makromechanismen unterstützt und erfordern einen ähnlich geringen Aufwand durch den Anwender:

- Per *Vererbung* (*BOAImpl*-Lösung): Die Implementierungsklasse erbt von einer generierten Skeletonimplementierung und implementiert per Vererbung vorgegebene abstrakte Methoden.
- Per *Delegation* (*TIE*-Lösung): Der BOA bindet eine generierte Klasse ein, die alle Aufrufe auf ein ihr zugeordnetes Objekt der Anwendungsimplementierungsklasse umlenkt. Diese Lösung erzeugt zusätzliche Delegationenobjekte, liefert dafür aber Flexibilität: Implementierungsobjekte können so mehrere nicht durch IDL-Vererbung verbundene Schnittstellen implementieren und die Wiederverwendung von Implementierungsklassen wird erleichtert, da diese keinen direkten Bezug zu einer Skeletonklasse aufweisen [IO95a; 1.4].

In der momentanen Version verwendet Orbix keine C++-Ausnahmen [IO95a; 3.15]. CORBA-Ausnahmen werden daher wie in der C-Schnittstelle über einen zusätzlichen expliziten Methodenparameter realisiert. Der folgende Block zeigt einen Ausschnitt aus einer nach dem BOAImpl-Ansatz realisierten Implementierungsklasse für das laufende Anwendungsbeispiel:

```
class MesswertDB_i {
public:
    class MessDB_i : public MesswertDB::MessDBBOAImpl {
    public:
        <...>
        virtual Messungen::MesswertListe suche
            (const char *      mst_name,
             const char *      bezeichnung,
             CORBA::Environment &IT_env
              =CORBA::default_environment);
    };
};
```

Zur Realisierung der serverseitigen Schnittstelle bietet ObjectBroker eine Reihe von Erweiterungen und Werkzeugen: über eine sogenannte *Method Mapping Language* [DEC94a; §5] können Entwickler bestimmen, welche Methoden zur Bearbeitung einer Operation im Server aufgerufen werden. Die Wahl einer Methode kann dabei auch dynamisch aufgrund von Server/ORB-Parametern erfolgen. Die über ein Werkzeug generierbare Voreinstellung nimmt eine direkte 1:1-Abbildung von Operationen auf globale C-Funktionen vor.

ObjektBroker implementiert die oben genannten Server-Aktivierungsmodi [DEC94a; §4].

3.5 Repositories

Konzepte

CORBA unterscheidet zwei Repositories, die in Abbildung 7 skizziert sind. Diese Trennung und die Implementierung der Repositories generell ist im Standard allerdings nicht festgelegt, so daß auch eine gemeinsame Implementierung denkbar ist. Unspezifiziert sind des weiteren alle Verwaltungsoperationen, der Grad der Nutzung der Repositories im „normalen“ Ablauf von Objektaufrufen und insbesondere ist auch die Art und Weise offen, wie Repositories persistent gespeichert werden.

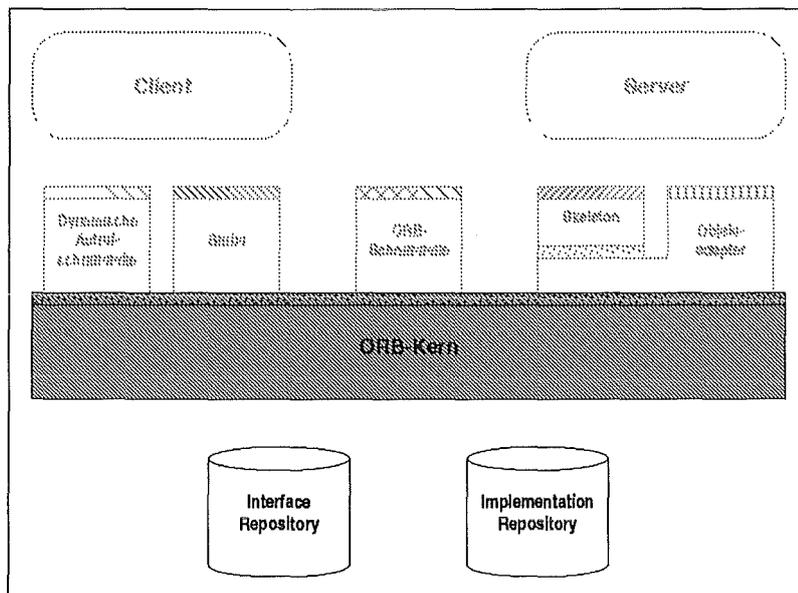


Abbildung 7: CORBA - Repositories

Der Inhalt des Interface Repository ist ab CORBA 1.2 [OMG93; §7] in Form von IDL-Schnittstellen spezifiziert. Etwas vereinfacht enthält das Interface Repository die in einer IDL-Datei enthaltenen Definitionen in strukturierter Form. Mit CORBA 2.0 wurden diese Spezifikationen um eine Modifikationsschnittstelle erweitert [OMG95a; §6], die die Änderung des Repositories gestattet.

Im Gegensatz zum Interface Repository läßt der Standard Inhalt und Semantik des Implementation Repository offen, es findet sich lediglich die Aussage, daß in der Basic Object Adapter-

Schnittstelle jeder Objektreferenz ein Beschreibungsobjekt im Implementation Repository zugeordnet ist [OMG95a; §8].

Die Bedeutung des Interface Repository wächst mit der Einführung von ORB-Interoperabilität ab CORBA 2.0: Eine sinnvolle Zusammenarbeit von ORBs setzt die Kopplung von Repositories voraus. Im Standard finden sich hierzu eher vage Angaben. Als Grundmechanismen werden die Vergabe universal eindeutiger Ids für Repositoryobjekte sowie deren Versionierung skizziert [OMG95a; §6.4, §6.6] — ohne allerdings Mechanismen und Konsistenzbedingungen näher festzulegen.

Produktspezifika

Sowohl Orbix als auch ObjectBroker implementieren die Interface Repository-Schnittstelle aus [OMG93; §7].

Orbix legt Repositoryinformationen im Dateisystem ab. Als Speicherformat für das Interface Repository wird der entsprechende IDL-Quelltext selbst verwendet, der auf Anforderung gegebenenfalls dynamisch analysiert wird [IO95b]. Das Implementation Repository hält Informationen über den Aktivierungsmodus von Servern und über die Zuordnung von Objekten/Schnittstellen zu Serverimplementierungen. Zur Verwaltung der Repositoryinformationen stehen einige Werkzeuge zur Verfügung, die im nächsten Abschnitt skizziert werden.

ObjectBroker legt seine Repositories ebenfalls im Dateisystem an, wobei Interface und Implementation Repository in einer Datei zusammengefaßt sind. Einträge in das Implementation Repository werden in einer dedizierten Sprache beschrieben, der sogenannten *Implementation Mapping Language* [DEC94a; §4]. ObjectBroker bietet bereits die oben genannten Mechanismen zur Interoperabilität von Repositories durch die Assoziation eindeutiger Bezeichner mit Repositoryobjekten [DEC94a; §8].

Weitere Bestandteile der ObjectBroker-Repositories sind die Verwaltung von Kontextobjekten, sowie von Benutzern und deren Zugriffsrechten [DEC94b; §2]. ObjectBroker bietet eine Verwaltung von Kontexten auf globaler, Gruppen- und Benutzerebene [DEC94a; §3.9], sowie eine sogenannte *Context Object Language* zur Beschreibung [DEC94b; §A].

Der folgende Block zeigt als Beispiel einen Eintrag in das Orbix Implementation Repository:

```
Name           : MesswertDB
Comms           : xdr/tcp
Activation      : shared
Owner          : guru
Launch         : ;
Invoke         : ;
Marker         :          Launch Command
* /usr/guru/Auswertung/dbSrv
```

3.6 CORBA-Systemkern

Dieser Abschnitt skizziert Aspekte der Systemimplementierung und die den betrachteten Systemen zugehörigen Werkzeuge.

Eine CORBA-Interaktion mit Orbix involviert zumindest drei Prozesse: den Klienten, den Server, sowie einen Dämonprozeß, der die auf einem Rechner laufenden Serverprozesse verwaltet. Dieser Dämonprozeß ist nur für die Kontaktaufnahme zwischen Klient und Server zuständig, im weiteren kommunizieren Klient und Server direkt [IO95a; §8.5].

Im Rahmen der Verwaltung von Serverprozessen realisiert der Dämonprozeß auch das Implementation Repository [IO95a; §9.3]. Über eine Reihe kommandozeilenbasierter Werkzeuge kann das Repository modifiziert, ausgelesen und durchsucht werden. Zudem können aktive Server aufgelistet, getestet oder beendet werden.

Obige Werkzeuge bieten eine rudimentäre Schnittstelle zur Verwaltung von Servern in einem Netzwerk, ohne jedoch durch ein Vorgehensmodell und komfortable Schnittstellen eine umfassende Unterstützung in der Administration bieten zu können.

Das zentrale Werkzeug in Orbix ist der kommandozeilenbasierte IDL-Schemaübersetzer. Dieser übernimmt zugleich die Verwaltung des Interface Repository, einschließlich des Lesezugriffs auf dieses.

ObjectBroker ist ähnlich strukturiert, wobei der hier Agent genannte rechnerlokale Prozeß analog der in Abschnitt 3.5 beschriebenen erweiterten Funktionalität auch umfangreichere Aufgaben übernimmt: die Verwaltung von Kontextobjekten, Benutzern und Zugriffsrechten [DEC94c]. In ObjectBroker ist der Agent allerdings im Unterschied zu Orbix nicht nur bei der Verbindungsaufnahme beteiligt, sondern bei jedem Operationsaufruf. Dies bringt Vorteile hinsichtlich der Umsetzung von Sicherheitskonzepten, die allerdings durch Laufzeitnachteile erkauft werden. Des weiteren ist die Skalierbarkeit einer solchen Architektur problematisch, da der Agent zum Flaschenhals werden kann.

Als Verwaltungswerkzeuge stehen eine graphische Oberfläche sowie ein Satz kommandozeilenbasierter Werkzeuge zur Verfügung [DEC94a; §2]. Diese sind - etwas vereinfacht - eine Obermenge der von Orbix angebotenen Werkzeuge, die auch die zusätzliche ObjectBroker-Leistungen wie die Verwaltung von Benutzern und Zugriffsrechten abdecken (vgl. Abschnitt 3.5).

3.7 CORBAservices und CORBAfacilities

In Abschnitt 3.1 wurde die Object Management Architecture (OMA) als Infrastruktur für Anwendungen in verteilten, heterogenen Umgebungen eingeführt. Abbildung 8 zeigt die Referenzarchitektur der OMA in etwas detaillierterer Sicht. Neben dem in den vorhergehenden Abschnitten betrachteten Object Request Broker wurden weitere Architekturkomponenten identifiziert [OMG90]:

- *CORBA services* (früher: *Object Services*) bieten standardisierte Komponenten, die im Laufe des Lebenszyklus eines Objekts benötigt werden. Diese sollten in jedem CORBA-basierten Anwendungssystem zur Verfügung stehen.
- *CORBA facilities* (früher: *Common Facilities*) subsumieren generische, konfigurierbare Anwendungsfunktionen, die optional in einer OMA-Umgebung vorzufinden sind.
- *Application Objects* schließlich ist die Sammelbezeichnung für alle anwendungsspezifischen Objekte, die außerhalb des Standardisierungsbereichs der OMG liegen.

Neben der Strukturierung des Anwendungsgebiets leitet sich aus obiger Einteilung zugleich eine Reihenfolge der Standardisierungsbestrebungen der OMG ab: Nachdem anfänglich der Schwerpunkt auf der Klärung der begrifflichen Grundlagen [OMG90] und der Standardisierung der Kernkomponente ORB lag [OMG93][OMG95a], richten sich die weiteren Arbeiten zunehmend auf höhere Systemfunktionen. So wurde inzwischen ein erster Satz an CORBA-services standardisiert [OMG95b] und erste Implementierungen sind für die nähere Zukunft angekündigt (vgl. Abschnitt 2.4).

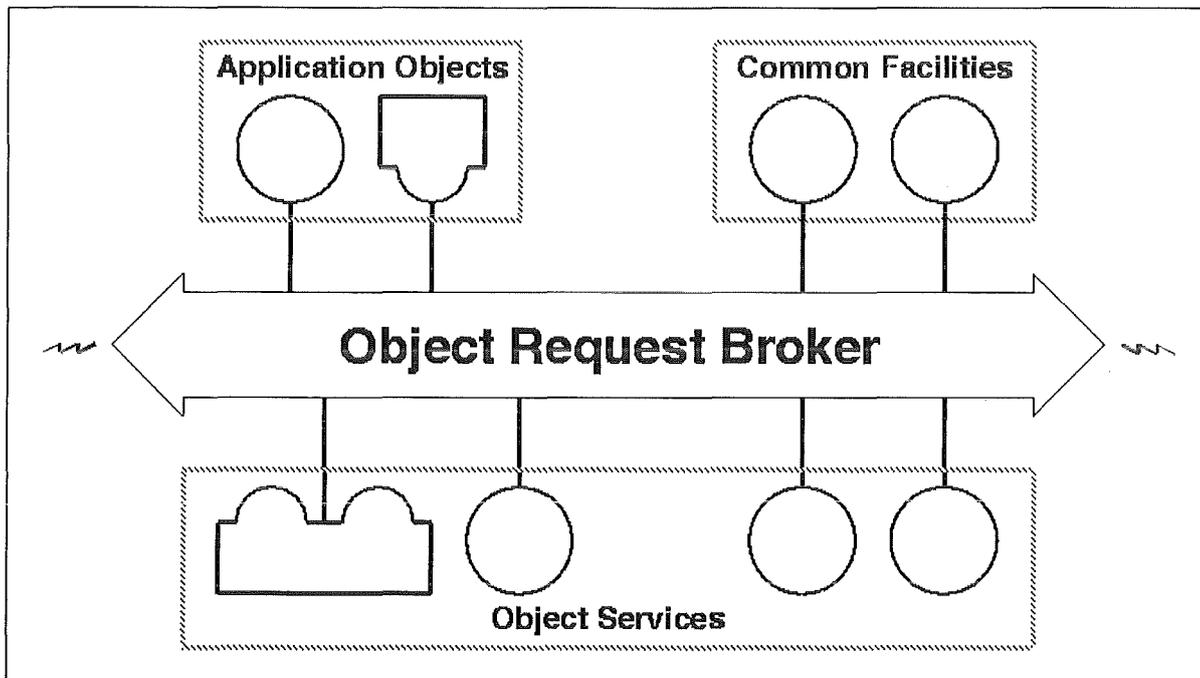


Abbildung 8: OMA-Referenzarchitektur

Im folgenden sollen die bisher identifizierten Dienste kurz skizziert werden:

- *LifeCycle Service*: Grundlegende Schnittstellen zum Erzeugen, Löschen, Kopieren und Bewegen verteilter Objekte.
- *Naming*: Verzeichnisdienst für verteilte Objekte innerhalb eines verteilten, strukturierten Namensraums.
- *Relationships*: Verwaltung von Beziehungen zwischen verteilten Objekten einschließlich Schnittstellen zur Traversierung eines Objektgraphs.
- *Externalization*: Schnittstelle zur Transformation von Objektzustand in einen flachen Datenstrom, sowie umgekehrt zur Extraktion von Objekt(geflechten) aus einem Datenstrom.

- *Events*: Propagieren von Ereignissen zwischen verteilten Objekten, die indirekt über einen Ereigniskanal oder auch direkt miteinander in Beziehung stehen.
- *Properties*: Dynamische Annotation von Objekten.
- *Change Management*: Verwaltung von Versionen und Konfigurationen von Schnittstellen, Implementierungen und Objekten.
- *Security*: Mechanismen zur Zugriffskontrolle und -protokollierung.
- *Licensing*: Einschränkung der Dienstzugriffe auf lizenzierte Objekte.
- *Persistenz*: Schnittstellen zur Verwaltung des persistenten Teils von Objektzuständen.
- *Concurrency Control*: Sperrmechanismen, die im Rahmen von Transaktionen oder auch explizit zur Synchronisation konkurrierender Operationen eingesetzt werden.
- *Transaktionen*: Realisierung geschachtelter und flacher Transaktionen für eine Folge von Aktionen mit Objekten.
- *Time*: Mechanismen zur Synchronisation von Uhren in einer verteilten Umgebung.
- *Anfragen*: Deklarative Anfragen auf Mengen verteilter Objekte sowie Änderungsoperationen auf solchen Mengen.
- *Trading*: Vermittlung zwischen Dienstinteressenten und Dienst Anbietern basierend auf Beschreibungen der angebotenen Dienste.

CORBAfacilities werden in der OMA grob in zwei Kategorien eingeteilt:

- *Vertical Facilities*: Dienste für bestimmte Anwendungsbereiche.
- *Horizontal Facilities*: Anwendungsübergreifende Dienste aus Bereichen wie Benutzungsschnittstellen, Aufgabenverwaltung u.ä.

Die betrachteten CORBA-Produkte umfassen noch keine Dienstimplementierungen. ObjectBroker beinhaltet eine teilweise Implementierung des Naming Service [DEC94b; §7.6].

3.8 Praktische Erfahrungen

In diesem Abschnitt fassen wir unsere Erkenntnisse im Umgang mit den CORBA-Produkten zusammen.

3.8.1 ObjectBroker

3.8.1.1 Grundsätzliches

Für unsere Tests stand uns ObjectBroker in der Version 2.5A-04 für SunOS mit einer Evaluationslizenz zur Verfügung. Als Übersetzer wurde der GNU gcc eingesetzt. DEC ordnet den ObjectBroker in die Klasse von Systemerweiterungen ein. Daraus resultiert, daß die Wartung und Pflege der Software dem Systemadministrator obliegt. Das Installieren und Starten der Kernsoftware ist nur mit den Zugriffsrechten eines Systemadministrators (Unix bzw. SunOS: root) möglich. Die Zeit für praktische Erfahrungen mit ObjectBroker war aufgrund der langen Lieferzeit knapp bemessen.

3.8.1.2 Installation

Der ObjectBroker zeigte sich bei der Installation zunächst unkompliziert. Ein Installationskript erledigt die Installation der kompletten Software automatisch. Probleme treten jedoch auf, wenn die von DEC geforderte Form der Installation mit der lokalen Rechnerumgebung in Konflikt geraten. ObjectBroker ist zwar weitgehend flexibel in der Wahl der zu verwendenden Verzeichnisse, besteht aber in einigen Fällen auf ein ganz bestimmtes Verzeichnis. Wird dies beachtet, so verläuft das Einrichten der Software problemlos.

3.8.1.3 Dokumentation

Die mitgelieferte Literatur ist ohne zu übertreiben als umfangreich zu bezeichnen. Sie besteht im wesentlichen aus: Handbuch für Installation und Management, Handbuch zur Systemintegration [DEC94a], Referenzhandbuch [DEC94b], ein allgemeines CORBA-Buch sowie einigen weiteren Informationen.

Wenn man sich mit Analyse und Design eines Systems beschäftigt und nur Code auf abstrakteren Ebenen wie IDL entwickelt, ist die Dokumentation als hervorragend zu bezeichnen. Die Schwächen offenbaren sich jedoch schon bald, wenn man die eigentliche Applikation in der betreffenden Programmiersprache (hier ANSI-C) schreiben möchte. Hierbei wird man allein gelassen. Die einzigen Quellen für fehlende Programminformationen sind der CORBA-Standard, der die Abbildung von IDL nach C vorschreibt, und die mitgelieferten Beispiele. Der sicherste aber auch der mühsamste Weg, sich mit ObjectBroker vertraut zu machen, ist der, sich intensiv mit den Beispielen und deren Programmcode zu beschäftigen.

Es fehlt ein Handbuch, das die Funktion eines sonst üblichen „Programmer's Guide“ erfüllt.

3.8.1.4 Handhabung bei der Programmentwicklung

In CORBA ist das zentrale Mittel Programmschnittstellen zu beschreiben, die IDL. IDL ist als Untermenge von C++ naturgemäß eher mit C++ als mit C verwandt. C stellt eben keine Objekte zur Verfügung. Konzepte der Kapselung und Namenshierarchie, wie sie in IDL zu finden sind, müssen in C mühsam und aufwendig „simuliert“ werden. In der C-Anbindung sind Programme merklich schwächer typisiert als in C++; so haben etwa alle Objektreferenzen in C denselben Typ. Dies macht sich bei der Programmentwicklung und Fehlersuche negativ bemerkbar.

Anfangs steht dem Programmierer die enorme Flexibilität von ObjectBroker im Weg. ObjectBroker generiert nämlich ausgehend von einer IDL-Beschreibung eine Reihe von Dateien. Bis zu diesem Zeitpunkt wurde noch keine Zeile der eigentlichen Anwendung programmiert. Wenn man die Funktion jeder einzelnen Datei erst einmal begriffen hat, stellt dieses Konzept,

Verhalten aufzuteilen und einzeln änderbar zu machen, die Mächtigkeit von ObjectBroker heraus.

ObjectBroker-Funktionalität wurde in Form einer Shared Library (analog einer DLL in PC-Betriebssystemen) ausgeliefert. Die Größe eines ObjectBroker-Programms wird daher durch den Umfang des Anwendungscodes bestimmt und lag bei unseren Tests in der Größenordnung unter 100 KByte.

3.8.1.5 Bedienung

ObjectBroker stellt zwei Möglichkeiten der Bedienung zur Verfügung. Das sind:

- (1) Die Bedienung über ein graphisches Werkzeug.
- (2) Die Bedienung über die Kommandozeile.

Ersteres ist vor allem zum Anschauen bereits installierter Objekte interessant. Repository-Einträge werden übersichtlich in Fenstern dargestellt und Definitionen (z.B. in IDL) können auf Wunsch angezeigt werden. Vom graphischen Werkzeug aus können auch wesentliche Systemadministrationsaufgaben erledigt werden.

Die zweite Bedienmöglichkeit liefert dem geübten Programmierer schnelleren Zugriff auf gewünschte Informationen. Außerdem können Übersetzungen (z.B. von IDL nach C) direkt angestoßen werden. Die installierte Version ließ Unterstützung für die Integration in Makefiles vermissen; entsprechende make-Regeln mußten erst entwickelt werden.

3.8.2 Orbix

3.8.2.1 Grundsätzliches

Für unsere Tests stand uns Orbix in der Version 1.3.1 für SunOS 4.* und SunOS 5.* (Solaris 2.*) zur Verfügung. Als Übersetzer wurde der GNU g++ eingesetzt. Orbix wurde im Evaluierungszeitraum auch für weitere Implementierungsaufgaben zumeist von Studenten eingesetzt, ohne daß hierbei Probleme bekannt wurden.

3.8.2.2 Installation

Die Installation von Orbix ist unkompliziert und erfordert keine gesonderten Privilegien. Eine Integration in das Installationskonzept des FZI ging problemlos vonstatten.

3.8.2.3 Dokumentation

Die mitgelieferte Literatur ist umfangreich, klar gegliedert und leicht zugänglich. Sie umfaßt zwei mit vielen Beispielen versehene Programmierhandbücher [IO95a, IO95b]. Die Systemauslieferung enthält eine Reihe von Programmbeispielen, die unterschiedliche Aspekte des Gesamtsystems demonstrieren.

3.8.2.4 Handhabung bei der Programmentwicklung

Anwendungsentwicklung in IDL ist für C++-Programmierer ohne großen Einarbeitungsaufwand möglich. Die C++-Abbildung insbesondere auf der Klientenseite wird als natürliche Einbettung in C++ angesehen, der Lernaufwand auf Serverseite ist nicht wesentlich größer. Erste Ergebnisse in Form von Prototypen eines Orbix-Programms sind schnell lauffähig und können hiervon ausgehend dann weiterentwickelt werden.

Orbix-Funktionalität stand nur in Form statischer Bibliotheken zur Verfügung, ein Orbix-Programm hat daher eine Mindestgröße von ca. 900 KByte.

3.8.2.5 Bedienung

Orbix wird im wesentlichen über kommandozeilenorientierte Werkzeuge bedient, die typischerweise über ein Makefile in den Entwicklungsprozeß integriert werden. Graphische Werkzeuge insbesondere zur Administration und zum Monitoring eines Orbix-Systems stehen nicht zur Verfügung. Inwieweit sich dies bei umfangreichen Anwendungen auswirkt, kann zur Zeit nicht beurteilt werden.

3.8.2.6 Support

Bei einigen im Ablauf der praktischen Erprobung aufgetretenen Fragen stand mit dem deutschen Distributor von IONA (iO Software) ein jederzeit hilfsbereiter, kompetenter und hochmotivierter Ansprechpartner zur Verfügung.

3.9 Resümee

In den vorhergehenden Abschnitten wurden CORBA-Implementierungen anhand des im Standard [OMG95a] erfaßten Umfangs betrachtet.

Beide Produkte realisieren den in Version 1.2 des Standards [OMG93] beschriebenen Stand; ObjectBroker mit teilweisen Einschränkungen (vgl. Abschnitt 3.2). Sie nehmen in der aktuellen Systemversion bereits - wenn auch unterschiedliche - Teile der mit CORBA 2.0

[OMG95a] bzw. den CORBAservices [OMG95b] einhergehenden Änderungen vorweg. Anhand eines vereinfachten Anwendungsbeispiels konnte die CORBA Funktionalität der Produkte exemplarisch gezeigt werden.

ObjectBroker weist umfangreichere Funktionen auf, insbesondere im Hinblick auf die Verwaltung verteilter Repositories und von Sicherungsmechanismen. Zudem bestehen feingranulare Eingriffsmöglichkeiten in den Ablauf der Operationsbearbeitung durch anwendungsspezifische Programmierung ansonsten systemgenerierter Programmfragmente. Diese Funktionalität bringt jedoch auch ihre Kosten mit sich, die sich vor allem im Einarbeitungsaufwand und im Aufwand zur Instantiierung eines Anwendungssystems niederschlagen. Ein Einsatz von ObjectBroker sollte daher durch Schulungen oder erfahrene Mitarbeiter vorbereitet und begleitet werden.

Orbix hingegen präsentiert sich als ein Produkt, daß sich auf den eigentlichen Kernbereich der Implementierung des CORBA-Standards konzentriert und sich leicht in bestehende Systemstrukturen integrieren läßt. Durch die Abbildung nach C++ entfallen einige der von ObjectBroker angebotenen Eingriffsmöglichkeiten. Zugleich wird dadurch allerdings die Anwendungsprogrammierschnittstelle deutlich vereinfacht.

4 Praktische Erprobung einer Funktionalität für das Trading

4.1 Einführung Trading

Durch die Anbindung von lokalen Netzen an Weitverkehrsnetze und deren Zusammenschluß haben verteilte Systeme in den vergangenen Jahrzehnten immer mehr an Bedeutung gewonnen. In verteilten Systemen werden Netzteilnehmern eine Vielzahl von Diensten bereitgestellt. Beispielsweise werden in WWW-UIS Dienste wie der UDK (Umweltdatenkatalog), TULIS (Technosphäre- und Luftinformationssystem), MEROS (Meßreihenoperationssystem), GRDB (Grenz-/Richtwert-Datenbank) sowie Karten und Darstellungsdienste angeboten.

Die steigende Zahl an Systemdiensten in verteilten Systemen führt zu einem unübersichtlichen Angebot. Dadurch wird es für Anwendungsentwickler immer schwerer, den zur Erfüllung einer Aufgabe geeignetsten Dienst auszuwählen, da hierfür Kenntnisse erforderlich sind, die sie nicht besitzen. Dies führt dazu, daß ein neuartiger Basisdienst in verteilten Systemen zunehmend an Bedeutung gewinnt: der *Trading-Dienst*.

Trading-Dienste stellen eine Erweiterung herkömmlicher Klient-Server-Systeme dar, indem Trader als eine zusätzliche Instanz zur Verfügung gestellt werden. Ein Klient wendet sich mit einer Dienstanfrage, d.h. mit dem Wunsch nach einem Server, der die benötigte Dienstleistung erbringt, an einen *Trader*, also die Instanz, die den Trading-Dienst bereitstellt. Der Trader seinerseits verwaltet das im System verfügbare Dienstangebot, in dem er die von den Servern verfügbaren Dienste mit ihren Eigenschaften registriert, und bei einer Dienstanfrage eines Klienten versucht, einen für ihn geeigneten Server zu vermitteln.

Aufgabe des Traders ist also neben der *Verwaltung* der im verteilten System vorhandenen Dienste auch die *Dienstevermittlung* zwischen Dienstanfragen des Klienten und Dienstangeboten des Servers.

Durch diesen neuartigen Dienst entsteht eine Art offener Dienstemarkt, in dem sich Dienstanbieter und Dienstbenutzer nicht mehr gegenseitig kennen müssen, wie dies in klassischen Klient-Server-Systemen der Fall ist.

4.2 Einsatzbeispiele des Trading im UIS Baden-Württemberg

Wie bereits im obigen Abschnitt dargestellt, ist der Trading-Dienst gut geeignet, die steigende Zahl an Diensten im UIS Baden-Württemberg zu verwalten. Durch den Einsatz eines Traders zur Dienstevermittlung zwischen Klient und Server können bereits bei der Auswahl des dienstbringenden Servers Kriterien, wie beispielsweise die Kosten für die Dienstbringung, berücksichtigt werden. Der Dienstbenutzer kann dann bei der Anforderung eines Dienstes Eigenschaften angeben, die vom Trader bei der Auswahl des Dienstbringers berücksichtigt werden (*statisches Trading*). Im Rahmen des WWW-UIS denkbare Eigenschaften sind zum Beispiel eine bestimmte Darstellungsart eines Ergebnisses (HTML oder ASCII), eine obere Grenze für die Auflösung einer Kartendarstellung oder auch der Aktualitätsgrad der gelieferten Daten.

In einem verteilten System können Rechnerknoten ausfallen, was dazu führt, daß Dienste auf Knoten vorübergehend nicht verfügbar sind. Um Ausfälle vor dem Benutzer zu verbergen, können identische Dienste auf unterschiedlichen Rechnerknoten implementiert sein. Bei Ausfall eines Knotens leitet der Trader die eingehenden Dienstanforderungen an einen funktionsfähigen Rechnerknoten weiter, der denselben Dienst erbringt (*dynamisches Trading*).

Der Einsatz eines Traders ermöglicht eine flexible und änderungsfreundliche Systemkonfiguration, bei der neue Versionen eines Dienstes in das System aufgenommen werden können (vgl. Abschnitt 4.3), ohne die Funktionsweise der weiterhin vorhandenen früheren Versionen zu beeinträchtigen. Insbesondere bei einem sich dynamisch ändernden System wie dem UIS Baden-Württemberg ist eine derartige Flexibilität von großer Bedeutung.

4.3 Der DAIS-Trader

Das DAIS-System geht auf das ANSA-Projekt zurück (Advances Network Systems Architecture [ANS91]), das eines der ersten Projekte war, die sich mit dem Konzept des Trading befassten. Es hatte Einfluß auf die Trader-Spezifikationen der OMG und der ISO/ODP, welche beide die zur Zeit stattfindende Integration des Trading in die CORBA-Architektur, neben anderen Trader-Proposals, stark beeinflussen.

Das DAIS-Software-Paket bietet als Werkzeug zur Entwicklung von Klient-Server-Systemen in heterogenen verteilten Systemen auch einen Trading-Mechanismus zur Dienstevermittlung

an. Das DAIS-System erfüllt zwar nicht die CORBA 1.2-Spezifikation, enthält aber zusätzlich eine Komponente für das Trading.

Um Dienste systemweit zugänglich zu machen, muß ein Server den von ihm angebotenen Dienst bei einem Trader registrieren lassen. Hierzu werden folgende Angaben benötigt:

1. Konformitätseigenschaften des Dienstes
2. Kontext des Dienstes (nicht identisch mit dem CORBA Kontext-Begriff aus Kapitel 3)
3. Eigenschaften des Dienstes

Mit Hilfe der *Konformitätseigenschaften* aller registrierten Dienste wird eine Vererbungshierarchie in Form eines gerichteten zyklensfreien Graphen im Trader festgelegt (vgl. Abbildung 9).

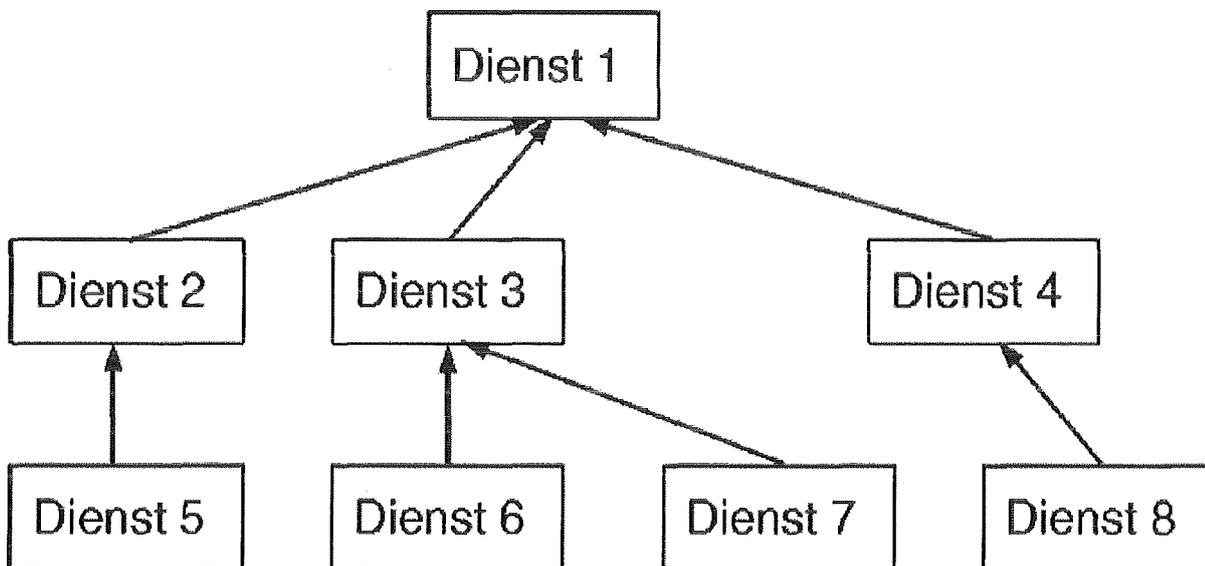


Abbildung 9: Konformitätshierarchie von Diensten im DAIS-Trader

In Abbildung 9 ist Dienst 1 Obertyp aller anderen Dienste. Dies bedeutet, daß Dienst 2 bis Dienst 8 mindestens die Funktionalität von Dienst 1 haben müssen und somit dessen Anforderungen beantworten können. In WWW-UIS könnte z.B. der Dienst 2 die allgemeine Objektsuche im UDK sein, welche durch den Dienst 5, die Detail-Objektsuche, verfeinert wird.

Falls ein bestimmter Dienst nicht verfügbar ist, so kann er von einem seiner Subtypen erbracht werden, ohne daß der Klient längere Zeit warten muß. Dies bedeutet eine deutliche Erhöhung der Gesamtleistung des verteilten Systemes. Der Typ-Begriff in DAIS ist identisch mit dem in IDL. Dies bedeutet, daß ein in IDL definiertes Objekt unter demselben Namen im DAIS-Trader registriert werden muß.

In der DAIS-Trader-Realisierung muß der Programmierer des Servers allerdings dafür Sorge tragen, daß die bei der Registrierung angegebene Konformität auch den realen Gegebenheiten entspricht. Der Aufbau einer Vererbungshierarchie in IDL führt nicht automatisch zu einem Aufbau der Konformitätshierarchie im Trader. Dies erhöht zwar die „Sorgfaltspflicht“ des Entwicklers, erhöht jedoch auch die Flexibilität.

Der *Kontext* dient der hierarchischen Strukturierung der exportierten Server-Dienstleistungen im Trader und ist einem Dateibaum vergleichbar. Er gestattet insbesondere dem Klient, den Suchraum des Traders durch Angabe des gewünschten Suchbereiches einzuschränken. Im WWW-UIS -Szenario könnten Dienste beispielsweise durch die Angabe des Kontextes einzelnen Abteilungen zugeordnet werden.

Die *Eigenschaften des Dienstes* ermöglichen dem Server seinen Dienst genauer zu beschreiben, um ihn von den Diensten anderer Server mit identischen Konformitäts- und Kontexteigenschaften zu unterscheiden. Hierbei werden allerdings nur statische Attribute berücksichtigt, die in Form von Attributname-Wert-Paaren dem Trader übergeben werden. Der Ort bzw. der Name des Dienstbringers (falls mehrere replizierte Dienste an unterschiedlichen Orten existieren) oder die Auflösung eines WWW-UIS-Kartendienstes, welche die Dienstqualität beschreibt, könnten Beispiele für Diensteseigenschaften sein.

Der Einsatz von Tradern in großen, offenen verteilten Systemen erfordert ihre dezentrale Realisierung. Jeder Trader verwaltet innerhalb seiner Domäne die verfügbaren Dienstangebote und vermittelt diese an seine Klienten. Da die Angebotsvielfalt eines Traders auf seine Domäne eingeschränkt ist, besteht die Notwendigkeit einer Interaktion der Trader über ihre Domäne hinaus. DAIS bietet hierzu folgende Möglichkeiten an:

1. Konzept der *Context Federation*
2. Konzept der *Proxy Offers*

Bei beiden Arten der Interaktion exportieren die Trader untereinander bestimmte Teile ihres Dienstangebotes, und die Dienstangebote der Trader enthalten sowohl lokale als auch entfernte Angebote. Bei der *Context Federation* werden über einen sogenannten Master Trader mehrere Trader miteinander verbunden, wobei dies für den Klient transparent ist, er sich also stets nur an „seinen“ Trader wendet. Durch den Trader wird in diesem Fall mittels einer Zufallsstrategie irgendein ein passender Dienst ausgewählt (ein lokal verfügbarer Dienst hat allerdings Priorität). Mit den *Proxy Offers* ist das Verbinden mehrerer Trader über eigene, d.h. selbst implementierte Auswahlstrategien möglich, wobei dieses Verfahren allerdings etwas „unhandlich“ erscheint, jedoch große Flexibilität bei der Diensteauswahl ermöglicht.

4.4 Praktische Erprobung Trading

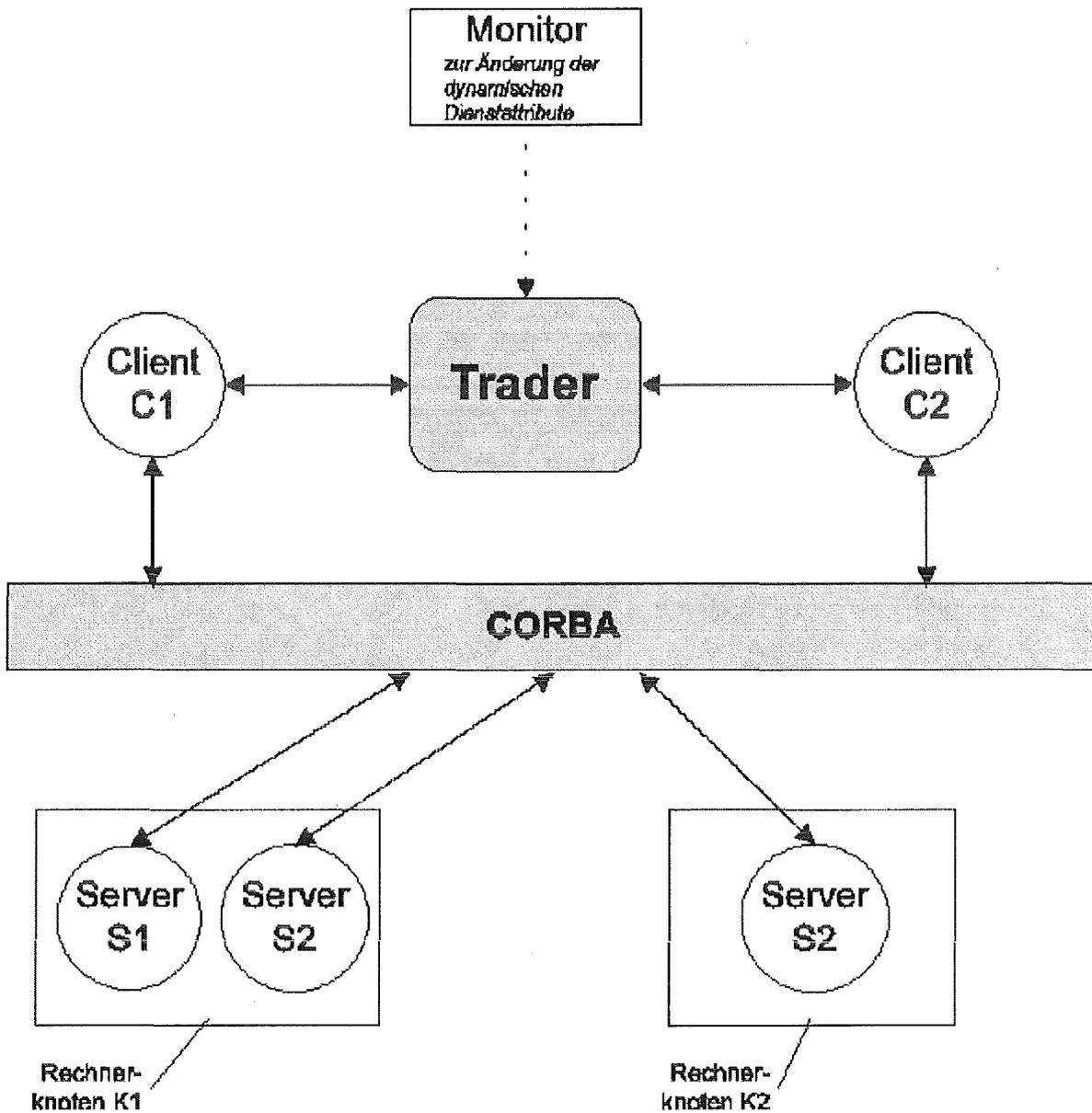


Abbildung 10: Szenario zur Erprobung des DAIS-Traders

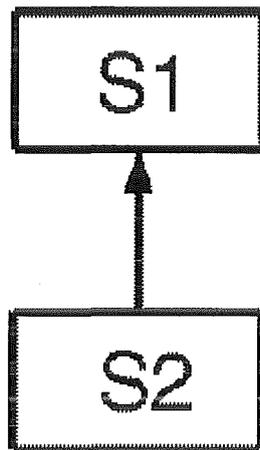


Abbildung 11: Konformitätshierarchie der Dienste der Beispielszenarien

Im Rahmen der praktischen Erprobung werden einige typische Szenarien für den Einsatz eines Traders entworfen, die dann im Rahmen einer Implementierung mit Hilfe des DAIS-Systems näher untersucht werden. Wie in Abbildung 10 ersichtlich, befinden sich auf Knoten 1 zwei Dienste S1 und S2, die sich dadurch unterscheiden, daß S2 eine Erweiterung des Dienstes S1 darstellt (vgl. Abbildung 11). Innerhalb des UDK-Dienstes im WWW-UIS könnte S1 beispielsweise die allgemeine Objektsuche und S2 die Objektdetailsuche darstellen. Auf Knoten 2 befindet sich eine identische Version des Dienstes S2.

Im Rahmen der Erprobung des Trading-Mechanismus werden die drei folgenden Szenarien implementiert und näher untersucht.

4.4.1 Szenario 1: statisches Trading

Die identischen Dienste S2 auf Knoten K1 und K2 sollen im Trader mit unterschiedlichen Dienstqualitäten registriert werden. Die Dienstqualität soll durch die Auflösung eines Kartendienstes dargestellt werden und soll innerhalb des Intervalles [1,100] jeden ganzzahligen Wert annehmen können. Der Dienstbenutzer C1 soll den Dienst S2 unter Vorgabe einer gewünschten Dienstqualität beim Trader anfordern.

Der Dienst S2 auf Knoten K1 wurde mit der Auflösung 50 beim Trader registriert, was einer mittleren Darstellungsgüte entspricht, wogegen der auf Knoten K2 ablaufende Dienst S2 mit der hohen Auflösung 80 registriert wurde. Ein Dienstbenutzer C1 forderte den Dienst S2 beim Trader an unter Vorgabe einer gewünschten Dienstqualität von mindestens 60. Der Trader wählte aus den beiden Diensten S2 denjenigen auf Knoten K2 aus, da dieser die geforderte Dienstqualität erbringen konnte.

Als Ergebnis ist festzuhalten, daß Szenario 1 ohne Probleme unter Verwendung des DAIS-Traders umgesetzt werden konnte.

4.4.2 Szenario 2: fehlertolerantes Trading

Im Rahmen einer Fehlertoleranzbetrachtung soll untersucht werden, wie sich der Trader beim Ausfall eines Dienstes bzw. Knotens verhält. Der Knoten K1 soll ausfallen und ein Klient fordert den Dienst S1 an. Es soll untersucht werden, wie der Trader auf diese Situation reagiert. Der Knoten K1 ist ausgefallen, und die darauf laufenden Dienste sind nicht verfügbar. Der Dienstbenutzer C1 fordert den Dienst S1, der beispielsweise die allgemeine Suche nach UDK-Objekten sein könnte, unter Vorgabe gewünschter Eigenschaften beim Trader an. Als Ergebnis soll er eine Liste mit den Objektreferenzen der Dienste erhalten, welche die geforderten Eigenschaften erfüllen.

Dieses Szenario konnte ebenso problemlos implementiert werden. In unserem Beispiel enthielt die Liste die Dienste S1, S2 auf Knoten K1 und S2 auf Knoten K2. Bevor der Klient einen dieser Dienste beauftragen konnte, mußte er selbst überprüfen, ob der jeweilige Dienst verfügbar war. Der verwendete DAIS-Trader überprüfte nicht die Verfügbarkeit der von ihm gelieferten Dienste. Der im Klient erforderliche Überprüfungsaufwand ist auf den DAIS-Trader zurückzuführen und stellt keine Schwäche des Trading-Prinzips dar.

4.4.3 Szenario 3: dynamisches Trading

Die Eigenschaften der im Trader registrierten Dienste sollen dynamisch verändert werden, um so einen wechselnden Systemzustand zu simulieren. Ein UIS-typisches Anwendungsbeispiel hierfür sind Kartendienste, deren Auflösung in Abhängigkeit von der Zahl der vorliegenden Anforderungen verändert wird, um die Wartezeit für die einzelnen Aufträge möglichst gering zu halten. Der Dienstbenutzer C1 soll mehrmals den Dienst S1 anfordern unter Vorgabe derselben Eigenschaften. Hierbei wird das Verhalten des Traders untersucht.

Da im Rahmen der Administrationswerkzeuge des verwendeten DAIS-Systems keine Unterstützung zur dynamischen Änderung der Diensteigenschaften geboten wurde, mußten diese durch ein eigens implementiertes Monitorprogramm (vgl. Abbildung 10) unter Verwendung von API-Funktionen geändert werden. Das Monitorprogramm wurde zyklisch aufgerufen, und es konnte beobachtet werden, wie der Klient C1 jeweils von wechselnden Servern bedient wurde.

Dieses Szenario konnte ebenso problemlos implementiert werden. Die Tatsache, daß zur Durchführung dynamischer Änderungen der registrierten Dienste ein Monitorprogramm selbst zu implementieren war, ist durch den derzeitigen Aufbau des DAIS-Trader bedingt und stellt wie im vorherigen Szenario keine Schwäche des Trading-Prinzips an sich dar.

4.5 Resümee

Der betrachtete Abschnitt 4 gibt eine kurze Einführung in den Trading-Dienst in verteilten Systemen. Anhand der im DAIS-System integrierten Traderkomponente wurde exemplarisch ein Trading-Dienst in einigen ausgewählten Szenarien prototypisch erprobt. Hierbei standen die Einsatzmöglichkeiten des Trading zur Bewältigung typischer Probleme in verteilten Systemen, wie Versionsvielfalt der vorhandenen Dienste, Fehlertoleranz beim Ausfall einzelner Dienste und dynamisch wechselnde Dienstattribute im Vordergrund.

Das Ergebnis der Tests war, daß der Trader-Mechanismus eine sinnvolle und nützliche Funktion zur Lösung dieser Probleme darstellt. Die Bedeutung eines Trading-Dienstes steht in engem Zusammenhang mit der Notwendigkeit in großen verteilten Systemen unterstützende Dienste zur Verwaltung von Klienten und Server bereitzustellen. Wegen seiner flexiblen Konzeption ist der Trading-Dienst ein wichtiges Werkzeug für eine Anwendung, um in Abhängigkeit ihrer Dienstanforderungen einen geeigneten Server auszuwählen.

Noch nicht alle Wünsche erfüllen konnte allerdings die hier speziell verwendete DAIS-Trader Implementierung. Zum einen war das in den Versuchen festgestellte Zeitverhalten des bei der Implementierung eingesetzten DAIS-Traders nicht immer ganz zufriedenstellend, zum anderen sind Funktionalitätserweiterungen, die dynamisches Trading noch besser unterstützen wünschenswert. Jedoch zeigen bereits die Möglichkeiten des DAIS-Trader die sinnvolle Einsetzbarkeit des Trading-Konzepts.

Generell ist davon auszugehen, daß der Trading-Dienst künftig eine immer wichtigere Rolle im Entwurf und in der Planung verteilter Anwendungen findet und sich als neuartiger Basismechanismus verteilter Systeme etablieren wird. Darauf deutet auch hin, daß zur Zeit im Rahmen der Standardisierungsbestrebungen der OMG ein Trader-Dienst als CORBAservice in die CORBA-Architektur integriert wird.

5 Leistungsmessungen

5.1 Überblick

Da CORBA als Kommunikationstechnologie für verteilte Anwendungen eingesetzt wird, zielen die in diesem Abschnitt zusammengefaßten (groben) Leistungsmessungen vornehmlich darauf ab, einen Überblick zu erhalten, wie feingranular die über CORBA abgewickelte Kommunikation sein kann. Zunächst werden daher in Abschnitt 5.2 ORB-Implementierungen anhand einiger Elementaroperationen betrachtet. Im folgenden Abschnitt 5.3 werden diese Zeiten in Relation zu den Gesamtlaufzeiten einer über ORBs kommunizierenden Anwendung - konkret Teilen aus WWW-UIS - gesetzt.

5.2 Messungen Elementaroperationen

Die folgende Tabelle stellt Leistungsmessungen zu einigen mit den betrachteten ORB-Implementierungen ausgeführten Operationsaufrufen dar. Ziel der Messungen war keine detailliert aufgeschlüsselte Leistungsanalyse der Systeme, sondern eine Aussage über die Größenordnung des Kommunikationsaufwands mittels eines ORB. Als Meßanwendung wurde ein einfacher Benchmark implementiert, der sich an Messungen orientiert, wie sie auch in Produktbeschreibungen häufig zu finden sind [IO95b]. Die Implementierung wurde möglichst direkt anhand der Produktdokumentation und in der Produktauslieferung enthaltener Programmbeispiele entwickelt. Sie erhebt daher keinen Anspruch auf Optimalität.

Die gemessenen Operationen unterscheiden sich einmal dahingehend, ob der Aufruf synchron oder asynchron ausgeführt wird. Des weiteren unterscheiden sich die übergebenen Parameter sowohl in der Komplexität ihrer Struktur wie auch in ihrem Umfang:

- Im synchronen Fall erhält der Aufrufer ein Operationsergebnis und die Ausgabeparameter der Operation, asynchron sind in CORBA nur Aufrufe von Operationen möglich, die weder einen Ergebniswert (`void`) noch Ausgabeparameter aufweisen.
- Betrachtet werden die Fälle kein Parameter, einige Parameter mit Basistypen, dieselben Parameter übergeben eingepackt in einen Verbund, ein Feld dieser Verbunde und schließlich ein kurzes und ein langes Feld aus Ganzzahlen.
- In den mit (*) markierten Fällen werden die Argumente jeweils als `in`-, `out`- und `inout`-Parameter (vgl. Abschnitt 3.2) übertragen, sonst nur als `in`-Parameter. Im (*)-Fall wird daher die 5-fache Datenmenge übertragen (request: 3-fach, reply: 2-fach).

Als Meßergebnisse dargestellt werden jeweils die Ausführungszeit in Millisekunden, wie auch reziprok die pro Sekunde möglichen Operationsaufrufe.² Aufgrund einer Spracheinschränkung konnten für ObjectBroker keine Verbundfelder gemessen werden (vgl. Abschnitt 3.2). Aus organisatorischen Gründen standen für ObjectBroker nur Sparc 2-Rechner zur Verfügung.

Ergebnisse der Messungen sind zum einen in der folgenden Tabelle 3 wie auch graphisch in Abbildung 12 und Abbildung 13 dargestellt. Der hellere Teil der Meßbalken gilt dabei jeweils für den asynchronen, der dunklere Teil für den synchronen Fall.

² Gemessen wurde SunOS 4.1.3 und SunOS 5.4 unter Benutzung des Übersetzers GNU gcc 2.6.3. Die gemessenen Programme wurden optimiert übersetzt (-O3). Die Meßplattformen sind über ein Standard-Ethernet verbunden. Während des Meßvorgangs wurde im Netz der normale Rechenbetrieb nicht eingeschränkt; auf den Meßrechnern waren ressourcenintensive Prozesse nicht vorhanden oder zumindest nicht aktiv.

		asynchron (oneway)						synchron (request & reply)						
		keine	Skalare	Verbund	Verbund-Feld [100]	long-Feld [10]	long-Feld [1000]	keine	Skalare (*)	Verbund (*)	Verbund-Feld (*) [100]	long-Feld [10]	long-Feld [1000]	
[Bytes]		[0]	[25]	[25]	[2500]	[40]	[4000]	[0]	[125]	[125]	[12500]	[40]	[4000]	
Orbix	Sparc 2	[ms]	2	2	2	12	1	7	4	5	5	36	4	7
	(lokal; OS4)	[1/s]	690	570	490	84	820	150	250	210	210	27,8	250	150
	Sparc 2	[ms]	1	2	1	12	1	10	5	6	6	54	5	12
	(Netz; OS4)	[1/s]	1500	690	710	87	1300	110	220	170	180	19	220	81
	Sparc 20	[ms]	1	1	1	4	1	3	2	3	3	17	3	4
	(lokal; OS4)	[1/s]	1700	870	1100	260	2000	310	450	390	380	59,2	390	270
	Sparc 20	[ms]	0,5	0,5	0,5	3	0,5	2	2	2	2	13	2	2
	(lokal; OS5)	[1/s]	5000	3500	3700	300	3500	570	630	480	550	75,2	670	470
Object-	Sparc 2	[ms]	61	64	62	¾	57	86	71	73	71	¾	69	82
	(lokal; OS4)	[1/s]	16	16	16	¾	18	12	14	14	14	¾	73	71
Broker	Sparc 2	[ms]	50	54	51	¾	46	78	77	78	77	¾	74	93
	(Netz; OS4)	[1/s]	20	19	20	¾	22	13	13	13	13	¾	14	11

Tabelle 3: Leistungsmessung ORB-Elementaroperationen

Die Ergebnisse zeigen eine deutliche Abhängigkeit vom übertragenen Datenumfang und von der Komplexität der Datenstrukturen. Dominierend ist allerdings die grundsätzliche Abweichung zwischen Orbix und ObjectBroker, die nach unserer Vermutung maßgeblich in der in Abschnitt 3.6 skizzierten Architektur begründet ist, nach der die Kommunikation bei Orbix direkt zwischen Klient und Server stattfindet, während bei ObjectBroker in jedem Fall ein weiterer zentraler Prozeß involviert ist.

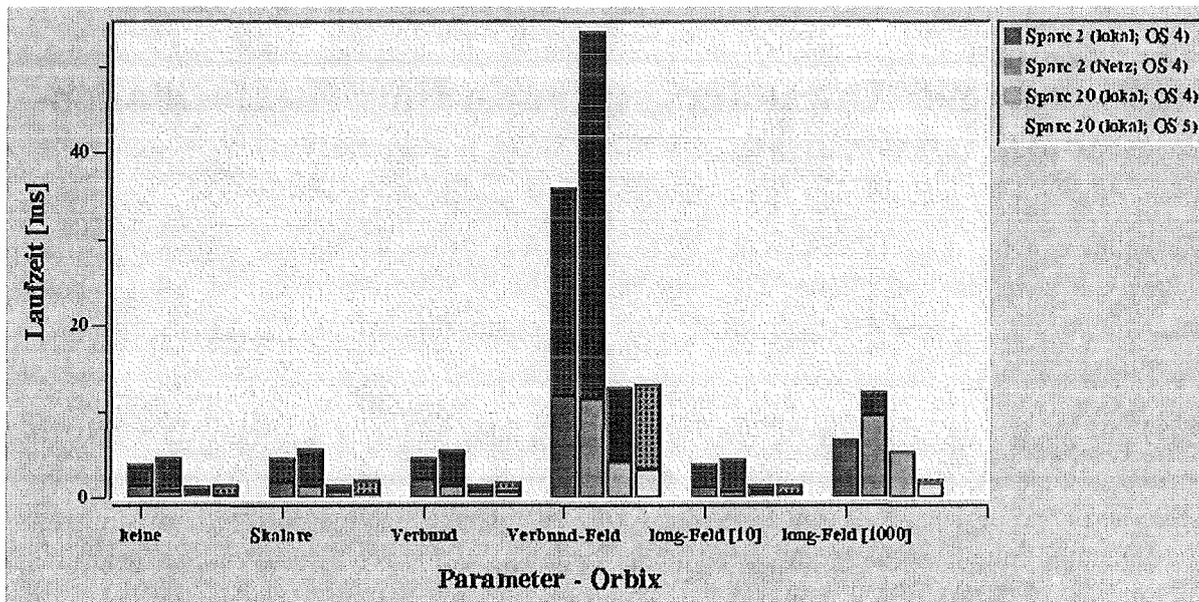


Abbildung 12: Graphik Meßergebnisse - Orbix

Die Laufzeiten bleiben bei Orbix selbst im ungünstigen Fall deutlich unterhalb der Schwelle, die für Operationen gilt, die Nutzer nach einigem Überlegen ausführen, wie z.B. das Abschicken eines Formulars. Im lokalen Fall sind die Laufzeiten selbst für die Übermittlung feingranularer Ereignisse aus Benutzungsschnittstellen ausreichend.

Diese günstige Einschätzung gilt um so mehr, als davon ausgegangen werden kann, daß mit umfangreicher Datenübertragung verbundene Operationsaufrufe in der Regel umfangreiche/langwierige Operationen ausführen, so daß dann der Aufwand zur Kommunikation über einen ORB von der eigentlichen Operationslaufzeit dominiert wird.

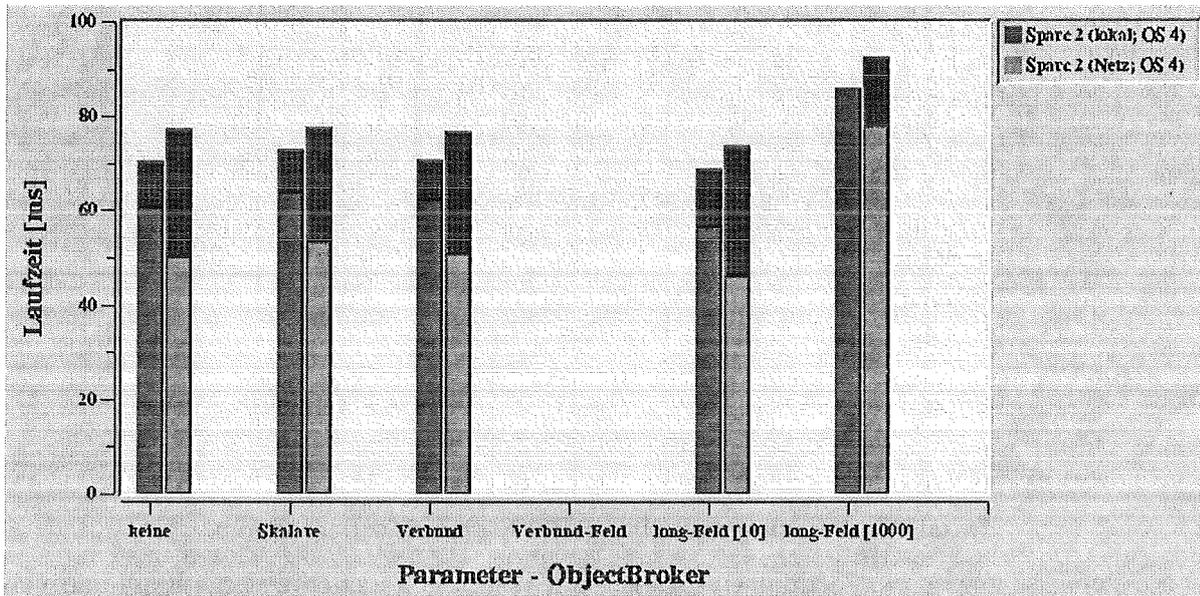


Abbildung 13: Graphik Meßergebnisse - ObjectBroker

Im Falle ObjectBroker ist diese Einschätzung weiterhin gültig. Wie die folgenden Abschnitte zeigen werden, dominieren die Ausführungszeiten der eigentlichen Anwendungsfunktionen im betrachteten Szenario die Kommunikationskosten über einen ORB deutlich.

5.3 Messungen Anwendungsszenario Umweltinformationssystem aus WWW-UIS

Da viele Elemente eines UIS nicht überall lokal zur Verfügung stehen und auch nicht in allen Netzen ein quasi unkontrollierter Zugang (wie z.B. bei WWW) erwünscht ist, müssen solche Systeme auf eine andere Art zugänglich gemacht werden. Eine Möglichkeit hierzu ist die Integration mittels CORBA.

Die Integration vorhandener UIS-Teile mit CORBA läßt sich, wie am Beispiel WWW-UIS gezeigt, auf einfache Weise durchführen. Man beschreibt alle Funktionen, die das vorhandene System zur Verfügung stellt, mit Hilfe der CORBA eigenen Schnittstellenbeschreibungssprache IDL, und integriert dann die vorhandenen Teile in die aus der Schnittstellenbeschreibung generierten (CORBA)-Programmrümpfe. Auf diese Weise hat man einen CORBA-Server erstellt, der die in der Schnittstellenbeschreibung festgelegten Funktionen zur Verfügung stellt. Die Implementierung eines CORBA-Klients ist genauso einfach. Man programmiert einen Klient wie sonst auch und ruft die Funktionen des Servers wie lokale Funktionen auf. Einziger Unterschied ist, daß bevor man eine Funktion eines Servers aufrufen kann, eine Verbindung zwischen Klient und Server hergestellt werden muß. Dies geschieht mittels eines einfachen Funktionsaufrufes: `_bind`. Das Lokalisieren eines passenden Servers wird durch CORBA übernommen, wodurch eine beliebige Verteilung von Klienten und Servern im Netz

möglich wird. Da verteilte Programme mehr Fehlerquellen (z.B. Netzwerkfehler) ausgesetzt sind, als Programme, die im gleichen Adreßraum ausgeführt werden, und ein Fehler in einem anderen Adreßraum auch nicht erkannt wird, bietet CORBA dem Klient-Programmierer die Möglichkeit, sich über den „Zustand“ eines Servers zu informieren. Dies gibt dem Programmierer die Möglichkeit, auf eventuelle Probleme, die aufgrund der Verteilung entstanden sind, angemessen zu reagieren.

Von uns wurden auf diese Weise die WWW-UIS Funktionen zur allgemeinen Objektsuche, allgemeinen Objektdarstellung und zur detaillierten Objektdarstellung des UDK implementiert. Ferner wurde die Kurzausgabe der GRDB implementiert. Jedes der Systeme UDK und GRDB wurde auf zwei Server aufgeteilt, einen für die HTML-Aufbereitung und einen für die Datenbankabfragen. Implementierungsgrundlage war Orbix.

Da durch eine Klient-Server Architektur ein größerer Datenaustausch notwendig ist, wurde ein direkter Vergleich der beiden Implementierungen durchgeführt. Da für einen Anwender nur der Unterschied zwischen einzelnen Abfragen relevant ist, wurden die Messungen mit Benutzung des WWW-Servers - also aus realer Anwendersicht - durchgeführt, d.h. die Zeitaussagen sind entsprechend grob.

Generelles Fazit ist allerdings, daß die zeitlichen Unterschiede beim Zugriff mittels CORBA - jedenfalls bei der von uns gewählten Einbindung der WWW-UIS Funktionen - gegenüber der bisherige in WWW-UIS genutzten Art gering waren. Der größte Teil der Antwortzeit wird für die Datenbankabfragen benötigt.

Programm	UDK	UDK	UDK	CORBA	CORBA	CORBA
	max	min	avg	max	min	avg
Test 1	25 s	4 s	11 s	21 s	5 s	11 s
Test 2	65 s	31 s	44 s	65 s	31 s	45 s
Test 3	20 s	10 s	12 s	21 s	10 s	12 s

Test 1: Das Aufrufen der Funktion „Objekte allgemeine Suche“.

Test 2: In der darauffolgenden Seiten wurde die Funktion „Objektliste“ mit dem Suchstring „b“ in allen Katalogen durchgeführt. Der Suchmodus war „Teilwort“ und „Insensitiv“. Das Ergebnis waren 611 Einträge.

Test 3: Wie Test 2 mit dem Unterschied, daß nur im Katalog „GER BW UDK Primärkatalog“ gesucht wurde und der Suchstring „boden“ gewählt wurde. Das Ergebnis waren 93 Einträge.

Tabelle 4: Anwendungsnahe Zeitmessungen

Allgemein läßt sich sagen, daß die Integration von UIS-Funktionen in der von uns durchgeführten Art und Weise nach einer gewissen Einarbeitungszeit einfach durchzuführen ist. Auch entstehen durch die Klient/Server Architektur keine ins Gewicht fallenden zusätzlichen Wartezeiten.

Die Tests liefen alle unter SUN-OS im Netz des FZI ab. Hierbei wurde der WWW-Browser Netscape in der Version 1.1 eingesetzt. Für die Tests mit CORBA wurden permanente Server benutzt, die nicht mit dem Klient zusammen auf einer Maschine liefen. Gemessen wurde die

Zeit vom Abschicken der Anforderung bis im Browser „*Document done*“ angezeigt wurde. Alle Tests wurden 20 Mal durchgeführt.

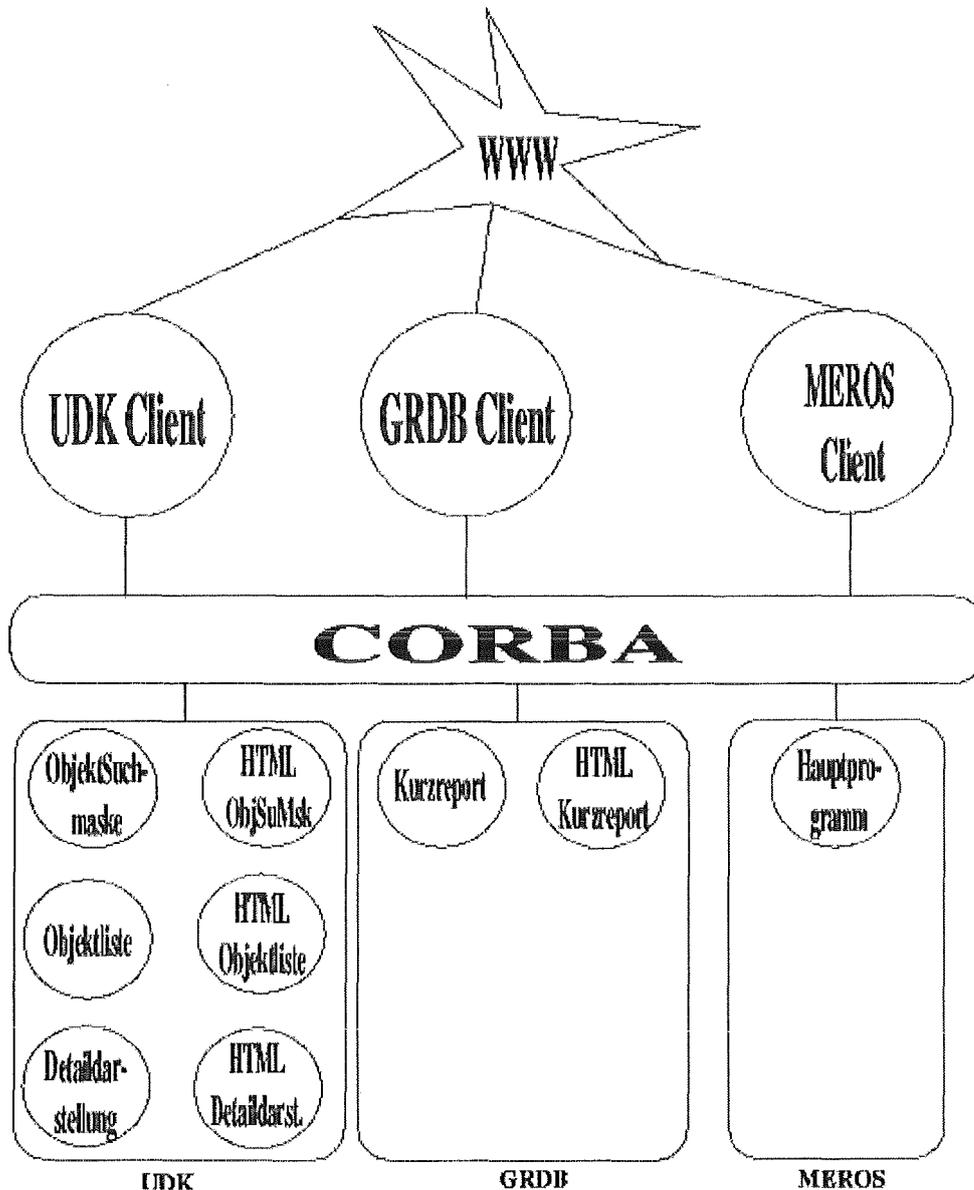


Abbildung 14: Testszenario: Integration einiger WWW-UIS Dienste mittels CORBA

5.4 Resümee

Thema dieses Abschnitts waren (grobe) Leistungsmessungen in CORBA-Implementierungen, einerseits anhand einfacher Objektaufrufe, andererseits anhand von Anwendungsbeispielen aus WWW-UIS. Wesentliches Fazit der Messungen ist, daß in den betrachteten CORBA-Implementierungen ObjectBroker und Orbix, selbst bei relativ feingranularen Objektaufrufen, keine größere Leistungseinschränkung auftritt, Orbix jedoch gegenüber ObjectBroker architekturbedingt die etwas bessere Leistung zeigt. Wird CORBA zudem in der UIS-typischen, WWW-basierten Gesamtarchitektur eingesetzt, so sind die Leistungsunterschiede für den Endanwender nahezu bedeutungslos.

6 Zusammenfassung und Ausblick

6.1 Wesentliche Ergebnisse

Eine der Hauptaufgaben des FZI im Projekt GLOBUS II waren die praktische Erprobung sowie Bewertung von aktuell am Markt verfügbaren Implementierungen des CORBA-Standards im Hinblick auf ihren Einsatz im Umweltinformationssystem Baden-Württembergs. Zu diesem Zweck wurde zunächst eine Marktübersicht der verfügbaren Implementierungen erstellt. Diese ergab, daß aktuell ca. 15 „echte“ Implementierungen verfügbar sind. Für den Einsatz im UIS müssen die Systeme eine Reihe von Anforderungen erfüllen. Anhand der aufgestellten priorisierten Anforderungsliste wurden die Systeme Orbix von IONA Technology und ObjectBroker von DEC für die praktische Erprobung der Kernfunktionalität ausgewählt.

Beide Produkte implementieren CORBA 1.2. Ferner umfassen sie bereits - jeweils unterschiedliche - Funktionalitäten des zwischenzeitlich verabschiedeten Standards 2.0 bzw. der CORBAServices. Der ObjectBroker weist insgesamt umfangreichere, allerdings teilweise nicht OMG konforme Funktionen auf. Dies bedingt eine umfangreichere Einarbeitung. Aufgrund der systemnäheren Programmierung birgt es ferner die Gefahr einer größeren Abhängigkeit vom eingesetzten Produkt. Demgegenüber konzentriert sich Orbix auf die Kernbereiche des CORBA-Standards. Statt der Programmiersprache C wie beim ObjectBroker wird C++ zur Verfügung gestellt. Dies vereinfacht die Benutzung. Insgesamt wird ein etwas geringerer Funktionsumfang zur Verfügung gestellt.

Als Produkt, das eine für das UIS Baden-Württembergs besonders wichtig erscheinende Funktionalität zur Verfügung stellt, wurde der Trader des DAIS-Systems von ICL anhand mehrerer Szenarien praktisch erprobt. Dabei erwies sich die Realisierung der benötigten Funktionalität als zum Teil noch etwas umständlich und daher verbesserungsfähig. Die Umsetzung der gewählten Szenarien erwies sich jedoch insgesamt als problemlos. Die Funktionalität des Trading insgesamt stellt eine für das UIS sinnvoll einsetzbare Funktionalität dar, deren praktische Umsetzung durch CORBA-Implementierungen unterstützt wird.

Wesentliches Fazit der durchgeführten (groben) Leistungsmessungen von Orbix und ObjectBroker ist, daß die verwendeten CORBA-Implementierungen effizient arbeiten, wobei Orbix etwas schneller als ObjectBroker ist. Für den Endanwender ergaben sich in der UIS-typischen WWW-Umgebung bei der gewählten Art der Datenbankanbindungen in den gemessenen Datenbankanfragen keine merkbaren Leistungseinbußen.

6.2 Empfehlungen für das weitere Vorgehen

Zum Zeitpunkt der Produktauswahl war CORBA 1.2 die aktuell in den Produkten implementierte Version des Standards. Inzwischen werden auch auf CORBA 2.0 basierende Implementierungen zunehmend verfügbar. Von daher stellen die praktische Erprobung sowie Bewertung der Systeme eher eine Momentaufnahme denn ein endgültiges Ergebnis dar. Ein abschließende Empfehlung zugunsten eines bestimmten Systems kann daher noch nicht sinnvoll gegeben werden. Insbesondere um der Gefahr einer Abhängigkeit von einem einzelnen Soft-

warehersteller zu entgehen, empfehlen wir, die mit CORBA 2.0 standardisierte Interoperabilität unterschiedlicher CORBA-Implementierungen anhand praktischer Szenarien und Anwendungen aus dem UIS Baden-Württemberg praktisch zu erproben. Besonderes Augenmerk ist hierbei auf die Integration der vorhandenen Softwarelasten unter proprietären Betriebssystemen wie beispielsweise DEC VMS zu legen, um so eine kontrollierte schrittweise Migration in Richtung offener Systeme vornehmen zu können.

Literatur

- [ANS91]
ANSAware 3.0 *Implementation Manual*. Document RM.097.01. February 1991.
- [DEC94a]
Digital Equipment Corp. *ObjectBroker Reference Manual*. ObjectBroker Version 2.5, Aug. 1994.
- [DEC94b]
Digital Equipment Corp. *System Integrator's Guide*. ObjectBroker Version 2.5, Aug. 1994.
- [DEC94c]
Digital Equipment Corp. *Overview and Glossary*. ObjectBroker Version 2.5, August 1994.
- [DEC94d]
Digital Equipment Corp. *Release Notes*. ObjectBroker Version 2.5A, November 1994.
- [ES90]
M.A. Ellis und B. Stroustrup. *The Annotated C++ Reference Manual*. Addison-Wesley, Reading, MA, 1990.
- [FZI95]
Günter v. Bültzingsloewen, Arne Koschel, Ralf Kramer, Rainer Schmidt, Dietmar Theobald. *CORBA-Evaluierung für das UIS Baden Württemberg*. Forschungszentrum Informatik (FZI). Karlsruhe. September 1995.
- [IO95a]
IONA Technologies Ltd. *Programmer's Guide*. release 1.3. Irland, April 1995.
- [IO95b]
IONA Technologies Ltd. *The Orbix Architecture*. Irland, 1995.
- [KL95]
Arne Koschel und Philipp Leibfried. *Die CORBA-Architektur in der Anwendung*. In 6. Kolloquium Software-Entwicklung: Methoden, Werkzeuge, Erfahrungen. Technische Akademie Esslingen, September 1995.
- [KL96]
Arne Koschel und Philipp Leibfried. *Experiences with the CORBA Implementation DSOM*. In International Conference on Distributed Platforms (ICDP). Universität Dresden, Februar 1996.
- [OMG90]
R.M. Soley (ed.). *Object management architecture guide*. OMG TC Document 90.9.1, Object Management Group, Inc., Nov. 1990. Version 1.0.
- [OMG93]
Object Management Group. *The Common Object Request Broker: Architecture and Specification*. OMG Document 93-12-43, Object Management Group, Inc., edition 1.2. Dezember 1993.
- [OMG94a]
Object Management Group. *CORBA Products Directory*. OMG TC Document 94-4-17, , Object Management Group, Inc., April 1994.

[OMG95a]

Object Management Group. *The Common Object Request Broker: Architecture and Specification*. Object Management Group, Inc., edition 2.0, Juli 1995.

[OMG95b]

Object Management Group. *CORBA services: Common object services specification*. OMG Document 95-3-31, Object Management Group, Inc., März 1995, revised edition.

[OMG95c]

Object Management Group. *CORBA Products Directory Addendum*. OMG TC Document 95-4-3. Object Management Group, Inc., April 1995.

[TS94]

Fritz Schmidt und Manfred Tischendorf. *INSERV - Spezifikation von Diensten im UIS*. Technical Report IKE 4 - 135 - 3, IKE, Universität Stuttgart. April 1994.

Management verteilter Umweltinformationsquellen

*Margit Gaul, Liljan Kadric, Gerhard Klügl, Thomas Krist, Wolf-Fritz Riekert, Gerlinde Wiest
Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW)
an der Universität Ulm
Helmholtzstr. 16, D-89081 Ulm
Postfach 2060, D-89010 Ulm*

1. MANAGEMENT VERTEILTER INFORMATIONSMARKT UIS BADEN-WÜRTTEMBERG – KONZEPTION UND PROTOTYPISCHE REALISIERUNG	96
1.1 EINFÜHRUNG	96
1.2 INFORMATIONSMARKT UIS	97
1.3 INFORMATIONSMARKT UIS	99
1.3.1 Daten als Informationsquellen	99
1.3.2 Methoden als Informationsquellen	99
1.3.3 Multimediale Dokumente als Informationsquellen	100
1.3.4 Dienste zur Erschließung von Informationsquellen	100
1.4 MANAGEMENT VON INFORMATIONSMARKT UIS IM WWW	100
1.4.1 Daten im WWW	102
1.4.2 Methoden im WWW	102
1.4.3 Multimediale Dokumente im WWW	103
1.5 PROTOTYPISCHE REALISIERUNG DES INFORMATIONSMARKT UISMANAGEMENTS	103
1.5.1 Unterstützung der Verbreitung und der Nutzung von Informationen aus Umweltdatenbanken über WWW	104
1.5.2 Bereitstellung von Berichten und anderen multimedialen Dokumenten über die Umwelt mittels WWW	105
1.5.3 Unterstützung des Retrievals von Umweltinformationen	107
1.5.4 Unterstützung der Erstellung von Berichten	109
1.6 AUSBLICK	110
2. DER UMWELTINFORMATIONSMARKT UIS KATALOG	110
2.1 ALLGEMEINE BESCHREIBUNG	110
2.2 DIE SYSTEMARCHITEKTUR	114
2.2.1 Überblick	114
2.2.2 Der Umweltdatenkatalog (UDK)	117
2.2.3 Das Datenbankschema des Umweltdatenkatalogs	118
2.2.3.1 Informationsquellenverwaltung	119
2.2.3.2 Benutzer- und Rechteverwaltung	122
2.2.3.3 Thesaurus	124
2.2.3.4 Gazetteer	125
2.3 DIALOGFÜHRUNG	130
2.3.1 Überblick	130
2.3.2 Deskriptor	134
2.3.3 Raumbezug	138
2.3.4 Zeitbezug	143
2.4 FUNKTIONALITÄTEN DES UMWELTINFORMATIONSMARKT UIS	146
2.4.1 Bericht eintragen	146
2.4.1.1 Überblick	146
2.4.1.2 „Bericht eintragen“ aus Benutzersicht	147
2.4.1.3 Programmablauf von „Bericht eintragen“	150
2.4.2 Dienst eintragen	155
2.4.2.1 „Dienst eintragen“ aus Benutzersicht	155
2.4.2.2 Programmablauf von „Dienst eintragen“	158
2.4.3 UMWELTINFORMATIONSMARKT UIS-Objekt suchen	159
2.4.3.1 „Umweltinformationsmarkts-Objekt suchen“ aus Benutzersicht	159
2.4.3.2 Programmablauf von „Umweltinformationsmarkts-Objekt suchen“	163
3. FAW-BASISTOOLS	165
3.1 WEBQUERY	165
3.1.1 WebQuery aus Benutzersicht	165
3.1.2 WebQuery aus der Sicht eines Datenbank-Diensteanbieters	168
3.1.3 Ablauf von WebQuery	174
3.2 SESSIONMANAGEMENT	178
3.2.1 Sessionmanager	179
3.2.1.1 Ablauf einer Session	180
3.2.1.2 Benutzung des Sessionmanagers	181
3.2.1.3 Zugangskontrolle zum Sessionmanager und den Diensten	182

3.2.1.4 Einrichten eines neuen Dienstes.....	183
3.2.2 <i>Sessiondaemon</i>	184
3.2.3 <i>Dokumentation der Programme</i>	185
3.2.3.1 Übersetzung der Programme.....	185
3.2.3.2 Aufbau von sessionmanager.....	185
3.2.3.3 Aufbau von newuser	187
3.2.3.4 Aufbau von sessiondaemon.....	189
LITERATUR	191

1. Management verteilter Informationsquellen im UIS Baden-Württemberg – Konzeption und prototypische Realisierung

Dieses Kapitel ist das erste von drei Kapiteln, in denen die Ergebnisse des vom FAW Ulm bearbeiteten Teilprojekts im Verbundprojekt GLOBUS II (Konzeption und prototypische Realisierung einer aktiven Auskunftskomponente für globale Umwelt-Sachdaten im Umweltinformationssystem Baden-Württemberg) beschrieben sind. Gegenstand dieses Kapitels ist die Konzeption eines Managements für verteilte Informationsquellen im UIS Baden-Württemberg als Beitrag zur Weiterentwicklung der UIS-Systemarchitektur im Rahmen des Landessystemkonzepts. Der Begriff Informationsquelle im UIS ist dabei recht weit gefaßt und umfaßt Daten, Methoden und multimediale Dokumente (z.B. Berichte). Es wird ein Modell für den Austausch von Umweltinformationen präsentiert, dem die Metapher eines elektronischen Informationsmarkts zugrundeliegt.

Anlaß für die Erstellung der Dokumentation in diesem Kapitel ist der UIS-Workshop vom 15. Mai 1995 in Stuttgart, auf dem das FAW vom Umweltministerium aufgefordert wurde, eine Konzeption für das Management verteilter Informationsressourcen im UIS vorzulegen. Die hier vorgestellte Konzeption wurde im Rahmen der anwendungsorientierten UIS-Forschungsvorhaben INTEGRAL (Integration heterogener Komponenten des Umweltinformationssystems Baden-Württemberg) und GLOBUS in den Jahren 1993 – 1995 am FAW entwickelt und in wesentlichen Teilen am Beispiel verschiedener UIS-Komponenten prototypisch realisiert.

Das diesem Kapitel folgende Kapitel 2 beschreibt die Realisierung des Umwelt-Informationsquellenkatalogs, der im Rahmen des Projekts GLOBUS II unter Nutzung von Synergien mit dem Projekt INTEGRAL III entwickelt wurde. Dieser Katalog bildet das Kernstück für das hier konzipierte Management von verteilten Informationsquellen im UIS Baden-Württemberg.

Die Dokumentation der Beiträge des FAW zum Projekt GLOBUS schließt ab mit dem Kapitel 3, in dem zwei am FAW entwickelte generische Tools zur Konstruktion von Informationsservern in Weitverkehrsnetzen beschrieben sind, die bei der Realisierung des Umwelt-Informationsquellenkatalogs genutzt wurden.

1.1 Einführung

Im UIS Baden-Württemberg wurden seit den Anfängen im Jahr 1987 inhaltlich und technisch weit fortgeschrittene Informationssysteme entwickelt, die inzwischen die Tauglichkeit für den praktischen Einsatz bewiesen haben.

Dennoch müssen die Akzeptanz und die Durchdringung informationstechnischer Lösungen in der Umweltverwaltung weiterhin verbessert werden. So bevorzugen beispielsweise auch heute noch viele Anwender analoge Lösungen (z.B. gedruckte oder gezeichnete Landkarten) gegen-

über digitalen Informationsbeständen (z.B. geographische Daten). Ziel muß es sein, mit dem UIS einen Attraktor für die Mitarbeiter in der Umweltverwaltung zu schaffen, der emotional positiv besetzt ist und allen Beteiligten Vorteile bringt.

Wesentliche Aufgabe ist hierbei die benutzerorientierte Bereitstellung informationstechnischer Unterstützung für das kooperative Arbeiten in der Umweltverwaltung. Die Zielsetzungen des UIS-Informationsmanagements, nämlich Qualität, Transparenz, Verknüpfbarkeit und Zugreifbarkeit der Informationen sind hierfür entscheidende Voraussetzungen.

Der Schatz des UIS besteht in dem vorhandenen großen Bestand an Informationen und Wissen, der in den einzelnen Dienststellen aufgebaut wurde und der zunehmend in Form von Daten, Methoden und multimedialen Dokumenten auf Computersystemen vorgehalten wird. Vereinfacht gesprochen geht es jetzt darum, den Schatz dieser Informationsbestände an den (PC-)Arbeitsplatz eines jeden Sachbearbeiters in der Umweltverwaltung zu bringen, soweit dies für die Erledigung der Dienstaufgaben förderlich ist. Dieses Ziel konnte mit der bisher verfügbaren Informationstechnik noch weitgehend nicht verwirklicht werden; vielmehr waren durch die in der Vergangenheit vorherrschenden Rahmenbedingungen bei der Entwicklung von UIS-Komponenten (proprietäre Hard- und Software-Plattformen und restriktives Netzwerkmanagement) Insellösungen entstanden, die in aller Regel nur lokal nutzbar waren. Mit der zunehmenden Verbreitung und Perfektionierung offener Vernetzungskonzepte sowie der wachsenden Leistungsfähigkeit moderner Weitverkehrsnetze haben sich die Voraussetzungen positiv verändert.

Ein frühes Ergebnis aus dem Projekt INTEGRAL war die Erkenntnis, daß es mit Hilfe moderner Telematiktechniken, also mit Hilfe einer Kombination aus Kommunikationstechnologie und Informatik möglich ist, die vorhandenen Insellösungen miteinander zu integrieren und zu einem verteilten Informationsverbund zusammenzuschließen. Die hierbei angewandten Vernetzungskonzepte, insbesondere Internet / World Wide Web besitzen auf dem Consumer-Markt eine zunehmende Anziehungskraft. Es besteht guter Grund zur Annahme, daß ihr Einsatz die Attraktivität des UIS und damit dessen Effizienz weiter steigern wird.

Mit der hier beschriebenen und in INTEGRAL und GLOBUS prototypisch realisierten Konzeption wird eine möglichst wirtschaftliche Nutzung der vorhandenen Hard- und Software-Ressourcen bezweckt. Dabei liegt der Vorteil in der leichteren Verfügbarkeit von Informationsquellen, die über das Netz angeboten werden. Diese Informationsquellen – Daten, Dienstprogramme, Dokumente – müssen nicht lokal vorgehalten und auch nicht lokal gepflegt werden. Daraus resultiert eine Minimierung der Hardware- / Software-Anforderungen und des Pflegeaufwandes für die einzelnen Rechnerarbeitsplätze.

1.2 Informationsmarkt UIS

Die hier präsentierte Konzeption für ein Management von Informationsquellen orientiert sich an einem Anbieter-Nutzer-orientierten Anwendermodell. Die zugrundeliegende Metapher ist die eines Informationsmarkts. Bedingt durch die technischen Rahmenbedingungen, die für die bisher verfügbare Informations- und Kommunikations-Technologie (IuK-Technologie) galten,

wurde das Informationsmanagement im UIS typischerweise an zentraler Stelle geplant und organisiert. Für solche *Informationstechnische Zentren* entsteht ein doppeltes Problem:

- Erstens müssen die Informationen inkl. ihrer Meta-Informationen bei den datenhaltenden Stellen „eingetrieben“ werden, was ein großes Maß von Kooperationsbereitschaft bei der datenhaltenden Stelle voraussetzt.
- Zweitens ist das Problem zu lösen, wie man mit diesen Informationen die potentiellen Nutzer erreicht.

Diese Herangehensweise, die, wie bereits angeführt, durch die technischen Rahmenbedingungen der Vergangenheit bedingt war, besitzt den Charakter einer „Planwirtschaft“: der Informationsbedarf wird an zentraler Stelle geplant und organisiert. Wünschenswert ist demgegenüber eine eher „marktwirtschaftliche“ Lösung, die durch die Bedürfnisse von Informationsanbietern und Informationsnutzern geregelt ist und sich durch ein Wechselspiel von Angebot und Nachfrage weitestgehend selbst organisiert. Das informationstechnische Zentrum stellt dann nur den technisch organisatorischen Rahmen für eine derartige „marktwirtschaftliche“ Lösung bereit. Ziel ist ein elektronischer Markt für Umweltinformationen, der die Beziehungen zwischen Informationsanbietern und Informationsnutzern unterstützt, indem auf eine Transparenz der Angebote und eine Eigenverantwortlichkeit der Akteure Wert gelegt wird. Damit werden im wesentlichen zwei Ziele verfolgt:

- Jeder Anbieter kann eigenverantwortlich seine Informationsangebote auf diesen Markt bringen.
- Jeder Nutzer kann sich über die aktuellen Informationsangebote informieren und bei Interesse und Bedarf von ihnen Gebrauch machen.

Dieser Markt ist nicht auf die Umweltverwaltung des Landes beschränkt, es können auch externe Anbieter wie auch externe Nutzer einbezogen werden: aus anderen Ländern, im Umweltbundesamt, in der europäischen Umweltagentur, aus der Wissenschaft, aus der Wirtschaft und nicht zuletzt aus der Öffentlichkeit. Dieser Markt ist ein elektronischer Markt; d. h. er ist online erreichbar für Informationssuchende und Informationsanbieter, und er erlaubt den Online-Durchgriff auf Originalinformationen und Originaldienste, soweit das immer mit vertretbarem technischen Aufwand möglich ist.

Zum Funktionieren einer marktwirtschaftlichen Lösung sind selbstverständlich Anreize erforderlich. Der naheliegendste Anreiz ist selbstverständlich Geld, z.B. durch die Verrechnung über Kostenstellen, aber auch immaterielle Anreize sind denkbar. Für die Nutzer auf der einen Seite ist ein reichhaltiges, qualitativ hochwertiges Informationsangebot ein wesentlicher Anreiz, den elektronischen Markt zu nutzen. Auf der anderen Seite kann der Umfang der Inanspruchnahme von Informationsangeboten als Anerkennung für die Leistung der Informationsanbieter gewertet werden. Durch automatisch erstellte Benutzungsstatistiken kann dieser Effekt noch verstärkt werden, insbesondere wenn dies von vorgesetzter Stelle als Leistungskriterium für die Arbeit der informations anbietenden Stelle herangezogen wird.

Der Übergang von der „planwirtschaftlichen“ zur „marktwirtschaftlichen“ Lösung bedeutet auch, daß das informationstechnische Zentrum im wesentlichen die Rolle eines Vermittlers übernimmt, der den Katalog der Informationsangebote führt. Dieser Katalog enthält lediglich die Meta-Information zu den Informationsangeboten; denn nur diese wird benötigt, um sich

über ein Informationsangebot zu informieren. Erst wenn das Angebot in Anspruch genommen wird, fließen die Informationen direkt vom Anbieter zum Nutzer; eine Inanspruchnahme der Dienste des informationstechnischen Zentrums ist hierbei nicht mehr erforderlich.

1.3 Informationsquellen im UIS

Wir unterscheiden im wesentlichen drei Arten von Informationsquellen im UIS, nämlich Daten, Methoden und multimediale Dokumente.

1.3.1 Daten als Informationsquellen

Die primären Informationsquellen im UIS sind Daten über die Umwelt, z.B. aus Meßnetzen, statistischen Erhebungen oder Satellitensensoren, aber auch aufbereitete Daten wie korrigierte Daten, aggregierte Daten oder komplexe Daten (z.B. Geodaten). Daten sind aber nur mittelbare Informationsquellen. Um Informationen zu erhalten, muß eine Interpretation noch erfolgen. Dies ist zugleich ein Nachteil und ein Vorteil: Zum einen erfordert die Interpretation von Daten Wissen, über das nicht jeder Informationssuchende verfügt. Andererseits sind Daten vielseitig interpretierbar und nicht auf eine einseitige Art der Nutzung festgelegt. Nutzer von Daten besitzen typischerweise geeignete Software-Tools und die zugehörige Erfahrung, um die Daten zu interpretieren. Dies können im einfachsten Fall Office-Softwareprodukte sein, wie z. B. eine Textverarbeitung oder eine Tabellenkalkulation, in komplizierteren Fällen sind komplexe Informationssysteme zur Nutzung der Daten erforderlich, wie bspw. ein Geoinformationssystem oder ein Statistikpaket.

1.3.2 Methoden als Informationsquellen

Viele Dienststellen im UIS verfügen über geeignete Verfahren, um aus Daten Informationen abzuleiten. Häufig sind diese Verfahren in Form von Methoden oder Dienstprogrammen auf einem Computer automatisch ablauffähig. Diese Methoden dienen als Informationsquellen, indem sie Informationen aus vorhandenen Daten generieren (virtuelle Informationen). Beispiele für Methoden sind Simulationen (z.B. die Schadstofftransportsimulation CRAYSIM), Datenbankzugriffe (z.B. die Selektoren des Umweltführungsinformationssystems UFIS), statistische Analysefunktionen (z.B. Methoden aus der Methodenbank ZEUS zum Betrieb eines Grundwassergütemeßnetzes) oder Präsentationsfunktionen (z.B. in Art eines Reportgenerators).

1.3.3 Multimediale Dokumente als Informationsquellen

In vielen Fällen ist es sinnvoll, nicht nur den Zugang zu unverarbeiteten Umweltdaten zu unterstützen, sondern darüber hinaus auch die Bereitstellung von Arbeitsergebnissen aus der Interpretation solcher Daten. Diese Arbeitsergebnisse liegen in der Regel in Form von Berichten und Karten, kurz, in Form multimedialer Dokumente vor. Dokumente wie Berichte und Karten sind sehr wichtige Informationsquellen, da sie – in impliziter Form – auch die Meta-Daten enthalten, die zur Interpretation der in das Dokument eingegangenen Basisdaten erforderlich sind. Insbesondere für die Öffentlichkeit und für Entscheidungsträger ist diese Art von Informationsquellen unverzichtbar.

1.3.4 Dienste zur Erschließung von Informationsquellen

Die oben genannten Informationsquellen sind infolge der zunehmenden Nutzung von Informationstechnologie bereits an vielen Stellen vorhanden. Um allerdings diese Informationsquellen nutzen zu können, ist es entweder erforderlich, sich physisch zu einer solchen Stelle zu bewegen, oder einen Mitarbeiter einer solchen Stelle zu beauftragen, die gewünschte Information der Informationsquelle zu „entlocken“.

Um nun auch diese Informationsquellen auf rationelle Weise von Ferne nutzen zu können, wird ein System von Netzwerkdiensten erforderlich. Diese Dienste ermöglichen den Zugriff auf die verschiedenen Arten von existierenden Informationsquellen. Hierfür sind prinzipiell ziemlich unterschiedliche Daten von Diensten denkbar. Telefon-, Fax- und Mailedienste ermöglichen die Kommunikation mit der datenhaltenden Stelle; allerdings ist in diesem Fall immer noch ein Mensch als Informationsvermittler erforderlich. Automatische Lösungen sind durch Netzwerkdienste wie Common Object Request Broker (CORBA), Remote Procedure Call (RPC), Remote Shell (RSH), Network File System (NFS) und World Wide Web (WWW) möglich, die allesamt aus der Welt der offenen Systeme und des Internet stammen. In den folgenden Abschnitten wird eine Dienstkonzeption präsentiert, die auf der Nutzung des WWW basiert. Diese Konzeption wurde vom FAW in den Projekten GLOBUS und INTEGRAL u.a. bei der Entwicklung eines generischen Datenbank-Gateways und eines Online-Informationsquellenkatalogs zugrundegelegt.

1.4 Management von Informationsquellen im WWW

Das netzübergreifende Hypermediasystem WWW stellt die heute wichtigste Benutzungsoberfläche für das weltweit größte Computernetz, das Internet, bereit. Die benutzerfreundliche Gestaltung des WWW-Systems hat dabei viel zur geradezu explosionsartigen Verbreitung des Internets beigetragen, ähnlich wie seinerzeit Windows den Markterfolg von DOS-basierten Personalcomputern beschleunigt hat.

Die Zugehörigkeit des WWW zur Internet-Welt bedeutet jedoch nicht, daß zur Nutzung des WWW ein Internet-Anschluß erforderlich ist. Die in den hier beschriebenen Projekten

verwendeten WWW-Protokolle und WWW-basierten Tools können durchaus auch auf privaten oder verwaltungseigenen Netzen betrieben werden; einzige Voraussetzung, die aber zunehmend von den verschiedensten Netzinstallationen erfüllt wird, so künftig auch vom Landesverwaltungsnetz Baden-Württemberg, ist die Unterstützung des TCP/IP-Protokolls. Mit Hilfe von Gateway-Rechnern und geschützt durch ein sogenanntes Firewall-Konzept kann jedoch auch, sofern gewünscht, aus einem derartig abgesicherten Netz heraus eine kontrollierte Anbindung an das „große“ Internet erfolgen.

Bei Verwendung eines interaktiven Browsersystems wie z.B. Mosaic oder Netscape besitzt WWW alle Eigenschaften eines modernen Hypermediasystems. Ein solcher Browser erlaubt zum einen die Präsentation von multimedialen Dokumenten, bestehend aus Text-, Graphik-, Audio- und Videoelementen. Zum anderen können die Hypermediadokumente über sogenannte Hyperlinks miteinander verknüpft werden; dies ermöglicht die Navigation zu (semantisch) benachbarten Dokumenten über sensitive Text- und Grafikelemente mit Hilfe von Mausklicks. Abgesehen von diesen Eigenschaften, die heute weitgehend Stand der Technik bei gängigen Hypertext- oder Hypermediasystemen sind, ist WWW als Client-/Server-Software für die Realisierung von Datenservern aus folgenden Gründen besonders geeignet:

- WWW ist eine offene, nicht-proprietäre Client-/Server-Lösung, die kostengünstig für die verschiedensten Plattformen zur Verfügung steht. Insbesondere sind auch PC-Systeme als Client- und zunehmend auch als Serversysteme nutzbar.
- WWW-Dokumente liegen verteilt auf unterschiedlichen Computerstationen im Rechnernetz. Jedes Hypermediadokument besitzt eine eindeutige Kennung, die sich (im wesentlichen) aus einem Dateinamen und dem Rechnernamen, auf dem sich die Datei befindet, zusammensetzt. Diese Kennung dient als netzweit eindeutige Referenz beim Aufbau von Hyperlinks zwischen Dokumenten. Dadurch lassen sich weltweite Informationsstrukturen aufbauen, die das gesamte Computernetz überspannen. Der Zugriff auf Informationen ist unabhängig davon, ob sich die Informationen auf dem lokalen oder einem fremden Rechner befinden, und ist auf diese Weise mittels einfacher Mausklicks möglich.
- WWW erlaubt auch den Einbezug von Datenfiles in das Netz der Hypertextdokumente. Wenn mit einem WWW-Browser eine Navigationsaktion auf ein solches Datenfile durchgeführt wird, so wird dieses automatisch zum Clientsystem übertragen und dort entweder im Filesystem abgelegt oder in ein Anwenderprogramm (einen sogenannten „Viewer“) geladen. Es können serverseitig verschiedenste Arten von Daten definiert werden und clientseitig beliebige Text-, Grafik- oder Geoinformationssysteme als Viewer bekanntgemacht werden, die der WWW-Browser automatisch aufruft, wenn das Serversystem einen Datenfile mit einer entsprechenden Kennung herausgibt. Für einige vordefinierte Datentypen wie z.B. Postscript-Dateien oder Rasterdaten im GIF-Format sind in den meisten Systemumgebungen bereits passende Viewer-Programme vorhanden.
- WWW-Dokumente können als Eingabemasken für Dienstprogramme dienen. Derartige Dokumente stellen sich dem Benutzer als graphisch interaktive Formulare dar. Über Eingabefelder, Menüs sowie durch maussensitive Selektions- und Optionsfelder können Dienste parametrisiert werden und durch Auslösen eines Datenfreigabefelds (OK-Button) kann das zugehörige Dienstprogramm gestartet werden. Als Ergebnis kann das Dienstprogramm dynamisch entweder ein neues

Hypermediadokument erzeugen, das die gesuchte Information enthält, oder auch Daten in anwendungsspezifischen Dateiformaten, die, wie oben beschrieben, im Clientsystems entweder auf dem Filesystem abgespeichert oder in ein Viewer-Programm geladen werden können.

Mit diesen Techniken ist es möglich, alle Arten von beschriebenen Informationsquellen, also Daten, Methoden und multimediale Dokumente, über das WWW in einem großen virtuellen Informationspool zu integrieren. Diese Informationsquellen sind dabei netzweit eindeutig durch eine sogenannte URL („Uniform Resource Locator“) identifiziert. Diese setzt sich zusammen aus einem Protokolltyp (z.B. http: Hypertext Transfer Protocol oder ftp: File Transfer Protocol), einer eindeutigen Identifikation des Serverrechners, auf dem sich die Informationsquelle befindet, einer Identifikation der Informationsquelle auf dem Serverrechner (im wesentlichen ein Dateiname, ggf. versehen mit Aktualparametern) sowie optional einem Port, über den die Kommunikation mit dem Serverrechner stattfindet. Durch textuelle Eingabe einer URL oder durch Anklicken eines Hyperlinks, der mit einer solchen URL hinterlegt ist, kann eine Aktivierung der betreffenden Informationsquelle erfolgen.

1.4.1 Daten im WWW

Einfache Daten im WWW sind repräsentiert durch Dateien, die unterschiedliche Dateitypen besitzen können. Die Aktivierung einer URL, die für eine solche Datei steht, bewirkt, daß die Datei, wie oben beschrieben, auf den Client-Rechner übertragen wird und dort entweder im Filesystem abgelegt oder in ein zugehöriges Viewer-Programm geladen werden kann. Ein solches Viewer-Programm kann den Nutzer dann bei der Interpretation der Daten unterstützen, d.h. bei der Extraktion von Information aus den Daten. Typische, im Rahmen des Projekts INTEGRAL genutzte Viewer sind insbesondere Tabellenkalkulationssysteme, Grafiksysteme, Textverarbeitungssysteme und Systeme für das Desktop Mapping.

Der Zugriff auf Daten in Datenbanken oder anderen datenhaltenden Systemen erfolgt über Zugriffsmethoden. Die Aktivierung von Methoden im WWW ist im nächsten Abschnitt beschrieben.

1.4.2 Methoden im WWW

Im WWW tritt eine Methode als ein parametrisierbares Dienstprogramm (ein sogenanntes CGI-Skript) in Erscheinung. Die Aktivierung einer Methode entspricht einem Aufruf eines solchen Dienstprogramms über das WWW und ist eindeutig identifiziert durch eine URL. Die URL eines Dienstaufrufs ist eine Zeichenkette, die sich dadurch auszeichnet, daß sie neben einer eindeutigen Bezeichnung des Dienstprogrammes eine Liste von Aktualparametern enthält. Der Aufruf einer Methode erfolgt im WWW durch eine „Navigationsaktion“ zu einer derartigen URL. Der Server veranlaßt dann den Aufruf des zugeordneten Dienstprogramms mit den jeweiligen Aktualparametern. Das Dienstprogramm erzeugt als sichtbares Ergebnis einen Datenstrom, der die Definition einer Hypertextseite enthält, die schließlich dem Dienstenutzer angezeigt wird.

Mit Hilfe von WWW-Formularen (das sind besondere Hypertextdokumente) können Aufrufe von Methoden auf bequeme Weise über einen WWW-Browser (z.B. Mosaic) veranlaßt werden. Dies geschieht dadurch, daß die in dem Formular spezifizierten Angaben zu einer einzigen URL zusammengefügt werden. Durch Drücken einer Schaltfläche (Submit-Button) erfolgt die „Navigation“ zur spezifizierten URL, d.h. der Aufruf des betreffenden Dienstprogramms.

1.4.3 Multimediale Dokumente im WWW

Die Repräsentation multimedialer Dokumente im WWW bereitet die geringsten Probleme, da WWW speziell hierfür konzipiert wurde. Mit Hilfe der im WWW verwendeten Dokumentendefinitionssprache HTML (Hypertext Markup Language) können multimediale Dokumente inklusive der enthaltenen Hyperlinks definiert werden.

Falls die Dokumente ursprünglich in anderen Systemen (z.B. Textverarbeitung, Desktop Publishing) erstellt wurden, ist es erforderlich, eine Formatwandlung nach HTML vorzunehmen. Inzwischen gibt es eine große Anzahl von Werkzeugen, die eine solche Konvertierung unterstützen; die neueren Systeme zur Textverarbeitung und zum Desktop Publishing stellen bereits Exportfilter für HTML bereit.

1.5 Prototypische Realisierung des Informationsquellenmanagements

Die Arbeiten des FAW zur prototypischen Realisierung des Informationsquellenmanagements zielten von Anfang an auf die Entwicklung offener, verallgemeinerungsfähiger Methoden und Tools zur Integration heterogener Komponenten ab und beschränken sich nicht auf die Optimierung eines speziellen WWW-Servers einer einzelnen Behörde. Für den ressort- und künftig auch für den länderübergreifenden Einsatz ist das von besonderer Bedeutung. So wurde im Projekt INTEGRAL ein generisches Umweltdatenbank-Gateway entwickelt und am Beispiel der UFIS-Datenbank demonstriert, das den Anschluß beliebiger SQL-fähiger Umweltdatenbanken erlaubt, ohne daß dafür eine Programmierung in 'C' o.ä. erforderlich ist. Diese Entwicklung stellt ebenso wie der vom FAW im Rahmen von GLOBUS entwickelte Informationsquellenkatalog ein Alleinstellungsmerkmal der FAW-Lösung dar: Mit dem Informationsquellenkatalog wurde erstmals eine Filter- und Broker-Software entwickelt, die das Retrieval von Umweltinformationen aus verteilten, heterogenen Informationsquellen auf dem WWW unterstützt, indem Metainformationen aus dem Umweltdatenkatalog (erweitert um ein Klassenkonzept), aus einem polyhierarchischen Thesaurus, einem Ortsverzeichnis, einem Netzwerkdienstverzeichnis und einem Benutzerverzeichnis als Navigationsunterstützung genutzt werden.

1.5.1 Unterstützung der Verbreitung und der Nutzung von Informationen aus Umweltdatenbanken über WWW

Eine wichtige Zielsetzung der hier beschriebenen Arbeiten ist es, die verwaltungsinterne und öffentliche Verbreitung von Informationen aus Umweltdatenbanken der öffentlichen Hand zu unterstützen. Der gewählte Ansatz besteht dabei darin, den Online-Zugriff auf Informationen in solchen Datenbanken über das WWW zu ermöglichen. Diese Zugriffsmöglichkeit wird primär für Nutzer aus der Umweltverwaltung in Bund und Land eingerichtet; darüber hinaus sollte sie, ggf. in kontrollierter Form, zunehmend auch der Öffentlichkeit ermöglicht werden.

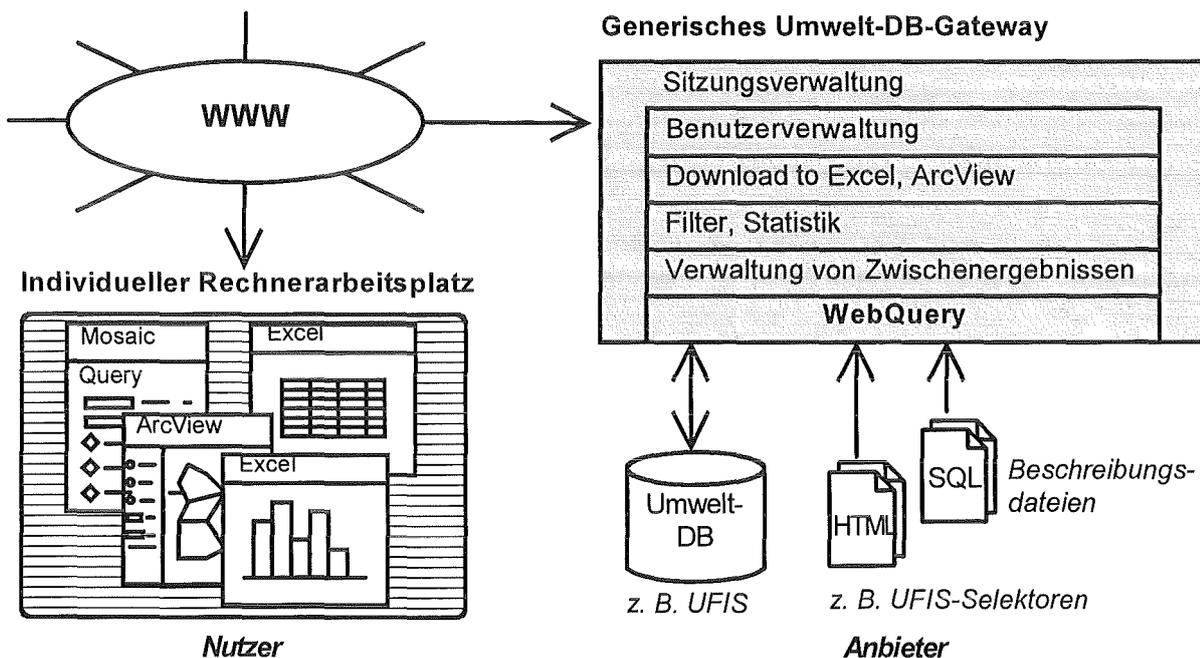


Abbildung 1-1: Anbindung von Umweltdatenbanken an das WWW

Das FAW hat hierfür eine Softwarelösung, ein *generisches Umweltdatenbank-Gateway*, in Form eines Prototyps entwickelt, der den Anschluß von Umweltdatenbanken an das WWW unterstützt (siehe Abbildung 1-1). Diese Lösung zeichnet sich durch folgende Leistungsmerkmale aus:

- Es werden Netzwerkdienste, sogenannte *Selektoren*, eingerichtet, die den Zugriff auf ausgewählte Informationsbestände aus der Datenbank erlauben. Diese Selektoren treten dem Benutzer gegenüber in Form von Bildschirmformularen (Masken) in Erscheinung. Die vom Benutzer eingetragenen Formularinhalte dienen zur Parametrisierung von Datenbankanfragen. Das Ergebnis einer Anfrage wird auf einem dynamisch generierten Hypertextdokument angezeigt und kann auf Wunsch auch in Datenform auf das Clientsystem („Individueller Rechnerarbeitsplatz“) des Benutzers übertragen werden.
- Für den Standardfall, d.h. für eine SQL-Datenbank, benötigt die datenanbietende Stelle *keine besonderen Programmierkenntnisse*, um Selektoren definieren zu können. Zum Anschluß einer Umweltdatenbank an das WWW genügt die Erstellung eines Satzes von Beschreibungsdateien, wofür lediglich Kenntnisse in der Erstellung von WWW-Seiten und der Datenbanksprache SQL erforderlich sind. Das generische

Umweltdatenbank-Gateway stellt unabhängig von der angeschlossenen Datenbank allgemeine Funktionen hinsichtlich Sitzungsverwaltung, Mehrbenutzerbetrieb, Benutzerverwaltung, Statistik, Verwaltung von Zwischenergebnissen, Download usw. bereit, ohne daß hierfür eine Programmierung erforderlich wäre.

- Für Spezialfälle, wie den Anschluß von Geodatenbanken oder nicht SQL-fähigen Datenbanken sowie für die Anbindung gänzlich anderer Anwendungen (z.B. Umweltsimulationen) kann ebenfalls auf Module des generischen Datenbank-Gateways zurückgegriffen werden, jedoch werden darüber hinaus im Einzelfall noch spezielle Software-Entwicklungen erforderlich.
- Die Selektoren der Umweltdatenbank sind einzelne, unabhängig voneinander ansprechbare und nutzbare Dienste. Für diese Dienste können an unterschiedliche Benutzergruppen unterschiedliche Berechtigungen vergeben werden, so daß sich die Datenbank speziellen Nutzergruppen (Einzelpersonen, Dienststellen, Verwaltung allgemein, Öffentlichkeit usw.) gegenüber mit unterschiedlichen Inhalten und Abfrageoptionen darstellen kann.

1.5.2 Bereitstellung von Berichten und anderen multimedialen Dokumenten über die Umwelt mittels WWW

In vielen Fällen ist es sinnvoll, nicht nur den Zugang zu unverarbeiteten Umweltdaten zu unterstützen, sondern darüber hinaus auch die Bereitstellung von Arbeitsergebnissen aus der Interpretation solcher Daten. Diese Arbeitsergebnisse liegen in der Regel in Form von Berichten und Karten, kurz, in Form multimedialer Dokumente vor. Solche multimedialen Dokumente können mit Hilfe geeigneter Tools im WWW als Hypertext zur Verfügung gestellt werden. Es ist dann möglich, im WWW durch ein solches Dokument entsprechend seiner Struktur hindurchzunavigieren.

Es hat sich jedoch gezeigt, daß die Navigation über Hyperlinks nicht ausreicht, um einen komfortablen Zugang zu den Seiten eines derartigen Dokuments zu ermöglichen. Gewünscht sind Möglichkeiten, um durch Angabe einiger Deskriptoren direkt an die gesuchten Informationen zu gelangen. Solch eine Funktionalität wurde am FAW im Rahmen des Projekts GLOBUS am Beispiel von Umweltberichten, konkret am Beispiel des Berichts der Baden-Württembergischen Landesanstalt für Umweltschutz prototypisch realisiert (Abbildung 1-2). In der Projektphase II des GLOBUS-Projekts wurde diese Lösung vom Forschungszentrum Karlsruhe für die Bereitstellung und Erschließung von Berichten zur Altlastenthematik in großem Stil angewandt und dabei in Teilen noch verfeinert.

Für den Informationssuchenden bestehen insgesamt drei Möglichkeiten des Zugangs zu den Dokumentseiten eines Berichts. Die erste Möglichkeit besteht wie bereits oben angeführt im hierarchischen Zugang zu den Informationen über ein systematisches Kapitelverzeichnis und zugehörige Unterverzeichnisse. Hierdurch ist der Benutzer allerdings auf die Systematik des Berichterstellers eingeschränkt. Die zweite Möglichkeit besteht über ein Volltextsuchprogramm, das über eine WWW-Formularseite gestartet werden kann und die gesuchten Berichtsseiten mit Hilfe eines dafür eingerichteten Volltext-Index lokalisiert. Vorteil dieses Verfahrens ist die Möglichkeit, beliebige Suchmuster verwenden zu können, Nachteil ist die u.U. längere Dauer der Suche und die fehlende Berücksichtigung von Synonymbegriffen.

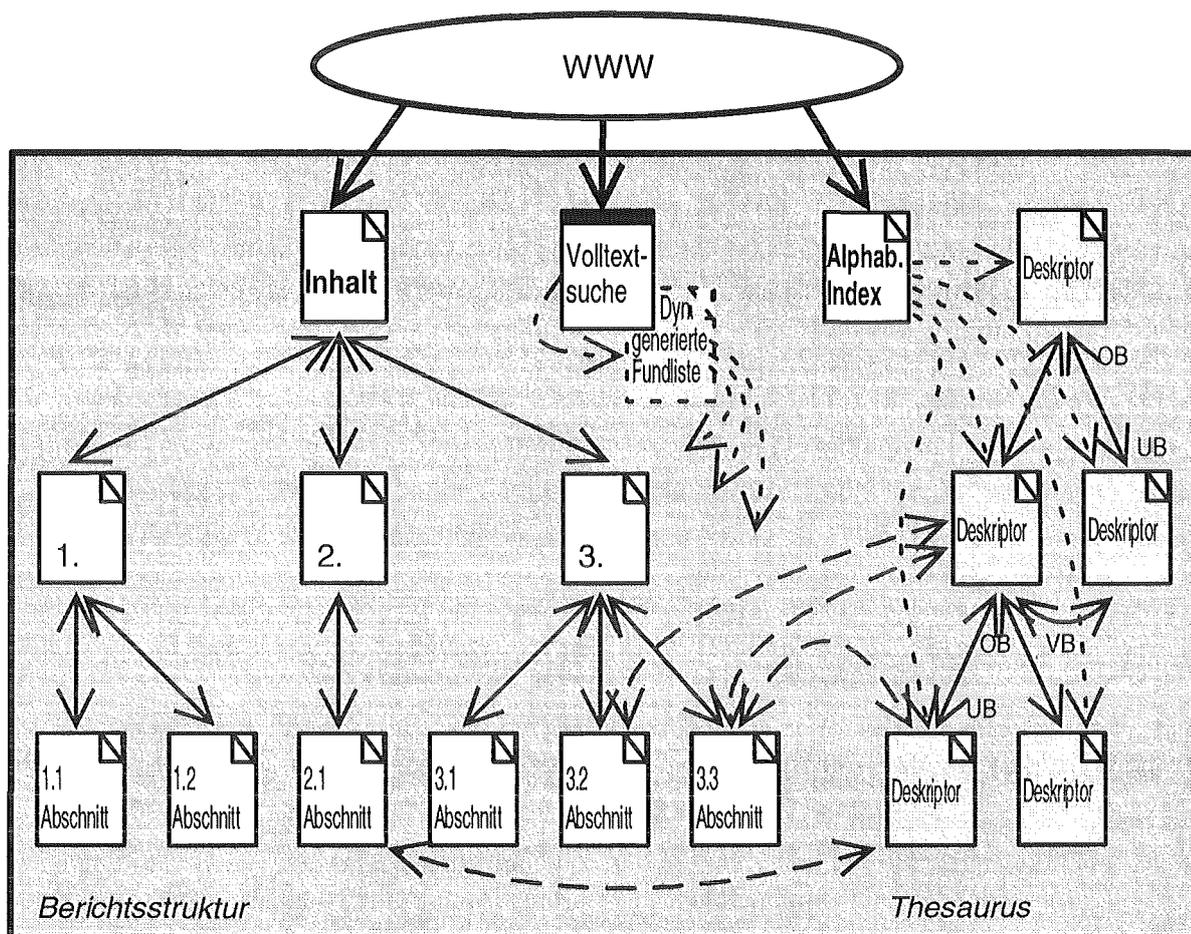


Abbildung 1-2: Bereitstellung von Berichten

Die dritte Möglichkeit besteht in der Suche über einen polyhierarchischen Thesaurus. Hierfür wurde der Thesaurus des Umweltbundesamts (UBA-Thesaurus) verwendet, der als Hypertext aufbereitet wurde, wobei für jeden Deskriptor ein Hypertext-Dokument angelegt wurde. Über einen alphabetischen Index ermittelt der Benutzer einen möglichen Deskriptorbegriff und läßt sich diesen als Hypertextdokument zeigen. In diesem Hypertextdokument sind die Titel aller mit dem betreffenden Deskriptor oder mit seinen Synonymen verschlagworteten Berichtsseiten aufgeführt. Durch Anklicken der Titel können die Berichtsseiten unmittelbar erreicht werden. Sollten die mit dem Deskriptor assoziierten Berichtsseiten nicht interessieren, so besteht die Möglichkeit, zu anderen Deskriptordokumenten (Oberbegriffe, Unterbegriffe, verwandte Begriffe) zu navigieren und die dort vermerkten Berichtsseiten zu inspizieren. Darüber hinaus ist es auch möglich, von einem gefundenen Berichtsdocument über Hyperlinks zu den Deskriptoren zu gelangen, mit denen sie verschlagwortet sind, so daß eine freie Navigation zwischen Deskriptoren und Berichtsseiten unterstützt wird.

Die Verschlagwortung der Berichtsseiten nach dem UBA-Thesaurus erfolgt vollautomatisch mit Hilfe eines effizienten Textvergleichsprogrammes. Nach Wunsch können die automatisch ermittelten Schlagwörter vom Berichtsanbieter noch modifiziert werden.

Insgesamt liegt mit dieser, weitgehend noch im Rahmen der Projektphase I des GLOBUS-Projekts erstellten Software bereits eine brauchbare Lösung für die Gestaltung eines Berichteservers auf dem WWW vor. Für die Archivierung von Berichten zur Altlastenthematik wird diese Software von Forschungszentrum Karlsruhe bereits prototypisch eingesetzt.

Weitergehende Anforderungen wurden in der Projektphase II des GLOBUS-Projekts gestellt. So wurde im Projektauftrag des Umweltministeriums an das FAW insbesondere gefordert:

Die nun vorliegende Version 2.0 des Umweltdatenkatalogs (UDK) umfaßt ein Thesaurus-Modul, in dem aktuell der UBA-Thesaurus bereits in relationalen Datenbanken zur Verfügung steht. Unter Berücksichtigung der weiteren Datenmodellierungen des UDK (insbesondere der UDK-Objekte) ist daher ein Konzept zur Nutzung dieses Thesaurus zum Zugriff auf entsprechend verschlagwortete Umweltberichte zu entwickeln und in Teilen prototypisch zu realisieren.

Im Rahmen der Arbeiten an GLOBUS II sollen die folgenden Anforderungen berücksichtigt werden:

- *thesaurus-unterstützter Zugriff auf mehrere Umweltdatenbestände auf unterschiedlichen Servern,*
- *Ausweitung des thesaurus-unterstützten Zugriffs auf andere allgemeine Daten und Dienste (nicht nur Umweltberichte),*
- *Integration weiterer Indizierungstechniken (Raumbezug, Zeitbezug, Datenherr).*

Die Weiterentwicklung soll insbesondere einen Zugriff über die logische Verknüpfung von Deskriptoren erlauben.

Diese Anforderungen wurden im Rahmen der Arbeiten am Informationsquellenkatalog realisiert, der im folgenden Abschnitt übersichtsartig dargestellt ist. Eine detaillierte Darstellung des Informationsquellenkatalogs findet sich im Kapitel 2 dieses Berichts.

1.5.3 Unterstützung des Retrievals von Umweltinformationen

Es ist zu erwarten, daß das Angebot von Umweltinformationen, die über WWW erreichbar sind, in beschleunigtem Maß zunimmt. Dazu zählen die in den vorigen beiden Abschnitten genannten Informationen aus Umweltdatenbanken und multimedialen Dokumenten ebenso wie allgemeine Informationsdienste (z.B. über Ozonmeßwerte) oder Berechnungsdienste (z.B. Schadstofftransportsimulationen). Insgesamt wird sich so ein komplexes Angebot von *Umweltinformationsquellen* auf dem Internet ergeben, die teils durch Dienststellen der Umweltverwaltung, teils durch externe Einrichtungen bereitgestellt werden.

Dieses wachsende Angebot von Informationsquellen bedeutet, daß einzelne Informationsangebote oft nur sehr schwer auffindbar sind. Erforderlich ist deshalb eine *Filter- und Brokerlösung* auf der Basis von Metainformation, mit welcher es Nutzern aus Verwaltung und Öffentlichkeit möglich ist, gesuchte Informationen zu finden (Brokerfunktionalität), ohne durch ungewünschte Informationen gestört zu werden (Filterfunktionalität). An der Filter- und Brokerthematik arbeitet das FAW schon lange mit vielen Partnern, wobei auch Entwicklungen wie das am MCC in Texas entwickelte KI-System CYC zur Repräsentation von Alltagswissen einbezogen sind. Die Enquete-Kommission Baden-Württemberg hat das Thema Filter und Broker zum Zukunftsthema erklärt.

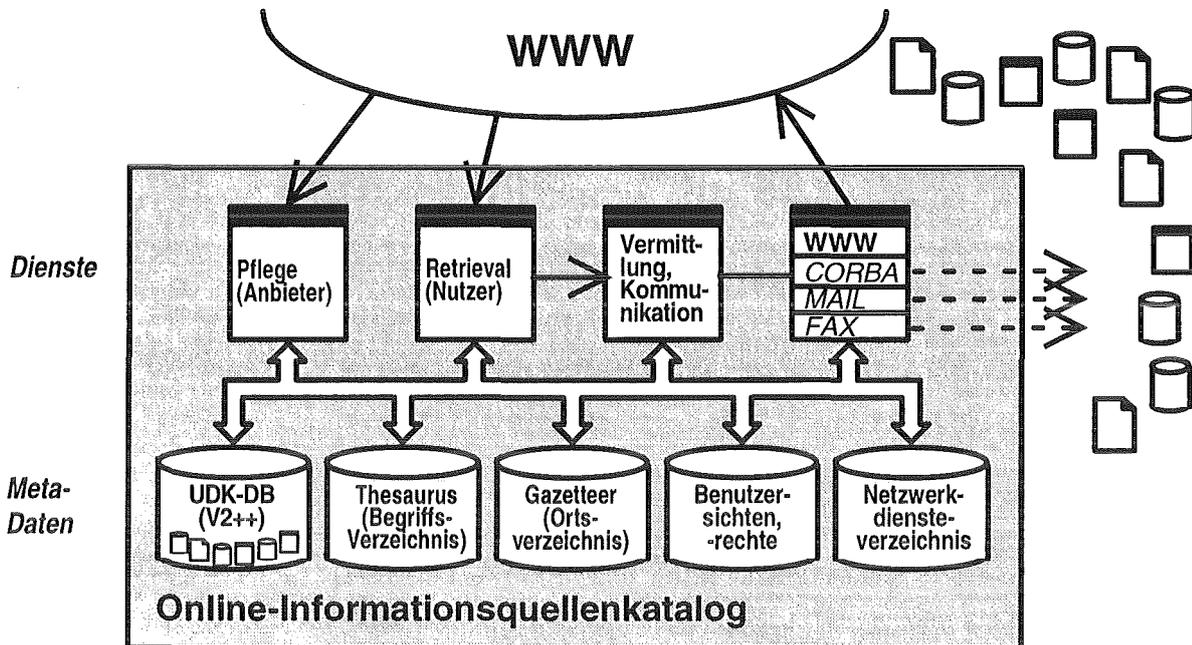


Abbildung 1-3: Der Umwelt-Informationsquellenkatalog

Das FAW hat für das Retrieval von Umweltinformationen im WWW einen Prototyp eines Filters und Brokers entwickelt, der den Namen *Online-Informationsquellenkatalog* trägt. Dieser Katalog zeichnet sich durch folgende Merkmale aus:

- Der Katalog enthält ein durchsuchbares Verzeichnis aller verfügbaren Arten von Informationsquellen (Daten, Dienstprogramme, Dokumente etc.). Diese Informationsquellen können hinsichtlich Ortsbezug, Raumbezug, Fachbezug und datenhaltender Stelle recherchiert werden. Das Verzeichnis der Informationsquellen wurde auf der Basis des Umweltdatenkatalogs (UDK) mit objektorientierter Erweiterung, wie für Version 3 geplant, realisiert. Für die Weiterentwicklung des Informationsquellenkatalogs sollen weiterführende Ansätze, insbesondere das geplante Verweis- und Kommunikationssystem (VKS) des UBA und der Catalogue of Data Sources (CDS) der Europäischen Umweltagentur einbezogen werden.
- Für die Eingabe von Fachbezügen wird auf einen polyhierarchischen Thesaurus, für die Eingabe von Raumbezügen auf ein strukturiertes Ortsverzeichnis („Gazetteer“) zurückgegriffen. Semantische Beziehungen zwischen Begriffen sowie geometrisch-topologische Beziehungen zwischen Raumbezügen werden vom System bei der Suche nach Informationsquellen ausgewertet und genutzt. Für einen künftigen internationalen Einsatz ist die Nutzung eines multilingualen Thesaurus geplant.
- Für die gefundenen Informationsquellen können Netzwerkdienste aktiviert werden, die den Zugang zu den gewünschten Informationsquellen ermöglichen. Damit sind in erster Linie die über das HTTP-Protokoll erreichbaren Dienste im WWW gedacht, die einen direkten Zugang zu den Informationsquellen ermöglichen. Alternativ sollten künftig aber auch andere Netzwerkprotokolle wie z.B. CORBA sowie eine Kommunikation mit der datenhaltenden Stelle über Electronic Mail oder Fax unterstützt werden.

- Es werden unterschiedliche Sichten auf den Informationsbestand unterstützt (Filterfunktion). Dadurch sind für die jeweilige Benutzergruppe nur diejenigen Informationsangebote sichtbar, die für diese von Interesse sind.
- Die Eintragung von Informationsquellen erfolgt vorzugsweise durch die datenhaltende Stelle. Der Informationsquellenkatalog hält für den Eintrag und die Pflege Dienste bereit, die von registrierten Angehörigen von UDK-Instanzen über WWW in Anspruch genommen werden können. Die Extraktion der Metainformation aus den Informationsquellen soll zunehmend durch automatische Verfahren unterstützt werden. Bereits in der jetzigen Lösung erfolgt die Verschlagwortung von Textdokumenten automatisiert durch Textvergleiche mit dem Thesaurus. Analoges ist für die Ermittlung des Raumbezugs mit Hilfe des Ortsverzeichnisses (Gazetteers) geplant.

Insgesamt stellt der Informationsquellenkatalog eine Art „elektronischen Marktplatz“ für Umweltinformationsangebote dar, der für Datenanbieter und Datennutzer gleichermaßen von Nutzen ist.

1.5.4 Unterstützung der Erstellung von Berichten

Eine Schlüsselaufgabe bei der Bereitstellung von Umweltinformation ist die Erstellung von Berichten; diese Leistung wird in der Umweltverwaltung an vielen Stellen erbracht. Zu diesem Zweck wird ein individueller Rechnerarbeitsplatz (i.d.R. auf PC-Basis) zur Erstellung multimedialer Berichte benötigt. Das FAW hat hierfür auf der Basis von Standardwerkzeugen aus der PC-Welt einen Prototyp entwickelt, der folgende Funktionen unterstützt:

- Abruf von Daten und Informationen aus den Serversystemen über einen WWW-Browser (Mosaic, Netscape etc.)
- Laden der Daten in ein Tabellenkalkulationssystem (Excel) oder Desktop-Mapping-System (ArcView). Laden von Berichten in Form von Hypertext in ein Textverarbeitungssystem (MS Word for Windows / Internet Assistant).

Diese Lösung ist noch um folgende Funktionalitäten erweiterbar:

- Analyse und Aufbereitung der Daten durch Statistikpakete, Tabellenkalkulation, Businessgraphik und Desktop Mapping.
- Integration der aufbereiteten Information in Berichtsform mit Hilfe eines Textverarbeitungssystems oder Desktop-Publishing-Systems.
- Bereitstellung des Berichts in Form von Hypertext mit Hilfe eines Network-Publishing-Tools. Dies umfaßt auch das Eintragen des Berichts in den Informationsquellenkatalog (siehe vorigen Abschnitt), womit er für den vorgesehenen Nutzerkreis allgemein nutzbar ist.

1.6 Ausblick

Aus der dienstorientierten Betrachtungsweise, die den beschriebenen Arbeiten zugrundeliegt, ergeben sich wesentliche Ansätze für ein Management heterogener Informationsquellen im UIS Baden-Württemberg. Die Leitidee dabei ist es, in UIS-Teilkomponenten mehrfach verwendbare Daten, Funktionalitäten und multimediale Informationen zu identifizieren und mit Hilfe von Netzwerkdiensten zur Verfügung zu stellen. Diese Dienste können im Rahmen einer Client-Server-Konzeption anderen Anwendungen und interaktiven Benutzern netzweit zur Verfügung gestellt werden. Einerseits läßt sich so eine möglichst wirtschaftliche Nutzung der vorhandenen Hard- und Software-Ressourcen erzielen. Andererseits wird so ein möglichst ungehinderter und bequemer Zugang zu den im Landesverwaltungsnetz angebotenen Daten, Funktionalitäten und Informationen verwirklicht; auf lange Sicht eröffnet dies den Weg hin zu einer computerunterstützten kooperativen Arbeitsweise in der Umweltverwaltung.

Eine entscheidende Rolle auf dem Weg zu diesem Ziel spielen fortschrittliche Techniken der Informationsverarbeitung wie die Bereithaltung von Metainformation in einem Online-informationsquellenkatalog oder der Einsatz einer netzübergreifenden Hypermedia-Software. Dabei können Schnittstellenstandards, die zumeist aus der Welt der offenen Systeme stammen, vorteilhaft eingesetzt werden.

Die beschriebene, als Prototyp vorliegende Entwicklung ist darüber hinaus ein geeigneter Ausgangspunkt für eine länderübergreifende Lösung, die sowohl in Deutschland, als auch in Europa oder weltweit im Rahmen der Aktivitäten der G7-Länder für eine *Global Information Infrastructure* Verwendung finden kann. Mit einer solchen Lösung wird eine komfortable, schnelle und skalierbare Möglichkeit zur Bereitstellung, Verbreitung und Nutzung von Umweltinformationen geschaffen, um damit Exekutive, Legislative, Verwaltung, Wirtschaft, Wissenschaft und nicht zuletzt die Öffentlichkeit zu erreichen. Es läßt sich ein virtueller Informationspool aufbauen, der verteilte Informationsquellen aus unterschiedlichen Organisationen und unterschiedlichen Ländern in einem Informationsnetz zusammenfaßt. Bereits vorliegende, oft kostenlos nutzbare Entwicklungen aus der Internet-Welt sowie marktgängige Werkzeuge zur Informationspräsentation, zur Berichtserstellung und zum Network Publishing wie z.B. Office-Softwareprodukte und Desktop-Mapping-Systeme sind in die Konzeption einbezogen, so daß sich der Umfang der Eigenentwicklungen auf ein minimales erforderliches Maß beschränkt.

2. Der Umweltinformationsquellenkatalog

2.1 Allgemeine Beschreibung

Ein Umweltinformationssystem hat in erster Linie die Funktion, Daten, Dienste und Berichte allgemein, flächendeckend und technisch uneingeschränkt zugänglich zu machen. Das UIS in der jetzigen Form beinhaltet bereits Applikationen, die den Zugang zu Daten, beispielsweise zu Meßdaten verschiedener Meßnetze, auf der Basis moderner Informations- und Kommunikationstechnologien ermöglichen und eine einheitliche Auswertung und

Interpretation der Daten zur Verfügung stellen. Dieser Zugang zu verschiedenen Meßdaten unterschiedlicher Meßnetze war bisher nur aus Monolithen wie z.B. TULIS und UFIS möglich, die als gesamte Applikation zugänglich waren und intern aus einzelnen Diensten bestanden.

Diese einzelnen Dienste aus den Monolithen herauszukristallisieren und einzeln von außen zugänglich zu machen, bedeutet einen weiteren Schritt in Richtung Effizienzsteigerung der Verwaltung von Umweltinformationen, da hiermit der direkte themenspezifische Zugriff auf Meßdaten ermöglicht werden kann. Den Funktions- und Datenumfang des gesamten UIS direkt themenspezifisch zugreifbar zu machen, führt aber sehr schnell zu einer unüberschaubaren Menge von Informationen, die schnell den Bedarf nach Hilfestellungen bei der zielgerechten Suche nach Informationen aufkommen läßt.

Neben Daten und Diensten gehören die Berichte zu der dritten Hauptgruppe von Umweltinformationen, die mittels eines Retrieval-Systems auffindbar und zugreifbar gemacht werden sollten. Hierbei zeichnet sich die Grundsituation problematischer als auf der Datenseite ab, da die Informationsquellen – sprich die Berichte – nicht selbstverständlich in digitalisierter Form vorliegen. Um eine Auskunftskomponente mit Filter- und Broker-Mechanismen für sämtliche Umweltinformationen zu realisieren, müssen diese in einem ersten Schritt auf ein gemeinsames digitales Format abhängig von der verwendeten Systemarchitektur gebracht werden.

Für die Systemarchitektur eines reinen Berichtsretrievalsystems eignen sich Konzepte wie Hypertext ausgezeichnet, weil sie alle Anforderungen für einen benutzerfreundlichen Zugang zu Informationen für die breite Öffentlichkeit erfüllen. Das auf dem Internet basierende System WorldWideWeb (WWW) ist ein solches Hypertextsystem, das vorhandene Informationen als Hypertext über eine einfach zu handhabende Benutzeroberfläche weltweit verfügbar macht. Über Hyperlinks in diesen Hypertexten können jedoch nicht nur weitere Hypertexte aktiviert, sondern auch Programme angestoßen werden, die ihre Ergebnisse wieder in Hypertextform präsentieren können. WWW als Hypertextsystem in einer heterogenen Client-/Server-Architektur eignet sich daher als Systembasis für ein effizientes, aktives Auskunftssystem für das UIS Baden-Württemberg. Mit der Einführung des WWW als Systemarchitektur werden folgende Grundvoraussetzungen für ein aktives Auskunftssystem ohne zusätzlichen Eigenaufwand geschaffen :

- heterogene Client-Server-Technik mit intelligenten Clients
- einheitliche Oberfläche bei heterogenen Rechnerplattformen
- plattformübergreifender Datenaustausch
- Integration in lokale Applikationen.

Die Arbeiten des FAW sowohl im Projekt INTEGRAL als auch im Verbundprojekt GLOBUS befassen sich mit der Problematik Retrieval bzw. Auskunftssysteme mit Filter- und Broker-Mechanismen im Rahmen des UIS Baden-Württemberg, wobei der typische Nutzerkreis Referenten des UM und der LfU umfaßt. Schwerpunkte liegen hierbei im Projekt INTEGRAL auf der Daten und Diensteseite, in GLOBUS hingegen in erster Linie auf der Seite Berichtsretrieval. Somit ergänzen sich die Aufgaben der beiden Projekte, und am FAW wurde in Synergie mit INTEGRAL und GLOBUS eine Auskunftskomponente, der *Umweltinformationsquellenkatalog* (UIK) entwickelt, der eine zielgerichtete, attributierte Suche nach Umweltinformationen mit anschließendem direktem Zugriff ermöglicht.

Bei der Entwicklung des Umweltinformationsquellenkataloges wurde auf dem Umweltdatenkatalog (UDK) aufgesetzt und dieser in den erforderlichen Bereichen erweitert und ergänzt. Voraussetzung des Systems ist derzeit, daß die Umweltinformationen in einem Hypertextformat vorliegen, um in das System integriert zu werden, und zwar in der Hypertext Markup Language HTML, die im WWW Verwendung findet. Bereits in GLOBUS I wurden erste Erfahrungen mit der Digitalisierung und Konvertierung von Textdokumenten in HTML-Format am Beispiel des Berichts „Umweltdaten 91/92“ gesammelt, und das Forschungszentrum Karlsruhe hat in der Phase II des Projekts GLOBUS diese Erfahrungen weiter ausgebaut.

Im Projekt INTEGRAL III wurde die Integration von Applikationen des UIS in das WWW am Beispiel UFIS konzipiert und prototypisch realisiert, indem ca. 50 Einzeldienste aus UFIS herauskristallisiert und in WWW integriert wurden. Diese Projekterfahrungen bestätigten die Tragfähigkeit des Ansatzes, ein operationelles Zugriffs- und Bereitstellungssystem für Umweltinformationen auf der Basis von WWW zu realisieren. Es wurde gezeigt, daß eine Bereitstellung der Informationen und Konvertierung bzw. Anpassung bereits bestehender Informationen mit vertretbarem Aufwand möglich ist.

Das Umweltinformationsauskunftssystem fungiert prinzipiell als Metainformationsserver, der Metadaten von verteilt verwalteten Informationen enthält und den direkten Zugriff darauf ermöglicht. In den Kernbereichen des Systems wurde auf den UDK aufgesetzt, der bereits eine Auskunftskomponente darstellt, die Metadaten von Informationen archiviert und über eine graphische Benutzeroberfläche zugreifbar macht. Diese Metadaten werden zusammen mit einem Thesaurus in einer relationalen Datenbank verwaltet.

In Zusammenhang mit der Anbindung der UFIS-Dienste an das WWW im Rahmen von INTEGRAL III wurde ein am FAW entwickeltes Basistool genutzt und weiterentwickelt, das eine Schnittstelle zwischen WWW und ORACLE darstellt und das Erstellen neuer Datenbankzugriffsdienste via WWW mittels einfacher in SQL und HTML + Zusatztags formulierter Beschreibungsfiles ohne zusätzlichen Programmier- und Compilieraufwand ermöglicht (siehe Kapitel 3.1). Dieses Basistool fand auch bei der Realisierung des Umweltinformationsquellenkatalogs Einsatz und ermöglicht eine beliebige Erweiterung und den flexiblen Ausbau, ohne erneuten Entwicklungsaufwand leisten zu müssen. Mit diesen technischen Voraussetzungen und dem UDK inkl. Thesaurus als Basis wurde mit dem Umweltinformationsquellenkatalog eine Auskunftskomponente entwickelt, die ein autonomes System darstellt. In diesem System können die Informationsanbieter eigenverantwortlich ihre Metadaten einbringen, wobei die Originalinformationen jedoch nach wie vor auf dem eigenen Informationsserver verbleiben können. Somit verteilt sich der administrative Aufwand im Bereich Pflege und Wartung sowie Datenintegrität auf die einzelnen Anbieter des Systems.

Sowohl beim Einbringen als auch beim Abfragen von Informationen besitzen die Metadaten für den Anwender große Bedeutung.

- Raumbezüge,
- Zeitbezüge,
- Schlagworte zur Festlegung der thematischen Zuordnung basierend auf dem Thesaurus des UDK,
- die datenhaltende Stelle und

- die Typisierung der Information

werden als Attribute der Metadaten abgelegt, die für die Selektion der Information entscheidenden Charakter besitzen.

Die Lokalität der Informationsquellen (grob gesprochen ihre Netzwerkadresse), die für die Realisierung des Broker-Mechanismus benötigt wird, ist definiert durch eine Reihe von Metadaten in einem Netzwerkdienstverzeichnis. Diese Metadaten umfassen Pfad und Namen der Dateien auf dem Serverrechner, welche die Daten, Dienste oder Berichte enthalten, zusammen mit der als eigene hierarchisch strukturierte Entität abgelegten Serverinformation der datenhaltenden Stellen, die den Instanzen der Umweltverwaltung entsprechen. Jede Person und Instanz wird darüber hinaus mit einer Zugriffsberechtigung und Eintragsberechtigung versehen, was eine Überwachung des Systems im Sinne von unberechtigtem Eintragen von Informationen bewirkt. Ebenso ermöglicht das Auskunftssystem durch die unterschiedlichen Zugriffsmodi

- Zugreifen
- Ausführen

dem Informationsanbieter, Sichten auf seine Daten, Dienste und Berichte zu bilden, ebenso wie den Durchgriff auf ausgewählte Informationen für bestimmte Benutzerkreise auf dem UIK zu sperren. Um eine sichere Zugriffsverweigerung zu realisieren, muß jedoch auf Anbieterseite dennoch eine separate Rechteverwaltung implementiert werden, da WWW einen direkten Zugang unter Umgehung des Auskunftssystems jederzeit ermöglicht. Darüber hinaus verwaltet das System Adreßdaten der zugriffs- und eintragsberechtigten Personen für die Bereitstellung einer Kommunikationskomponente zwischen Anbietern und Abfragenden.

Der Umweltinformationsquellenkatalog besitzt folgende prinzipiellen Funktionen

- Bericht katalogisieren,
- Dienst katalogisieren,
- Informationen suchen.

Die Metadaten von Berichten und Diensten können in dem System über eine HTML-Oberfläche interaktiv eingetragen werden. Hierbei basiert die thematische Zuordnung auf dem Thesaurus des UDK, der beim Eintragen und Suchen von Informationen mit verschiedenen Mechanismen durchsucht und visualisiert werden kann. Der Raumbezug bezieht sich ebenfalls auf einen hierarchischen Datenbestand, der jedoch im System von den Anwendern erweitert werden kann. Basis für den Raumbezug stellt der Gazetteer dar, der sämtliche Gebiete mit Namen und umschließenden Rechteck hierarchisch gemäß Überlappungen sortiert verwaltet.

Das Katalogisieren von Diensten erfolgt durch Eintragen sämtlicher Metadaten durch den Anbieter in das System. Zum Eintragen von Berichten enthält der UIK einen Verschlagwortungs-Mechanismus, der den Bericht des Anbieters temporär auf den Metadatenserver portiert, dort basierend auf dem Thesaurus verschlagwortet, mit seinen originalen Verweisen auf den Anbieter-Server in den Katalog einträgt und abschließend die Dokumente auf dem Metadatenserver wieder löscht. Eine solche Verschlagwortung läßt sich

für die Zuordnung der Raumbezüge ebenfalls realisieren, basierend auf den Gebietsnamen des Gazetteers.

Die Informationssuche beinhaltet prinzipiell die gleichen Oberflächenfunktionen wie beim Eintragen der Daten und Berichte. Der Suchende muß die von ihm gewünschten Selektionskriterien

- Informationstyp,
- Datenhaltende Stelle,
- Sachbezüge,
- Raumbezüge und
- Zeitbezüge

setzen und das System generiert eine Ergebnisseite mit den wichtigsten Metadaten der Informationen, deren Attribute mit den gewählten übereinstimmen und auf die der Abfragende entsprechende Zugriffsrechte besitzt. Hierbei werden nicht nur einfache Übereinstimmungen der Attribute berücksichtigt, sondern vielmehr die hierarchischen Strukturen der Datenbestände Thesaurus und Gazetteer ausgewertet und mit berücksichtigt. Falls entsprechende Zugriffsrechte vorhanden sind, kann von der Ergebnisseite auf jeden Dienst oder Bericht direkt beim Anbieter zugegriffen werden. Hierzu muß keine weitere Applikation gestartet werden.

Die als Ergebnis der Projekte GLOBUS I und INTEGRAL III entstandenen Berichte in Hypertextformat bzw. UFIS-Dienste mit WWW-Anbindung wurden bereits interaktiv in das System eingetragen und selektiv abgefragt. Hierbei hat sich das System als eine einfach bedienbare, die Heterogenität überwindende Auskunftskomponente mit Brokerfunktion erwiesen, deren Ergebnisse direkt in die Arbeitsumgebung des Clientsystems integriert und dort weiterverarbeitet werden können. Auf die einzelnen Funktionalitäten und die Systemarchitektur des Umweltinformationsquellenkataloges wird in den folgenden Kapiteln detailliert eingegangen.

2.2 Die Systemarchitektur

2.2.1 Überblick

Um dem Grundgedanken eines einheitlichen Zugriffs bzw. der Bereitstellung von Informationsquellen-Objekten, sowie zugrundegelegten Qualitätsansprüchen, Zugriffstransparenz und Verknüpfbarkeit von Informationen Rechnung zu tragen, wurde bei der Erarbeitung der Systemarchitektur des Umweltinformationsquellenkataloges besonderen Wert gelegt auf:

- einen modularen Aufbau des Gesamtsystems,
- die Möglichkeit der ständigen Erweiterbarkeit,
- die Offenheit des Gesamtsystems,
- die Verwendung von Standardsystemen (Betriebssystem, Datenbank, User-Interface)
- die explizite Darstellung des dahinterliegenden Wissens und

- die Einbindung bereits bestehender Informationsquellensysteme

Der Informationsquellenkatalog fungiert in seiner Gesamtkonzeption wie ein zentrales Repository-Managementsystem, wobei die verwalteten Informationen nicht zentralisiert, sondern verteilt auf die Server der datenhaltenden Stellen vorliegen. Der Informationsquellenkatalog stellt Wissen über die Suche, die Lokalität und den Zugriff auf Informationen zur Verfügung und erfüllt damit die Funktion eines Filters und Brokers für Umweltinformationen.

Die zugrundeliegende Systemarchitektur mit allen ihren Komponenten soll es jedem Sachbearbeiter an seinem (PC-)Arbeitsplatz ermöglichen, über den Informationsquellenkatalog die von verschiedenen datenhaltenden Stellen eingebrachten Informationen (Dienste, Berichte) gezielt zu suchen, zu finden und falls gewünscht direkt darauf zuzugreifen. Dies war bisher nicht möglich, denn durch die in der Vergangenheit vorherrschenden Rahmenbedingungen bei der Entwicklung, der technischen Möglichkeiten (proprietäre Hard- und Software-Plattformen und Middleware-Komponenten) bzw. der zur Verfügung stehenden Kommunikationsinfrastruktur sind Insellösungen entstanden, die in aller Regel nur lokal nutzbar sind. Durch die Ausnutzung moderner Telematiktechniken (Kombination aus Kommunikationstechnologie und Informatik) wurden diese Einschränkungen aufgehoben. Mit der zunehmenden Verfügbarkeit offener Rechnernetze sind die technischen Grundvoraussetzungen für das Erreichen dieser Zielsetzung inzwischen vorhanden. Die verfügbaren weltweiten Rechnernetze, insbesondere das Internet und das darauf basierende World Wide Web (WWW), besitzen alle Voraussetzungen, um auch für den Einsatz des Informationsquellenkataloges die technische Basis für die Kommunikation bzw. den Zugriff auf Umweltinformationen zu ermöglichen. Der Zugriff auf die Inhalte des Informationsquellenkataloges sollte unter einer einheitlichen Oberfläche unabhängig von dem jeweiligen Betriebssystem des Arbeitsplatzrechners erfolgen können. Das World Wide Web (WWW) ist als Client-Server-System konzipiert und bietet jedem über das Internet erreichbaren Rechner die Möglichkeit, als Server-System Informationen weltweit zur Verfügung zu stellen bzw. als Client-System bereitgestellte Informationen abzurufen und erfüllt daher die aufgestellten Forderungen. Dem WWW liegt des weiteren eine hypertextartige Struktur zugrunde, die sich im Gegensatz zu „flachen“ Strukturen durch vielfältige Querverweise auszeichnet. Um nun über das World Wide Web zur Verfügung gestellte Informationen für einen Benutzer in komfortabler Weise zugänglich zu machen, werden WWW-Browser eingesetzt. Die bekanntesten Vertreter sind das kostenlos erhältliche „Mosaic“ bzw. der „Netscape-WWW-Browser“.

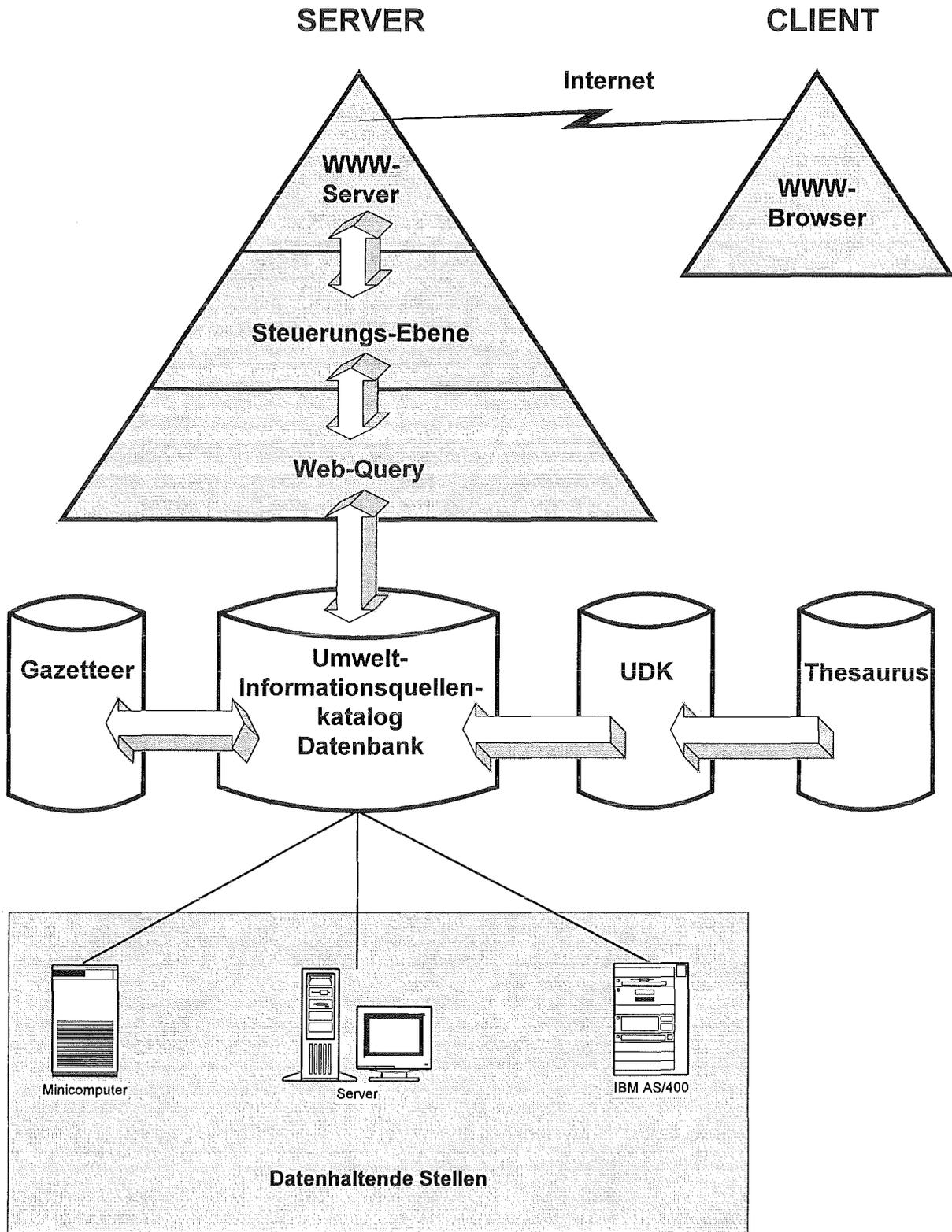


Abbildung 2-1: Systemarchitektur des Umweltinformationsquellenkataloges

Um nun Umweltinformationen über die WWW-Oberfläche des Informationsquellenkataloges identifizieren zu können, müssen im Rahmen der Systemarchitektur die entsprechenden Voraussetzungen für eine gezielte Suche geschaffen werden. Dies wird erreicht, indem alle verwalteten Umweltinformationen (UIK-Objekte) innerhalb des Repositories des Informationsquellenkataloges detailliert durch weitergehende Attribute beschrieben werden.

So sollen relevante Umweltinformationen z.B. aufgrund zugeordneter Schlagwörter (Deskriptoren), den zugeordneten Raumbezügen und / oder Zeitbezügen identifiziert und für den Benutzer transparent zur Verfügung gestellt werden. Insgesamt werden im Meta-Schema des Informationsquellenkataloges folgende beschreibende Zuordnungen für UIK-Objekte (Umweltinformationsquellenkatalog-Objekte) festgehalten und verwaltet:

- Art der Umweltinformation (Bericht, Dienst)
- Aufbau bzw. Zusammensetzung der Umweltinformation (Berichtshierarchien)
- Deskriptoren (Schlagwörter)
- Raumbezüge
- Zeitbezüge
- Benutzer- bzw. Zugriffsinformationen
- technische Zugriffsinformationen (datenhaltende Stelle, Server)

Somit wird jedem Benutzer die Möglichkeit gegeben, den Suchraum über alle zur Verfügung stehenden und verteilt auf verschiedenen Servern gehaltenen Umweltinformationen gezielt durch die Angabe der entsprechenden Suchkriterien einzuschränken. Einschränkungen können somit erfolgen:

- nach der Art der Umweltinformation (Dienst, Bericht),
- den gewünschten Themenbereichen,
- dem gewünschten Raumbezug (Gebiet) und
- dem gewünschten Zeitbezug (Zeitpunkt oder Zeitraum)

Alle Daten, die im Rahmen des Informationsquellenkataloges zur Verfügung gestellt werden, werden in einem relationalen Datenbankmanagementsystem (RDBMS) verwaltet. Durch die dynamische Verwaltung von Informationsquellen mit Hilfe einer Datenbanklösung wird u.a. die laufende Aktualität der angebotenen Informationen sichergestellt. Im Hinblick auf die räumlich verteilte und zeitlich versetzte (parallele) Nutzung des Informationsquellenkataloges, sowohl von Informationsanbietern als auch Nutzern, ist es zudem wichtig, auf der technischen Seite die erforderlichen Zugriffs- und Transaktionsmechanismen bereitzustellen, damit die Konsistenz und Integrität der verwalteten Datenbestände zu jeder Zeit sichergestellt werden kann. Der Einsatz einer relationalen Datenbank bildet hierfür den notwendigen technischen Hintergrund.

2.2.2 Der Umweltdatenkatalog (UDK)

Der Umweltdatenkatalog (UDK) ist als Meta-Informationssystem für Umweltdaten konzipiert, das Orientierung schaffen soll, welche Informationen zur Umwelt verfügbar sind. Die verwalteten Datenbestände entstehen i.d.R. in den Fachverwaltungen und liegen auch dort vor. Um diese Informationen zugänglich zu machen, versucht der UDK eine Übersicht und Transparenz darüber herzustellen, wer welche Informationen wo und wie verfügbar hält. Die Grundfunktion des UDK ist somit mit dem Auskunftssystem einer Bibliothek vergleichbar, bei dem die verwalteten Datenbestände bzgl. ihrer Inhalte und Eigenschaften beschrieben werden. Verschiedene Ordnungshinweise oder Schlagworte, die das Auffinden von Umweltinformationen bzw. auch thematisch verwandten Einträgen gestatten, sind im Rahmen des UDK über ein Thesaurus-Modul realisiert.

Das Ziel der verteilten, netzwerkweiten digitalen Bereitstellung von Umweltinformationen geht über die Grundfunktionalitäten des UDK hinaus und wurde im Rahmen der Entwicklung des Umweltinformationsquellenkataloges (UIK) verwirklicht. Die technische Basis für die netzweite Bereitstellung von Umweltinformationen innerhalb des UIK stellt das Internet / WWW dar und ersetzt somit Überlegungen hinsichtlich der angedachten Kommunikations- und verteilten Nutzungsmöglichkeiten des UDK (Modemverbindungen, Datex-P).

Bei der Planung und Entwicklung des UIK wurde der UDK als Basis verwendet. Somit setzt der Umweltinformationsquellenkatalog auf den Grundkonzepten des UDK auf, gestattet die durchgängige Integration von Meta-Komponenten des UDK (UDK-Objekt, Thesaurus) in den UIK und erweitert diese Konzepte zum einen bzgl. den Anforderungen einer verteilten netzwerkweiten Bereitstellung, Nutzung und dem direkten Zugriff auf Umweltinformationen. Zum anderen werden innerhalb des UIK erweiterte Konzepte hinsichtlich des parametrisierten Zugriffs auf UIK-Objekte realisiert. Diese beziehen sich vor allem auf die Darstellung und die Nutzung von räumlichen Zuordnungen zu Umweltinformationen, die innerhalb des Umweltinformationsquellenkataloges verwaltet werden.

2.2.3 Das Datenbankschema des Umweltinformationsquellenkatalogs

Die Verwaltung aller Informationen des Umweltinformationsquellenkatalogs wird mit Hilfe des relationalen Datenbankmanagementsystems Oracle7 realisiert. Das nachfolgend abgebildete ER-Modell stellt das Datenbankschema dar und besteht i.w. aus den vier Hauptkomponenten:

- Informationsquellenverwaltung (UIK-Objekte)
- Benutzer- und Rechteverwaltung
- Thesaurus
- Gazetteer (Ortsverzeichnis und Raumbezüge)

Die Entwicklungen, die im Rahmen des UDK vorgenommen worden sind, stellen mit ihren Ergebnissen die wesentliche Basis für den Umweltinformationsquellenkatalog dar. So entspricht von seiner Definition her ein UIK-Objekt i.w. einem UDK-Objekt mit der zusätzlichen Erweiterung bzw. Einführung einer Objektklassifizierung nach Berichten und Diensten als die eigentlichen Ausprägungen von UIK-Objekten.

Des Weiteren wird für die Zuordnung von Schlagwörtern und Themenbereichen der Thesaurus des UDK verwendet und in seiner vorhandenen Struktur vollkommen unverändert in das Datenbankschema des UIK integriert. Die Zuordnung von Deskriptoren des Thesaurus zu den UIK-Objekten wird über eine Beziehungsrelation (Deskriptor_Referenz) realisiert und erlaubt so die uneingeschränkte Zuordnung von Thesaurus-Begriffen zu UIK-Objekten, ohne Änderungen am Thesaurus vornehmen zu müssen.

Die Abbildung von UDK-Instanzen wird über die Tabelle „Instanz“ ermöglicht. Hier sind durch die modellierte „Teil_von“-Beziehung der Tabelle „Instanz“ beliebige Hierarchien von UDK-Instanzen abbildbar.

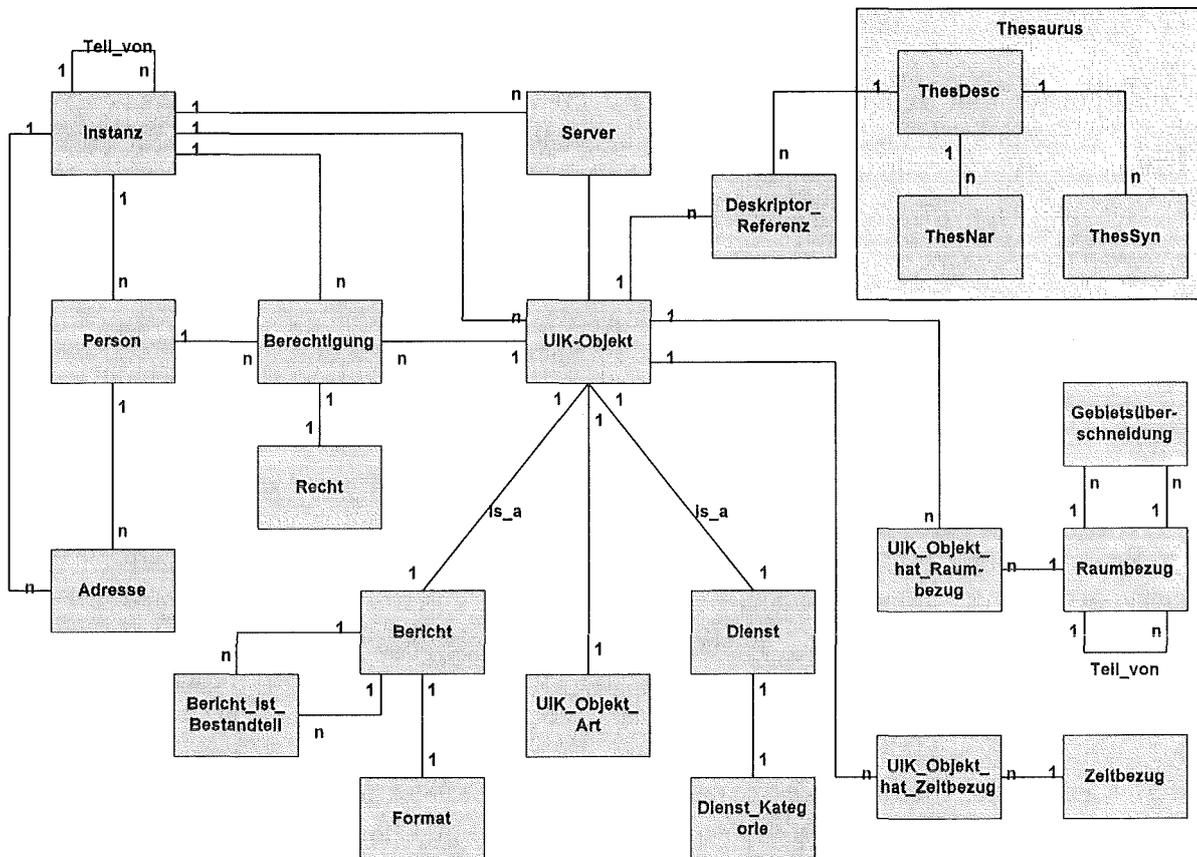


Abbildung 2-2: Entity-Relationship-Modell des Umweltinformationsquellenkatalogs

Zwischen einem UIK-Objekt und einem UDK-Objekt besteht eine logische 1:1 Beziehung. Dies bedeutet, daß ein UIK-Objekt einem UDK-Objekt entspricht, mit dem Unterschied, daß die Definition von UIK-Objekten um zusätzliche Eigenschaften erweitert wurde.

2.2.3.1 Informationsquellenverwaltung

Zu der Komponente „Informationsquellenverwaltung“ sind folgende Tabellen des Datenbankschemas zu zählen:

- UIK-Objekt
- UIK-Objekt_Art
- Bericht
- Format
- Bericht_ist_Bestandteil
- Dienst
- Dienst_Kategorie
- Server

Die Tabelle „UIK-Objekt“ stellt die eigentliche Zugriffsklasse aller verwalteten Umweltinformationsquellen dar. Sowohl die Suche nach Informationsquellen, als auch das Eintragen von neuen Informationsquellen geschieht über die Tabelle „UIK-Objekt“. Sie stellt des weiteren die Generalisierung von Berichten und Diensten, die innerhalb des Umweltinformationsquellenkataloges verwaltet werden, dar.

Beim Eintragen von Berichten oder Diensten wird sowohl ein Eintrag in der Tabelle „UIK-Objekt“ vorgenommen, als auch ein Eintrag in der jeweiligen spezialisierten Tabelle (Bericht oder Dienst). Dabei werden innerhalb der Tabelle „UIK-Objekt“ alle Eigenschaften von UIK-Objekten abgelegt, die für alle Spezialisierungen identisch sind. Innerhalb der jeweiligen Spezialisierungen von UIK-Objekten, also Berichten oder Diensten, werden dann zusätzlich die Eigenschaften abgebildet, die hierfür charakteristisch sind. Die Objektidentität bzw. die referenzielle Integrität über die Tabellen „UIK-Objekt“ und „Bericht“ bzw. „Dienst“ wird über eine tabellenübergreifende, eindeutige Objekt-ID sichergestellt. Über die zusätzlich zum statischen Teil (Datenbankschema) notwendigen Funktionen läßt sich sowohl beim Eintragen, als auch bei der Auswertung eine Spezialisierungshierarchie („is_a“ - Beziehung) zwischen UIK-Objekten und Berichten / Diensten realisieren. Der Vorteil einer solchen Lösung liegt zum einen in der homogenen Zugriffs- bzw. Eintragsverwaltung, da unabhängig von der jeweiligen Art einer Informationsquelle immer die Tabelle „UIK-Objekt“ ausgewertet bzw. instanziiert wird. Darüberhinaus lassen sich durch diese Architektur der Informationsquellenverwaltung mühelos weitere Arten von Informationsquellen unter die Tabelle „UIK-Objekt“ einhängen und sich damit das Schema bzw. der Informationsgehalt erweitern, ohne daß Änderungen an bereits realisierten Funktionen des Umweltinformationsquellenkataloges vorgenommen werden müssen. Auch weitergehende, tiefer geschachtelte Hierarchien lassen sich durch diese Architekturvorgabe realisieren, so daß z.B. Dienste oder Berichte - falls gewünscht oder benötigt - noch feiner gegliedert werden könnten. Zusammenfassend läßt sich somit sagen, daß sich sowohl horizontale, als auch vertikale Erweiterungen bzw. Ergänzungen mühelos mit der aufgezeigten Architektur der Informationsquellenverwaltung umsetzen lassen.

Über die Tabelle „Bericht_ist_Bestandteil“ lassen sich zudem die für Berichte charakteristischen Berichtshierarchien aufbauen und verwalten. Die dazu notwendigen Einträge in die Tabellen „Bericht“ und „Bericht_ist_Bestandteil“ werden beim Eintragen eines neuen Berichtes automatisch vorgenommen, wobei der Zugriffsschlüssel eines Berichtes der eindeutig vergebenen Objekt-ID eines UIK-Objektes entspricht, welche gleichzeitig dazu angelegt wird. Zudem wird jedem Bericht über die Tabelle „Format“ ein bestimmtes Darstellungsformat zugewiesen. Eine Auswertung dieses Eintrages kann dafür genutzt werden, um abhängig vom jeweiligen Repräsentationsformat (ASCII, EPS, HTML,..) des Berichtes eine für den Benutzer relevante Darstellungsart zu wählen und die dafür notwendigen Darstellungsprogramme (Editoren, Viewer) zu starten.

Tabelle "Bericht_ist_Bestandteil"

abzubildende Berichts-Hierarchie

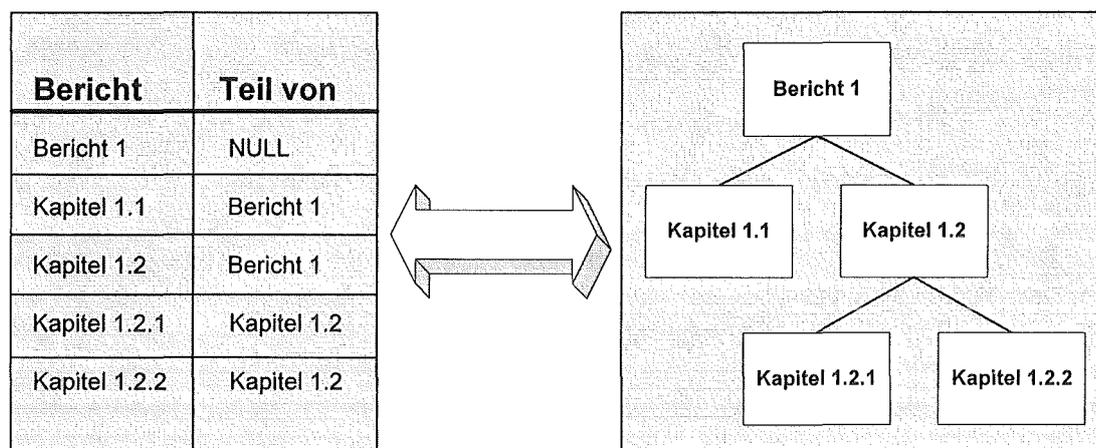


Abbildung 2-3: Berichtshierarchie

Mit Hilfe der Tabelle „Dienst“ lassen sich Dienste, die durch den Umweltinformationsquellenkatalog verwaltet werden, abbilden. Jeder Dienst wird über die Tabelle „Dienst_Kategorie“ einer bestimmten Kategorie zugeordnet. Dies soll für eine später zu realisierende Trader-Funktionalität für Dienste genutzt werden und im Zusammenwirken mit einer noch zu realisierenden Dienste-Hierarchie das Auffinden bzw. Identifizieren von geeigneten bzw. ähnlichen Diensten ermöglichen. Darüberhinaus ist im vorliegenden Modell ein Dienst bereits mit Input- und Output-Feldern modelliert und im Datenbankschema eingetragen. Durch die Einführung und Verwendung von Eingabe- und Ausgabe-Beschreibungen zur Schnittstellencharakterisierung von Diensten verfügt die realisierte Strukturierung und Modellierung von Diensten bereits jetzt über die notwendige Ausdrucksmächtigkeit, um im Rahmen einer verteilten Ausführungsumgebung wie z.B. CORBA Verwendung finden zu können. Die im Rahmen der CORBA-Spezifikation festgeschriebene IDL (Interface Definition Language) zur plattformunabhängigen Beschreibung der Dienste-Schnittstellen kann in der hier erarbeiteten Dienste-Struktur verwendet werden.

Damit auf Berichte und Dienste, also auf alle UIK-Objekte, die durch den Umweltinformationsquellenkatalog verwaltet werden, direkt durchgegriffen werden kann, besitzt jedes UIK-Objekt eine logische Zugriffsadresse. Diese logische Zugriffsadresse (URL) bezieht sich auf einen beim jeweiligen Anbieter vorhandenen UIK-Objekt-Server. Die für einen direkten Zugriff relevanten Daten werden innerhalb der Tabelle „Server“ gehalten und über eine entsprechende Fremdschlüsselvergabe mit der Tabelle „UIK-Objekt“ verknüpft. Auch an dieser Stelle kommt wieder der bereits oben angesprochene Vorteil zum Tragen, daß die Informationsquellenverwaltung (UIK-Objekte) über eine Spezialisierungs-Hierarchie realisiert worden ist. Erst dadurch wird es auf eine einfache Art und Weise möglich, mit nur einer Beziehung innerhalb des Datenbankschemas (UIK-Objekt <--> Server) eine Adreßvergabe für alle Arten von UIK-Objekten einheitlich zu gestalten und für evtl. notwendige Erweiterungen offen zu halten.

2.2.3.2 Benutzer- und Rechteverwaltung

Zu der Komponente „Benutzer- und Rechteverwaltung“ sind folgende Tabellen des Datenbankschemas zu zählen:

- Instanz
- Person
- Adresse
- Berechtigung
- Recht

Um den Zugriff auf UIK-Objekte regeln und verwalten zu können, ist eine explizite Benutzer- und Rechteverwaltung notwendig. Die für bestimmte Nutzer vergebenen Änderungs- bzw. Zugriffsrechte werden innerhalb der Tabelle „Berechtigung“ verwaltet, die wiederum auf die Sekundärtabelle „Recht“ zugreift. Die Abbildung bzw. Verwaltung aller zur Verfügung stehenden Rechtearten in einer eigenen Tabelle hat den Vorteil, daß bei Änderungen nur diese Tabelle betroffen ist und eine Erweiterung des Rechteverzeichnisses problemlos vorgenommen werden kann.

Innerhalb der Tabelle „Berechtigung“ werden die Beziehungen zwischen Instanzen, Personen, Rechten und UIK-Objekten abgebildet. Die Auswertung der jeweiligen Rechte beim Zugriff auf den Umweltinformationsquellenkatalog geschieht im Zusammenspiel mit der Abbildung von Organisationsstrukturen über die Tabellen „Instanz“ und „Person“. So ist es zum einen möglich, Personen bestimmten Instanzen zuzuordnen, die wiederum einer übergeordneten Instanz unterstellt sein kann usw. Grundsätzlich sind alle Instanzen und Personen der Instanz (Gruppe) „ALLE“ direkt, oder indirekt über andere Instanzen zugeordnet und besitzen damit zum Zeitpunkt ihrer Bekanntmachung im System automatisch alle Rechte dieser Gruppe. Die Instanz „Alle“ stellt das Wurzelobjekt der Instanzenhierarchie dar (Abbildung 2-4: Instanzen- und Personenhierarchie) und vererbt alle Rechte, die dieser Gruppe zugeordnet sind, an die untergeordneten Instanzen.

Die Abbildung von Personen und Organisationsstrukturen innerhalb der Architektur des Umweltinformationsquellenkataloges folgt folgendem Schema:

- Personen bilden die Blätter der Personen- und Instanzenhierarchie, sind also am Ende der Hierarchie angesiedelt und i.d.R. immer einer bestimmten Instanz zugeordnet.
- Instanzen bilden die Knoten innerhalb der Instanzenhierarchie und können Beziehungen zu mehreren nachfolgenden Instanzen, aber immer nur zu einer übergeordneten Instanz besitzen.

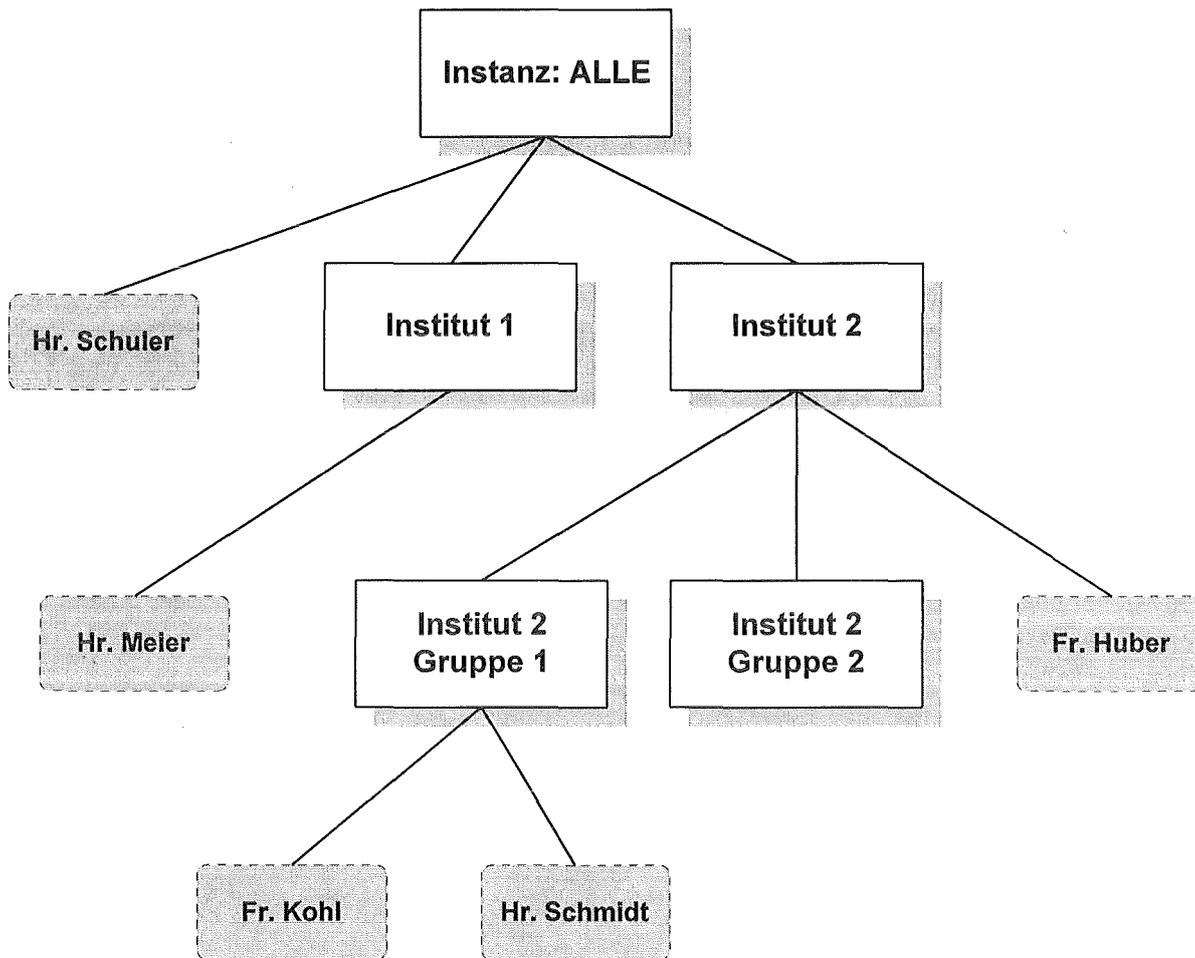


Abbildung 2-4: Instanzen- und Personenhierarchie

Durch diese Art der Strukturierung von organisatorischem Wissen ist es nicht mehr nötig, für alle Personen, die jemals auf den Umweltinformationsquellenkatalog zugreifen, gesondert Zugriffsrechte zu vergeben. Gehört eine Person z.B. einer bestimmten Gruppe (Instanz) an, so reicht es, wenn für diese Instanz Zugriffsrechte innerhalb der Tabelle „Berechtigung“ vergeben sind. Genau die für die jeweilige Instanz definierten Rechte erbt dann auch die Person, die den Umweltinformationsquellenkatalog nutzen möchte. Existieren jedoch für die Person gesonderte Zugriffsrechte, so geht Personenrecht stets vor Gruppenrecht. Dieser Mechanismus garantiert eine enorme Änderungsfreundlichkeit, was vor allem den sich immer wieder ändernden Organisations- und Ablaufstrukturen von Organisationen (Instanzen) zu Gute kommt. So müssen z.B. einem neuen Mitarbeiter einer Instanz nicht explizit Rechte auf UIK-Objekte erteilt werden. Es ist vollkommen ausreichend, wenn der neue Mitarbeiter in die Tabelle „Person“ eingetragen und dabei festgehalten wird, daß er der jeweiligen Abteilung angehört. Damit ist er automatisch im Besitz aller Gruppenrechte dieser Abteilung. Natürlich ist es zudem möglich, dem Mitarbeiter gesonderte Rechte zu geben.

Die Auswertung der abgebildeten Organisationsstrukturen und Hierarchien und der damit in Beziehung stehenden Vergabe von Zugriffs- bzw. Änderungsrechten auf UIK-Objekte über die Tabelle „Berechtigung“ wird über entsprechende Datenbankzugriffe realisiert. Dabei wird folgende Vorgehensweise gewählt:

1. jeder Benutzer identifiziert sich beim System mit Benutzername und Paßwort (Account)
2. über eine Auswertung der Tabelle „Person“ bzw. „Instanz“ wird geprüft, ob die vom Benutzer eingegebene Benutzerkennung sowie das Paßwort richtig sind
3. beim Zugriff auf UIK-Objekte wird geprüft, ob für die Person oder die Gruppe (Instanz) innerhalb der Tabelle „Berechtigung“ Rechte vergeben sind. Sind Berechtigungen eingetragen, werden diese beim Zugriff auf das jeweilige UIK-Objekt ausgewertet.
4. sind für die Person oder die Gruppe keine Rechte eingetragen, wird innerhalb der abgebildeten Organisationshierarchie nach der übergeordneten Instanz gesucht und nach eingetragenen Rechten für die gefundene Instanz innerhalb der Tabelle Berechtigung gesucht. Führt dies auch zu keinem Erfolg, wird in der Instanzenhierarchie eine weitere Stufe nach oben gegangen usw., bis eingetragene Berechtigungen gefunden werden.

Um sicherzustellen, daß für alle eingetragenen Personen und Instanzen Zugriffsberechtigungen auf UIK-Objekte vergeben sind und überprüft werden können, werden beim Eintragen neuer Personen und Instanzen diese sofort der Instanz (Gruppe) „Alle“ zugeordnet, sofern sie nicht einer anderen Instanz unterstellt werden. Somit ist immer sichergestellt, daß neue Benutzer zumindest die Rechte „aller“ besitzen und keine explizite Rechtevergabe für die Benutzung des Systems notwendig ist.

2.2.3.3 Thesaurus

Zu der Komponente „Thesaurus“, die aus dem UDK Verwendung findet, sind folgende Tabellen des Datenbankschemas zu zählen:

- ThesDesc
- ThesNar
- ThesSyn

Die oben aufgeführten Tabellen wurden für den Umweltinformationsquellenkatalog übernommen und als Teil des UDK eingebunden. In der Tabelle „ThesDesc“ werden die Deskriptoren des Thesaurus gespeichert. In der Tabelle „ThesNar“ wird eine Verbindung zwischen den Deskriptoren und ihren Unterbegriffen festgelegt. Da sowohl Unter- als auch Oberbegriffe Deskriptoren sind, wird innerhalb dieser Tabelle darauf verzichtet, die Begriffe (Deskriptoren) nochmals abzulegen. Statt dessen werden für die Abbildung der Oberbegriff-Unterbegriff-Beziehung in dieser Tabelle nur die Zugriffsschlüssel der Deskriptoren verwendet. Die Tabelle „ThesSyn“ speichert die Synonyme zu einem Deskriptor. Durch die vollständige, unveränderte Verwendung und Einbindung des Thesaurus des UDK kann dieser in bekannter Weise verwendet und für den Umweltinformationsquellenkatalog genutzt werden. Dies gestattet auch, Änderungen am Thesaurus ohne Änderungen am Datenbankschema des Umweltinformationsquellenkataloges vornehmen zu können, solange die Struktur des Thesaurus in der hier beschriebenen Art beibehalten wird.

Über die Beziehungstabelle „Deskriptor_Referenz“ des Datenbankschemas ist es nun möglich, einem UIK-Objekt mehrere Deskriptoren zuzuordnen (Abbildung 2-5: Zuordnung von Deskriptoren zu UIK-Objekten). In dieser Beziehungstabelle wird ebenfalls auf eine

redundante Speicherung von Deskriptoren bzw. UIK-Objekten verzichtet. Für die Herstellung der Referenz-Beziehung zwischen UIK-Objekten und Deskriptoren werden nur die jeweiligen Schlüsselattribute *UIK-Objekt-ID* und *desc_no* verwendet.

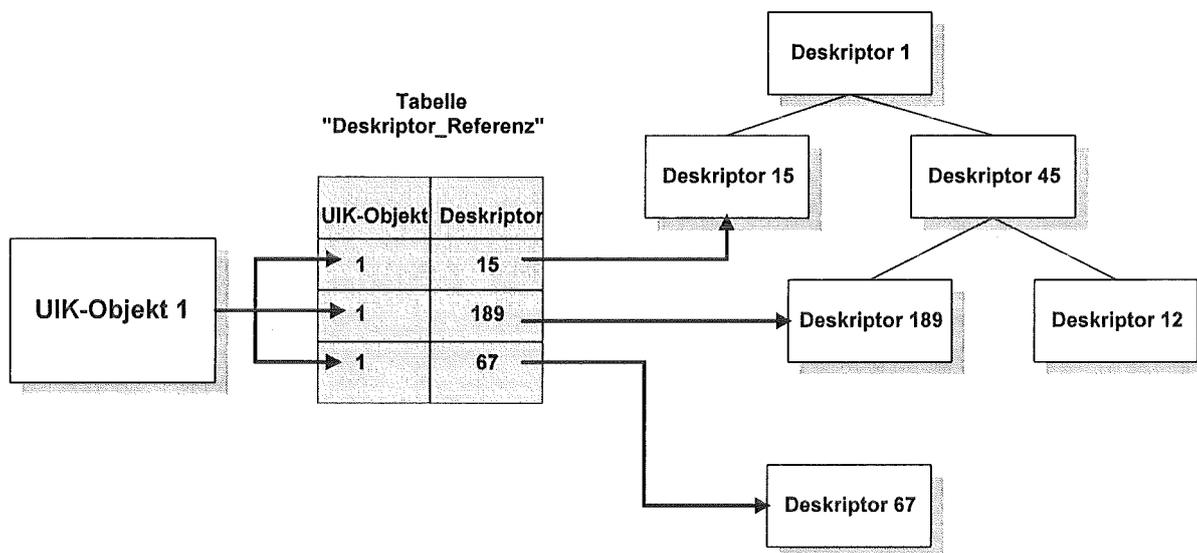


Abbildung 2-5: Zuordnung von Deskriptoren zu UIK-Objekten

Über die eindeutige UIK-Objekt-ID werden die zugeordneten Deskriptoren den modellierten Spezialisierungen des UIK-Objektes zugänglich gemacht. (siehe dazu: 2.2.3.1 Informationsquellenverwaltung).

2.2.3.4 Gazetteer

Zu der Komponente „Gazetteer“ sind folgende Tabellen des Datenbankschemas zu zählen:

- Raumbezug
- Gebietsüberschneidung

In der Tabelle „Raumbezug“ werden Gebiete (Gebietsbezeichnungen) und deren geographische Koordinaten abgelegt. Durch die Struktur selbst ist keine Beschränkung hinsichtlich der Granularität der Gebietsgröße gegeben. Die Organisation der dargestellten Gebiete beschränkt sich nicht auf implizites Wissen über Gebietsnamen und Koordinaten, die zu jedem Gebiet abgelegt werden. Darüberhinaus werden die dargestellten Gebiete im Sinne einer „Teil_von“-Beziehung untereinander strukturiert und organisiert.

Um dies mittels des dargestellten Schemas zu realisieren, wurden folgende Maßnahmen getroffen (siehe auch Abbildung 2-6: Verwendung des Gazetteers):

- jedes Gebiet (Raumbezug) wird durch eine eindeutige ID beschrieben (Raumbezugs-ID). Diese ID bildet den Zugriffsschlüssel auf den Raumbezug.
- jeder Eintrag in der Tabelle „Raumbezug“ erhält zusätzlich das Attribut „Teil_von“. Die Einträge für dieses Attribut entsprechen wieder einer Raumbezugs-ID und verweisen damit auf den Raumbezug, der dem aktuellen übergeordnet ist.

- Der Raumbezug, der an der Spitze der sich daraus ableitbaren Hierarchie steht, besitzt keinen Eintrag mehr im Feld „Teil_von“ und ist damit eindeutig als Wurzel-Objekt im Kontext der Raumbezüge identifizierbar.

Tabelle "Raumbezug"

darzustellende Gebiete

Raumbezug	Teil_von
Erde	
Europa	Erde
BRD	Europa
Österreich	Europa
Baden-W.	BRD
Stuttgart	Baden-W.

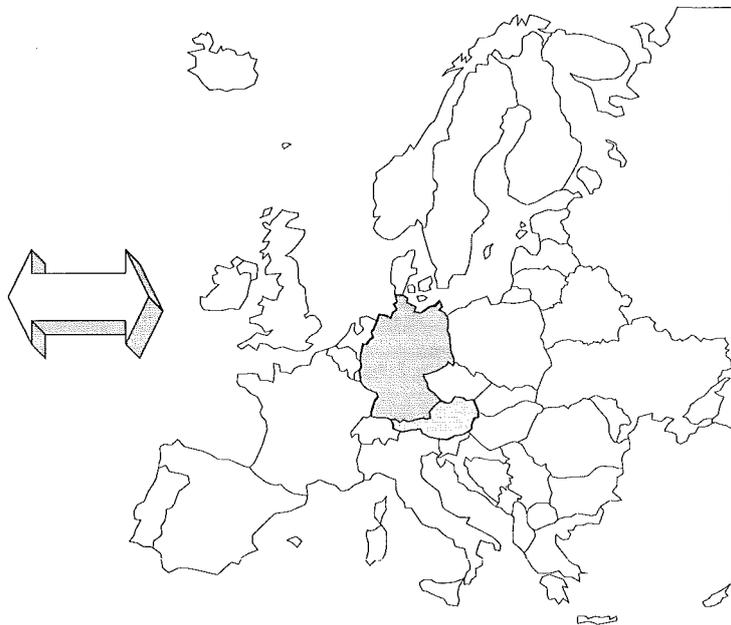


Abbildung 2-6: Verwendung des Gazetteers

Über die Beziehungstabelle „UIK-Objekt_hat_Raumbezug“ lassen sich nun Raumbezüge (Gebiete) UIK-Objekten zuordnen. In dieser Beziehungstabelle wird wiederum auf eine redundante Speicherung von Raumbezügen (Gebieten) bzw. UIK-Objekten verzichtet. Für die Herstellung der Referenz-Beziehung zwischen UIK-Objekten und Gebieten werden nur die jeweiligen Schlüsselattribute *UIK-Objekt-ID* und *RB_ID* verwendet.

Die realisierte Struktur von Raumbezügen gestattet es, unabhängig von der Suche über Koordinaten oder Gebietsnamen Strukturen und Hierarchien von Gebieten erkennen zu lassen. Diese Eigenschaft läßt sich sowohl beim Suchen bzw. Zuordnen von Raumbezügen zu UIK-Objekten, als auch beim Eintragen neuer Raumbezüge sinnvoll nutzen.

Die im Gazetteer aufgebauten Gebietshierarchien bilden den Suchraum für die Identifikation von relevanten Raumbezügen und damit den zugeordneten UIK-Objekten. Im Rahmen der Auswertung des Gazetteers für die Identifikation von UIK-Objekten finden zwei sich ergänzende Methoden Verwendung:

1. eine Auswertung des Suchraumes über die „Teil_von“ - Beziehung von Raumbezügen (vertikale Auswertung)
2. eine Auswertung des Suchraumes bzgl. Gebietsüberschneidungen von Raumbezügen über die Tabelle „Gebietsüberschneidung“ (horizontale Auswertung)

1. Auswertung von „Teil_von“ - Beziehungen zwischen Raumbezügen (vertikale Auswertung)

Durch Ausnutzung und Interpretation der „Teil_von“ - Beziehung werden beim Suchen von UIK-Objekten auch die UIK-Objekte gefunden, die nicht direkt einen angegebenen Raumbezug referenzieren. Die „Teil_von“ - Beziehung wird jedoch nicht nur unidirektional - von oben nach unten - sondern bidirektional ausgenutzt und für die Suche nach relevanten UIK-Objekten herangezogen.

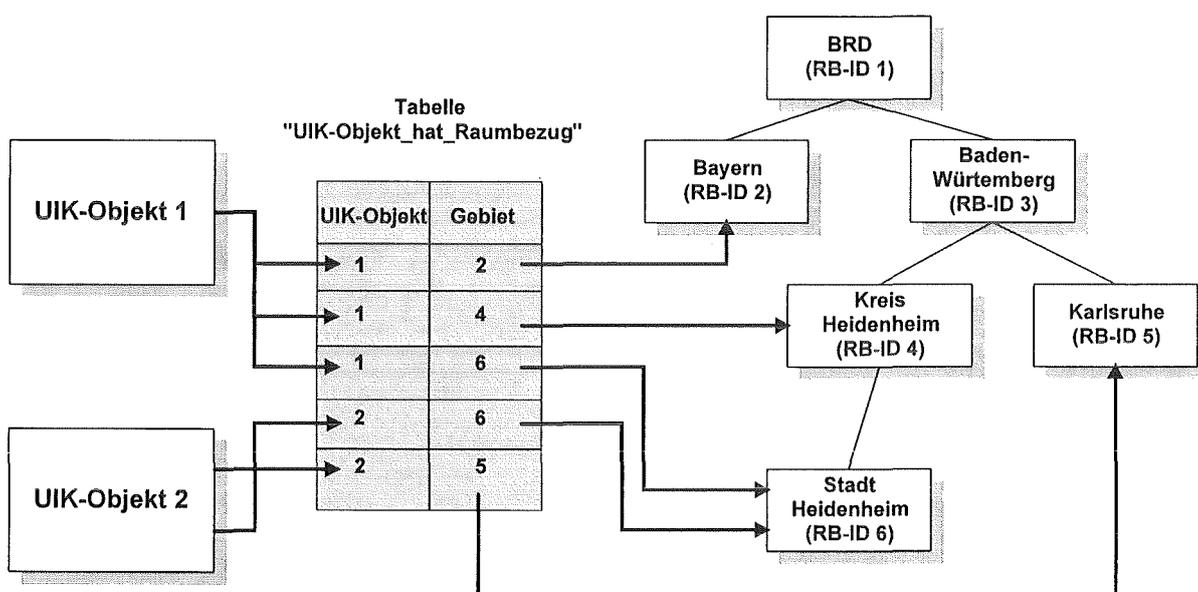


Abbildung 2-7: Zuordnung von Raumbezügen zu UIK-Objekten

Zur Veranschaulichung der Nutzungsmöglichkeiten der modellierten Beziehung zwischen Raumbezügen soll für das folgende Beispiel Abbildung 2-7: „Zuordnung von Raumbezügen zu UIK-Objekten“ herangezogen werden.

Wird z.B. bei einer Suche nach UIK-Objekten der Raumbezug 4 „Kreis Heidenheim“ als für die Suche relevant angegeben, so werden über eine geeignete Interpretation und Auswertung der Beziehungen zwischen den Raumbezügen auch UIK-Objekte gefunden, die über die Tabelle „UIK-Objekt_hat_Raumbezug“ Gebiete referenzieren, die bei der Definition der Suchkriterien nicht explizit aufgeführt worden sind. Dies bedeutet, daß im oben angegebenen Beispiel nicht nur das UIK-Objekt 1, sondern auch das UIK-Objekt 2 gefunden und in die Ergebnisliste mitaufgenommen werden würde. Das UIK-Objekt 2 hat u.a. den Raumbezug „Stadt Heidenheim“. Dieses Gebiet ist aber im Gazetteer als Teil des Gebietes „Kreis Heidenheim“ eingetragen. Gibt der Informationssuchende nun wie im vorliegenden Beispiel als Raumbezug „Kreis Heidenheim“ an, so ist es nur richtig, auch alle UIK-Objekte zu suchen, die Raumbezüge referenzieren, die dem gewünschten Raumbezug untergliedert sind und somit innerhalb des angegebenen Gebietes liegen.

Da die Suche nach relevanten UIK-Objekten, wie oben bereits erwähnt, nicht nur unidirektional vom angegebenen Raumbezug zu den darunterliegenden erfolgt, sondern auch vom angegebenen Raumbezug zu den übergeordneten, werden alle UIK-Objekte gefunden, die einen bestimmten Pfad innerhalb der Gebietshierarchie referenzieren. Dabei ist es völlig

unerheblich, ob bei der Suche nach UIK-Objekten die Raumbezüge durch Angabe ihrer Koordinaten oder durch Angabe des Gebietsnamens definiert werden.

2. Auswertung von Gebietsüberschneidungen (horizontale Auswertung)

Sollen bei der Suche nach UIK-Objekten nicht nur jene gefunden werden, die im Rahmen der definierten Raumbezüge direkt oder über die oben beschriebene „Teil_von“-Beziehung referenziert sind, sondern auch jene, die mit einem Raumbezug verknüpft sind, der den bei der Suche definierten Raumbezug schneidet, muß zusätzlich zur vertikalen Auswertung des Suchraumes auch eine Auswertung der Raumbezüge hinsichtlich von Gebietsüberschneidungen erfolgen.

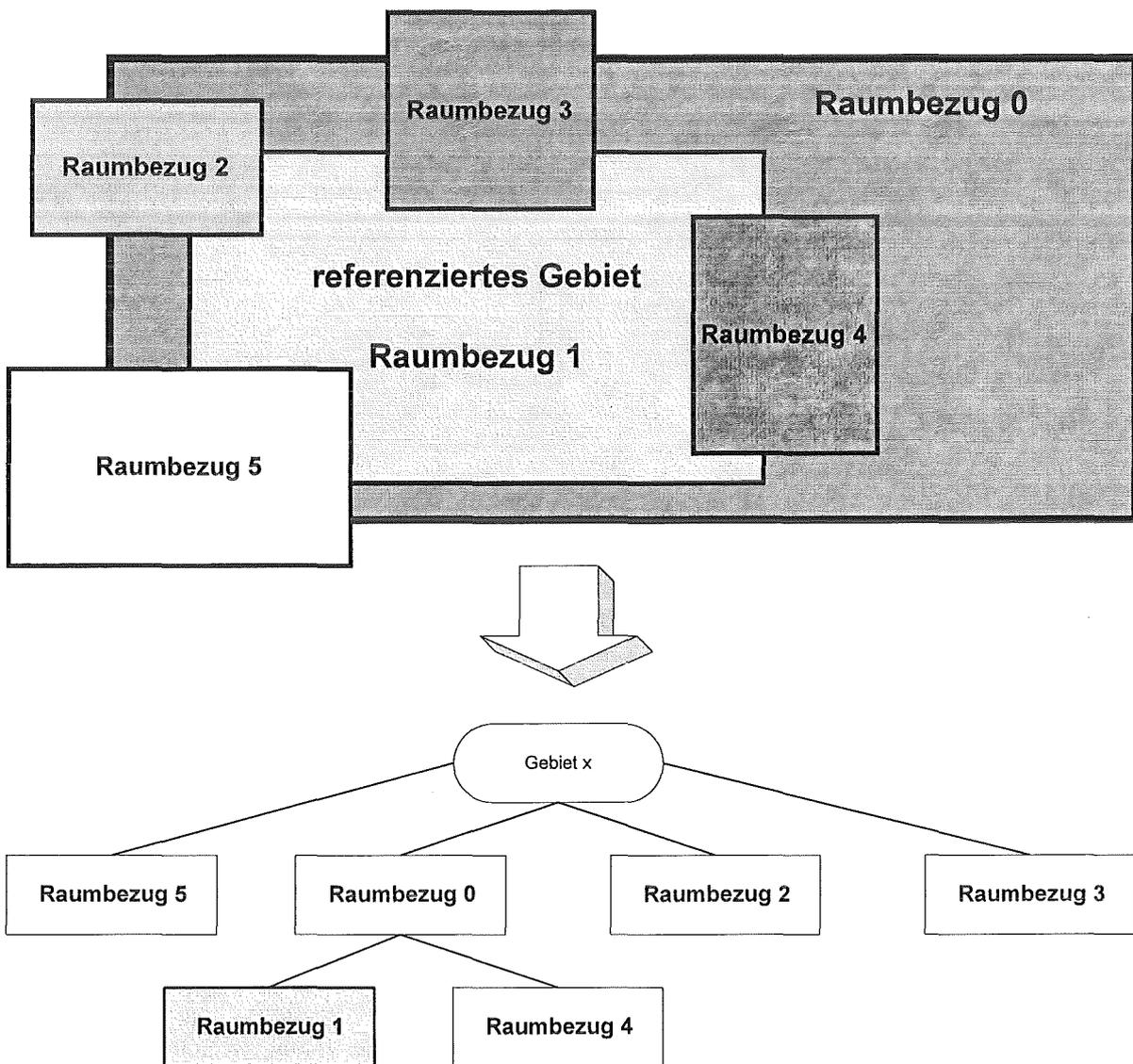


Abbildung 2-8: Abbildung sich überschneidender Gebiete im Gazetteer

Wird wie in Abbildung 2-8: „Abbildung sich überschneidender Gebiete im Gazetteer“ gezeigt, der Raumbezug 1 innerhalb einer Suche nach UIK-Objekten als relevant angegeben, so wird mittels der vertikalen Auswertung des Gazetteer's der Raumbezug 0 und damit die diesem Raumbezug zugeordneten UIK-Objekte gefunden. Da jedoch wie in Abbildung 2-8 gezeigt andere Raumbezüge (RB2, RB3, RB4, RB5) den referenzierten Raumbezug 1

schneiden, sollen auch diejenigen UIK-Objekte gefunden werden, die Beziehungen zu diesen Raumbezügen aufweisen.

Dies kann nun nicht mehr über eine Auswertung der modellierten „Teil_von“-Beziehung (vertikale Auswertung) geschehen, da die aufgeführten Raumbezüge nicht mehr Teil des Raumbezugs 1 sind, sondern diesen nur überlappen. Zur Erreichung dieses Ziels muß nun die Tabelle „Gebietsüberschneidung“ ausgewertet werden. Innerhalb dieser Tabelle werden die Gebietsüberschneidungen von Raumbezügen festgehalten. Da ein bestimmter Raumbezug von mehreren anderen Raumbezügen überlappt werden kann, kann diese Beziehung nicht mehr im Rahmen der Modellierung der Tabelle „Raumbezug“ erfolgen, sondern muß in eine externe Tabelle ausgelagert werden. Zu diesem Zweck werden innerhalb der Tabelle „Gebietsüberschneidung“ die bestehenden Beziehungen zwischen Raumbezügen mit Hilfe der für Raumbezüge vergebenen IDs abgebildet.

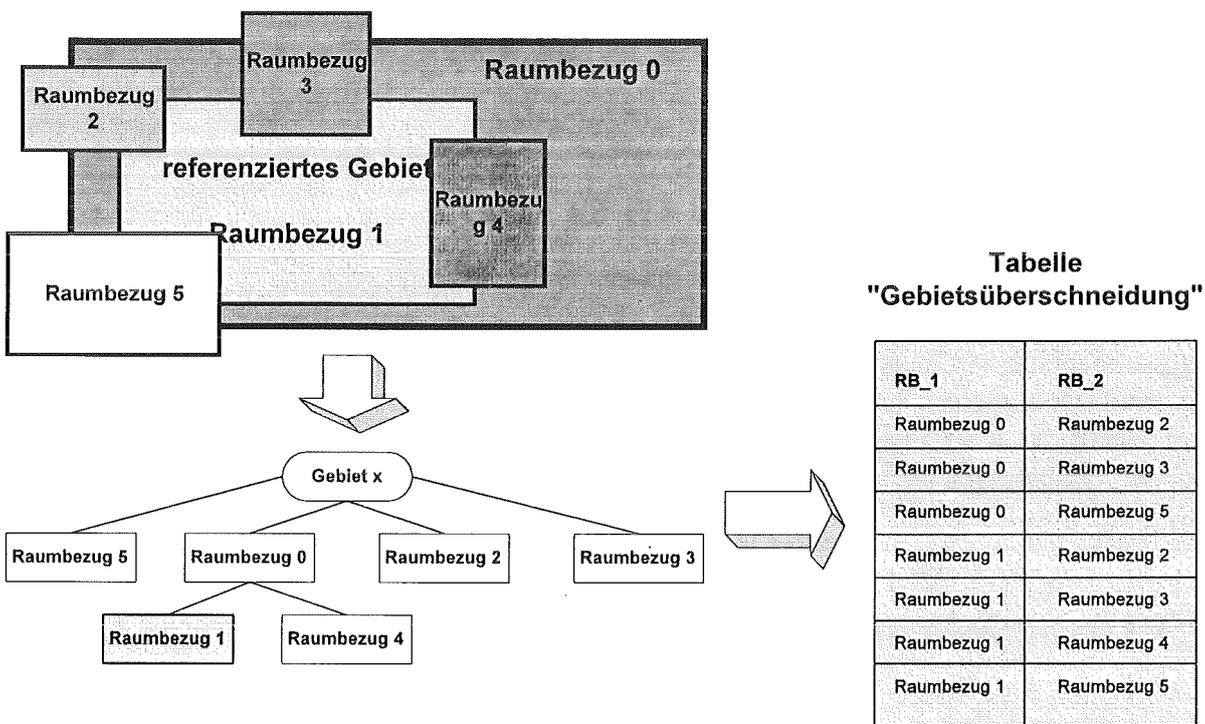


Abbildung 2-9: Tabelle „Gebietsüberschneidung“

Wird sowohl die Auswertung der „Teil_von“-Beziehung, als auch der Gebietsüberschneidungen vorgenommen, können alle UIK-Objekte, welche auf die gefundenen Raumbezüge verweisen, in die Ergebnisliste der gefundenen UIK-Objekte eingetragen werden. Um die UIK-Objekte zu finden, wird die Tabelle „UIK-Objekt_hat_Raumbezug“ ausgewertet.

3. Eintragen von Raumbezügen

Für den Aufbau der Raumbezugs-Hierarchie des Gazetteers sowie für die spätere vertikale und horizontale Auswertung ist es allerdings notwendig, daß bereits beim Eintragen neuer Raumbezüge diese innerhalb der Hierarchie richtig eingeordnet werden. Dazu wird mittels eines Koordinatenvergleichs der bereits innerhalb des Gazetteers vorhandenen Raumbezugseinträge der relevante Einfügeknoten gesucht. Anschließend wird der neue Raumbezug über dem gefundenen Raumbezug eingetragen. Danach wird die „Teil_von“-

Beziehung des gefundenen Einfügeknotens auf die beim Eintragen des neuen Knotens vergebene RB-ID gesetzt. Somit ist der neue Knoten an der richtigen Stelle innerhalb der Hierarchie eingeordnet worden.

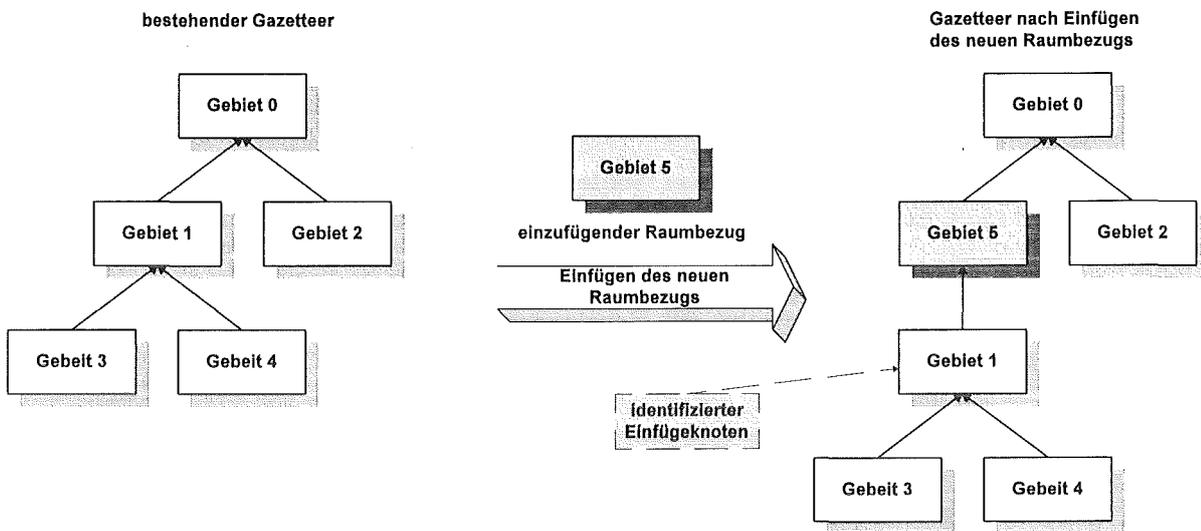


Abbildung 2-10: Einfügen eines neuen Raumbezuges

Um zusätzlich eine Auswertung von Gebietsüberschneidungen realisieren zu können, müssen nun die für jeden neuen einzutragenden Raumbezug angegebenen Koordinaten mit den bereits eingetragenen Koordinaten von Raumbezügen verglichen werden. Werden nun mit Hilfe eines Koordinatenvergleichs Überschneidungen der untersuchten Gebiete zum neuen Raumbezug festgestellt, so werden die überlappenden Raumbezüge in die Tabelle „Gebietsüberschneidungen“ eingetragen (siehe auch: Abbildung 2-9: Tabelle „Gebietsüberschneidung“).

2.3 Dialogführung

2.3.1 Überblick

Da der Zugriff auf den Umweltinformationsquellenkatalog über das World Wide Web realisiert wird, erfolgt der Zugang für jeden Benutzer über eine Einstiegsseite des UIK. Dem Benutzer wird dabei eine Auswahl über die drei generellen, im Rahmen des Umweltinformationsquellenkatalogs realisierten Dialogwege, angeboten. Der Benutzer hat die Auswahl zwischen:

1. neuen Bericht katalogisieren
2. neuen Dienst katalogisieren
3. nach UIK-Objekten suchen

Der Benutzer hat die Möglichkeit, eine der angebotenen Auswahlmöglichkeiten zu treffen und verzweigt dann zu den jeweiligen Folgeseiten des Umweltinformationsquellenkataloges. Stehen auf einer dargestellten Seite der Anwendung mehrere unterschiedliche Aktionen zur

Verfügung, so kann sich der Benutzer für eine Aktion entscheiden und diese anschließend ausführen lassen (z.B. Aktion: Raumbezug browsen).

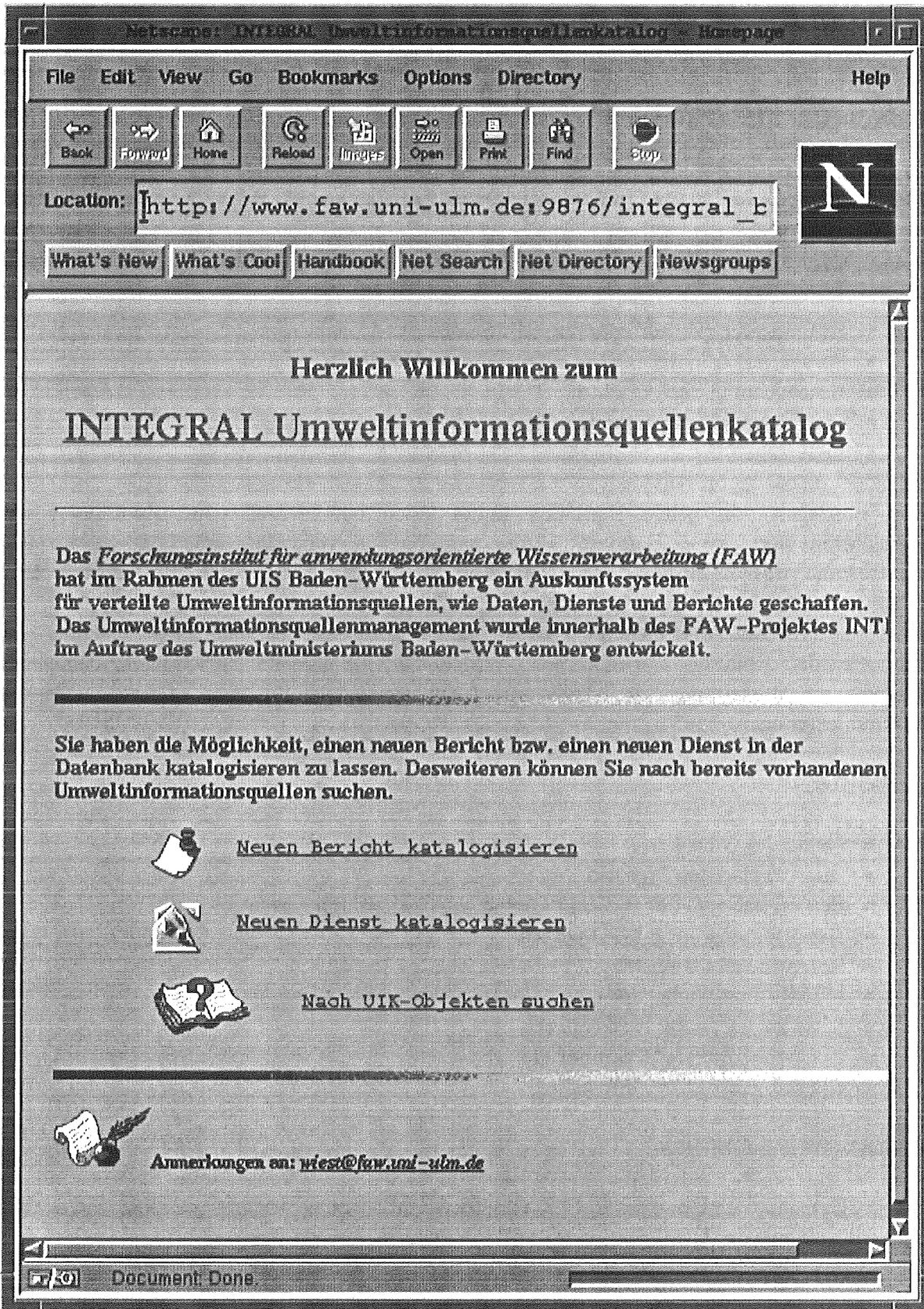


Abbildung 2-11: Einstiegsseite des Umweltinformationsquellenkataloges

Soll ein UIK-Objekt (Bericht, Dienst) eingetragen oder gesucht werden, muß eine Bekanntmachung des Benutzers im System erfolgen (siehe 2.2.3.2 Benutzer- und Rechteverwaltung). Er muß sich dazu mit seiner Benutzerkennung identifizieren. Erst nach erfolgreicher Überprüfung der eingegebenen Parameter gelingt der Eintritt in das System bzw. zu den Optionen für das Eintragen oder Suchen von UIK-Objekten.

Eintragen von UIK-Objekten:

Sollen neue UIK-Objekte eingetragen werden, so müssen auf den Folgeseiten der Anwendung folgende Attribute für das zu katalogisierende UIK-Objekt angegeben werden:

für Berichte:

- Anfangspfad (Zugriffspfad) des Berichtes
- Server, auf dem der Bericht physikalisch plaziert ist
- Beschreibung des Berichtes
- Liste zugeordneter Raumbezüge
- Liste zugeordneter Zeitbezüge

Beim Eintragen von neuen Berichten findet keine benutzergesteuerte Zuordnung von Deskriptoren statt. Neue Berichte werden automatisch verschlagwortet, indem auf den Datenbestand des Thesaurus zurückgegriffen wird. Die neuen Berichte werden nach Schlagworten (Deskriptoren) aus dem Thesaurus durchsucht und die Häufigkeit des Auftretens ermittelt. Anschließend werden innerhalb der Tabelle „Deskriptor_Referenz“ die entsprechenden Einträge vorgenommen, welche die vorhandenen Beziehungen des neuen Berichtes zu Deskriptoren des Thesaurus beschreiben und für eine spätere Suche zur Verfügung stellen.

für Dienste:

- Name des Dienstes
- Zugriffspfad des Dienstes
- Server, auf dem der Bericht / Dienst physikalisch plaziert ist
- Beschreibung des UIK-Objektes
- Kategorie des Dienstes
- HTML-Oberfläche (J/N)
- Liste zugeordneter Deskriptoren
- Liste zugeordneter Raumbezüge
- Liste zugeordneter Zeitbezüge

Suchen von UIK-Objekten:

Soll nach bereits eingetragenen UIK-Objekten im System gesucht werden, so müssen folgende Attribute als Suchkriterien für UIK-Objekte angegeben werden:

- Art des UIK-Objektes, das gefunden werden soll (Bericht, Dienst, Alle)
- datenhaltende Stelle(n)
- Liste zugeordneter Deskriptoren

- Liste zugeordneter Raumbezüge
- Liste zugeordneter Zeitbezüge

Für die Aktionen „neuen Bericht katalogisieren“ und „neuen Dienst katalogisieren“ werden dem Benutzer jeweils eigene Seiten (Eintragungsseiten) angezeigt. Diese Seiten stellen den Ausgangspunkt für die Belegung von Teilen der oben angegebenen Attribute dar. So wird die Zuordnung von Deskriptoren, Raumbezügen und Zeitbezügen zu dem neuen UIK-Objekt wiederum über eine Dialogfolge anderer Seiten abgewickelt. Im Gegensatz dazu werden elementare Attribute wie z.B. der Name des UIK-Objektes sofort auf der Eintragungseite belegt.

Einen Gesamtüberblick über die innerhalb des Umweltinformationsquellekataloges realisierten Dialogfolgen gibt Abbildung 2-12: Überblick über die Dialogfolgen des UIK.

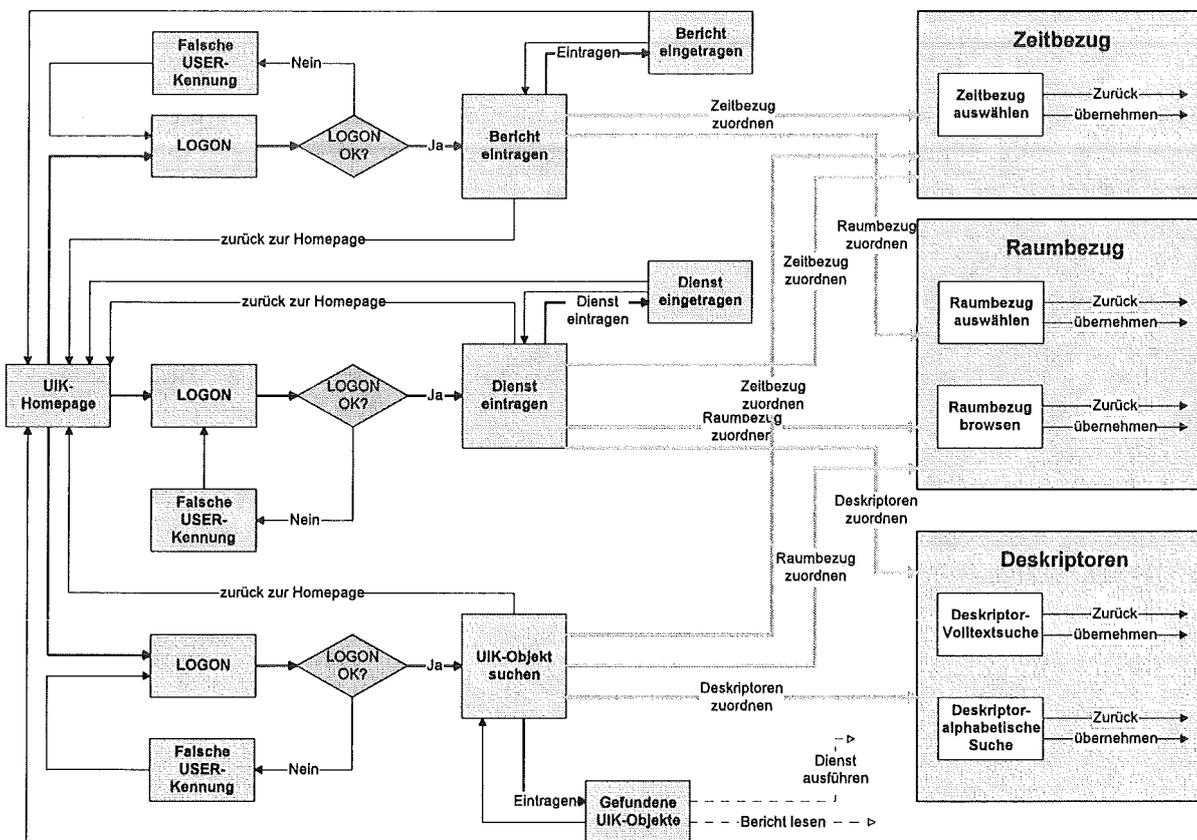


Abbildung 2-12: Überblick über die Dialogfolgen des UIK

Auch bei der Suche nach relevanten UIK-Objekten wird dem Benutzer eine eigene Dialogseite angeboten. Diese Seite (Aktion) kann von jedem Benutzer ausgeführt werden, der sich gegenüber dem System bekanntmacht. Dies ist notwendig, um im Rahmen des Zugriffs auf UIK-Objekte eine Rechteüberprüfung durchführen zu können. Über die Benutzer- und Rechteverwaltung kann somit festgelegt werden, welche Art von Zugriff auf welche UIK-Objekte gestattet wird (siehe dazu auch: 2.2.3.2 Benutzer- und Rechteverwaltung).

Im folgenden soll die Zuordnung von Deskriptoren, Raum- und Zeitbezügen zu UIK-Objekten näher beschrieben werden. Dabei werden die in Abbildung 2-12: Überblick über die Dialogfolgen des UIK aufgezeigten Blöcke Zeitbezug, Raumbezug und Deskriptoren

detaillierter dargestellt und beschrieben Die nachfolgend aufgeführten Dialogfolgen sind sowohl für das Eintragen von UIK-Objekten als auch für die Suche nach UIK-Objekten identisch und können somit unabhängig von der jeweils ausgeführten Aktion betrachtet werden.

2.3.2 Deskriptor

Bei der Bestimmung der Deskriptoren wurden im Rahmen der Entwicklung des Umweltinformationsquellenkataloges zwei unterschiedliche, aber sich ergänzende Möglichkeiten realisiert:

1. Deskriptoren Volltextsuche
2. Deskriptoren alphabetische Suche

Beide Möglichkeiten stehen sowohl beim Eintragen neuer Dienste (Anm.: die Deskriptor-Zuordnung steht nur beim Eintragen neuer Dienste zur Verfügung, da das Eintragen neuer Berichte mit einer automatischen Verschlagwortung realisiert wird), als auch bei der Suche nach bereits eingetragenen UIK-Objekten zur Verfügung.

Die HTML-Seiten „Deskriptor Volltextsuche“ und „Deskriptor alphabetische Suche“ können beim Eintragen eines neuen Dienstes und beim Suchen von UIK-Objekten aufgerufen werden. Nachfolgend soll der Dialogablauf bei der Deskriptor Volltextsuche und der alphabetischen Suche stichpunktartig dargestellt werden.

Von beiden Suchmöglichkeiten aus besteht die Möglichkeit, den Deskriptor-Browser aufzurufen und zu nutzen. Um diesem Punkt gerecht zu werden, sollen die Dialogmöglichkeiten des Deskriptor-Browsers getrennt beschrieben werden.

Deskriptor- Zuordnung

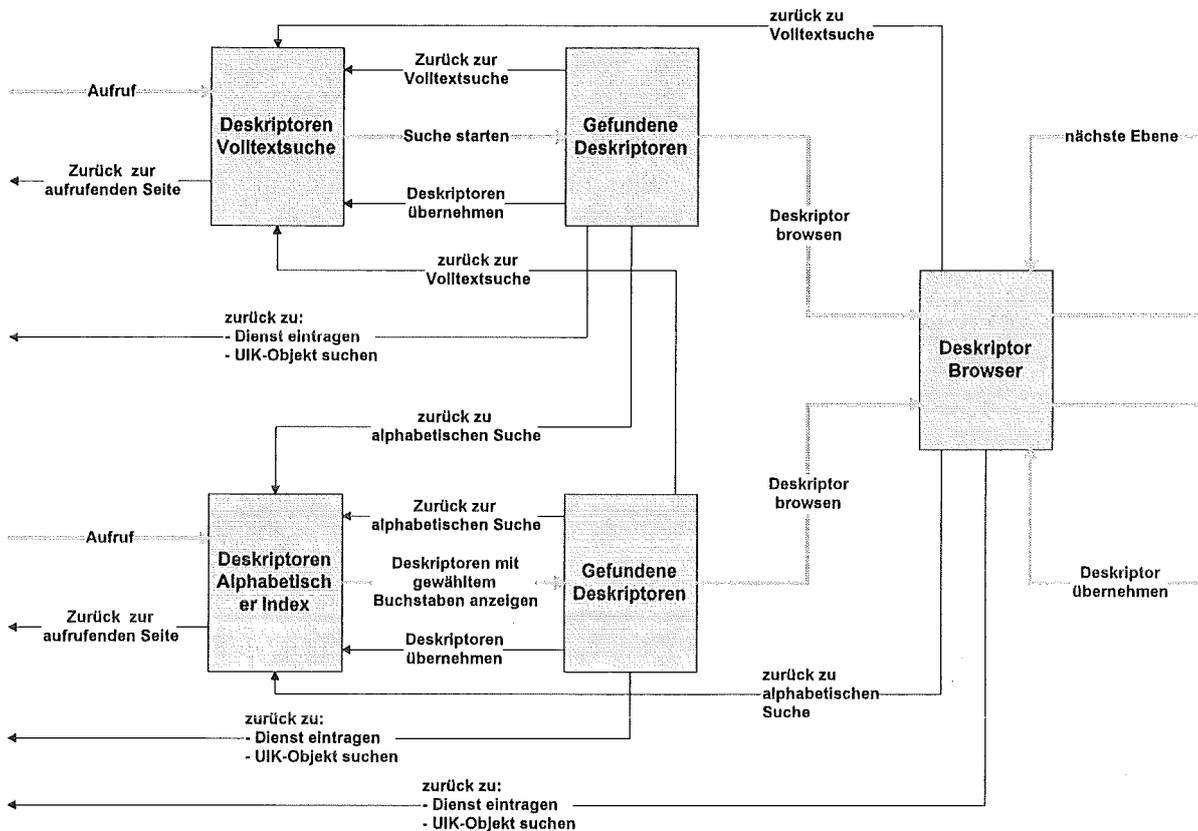


Abbildung 2-13: Dialogabläufe bei der Zuordnung von Deskriptoren

Deskriptor Volltextsuche:

1. Dem Anwender wird auf dieser Seite die Möglichkeit gegeben, einen bestimmten Suchbegriff einzugeben, nach welchem anschließend innerhalb des Thesaurus gesucht wird.
2. Durch Ausführen der Aktion „Suche starten“ wird auf der Basis des eingegebenen Suchbegriffs eine Datenbankabfrage mit dem Ziel gestartet, alle Deskriptoren des Thesaurus zu ermitteln, die dem eingegebenen Suchbegriff entsprechen bzw. bei denen der Suchbegriff vorkommt.
3. Die gefundenen Deskriptoren werden anschließend auf der HTML-Seite „Gefundene Deskriptoren“ angezeigt. Der Anwender hat nun mehrere Aktionsmöglichkeiten:
 - er kann ohne weitere Bearbeitung zur Einstiegsseite „Dienst eintragen“ oder zur Seite „UIK-Objekt suchen“ zurückkehren. Dies ist abhängig davon, von welcher Seite aus die Volltextsuche aufgerufen wurde.
 - er kann ohne weitere Bearbeitung der Deskriptoren zur Volltextsuche zurückkehren
 - er kann zur alphabetischen Suche wechseln
 - er kann aus der Menge der gefundenen Deskriptoren die gewünschten markieren und übernehmen, d.h. einem neuen Dienst zuordnen oder für die Suche nach einem Dienst oder einem Bericht verwenden
 - er kann sich für einen der gefundenen Deskriptoren entscheiden, diesen markieren und den Deskriptor-Browser aufrufen

Deskriptor alphabetische Suche:

1. Dem Anwender wird bei der alphabetischen Suche die Möglichkeit gegeben, einen bestimmten Buchstaben auszuwählen, der den Anfangsbuchstaben aller Deskriptoren darstellt, nach denen innerhalb des Thesaurus gesucht werden soll.
2. Durch Ausführen der Aktion „Deskriptoren mit gewähltem Buchstaben anzeigen“ wird auf der Basis des gewählten Buchstabens eine Datenbankabfrage mit dem Ziel gestartet, alle Deskriptoren des Thesaurus zu ermitteln, die mit dem gewählten Buchstaben beginnen.
3. Die gefundenen Deskriptoren werden anschließend auf der HTML-Seite „Gefundene Deskriptoren“ alphabetisch sortiert angezeigt. Der Anwender hat nun mehrere Aktionsmöglichkeiten:
 - er kann ohne weitere Bearbeitung zur Einstiegsseite „Dienst eintragen“ oder zur Seite „UIK-Objekt suchen“ zurückkehren. Dies ist abhängig davon, von welcher Seite aus die alphabetische Suche aufgerufen wurde.
 - er kann ohne weitere Bearbeitung der Deskriptoren zur alphabetischen Suche zurückkehren
 - er kann zur Volltextsuche wechseln
 - er kann aus der Menge der gefundenen Deskriptoren die gewünschten markieren und übernehmen, d.h. einem neuen Dienst zuordnen oder für die Suche nach einem Dienst oder einem Bericht verwenden
 - er kann sich für einen der gefundenen Deskriptoren entscheiden, diesen markieren und den Deskriptor-Browser aufrufen

Deskriptor-Browser:

Wird der Deskriptor-Browser mit einem gefundenen Begriff aufgerufen, so stellt der Browser den ausgewählten Begriff nochmals dar. Zudem listet er alle innerhalb des Thesaurus gefundenen Synonyme zu dem Begriff, die Oberbegriffe des gewählten Deskriptors sowie alle Unterbegriffe auf. Der Deskriptor-Browser bietet dem Anwender nun folgende Aktionsmöglichkeiten:

- Der dargestellte, gewählte Deskriptor kann übernommen werden. Er wird daraufhin in eine temporäre Tabelle eingetragen und für das Eintragen eines neuen UIK-Objektes bzw. für die Suche nach UIK-Objekten verwendet.
- Wurde ein Deskriptor übernommen, wird die Browser-Seite erneut angezeigt und der Benutzer hat die Möglichkeit, weitere Deskriptoren auszuwählen.
- Da zu jedem gewählten Deskriptor alle Ober- und Unterbegriffe angezeigt werden, hat der Benutzer die Möglichkeit, entweder aus der Liste der Oberbegriffe oder aus der Liste der Unterbegriffe einen Begriff auszuwählen und mit der Aktion „nächste Ebene“ entweder eine Ebene tiefer (Unterbegriff gewählt) oder eine Ebene höher (Oberbegriff gewählt) zu browsen.
- Wurde die Aktion „nächste Ebene“ gewählt, so wird daraufhin eine Datenbankabfrage auf dem Datenbestand des Thesaurus durchgeführt und das gefundene Ergebnis, welches wiederum aus dem gewählten Deskriptor, den Synonymen und allen gefundenen Ober- und Unterbegriffen besteht, im Browser angezeigt. Für den Anwender besteht nun wieder die Möglichkeit, die Aktion „Deskriptor übernehmen“ zu wählen oder noch weiter im Netzwerk der vorhandenen Begriffe zu browsen.

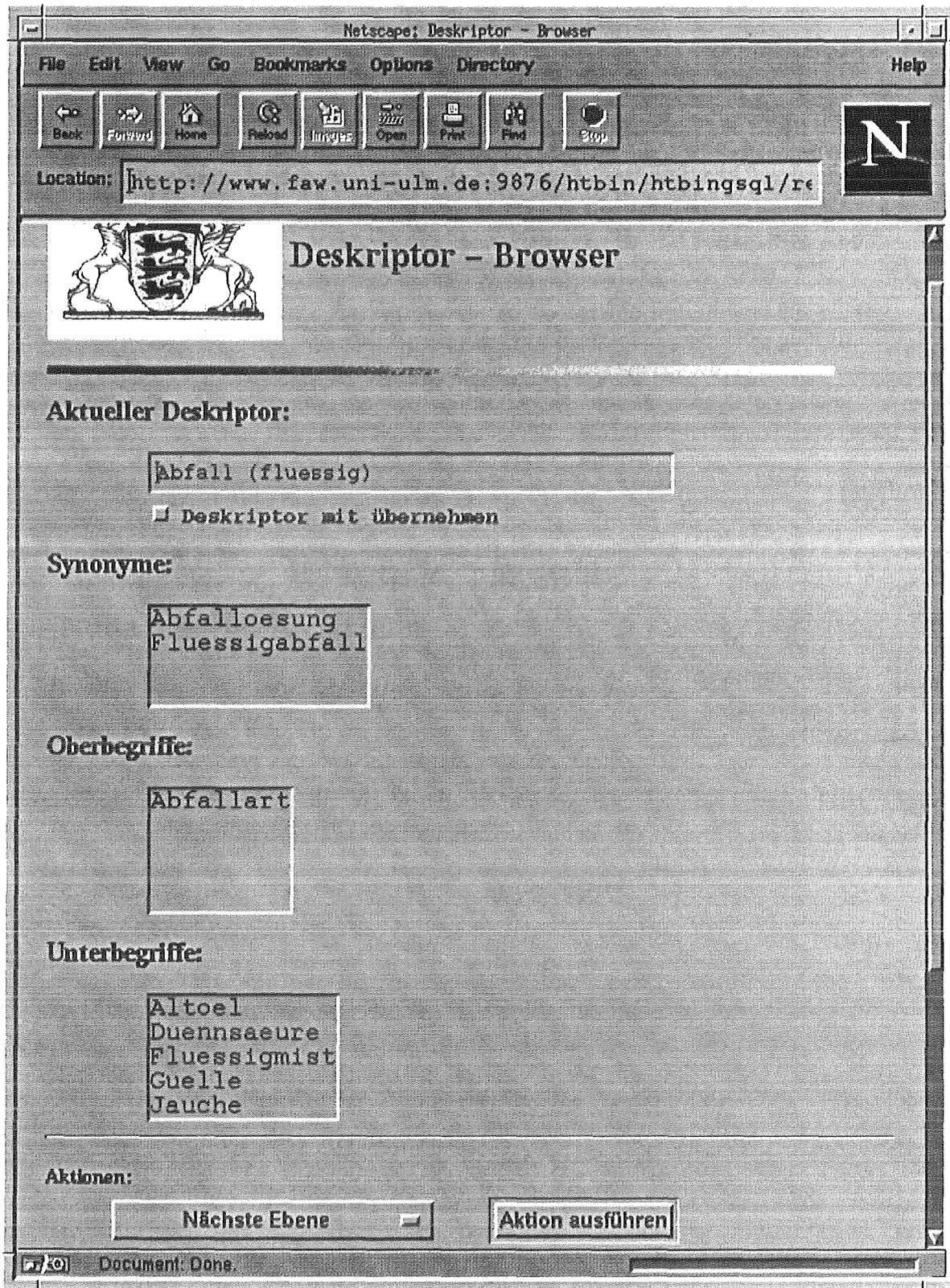


Abbildung 2-14: Deskriptor-Browser

2.3.3 Raumbezug

Auch bei der Definition von Raumbezügen, die entweder für die Zuordnung zu UIK-Objekten beim Eintragen neuer Objekte oder als Suchkriterium für das Auffinden bereits eingetragener UIK-Objekte verwendet werden, wurden im Rahmen der Entwicklung des Gazetteers des Umweltinformationsquellenkataloges zwei unterschiedliche, aber sich ergänzende Möglichkeiten realisiert:

1. Raumbezug (Gebiet) browsen
2. Raumbezug (Gebiet) gezielt auswählen

Beide Möglichkeiten stehen sowohl beim Eintragen neuer UIK-Objekte, als auch bei der Suche nach bereits eingetragenen UIK-Objekten zur Verfügung.

Beim Eintragen eines neuen UIK-Objektes muß diesem ein Raumbezug zugeordnet werden. Dazu können wahlweise die HTML-Seiten „Raumbezug-Browser“ und „Gebiet auswählen“ aufgerufen werden. Für die Suche nach UIK-Objekten stehen ebenso beide Möglichkeiten zur Verfügung. Der Aufruf der Seiten „Raumbezug-Browser“ und „Gebiet auswählen“ kann somit von den Seiten:

- Bericht eintragen
- Dienst eintragen
- UIK-Objekt suchen

erfolgen. Nachfolgend soll der Dialogablauf bei der Verwendung des Raumbezug-Browsers und bei der gezielten Auswahl von Gebieten stichpunktartig dargestellt werden.

Raumbezug - Zuordnung

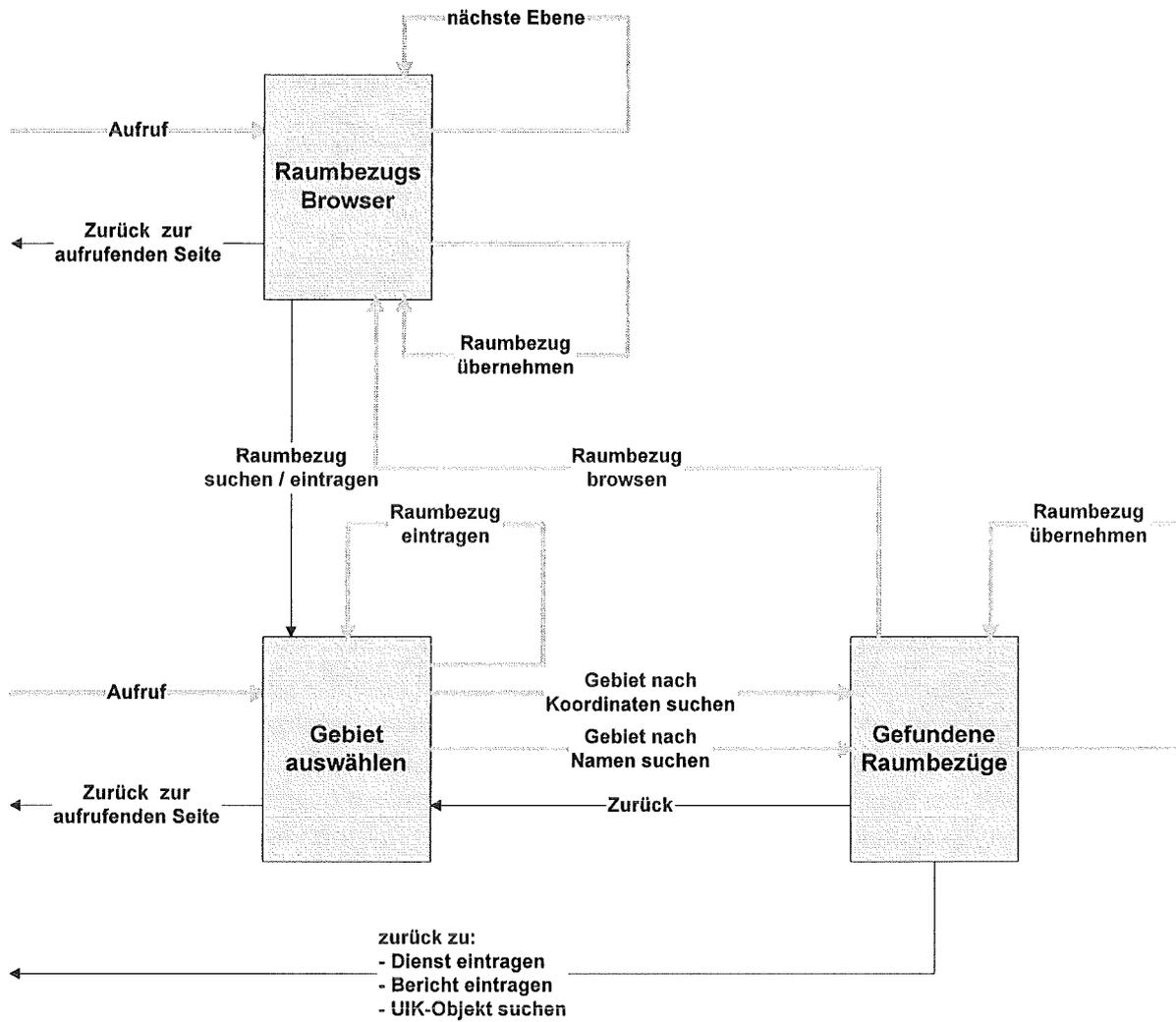


Abbildung 2-15: Dialogfolgen bei der Zuordnung von Raumbezügen

Raumbezug-Browser:

Wird der Raumbezug-Browser des Gazetteers aufgerufen, so stellt der Browser zuerst das innerhalb der Gebietshierarchie definierte „Top-Level“-Gebiet dar. Dieses Gebiet (Raumbezug) ist dadurch gekennzeichnet, daß für dieses Gebiet keine übergeordneten Gebiete definiert sind. Zudem listet er alle Raumbezüge auf, die Teil dieses identifizierten Top-Level-Gebietes sind (alle Untergebiete). Der Raumbezug-Browser bietet dem Anwender nun folgende Aktionsmöglichkeiten:

- Der Anwender kann ohne weitere Aktion den Raumbezug-Browser verlassen und zu der Seite zurückkehren, von der aus der Raumbezug-Browser aufgerufen wurde.
- Der dargestellte, gewählte Raumbezug kann übernommen werden. Er wird daraufhin in eine temporäre Tabelle eingetragen und für das Eintragen eines neuen UIK-Objektes bzw. für die Suche nach UIK-Objekten verwendet.
- Wurde ein Raumbezug übernommen, wird die Browser-Seite erneut angezeigt und der Benutzer hat die Möglichkeit, weitere Gebiete auszuwählen.
- Da zu jedem gewählten Raumbezug alle Ober- und Untergebiete angezeigt werden, hat der Benutzer die Möglichkeit, entweder aus der Liste der Obergebiete oder aus der Liste der Untergebiete einen Raumbezug auszuwählen und mit der Aktion „nächste Ebene“ entweder eine Ebene tiefer (Untergebiet gewählt) oder eine Ebene höher (Obergebiet gewählt) zu browsen.
- Wurde die Aktion „nächste Ebene“ gewählt, so wird daraufhin eine Datenbankabfrage auf dem Datenbestand des Gazetteers durchgeführt und das gefundene Ergebnis, welches wiederum aus dem gewählten Raumbezug und den gefundenen Ober- und Untergebieten besteht, im Browser angezeigt. Für den Anwender besteht nun wieder die Möglichkeit, die Aktion „Raumbezug übernehmen“ zu wählen oder noch weiter im Netzwerk der vorhandenen Gebiete zu browsen.
- Der Anwender kann zur Seite „Gebiet auswählen“ wechseln.

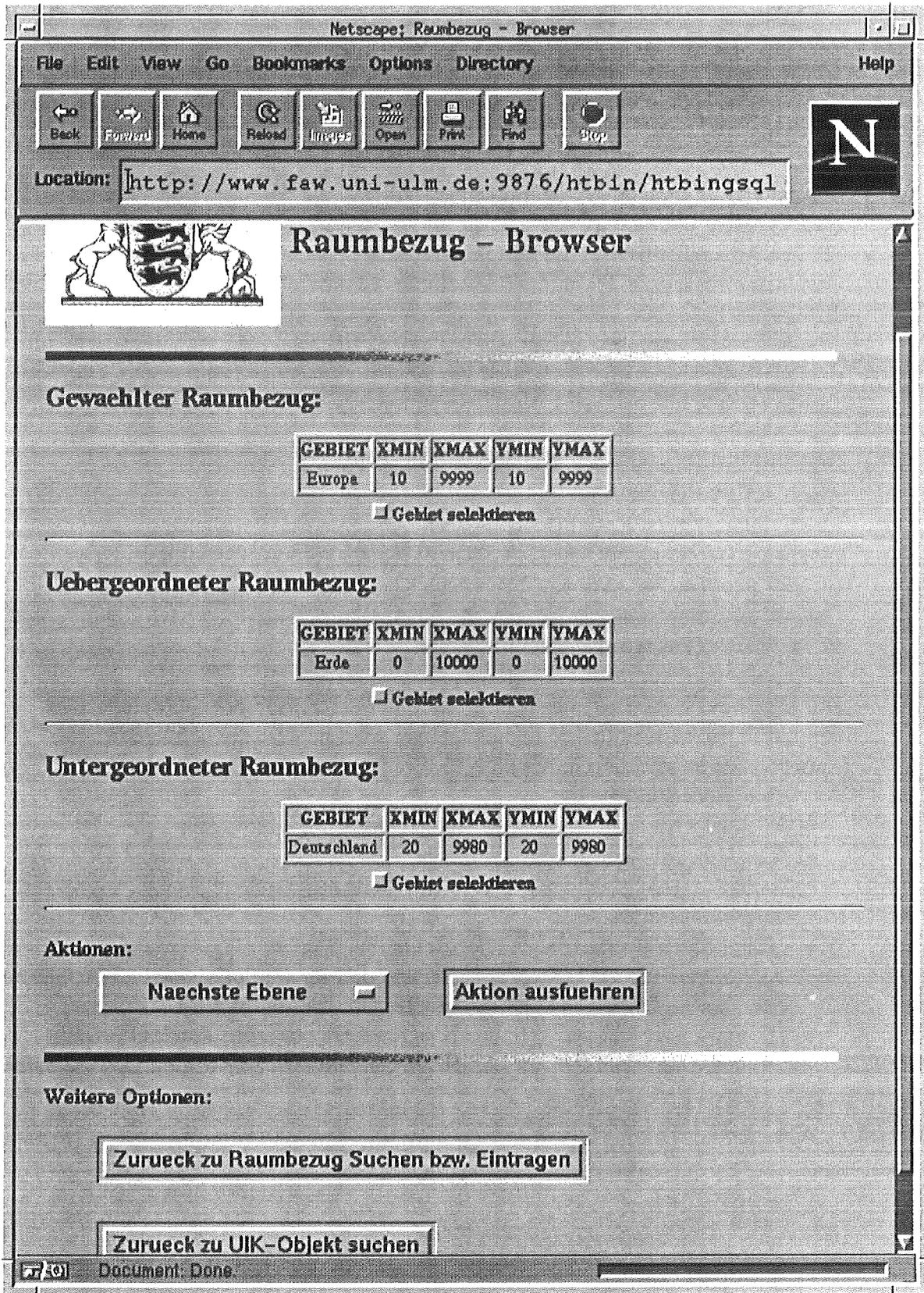


Abbildung 2-16: Der Raumbezug-Browser

Gebiet auswählen:

Wird die HTML-Seite „Gebiet auswählen“ aufgerufen, bestehen für den Anwender im Prinzip die gleichen Möglichkeiten wie bei der Deskriptor-Volltextsuche. Ihm werden folgende Optionen geboten:

- Der Anwender kann ohne weitergehende Aktion direkt zur aufrufenden Seite zurückkehren
- Für den Anwender besteht die Möglichkeit, einen neuen Raumbezug in den Gazetteer einzutragen. Dazu ist es notwendig, daß sowohl der Gebietsname, als auch die entsprechenden Koordinaten eingegeben werden. Anschließend muß die Aktion „Raumbezug eintragen“ gewählt werden, mit welcher der neue Raumbezug an die richtige Stelle innerhalb der Gebietshierarchie des Gazetteers eingetragen wird (siehe dazu auch: 2.2.3.4 Gazetteer). Nach Ausführen dieser Aktion wird wiederum die Seite „Gebiet auswählen“ angezeigt.
- Innerhalb der HTML-Seite „Gebiet auswählen“ kann der Anwender einen Raumbezug entweder anhand eines eingegebenen Gebietsnamens suchen oder die für ihn interessanten Koordinaten angeben. Anschließend kann entsprechend der Eingaben (Gebietsname oder Koordinaten) mit der Aktion „Gebiet nach Namen suchen“ oder mit der Aktion „Gebiet nach Koordinaten suchen“ eine Datenbankabfrage gestartet werden, die innerhalb des Gazetteers nach Raumbezügen sucht, die dem angegebenen Gebietsnamen oder den Koordinaten entsprechen oder als Teil darin vorkommen.
- Das Ergebnis dieses Suchvorgangs wird auf der Seite „Gefundene Raumbezüge“ dargestellt. Es besteht die Möglichkeit, einen oder mehrere der gefundenen Raumbezüge zu selektieren. Nachfolgend werden dem Anwender folgende Auswahlaktionen angeboten:
 - Rückkehr zur Seite „Gebiet auswählen“ ohne weitere Aktion
 - Rückkehr zur Seite „Dienst eintragen“, „Bericht eintragen“ oder „UIK-Objekt suchen“, abhängig davon, von welcher Seite aus die Seite „Gebiet auswählen“ aufgerufen wurde.
 - „Raumbezug übernehmen“. Die selektierten Raumbezüge werden in einer temporären Tabelle zwischengespeichert und für die Suche nach UIK-Objekten bzw. für das Eintragen eines neuen UIK-Objektes verwendet.
 - „Raumbezug browsen“. Mit den selektierten Raumbezügen wird die Seite „Raumbezug-Browser“ aufgerufen (Beschreibung siehe: Raumbezug-Browser)

2.3.4 Zeitbezug

Im Umweltinformationsquellenkatalog (UIK) ist neben der Zuordnung von Deskriptoren und Raumbezügen auch die Zuordnung von Zeitbezügen zu UIK-Objekten vorgesehen. Neben dem Thema und dem Gebiet, auf das sich ein UIK-Objekt (Bericht, Dienst) bezieht, ist natürlich auch die Zeitkomponente von Bedeutung (z.B. Erhebungszeitraum von Umweltdaten). Bei der Betrachtung und Realisierung dieser Komponente erwies es sich als notwendig, zwischen Zeitpunkten und Zeiträumen zu unterscheiden. Genau diese Anforderungen beeinflussten sowohl die Gestaltung des entsprechenden Teils des Datenbankschemas des UIK (Tabelle „Zeitbezug“), als auch die Dialogführung bzw. die zur Verfügung gestellten Eingabemöglichkeiten.

Beim Eintragen von UIK-Objekten müssen den neuen UIK-Objekten Zeitbezüge zugeordnet werden. Diese Zeitbezüge können entweder Zeitpunkte oder Zeiträume sein. Beim Suchen von UIK-Objekten werden Zeitbezüge als Identifikationskriterien für UIK-Objekte herangezogen. Bei der Suche nach UIK-Objekten werden die zugeordneten Zeitbezüge bereits gespeicherter UIK-Objekte mit den ausgewählten Zeitbezügen verglichen. Dem Anwender werden sowohl beim Eintragen neuer UIK-Objekte als auch beim Suchen nach UIK-Objekten zwei Möglichkeiten angeboten:

1. Zeitpunkt festlegen
2. Zeitraum festlegen

Um einen Zeitbezug für die Suche oder das Eintragen festzulegen, kann die Seite „Zeitbezug eintragen“ bzw. „Zeitbezug Suchen“ aufgerufen werden. Diese Seiten sind bis auf den unterschiedlichen Kontext (eintragen / suchen) identisch aufgebaut. Der Aufruf der Seite „Zeitbezug eintragen“ kann von den Seiten:

- Bericht eintragen
- Dienst eintragen

erfolgen. Die Seite „Zeitbezug suchen“ kann von der Seite:

- UIK-Objekt suchen

aufgerufen werden.

Nachfolgend soll der Dialogablauf bei der Festlegung von Zeitbezügen stichpunktartig am Beispiel der Seite „Zeitbezug eintragen“ dargestellt werden.

Zeitbezug - Zuordnung

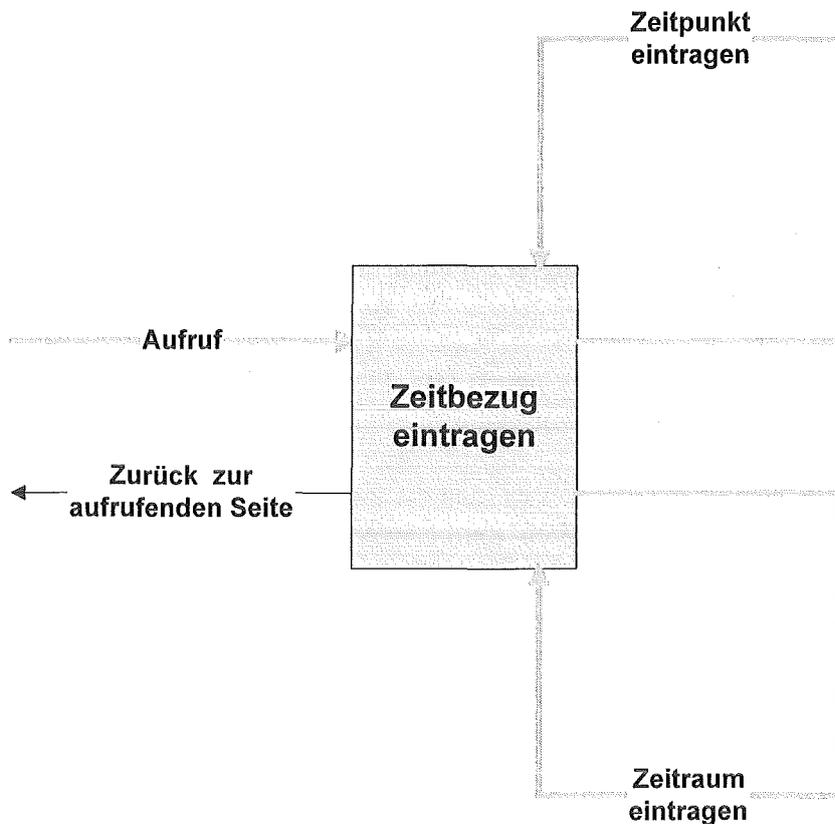


Abbildung 2-17: Dialogfolgen bei der Zuordnung von Zeitbezügen

Nach Aufruf der Seite „Zeitbezug eintragen / auswählen“ stehen dem Anwender folgende Aktionsmöglichkeiten zur Verfügung:

- Der Anwender kann ohne weitere Aktionen zur aufrufenden Seite (neuen Bericht eintragen, neuen Dienst eintragen, nach UIK-Objekt suchen) zurückkehren.
- Es besteht die Möglichkeit, einen Zeitpunkt einzugeben. Für die Angabe des Zeitpunktes muß der Tag, der Monat und das Jahr festgelegt werden. Anschließend wird mittels der Aktion „Zeitpunkt übernehmen“ der gewählte Zeitpunkt in eine temporäre Tabelle übernommen und für die Zuordnung zu dem einzutragenden UIK-Objekt verwendet. Nach Ausführen der Aktion „Zeitpunkt übernehmen“ wird die Seite „Zeitbezug eintragen“ erneut angezeigt. Damit besteht für den Anwender die Möglichkeit, weitere Zeitbezüge einzutragen.
- Neben der Möglichkeit, einen Zeitpunkt einzutragen, besteht die Möglichkeit, einen Zeitraum einzugeben. Für die Angabe des Zeitraumes müssen der Anfangszeitpunkt und der Endzeitpunkt des Zeitraumes im Format Tag, Monat und Jahr festgelegt werden. Anschließend wird mittels der Aktion „Zeitraum übernehmen“ der gewählte Zeitraum in eine temporäre Tabelle übernommen und für die Zuordnung zu dem einzutragenden UIK-Objekt verwendet. Nach Ausführen der Aktion „Zeitraum übernehmen“ wird die Seite „Zeitbezug eintragen“ erneut angezeigt. Damit besteht für den Anwender die Möglichkeit, weitere Zeitbezüge einzutragen.

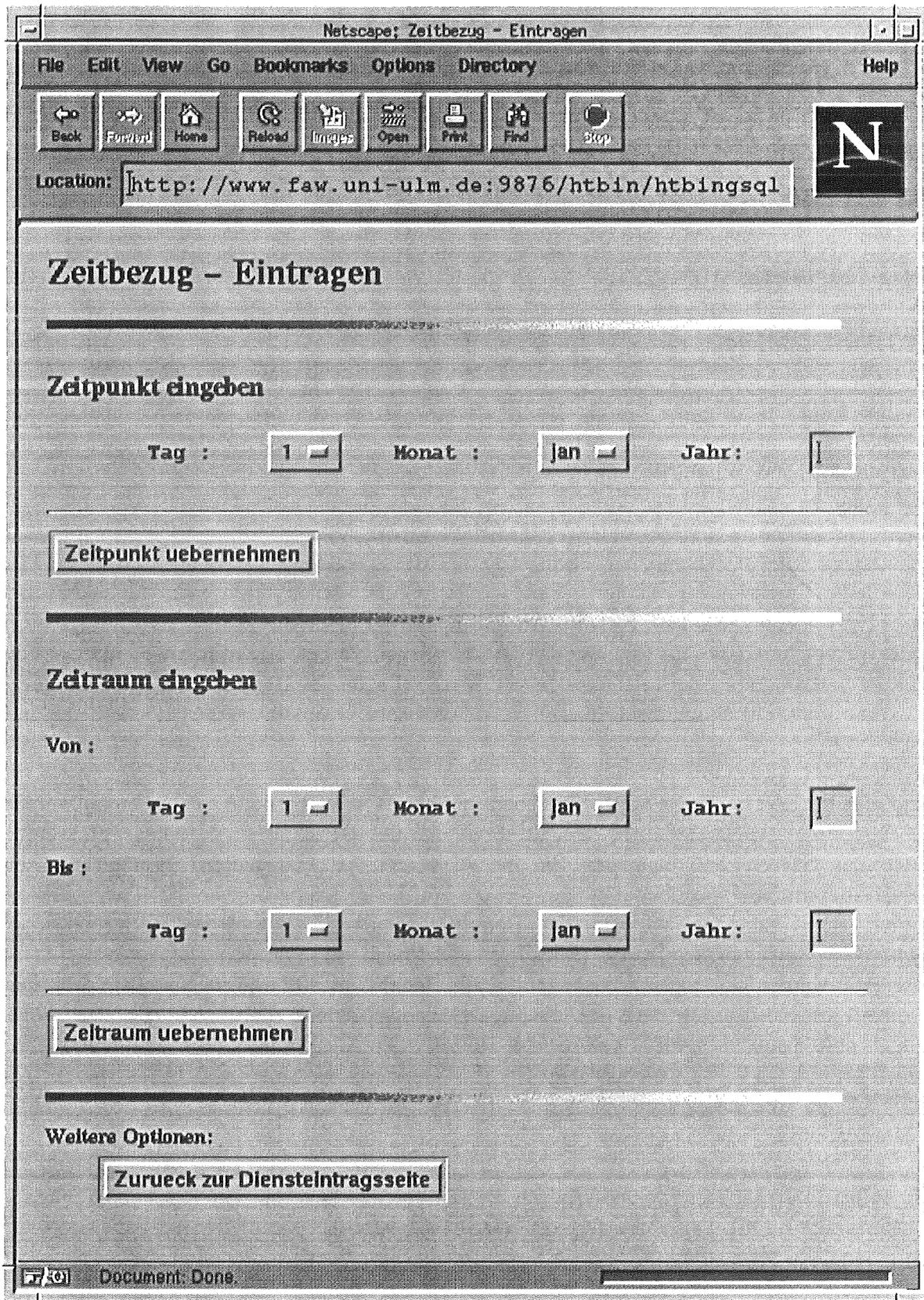


Abbildung 2-18: Dialogseite „Zeitbezug eintragen“

Um die Eingabemöglichkeiten für Zeitbezüge so komfortabel wie möglich zu gestalten und Fehlerquellen bei der Eingabe soweit wie möglich auszuschließen, werden dem Anwender bei

der Festlegung des Tages und des Monats vordefinierte Listen angeboten. Der Anwender hat somit die Möglichkeit, den Tag und den Monat aus diesen Listen auszuwählen. Einzig die Eingabe der Jahreszahl ist erforderlich.

2.4 Funktionalitäten des UIK

2.4.1 Bericht eintragen

2.4.1.1 Überblick

Mit der Funktion „Neuen Bericht katalogisieren“ ist es möglich, Metainformation über Berichte, die in Form von HTML-Dokumenten vorliegen, in die Datenbank des Umweltinformationsquellenkataloges (UIK) einzutragen und damit anderen Nutzern zur Verfügung zu stellen. Dabei können dem gesamten Bericht verschiedene Attribute (z.B. Raum- und Zeitbezüge, Deskriptoren) zugeordnet werden. Voraussetzung dafür, einen Bericht einzutragen, ist, daß dieser in Form von HTML-Dokumenten vorliegt; diese HTML-Dokumente müssen in einer Verzeichnisstruktur unterhalb eines anzugebenden Wurzelverzeichnisses liegen. Zusätzlich muß der Bericht (für die Dauer des Eintragens) in einem Verzeichnis auf dem Rechner des UIK vorliegen. In der derzeitigen Version des UIK muß die Übertragung der Berichtsseiten vor dem Eintragen des Berichtes erfolgen, zukünftig soll diese Funktion automatisch über WWW-Formulare veranlaßt werden können (z.B. mit ftp).

Liegt der Bericht in dieser Form vor, kann er über WWW-Formulare eingetragen werden. Dabei muß der Anfangspfad, in dem die Startseite des Berichtes liegt, angegeben werden, sowie der Server, auf dem der Bericht zu finden ist. Zu jedem Bericht kann ein Beschreibungstext eingegeben und über weitere Formulare Raum- und Zeitbezüge zugeordnet werden. Beim Eintragen werden allen Berichtsabschnitten, die sich im Verzeichnisbaum unterhalb der angegebenen Startseite des Berichtes finden, diese Attribute zugeordnet.

Zusätzlich werden automatisch Deskriptoren aus einem Umweltthesaurus vergeben. Für diese automatische Indexierung wird der jeweilige Berichtsabschnitt zunächst in einzelne Wörter zerlegt und dann überprüft, welche der Begriffe als Deskriptoren im Umweltthesaurus aufgeführt sind bzw. zu welchen Begriffen im Thesaurus Synonymbegriffe existieren. Aus dieser Menge von Deskriptoren wird eine Auswahl getroffen, die dann dem Berichtsabschnitt zugeordnet wird.

Als weiteres Attribut wird beim Eintragen der Berichte die hierarchische Beziehung zwischen den einzelnen HTML-Dokumenten eines Berichtes anhand der Verzeichnisstruktur festgelegt. Für jeden Berichtsabschnitt wird dabei überprüft, welche übergeordneten HTML-Dokumente existieren; diese Information wird ebenfalls im UIK abgelegt.

2.4.1.2 „Bericht eintragen“ aus Benutzersicht

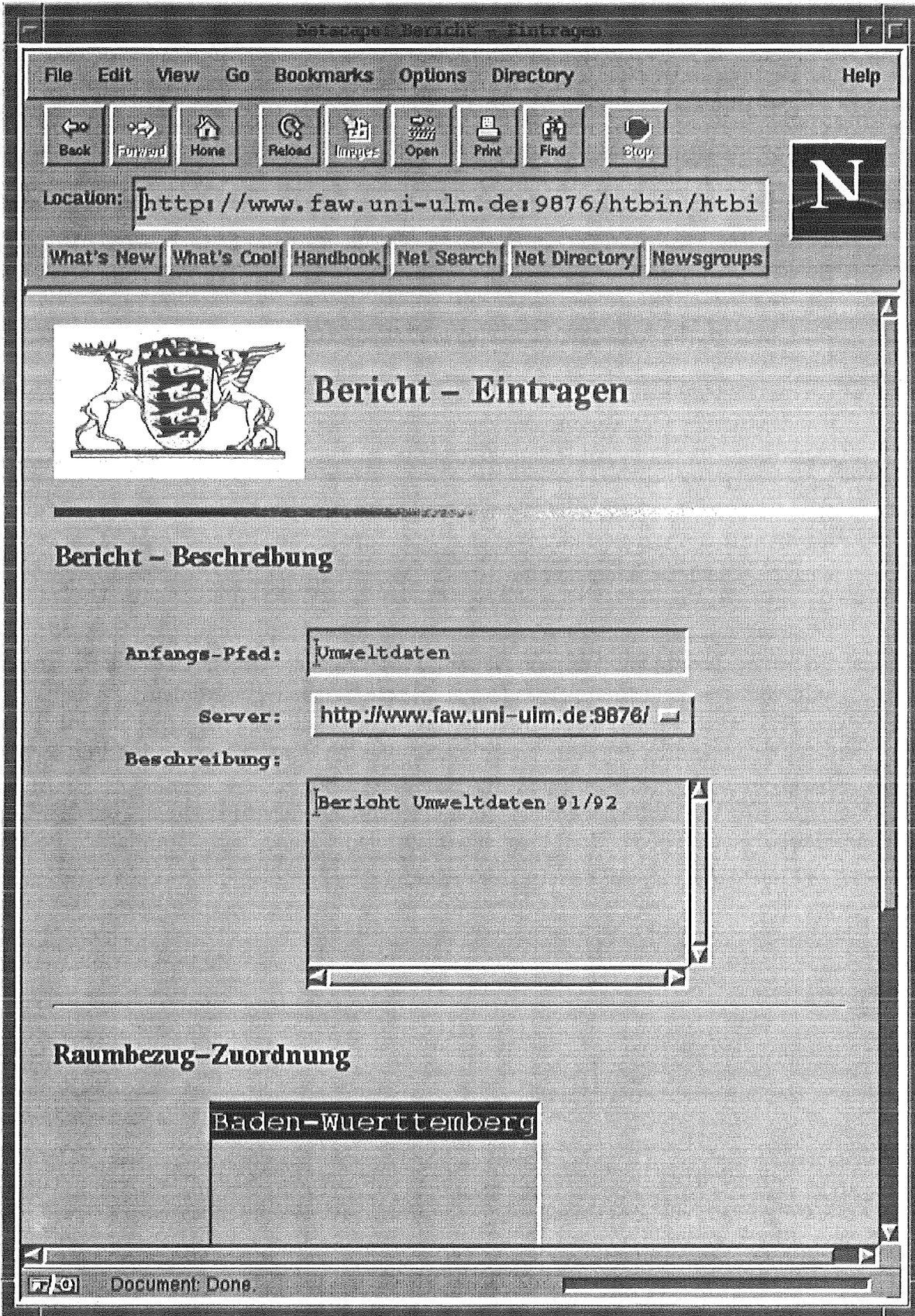


Abbildung 2-19: WWW-Formular für „Neuen Bericht katalogisieren“

Die Funktion „Bericht eintragen“ kann über die Einstiegsseite des UIK aufgerufen werden, indem dort der Verweis auf „Neuen Bericht katalogisieren“ angewählt wird. Um das Eintragsformular für „Bericht eintragen“ (s. Abbildung 2-19) zu erhalten, ist es notwendig, sich mit Benutzernamen und Paßwort einzuloggen.

Bevor in diesem Eintragsformular mit der Aktion „Eintragen mit Verschlagwortung“ das eigentliche Eintragen des Berichtes in den UIK gestartet werden kann, müssen in die Eingabefelder des Formulars Werte eingetragen werden. Im einzelnen sind dies folgende Felder:

- Anfangspfad: Anfangspfad ist das Verzeichnis, zu dem relativ die Verweise in den HTML-Dokumenten angegeben sind. Ist die Startseite eines Berichtes beispielsweise über die URL-Adresse *http://www.faw.uni-ulm.de/Umweltdaten/start.html* zu erreichen, ist als Pfadname „Umweltdaten“ anzugeben (ohne „/“ am Anfang und am Ende !).
- Server: Hier kann aus einer Liste von Servern der Server ausgewählt werden, über den der Bericht gefunden werden kann. Zur Auswahl angeboten werden dabei nur diejenigen Server, die der Instanz zugeordnet sind, der auch die eintragende Person zugeordnet ist.
- Beschreibung: In das Beschreibungsfeld kann ein beliebiger Text eingegeben werden. Dieser Beschreibungstext wird allen HTML-Seiten zugeordnet, die zu dem einzutragenden Bericht gehören.
- Raumbezug-Zuordnung: Dieses Feld kann mit Hilfe der Aktionen „Raumbezüge browsen“ oder „Raumbezüge auswählen“ (s. 2.3.3 Raumbezug) gefüllt werden.
- Zeitbezug-Zuordnung: Über die Aktion „Zeitbezüge auswählen“ (s. 2.3.4 Zeitbezug) kann in dieses Feld ein Zeitbezug (oder mehrere) eingetragen werden.

Wenn in den Eingabefeldern des Formulars die gewünschten Werte eingetragen sind, kann die Aktion „Eintragen mit Verschlagwortung“ (s. Abbildung 2-20) gestartet werden.

Nach dem erfolgreichen Eintragen des Berichtes erscheint eine Ergebnisseite mit der Meldung „Bericht wurde eingetragen“, von hier aus kann man wieder zum Formular „Bericht eintragen“ zurückkehren.

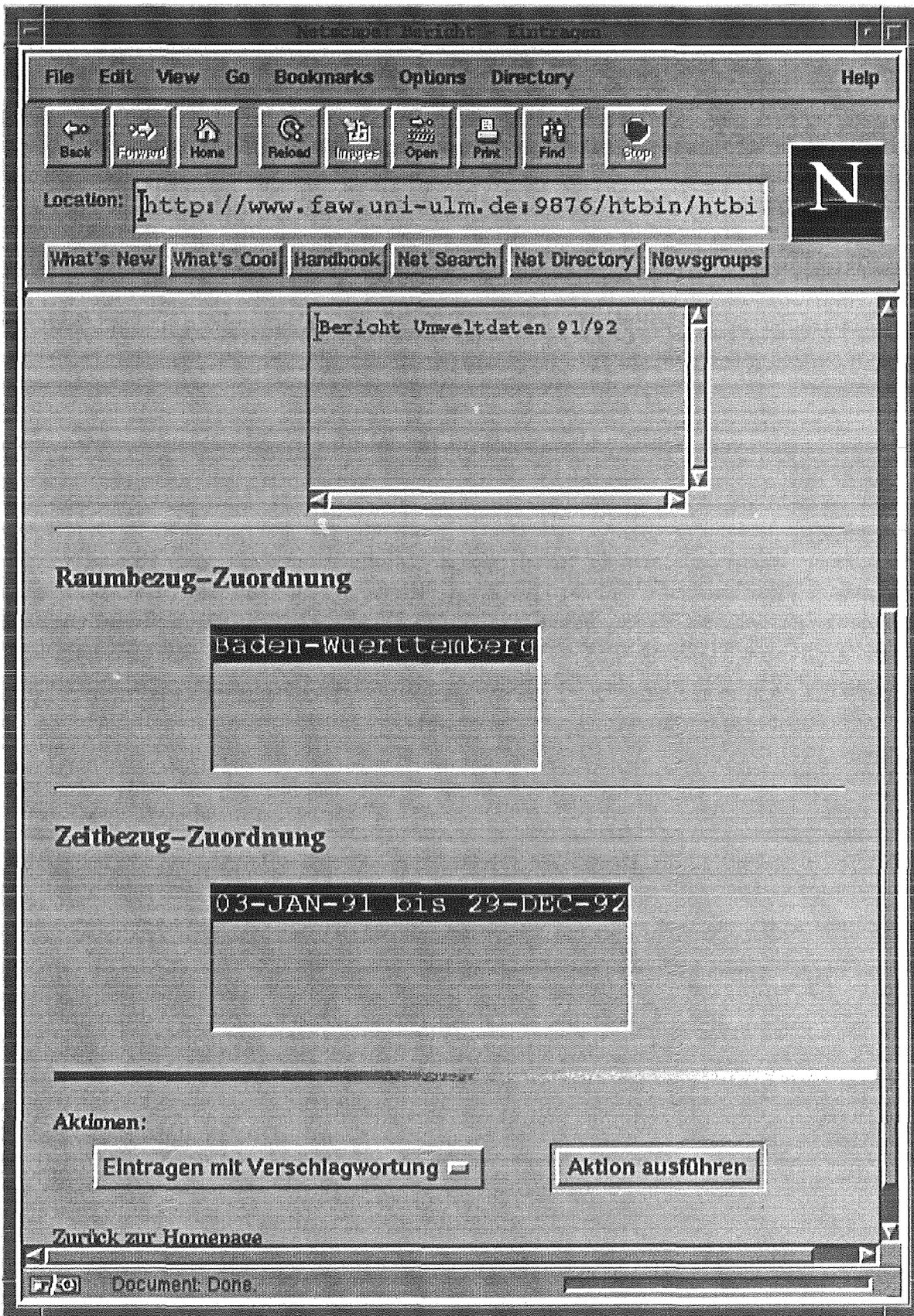


Abbildung 2-20: WWW-Formular für „Neuen Bericht katalogisieren“

2.4.1.3 Programmablauf von „Bericht eintragen“

Wenn in der Homepage des UIK die Funktion „Neuen Bericht katalogisieren“ ausgewählt und Benutzerkennung und Paßwort eingegeben wurde, werden diese Benutzerangaben überprüft. Falls für die Benutzerkennung ein Recht zum Eintragen von Berichten besteht, wird ein C-Programm aufgerufen, welches aus der Datei „Bericht_eintragen.sql“ das HTML-Formular für das Eintragen eines Berichtes generiert.

Bei „Aktion ausführen“ („Eintragen mit Verschlagwortung“) in diesem Formular wird ein Shell-Skript („FilterOption.sh“) aufgerufen, dem als Parameter die Werte aus dem Eingabeformular in der Variablen QUERY_STRING übergeben werden. „FilterOption.sh“ ruft mit diesen Werten das C-Programm „FilterOption“ auf, das in Abhängigkeit von der ausgewählten Aktion verschiedene Rückgabewerte liefert. Anhand dieses Rückgabewertes erkennt „FilterOption.sh“ die ausgewählte Aktion und ruft entsprechende Programme auf, hier das Shell-Skript „berichte.sh“, an das die Variable QUERY_STRING weitergegeben wird. Dieses Skript erledigt die eigentliche Aufgabe, d.h. es sucht die einzelnen HTML-Seiten des einzutragenden Berichtes, ruft für diese das Programm zur Verschlagwortung auf und sucht übergeordnete Berichtsabschnitte. Alle diese Informationen, die zusätzlich zu den bereits übergebenen Werten aus dem Formular (wie z.B. Raum- und Zeitbezug) erzeugt werden, werden wie diese in der Variablen QUERY_STRING abgelegt. Für das Eintragen in die Datenbank werden mit diesen aktuellen Werten aus QUERY_STRING SQL-Statements generiert und anschließend abgesetzt. Danach kehrt der Ablauf mit dem Anzeigen der Ergebnisseite wieder zum Browser zurück. Abbildung 2-21 zeigt den Ablauf von „berichte.sh“.

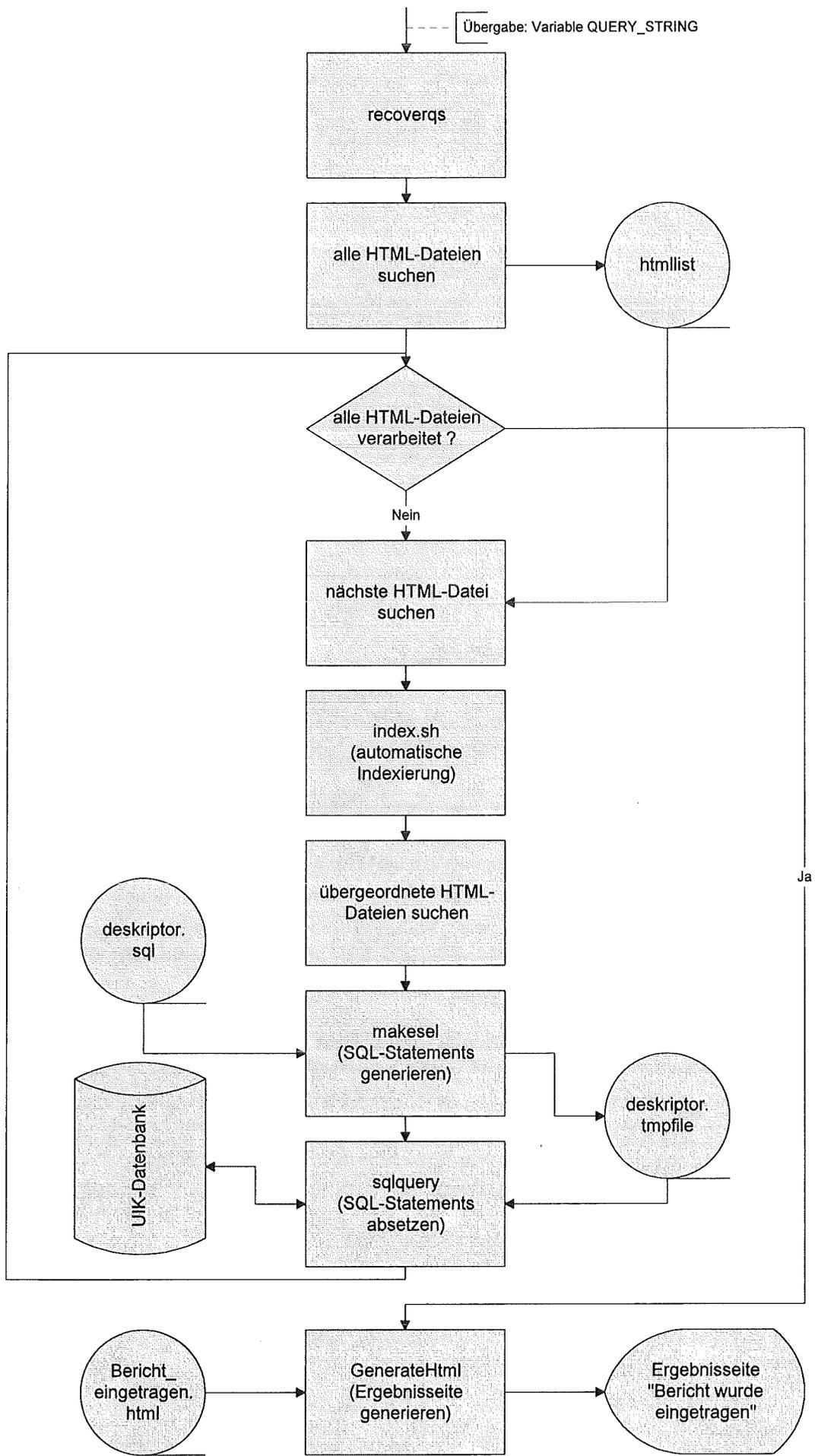


Abbildung 2-21: Ablauf von „berichte.sh“

Im folgenden wird der Ablauf von „berichte.sh“ und „index.sh“ (automatische Verschlagwortung) genauer beschrieben.

berichte.sh

Zunächst „restauriert“ ein C-Programm („recoverqs“) die Variable QUERY_STRING, d.h. HTML-Steuerzeichen, die vom Browser konvertiert wurden, werden auf ihren ursprünglichen Wert zurückgesetzt. Danach wird aus QUERY_STRING der Pfadname ausgelesen und anschließend daraus gelöscht, da später dort der aktuelle Pfad des jeweiligen Berichtsabschnittes eingesetzt wird. Mit einem „find“-Befehl wird dann das Verzeichnis, in dem die Berichte lokal abgelegt sind, rekursiv nach HTML-Dateien durchsucht (Tiefensuche) und das Ergebnis in die Datei „htmllist“ geschrieben.

Im Anschluß daran werden in einer Schleife alle Dateien aus „htmllist“ verarbeitet, wobei der Verzeichnisbaum jetzt in einer Breitensuche durchlaufen wird. Auf diese Weise wird sichergestellt, daß die einem Berichtsabschnitt übergeordneten Berichtsabschnitte bereits eingetragen sind (was nicht der Fall wäre, ginge man in der Reihenfolge vor, in der die Dateinamen der Berichtsabschnitte in „htmllist“ abgelegt sind).

Bei der Verarbeitung einer HTML-Datei werden im ersten Schritt Deskriptoren für diesen Berichtsabschnitt gesucht. Dazu wird das Shell-Skript „index.sh“ aufgerufen, das unten genauer beschrieben ist. Als Ergebnis liefert dieses Skript einen String mit den herausgefilterten Deskriptoren, der an die Variable QUERY_STRING angehängt wird.

Im nächsten Schritt werden die dem Berichtsabschnitt übergeordneten Berichtsabschnitte gesucht. Dabei wird so vorgegangen, daß der Verzeichnisbaum vom jeweiligen Berichtsabschnitt ausgehend nach oben durchsucht wird. Sobald in einem Verzeichnis ein Berichtsabschnitt gefunden wird, wird die Suche beendet, ebenso, wenn das Wurzelverzeichnis erreicht ist. Wird kein übergeordneter Berichtsabschnitt gefunden (was beim obersten Abschnitt der Fall ist), wird „NULL“ als Verweis auf den oberen Bericht eingetragen.

Nachdem alle notwendigen Informationen in QUERY_STRING abgelegt sind, wird das Programm „makesel“ (C-Programm) aufgerufen, dem als Parameter diese Variable übergeben wird. „makesel“ generiert aus der in QUERY_STRING angegebenen Datei mit SQL-Statements (die noch Variablen enthalten) und den aktuellen Variablenwerten eine Datei mit SQL-Statements. Diese Datei wird als Parameter an das Programm „sqlquery“ (C-Programm) übergeben, das die SQL-Statements dann absetzt (zu „makesel“ und „sqlquery“ s. 3.1 WebQuery).

index.sh

Der Gesamtablauf von index.sh ist in Abbildung 2-22 dargestellt.

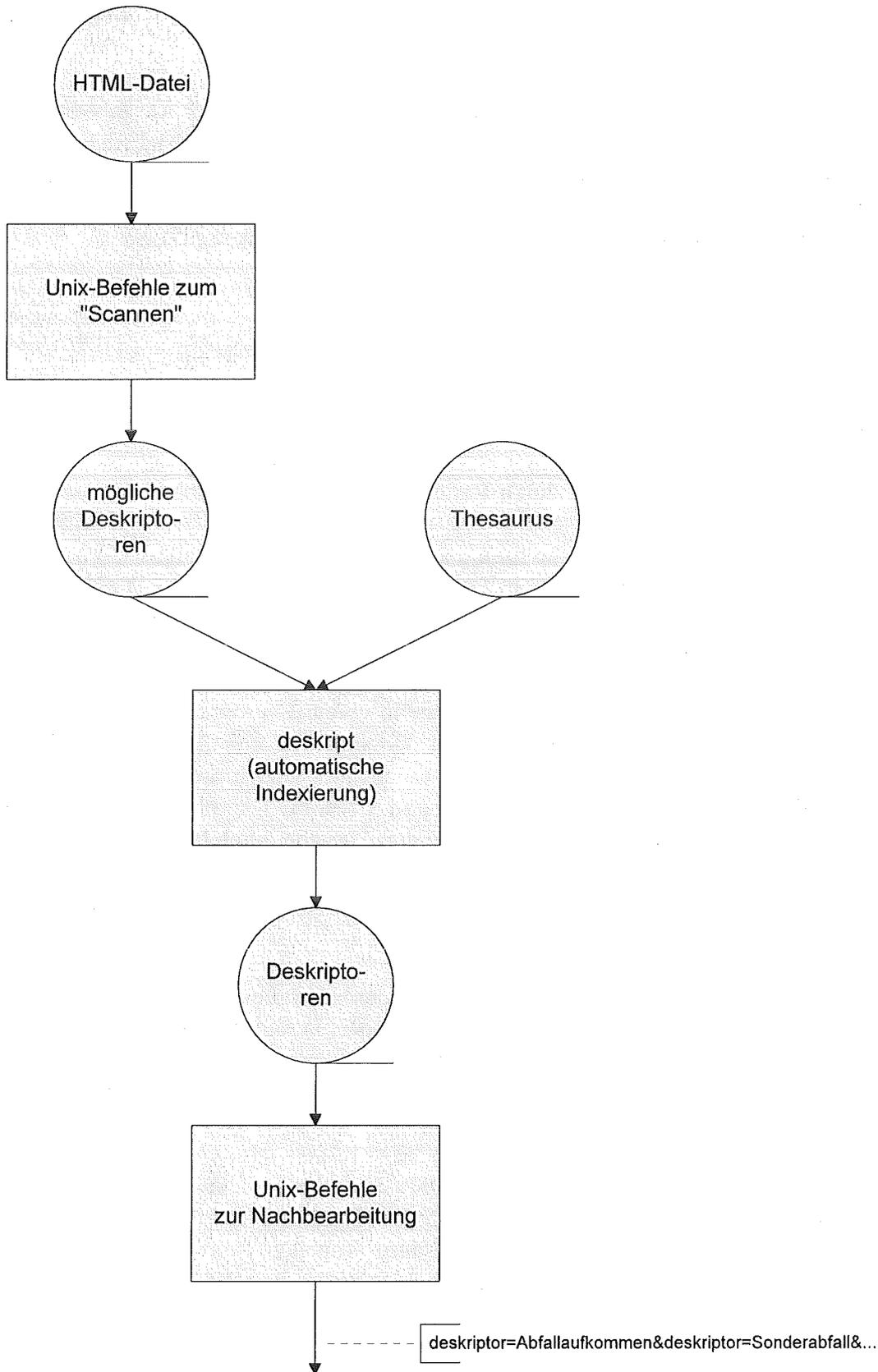


Abbildung 2-22: Ablauf von „index.sh“

„index.sh“ führt die Indexierung für eine HTML-Datei (Berichtsabschnitt) durch. Rückgabewert dieses Skripts ist ein String mit den herausgefilterten Deskriptoren, wobei der

String wie die Variable QUERY_STRING aufgebaut ist (also in der Form deskriptor=wert1&deskriptor=wert2&deskriptor=...).

Zuerst wird die HTML-Datei in einzelne Wörter aufgeteilt, d.h. jede Zeile der Ausgabe enthält dann ein Wort der HTML-Datei, wobei bereits eine Vorverarbeitung stattfindet. Die Aufteilung in Wörter erfolgt in mehreren Schritten mit Hilfe von Unix-Befehlen, die durch „Pipes“ miteinander verbunden sind.

Im ersten Teilschritt durchlaufen die Eingabedaten ein sed-Skript („deltag.sed“), das

- alle Zeilen bis <BODY> löscht,
- alle Hyperlinks löscht,
- alle sonstigen HTML-Tags löscht,
- Sonderzeichen durch Blanks ersetzt und
- Umlaute durch ae (oe, ue) bzw. ß durch ss ersetzt.

Danach ersetzen tr-Befehle alle Blanks durch Return (d.h. jede Zeile enthält nur noch ein Wort) und alle Groß- durch Kleinbuchstaben. Die daraus entstehende Wortfolge wird mit „sort“ alphabetisch sortiert und als Eingabe für ein awk-Programm („stopwort.awk“) benutzt, welches Stoppwörter, Zahlen, einbuchstabile Wörter und Leerzeilen entfernt. Die Ausgabe bis zu diesem Zeitpunkt besteht also aus einer Liste von Wörtern, die möglicherweise als Deskriptoren zu verwenden sind.

Diese Liste wird zusammen mit dem Thesaurus (s.u.) in einem C-Programm („deskript.c“) verarbeitet, welches aus den beiden Eingabedateien diejenigen Wörter herausucht, die in beiden Dateien vorkommen bzw. die sich nur durch die letzten beiden Buchstaben unterscheiden. Falls gleiche Wörter gefunden werden, zu denen im Thesaurus statt dessen zu benutzende Synonyme aufgeführt sind, werden sie in der Ausgabe durch die entsprechenden Begriffe ersetzt. Als Ergebnis liefert „deskript.c“ also eine Liste von Deskriptoren, wobei Deskriptoren, die im HTML-Dokument mehrfach vorkommen, auch mehrfach aufgeführt sind.

Diese Liste wird so bearbeitet, daß sie anschließend nach der Häufigkeit, mit der die einzelnen Deskriptoren vorkommen, sortiert ist; außerdem ist jeder Deskriptor nur noch einmal aufgeführt und alle Mehrwortdeskriptoren sind entfernt. Aus dieser Liste wird mit einem awk-Befehl der Rückgabestring zusammengesetzt, wobei hier nur die sieben häufigsten Deskriptoren berücksichtigt werden.

Für die Indexierung wird der in den UDK integrierte Umweltthesaurus des Umweltbundesamtes verwendet, der dafür als ASCII-Datei („thesaurus.txt“) zur Verfügung stehen muß. Diese ASCII-Datei wird aus den Thesaurus-Tabellen generiert und kann bei Bedarf (z. B. bei einer neuen Thesaurus-Version) in zwei Schritten neu erstellt werden:

1. Erstellen der Datei „thesaurus.lst“ unter „sqlplus“ (benötigte SQL-Statements s. Anhang). Hierbei werden die Tabellen thesdesc (Tabelle mit Deskriptoren) und thessyn (Tabelle mit Synonymen, s. 2.2.3.3 Thesaurus) benutzt, um eine temporäre Tabelle tmp_desc zu füllen, aus der dann „thesaurus.lst“ erstellt werden kann. „tmp_desc“ enthält eine Liste von Deskriptoren und Synonymen mit zugeordneten Deskriptoren.
2. Aufruf des Shell-Skripts „thesaurus.sh“, dem als Parameter die erstellte Liste aus Schritt 1 („thesaurus.lst“) zu übergeben ist. Dieses Skript entfernt Leerzeilen,

Leerzeichen am Zeilenende und doppelte Begriffe bei Kombinationen und schreibt das Ergebnis in die Datei „thesaurus.txt“.

Die Datei „thesaurus.txt“ enthält anschließend alle Deskriptoren und Nicht-Deskriptoren aus dem Thesaurus in alphabetischer Reihenfolge, wobei zu den Nicht-Deskriptoren in der folgenden Zeile der statt dessen zu verwendende Deskriptor aufgeführt ist (bzw. in den nächsten Zeilen mehrere Deskriptoren bei sog. Begriffs-Kombinationen).

2.4.2 Dienst eintragen

2.4.2.1 „Dienst eintragen“ aus Benutzersicht

Um das WWW-Formular für die Funktion „Dienst eintragen“ zu erhalten, muß in der Einstiegsseite des UIK der Verweis „Neuen Dienst katalogisieren“ ausgewählt und anschließend eine gültige Benutzerkennung und das Paßwort eingegeben werden (s.a. 2.4.1.2 „Bericht eintragen“ aus Benutzersicht)

Ein neuer Dienst kann katalogisiert werden, indem im WWW-Formular (s. Abbildung 2-23 für diese Funktionalität die Eingabefelder des Formulars mit den entsprechenden Attributwerten gefüllt werden und anschließend die Aktion „Dienst eintragen“ gestartet wird. Beim Katalogisieren eines Dienstes muß eine eindeutige Adresse festgelegt werden, über den der Dienst aufgerufen werden kann. Dies geschieht über die folgenden Felder, die zusammen die vollständige URL-Adresse bilden:

- **Dienstname:** Bei „Dienstname“ muß das Programm angegeben werden, über den der Dienst aufgerufen werden kann (z.B. ein Shell-Skript).
- **Pfad:** Der Pfad gibt den Suchweg bis zum Dienstprogramm an. Soll z.B. ein Dienst eingetragen werden, der die URL-Adresse *http://www.faw.uni-ulm.de/UFIS/Selektoren/abfallaufkommen.sh* hat, muß als Pfadname „UFIS/Selektoren/“ eingegeben werden.
- **Server:** Der Server, über den der Dienst zu erreichen ist, kann aus einer Liste von Servern ausgewählt werden. Zur Auswahl stehen alle Server, die der gleichen Instanz wie die eintragende Person zugeordnet sind.

Weiter sind folgende Felder von Bedeutung:

- **Beschreibung:** Hier kann ein beliebiger Text, der den Dienst beschreibt, eingegeben werden. Wenn der Dienst später Ergebnis einer Suche nach einem UIK-Objekt ist, wird u.a. dieser Beschreibungstext angezeigt.
- **Kategorie:** Für jeden Dienst kann eine Kategorie ausgewählt werden, zur Bedeutung dieses Attributes s. 2.2.3.1 Informationsquellenverwaltung.
- **HTML-Oberfläche:** Dieses Attribut gibt an, ob der einzutragende Dienst über eine HTML-Oberfläche verfügt oder nicht (derzeit können nur Dienste mit HTML-Oberfläche eingetragen werden).
- **Deskriptoren-Zuordnung:** Über dieses Feld können beliebige Deskriptoren aus dem UDK-Thesaurus mit dem Dienst verknüpft werden. Ausgewählt und in das

Eingabefeld übernommen werden die Deskriptoren über die Aktionen „Deskriptoren Volltextsuche“ oder „Deskriptoren alph. Suche“ (s. 2.3.2 Deskriptor)

- Raumbezug-Zuordnung: Mit Hilfe der Aktionen „Raumbezüge browsen“ und „Raumbezüge auswählen“ (s. 2.3.3 Raumbezug) können in dieses Eingabefeld Raumbezüge eingetragen werden.
- Zeitbezug-Zuordnung: Die Zeitbezug-Zuordnung erfolgt über die Aktion „Zeitbezüge auswählen“ (s. 2.3.4 Zeitbezug).

Neben den Aktionen zum Füllen der Eingabefelder gibt es in diesem Formular die Aktion „Dienst eintragen“, mit der nach dem Eintragen der gewünschten Werte das Eintragen dieser Metainformationen in den UIK veranlaßt werden kann.

Das erfolgreiche Eintragen des Dienstes wird mit einer Ergebnisseite angezeigt, von der aus man wieder zum Formular „Dienst eintragen“ zurückkehren kann.

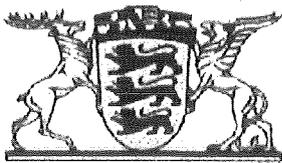
Netcape! Dienst - Eintragen

File Edit View Go Bookmarks Options Directory Help

Back Forward Home Reload Images Open Print Find Stop

Location:

What's New What's Cool Handbook Net Search Net Directory Newsgroups



Dienst - Eintragen

Allgemeine Dienstbeschreibung

Dienstname:

Pfad:

Server:

Beschreibung:

Kategorie:

HTML-Oberfläche: Ja Nein

Deskriptoren-Zuordnung

Document: Done

Abbildung 2-23: WWW-Formular für „Neuen Dienst katalogisieren“

2.4.2.2 Programmablauf von „Dienst eintragen“

Der Programmablauf bis zum Aufruf von „FilterOption.sh“ entspricht sinngemäß dem von „Bericht eintragen“ (s. 2.4.1.3 Programmablauf von „Bericht eintragen“), verwendet wird zum Generieren des WWW-Formulars für das Eintragen von Diensten die Beschreibungsseite „Dienst_eintragen.sql“.

Wurde im Eingabeformular die Aktion „Dienst eintragen“ ausgewählt, werden in „FilterOption.sh“ die in der Abbildung 2-24 dargestellten Programme zum Eintragen eines Dienstes aufgerufen.

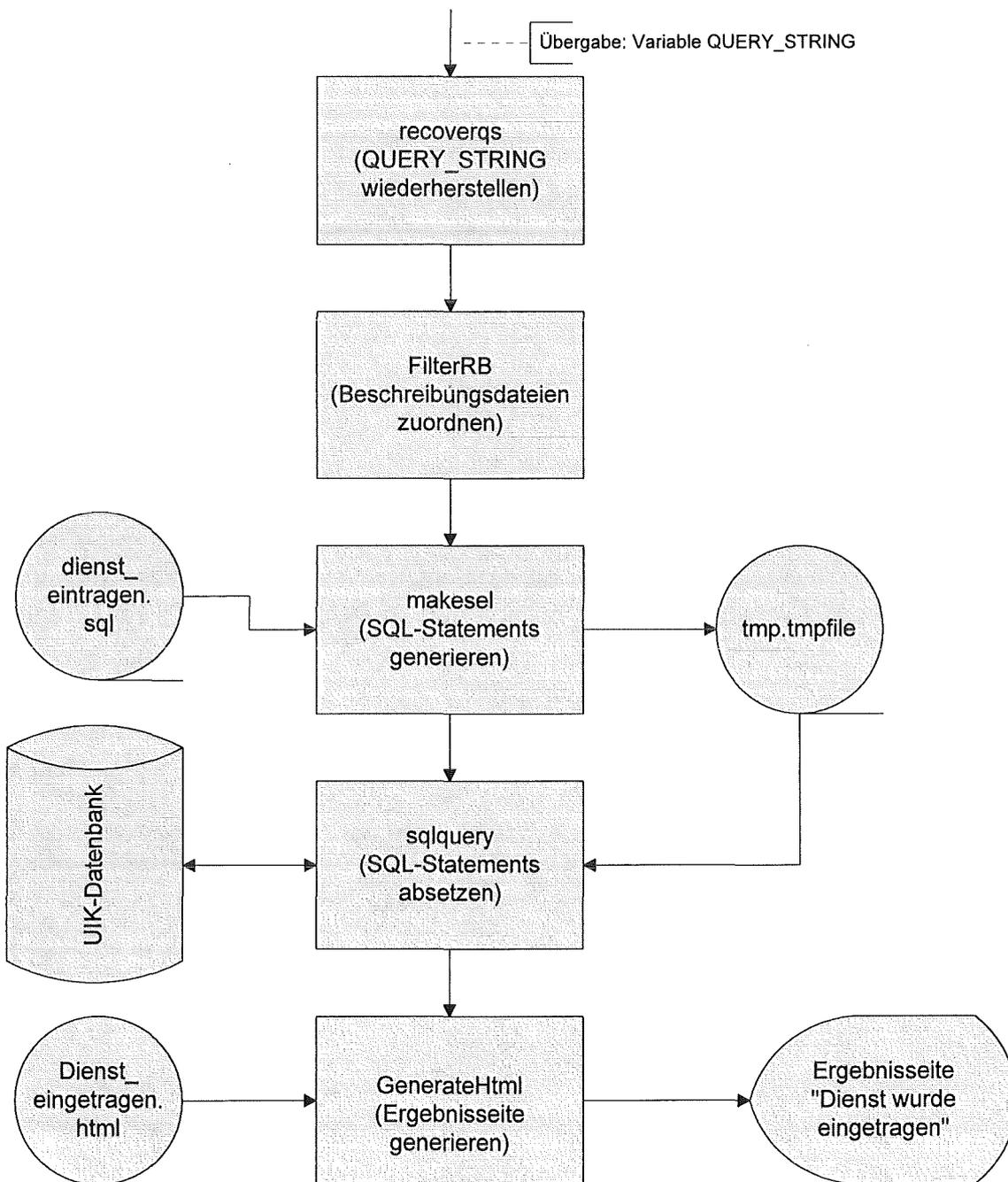


Abbildung 2-24: Programmablauf von „Dienst eintragen“

Als erstes wird das C-Programm „recoverqs“ zum Wiederherstellen der Variablen QUERY_STRING aufgerufen (s.a. 2.4.1.3 Programmablauf von „Bericht eintragen“). Da vom Formular „Dienst eintragen“ aus verschiedene Aktionen gewählt werden können, die Datenbankzugriffe erfordern, muß eine Zuordnung der für diesen Datenbankzugriff nötigen Beschreibungsdateien zu der jeweiligen Aktion erfolgen. Erledigt wird diese Aufgabe durch das C-Programm „FilterRB“, das der Aktion „Dienst eintragen“ die Dateien „dienst_eintragen.sql“ und „Dienst_eingetragen.html“ zuordnet. Aus den Angaben in „dienst_eintragen.sql“ und den Werten in QUERY_STRING generiert „makesel“ eine Datei mit SQL-Statements, die dann von „sqlquery“ abgesetzt werden. Danach wird „GenerateHtml“ zum Anzeigen der Ergebnisseite aufgerufen.

2.4.3 UIK-Objekt suchen

2.4.3.1 „UIK-Objekt suchen“ aus Benutzersicht

Das WWW-Formular zum Suchen nach UIK-Objekten erreicht man (wie die Formulare zum Eintragen von UIK-Objekten auch) über die Einstiegsseite des UIK.

Die Suche nach UIK-Objekten kann durch die folgenden Eingabefelder im Formular (s. Abbildung 2-25 eingeschränkt werden:

- Art des UIK-Objektes: Hier kann angegeben werden, ob nur nach bestimmten UIK-Objekten gesucht werden soll (Option „Bericht“ oder „Dienst“) oder ob sich die Suche auf sämtliche UIK-Objekte erstrecken soll (Option „Alle“).
- Datenhaltende Stelle: Die datenhaltenden Stellen sind in einer Liste aufgeführt, aus der ausgewählt werden kann. Soll sich die Suche nicht auf eine bestimmte datenhaltende Stelle beziehen, muß die Option „Alle datenhaltenden Stellen“ ausgewählt werden.
- Deskriptoren-Zuordnung: In dieses Feld können über die Aktionen „Deskriptoren Volltextsuche“ oder „Deskriptoren alph. Suche“ (s. 2.3.2 Deskriptor) beliebige Deskriptoren aus dem UDK-Thesaurus eingetragen werden. Bei der Suche werden diese Deskriptoren durch eine ODER-Beziehung verknüpft, d.h. ein UIK-Objekt muß nur mit mindestens einem der gewählten Deskriptoren verknüpft sein, um als Suchergebnis angezeigt zu werden. Zusätzlich wird bei der Suche nach Diensten auch nach solchen Objekten gesucht, deren Deskriptoren Unterbegriffe der ausgewählten Deskriptoren sind.
- Raumbezug-Zuordnung: Raumbezüge, nach denen gesucht werden soll, können über die Aktionen „Raumbezüge browsen“ und „Raumbezüge auswählen“ eingetragen werden (s. 2.3.3 Raumbezug). Bei der Suche nach UIK-Objekten werden auch diejenigen Objekte berücksichtigt, deren Raumbezug Überschneidungen mit den hier gewählten Raumbezügen aufweist.
- Zeitbezug-Zuordnung: Die Zeitbezug-Zuordnung erfolgt über die Aktion „Zeitbezüge auswählen“ (s. 2.3.4 Zeitbezug); auch hier werden wie bei den Raumbezügen UIK-Objekte gesucht, deren Zeitbezug sich mit den angegebenen Zeitbezügen überschneidet.

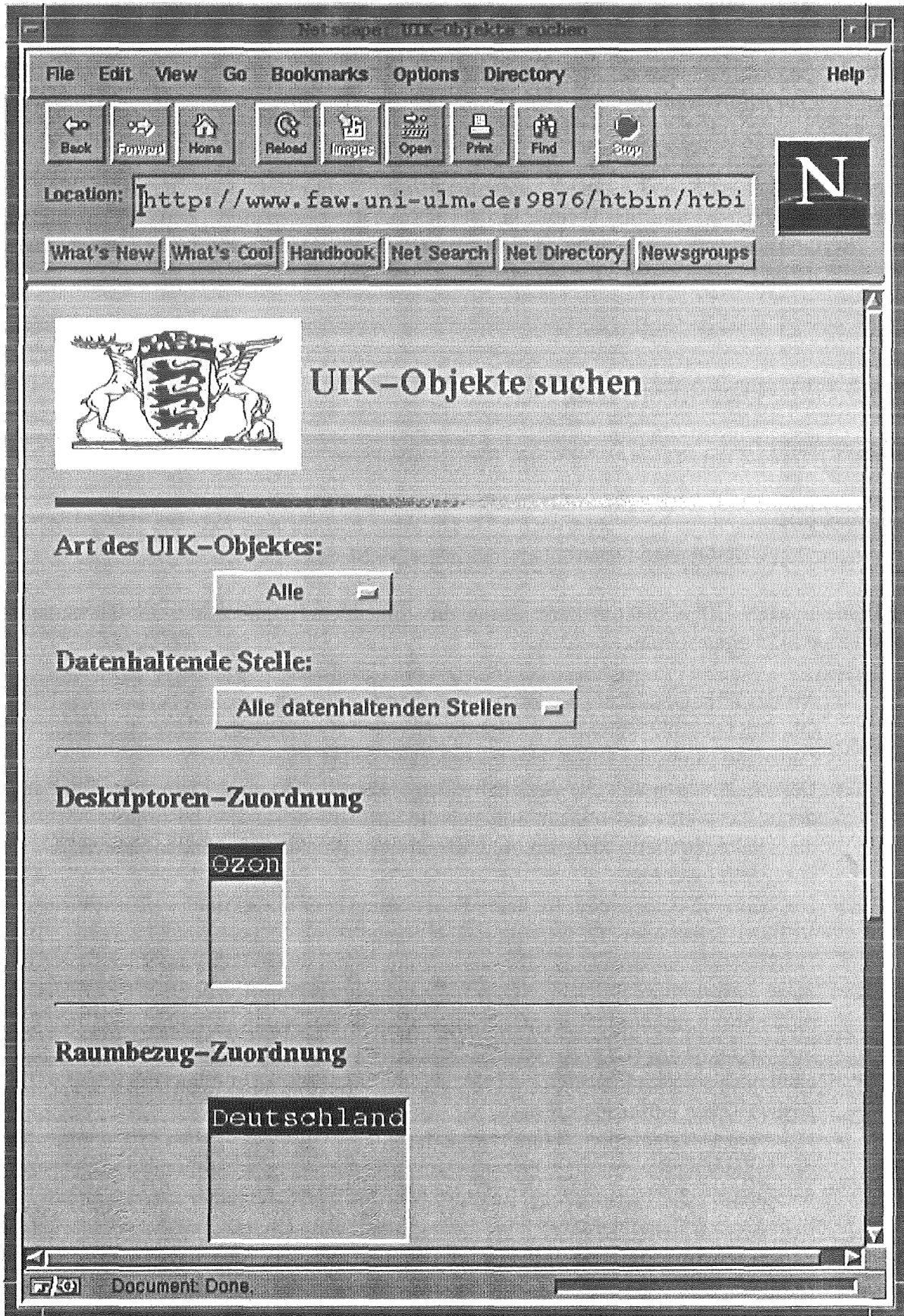


Abbildung 2-25: WWW-Formular für „UIK-Objekt suchen“

Werden in den Feldern „Deskriptoren-Zuordnung“, „Raumbezug-Zuordnung“ und „Zeitbezug-Zuordnung“ keine Angaben gemacht, bedeutet das, dass die Suche hinsichtlich dieser Attribute nicht eingengt wird (also z.B. alle UIK-Objekte gesucht werden, unabhängig davon, welchen Raumbezug sie haben).

Sind alle gewünschten Eintragungen gemacht, kann über die Aktion „UIK-Objekt suchen“ der Suchvorgang angestoßen werden.

Ist der Suchvorgang beendet, wird eine Ergebnisseite angezeigt mit allen UIK-Objekten, die die Suchkriterien erfüllen (Bsp. s. Abbildung 2-26). Diese Ergebnisseite ist nach den verschiedenen UIK-Objektarten geordnet, innerhalb jeder Objektart findet eine weitere Untergliederung nach der Zugriffsberechtigung statt (d.h. es gibt beispielsweise „Dienste mit Zugriffsrecht“, „Dienste ohne Zugriffsrecht“ oder „Berichte mit Zugriffsrecht“). Zu jedem gefundenen Objekt gibt es eine Beschreibung, die aus dem Objekt-Namen und einem Beschreibungstext besteht. Handelt es sich um ein Objekt, für das der Benutzer zugriffsberechtigt ist, wird unter der Beschreibung ein Hyperlink angezeigt. Durch Anklicken dieses Verweises kann auf das jeweilige UIK-Objekt zugegriffen werden, also z.B. ein Bericht angezeigt oder ein Dienst aufgerufen werden.

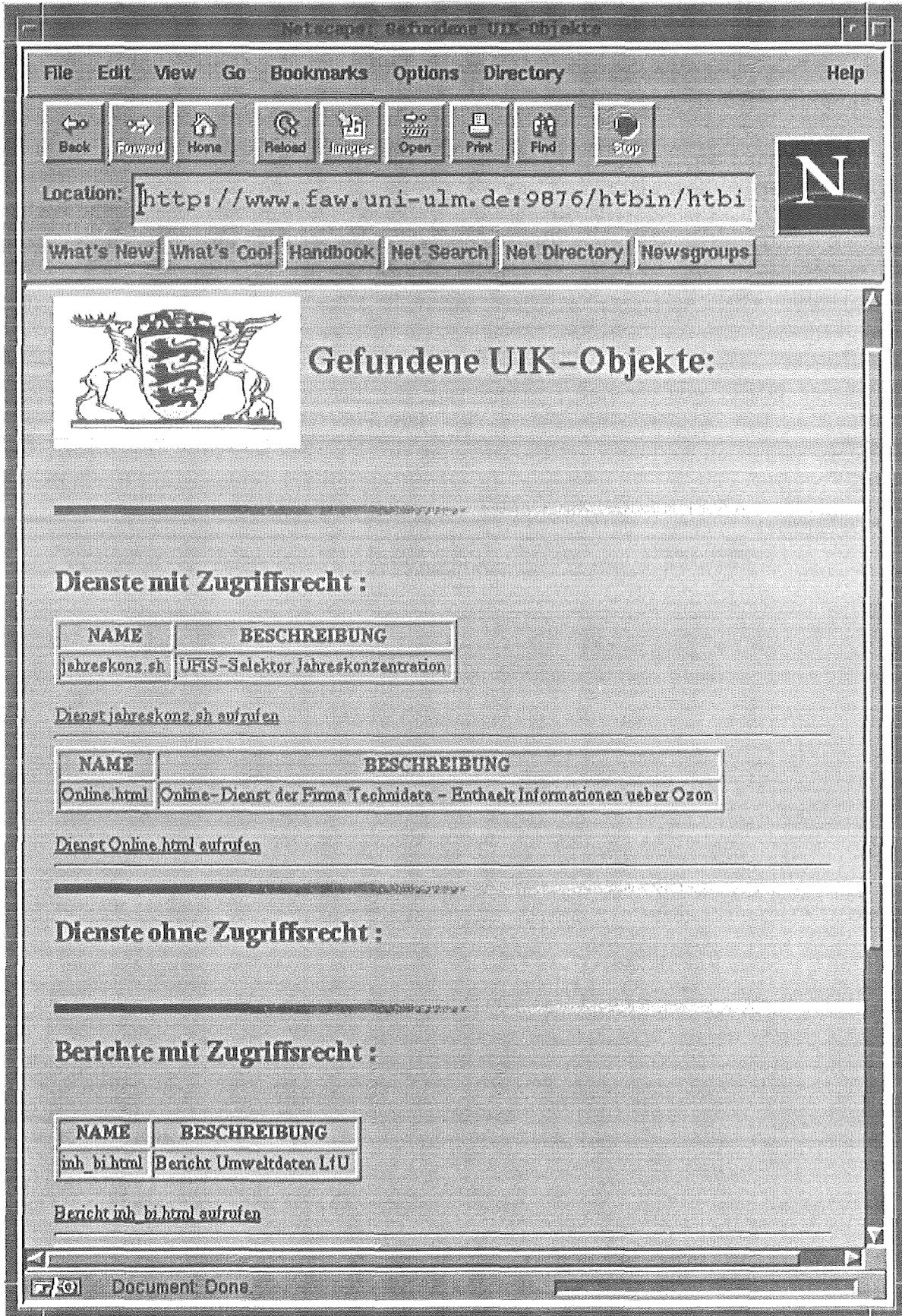


Abbildung 2-26: Ergebnisseite eines Suchvorganges

2.4.3.2 Programmablauf von „UIK-Objekt suchen“

Der Programmablauf von „UIK-Objekt suchen“ (s. Abbildung 2-27) entspricht weitgehend dem Eintragen eines neuen Dienstes (s. dazu 2.4.2.2 Programmablauf von „Dienst eintragen“). Im Unterschied zum „Dienst eintragen“, bei dem die Datenbankzugriffe im wesentlichen aus Einfügungen in Tabellen bestehen, werden hier jedoch Suchergebnisse zurückgeliefert, die in der Datei „tmp.suchen.tmpfile“ abgespeichert werden. Diese Datei wird vom C-Programm „htmlaufb“ eingelesen, welches die Werte für eine Darstellung im HTML-Format aufbereitet. Die Art der Aufbereitung wird dabei durch die Beschreibungen in der Datei „suchen.html“ festgelegt. Das aufbereitete Ergebnis wird in der Datei „tmp.html“ abgelegt und durch das Programm „GenerateHtml“ angezeigt.

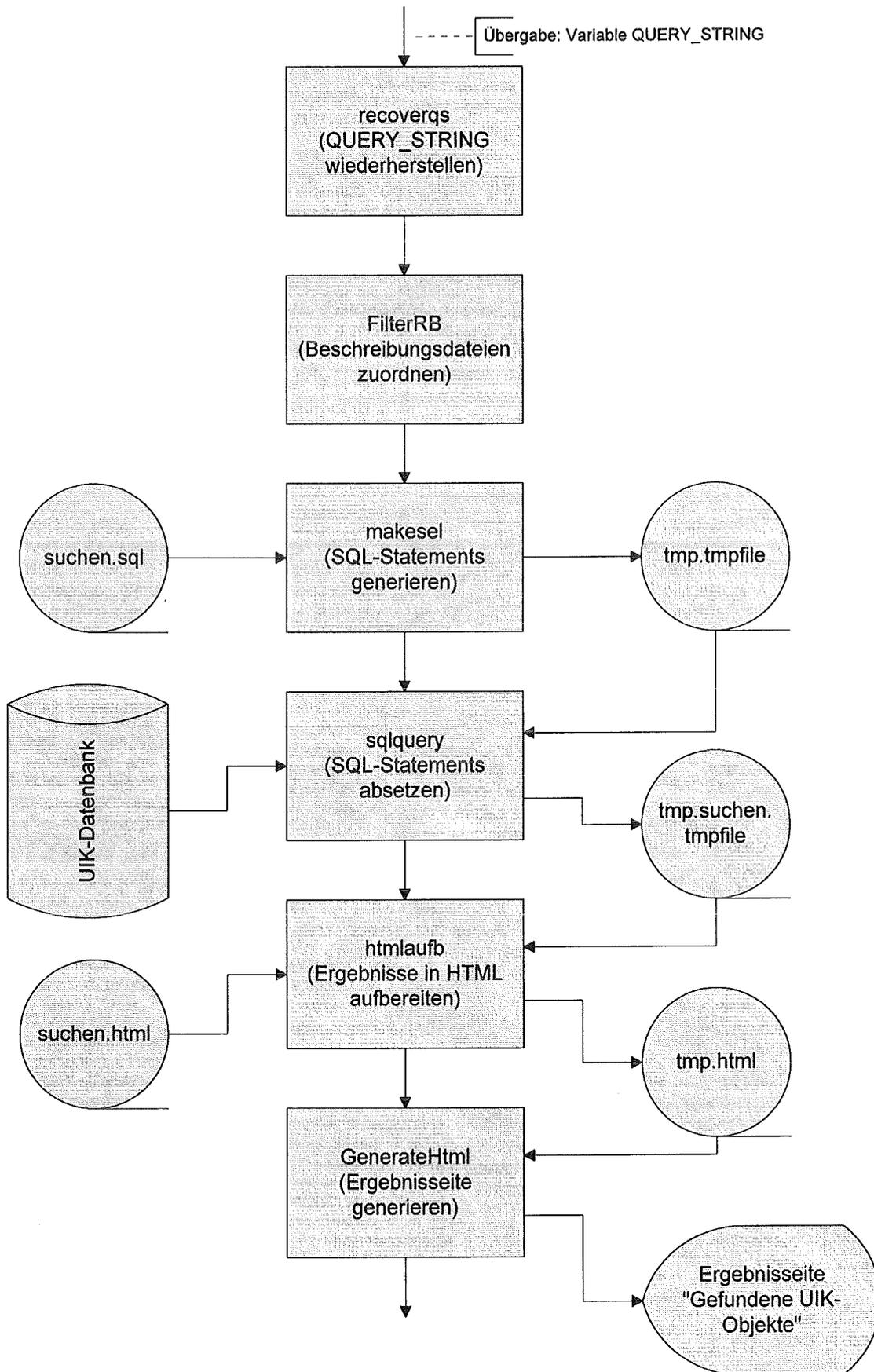


Abbildung 2-27: Programmablauf von „UJK-Objekt suchen“

3. FAW-Basistools

3.1 WebQuery

WebQuery ist ein Tool, das es erlaubt, Abfragen von ORACLE-Datenbanken als Dienste unter WWW zur Verfügung zu stellen. Bei der Entwicklung des Systems wurde streng auf den modularen Aufbau der einzelnen Komponenten geachtet. Das Tool wurde in drei Hauptkomponenten zerlegt, die jeweils Abhängigkeiten von der Systemumgebung aufweisen; somit können diese Programmodule jeweils durch weitere systemabhängige Module ersetzt werden. Damit wird eine weitgehende Systemunabhängigkeit und Erweiterbarkeit des Tools durch die Austauschbarkeit einzelner Module und durch einfaches Einsetzen weiterer Module erreicht. WebQuery wurde am FAW entwickelt und in seiner jetzigen Form erstmals für die Anbindung von UFIS-Selektoren eingesetzt. Desweiteren fand es Anwendung bei der Realisierung des Umweltinformationsquellenkatalogs, sowie in anderen Bereichen am FAW in Ulm. Das Tool WebQuery hat sich beim Einsatz in der Entwicklung von Datenbankabfragesystemen durch seine leicht verständlichen Definitionsformen als schnell erlernbar und auch von einem EDV-Amateur anwendbar erwiesen. Prinzipiell sind nur Kenntnisse über HTML-Seitenerstellung und SQL-Abfrage-Formulierung von Nöten.

3.1.1 WebQuery aus Benutzersicht

Der Dienstenutzer kann über eine Einstiegsseite, die als WWW-Formular realisiert ist, einen Datenbank-Dienst nutzen (siehe Abb. 3-1). Auf der Eingabeseite befinden sich die von ihm frei wählbaren Parameter für die entsprechende Abfrage. Hierbei ist auch eine multiple Auswahl von Parametern erlaubt. Dazu füllt er die Eingabefelder des Formulars entsprechend aus oder wählt vorformulierte Parameter aus den Listboxen und startet dann über eine Schaltfläche zur Datenfreigabe („Submit“-Button) die eigentliche Anfrage an die Datenbank. Hierbei werden die von ihm selektierten Parameter auf der HTML-Seite in die parametrisierten Select-Statements eingesetzt und diese dann vom Programm als Anfrage an die Datenbank weitergegeben.

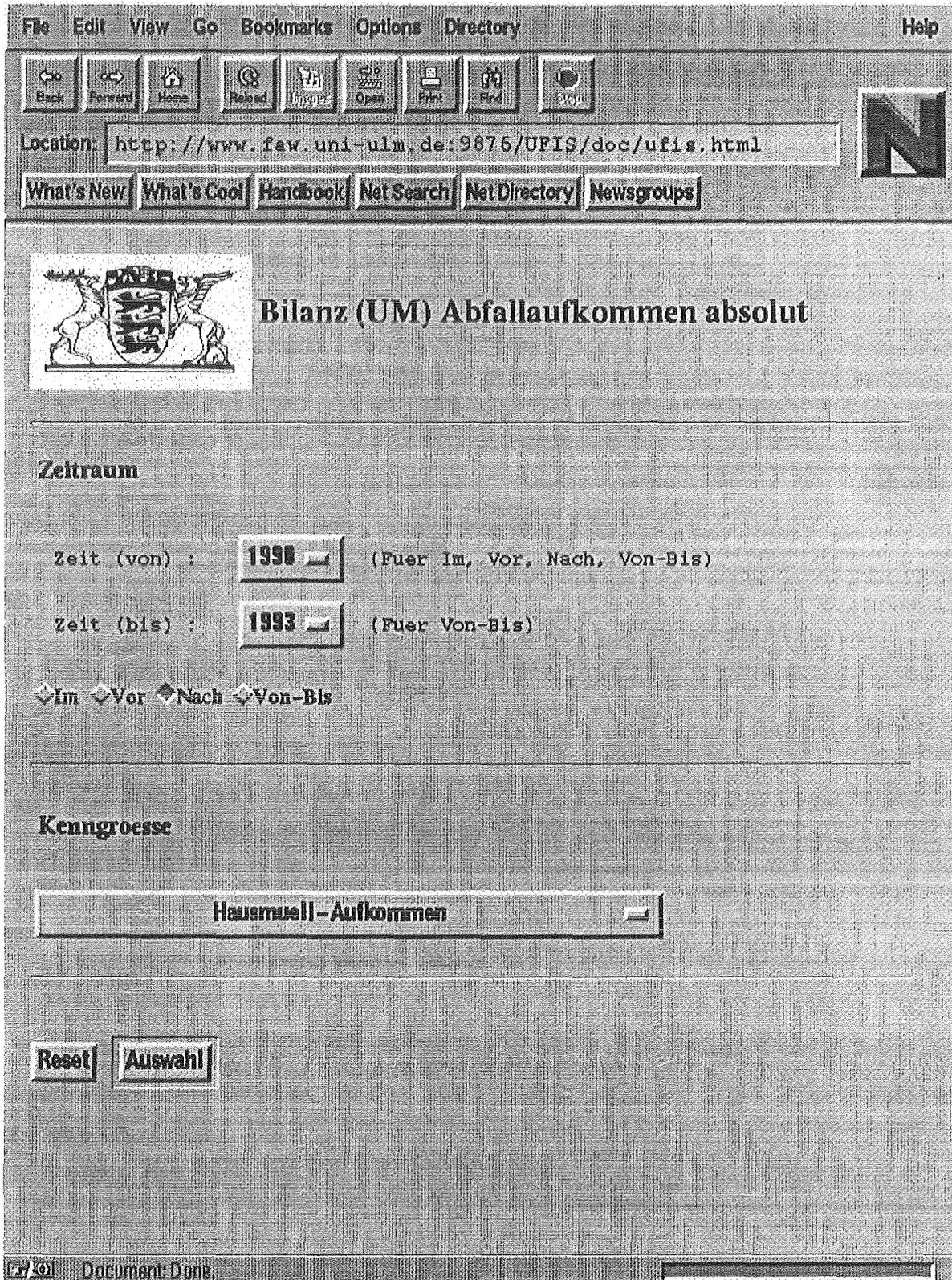


Abbildung 3-1: Einstiegsseite für die Datenbankabfrage

Nach Betätigen des Submit-Buttons wird die benutzerdefinierte Abfrage abgesetzt und die Ergebnisse dieser Anfrage werden in vom Diensteanbieter definierter Form aufbereitet und dem Benutzer in einer WWW-Seite präsentiert (z.B. als Tabelle, Listbox ...; siehe Abb. 3-2). Hierbei können die Ergebniswerte wiederum als selektierbare Auflistungen in der WWW-

Ergebnisseite aufbereitet und als Eingabefelder für eine erneute Anfrage verwendet werden. Somit ist es möglich, ganze Abfragesequenzen zu realisieren. Mit dem Tool WebQuery können aus einer Eingabeseite nicht nur Selektionen auf die Datenbank formuliert werden, sondern ebenso Insert-, Delete- und Update-Statements, die ein interaktives Arbeiten mit Oracledatenbeständen in jeder Hinsicht erlauben.

Content-type: text/html

Liste der gefundenen Werte

VE_KENNGROESSEN_NR	VE_KENNZ	ZEIT	WERT	HINWEIS
168	8	1992	2038314	0
168	8111	1992	154587	0
168	8115	1992	63122	0
168	8116	1992	80195	0
168	8117	1992	61433	0
168	8118	1992	123579	0
168	8119	1992	68556	0
168	8121	1992	27557	0
168	8125	1992	49767	0
168	8126	1992	23514	0
168	8127	1992	48607	0
168	8128	1992	18764	0
168	8135	1992	33434	0
168	8136	1992	54366	0
168	8211	1992	14162	0
168	8212	1992	65757	0
168	8215	1992	63798	0
168	8216	1992	40145	0
168	8221	1992	28997	0
168	8222	1992	90400	0
168	8225	1992	26104	0
168	8226	1992	86505	0
168	8231	1992	25203	0
168	8235	1992	27172	0
168	8236	1992	44005	0

Document Done

Abbildung 3-2: WWW-Seite der aufbereiteten Daten resultierend aus der Abfrage

3.1.2 WebQuery aus der Sicht eines Datenbank-Diensteanbieters

Um diesen für den Dienstenutzer beschriebenen Ablauf zu ermöglichen, muß der Diensteanbieter für jeden Datenbankdienst drei Beschreibungsdateien erstellen:

- **Beschreibung des Eingabefelds (x.html)**

Bei dieser Datei handelt es sich um ein HTML-Formular, das heißt, es werden mit den üblichen HTML-Befehlen die für die jeweilige Anwendung benötigten Eingabefelder festgelegt. Die Namen der Variablen des HTML-Formulars entsprechen den Variablen der parametrisierten Select-Statements, die mit den benutzerdefinierten Werten aus dem Eingabefeld bei der Anwendung ersetzt werden. Zusätzlich existieren zwei verborgene Formularelemente, die den Dateinamen der x.result-Datei (im Beispiel Abfall1.result) und des Files mit den Select-Statements x.query (im Beispiel Abfall1.query) festlegt.

```
<HTML>
<TITLE>Bilanz (UM) Abfallaufkommen absolut</TITLE>

<Form method="GET" ACTION="http://faw.uni-ulm.de:9876/cgi-
bin/htbingsql/setval.sh">
<H2><img SRC=http://Umweltdaten/lud/1_titel/glawapp.gif ALIGN=MIDDLE> Bilanz (UM)
Abfallaufkommen absolut </H2>
<HR>
<PRE>
<H3> Zeit</H3>

    Zeitpunkt :
<SELECT NAME="zeit">
    <OPTION SELECTED> 1989
    <OPTION> 1990
    <OPTION> 1991
    <OPTION> 1992
    <OPTION> 1993
</SELECT>

<HR>

<H3> Kenngröesse </H3>

<SELECT NAME="Kenngröesse">
<OPTION> Altglas-Aufkommen
<OPTION> Altpapier-Aufkommen
<OPTION> Bio-u. Gruenabfall-Aufkommen
<OPTION> Deponierte Muellmenge
<OPTION> Gesamtmuellaufkommen (incl. Wertstoffe usw.)
<OPTION> Gewerbe-u.Baustellenabfaelle-Aufkommen
<OPTION> Hausmuell+Sperrmuell - Aufkommen
<OPTION> Hausmuell+Sperrmuell+Gewerbemuell+Baustellenabfaelle
<OPTION SELECTED> Hausmuell-Aufkommen
<OPTION> Problemstoffe-Aufkommen
```

```
<OPTION> Schrott-Aufkommen
<OPTION> Sperrmuell-Aufkommen
<OPTION> Wertstoff-Aufkommen
</SELECT>

<HR>

<input Type="reset" value="Reset"> <input Type="submit" value="Auswahl">

<input Type="hidden" name=queryfile value="Abfall1.query">
<input Type="hidden" name=resultfile value="Abfall1.result">

</PRE>
</FORM>
</HTML>
```

Abbildung 3-3: Die Datei x.html

- **Beschreibung der Select-Statements (x.query)**

In dieser Datei werden die Select-Statements definiert, die in der Ergebnisseite dargestellt werden sollen. Die Select-Statements werden in ihrer grundsätzlichen Formulierung nicht verändert. D.h. in der Datei x.query werden prinzipiell nur SQL-Statements eingetragen, die jedoch einen eindeutigen Namen erhalten müssen, damit sie bei der Ergebnisaufbereitung den Aufbereitungsattributen der Ergebnisseite zugeordnet werden können, da in der Ergebnisseite mehrere Datenbankabfrageergebnisse dargestellt werden können. In der x.select-Datei wird die Kennung des Select-Statements mit dem Schlüsselwort NAME, gefolgt von dem eindeutigen Namen des Statements, eingetragen und mit einem Semikolon und einem Return abgeschlossen. In der darauffolgenden Zeile wird die SQL-Abfrage eingetragen, in der die durch Werte des Eingabefelds zu ersetzenden Parameter durch ein '\$' am Anfang und Ende des Variablennamens gekennzeichnet werden (\$var\$). Da diese Parameter der SQL-Statements durch die Werte der Variablen aus den HTML-Formularen ersetzt werden, können diese mehrere Werte besitzen. Somit müssen Teile des Statements vervielfacht und durch „oder“ verknüpft werden. Diese zu multiplizierenden Teile werden mit einem '^' am Anfang und am Ende signalisiert. Ein mit „**^upper(ZEIT) >= upper('\$zeit\$')^**“ eingetragenes Statement wird wie folgt konvertiert : **upper(ZEIT) >= upper('1987') or upper(ZEIT) >= upper('1987')**. Ein zu wiederholender Block kann auch mehrere Parameter gleichzeitig beinhalten. Beim Ersetzen mit den entsprechenden Werten werden sämtliche Listen bis zu ihrem Ende durchlaufen und deren Werte werden eingetragen. Sind die Listen unterschiedlich lang, wird bis zum Abarbeiten der letzten Liste am Ende der kürzeren gepollt. Darüberhinaus existieren SQL-Statements, die als Parameter Ergebniswerte vorhergehender Datenbankabfragen benötigen. Diese vorrangig abzusetzende Datenbankabfrage wird in der oben beschriebenen Form in das x.select-File eingetragen und mit einem eindeutigen Namen versehen. Dieser Name wird als Parameter in das die Ergebniswerte beinhaltende Statement mit einem '@' gekennzeichnet eingetragen. An diese Stelle werden dann bei der Datenbankabfrage die Ergebnisse des vorherigen Statements eingetragen.

```
#-----  
#--- Variablenname des SELECT_STATEMENTS  
NAME sell;  
  
#--- SELECT_STATEMENTS  
SELECT * FROM VE_KENNGROESSEN_WERTE WHERE ^upper(ZEIT) >= upper('$zeit$')^ AND  
^VE_KENNGROESSEN_WERTE.VE_KENNGROESSEN_NR = '$Kenngroesse$' and GEBIET =  
@GEBIETS_SELECT^;  
  
#----ENDE_SELECT_STATEMENT
```

Abbildung 3-4: Die Datei x.query

- **Beschreibung der Ergebnisseite (x.result)**

In dieser Datei werden Aufbau und Inhalt der Ergebnisseite festgelegt. Das Aussehen der Ergebnisseite wird durch Standard-HTML-Befehle definiert, gekennzeichnet durch das Schlüsselwort HTML zu Beginn. Die Stelle, an der jeweils Ergebnisdaten eingefügt werden sollen, wird mit dem Schlüsselwort QUERY gekennzeichnet, gefolgt von einem Variablennamen, der im x.select-File das Select-Statement definiert, dessen Abfrageergebnisse an dieser Stelle eingefügt werden sollen. In welcher Form die Ergebnisdaten dargestellt werden sollen, wird durch zusätzliche Schlüsselwörter (z.B. FORMAT, BORDER) anschließend an das Schlüsselwort QUERY beschrieben. Abgeschlossen werden sämtliche mit Schlüsselworten gekennzeichneten Angaben jeweils durch Semikolon und Return. (Abb. 3-5)

```
#-----  
#--- HTML Text  
HTML  
<H2><IMG SRC= "/Umweltdaten/1ud/1_titel/glawapp.gif" align=middle>  
  Liste der gefundenen Werte</H2>  
<FORM method="POST">;  
  
#-----  
#--- Query-String  
QUERY  
sell;  
  
#-----  
#--- Definition of table for the DB-Data  
FORMAT table;  
BORDER yes;  
STYLE vertikal;  
ALIGN center;  
VALIGN center;  
  
#-----  
#--- Additional HTML Text  
  
HTML  
</FORM>
```

```
<P> <A HREF="http://www.faw.uni-ulm.de:9876/UFIS/doc/ufis.html"><img  
SRC="/icons/back.xbm" ALIGN=MIDDLE> Zurueck zum Selektor</A>  
<BR><A HREF="http://www.faw.uni-ulm.de:9876/UFIS/doc/ufis_aufb.html"><img  
SRC="/icons/forward.xbm" ALIGN=MIDDLE> Weiter zur Aufbereitung</A>;
```

Abbildung 3-5: Die Datei x.result

Im folgenden wird eine Übersicht über die neben HTML und QUERY verwendeten Schlüsselwörter gegeben. Prinzipiell müssen die Schlüsselwörter HTML und QUERY mit dazugehörigem FORMAT mit seinen Attributen in der Reihenfolge definiert werden, in der sie in der Result-Datei dargestellt werden sollen.

HTML

Definition der HTML-Seite ausschließlich des Blocks mit den aufbereiteten Daten aus der Datenbank-Anfrage.

QUERY

Kennzeichnet das Select-Statement, das an dieser Stelle in der mit FORMAT angegebenen Form dargestellt werden soll.

FORMAT [table | list | listbox | pulldown]

legt fest, in welcher Form die Daten aufbereitet werden

WHICH_COL

Nur die nachfolgend aufgelisteten, mit ‘&’ getrennten Spaltennamen, werden aus der Ergebnistabelle aufbereitet und in der HTML-Seite visualisiert. Fehlt diese Angabe, werden automatisch alle Spalten dargestellt. Es kann auch explizit das Schlüsselwort „all“ oder „no“ eingetragen werden, um alle oder keine Spalten aufzulisten.

COL_AS_URL

Hier können direkt HTML-Statements eingetragen werden, in denen die Spaltennamen mit ‘\$’ gekennzeichnet werden, deren Werte eingetragen werden sollen:

<HREF = \$PFAD> DAS BILD \$BILD_NAME

FORMAT table benötigt folgende Schlüsselwörter als Attribute:

BORDER [yes | no]
STYLE [vertical | horizontal]

vertical

Spalte1	Spalte2
wert1	wert3
wert2	wert4

horizontal

Spalte1	wert1	wert2
---------	-------	-------

Spalte2	wert3	wert3
---------	-------	-------

ALIGN [left | center | right]
VALIGN [top | center | bottom]
SINGLE_TAB_ROWS [yes | no]

no

Spalte1	Spalte2
wert1	wert3
wert2	wert4

yes

Spalte1	Spalte2
wert1	wert3

Spalte1	Spalte2
wert2	wert4

Folgende Attribute haben nur Gültigkeit wenn SINGLE_TAB_ROW mit „yes“ belegt wurde.

TAB_ROW_SEP

<HR>

Spalte1	Spalte2
wert1	wert3

Spalte1	Spalte2
wert2	wert4

TAB_ROW_SELECT [no | single | multiple]

Zwischen jeder Zeile werden Buttons eingetragen und die Zeile somit selektierbar gemacht, falls nicht no eingetragen wurde. Wird „single“ eingetragen, werden Toggle-Buttons eingetragen, sonst Checkboxes. Als Name der Variablen wird der Name des Select-Statements, dessen Ergebnisse eingetragen werden, verwendet und Value wird die Zeilennummer. Das Aufbereitungsprogramm prüft, ob eine Variable mit dem Namen des darzustellenden Statements gesetzt ist und stellt dann nur die entsprechende Zeile der Ergebnistabelle dar.

TAB_COL_NAME [yes | no]

Legt fest, ob die Ergebnisse mit Spaltenbezeichner aufbereitet werden oder nicht.

FORMAT list benötigt folgende Schlüsselwörter als Attribute:

TYP [column, field]

COL_SEP [Trennzeichen]

LIST_ROW_SEP [Trennzeichen] (Nur bei Typ field relevant!!)

z.B. TYP column COL_SEP <HR>

Spalte1:

wert1

wert2

wert3

Spalte2:

wert4

wert5

wert6

z.B. TYP column COL_SEP <HR> <HR>

LIST_ROW_SEP <center>-----</center>

Spalte1:

wert1

wert2

wert3

Spalte2:

wert4

wert5

wert6

LIST_ROW_SELECT [no | single | multiple]

Zwischen jeder Zeile werden Buttons eingetragen und die Zeile somit selektierbar gemacht, falls nicht no eingetragen wurde. Wird „single“ angegeben, werden Toggle-Buttons eingetragen, sonst Checkboxen. Name der Variablen wird der Name des Select-Statements, dessen Ergebnisse eingetragen werden und Value wird die Zeilennummer. Das Aufbereitungsprogramm prüft, ob eine Variable mit dem Namen

des darzustellenden Statements gesetzt ist und stellt dann nur die entsprechende Zeile der Ergebnistabelle dar.

LIST_COL_NAME [yes | no]
Legt fest, ob die Ergebnisse mit Spaltenbezeichner aufbereitet werden oder nicht.

FORMAT listbox benötigt folgende Schlüsselwörter als Attribute:

SIZE	[zahl] (Anzahl der Spalten in der Listbox)
MULTIPLE_SEL	[yes no] (Mehrfache Selektionen erlaubt)
BOX_COL_NAME	[variablenname] (Variablenname der Box)

FORMAT pulldown benötigt folgendes Schlüsselwort als Attribut:

PULL_COL_NAME	[variablenname] (Variablenname des Pulldowns)
----------------------	---

3.1.3 Ablauf von WebQuery

Der grundsätzliche Ablauf des Programmes "WebQuery" ist in der folgenden Abbildung dargestellt.

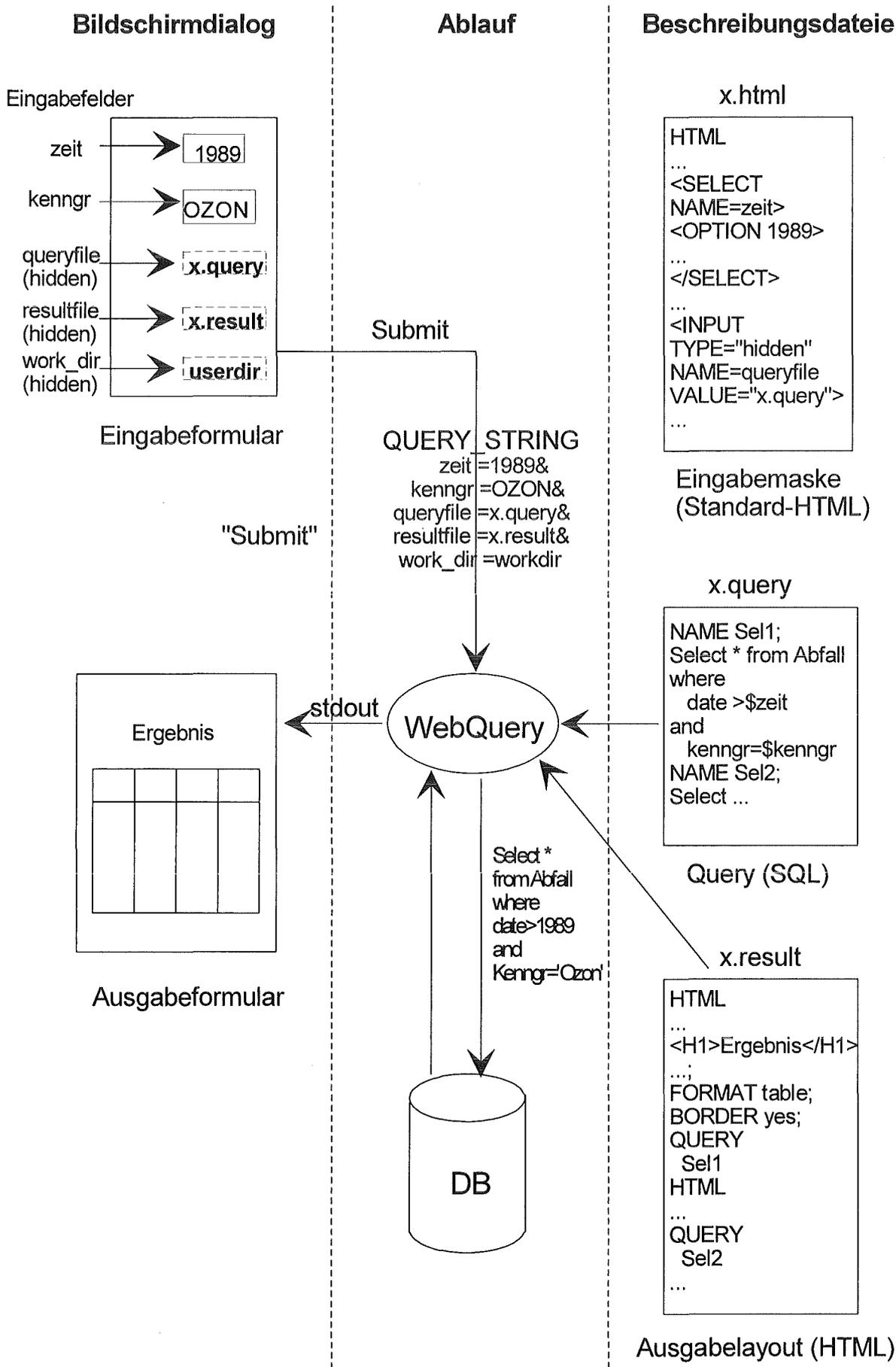


Abbildung 3-6: Ein- und Ausgabedateien von WebQuery

Im HTML-Formular (Datei x.html) müssen zunächst die festgelegten Eingabefelder ausgefüllt werden, die später das Select-Statement definieren werden. Wenn dieses Formular durch "Submit" bestätigt wird, wird die Variable „QUERY_STRING“ mit den Variablen und ihren Werten in der Form : var1=wert1&var2=wert2&var3=wert3 belegt und das Programm "WebQuery" wird mit dem String der Variablen „QUERY_STRING“ als Übergabeparameter gestartet. QUERY_STRING enthält die Namen des queryfiles x.query, welches die parametrisierten Select-Statements enthält, die Werte der Parameter mit ihren Namen und das Resultfile x.result, das die Definition für das Ausgabeformular und die Festlegung der Aufbereitung der Abfrageergebnisse enthält.

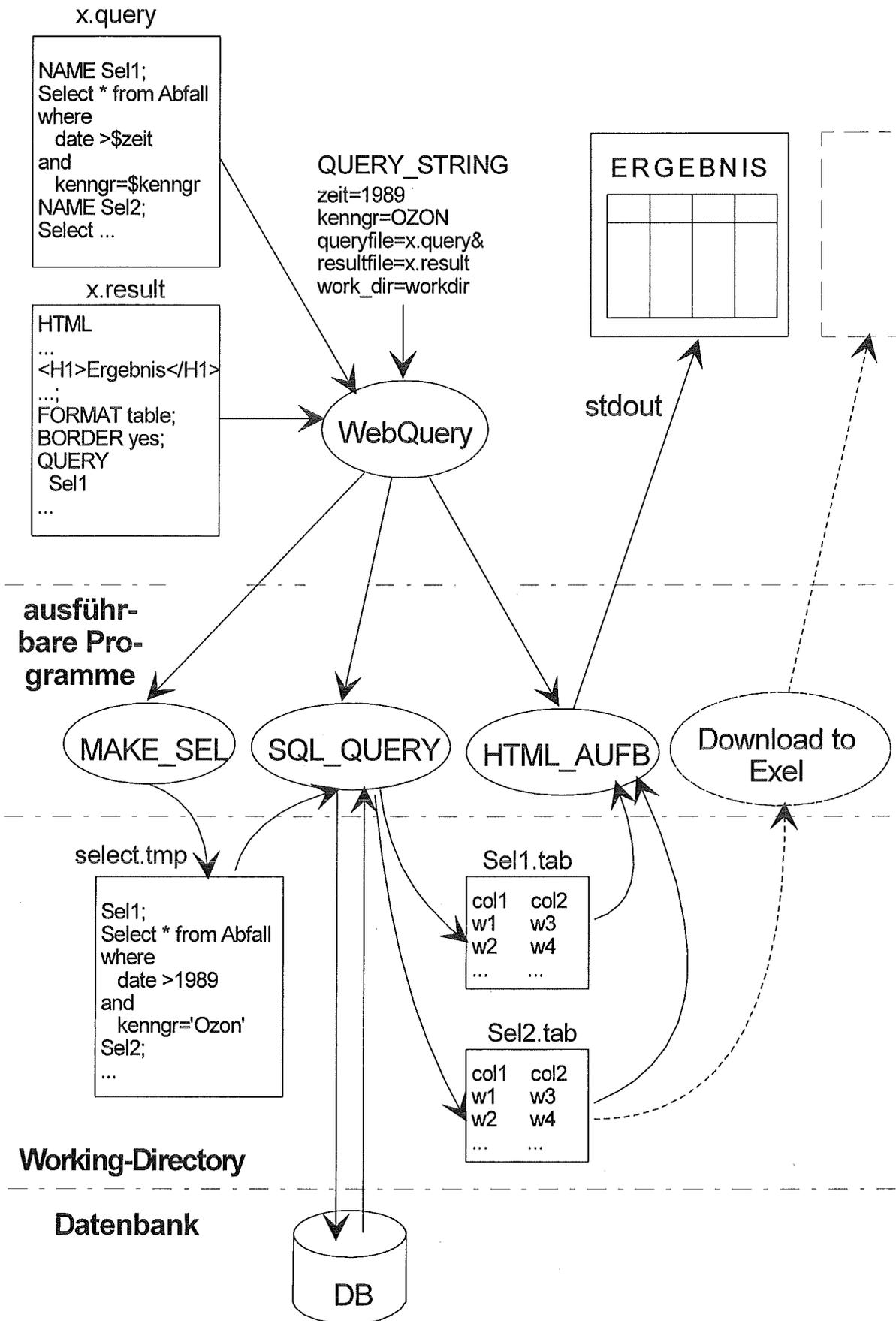


Abbildung 3-7: Ablauf von WebQuery

„Web_Query“ ruft das Executable „Make_Sel“ mit der QUERY_STRING-Variablen als Übergabeparameter auf. „Make_Sel“ zerlegt zuerst QUERY_STRING in einzelne Variablen mit Werten und speichert diese ab. Der Wert der Variablen „queryfile“ definiert den Namen der x.select-Datei, die die parametrisierten Select-Statements und deren Kennung enthält. Diese SQL-Abfragen und deren Kennung werden von „Make_Sel“ aus der x.select-Datei eingelesen und die Variablen (\$var\$) der Select-Strings werden durch die entsprechenden Werte der gleichnamigen Variablen des Übergabeparameters QUERY_STRING ersetzt. Diese ersetzten Select-Statements werden in das Workingdirectory (Wert der Variablen working_dir) in das File select.tmp zusammen mit ihren Kennungen eingetragen. „Web_Query“ ruft nun das Executable „Sql_Query“ ebenfalls mit QUERY_STRING als Übergabeparameter auf. Auch in diesem Modul wird zuerst die Variable QUERY_STRING zerlegt, wobei hier nur die Variable „work_dir“ mit ihrem Wert relevant ist. Die select.tmp-Datei in diesem work_dir-Verzeichnis wird eingelesen und die einzelnen Selects abgesetzt. Die Datenbankabfrageergebnisse der Selects werden im Directory „work_dir“ in Files mit den Namen „Kennung des Selects.tmpfile“ abgespeichert. „Web_Query“ ruft nun das Programm „Html_Aufb“ mit Parameter QUERY_STRING auf, das prinzipiell die SQL-Abfrageergebnisse der Datentöpfe als HTML-Seite aufbereitet und visualisiert. Hierzu parst es QUERY_STRING nach den Variablen „resultfile“ und „work_dir“. Die in der Variablen „resultfile“ angegebene Datei (x.result) wird nach ihren Schlüsselwörtern geparkt und die dazugehörigen Benutzerdefinitionen der Reihenfolge nach abgespeichert. Anschließend werden die unter dem Schlüsselwort QUERY gespeicherten Select-Statement-Kennungen durch die dazugehörigen .tab-Files mit dem working_dir als Pfad ersetzt. Somit befinden sich unter dem Schlüsselwort QUERY gespeichert alle Datentöpfe mit den Ergebnisdaten der entsprechenden Select-Statements. Nach dem vollständigen Parsen von x.result werden die unter den Schlüsselwort HTML abgelegten HTML-Definitionen auf Standardausgabe ausgegeben, bis an die Stelle, an der der Benutzer eine QUERY vorgesehen hat. Nun wird der erste unter QUERY abgelegte Datentopf geparkt und dessen Inhalt aufbereitet und entsprechend dem in x.result definierten Format auf Standardausgabe ausgegeben. Dieser Zyklus wiederholt sich, bis alle unter dem Schlüsselwort HTML abgespeicherten HTML-Definitionen ausgegeben und sämtliche Datentöpfe eingelesen und aufbereitet wurden.

Dieses Verfahren ist weitgehend unabhängig von der zugrundegelegten Anwendung (hier UFIS) und des verwendeten Datenbankmanagementsystems (hier ORACLE) und kann daher leicht auf andere Anwendungsfälle und Datenhaltungssysteme übertragen werden, indem das Modul Sql_Query ersetzt wird. Ebenso sind alle drei Module „Make_Sel“, „Sql_Query“ und „Html_Aufb“ voneinander unabhängig. Dem Modul Sql_Query kann somit jede andere Art der Aufbereitung folgen, da der vorherige Zyklus HTML-unabhängig ablaufen kann.

3.2 Sessionmanagement

Dieser Abschnitt beschreibt die am FAW entwickelte und implementierte Konzeption eines sogenannten Sessionmanagements für Dienste, die über WWW zur Verfügung gestellt werden. Dieses Sessionmanagement wird vom FAW in verschiedenen Forschungsvorhaben eingesetzt.

F. Schmidt definiert den Begriff *Dienst* im Kontext des UIS Baden-Württemberg als eine „Menge von Operationen an einer Schnittstelle“ (Dokumentation des UIS Workshop am 15.

5. 1995). Im WWW tritt ein Dienst als ein parametrisierbares Dienstprogramm (ein sogenanntes CGI-Skript) in Erscheinung.

Eine *Dienstanforderung* entspricht einem Aufruf eines solchen Dienstprogramms über das WWW und ist eindeutig identifiziert durch eine URL („Uniform Resource Locator“). Eine solche URL ist eine Zeichenkette, die im wesentlichen aus folgenden Informationen besteht: 1. einer netzweiten ID des Serverrechners, der den Dienst bereitstellt, 2. dem Namen des zugrundeliegenden Dienstprogramms sowie 3. einer Liste von Aktualparametern. Der Aufruf eines Dienstes erfolgt im WWW durch eine „Navigationsaktion“ zu einer derartigen URL. Der Server veranlaßt dann den Aufruf des zugeordneten Dienstprogramms mit den jeweiligen Aktualparametern. Das Dienstprogramm erzeugt als sichtbares Ergebnis einen Datenstrom, der die Definition einer Hypertextseite enthält, die schließlich dem Dienstenutzer angezeigt wird.

Mit Hilfe von WWW-Formularen (das sind besondere Hypertextdokumente) können Dienstanforderungen auf bequeme Weise über einen WWW-Browser (z.B. Mosaic) veranlaßt werden. Dies geschieht dadurch, daß die in dem Formular spezifizierten Angaben zu einer einzigen URL zusammengefügt werden. Durch Drücken einer Schaltfläche (Submit-Button) erfolgt die „Navigation“ zur spezifizierten URL, d.h. der Aufruf des betreffenden Dienstprogramms.

Viele Dienste im UIS benötigen für die ordnungsgemäße Ausführung einen *Kontext*. Bei einem solchen Kontext handelt es sich im weitesten Sinn um eine Menge von impliziten Parametern, die der Dienst lesend und verändernd verwendet, ohne daß diese vor oder nach dem Dienstaufruf explizit übergeben werden. Ein Beispiel für einen solchen Kontext ist ein Zwischenergebnis (z.B. ein „Datentopf“), das durch einen vorangegangenen Dienstaufruf erzeugt wurde (z.B. einem Selektor des Umweltführungsinformationssystems UFIS) und das durch einen nachfolgenden Dienstaufruf wieder genutzt wird (z.B. zur Präsentation).

Solche Kontexte müssen zumindest für die Dauer einer Session (also einer Sitzung an einem Server) aufrechterhalten werden. Unter einer *Session* soll hier eine zeitliche Abfolge von *Dienstanforderungen* (d.h. Zugriffe auf WWW-URLs) verstanden werden, die zwischen dem *Sessionstart* und dem *Sessionende* von einem Benutzer eines Clientsystems an ein Serversystem gerichtet werden.

Da WWW derzeit noch keine Werkzeuge bietet, um Sessions zu verwalten, wurde vom FAW ein geeignetes Sessionmanagement konzipiert und prototypisch realisiert. Hierbei können Gruppen von Diensten auf einem Server der Kontrolle des Sessionmanagements unterstellt werden. Das Sessionmanagement übernimmt die Aufgabe, Kontexte beim Start einer Session einzurichten und diese bei jeder nachfolgenden Dienstanforderung, die der Kontrolle des Sessionmanagements untersteht, wiederherzustellen und zu aktualisieren sowie zum Ende einer Session zu löschen und ggf. zuvor noch permanent zu sichern.

3.2.1 Sessionmanager

Der Sessionmanager ist eine anwendungsneutrale Komponente, um Sessions zu verwalten. Es können Gruppen von Diensten der Kontrolle des Sessionmanagers unterstellt werden. Das

Sessionmanagement übernimmt die Aufgabe, Kontexte beim Start einer Session einzurichten und diese bei jeder nachfolgenden Dienstanforderung, die der Kontrolle des Sessionmanagements untersteht, wiederherzustellen und zu aktualisieren.

3.2.1.1 Ablauf einer Session

Ein Dienst, der unter der Kontrolle des Sessionmanagements ablaufen soll, muß mittels des Sessionmanagers aufgerufen werden. Der Sessionmanager ist selbst ein CGI-Skript, dem u.a. der Name des aufzurufenden Dienstprogramms als Parameter übergeben wird. In aller Regel wird der Sessionmanager hierfür über ein WWW-Formular aufgerufen. Dieses Formular enthält die Aktualparameter für den aufzurufenden Dienst, darunter auch sogenannte versteckte Parameter, die für den Benutzer auf dem Formular nicht sichtbar sind.

Als einer dieser versteckten Parameter kann eine sogenannte Session-ID übergeben werden, die die Zugehörigkeit des Dienstaufrufs zu einer bestimmten Session festlegt. Wird der Sessionmanager ohne eine solche Session-ID aufgerufen oder ist die übergebene Session-ID ungültig, startet der Sessionmanager eine neue Session, sofern dies möglich ist, und vergibt hierfür eine neue Session-ID. Besteht eine Session noch, so wird der Sessionkontext aktualisiert und die Session fortgeführt. Über die aktiven Sessions führt der Sessionmanager in einer Verwaltungsdatei Buch.

Der Sessionmanager ruft anschließend den angewählten Dienst als Subprozeß auf. Der Dienst generiert als Ergebnis ein Hypertext-Dokument – in der Regel wiederum ein Formular für die nächste Interaktion – und liefert dieses an den Sessionmanager zurück. Der Sessionmanager fügt den Sessionkontext an das Dokument an und leitet dieses an den WWW-Browser weiter. Teile des Sessionkontexts werden in einer Fußzeile – dem sogenannten Session-Bar – im neuen Dokument sichtbar dargestellt, die Session-ID wird vom Session-Manager als versteckter Parameter in das Formular eingefügt.

Wird eine Session nach einer einstellbaren Zeit nicht mehr benützt, löscht der Sessiondaemon automatisch die Session. Dazu wird die Session aus der Verwaltungsdatei entfernt und das temporäre Arbeitsverzeichnis mit allen Dateien gelöscht. Außerdem kann künftig eine Session auch explizit durch den Benutzer beendet werden, indem dieser eine Schaltfläche auf dem Session-Bar betätigt; dieses Feature ist zum Zeitpunkt der Berichtserstellung noch in Vorbereitung.

Ein weiteres Feature, das derzeit in Vorbereitung ist, ist die Einrichtung eines *Init-Hooks* und eines *Exit-Hooks* für das Sessionmanagement. Mit Hilfe des Init-Hooks kann dem Sessionmanager ein Init-Programm bekannt gemacht werden, das als erste Aktion nach der Einrichtung einer Session mit dem aktuellen Sessionkontext aufgerufen wird. Dieses Programm kann erforderliche anwendungsspezifische Initialisierungen für die Session vornehmen. In analoger Weise kann über den Exit-Hook ein Exit-Programm bekannt gemacht werden, das als letzte Aktion vor dem Beenden einer Session aufgerufen wird und Aufräumarbeiten durchführen kann. Programmtechnisch sind Init-Hook und Exit-Hook lediglich zwei zusätzliche Parameter für den Sessionmanager, die die Pfadnamen der betreffenden Init- und Exit-Programme festlegen.

3.2.1.2 Benutzung des Sessionmanagers

Dem Sessionmanager müssen neben den Aktualparametern für den eigentlichen Dienstaufruf, die in Form von Attribut-Wert-Paaren vorliegen, noch folgende Parameter mitgegeben werden:

- 'service' - Name des Dienstes mit Pfad ab UIS_SERVICE_DIR.
- 'method' - Methode, mit der der Sessionmanager den Dienst aufruft. Wird keine Methode angegeben, ruft der Sessionmanager den Dienst mit der Methode 'GET' auf.

Außerdem wird die Session-ID, sofern eine solche mit den Aktualparametern übergeben wird, vom Sessionmanager ausgewertet.

Alle anderen Attribute werden mit ihren Werten an den aufgerufenen Dienst weitergereicht.

Sessionkontext

Der Sessionkontext besteht aus den folgenden Elementen:

- Session-ID, eine eindeutige Zeichenkette, die eine Session und damit auch den zugehörigen Sessionkontext auf einem Server für alle Zeiten eindeutig identifiziert. Die Eindeutigkeit der Session-ID wird durch Konkatenation der Systemzeit mit einer Zufallszahl gewährleistet.
- Benutzername, der Name, unter dem sich der Benutzer mittels des WWW-Zugriffsmechanismus angemeldet hat.
- Datum, der letzte Zeitpunkt, an dem der Sessionmanager auf die Session zugegriffen hat.

Konfiguration des Sessionmanagers

Es gibt mehrere Konfigurationsmöglichkeiten, um den Sessionmanager optimal auf die Bedürfnisse anzupassen. Seine Einstellungen liest der Sessionmanager aus der Konfigurationsdatei '.sessionmgr', die der Manager über die Environment-Variable 'UIS_CONFIG_DIR' lokalisiert. Folgende Steuermöglichkeiten gibt es:

- maximale Anzahl aller Sessions
- maximale Anzahl der Sessions für den Benutzer 'gast'
- maximale Anzahl der Sessions für den Benutzer 'user'
- das 'Wurzel-Verzeichnis' für alle Dienste
- das Verzeichnis, in dem der Sessionmanager liegt
- das temporäre 'Wurzel-Verzeichnis'
- das persistente 'Wurzel-Verzeichnis'

Fehlt eine dieser Variablen, bricht der Sessionmanager mit einer Fehlermeldung ab.

Da der Sessionmanager mittels einer Remote-Shell aufgerufen wird, ist es empfehlenswert, die Environment-Variable 'UIS_CONFIG_DIR' in einer Konfigurationsdatei, z.B. der '.cshrc' des Benutzers zu setzen. Die Variable muß dabei auf das Verzeichnis gesetzt werden, in dem die Konfigurationsdatei '.sessionmgr' steht. Dazu ist z.B. der Eintrag 'setenv UIS_CONFIG_DIR /users/lambach/integral/UIS_SERVICE_DIR' notwendig. Findet der Sessionmanager die Konfigurationsdatei nicht, bricht er mit einer entsprechenden Fehlermeldung ab.

Neben den Einstellungen für den Sessionmanager stehen in dieser Datei auch die Einstellungen für den Sessiondaemon, die im Kapitel 'Sessiondaemon' besprochen werden.

3.2.1.3 Zugangskontrolle zum Sessionmanager und den Diensten

Der Zugang zum Sessionmanager und den von ihm kontrollierten Diensten wird über die WWW-Benutzeridentifizierung geregelt. Das bedeutet, daß die Zugangskontrolle unabhängig vom Sessionmanager ist und einfach geändert werden kann, falls sich der WWW-Zugriffsschutz ändert.

Der WWW-Zugriffsschutz, der hier angewandt wird, schützt Verzeichnisse und Dateien, die über WWW adressiert werden. Die Dokumente in derart geschützten Verzeichnissen sind aber dadurch nicht automatisch gegen andere als WWW-Zugriffe geschützt. Deshalb kann auf die üblichen Zugriffsschutzmöglichkeiten des Betriebssystems nicht verzichtet werden.

Der Zugriffsschutz, der beim Sessionmanagement Anwendung findet, heißt 'By-Password Authentication'. Mit dieser Methode lassen sich Verzeichnisse, bzw. Dokumente, die in geschützten Verzeichnissen oder Unterverzeichnissen liegen, schützen. Der Zugriff kann für einzelne Benutzer oder für Benutzergruppen freigegeben werden.

Hat sich ein Benutzer einmal beim WWW-Browser eingeloggt, so gilt dieser Name so lange, wie der Browser läuft. Dies bedeutet daß der Benutzer nur einmal seinen Namen und sein Paßwort eingeben muß, unabhängig davon, wie oft er auf geschützte Dienste oder Dokumente zugreift. Weitere Details zur WWW-Zugangskontrolle finden sich in der Dokumentation des verwendeten WWW-Servers.

Zugangskontrolle zu Diensten

Da viele Benutzer das System benutzen können, wird der Zugriff über Benutzergruppen und nicht über Einzelpersonen geregelt. Jeder Benutzer wird beim Erstellen seines Accounts einer Gruppe zugeordnet. Dienste werden in Schutzgruppen aufgeteilt, und jedem Verzeichnis wird eine oder mehrere Gruppen zugewiesen. Diese Verzeichnisse sind damit für die angegebenen Benutzergruppen freigegeben. Dazu muß lediglich eine Datei mit dem Namen '.htaccess' in dem Verzeichnis erstellt werden. Möchte der Benutzer auf Dienste zugreifen, fordert der Browser einen Namen und ein Paßwort an. Ist beides korrekt, ruft der Browser den Dienst auf.

Zugangskontrolle zum Sessionmanager

Der Zugang zum Sessionmanager ist genau wie jeder andere Dienst über WWW-Mechanismen geschützt.

3.2.1.4 Einrichten eines neuen Dienstes

Alle UIS-Dienste sind in dem Verzeichnis 'SRV_CGI' bzw. in einem Unterverzeichnis lokalisiert. In diesem Verzeichnis gibt es Unterverzeichnisse, die die Funktion von sogenannten *Zugriffsgruppen* haben. Alle Dienste in einer Zugriffsgruppe können von den gleichen Benutzergruppen angesprochen werden.

Das Wurzelverzeichnis für alle Dienste, in der nachfolgenden Abbildung als Beispiel das Verzeichnis '/users/lambach/integral/integral_ges/integral_bin/SRV_CGI', ist in der Konfigurationsdatei '.sessionmgr' abgelegt. Soll einmal der Verzeichniszweig mit den Diensten verschoben werden, müssen nicht alle Diensteaufrufe geändert werden, sondern nur der Eintrag in der Konfigurationsdatei.

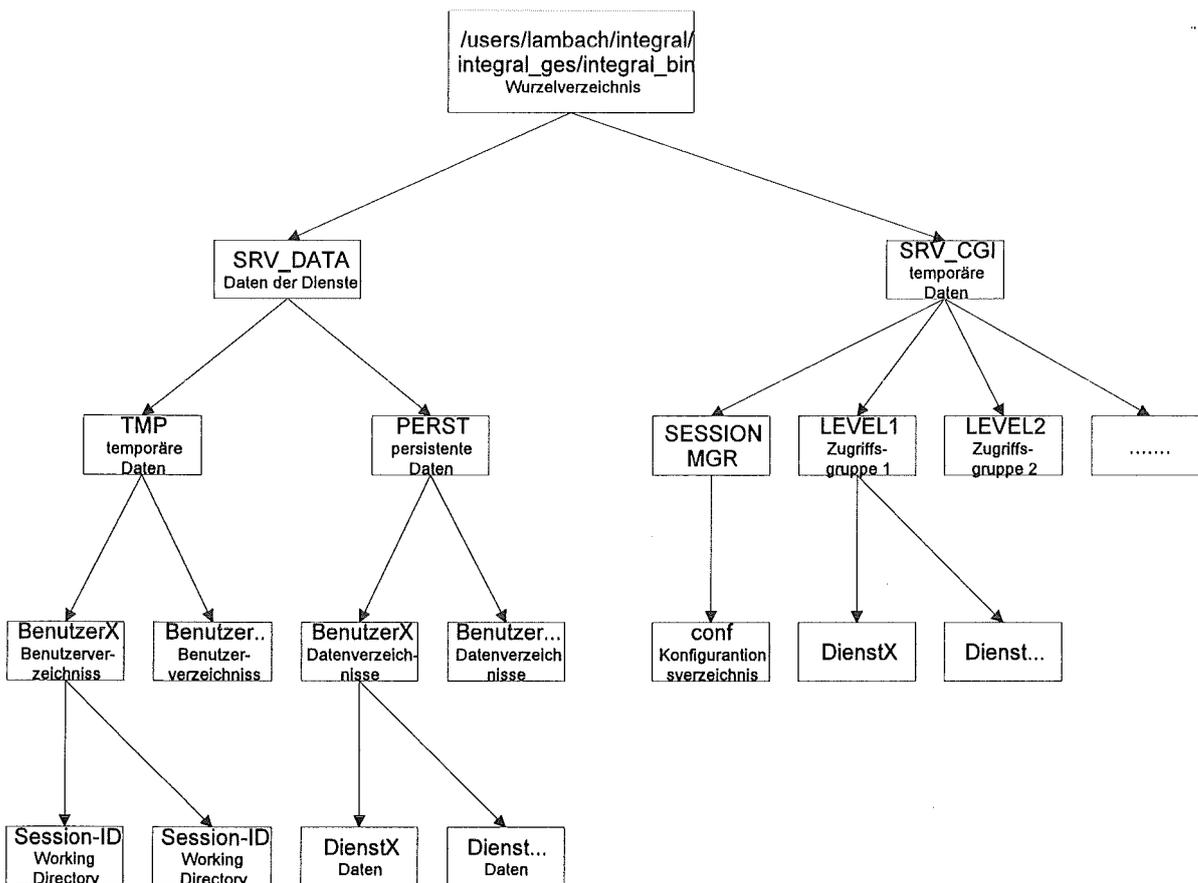


Abbildung 3-8: UIS Verzeichnisstruktur

(Anm.: Die Verzeichnisstruktur geht auf einen Vorschlag des FZI am 28.07.1995 zurück, s.a. FZI-Bericht 10/95, Studienarbeit von Hr. Kirsch, S. 14, Abb. 14)

Einrichten einer neuen Zugriffsgruppe

Soll eine neue Zugriffsgruppe erstellt werden, muß im Verzeichnis 'SRV_CGI' ein neues Unterverzeichnis erstellt werden. Danach kopiert man die Datei '.htaccess' aus einem anderen Zugriffsverzeichnis, z.B. von SESSIONMGR, in dieses neue Verzeichnis. Nun muß nur noch die Datei angepaßt werden.

Die Datei '.htaccess' im Verzeichnis SESSIONMGR sieht beispielsweise folgend aus:

```
AuthUserFile /usr/lambach/integral/integral_ges/integral_bin/SRV_CGI/SESSIONMGR/conf/.htpasswd
AuthGroupFile /usr/lambach/integral/integral_ges/integral_bin/SRV_CGI/SESSIONMGR/conf/.htpasswd
AuthName ByPassword
AuthType Basic
<Limit GET>
require group all
</LIMIT>
```

Zugriff auf das Verzeichnis haben alle Benutzer, die der Gruppe 'all' angehören. Diese Gruppe muß auf die gewünschte Gruppe abgeändert werden. Es kann auch mehreren Gruppen der Zugriff erlaubt werden.

Dienst hinzufügen

Wird für den Dienst eine neue Zugriffsgruppe benötigt, muß diese zuerst, wie im vorigen Kapitel beschrieben, erstellt werden. Danach wird in der gewünschten Zugriffsgruppe, z.B. LEVEL1, lediglich ein Unterverzeichnis erstellt. In dieses Unterverzeichnis wird dann der Dienst hineingestellt. Zugriff auf den Dienst erhalten alle Benutzer, die Zugriff auf LEVEL1 haben.

3.2.2 Sessiondaemon

Sessions müssen vom Benutzer, selbst bei Beendigung des Browsers, nicht beendet werden. Daher kann die Anzahl der aktiven Sessions sehr schnell anwachsen. So könnte ein einzelner Benutzer schnell zu mehreren hundert aktiven Sessions kommen. Für jede Session existiert ein Arbeitsverzeichnis und ein Eintrag in einer Verwaltungsdatei. Damit würden nach kurzer Zeit enorme Systemressourcen gebunden und das Sessionmanagement wäre nicht mehr handhabbar. Daher müssen alle Sessions nach einer Zeitspanne, nach der sie nicht mehr benutzt werden, gelöscht werden.

Der Sessiondaemon ist ein Programm, das Sessions, die länger als eine konfigurierbare Zeitspanne laufen, löscht. Die maximale Zeit, die Sessions aktiv sein können, sowie das Zeitintervall, nach dessen Ablauf der Sessionmanager periodisch alle Sessions überprüft, ist in der Konfigurationsdatei '.sessionmgr' abgelegt.

In der Sessionverwaltungsdatei wird zu jeder Session der Zeitpunkt des letzten Zugriffs mitgeführt. Der Sessionmanager überprüft alle Zeitstempel in dieser Datei und löscht eine Session, falls sie zu lange nicht angesprochen wurde. Dabei entfernt der Sessiondaemon den

Eintrag aus der Verwaltungsdatei und löscht das temporäre Arbeitsverzeichnis der Session, ohne eventuell vorhandene Daten zu sichern.

Hat der Sessiondaemon alle Einträge in der Verwaltungsdatei überprüft, legt er sich eine konfigurierbare Zeit schlafen. Danach startet er den nächsten Überprüfungslauf.

Werden Einstellungen des Dämons in der Konfigurationsdatei geändert, muß der Sessiondaemon neu gestartet werden, damit diese Änderungen wirksam werden können.

3.2.3 Dokumentation der Programme

Die Funktionen sind in den Quelldateien ausführlich kommentiert. Den kompletten Ablauf der Programme natürlichsprachlich zu beschreiben, würde eher zur Verwirrung führen. Daher ist der Ablauf der Programme jeweils mit einem Diagramm beschrieben.

3.2.3.1 Übersetzung der Programme

Die Übersetzung der Programme wird über ein gemeinsames Makefile gesteuert. Es kann jedoch ein einzelnes Programm, z.B. der Sessionmanager mit 'make sessionmgr', übersetzt werden. Desweiteren benützen die Programme eine gemeinsame Header-Datei, in der verschiedene Konstanten definiert sind.

Die Programme *sessionmanager* und *newuser* benutzen überwiegend die Funktionen, die in *util.c*, *html.c* und *parse.c* codiert sind. Das Programm *sessiondaemon* wird nicht mit anderen Dateien zusammengelinkt. Alle Funktionen, die benötigt werden, befinden sich in der Datei *newuser.c*.

3.2.3.2 Aufbau von sessionmanager

Der Sessionmanager ist das zentrale Element des UIS-Sessionmanagements. Er regelt die Zugriffe auf UIS Dienste.

Relevante Dateien:

- sessionmgr.c
- sessionmgr.h
- util.c
- html.c
- parse.c

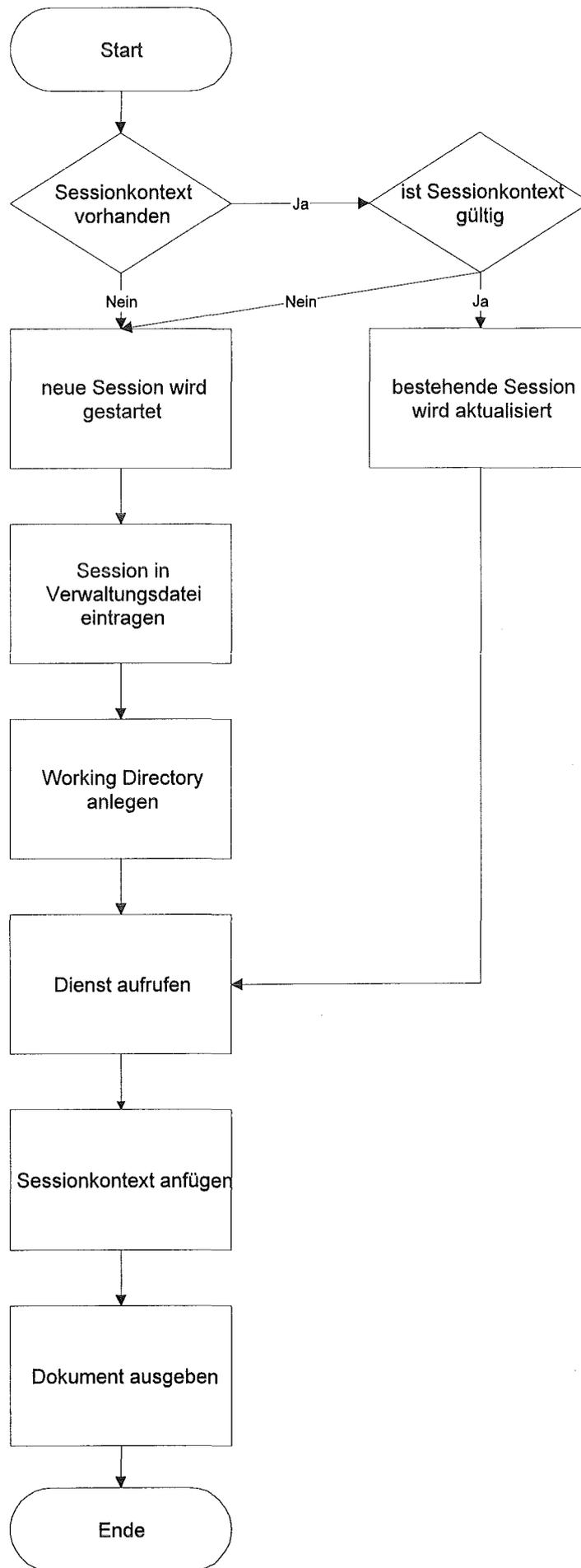


Abbildung 3-9: Ablaufplan Sessionmanager

3.2.3.3 Aufbau von *newuser*

Das Programm *newuser* setzt einen neuen Benutzer auf oder fügt den Benutzernamen anderen Benutzergruppen hinzu.

Achtung:

Das Paßwort wird immer auf das eingegebene gesetzt, auch wenn bereits ein Paßwort für den Benutzer existiert.

Relevante Dateien:

- *newuser.c*
- *sessionmgr.h*
- *util.c*
- *html.c*
- *parse.c*
- *htpasswd.c*

Hinweis:

Die Datei '*htpasswd.c*' ist dem NCSA Browser Mosaic entnommen. Mit dem ursprünglichen '*htpasswd*'-Programm lassen sich Einträge in einer Paßwortdatei für den WWW-Zugriffsschutz generieren. Funktionen aus dieser Datei wurden für die Bedürfnisse des Programms *newuser* angepaßt; mit ihrer Hilfe werden die Paßworteinträge erzeugt.

Erweiterungen:

Es ist denkbar, das Programm *newuser* dahingehend zu ändern, daß, falls ein Paßwort angegeben wurde und der Benutzer bereits einer Benutzergruppe angehört, das Paßwort nicht zurückgesetzt wird. Allerdings muß dann ein Mechanismus geschaffen werden, das Paßwort auf ein anderes zurückzusetzen.

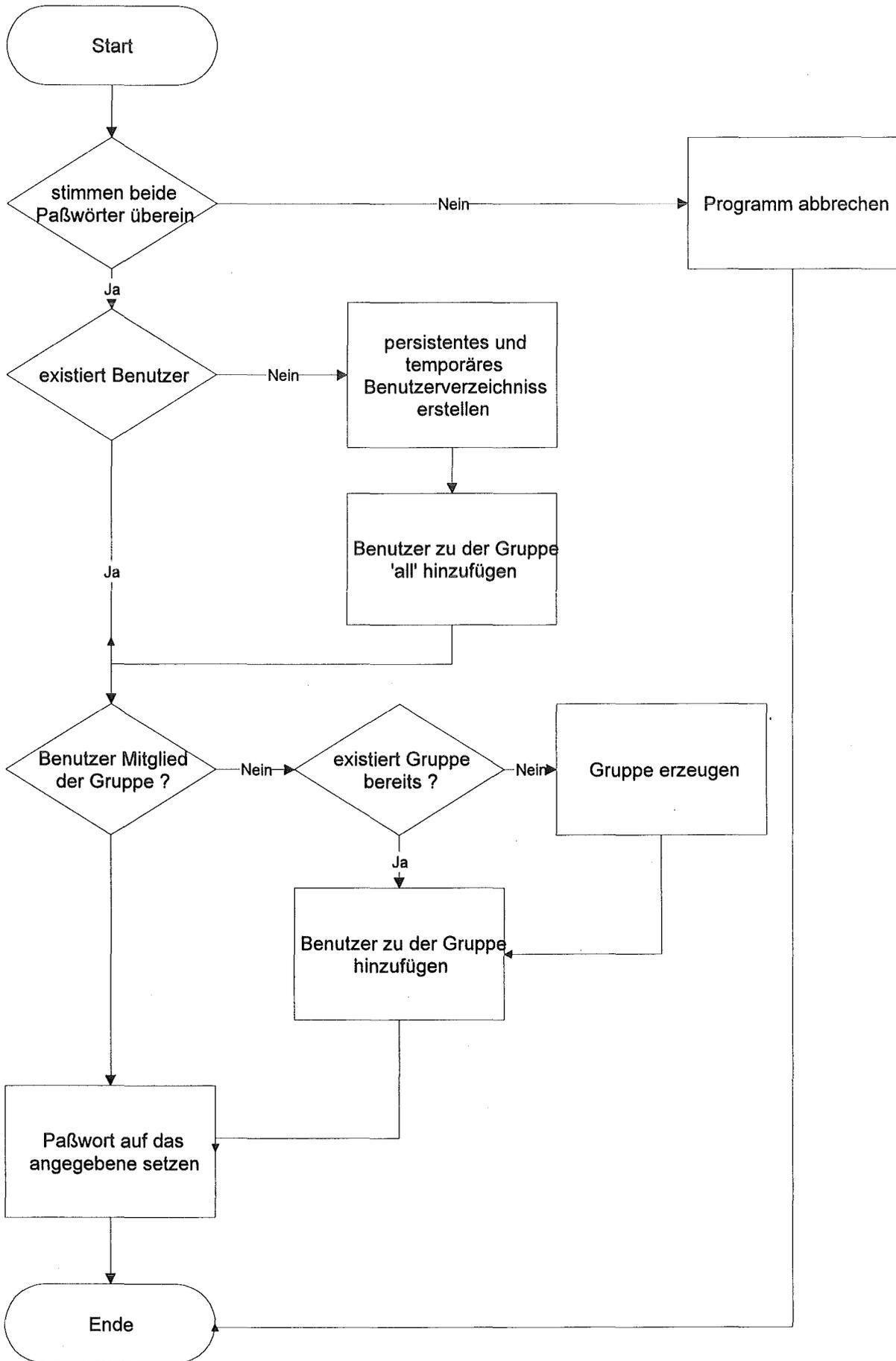


Abbildung 3-10: Ablaufplan newuser

3.2.3.4 Aufbau von sessiondaemon

Relevante Dateien:

- sessionmgr.c
- sessionmgr.h

Programmaufbau:

Um den Sessiondaemon möglichst schlank zu halten, sind alle Funktionen in der Datei sessiondaemon.c untergebracht. Mehrere Funktionen ähneln Funktionen aus anderen Modulen stark. Jedoch kann eine eventuelle Fehlerausgabe, wie in newuser oder sessionmgr, nicht über ein HTML-Dokument erfolgen.

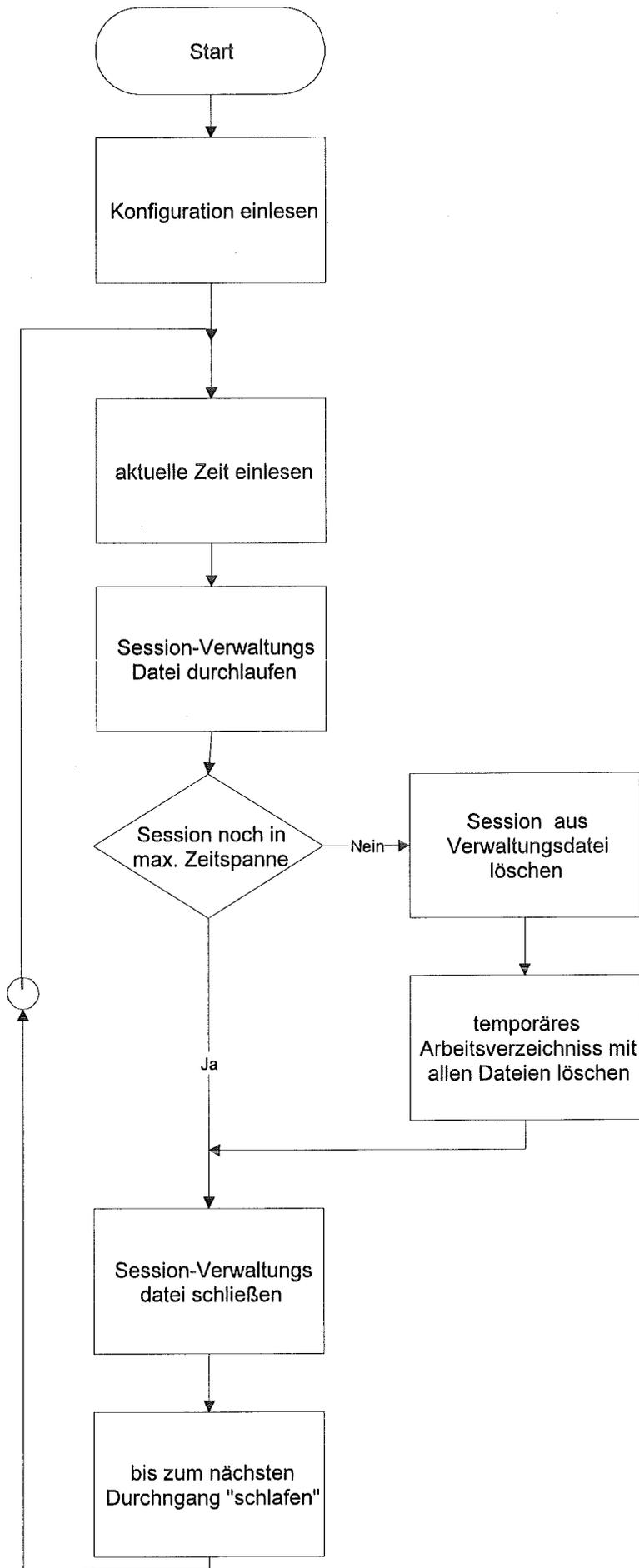


Abbildung 3-11: Ablaufplan sessiondaemon

Literatur

F. J. Radermacher. The Importance of Metaknowledge for Environmental Information Systems. In: O. Günther, H.-J. Schek (Hrsg.). *Advances in Spatial Databases*, Lecture Notes in Computer Science 525, Springer-Verlag, Berlin [u.a.], 1991.

A. Jaeschke, A. Keitel, R. Mayer-Föll, F. J. Radermacher, J. Seggelke. Metawissen als Teil von Umweltinformationssystemen. In: O. Günther, H. Kuhn, R. Mayer-Föll, F. J. Radermacher (Hrsg.). *Konzeption und Einsatz von Umweltinformationssystemen*, Informatik-Fachberichte 301, Springer-Verlag, Berlin [u.a.], S. 115 – 130, 1992.

O. Günther, I. Henning, W.-F. Riekert, F. Schmidt. Ein Schichtenmodell zur Integration heterogener Umweltinformationssysteme. In: R. Denzer, W. Geiger, R. Güttler (Hrsg.): *1. Workshop „Integration von Umweltdaten“*, Kernforschungszentrum Karlsruhe, Bericht KfK-5187, S. 32 – 37, August 1993.

W.-F. Riekert, I. Henning, F. Schmidt. Integration von heterogenen Komponenten des Umweltinformationssystems (UIS) Baden-Württemberg. In: R. Güttler, W. Geiger (Hrsg.): *2. Workshop „Integration von Umweltdaten“*, Metropolis-Verlag, Marburg, 1994.

K. Blank, W.-F. Riekert, F. Schmidt, M. Tischendorf: *Projekt Integral: Integration von heterogenen Komponenten des Umweltinformationssystems (UIS) Baden-Württemberg*. Abschlußbericht der Phase I. FAW Ulm und IKE Universität Stuttgart, März 1994.

K. Blank, W.-F. Riekert, Th. Krist, R. Münster, F. Schmidt, M. Tischendorf, K. Kübler: *Projekt Integral: Integration von heterogenen Komponenten des Umweltinformationssystems (UIS) Baden-Württemberg*. Abschlußbericht der Phase II. FAW Ulm und IKE Universität Stuttgart, Oktober 1994.

Institut für Photogrammetrie und Fernerkundung der Universität Karlsruhe, Institut für Kernenergetik und Energiesysteme der Universität Stuttgart, Forschungsinstitut für anwendungsorientierte Wissensverarbeitung an der Universität Ulm und Forschungszentrum Informatik an der Universität Karlsruhe: *GLOBUS – Konzeption und Realisierung einer aktiven Auskunfts-komponente für globale Sachdaten im Umweltinformationssystem Baden-Württemberg*. Abschlußbericht. Dezember 1994.

W.-F. Riekert, J. Ebbinghaus: Geodatenserver in Wide Area Networks. In: *Geographische Informationssysteme '95*. Tagungsband. Wiesbaden, 1995.

M. Gaul: *Thesaurusgestützter Zugriff zu Umweltberichten in einem netzübergreifenden Hypertextsystem*. Forschungsbericht FAW-TR-95006, Ulm, 1995.

W.-F. Riekert: Cooperative Management of Data and Services for Environmental Applications. In F. Huber-Wäschle, H. Schauer, P. Widmayer (Hrsg.): *GISI 95 –*

Herausforderungen eines globalen Informationsverbundes für die Informatik. Informatik aktuell, Springer. Berlin – Heidelberg – New York, 1995

W.-F. Riekert: Thesaurusgestützter Zugriff zu Umweltinformationen in einem netzübergreifenden Hypertextsystem. In: 3. *Workshop „Integration von Umweltdaten“*, Metropolis-Verlag, Marburg. In Vorbereitung.

J. Ebbinghaus, M. Gaul, G. Wiest: Metainformation für den Zugang zu Umweltdaten in Globalen Netzen. In: H. Kremers (Hrsg.): *Umweltdaten verstehen durch Metainformation. Praxis der Umweltinformatik*, Band 6. Metropolis-Verlag, Marburg. In Vorbereitung.

Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW) an der Universität Ulm und Institut für Kernenergetik und Energiesysteme (IKE) der Universität Stuttgart: *Integration heterogener Komponenten des Umweltinformationssystems Baden Württemberg (INTEGRAL), Phase III.* Abschlußbericht, in Vorbereitung.

Nutzung von Diensten im WWW mit dem Common Client Interface (CCI)

*R. Kopetzky; A. Lurk; F. Schmidt
Universität Stuttgart,
Institut für Kernenergetik und Energiesysteme (IKE)
Pfaffenwaldring 31
D-70569 Stuttgart*

KURZFASSUNG	195
1 EINLEITUNG	195
2 DAS COMMON CLIENT INTERFACE (CCI).....	196
2.1 SPEZIFIKATION DES CCI-PROTOKOLLS	196
2.1.1 <i>Datenübertragung auf unterschiedlichen Systemen.....</i>	<i>196</i>
2.1.2 <i>Funktionalitäten des CCI-Protokolls</i>	<i>197</i>
2.2 ANWENDUNGEN DES CCI-PROTOKOLLS	200
2.2.1 <i>CCI unter Windows</i>	<i>200</i>
2.2.1.1 <i>Kommunikation unter Windows-Applikationen</i>	<i>200</i>
2.2.1.2 <i>Das "OLE Automation Interface".....</i>	<i>202</i>
2.2.2 <i>CCI über TCP/IP.....</i>	<i>207</i>
2.2.2.1 <i>Das Application Programmers Interface (API)</i>	<i>207</i>
2.2.2.2 <i>Tel/Tk und CCI</i>	<i>212</i>
2.2.2.3 <i>C-API oder CCI-Shell</i>	<i>214</i>
3 PRAKTISCHE ANWENDUNGEN DER CCI-SCHNITTSTELLE.....	215
3.1 ANWENDUNGSBEREICHE	215
3.2 UMSETZUNG ÜBER OLE UND TCP/IP	216
3.2.1 <i>Nutzung des OLE Application Interface.....</i>	<i>217</i>
3.2.2 <i>Nutzung von CCI über TCP/IP.....</i>	<i>218</i>
3.2.3 <i>Beispiel: Simulation atmosphärischer Schadstoffausbreitung (CRAYSIM)</i>	<i>219</i>
3.2.3.1 <i>Aufgabenstellung</i>	<i>220</i>
3.2.3.2 <i>Realisierung über CCI.....</i>	<i>222</i>
3.2.3.3 <i>Probleme und Lösungsansätze</i>	<i>223</i>
3.2.4 <i>Beispiel: Inanspruchnahme von Diensten mit dem Kommunikationsinterpreter (KIP) über das WWW.....</i>	<i>223</i>
3.2.4.1 <i>Hauptmenü</i>	<i>224</i>
3.2.4.2 <i>Dienstanforderung</i>	<i>225</i>
3.2.4.3 <i>Dienstinanspruchnahme</i>	<i>226</i>
3.2.4.4 <i>Ergebnisanforderung</i>	<i>230</i>
3.2.4.5 <i>Statusanfrage.....</i>	<i>231</i>
3.2.4.6 <i>Übersicht über aktive Dienstaufträge</i>	<i>231</i>
3.2.4.7 <i>Löschen von Dienstaufträgen.....</i>	<i>232</i>
3.2.4.8 <i>Fehlermeldungen des Kommunikationsinterpreters</i>	<i>232</i>
3.3 KOOPERATIVES ARBEITEN MIT WINDOWS-MOSAIC	233
4 ZUSAMMENFASSUNG	236
LITERATUR.....	237

Kurzfassung

Das World-Wide-Web gewinnt als Informationssystem durch die Ausweitung des Angebots an Informationen immer mehr an Bedeutung. Um Zugang zu diesen Informationen zu erhalten, stehen dem Anwender eine Auswahl von Browsern zur Verfügung. Einer dieser Browser ist Mosaic, der am National Center for Supercomputer Applications (NCSA) entwickelt wird.

Dieser Bericht untersucht die neue Common Client Interface-Schnittstelle (CCI), die erst seit kurzer Zeit in Mosaic implementiert ist. Hierbei wurde besonderer Wert auf die Darstellung der Unterschiede zwischen den Versionen für MS-Windows und X-Window sowie den möglichen Anwendungsbereiche beider Versionen gelegt.

1 Einleitung

Im Internet, einem weltumspannenden, offenen Verbund von lokalen, regionalen und überregionalen Rechnernetzen, werden zahlreiche Informationssysteme und Dienste angeboten. Neben WAIS, FTP oder GOPHER gewinnt dabei das World-Wide-Web (kurz WWW oder 3W) immer mehr an Bedeutung.

Das WWW ist ein Informationssystem, das zur Erschließung der im Internet erhältlichen Ressourcen (bzw. Informationen) eingesetzt wird. Die Ressourcen können dabei beispielsweise Texte, Daten oder auch Programme sein, die Informationen aus den unterschiedlichsten Bereichen bieten und den Zugang zu anderen Internetdiensten wie z.B. FTP oder ARCHIE erlauben.

WWW besteht aus einer Vielzahl von Web-Clients und Web-Servern, die sich, da das Internet ein verteiltes System darstellt, auf verschiedenen Rechnern innerhalb des Internet befinden können. Die Web-Clients, auch Web-Browser genannt, bilden den Ausgangspunkt, um auf sämtliche Ressourcen, die von Web-Servern angeboten werden, zuzugreifen.

Mit SGML (Standard Generalized Markup Language) entstand 1988 ein Standard zur Definition plattformunabhängiger Dokumenttypen. Um einen solchen handelt es sich bei der "Hypertext-Markup-Language" (HTML). Als SGML-Sprache dient HTML der Gestaltung von Strukturelementen wie Kapitel, Überschriften und Aufzählungen. Sie ist also für die "Document Type Definition" (DTD) zuständig, wobei die Umsetzung der Dokumentenstruktur in ein Layout von dem jeweiligen Browser übernommen wird. HTML bietet eine stetig wachsende Anzahl von Befehlen, die es dem Browser ermöglichen, Text, Graphik und Hypertext-Links zu anderen Dokumenten oder Diensten darzustellen. [1]

Bei den Dokumenten wird innerhalb des World Wide Web zwischen statischen und dynamischen HTML-Dokumenten unterschieden. Statische Dokumente werden einmal erstellt und in der Regel nicht mehr verändert. Dynamische Dokumente hingegen werden erst zum Zeitpunkt einer Client-Anfrage von einem Programm oder Skript ("Gateway"-Programm) dynamisch erzeugt.

Der gebräuchlichste Web-Browser ist Mosaic, ein Tool, das durch seine graphische Benutzeroberfläche sehr benutzerfreundlich und einfach zu bedienen ist. Es wurde am National Center for Supercomputer Applications (NCSA) entwickelt und wird für die meisten gängigen Betriebssysteme angeboten. Seit kurzer Zeit ist in NCSA Mosaic (ab v2.5 für X und v2.0b4 für Windows) eine neue Schnittstelle, das "Common Client Interface" (CCI), integriert. Ergänzend zu dem "Common Gateway Interface" (CGI), das die Kommunikation zwischen HTTP-Server und "Gateway"-Programm ermöglicht, wurde CCI für die Kommunikation zwischen externen Anwendungen und Mosaic entwickelt.

Diese Arbeit beschäftigt sich mit der Spezifikation dieser beiden Schnittstellen und zeigt auf, welche Möglichkeiten die derzeit verfügbaren Funktionalitäten speziell im Bereich der Kommunikation zwischen Mosaic und Anwendungsprogrammen bieten.

2 Das Common Client Interface (CCI)

Während CGI über "Gateway"-Programme den Zugriff auf Informationen ermöglicht, die der Browser in ihrem ursprünglichen Format nicht darstellen kann, erlaubt das Common Client Interface (CCI) externen Anwendungen, auf Informationen und Funktionalitäten aus dem WWW zuzugreifen.

2.1 Spezifikation des CCI-Protokolls

Das Mosaic CCI definiert ein Protokoll im ASCII-Format für den Einsatz zwischen Mosaic und externen Applikationen, das den Anforderungen nach RFC 822 entspricht. Nahezu beliebige Anwendungsprogramme werden dadurch in die Lage versetzt, Mosaic zu kontrollieren und Informationen aus dem WWW zu erhalten. Basierend auf diesem Protokoll können Funktionsbibliotheken und Skriptsprachen entwickelt werden, mit deren Hilfe Anwendungen realisiert werden können.

2.1.1 Datenübertragung auf unterschiedlichen Systemen

Für die beiden gängigsten Window-Systeme wurden am NCSA zwei unterschiedliche Versionen von NCSA Mosaic entwickelt. Diese unterscheiden sich hinsichtlich der CCI-Schnittstelle vor allem in den jeweils für das CCI-Protokoll verfügbaren Datentransport-Mechanismen.:

Bei **X-Window Systemen** erfolgt der Datenaustausch ausschließlich über TCP/IP, was sowohl den lokalen, als auch den räumlich getrennten Einsatz von Mosaic und CCI-Applikation gestattet. NCSA Mosaic für X-Window steht in der derzeitigen Version 2.7b2 auf dem FTP-Server von NCSA für folgende Plattformen zur Verfügung:

Rechner:	Betriebssystem:
DEC Alpha	OSF1 3.0
DEC 3100	Ultrix 4.0
HP 9000/735	HP-UX 9.01
IBM RS/6000	AIX 3.2
SGI Indy	IRIX 5.2
SGI Indigo	IRIX 4.0.5
Pentium	Linux 1.1.94
Sun	Solaris 2.3 / 2.4 und SunOS 4.1.3

Unter **MS Windows 3.1 (bzw. Windows für Workgroups 3.11)** besteht neben dem Datenaustausch über TCP/IP die Möglichkeit des lokalen Datenaustausches über "Dynamik Data Exchange" (DDE) oder "Object Linking and Embedding" (OLE). Damit ist mit Windows-Mosaic 2.0 sowohl der lokale Einsatz von CCI-Applikationen (OLE und TCP/IP) als auch der entfernte Zugriff über TCP/IP möglich.

Unabhängig von der Plattform werden die CCI-Daten gemäß der MIME 1.0 - Spezifikation, definiert in RFC 1521, übertragen. Bei den "Multipurpose Internet Mail Extensions" (MIME) handelt es sich um eine Spezifikation der Formate von im Internet gebräuchlichen Message-Bodies.

Beispiele für MIME-Typen, die Mosaic ohne weitere Definitionen erkennt, sind:

text/html	- HTML-Dateien
audio/*	- alle in MIME festgelegten Ton- und Klangformate
image/*	- alle in MIME festgelegten Bildformate
application/postscript	- Postscript-Dateien

2.1.2 Funktionalitäten des CCI-Protokolls

Für den Umgang mit diesen Daten stellt die CCI-Schnittstelle, wie in Abbildung 2.1 gezeigt, eine Reihe von Funktionalitäten bereit:

- GET - Anfordern einer URL
- DISPLAY - Darstellen von Informationen im Browser
- POST - Daten an Server schicken
- SEND - Registrierung eines Ports für Output vom Browser
- DISCONNECT - Verbindungsabbruch der Anwendung
- QUIT - Beenden des Browsers

Jeder Zugriff auf den Browser liefert einen numerischen Code, gefolgt von einem Text, an den Client zurück, der die Anforderung geschickt hat (ausführlicher Code siehe Anhang A.1):

1xx	- Informative Nachricht
2xx	- Kommando OK
3xx	- Kommando OK; zusätzlicher OUTPUT folgt
4xx	- Client Error; (z.B. Request hat fehlerhafte Syntax)
5xx	- Server Error; (z.B. Befehl kann nicht ausgeführt)

werden)

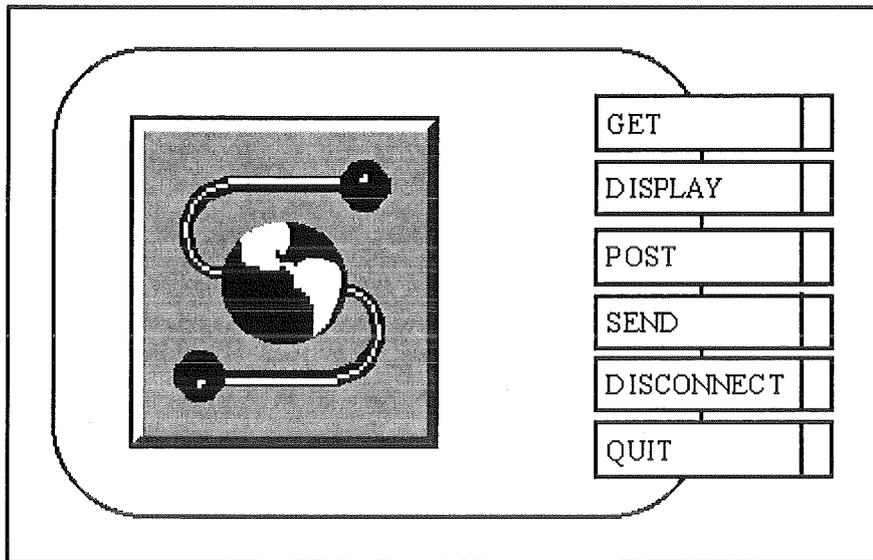


Abbildung 2.1 - Funktionalitäten des CCI-Protokolls

Da es sich bei CCI um eine sehr neue Schnittstelle handelt, unterliegt die Spezifikation ständigen Änderungen und Erweiterungen. In der Fassung vom 2. April 1995 sind die Funktionalitäten wie folgt festgelegt:

- GET
GET {URL|URN} <url> [OUTPUT {CURRENT|NEW|NONE}]
GET {URL|URN} <url> [OUTPUT {CURRENT|NEW|NONE}] HEADER
Content-Length: (length of additional header)
...additional header...
Das GET URL - Kommando veranlaßt den Web-Browser, die gegebene URL anzufordern. Das Ergebnis wird wahlweise im aktuellen (CURRENT) oder einem neuen Fenster (NEW) dargestellt. Voreingestellt ist dabei das aktuelle Fenster des Browsers.
Wurde OUTPUT NONE gewählt, so wird das Dokument im Browser nicht dargestellt. Über SEND besteht die Möglichkeit, das Dokument in der CCI-Client-Applikation selbst zu bearbeiten oder an eine dem MIME-Typ entsprechende Anwendung (z.B. Konverter, Betrachter, ...) weiterzuleiten. In diesen Fällen wird der Browser zugleich als Web-Client und CCI-Server eingesetzt.
- DISPLAY
DISPLAY [OUTPUT {CURRENT|NEW}]
Content-Type: data type
Content-Length: (length of data body)
...data body...
Das DISPLAY - Kommando weist den Browser an, die folgenden Daten darzustellen. Der Datenbereich ist, getrennt durch einen <CR><LF>, in die Information über die Länge des folgenden Datenabschnittes und den eigentlichen Datenbereich eingeteilt. Die optionale Verwendung von OUTPUT entspricht der bei dem Kommando GET.

- POST
POST <url> mimeType [OUTPUT {CURRENT|NEW|NONE}]
Content-Length: (length of MIME Body)
...MIME body...
Das Kommando POST veranlaßt den Browser, den folgenden MIME-Body an den in der URL bezeichneten HTTP-Server zu schicken.
Der OUTPUT-Parameter gibt an, ob der OUTPUT des HTTP-Server im aktuellen oder in einem neuen Fenster dargestellt bzw. bei NONE über den CCI-Port zurückgeschickt werden soll.
Dabei geschieht die Datenrückgabe in folgender Form:
304 POST output
Content-Length: (length of data body)
...data body...

Das Kommando SEND wird verwendet, um eine Port-Adresse für die vom Browser gesendeten Informationen zu registrieren:

- SEND ANCHOR
SEND ANCHOR {BEFORE|AFTER|STOP}
Diese Anweisung registriert eine Adresse, an die aktivierte URL-Anchors geschickt werden.
Form der Datenrückgabe:
301 ANCHOR <url>
Die Adresse bleibt so lange aktiv, bis die Anweisung SEND ANCHOR STOP gesendet wird.
- SEND OUTPUT
SEND OUTPUT [STOP] mimeType
Dieses Kommando veranlaßt den Browser, den spezifizierten MIME-Typ zu schicken. Der Empfänger ist dann für die Darstellung verantwortlich.
Form der Rückgabe:
302 Viewer output
Content-Type: mimeType
Content-Length: (length of data body)
...data body...
- SEND BROWSERVIEW
SEND BROWSERVIEW [STOP]
Damit wird der Browser angewiesen, eine Kopie des gerade dargestellten Dokuments an die aktive CCI-Applikation zu schicken. Dies beinhaltet sowohl das text/html-Dokument als auch zugeordnete Graphiken, die so weitergegeben werden, wie sie entweder von einem Server oder dem Browser-Cache empfangen werden.
Form der Rückgabe:
303 url
Content-Type: mimeType
Content-Length: (length of data body)
...data body...
- DISCONNECT
Damit wird der Browser "davon unterrichtet", daß die Verbindung zu der Applikation abgebrochen wird.

- **QUIT**
Dieser Befehl veranlaßt die Beendigung des Browsers. Das kann nötig werden, wenn der Browser als CCI-Server fungiert und nach der Beendigung des CCI-Client nicht mehr benötigt wird.
Der Browser gibt
213 QUIT request received exiting...
zurück und wird beendet.

2.2 Anwendungen des CCI-Protokolls

Wie in 2.1.1 beschrieben, kann der Zugriff auf CCI entweder lokal oder verteilt erfolgen. Der verteilte Zugriff basiert auf TCP/IP und wurde in X-Mosaic (ab der Version 2.5) und in Windows-Mosaic (ab der Version 2.0) realisiert. Dabei können CCI-Anwendung und Browser sowohl auf lokalen Rechnern wie auch auf örtlich entfernten Rechnern installiert sein. In Windows-Mosaic (ab der Version 2.0b4) wurde zusätzlich der lokale Zugriff über OLE 2.0 realisiert.

2.2.1 CCI unter Windows

Bei der Entwicklung der CCI-Schnittstelle in Windows-Mosaic wurde am NCSA darauf geachtet, die von Microsoft zur Verfügung gestellten Kommunikationsmöglichkeiten zu unterstützen. Der weitestgehende Mechanismus ist dabei das OLE 2.0, das im folgenden Abschnitt kurz erläutert wird. Im Anschluß daran folgt eine Einführung in das in Windows-Mosaic implementierte "OLE Automation Interface".

2.2.1.1 Kommunikation unter Windows-Applikationen

Unter Windows können Programme innerhalb eines Netzwerks als Teil eines größeren, verbundenen Systems ablaufen [2]. Die Daten verschiedener Programme können damit Teil einer Anwendung sein. Diese Fähigkeit zur Zusammenarbeit wird durch unterschiedliche Mechanismen erzielt:

- Die Zwischenablage (Clipboard)
- gemeinsame Speicherbereiche (Shared Memory)
- Dynamik Data Exchange (DDE)
- Object Linking and Embedding (OLE)

Bei der **Zwischenablage** handelt es sich um eine zeitweilige Ablage, die nur für rechnerinterne Übertragungen geeignet ist, da sie den manuellen Eingriff des Bedieners erfordert. Allerdings kann mit ihr ein Anwender ausgewählte Daten allen anderen Programmen, die im System vorhanden sind, zur Verfügung stellen.

Im Gegensatz dazu sind **gemeinsame Speicherbereiche** nur für Programme geeignet, die speziell aufeinander abgestimmt sind. Innerhalb von Windows wird dieser Datenaustausch über Dynamik Link Libraries (*.DLL-Dateien) realisiert. Dabei ist der Datenaustausch auf eine Programmeinheit begrenzt, da dieser nicht über standardisierte Protokolle geschieht.

Bei **DDE** handelt es sich um ein offizielles Kommunikationsprotokoll, das intern auf dem Austausch von Meldungen basiert. Für die Programmierung steht eine Sammlung von DDE-Funktionen zur Verfügung, die die Kommunikation von Programm zu Programm ermöglichen.

Wie DDE ist auch **OLE** ein Standard-Protokoll, um Daten und Befehle auszutauschen. OLE ist so realisiert, daß es auf dem DDE-Protokoll aufbaut, wobei die Übermittlung von Befehlszeichenketten über DDE erfolgt.

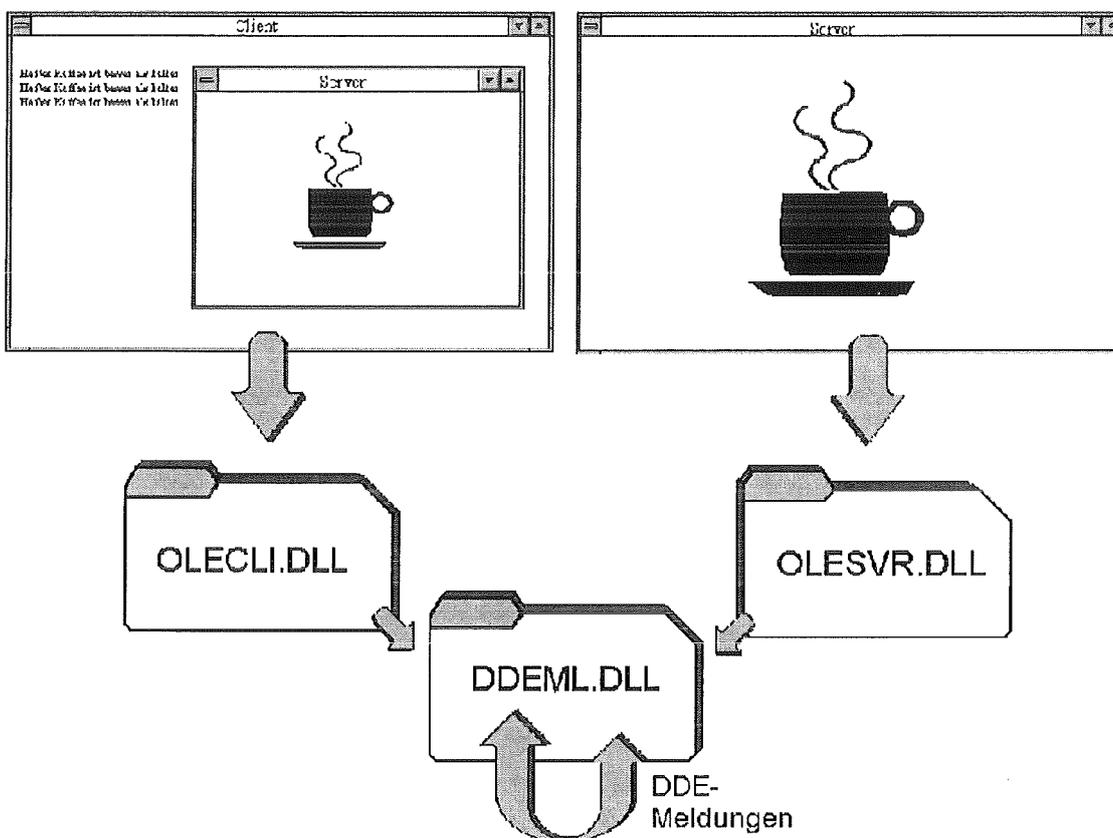


Abbildung 2.2 - Kommunikationsablauf unter Windows (DDE/OLE)

Die Programmierung einer OLE-Applikation erfolgt unter Zuhilfenahme von Windows-API-Aufrufen. Diese Funktionen werden innerhalb von Windows durch DLL's bearbeitet (OLECLI.DLL für die Client-, OLESVR.DLL für die Server-Seite). Abbildung 2.2 zeigt den Kommunikationsablauf unter Windows (DDE/OLE). Die OLE-Objekte selbst sind gekapselte Daten unterschiedlichster Art, die sich z.B. aus Texten, Zahlen, Bildern oder auch Tönen zusammensetzen können.

2.2.1.2 Das "OLE Automation Interface"

Für den Einsatz auf PC's unter MS-Windows wurde CCI zusätzlich als OLE2.0-Automation-Server in Windows-Mosaic implementiert. Das bedeutet, daß jede Anwendung, die als OLE Client eingesetzt werden kann, auf Mosaic über diese Schnittstelle zugreifen kann. Für die praktische Nutzung wurde von NCSA der Befehlssatz des "OLE Automation Interface" entwickelt, der lokalen Applikationen eine stabil laufende, objektorientierte Basis zur Verfügung stellt, auf der diese mit Mosaic kommunizieren und es kontrollieren können.

Ebenso wie die CCI-Spezifikation unterliegt auch das "OLE Automation Interface" (OLE-AI v0.50) als neue Schnittstelle ständigen Änderungen und Erweiterungen. Folgende Funktionen sind derzeit in NCSA Mosaic für Windows 2.0 implementiert:

-  DoActivate
-  ResolveUrl(LPCTSTR szURL)
-  DownloadUrl(LPCTSTR szURL, LPCTSTR szPath)
-  VERSIONSTRING GetVersion()
-  URLSTRING GetLastLaunchedAnchor(LPCTSTR szName)
-  ITEMCOUNT GetAnchorCount()
-  DESKSTRING GetAnchorDesk(short index)
-  URLSTRING GetAnchorUrl(short index)
-  ITEMCOUNT GetHistoryCount()
-  ITEMSTRING GetHistoryItem(short index)
-  URLSTRING GetCurrentDocUrl()
-  TITLESTRING GetCurrentDocTitle()
-  VOID UserCommand(short)

Die Abbildung 2.3 zeigt die Implementierung dieses OLE-Befehlssatzes in Windows-Mosaic, wobei für die Funktionen ResolveUrl() und DownloadUrl() beispielhaft die verwendeten Funktionalitäten des CCI-Protokolls eingezeichnet sind:

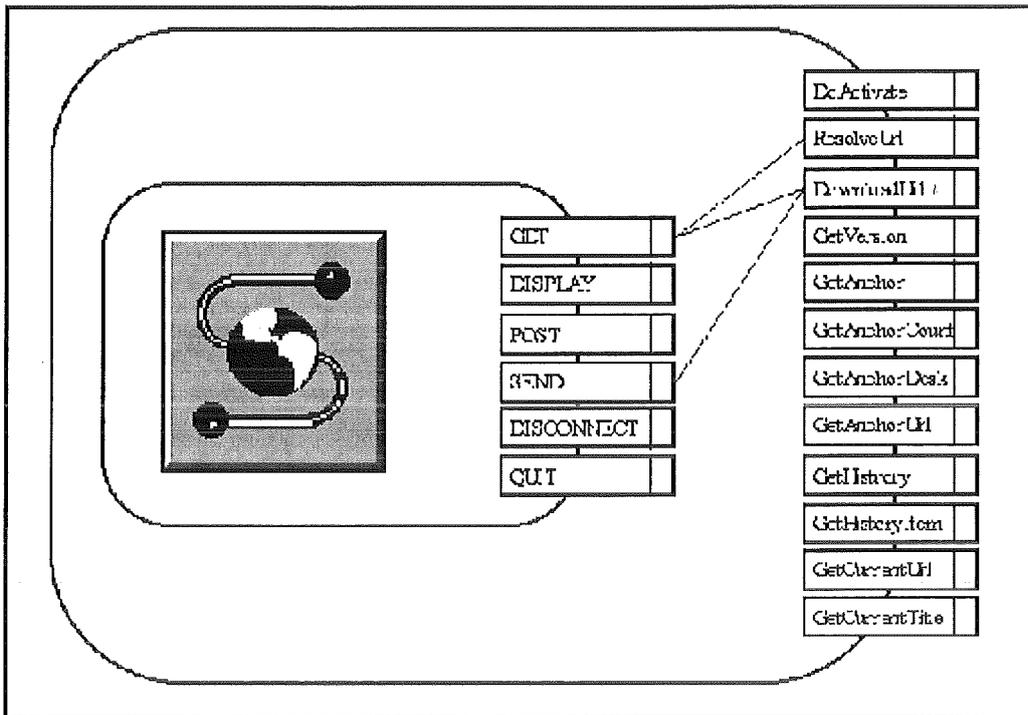


Abbildung 2.3 - Implementierung des "OLE Automation Interface" in Windows-Mosaic

Im Gegensatz zu der C-API (siehe Kapitel 2.2.2.1) beinhaltet der OLE-Befehlssatz keine Funktionen zum Aufbau und zum Abbruch der CCI-Verbindung. Dies wird durch die allgemeinen Funktionen für die Kommunikation mit OLE-fähigen Programmen, die in der jeweiligen Programmiersprache realisiert sind, abgedeckt. In Visual Basic beispielsweise lautet der Aufruf zur Initialisierung einer CCI-Verbindung wie folgt:

```
Setze Mosaic_CCI_Obj = ErstelleObjekt("Mosaic.MosaicCCI")
```

Visual Basic (VB) ist für eigenständige CCI-Applikationen besonders geeignet, da diese Programmiersprache die OLE Automation unterstützt und das Erstellen selbstlaufender Programme ermöglicht.

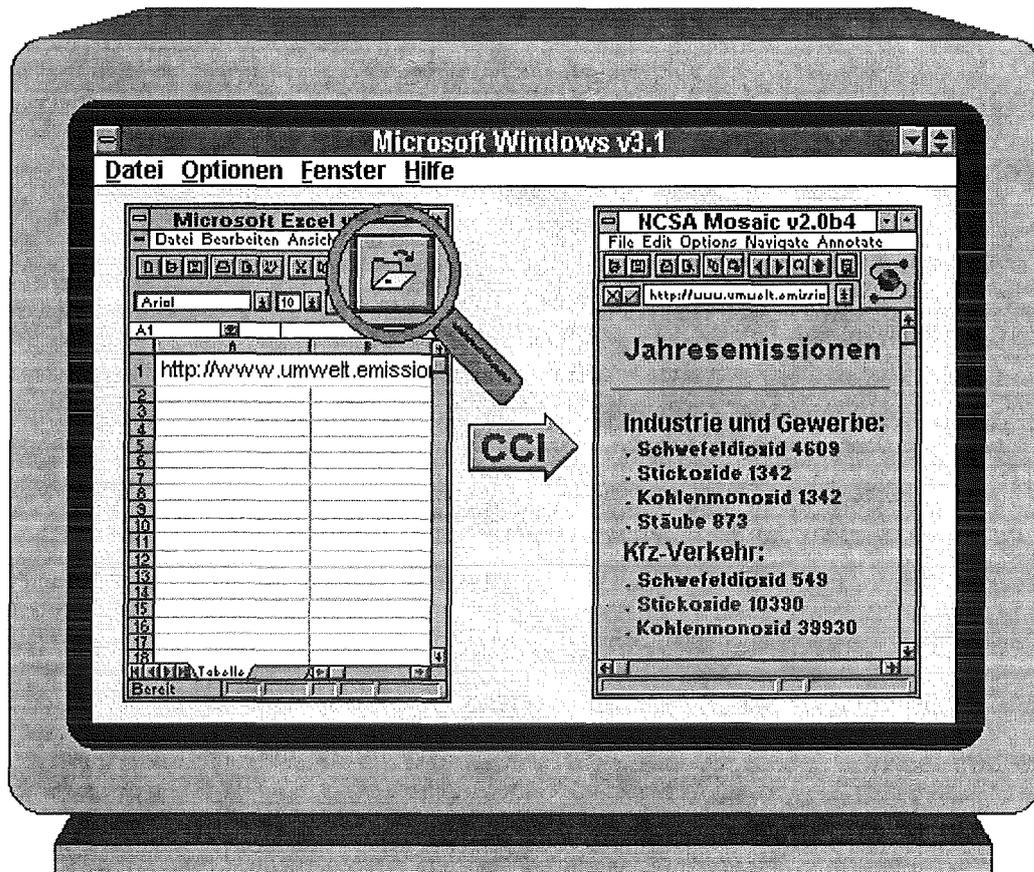


Abbildung 2.4 - Anwendungsbeispiel der Funktion `ResolveUrl()` in Excel

Der einfachere Weg, das "OLE Automation Interface" zu nutzen, ist der Einsatz eines OLE-fähigen Anwendungsprogramms wie zum Beispiel Winword 6.0 oder Excel 5.0, wobei in Excel die Sprache Visual Basic 3.0 als Makrosprache integriert ist. Abbildung 2.4 zeigt, wie ein einfacher Aufruf einer URL mittels `ResolveUrl()` aus Excel heraus aussehen kann. Hierbei wurde das entsprechende Makro über ein in die Funktionsleiste integriertes Icon aktiviert.

Im Folgenden werden die einzelnen Funktionen mit ihren Möglichkeiten ausführlich erläutert:



DoActivate

Mit diesem Befehl ist es möglich, Mosaic (als Icon oder im Fenstermodus) als aktives Programm in den Vordergrund zu schieben.



`ResolveUrl(LPCTSTR szURL)`

Mosaic fordert die übergebene URL an und stellt das empfangene Dokument im aktiven Fenster von Mosaic dar.

Beispiel: `ResolveUrl("http://www.rus.uni-stuttgart.de")` veranlaßt Mosaic, die Homepage des RUS darzustellen.



`DownloadUrl(LPCTSTR szURL, LPCTSTR szPath)`

Mit dieser Funktion wird Mosaic angewiesen, die übergebene URL zu laden und in die vorgegebene Datei zu speichern, ohne das Dokument anzuzeigen.

Beispiel: DownloadUrl("http://www.uni-stuttgart.de";"H.HTM")



VERSIONSTRING GetVersion()

GetVersion() liefert als Ergebnis einen String mit Informationen zu den Versionen von Mosaic und der CCI-Schnittstelle.

Beispiel: GetVersion() = "NCSA MOSAIC/2.0.0b4(x86)/CCI 0.11"



URLSTRING GetLastLaunchedAnchor(LPCTSTR szName)

Diese Routine hilft einem externen Programm, den vollständigen Anker eines angeforderten Dokuments zu erhalten. Wird kein Dateiname übergeben, so liefert die Funktion die URL des zuletzt erhaltenen Dokuments. Wird ein Name übergeben, so vergleicht Mosaic diesen mit dem zuletzt erhaltenen Dokument: Ist dieser identisch, liefert Mosaic die entsprechende URL, ist dies nicht der Fall, so erhält man einen leeren String.

Beispiel: Wird eine URL mit einem Dateinamen aktiviert, der aus mehr als den in Windows erlaubten 8+3 Zeichen besteht (z.B. "bigfilename.xyz"), so wird an den mit der Dateiendung "*.xyz" verbundenen externen Viewer nur der Name "bigfilen.xyz" weitergegeben. Mit Hilfe der Funktion GetLastLaunchedAnchor() erhält der Viewer die vollständige URL (z.B. http://sdg.ncsa.uiuc.edu/bigfilename.xyz).

Die folgenden drei Funktionen liefern Informationen über alle Links, die das zum Zeitpunkt der Anforderung im Mosaic-Fenster dargestellte Dokument enthält:



ITEMCOUNT GetAnchorCount()

Diese Funktion liefert als Integer-Ergebnis die Anzahl der Links im aktuellen Mosaic-Dokument.



DESKSTRING GetAnchorDesk(short index)

GetAnchorDesk() liefert nach Übergabe einer "Integer"-Zahl N die Beschreibung des Nten Link im aktuellen Mosaic-Dokument (z.B. Titel).

Beispiel: Die Indizierung der Links beginnt mit "1". Daraus folgt, daß der Aufruf GetAnchorDesk(1) die Beschreibung des ersten Link liefert.



URLSTRING GetAnchorUrl(short index)

Diese Funktion liefert nach Übergabe einer Integer Zahl N einen String, der die URL repräsentiert, auf die im Nten Link des aktuellen Dokuments verwiesen wird.

Die folgenden beiden Funktionen ermöglichen die Nutzung der Informationen aus Mosaics History-List:



ITEMCOUNT GetHistoryCount()

GetHistoryCount() liefert als Integer-Ergebnis die Anzahl der Einträge der History-List des zu diesem Zeitpunkt aktiven Mosaics.



ITEMSTRING GetHistoryItem(short index)

Das Ergebnis dieser Funktion ist nach Übergabe einer Integer-Zahl N die URL des Nten Eintrages der History-List. In der praktischen Anwendung wird vor jedem Aufruf dieser Funktion die Anzahl der Einträge mit GetHistoryCount() ermittelt.

Beispiel: N = GetHistoryCount(); LastURL = GetHistoryItem(N) ... liefert die URL des letzten Eintrags in der History-List.

Die nächsten beiden Funktionen liefern Informationen über das derzeit in Mosaic dargestellte Dokument selbst:



URLSTRING GetCurrentDocUrl()

Analog zu GetAnchorUrl() liefert diese Funktion die URL des in Mosaic dargestellten Dokumentes.



TITLESTRING GetCurrentDocTitle()

GetCurrentDocTitle() liefert den Titel des in Mosaic dargestellten Dokumentes.



VOID UserCommand(short)

Diese Funktion schickt die folgenden 6 Anweisungen an Mosaic:

```
#define BACK          0
#define FORWARD      1
#define RELOAD        2
#define HOME          3
#define EXIT          4
#define PRESENT       5
```

Im folgenden werden zwei kleine **Beispielprogramme** beschrieben, die die Umsetzung dieser Funktionen in Excel 5.0 zeigen. Die zugehörigen Makros finden sich im Anhang A.2.

Das Makro **SlideShow()** setzt die Funktionen ResolveUrl() und GetCurrentDocUrl() ein, um eine "Slide-Show" ablaufen zu lassen. Die gewünschten URL werden in Excel in untereinander liegende Zellen eingetragen. Die "Show" beginnt bei der obersten URL und stellt so lange die URL der jeweils darunter liegenden Zelle dar, bis eine leere Zelle erreicht wird.

Das zweite Makro **GetAnchors()** nutzt die drei Funktionen GetAnchorCount(), GetAnchorUrl() und GetAnchorDesc(), um die URL's des in Mosaic dargestellten Dokuments mit deren Beschreibung aufzulisten.

2.2.2 CCI über TCP/IP

Im Gegensatz zu der OLE-Verbindung, bei der sich Client-Anwendung und Mosaic lokal in einem Netzwerk befinden müssen, läßt die CCI-Verbindung über TCP/IP sowohl lokalen als auch entfernten Zugriff zu. In den folgenden beiden Abschnitten werden die von NCSA zur Verfügung gestellte Funktionsbibliothek und die darauf aufbauende CCI-Shell von Stan Letovsky erläutert.

2.2.2.1 Das Application Programmers Interface (API)

Mit dem Application Programmer's Interface (API) stellt NCSA eine Funktionsbibliothek für die Programmiersprache "C" zur Verfügung. Mit ihr ist die Entwicklung von Programmen möglich, die lokal und entfernt über TCP/IP die CCI-Schnittstelle nutzen können. Mit Hilfe der API-Funktionen kann der Benutzer z.B. eigene Browser entwickeln, die als selbständige Anwendungen parallel zu der laufenden Web-Session ausgeführt werden und die zusätzliche Funktionen bereitstellen können. In Kapitel 4 werden denkbare Anwendungen erläutert. Die derzeitige Version (V1.1 vom 31. März 95) umfaßt folgende Funktionen:



MCCIInitialize()



MCCIPort MCCIConnect(SvAddress,port,callback,callbackData)



int MCCIGetSocketDescriptor(serverPort)



int MCCIIsConnected(serverPort)



int MCCIPoll(serverPort)



int MCCIIsThereInput(SvPort)



int MCCISendAnchor(SvPort,status,callback,callbackData)



int MCCIGet(SvPort,uri,output,absRelative,additionalHeader)



int MCCINBGet(SvPort,uri,output,absRelative,additionalHeader)



int MCCIPost(SvPort,url,contentType,data,dataLength,output)



int MCCISendOutput(SvPort,mimeType,callback,callbackData)



int MCCISendOutputStop(SvPort,mimeType)



int MCCISendBrowserView(SvPort,status,callBack,callBackData)



int MCCIDisconnect(SvPort)

Gegen Ende November 95 soll eine Neufassung mit erweitertem Funktionsumfang zur Verfügung stehen.

Die Integer-Werte, die von den meisten dieser Funktionen zurückgegeben werden, geben Aufschluß über Erfolg oder Mißerfolg des Funktionsaufrufs:

- 0 - MCCI_OK
- 1 - MCCI_FAIL
- 2 - MCCI_OUTOFMEMORY
- 3 - MCCI_REQUEST_FAIL
- 4 - MCCI_NETWORK_ERROR

In der Abbildung 2.5 ist die Anwendung der C-API dargestellt. Im Gegensatz zu der OLE-AI, die in Windows-Mosaic selbst implementiert ist, ist die C-API als C-Library (libcci.a) auf dem FTP-Server des NCSA erhältlich.

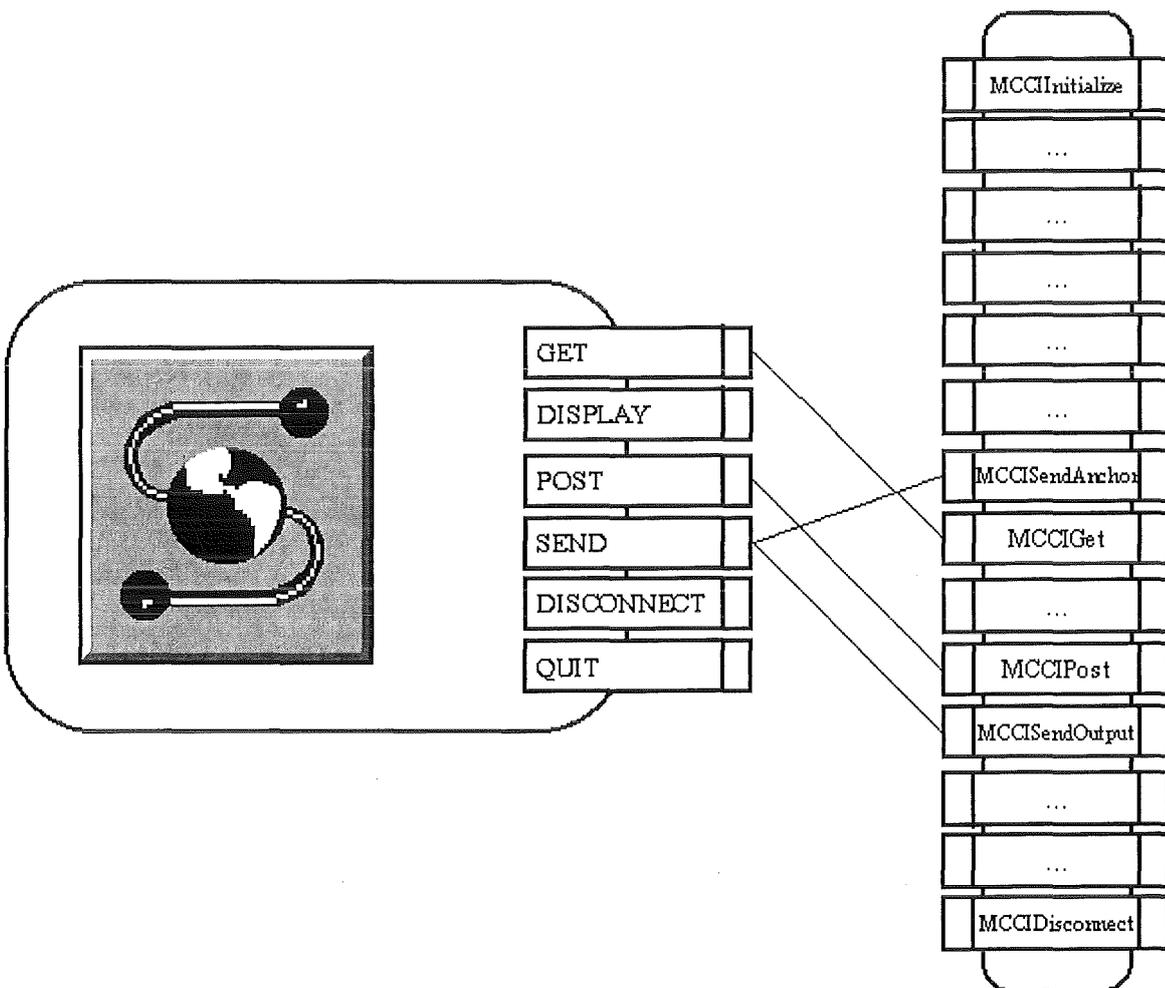


Abbildung 2.5 - Anwendung der C-API (libcci.a) bei X-Mosaic

Nach dem Einbinden der mitgelieferten Header-Datei <cci.h> steht dem Programmierer der volle Funktionsumfang der C-API zur Verfügung.

Im Einzelnen bietet die C-API derzeit folgende Möglichkeiten:



MCCIInitialize()

Diese Funktion wird vor allen anderen aufgerufen. Sie initialisiert die CCI-Bibliothek.



MCCIPort MCCIConnect(serverAddress,port,callback,callbackData)

```
char *serverAddress;  
int port;  
void (*callback) ()  
void *callbackData;
```

Diese Funktion baut eine Verbindung zu einer laufenden Mosaic-Anwendung auf. Dabei sind die Variablen folgendermaßen definiert:

serverAddress ist der hostname oder die IP-Adresse des Rechners, auf dem Mosaic läuft;

port ist die Nummer des Ports, auf dem Mosaic auf CCI-Verbindungen wartet;

callback() ist der Name einer Funktion, die bei Beendigung der Verbindung aufgerufen wird;

callbackData wird an die *callback()*-Funktion übergeben, die folgende Struktur aufweisen sollte:

```
void CCICallback(MCCIPort serverPort, void *callbackData)
```

MCCIConnect gibt einen Wert vom Typ MCCIPort zurück, der dazu benutzt wird, Mosaic einer CCI-Verbindung zuzuordnen.



int MCCIGetSocketDescriptor(serverPort)

```
MCCIPort serverPort;
```

Mit Hilfe dieses Aufrufs ist es möglich, die Socket-Beschreibung des serverPort anzufordern.



int MCCIIsConnected(serverPort)

```
MCCIPort serverPort
```

Diese Funktion liefert den Wert "1", falls der serverPort einer aktiven Netzwerkverbindung zugeordnet ist. Andernfalls erhält man den Wert "0".



int MCCIPoll(serverPort)

```
MCCIPort serverPort
```

Diese Routine sollte in periodischen Abständen aufgerufen werden. Sie kontrolliert den serverPort, ob Mosaic Daten an die Applikation übermittelt. Ist dies der Fall, werden die Daten an die in MCCISendAnchor() und MCCISendOutput() spezifizierten callback()-Funktionen weitergegeben. Hier liefert MCCIPoll() den Wert "1".

Kommen keine Daten, erhält man den Wert "0", wurde die Verbindung abgebrochen, liefert MCCIPoll() den Wert "-1".



int MCCIsThereInput(serverPort)

MCCIPort serverPort

Die Funktion liefert den Wert "1", wenn von der Verbindung, die mit serverPort verknüpft ist, Daten gesendet werden. Andernfalls erhält man "0". Zur weiteren Bearbeitung ankommender Daten sollte die Funktion MCCIPoll() aufgerufen werden.



int MCCISendAnchor(serverPort,status,callback,callbackData)

MCCIPort serverPort;

int status;

void (*callback) ();

void *callbackData;

Diese Funktion weist Mosaic an, der Client-Applikation bei der Aktivierung eines neuen Ankers mittels Open, Back, Forward oder Home eine Mitteilung zu schicken. Die Variablen haben dabei folgende Bedeutung:

serverPort verweist auf eine aktive CCI-Verbindung;

status kann auf MCCI_SEND_BEFORE bzw. MCCI_SEND_AFTER gesetzt werden, je nachdem, ob Mosaic die Mitteilung vor oder nach dem Laden des Ankers absetzen soll. Durch *status* = 0 wird die Anweisung annulliert.

callback() ist eine Funktion, die bei der Aktivierung eines Ankers aufgerufen werden soll. Diese *callback()*-Funktion sollte dabei folgende Struktur aufweisen:

*void callback(char *anchor, void *callbackData)*

anchor ist die URL, die in Mosaic aktiviert wurde.



int MCCIGet(serverPort,uri,output,absRelative,additionalHeader)

MCCIPort serverPort;

char *uri;

int output;

int absRelative;

char *additionalHeader;

Die Funktion weist Mosaic an, eine URL anzufordern. Die Wirkung ist mit dem manuellen Aufruf innerhalb von Mosaic identisch.

serverPort verweist auf eine aktive CCI-Verbindung;

uri ist die zu öffnende URL;

output weist Mosaic an, wie mit dem erhaltenen OUTPUT zu verfahren ist:

MCCI_OUTPUT_NONE - Unterdrückung der Darstellung;

MCCI_OUTPUT_CURRENT - Darstellung im aktuellen Fenster;

MCCI_OUTPUT_NEW - Darstellung in einem neuen Fenster;

MCCI_DEFAULT - (wie MCCI_OUTPUT_CURRENT)

absRelative definiert die Ausführung der URL als relativen oder absoluten Hyperlink:

MCCI_ABSOLUTE - URL als absoluter Hyperlink;

MCCI_RELATIVE - URL als relativer Hyperlink;

MCCI_DEFAULT - (wie MCCI_ABSOLUTE);

additionalHeader ist ein String, der zusätzliche HTTP-Header-Informationen zur Weitergabe an den HTTP-Server enthalten kann. Sollen keine Informationen weitergegeben werden, so erhält *additionalHeader* den Wert "NULL".

Ist der Aufruf erfolgreich, so gibt MCCIGet den String "MCCI_OK" zurück.



`int MCCINBGet(serverPort,uri,output,absRelative,additionalHeader)`

Diese Funktion ist bezüglich der Funktionsweise mit `MCCIGet` identisch. Der Unterschied besteht darin, daß die Applikation nicht blockiert, während Mosaic die übergebene URL erwartet. Die Funktion schickt die Anforderung an Mosaic und kehrt zum Programm zurück. Dabei ist der zurückgegebene String "MCCI_OK" oder "MCCI_FAIL", je nachdem, ob die Anforderung abgeschickt werden konnte oder nicht.



`int MCCIPost(serverPort,url,contentType,data,dataLength,output)`

```
MCCIPort serverPort;  
char *url;  
char *contentType;  
char *data;  
int dataLength;  
int output;
```

Diese Funktion weist Mosaic an, die in `data` enthaltenen Daten an den in der URL definierten HTTP-Server zu schicken. Dabei bedient sie sich der HTTP-Methode POST.

`serverPort` verweist auf eine aktive CCI-Verbindung;

`url` definiert die URL des HTTP-Servers;

`contentType` ist eine String-Variable, die einen gültigen MIME-Typ enthält. Mit ihr wird dem HTTP-Server mitgeteilt, welchem Content-Typ die folgenden Daten zuzuordnen sind.

`data` enthält die Daten, die an den HTTP-Server geschickt werden sollen;

`dataLength` spezifiziert die Länge von `data`;

`output` weist Mosaic an, wie mit dem erhaltenen OUTPUT zu verfahren ist:

```
MCCI_OUTPUT_NONE - Unterdrückung der Darstellung und Rückgabe über den  
CCI-Port;  
MCCI_OUTPUT_CURRENT - Darstellung im aktuellen Fenster;  
MCCI_OUTPUT_NEW - Darstellung in einem neuen Fenster;  
MCCI_DEFAULT - (wie MCCI_OUTPUT_CURRENT)
```



`int MCCISendOutput(serverPort,mimeType,callback,callbackData)`

```
MCCIPort serverPort;  
char *mimeType;  
void (*callback) ();  
void *callbackData;
```

Diese Funktion erlaubt es, den OUTPUT eines bestimmten MIME-Typs an die Client-Anwendung zurückzuschicken, ehe dieser in Mosaic im Dokumenten-Fenster oder über eine externe Applikation dargestellt wird.

`serverPort` verweist auf eine aktive CCI-Verbindung;

`mimeType` legt den MIME-Typ fest (z.B. "text/html");

`callback()` ist die Funktion, an die der OUTPUT eines bestimmten MIME-Typ je nach MIME-Typ übergeben wird;

`callbackData` wird als Argument an die `callback()`-Funktion übergeben. Diese hat dabei folgende Form:

```
void SendOutCallback(char *mimeType,char *data,int length,void *callbackData)
```

Dabei entspricht `mimeType` dem MIME-Typ String und `data` dem aktuellen Datenblock vom Typ MIME-Typ.



`int MCCISendOutputStop(serverPort,mimeType)`

```
MCCIPort serverPort;  
char *mimeType;
```

Diese Funktion weist Mosaic an, die mit `MCCISendOutput()` initiierte Datenübertragung zu beenden.



`int MCCISendBrowserView(serverPort,status,callBack,callBackData)`

```
MCCIPort serverPort;  
int status;  
void (*callBack) ();  
void *callBackData;
```

Diese Routine ermöglicht bzw. verhindert die Ausgabe des Inhalts des Browser-Fensters an die in `callBack()` bezeichnete Funktion. Der Wert von `status` entscheidet darüber, ob diese Funktionalität genutzt (`status = 1`) oder nicht genutzt (`status = 0`) wird. Die am häufigsten auftretenden MIME-Typen sind dabei "text/html" und "image/*". Die Daten werden in der Reihenfolge geschickt, in der Mosaic diese vom HTTP-Server erhält.

Hat Mosaic beispielsweise ein "text/html" - Dokument angefordert, das diverse Bilder enthält, wird die `callBack()`-Funktion zuerst mit dem "text/html" und dann jeweils für jedes eintreffende Bild aufgerufen. Hierbei ist zu beachten, daß Inline-Objekte wie z.B. Graphiken derzeit nicht übermittelt werden, wenn sie sich im Cache-Speicher von Mosaic befinden. Das bedeutet, daß eine Anwendung, die auf eine Graphik wartet, die sich bereits im Cache-Speicher befindet, an dieser Stelle hängen bleibt.

Die `callBack()`-Funktion sollte für `MCCISendBrowserView()` folgende Struktur aufweisen:

```
void SendBVCallBack(char *url,char *mimeType,char *data, int length,void *callBackData)
```

Dabei entspricht `mimeType` dem MIME-Typ String und `data` dem aktuellen Datenblock vom Typ MIME-Typ.



`int MCCIDisconnect(serverPort)`

```
MCCIPort serverPort;
```

Diese Funktion beendet die durch den `serverPort` bezeichnete CCI-Verbindung.

Zwei Beispiele für die Anwendung dieser Funktionen finden sich im [Anhang A3](#).

2.2.2.2 Tcl/Tk und CCI

Tcl und Tk sind zwei Software-Pakete, die zusammen ein System zur Entwicklung und zum Einsatz graphischer Benutzerschnittstellen bilden. Tcl, eine Abkürzung für "Tool Command Language", ist eine Skriptsprache, mit der neue Applikationen geschrieben und bestehende erweitert werden können. Dazu stellt sie typische Kontrollstrukturen wie Variablen, Schleifen und Prozeduren bereit, wobei ihre grundlegenden Sprachelemente durch anwendungsspezifische Funktionen erweitert werden können: der Tcl-Interpreter besteht einfach aus einer Library von C-Funktionen, die zu Programmen hinzugelinkt werden kann und die neben der eigentlichen Implementierung der Sprache eine Reihe von Funktionen beinhaltet, mit denen der Befehlssatz von Tcl beliebig erweitert werden kann. [3]

Eine solche Erweiterung ist Tk, ein Toolkit für das X Window System. Mit ihm wird der Funktionsumfang von Tcl dahingehend erweitert, daß Entwickler Motif-ähnliche graphische Benutzeroberflächen durch Tcl-Skripte anstatt durch C-Code erstellen können. (siehe Abbildung 2.6)

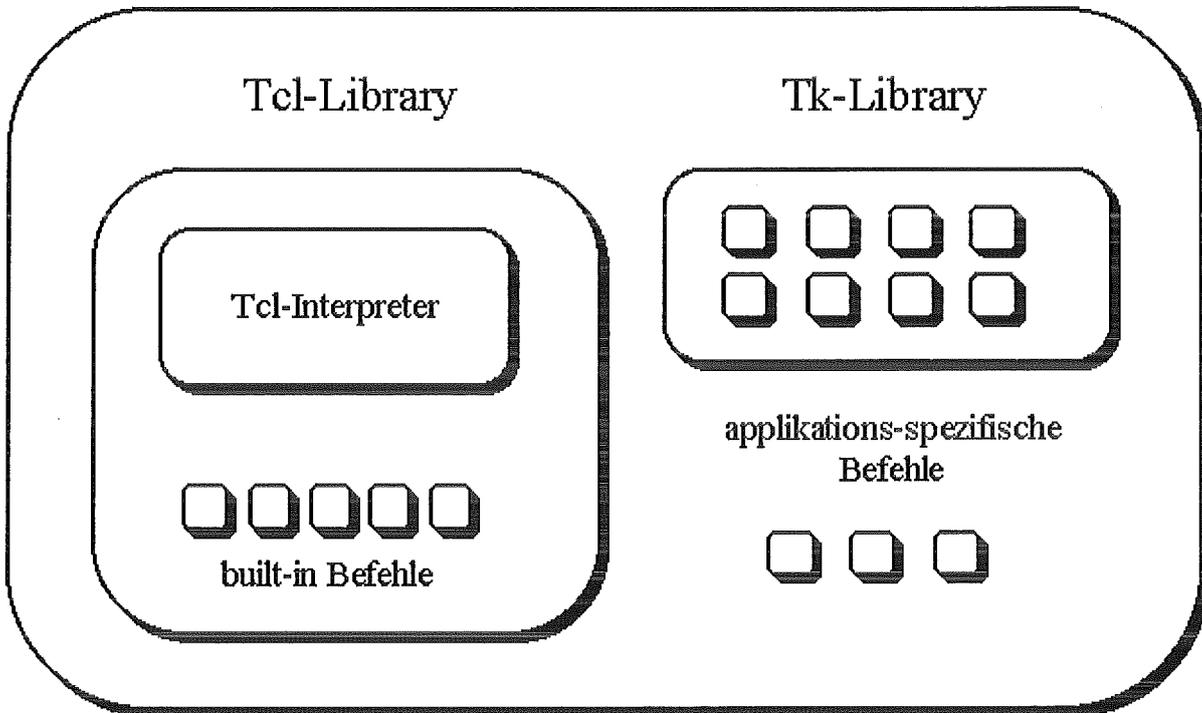


Abbildung 2.6 - Beispiel für eine Tcl/Tk-Applikation

Eine weitere Erweiterung von Tcl/Tk ist Safe-Tcl, eine Skriptsprache, deren Syntax mit Tcl weitgehend identisch ist. Der Unterschied zwischen Tcl und Safe Tcl besteht darin, daß Safe Tcl um zahlreiche "gefährliche" Funktionen reduziert und um neue Funktionen erweitert wurde, die speziell für Anwendungen im Bereich "Mail"-Programme benötigt werden. Damit wird sichergestellt, daß - absichtlich oder unabsichtlich - schlecht geschriebene Skripte keine Möglichkeit haben, die Systemumgebung zu schädigen.

Aufbauend auf Safe-Tcl wurde von Stan Letovsky die CCI-Shell entwickelt, ein Skript-Interpreter für Client-Anwendungen im World Wide Web. Das Tcl CCI API basiert auf dem unter 2.2.2.1 beschriebenen C-API, wobei einige Änderungen vorgenommen wurden, um die Kompatibilität mit Tcl/Tk zu gewährleisten.

Die derzeitige Version umfaßt folgende Funktionen, die wie in 2.2.2.1 beschrieben gehandhabt werden können:



sid McciConnect <S_Adr> <port> [<callBack>] [<cB-Data>]



status McciIsConnected <sid>



status McciPoll <sid>



McciAutoPoll <sid>



status McciIsThereInput <sid>



McciSendAnchor <sid> <status> <callBack> [<cB-Data>]



McciGet <sid> <uri> [<output>] [<absrel>] [<header>]



McciNBGet <sid> <uri> [<output>] [<absrel>] [<header>]



McciPost <sid> <URL> <mimeType> <data> [<output>]



McciSendOutput <sid> <mimeType> <callBack> [<cB-Data>]



McciSendOutputStop <sid> <mimeType>



McciSendBrowserView <sid> <status> [<callBack>] [<cB-Data>]



McciDisconnect <sid>



McciOff

Bei der Konzeption der CCI-Shell (ccish) wurde auf die Funktionen MCCIInitialize() und MCCIGetSocketDescriptor() verzichtet. Die Initialisierung der Bibliothek erfolgt automatisch zusammen mit den Funktionen von Safe-Tcl bei Aufruf der ccish.

Dazugekommen sind die Funktionen McciAutoPoll und McciOff, wobei McciAutoPoll die Tk-spezifische "Event-Loop" nutzt. Diese "Event-Loop" ermöglicht den ereignisgesteuerten Einsatz von Tk-Applikationen über eine Programmierschleife, in der die jeweilige Applikation auf das Eintreten von Ereignissen wartet.

2.2.2.3 C-API oder CCI-Shell

Während die Frage "OLE-Automation Interface oder C-Bibliothek" durch den benötigten Transportmechanismus zu beantworten ist, stellt sich bei X Window-Systemen durchaus das Problem, welcher Weg für die Realisierung einer CCI-Applikation zu wählen ist. Gründe lassen sich sowohl für die Implementierung der CCI-Funktionen in Form von C-Code als auch für die Verwendung von Tcl/Tk anführen:

- Der größte Vorteil, den Tcl mit seinen Erweiterungen bietet, ist sicherlich die vergleichsweise einfache Erstellung komplexer Programme. Zu den umfangreichen Möglichkeiten von Tcl/Tk gehören:

- Eine Vielzahl von Widget-Klassen wie z.B. Menue- und Scrollbar-Widgets, eine Widget-Klasse für Zeichnungen und das Text-Widget, mit dem auch Hypertext-Effekte möglich sind;
- Zugriff auf Ressourcen des X Window-Systems;
- Kommunikation zwischen auf Tk basierenden Applikationen über den Austausch von Tcl/Tk-Skripten.

Beachtung verdient insbesondere die Möglichkeit, über eine Schnittstelle in "C" geschriebene Programmteile als einfache Tcl-Befehle zu integrieren. Bei konsequenter Umsetzung moderner Entwurfsprinzipien vor allem im Bereich der Schnittstellendefinition ist eine sinnvolle Aufgabenteilung zwischen C-Code und Tcl-Skripten einfach zu realisieren. Dabei wird auch der Forderung nach Wiederverwendbarkeit, dem Geheimnisprinzip und der Lokalität von Entscheidungen Rechnung getragen.

Die Programmierung in C-Code bietet im Wesentlichen folgende Vorteile:

- Portierbarkeit der Programme auf unterschiedlichste Plattformen;
- Einbindung der CCI-Funktionalitäten in bereits bestehende Programme;
- Hardwarenahe Programmierung mit der daraus resultierenden erhöhten Performance.

Unter Berücksichtigung obiger Kriterien muß je nach Aufgabenstellung und Voraussetzungen die Entscheidung getroffen werden, welche der beiden Möglichkeiten zur Lösung des Problems günstiger ist.

3 Praktische Anwendungen der CCI-Schnittstelle

Nach der theoretischen Erörterung der Funktionalitäten der CCI-Schnittstelle folgt nun eine Betrachtung der möglichen Szenarien von Anwendungen.

3.1 Anwendungsbereiche

Der erste Anwendungsbereich beinhaltet den Zugriff aus einer Applikation über Mosaic auf Informationen im WWW. Dies kann sowohl lokal über OLE oder TCP/IP als auch verteilt über TCP/IP erfolgen. Vor allem im Bereich von Büroapplikation wie z.B. MS-Excel oder MS-Word für Windows gestattet CCI dem Anwender, über Makros Informationen aus dem WWW anzufordern und in Dateien abzulegen, ohne in eine andere Softwareumgebung wechseln zu müssen. Über entsprechende Konverter können diese Informationen leicht in die jeweilige Anwendung übernommen werden.

Ein weiterer Anwendungsbereich beschäftigt sich mit der Möglichkeit, Mosaic als Benutzeroberfläche - insbesondere im Bereich der "Gateway"-Applikationen - einzusetzen. Wie von Thomas Hofer ausgeführt wurde, ergeben sich hierbei gravierende Probleme, die es bisher unmöglich machten, Mosaic in dieser Funktion sinnvoll zu verwenden [4]. Dazu gehören:

- Geringe Anzahl von Schnittstellenelementen;
- Blockierter Zustand von Mosaic bis zur Beendigung einer Anfrage;
 - Keine Möglichkeit für die Applikation, Rückmeldungen an Mosaic zu geben;
 - Keine Möglichkeit für Mosaic, interaktiv auf eine gestartete Applikation Einfluß zu nehmen.

Für eines der Probleme, die sich aus dem blockierten Zustand von Mosaic ergeben, bietet CCI bereits jetzt einen praktikablen Lösungsansatz. Über ein zusätzlich zu öffnendes (Slave-)Mosaic besteht für eine Applikation die Möglichkeit, dem Anwender Zwischenergebnisse und Zustandsmeldungen zu liefern. Eine ausführliche Beschreibung dieses Ansatzes und die praktische Umsetzung für die Ergebniserückgabe bei CRAYSIM folgt in Abschnitt 3.2.3.

Das Problem der interaktiven Einflußnahme soll durch eine neue Methode zur Übertragung der FOF-Daten, der `<METHOD=CCIPPOST>`, behoben werden, die ab der Version 2.7 von X-Mosaic zur Verfügung stehen wird.

3.2 Umsetzung über OLE und TCP/IP

Die angesprochenen Anwendungsbereiche verwenden zum einen den lokalen, zum anderen den verteilten Zugriff einer Applikation auf Mosaic. Der lokale Zugriff erfolgt je nach System über OLE (nur MS-Windows) oder TCP/IP (Abbildung 3.1). An der Abarbeitung der Anforderungen auf Seiten des WWW-Servers ändert sich dabei durch die Nutzung der CCI-Schnittstelle nichts. Für den Server macht es keinen Unterschied, ob z.B. die Anforderung einer URL aus einem HTML-Dokument heraus über einen Hyperlink oder mittels eines GET-Befehls über CCI gestartet wurde.

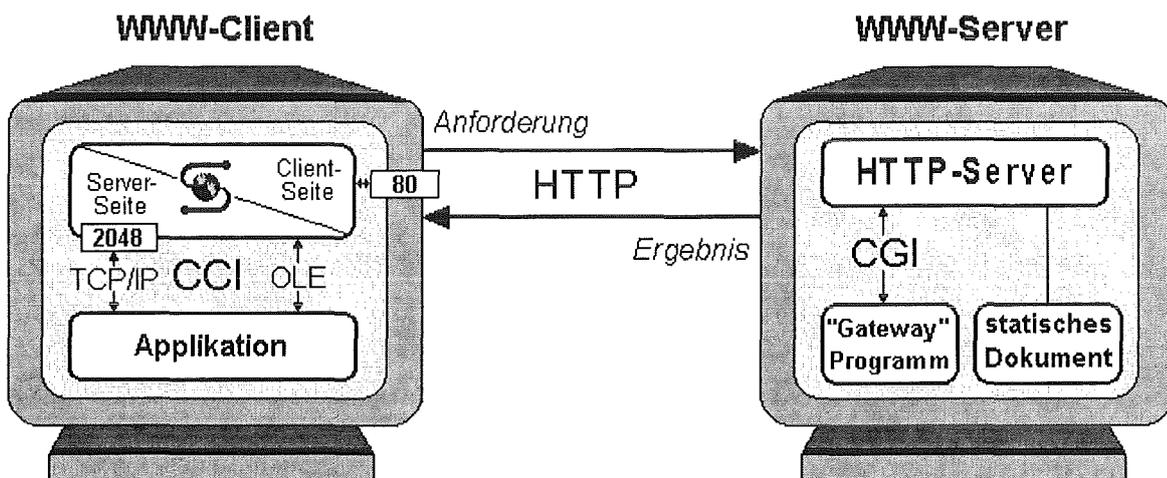


Abbildung 3.1 - Möglichkeiten des lokalen Zugriffs auf Mosaic

3.2.1 Nutzung des OLE Application Interface

Ausgangspunkt für aktuelle Informationen zu der neuesten Version von Mosaic für Windows und die jeweils gültige Spezifikation ist die URL

<http://www.ncsa.uiuc.edu/SDG/Software/WinMosaic/>.

Bei der Programmierung von CCI-Applikationen mit dem OLE-AI bestehen, im Gegensatz zu der C-API-Implementierung, zwei prinzipielle Möglichkeiten des Verbindungsaufbaus: Zum einen kann der Aufbau der CCI-Verbindung mit einer laufenden Mosaic-Session erfolgen, zum anderen kann Mosaic aber auch erst bei Bedarf automatisch gestartet werden.

Im ersten Fall signalisiert Mosaic den Aufbau einer CCI-Verbindung durch ein Symbol in der unteren Statuszeile (Abbildung 3.2), bietet aber nach wie vor alle Möglichkeiten der manuellen Navigation (Back, Forward, ...). Dies ermöglicht das Erstellen von Anwendungen, die auf eine Erweiterung der Funktionalitäten von Mosaic unter Beibehaltung der Browser-Oberfläche abzielen. Denkbar ist hier beispielsweise eine Erweiterung der "History-List" mit Datenbankfunktionen, wobei mit Hilfe der drei "Anchor"-Funktionen besuchte URL's abgelegt und dokumentiert werden können.

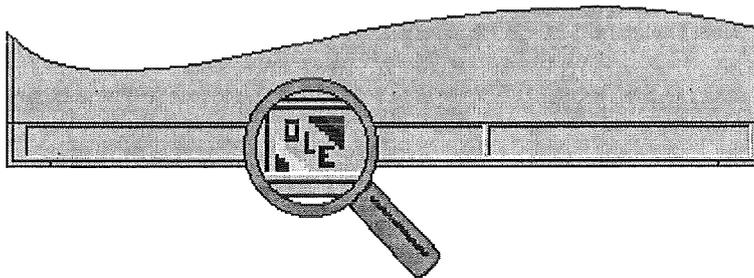


Abbildung 3.2 - OLE-Symbol bei Aufbau einer CCI-Verbindung zu einer Aktiven Mosaic-Session

Der zweite Fall nutzt die Möglichkeit von OLE, Server-Applikationen bei Bedarf selbständig zu starten. Dabei wird Mosaic im Hintergrund als Icon gestartet, was zur Folge hat, daß der Anwender keinen direkten Zugriff auf die Browser-Oberfläche hat. Die benötigten Operationen (ResolveUrl(), ...) werden von der CCI-Applikation mit interaktiver Einflußnahme des Anwenders ausgeführt. Bei Abbruch der CCI-Verbindung wird Mosaic automatisch beendet. Denkbare Anwendungen sind hier Programme, die nur die Funktionalitäten von Mosaic, nicht aber die Browser-Oberfläche nutzen, um Daten aus dem WWW zu erhalten.

Im Anhang A.2 sind zwei Beispielmakros aufgeführt, die die Anwendung des OLE-AI in der Tabellenkalkulation Excel 5.0 zeigen.

3.2.2 Nutzung von CCI über TCP/IP

Ausgangspunkt für Informationen zu Mosaic für X-Window und die aktuelle Spezifikation der C-API sind die URL's

<http://www.ncsa.uiuc.edu/SDG/Software/XMosaic/>

<http://www.ncsa.uiuc.edu/SDG/Software/XMosaic/CCI/cci-api.html>.

Dieser Bericht basiert auf den Versionen 2.7b2 (XMosaic), der Version 2.0 (Windows-Mosaic) und der Version 1.1 (C-API).

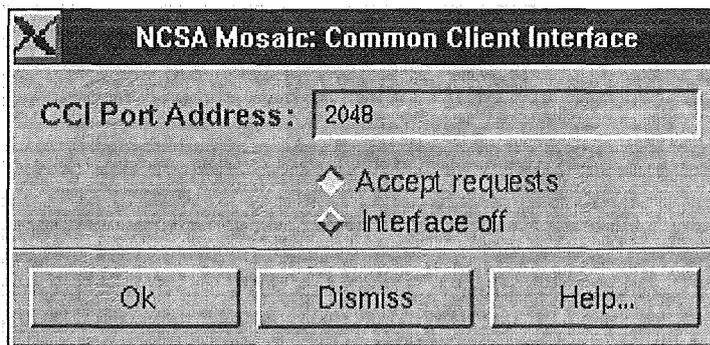


Abbildung 3.3 - Untermenü "CCI..." in Mosaic, in dem die CCI-Port Adresse eingetragen wird

Im Gegensatz zu der Nutzung von CCI als OLE-Server kann die Verbindung über TCP/IP nur mit einer laufenden Mosaic-Session erfolgen. Zusätzlich muß bei XMosaic in dem Menü "File" im Untermenü "CCI" die CCI-Port Adresse eingetragen und aktiviert werden, auf der Mosaic auf CCI-Verbindungen wartet. Bei Windows-Mosaic geschieht dieser Eintrag unter "Preferences" im Bereich "Services". Alternativ dazu besteht die Möglichkeit, Mosaic mit dem Parameter "-cci <Port>" zu starten, wobei der <Port> im Bereich von "1024" bis "65535" liegen kann. Dies ermöglicht eine flexible Anpassung des Ports an lokale Gegebenheiten. Desweiteren würden durch einen dauerhaft voreingestellten Port wichtige Sicherheitsaspekte verletzt, da bei aktiviertem CCI-Port von jedem Rechner im Internet aus die Möglichkeit besteht, Mosaic über die CCI-Schnittstelle ohne weitere Zugangsabfragen zu beeinflussen.

Aus diesem Grund ist es auch ratsam, in der Praxis von der Verwendung des Standard-Ports "2048" abzusehen.

Die Verbindung über TCP/IP erlaubt sowohl die lokale (Abbildung 3.1) als auch die verteilte Anordnung von Mosaic und CCI-Applikation (Abbildung 3.4).

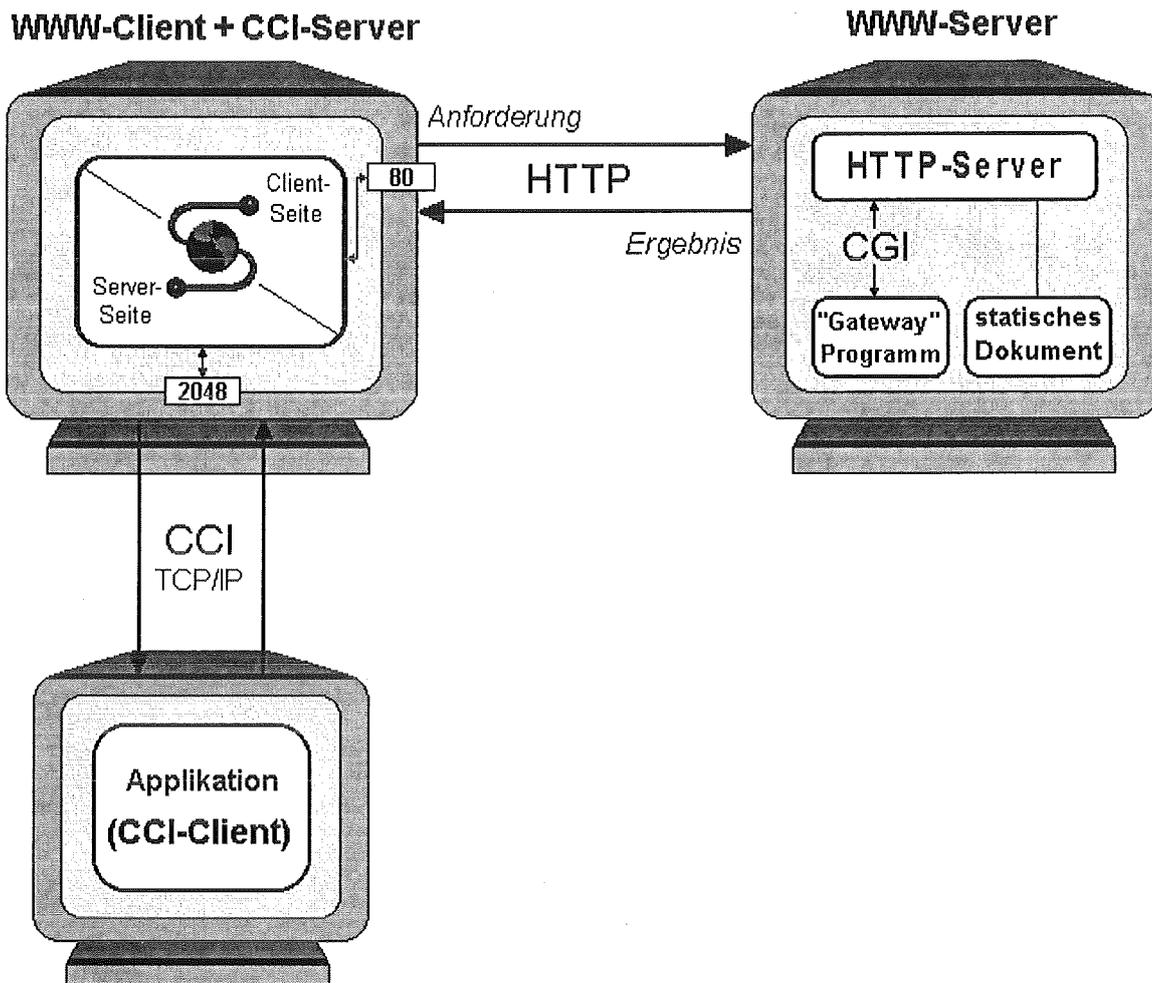


Abbildung 3.4 - Verteilter Zugriff auf Mosaic über die CCI-Schnittstelle

Damit ist, ergänzt durch den Befehlssatz der C-API, ein breites Spektrum verschiedenster Anwendungen denkbar. Unter anderem kann Mosaic auch (siehe Abschnitt 3.2.3) für die Rückmeldung von Zwischenergebnissen einer "Gateway"-Anwendung eingesetzt werden. Eine weitere Möglichkeit ist die Verwendung von Mosaic für Präsentationen oder Schulungen: Eine CCI-Applikation steuert dabei eine größere Anzahl von Mosaic-Sessions auf vernetzten Rechnern und führt den Anwender durch die Hypertext-Seiten.

Im Anhang A.3 finden sich zwei weitere Beispiele mit dem zugehörigen Quellcode für die Anwendung einiger wichtiger C-API Funktionen.

3.2.3 Beispiel: Simulation atmosphärischer Schadstoffausbreitung (CRAYSIM)

Die vom Institut für Kernenergetik und Energiesysteme (IKE) bereitgestellte Simulation atmosphärischer Schadstoffausbreitung wird als Dienst im Umweltbereich eingesetzt. Mit Hilfe einer solchen Simulation möchte man, ausgehend von Informationen über die

Freisetzung eines Schadstoffes in der Umwelt (Emission), zu einer Beurteilung der Auswirkungen in der Umgebung der Quelle (Immission) kommen. [5]

3.2.3.1 Aufgabenstellung

Um die Simulation atmosphärischer Schadstoffausbreitung zu demonstrieren, wurde nach einer graphischen Benutzeroberfläche gesucht, die es dem Anwender einerseits erlaubt, die Simulation einfach zu starten, und die andererseits eine ansprechende Visualisierung der Ergebnisse ermöglicht. Diese Anforderungen werden von Mosaic erfüllt. Über eine HTML-Seite mit Fill-Out-Form ist es möglich, die entfernt liegende Simulationsrechnung als "Gateway"-Programm zu starten und diesem die benötigten Werte zu übermitteln. Die Endergebnisse können nach Abschluß der Simulationsrechnung im Browserfenster sowohl in Textformat als auch als Graphik dargestellt werden.

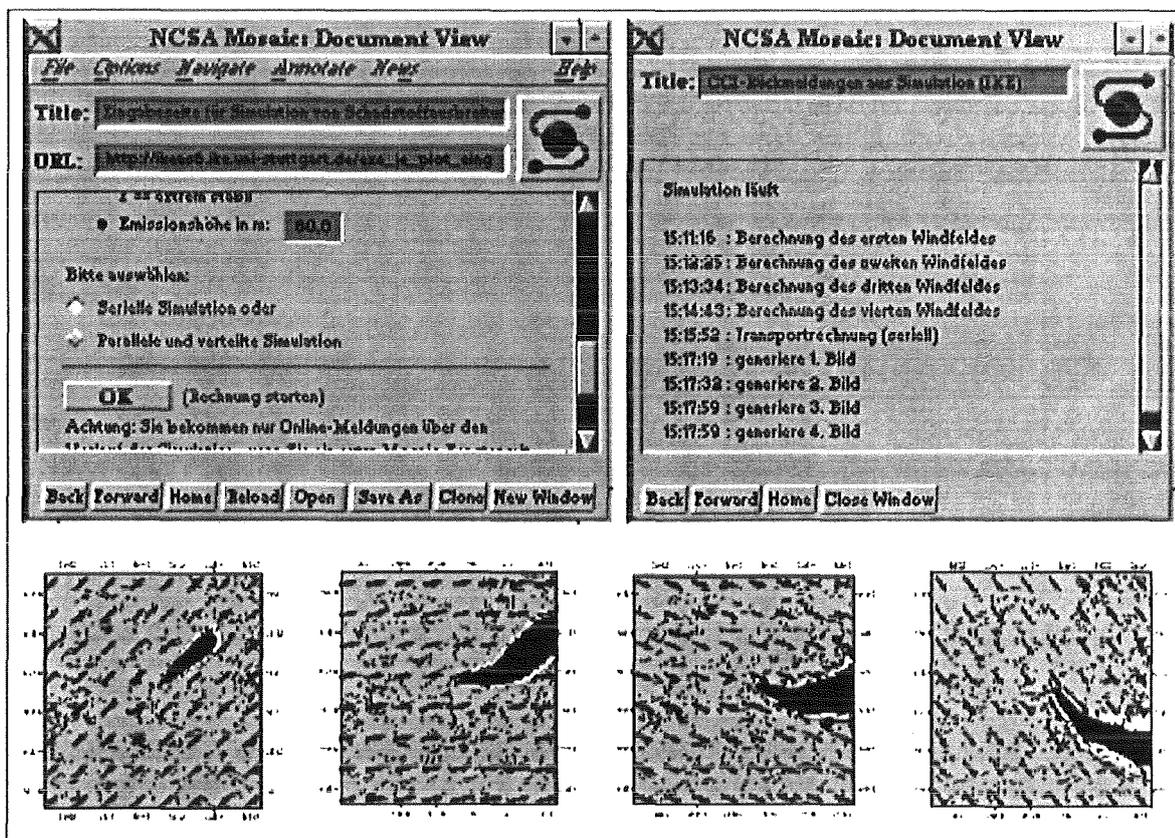


Abbildung 3.5 - Master- und Slave-Mosaic mit in vier Schritten errechneten Windfeldern

Da die Berechnungen mehrere Minuten in Anspruch nehmen, wäre es wünschenswert, Zwischenergebnisse an Mosaic zurückgeben zu können, ohne die Simulation zu beenden. Bei der Simulation von atmosphärischer Schadstoffausbreitung sind dies:

- Zwischenergebnisse einzelner Zeitschritte (image/gif)
 - Windfelder zu einem bestimmten Zeitpunkt der Simulationszeit
 - Konzentrationsfelder zu einem bestimmten Zeitpunkt der Simulationszeit
- Informationen über den Verlauf bestimmter Algorithmen (text/html)
 - Windberechnung - Iterationszahl

Die im Rahmen dieses Projektes erstellte Beispielfunktion ZwErg() arbeitet mit der Funktion MCCIGet(), was zur Folge hat, daß das "Gateway"-Programm bis zur Rückmeldung über Erfolg oder Mißerfolg der Datenübertragung in einen Wartezustand übergeht. Dies vereinfacht die Fehlerbehandlung, kostet aber auch die Rechenzeit, die während der Datenübertragung nicht genutzt wird. Die Entscheidung, welche der Funktionen verwendet wird, ist von den jeweiligen Rahmenbedingungen abhängig und muß von Fall zu Fall neu überdacht werden.

3.2.3.2 Realisierung über CCI

Das Funktionsmodul ZwErg() übernimmt die Aufgabe, die Zwischenergebnisse vom MIME-Typ "text/html" oder "image/gif" an das (Slave-)Mosaic zu übermitteln. Hierzu wurden folgende Schnittstellen definiert:

```
-> int    cciport      * Standard: 2048
-> char  datentyp[5]  * je nach Zwischenergebnis entweder "html"
                    * oder "gif"
-> char  *datenbody   * Daten des Zwischenergebnisses
-> int   length       * Länge der Daten in Byte
<- int  ierr         * Statuscode: 0=Erfolgreich,
                    * 1-5=Fehlermeldung (Anhang A.4)
```

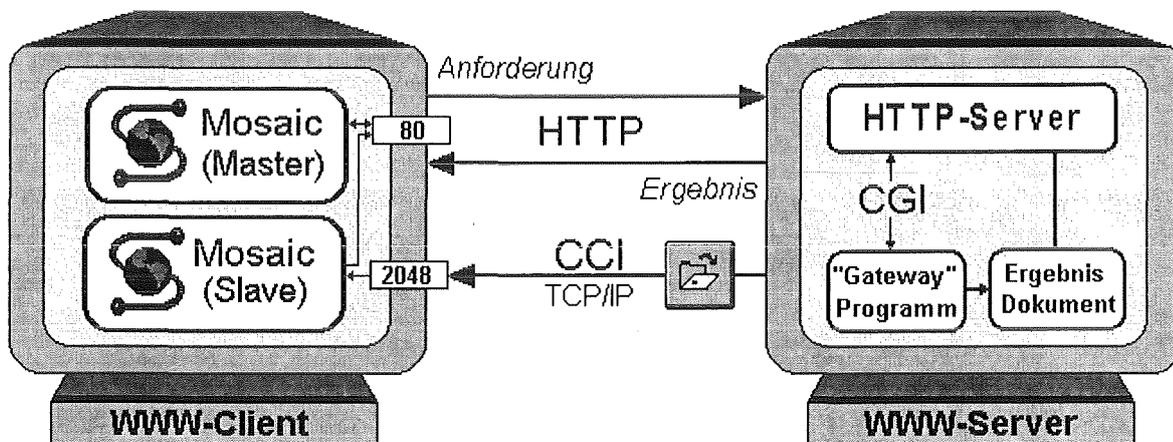


Abbildung 3.7 - Ablauf der Rückgabe der Zwischenergebnisse

Abbildung 3.7 zeigt die Rückgabe der Ergebnisse in sechs Schritten:

- Kontrolle der Vollständigkeit der übergebenen Daten
- Anlegen von temporären Dateien (Ergebnisdokumenten):
 - HTML-Daten: "ZwErg.html"
 - GIF-Daten: "ZwErg.html" und "ZwErg.gif"
- Ablegen des datenbody in die temporären Dateien:
 - HTML-Daten: datenbody wird in "ZwErg.html" gespeichert
 - GIF-Daten: datenbody wird in "ZwErg.gif" gespeichert und in die Datei "ZwErg.html" wird der Verweis auf die GIF-Datei im HTML-Format

geschrieben:

```
<IMG SRC="ZwErg.gif" >
```

- Aufbau der CCI-Verbindung mit dem (Slave-)Mosaic über den Port cciport
- Anweisung an Mosaic, die HTML-Datei "ZwErg.html" zu laden
- Schließen der CCI-Verbindung und Löschen der temporären Datei(en)

Der zu ZwErg() gehörige C-Quellcode sowie eine detaillierte Beschreibung des Rückgabecodes findet sich im Anhang A.4.

3.2.3.3 Probleme und Lösungsansätze

Diese Art der Rückgabe von Daten beinhaltet einige Einschränkungen, die eine einfachere und effektivere Lösung für wünschenswert erscheinen lassen:

- Der Anwender muß auf seinem Rechner das zweite (Slave-)Mosaic von Hand öffnen und den CCI-Port einstellen.
- Es ist nicht möglich, interaktiv (z.B. aufgrund der erhaltenen Daten) in den Ablauf der Simulation einzugreifen.
- Es sind nur die am NCSA entwickelten Browser XMosaic und Windows-Mosaic einsetzbar, da andere Browser CCI (noch) nicht unterstützen.

Von Thomas Hofer wurden folgende Lösungsansätze für diese Probleme vorgeschlagen:

- Das Anlegen von Dateien mit Zustandsdaten des "Gateway"-Programms auf Server- oder Clientseite. Der nächste Programmaufruf kann dann, basierend auf diesen Zustandsdaten und weiteren Angaben des Anwenders, das Programm fortführen.
- Die Implementierung eines zweiten Server, der seinen momentanen Zustand beibehält, während der erste Server Zwischenergebnisse an die Client-Anwendung zurückgibt und terminiert.

Aber auch die Weiterentwicklung der CCI-Schnittstelle kann dazu geeignet sein, zumindest die interaktive Einflußnahme auf ein laufendes "Gateway"-Programm zu ermöglichen. Ein Ansatz hierzu ist in der Ankündigung einer weiteren Methode für Fill-Out-Forms zu erkennen: Mittels der <METHOD=CCIPost> soll es möglich sein, Mosaic als Benutzer-Interface für Applikationen einzusetzen.

3.2.4 Beispiel: Inanspruchnahme von Diensten mit dem Kommunikationsinterpret (KIP) über das WWW

Der Kommunikationsinterpret (KIP) ist das zentrale Werkzeug des Projektes "Technischer Zugang zu Umweltinformationen (TZUI). Ziel von TZUI ist es, die unterschiedlichsten Anwendungen innerhalb des Umweltinformationssystems (UIS) beim Zugriff auf Datenbestände, die auf z.T. im ganzen Land Baden-Württemberg verteilten Rechnern gespeichert sind, zu unterstützen. Die zugreifenden Anwendungen laufen sowohl auf

Rechnern im Geschäftsbereich des Umweltministeriums, als auch auf Rechnern anderer Dienststellen mit Umweltaufgaben, die Zugang zu zentralen umweltrelevanten Datenbeständen benötigen. So müssen z.B. die übergreifenden UIS-Komponenten einheitlich auf Daten der Grundkomponenten zugreifen können. Eine detaillierte Beschreibung der Funktionalitäten des Kommunikationsinterpreters kann im "Benutzerhandbuch des Kommunikationsinterpreters" [6] nachgelesen werden.

Um den Einsatz des Kommunikationsinterpreters für den Benutzer komfortabel und anwenderfreundlich zu gestalten, werden grafische Benutzeroberflächen in Form von HTML-Dokumenten zur Verfügung gestellt. Damit wird es ermöglicht, den Kommunikationsinterpreter über einen WWW-Client (z. B. Mosaic) in Anspruch zu nehmen. Voraussetzung hierfür ist das Vorhandensein eines KIPs und eines WWW-Servers auf derselben Maschine (Abbildung 3.8). Zusätzlich wird es, durch den Einsatz der in Mosaic implementierten Schnittstelle Common Client Interface (CCI) dem KIP ermöglicht, sowohl Rückmeldungen als auch Endergebnisse zum Dienstanforderer (Slave-Mosaic) zu schicken.

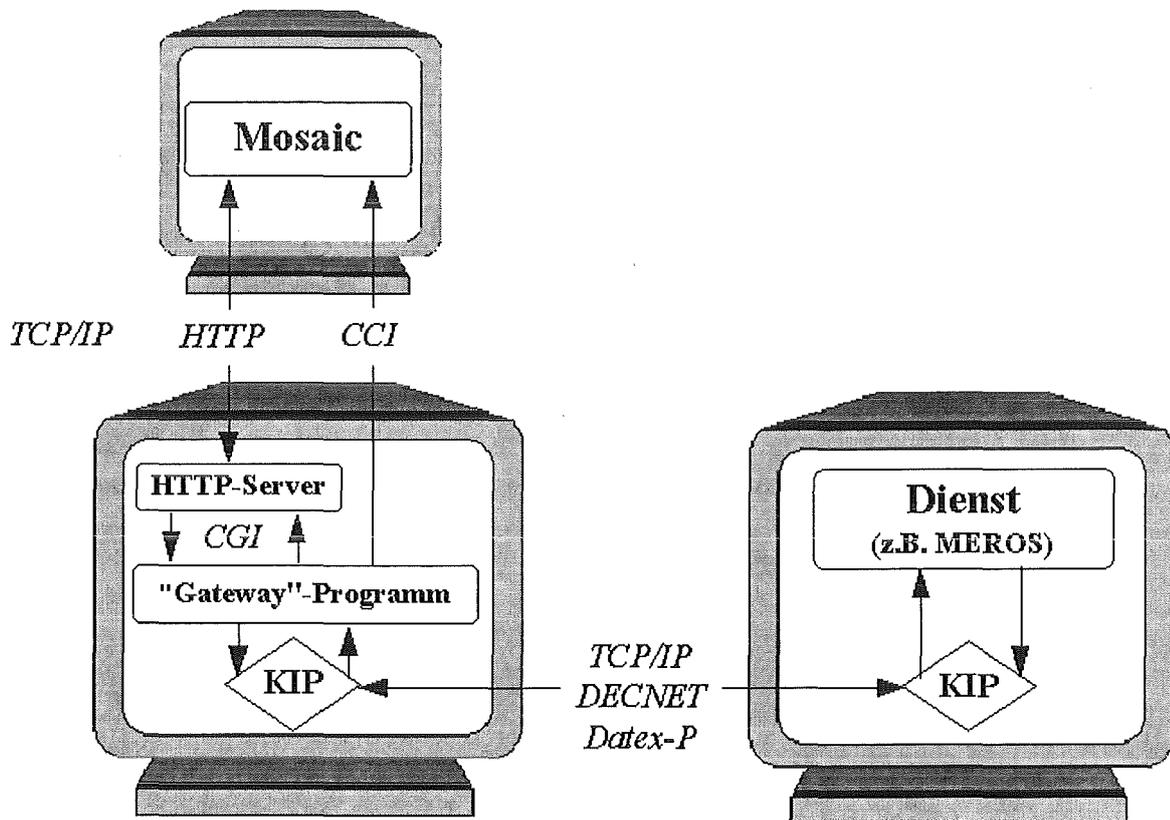


Abbildung 3.8 - Kombiniertes Einsatz von KIP und CCI für den Dienst MEROS

Im folgenden werden die in HTML generierten Benutzeroberflächen und der Einsatz der Schnittstelle CCI näher erläutert.

3.2.4.1 Hauptmenü

Über die "Home Page" gelangt man zu der Seite "Inanspruchnahme von Diensten über den Kommunikationsinterpreter" (siehe Abbildung A1 im Anhang A.6). Diese Seite wird als

Hauptmenü bezeichnet. Sie ist der Ausgangspunkt für weitere HTML-Seiten, von denen aus Dienste über den KIP in Anspruch genommen werden können. Folgende Funktionalitäten stehen zur Verfügung:

- **Dienstanforderung:**
Mit der Dienstanforderung erstellt der Dienstanforderer einen Dienstauftrag. Der Dienstanforderer erhält eine Kennzeichnung des Dienstauftrages zurück, falls der Dienst verfügbar ist. Ist der Dienst nicht verfügbar, so erhält er eine entsprechende Mitteilung.
- **Dienstinanspruchnahme:**
Im nächsten Schritt übergibt der Dienstanforderer die für die Ausführung des gewünschten Dienstes notwendigen Daten an den Kommunikationsinterpret zur Weiterleitung an den gewünschten Dienst. Als Ergebnis der Dienstinanspruchnahme erhält der Dienstanforderer eine Nachricht wo bzw. daß die Ergebnisdaten vorliegen.
- **Ergebnisanforderung:**
Mittel der Ergebnisanforderung kann der Dienstanforderer auf die Ergebnisdateien der erledigten Dienstaufträge zugreifen. Dieser Zugriff erfolgt über die in Mosaic zur Verfügung gestellten Schnittstelle CCI.
- **Statusanfrage:**
Über die Statusanfrage kann sich der Dienstanforderer den Status eigener, noch nicht abgeschlossener Dienstaufträge geben lassen.
- **Übersicht über aktive Dienstaufträge:**
Um bei mehreren aktiven Dienstaufträgen die Übersicht nicht zu verlieren, kann sich der Dienstanforderer eine Übersicht über seine Dienstaufträge geben lassen.
- **Löschen von Dienstaufträgen:**
Über den Löschauftrag kann der Dienstanforderer nach Beendigung des Dienstauftrags diesen löschen oder Dienstaufträge abrechnen.

3.2.4.2 Dienstanforderung

Die Aufgabe der Dienstanforderung ist es, eine Anfrage des Dienstanforderers nach einem Dienst zu befriedigen. Der Dienstanforderer erhält nach der Dienstanforderung eine Kennzeichnung des Dienstauftrages zugewiesen. Mit dieser Kennung kann er den gewünschten Dienst in Anspruch nehmen. Kann der aufgerufene Kommunikationsinterpret die Dienstanforderung nicht befriedigen, so besteht die Möglichkeit, auf die Informationen eines anderen bekannten Kommunikationsinterpreters zuzugreifen. Dies geschieht für den Dienstanforderer transparent.

Für eine ordnungsgemäße Dienstanforderung müssen folgende Werte eingegeben werden (siehe Abbildung A2 im Anhang A.6):

- Name des Dienstes (Dienstname).
- Schlüssel für den Zugriff auf externe Dienste (ID-Nummer).
- Benutzerkennung zur eindeutigen Identifikation der Ergebnisdaten und Dienstaufträge (Benutzerkennung).

- **Auftragsbezeichnung.**
Die Auftragsbezeichnung wird verwendet, um später zwischen mehreren eigenen Dienstaufträgen unterscheiden zu können (Auftragsbezeichnung).

Die Benutzerkennung und Auftragsbezeichnung müssen für jeden Dienstauftrag eindeutig sein. D.h. ein Benutzer muß jedem Auftrag eine eigene Auftragsbezeichnung geben. Zwei Dienstaufträge mit der gleichen Auftragsbezeichnung können nicht gleichzeitig existieren.

Wurden die Werte eingegeben, so kann die Dienstanforderung durch aktivieren des OK-Buttons gestartet werden.

Als Ergebnis der Dienstanforderung werden folgende Werte zurückgeliefert und gleich in das HTML-Dokument für die Dienstinanspruchnahme übernommen (siehe Abbildung A3 im Anhang A.6):

- **Nummer des Dienstauftrags:**
Für jede erfolgreiche Dienstanforderung wird ein Dienstauftrag erstellt. Dem Dienstauftrag wird, um die spätere Zuordnung zu ermöglichen, eine Nummer als Kennzahl vergeben (Auftragsnummer).
- **Mögliche Art der Inanspruchnahme:**
Die mögliche Art der Inanspruchnahme gibt an, wie der Dienst in Anspruch genommen werden kann. Es besteht die Möglichkeit, daß die Inanspruchnahme des gewünschten Dienstes auf verschiedene Arten möglich ist. In diesem Fall kann der Dienstanforderer selber entscheiden, welche er davon beanspruchen möchte (Mögliche Art der Inanspruchnahme).
- **Pfad für die Eingabedateien:**
Der Pfad kennzeichnet ein Verzeichnis, indem die Eingabedateien des Dienstanforderers abgelegt werden können. Dadurch ist es möglich, mehrere Eingabedatensätze für verschiedene Dienstaufträge in unterschiedliche Verzeichnisse abzulegen (Dateipfad).
- **Fehlernachricht:**
Tritt bei der Dienstanforderung ein Fehler auf, wird über den Rückgabewert eine Fehlernummer zurückgegeben. Da Fehlernummern im allgemeinen nicht sehr aussagekräftig sind, wird dem Dienstanforderer im Fehlerfall zusätzlich eine Fehlernachricht übergeben.

Der Dienstanforderer kann nun anschließend mit der Dienstinanspruchnahme fortfahren, oder sich die, aus der Dienstanforderung resultierten Werte speichern, und zu einem späteren Zeitpunkt die Dienstinanspruchnahme starten.

3.2.4.3 Dienstinanspruchnahme

In diesem Abschnitt wird gezeigt, wie ein Dienst nach einer erfolgreichen Dienstanforderung in Anspruch genommen werden kann. Bei der Dienstinanspruchnahme wird zwischen blockierender und nichtblockierender Ausführung unterschieden. Blockierend heißt, daß der Dienstanforderer wartet, bis der Dienst beendet ist und er die Ergebnisdaten erhält oder bis eine Fehlermeldung auftritt. Wird der Dienst nichtblockierend in Anspruch genommen, dann

muß der Dienstanforderer nicht bis zur Beendigung der Durchführung des Dienstes warten, sondern kann in der Programmausführung fortfahren und sich auch abmelden (d.h. das Programm beenden). Die Ergebnisse werden in diesem Fall vom Kommunikationsinterpreter auf einem permanenten Speicher zwischengespeichert und können zu einem späteren Zeitpunkt vom Dienstanforderer angefordert werden. Dazu wird vom Kommunikationsinterpreter eine Benachrichtigung an den Empfänger generiert. Die Benachrichtigung erfolgt entweder über eine RÜchmeldung über die CCI-Schnittstelle an das Slave-Mosaic des Benutzers, oder durch eine Mail. Ein RÜchmeldung über CCI kann nur dann erfolgen, wenn der Benutzer in einem Mosaic-Fenster den entsprechenden Port aktiviert hat. Ansonsten schickt der KIP dem Dienstanforderer eine Mail mit der Nachricht, daß die Ergebnisdaten vorliegen.

Wie schon erwähnt gibt es zwei mögliche Arten der Inanspruchnahme eines Dienstes:

- blockierend
- nichtblockierend

Bei der nichtblockierenden Inanspruchnahme eines Dienstes ergibt sich für die Zwischenspeicherung der Dienstinanspruchnahme eine Kombination von zwei möglichen Varianten:

- Dienstinanspruchnahme wird auf der Clientseite zwischengespeichert, um beispielsweise günstigere Übertragungsbedingungen zu einem späteren Zeitpunkt zu nutzen.
- Dienstinanspruchnahme wird auf der Serverseite zwischengespeichert, um den externen Dienst verzögert zu starten.

Als Default-Wert wird der Wert für die mögliche Art der Inanspruchnahme eines Dienstes, wie sie die Dienstanforderung zurückliefert, verwendet. Besteht dabei die Möglichkeit sowohl blockierend als auch nichtblockierend auf den Dienst zuzugreifen, erhält die blockierende Inanspruchnahme den Vorzug.

Nach der erfolgreichen Dienstanforderung erscheint, wie oben schon angesprochen, automatisch ein HTML-Dokument für die Inanspruchnahme dieses Dienstes. Dieses Dokument besitzt folgende Eingabewerte, die zum Teil als Ergebnis der Dienstanforderung vorgegeben werden (siehe Abbildung A3):

- Name des Dienstes (Dienstname).
- Nummer des Dienstauftrags (Auftragsbezeichnung).
Die Nummer wird bei einer erfolgreichen Dienstanforderung geliefert.
- Benutzerkennung zur eindeutigen Identifikation der Ergebnisdaten und Dienstaufträge (Benutzerkennung).
Die Benutzerkennung muß die gleiche sein wie bei der Dienstanforderung. Ansonsten kann der Dienstauftrag nicht ausgeführt werden.
- Tatsächliche Art der Inanspruchnahme (Art der Inanspruchnahme).
Man unterscheidet bei der Dienstinanspruchnahme vier Fälle, wobei in Abhängigkeit des Dienstes als Ergebnis sowohl nur eine als auch mehrere dieser vier Arten zurückgeliefert werden:
 - blockierend

- nichtblockierend auf der Clientseite zwischengespeichert
- nichtblockierend auf der Serverseite zwischengespeichert
- nichtblockierend auf der Client- und Serverseite zwischengespeichert.
- Anzahl der Dateien (Anzahl der Dateien).
Die Anzahl der Dateien zeigt an, wieviele Eingabedateien bei der Dienstanspruchnahme verwendet werden..
- Angabe ob der vorgeschlagene Pfad des Verzeichnisses, in dem sich die Eingabedateien für den Dienst befinden, übernommen werden soll oder nicht (Pfad).
- CCI-Port, über den die Meldung daß die Dienstanforderung beendet ist, geschickt wird (CCI-Port).
Der CCI-Port wird nur im Falle der nichtblockierenden Dienstanspruchnahme benötigt. Bei der blockierenden Dienstanspruchnahme wird der CCI-Port ignoriert.

Wurden die benötigten Angaben eingegeben, so erscheint nach Drücken des OK-Button ein weiteres HTML-Dokument (Abbildung A4 im Anhang A.6), in dem weitere, für die Dienstanspruchnahme benötigten Werte, eingegeben werden müssen. Hierbei handelt es sich um folgende Angaben:

- Zeitpunkt, zu dem eine auf der Clientseite zwischengespeicherte Dienstanspruchnahme weitergeleitet werden soll (Zeitpunkt zum Versenden).
Die Zeitpunktangabe hat das Format "ttmmjjjjhhxx". Die einzelnen Zeichen der Zeichenkette haben folgende Bedeutung:
tt: Tag
mm: Monat
jjjj: Jahr
hh: Stunde
xx: Minute

Beispiel: "101119952035" bedeutet z.B., daß der Start des Dienstes am 10.11.1995 um 20.35 Uhr erfolgt. Bei der Verwendung eines Datums aus der Vergangenheit wird die nichtblockierende Ausführung sofort durchgeführt. Bei der Verwendung eines Datums, welches länger als einen Monat voraus liegt, erfolgt die Ausführung spätestens nach einem Monat.

- Zeitpunkt, zu dem der Dienst gestartet werden soll (Zeitpunkt zum Starten).
Die Zeitpunktangabe für das Starten des Dienstes hat das gleiche Format wie der Zeitpunkt zum Versenden des Dienstes.
- Namen der Eingabedateien (Dateinamen).
Das Format der Eingabedatei ist durch den Dienst vorbestimmt. Diese Datei wird vom Kommunikationsinterpreter vom Dienstanforderer an den Dienst übermittelt.
- Angabe ob die Eingabedateien übernommen oder neu erstellt werden soll (Datei).

Werden schon existierende Eingabedateien übernommen, so erscheint nach drücken des OK-Buttons ein HTML-Dokument mit den zuvor eingegebenen Werten. Es besteht die Möglichkeit sie nochmals auf Richtigkeit zu überprüfen. Möchte man bestimmte Angaben ändern, so kann man mittels dem Back-Button zurückblättern und Änderungen durchführen. Durch Aktivieren des OK-Buttons wird anschließend die Dienstanspruchnahme gestartet.

Wählt der Benutzer den Punkt „neue Eingabedatei erstellen“, so gelangt er zu einem HTML-Dokument, mittels dem die Eingabedatei generiert werden kann. Die Form dieses HTML-Dokumentes ist dienstabhängig.

Beispiel: HTML-Seite zum Erstellen einer Eingabedatei für den Dienst Meros. (Abbildung A5 im Anhang A.6)

Nachdem eine oder mehrere Eingabedateien erstellt wurden, wird analog wie oben beschrieben die Dienstinanspruchnahme gestartet.

In Abhängigkeit der Art der Inanspruchnahme (blockierend oder nichtblockierend) erhält man unterschiedliche Ergebnisse zurück:

Fall I: blockierende Art der Inanspruchnahme

Als Ergebnis werden folgende Werte zurückgeschickt:

- Anzahl der Dateien (Dateianzahl).
Die Anzahl der Dateien gibt an, wieviele Ergebnisdateien eingetroffen sind.
- Nummer der Datei (Dateinamen)
Erhält man als Ergebnis einer Dienstanforderung mehrere Ergebnisdateien zurück, so kann man aus den Dateinummern die Reihenfolge, in der sie erzeugt wurden, rekonstruieren.
- Name der Dateien, mit den Ergebnisdaten für den Dienst (Dateiname:).
Das Format der Ergebnisdatei ist durch den Dienst vorbestimmt. Da diese Datei vom Kommunikationsinterpreter nur vom Dienst an den Dienstanforderer übermittelt wird, kann zu dieser im weiteren keine Aussage getroffen werden.
- Pfadname des Verzeichnisses, in dem sich die Ergebnisdaten des Dienstes in einer Datei befinden (Pfadname).
- Fehlernachricht.
Tritt bei der Dienstinanspruchnahme ein Fehler auf, wird über den Rückgabewert eine Fehlernummer zurückgegeben. Da Fehlernummern im allgemeinen nicht sehr aussagekräftig sind, wird dem Dienstanforderer im Fehlerfall zusätzlich eine Fehlernachricht übergeben.

Anschließend können die Ergebnisdateien mittels der "Ergebnisanforderung" angefordert werden.

Fall II: nichtblockierende Art der Inanspruchnahme

Als Ergebnis wird in diesem Falle nur eine Bestätigung, daß der gewünschte Dienstauftrag nichtblockierend ausgeführt wird, zurückgeschickt. Wurde der Dienst zu dem angegebenen Zeitpunkt erfolgreich in Anspruch genommen, so wird zuerst versucht über die CCI-Schnittstelle dem Dienstanforderer die Nachricht zu schicken, daß der Dienstauftrag erledigt ist und die Ergebnisse angefordert werden können. Kann der KIP keine Verbindung über die CCI-Schnittstelle aufbauen, so wird der Dienstanforderer mittels einer Mail benachrichtigt. Dies ist dann der Fall, wenn der Dienstanforderer keine oder eine falsche Portnummer in Slave-Mosaic eingestellt hat oder er einfach zum Zeitpunkt der Benachrichtigung nicht mehr in seinem Rechner eingeloggt ist.

3.2.4.4 Ergebnisanforderung

Nach einer erfolgreichen Dienstinanspruchnahme (blockierend oder nichtblockierend) muß der Dienstanforderer auf die Ergebnisdateien, die beim Auftragsverwalter zwischengespeichert sind, zugreifen können. Der Zugriff auf abgelegte Ergebnisdateien ist die Aufgabe der Ergebnisanforderung.

Für einen ordnungsgemäßen Aufruf der Ergebnisanforderung werden folgende Werte benötigt (siehe Abbildung A6 im Anhang A.6):

- Nummer des Dienstauftrags (Auftragsnummer).
Die Nummer wird bei einer erfolgreichen Dienstanforderung geliefert. Sie kann auch über die Anforderung der "Übersicht über aktive Dienstaufträge" ermittelt werden.
- Benutzerkennung zur eindeutigen Identifikation der Ergebnisdaten und Dienstaufträgen (Benutzerkennung).
Die Benutzerkennung muß die gleiche sein, wie in der Dienstanforderung angegeben wurde.
- CCI-Port, über den die Ergebnisdateien zum Dienstanforderer geschickt werden (CCI-Port).

Als Ergebnis der Ergebnisanforderungen erhält man folgende Werte im Master-Mosaic zurück:

- Anzahl der Dateien (Dateianzahl).
Die Anzahl der Dateien gibt an, wieviele Ergebnisdateien eingetroffen sind.
- Nummer der Datei (Dateinamen)
Erhält man als Ergebnis einer Dienstanforderung mehrere Ergebnisdateien zurück, so kann man aus den Dateinummern die Reihenfolge, in der sie erzeugt wurden, rekonstruieren.
- Name der Dateien, mit den Ergebnisdaten für den Dienst (Dateiname:).
Das Format der Ergebnisdatei ist durch den Dienst vorbestimmt. Da diese Datei vom Kommunikationsinterpreter nur vom Dienst an den Dienstanforderer übermittelt wird, kann zu dieser im weiteren keine Aussage getroffen werden.
- Pfadname des Verzeichnisses, in dem sich die Ergebnisdaten des Dienstes in einer Datei befinden (Pfadname).
- Fehlernachricht.
Tritt bei der Dienstinanspruchnahme ein Fehler auf, wird über den Rückgabewert eine Fehlernummer zurückgegeben. Da Fehlernummern im allgemeinen nicht sehr aussagekräftig sind, wird dem Dienstanforderer im Fehlerfall zusätzlich eine Fehlernachricht übergeben.

Zusätzlich schickt der KIP über CCI die Ergebnisdateien zurück. Man kann diese entweder im Slave-Mosaic visualisieren (Abbildung A7 im Anhang A.6), oder lokal zur weiteren Bearbeitung abspeichern.

3.2.4.5 Statusanfrage

Jedem Schritt des Dienstauftrages ist ein bestimmter Statuswert zugeordnet, so daß sich der Dienstanforderer jederzeit über die Ausführung eines Dienstauftrags informieren kann. Um den Status eines Dienstauftrags zu bestimmen, kann der Dienstanforderer an den Kommunikationsinterpret eine Anfrage richten.

Für einen ordnungsgemäßen Aufruf der Statusanfrage müssen folgende Werte angegeben werden (zugehörige Abbildung A8 im Anhang A.6):

- Nummer des Dienstauftrags (Auftragsnummer).
Die Nummer wird bei einer erfolgreichen Dienstanforderung geliefert.
- Benutzerkennung zur eindeutigen Identifikation der Ergebnisdaten und Dienstaufträge (Benutzerkennung).
Die Benutzerkennung muß die gleiche sein, wie in der Dienstanforderung angegeben wurde.

Als Ergebnis der Statusabfrage werden folgende Werte zurückgeliefert:

- Statusnummer des Dienstauftrags
- Statusmeldung
- Fehlernachricht
Tritt bei der Dienstinanspruchnahme ein Fehler auf, wird über den Rückgabewert eine Fehlernummer zurückgegeben. Da Fehlernummern im allgemeinen nicht sehr aussagekräftig sind, wird im Fehlerfall zusätzlich eine Fehlernachricht dem Dienstanforderer übergeben.

3.2.4.6 Übersicht über aktive Dienstaufträge

Wenn ein Benutzer mehrere Dienstaufträge gleichzeitig aktiv hat, kann er eine Übersicht über diese Dienstaufträge abrufen. Als Dienstanforderer kann er eine Anfrage an den Kommunikationsinterpret schicken, mit dem Wunsch, alle seine noch aktiven Dienste anzuzeigen.

Für einen ordnungsgemäßen Aufruf der "Übersicht über Dienstaufträge" muß folgender Wert angegeben werden (zugehörige Abbildung A9 im Anhang A.6):

- Benutzerkennung zur eindeutigen Identifikation der Ergebnisdaten und Dienstaufträgen (Benutzerkennung).
Die Benutzerkennung muß die gleiche sein, wie in der Dienstanforderung angegeben wurde.

Als Ergebnis der Dienstübersichtsabfrage werden folgende Werte zurückgeliefert:

- Anzahl der aktiven Dienstaufträge.

- Die Auftragsbezeichnungen, die der Dienstanforderer bei der Dienstanforderung den einzelnen Dienstaufträgen gegeben hat.
- Fehlernachricht.
Tritt bei der Dienstinanspruchnahme ein Fehler auf, wird über den Rückgabewert eine Fehlernummer zurückgegeben. Da Fehlernummern im allgemeinen nicht sehr aussagekräftig sind, wird im Fehlerfall zusätzlich eine Fehlernachricht dem Dienstanforderer übergeben.

3.2.4.7 Löschen von Dienstaufträgen

Wenn der Dienstanforderer an der Durchführung eines Dienstauftrags nicht mehr interessiert ist oder sich die Notwendigkeit ergibt, alte Dienstaufträge zu entfernen, kann er Dienstaufträge löschen. Der Dienstanforderer muß dazu einen Löschauftrag an den Kommunikationsinterpret er richten. Das Löschen von registrierten Dienstaufträgen ist immer erforderlich. Bis zum Löschen von registrierten Dienstaufträgen bleiben diese mit all ihren Daten (auch Ergebnisdaten) erhalten. Alle Informationen über den Dienstauftrag (z.B. Ergebnisdaten) müssen für eine Verwendung nach dem Löschen eines Dienstauftrags vom Dienstanforderer kopiert werden. In der Regel sollte ein Löschauftrag erst erfolgen, wenn die Ergebnisdaten nicht mehr verwendet werden, um diesen Kopiervorgang zu vermeiden.

Für einen ordnungsgemäßen Aufruf des Löschauftrags müssen folgende Werte angegeben werden (zugehörige Abbildung A10 im Anhang A.6):

- Nummer des Dienstauftrags (Auftragsnummer).
Die Nummer wird bei einer erfolgreichen Dienstanforderung geliefert.
- Benutzerkennung zur eindeutigen Identifikation der Ergebnisdaten und Dienstaufträge (Benutzerkennung).
Die Benutzerkennung muß die gleiche sein, wie in der Dienstanforderung angegeben wurde.

Wurde der Löschauftrag erfolgreich durchgeführt, so wird dies durch eine Meldung bestätigt. Konnte der Dienstauftrag nicht gelöscht werden, erscheint eine entsprechende Fehlermeldung.

3.2.4.8 Fehlermeldungen des Kommunikationsinterpreters

Im Anhang A.5 sind alle Fehlermeldungen aufgeführt und beschrieben, die der Dienstanforderer vom Kommunikationsinterpret erhalten kann. Jede Routine, die vom Dienstanforderer aufgerufen wird, gibt eine Fehlermeldung zurück. Aufgrund der Nummer ist es möglich, festzustellen, welche Fehlerbedingung vorhanden ist. Neben der Fehlernummer wird im Fehlerfall mit dem zurückgelieferten Formular zusätzlich eine Fehlermeldung mitgeliefert.

Im Falle eines Fehlers hat die Fehlernummer eine der nachfolgenden Bedeutungen. Nicht aufgeführte Fehler sind interne Fehler des Kommunikationsinterpreters. Anhand des

Fehlertextes ist es möglich, die Ursache festzustellen. Manche internen Fehler (z.B.: Speicherplatzmangel) können unter Umständen vom Anwender behoben werden.

3.3 Kooperatives Arbeiten mit Windows-Mosaic

Neben der Integration der CCI-Schnittstelle für die Nutzung von CCI über TCP/IP bietet Windows-Mosaic ab der Version 2.0 die Möglichkeit einer sogenannten "Collaborative Session". Diese ähnelt dem von Alan Braverman entwickelten Programm "X Web Teach", das es über die CCI-Schnittstelle einem "Teacher-Mosaic" erlaubt, mehrere "Student-Mosaic" zu steuern. Voraussetzung für den vollen Funktionsumfang ist die Verwendung von Windows für Workgroups 3.11 oder Windows 95.

Neben dem Vorteil, das sie in Mosaic selbst implementiert ist, bietet die "Collaborative-Session" allerdings noch wesentlich weitreichendere Möglichkeiten:

- Kommunikation beliebig vieler Teilnehmer über ein Konferenz-Fenster;
- Versenden von Dateien (Grafiken, Text, Sound) über "drag&drop" an alle Teilnehmer der Konferenz;
- Gemeinsames Betrachten von HTML-Dokumenten über <Send Link>- und <Follow Link>;

Vor **Beginn einer Konferenz** wählt man unter "Options/Preferences/Sevices" (Abbildung 3.9) einen sogenannten "Chat Alias", also einen Namen, unter dem man an der Konferenz teilnehmen will. Dieser erscheint einmal in der Teilnehmerliste, zum anderen wird er eigenen Mitteilungen im Konferenzfenster vorangestellt.

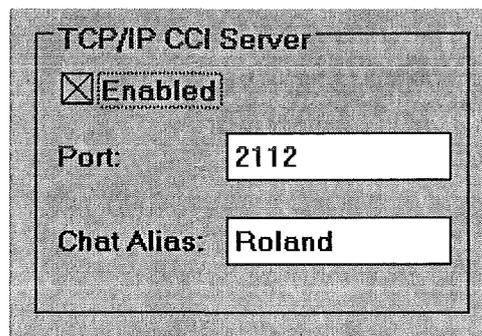


Abbildung 3.9 - CCI-Menü in Windows-Mosaic

Die beiden weiteren Auswahlpunkte (Port und Enabled) beziehen sich auf die Öffnung einer CCI-Verbindung und haben keinen Einfluß auf die "Collaborative Session", da eventuell bestehende CCI-Verbindungen vor Beginn der Konferenz beendet werden.

Um an einer Konferenz teilzunehmen oder um selbst eine zu starten, wählt man im Menü "File" den Unterpunkt "Collaborate....".

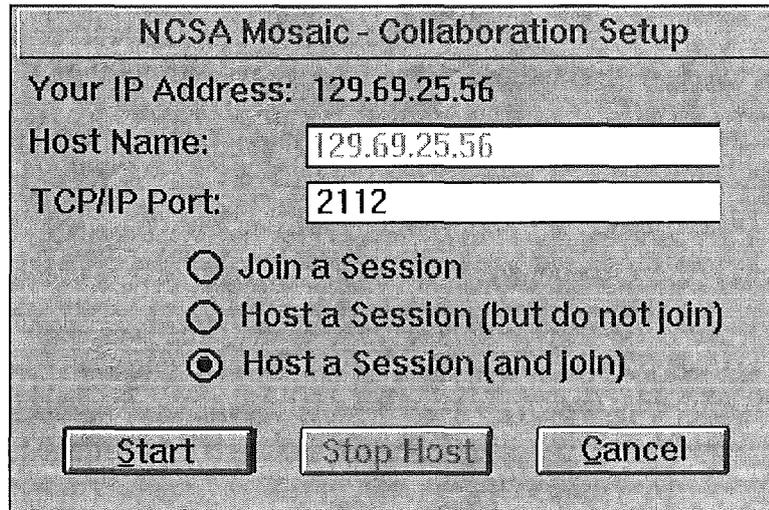


Abbildung 3.10 - Auswahlmenü für kooperatives Arbeiten

Hier stehen mehrere Möglichkeiten zur Auswahl:

- Mit "Join a Session" kann man an einer laufenden Konferenz teilnehmen. Dabei müssen Port und IP des Initiators in die entsprechenden Felder eingetragen werden.
- "Host a Session" ermöglicht den Start einer neuen Konferenz, an der man selbst gleich teilnehmen kann. IP-Adresse ist dabei automatisch die des eigenen Rechners.

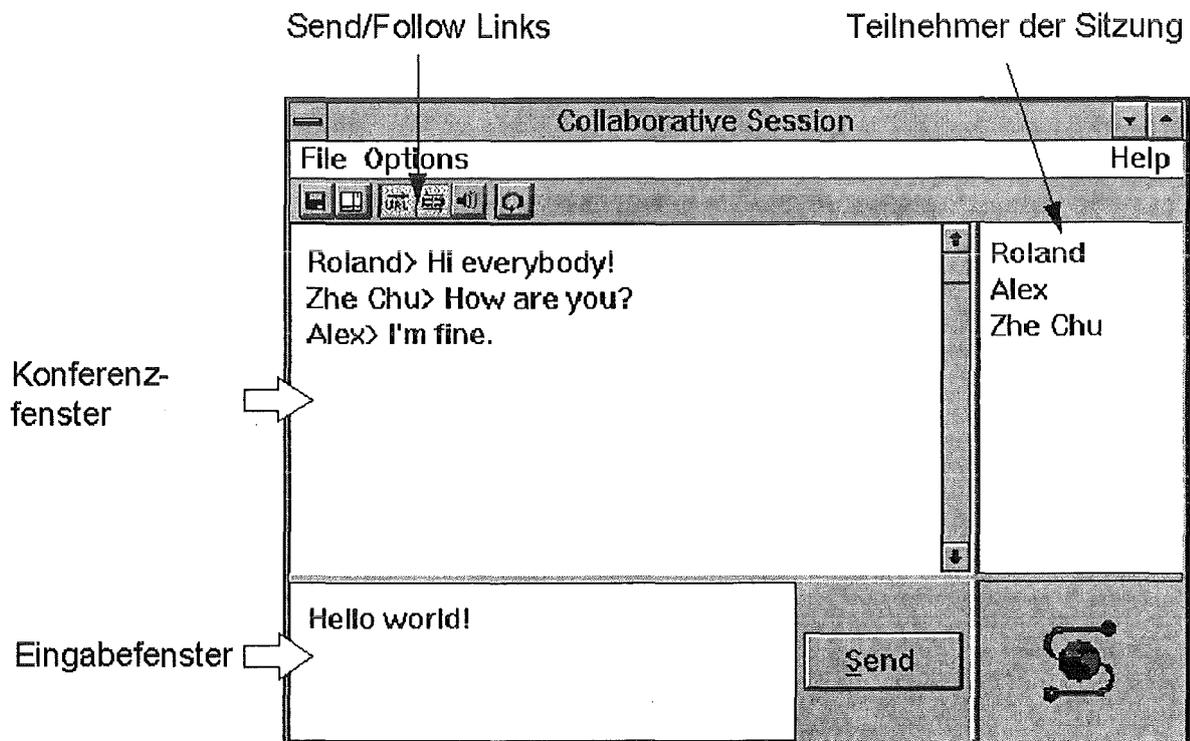


Abbildung 3.11 - Kooperationsfenster von Windows-Mosaic

Das Kooperationsfenster (Abbildung 3.11) gliedert sich in 3 Bereiche:

- Im Konferenzfenster werden die Beiträge der einzelnen Teilnehmer dargestellt, wobei jedem Beitrag der "Chat Alias" des Urhebers vorangestellt wird. Es ist in der Lage, die jeweils letzten 32.000 Byte zu speichern.
- Das Teilnehmer-Fenster listet alle an der Konferenz beteiligten Personen mit ihrem "Chat Alias" auf. Per Mausklick auf den jeweiligen Alias erhält man Auskunft über dessen vollständigen Namen, IP und E-Mail Adresse.
- Das Eingabefenster dient zur Eingabe der eigenen Beiträge, die nach Betätigung des "Send"-Button den anderen Teilnehmern zugeschickt werden.

Die **Menüleiste** bietet folgende Optionen:



"Save": Abspeichern des Inhalts des Konferenzfenster.



"Clear": Löschen des Inhalts des Konferenzfensters.



"Send Link": Aktiviert die Weitergabe der vom eigenen Mosaic aufgerufenen HTML-Seiten an alle Teilnehmer, die "Follow Link" aktiviert haben.



"Follow Link": Aktiviert die Option, in Mosaic HTML-Seiten darzustellen, die von Teilnehmern besucht werden, die "Send Link" aktiviert haben.



"Audio Notification": Aktiviert ein akustisches Signal, das einen neuen Teilnehmer ankündigt.



"Refresh User List": Baut die Teilnehmerliste neu auf.

Kommandozeilen-Optionen beim Start von Mosaic:

"-hostcollab"	startet eine Konferenz;
"-port xxx"	Port, auf dem die Konferenz gestartet werden soll;
"-collablimit N"	begrenzt die maximale Teilnehmerzahl einer Konferenz;
"-collabdroplimit N"	begrenzt die Größe von Dateien, die während einer Konferenz übertragen werden können;

Auch für **XMosaic** wird am NCSA von Stanford S. Guillory derzeit an einem "collab X tool" gearbeitet, das im wesentlichen alle Funktionen der Windows-Variante unterstützen wird. Damit wird ein kooperatives Arbeiten mit Datenaustausch über unterschiedliche Plattformen hinweg unter einer einheitlichen Oberfläche möglich.

4 Zusammenfassung

Das World-Wide-Web gewinnt als Informationssystem durch die Ausweitung des Angebots an Informationen immer mehr an Bedeutung. Um Zugang zu diesen Informationen zu erhalten, stehen dem Anwender eine Auswahl von Browsern zur Verfügung. Einer dieser Browser ist Mosaic, der am NCSA entwickelt wird.

Dieser Bericht untersucht die neue CCI-Schnittstelle, die erst seit kurzer Zeit in Mosaic implementiert ist. Hierbei wurde besonderer Wert auf die Darstellung der Unterschiede zwischen den Versionen für MS-Windows und X-Window sowie der dadurch möglichen Erweiterung der CGI-Schnittstelle gelegt.

Bei CGI handelt es sich um eine serverseitige Schnittstelle, die es einem HTTP-Server ermöglicht, über "Gateway"-Programme auf Informationen zuzugreifen und, für einen beliebigen WWW-Browser darstellbar, zur Verfügung zu stellen. Mittels CGI ist es nicht möglich, von diesen "Gateway"-Programmen vor der Beendigung der Anfrage Informationen zu erhalten oder interaktiv auf den Programmablauf Einfluß zu nehmen.

Ein Ansatz zur Beseitigung dieser Problematik ist CCI, eine bisher nur in Mosaic implementierte Schnittstelle, die es Applikationen gestattet, lokal oder verteilt auf die Funktionalitäten von Mosaic zuzugreifen und so an Informationen aus dem WWW zu gelangen. Zugleich gestattet die derzeitige CCI-Implementation, dem Anwender auf dem Umweg über ein (Slave-)Mosaic Meldungen und Zwischenergebnisse eines "Gateway"-Programms zu übermitteln. Das Problem der fehlenden Interaktion soll nach Aussage des NCSA mit der Mosaic-Version 2.7 beseitigt werden, in die eine neue Methode für die Übermittlung von FOF-Daten (<METHOD=CCIPPOST>) implementiert wurde.

Der wesentliche Unterschied zwischen Mosaic für Windows und X-Mosaic besteht in der Realisierung der CCI-Schnittstelle. Unter MS-Windows wurde CCI zum einen als OLE 2.0-Server integriert, zum anderen besteht die Möglichkeit einer Verbindung über TCP/IP. Die Kommunikation unter X-Window läuft ausschließlich über TCP/IP und einen CCI-Port, der in Mosaic separat eingestellt werden muß. Damit ist sowohl der lokale als auch der verteilte Zugriff auf Mosaic möglich, sobald der Port vom Anwender aktiviert wurde.

Insgesamt hat NCSA mit der CCI-Schnittstelle die Basis für eine breite Palette neuer Anwendungsmöglichkeiten geschaffen, mit denen voraussichtlich viele der heutigen Probleme zu lösen sein werden. Insbesondere für den Einsatz von Mosaic als graphische Benutzeroberfläche wurde mit CCI der Grundstein für künftige Entwicklungen gelegt.

Literatur

- [1] Bükler, Töns: Kreativer gestalten; c't 7/95; S.250 ff.
- [2] Bär, J.: Windows Intern 3.1, S. 580 ff; Data Becker Verlag, 1992.
- [3] Ousterhout, John K.: Tcl und Tk; Addison-Wesley GmbH; München, 1995.
- [4] Hofer, Thomas: Untersuchung von HTML im Bezug auf die Eignung zur Implementierung graphischer Benutzeroberflächen; Studienarbeit; 1995; S. 31 ff.
- [5] Kaltenbach, J.: Parallele und verteilte Simulation technischer Vorgänge; in: 3.Berichtskolloquium des GK PVS; IND; Stuttgart, 1995.
- [6] Bußmann, M., Schuch, A., Tischendorf, M., Schmidt, F.: Benutzerhandbuch Kommunikationsinterpreter, LFU Karlsruhe, Informationstechnisches Zentrum, Oktober 1995.

Visualisierung von Umweltdaten

*Joachim Wiesel, Wilhelm Hagg
Institut für Photogrammetrie und Fernerkundung
Universität Karlsruhe
Englerstraße 7
76128 Karlsruhe*

*unter Mitarbeit von
Heiko Jacobs
Ingenieurbüro Jacobs*

*Michael Böse, Claus Hofmann, Heiko Krassin
Institut für Photogrammetrie und Fernerkundung*

1 DIE ROLLE VON VIEWERN IN DER WWW-ARCHITEKTUR.....	241
1.1 KARTEN	241
1.2 BILDER	241
1.3 GESCHÄFTSGRAFIK	241
2 SERVER-BASIERENDE ANSÄTZE.....	242
2.1 ARCHITEKTUR	242
2.2 ENTWICKLUNGSUMGEBUNG UND SOFTWARE- KOMPONENTEN	244
2.3 DIE ANBINDUNG AN DAS WWW	245
2.4 MODULARCHITEKTUR.....	249
2.5 CGI - AKTIVITÄTSBEHANDLUNGSPROGRAMM	253
2.6 AUFBAU EINER SEITENGENERIERERFUNKTION.	255
3 KLIENTEN-BASIERENDE ANSÄTZE.....	257
3.1 HELFERANWENDUNGEN.....	257
3.2 MIME-MECHANISMUS	257
3.3 INTER-CLIENT-KOMMUNIKATION.....	259
4 DIE ROLLE VON ARCVIEW2	260
4.1 ARCVIEW 2.1 EIGENSCHAFTEN	260
4.2 ERWEITERUNGSMÖGLICHKEITEN VON AV2.....	261
4.2.1 Avenue - Übersicht.....	261
4.2.2 Avenue - Entwicklungswerkzeuge	262
4.2.3 Vor- und Nachteile der Entwicklung in Avenue	262
4.3 DATENZUGRIFFSFÄHIGKEITEN	263
4.4 DATENKOMMUNIKATIONSMECHANISMEN.....	263
4.4.1 DDE	263
4.4.2 RPC	263
4.4.3 System.....	264
5 VORSCHLAG FÜR EINE ARCHITEKTUR	264
6 GEODATENSERVER.....	264
6.1 BISHERIGE ARBEITEN	265
6.1.1 FAW	265
6.1.2 OGIS.....	265
6.1.3 GeoScope	268
6.1.4 Firmen: ESRI, Oracle, Computer Associates.....	269
7 ZUSAMMENFASSUNG UND AUSBLICK	269

1 Die Rolle von Viewern in der WWW-Architektur

Wie bereits aus verschiedenen Studien und Berichten bekannt (z.B. Globus-I, Integral) beruht die WWW-Architektur auf einem Client/Server-Modell. Als Client operiert einer der bekannten WWW-Browser (z.B. Arena, lynx, mosaic, netscape, hotjava), der bestimmte Dokumententypen auf dem Arbeitsplatzrechner anzeigen kann. In der Regel bestehen HTML-Dokumente aus einfachen oder speziell aufbereiteten Texten (z.B. Tabellen, Formularen) und Rasterbildern. Die WWW-Browser sind in der Lage, HTML-Dokumente, die bestimmten Normumfängen entsprechen, darzustellen. Da sich die HTML-Standardisierung rasch weiterentwickelt, kommt es zu Unverträglichkeiten, da die Entwicklungen der verschiedenen Browser mit dem Fortschreiten des Sprachumfangs nicht immer Schritt hält.

Farbige Rasterbilder werden zur Illustration von Texten oder als Schaltflächen verwendet. Alle Browser (außer lynx) können Rasterbilder im GIF-Format direkt darstellen.

Andere Dokumentenarten, wie z.B. Videofilme, Vektorbilder im DXF-Format, PostScript-Dokumente oder TIFF-Bilder lassen sich mit den genannten Browsern nicht darstellen.

1.1 Karten

Karten dienen im UIS (z.B. ALBIS) als Hintergrundinformation oder auch als das eigentliche Darstellungsziel, wenn beispielsweise thematische Kartenausgaben gewünscht sind. Kartendaten liegen in der Regel im RIPS-Pool abgespeichert vor. Die zu den Karten führenden Geodaten im RIPS-Pool sind teilweise als Vektordaten (mehrheitlich) aber auch als Rasterdaten vorhanden. Zur Verwaltung des RIPS-Pool wird die GIS-Software Smallworld-GIS eingesetzt. Geodaten können als ARC/Info GENERATE- (Vektorformat) und als TIFF-Dateien (Rasterformat) exportiert werden.

1.2 Bilder

Mit Bildern sind hier schwarzweiße oder farbige Halbtonbilder gemeint, die zum Beispiel durch Scannen von Photographien, Digitalisieren von Videobildern, Einspielen von Foto-CD oder durch Aufnahmen mit Stillvideo Kameras entstehen. Sie werden hauptsächlich zur Illustration von Sachverhalten benutzt (Beispiel: Foto eines geschützten Vogels).

1.3 Geschäftsgrafik

Unter Geschäftsgrafiken sind hier alle Darstellungen von Meßreihen (z.B. XY-Graphen), Häufigkeitsverteilungen (Balken-, Tortendiagramme) und ähnliche zusammengefaßt. Ge-

schäftsgrafiken müssen sowohl autonom aus Zahlentabellen herstellbar sein als auch in Kartenbilder eingebildet bzw. zusammenhängend dargestellt werden können. Sie dienen der Veranschaulichung von Meß- und Grenzwerten oder von Statistiken aller Art.

2 Server-basierende Ansätze

Da, wie schon dargestellt, WWW-Browser nur eine definierte Menge von Dokumententypen anzeigen können, müssen alle nicht konformen Dokumente in ein geeignetes, dem Browser bekanntes Format transformiert werden.

Ein Ansatz ist, alle Konvertierungen in einem ausreichend ausgestatteten Server vorzunehmen. In der folgenden Tabelle sind Vor- und Nachteile einer Server-zentrierten Lösung gegenübergestellt:

Vorteil	Nachteil
Einfache Administration	Hoher Ressourcenbedarf im Server
Geringe Softwarekosten	
Skalierbare Server sind an den Bedarf anpaßbar	
Standardbrowser verwendbar	Dokumentendarstellung nicht optimal in der Qualität
Niedrige Anforderungen an Arbeitsplatzrechner	Datenleitungen werden stark belastet
Vollständige Kontrolle über Dokumentenexport	Darstellungsgeschwindigkeit eher niedrig

2.1 Architektur

Das Architekturkonzept für die Anbindung eines Geoinformationssystems an eine Client-Server Applikation (WWW) ist in Abbildung 1 schematisch dargestellt. Auf der linken Seite der Abbildung ist der Zugriff auf verschiedene Datenbanksysteme mit punktuellen Daten in vereinfachter Form dargestellt, auf der rechten Seite befinden sich die Komponenten für den Zugriff auf die GIS-Daten. Dazwischen befinden sich die Komponenten zur Aufbereitung und graphischen Darstellung der Daten. In vertikaler Richtung sind zunächst WWW-Client und WWW-Server dargestellt. Der Server hat die Aufgabe, die Anwenderdienste für Sach- und GIS-Daten zur Verfügung zu stellen und die Ergebnisse der Dienste zum Client zu übermitteln. Die Anwenderdienste dienen der Bereitstellung der gesamten serverseitigen Funktionalität gegenüber dem Client. Sie bedienen sich hierzu einer Reihe von zur Verfügung stehenden Systemdiensten. So werden Sach- und Geometriedaten aus Datenbanken und Geoinformationssystemen extrahiert, dargestellt und miteinander kombiniert. Die GIS-Daten gliedern sich in die Geometriedaten, die in Form von Raster- oder komplexen Vektor-daten vorliegen, und

die Sachdaten, welche in Form von Datenbanktabellen oder Ascii-Dateien vorliegen. Der Bezug wird über einen eindeutigen Objekt-Index hergestellt. Die Sachdaten dienen in erster Linie der Erstellung von aussagekräftigen Legenden, können aber auch für komplexe Verarbeitungsmethoden herangezogen werden.

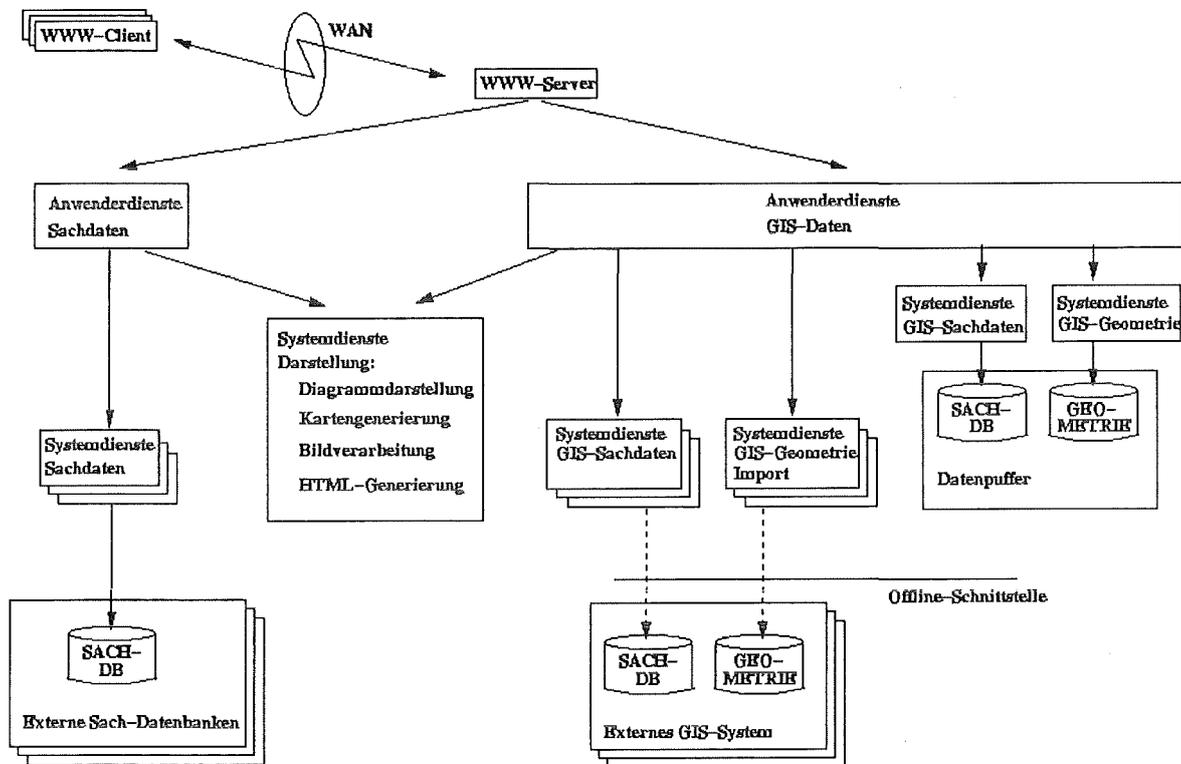


Abbildung 1: Architektur einer Anbindung von GIS im WWW

Die im Rahmen von WWW-UIS (Umwelt Informations System) verwendeten GIS-Daten stammen aus dem Geoinformationssystem Smallworld, in dem unter der Bezeichnung „RIPS“ (Räumliches Informations- und Planungs- System) ein großer Teil der in Baden-Württemberg vorhandenen Geoinformationen vorliegen. Bedingt durch eine noch nicht vorhandene Online-Schnittstelle zum GIS Smallworld erfolgt die Zwischenspeicherung der Geo-Daten in einem Datenpuffer, aus dem die relevanten Daten mittels der Systemdienste für GIS Sach- und Geometriedaten extrahiert werden. Die Anbindung von Smallworld erfolgt über eine Offline-Schnittstelle, wobei die Daten nach einem Export aus Smallworld mittels der Systemdienste für den Import in den Datenpuffer gelesen werden können. Die Methode der Datenpufferung ermöglicht aber auch einen effizienteren Datenzugriff durch den Einsatz mehrere lokaler Server, was angesichts der enormen Datenmengen (insbesondere bei Rasterdaten) und knapper Netzwerk-Ressourcen sicherlich nicht uninteressant sein dürfte.

Die Darstellung und Kombination verschiedenartiger Daten wie Karten und Business-Graphiken wird von den „Systemdiensten Darstellung“ übernommen. Diese sind auch für die Generierung der HTML-Seiten verantwortlich. Diese Systemdienste übernehmen die Aufbereitung der Daten, die Umsetzung in verschiedene Diagrammarten, die Erstellung von Karten mit mehreren Informationsebenen und die Kombination der Karten mit den Diagrammen. Sie bedienen sich hierzu externer Programme, die ein umfangreiches Repertoire an Methoden zur Verfügung stellen.

2.2 Entwicklungsumgebung und Software- Komponenten

Die Implementierung der einzelnen Module für die GIS-Anbindung im WWW-UIS erfolgt in TCL. TCL ist eine Interpretersprache mit einem relativ mächtigen Sprachumfang. Gegenüber herkömmlichen Shell- oder C-shell- Skripten existieren insbesondere verschiedene Datentypen wie Listen und Assoziative Arrays, umfangreiche Bibliotheken für die Behandlung von Listen, Strings und Dateizugriffen, und die Möglichkeit, weitere Funktionspakete wie ein Oracle-Interface in den Interpreter einzubinden. Insbesondere waren die folgenden Argumente für den Einsatz von TCL bei der Implementierung ausschlaggebend:

- Flexible Handhabung von Strings
- Einfacher Dateizugriff
- Einfache Portierung auf versch. Rechnerarchitekturen
- Keine Fehler durch dynamische Speicherallokierung
- Komfortable Fehlerbehandlung
- Keine zeitkritische Anwendung
- Aufgerufene externe Module sind ebenfalls Skripten

Im Rahmen der Realisierung des WWW-UIS wurden die folgenden externen public-domain Software-Komponenten verwendet:

Anwender- und Systemdienste

Sämtliche Anwender- und Systemdienste sind in TCL implementiert. Verwendet wurde TCL in der Version 7.3.

GIS-Funktionalität

Zur Erzeugung von Karten und für die Realisierung von standardisierten komplexen GIS-Operationen wurde GRASS in der Version 4.1 eingesetzt. GRASS ist ein rasterorientiertes Geoinformationssystem mit Funktionen zu digitaler Bildverarbeitung, Kartenerstellung und Vektordatenoperationen. Hardwarebasis für GRASS sind UNIX-Workstations. Entwickelt wurde das System von der US-Army, Corps of Engineers. Verantwortlich ist das GRASS Information Center in Champaign, Illinois, es bietet über Internet einen Support bei auftretenden Problemen an.

Generierung von Business-Graphiken

Für die Darstellung von Tabellen in Form von Business-Graphiken wird GNUPLOT in der Version 3.6 eingesetzt. Bei GNUPLOT handelt es sich um ein wissenschaftlich orientiertes Programm zur Darstellung von mathematischen Funktionen und Zahlenreihen. Die Anforderung die typischerweise an die Darstellung von Business-Graphiken gestellt werden, können daher nur zum Teil erfüllt werden. Eine Alternative zu GNUPLOT ist aber zumindest im public-domain Bereich nicht verfügbar.

Kombination von Karten und Graphiken

Für die Kombination von Karten und Business-Graphiken und zur Konvertierung verschiedener Bildformate wird das FLY -Tool von Martin Gleeson, welches auf der gd - Bibliothek von Tom Boutell basiert, eingesetzt. FLY ist ein Kommandozeilen- Interface, welches den Einsatz

der gd - Bibliothek im Rahmen von TCL erlaubt. Damit lassen sich umfangreiche Manipulationen an GIF-Bildern vornehmen, wie etwa die Einblendung anderer Bilder oder die Veränderung von Farben.

2.3 Die Anbindung an das WWW

Das WWW (World Wide Web) bietet die Möglichkeit hypertextartig zwischen einzelnen Seiten zu wandern. Eine Erweiterung dieses Konzeptes ist die Auswahl der Folgeseite durch aktivieren von Aktionen, die dynamische HTML-Seiten generieren. Zeichnet man alle darzustellenden Seiten und die möglichen Verbindungen zwischen diesen auf, so erhält man einen Graphen. Der Aufbau ist in Abbildung 2 angedeutet. Für den Einstieg in den Darstellungsgraphen (Aufruf der Darstellungsdienste) sind spezielle Dienstaufrufprogramme verantwortlich, die somit eine Schnittstelle nach außen bilden und zwischen Dienstaufruf und Realisierung vermitteln. Durch das sessionlose HTTP-Protokoll und die Aufteilung in WWW-Server und WWW-Client als zwei eigenständige Prozesse, die nur bei Aktivitätsauslösung kommunizieren, ergeben sich Probleme mit der session- und seitenübergreifenden Datenhaltung. Die einfachste Lösung der Datenhaltung über die Integration in die übertragene Seite (bei HTML: HIDDEN-Variablen) erweist sich bei der Anzahl von aufeinander aufbauenden Seiten als unhandlich. Um Daten auch nach Aktivitätsabgabe des WWW-Servers zu halten, arbeiten alle Seiten bzw. alle Seitengenerierer auf einer gemeinsamen Datenbasis, die in Abbildung 2 angedeutet ist.

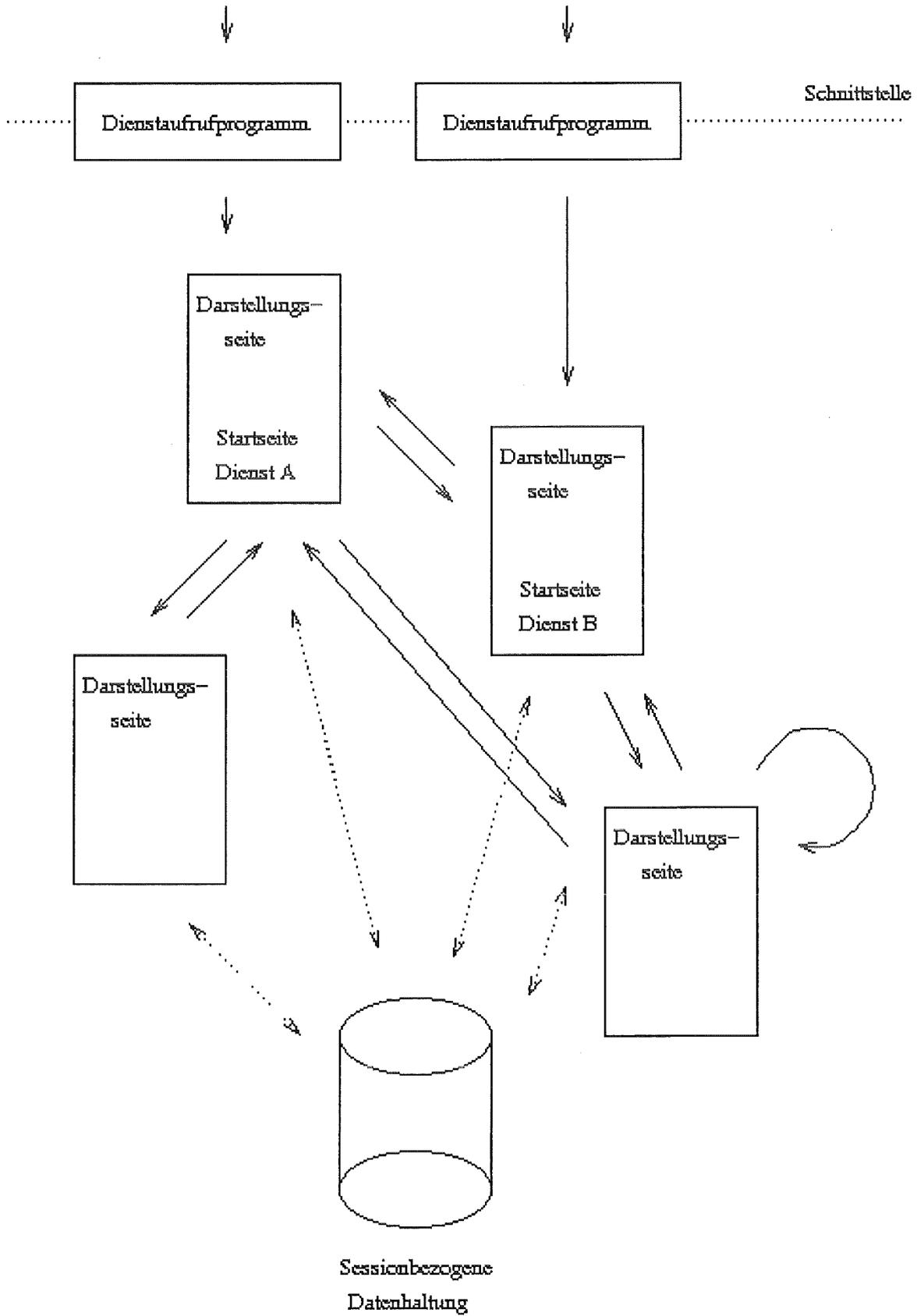


Abbildung 2: Datenbasis und Architektur der Seitengenerierer

Der Graph hat somit folgenden Aufbau:

- Knoten \Rightarrow Darstellungsseiten.

Zunächst also HTML-Seiten. Es sind aber auch andere Ausgabeformate wie Tk-Skripten zur Erzeugung von Darstellungsfenstern denkbar. Jede erzeugte Darstellungsseite des Graphen bekommt eine eindeutige PageId, die wie folgt aufgebaut ist:

HostId:UserId:SessionId:PageType:PageNr

- HostId

Ergebnisstring des 'hostname' Kommandos.

- UserId

String der den Benutzer oder auch eine Benutzergruppe beschreibt.

- SessionId

String, der bei Dienstaufwurf von den Dienstaufwurfprogrammen generiert wird. Nötig da einzelne Benutzer innerhalb von Benutzergruppen unterschieden werden müssen. Weiterhin kann ein Benutzer das System quasi parallel nutzen. Der String wird aus der ProzeßId des Dienstaufwurfprogramms und einem Zeitstempel ('date'- Kommando) im Sekundenbereich generiert.

- PageType

String, der den Seitentyp beschreibt. Jedes mögliche Seitenformular stellt einen eigenen Seitentyp dar.

- PageNr

Ganze Zahl, die die Anzahl der bislang zu diesem Seitentyp generierten Seiten angibt.

Sämtliche Id's dürfen keine Leerzeichen, Doppelpunktzeichen oder Kochkommata enthalten. Diese PageId wird in jede generierte Seite fest encodiert (bei HTML durch eine hidden Variable). Dadurch ist es möglich einen Sessionzustand bei jeder vom Benutzer ausgehenden Aktion zu berücksichtigen. Alle innerhalb einer Darstellungsseite veränderbaren Elemente/Widgets (Eingabefelder, Check- und Radiobuttons ...) bekommen ebenfalls einen im Graphen eindeutigen Namen, der sich wie folgt zusammensetzt: PageType.Widgetname Dieser "Pfadname" ist für alle Seiten und Elemente eindeutig zu definieren und wird auch in der Datenbasis verwendet.

- Kanten \Rightarrow Aktionen.

Betätigt der Benutzer einen Aktivitätsknopf¹ (HTML-Submitbutton), veranlaßt er die Ausgabe einer neuen Seite. Der WWW-Server startet dazu ein Programm das auf die Benutzeraktion reagiert und als Ausgabe eine neue Darstellungsseite erzeugt. Jede Aktion startet das gleiche CGI -Programm. Dieses Programm kann dann anhand des aktivitätsauslösenden Knopfes erkennen, welche Seite zu erzeugen ist.

Jede mögliche Darstellungsseite wird von einer einzigen nur für diese Seite zuständigen Funktion erzeugt. Alle Eingangskanten einer Seite werden somit zu einer Funktion zusammengefaßt. Das CGI-Programm ruft dann je nach Aktivitätsauslöser die entsprechende seiten-generierende Funktion auf, vergleichbar mit Callbacks in Toolkit-Terminologie.

Die Funktion kann anhand der ihr übergebenen PageId der aktionsauslösenden Seite erkennen, welche Aktivitäten nötig sind um die neue Seite zu generieren. Es kann sich alle Informationen, die es zum Erzeugen der neuen Seite benötigt werden, aus der Datenbasis besorgen. Die Datenbasis hält hierzu einen Datenbestand, der zu dem Benutzungsverlauf der zu generierenden Seite paßt, vor.

- Dienstaufwurfprogramme

¹ Seitenwechsel, die durch einen Link ausgelöst werden sind hier nicht berücksichtigt.

Die Dienstaufrufprogramme stellen eine Schnittstelle nach außen zur Verfügung und schaffen eine Systemumgebung in der die Aktionsprogramme laufen. Hier sollen Dienstübergabeparameter geparkt, die Datenbasis initialisiert werden. An dieser Stelle kann auch ein Persistenzmechanismus, der alte Sessions wieder herstellt, implementiert werden. Nachdem alle Vorbereitungen getroffen sind, ruft dieses Programm das Aktivitätsprogramm auf, das die Startseite des jeweiligen Dienstes erzeugt, so als würde der Aufruf aus dem Darstellungsgraphen erfolgen.

- Datenbasis

Die Datenbasis realisiert einen "session-bezogenen Datenpuffer". Da der WWW-Server nach einer Seitenerzeugung den Aktivitätsfaden abgibt, müssen alle bei der neuen Seitengenerierungen benötigten Daten in Dateien zwischengelagert werden. Die Datenbasisdatei wird unter dem Namen `UserId.SessionId.db` im jeweiligen Sessionverzeichnis des Benutzers angelegt. Durch die BACK-, FORWARD- und CLONE-Problematik der WWW-Clients entstehen Probleme mit der Konsistenz der Datenbasis. Unterschiedliche Seiten können sich auf unterschiedliche Datenzustände beziehen, da durch diese Clienteingenschaften mehrere gleichzeitige Pfade durch den Seitengraph möglich sind. Dieser Benutzungsablauf kann ebenfalls in einem Graph verdeutlicht werden. Der Benutzer befindet sich zuerst auf der Startseite, d.h. der Graph besteht zu diesem Zeitpunkt aus nur einem Knoten. Bei jeder Aktivitätsauslösung des Benutzers kommt ein neuer Knoten hinzu, da eine neue Seite mit eindeutiger PageId erzeugt wird. Drückt der Benutzer die BACK Taste, so wird ein Knoten aus dem Graphen entfernt. Die CLONE Taste kann man sich als Kopie des gesamten Graphen vorstellen, die parallel zu dem Ausgangsgraphen existiert. Um diese Problematik zu lösen werden die Daten unter der jeweiligen PageId gespeichert und bei Bedarf kopiert. Jede Seite hat demnach Ihren eigenen Datenbasiszustand, der dem Benutzungsverlauf entspricht.

Der Zugriff auf die Daten in der Datenbasis erfolgt über einen Datenschlüssel (Variablennamen). Es gibt zwei Datentypen:

- globale Daten/Variablen

Global bezieht sich hier auf den zeitlichen Verlauf der Session. Globale Daten sind unabhängig von dem Benutzungsverlauf. Es sind Daten die für alle Seiten gleichzeitig gelten und innerhalb einer Session nicht veränderbar sein sollen. Dies können z. B. Dienstaufrufparameter sein. Aber auch Aktivitätsprogramme oder Hilfsprogramme können hier Zwischenergebnisse, die für alle Zeitverläufe gültig sind, ablegen. Globale Daten werden in der Datenbasis nicht bezogen auf eine PageId und somit nicht auf einen speziellen Benutzungsablauf abgespeichert.

- lokale Daten/Variablen

Lokal bezieht sich hier auf den zeitlichen Verlauf der Session. Dieser Datentyp wird immer bezogen auf einen bestimmten Benutzungsablauf anhand der PageId abgespeichert. Veränderliche Daten wie Widgeteinstellungen oder nicht sichtbare Zusatzinformationen für eine spezielle Seite sind Beispiele für diesen Typ.

Lokal und Global bezieht sich hier also nicht, wie sonst üblich, auf die Datenkapselung einzelner Funktionen/Seitengenerierer, sondern nur auf die jeweilige Sicht auf den Datenzustand. Die Abbildung 3 verdeutlicht die Sichtbarkeit von Variablen in dem Benutzungsgraphen. Die Seitennummer entsprechen den generierten Seiten und haben nichts mit dem Seitentyp zu tun. Beispielsweise können die Seiten 1 und 2 vom gleichen Seitentyp sein, wenn die Seite sich selbst aufrufen kann.

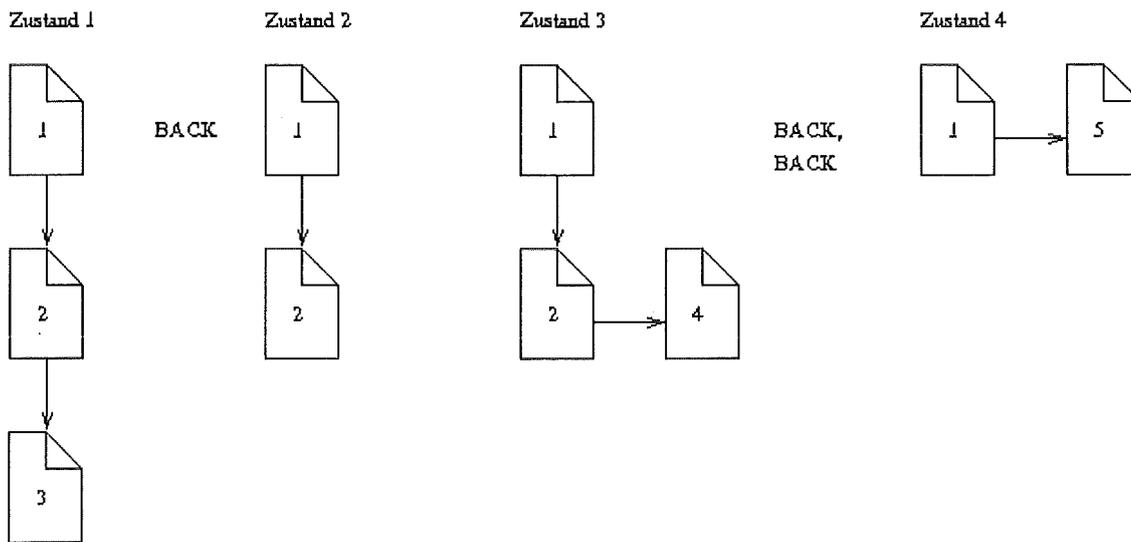


Abbildung 3: Seitzustandsdiagramme

Globale Variablen sind in gesamten Graphen, also auf den Seiten 1 bis 5, sichtbar. Lokale Variablen sind nicht überall sichtbar. Nur Variablen von Seiten, die auf dem direkten Benutzungspfad zwischen Startseite und aktueller Seite definiert wurden, sind auf der aktuellen Seite sichtbar. Folgende Beispiele verdeutlichen das Verhalten:

- In Zustand 1 kennt die Seite 2 nur Variablen von sich und der Seite Nummer 1. Seite 3 kennt alle Variablen.
- In Zustand 3 kennt die Seite 4 nur die Variablen der Seiten 1, 2 und 4. Nicht jedoch Variablen von Seite 3. Seite 3 ist aber immer noch in der Datenbasis vertreten, da durch cache-Mechanismen der Clients von dieser Seite immer noch Aktivitäten ausgehen können.

2.4 Modularchitektur

Die implementierten Karten- und Darstellungsdienste sind modular aufgebaut. Dies hat den Vorteil, daß das gesamte Paket übersichtlich gestaltet werden kann, was die Software-Pflege und die Fehlersuche in einem derart komplexen System wesentlich erleichtert, wenn nicht sogar überhaupt erst ermöglicht. Außerdem wird durch die mehrmalige Verwendung von Code in den einzelnen Funktionen die Fehleranzahl insgesamt reduziert. Erweiterungen sind so jederzeit mit tragbarem Aufwand vorzunehmen ohne die gesamte Software umbauen zu müssen. Bei Erweiterungen sind lediglich die neuen Module auf ihre Funktionalität hin zu testen, der bereits bestehende größte Teil kann ohne aufwendige Tests übernommen werden. Damit ist auch die Austauschbarkeit einzelner externer Softwaremodule, wie zum Beispiel die Erzeugung der Business-Graphiken oder das verwendete GIS, hinreichend gewährleistet.

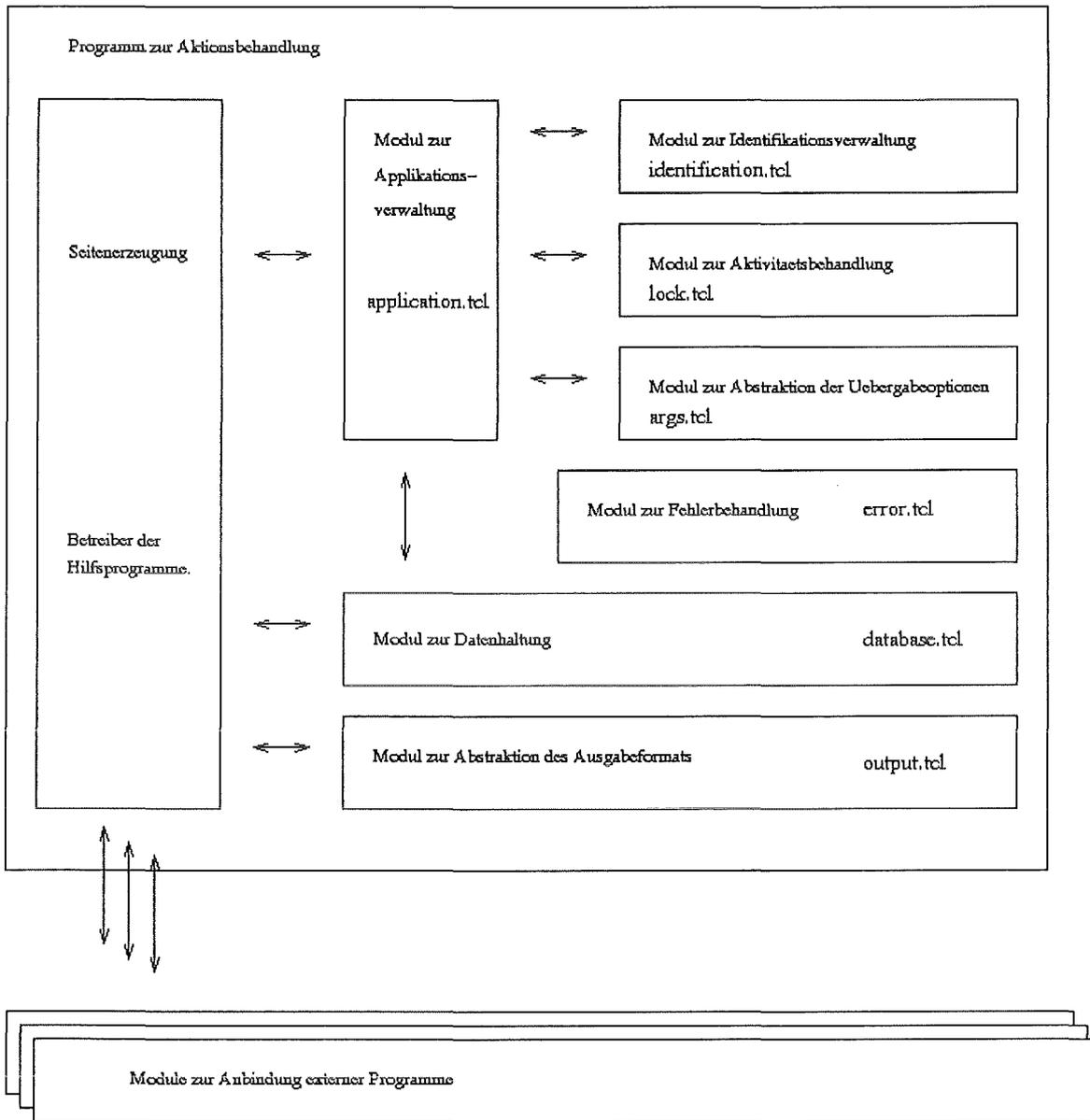


Abbildung 4: Modulübersicht im Server

Module zur Anbindung externer Programme

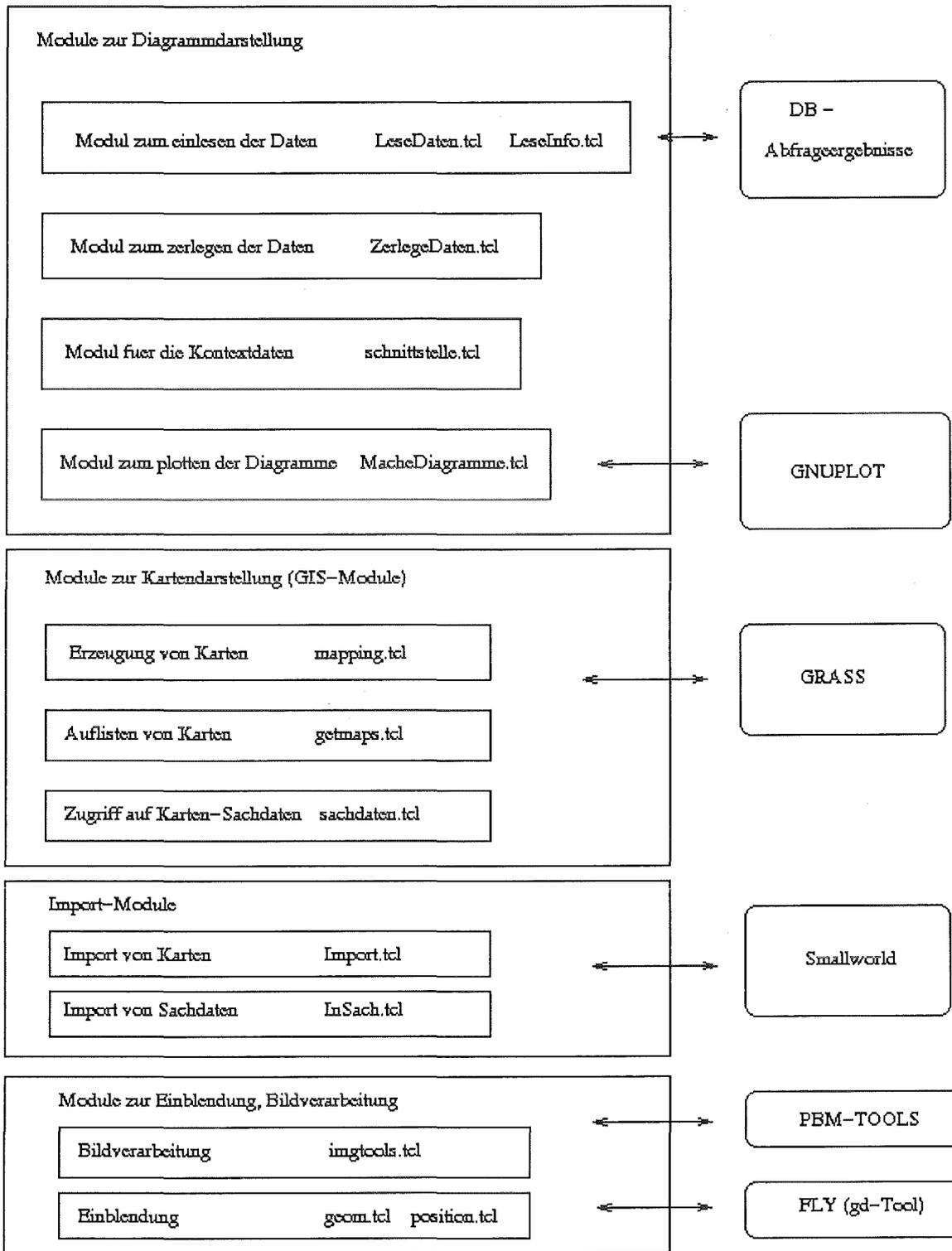


Abbildung 5: Anbindung externer Programme an den WWW-Server

Eine Übersicht über die Module ist in Abbildung 4 dargestellt. Die Module zur Anbindung externer Programme sind in Abbildung 5 dargestellt. Im einzelnen erfüllen die Module die folgenden Aufgaben:

Modul zur Applikationsverwaltung *application.tcl*

Das Modul gruppiert Aufgaben, die in allen Programmen gleichartig sind. Alle geparsten Übergabewerte werden in die Datenbasis übernommen. Der Seitengenerierer arbeitet somit immer auf der Datenbasis und parst die Übergabeoptionen nicht selbst.

Modul zur Identifikationsverwaltung *identification.tcl*

In dem gesamten System müssen bestimmte Objekte durch die Vergabe von Identifikationen gekennzeichnet werden. In diesem Modul sind alle Funktionen die Identifikationen erzeugen oder auswerten zusammengefaßt.

Modul zur Aktivitätsbehandlung *lock.tcl*

Um auszuschließen, daß zwei Aktivitäten eines Benutzers gleichzeitig ablaufen (möglich z. B. durch das Clonen von Fenstern), wird die Seitenerzeugung in einem gesperrten Bereich gemacht. Die erste Handlung eines Aktionsprogramms versucht im Session-Verzeichnis des Benutzers eine "lock"-Datei anzulegen. Gibt es diese Datei noch nicht, ist keine andere Aktivität am laufen und die Seitenerzeugung kann fortgesetzt werden. Ist diese Datei jedoch schon vorhanden, wird eine Informationsseite für den Benutzer generiert, und das Programm beendet. Der letzte Befehl des Aktionsprogramms hebt die Sperrung wieder auf, indem er die "lock"-Datei löscht.

Modul zur Abstraktion der Übergabeoptionen. *args.tcl*

Dieses Modul hat den Zweck das tatsächliche Übergabeformat vor der Seitenerzeugung zu verbergen. In dem Modul wird die Übertragungsart (GET,POST) behandelt. Auch das Format der Optionen wird hier umgesetzt ("option=wert" oder "option wert" oder ...). Falls später ein anderes Ausgabeformat gewünscht wird, kann der dort benötigte Übergabemechanismus unabhängig von der Seitenerzeugung geändert werden.

Modul zur Datenhaltung *database.tcl*

Dieses Modul hat die Aufgabe Daten zwischen Aktivitätswechseln zu speichern und den Seitengenerierern eine konsistente Sicht auf die Daten zu geben.

Modul zur Fehlerbehandlung *error.tcl*

Funktionen zum behandeln von internen Ausnahmezuständen sowie Benutzerfehlern. Diese Funktionen arbeiten auf Fehlernummern, um Sprach- und Formulierungsunabhängig zu bleiben.

Modul zur Abstraktion des Ausgabeformats *output.tcl*

Bietet Funktionen zum abstrakten Umgang mit Eingabeelementen und Formaten an. Die Routinen bewirken die Ausgabe auf Stdout im HTML-Format. Sie können in beliebiger Reihenfolge verwendet werden.

Module zum einlesen von *DatenLeseDaten.tcl LeseInfo.tcl*

Die Routinen lesen die Dateien, welche als Ergebnis einer Datenbank-Abfrage in Form von Diagrammen dargestellt werden sollen, ein. Durch eine mögliche Anpassung dieses Moduls sind beliebige Datenformate möglich.

Modul zur Aufbereitung der Daten *ZerlegeDaten.tcl*

Das Modul zerlegt die Eingabedaten entsprechend den Anforderungen an die zu erstellenden Diagramme. Ziel ist ein einfaches Dateiformat, welches von Programmen zur Diagrammdarstellung direkt gelesen werden kann.

Modul für die Erfragung von Kontextdaten *Schnittstelle.tcl*

Das Modul enthält alle Routinen welche die Kontextdaten für die Diagrammdarstellung erfragen. Dies sind sowohl Eingaben des Benutzers, als auch Informationen über die Daten selbst.

Modul zum erzeugen der Diagramme *MacheDiagramm.tcl*

Verschiedene Funktionen zum erzeugen von Business-Graphiken aus den zuvor aufbereiteten Daten. Das Modul stellt die Schnittstelle zu GNUPLOT dar.

Modul zur Erzeugung von Karten *mapping.tcl*

Das Modul ist für die Erzeugung der Karten verantwortlich und bildet die Schnittstelle zu GRASS.

Modul zum Auflisten von Karten *getmaps.tcl*

Funktionen, die für die Ausgabe von Kartenlisten erforderlich sind. Es werden jeweils die Karten der verschiedenen Kartentypen aufgelistet.

Modul für den Zugriff auf Karten-Sachdaten *sachdaten.tcl*

Das Modul enthält alle Funktionen die Sachdaten zu Karten bereitstellen. Durch die Anpassung dieses Modul können Sachdaten aus beliebigen Quellen entnommen werden.

Module für den Import *Import.tcl InSach.tcl*

Die Module enthalten verschiedene Funktionen für den Import von Sach- und Kartendaten. Als Eingabeformat wird das von Smallworld standardmäßig exportierte Format bezüglich aller Datenarten erwartet. Die Daten werden dann in ein internes Format (GRASS bei Kartendaten) gewandelt.

Modul zur Bildverarbeitung *imgtools.tcl*

Allgemeine Bildverarbeitungs-Funktionen (Formatwandlung, ...)

Module zur Einblendung von Diagrammen *geom.tcl position.tcl*

Gundlagenfunktionen zur Rechteckflächenverwaltung. Funktionen zum Positionieren der Diagrammbilder innerhalb und um die Karte.

2.5 CGI - Aktivitätsbehandlungsprogramm

Bei jeder Aktion die vom Benutzer ausgelöst wird, wird immer das gleiche CGI (Aktionsbehandlungsprogramm) vom WWW-Server gestartet. Dieses Programm parst zunächst die Übergabeparameter und liest die Datenbasis ein. Weiterhin werden globale Variablen initialisiert und die neue Seitengenerierung vorbereitet. Die Übergabe muß (per Definition) einen Parameter `pagetype.submit` enthalten. Von diesem Knopf ist die Aktivität des Benutzers ausgegangen. Der Wert des Parameters enthält die Knopfaufschrift. Anhand dieser Aufschrift entscheidet das Programm welche Seitengenerierfunktion aufgerufen werden soll.

Um eine neue Seite zu erzeugen, muß man also eine neue Funktion schreiben und durch einen Knopf auf anderen Seiten aufrufbar machen.

```
#!/usr/local/bin/tcl
# -----
# Copyright 1995 Universitaet Karlsruhe,
#           Institut fuer Photogrammetrie und Fernerkundung (IPF)
#
# CGI      : gis
# Date    : 26.8.95
# Author  : Claus Hofmann
#
# Das ist das General-CGI. Alle Benutzeraktionen starten dieses Programm.
# -----
#
# -----
# TCL und Module initialisieren
#
# Umgebungsvariable WWWUIS_HOME muss gesetzt sein.
#
if ![info exists env(WWWUIS_HOME)]
    # Fehlermeldung auf stderr und stdout.
    #
    puts stderr "ERROR: Umgebungsvariable WWWUIS_HOME nicht definiert!"

    puts stdout "Content-type: text/html

        <HTML>
        <HEAD>
        <TITLE>ERROR</TITLE>
        </HEAD>
        ERROR: Umgebungsvariable WWWUIS_HOME nicht definiert!
        </BODY>
        </HTML>
    "

# Eigene TCL-Libraryfunktionen und Seitengeneriererefunktionen bekannt machen.
#
set auto_path "$env(WWWUIS_HOME)/SRV_CGI/GIS/TCL_LIB $auto_path"
set auto_path "$env(WWWUIS_HOME)/SRV_CGI/GIS/TCL_PAGES $auto_path"

# globale assoziative Arrays definieren
#
global App AppErr ArgArr ResDB

# Application initialisieren:
#
# 1) Datenbasis initialisieren und einlesen.
# 2) Uebergabeparameter parsen
# 3) Folgende Verwaltungsvariablen zum allgemeinen Gebrauch initialisieren:
#
# App(OldPageId)      : PageId des Aufrufers.
# App(OldPageType)   : PageType des Aufrufers
# App(OldFormId)     : FormularId des aktionausloesenden Formulars.
# App(HostId)        : Hostname
```

```
# App(UserId)      : UserId
# App(SessionId)  : SessionId
# App(NewPageId)  : Neue, fuer diese Seitengenerierung erzeugte PageId.
# App(NewPageType) : PageType der neu zu generierenden Seite.
# App(SessionDir) : Session Verzeichnispfad
# App(CgiUrl)     : http-Pfad der CGIs.
#
AppInit

# =====
# Gewuenschten Seitengenerierer aufrufen.
#

# Callback aus den Uebergabeparametern ermitteln
#
set NextPage $ArgArr($App(OldPageType).submit)

# Seitengeneriererefunktion aufrufen.
#
if [catch $NextPage "" res]
    PageError $res

# =====

# -----
# Datenbasis in Datei schreiben; Lock-Datei loeschen
#
AppExit
```

2.6 Aufbau einer Seitengeneriererefunktion.

Die Seitengeneriererefunktion wird von dem Aktionsbehandlungsprogramm aufgerufen. Sie hat genau einen Uebergabeparameter, in dem ein ev. Statustext uebergeben wird. Wird die Funktion direkt von dem Hauptprogramm aufgerufen, ist dieser Statustext leer. Ruft man jedoch die Funktion aus einer anderen Seitengeneriererefunktion auf, so sollte in dem Statustext der Grund des Ausnahmeaufrufs enthalten sein. Der Statustext ist hauptsaechlich fuer die Ausgabe von Fehlermeldungen und Benutzungsanweisungen gedacht. Ein Beispiel soll den Zweck einer Statuszeile verdeutlichen: Ein Benutzer hat eine Aktivitaet ausgeloeset, ohne zuvor in einer Listbox einen Eintrag auszuwaehlen. Die gestartete Seitengeneriererefunktion weiss dadurch z. B. nicht auf welchen Daten sie arbeiten soll und kann die neue Seite somit nicht erzeugen. In diesem Fall wird die Aktivitaetsausloesende Seite erneut aufgerufen und die Fehlbenutzung mit Hilfe der Statuszeile dem Benutzer angezeigt. Das folgende Skript zeigt den Rahmen einer Seitengeneriererefunktion.

```
# -----
# Copyright 1995 Universitaet Karlsruhe,
#           Institut fuer Photogrammetrie und Fernerkundung (IPF)
#
# Seite :
# Date  :
# Author:
```

```
#
# -----
#
proc template status

global App

# Seite und Datenbasis initialisieren
#
# _____ Hier PageType dieser Seite angeben = App(PageType)
#
AppPageInit xx

# =====
# Auf folgende initialisierten Variablen kann jetzt zugegriffen werden:
#
# App(ArgCount)      : Anzahl der uebergebenen Argumentpaare
#                    (einschiesslich PageId und FormId).
# App(OldPageId)     : PageId des Aufrufers.
# App(OldPageType)   : PageType des Aufrufers.
# App(HostId)        : HosrId
# App(UserId)        : UserId
# App(SessionId)     : SessionId
# App(SessionDir)    : Verzeichnispfad des Session-Verzeichnis.
# App(CgiUrl)        : http-Pfad der CGIs.
# App(PageId)        : Neue, fuer diese Seitengenerierung erzeugte PageId.
# App(PageType)      : PageType der neu zu generierenden Seite.
# App(StdActionURL) : Default CGI
# App(Language)      : Ausgabesprache (derzeit d=deutsch)
# =====
#

# =====
# Seitengenerierung
#
PageStdHead Templateseite

# Ev. Statuszeile erzeugen
#
if $status != ""
    PageStatus $status

PageEnd

# =====
# end template
```

3 Klienten-basierende Ansätze

Nahezu alle WWW-Browser sind über verschiedene Konzepte erweiterbar. Dokumente, die sie nicht „kennen“ übergeben sie an geeignete Helfer-Anwendungen (Helper applications), die für die weitere Präsentation des Dokumenteninhalts verantwortlich sind. Die Browser (z.B. Mosaic, Netscape) sind nur noch für den Transport der das Dokument enthaltenden Dateien zuständig; sie geben nach dem erfolgreichen Dateitransfer die Kontrolle komplett an die Helfer-Anwendungen ab.

3.1 Helfer-Anwendungen

Typische Helfer-Anwendungen sind unter dem Betriebssystem MS-Windows:

telnet	Interaktive Session mit VT100/200 Protokoll
gsvieiw	Darstellung von PostScript-Dokumenten
lview	Darstellung von TIFF, BMP, RAS, TGA, JPEG Rasterbildern
qws3270	Interaktive Session mit dem IBM 3270 Protokoll
acroread	Adobe PDF Reader

Typische Helfer-Anwendungen sind unter dem Betriebssystem Unix:

telnet	Interaktive Session mit VT100/200 Protokoll
ghostview	Darstellung von PostScript-Dokumenten
xv	Darstellung von TIFF, BMP, RAS, TGA, JPEG Rasterbildern
x3270	Interaktive Session mit dem IBM 3270 Protokoll
xdvi	TeX Device Independent Datei darstellen
acroread	Adobe PDF Reader

3.2 MIME-Mechanismus

Wie erkennt ein Browser, welche Anwendung für einen bestimmten Dokumententyp zuständig ist?

In Ermangelung einer besseren Lösung haben sich die WWW-Architekten des bereits aus dem E-Mail-Bereich bekannten MIME-Mechanismus bedient (Multipurpose Internet Mail Extensions, Internet Standards RFC 1049, RFC 1341, RFC 1521, RFC 1522). Dieser basiert u.a. darauf, daß die Dateiendung einer Dokumentendatei eindeutig die enthaltene Dokumentenart beschreibt. Beispielsweise führt dies dazu, daß alle Dateien mit dem Suffix *.eps* als *Encapsulated Postscript* identifiziert werden, während *.dxf* eine Datei im *AutoCAD Drawing eXchange Format* bezeichnet.

In der folgenden Tabelle sind einige gängige MIME-Zuordnungen zusammengefaßt:

Suffix	Unix-Anwendung	Windows-Anwendung	Dokumentenart
.ps	gs	gs	PostScript
.eps	gs	gs	Encapsulated PostScript
.tif	xv	lview	TIFF-Datei
.wav	play	Medienwiedergabe	Sound im WAV-Format
.mpg	mpeg_play	Medienwiedergabe	Videofilm im MPEG-Format
.jpg	xv	lview	Komprimiertes Rasterbild im JPEG-Format
.hpg	xhpgl	??	Zeichnung im HP/GL Format
.dxf	??	CorelDraw4/5	Zeichnung im AutoCAD Drawing Exchange Format
.pdf	acroread	acroread	Adobe Acrobat Dokument (Portable Document Format)

Was passiert, wenn dem Klienten ein unbekanntes Dokument angeboten wird?

In diesem Fall greift eine Fallback-Routine ein, die einfach das Dokument binär in das Dateisystem des Klienten (nach Rückfrage) überträgt.

Wie wird die MIME-Zuordnung von Dokument zu Anwendung definiert? Dies ist abhängig von der Betriebssystemumgebung. In der Unix-Welt sind dafür Zuordnungstabellen in Textdateien (mailcap-Dateien) verantwortlich. Sie sind in der Regel in bestimmten Systemverzeichnissen und in den HOME-Verzeichnissen der Nutzer abgelegt. Beispielsweise sieht eine typische Eintragung für die Zuordnung des Adobe Acrobat-Readers zu einem PDF-Dokument so aus:

```
application/pdf; acroread %s
```

Dies bewirkt, daß das Programm `acroread` mit dem Namen einer Datei als Argument (`%s`) gestartet wird, wenn ein Dokument in einer Datei mit der Endung `.pdf` übergeben wird.

Im MS-Windows Betriebssystem kann die Zuordnung nur applikationsspezifisch vorgenommen werden, da es Unix-äquivalente Standardisierung von MIME dort nicht gibt. Die einzelnen Anwendungen benutzen deshalb für die Zuordnung von Dokumenten zu Anwendungen eigene Zuordnungstabellen (in der Regel private Initialisierungsdateien). Wünschenswert wäre eine OLE-Einbettung, die aber bisher von keinem Browser angeboten wird (allerdings scheint der NCOMPASS-Prototyp dies in rudimentärer Form zu unterstützen).

Komfortablere WWW-Browser bieten eine Menü-geführte Konfiguration von Helfer-Anwendungen an (z.B. wie in Abbildung 6 mit Netscape gezeigt).

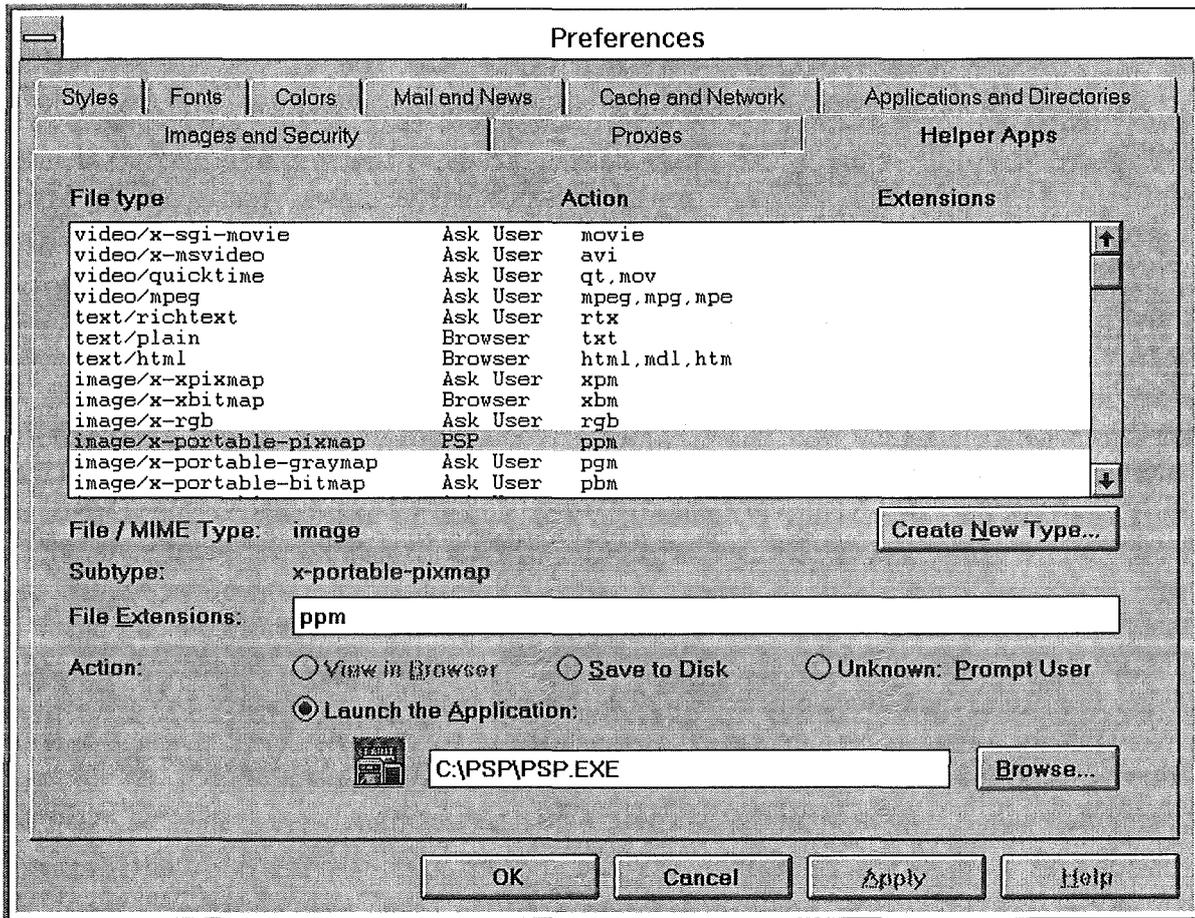


Abbildung 6: Beispiele für Helferapplikationen in Netscape

3.3 Inter-Client-Kommunikation

Um dem Nachteil zu begegnen, daß nach Übergabe der Kontrolle an eine Helferapplikation der Programmfluß abgegeben wird, muß diese vom Browser steuerbar sein. Es ist auch die umgekehrte Situation möglich, daß eine Applikation sich der Fähigkeiten eines Browsers bedient, um z.B. Datensätze aus dem Web zu beschaffen. Ein über alle Betriebssysteme gleichermaßen verfügbare Methode für die Inter-Client-Communication (ICC) ist derzeit nicht verfügbar, wenn auch die Basiskomponenten benutzbar sind. Dies sind in der Unixwelt insbesondere:

- ONC/RPC
- DCE/RPC
- auf Sockets basierende Spezialanwendungen
- auf CORBA basierende Spezialanwendungen
- OpenDoc-Anwendungen

In der Microsoft Windows 3/NT/95 Umgebung sind verfügbar:

- ONC/RPC

- DCE/RPC
- auf Sockets basierende Spezialanwendungen
- auf CORBA basierende Spezialanwendungen
- DDE, NET-DDE basierende Spezialanwendungen
- OLE2 als Anwendungsschale auf DDE/NET-DDE
- OpenDoc-Anwendungen

4 Die Rolle von ArcView2

Eines der bekanntesten (und wohl neben Produkten von MapInfo, Intergraph, SPANS) und am weitesten verbreitete Desktop-Mapping Programm ist Arcview2 von ESRI. Arcview2 (AV2) wurde von ESRI auf den Markt gebracht, um Endnutzern ein komfortables Werkzeug zum Darstellen und Abfragen von Geodaten und Sachdaten auf Arbeitsplatzrechnern zu ermöglichen. Die Idee ist, daß Geodatenbestände mit ARC/INFO aufgebaut und gepflegt werden, die dann via AV2 auch weniger geübten Nutzern zugänglich gemacht werden sollen. AV2 ist auf allen gängigen Plattformen verfügbar: MS-Windows3.1x, Windows 95, Windows NT (x86), Apple MacOS, verschiedene Unix-Varianten. In der Anwendungspraxis ergab sich jedoch die Notwendigkeit, die mächtige Funktionalität zu reduzieren, bzw. an die Fachbedürfnisse anpassen zu können. Für diese Fälle wurde ein Skriptsprache (Avenue) entwickelt, die zur Programmierung von „Custom-Applikationen“ verwendet werden muß.

4.1 ArcView 2.1 Eigenschaften

Die Version 2.1 von AV2 ist seit etwa Mitte Oktober auf dem deutschen Markt erhältlich. Neben der Korrektur sehr vieler Fehler in 2.0 (ca. 700) hat ESRI folgende Ergänzungen am Lieferumfang vorgenommen:

- Avenue ist im Preis inbegriffen (war ein Extra unter 2.0)
- Verbesserter Zugriff auf ARC-Libraries und ARCSTORM Datamanager
- Import von MapInfo-Dateien
- Integration von AutoCad DXF- und DWG-Dateien in Projekte
- ONC/RPC Client und Server auch unter MS-Windows
- DLL-Unterstützung in Avenue unter MS-Windows für die lokale Interprozeßkommunikation

AV2 kann mit den üblichen Netzwerkfunktionen (NFS, LAN-Manager, Netware) auf Geodatenbestände zugreifen, die aber in einem bestimmten Format vorliegen müssen.

Projekte werden aus einer Menge von *Views*, *Tabellen* und anderen Dokumenten zusammengefaßt und unter einem Namen abgelegt. Ein *View* besteht aus mehreren Ebenen mit thematisch organisierten Daten, genannt *Theme*. Zu einem View gehört auch immer eine *Legende*. *Themes* können zur Darstellung sehr einfach mit einer Checkbox ein- oder ausgeschaltet werden. Alle geographischen Objekte eines *Theme* sind von der gleichen Objektart: Punkt, Linie oder Fläche. Geographische Objekte werden in *Shape*-Dateien gespeichert. Zur Speicherung

von Sachdaten benutzt *AV Tabellen*. Tabellen können in vielfältiger Weise - ähnlich einem Tabellenkalkulationsprogramm - bearbeitet werden, Tabellendaten können außerdem mit geographischen Objekten verknüpft und in thematischen Karten visualisiert werden. Mit einfachen Geschäftsgrafiken (Balken-, Kreisdiagramme usw.) können Zahlenreihen grafisch dargestellt werden.

Tabellen sind dynamisch, d.h. ihr Inhalt ändert sich, sobald sich ihre Datenquelle ändert. Tabellen können sein:

- ARC/INFO Attributtabellen
- DBASE, INFO- und ASCII-Tabellen
- SQL-Datenbankabfragen

SQL-Datenbanken werden unter MS-Windows via ODBC-Treiber angesprochen. AV speichert in der einem Theme zugeordneten Tabelle nicht die Ergebnisse einer Datenbankquery, sondern nur die SQL-Anweisungen, sodaß immer eine Datenbankabfrage gestartet wird, wenn die Tabelle innerhalb eines Projekts aktiviert wird.

Wichtig ist anzumerken, daß AV2 ohne ARC/INFO als Werkzeug zum Erfassen und Aufbereiten von Datenbeständen nicht brauchbar ist. Eine Digitalisierungsfunktion fehlt vollständig. In Kooperation mit ARCSTORM (ARC STORAge Manager) ist es auch möglich im Mehrbenutzerbetrieb via NFS o.ä. auf einen gemeinsamen Datenbestand zuzugreifen.

Die Dokumentation zum Produkt ist eher knapp gefaßt und stark auf Endbenutzer ausgerichtet, da sie als Begleitbuch zu den mitgelieferten Beispieldaten verfaßt ist. Ein Referenzhandbuch, das komplett alle internen Datenstrukturen und Zusammenhänge dokumentiert, fehlt. Selbst im „Introducing ArcView“ [ESRI 1995] wird ständig auf die Online-Hilfe verwiesen.

4.2 Erweiterungsmöglichkeiten von AV2

Um AV2 an Nutzerbedürfnisse anzupassen, hat ESRI eine Objekt-orientierte Skriptsprache (Avenue) entwickelt. Diese Sprache ist auf allen unterstützten Plattformen identisch. Nur einige OS-spezifische Methoden (z.B. der DLL Zugriff unter Windows) sind nicht portierbar.

4.2.1 Avenue - Übersicht

Avenue bietet Entwicklern eine Reihe von ArcView Klassen an, die sich grob in die Gruppen:

1. Document User Interface
2. ArcView Dokumente (Chart, Layout, Script Editor, Tabelle, View)

Die Dokumentklassen sind wieder aus Subklassen zusammengesetzt:

1. Chart z.B. aus Legende, Titel, Achsen, Datenquelle.
2. Die Layout Klasse definiert in den Subklassen Grafik, Graphik-Text, Rahmen, Nordpfeil usw. aus welchen graphischen Komponenten eine Kartendarstellung besteht und wie sie in der Zeichnung angeordnet werden.
3. Der Script Editor stellt die Methoden zum Editieren, Übersetzen, Ablaufen und Testen von Skripten zur Verfügung.
4. In der Tabellenklasse (Table) werden Methoden zur Bearbeitung, Berechnung, Import von Daten in Tabellen bereitgestellt.
5. In der View-Klasse findet man alle Methoden zum visuellen Präsentation von Datenquellen: Features und Bilder (Images) inklusive Maßstäbe, Geocoding, Farbtabelle, Legenden usw.

Daneben gibt es noch eine Reihe von weiteren für die Anwendungssteuerung und Dateneingabe wichtige Klassen:

1. Dialogboxen zum Gestalten von textuellen Benutzerdialogen
2. Dateiklassen für Datenein- und Ausgabe.
3. Interprozesskommunikationsklassen zur Austausch von Daten und Steueranweisungen zwischen verschiedenen Applikationen (DDE und ONC/RPC).

Das Document User Interface erlaubt Entwicklern grafische Interaktionen mit Schaltflächen, Radioknöpfen, Menüleisten, Schaltelementen usw. zu gestalten. Damit sind moderne, unter den verschiedenen Fenstersystemen angepasste Bedieneroberflächen (MOTIF-Style, MS-Windows 3, Apple Mac) herstellbar.

4.2.2 Avenue - Entwicklungswerkzeuge

Für die Entwicklung von Avenueanwendungen stehen ein einfacher Texteditor (Script Editor), ein Compiler und eine einfache Testhilfe zur Ausführung von Programmen im Einzelschrittverfahren, Variableninspektion und mit Haltepunkten zur Verfügung. Avenue-Skripten können auch als ASCII-Texte extern erstellt und dann in den Editor eingelesen werden. Übersetzte Avenueprogramme können an Projekte versteckt angebunden werden, sodaß sie von Endanwendern nicht modifiziert werden können.

4.2.3 Vor- und Nachteile der Entwicklung in Avenue

Avenue ist eine komplett neue Entwicklung von ESRI. Es ist mit keiner der häufig genutzten Sprachen unter MS-Windows, MacOS oder Unix verträglich (z.B. C++ oder Visual Basic). Darüberhinaus hat es auch nichts mit der Skriptsprache von Arc/Info (AML) oder PC-Arc/Info (SML) gemeinsam. Programmierer müssen sich neu einarbeiten, Lehrbücher sind außer mit „Introducing Avenue 2.0“, der Online-Hilfe und ein paar Beispielskripten nicht verfügbar.

Andererseits ist Avenue relativ leicht erlernbar und - das ist m.E. sehr wichtig - auf allen Plattformen mit dem gleichen Funktionsumfang ausgestattet.

4.3 Datenzugriffsfähigkeiten

AV2 erlaubt den Zugriff auf Daten von MS-ACCESS, Dbase-II und IV, Excel-Tabellen, ASCII-Tabellen, INFO-Dateien und SQL-Datenbanken via ODBC (Open Database Connectivity) auch über Netze. Standardnetzwerkdateisysteme, wie z.B. Netware, NFS, LAN-Manager werden voll unterstützt. Arc-Coverages müssen allerdings erst in das Format der Arc/Info Version 7 konvertiert werden. Geometrische Daten von Arc/Info (Coverages) werden direkt unterstützt (ohne Konvertierung), wie auch AutoCAD Drawings im DWG-Format oder im DXF. MapInfo Dateien müssen über ein Import-Werkzeug zuerst in Shape-Files konvertiert werden.

4.4 Datenkommunikationsmechanismen

Zur Kommunikation mit anderen Anwendungsprogrammen (z.B. Excel, Access, Visual Basic) enthält Avenue Methoden, um AV2 von anderen Prozessen steuern zu lassen oder um AV2 als Steuerung für andere Prozesse verwenden zu können.

4.4.1 DDE

Unter MS-Windows (3.11, 95, NT) steht DDE (Dynamic Data Exchange) sowohl als Client als auch als Server zur Verfügung. Die wesentlichen Dienste der Servervariante sind, daß eine andere Anwendung (z.B. ein Visual Basic Programm) Avenue-Anweisungen an AV2 senden kann. Umgekehrt kann mit den entsprechenden Anweisungen z.B. Excel dazu veranlaßt werden, eine Kalkulation auszuführen und deren Ergebnisse in eine AV2 Table zu übertragen.

4.4.2 RPC

Die SUN ONC/RPC (Remote Procedure Calls) stehen seit der Version 2.1 sowohl unter Unix als auch unter MS-Windows zur Verfügung. Auch hier werden Avenue-Skripts an AV2 gesendet, ausgeführt und deren Ausführungsstatus zurückübertragen. Da Arc/Info seit Version 7.0 ebenfalls eine RPC-Server-Funktion anbietet, ist es auch möglich AML-Skripten an einen AI-RPC-Server zu übertragen und sie dort ausführen zu lassen.

4.4.3 System

Mit der System-Klasse kann man beliebige Host-Kommandos im lokalen System ablaufen lassen. Dies kann z.B. nötig sein, wenn eine Verbindung mit einer anderen Applikation gebraucht wird, diese aber noch nicht gestartet wurde.

5 Vorschlag für eine Architektur

In einer Client-orientierten Sicht, muß dafür gesorgt werden, daß die zur Versorgung eines Viewers erforderlichen Dokumente und Dokumentteile vollständig vom Server übertragen werden. Im Fall von Arcview-Projekten (bestehend aus Projektdatei plus Views, Skripten usw) kann das so aussehen:

1. Die AV-Projekte werden im Server mittel pkzip o. ä. in ein Archiv verpackt, das in einer einfachen flachen Datei mit einer speziellen MIME-Kennung abgespeichert wird.
2. Im Client wird eine Applikation als Batch-Datei definiert, die im Fall der Übertragung o.g. Projektarchivs gestartet wird:
3. `application/avz; unpav2 %s`
4. unpav2 ist ein Batch-Programm, das den Entpacker aufruft und anschließend AV2 mit dem Projekt als Argument startet.

6 Geodatenserver

Geodaten, wie sie den Nutzern heute angeboten werden, sind gekennzeichnet durch proprietäre Formate, häufig unzureichende Beschreibung und daß sie nicht auf einem einheitlichen Referenzmodell definiert sind. Interoperabilität zwischen unterschiedlichen Softwarelösungen verschiedener Hersteller ist deshalb nicht vorhanden. Um Daten in einem System X nutzen zu können, die von einem System Y erzeugt wurden, müssen (z.T. mit Informationsverlust behaftete) Konvertierungen durchgeführt werden. Dieser Mangel (u.a. ausführlich im RIPS Feinkonzept erörtert) führte zu verschiedenen Entwicklungsansätzen sowohl im UIS-Umfeld, als auch in anderen Ländern, wie z.B. in den USA.

Die Grundidee eines Geodatenservers besteht darin, Anwendungsprogrammen Geodaten in geeigneter - möglichst transparenter - Weise verfügbar zu machen. Datenmanagement und Datensicherheit ist am Standard kommerzieller Datenbanksysteme angelehnt. Dazu gehört auch, Metadaten (Daten über Geodaten) bereitzustellen und abfragbar zu machen.

6.1 Bisherige Arbeiten

An dieser Stelle sollen einige neuere Arbeiten auf dem Gebiet der Geodatenservices dargestellt und bewertet werden.

6.1.1 FAW

In [Riekert, Ebbinghaus 1995] wird das Konzept eines Geodatenservers innerhalb des Projektes GODOT beschrieben. Wesentliche Designziele waren:

- Blattschnittfreie Speicherung landesweiter ATKIS DLM25-Daten.
- Karten- und Teilobjektübergreifende Geoobjektbildung.
- Rasterdatenhaltung.
- Metadatenbereitstellung zur Unterstützung von Nutzeranfragen
- Eine Abfragekomponente für zeitliche, thematische und räumliche Kriterien.
- Lieferung von Daten in Standardformaten (soweit vorhanden), wie z.B. EDBS, ETF und Firmenformaten.
- Kostenverrechnung für abgerufene Geodaten.
- Sicherheit und Zugangskontrolle inkl. Benutzeridentifikation.
- Mehrbenutzerfähigkeit.
- Konzeption einer Clientsoftware

Für die Implementierung der Geodatenbank verwendete man ein kommerzielles OODBMS (ObjectStore). Als Clientsoftware und Netzwerkprotokoll wurden Mosaic und das HTTP (d.h. WWW) eingesetzt. CORBA als alternative Vermittlungsschicht ist angedacht und für die spätere Projektarbeit vorgesehen. Die Entwicklung des Systems befindet sich im Prototyp-/Demonstratorzustand.

6.1.2 OGIS

OGIS (Open Geodata Interoperability Specification) ist eine Spezifikation für die objektorientierte Definition von Geodaten. Ihr Ziel ist, die offene, verteilte Verarbeitung von Geodaten innerhalb großer Netzwerke zu ermöglichen und als Basis für die Entwicklung von interoperablen Anwendungslösungen zu dienen. Die Entwicklung von OGIS wird vom OGIS Technical Committee (OTC) des Open GIS Consortium (OGC) gesteuert. Das OTC bringt die geleistete Arbeit in nationale und internationale Standardisierungsgremien ein, wie z.B. in das ISO TC211 GIS/Geomatics Committee, ANSI X3L, die OMG und das FGDC (Federal Geographic Data Committee) der US-Bundesregierung.

Technische Ziele des OGC sind:

- Ein universelles räumlich/zeitliches Daten- und Prozessmodell, das alle derzeitigen und denkbaren räumlichen Anwendungen abdeckt

- Eine Spezifikation für alle wesentlichen Datenbanksprachen (SQL, OQL), um in ihnen das OGIS Datenmodell implementieren zu können
- Eine Spezifikation für alle wesentlichen verteilten Verarbeitungsmodelle (CORBA, OLE), um in ihnen das OGIS Verarbeitungsmodell zu implementieren

Dazu wird OGIS folgende Mittel bereitstellen:

- Eine interoperable Anwendungsumgebung, bestehend aus einer konfigurierbaren Arbeitsumgebung mit Werkzeugen und Daten, um Probleme zu lösen
- Einen gemeinsam nutzbaren Datenraum und ein generisches Datenmodell um möglichst viele analytische und kartographische Anwendungen abzudecken
- Einen interoperablen Ressourcen-Browser, um Metadaten, Daten und Verarbeitungsmethoden in einem Netzwerk zu erkunden und auf sie zuzugreifen
- Definition eines Satzes von Schnittstellenstandards für Geodatenzugriff und Verarbeitungsfunktionen
- Kapselungstechnik, um Altdatenbestände nahtlos zu integrieren

Die zwei Hauptkomponenten innerhalb des Architekturrahmens der OGIS sind das *OGIS Geodata Model* (OGM) und das *OGIS Reference Model* (ORM).

OGM bildet den eigentlichen Kern; es besteht aus einer hierarchischen Klassenbibliothek für Geodatentypen und den dazugehörigen vereinheitlichten Programmierschnittstellen (Methoden). Sie wird alle Aspekte räumlicher und zeitlicher Beschreibung von Daten, Visualisierungs-, Analyseverfahren, Speicher- und Kommunikationsaspekte abdecken. OGM wird als Schnittschicht über die in den USA bereits vorhandenen Standards für Geodaten austausch (SDTS), Archivierung (SAIF) und DIGEST eingezogen werden. Es wird mit SQL3-MM interoperabel sein. OGC arbeitet aktiv bei der SQL3-Spezifikation mit.

OGM wird Objektanfragen, Ein-/Ausgabe verschiedener Formate, Modellierung und lange Transaktionen unterstützen. Es wird deshalb detaillierte Definitionen räumlicher Objekte und Funktionen enthalten.

Das ORM beschreibt eine offene Entwicklungsumgebung, die auf wiederverwendbaren Klassenbibliotheken und einer Menge von Diensten beruhen. Der Entwurf des OGM bestimmt die Menge von Diensten, die vom ORM bereitgestellt werden müssen. ORM benötigt deshalb Dienstverzeichnisse und Broker für die Dienstvermittlung. Es legt Standardverfahren für Transformationen und Verarbeitungsschritte fest, wie z.B. Koordinatentransformationen, Raster/Vektorkonvertierung, Visualisierung und die Kapselung existierender nicht konformer Datenbestände.

Mitglieder des OGC sind derzeit:

GIS-Firmen:

- ESRI
- Genasys
- Graphic Data Systems
- Intergraph
- MapInfo

- MITRE Corp.
- PCI
- Smallworld Systems

Computerfirmen:

- Hewlett Packard
- IBM
- Sun Microsystems

Staatliche Organisationen:

- ARPA
- CSIRO, Australian National Research Laboratory
- DMA, Defense Mapping Agency
- NASA
- NOAA
- NRCS
- USACERL
- USDA Natural Resources Conservation Service
- USGS

Nichtstaatliche Organisationen:

- National Geographic Society
- OMG (Object Management Group)

Universitäten:

- UCB, UCLA, Rutgers Univ., Univ. of Arkansas

Wie man sieht, sind neben einer Reihe führender GIS-Softwarehersteller, Computerfirmen und Universitäten alle US-Ministerien mit einem Bezug zu geographischen Daten vertreten. Microsoft hat seinen Beitritt angekündigt.

Der Zeitplan für OGIS sieht vor, daß bis Ende 1996 ein Satz von Standarddokumenten erarbeitet sein wird, der dann zur formellen Verabschiedung als ISO- und ANSI-Normen an die entsprechenden Gremien weitergeleitet wird.

Einige Firmen haben bereits mit der Entwicklung von OGIS-Draft-konformen Komponenten begonnen: Genasys, ESRI, Autometric, Oracle, Intergraph und AVS. Sie werden mit Sicherheit auf dem größten GIS-Markt der Welt (USA + Australien) große Wettbewerbsvorteile haben.

Um auch die Machbarkeit von Konzepten zu verifizieren, wird parallel zur Spezifikationsarbeit ein "Testbed Program" aufgebaut. Innerhalb dieses Programms implementieren Teilnehmer Prototypen wesentlicher Komponenten der OGIS. Das derzeit wichtigste Vorhaben wird von der NASA finanziert und läuft unter dem Namen *Universal Spatial Data Access Consortium* (USDAC). Es wird von den Teilnehmern: Bell Communications Research (Bellcore),

Rutgers University, Camber Corporation, NASA-Goddard Space Flight Center mit Unterstützung durch die California Resources Agency, NOAA, Intergraph, USACERL, National Geographic Society, Weltbank, MITRE, QUBA, MacDonald Dettwiler, BBN, U. of Arkansas und UCB durchgeführt.

Nach Korrespondenz mit dem OGC kann man davon ausgehen, daß etwa Mitte 1996 ein erster Entwurf der OGIS den Konsortiumsmitgliedern vorliegen wird.

6.1.3 GeoScope

GeoScope ist eines der wichtigsten Projekte innerhalb von USDAC. Es hat das Ziel, die umfangreichen Fernerkundungs- und Geodatenbestände der NASA einem breiten Publikum über das Internet zugänglich zu machen. Dabei sollen Probleme der Interoperabilität und des netzwerkweiten Zugangs identifiziert und gelöst werden. Es wird mit einem Finanzvolumen von US\$ 2.3 Mio. von der NASA gefördert.

Die prinzipiellen Ansätze sind:

- Benutzung von OGM für die Kapselung von Fernerkundungs- und Geodaten in Objekten
- Implementierung von Objektzugriffsdiensten auf der Basis von CORBA
- Entwicklung eines komfortablen graphischen Browsers
- Public Domain Software und Standards zu nutzen, wo immer möglich

GeoScope besteht aus einer GUI/Browser Endbenutzerapplikation und verteilten Datenzugriffsmanagern und Diensten. Ein Katalogserver basierend auf dem "InfoHarness" Tool [Shklar 1995] extrahiert und katalogisiert Metadaten entsprechend dem FDGC Metadaten-Standard aus vorhandenen Metadaten. Es speichert sie in temporären Repositories, um kommende Anfragen beantworten zu können. Weitere Dienste sind für Datenextraktion, Konversion usw. zuständig.

Der Datenzugriffsmanager basiert hauptsächlich auf GRASS-Code eingekapselt in Klassenbibliotheken.

Das Endbenutzerwerkzeug wird mit Tcl/Tk implementiert und Public Domain Software, wie z.B. GRASS, Khoros, EOSDIS und andere umfassen.

Alle Schnittstellen zu Diensten, Managern usw. werden mit CORBA IDL realisiert. HTTP und RPC werden für die Kommunikation zwischen Servern und Klienten verwendet.

6.1.4 Firmen: ESRI, Oracle, Computer Associates

ESRI und Oracle bieten gemeinsam einen Geodatenserver auf der Basis von Oracles GeoObject Extension (ehemals Multidimension) an. Beide haben auch einen gemeinsamen Vorschlag an OGC über ein Geodatenserver-API eingereicht.

Computer Associates bietet mit der Spatial-Extension für INGRES einen Objekt-orientierten Ansatz für das Speichern, Auslesen und Verarbeiten von Geodaten als proprietäre Lösung an.

7 Zusammenfassung und Ausblick

Geodatenserver müssen unter heutigen Gesichtspunkten (relative teure Datenübertragung, sehr billige Datenverteilung mit CDROM) anders bewertet werden:

Da Geodaten (d.H. Basisdaten) sich nur relativ langsam ändern, ist eine Verteilung von Massendaten via CDROM eine wirtschaftliche Lösung, die ein vielfaches Übertragen vermeiden hilft. Dazu müssen diese Daten jedoch in gleicher Weise aufbereitet sein, wie es auch in einem Geodatenserver erforderlich wäre [Riekert, Ebbinghaus 1995], da die Objektbildung, Plausibilitätsprüfung usw. unabhängig vom Transportweg der Daten ist. Diese Arbeit (siehe OGIS) muß trotzdem geleistet werden. Variable Daten sollten dazu via Netzwerkzugriff kombinierbar sein.

Im Viewerbereich läßt sich auf relativ einfache Weise ein Dateitransfer von ArcView2-Projekten in einen PC-Clients initiieren. Diese können dann mit AV2 visualisiert werden. Jedoch ist eine Anpassung des mächtigen AV2 auf Benutzer mit geringer Praxis notwendig. Im Viewerbereich stellt die Server-basierte Lösung einen brauchbaren, preiswerten Weg dar, Nutzern mit Standardbrowsern Zugang zu Geodaten zu verschaffen. Probleme gibt es mit der Belastung des Servers und (wichtiger) der hohen Last auf den Datenleitungen durch die Übertragung der Bilder als Rasterdaten. Bestimmte Operationen, wie z.B. Farb- und Darstellungsänderungen, Layoutgestaltung sollten in einen geeigneten erweiterbaren Clients verlagert werden. Clients, die mit der Multiplattformsprache Java erweiterbar sind, versprechen diese Funktionen übernehmen zu können.

Literatur:

- ESRI: ArcView Quick Start Guide, Version 2.1, Redlands, CA, 1995, 45 S.
- ESRI: ArcView Installation Guide, Version 2.1, Redlands, CA, 1995, 31 S.
- ESRI: Introducing Avenue. Version 2.1, Redlands, CA, 1995, 120S.
- OGC: OGIS Technical Summary. [Http://www.ogis.org](http://www.ogis.org), 10/1995
- Buehler, K.: The OGIS Technical Committee: Purpose, Status, Future. Geo Info Systems, 1/1995, S.48
- Buehler, K.(ed.): The Open Geodata Interoperability Specification. Draft Base Document - OGIS Project Document 94-025R2, Dec 22, 1994
- Cornelio, A.; Thomas, G.: GeoScope Requirements. Bellcore Document BD-GEO-REQ-1.0-1 ISSUE 1, June 1995, 53 S.
- FGDC: Content Standards for Digital Geospatial Metadata. USGS, Reston, VA, 6/1994
- Dept. of Commerce: Spatial Data Transfer Standard (SDTS). Federal Information Processing Standard 173, Washington, D.C., NIST, 1992
- Borenstein, N. et al.: MIME (Multipurpose Internet Mail Extensions). Network Working Group, RFC 1521 and 1522
- Riekert, W.-F.; Ebbinghaus, J.: Geodatenserver in Wide Area Networks. FAW Ulm, 1995, 13S.
- Shklar, L. et al.: GeoScope Detailed Design. Bellcore Document, URL <http://www.ogis.org>, 1/1995
- Shklar, L. u.v.a.: Use of Automatically Generated Metadata for Search and Retrieval of Heterogeneous Information. Proceedings of CAISE 95, June 12-16, Jyvaskyla, Finland, Lecture Notes in Computer Science 932, Springer-Verlag, 1995

Entwicklung eines WWW-basierten Altlasten-Informationssystems

R. Weidemann; W. Geiger; A. Jaeschke; M. Reißfelder
Forschungszentrum Karlsruhe GmbH Technik und Umwelt
Institut für Angewandte Informatik (IAI)
Postfach 3640
D-76021 Karlsruhe

1. EINLEITUNG.....	273
2. DAS VORHABEN ALFAWEB.....	274
2.1 ZIELE	274
2.2 AUFGABEN.....	274
2.2.1 Erschließung und Konvertierung vorhandener Dokumente	275
2.2.2 WWW-Schnittstelle zu Fachsystemen	276
2.2.3 Anforderungen an die Erstellung neuer Handbücher und Fachsysteme	276
2.2.4 Entwicklung eines Zugangssystems	276
2.2.5 Ermittlung weiterer Informationsanbieter	277
2.3 VORGEHENSWEISE	277
2.3.1 Generelle Strategie	277
2.3.2 Arbeiten im Rahmen von GLOBUS II (1995)	278
3. ERGEBNISSE IN GLOBUS II.....	278
3.1 KONVERTIERUNG UND AUFBEREITUNG VON BERICHTEN.....	278
3.1.1 Vorgehensweise	279
3.1.2 Aufbereitung der Berichte in das Zwischenformat RTF	281
3.1.2.1 Einscannen und Verarbeiten der Abbildungen	281
3.1.2.2 Einscannen von Texten.....	282
3.1.2.3 Aufbereitung der Texte.....	283
3.1.3 Konvertierung von RTF nach HTML.....	283
3.1.3.1 rftohtml und rftoweb	284
3.1.3.2 Das Konvertierungsprogramm finish.perl.....	284
3.1.4 Einfügen von Verweisen nach fachlichen Kriterien.....	285
3.2 ANLEITUNG ZUR ERSTELLUNG NEUER BERICHTE.....	286
3.3 DER PROTOTYP DES ATLASTEN-INFORMATIONSSYSTEMS IM WWW	287
3.3.1 Allgemeine Beschreibung	287
3.3.2 Zugang über Berichtsreihe und Titel.....	289
3.3.3 Zugang über Volltextsuche	291
3.3.4 Zugang über den UBA-Thesaurus	293
3.3.5 Zugang über den Umweltdatenkatalog (UDK).....	295
3.4 ERMITTLUNG WEITERER INFORMATIONSANBIETER.....	295
4. KÜNFTIGE ARBEITEN.....	296

1. Einleitung

Die Altlastenbearbeitung hat sich in den letzten Jahren zu einer großen Herausforderung für alle Beteiligten - Sanierungspflichtige, Ingenieurbüros und Verwaltung - entwickelt. Die Bearbeitung des Einzelfalls erfolgt in aller Regel durch Ingenieurbüros im Auftrag des pflichtigen Handlungs- oder Zustandstörers; die Behörde gibt dabei die Erkundungs- und Sanierungsziele vor und erteilt auch die erforderlichen Genehmigungen.

Zuständige Fachbehörden für die Altlastenbearbeitung in Baden-Württemberg waren bis Mitte 1995 die Ämter für Wasserwirtschaft und Bodenschutz mit 23 Dienststellen. Danach wurden diese Aufgaben in die Stadt- und Landkreise mit 44 Dienststellen integriert. Weiter sind aufgrund der Verwaltungsstruktur die Gewerbeaufsichtsämter (9), die Regierungspräsidien (4), das Umweltministerium, die Landesanstalt für Umweltschutz (LfU), das Geologische Landesamt und die Gesundheitsverwaltung bei der Altlastenbearbeitung im Rahmen ihrer Zuständigkeit eingebunden. Die Abteilung 5 - Boden, Abfall, Altlasten - der LfU hat dabei u.a. die Aufgabe, die Grundlagen für eine landeseinheitliche, systematische Altlastenbearbeitung für alle Beteiligten, also die Pflichtigen, die Ingenieurbüros und die Verwaltung, bereitzustellen. Dies erfolgt durch die Erstellung von Arbeitshilfen in Form von Handbüchern und Fachberichten, zunehmend auch von Datenbanken und Fachprogrammen, sowie durch Schulungs- und Fortbildungsveranstaltungen.

Typisch für die Altlastenbearbeitung ist, daß noch keine allgemein gültigen Regelwerke - wie etwa auf dem Gebiet der Abwassertechnik oder der Luftreinhaltung - bundesweit zur Verfügung stehen. Kenntnis und Information über das im Altlastenbereich vorhandene Wissen und der entsprechenden Regelungen anderer Ebenen und Stellen (Länder, Bund, EG, Umweltbundesamt) sind daher um so wichtiger. Dies erfordert zum einen, das Wissen über die verschiedenen Quellen ständig aktuell zu halten und zum anderen, den Zeitaufwand zur Beschaffung benötigter Informationen zu minimieren.

Ähnliche Anforderungen lassen sich in vielen anderen Gebieten ebenfalls beobachten. Die wachsende weltweite Verflechtung in Wirtschaft und Wissenschaft und der zunehmende Bedarf an aktueller Information hat u.a. dazu geführt, daß seit 1993 das World-Wide Web (WWW), der Hypertext-Dienst des Internet, explosionsartig ansteigende Teilnehmerzahlen verzeichnet (Verdopplung alle 2-3 Monate). Das WWW ist ein Client-Server-System, bei dem jeder mit dem Internet verbundene Rechner Informationen zur weltweiten Einsicht bereitstellen (= Server) und/oder bereitgestellte Informationen abrufen (= Client oder Browser) kann. Basisobjekt des WWW ist die Informationsseite, die zunächst nur aus Texten und aktiven Verknüpfungen zu anderen Informationsseiten bestand (Hypertext). Das Anklicken einer aktiven Verknüpfung im Browser führt dazu, daß die entsprechende Informationsseite geholt und angezeigt wird, egal auf welchem Server weltweit diese liegt. Das WWW erlaubt weiterhin die Übertragung beliebiger Dokumente (z.B. ausdrückbare Berichte, Software, ..) und

entwickelt sich durch Integration von Bildern, Videos und Tondokumenten zu einem Hypermedia-System.¹

2. Das Vorhaben AlfaWeb

2.1 Ziele

Im Rahmen von AlfaWeb (Altlasten-Fachinformationen im World-Wide Web) soll ein WWW-Server aufgebaut werden, auf den die Altlasten-Sachbearbeiter von ihrem Arbeitsplatz aus mittels WWW-Browser über das Internet zugreifen können. Der Server soll folgende Aufgaben erfüllen:

- Die Handbücher und Fachberichte der LfU sollen als Hypertext-Dokumente verfügbar gemacht werden.
- Datenbanken und einfache Auswerteprogramme der LfU sollen über eine formularbasierte Schnittstelle abgefragt bzw. verwendet werden können.
- Ein übergeordnetes Zugangssystem soll verschiedenartige Zugriffspfade auf die Informationen der LfU anbieten (z.B. hierarchische Suche, Thesaurus, ..).
- Informationsseiten anderer Anbieter sollen über das Zugangssystem ebenfalls erreichbar sein.

2.2 Aufgaben

Aufgabe des IAI im Rahmen von GLOBUS ist der prototypische Aufbau des in Kap. 2.1 beschriebenen Servers und die Bereitstellung von Werkzeugen und Hilfsmitteln, die es der LfU erlauben, den Server weiter auszubauen und aktuell zu halten.

Dabei sind folgende Aufgabenfelder zu bearbeiten:

- Beratung und Unterstützung der LfU bei der Einrichtung des Servers (Hardware, Basissoftware, Anschluß an das Internet). In diesem Zusammenhang sind auch Sicherheitserfordernisse zu erfüllen.
- Erschließung einiger vorhandener Dokumente und Konvertierung in das WWW-Format HTML.
- Erarbeitung von Anleitungen und Bereitstellen von Hilfsmitteln für die Erschließung und Konvertierung weiterer Berichte.
- Entwicklung einer formular-basierten WWW-Schnittstelle zu ausgewählten PC-Datenbanken und -Fachsystemen der LfU.

¹ Für weitere Informationen zu Internet und WWW sei auf die einschlägige Literatur verwiesen, z.B.: Scheller, M.; Boden, K.-P.; Geenen, A.; Kampermann, J.: Internet: Werkzeuge und Dienste. Springer-Verlag, Berlin/Heidelberg, 1994.

- Beratung der LfU bei der Erstellung neuer Handbücher und Fachsysteme mit dem Ziel einer problemlosen Einbindung in das WWW.
- Entwicklung eines Zugangssystems auf lokal- (CD) und netz-verfügbare (WWW) Altlasteninformationen.
- Ermittlung weiterer für die Benutzer relevanter Informationsangebote im WWW; Einbindung entsprechender Verknüpfungspunkte (Links) im Zugangssystem.

Eine detailliertere Betrachtung der Aufgabenfelder erfolgt in den Unterkapiteln 2.2.1 - 2.2.5:

2.2.1 Erschließung und Konvertierung vorhandener Dokumente

Die bereits existierenden Handbücher und Fachberichte der LfU liegen z.T. nur in Papierform, teilweise aber auch in elektronischer Form, d.h. in dem Format des verwendeten Textverarbeitungs- bzw. Graphikprogramms vor. Da bisher den Erstellern der Berichte die Art deren Bearbeitung freigestellt war, ist das Spektrum der Formate (Software-System und -Version) sehr breit. Vom Inhalt her sind, wie bei technischen Berichten üblich, formatierte Texte, Tabellen und Abbildungen enthalten.

Hypertext-Dokumente haben typischerweise einen anderen Aufbau als "konventionelle" Dokumente. Letztere haben eine "flache" Struktur. Querverweise über einen Index oder aus dem Text heraus erfordern das Blättern und Suchen im Dokument. Zwar kann man das Dokument auch in dieser Form in das WWW einhängen, doch ist dies nicht unbedingt zweckmäßig, da man so die Vorteile des Hypertext-Ansatzes teilweise wieder verschenkt. Um ein konventionelles Dokument ins WWW einbringen zu können, ist dies in einzelne Teile von etwa Bildschirmgröße zu zerlegen, die untereinander verzeigert werden. Das konventionelle Dokument wird in ein Netz von Hypertext-Seiten aufgelöst. Da die Verzeigerung auch nach inhaltlichen Kriterien erfolgen muß, ist eine vollständige Automatisierung nicht möglich!

Um vorhandene Fachberichte ins WWW einbringen zu können, sind grob folgende Arbeiten erforderlich:

- Einscannen von nur in papierform vorliegenden Dokumenten.
- Konvertierung aus dem vorhandenen Dokumentenformat in ein definiertes Zwischenformat.
- Automatische Vorbearbeitung der Dokumente im Zwischenformat (z.B. Zerlegung entsprechend der Kapitelstruktur, Indexierung, Einfügen der Bilder).
- Manuelle Nachbearbeitung der Dokumente mit dem Editor (Verzeigerung nach inhaltlichen Gesichtspunkten).
- Konvertierung nach HTML.

Werkzeuge zur Unterstützung des Vorgangs sind für Teilaufgaben in unterschiedlicher Qualität und Brauchbarkeit vorhanden. Im Rahmen des Vorhabens ist die Auswahl geeigneter Werkzeuge, deren Erweiterung (in Kooperation mit den GLOBUS-Partnern) und die Zusammenstellung in einem durchgehenden Werkzeugkasten vorgesehen.

2.2.2 WWW-Schnittstelle zu Fachsystemen

Das WWW wurde anfangs vornehmlich zur Verteilung statischer Dokumente benutzt. Mit der Erweiterung von HTML zur Definition von Eingabefeldern in Dokumenten und einer Schnittstelle zur Erzeugung dynamischer Dokumente werden inzwischen weitergehende Möglichkeiten eröffnet. Es können somit einfache Datenbankabfragen bzw. Auswertungen realisiert werden.

Im Rahmen dieses Vorhabens soll exemplarisch für einige LfU-Anwendungssysteme, wenn diese erst einmal als selbständige Programme verfügbar sind, eine zusätzliche Zugangsschnittstelle über das WWW implementiert werden.

2.2.3 Anforderungen an die Erstellung neuer Handbücher und Fachsysteme

Berücksichtigt man bei der Erstellung neuer Handbücher und Fachsysteme von vorneherein die spätere Einbindung in das WWW, so kann man sich einen großen Teil des Nachbearbeitungsaufwands sparen. Im Rahmen des Vorhabens werden die Anforderungen an neue Handbücher und Fachsysteme in einem Leitfaden detailliert und durch Mitarbeit in laufenden Arbeitskreisen für den Einzelfall konkretisiert.

2.2.4 Entwicklung eines Zugangssystems

Die Art des Zugangssystems, d.h. auf welche Weise bestimmte Informationen im Altlasten-Informationssystem gefunden werden können, wird i.w. dadurch beeinflusst, daß

- die Hauptnutzergruppe Altlasten-Sachbearbeiter sind, d.h. fundiertes Basiswissen aus dem Anwendungsgebiet vorausgesetzt werden kann und
- alternativ eine lokale Nutzung auf einem PC ohne Netzanschluß (CD-ROM Version) möglich sein muß.

Um die Altlasten-Sachbearbeiter bei der Suche nach der für sie relevanten Information zu unterstützen, ist ein Zugangssystem zu schaffen, das den Informationspool über verschiedene Zugriffspfade erschließt. Mögliche Zugriffspfade sind z.B.:

- Hierarchisch über Organisation (LfU, UM, ..), Berichtsnummer, Jahr
- Thesaurus, Schlüsselwörter
- Volltextsuche

Im Rahmen des Vorhabens wird ermittelt, welche Zugriffspfade technisch sinnvoll und von den Benutzern gewünscht werden. Ein entsprechendes Zugangssystem soll realisiert werden.

In diesem Zusammenhang ist auch zu prüfen, ob ein Teil der bereitgestellten Informationen nur für einen eingeschränkten Benutzerkreis zugänglich sein soll und nicht, wie es standardmäßig der Fall ist, für alle Interessierte weltweit. Falls vertrauliche Informationen enthalten sind, müssen entsprechende Schutzmechanismen (z.B. Password) eingebaut werden.

2.2.5 Ermittlung weiterer Informationsanbieter

Die Anzahl der Informationsangebote im WWW wächst derzeit sehr rasch. Aus diesen ohne Unterstützung durch das Zugangssystem diejenigen Angebote herauszufinden, die für die Altlastenbearbeitung relevant sind, gleicht der Suche nach der Nadel im Heuhaufen, da das WWW, wie der Name schon sagt, ein Informations-Netz ist und keine hierarchische Struktur besitzt, die ohne weiteres einen gezielten Zugriff auf die relevanten Anbieter erlauben würde. Um in das LfU-Zugangssystem ein Verzeichnis relevanter Informationsquellen einbauen zu können, sollen folgende Wege beschriftet werden:

- Abfrage verschiedener Kataloge im Netz, die Informationen über Anbieter enthalten.
- Beobachtung der Informationsseiten und USENET-Gruppen mit Neuankündigungen von Informationsquellen.
- Gezielte Nachfrage bei in Frage kommenden Institutionen (z.B. UBA).

2.3 Vorgehensweise

2.3.1 Generelle Strategie

Die Vorgehensweise bei der Bearbeitung der genannten Aufgabenfelder wird von der Tatsache beeinflusst, daß durch das rapide Wachstum des WWW auch eine Flut Software-Werkzeuge unterschiedlichster Funktionalität und Qualität auf den Markt kommt. Bevor daher selbst eine spezielle Einzellösung entwickelt wird, ist immer zu prüfen, ob nicht bereits ein geeignetes Werkzeug vorhanden ist oder in absehbarer Zeit erwartet werden kann.

Ein kritisches Element des WWW ist die Dokumenten-Beschreibungssprache HTML. Das WWW lebt von der Vielzahl der Browser, die für nahezu alle Rechnersysteme angeboten werden und HTML als gemeinsame Basis verstehen. Falls spezielle Anforderungen aus dem Altlastenbereich mit dem existierenden HTML-Standard nicht abgedeckt werden können, ist besonders hier jede Speziallösung kritisch zu sehen. In diesem Fall ist es besser, die benötigten HTML-Erweiterungen in die Diskussionen um die Fortschreibung des HTML-Standards einzubringen. Da sich der Standard mit dem gleichen Tempo weiterentwickelt, wie das ganze Gebiet, ist dies im Vergleich zu sonstigen Standardisierungs-Bestrebungen in einem vertretbaren Zeithorizont möglich.

2.3.2 Arbeiten im Rahmen von GLOBUS II (1995)

Die Bearbeitung des Vorhabens erfolgt in einzelnen, überschaubaren Stufen. Für die erste Stufe (Arbeitsprogramm für 1995) wurden gemeinsam mit der LfU die existierenden Handbücher und Fachberichte gesichtet und 3 Berichte ausgewählt, die sowohl typisch hinsichtlich der enthaltenen Strukturen (Texte, Tabellen, Abbildungen) als auch bezüglich der vorliegenden Form sind. Bis Ende des Jahres 1995 sollen diese Berichte als WWW-Hypertext-Dokumente aufbereitet und unter einer ersten Version des Zugangssystems im WWW angeboten werden.

Die LfU entwickelt derzeit die fachlichen Grundlagen von weiteren Fachsystemen. Durch die Mitarbeit in den Arbeitskreisen:

- Methoden und Strategien für die Erstellung von Umweltbilanzen bei der Altlasten- und Schadensfallbearbeitung.
- Leitfaden Erkundungsstrategie Grundwasser
- Referenzkatalog Altlasten/Schadensfälle

wird auf die Berücksichtigung der Anforderungen an eine spätere WWW-Einbindung hingewirkt. Bei den Projekten Umweltbilanz und Referenzkatalog sollen, neben den Berichten, Fachsysteme auf PC-Basis entwickelt werden. Nach deren Fertigstellung kann eine Einbindung in den WWW-Server als erste aktive Anwendungssysteme in der nächsten Stufe des Vorhabens (1996) angegangen werden.

3. Ergebnisse in GLOBUS II

3.1 Konvertierung und Aufbereitung von Berichten

Den Kern des Altlasten-Informationssystems bilden die Fachberichte und Handbücher der LfU. Man kann diese grob in 3 Kategorien klassifizieren:

- Bericht liegt nur in Papierform vor.
- Bericht liegt (auch) als Dokument im speziellen Format eines Textverarbeitungssystems vor.
- Bericht wird erst noch erstellt. Die LfU hat sich für künftige Berichte auf Microsofts Word für Windows² festgelegt. Im Einzelnen kann auf die Form der Berichte noch Einfluß genommen werden.

Obwohl in der Regel mit einem Textverarbeitungssystem erstellt, wurden Berichte bisher ausschließlich in Papierform herausgegeben. Zu Beginn des Vorhabens wurde von der LfU recherchiert, in welchen Fällen eine elektronische Form des Berichts beschaffbar war. Für 12 Berichte wurden Dateien in unterschiedlichen Formaten gefunden, die allerdings teilweise

² In diesem Bericht wird meist WinWord als Abkürzung für „Microsoft Word für Windows“ benutzt.

Berichte nicht vollständig abdecken (z.B. keine Abbildungen; nur einzelne Berichtsteile). Da es sich bei einem der Berichte um einen Tagungsband handelt, der aus Vorwort und 19 unterschiedlichen Einzelbeiträgen besteht, lagen insgesamt 31 Berichte bzw. Berichtsteile vor. Tabelle 1 enthält eine Übersicht über die unterschiedlichen Formate dieser 31 Berichts(-teile):

(Teil-)Berichte	Format der Texte	Abbildungen
2	ASCII (MS DOS)	fehlen
7	nur Papierversion	fehlen
1	Pagemaker	Designer 3.1
6	Star Writer	fehlen
1	unbekannt	fehlen
3	Word für DOS 3.x - 5.x	fehlen
2	Word für DOS 6.0	fehlen
1	Word für DOS 6.0	Designer 3.1
2	Word für Windows 2.0	keine
1	Word für Windows 2.0	Windows Metafile (WMF)
3	Word für Windows 6.0	fehlen
2	Word Perfect 5.1	fehlen

Tabelle 1: Übersicht der verwendeten Formate

Bei einigen wenigen Berichten lagen Tabellen als separate Dateien vor (Quattro Pro).

Aus dieser Bestandsaufnahme lassen sich folgende Randbedingungen für die Entwicklung eines Werkzeugs zur Konvertierung nach HTML und die WWW-gerechte Aufbereitung ableiten:

- Es gibt viele Berichte, die nur in Papierform vorliegen. Diese müssen mit berücksichtigt werden.
- Die LfU hat sich auf Word für Windows für zukünftige Berichte festgelegt. Damit genießt dieses Format die höchste Priorität.
- Da für jedes Format der vorhandenen Berichte nur wenige Exemplare vorhanden sind, lohnt sich der Aufwand für die Bereitstellung von Tools zur direkten Konvertierung nach HTML nicht. Statt dessen ist insgesamt betrachtet der Umweg über WinWord sicherlich effizienter.
- Abbildungen liegen in der Regel nur in Papierform vor. Das Einscannen der Abbildungen ist daher der Standardfall.

3.1.1 Vorgehensweise

Zusammen mit der LfU wurden drei typische Berichte ausgewählt, die exemplarisch bearbeitet und im WWW verfügbar gemacht werden sollten. Es handelt sich dabei um:

- Materialien zur Altlastenbearbeitung Band 12: Das Modellstandortprogramm des Landes Baden-Württemberg, April 1993 (Tagungsband mit Vorwort und 19 Einzelbeiträgen).
- Gemeinsame Verwaltungsvorschrift des Umweltministeriums und des Sozialministeriums Baden-Württemberg über Orientierungswerte für die Bearbeitung von Altlasten und Schadensfällen vom 16. September 1993.
- Texte und Berichte zur Altlastenbearbeitung, Band 10/94: Eingehende Erkundung für Sanierungsmaßnahmen/Sanierungsvorplanung (E3-4), Mai 1994.

Weitere 7 Berichte (s. Kap. 3.3.2 Zugang über Berichtsreihe und Titel) wurden zusätzlich zu den 3 geforderten bearbeitet und sind im WWW verfügbar. Bei deren Aufbereitung konnte jedoch nicht derselbe Aufwand investiert werden, sodaß die Qualität geringer ist (z.B. keine Abbildungen).

Für den Entwurf eines Ablaufschemas zur Konvertierung und Aufbereitung von Berichten für das WWW war ausschlaggebend, daß der hohen Dynamik in diesem Bereich wegen ein leistungsfähiges Werkzeug mit vertretbarem Aufwand nur dann entwickelt werden kann, wenn man sich konzentriert. Dies hat zur Konsequenz, daß man sich auf Seiten der Textverarbeitung auf ein bestimmtes Format festlegt. Weiterhin sollte möglichst auf vorhandene Software (Freeware oder Shareware) zurückgegriffen werden, wiederum um den Aufwand zu begrenzen.

Unter Berücksichtigung der weiter oben genannten Randbedingungen und nach Prüfung der verfügbaren Konvertierungstools wurde ein Ablaufschema erarbeitet (s. Abb. 1), das auf folgenden Eckpfeilern ruht:

- Vorhandene Abbildungen und Berichte, die nur in Papierform vorliegen, werden eingescannt.
- Ausgangsformat für die Konvertierung nach HTML ist RTF (Rich Text Format). RTF hat den Vorteil, daß
 - alle wichtigen Textverarbeitungssysteme in dieses Format exportieren können;
 - es sich um ein ASCII-Format handelt (nur darstellbare Zeichen) und daher leicht bearbeitbar ist;
 - es bereits Konvertierungstools mit einer ausbaufähigen Funktionalität gibt.
- Die Bearbeitung neuer Fachberichte erfolgt in Word für Windows. Über eine Dokumentenvorlage und eine Anleitung zu deren Benutzung wird sichergestellt, daß die neuen Berichte leicht konvertiert werden können.
- Das Konvertierungswerkzeug wird als Erweiterung der zusammengehörigen Freeware-Produkte rtftohtml und rtftoweb realisiert. Diese Produkte haben sich bei einer Sichtung der verfügbaren Tools³ als am erfolgversprechensten erwiesen.

³ Über WWW und Newsgroups wurden die vorhandenen Tools und Erfahrungsberichte von Anwendern ermittelt. Verzeichnisse mit Konvertern sind im Anhang unserer „Richtlinie zur Erstellung WWW-verfügbarer Berichte“ aufgeführt.

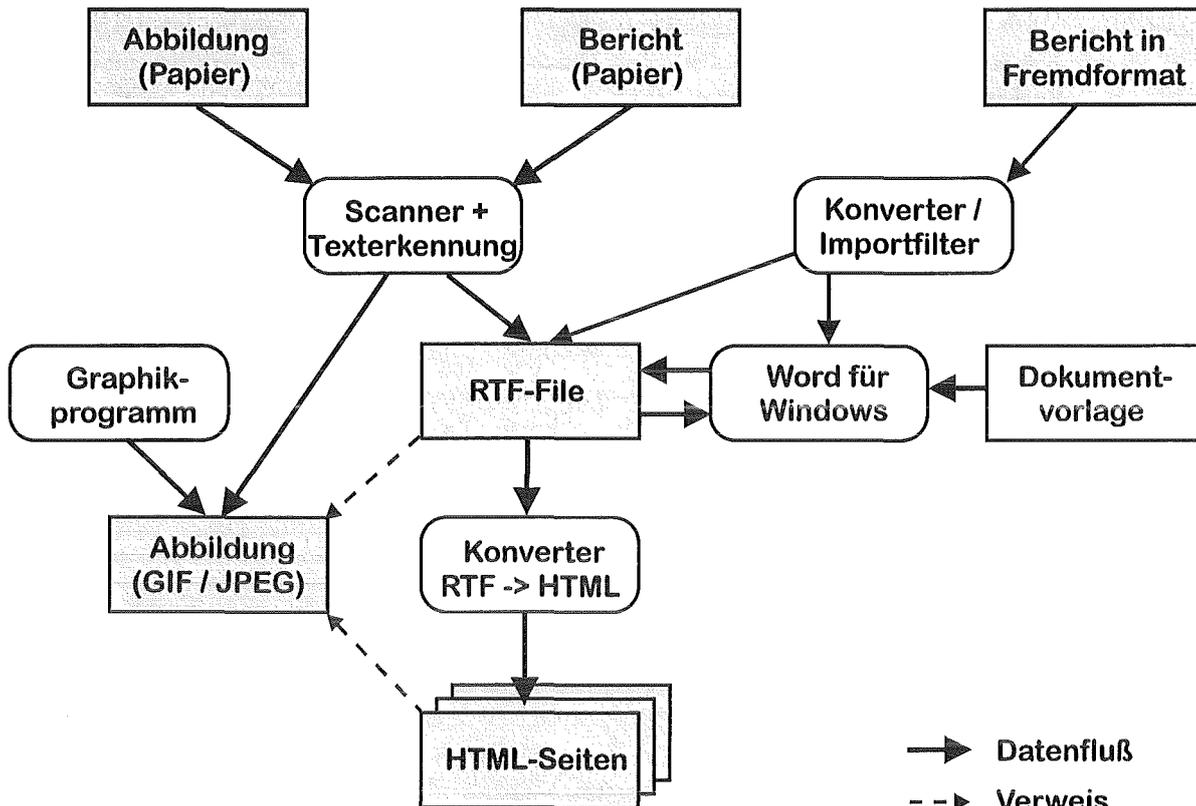


Abbildung 1: Ablaufschema für die Aufbereitung und Konvertierung von Berichten

3.1.2 Aufbereitung der Berichte in das Zwischenformat RTF

3.1.2.1 Einscannen und Verarbeiten der Abbildungen

Zum Einscannen und Verarbeiten der nur in gedruckter Form vorliegenden Abbildungen wurden folgende Werkzeuge verwendet:

- DeskScan II - Scanner von Hewlett Packard mit zugehöriger Software zum Einscannen der Abbildungen und Abspeichern im Bitmap-Format
- Lview-Software zur Konvertierung des Dateiformats von Bitmap nach GIF. In einigen Fällen wurden damit auch die Abbildungen gedreht.

Die Abbildungen wurden überwiegend als Schwarz-Weiß-Zeichnungen, d.h. ohne Graustufen, eingescannt und verarbeitet. Beim Einscannen wurde eine Auflösung von 75 dpi (Bildpunkte pro Zoll) gewählt. Mit Hilfe einer entsprechenden Funktion der HP-Software wurde noch die Schärfe der Abbildungen verbessert. Einige Abbildungen, insbesondere Fotos von Altlasten und Geräten, wurden als Schwarz-Weiß-Fotos mit 256 Graustufen und 75 dpi Auflösung eingescannt, wobei ebenfalls die Funktion zur Verbesserung der Bildschärfe eingesetzt wurde. Die Bildunterschriften wurden nicht mit den Abbildungen eingescannt, sondern zusammen mit dem Text des jeweiligen Dokuments behandelt. Als Abbildungsgröße wurde i.allg. die Originalgröße (Skalierung 100%) verwendet, d.h. die Abbildungen wurden weder verkleinert noch vergrößert; bei einigen Zeichnungen mit kleiner Schriftgröße der Be-

schriftung wurden auch Skalierungsfaktoren im Bereich 120 - 150% gewählt. Bei der Helligkeit wurde meist der vom HP-System auf der Basis einer Voransicht der Abbildung vorgeschlagene Wert benutzt.

Die DeskScan II - Software bietet bisher nicht die Möglichkeit zur direkten Abspeicherung im GIF-Format, welches für das WWW benötigt wird. Deshalb wurden die Abbildungen zunächst im Bitmap-Dateiformat (MS Windows 3.0 Bitmap) gespeichert und dann mit dem Programm Lview31 von L. Loureiro in das GIF-Format (GIF87a) konvertiert. Dabei wurden, wenn erforderlich, auch Drehungen der Abbildungen, insbesondere vom Querformat ins Hochformat, vorgenommen.

Ein Hauptproblem bei den Abbildungen ist, einen vernünftigen Kompromiß zwischen ihrer späteren Übertragungszeit über das Netz und ihrer Auflösung und damit Qualität zu finden. Jeder, der mit dem WWW schon gearbeitet hat, weiß, daß ab einer Dateigröße von größenordnungsweise 10 KB (der Wert hängt natürlich vom Übertragungsweg und der aktuellen Netzperformance ab) die Wartezeiten auf Empfängerseite als lästig empfunden werden. Selbst bei einer Schwarz-Weiß-Zeichnung ohne Graustufen mit einer Auflösung von 75 dpi (die etwa der Auflösung des Bildschirms entspricht) ergibt eine Abbildung von ½ Seite DIN A4 bereits eine unkomprimierte Dateigröße von ca. 30 KB und nach der Kompression im GIF-Format eine Dateigröße von ca. 10 KB. Eine solche Auflösung entspricht aber deutlich noch nicht der Qualität des Originals oder der Qualität (Auflösung) von Laser-Druckern. Bei einer Auflösung von 75 dpi ist ein Abbildungstext in kleiner Schrift, insbesondere wenn die Vorlage in mäßiger Qualität vorliegt, nur schlecht oder überhaupt nicht erkennbar. Falls die Bildqualität von Laser-Druckern gewünscht wird, muß zu einer höheren Auflösung mit entsprechend höheren Übertragungszeiten im Netz übergegangen werden. Das Problem der Übertragungszeit tritt bei Fotos (mit Graustufen) und farbigen Abbildungen noch verstärkt auf. Bei lokalem Betrieb des Altlasten-Informationssystems über CD-ROM entfällt es allerdings.

3.1.2.2 Einscannen von Texten

Zum Einscannen der Texte wurde ebenfalls der DescScan II - Scanner von Hewlett Packard verwendet. Dabei wurde mit folgenden Parametern gearbeitet: Schwarz-Weiß-Zeichnung (scharf), Auflösung: 300 dpi, Skalierung: 100%, automatische Helligkeitssteuerung. Zur optischen Zeichenerkennung (optical character recognition, OCR) wurde das Softwarepaket TextBridge von Xerox Imaging Systems eingesetzt, das auch ein Sprachpaket für die deutsche Sprache enthält. Die Texte wurden mit TextBridge im RTF-Format abgespeichert und mit Word für Windows nachbearbeitet.

Die Erfahrungen haben gezeigt, daß das Einscannen von Texten doch mit einem erheblichen Nachbearbeitungsaufwand verbunden ist und deshalb nur für besonders wichtige textliche Ressourcen sinnvoll ist. Der Nachbearbeitungswand liegt bei grauem Umweltpapier minderer Qualität merklich höher, weshalb weißes, holzfreies Papier für die Vorlagen verwendet werden sollte.

3.1.2.3 Aufbereitung der Texte

Die Bearbeitung der Berichte erfolgt, wie bereits erwähnt, mit Word für Windows im RTF-Format. Um Berichte, die in einem anderen Format erstellt wurden, nach RTF zu konvertieren, sind mehrere Alternativen offen:

- Das Textverarbeitungssystem, mit dem der Bericht erstellt wurde, kann diesen nach RTF exportieren (z.B. Star Writer).
- Word für Windows besitzt einen Importfilter für das fremde Berichtsformat (z.B. Word für DOS, Word Perfect).
- Ein allgemeines Konvertierungsprogramm (z.B. Word für Word der Firma Mastersoft) kann eingesetzt werden.

I.d.R. ergaben sich nach der Konvertierung größere Abweichungen im Layout gegenüber der im ursprünglichen Format ausgedruckten Version. Einige Berichte konnten über verschiedene Wege konvertiert werden, wobei sich die Ergebnisse z.T. deutlich unterschieden.

Aber auch die Berichte, die sich gut konvertieren ließen oder bereits in WinWord 6.0 Format waren, erforderten eine Nachbearbeitung. Eine wesentliche Voraussetzung für die weitere Konvertierung nach HTML ist nämlich, daß die Gestaltung der Berichte über strukturelle Attribute und nicht über das Aussehen erfolgt. Beispielsweise muß eine Überschrift auch als Überschrift gekennzeichnet sein und nicht nur durch größere Buchstaben im Text hervorgehoben werden. Bei den meisten Berichten wurde dies bei der Erstellung nicht beachtet oder die entsprechenden Informationen gingen bei der Konvertierung verloren.

Da parallel zur Aufbereitung der Berichte und aufbauend auf den dabei gewonnenen Erfahrungen eine Richtlinie zur Erstellung neuer Fachberichte erarbeitet wurde (s. Kap. 3.2 Anleitung zur Erstellung neuer Berichte), wurden auch die vorhandenen Berichte nach dieser Richtlinie umgestaltet. Der dazu erforderliche Aufwand ist sehr stark vom Inhalt der einzelnen Berichtsseite abhängig und lag bei weniger als einer Minute für Seiten, die nur Fließtext enthalten, bis zu einer halben Stunde z.B. für Seiten mit komplexen Tabellen, die praktisch eine Neueingabe erforderlich machten.

3.1.3 Konvertierung von RTF nach HTML

Die Software zur Konvertierung des im RTF-Format vorliegenden Fachberichts nach HTML hat im wesentlichen folgende Aufgaben:

1. Umsetzen der RTF-Beschreibungen für einzelne Textsegmente in die entsprechenden HTML-Anweisungen (z.B. normaler Text, Hervorhebungen, Listen, Tabellen, ..).
2. Zerlegung des Berichts in einzelne HTML-Seiten gemäß der Gliederungsstruktur (Überschriften) bis zu einer gewünschten Tiefe.
3. Anlegen eines separaten Inhalts- und eines Indexverzeichnisses.

4. Einfügen von Verweisen (Hyperlinks) zum Blättern innerhalb des Bericht (nächste Seite, vorherige Seite, nächst-höheres Kapitel, Titelseite, Inhaltsverzeichnis, Index)
5. Einfügen von Verweisen auf die in separaten Dateien liegenden Abbildungen.
6. Einfügen von Verweisen auf andere Berichte und Informationsquellen.
7. Einfügen von Verweisen auf die Startseite des Altlasten-Informationssystems.
8. Einfügen von Informationen für ein einheitliches Layout (Hintergrundfarbe, Abspann, ...)

Realisiert wurde die Konvertierungssoftware in der Programmiersprache Perl⁴, die sich besonders zur Zeichenkettenbearbeitung (Texten) eignet und flexible Möglichkeiten zur Datei-manipulation besitzt. Den Kern des System bildet jedoch ein Freeware-Produkt (rtftohtml / rftoweb), das den Hauptteil der Konvertierung übernimmt. Die Software wurde auf dem Rechner entwickelt, der derzeit auch als WWW-Server für das Altlasten-Informationssystem dient (Sun SPARCstation 2 mit dem Betriebssystem SunOS 4.1.x).

3.1.3.1 rtftohtml und rftoweb

Das Freeware-Produkt „rtftohtml“ wurde von Chris Hector, Mitarbeiter der Firma Cray, entwickelt. Es dient dazu, RTF-Dokumente in HTML-Dokumente umzusetzen und realisiert i.w. den Punkt 1 und Teile von Punkt 5 der Aufgabenliste in Kap. 3.1.3. Christian Bolik (Universität Hannover) hat mit „rftoweb“ eine Erweiterung dazu implementiert, die als zusätzliche Funktionen die Zerlegung in ein Netz von HTML-Seiten und die Erzeugung separater Inhalts- und Indexverzeichnisse vornimmt (Punkt 2 - 6). Jede HTML-Seite wird als Datei in demselben Verzeichnis wie der zerlegte RTF-File abgespeichert. Beide Produkte sind im Quellcode (Programmiersprache: C) auf UNIX-Rechnern verfügbar, rtftohtml zusätzlich auch auf Macintosh und PC.

Die Konvertierung wird teilweise über Konfigurationsfiles gesteuert. Durch das Editieren dieser Files kann in gewissen Grenzen der Konvertierungsvorgang beeinflusst werden. Dies wurde u.a. dazu benutzt, die Konvertierung von Sonderzeichen zu verbessern und zu vervollständigen. Eine weitere Anwendungsmöglichkeit ist die Definition zusätzlicher Formatvorlagen auf WinWord/RTF Seite und die Spezifikation deren Umsetzung nach HTML. Wegen der beschränkten Darstellungsmöglichkeiten von HTML sind dabei allerdings keine wesentlichen Verbesserungen zu erreichen.

3.1.3.2 Das Konvertierungsprogramm finish.perl

Das Konvertierungsprogramm „finish.perl“ bereitet den RTF-File für die Konvertierung vor, startet das mit rftoweb erweiterte rtftohtml und bearbeitet anschließend die erzeugten HTML-Files nach. Es wird davon ausgegangen, daß für jeden Bericht (RTF-File, HTML-Files, Abbildungen, ..) ein eigenes Verzeichnis verwendet wird. Pro Bericht ist ein Konfigurationsfile zu erstellen und ebenfalls in diesem Verzeichnis abzulegen. Der Konfigurationsfile enthält eine ungeordnete Liste von Attribut-Werte-Paaren, die folgendes festlegen:

⁴ siehe z.B.: Randal L. Schwartz: Einführung in Perl. O'Reilly, 1995.

- Die Aufrufparameter für rtftohtml. Damit ist sichergestellt, daß der Konvertierungsvorgang nachvollzogen werden kann und bei Änderungen am Bericht, die HTML-Seiten in gleicher Struktur reproduzierbar sind.
- Die Adresse des obersten Verzeichnisses (Startseite) von AlfaWeb relativ zu dem Verzeichnis des Berichts.
- Bei einem Teilbericht (z.B. Beitrag in Tagungsband) die relative Adresse und den Titel der Startseite des Gesamtberichts.
- Das Format (File-Extension) der Abbildungen.

Diese Angaben werden zur Ablaufsteuerung des Konvertierungsprogramms verwendet. Neben dem Starten von rtftohtml sind die wesentlichen Funktionen des Konvertierungsprogramms „finish.perl“:

- Einfügen bzw. Ändern von Verweisen auf Abbildungen.
- Einfügen der Verweise auf die Startseite von AlfaWeb und Gesamtbericht
- Einfügen des Titels des Berichts auf jeder HTML-Seite.
- Definition des Hintergrundbildes.
- Generieren der Schlußzeile der HTML-Seiten (z.B. Mailadresse, Datum)
- Einfügen des Bearbeiters und des Bearbeitungsdatums am Seitenende.
- Aktualisierung der Angaben über Anzahl HTML-Seiten und Abbildungen auf der Startseite des Altlasten-Informationssystems.
- Freigabe der HTML-Seiten für den Zugriff über den WWW-Server

3.1.4 Einfügen von Verweisen nach fachlichen Kriterien

Während der Konvertierung werden Verweise zwischen den HTML-Seiten auf der Basis der Gliederungsstruktur (Links zwischen den Seiten und vom Inhaltsverzeichnis aus) und, wenn im Original ein Index definiert war, aus dem Indexverzeichnis auf die entsprechenden Seiten definiert. Die Möglichkeiten eines Hypertext-Systems werden allerdings erst dann richtig genutzt, wenn über aktive Verknüpfungen Querbezüge zu anderen Informationsquellen aufgezeigt werden. In erster Linie ist beim Altlasten-Informationssystem dabei an Verweisen auf andere Fachberichte und Handbücher zu denken.

Da solche Querbezüge in den aufbereiteten Berichten bisher fehlen, wurde von der LfU ein Ingenieurbüro beauftragt, die drei zum Pflichtprogramm für 1995 gehörigen Berichte (s. Kap. 3.1.1 Vorgehensweise) um fachliche Querbezüge innerhalb einzelner Berichte, zwischen den Berichten und auf weitere bisher nicht im WWW-verfügbare Berichte zu ergänzen. Entsprechende Verweise werden ins Textdokument eingefügt (vgl. Kap. 3.2 Anleitung zur Erstellung neuer Berichte) und bei der Konvertierung nach HTML automatisch in aktive Verweise umgesetzt.

3.2 Anleitung zur Erstellung neuer Berichte

Die Landesanstalt für Umweltschutz erarbeitet zur Unterstützung der Altlasten-Sachbearbeiter ständig neue Fachberichte und Handbücher und aktualisiert bestehende entsprechend dem fortschreitenden Stand der Technik. Um das neue Altlasten-Informationssystem gleich mit einer „kritischen Masse“ an Informationen zu füllen, die einen nutzbringenden Einsatz erlaubt, ist es zwar notwendig, bestehende Berichte in das System einzuhängen, doch längerfristig betrachtet ist es wichtiger, dafür zu sorgen, daß neue oder überarbeitete Berichte ohne großen Aufwand auch im WWW bereitgestellt werden können. Diesem Gesichtspunkt wird mit der „Richtlinie zur Erstellung WWW-verfügbarer Berichte“ Rechnung getragen.

Die Richtlinie enthält Vorgaben zur Erstellung von Dokumenten unter Word für Windows und wird ergänzt um eine Dokumentvorlage, welche die Einhaltung der Vorgaben unterstützt. Da HTML bei weitem nicht die ganze Vielfalt an Darstellungsmöglichkeiten eines Textverarbeitungssystems wie WinWord bietet, kann nur über eine gezielte Beschränkung die Konvertierbarkeit gewährleistet werden. Andererseits wird von WinWord die Definition von aktiven Verweisen praktisch nicht unterstützt, sodaß man vorhandene Eigenschaften des Textverarbeitungssystems zweckentfremden muß.

Um die Anwendbarkeit der Richtlinie zu demonstrieren, wurde diese bereits bei ihrer eigenen Erstellung befolgt und als Basis für den GLOBUS II Abschlußbericht vorgegeben. Außerdem wurde die Richtlinie dazu benutzt, die mögliche Aktualität von Informationen als Vorteil einer rechnergestützten Verbreitung von Berichten zu demonstrieren. Deshalb wurde die Richtlinie auch nicht erst am Ende des Projekts als vollständige Version verfügbar gemacht, sondern es wurden aufbauend auf einer ersten Basisversion die während der Aufbereitung der bestehenden LfU-Dokumente gewonnenen Erfahrungen direkt eingearbeitet. Die jeweils aktuelle Version des Dokuments konnte unmittelbar im WWW eingesehen werden und ist - außer im Anhang des GLOBUS II Abschlußberichts - auch weiterhin im Altlasten-Informationssystem unter der Adresse (URL) <http://www.iai.fzk.de/~weideman/lfu/berichte/richtl/richtl.html> verfügbar.

Im Anhang der Richtlinie werden zur Einführung in die Thematik Internet, WWW und HTML sowohl konventionelle Informationsquellen (Bücher) genannt, als auch auf WWW-verfügbare Dokumente und Verzeichnisse verwiesen. Weiterhin wird kurz auf

Texteditoren,
HTML-Editoren und
Konverter

als Alternativen zur Erstellung von HTML-Seiten eingegangen. Da Editoren und Konverter für HTML größtenteils als Shareware-Produkte erhältlich sind, werden Verweise auf spezifische Verzeichnisse und Übersichten angegeben.

3.3 Der Prototyp des Altlasten-Informationssystems im WWW

3.3.1 Allgemeine Beschreibung

Die im Rahmen des GLOBUS-Projekts konvertierten und aufbereiteten Berichte wurden jeweils unmittelbar über das WWW den Anwendern zur Verfügung gestellt. Der Zugriff erfolgt über die Startseite des Altlasten-Informationssystems, die unter der Adresse (URL) <http://www.iai.fzk.de/~weideman/lfu/lfu.htm> erreichbar ist⁵. Abbildung 2 zeigt als Hardcopy den Stand der Seite am 07.11.1995.

Die Startseite bietet verschiedene Zugriffspfade auf die gespeicherten Berichte und Handbücher. Derzeit ist ein Zugriff möglich über

- Berichtsreihe und Titel, d.h. über eine Auswahl aus der Liste der Berichte,
- eine Schlagwortsuche in dem vom Umweltbundesamt herausgegebenen Thesaurus der Umweltbegriffe,
- eine Volltextsuche über frei wählbare Begriffe und Begriffskombinationen.

In allen drei Fällen wird direkt auf die Berichte zugegriffen, d.h. der WWW-Server, der die Altlasten-Berichte verwaltet, stellt auch die Zugriffsmechanismen zur Verfügung. Informationen über die Fachberichte sollen aber auch in den in Entwicklung befindlichen Metainformationssystemen Umweltdatenkatalog (Projekt WWW-UIS des FZI) und Ressourcenkatalog (Projekt Integral des FAW) bereitgestellt werden⁶. Diese Kataloge sind als übergreifende Auskunftssysteme im Rahmen des Umweltinformationssystems Baden-Württemberg konzipiert und damit natürlich auch für die Altlasten-Sachbearbeiter relevant und nutzbar.

Weitere auf der Startseite des Altlasten-Informationssystem enthaltene oder von dort über Verweise erreichbare Informationen sind u.a.:

- aktueller Umfang (Anzahl WWW-Seiten und Abbildungen),
- umweltrelevante Informationen im WWW,
- Informationsseite zum GLOBUS-Projekt,
- die Richtlinie zur Erstellung WWW-verfügbarer Berichte, ..

⁵ Das Altlasten-Informationssystem wird derzeit auf einem Rechner des FZK betrieben. Im weiteren Projektverlauf ist die Übertragung auf einen LfU-Rechner vorgesehen.

⁶ Informationen über die Fachberichte sind im UDK bereits testweise enthalten.

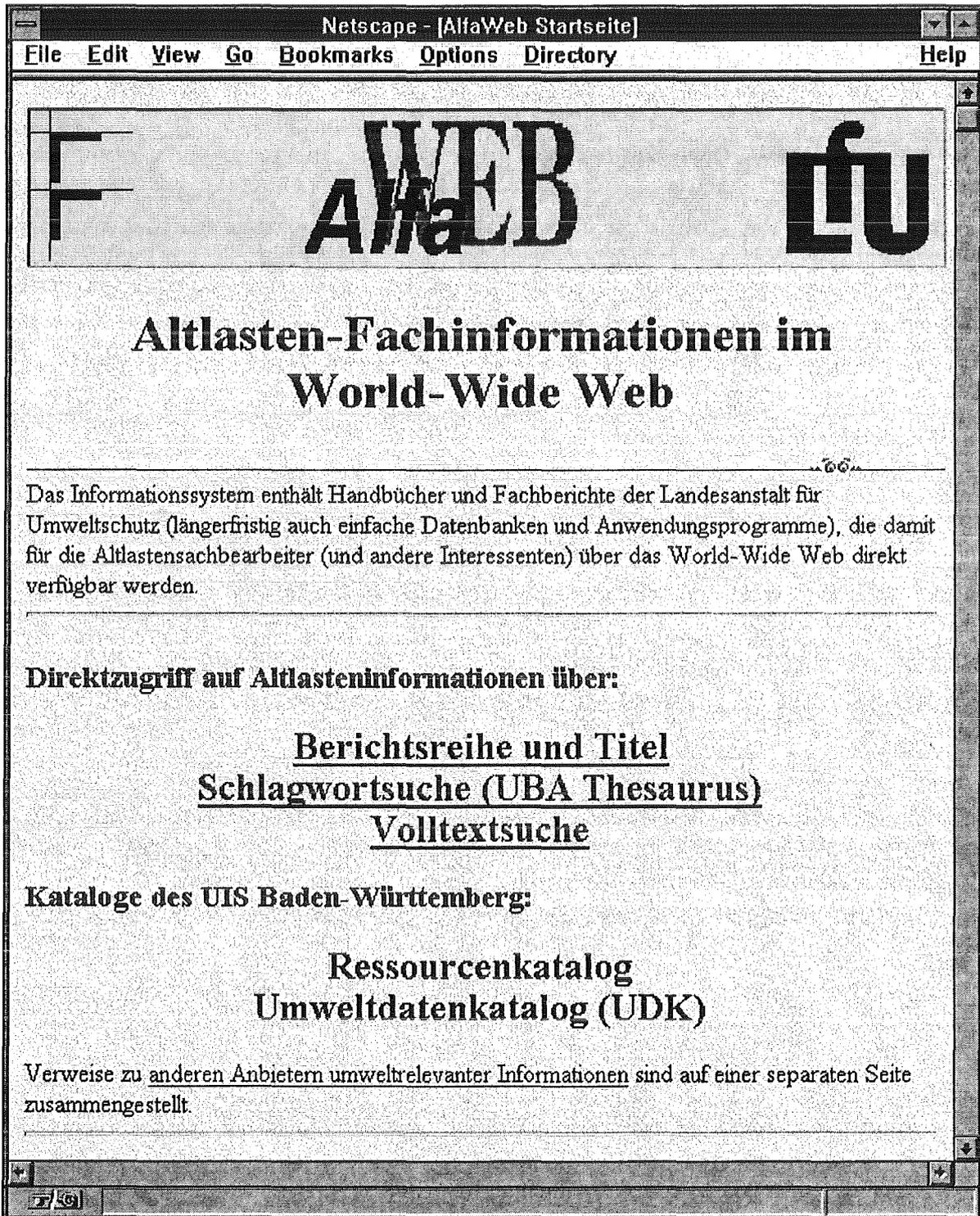


Abbildung 2: Startseite des Altlasten-Informationssystems

3.3.2 Zugang über Berichtsreihe und Titel

Ein Zugang über Berichtsreihe und Titel ist dann sinnvoll, wenn der Benutzer sich darüber informieren will, welche Berichte insgesamt verfügbar sind oder genau weiß, welchen Bericht er sich ansehen möchte. Untergliedert in die Abschnitte „Materialien zur Altlastenbearbeitung“, „Texte und Berichte zur Altlastenbearbeitung“ und „Sonstige Berichte“ werden die zugreifbaren Berichte mit den wichtigsten Informationen zu ihrer Identifizierung gelistet. Die Auswahl eines Berichts führt direkt zu dessen Startseite, von wo aus dann im Bericht geblättert werden kann.

Abbildung 3 zeigt eine Hardcopy der Auswahlseite (Stand: 07.11.1995), zum aktuellen Stand sei auf das WWW verwiesen.

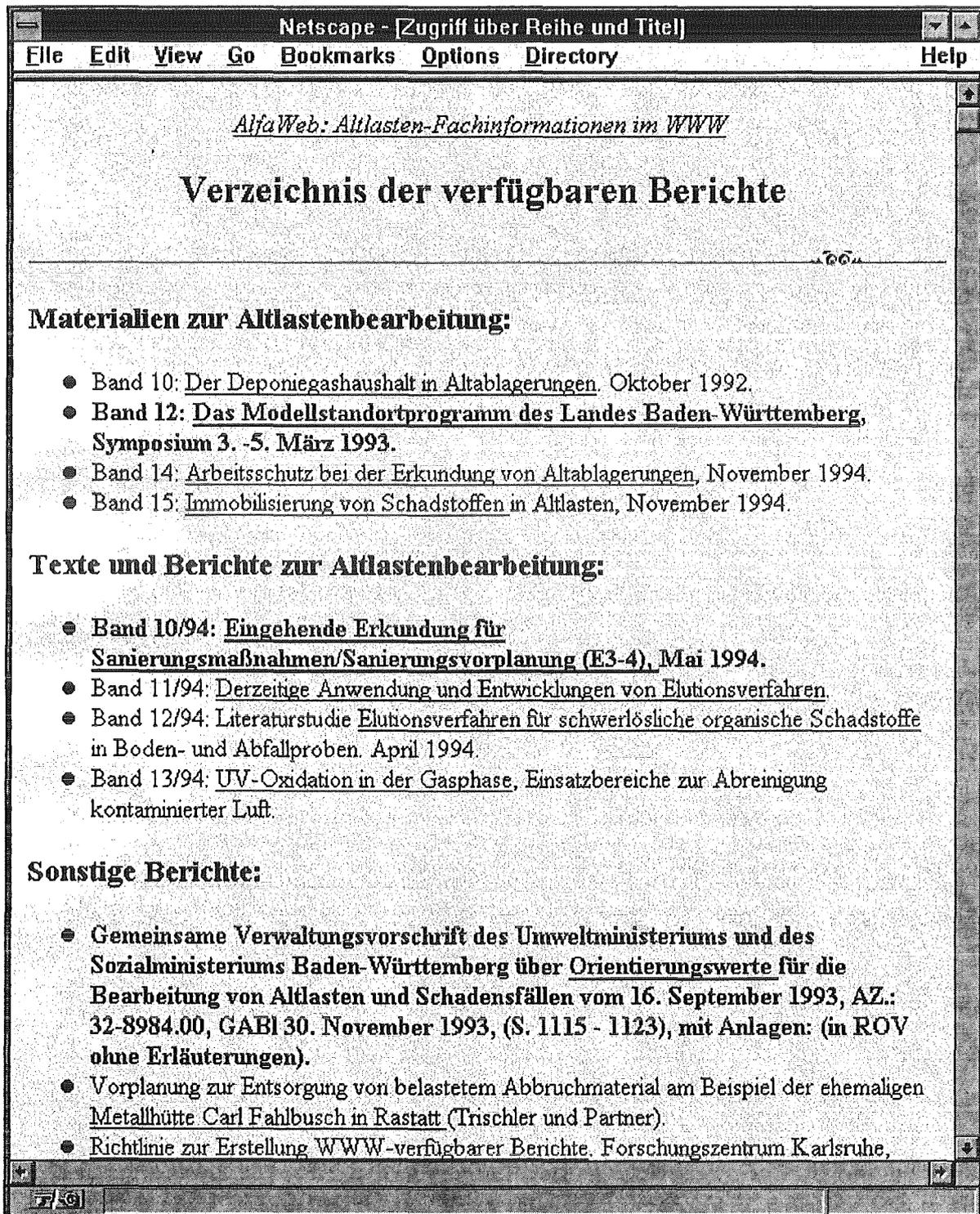


Abbildung 3: Derzeit im WWW verfügbare Berichte

3.3.3 Zugang über Volltextsuche

Die Volltextsuche erlaubt nach einzelnen Berichtsabschnitten (WWW-Seiten) zu suchen, in denen vom Benutzer frei wählbare Begriffe oder Begriffskombinationen enthalten sind. Das Suchformular (Hardcopy siehe Abbildung 4) enthält ein Eingabefeld, in das die Suchbegriffe einzutragen sind. Begriffe können durch „and“ und/oder „or“ verknüpft werden, wobei der Eindeutigkeit wegen Klammern einzufügen sind, wenn beide Operatoren gleichzeitig verwendet werden. Groß-/Kleinschreibung wird bei der Suche nicht berücksichtigt. In den Suchbegriffen kann „*“ als Platzhalter für 0-n beliebige Zeichen angegeben werden.

Die Suche, die durch Anklicken des Auswahl-Buttons oder Betätigen der „Return“-Taste im Eingabefeld angestoßen wird, erfolgt immer über alle Berichte des Altlasten-Informationssystems.⁷ Das System liefert als Suchergebnis eine WWW-Seite, welche als wesentlichen Inhalt eine Liste aktiver Verweise auf die gefundenen Berichtsabschnitte hat. Ein Berichtsabschnitt wird über eine Bezeichnung identifiziert, die sich aus dem Namen des Berichts und der Kapitelüberschrift zusammensetzt. Zusätzlich ist die Größe (Anzahl Zeichen) des Abschnitts und eine Bewertungsziffer angegeben. Die Bewertungsziffer, nach der die Verweise geordnet sind, dient als Maß dafür, wie gut eine Abschnitt die Suchkriterien erfüllt. Es wird beispielsweise berücksichtigt, wie oft Suchbegriffe im Dokument auftauchen und an welcher Stelle (Überschrift oder normaler Text). Die Bewertungsziffer des am besten passenden Abschnitts wird auf 1000 normiert.

Kern der Volltextsuche ist das Public Domain System SWISH, das von Kevin Hughes in der Programmiersprache C für UNIX-Rechner implementiert wurde und im Quellcode verfügbar ist. SWISH wurde speziell dafür konzipiert, suchbare Indexverzeichnisse von WWW-Seiten zu erzeugen. Die Benutzeroberfläche für das WWW, das heißt die Suchformulare und Ergebnistabellen, sind nicht mehr Teil von SWISH. Hier konnte jedoch auf Ergebnisse des FAW aus den Projekten Integral und GLOBUS zurückgegriffen werden. Die zur Verfügung gestellte Software wurde an die spezifischen Randbedingungen des Altlasten-Informationssystems (Adressen, Serverkonfiguration) angepaßt und in einigen Punkten erweitert (Umlaute, Darstellung des Suchbegriffs, ..).

⁷ In späteren Versionen ist eine benutzerdefinierte Einschränkung auf eine Untermenge der Berichte vorgesehen.

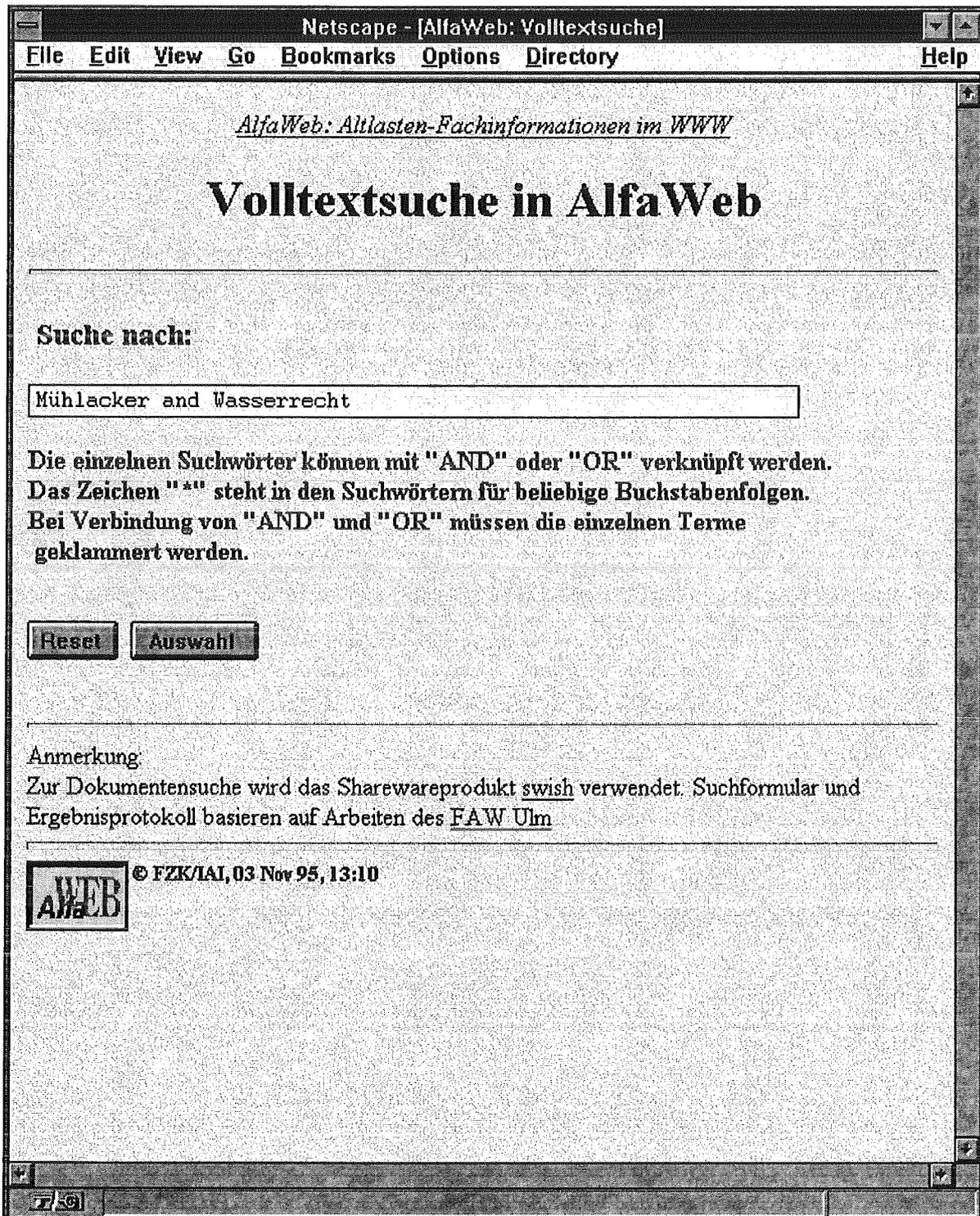


Abbildung 4: Suchmaske für die Volltextsuche

3.3.4 Zugang über den UBA-Thesaurus

Der Thesaurus umweltrelevanter Begriffe des Umweltbundesamts ist Basis der Schlagwortsuche im Altlasten-Informationssystem und umfaßt in der vorliegenden Form mehr als 7000 Schlagworte (Deskriptoren) und zusätzlich synonyme Begriffe in derselben Größenordnung. Die Begriffe sind in ein Begriffsnetz eingebettet (Ober-/Unterbegriffe, verwandte Begriffe, Synonyme).

Über den alphabetischen Index, welcher der Handhabbarkeit wegen auf mehrere WWW-Seiten aufgeteilt ist, kann ein Schlagwort ausgewählt werden. Abbildung 5 zeigt beispielhaft einen Ausschnitt der WWW-Seite für den Begriff Altöl. Die Berichtsabschnitte, in denen das Schlagwort enthalten ist, werden als aktive Verweise aufgeführt. Außerdem werden alle Begriffe des Thesaurus, die zu dem aktuellen Begriff in einem logischen Zusammenhang stehen, mit entsprechenden Verweisen auf deren Deskriptorseiten aufgeführt.

Die Schlagwortsuche ist über statische HTML-Seiten realisiert, d.h. für jeden Deskriptor ist eine eigene HTML-Seite abgelegt, in welche die Verweise auf die Berichtsabschnitte eingetragen werden müssen. Gegenüber einer Datenbanklösung mit dynamischen HTML-Seiten hat dies den Vorteil, einen schnelleren Zugriff zu ermöglichen und ist daneben problemlos in der geplanten lokalen Version (CD-ROM) des Altlasten-Informationssystems nutzbar. Der erforderliche Speicherplatz von ca. 12MB ist bei den Kapazitäten heutiger Datenträger ohne wesentliche Bedeutung.

Zur Verschlagwortung der Berichte stehen Programme zur Verfügung, die getrennt für jede HTML-Seite die darin enthaltenen Begriffe gemäß UBA-Thesaurus ermitteln. Dabei wird nicht nur nach den Grundformen der Begriffe (1:1 Übereinstimmung mit den Thesaurus-Begriffen) gesucht, sondern es sollen auch abgeleitete Formen wie z.B. der Plural eines Begriffs identifiziert werden. Der momentan verwendete Algorithmus, der die letzten beiden Buchstaben eines Wortes ignoriert und damit manchmal zu unzutreffenden Schlagworten führt, soll in späteren Versionen des Altlasten-Informationssystem durch einen leistungsfähigeren ersetzt werden.

UBA-Thesaurus, Deskriptorseiten und die Werkzeuge zur Verschlagwortung von Berichten wurden vom FAW zur Verfügung gestellt und dort im Rahmen der Projekte Integral und GLOBUS entwickelt. Ergänzt wurden diese i.w. um Programme zur Aufbereitung der Deskriptorseiten (Adressen, einheitliches Layout im Altlasten-Informationssystem) und zu deren Pflege, wie z.B. zum selektiven Entfernen einzelner Berichte aus der Verschlagwortung.

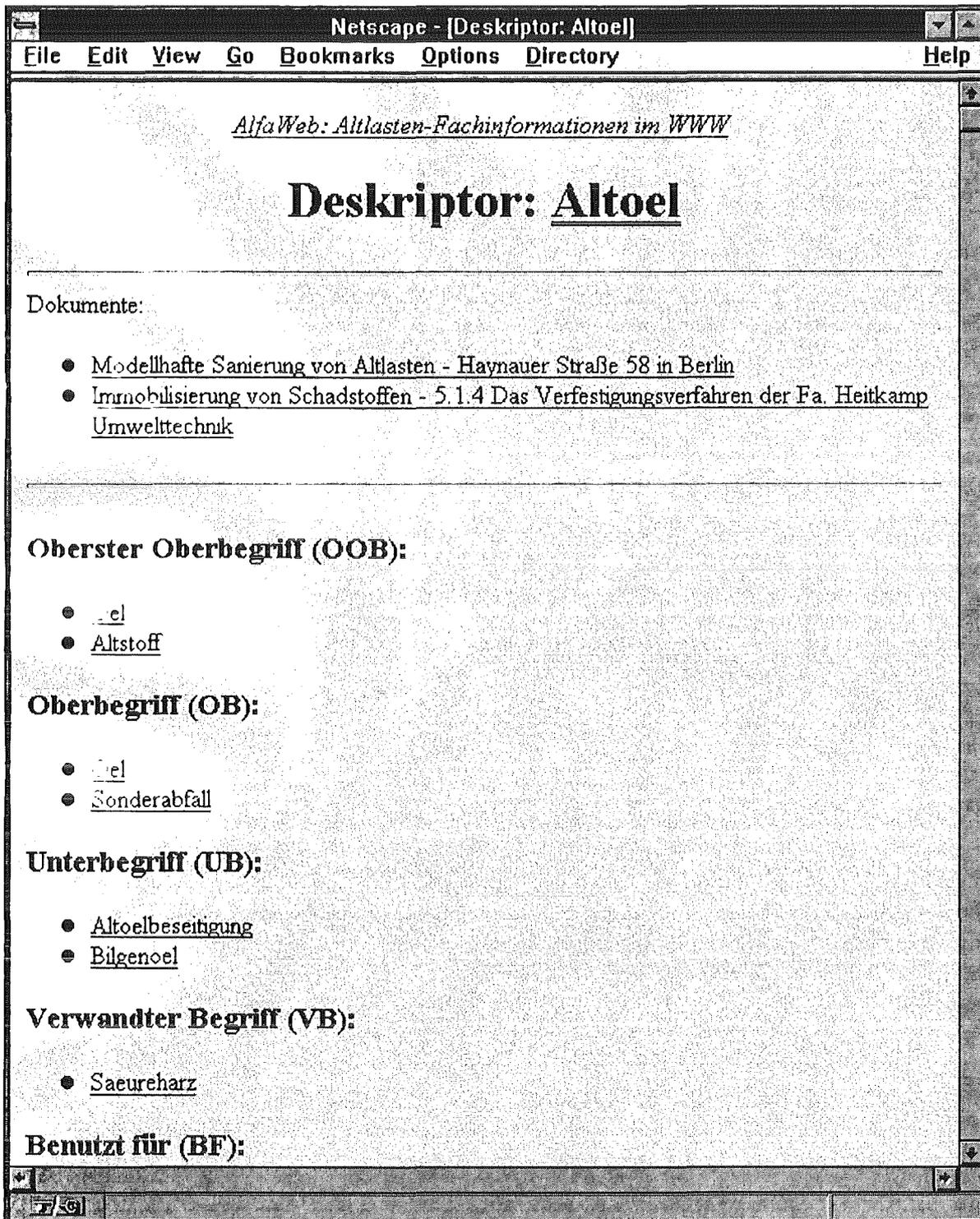


Abbildung 5: Beispiel für eine Deskriptorseite

3.3.5 Zugang über den Umweltdatenkatalog (UDK)

Das FZI entwickelt im Rahmen des Projekts WWW-UIS den in Regie des Umweltministeriums Niedersachsen entstandenen Umweltdatenkatalog weiter zu einem datenbank-gestützten Metainformationssystem für Dienste und Berichte im Umweltinformationssystem Baden-Württemberg. Um beispielhaft die Einbindung von Berichten demonstrieren zu können, wurden die im Altlasten-Informationssystem vorhandenen Fachberichte in den UDK aufgenommen.

Pro Berichtsteil, der im UDK als eigenständiges Objekt eingetragen werden soll, sind - soweit vorhanden - folgende Metainformationen bereitzustellen:

- Bezeichnung des Berichts bzw. Kapitels
- Autoren bzw. Herausgeber
- Erscheinungsjahr
- Kurzbeschreibung
- Schlagworte
- Adresse (URL)
- Einordnung in die Berichtshierarchie (Vorgänger/Nachfolger)

Diese Informationen sind größtenteils durch die Berichtersteller zu liefern, wobei zur Eingabe das Datei-Info von WinWord herangezogen wird. Dort können neben dem Titel des Berichts die Autoren, die Kurzbeschreibung und benutzerdefinierte Schlagworte angegeben werden. Bei der Konvertierung des Berichts werden in einem separaten File die Metainformationen für den UDK angelegt. Adressen und Unterkapitel werden automatisch ergänzt, weitere Schlagworte bei der Verschlagwortung des Berichts (s. Kap. 3.3.4 Zugang über den UBA-Thesaurus) erzeugt. Der in einem definierten Format aufbereitete File wird zur weiteren Bearbeitung an das FZI übergeben.

3.4 Ermittlung weiterer Informationsanbieter

Die Altlasten-Sachbearbeiter können dadurch, daß das Altlasten-Informationssystem über WWW angeboten wird, natürlich auch weitere umweltrelevante Informationsquellen im Netz nutzen. Gerade gelegentliche Benutzer haben aber wegen der Struktur des WWW oft Schwierigkeiten, für sie interessante Informationen aufzufinden. Es wurde deshalb auf einer separaten WWW-Seite eine Liste mit Verzeichnissen von Links zu umweltrelevanten Informationsquellen zusammengestellt. Weiterhin ist eine Seite mit den wichtigsten Server-Listen, Such-Werkzeugen und allgemeinen themen-orientierten Verzeichnissen des WWW verfügbar.

Ein eigenes Verzeichnis umweltrelevanter Links, das sowohl für das Altlasten-Informationssystem als auch für den im FZK-IAI betriebenen WWW-Server der GI-Fachgruppe „Informatik im Umweltschutz“ verwendet wird, ist ebenfalls im Aufbau. Zu diesem Zweck werden regelmäßig folgende Internet-Ressourcen ausgewertet:

- Newsgroups:
 - [de.comp.infosystems](#)
 - [de.comm.internet](#)
 - [comp.infosystems.www.announce](#)
- Neuankündigungen:
 - [What's New \(NCSA Mosaic\)](#)
 - [What's New on Yahoo](#)

Mittelfristig ist geplant, in Zusammenarbeit mit der LfU eine Bewertung und Auswahl der für die Altlasten-Sachbearbeiter des Landes besonders relevanten Informationsquellen vorzunehmen.

4. Künftige Arbeiten

Die im Rahmen von GLOBUS II begonnenen Arbeiten an dem Altlasten-Informationssystem im WWW sollen in den Folgejahren fortgesetzt werden. Im Vordergrund steht dabei, ein für die Altlasten-Sachbearbeiter in ihrer täglichen Arbeit nutzbringend-einsetzbares Werkzeug zu schaffen. Als unmittelbare Anschlußarbeiten sind geplant:

- Einbindung von Anwendungsprogrammen
An einem (maximal 2) geeigneten Systemen soll die Einbindung von Anwendungsprogrammen in das Altlasten-Informationssystem beispielhaft durchgeführt werden. Dabei sollen die in der vorhandenen Richtlinie (Kap. 3.2 Anleitung zur Erstellung neuer Berichte) aufgeführten Kriterien zur Gestaltung von ins WWW-einbindbaren Anwendungsprogrammen vervollständigt und weiter detailliert werden.
- Weiterentwicklung des Konvertierungsprogramms
Parallel zur Fortschreibung des HTML-Standards sollen für technische Berichte wichtige Gestaltungsmöglichkeiten für das Altlasten-Informationssystem verfügbar gemacht werden.
- Einbindung weiterer Berichte
Weitere Berichte sollen vordringlich von der LfU selbst oder in deren Auftrag gemäß der vorhandenen Richtlinie aufbereitet werden. FZK-IAI berät die LfU dabei und führt die Konvertierung der Berichte nach HTML und die Einbindung in das Altlasten-Informationssystem durch.
- Weiterentwicklung des Zugangssystems
 - Verschlagwortung
 - Das verwendete Verfahren muß weiterentwickelt werden, um die Behandlung der unterschiedlichen Wortendungen zu verbessern.
 - Es ist ein Werkzeug zu entwickeln, das die manuelle Nachbearbeitung der zu einem Dokument erzeugten Schlagwort-Liste erlaubt.
 - Volltextsuche
Mit zunehmender Informationsmenge im Altlasten-Informationssystem wird es wichtig, den Suchraum für die Volltextsuche einschränken zu können (z.B. auf bestimmte Berichte). Außerdem soll die Volltextsuche auch lokal betreib-

bar sein (ohne Zugriff auf den zentralen Server des Altlasten-Informationssystems)

- Kataloge des UIS Baden-Württemberg
Auf die Altlasteninformationen soll auch über den Umweltdatenkatalog (FZI Karlsruhe) bzw. den Ressourcenkatalog (FAW Ulm) zugegriffen werden können. In Kooperation mit den genannten Institutionen sind Form und Umfang der dafür benötigten Meta-Informationen zu spezifizieren. Außerdem sind Werkzeuge zu entwickeln, die diese Meta-Informationen soweit möglich automatisch aus den Berichten extrahieren.
- CD-ROM Version zur lokalen Verwendung auf PCs
Um auch auf nicht vernetzten PCs das Altlasten-Informationssystem nutzen zu können, ist eine CD-ROM-Version zu entwickeln, welche die wichtigsten Berichte der LfU umfaßt.
- Entwicklung eines Werkzeugs für den Endanwender
Damit die LfU das Altlasten-Informationssystem selbständig ergänzen kann, sind die verschiedenen Konvertierungs- und Aufbereitungsprogramme in ein durchgehendes Werkzeug mit benutzerfreundlicher Oberfläche zu integrieren.
- Installation des Altlasten-Informationssystems auf einem LfU-Server
Das derzeit auf einem WWW-Server des FZK betriebene System soll längerfristig auf einen LfU-Server übertragen werden. Als ersten Schritt in diese Richtung ist zu entscheiden, auf welcher Basis (Hardware, Betriebssystem) der LfU-Server eingerichtet werden soll, um die zu entwickelnde Software später problemlos portieren zu können.

Ausblick - Fortführung des Projekts GLOBUS

Die Ergebnisse der F+E-Arbeiten in den Teilprojekten von GLOBUS II sind jeweils am Ende der voranstehenden Berichte der Forschungsinstitute zusammengefaßt. Aufbauend auf den Resultaten der ersten und zweiten Phase von GLOBUS werden die folgenden Aspekte neben anderen Aufgabenstellungen untersucht:

Aspekt UIS-Architektur und CORBA

Hier zeichnen sich folgende Aufgaben ab:

- a) Fortführung der Arbeiten der GLOBUS-Arbeitsgruppe Dienstedefinition im UIS
- b) Evaluierung und Integration neuer CORBA-Entwicklungen
- c) Prototypische Implementierungen von Anwender-Applikationen auf der Basis von CORBA.

zu a)

Die Zielsetzung der Arbeitsgruppe ist ausgerichtet auf die Definition konkreter Dienste im UIS. Hierzu werden Anwender in die Gruppe einbezogen. Die Arbeiten sind mit dem Projekt 'UIS-RK-Fortschreibung' des FAW abzustimmen.

zu b)

Die Entwicklung der CORBA-Standards und -Implementierungen (sowie auch der Alternativ-Konzepte) ist derzeit noch sehr dynamisch. Im Rahmen von GLOBUS sollte daher kontinuierlich sichergestellt werden, daß neue Produkte und Konzepte aus dem Bereich und dem Umfeld von CORBA, z. B. Tradingmechanismen für die Dienstauswahl, evaluiert und ggfs. in die UIS-Architekturkonzepte integriert werden.

zu c)

Im Rahmen prototypischer Implementierungen sollten mehrere ausgewählte Applikationen unterschiedlicher Komplexität auf unterschiedlichen Plattformen (Rechner, Betriebssysteme, Sprache ...) unter der neuen Architektur auf der Basis von CORBA realisiert werden.

Damit werden folgende Ziele erreicht:

- Es wird geklärt, inwieweit sich die derzeit verfügbaren CORBA-Implementierungen (speziell die in GLOBUS II ausgewählten) in der praktischen Anwendung bewähren und welche Defizite noch bestehen.
- Es wird demonstriert, welche Leistungsvorteile diese Architektur im Rahmen des UIS in der praktischen Anwendung bringt.
- Das Entwicklungsrisiko bei einer Umstellung des UIS auf CORBA-Standards wird deutlich reduziert.

- Es wird gezeigt, daß das Konzept auch auf komplexe Applikationen - wie sie in UIS typisch sind - anwendbar ist.
- Es wird eine Basis für eine Architektur geschaffen, die für den praktischen Einsatz im UIS geeignet ist und in die schrittweise die diversen Dienste des UIS eingehängt werden können.

Als prototypische Applikationen erscheinen geeignet:

- Eine vereinfachte Version des Programms CRAYSIM,
- im Rahmen der Entwicklung des Altlasten-Informationssystems
 - ein Werkzeug für die Erstellung von Umweltbilanzen bei der Altlasten- und Schadensfallbearbeitung,
 - eine Komponente (Analysenplanerstellung) des Expertensystems Umweltgefährlichkeit von Altlasten XUMA,
- eine Teilfunktionalität des RIPS-Pools (auf der Basis einer Vorstudie wegen der Komplexität der Anwendung).

Die hierbei durchzuführenden Arbeiten umfassen die Installation der ausgewählten CORBA-Implementierung auf der Server-Plattform, ggfs. die Aufgliederung der Applikation in Dienste sowie die Implementierung.

Aspekt Auskunftsdienste und Metainformation im UIS

Ziel der künftigen Arbeiten in GLOBUS sollten hier die Integration und Weiterentwicklung der in GLOBUS II und in den anderen UIS-Projekten erarbeiteten Ergebnisse sein. Hierzu ist insbesondere eine Abstimmung der in den Projekten INTEGRAL und WWW-UIS erreichten Resultate und deren Integration in ein gemeinsames UIS-Konzept vorzunehmen. In der Abstimmung und dem Konzept sind der Bedarf und die Anforderungen der unterschiedlichen Nutzer zu berücksichtigen (z. B. globale Suche eines gelegentlichen Benutzers gegenüber lokaler Suche eines Spezialisten innerhalb seines Sachgebiets).

Die Weiterentwicklungen im Rahmen von GLOBUS sollte sich konzentrieren auf:

- Ausbau der Suchfunktionalität; dies beinhaltet:
 - Verbesserung der Verfahren und Werkzeuge zur Volltextsuche,
 - Erweiterungen der Verschlagwortungsmethoden, z. B. durch Einführung von Ober- und Unterbegriffen, durch räumliche Indexierung von Informationen, Verschlagwortung nach geographischen Namen,
 - intelligente Nutzung der semantischen Beziehungen zwischen Schlagworten bzw. der geometrisch-topologischen Beziehungen zwischen Raumbezügen,
- Verbesserung der Methoden zur Metadatenpflege; dies umfaßt:
 - Mechanismen für das Updaten der Metadaten durch den Informationsanbieter,
 - Mechanismen für ein automatisches Eintragen neuer Metainformationen,
- Generische Tools für die Anbindung von Auskunftssystemen:

- Tools für die Einrichtung neuer Anwenderdienste zur Abfrage von Umweltdatenbanken, ohne daß hierzu Softwareexperten herangezogen werden müssen,
- Informationspräsentation, Werkzeuge / Dienste für das Bearbeiten von (Business-) Grafiken, Karten, Bildern etc.:
 - Evaluierung von Produkten,
 - Einbindung in UIS / PC-Arbeitsumgebungen.

Die unter diesen Aspekten (weiter-) entwickelten Werkzeuge sollten direkt im praktischen Betrieb erprobt und eingesetzt werden. Hierfür eignen sich die in GLOBUS II begonnenen Arbeiten zur Erstellung des Altlasten-Informationssystems; speziell die Entwicklung eines integrierten Werkzeugs zur Konvertierung, Verschlagwortung und Volltextsuche.

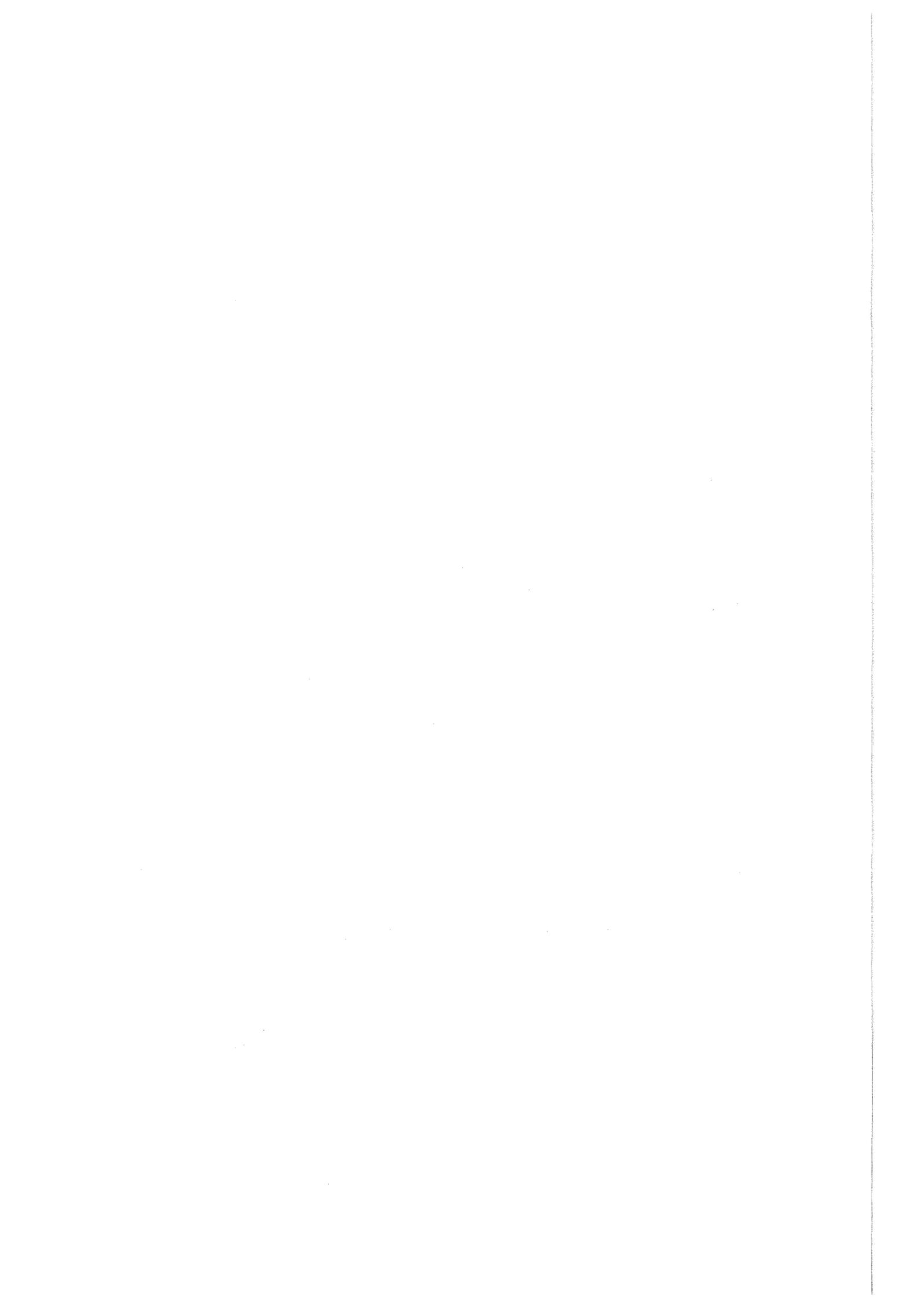
Aspekt Hypermediatechnik für Umweltdaten

Gemeinsam mit dem Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit und dem Umweltbundesamt ist ein F+E-Projekt 'Hypermediatechnik für Umweltdaten' (HUDA) geplant. Im Umweltbereich gibt es fest etablierte Berichte des Bundes und der Länder. Diese sind in der Regel mit Hilfe verschiedener IuK-technischer Hilfsmittel entstanden. Bislang war es nicht möglich, die komplex aufgebauten Umweltberichte aus einem Guß mit Hilfe kooperierender IuK-Anwendungen zu erstellen. Ziel ist die konzeptionelle Entwicklung und softwaretechnische Erstellung eines Werkzeugkastens. Dieser soll die notwendigen Module enthalten, um einen Umweltbericht zu erstellen und gleichzeitig in einer Hypermediaversion verfügbar zu machen, die auf einer CD-ROM und on-line über Weitverkehrsnetze angeboten werden kann. Für diese Aufgabenstellungen soll zusätzlich zu den Instituten des GLOBUS-Konsortiums das Informatik Centrum Dortmund e.V. (ICD) gewonnen werden.

In einer GLOBUS-Konsolidierungsphase ist die Integration weiterer WWW-konformer UIS-Dienste sowie eine enge Abstimmung mit der Neukonzeption des Umwelt-Führungs-Informationssystems erforderlich. Die Prototypen von GLOBUS II, Integral III und WWW-UIS I sollen in einen einheitlichen Systemkern zur Entwicklung einer Betriebsversion überführt werden. Dieser GLOBUS-Systemkern muß bei UM und LfU lauffähig sein.

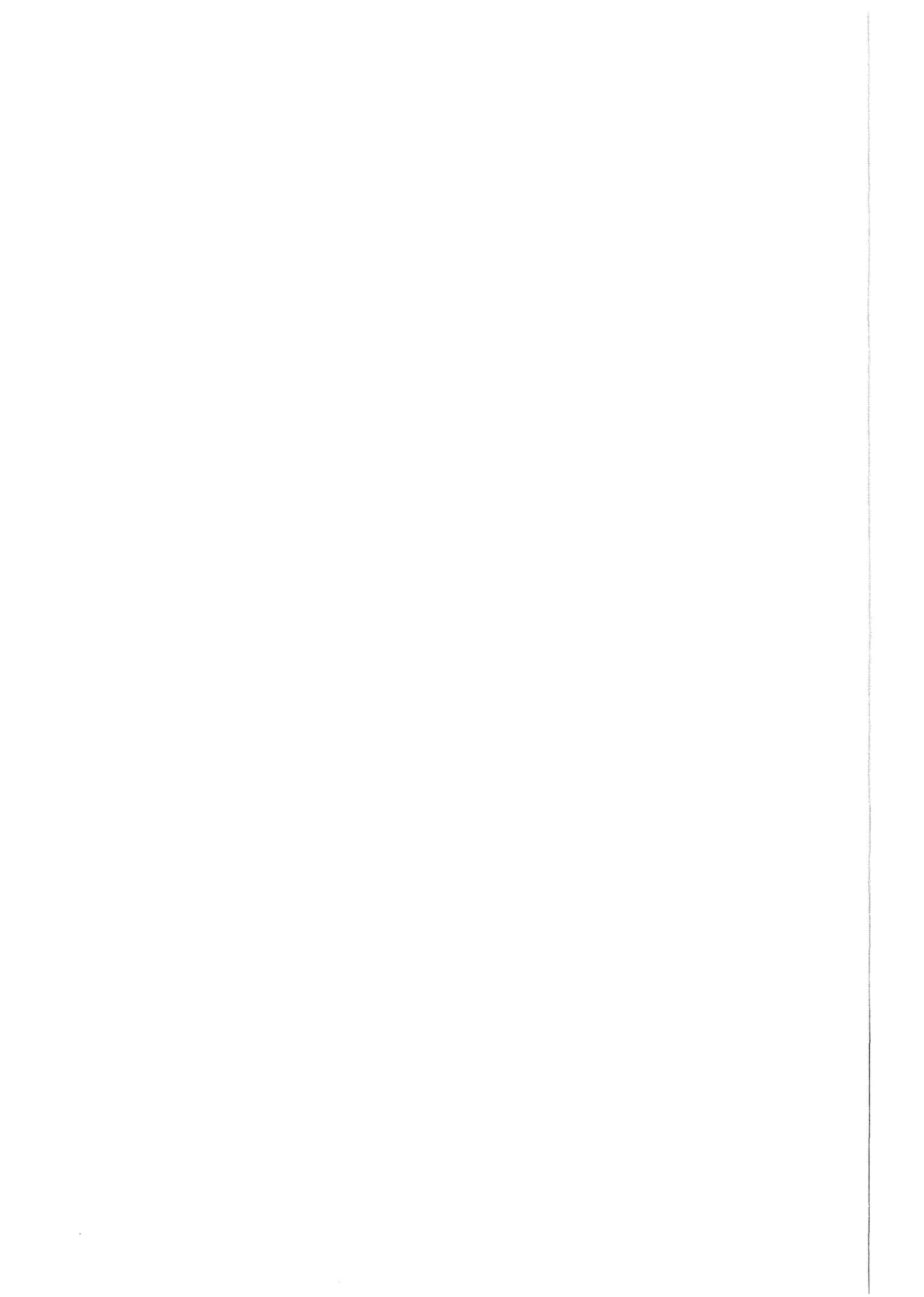
Die vorgesehenen und teilweise bereits begonnenen F+E-Arbeiten sind von Anfang an darauf ausgerichtet, die Wirtschaftlichkeit bei der Erledigung von Umweltaufgaben weiter zu erhöhen. Der inhaltliche Abgleich mit anderen Vorhaben des LSK sowie im Umweltinformatikbereich wird laufend vorgenommen. Kooperationen mit EU, Bund, anderen Ländern und dem kommunalen Bereich werden angestrebt, soweit es die Zielsetzungen des UIS zulassen.

Umweltministerium und Landesanstalt für Umweltschutz Baden-Württemberg werden die Zusammenarbeit mit FAW, FZI, FZK/IAI, IKE und IPF im Rahmen von GLOBUS fortsetzen, um Synergieeffekte mit anderen Vorhaben der Institute zu erreichen und das dort aufgebaute UIS-Know-how zum Vorteil für alle Partner zu nutzen.



Anhang

.....



Architektur des WWW- und CORBA- basierten UIS

- Anlage -

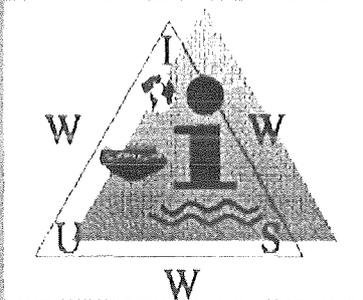
*Arne Koschel, Ralf Kramer, Ralf Nikolai
Forschungszentrum Informatik (FZI)
Haid-und-Neu-Str. 10-14
D-76131 Karlsruhe*

Unter Mitwirkung von:

*Sven Behrens, Uwe Beutler, Corinna Habeck, Philipp Kirsch, Bettina Mayer, Peter Palmer,
Claudia Rolker, Oliver Scheffczyk, Martin Schoel, Manfred Walz*

File Edit View Go Bookmarks Options Directory Help

Location: 



WWW-UIS

Willkommen im Umweltinformationssystem des Landes Baden-Württemberg!

Auskunftsdienste – Suchauswahl

- [Adressen allgemeine Suche](#)
- [Objekte allgemeine Suche](#)
- [Objekte Detailsuche](#)
- [Objekte Schlagwortsuche](#)
- [Objekte Geosuche](#)
- [Objekte Kartensuche](#)

Anmeldung

Um Zugang zum WWW-UIS-System zu erhalten, müssen Sie sich dem System bekannt machen. Sie erhalten eine Benutzerkennung und ein Paßwort zugeteilt. Beim ersten Aufruf der WWW-UIS-Dienste in einer WWW-Sitzung werden Sie nach diesen Angaben gefragt.

- [Anmeldung](#)
- [Paßwort ändern](#)



This page is also available in [english](#) |



La pagina anche disponibile en [italiano](#) |



Esa pagina esta disponible en [español](#) también |



[français](#)

Realisiert im Auftrag des Umweltministeriums Baden-Württemberg

 © (1994/95) FZI Karlsruhe und IPF – Alle Rechte vorbehalten



1. Einstiegsseite in WWW-UIS mit verschiedenen Such- und Navigationsmöglichkeiten

File Edit View Go Bookmarks Options Directory Help

Location: **N**

WWW-UIS

Auskunftsdienst: Adressen allgemeine Suche

Gewünschtes Land:

GER	BB
GER	BE
GER	BL
GER	BW
GER	BY
GER	HB
GER	HE
GER	HH

Bitte geben Sie den Suchbegriff im folgenden Feld ein

Suchmodus:

<input checked="" type="checkbox"/> Teilwort	<input type="checkbox"/> Sensitiv
<input type="checkbox"/> ganzes Wort	<input checked="" type="checkbox"/> Insensitiv

2.1 Suche nach Adressen für „Bodenschutz“ im Bundesland „BW“ (Baden-Württemberg)

File Edit View Go Bookmarks Options Directory Help

Location:

Ausgewähltes Land: GER BW

Anzahl der gefundenen Adressen: 35

Adreßkennzeichen	Institution
<input type="checkbox"/> GERBWL FU 5-51 51	Landesanstalt für Umweltschutz Hildenbrand
<input checked="" type="checkbox"/> GERBWL FU 5-51 511	Landesanstalt für Umweltschutz Manfred Dr. Schöttle
<input type="checkbox"/> GERBWL FU 5-51 511-1	Landesanstalt für Umweltschutz Thomas Dr. Nöltner
<input type="checkbox"/> GERBWL FU 5-51 511-2	Landesanstalt für Umweltschutz Raimund Dr. Kohl
<input type="checkbox"/> GERBWL FU 5-51 511-3	Landesanstalt für Umweltschutz Andreas Hausmann
<input type="checkbox"/> GERBWL FU 5-51 512-1	Landesanstalt für Umweltschutz Volker Dr. Schweikle
<input type="checkbox"/> GERBWL FU 5-51 512-2	Landesanstalt für Umweltschutz Kay Rahtkens
<input type="checkbox"/> GERBWL FU 5-51 513	Landesanstalt für Umweltschutz Dieter Wolf
<input type="checkbox"/> GERBWL FU 5-51 513-1	Landesanstalt für Umweltschutz Manfred Dr. Lehle
<input type="checkbox"/> GERBWWBA HD21 21-1	Amt für Wasserwirtschaft und Bodenschutz Frank
<input type="checkbox"/> GERBWWBA HD21 211-1	Amt für Wasserwirtschaft und Bodenschutz Martin
<input type="checkbox"/> GERBWWBA HD21 212-1	Amt für Wasserwirtschaft und Bodenschutz Heidelberger
<input type="checkbox"/> GERBWWBA HD21 213-1	Amt für Wasserwirtschaft und Bodenschutz Hofer
<input type="checkbox"/> GERBWWBA HD22 22-1	Amt für Wasserwirtschaft und Bodenschutz Künzel

2.2 Ergebnisliste der Adressensuche: Auswahl von „LfU, Dr. Schöttle“ zur detaillierten Beschreibung

The screenshot shows a Netscape browser window titled "Netscape: Adre_detaildarstellung". The address bar contains the URL "http://havanna.fzi.de:9000/cgi-bin/WWW-UIS/SRV_C". The browser interface includes a menu bar (File, Edit, View, Go, Bookmarks, Options, Directory, Help) and a toolbar with icons for Back, Forward, Home, Reload, Images, Open, Print, Find, and Stop. Below the address bar are navigation buttons for "What's New", "What's Cool", "Handbook", "Net Search", "Net Directory", and "Newsgroups".

The main content area displays the following information:

Auskunftsdienst: Adreßdetaildarstellung

1. UDK-Adresse

Institution:
LfU

Name:
Landesanstalt für Umweltschutz

Abteilung:
S-51

Name:
Boden, Abfall, Altlasten

Bereich:
Bodenschutz Referat 51

Strasse:
Griesbachstr. 1

Stadt/PLZ/Ort:
GER 76185 Karlsruhe

Stelle:
511

Name:
Dr. Schöttle

Vorname:
Manfred

Telefon/Fax:
0721/983-1559

Suchbegriffe:
Bodenschutz
Geologie
Landesanstalt für Umweltschutz
LfU

Aufgaben:
Notiz:

Realisiert im Auftrag des Umweltministeriums Baden-Württemberg

©(1994/95) FZI Karlsruhe - Alle Rechte vorbehalten

Document: Done.

2.3 Detaillierte Beschreibung eines Zuständigen für Bodenschutz (LfU, Dr. Schöttle)

File Edit View Go Bookmarks Options Directory Help

Location: 

Auskunftsdienst: Objekte Detailsuche

Gewünschter Katalog:

- GER BW GDK Querschnittsbaum
- GER BW RAS Primärkatalog
- GER BW SBT Querschnittsbaum
- GER BW UDK Primärkatalog
- GER NI GDK Querschnittsbaum
- GER NI UDK Fachbaum
- Alle Kataloge

<input type="text" value="Raumbezug"/>	<input type="text" value="-"/>	<input type="text" value="Wiesloch"/>	<input type="text" value="Oder"/>
<input type="text" value="Objektname"/>	<input type="text" value="-"/>	<input type="text" value="Bergbau"/>	<input type="text" value="Ende"/>
<input type="text" value="Objektname"/>	<input type="text" value="-"/>	<input type="text" value=""/>	<input type="text" value="Ende"/>
<input type="text" value="Objektname"/>	<input type="text" value="-"/>	<input type="text" value=""/>	<input type="text" value="Ende"/>
<input type="text" value="Objektname"/>	<input type="text" value="-"/>	<input type="text" value=""/>	<input type="text" value="Ende"/>

3.1 Suche im UDK Baden-Württemberg nach Objekten zu „Bergbau“ im Raum „Wiesloch“

File Edit View Go Bookmarks Options Directory Help

Location: N

Ausgewählter Katalog: GER BW UDK Primärkatalog
Anzahl der gefundenen Objekte: 11

Objektname

- Schwermetalluntersuchungen im Umfeld des Zementwerks Leimen
01.03.01.14.01.03.01
- Bodenbelastungen durch Bergbau und Aufbereitung von Rohstoffen
01.03.01.14.01.04
- Bodenbelastungen durch Bergbauaktivitäten im Münstertal
01.03.01.14.01.04.03
- Bodenbelastungen durch Bergbauaktivitäten in Wittichen-Schenkenzell
01.03.01.14.01.04.04
- Bodenbelastungen durch Bergbauaktivitäten in Neubulach
01.03.01.14.01.04.05
- RVO des LRA Rhein-Neckar-Kreis über Erklärung der Flächen entlang des Leimbachs – Gewässer
II.Ordnung und Überschwemmungsgebiete, Wiesloch
01.06.03.02.02.01.02.01
- Bergbau Wiesloch
01.06.03.02.07.01.02.02
- Schwermetallbelastung im Bergbaugbiet Wiesloch (Eichteich, Balertal) Schwermetalltransfer Boden/Pflanze
01.06.03.02.07.01.02.02.01
- Boden- und Pflanzenuntersuchungen entlang des Leimbach und Hardtbach Leimbachprogramm
01.06.03.02.07.01.02.02.02
- Bodenuntersuchungen im weiteren Umfeld des Bergbaus Wiesloch St. Ilgen, Nußloch
01.06.03.02.07.01.02.02.05
- Ehemaliges Bergbaugbiet Weinheim
01.06.03.02.07.02.03.06

Realisiert im Auftrag des Umweltministeriums Baden-Württemberg

 ©(1994/95) FZI Karlsruhe – Alle Rechte vorbehalten
Generierungsdatum: 11.12.'95, 19:31 Uhr (Lokalzeit in Karlsruhe)

Document: Done

3.2 Ergebnisliste der Objekte zu „Bergbau“ im Raum „Wiesloch“

File Edit View Go Bookmarks Options Directory Help

Location: http://havanna.fzi.de:9000/cgi-bin/WWW-UIS/SRV_CGI/umlenk **N**



WWW-UIS

Auskunftsdienst: Objektdetailausgabe

1. UDK-Objekt

Objektname:
Schwermetalluntersuchungen im Umfeld des Zementwerks Leimen

Dekadische Notation:
01.03.01.14.01.03.01

Suchbegriff:
Bodenschutz, Bodenbelastung, Thallium, Zementwerk, Schadstoff,

Adresskennzeichen:
GERBWLFU 5-51 511-1

Datenauskunft:
GERBWLFU 5-51 511-1

Freigabe:
j

Datenart:
AO

Beschreibung:
Erhöhte Thalliumgehalte im Umfeld des Zementwerks Leimen wurden in Nahrungs- und Futtermitteln sowie in Böden gefunden. Diese Tatsache führte zu Anbauempfehlungen u.a. für alle Kohlarten sowie für Grünraps, Grünmais und Silomais. Die Tl-Einträge entstanden durch die damals eingesetzten Tl-haltigen Zuschlagstoffe, die daraufhin verboten wurden. Für die Zementherstellung wird hier u.a. Oberer Muschelkalk (z.T. mit Vererzungen s. Bergbau Wiesloch) eingesetzt. Im Umfeld des Zementwerks wurden seit 1979 mehrere Bodenproben (> 300) und Pflanzenproben auf Schwermetall-Gesamtgehalte und seit 1990 zusätzlich auf NH₄NO₃-extrahierbare (mobile) Gehalte analysiert.

Notiz:
s. Untersuchungen des RP KA, WBA HD, LUFA Augustenberg, Diss. Andreas Prödel 1993,

Document Done

3.3 Detaillierte Beschreibung der „Schwermetalluntersuchungen im Bereich des Zementwerks Leimen“, 1. Teil

File Edit View Go Bookmarks Options Directory Help

Location: 

Erläuterung:

Raumbezug

Land:
Baden-Württemberg

Raumeinheit:
ZWL

Kreis:
Rhein-Neckar-Kreis

Gemeinde:
Leimen

Karte:
kreis.gemeinde

Maßstab:
1:25.000

Blattnummer:
6618

Blattname:
Heidelberg Süd

Koord.system:
Koordinaten
X: 34 68 250
Y: 54 49 750
Z:

Bounding Box
X1,2:
Y1,2:
Z1,2:

Erläuterung:
Untersuchungsgebiet: Leimen, HD Süd, Wiesloch, Nußloch

[Zur Kartendarstellung](#)

Zeitbezug

von: 1979

bis: 1994

Periode:
Untersuchungen von verschiedenen Ämtern und Institutionen

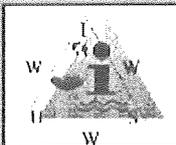
Erläuterung:
Untersuchungen von verschiedenen Ämtern und Institutionen

Document Done

3.4 Detaillierte Beschreibung der „Schwermetalluntersuchungen im Bereich des Zementwerks Leimen“, 2. Teil

File Edit View Go Bookmarks Options Directory Help

Location: 



WWW-UIS

Auskunftsdienst: Thesaurus Begriffssuche

[Anleitung zum Ausfüllen dieses Formulars](#)

Bitte geben Sie den Suchbegriff im folgenden Feld ein:

Suchmodus:

<input checked="" type="checkbox"/> Teilwort	<input checked="" type="checkbox"/> Sensitiv
<input checked="" type="checkbox"/> ganzes Wort	<input checked="" type="checkbox"/> Insensitiv

Realisiert im Auftrag des Umweltministeriums Baden-Württemberg

 ©(1994/95) FZI Karlsruhe – Alle Rechte vorbehalten
Generierungsdatum: 11.12.'95, 19:38 Uhr (Lokalzeit in Karlsruhe)



4.1 Suche im UBA - Thesaurus nach dem Begriff „Boden“

File Edit View Go Bookmarks Options Directory Help

Location: 



WWW-UIS

Auskunftsdienst: Thesaurusliste

Anzahl der gefundenen Begriffe: 86

Begriffe

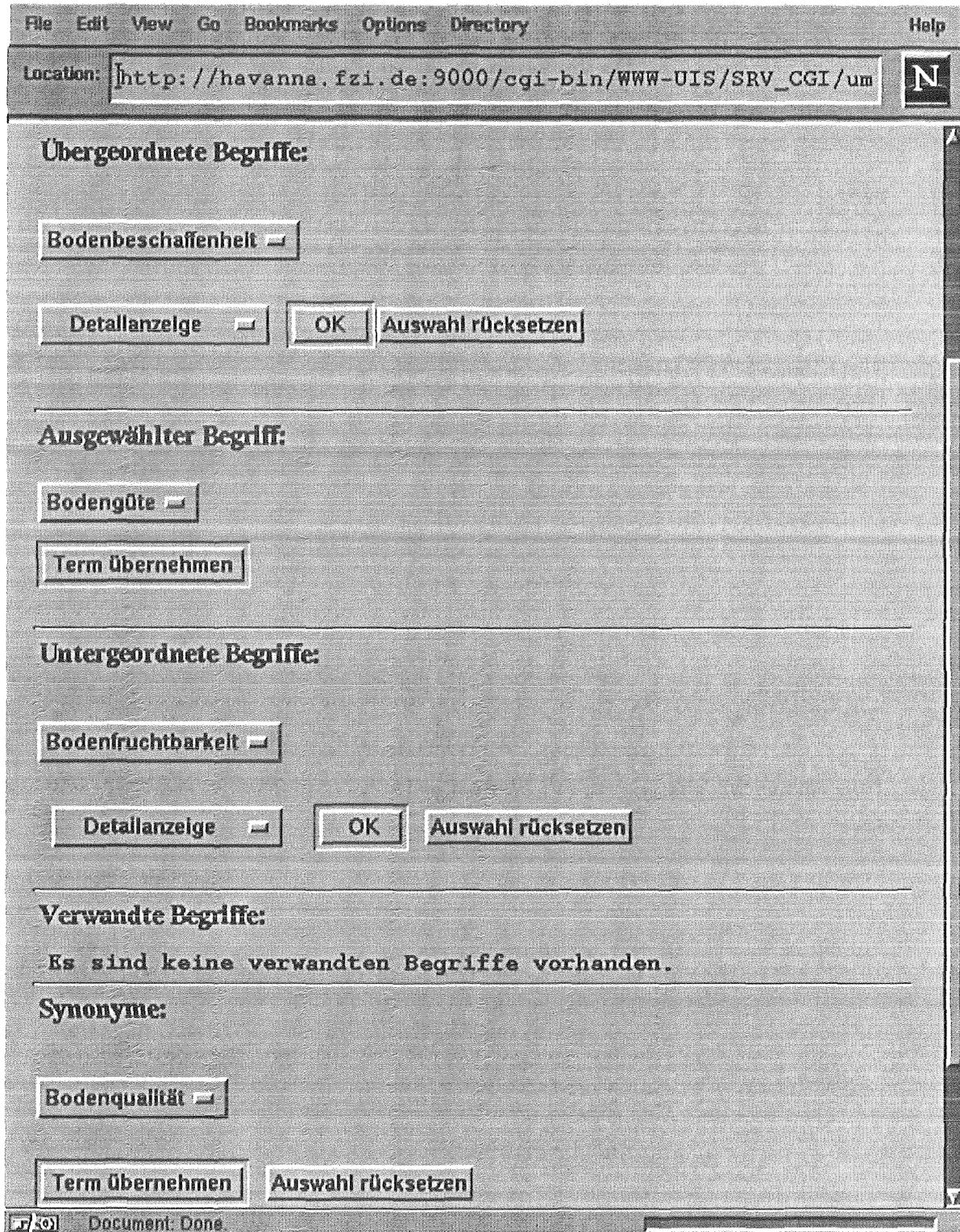
Bodenfiltration
Bodenflora
Bodenfruchtbarkeit
Bodenfunktion
Bodengüte
Bodenhilfsstoff
Bodenkalkung
Bodenkarte

Realisiert im Auftrag des Umweltministeriums Baden-Württemberg

 ©(1994/95) FZI Karlsruhe - Alle Rechte vorbehalten
Generierungsdatum: 11.12.'95, 19.39 Uhr

Document Done

4.2 Ergebnisliste der Thesaurussuche zu „Boden“, Auswahl des Begriffs „Bodengüte“



4.3 Anzeige übergeordneter, untergeordneter und verwandter Thesaurusbegriffe sowie Synonyme zu „Bodengüte“

Nutzung von Diensten im WWW mit dem Common Client Interface (CCI)

- Anlage -

*R. Kopetzky; A. Lurk; F. Schmidt
Universität Stuttgart,
Institut für Kernenergetik und Energiesysteme (IKE)
Pfaffenwaldring 31
D-70569 Stuttgart*

A.1 CCI RÜCKGABE-CODES.....	319
A.2 EXCEL BEISPIELMAKROS	319
A.3 CCI-BEISPIEL-PROGRAMME IN C.....	320
A.4 QUELLCODE FÜR DAS FUNKTIONSMODUL ZWERG().....	323
A.5 FEHLERMELDUNGEN DES KIP.....	327
A.6 ABBILDUNGEN ZUM BEISPIEL "KIP UND CCI"	328

A.1 CCI Rückgabe-Codes

MCCIR_OK	200	
MCCIR_GET_OK	210	
MCCIR_DISPLAY_OK	211	
MCCIR_DISCONNECT_OK	212	
MCCIR_QUIT_OK	213	
MCCIR_SEND_OUTPUT_OK	214	
MCCIR_SEND_O_STOP_OK	215	
MCCIR_SEND_ANCHOR_OK	216	
MCCIR_SEND_A_STOP_OK	217	
MCCIR_POST_OK	218	
MCCIR_SEND_ANCH_BEF_LINK_OK	220	/* clicked link */
MCCIR_SEND_ANCH_BEF_OPEN_OK	221	/* used open dialog */
MCCIR_SEND_ANCH_BEF_EDIT_OK	222	/* edited URL field */
MCCIR_SEND_ANCH_BEF_FORW_OK	223	/* clicked forward */
MCCIR_SEND_ANCH_BEF_BACK_OK	224	/* clicked back */
MCCIR_SEND_ANCH_BEF_RELO_OK	225	/* clicked reload */
MCCIR_SEND_ANCH_BEF_HIST_OK	226	/* from history */
MCCIR_SEND_ANCH_BEF_HOTL_OK	227	/* from hotlist */
MCCIR_SEND_ANCH_BEF_MCCI_OK	228	/* from CCI */
MCCIR_SEND_ANCH_BEF_OTHR_OK	229	/* from other source */
MCCIR_SEND_ANCH_AFT_LINK_OK	230	/* clicked link */
MCCIR_SEND_ANCH_AFT_OPEN_OK	231	/* used open dialog */
MCCIR_SEND_ANCH_AFT_EDIT_OK	232	/* edited URL field */
MCCIR_SEND_ANCH_AFT_FORW_OK	233	/* clicked forward */
MCCIR_SEND_ANCH_AFT_BACK_OK	234	/* clicked back */
MCCIR_SEND_ANCH_AFT_RELO_OK	235	/* clicked reload */
MCCIR_SEND_ANCH_AFT_HIST_OK	236	/* from history */
MCCIR_SEND_ANCH_AFT_HOTL_OK	237	/* from hotlist */
MCCIR_SEND_ANCH_AFT_MCCI_OK	238	/* from CCI */
MCCIR_SEND_ANCH_AFT_OTHR_OK	239	/* from other source */
MCCIR_ANCHOR_INFO	301	
MCCIR_SEND_DATA_OUTPUT	302	
MCCIR_POST_OUTPUT	303	
MCCIR_UNRECOGNIZED	401	/* what's this? */
MCCIR_ERROR	402	/* does not follow protocol */
MCCIR_REQ_FAILED	500	/* request failed */
MCCIR_GET_FAILED	501	/* request failed */

A.2 Excel Beispielmakros

LoadMosaic() öffnet die OLE Verbindung zu Mosaic:

```
Dim Mosaic_CCI_Obj Als Objekt
```

```
Sub LoadMosaic()
```

```
    Setze Mosaic_CCI_Obj = ErstelleObjekt("Mosaic.MosaicCCI")
```

```
Ende Sub
```

UnloadMosaic() schließt die OLE Verbindung zu Mosaic:

```
Sub UnloadMosaic()
```

```
Setze Mosaic_CCI_Obj = Nichts  
Ende Sub
```

SlideShow() setzt verschiedene Methoden ein, um eine "Slide-Show" ablaufen zu lassen, wobei mit dem URL in der aktiven Zelle begonnen und so lange die URL der darunter liegenden Zelle geladen wird, bis eine leere Zelle erreicht wird:

```
Sub SlideShow()  
Item = AktiveZelle.Wert  
  Durchlaufe Solange Item <> ""  
    Mosaic_CCI_Obj.ResolveUrl (Item)  
    Anwendung.Warten Jetzt + Zeitwert("00:00:05")  
    Item = Mosaic_CCI_Obj.GetCurrentDocUrl()  
    Wenn (Item = AktiveZelle.Wert) Dann  
      AktiveZelle.Versetzen(1; 0).Aktivieren  
      Item = AktiveZelle.Wert  
    Sonst  
      Item = ""  
    Ende Wenn  
  Schleife  
Ende Sub
```

GetAnchors() nutzt die Methoden GetAnchorCount(), GetAnchorUrl() und GetAnchorDesc(), um die URL's des aktuellen Dokumentes mit Beschreibung zu listen:

```
Sub GetAnchors()  
  Index = 1  
  Total = Mosaic_CCI_Obj.GetAnchorCount()  
  AktiveZelle.Wert = Total  
  AktiveZelle.Versetzen(1; 0).Aktivieren  
  Durchlaufe Solange Index <= Total  
    DocUrl = Mosaic_CCI_Obj.GetAnchorUrl(Index)  
    DocDsc = Mosaic_CCI_Obj.GetAnchorDesc(Index)  
    AktiveZelle.Wert = DocDsc  
    AktiveZelle.Versetzen(0; 1).Wert = DocUrl  
    AktiveZelle.Versetzen(1; 0).Aktivieren  
    Index = Index + 1  
  Schleife  
Ende Sub
```

A.3 CCI-Beispiel-Programme in C

cciget.c

Dieses Beispielprogramm stellt die Verbindung mit Mosaic her und weist es an, eine vorgegebene URL anzufordern.

Verwendung finden die Funktionen MCCIInitialize(), MCCIConnect(), MCCIGet() und MCCIDisconnect().

- * Anwendung: cciget <host> <port> <URL>
- * Beispiel:

```
* cciget 129.69.25.7 2048 http://www.rus.uni-stuttgart.de/

#include <stdio.h>
#include "cci.h"

main(int argc, char **argv)
{
    MCCIPort serverPort;
    char *host;      * Rechner, auf dem Mosaic läuft
    int port;       * Port, auf dem Mosaic auf CCI-Verbindungen wartet
    char *url;      * URL, die geladen werden soll
    * Kontrolle der Anzahl der Parameter
    if (argc<4) {
        fprintf(stderr, "Aufruf: %s host port URL\n", argv[0]);
        exit(0);
    }
    * Zuweisung der Parameter
    host = argv[1];
    port = (int)argv[2];
    url = argv[3];
    * Initialisierung der CCI-Bibliothek
    MCCIInitialize();
    * Verbindungsaufbau mit Mosaic
    serverPort = (MCCIPort) MCCIConnect(host, port, 0, 0);
    * Überprüfung der Verbindung
    if (!serverPort) {
        printf("Keine Verbindung mit %s:%d !\n", host, port);
        exit(0);
    }
    * Anweisung an Mosaic, die URL zu laden
    MCCIGet(serverPort, url, MCCI_DEFAULT, MCCI_ABSOLUTE, 0);
    * Verbindung mit Mosaic beenden
    MCCIDisconnect(serverPort);
}
```

cs0.c

Dieses Beispielprogramm zeigt die Verwendung des MCCISendOutput()-Aufrufs. Mosaic wird dabei angewiesen, alle eingehenden image/gif und image/jpeg - Typen an diese Anwendung zu schicken.

Folgende Funktionen werden verwendet: MCCIInitialize(), MCCIConnect(), MCCISendOutput(), MCCIPoll() und MCCIDisconnect()

```
#include <stdio.h>
#include "cci.h"

* callBack()-Funktion, die aufgerufen wird, wenn Mosaic die Verbindung abbricht.
void DisconnectCB(MCCIPort port, void callBackData)
{
    printf("Mosaic hat die Verbindung abgebrochen \n");
    printf("Die Anwendung wird beendet...\n");
}

* callBack()-Funktion, die aufgerufen wird, wenn von Mosaic Daten übertragen
werden, die sich auf die
* Funktion MCCISendOutput() beziehen.
```

```
void ReceiveOutput(char *type, char *data, int length,
void callBackData)
{
FILE *fp;
char *dateiname;
char displayCommand[1024];
* Schreiben der Daten in eine temporäre Datei
dateiname = tmpnam(NULL);
fp = fopen(dateiname,"w");
fwrite(data,1,length,fp);
fclose(fp);
* Aufruf des vi-Editors mit Übergabe der temporären Datei
sprintf(displayCommand,"vi %s",filename);
system(displayCommand);
* Löschen der temporären Datei
unlink(filename);
}

* Beginn des Hauptprogramms
main(int argc, char **argv)
{
MCCIPort serverPort;
char *host;          * Rechner, auf dem Mosaic läuft
int port;           * Port, auf dem Mosaic auf CCI-Verbindungen wartet
* Kontrolle der Anzahl der Parameter
if (argc<3) {
    fprintf(stderr,"Aufruf: %s host port\n",argv[0]);
    exit(0);
}
* Zuweisung der Parameter
host = argv[1];
port = (int)argv[2];
* Initialisierung der CCI-Bibliothek
MCCIInitialize();
* Verbindungsaufbau mit Mosaic
serverPort = (MCCIPort) MCCIConnect(host,port,0,0);
* Überprüfung der Verbindung
if (!serverPort) {
    printf("Keine Verbindung mit %s:%d !\n",host,port);
    exit(0);
}
* Anweisung an Mosaic, OUTPUT vom MIME-Typ "text/html" zu schicken
MCCISendOutput(serverPort,"text/html",ReceiveOutput,0);
* Kontrolle, ob Mosaic Daten sendet
while(-1 != MCCIPoll(serverPort)) {
    sleep(1);
}
* MCCIPoll liefert den Wert "-1", wenn die Verbindung von Mosaic abgebrochen
* wird. Für diesen Fall wird die bei Verbindungsaufbau angegebene callBack()-
* Funktion DisconnectCB()
* aufgerufen, bevor das Programm weiter abgearbeitet wird.
exit(0);
}
```

A.4 Quellcode für das Funktionsmodul ZwErg()

Mit der Funktion

```
int ZwErg(MCCIPort cciport, char datentyp char *datenbody)
```

können über CCI Zwischenergebnisse vom MIME-Typ text/html oder image/gif an einen Web-Client übermittelt werden. Dazu müssen folgende Header-Dateien in das Hauptprogramm eingebunden werden:

```
#include <stdio.h> C-Standard Header  
#include <string.h> C-Standard Header  
#include <stdlib.h> C-Standard Header  
#include <cci.h> Header der CCI-Bibliothek
```

Desweiteren ist die Einbindung der Funktion ZwErg() und der callBack()-Funktion DisconnectCB() erforderlich:

```
#include "ZwErg.c"
```

Die Kompilierung muß zusammen mit der Bibliotheks-Datei "libcci.a" erfolgen.

Beispiel: `cc test.c libcci.a -o test`

Schnittstellen-Variable:

cciport : normalerweise 2048 (oder von Web-Client vorgegeben)

datentyp : "html" oder "gif"

datenbody: entsprechende Daten im HTML-oder GIF-Format

length : Länge des datenbody in Byte

Rückgabe-Codes:

0 - Ergebnismrückgabe erfolgreich

1 - Fehler: Kein oder falscher Datentyp

2 - Fehler: Keine Daten zur Rückgabe

3 - Fehler: HTML-Datei kann nicht geöffnet werden

4 - Fehler: GIF-Datei kann nicht geöffnet werden

5 - Verbindung zu Slave-Mosaic kann nicht aufgebaut werden

Quellcode von ZwErg():

```
* Von MCCICConnect() benötigte callBack()-Funktion, die bei
```

```
* Abbruch der CCI-Verbindung durch Mosaic aufgerufen wird.
```

```
void DisconnectCB(MCCIPort port, void *callBackData)
```

```
{  
}
```

```
* Funktionsmodul ZwErg()
```

```
int zwerg(int cciport, char *datentyp, char *datenbody, int length)
```

```
{
```

```
    MCCIPort serverPort;          *CCI-Port des Slave-Mosaic
```

```
    FILE *fhtml, *fgif;          *File-Pointer Rückgabe-Dateien
```

```
    char *htmlname, *gifname;    *Namen Rückgabe-Dateien
```

```
    int ierr, gif_datei, geschr;
```

```
    char mimetype;
```

```
char *host;
char url[200];

* Setzen der Default-Werte
ierr=0;
gif_datei=0;          * "0" = HTML, "1" = GIF
htmlname="ZwErg.html";
gifname="ZwErg.gif";

* Schreiben der Zwischenergebnisse in die Rückgabe-Datei(en)
while(1) {

* Kontrolle der Variablen datentyp
  if((strcmp(datentyp, "html") && strcmp(datentyp, "gif"))) {
    ierr=1;
    break;
  }

* Kontrolle der Variablen datenbody
  if(length==0) {
    ierr=2;
    break;
  }

* Öffnen der Rückgabedatei(en)
  fhtml=fopen(htmlname, "w");
  if(fhtml==NULL) {
    ierr=3;
    break;
  }

  if(!strcmp(datentyp, "gif")) {
    gif_datei=1;
    fgif=fopen(gifname, "w");
    if(fgif==NULL) {
      ierr=4;
      break;
    }
  }

* Schreiben der Daten in GIF-Datei
  fwrite(datenbody, 1, length, fgif);

* Schließen der GIF-Datei
  fclose(fgif);
}

* Schreiben der HTML-Daten; bei GIF-Daten wird ein Verweis auf die GIF-Datei
* übergeben.
  if(gif_datei==1)
    fputs("<IMG SRC=\"ZwErg.gif\" align=middle>\n", fhtml);
  else
    fwrite(datenbody, 1, length, fhtml);

* Schließen der HTML-Datei
  fclose(fhtml);
  break;
}

* Damit ist die Speicherung des Zwischenergebnisses abgeschlossen.
* Es folgt der Verbindungsaufbau mit Mosaic und der Befehl, die
```

```
* Zwischenergebnisse darzustellen.
while((1)&&(ierr==0))
{

* Initialisierung der CCI-Bibliothek
  MCCIInitialize();

* Auslesen der Server-spezifischen Umgebungsvariablen
  host=getenv("REMOTE_ADDR");
* Erstellen der URL auf die HTML-Datei
  sprintf(url,"http://%s%/ZwErg.html",getenv("SERVER_NAME"),
          getenv("EXTRA_PATH"));

* Verbindungsaufbau mit Slave-Mosaic
  serverPort=(MCCIPort)MCCIConect(host,cciport,DisconnectCB,0);
  if(!serverPort){
    ierr=5;
    break;
  }

* Auftrag an Slave-Mosaic, Zwischenergebnisse über den Server
* anfordern
  MCCIGet(serverPort,url,"MCCI_DEFAULT","MCCI_DEFAULT",0);

* Schließen der CCI-Verbindung
  MCCIDisconnect(serverPort);
  break;
}

* Löschen der angelegten Rückmeldedatei(en)
if((ierr>3)|| (ierr==0))
  unlink(htmlname);
if((gif_datei==1)&&((ierr>4)|| (ierr==0)))
  unlink(gifname);

* Rückgabe des Fehler-Codes
return ierr;

}
```

Es folgt ein Beispielprogramm, in dem die Einbindung von ZwErg() in ein simuliertes "Gateway"-Programm demonstriert wird. Das Testprogramm liest GIF-Daten aus einer Datei und übergibt diese an ZwErg() zur Darstellung in Mosaic.

```
#include <stdio.h>
#include <stddef.h>
#include <stdin.h>
#include <string.h>
#include "cci.h"

#include "zwerg.c"

main()
{
  MCCIPort cciport;
  char *datenbody;
  char datentyp[5];
  int fehler;
```

```
FILE *fp;
int c,length;

cciport=2048;
strcpy(datentyp,"gif");

* Lesen der GIF-Daten aus der Datei "icon.gif" in datenbody; dazu wird die
* Länge der Datei bestimmt, der erforderliche Speicherplatz angefor-
* dert und die Daten in datenbody abgelegt.
fp=fopen("icon.gif","r");
length=0;
while((c=getc(fp))!=EOF)
    length++;
datenbody=(char *)malloc(length);
rewind(fp);
fread(datenbody,1,length,fp);
fclose(fp);

* Aufruf der Funktion ZwErg() zur Übertragung der Daten
fehler=zwerg(cciport,datentyp,datenbody,length);

* Fehlerbehandlung
if(fehler==0)
    printf("Ergebnisübertragung erfolgreich!\n");
else
    printf("Fehler Nr. %d\n",fehler);

* Freigabe des Speichers für datenbody
free(datenbody);
}
```

A.5 Fehlermeldungen des KIP

- 301 Format der Datumsangabe ist inkorrekt (siehe Kapitel Fehler! Verweisquelle konnte nicht gefunden werden.).
- 302 Die Datumsangabe entspricht keinem zulässigen Datumswert.
- 550 Die Umgebungsvariablen des Kommunikationsinterpreters sind nicht richtig oder gar nicht gesetzt (siehe Kapitel Fehler! Verweisquelle konnte nicht gefunden werden.).
- 1001 Die zu sendende Datei kann nicht geöffnet werden. Überprüfen Sie die Existenz der angegebenen Datei im angegebenen Verzeichnis.
- 1005 Die Interprozesskommunikation konnte nicht durchgeführt werden. Damit konnte nicht auf den Kommunikationsinterpreter zugegriffen werden. Dies kann durch eine unzulässige Verzeichnisangabe der Umgebungsvariablen KIP_SPOOL verursacht werden.
- 1006 Die Verbindung zum Server konnte nicht aufgebaut werden.
- 1007 Unter der angegebenen Portnummer der Datentransferkomponente des Servers im Auskunftsdienst (in der Datei kad.dat) ist kein Verbindungsaufbau möglich.
- 1008 Mit dem Namen bzw. der Telefonnummer des Servers im Auskunftsdienst (in der Datei kad.dat) ist kein Verbindungsaufbau möglich
- 10001 Der Wert der Art der Inanspruchnahme liegt außerhalb des Wertebereiches (siehe Kapitel Fehler! Verweisquelle konnte nicht gefunden werden.).
- 12001 Die Verbindung zwischen der Anwendung und dem Kommunikationsinterpreter wurde unterbrochen. Dieser Fehler tritt auf, wenn der Prozess des Kommunikationsinterpreters oder der Datentransferkomponenten unvorhergesehen abbricht.
- 12101 Ungültige Dienstauftragsnummer.
- 12110 Fehler beim Laden der Datei kad.dat. Diese entspricht nicht der Beschreibung aus dem Kapitel Fehler! Verweisquelle konnte nicht gefunden werden
- 12201 Der gewünschte Dienst ist dem Kommunikationsinterpreter unbekannt.
- 12202 Der ID Schlüssel berechtigt nicht zum Zugriff auf den Dienst.
- 12260 Die Datei kad.dat kann nicht geöffnet werden.
- 12401 Die Benutzerkennung erlaubt den gewünschten Zugriff nicht.
- 12402 Unter der angegebenen Auftragsnummer besteht kein Dienstauftrag.
- 12403 Es besteht schon ein Auftrag mit der gleichen Benutzerkennung und Auftragsbezeichnung (siehe Kapitel Fehler! Verweisquelle konnte nicht gefunden werden.).
- 12580 Der Dienstauftrag befindet sich in einem Zustand, der die gewünschte Aktion nicht erlaubt.
- 12581 Die Nummer der Datei liegt außerhalb des angegebenen Wertebereichs.
- 12582 Die Datei mit der übergebenen Nummer ist schon vorhanden.
- 12583 Die gewünschte Art der Dienstinanspruchnahme ist nicht möglich.
- 12584 Die maximale Anzahl der Eingabedateien hat einen unzulässigen Wert.
- 12585 Die Anzahl der Eingabedateien innerhalb des Formulars hat einen unzulässigen Wert.
- 12586 Die maximale Anzahl der Ergebnisdateien hat einen unzulässigen Wert.
- 12587 Die Anzahl der Ergebnisdateien innerhalb des Formulars hat einen unzulässigen Wert.
- 12588 Auf die Datei mit den Daten kann nicht zugegriffen werden.

A.6 Abbildungen zum Beispiel "KIP und CCI"

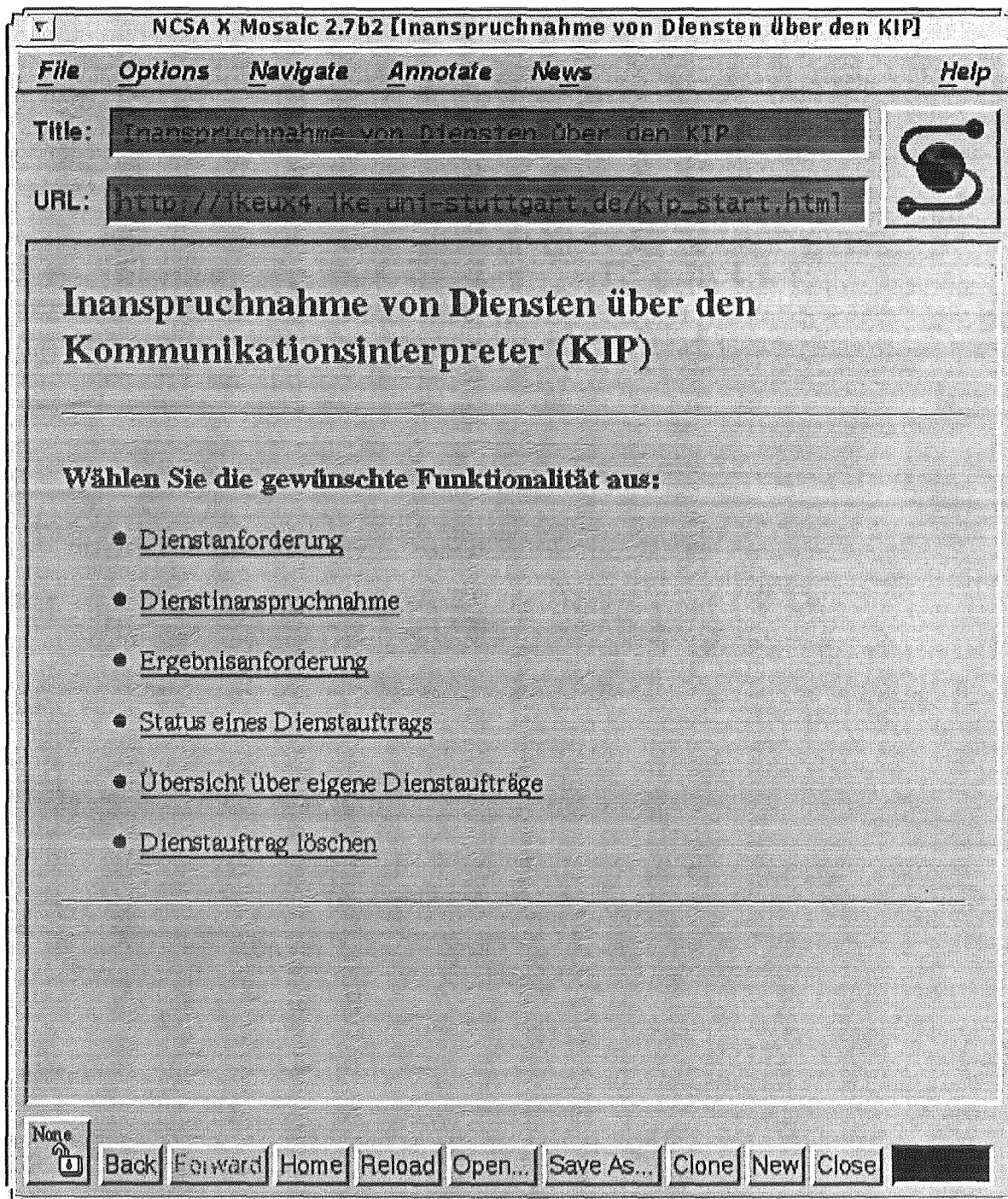


Abbildung A1 - Inanspruchnahme von Diensten über den Kommunikationsinterpreter

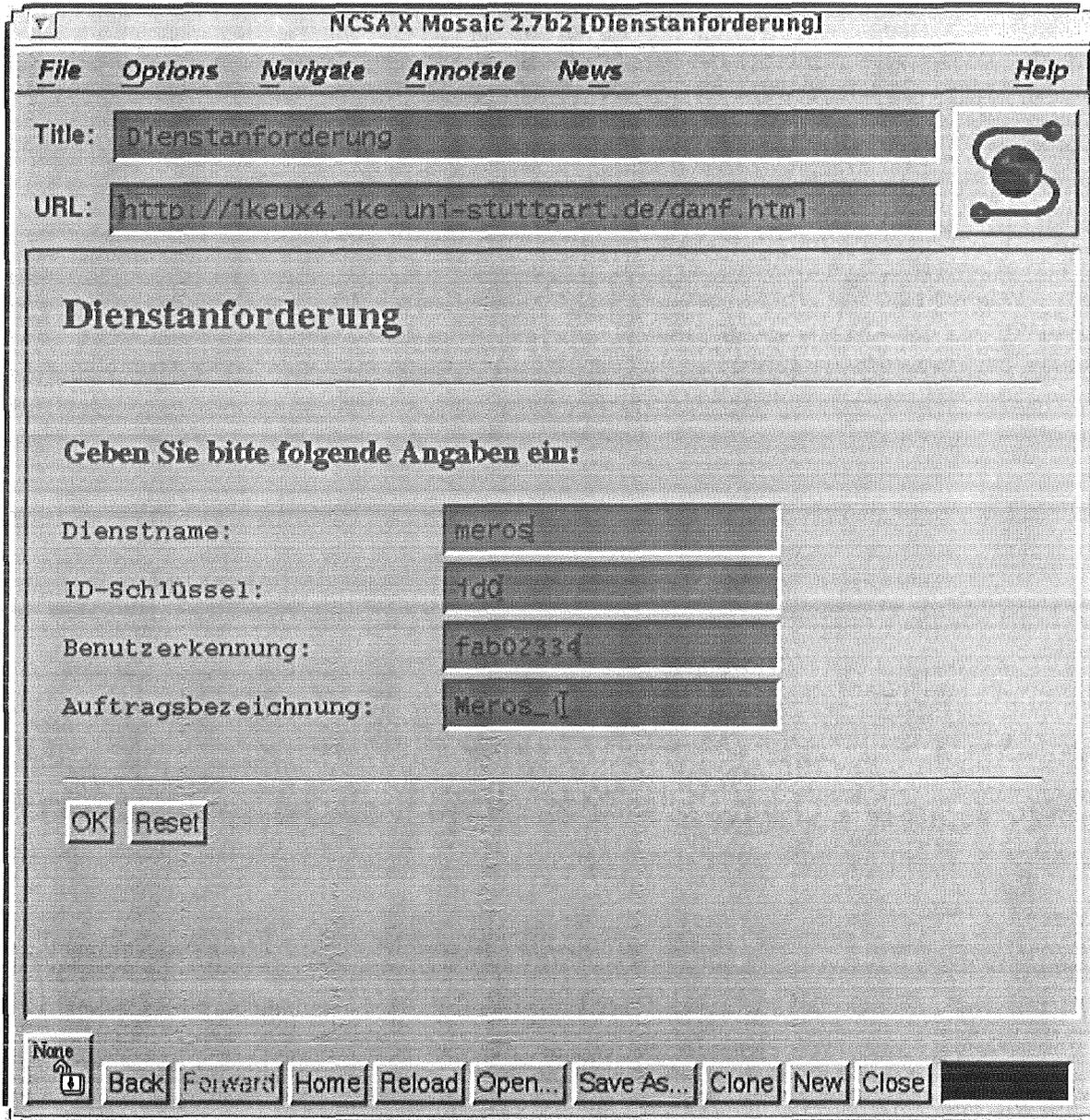


Abbildung A2 - HTML-Seite "Dienststanforderung"

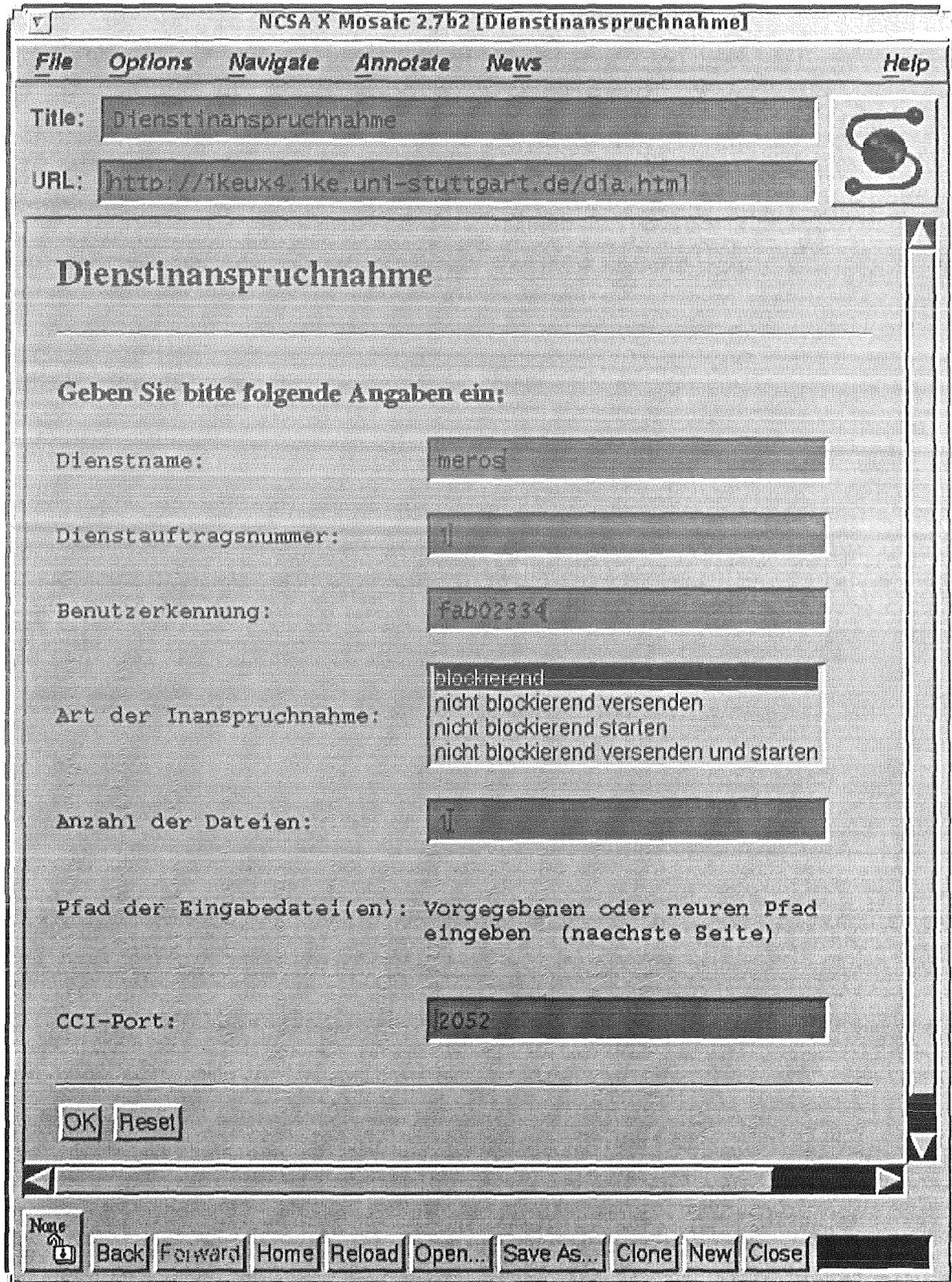


Abbildung A3 - HTML-Seite "Dienstinanspruchnahme"

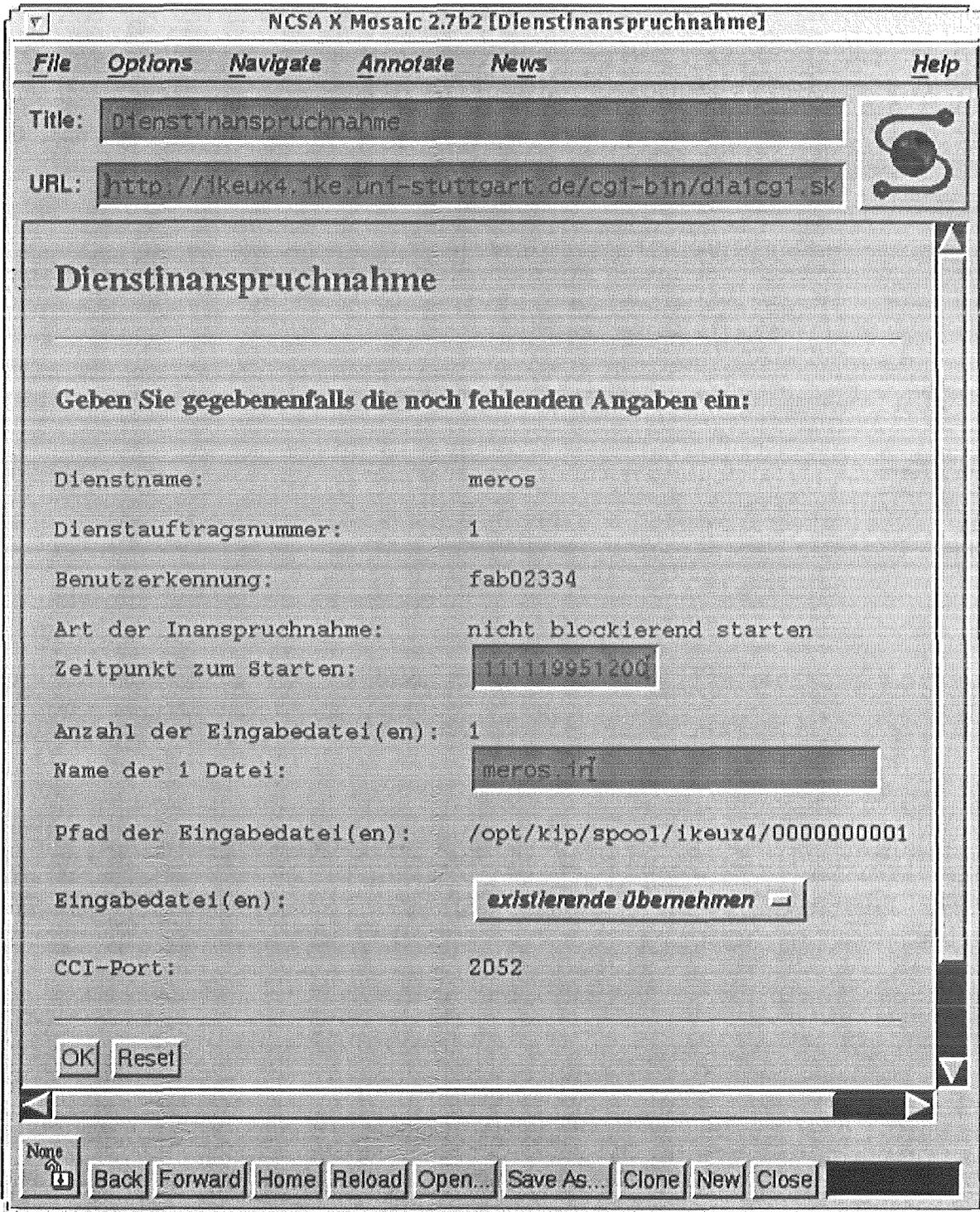


Abbildung A4 - HTML-Seite mit weiteren Werten für die Dienstinanspruchnahme

NCSA X Mosaic 2.7b2 [DienstInanspruchnahme]

File Options Navigate Annotate News Help

Title: dienstInanspruchnahme

URL: http://ikeux4.ike.uni-stuttgart.de/cgi-bin/meroscgi.sk

DienstInanspruchnahme

Neue Eingabedatei(en) für den Dienst "meros" erstellen

Geben Sie bitte die Werte für die Datei "meros.in" ein:

Steuercode:	<input type="text"/>	keine Aggregation
Meßstellenummer des Lfu-MVZ:	<input type="text" value="4444"/>	
Parameternummer des Lfu-Parameterverzeichnis:	<input type="text" value="336"/>	
Meßstypnummer der Lfu-Schlüssellistenverwaltung:	<input type="text" value="1"/>	
Meßobjektnummer der Lfu-Schlüssellistenverwaltung:	<input type="text" value="1"/>	
Dimensionsnummer des Lfu-Dimensionsverzeichnis:	<input type="text" value="0"/>	
Bezugszeitraum Jahr:	<input type="text" value="0"/>	
Bezugszeitraum Monat:	<input type="text" value="0"/>	
Bezugszeitraum Tag:	<input type="text" value="0"/>	
Bezugszeitraum Stunde:	<input type="text" value="0"/>	
Bezugszeitraum Minute:	<input type="text" value="30"/>	
Periode Jahr:	<input type="text" value="0"/>	
Periode Monat:	<input type="text" value="0"/>	

None Back Forward Home Reload Open... Save As... Clone New Close

Abbildung A5 - HTML-Seite zum Erstellen einer Eingabedatei für den Dienst Meros

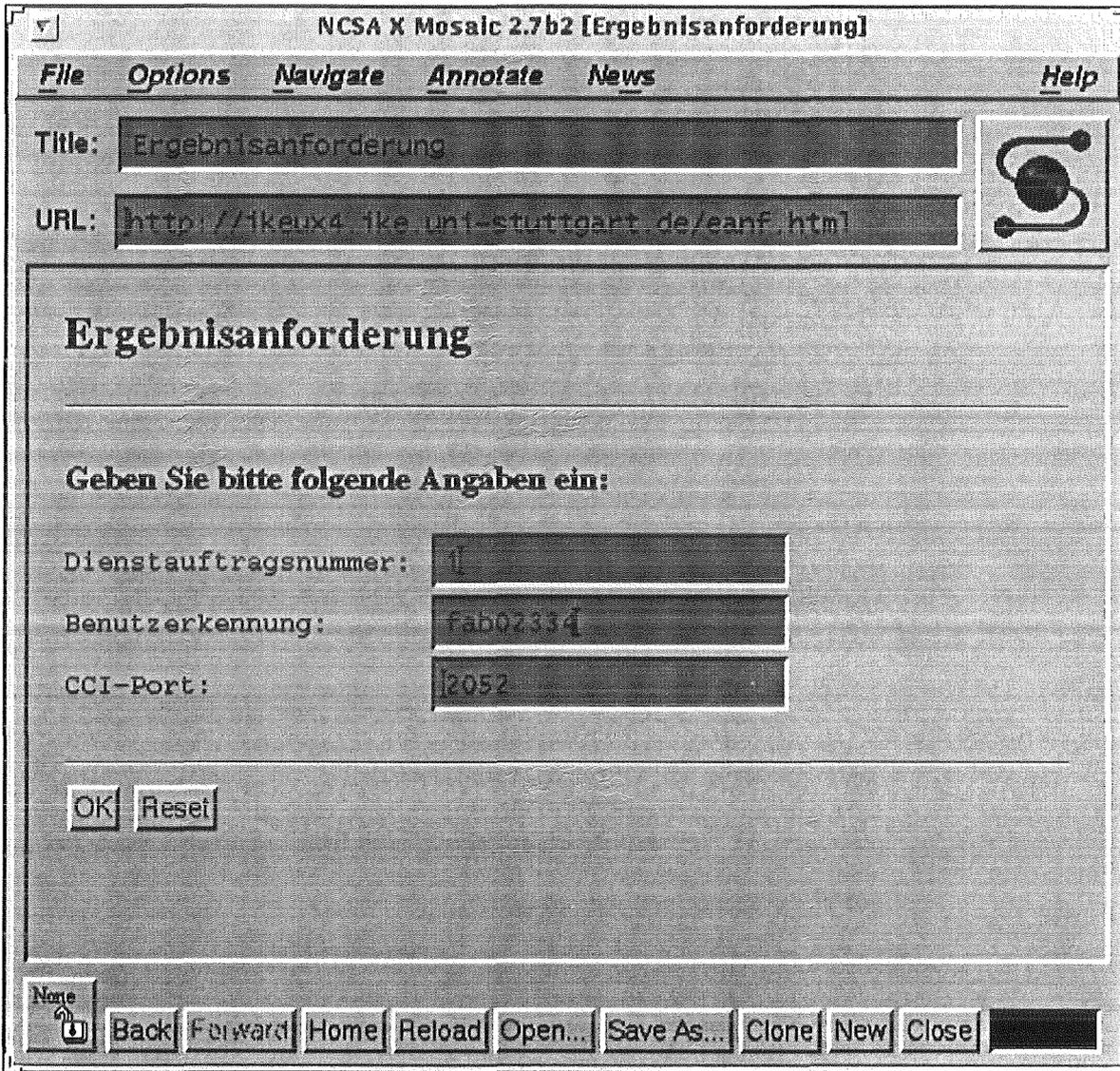


Abbildung A6 - HTML-Seite "Ergebnisanforderung"

The screenshot shows the NCSA X Mosaic 2.7b2 browser window. The title bar reads "NCSA X Mosaic 2.7b2 [Untitled, URL http://ikeux4.ike.uni-stuttgart.de/meros.dat]". The menu bar includes "File", "Options", "Navigate", "Annotate", "News", and "Help". The "Title:" field contains "Untitled, URL http://ikeux4.ike.uni-stuttgart.de/meros.dat" and the "URL:" field contains "http://ikeux4.ike.uni-stuttgart.de/meros.dat". The main content area displays a table with 20 rows of data. The table has 7 columns: a column of zeros, a column of "4444", a column of "336", a column of 10-digit IDs, a column of floating-point values, and two columns of zeros. The data is as follows:

0	4444	336	199505270000	66.0000	0	0
0	4444	336	199505270030	70.0000	0	0
0	4444	336	199505270100	77.0000	0	0
0	4444	336	199505270130	78.0000	0	0
0	4444	336	199505270200	81.0000	0	0
0	4444	336	199505270230	79.0000	0	0
0	4444	336	199505270300	77.0000	0	0
0	4444	336	199505270330	78.0000	0	0
0	4444	336	199505270400	79.0000	0	0
0	4444	336	199505270430	79.0000	0	0
0	4444	336	199505270500	60.0000	0	0
0	4444	336	199505270530	56.0000	0	0
0	4444	336	199505270600	56.0000	0	0
0	4444	336	199505270630	49.0000	0	0
0	4444	336	199505270700	43.0000	0	0
0	4444	336	199505270730	43.0000	0	0
0	4444	336	199505270800	52.0000	0	0
0	4444	336	199505270830	42.0000	0	0
0	4444	336	199505270900	50.0000	0	0
0	4444	336	199505270930	59.0000	0	0
0	4444	336	199505271000	69.0000	0	0
0	4444	336	199505271030	87.0000	0	0
0	4444	336	199505271100	92.0000	0	0
0	4444	336	199505271130	94.0000	0	0
0	4444	336	199505271200	104.0000	0	0
0	4444	336	199505271230	106.0000	0	0
0	4444	336	199505271300	95.0000	0	0
0	4444	336	199505271330	97.0000	0	0
0	4444	336	199505271400	91.0000	0	0
0	4444	336	199505271430	84.0000	0	0
0	4444	336	199505271500	86.0000	0	0
0	4444	336	199505271530	90.0000	0	0
0	4444	336	199505271600	95.0000	0	0
0	4444	336	199505271630	94.0000	0	0
0	4444	336	199505271700	88.0000	0	0
0	4444	336	199505271730	87.0000	0	0

The bottom of the window features a toolbar with buttons: "None", "Back", "Forward", "Home", "Reload", "Open...", "Save As...", "Clone", "New", and "Close".

Abbildung A7 - Visualisierung von Ergebnisdaten im Slave-Mosaic

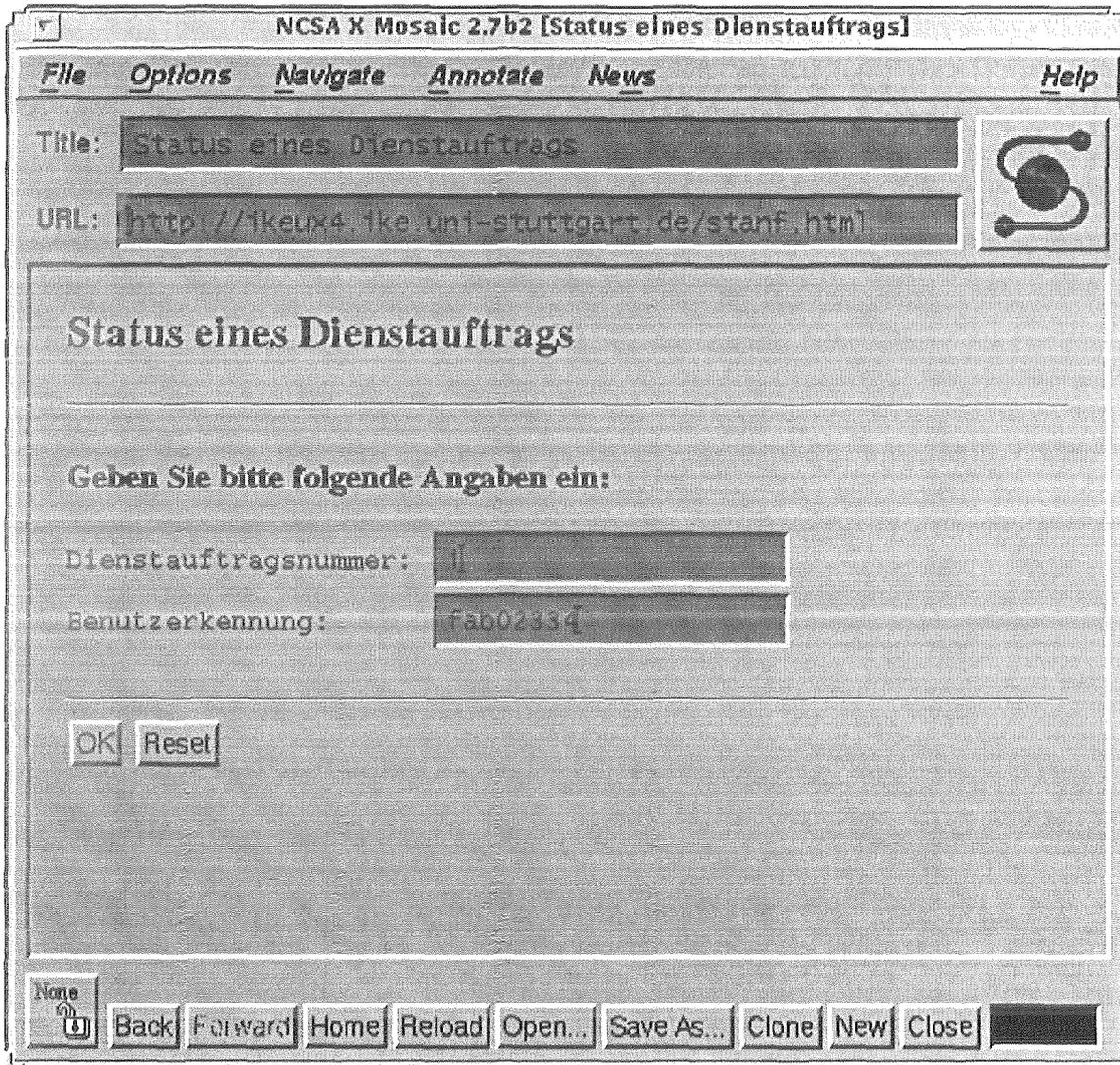


Abbildung A8 - HTML-Seite "Aufruf der Statusanfrage"

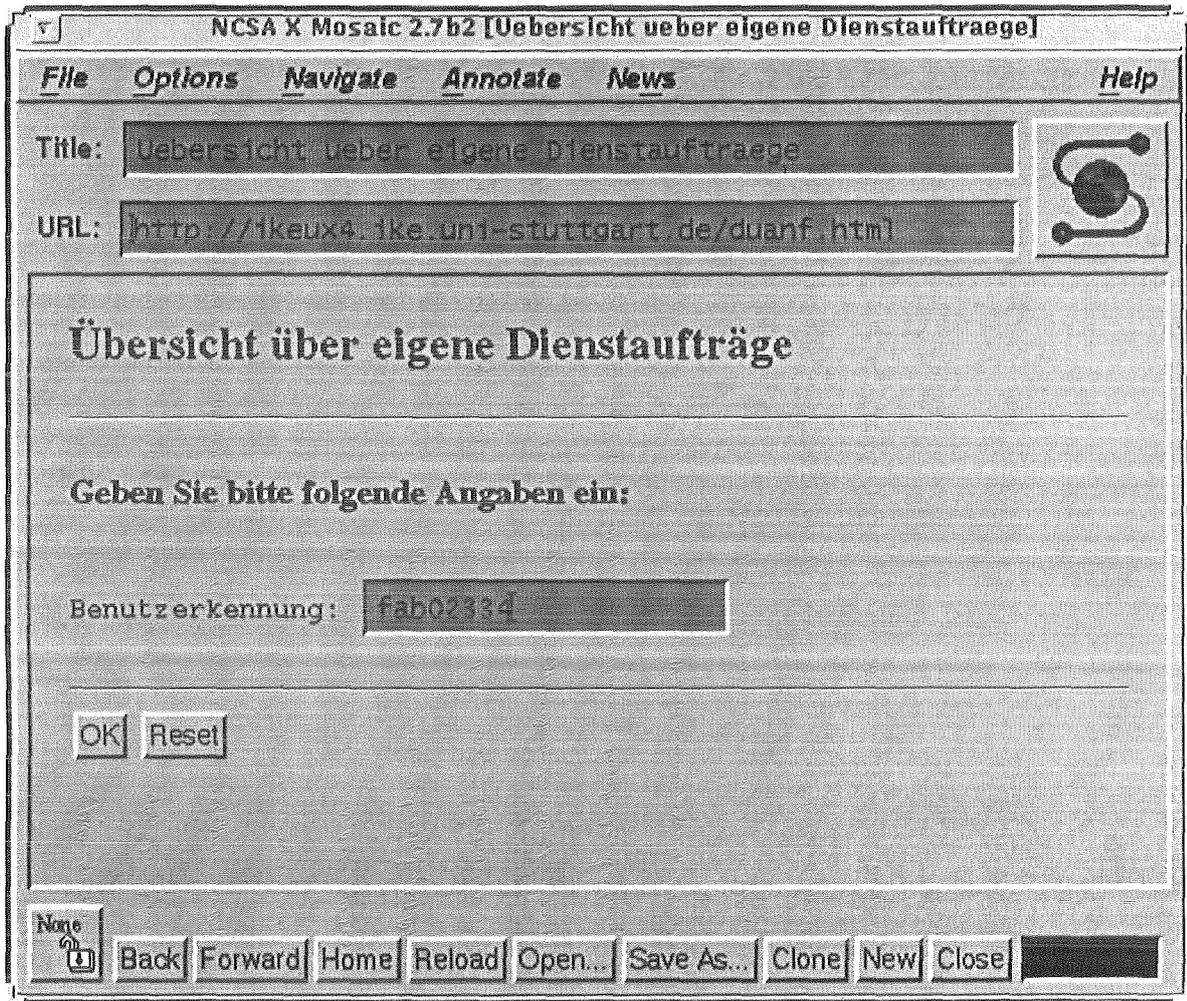


Abbildung A9 - HTML-Seite "Übersicht über eigene Dienstaufträge"

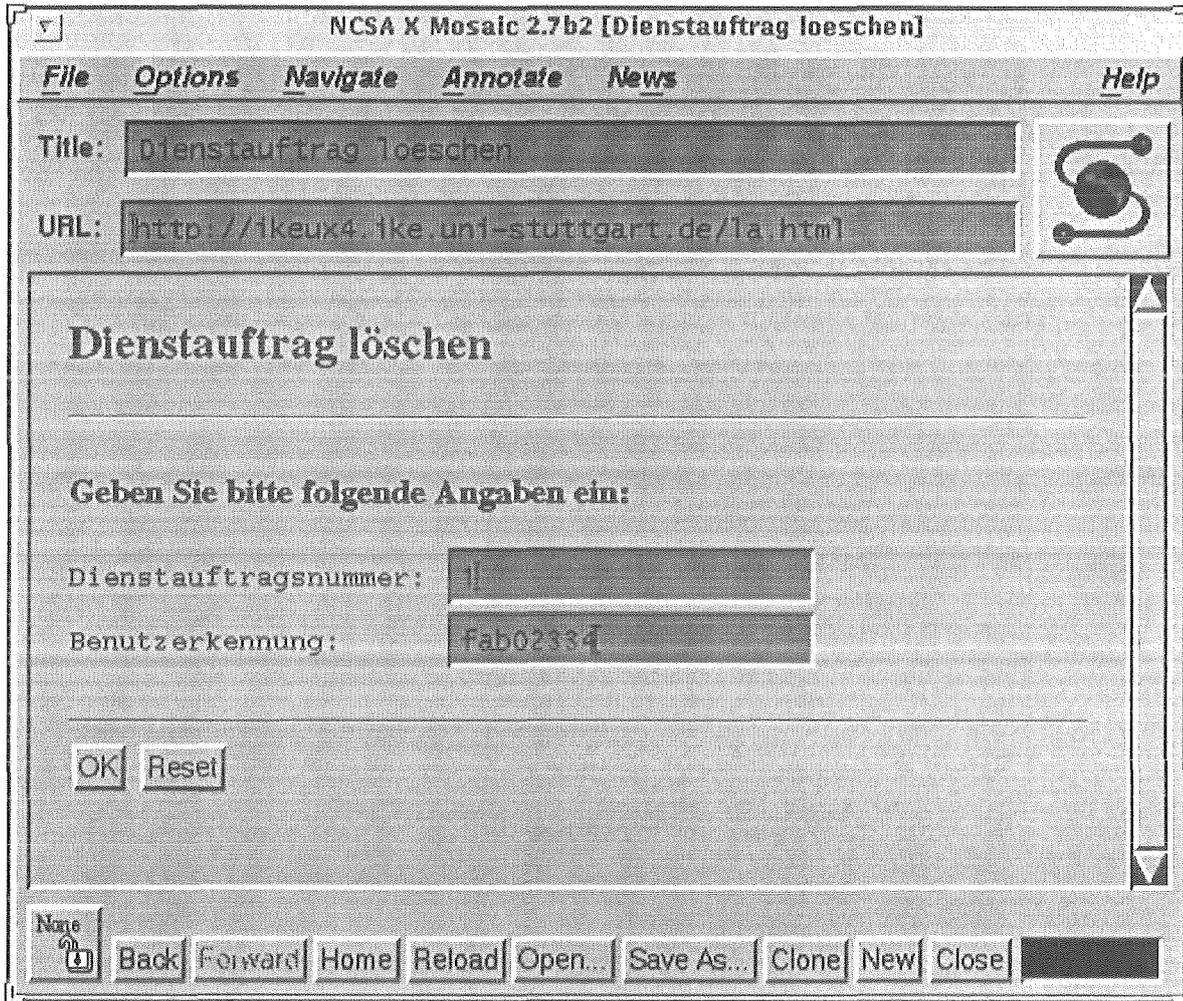


Abbildung A10 - HTML-Seite "Dienstauftrag löschen"

Visualisierung von Umweltdaten

- Anlage -

*Joachim Wiesel, Wilhelm Hagg
Institut für Photogrammetrie und Fernerkundung
Universität Karlsruhe
Englerstraße 7
76128 Karlsruhe*

*unter Mitarbeit von
Heiko Jacobs
Ingenieurbüro Jacobs*

*Michael Böse, Claus Hofmann, Heiko Krassin
Institut für Photogrammetrie und Fernerkundung*

Schaubilder

Auf den folgenden Seiten sind einige Screendumps einer typischen Sitzung in folgender Reihenfolge wiedergegeben:

m1.gif Einstiegsseite, Auswahl zwischen Karten- und Diagramm- und kombinierter Darstellung

m2.gif Auswahl der Region

m3.gif Auswahl des Regierungsbezirks Karlsruhe

m4.gif Auswahl der Hintergrundkarte "Einfärbung: Kreis"

m5.gif Auswahl der Vektorkarten "Gemeinde und Forstl. Regionalges."

m6.gif Auswahl der Darstellungsgroesse und Farben

m7.gif Belegung der Diagrammachsen

m8.gif Auswahl der darzustellenden Diagramme

m9.gif Auswahl von Diagrammart und Groesse

ma.gif Auswahl der Diagrammposition: Am Rand der Karte

mb.gif Darstellungsseite (obere Hälfte)

mc.gif Darstellungsseite (untere Hälfte)

md.gif Darstellung mit Diagrammen in der Karte

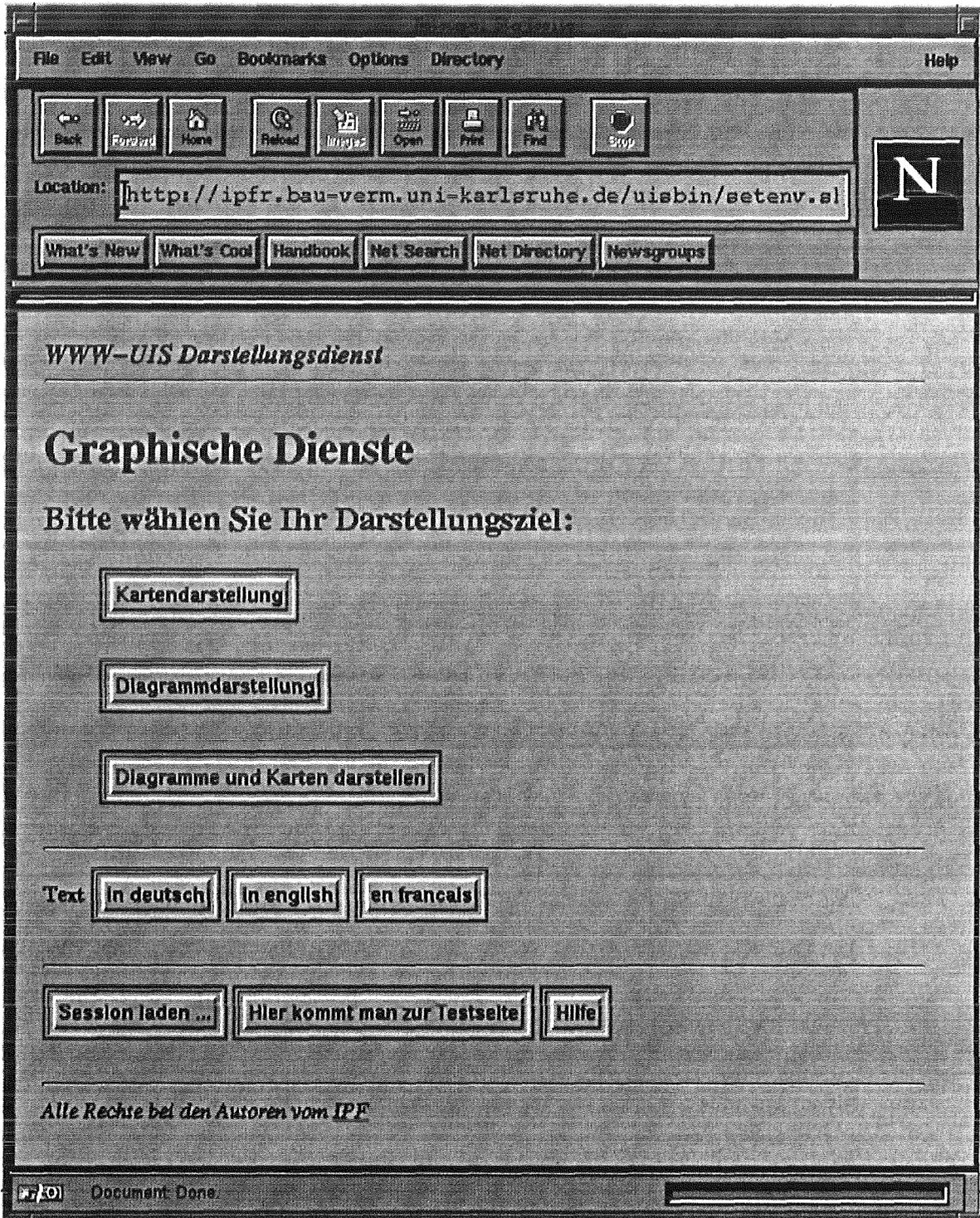
me.gif Darstellung gezoomt (Faktor 4)

mf.gif Vollbilddarstellung eines Diagramms

mg.gif Objektinformationen

mh.gif Online-Hilfeseite (für Darstellungsseite)

mi.gif Darstellung mit Kreisdiagrammen



File Edit View Go Bookmarks Options Directory Help

Back Forward Home Reload Images Open Print Find Stop

Location:

What's New What's Cool Handbook Net Search Net Directory Newsgroups

WWW-UIS Darstellungsdienst

Auswahl der darzustellenden Region

Eingangsgröße: Bounding Box aus Diagrammen: 3407640 5285257 3589041 5493051

aktuelle Box: 3388362 5266822 3610194 5517393

Auswahl der Region fertig. akt. Box ok, weiter zur Kartenauswahl

oder Wahl der Region ändern:

Eingangsgröße wählen

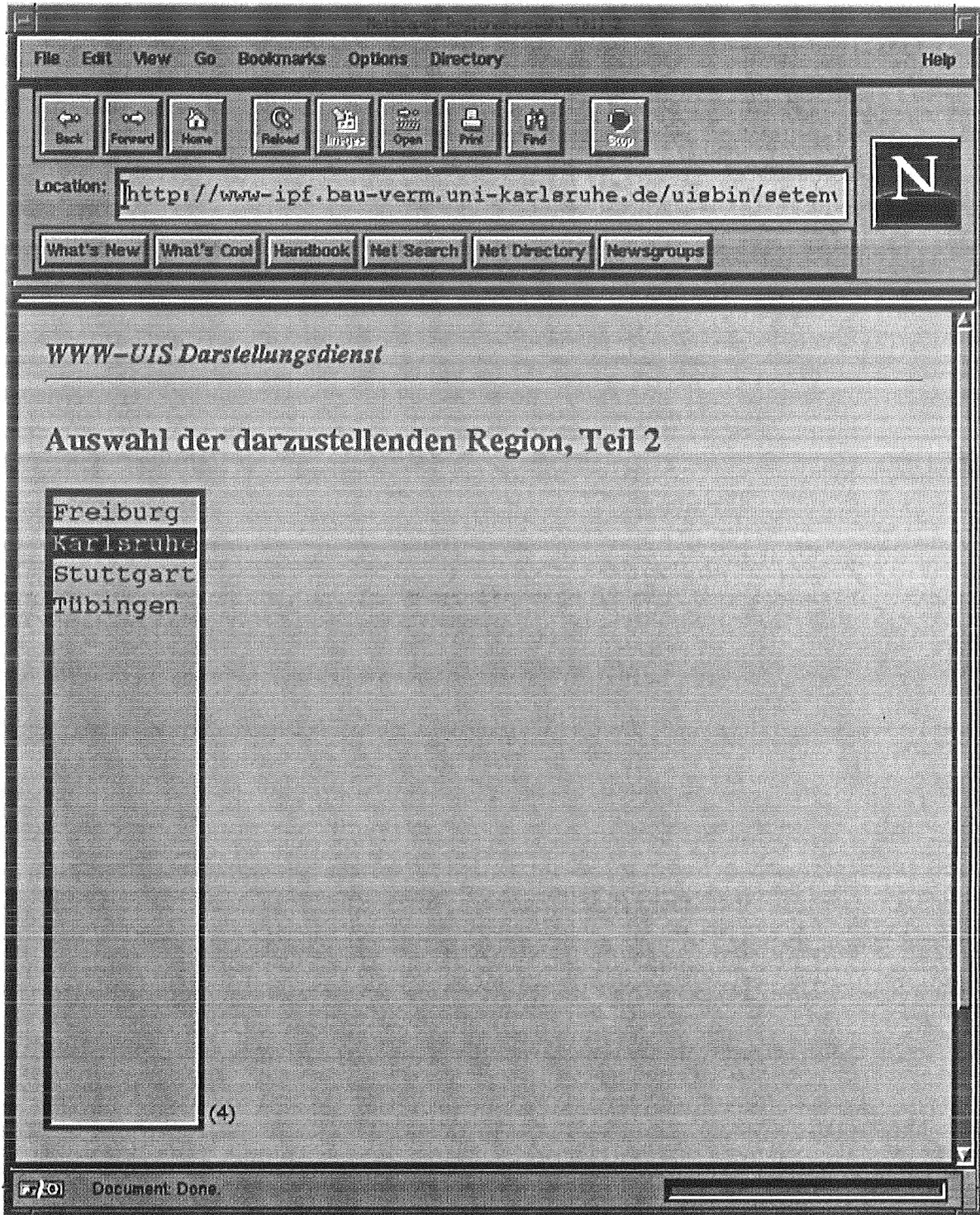
Box wählen: (In Gauß-Krüger-Koordinaten)

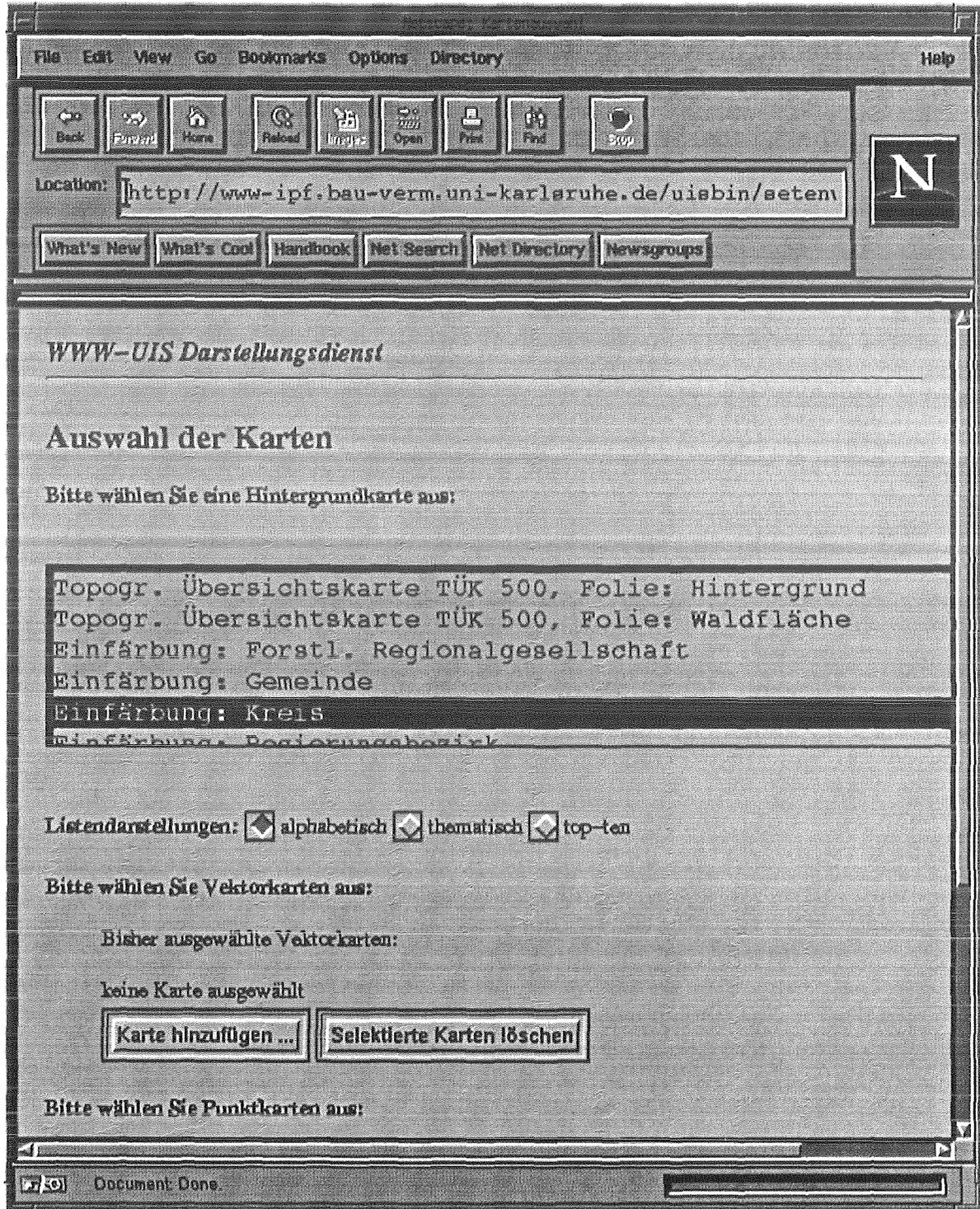
<input type="text" value="3388362"/>	<input type="text" value="5266822"/>	unten links	(Rechtswert/Hochwert) (x/y)
<input type="text" value="3610194"/>	<input type="text" value="5517393"/>	oben rechts	(Rechtswert/Hochwert) (x/y)

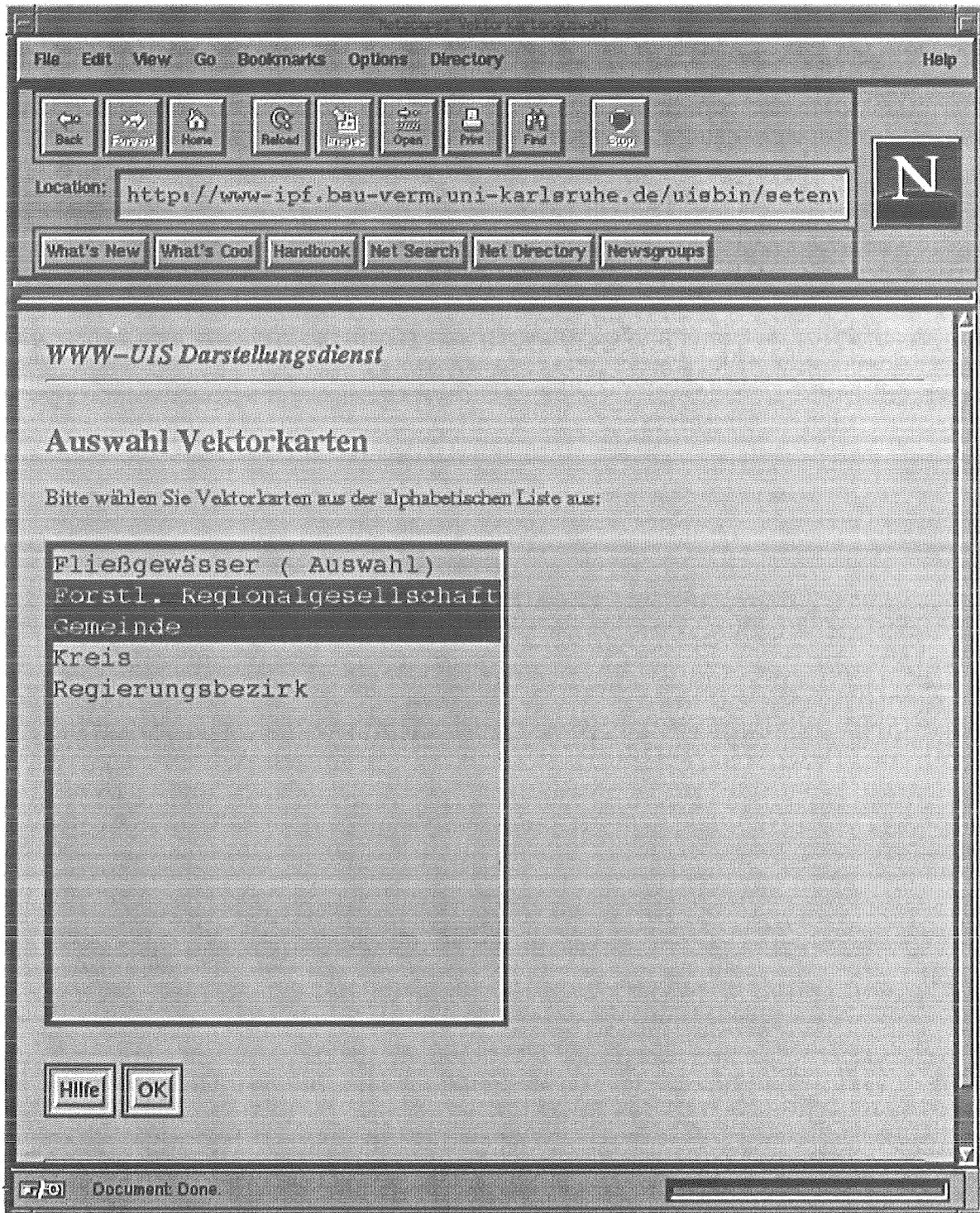
Bundesland:

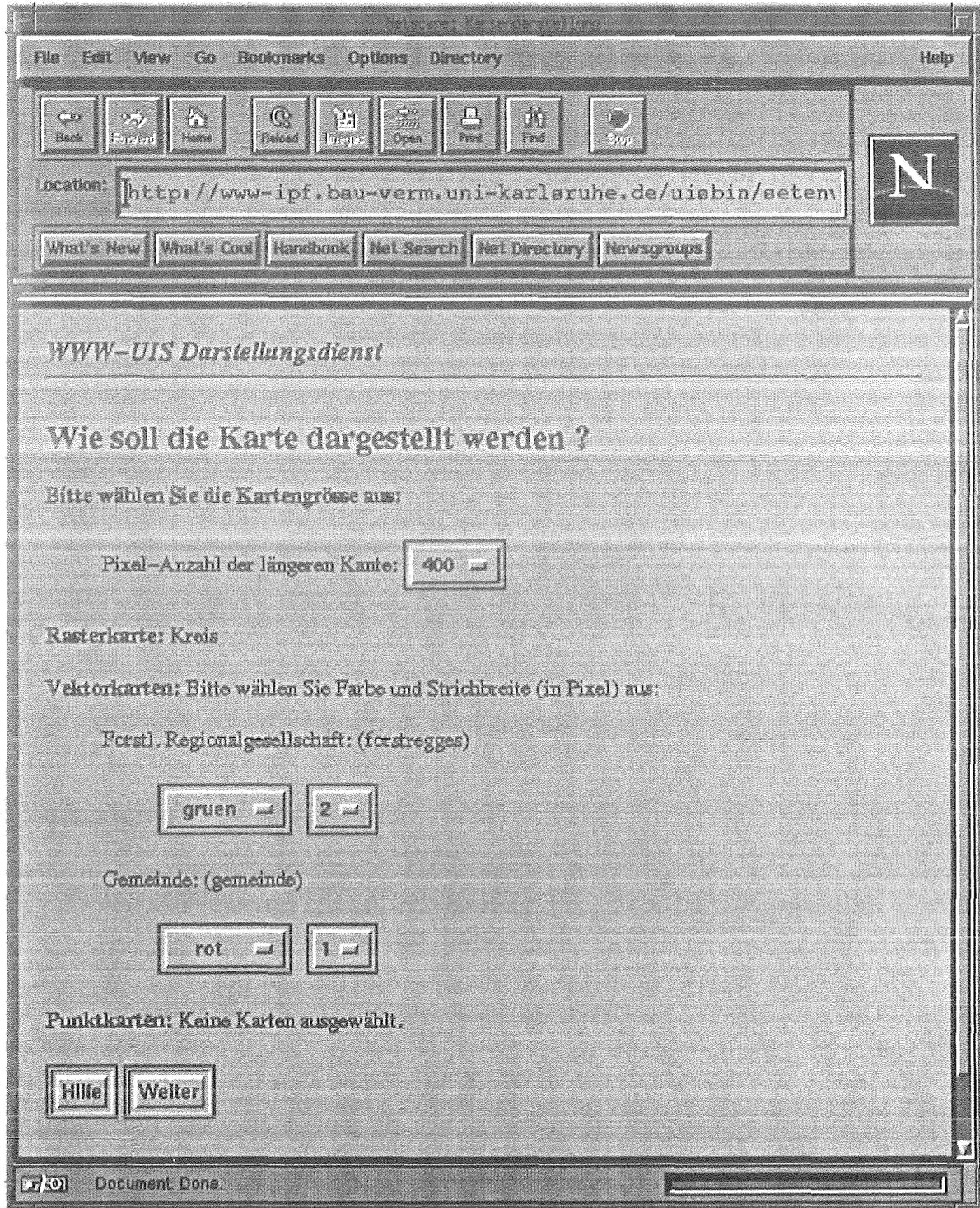
Regierungsbezirk aus Bundesland:

Document: Done



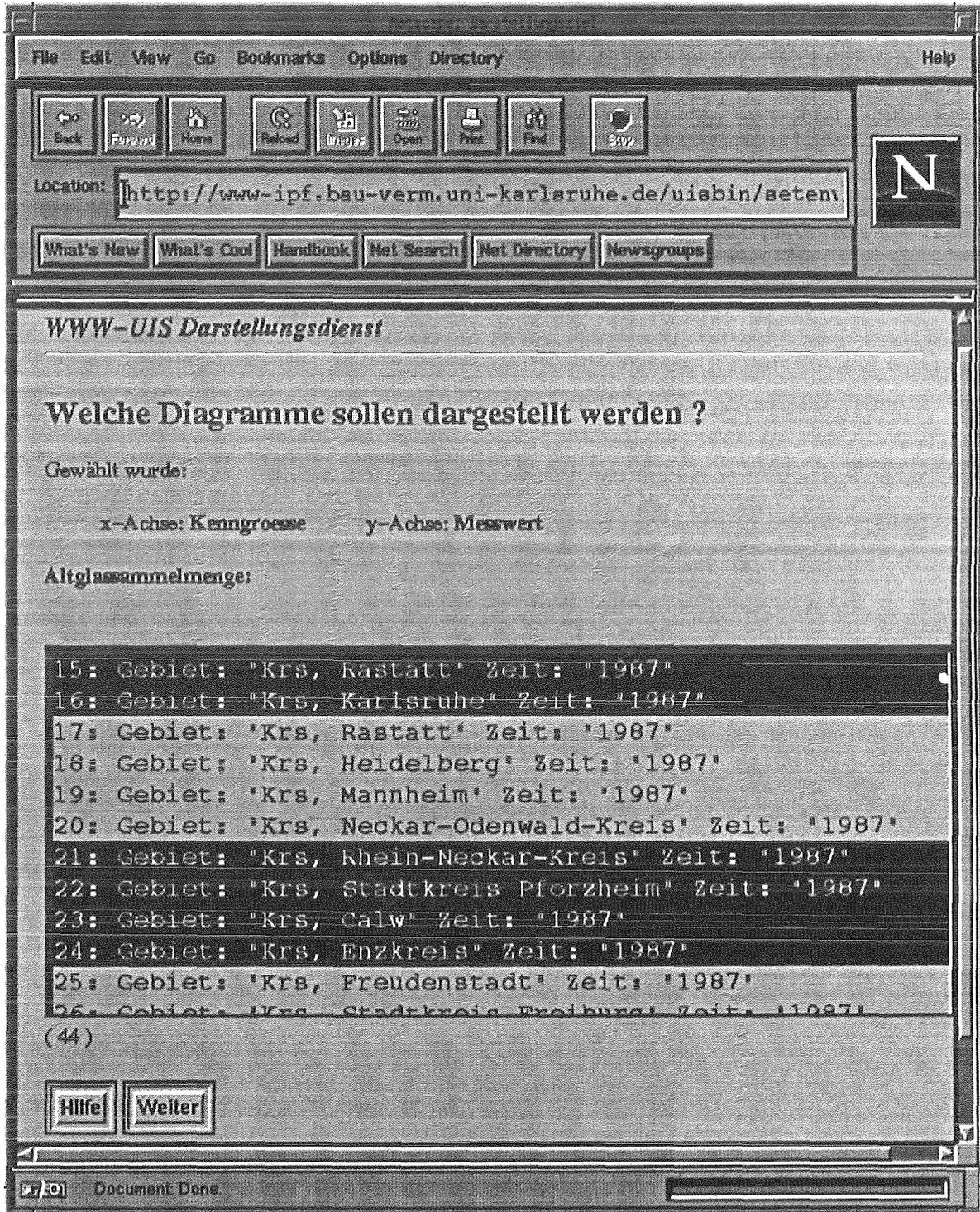


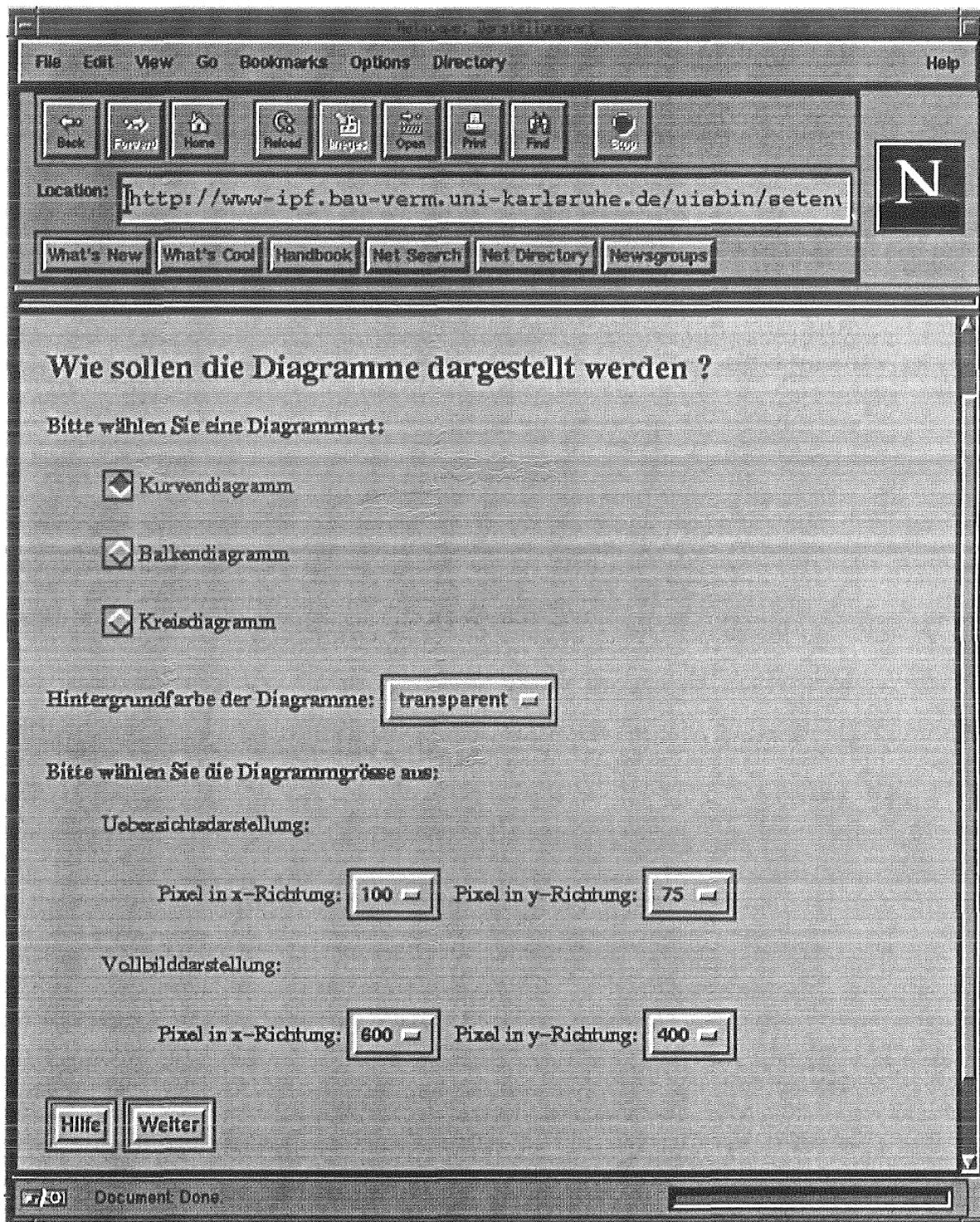


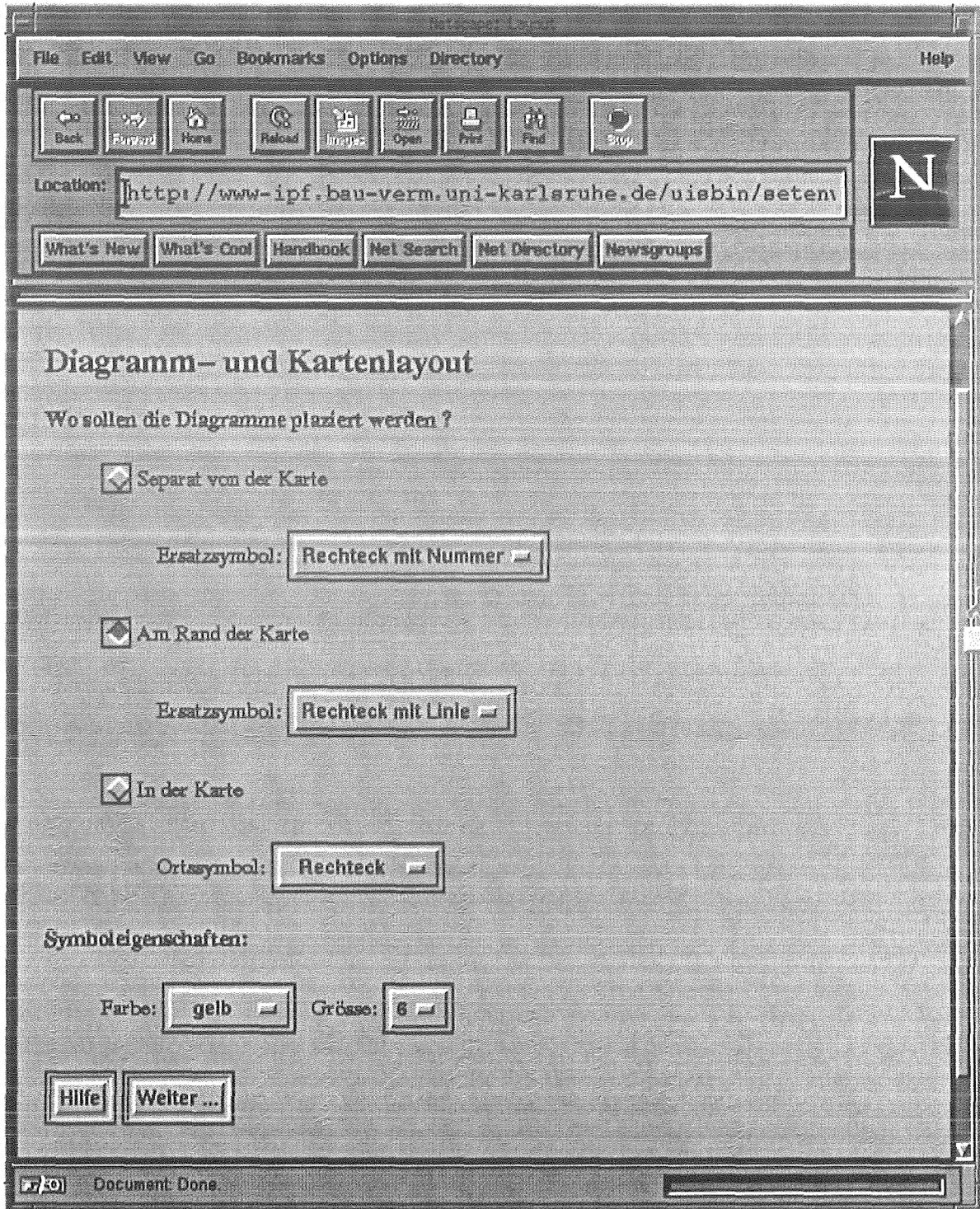


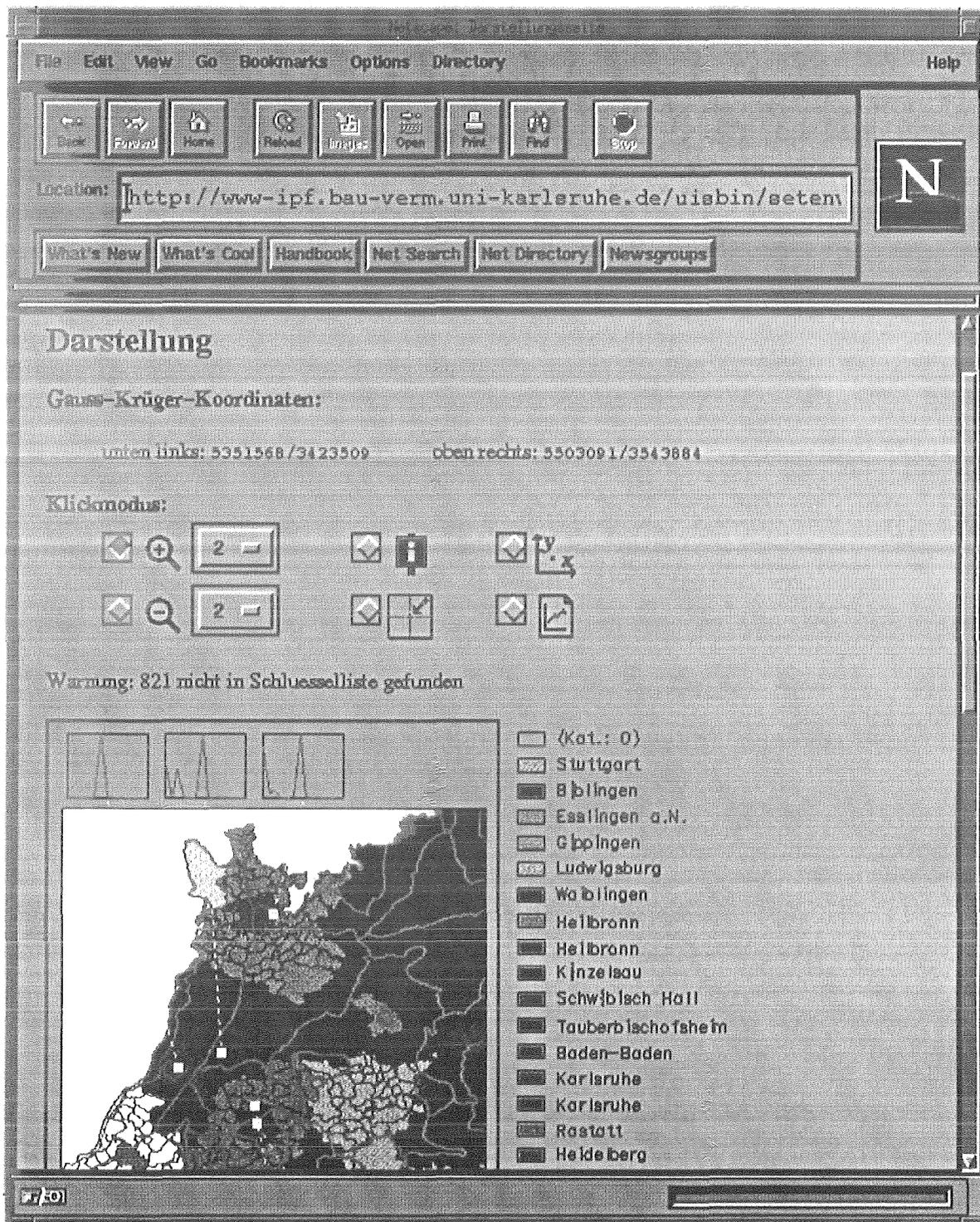
The screenshot shows a Netscape browser window with the following elements:

- Menu Bar:** File, Edit, View, Go, Bookmarks, Options, Directory, Help.
- Toolbar:** Back, Forward, Home, Reload, Stop, Open, Print, Find.
- Location Bar:** http://www-ipf.bau-verm.uni-karlsruhe.de/uisbin/setem
- Navigation Buttons:** What's New, What's Cool, Handbook, Net Search, Net Directory, Newsgroups.
- Page Content:**
 - Title: *WWW-UIS Darstellungsdienst*
 - Section Header: **Wie sollen die Diagramme aufgebaut sein?**
 - Text: **x-Achse:**
 - Diagram: A box with 'Gebiet' and 'Kenngrösse' labels.
 - Text: **y-Achse:**
 - Text: **Messwert**
 - Buttons: **Hilfe**, **Weiter**
 - Footer: *Alle Rechte bei den Autoren vom IPF*
- Status Bar:** Document: Done









The screenshot shows a Netscape browser window with the following elements:

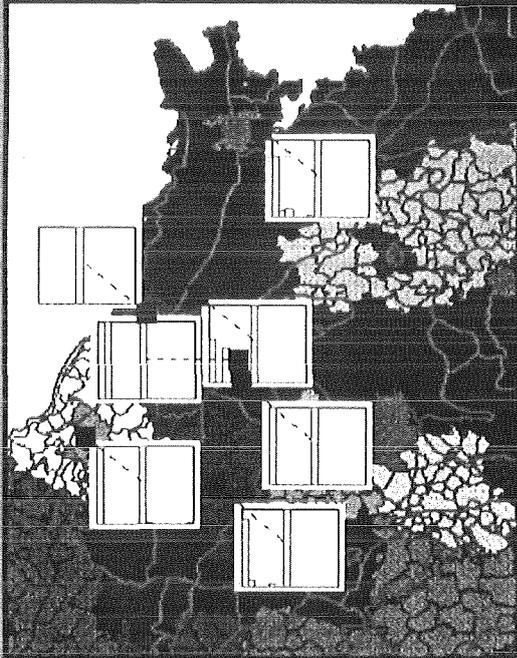
- Menu Bar:** File, Edit, View, Go, Bookmarks, Options, Directory, Help.
- Toolbar:** Back, Forward, Home, Reload, Images, Open, Print, Find, Stop.
- Location Bar:** http://www-ipf.bau-verm.uni-karlsruhe.de/uisbin/etem
- Navigation Buttons:** What's New, What's Cool, Handbook, Net Search, Net Directory, Newsgroups.
- Main Content Area:**
 - Map:** A grayscale map of a region with a legend on the right.
 - Legend:**
 - Heidelberg
 - Mannheim
 - Mosbach
 - RHEIN-NECKAR-KREIS
 - Pforzheim
 - Calw
 - Enzkreis
 - Freudenstadt
 - Offenburg
 - Rottweil
 - Reutlingen
 - Tübingen
 - Balingen
 - AB-Donau-Kreis
 - Forstl. Regionalgeseilschaft
 - Gemeinde
 - Navigation Controls:**
 - Verschiebung:** A 3x3 grid of directional arrows.
 - Verschiebungweite:** A set of four square icons representing different zoom levels.
 - Buttons:**
 - Row 1: Darstellungsregion, Kartenauswahl, Kartenlayout, Kartenexport
 - Row 2: Diagrammart, Diagrammauswahl, Diagrammlayout, Diagrammexport

File Edit View Go Bookmarks Options Directory Help

Back Forward Home Reload Images Open Print Find Stop

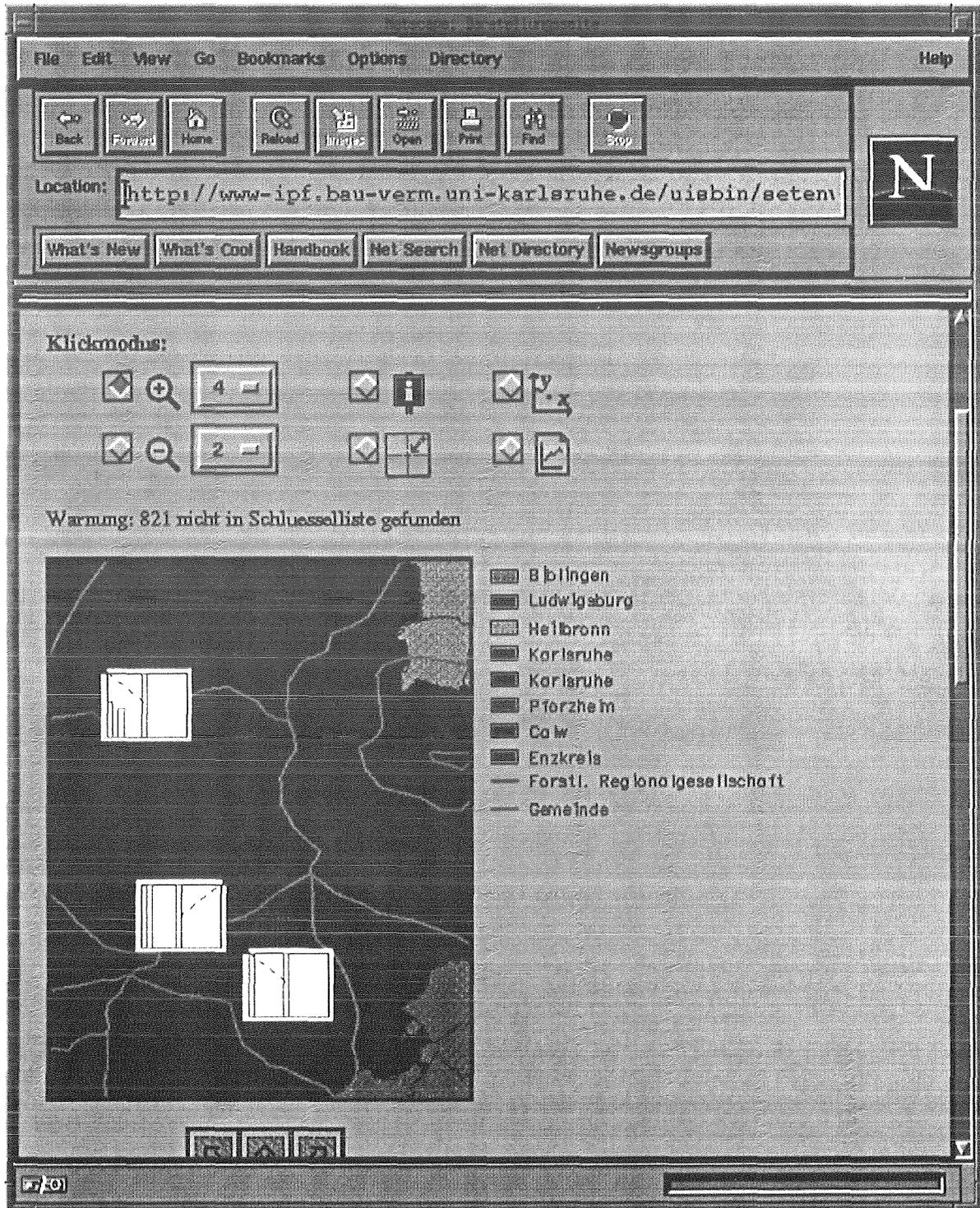
Location: <http://www-ipf.bau-verm.uni-karlsruhe.de/uiabin/setem>

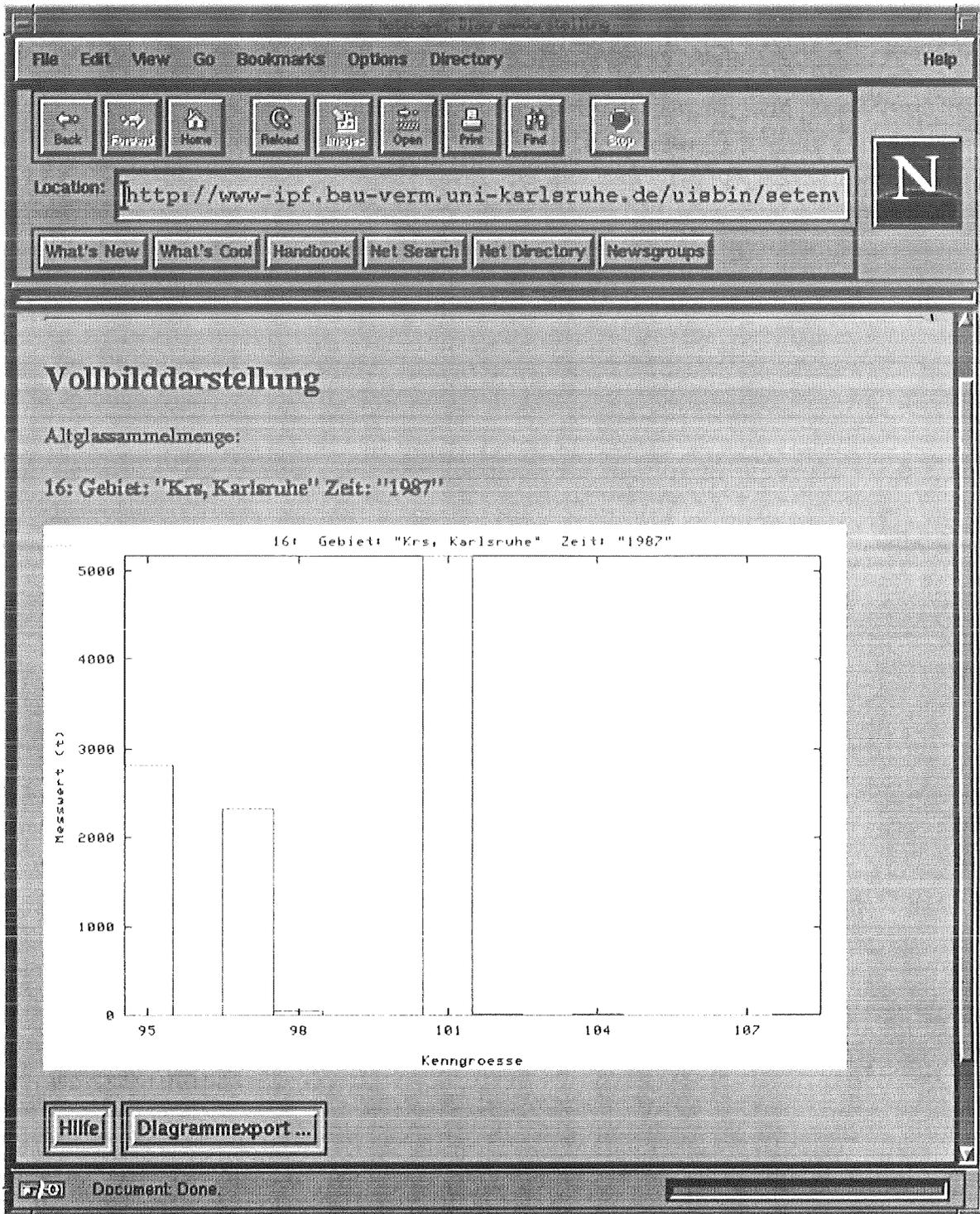
What's New What's Cool Handbook Net Search Net Directory Newsgroups



- (Kat.: 0)
- Stuttgart
- Balingen
- Esslingen a.N.
- Göppingen
- Ludwigsburg
- Waiblingen
- Heilbronn
- Heilbronn
- Künzelsau
- Schwäbisch Hall
- Tauberbischofsheim
- Baden-Baden
- Karlsruhe
- Karlsruhe
- Rastatt
- Heidelberg
- Mannheim
- Mosbach
- RHEIN-NECKAR-KREIS
- Pforzheim
- Calw
- Enzkreis
- Freudenstadt
- Offenburg
- Rotweil
- Reutlingen
- Tübingen
- Balingen
- Ab-Donau-Kreis
- Forstl. Regionalesellschaft
- Gemeinde

01





The screenshot shows a Netscape browser window with the following elements:

- Menu Bar:** File, Edit, View, Go, Bookmarks, Options, Directory, Help
- Toolbar:** Back, Forward, Home, Reload, Images, Open, Print, Find, Stop
- Location Bar:** http://www-ipf.bau-verm.uni-karlsruhe.de/uibin/setem
- Navigation Buttons:** What's New, What's Cool, Handbook, Net Search, Net Directory, Newsgroups
- Page Content:**
 - Darstellung**
 - Kartenobjekte unter der Koordinate R=3485794 H=5425233**
 - Flächenkarte Forstl. Regionalgesellschaft:**
 - Objekt Nr. 402: Kolline Laubwälder, meist eichenreich
 - nummer: 402
 - rechtswert: 3514564
 - hochwert: 5441226
 - name:
 - bbox_xl: 0
 - bbox_yl: 0
 - bbox_xh: 0
 - bbox_yh: 0
 - typ: Kolline Laubwälder, meist eichenreich
 - Flächenkarte Gemeinde:**
 - Objekt Nr. 8236050: Ötisheim
 - nummer: 8236050
 - rechtswert: 3485776
 - hochwert: 5424974
 - name: Ötisheim
 - kreisname: Enzkreis
 - regbezname: Karlsruhe
 - wwa: 0
 - wwaa: 0
 - kreis_nummer: 236
 - regbezirk_nummer: 2
 - bbox_xl: 3483080
 - bbox_xh: 3488391

At the bottom of the browser window, the status bar shows "Document Done" and a progress indicator.

The screenshot shows a Netscape browser window with the following elements:

- Menu Bar:** File, Edit, View, Go, Bookmarks, Options, Directory, Help.
- Toolbar:** Back, Forward, Home, Reload, Images, Open, Print, Find, Stop.
- Location Bar:** http://www-ipf.bau-verm.uni-karlsruhe.de/uibin/seten
- Navigation Buttons:** What's New, What's Cool, Handbook, Net Search, Net Directory, Newsgroups.
- Page Content:**
 - Darstellungsseite** (Section Header)
 - In Abhängigkeit davon, ob Sie**
 - nur Karten darstellen wollen,
 - nur Diagramme darstellen wollen oder
 - Diagramme zusammen mit Karten darstellen wollen,hat die Seite unterschiedliche Inhalte. Einzelne Punkte hängen zudem von den Auswahlen der vorhergehenden Seiten ab.
 - Aufgabe dieser Seite:**

Auf dieser Seite werden die gewählten Karten und Diagramme dargestellt. Einzelne Diagramme können per Auswahl großformatig dargestellt werden. Kartenausschnitte können verändert werden. Informationen zu einzelnen Kartenobjekten sind noch nicht implementiert.
 - Koordinatenbereich der dargestellten Karte:**

Wird in Gauß-Krüger-Koordinaten angegeben.
 - Klickmodus:**

Hier wird festgelegt, welche Aktion durch einen Klick in die Karte ausgelöst wird:

 - : Die Region, die mit der Maus mittig angeklickt wird, wird n-mal vergrößert dargestellt. Der Vergrößerungsfaktor wird im Menü rechts davon ausgewählt. Es wird somit eine kleinere Kartenregion dargestellt.
 - : Die Region, die mit der Maus mittig angeklickt wird, wird n-mal verkleinert dargestellt. Der Verkleinerungsfaktor wird im Menü rechts davon ausgewählt. Es wird somit eine
- Status Bar:** Document Done.

File Edit View Go Bookmarks Options Directory Help

Back Forward Home Reload Images Open Print Find Stop

Location: <http://www-ipf.bau-verm.uni-karlsruhe.de/uibin/aeten>

What's New What's Cool Handbook Net Search Net Directory Newsgroups

(Kat.: D)
Stuttgart
Balingen
Esslingen a.N.
Ludwigsburg
Waiblingen
Heilbronn
Heilbronn
Karlsruhe
Karlsruhe
Rastatt
Heidelberg
Mosbach
RHEIN-NECKAR-KREIS
Pforzheim
Calw
Enzkreis
Freudenstadt
Offenburg
Reutlingen
Tübingen
Forstl. Regionalgeseilschaft
Gemeinde

7:01

ArcView 2.0 - Leistungsumfang 18.7.95

- ARC/INFO-Daten anzeigen
- Bilder, Videos anzeigen
- Abfragefunktionen

- Projekte, Views, Themen
- ARC-Coverages (Kartenebenen), ArcView Shapes
- ARC-GRID, TIFF, ERDAS, Sun-Raster, RLC
- Oracle, Ingres, Sybase, Informix via SQL/ODBC
- Dbase-III, IV, INFO-Tabellen, Kommagetrennte Texttabellen
- Hot-Links
- Tabellenbearbeitung - Formeln, Sortieren, Aufbereiten

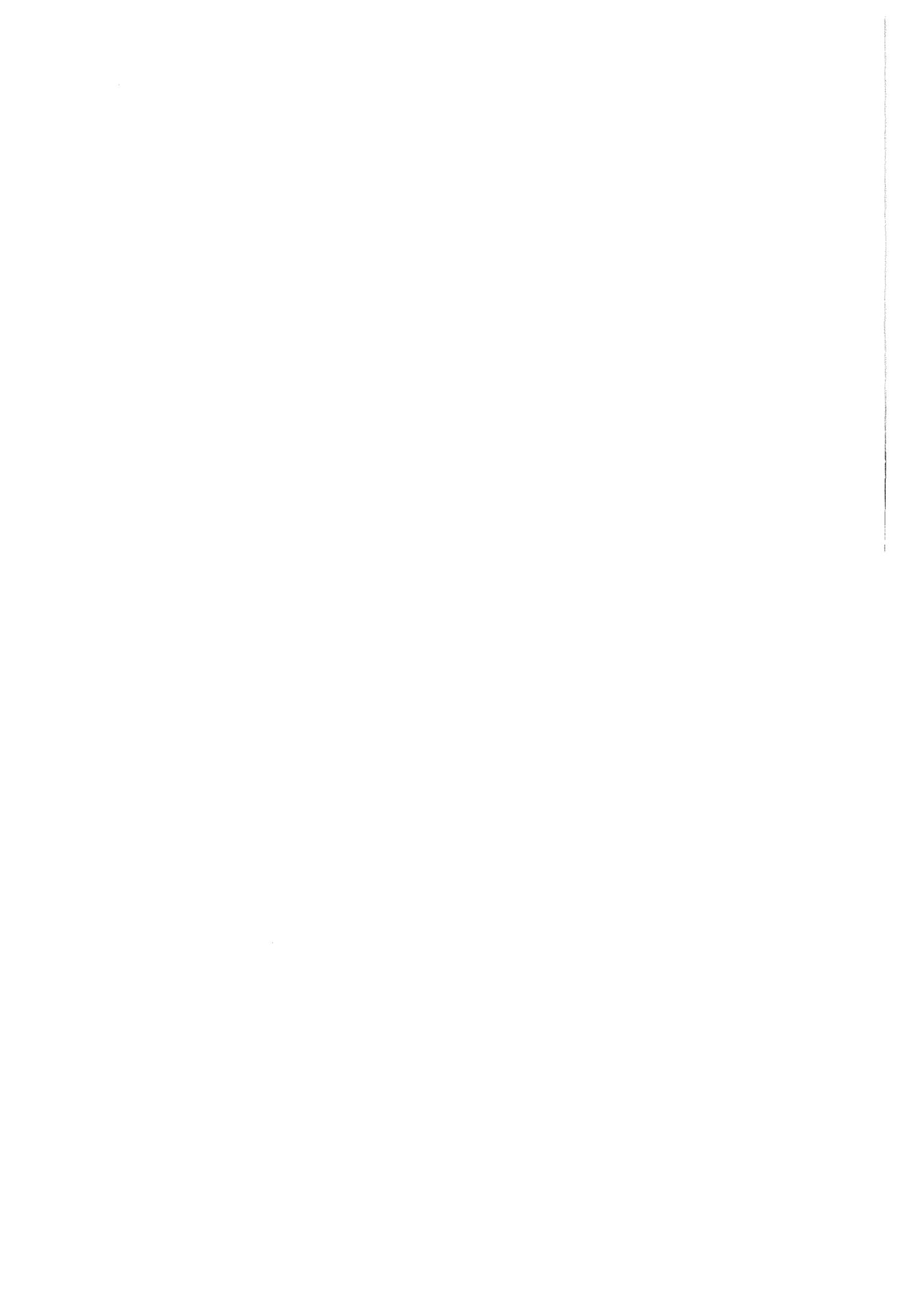
- Objektauswahlfunktionen: Mausclick, Rechteck, Shapes, Attributsuche, Abfragemanager für komplexe Attributsuche

- Kartographische Aufbereitung inkl. Geschäftsgraphik: Flächendiagramme, Balken-, Linien-, Kreis-, XY-Streudiagramme
- Kartenlayout: Rahmen, Legenden, Graphiken, Bilder

- Scripts: Avenue - objektorientierte Script-Sprache, C-ähnliche Syntax
- ArcView Klassen: Abstrakte und Konkrete Klassen
- Persistente Objekte
- Interapplikationskommunikation: DDE (Client & Server), RPC (Client & Server nur Unix)
- Kommunikation mit dem Gastgeberbetriebssystem via System-Klasse

- Betriebssysteme: MS-Windows 3, Windows NT, Windows 95, MacOS, Unix, OS/2 3.0 (WinApp), mindestens i386
- Mindestbedarf: 8MB Arbeitsspeicher (im praktischen Betrieb jedoch nicht nutzbar - eher 16MB)
- Deutsche Bedieneroberfläche für 2.0c (MS-Windows)
- Avenue im Lieferumfang enthalten
- "Canned Applications" möglich mit nicht veränderbaren Scripts

- Ressourcenbedarf recht hoch
- Nur wenige Austauschformate (Shapes neu und nicht dokumentiert)
- RPC nur mit ARC/INFO als Server (mindestens Version 7.0 erforderlich)
- Interapplication-Kommunikation braucht zwingend 16MB oder mehr
- keine OLE-Automationschnittstelle
- **Erfordert Avenue-Programmierung**



Richtlinie zur Erstellung WWW-verfügbarer Berichte

*Rainer Weidemann; Werner Geiger; Matthias Reißfelder
Forschungszentrum Karlsruhe GmbH - Technik und Umwelt
Postfach 3640
76021 Karlsruhe*

1. EINLEITUNG	363
1.1 WORLD-WIDE WEB	363
1.2 KONSEQUENZEN FÜR DIE ERSTELLUNG VON BERICHTEN	363
1.2.1 <i>Alternative 1: Nachträgliche Konvertierung nach HTML</i>	364
1.2.2 <i>Alternative 2: Gleichzeitige Erstellung beider Versionen</i>	365
1.3 ZU DIESEM BERICHT	365
2. ERSTELLUNG VON BERICHTEN.....	366
2.1 DIE DOKUMENTVORLAGE.....	366
2.2 GLIEDERUNGSELEMENTE.....	368
2.2.1 <i>Titel</i>	368
2.2.2 <i>Überschriften</i>	368
2.2.3 <i>Index</i>	369
2.2.4 <i>Weitere Gliederungselemente</i>	369
2.2.4.1 <i>Kopf- und Fußzeilen</i>	369
2.2.4.2 <i>Verzeichnisse</i>	369
2.2.4.3 <i>Seitenwechsel</i>	370
2.3 TEXTE.....	370
2.3.1 <i>„Normaler“ Fließtext</i>	370
2.3.2 <i>Vorformatierter Text</i>	371
2.3.3 <i>Weitere Hervorhebungen</i>	372
2.3.4 <i>Zeichenvorrat</i>	372
2.4 LISTEN, AUFZÄHLUNGEN, TABELLEN	373
2.4.1 <i>Listen und Aufzählungen</i>	373
2.4.2 <i>Weitere Formatvorlagen für Listen/Aufzählungen</i>	374
2.4.3 <i>Tabellen</i>	375
2.5 ABBILDUNGEN.....	375
2.5.1 <i>Voraussetzungen für eine automatische Konvertierung</i>	376
2.5.2 <i>Verknüpfte Abbildungen in anderen Formaten</i>	377
2.5.3 <i>Eingefügte Abbildungen</i>	378
2.5.4 <i>Platzhalter für Abbildungen</i>	379
2.5.5 <i>Verwendung verschiedener Alternativen in einem Dokument</i>	380
2.5.6 <i>Empfehlungen für die Erstellung von Abbildungen</i>	380
2.6 VERWEISE.....	381
2.6.1 <i>Fußnoten</i>	381
2.6.2 <i>Verweise auf Kapitelüberschriften</i>	381
2.6.3 <i>Sonstige Verweise</i>	381
2.7 SPEZIELLE HTML-FORMATIERUNGEN	382
2.8 DAS DATEI-INFO	383
3. DATENBANKEN UND ANWENDUNGSPROGRAMME.....	384
ANHANG.....	385
A. ZUR ERSTELLUNG VON WWW-SEITEN	385
A.1 <i>Texteditoren</i>	385
A.2 <i>HTML-Editoren</i>	385
A.3 <i>Konverter</i>	386

1. Einleitung

Die Landesanstalt für Umweltschutz (LfU) und das Forschungszentrum Karlsruhe (FZK) entwickeln im Rahmen des GLOBUS-Projekts ein Altlasteninformationssystem, das den Altlastensachbearbeitern Handbücher und Fachberichte der LfU über das World-Wide Web an ihrem Arbeitsplatz verfügbar machen soll. Längerfristig ist auch eine Einbindung von einfachen Datenbanken und Anwendungsprogrammen geplant.¹

1.1 World-Wide Web

Das WWW ist ein Client-Server-System, bei dem jeder mit dem Internet verbundene Rechner Informationen zur weltweiten Einsicht bereitstellen (= Server) und/oder bereitgestellte Informationen abrufen (= Client / Browser) kann. Basisobjekt des WWW ist die Informationsseite, die zunächst nur aus Texten und aktiven Verknüpfungen zu anderen Informationsseiten bestand (Hypertext). Das Anklicken einer aktiven Verknüpfung im Browser führt dazu, daß die entsprechende Informationsseite geholt und angezeigt wird, egal auf welchem Server weltweit diese liegt. Das WWW erlaubt weiterhin die Übertragung beliebiger Dokumente (z.B. ausdruckbare Berichte, Software, ..) und entwickelt sich durch Integration von Bildern, Videos und Tondokumenten zu einem Hypermedia-System.

Hypertext-Dokumente haben typischerweise einen anderen Aufbau als "konventionelle" Dokumente. Letztere haben eine "flache" Struktur. Querverweise über einen Index oder aus dem Text heraus erfordern das Blättern und Suchen im Dokument. Zwar kann man das Dokument auch in dieser Form in das WWW einhängen, doch ist dies i.a. nicht zweckmäßig, da man so die Vorteile des Hypertext-Ansatzes teilweise wieder verschenkt. Um ein konventionelles Dokument ins WWW einbringen zu können, ist dies in einzelne Teile von etwa Bildschirmgröße zu zerlegen, die untereinander verzeigert werden. Das konventionelle Dokument wird damit in ein Netz von Hypertext-Seiten aufgelöst. Da die Verzeigerung nach inhaltlichen Kriterien erfolgen muß, ist eine vollständige Automatisierung in aller Regel nicht möglich!

1.2 Konsequenzen für die Erstellung von Berichten

Im Anhang dieses Berichts (s. Anhang A. Zur Erstellung von WWW-Seiten) werden prinzipielle Alternativen zur Erstellung von WWW-Seiten beschrieben. In diesem Kapitel soll nun untersucht werden, welche Vorgehensweisen bei der Erstellung neuer Handbücher und Fachsysteme der LfU sinnvoll sind. Es wird davon ausgegangen, daß durch rechtzeitige Berücksichtigung einer späteren Einbindung in das WWW ein großer Teil des Nachbearbeitungsaufwands entfallen kann. Es werden zwei Alternativen betrachtet:

¹ Das GLOBUS-Projekt wird vom Umweltministerium Baden-Württemberg gefördert. Neben LfU und FZK sind das FAW Ulm, das Forschungszentrum Informatik Karlsruhe und die Universitäten Stuttgart (IKE) und Karlsruhe (IPF) beteiligt.

- Zunächst Erstellung der Papierversion der Berichte mit einem Textverarbeitungssystem und nachträglich (weitgehend) automatische Konvertierung nach HTML, dem Dateiformat der WWW-Informationseiten.
- Gleichzeitige Erstellung der Papier- und der WWW-Version des Berichts.

Beide Alternativen haben charakteristische Vor- und Nachteile, die in Tabelle 1 gegenübergestellt und in den folgenden Unterkapiteln näher erläutert werden.

	Vorteile	Nachteile
Nachträgliche Konvertierung	Weniger Aufwand bei der Berichtserstellung. Keine Konsistenzprobleme.	Hoher Aufwand für die nachträgliche Konvertierung, Zerlegung und Verzögerung der Dokumente.
Gleichzeitige Erstellung beider Versionen	Vorhandenes Fachwissen kann unmittelbar zur Verzögerung der Dokumente benutzt werden.	Höherer Aufwand bei der Berichtserstellung selbst. Konsistenzprobleme während der Erstellung, da zwei Versionen des Berichts gepflegt werden müssen.

Tabelle 1: Vor- und Nachteile der beiden Alternativen

1.2.1 Alternative 1: Nachträgliche Konvertierung nach HTML

Die erste Alternative geht davon aus, daß ein Handbuch oder Fachbericht mit einem Textverarbeitungssystem erstellt und daraus eine Papierversion erzeugt wird. Die Integration des Berichts ins Altlasten-Informationssystem gehört nicht zur Erstellung des Fachberichts und erfolgt in einer separaten nachgeschalteten Stufe.

Dort wird mittels geeigneter Software-Werkzeuge das im Format des Textverarbeitungssystems gespeicherte Dokument in einzelne WWW-Seiten zerlegt, das Layout in das HTML-Format konvertiert und eine Verweisstruktur zwischen den Seiten aufgebaut. Die Bereitstellung eines solchen Konvertierungs-Werkzeugs ist mit beträchtlichem Aufwand verbunden. Um den Aufwand zu minimieren, wird ein bestimmtes Textverarbeitungssystem ausgewählt, und nur für dieses werden die entsprechenden Softwarewerkzeuge entwickelt. Das Textverarbeitungssystem muß folgende Bedingungen erfüllen:

- Es läuft auf PCs.
- Es muß in die DV-Landschaft der LfU passen.
- Eine weite Verbreitung sorgt für eine gewisse Zukunftssicherheit und erleichtert die Durchsetzbarkeit bei den Auftragnehmern der Fachberichte.

- Es sind Werkzeuge (Freeware, Shareware, kommerziell) verfügbar, die einen großen Teil der Konvertierung und Aufbereitung für das WWW übernehmen.
- Idealerweise können damit auch die HTML-Dokumente nachbearbeitet werden (um z.B. Querverweise auf andere Dokumente einzutragen).

Diese Kriterien werden am besten durch Word für Windows 6.0x von Microsoft erfüllt. Selbstverständlich kann in dem schnell veränderlichen DV-Markt die Softwareversion des Textverarbeitungssystems nicht festgeschrieben werden. Die Werkzeuge müssen immer an die aktuelle Version angepaßt werden.

Allerdings reicht die Einschränkung auf ein bestimmtes Textverarbeitungssystem allein noch nicht aus. Da das HTML-Format nur eine Untermenge der Gestaltungsmöglichkeiten der üblichen Textverarbeitungssysteme kennt, müssen bei der Erstellung der Berichte bestimmte Richtlinien befolgt werden, wenn eine weitgehend automatische Umsetzung möglich sein soll.

1.2.2 Alternative 2: Gleichzeitige Erstellung beider Versionen

Bei der Erstellung der Handbücher und Fachberichte wird das vorhandene Fachwissen besser genutzt, wenn nicht nur der konventionelle Bericht angefertigt, sondern auch gleich die WWW-Seiten erstellt werden. Da konventionelle Berichte und Hypertexte nicht denselben Designkriterien genügen, ist es allerdings nicht damit getan, den Bericht einmal mit einem Textsystem zu erstellen und dann in zwei Formaten abzuspeichern. Es ist tatsächlich mit einem gewissen Mehraufwand zu rechnen.

Diese Alternative läßt sich dadurch realisieren, daß ein Textverarbeitungssystem verwendet wird, welches als zusätzliche Option die Erstellung und Bearbeitung von HTML-Dokumenten erlaubt (siehe Anhang A.2 HTML-Editoren). Legt man sich auf dasselbe System wie bei Alternative 1 fest, kann bei einem bestimmten Bericht wahlweise eine der beiden Alternativen verwendet werden.

1.3 Zu diesem Bericht

Der vorliegende Bericht definiert Voraussetzungen zur Umsetzung der oben genannten Alternative 1, d.h. er enthält Richtlinien zur Gestaltung von neuen Handbüchern und Fachberichten der LfU. Werden diese eingehalten, sollte eine nachträgliche automatische Konvertierung eines Dokuments in das HTML-Format des WWW weitgehend problemlos möglich sein. Um die praktische Anwendbarkeit zu demonstrieren, ist dieser Bericht entsprechend den genannten Richtlinien erstellt worden und daher sowohl in Papierform als auch im WWW verfügbar.

Die längerfristig geplante Einbindung von Datenbanken und Anwendungsprogrammen in das Altlasten-Informationssystem bringt - unabhängig von der für die Einbindung der Texte gewählten Alternative - zusätzliche Anforderungen, die bei der Entwicklungen dieser Systeme berücksichtigt werden sollte. Diese Anforderungen werden ebenfalls kurz angeschnitten (s. Kap. 3. Datenbanken und Anwendungsprogramme).

Ein Vorteil der rechnergestützten Verbreitung von Berichten ist die Aktualität der Information. Dies soll ebenfalls mit dem vorliegenden Bericht demonstriert werden, indem nicht erst am Ende des Projekts eine vollständige Version verfügbar gemacht wird, sondern aufbauend auf einer ersten Basisversion die während der Aufbereitung der bestehenden LfU-Dokumente gewonnenen Erfahrungen unmittelbar eingearbeitet werden. Die jeweils aktuelle Version des Dokuments kann dann unmittelbar im WWW eingesehen werden.

2. Erstellung von Berichten

Dieses Kapitel beschreibt die Vorgehensweise zur Erstellung von Fachberichten und Handbüchern, die sowohl in Papierform ausgegeben als auch - nach weitgehend automatischer Konvertierung - ins WWW eingestellt werden können.

Damit dieser Ansatz überhaupt sinnvoll anwendbar ist, muß ein Dokument **klar und in ausreichender Tiefe strukturiert** sein, um eine Auftrennung in einzelne Hypertext-Seiten gemäß der Kapitelstruktur zu ermöglichen. Als grobe Richtschnur kann gelten, daß die tiefsten Unterkapitel, das sind die, welche nicht mehr weiter unterteilt werden, idealerweise zwischen 12 und 30 Zeilen lang sind. Die zweite Randbedingung der hier beschriebenen Vorgehensweise verlangt, daß die Dokumente in einem **definierten Format** erstellt werden. Um dies zu erreichen, wird davon ausgegangen, daß die Texterstellung am PC mit Word für Windows 6.0x² von Microsoft erfolgt. Es steht eine Dokumentvorlage zur Verfügung, die nur konvertierbare Formatvorlagen enthält und damit die Berichterstellung erleichtert. Bei Benutzung der Vorlage wird das Textdokument ähnlich dem Layout der WWW-Seiten gestaltet.

Die Dokumentvorlage ist Teil des frei verfügbaren Softwarepakets „rtftoweb“ von Christian Bolik, das wir als Basiskomponente des Konvertierungswerkzeugs einsetzen.

2.1 Die Dokumentvorlage

Um die Dokumentvorlage nutzen zu können, muß diese in Word für Windows geladen werden. Dazu sind folgende Schritte erforderlich:

- Menüpunkt "Datei/Dokumentvorlage..." aufrufen;
- Unter "Dokumentvorlage/Verbinden..." rtftoweb.dot auswählen;
- "Formatvorlagen automatisch aktualisieren" anwählen;
- Unter "Globale Vorlagen und Add-Ins" ebenfalls rtftoweb.dot auswählen und aktivieren;
- Mit "OK" Dialog beenden.

Es werden folgende Formatvorlagen geladen:

² In diesem Bericht wird meist als Abkürzung „WinWord“ verwendet.

- address
- blockquote
- bullet list; bullet list 1; bullet list 2
- DEF; DEF 1; DEF 2
- dir; dir 1; dir 2
- glossary; glossary 1; glossary 2
- hr
- HTML
- menu; menu 1; menu 2
- numbered list; numbered list 1; numbered list 2
- pre
- Standard
- TERM; TERM 1; TERM 2
- Überschrift 1; .. Überschrift 6

Weiterhin werden eine Reihe von Tastatur-Shortcuts³ definiert:

Strg-Shift -1 ... Strg-Shift -6	Formatvorlagen für Überschrift 1 bis 6
Strg-Shift -p	Formatvorlage „pre“.
Strg-Shift -b	Formatvorlage „bullet list“.
Strg-Shift -n	Formatvorlage „numbered list“.
Strg-Shift -g	Formatvorlage „glossary“.
Strg-Shift -h	Formatvorlage HTML.
Strg-Shift -r	Definiert den ausgewählten Text als Verweis auf eine (Unter-)Kapitelüberschrift.
Strg-Shift -i	Definiert den ausgewählten Text als URL (Adresse einer WWW-Seite).
Strg-Shift -u	Der ausgewählte Text wird auf der WWW-Seite zum Verweis auf eine andere WWW-Seite (deren Adresse mit Strg-Shift -i markiert wird).
Strg-Shift -a	Der ausgewählte Text wird zum Anker für einen Verweis.
Strg-Shift -c	Dem ausgewählten Text wird der Font „Courier New“ zugewiesen.
Strg-Shift -t	Dem ausgewählten Text wird der Font „Times New Roman“ zugewiesen.

In den folgenden Kapiteln werden die Formatvorlagen und Tastatur-Shortcuts näher beschrieben.

³ Strg-Shift -p bedeutet z.B.: während die beiden Tasten „Strg“ (auf manchen Tastaturen „Ctrl“) und „Shift“ (Umschalt-Taste) gleichzeitig gedrückt sind, wird zusätzlich die Taste „p“ betätigt.

2.2 Gliederungselemente

2.2.1 Titel

Der Titel eines Fachberichts wird auf jeder zugehörigen WWW-Seite wiedergegeben, um eine schnelle Zuordnung der Seite zu ermöglichen und damit dem „Lost in Cyberspace“-Effekt, der sich beim exzessiven Herumspringen im WWW leicht einstellt, entgegenzuwirken. Definiert wird der Titel des Berichts über das Datei-Info (s. Kap. 2.8 Das Datei-Info).

Möchte man den Titel auch dem eigentlichen Dokumenttext voranstellen, kann, um eine zweimalige Ausgabe des Titels auf der Titelseite der WWW-Version zu verhindern, wie folgt vorgegangen werden:

- Das Titelblatt des Berichts wird als separates WinWord-Dokument erstellt und nicht mit ins WWW übertragen.
Oder:
- Der Titel wird in das Textdokument eingefügt und mit der Formatvorlage „Titel“ versehen. Diese Vorlage kann über den Menüpunkt „Format/Formatvorlage“ von WinWord gefunden werden. Texte, die diese Formatvorlage verwenden, werden bei der Konvertierung des Dokuments nach HTML ignoriert.

2.2.2 Überschriften

(Kapitel-)Überschriften benennen logisch zusammengehörige Textteile und sind damit nahe-liegende Gliederungspunkte für die Aufteilung eines Berichts in einzelne WWW-Seiten. Die Gliederung eines Berichts in Kapitel erfolgt streng hierarchisch, d.h. die oberste Gliederungsebene sind die Hauptkapitel, die als „Überschrift 1“ zu kennzeichnen sind. Jedes Hauptkapitel kann in Unterkapitel unterteilt werden, die als „Überschrift 2“ gekennzeichnet werden. Diese wiederum lassen sich untergliedern in Unterkapitel mit „Überschrift 3“ als Formatvorlage usw. Wichtig ist, daß keine Sprünge auftreten, d.h. ein Kapitel mit Überschrift n wird immer nur in Unterkapitel mit Überschrift n+1 zerlegt.

Es sind Formatvorlagen für die Überschriften 1 bis 6 vorhanden, was in aller Regel genügend Gliederungsebenen bietet. Die Zuweisung der Formatvorlagen kann bei der Textbearbeitung auch über die Tastatur-Shortcuts Strg-Shift -1 ... Strg-Shift -6 vorgenommen werden.

Bei der Konvertierung des Dokuments kann angegeben werden, bis zu welcher Hierarchie-Ebene eine Aufteilung in separate WWW-Seiten vorgenommen werden soll. Dieses Dokument wird beispielsweise bis einschließlich „Überschrift 3“ zerlegt, d.h. tiefere Überschriften wie in Kapitel 2.2.4 bleiben auf einer WWW-Seite zusammen. Die Seiten werden automatisch verzeigert, d.h. es werden Verweise eingefügt auf die vorherige und nachfolgende Seite (entsprechend der Reihenfolge im Textdokument), auf die nächst-höhere Hierarchie-Ebene, auf die Titelseite, auf das Inhaltsverzeichnis und den Index.

Achtung: Leer- und sonstige Zeilen (Seitenwechsel, Abschnittswechsel) vor und nach einer Überschrift dürfen **nicht** mit einer Überschrift-Formatvorlage versehen sein, weil ansonsten eine leere und damit überflüssige WWW-Seite entsteht oder sogar die Zerlegung des Dokuments scheitert!

2.2.3 Index

Ein Index-Verzeichnis ist ein wichtiges Element zur Erschließung umfangreicher Texte und sollte deshalb unbedingt verwendet werden. Ein im Textdokument enthaltener Index wird bei der Konvertierung ins WWW übernommen, wo eine eigene Index-Seite eingerichtet wird.

Achtung: Bei der Erstellung des Index (Menüpunkt: „Einfügen/Index und Verzeichnisse“) muß als Format „Einfach“ angegeben werden.

Während im Textdokument ein Index-Eintrag die Nummern der Seiten listet, auf denen der jeweilige Begriff auftritt, enthält der WWW-Index direkte Verweise auf die relevanten Seiten. Da für jeden Verweis der Begriff wieder neu aufgeführt werden muß, werden Index-Verzeichnisse im WWW wesentlich voluminöser als im Textdokument. Als Konsequenz daraus wird empfohlen, beim Aufbau des Index im Textdokument genau abzuwägen, ob das Auftreten eines Begriffs an einer bestimmten Stelle wichtig genug ist, einen Eintrag im Index zu rechtfertigen. D.h. also statt der automatischen Aufnahme aller Erwähnungen eines Begriffs ist eine gezielte Auswahl vorzunehmen.

2.2.4 Weitere Gliederungselemente

2.2.4.1 Kopf- und Fußzeilen

In Word für Windows können Kopf- und Fußzeilen definiert werden, die z.B. die Seitenzahl, das Datum, beliebige Texte und auch Bilder enthalten. Dazu gibt es keine Entsprechung im WWW. Kopf- und Fußzeilen (z.B. Seitenangabe) werden daher beim Konvertieren verworfen.

Achtung: Bilder in Kopf- und Fußzeilen bereiten bei der Konvertierung derzeit Probleme und sind deshalb zu vermeiden!

2.2.4.2 Verzeichnisse

Bei der Konvertierung des Textdokuments wird automatisch ein Inhaltsverzeichnis erzeugt, das die Gliederungsstruktur des Dokuments vollständig wiedergibt und von einer separaten WWW-Seite aus jedes (Unter-) Kapitel über einen Verweis direkt zugreifbar macht.

Achtung: Abbildungs- und Inhaltsverzeichnis im Originaldokument (eingefügt über den Menüpunkt: „Einfügen/Index und Verzeichnisse“) führen zu massiven Problemen bei der Konvertierung und sind deshalb zu vermeiden!⁴

2.2.4.3 Seitenwechsel

Ein manuell definierter Seitenwechsel im Textdokument, der hier beispielsweise dafür sorgt, daß eine Überschrift nicht am unteren Ende einer Seite hängt, sondern auf einer neuen Seite erscheint, ist ein Gliederungselement, welches für das WWW keine Bedeutung hat und daher genau wie ein Abschnittswechsel beim Konvertieren ignoriert wird.

Dagegen wird auf WWW-Seiten oft eine horizontale Linie zur optischen Gliederung verwendet. Die Formatvorlage „hr“ fügt eine Linie in das Textdokument ein, die dann auch an die entsprechende Stelle der WWW-Seite übernommen wird.

2.3 Texte

2.3.1 „Normaler“ Fließtext

Die größten Teile eines Berichts werden typischerweise als Fließtexte (d.h. mit automatischem Zeilenumbruch) geschrieben. Solche Abschnitte sind mit der Formatvorlage „Standard“ zu formatieren. Unter Word für Windows gibt es zahlreiche Fonts mit verschiedenen Fontgrößen, die eine Gestaltung der Texte zulassen. Für die Konvertierung nach HTML ist davon jedoch nur die Unterscheidung der Fonts in proportionale Fonts (d.h. „i“ braucht weniger Platz als „m“) und nicht-proportionale Fonts (alle Buchstaben werden auf die gleiche Breite gebracht) von Bedeutung.

Über die Dokumentvorlage (siehe Kap. 2.1 Die Dokumentvorlage) sind die Fonts „Times New Roman“ (proportional) und „Courier New“ (nicht-proportional) als Standard-Schriftarten für das Textdokument definiert worden. Zwischen den beiden Schriftarten kann mittels Tastatur-Shortcuts gewechselt werden. Dazu ist der gewünschte Textabschnitt auszuwählen (hervorzuheben) und mit „Strg-Shift -c“ in „Courier New“ bzw. mit „Strg-Shift -t“ in „Times New Roman“ zu formatieren. Dieser Abschnitt ist beispielsweise im Textdokument in der Schriftart „Times New Roman“ gesetzt.

Mit der Formatvorlage „Standard“ belegte Textabschnitte werden bei der Konvertierung in Fließtext mit Proportionalschrift auf Seiten des WWW abgebildet, es sei denn, es wurde ein anderer Font (Schriftart) eingestellt, der dem Konvertierungsprogramm

⁴ Bei Bedarf kann das Textdokument vor dem Ausdrucken kopiert und in der Kopie mit Inhalts- und Abbildungsverzeichnis versehen werden.

als nicht-proportionaler Font bekannt ist. Dies ist bei „Courier New“, der Schriftart dieses Abschnitts, der Fall.

Wählt man einen anderen Font (wie z.B. hier „Monospaced“), so muß man in der Regel davon ausgehen, daß dieser auf den standardmäßigen Proportionalfont in HTML abgebildet wird, weil er dem Konvertierungs-Werkzeug nicht bekannt ist.

Die Größe des Fonts kann im Fließtext beliebig gesetzt werden. Eine Auswirkung auf die Fontgröße im WWW ist nicht gegeben.

Möchte man einzelne Worte oder Wortfolgen in einem Fließtext besonders hervorheben, so können diese entweder **fett** oder *kursiv* gesetzt werden. Auch eine Kombination aus beiden also **fett und kursiv** ist möglich. Man kann aber auch in einem Abschnitt, der in Proportional-schrift geschrieben ist, einzelne Teile durch nicht-proportionale Schrift hervorheben. Auch eine Kombination der anderen Schriftart mit **fett** und *kursiv* bleibt bei der Umsetzung ins WWW erhalten. Unterstreichungen sollten allerdings nicht als Mittel zur Hervorhebung verwendet werden. Wie wir später sehen werden, hat das Unterstreichen eine besondere Bedeutung im Zusammenhang mit der Definition von Verweisen (s. Kap. 2.6 Verweise).

2.3.2 Vorformatierter Text

Man kann zwar nicht-proportionale Schriften auch für Fließtext einsetzen, wie wir im letzten Kapitel gesehen haben, doch werden diese hauptsächlich für vorformatierte Texte benutzt. Dabei wird die Position der Zeichen auf der Seite bzw. in der Zeile als gestalterisches Element verwendet und soll deshalb bei der Umsetzung ins WWW und unabhängig von der Größe des Browser-Fensters erhalten bleiben.

Beispiel für einen vorformatierten Text:

Rechnung:	PC 486/DX2	1600,-- DM
	15''-Bildschirm	500,-- DM
	CD-ROM	450,-- DM
	Drucker	600,-- DM
		=====
	Summe:	3150,-- DM

Für vorformatierten Text ist die Formatvorlage „pre“ zu verwenden. Die Vorlage kann auch über den Tastatur-Shortcut „Strg-Shift -p“ zugewiesen werden. Im Gegensatz zu Fließtext sind bei vorformatiertem Text keine Hervorhebungen zulässig.

2.3.3 Weitere Hervorhebungen

Zum Hervorheben bestimmter Textabschnitte stehen zwei weitere Formatvorlagen zur Verfügung:

Die Formatvorlage „address“ wird für Adressen verwendet. Normalerweise wird nur die Anschrift der Berichtsersteller in dieses Format gesetzt, wie es auch bei dem vorliegenden Dokument auf der Titelseite zu sehen ist.⁵

Für Zitate, d.h. Texte, die von anderen Autoren übernommen wurden, kann die Formatvorlage „blockquote“ benutzt werden. Eine Zuweisung ist nur zu vollständigen Absätzen möglich, nicht aber zu Textabschnitten innerhalb eines Absatzes, wie es vielleicht wünschenswert wäre. Hier gilt

nobody is perfect,

was gleichzeitig das Beispiel für die Formatvorlage „blockquote“ ist.

2.3.4 Zeichenvorrat

Die Menge der im Text verwendbaren Zeichen wird begrenzt durch die in HTML definierten und damit im WWW darstellbaren Zeichen. Im einzelnen sind dies:

- Alle Buchstaben (inkl. Umlaute und ß) und die 10 Ziffern 0...9.
- Die auf der (deutschen) Tastatur vorhandenen Sonderzeichen:
! " § \$ % & / () = ? * + ' # < > , . - ; : _
- Viele der Sonderzeichen, die über den Menüpunkt Einfügen/Sonderzeichen... (Auswahl 1; Schriftart: (normaler Text)) in den Text eingefügt werden können:
@ [\] ^ _ ` { | } ~
¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾
¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
- Weitere über den genannten Menüpunkt einfügbare Sonderzeichen sind in HTML nicht definiert, werden aber bei der Konvertierung in eine Ersatzdarstellung (= ähnliche Darstellung) gewandelt. Es sind dies die Zeichen⁶:
, f „ ... † ‡ ^ % Œ ‹ € ‘ ’ “ ” • – — ~ ™ § › œ Ÿ

Weitere Zeichen, insbesondere die über die Schriftart „Symbol“ verfügbaren mathematischen Zeichen und griechischen Buchstaben sind in der aktuellen Version von HTML nicht definiert und fallen daher bei der Konvertierung ersatzlos weg.

⁵ Die Autoren der im Konvertierungswerkzeug eingesetzten Basissoftware planen in einer späteren Version, die mit dieser Formatvorlage versehenen Texte auf jeder HTML-Seite zu wiederholen! Deshalb sollte die Vorlage maximal einmal pro Bericht verwendet werden.

⁶ In Originaldarstellung sind diese Zeichen natürlich nur in der Papierversion des Berichts enthalten!

Eine weitere Texteigenschaft mit nicht unerheblicher Bedeutung für technische Berichte ist die Hoch- bzw. Tiefstellung von Zeichen (z.B. H_2O , m^3 , 10^{-9}). Man markiert dazu die gewünschten Zeichen und wählt in dem Formular, das man über den Menüpunkt „Format/Zeichen...“ erhält, unter „andere Stilmerkmale“ entweder „Hochgestellt“ oder „Tiefgestellt“ aus. Bei der Konvertierung werden die entsprechenden HTML-Anweisungen erzeugt. Die aktuellen Versionen der WWW-Browser können die Hoch- bzw. Tiefstellung von Text noch nicht anzeigen (Ausnahme: Arena), doch ist die entsprechende Funktionalität in absehbarer Zeit zu erwarten (z.B. angekündigt für Netscape 2.0).

2.4 Listen, Aufzählungen, Tabellen

2.4.1 Listen und Aufzählungen

Die Erstellung von Listen und Aufzählungen erfolgt über die Formatvorlagen:

- bullet list; bullet list 1; bullet list 2
- glossary; glossary 1; glossary 2
- numbered list; numbered list 1; numbered list 2

Diese haben folgende Bedeutung:

bullet list	(normale) Liste Die einzelnen Element der Liste werden eingerückt dargestellt, können mehrere Zeilen überspannen und umfassen genau einen Absatz. Wahlweise kann jedem Listenelement ein Aufzählungszeichen vorangestellt werden (bullet = Punkt). Die Liste am Anfang dieses Unterkapitels ist mit der Vorlage „bullet list“ formatiert.
numbered list	nummerierte Liste Entspricht der normalen Liste, die Listenelemente werden jedoch fortlaufend durchnummeriert (nach Auswahl des entsprechenden Icons für die Nummerierung).
glossary	Glossar Ein Glossar enthält Begriffe und zugehörige Erläuterungen. Diese Liste ist ein Beispiel für ein Glossar. Bei der Erstellung des Glossars wird bei jedem Element der Liste nach dem Begriff ein TAB-Zeichen und anschließend dessen Erläuterung eingegeben.

Um eingerückte Listen erstellen zu können, gibt es für jede der drei Listenarten 3 Formatvorlagen. So ist die „bullet list 1“ eine normale Liste, die um eine Stufe gegenüber „bullet list“

eingerrückt ist und „bullet list 2“ eine um eine weitere Stufe eingerrückte Liste. Entsprechendes gilt für nummerierte Listen und Glossare.

Die Formatvorlagen können auch kombiniert werden. Zu beachten ist lediglich, daß keine Einrückungsstufe übergangen wird. In dem folgenden Beispiel entsprechen die Texte der Listenelemente jeweils der verwendeten Formatvorlage:

1. numbered list
 - bullet list 1
 - eins glossary 2
 - zwei glossary 2
 - bullet list 1
2. numbered list
 - bullet list 1
 - bullet list 2
 - bullet list 1

Die drei Formatvorlagen können Absätzen auch über Tastatur-Shortcuts zugewiesen werden und zwar „Strg-Shift -b“ für „bullet list“, „Strg-Shift -n“ für „numbered list“ und schließlich „Strg-Shift -g“ für „glossary“.

2.4.2 Weitere Formatvorlagen für Listen/Aufzählungen

In der Regel sind die Formatvorlagen in Kapitel 2.4.1 ausreichend. HTML kennt jedoch noch zwei Spezialformen obiger Listen, für die ebenfalls WinWord-Formatvorlagen vorhanden sind. Es sind dies:

- dir; dir 1; dir 2
- menu; menu 1; menu 2

Ein Menü (Formatvorlage „menu“) ist eine normale Liste (bullet list), bei der die einzelnen Elemente maximal eine Zeile lang sein dürfen. Ein HTML-Browser, wie z.B.

Netscape
Mosaic
Arena

(Beispiel für „menu“) kann (muß aber nicht) diese spezielle Listenform kompakter als normale Listen darstellen. Eine u.U. noch kompaktere Darstellung kann mit der Verwendung der Formatvorlage „dir“ verbunden sein. Hier dürfen die einzelnen Listenelemente maximal 20 Zeichen lang sein und können auch, durch TABs getrennt, im Textdokument in einer Zeile aufgeführt werden, wie z.B.

Montag	Dienstag	Mittwoch
Donnerstag	Freitag	Samstag

Achtung: Bei den beiden genannten Formatvorlagen dürfen **keine** Aufzählungszeichen verwendet werden!

2.4.3 Tabellen

Tabellen, erstellt über den Menüpunkt „Tabelle/Tabelle einfügen...“ in WinWord, können in Fachberichten benutzt werden. Beim Konvertieren und Aufbereiten des Fachberichts für das WWW werden Tabellen derzeit in vorformatieren Text (vgl. Kap. 2.3.2 Vorformatierter Text) gewandelt, wobei lediglich die Grundstruktur der Tabelle erhalten bleibt.⁷

Beispiele für Tabellen finden Sie in diesem Bericht in Kapitel 1.2 (Kap. 1.2 Konsequenzen für die Erstellung von Berichten) und hier direkt anschließend:

Überschrift 1	Überschrift 2	Überschrift 3
Zeile 1	Text 1	Text 2
Zeile 2	Text 3	Text 4
Zeile 3	Hier wurden zwei Zellen zusammengefaßt.	

Tabelle 2: Beispiel für eine Tabelle

Bei der Erstellung sollte beachtet werden:

- In einer Zelle darf nur **ein** Absatz enthalten sein.
- Fußnoten in Tabellen sind nicht möglich.

2.5 Abbildungen

Leistungsfähige Textverarbeitungssysteme erlauben die Einbindung von Abbildungen in Dokumente:

- Abbildungen werden in das Dokument integriert, indem entweder ein mit einem anderen Software-Werkzeug erstelltes Bild eingefügt wird oder dadurch, daß die Abbildung mit einer entsprechenden Komponente des Textverarbeitungssystems direkt erstellt wird.
- Alternativ können die Abbildungen auch in separaten Dateien gespeichert bleiben, zu denen dann Verknüpfungen aus dem Textdokument hergestellt werden können.

Ebenso wie für Textdokumente gibt es für Graphiken sehr viele unterschiedliche Dateiformate. Man unterscheidet dabei zwischen Vektorgraphikformaten, wie sie typischerweise von Programmen wie Corel Draw verwendet werden - hier bleiben die einzelnen Objekte einer

⁷ Da HTML in der nächsten Version auch direkt Tabellen darstellen kann, soll das Konvertierungswerkzeug mittelfristig diese auch unterstützen.

Zeichnung separat bearbeitbar - und Rastergraphiken (Bitmaps; z.B. GIF, JPEG, BMP), die mit Malprogrammen wie Paintbrush oder Corel Photo Shop auf Pixelebene bearbeitet werden können.

Während Textverarbeitungsprogramme meist mit einer größeren Zahl verschiedener Graphikformate umgehen können, beschränken sich die WWW-Browser bisher i.w. auf die Rasterformate GIF und JPEG. Bei anderen Formaten ist man auf externe Viewer, d.h. separate Softwaretools, die durch den Browser aufgerufen werden, wenn dieser die Formate nicht selbst bearbeiten kann, angewiesen. Bei diesen Viewern handelt es sich meist um kleinere Shareware-Programme, die im Gegensatz zu den Browsern jeweils nur für spezielle Plattformen erhältlich sind und bei Bedarf im Internet besorgt werden können. Um eine möglichst problemlose Verbreitung von Informationen im WWW zu erreichen, empfiehlt es sich, nur die von den Browsern selbst unterstützten Formate zu verwenden.

Da es für die schrittweise Erstellung von Dokumenten andererseits meist günstiger ist, Abbildungen in einem Vektorgraphik-Format zu bearbeiten, können sich durch die Abbildungen eines Dokuments u.U. Probleme bei der Bereitstellung für das WWW ergeben. Kapitel 2.5.1 beschreibt den **Idealfall**, bei dem eine Konvertierung automatisch durchgeführt werden kann. Die folgenden Unterkapitel beschreiben mögliche Alternativen, bei denen ein gewisser Zusatzaufwand erforderlich ist.

2.5.1 Voraussetzungen für eine automatische Konvertierung

Um Abbildungen problemlos konvertieren zu können, müssen folgende Randbedingungen eingehalten werden:

- Jede Abbildung liegt als separate Datei im GIF- oder JPEG-Format vor.
- Die Abbildung wird mit dem Textdokument verknüpft (und nicht eingefügt).
- Das Textdokument wird mit Word für Windows 6.0 erstellt.

Bei der Konvertierung nach HTML wird an der Stelle im Dokument, an der die Verknüpfung zur Abbildung verankert ist, ein Link zur Graphik in Form eines Icons platziert. Die Abbildung wird also im WWW nicht direkt in der Textseite angezeigt, sondern erst auf Anforderung geladen. Die Bildunterschrift verbleibt auf der Textseite. Abbildung 1 zeigt ein Beispiel für eine verknüpfte Abbildung, die in einem GIF-File liegt.

**Abb. 1: mit Textdokument
verknüpfte GIF-Datei**

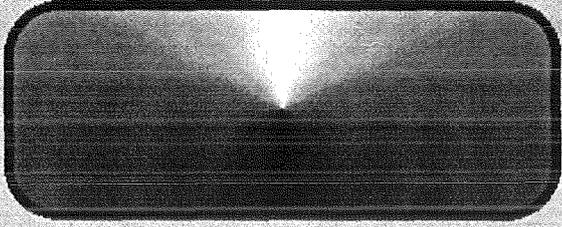


Abbildung 1: Beispiel für eine verknüpfte GIF-Datei

2.5.2 Verknüpfte Abbildungen in anderen Formaten

Die zweite Alternative geht davon aus, daß Abbildungen in separaten Dateien vorliegen, mit dem Textdokument verknüpft sind und (auch) von GIF und JPEG abweichende Formate verwendet wurden. Im Prinzip ist in diesem Fall dieselbe Vorgehensweise wie in 2.5.1 möglich. Im WWW werden dann Links auf die Graphikdateien im jeweiligen Format erzeugt. Da dies die Browser - wie oben geschildert - nicht ohne weiteres verarbeiten können, ist diese Vorgehensweise nicht sonderlich zweckmäßig.

Statt dessen kann bei der Konvertierung des Textdokuments nach HTML ein Graphikformat angegeben werden (sinnvollerweise GIF oder JPEG), für das Links erzeugt werden sollen. Wurde beispielsweise in der Textverarbeitung eine Verknüpfung mit der im Computer Graphics Metafile Format erstellten Datei „abb2.cgm“ hergestellt und bei der Konvertierung als Format GIF angegeben, wird in der entsprechenden HTML-Seite ein Link auf die Datei „abb2.gif“ verankert (siehe Abb.2). Das angegebene Graphikformat wird immer für **alle** Links des Dokuments auf Abbildungen verwendet!

**Abb.2: Original in CGM
fürs WWW gewandelt nach GIF**

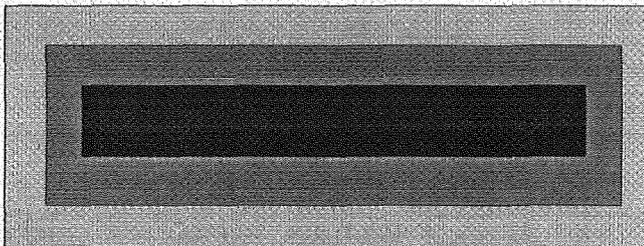


Abbildung 2: Beispiel für eine verknüpfte Datei, die fürs WWW konvertiert werden muß.

Die Konvertierung der Graphikdatei in das GIF-Format wird nicht automatisch durchgeführt, sondern muß separat erfolgen. Es gibt verschiedene Möglichkeiten, wobei es von Quell- und Zielformat abhängt, welche davon im Einzelfall verwendet werden können, da jedes Werkzeug immer nur eine Teilmenge der Formate kennt:

- über spezielle Konvertierungsprogramme, die es z.T. als Shareware gibt (z.B. LVIEW⁸)
- über die verbreiteten PC-Graphikprogramme, da diese i.d.R. neben ihrem eigenen Format auch fremde Formate im- und exportieren können.

Abbildung 2 wurde z.B. im CGM-Format nach CorelDraw importiert und im GIF-Format wieder exportiert.

2.5.3 Eingefügte Abbildungen

Bilder, die in das Textdokument eingefügt wurden, die also nicht mehr in einer separaten Datei liegen, können verwendet werden, wenn diese entweder das Windows Meta-File- (WMF) oder das Windows Bit-Map-Format (BMP) verwenden. Bei der Konvertierung wird die eingefügte Abbildung in einen separaten WMF-File exportiert. Der Name der exportierten Datei wird dabei automatisch aus dem Namen des Textdokuments (RTF-File⁹) und einer laufenden Nummer generiert. Je nach dem, ob bei der Konvertierung ein Graphik-Format angegeben wurde, wird entweder ein Link aus der betreffenden HTML-Seite auf den WMF-File (wenig sinnvoll) oder auf einen File in dem angegebenen Format angelegt. Dieser File muß aus dem WMF-File oder dem Original, das ursprünglich in das Textdokument eingefügt wurde, erzeugt werden (s. Kap. 2.5.2 Verknüpfte Abbildungen in anderen Formaten).

In dem nachfolgenden Beispiel (Abb. 3) wurde ein WMF-File in das Dokument eingefügt. Bei der Konvertierung wird als gewünschtes Graphik-Formats für die HTML-Links GIF angegeben. Für das WWW wird daher ein Link auf die Datei „richtl1.gif“ angelegt. Der Name der Datei entsteht aus der Bezeichnung der RTF-Datei (richtl.rtf) und der laufenden Nummer 1.

⁸ Das Buch „Murray, J.D./ vanRyper, W. (1994): Encyclopedia of Graphics File Formats. O'Reilly, Sebastopol, CA“ enthält neben der Beschreibung von ca. 100 Graphik-Formaten auch eine CD-ROM mit Shareware-Konvertierungsprogrammen für DOS, Windows, Macintosh und UNIX.

⁹ Das Textdokument muß zur Konvertierung nach HTML statt im normalen WinWord-Format als RTF-Datei (Rich Text Format) abgespeichert werden. RTF wird als Austauschformat von einer Vielzahl von Textverarbeitungssystemen unterstützt.

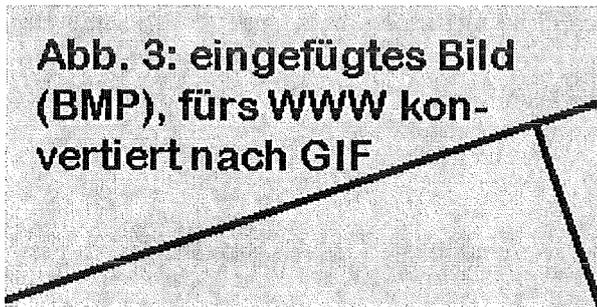


Abbildung 3: Beispiel für eine eingefügte Abbildung

2.5.4 Platzhalter für Abbildungen

Bei vorhandenen Dokumenten ist oftmals nur der Textteil auf dem Rechner verfügbar, während die Abbildungen (Fotos, Zeichnungen) manuell in die ausgedruckten Seiten eingeklebt werden. Will man diese Berichte im WWW verfügbar machen, müssen die Bilder nachträglich eingefügt werden. Dazu gibt es zwei Möglichkeiten:

- Die WWW-Seiten werden erst einmal ohne Bilder erzeugt und anschließend, wenn die Bilder auf dem Rechner verfügbar sind (eingescannt), mit einem HTML-Editor nachbearbeitet, d.h. um die Links ergänzt.
- Im Originaldokument werden Platzhalter definiert, die während der Konvertierung in Links umgesetzt werden. Diese Links zeigen solange ins Leere, bis die Bilder eingescannt sind.

Da die zweite Alternative Änderungen an den Originaldokumenten flexibler handhabt (die Links müssen nicht immer wieder neu definiert werden), wird diese empfohlen. Für Abbildung 4 (im Textdokument!) ist unten vor der Abbildungsbeschriftung ein Platzhalter definiert, der sich aus einem Schlüsselwort (XXXPICT) und dem Namen der Graphik-Datei „abb4.gif“ zusammensetzt. Der Platzhalter ist als verborgener Text formatiert, sodaß dieser standardmäßig nicht angezeigt und ausgedruckt wird. In der WWW-Version dieses Berichts ist der Platzhalter durch die entsprechende Abbildung ersetzt:

Abbildung 4: Beispiel für eine erst nachträglich verfügbare Abbildung

Im Platzhalter muß das Schlüsselwort in Großbuchstaben stehen und der Dateiname in Leerzeichen eingeschlossen werden. Alternativ kann auch ein Platzhalter (Schlüsselwort: XXXIMG) für eine Graphik definiert werden, die direkt in der HTML-Seite erscheinen soll, d.h. die nicht erst durch Anklicken eines Icons geladen werden muß. Diese, in der folgenden Abbildung demonstrierte Möglichkeit, sollte man allerdings nur bei sehr kleinen (auf Dateigröße bezogen: < 10kB) Abbildungen wahrnehmen.

Abbildung 5: Beispiel für eine erst nachträglich verfügbare Abbildung, die direkt in die WWW-Seite eingefügt wird.

Wie in 2.5.2 und 2.5.3 gilt, daß die Angabe eines Graphikformats bei der Konvertierung den erzeugten Link dahingehend ändert, daß auf einen File gleichen Namens aber mit einer dem angegebenen Format entsprechenden Endung verwiesen wird.

2.5.5 Verwendung verschiedener Alternativen in einem Dokument

Entscheidend für den Aufbau der Links zu den Abbildungen ist, ob bei der Konvertierung ein Graphikformat angegeben wurde. Diese Angabe gilt jeweils generell für alle Links eines Berichts und bestimmt somit auch, welche der Alternativen in den Kapitel 2.5.1-4 (in Kombination) verwendet werden können. Sinnvoll ist:

Bei der Konvertierung wird kein Graphikformat angegeben:

- verknüpfte Graphiken im GIF und/oder JPEG-Format (Kap. 2.5.1) und/oder
- Platzhalter für Graphiken im GIF und/oder JPEG-Format (Kap. 2.5.4)

Es wird ein Graphikformat angegeben (GIF oder JPEG):

- verknüpfte Graphiken in beliebigem Format (Kap. 2.5.2) und/oder
- eingefügte Graphiken (Kap. 2.5.3) und/oder
- Platzhalter für Graphiken im angegebenen Format (Kap. 2.5.4)

2.5.6 Empfehlungen für die Erstellung von Abbildungen

Die unterschiedlichen Darstellungsmöglichkeiten von Bildschirmgeräten und (Laser-) Druckern führen dazu, daß bereits bei der Erstellung von Abbildungen auf deren zweifache Verwendung Rücksicht genommen wird. Aus den bisherigen Erfahrungen ergeben sich folgende Empfehlungen für die Erstellung und das Einscannen von Abbildungen (bei späterem Zugriff auf die Abbildungen über Netz):

- Bei der Beschriftung von Zeichnungen sollte auf eine Schriftgröße im Enddokument von mindestens 10 Punkten, möglichst sogar 12 Punkten geachtet werden (1 Punkt entspricht 0.035 cm).
- Eine Auflösung von 75 dpi beim Scannen genügt i.allg. für die Darstellung auf Bildschirm. Bei kleiner Schriftgröße der Beschriftung von Zeichnungen oder bei feinen relevanten Details sollte beim Einscannen mit einem Skalierungsfaktor gearbeitet werden. Daumenregel: Skalierungsfaktor so wählen, daß die Schrift auf eine Größe von 10-12 Punkten vergrößert wird. Bei Qualitätsanforderungen entsprechend Laser-Druckern muß zu einer höheren Auflösung übergegangen werden.
- Schwarz-Weiß-Fotos, Rasterbilder und farbige Abbildungen sollten nur in Ausnahmefällen verwendet werden.

Die Vorlagen sollten möglichst auf weißem Papier, nicht auf grauem Umweltpapier, vorliegen.

2.6 Verweise

2.6.1 Fußnoten

Im Text können an beliebiger Stelle Fußnoten¹⁰ bzw. Endnoten eingefügt werden. Bei der Konvertierung werden diese in **eine** eigene WWW-Seite geschrieben. An der Stelle im Text, an der die Fuß- bzw. Endnoten verankert wurden, wird jeweils ein aktiver Verweis¹¹ eingerichtet, über den die zugehörigen Texte erreichbar sind. Um von dort wieder zu der Seite zurückzukehren, auf welcher der Anker ausgewählt wurde, ist die „Back“-Option des HTML-Browsers zu verwenden.

Achtung: In einem Bericht sollten entweder nur Fußnoten oder nur Endnoten verwendet werden, nicht aber beides gleichzeitig! Bei der Konvertierung ergeben sich ansonsten Nummerierungsprobleme.

2.6.2 Verweise auf Kapitelüberschriften

Innerhalb eines Berichts können an beliebiger Stelle Verweise auf andere Kapitel definiert werden. Dazu ist die Kapitelüberschrift, auf die verwiesen werden soll, vollständig und in exakt der gleichen Weise, wie sie im Bericht erscheint, in den Text einzufügen (am besten kopieren) und mit dem Tastatur-Shortcut (s. Kap. 2.1 Die Dokumentvorlage) „Strg-Shift -r“ als Verweis zu kennzeichnen. Der entsprechende Text erscheint anschließend in roter Schrift.

Beim Konvertieren wird daraus ein aktiver Verweis auf das entsprechende Kapitel erzeugt, d.h. im WWW kann man den Text anklicken, worauf die entsprechende Seite angezeigt wird.

Achtung: Der als Verweis markierte Text sollte im Textdokument möglichst in einer Zeile stehen und nicht unterbrochen werden (z.B. durch einen Zeilenumbruch).

2.6.3 Sonstige Verweise

Prinzipiell können im Textdokument Verweise auf beliebige Stellen im selben Bericht, in anderen Berichten oder auf jede WWW-Seite weltweit definiert werden. Das Problem ist dabei allerdings, daß man dazu die genaue Adresse (URL) kennen muß, was wiederum eine gewisse Vertrautheit mit den Adressierungsprinzipien des WWW voraussetzt und man sich außerdem von Dritten abhängig macht (z.B. wegen den häufigen Änderungen von WWW-

¹⁰ Über das Menü „Einfügen/Fußnote...“: Die Felder „Fußnote“ (bzw. „Endnote“) und „Nummerierung/automatisch“ auswählen.

¹¹ Ein aktiver Verweis ist eine markierte Textstelle (normalerweise farblich hervorgehoben und unterstrichen), die mit der Maus angeklickt werden kann, worauf der Browser die WWW-Seite, auf die der Verweis zeigt, aufblättert.

Adressen). In der Regel wird man diese Methode in den Fachberichten der Umweltverwaltung daher nicht anwenden.

Normalerweise will man die WWW-Adressen verbergen und statt dessen verständlichere Texte als Verweise angeben. Dazu schreibt man in das Dokument zuerst die Adresse und anschließend - ohne ein Leerzeichen dazwischen - den Text, der als Verweis dienen soll. Als nächstes ist der Verweistext auszuwählen und mit dem Tastatur-Shortcut „Strg-Shift -u“ zu kennzeichnen. Danach wird die Adresse ausgewählt und mit „Strg-Shift -i“ markiert.

Der voranstehende Verweis auf die Liste der Tastatur-Shortcuts wurde auf diese Weise definiert. Als Adresse ist dabei angegeben: „richtl-2.1.html#Shortcut“. Die Adresse setzt sich in diesem Fall zusammen aus dem Namen der HTML-Datei (richtl-2.1.html), die sich im selben Verzeichnis befinden muß, wie die Datei mit dem Verweis, und der Bezeichnung eines Ankers (hier: „Shortcut“). Ein Anker wird benötigt, wenn man eine HTML-Seite nicht am Anfang, sondern an einer bestimmten Position aufschlagen will. Ein Anker wird angelegt, indem man

1. die Eingabemarke an die gewünschte Stelle bringt,
2. die gewünschte Bezeichnung des Ankers eingibt,
3. die Bezeichnung auswählt und
4. über den Tastatur-Shortcut „Strg-Shift -a“ als Anker kennzeichnet (verborgener Text).

Zahlreiche weitere Beispiele für Verweise finden sich im Anhang.

2.7 Spezielle HTML-Formatierungen

Formatvorlagen, die sehr HTML-spezifische Formatierungen unterstützen und daher auch genauere Kenntnisse von HTML erfordern, haben Bezeichnungen mit Großbuchstaben. Es sind dies:

- DEF; DEF 1; DEF 2
- TERM; TERM 1; TERM 2
- HTML

Die ersten beiden erlauben die Erstellung einer „Definition List“. Dabei handelt es sich um die Verallgemeinerung des Glossars (s. Kap. 2.4.1 Listen und Aufzählungen). Eine Definition List dient zur Definition von Begriffen. Einem oder mehreren Begriffen (Terme) folgt jeweils deren Definition. Beispiel:

TERM

TERM1

TERM2

Die Formatvorlage TERM (bzw. die entsprechenden Einrückungen TERM1 und TERM2) wird zur Formatierung der Begriffe (Terme, Bezeichnungen) verwendet.

DEF

DEF1

DEF2

Die Formatvorlage DEF (bzw. die entsprechenden Einrückungen DEF1 und DEF2) wird zur Formatierung der Erläuterungen (Definitionen) benutzt.

Die beiden Formatvorlagen „TERM“ und „DEF“ gehören also zusammen und müssen immer in Kombination verwendet werden!

Schließlich gibt es noch die Möglichkeit, in das Textdokument direkt HTML-Anweisungen einzufügen. Damit diese bei der Konvertierung erkannt und unverändert in den HTML-File weitergeleitet werden, sind sie mit der Formatvorlage „HTML“ zu kennzeichnen. Diese Vorlage kann auf beliebige Textabschnitte angewendet werden und nicht nur auf vollständige Absätze, wie z.B.:

Dieser Text wurde mit HTML-Anweisungen zur Zentrierung eingerahmt!

Als HTML-Anweisungen markierte Texte werden als „verborgen“ gekennzeichnet. Sie sind also im Normalfall im Textdokument nicht sichtbar. Die Formatvorlage kann markierten Textabschnitten auch über den Tastatur-Shotcut „Strg-Shift -h“ zugewiesen werden.

2.8 Das Datei-Info

Das Datei-Info wird in WinWord über das Menü „Datei/Datei-Info...“ aufgerufen und erlaubt in einem Formular die Eingabe beschreibender Informationen zu dem aktuellen Bericht. Bei der Konvertierung eines Berichts werden die enthaltenen Informationen ausgewertet, um den Bericht im Umweltinformationssystem von Baden-Württemberg bekannt zu machen (Eintrag in die Metainformationssysteme Ressourcenkatalog/Umweltdatenkatalog). Der aus dem Datei-Info entnommene Titel des Berichts (s. Kap. 2.2.1 Titel) wird zusätzlich in die erzeugten WWW-Seiten eingefügt.

Das Datei-Info enthält insgesamt folgende Felder:

Dateiname	Name der aktuellen Datei (Ausgabefeld)
Verzeichnis	Verzeichnis, in dem die Datei abgelegt ist (Ausgabefeld)
Titel	Titel des Berichts
Thema	Inhalt (dieses Feld wird bei der Konvertierung derzeit ignoriert)
Autor	Autor(en) bzw. Herausgeber. Falls mehrere Personen angegeben werden, sind diese durch „;“ zu trennen.
Stichwörter	Es können ein oder mehrere Stichwörter (Keywords) angegeben werden. Stichwörter nach dem Thesaurus des Umweltbundesamts werden später automatisch nachgetragen und brauchen daher nicht explizit eingetragen werden.
Kommentar	Kurze Beschreibung des Inhalts des Berichts (Abstract).

3. Datenbanken und Anwendungsprogramme

Bei den in Frage kommenden Datenbanken und Anwendungsprogrammen handelt es sich bzgl. der Programmkomplexität um einfachere DV-Systeme, die für die Altlasten-Sachbearbeiter (i.d.R. DV-Laien) konzipiert werden und über die Benutzeroberfläche vorgegebene Funktionen anbieten. Frei definierte Anfragen, wie sie beispielsweise mit dem SELECT-Statement von SQL formulierbar sind, können nicht verwendet werden.

Ziel ist es in erster Linie, die vorhandene Funktionalität auch über WWW anzubieten. Eine weitergehende Öffnung der Systeme wie sie im Rahmen des GLOBUS-Projekts (Stichwort: Dienstekonzept) diskutiert wird, d.h. das Anbieten einer (objekt-orientierten) Schnittstelle zur Integration der jeweiligen Anwendung in andere Umweltinformations-Systeme ist eine weiterführende Aufgabe, die derzeit für das Altlasten-Informationssystem nicht unmittelbar relevant ist.

Eine detaillierte Spezifikation der Anforderungen an ins WWW integrierbare Datenbanksysteme und Anwendungsprogramme soll 1996 am praktischen Beispiel eines oder mehrerer existierender Systeme der LfU erfolgen. An dieser Stelle sollen vorab einige Grundvoraussetzungen genannt werden:

- Anwendungsprogramm und Benutzeroberfläche sind klar zu trennen, um ein Austausch der Benutzeroberfläche ohne Eingriffe ins Anwendungsprogramm zu ermöglichen.
- Die Benutzeroberfläche muß relativ einfach aufgebaut sein. Entsprechend den Möglichkeiten von HTML ist eine Formular-orientierte Schnittstelle anzustreben.
- Wegen der Antwortzeiten im Internet sollte die auszutauschende Datenmenge und besonders die Anzahl der Kommunikationsvorgänge zwischen Anwendungsprogramm und Benutzer minimiert werden.
- Das WWW kennt kein Sessionmanagement oder Transaktionskonzept, d.h. das Aufrufen einer WWW-Adresse kann zwar das Ausführen irgendwelcher Programme auslösen und von dort Ergebnisse auf einer dynamisch-erzeugten Seite präsentieren, ein Aufsetzen auf den Ergebnissen in einer späteren Anfrage ist jedoch nicht unmittelbar möglich. Am einfachsten integrierbar sind deshalb Systeme, bei denen
 - die Benutzeranfragen den inneren Zustand des Systems überhaupt nicht ändern (z.B. Datenbankabfrage) oder
 - zwar Fall-/Anfrage-spezifische Daten und Zwischenergebnisse anfallen, diese aber vollständig eingekapselt und über eine eindeutige Kennung vom Benutzer bei späteren Anfragen wieder verwendet werden können.
- Das Anwendungsprogramm muß in einer Multitasking-Umgebung laufen und vom WWW-Server über Interprozess-Kommunikation ansprechbar sein (z.B. UNIX, Windows NT).
- Eine objekt-orientierte Realisierung des Anwendungssystems erleichtert zum einen die bereits angeschnittene Trennung von Benutzeroberfläche und Kernsystem und schafft andererseits die Voraussetzung für eine Öffnung nach dem Dienstekonzept.

Anhang

A. Zur Erstellung von WWW-Seiten

Die Grundlagen von Internet und WWW werden inzwischen in zahlreichen Büchern und online verfügbaren Artikeln beschrieben, wie z.B.:¹²

- „Internet: Werkzeuge und Dienste“ von Scheller et al. (Springer Verlag),
- „Das WWW-Kompodium“ von Rainer Klute ist sowohl online als auch bei Addison-Wesley als Buch erhältlich,
- Die „HTML-Einführung“ von Hubert Partl (Universität für Bodenkunde, Wien) gibt es derzeit nur im Netz,
- „Spinning the Web. How to provide Information on the Internet“ von Andrew Ford (Thomson Publishing),
- mehr als 30 online verfügbare Dokumente zur Einführung ins WWW werden in der Yahoo-Datenbank (Verzeichnis: Computers and Internet/Internet/World Wide Web/Beginner's Guides) aufgelistet.

WWW-Seiten werden im HTML-Format (HyperText Markup Language) beschrieben. Die Bereitstellung von Informationen im WWW erfordert daher die Erstellung von Dateien in diesem Format. Prinzipiell kann dies auf 3 verschiedenen Wegen erreicht werden, was in den folgenden Unterkapiteln beschrieben wird.

A.1 Texteditoren

Im Gegensatz zu den Formaten vieler Textverarbeitungssysteme sind die Quelltexte der WWW-Seiten, d.h. die Dateien, welche die HTML-Beschreibungen enthalten, direkt mit einfachen Texteditoren bearbeitbar. Die HTML-Steueranweisungen sind in den ASCII-Text als sogenannte „Tags“ (in eckige Klammern eingeschlossene Zeichenfolgen) eingebettet. Die Erstellung von WWW-Seiten auf diesem Weg erfordert eine detailliertere Kenntnis der Syntax von HTML und ist daher fehleranfällig. Für die Erstellung von Fachberichten der LfU ist diese Alternative ungeeignet.

A.2 HTML-Editoren

HTML-Editoren sind spezielle Editoren, welche HTML direkt unterstützen. Man kann diese i.w. nach zwei Kriterien klassifizieren:

¹² An dieser Stelle werden daher nur einige Stichpunkte angerissen und Verweise auf die ausführlichere Dokumentation gegeben.

- Darstellung der HTML-Beschreibung:
 - ASCII-Text: Wie bei den normalen Texteditoren sieht der Benutzer die vollständige HTML-Beschreibung inkl. aller HTML-Tags. Über Menüs und Icons wird eine möglichst syntax-konforme Bearbeitung unterstützt.
 - WYSIWYG (What you see is what you get): Der Benutzer sieht die WWW-Seite in ihrer endgültigen Form und bearbeitet diese. Die HTML-Beschreibung wird vom System intern bearbeitet und vor dem Benutzer standardmäßig verborgen.
- Integration:
 - Eigenständiges System: Der HTML-Editor ist ein spezielles Produkt nur für die Bearbeitung von WWW-Seiten. Vorteil dieses Ansatzes ist, daß das Produkt auf die Eigenheiten von HTML optimal zugeschnitten werden und gleichzeitig kompakt bleiben kann.
 - Add-On: Ein (meist) Textverarbeitungssystem erhält zusätzliche Funktionen zur Bearbeitung von HTML-Seiten. Für den Benutzer ist hier von Vorteil, daß sich der Lernaufwand in Grenzen hält, da er sich in seiner gewohnten Umgebung bewegt.

Zur Erstellung der Fachberichte der LfU - solange diese auch in Papierform erscheinen sollen - sind HTML-Editoren zu empfehlen, die als Add-Ons zu Textverarbeitungssystemen dem Benutzer die Seiten in endgültiger Form (WYSIWYG) zeigen. Auf PC kommen dafür derzeit z.B. der Internet Assistant für Word für Windows und der Internet Publisher für WordPerfect in Frage.

Derzeit ist eine beinahe unüberschaubare Zahl von HTML-Editoren als Freeware, Shareware und kommerzielle Produkte verfügbar. Meist können diese direkt über das Netz auf den eigenen Rechner kopiert werden. Informationen über HTML-Editoren werden an folgenden Stellen angeboten:

- World Wide Web Consortium: <http://www.w3.org/hypertext/WWW/Tools>
- Yahoo: http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/HTML_Editors/
- Carl Davis HTML Editor Reviews: http://www.interaccess.com/users/cdavis/edit_rev.html
- Gabriel's HTML Editor List: <http://luff.ladrobe.edu.au/~medgjw/editors/index.html>
- Mag's Big List of HTML Editors: <http://union.ncsa.uiuc.edu/HyperNews/get/www/html/editors.html>

A.3 Konverter

Eine häufige Aufgabenstellung ist das Einbringen existierender Dokumente ins WWW. Wegen der Vielzahl an Textverarbeitungs-Programmen und deren diverse Versionen muß auch mit unterschiedlichsten Dateiformaten umgegangen werden. Zur Konvertierung der Dokumente kann wie folgt vorgegangen werden:

- Direkte Konvertierung nach HTML mit einem speziellen Konvertierungsprogramm oder
- importieren in einen HTML-Editor zur weiteren Bearbeitung.

Falls für ein bestimmtes Format weder ein Konvertierungsprogramm noch ein Importfilter für einen HTML-Editor vorhanden ist, kann das Dokument eventuell vorher in ein geeigneteres Format übergeführt werden (Import- und Export-Funktion eines Textverarbeitungssystems; allgemeines Konvertierungsprogramm wie z.B. Word für Word von Mastersoft, das allerdings sicher bald HTML auch direkt unterstützen wird).

Die vorhandenen Konverter sind meist relativ einfache Werkzeuge, die als Free- oder Shareware verfügbar sind. In der Regel unterstützen diese nur einen Teil der HTML-Features und auch nur eine kleine Teilmenge der Gestaltungsmöglichkeiten des Textverarbeitungssystems. Eine Nachbearbeitung der konvertierten Dokumente, z.B. eine Zerlegung in WWW-Seiten geeigneter Größe, die separate Konvertierung von Abbildungen und Tabellen und den manuellen Aufbau von Hyperlinks kann man selten vermeiden.

Steht man vor der Aufgabe, ein bestimmtes Dokument, das nicht mehr fortgeschrieben werden soll, einmal ins WWW einzubringen, so kann man sich einen geeigneten Konverter suchen, das Dokument konvertieren und das erhaltene HTML-Dokument solange weiterbearbeiten, bis die WWW-Seiten den Anforderungen genügen. Konverter findet man z.B. in folgenden Verzeichnissen:

- World Wide Web Consortium:
<http://www.w3.org/hypertext/WWW/Tools/Filters.html>
- Yahoo:
http://www.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/HTML_Converters/
- HyperNews: <http://union.ncsa.uiuc.edu/HyperNews/get/www/html/converters.html>

Dieser Ansatz ist jedoch nicht sinnvoll, wenn neue Dokumente erstellt oder Dokumente fortgeschrieben werden sollen, wie es bei der LfU der Fall ist. Hier ist anzustreben, daß das Originaldokument automatisch in ein Netz von WWW-Seiten umgesetzt werden kann. Dazu ist es erforderlich, die Gestaltungsmöglichkeit des Textverarbeitungs-Systems soweit einzuschränken und festzuschreiben, daß ein mit vertretbarem Aufwand realisierbares Konvertierungstool entwickelt werden kann, das eine vollständige Umsetzung erlaubt.