T U M

I N S T I T U T   F Ü R   I N F O R M A T I K

Systemarchitektur und Funktionalität des multimedialen digitalen Meta-Bibliothekssystems OMNIS/2

Günther Specht, Michael G. Bauer

TUM-I9924
November 99

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM-INFO-11-I9924-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

Systemarchitektur und Funktionalität des multimedialen digitalen Meta-Bibliothekssystems OMNIS/2

Günther Specht, Michael G. Bauer
{specht,bauermi}@in.tum.de

Inhaltsverzeichnis

1 Einleitung und Zielsetzung

Im Projekt "OMNIS/2: Integration von multimedialen Datenbanksystemen und Bibliotheksrecherchesystemen" des DFG-Schwerpunktprogramms (SPP) "Verteilte Verarbeitung und Vermittlung digitaler Dokumente" (V3D2)[1] wird ein Bibliothekssystem entwickelt, das beliebige, bestehende (multimediale) Bibliotheksrecherchesysteme in einer Metaschicht erweitert. Bestehende digitale Bibliothekssysteme enthalten oft große Dokumentenbestände und erlauben unterschiedlichste Retrievalmöglichkeiten von der Recherche in Katalog- und Metadaten bis hin zu komplexen Volltextqueries oder inhaltsbasierten Suchanfragen. Oftmals wird gewünscht, die gefundenen Dokumente wären wiederum automatisch verlinkt, im einfachsten Fall z.B. über die Literaturlisten in den Volltextdokumenten. Weiterhin können diese Systeme als reine Nachweisdatenbanken den Benutzern keine Möglichkeit bieten, darin zu schreiben, was nötig wäre, um von Retrievalsystemen zu interaktiven Bibliothekssystemen zu kommen, mit denen man direkt "arbeiten" kann. D.h. in denen man Bibliotheksdokumente entspechend den Benutzerinteressen verlinken kann, mit eigenen Annotationen versehen kann, personalisierte Benutzersichten aufbauen kann und in die man eigene (multimediale) Dokumente einfügen oder um bestehende erweitern kann.
Mit OMNIS/2 wird ein Bibliothekssystem entwickelt, das beliebige, bestehende (multimediale) Bibliotheksrecherchesysteme in einer Metaschicht um diese Funktionalitäten erweitert und sie dem Benutzer transparent zur Verfügung stellt. Dabei können mehrere Systeme gleichzeitig unter einer Oberfläche eingebunden werden. Andererseits stellt OMNIS/2 durch seine Fähigkeit eigene multimediale Dokumente erstellen, speichern und volltextindizieren zu können, aber auch ein eigenständiges System dar, in dem alle Fähigkeiten digitaler Bibliothekssysteme und multimedialer Datenbanksysteme integriert sind.
Dieser Bericht enthält die wesentlichen Ergebnisse der 1. Förderphase (1998-1999). Insgesamt ist das Projekt auf 6 Jahre angelegt (1998-2004).

1.1 Übersicht über die Projektarbeit

Die erste Projektphase (1998-1999) war gekennzeichnet von der Konzeption der Architektur des Gesamtsystems, einer Grob- und einer Feinspezifikation und der Entwicklung eines ersten Demonstrators. Dabei wurde die Architektur und Funktionalität des Gesamtsystems gegenüber den ersten Überlegungen deutlich erweitert, so daß auch externe Bibliotheksysteme transparent angebunden werden können. Die ursprüngliche Zielsetzung einer Integration eines Dokumenten-Retrieval-Systems und eines Multimedia-Datenbanksystems, namentlich der Systeme OMNIS und MultiMAP mit der Zielsetzung ein einheitliches Software-System zu erhalten, mit dem interaktive, multimediale Dokumente erstellt, gespeichert, gesucht und repräsentiert werden können, wurde so um wesentliche Punkte ausgedehnt. Darüberhinaus konnte in der ersten Projektphase eine erfolgreiche Kooperation mit dem SPP-Teilprojekt AVAnTA in Bremen (Prof. Herzog) durchgeführt werden.

1.2 Erweiterte Zielsetzung

Bereits während der Grobspezifikation zeigte sich, daß es oft wünschenswert ist, zusätzlich andere bestehende Bibliothekssysteme (seien sie katalogbasiert, volltextbasiert oder multimedial) transparent einbinden zu können. Die Suchanfragen können dann an diese

[1] http://www.cg.cs.tu-bs.de/v3d2/

intern weitergereicht werden und die Ergebnisse genauso wie lokale Dokumente der eigenen Dokumentenspeicherschicht weiterbehandelt und verknüpft werden. Damit erhält man zwei wesentliche Erweiterungen gegenüber der ursprünglichen Zielsetzung. Zum einen wird dadurch die Funktionalität und Mächtigkeit des OMNIS/2-Systems wesentlich ausgebaut, zum anderen kann damit OMNIS/2 auch als Metasystem über beliebige andere Bibliotheksrecherchesysteme gesetzt werden, wodurch deren Dokumente - auch und gerade wenn die Systeme selbst diese Möglichkeiten nicht bieten -, durch die OMNIS/2-Schicht untereinander verlinkt, annotiert und multimedial erweitert werden können.
Dies bedeutet eine deutliche Flexibilität und Verallgemeinerung gegenüber den ursprünglich genannten Zielen nur OMNIS und MultiMAP zu integrieren. Erste Versuche mit dieser erweiterten Zielsetzung waren sehr vielversprechend und fanden auch großes Interesse bei anderen Teilprojekten im SPP, wie z.B. AVAnTA Uni Bremen (Videobibliothekssystem) und Paddle FAW Ulm (Umweltinformationssystem).

1.3 Gliederung des Berichts

Im folgenden werden die einzelnen Arbeitsschritte und Ergebnisse genauer beschrieben. Zuerst wird in Abschnitt 2 die erarbeitete Spezifikation von OMNIS/2 dargestellt. Anschließend gehen wir in Abschnitt 3 auf die Architektur genauer ein. Es folgen Abschnitte über den erstellten Demonstrator (4) und die Feinspezifikation (5). Ein Abschnitt (5.3) beschäftigt sich mit den Forschungsergebnissen bzgl. den Integritätsmechanismen bei der Speicherung von XML-Dokumenten. Der Bericht endet mit den Ergebnissen zur Personalisierung (6) und zur Kooperation (7) mit dem Projekt AVAnTA.

2 Funktionalität von OMNIS/2

Im Teilprojekt wurde eine umfangreiche Spezifikation des OMNIS/2-Systems unter Beachtung der erweiterten Zielsetzung erarbeitet. Diese zusammenfassend, läßt sich das System in seiner Funktionalität wie folgt beschreiben:

OMNIS/2 soll die Fähigkeiten von OMNIS, einem digitalen Bibliothekssystem, und MultiMAP, einem multimedialen Datenbanksystem, integrieren, so daß ein einheitliches Software-System entsteht, mit dem interaktiv multimediale Dokumente erstellt, gespeichert, gesucht und über das WWW repräsentiert werden können. Die einzelnen Dokumente, bestehend aus Texten, Bildern, Audio, Video und Tabellen, etc. sollen nicht nur über typische Bibliothekssystemanfragen wie Recherche nach Katalogdaten (Autor, Titel, Stichworte, etc.) und Volltextsuche untereinander recherchierbar sein, sondern darüberhinaus, mittels eines wesentlich mächtigeren Linkkonzepts als das jetzige WWW, auch beliebig verlinkbar sein, ohne dabei für die Links in die Originaldokumente hineinschreiben zu müssen. Damit ist OMNIS/2 als selbständiges Bibliothekssystem nutzbar und kann alle Daten aus den bisherigen OMNIS Systemen und deren Erweiterungen (OMNIS als Volltext-Bibliothekssystem der Fakultätsbibliothek Mathematik und Informatik der TU München; Elektra, einem elektronischen Aufsatzdienst; oder VD17, einem Volltextverzeichnis aller Drucke des 17. Jahrhunderts) übernehmen und multimedial sowie hypermedial ergänzen.
Zusätzlich soll OMNIS/2 die Möglichkeit bieten, bestehende katalogbasierte, volltextbasierte oder multimediale Bibliothekssysteme transparent einzubinden. D.h. über OMNIS/2

kann in diesen Systemen recherchiert werden und - auch wenn die Systeme selbst diese Möglichkeiten nicht bieten -, deren Dokumente verlinkt, annotiert, multimedial erweitert und personalisiert werden. Die Dokumente verbleiben dabei in den Originalbibliothekssystemen. Sie werden in der OMNIS/2-Datenbank nur durch Ihre Adresse und Metadaten repräsentiert. Die Verlinkung, einschließlich der Linkankerpositionen wird ausschließlich in OMNIS/2 gespeichert und zur Präsentationszeit dynamisch den Ergebnisdokumenten in der Anzeige hinzugefügt. Außerdem können Bibliotheksdokumente mittels OMNIS/2 mit persönlichen, gruppenbezogenen oder allgemeinen Annotationen versehen werden. Diese können auch multimedial und rekursiv (Annotationen auf Annotationen) sein. Für OMNIS/2 ist auch eine Personalisierungskomponente spezifiziert worden, die zur besseren Filterung entsprechend persönlichen Interessen und zum persönlichen Ranking der Ergebnisse der eingebundenen Bibliothekssysteme verwendet werden kann. Anschlüsse zum Ausbau zu einem späteren Benachrichtigungssystem bei relevanten Neueingängen (Push- und Pull-Strategie) sind vorgesehen. Sowohl die Annotationen als auch die Personalisierung werden ausschließlich in der OMNIS/2-Metadatenbank durchgeführt, ohne Änderungen oder Ergänzungen in den Orginalbibliotheksystemen, die als Recherchesysteme ja gerade keine schreibenden Einträge der Benutzer zulassen können.
OMNIS/2 bietet hier die Möglichkeit aus bestehenden Bibliotheksrecherchesystemen persönliche oder gruppenbezogene transparente Arbeitsumgebungen zu machen mit der Möglichkeit eigene Dokumente, Verlinkungen und Annotationen transparent in das System einbringen zu können.
Es können zweierlei Arten an Einbindung unterschieden werden, die aber gleichzeitig zum Einsatz kommen können: Eine enge Integration und eine Kopplung über XML. Bei der engen Integration werden die eingebundenen Systeme über interne Schnittstellen direkt in OMNIS/2 eingebunden. Bei der XML-Kopplung brauchen die Systeme nur um eine XML-Ausgabeschnittstelle erweitert zu werden. OMNIS/2 verwaltet dann die gemeinsame DTD (Data Type Definitions) und nutzt zusätzlich die getaggten Informationen zur Metadatenverwaltung.

3 Architekturfestlegung von OMNIS/2

Im Anschluß an die Grobspezifikation wurde die Architektur des Systems festgelegt. Dabei bieten sich eine Reihe an Freiheitsgraden, die untersucht werden mußten: die WWW-Anbindungstechniken, die konzeptionelle und die technische Schichtenarchitektur.

Wahl der WWW-Anschlußtechnik:
Die Wahl der WWW-Anschlußtechnik hat einen großen Einfluß auf die weitere Implementation. Während das ursprüngliche OMNIS eine CGI-Anbindung über Server-parsed-HTML enthält, war MultiMAP mittels Java-Applets und einem DB-Client zur Transaktionssynchronisation ans Web angebunden. Während ersteres für interaktive multimediale Repräsentationen zu wenig mächtig ist, erzeugt letzteres eine zu hohe Last auf dem Client. Daher wurde diese Frage entsprechend dem aktuellen Stand der Technik für OMNIS/2 neu aufgerollt. Einen Überblick bietet Abbildung 1. Die folgenden verschiedenen Möglichkeiten wurden im Einzelnen im Rahmen der Grobspezifikation geprüft:
Datenbankanbindungen ans WWW können in clientseitige und serverseitige Anbindungen unterschieden werden. Bei einer clientseitigen Anbindung kann über ein Java-Applet eine

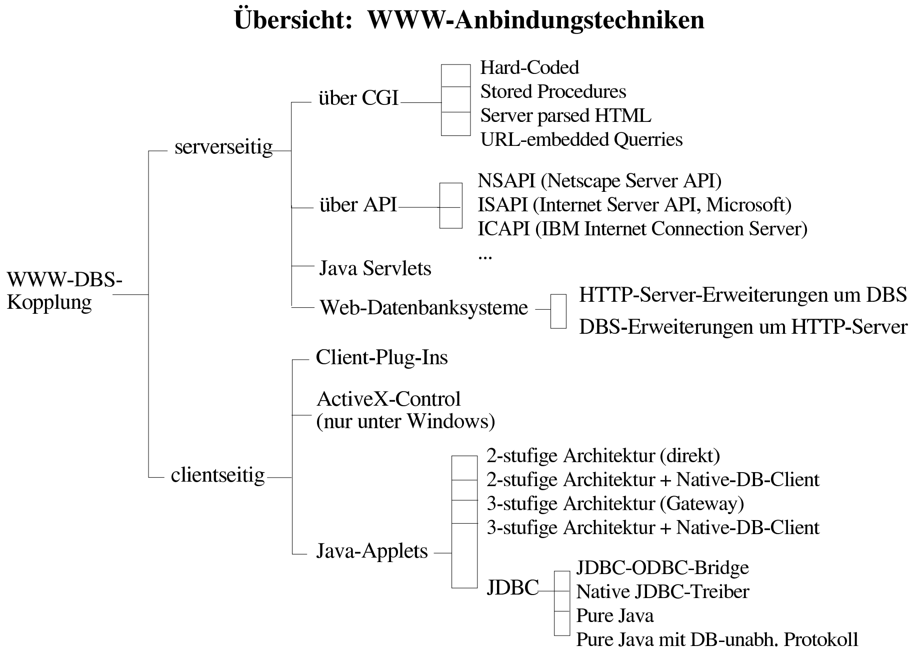

Abbildung 1: WWW-Anbindungstechniken an Datenbank- und Bibliothekssysteme

direkte Verbindung (außerhalb der HTTP-Verbindung) zu einem Datenbankserver aufgebaut werden. Hinderlich ist hierbei, daß herkömmliche Java-Applets aus Sicherheitsgründen nur zu dem Server direkte Verbindung aufnehmen können von dem sie geladen wurden. Dies erfordert, daß sich der Datenbankserver auf der gleichen Maschine wie der Webserver befinden muß. Zwar wurde diese Einschränkung durch den Einsatz von sog. "trusted" Applets (d.h. signierten und damit vertrauenswürdigen Applets) aufgehoben, allerdings zeigte sich in beiden Fällen der Nachteil, daß das Applet dann eine immense Programmgröße erreichen würde, da die Funktionalität eines Datenbankclients nachgebildet werden müßte. Weiterhin wurde eine Architektur in Betracht gezogen, bei der auf dem Webserver nur ein Gateway läuft, das die eigentliche Anbindung an die Datenbank übernimmt. Man könnte mit einer Architektur dieser Art zwar den Webserver und den Datenbankserver trennen, hat aber den Nachteil, daß die Clients immer noch groß sind, da die Anbindung an das Gateway realisiert werden muß.
Bei einer serverseitigen Anbindung an die Datenbank gibt es wiederum verschiedene Möglichkeiten der Realisierung. Eine weitverbreitete Technik ist die Nutzung von CGI (Common Gateway Interface). Dabei wird durch den Aufruf einer URL ein Programm auf dem WWW-Server gestartet, das dann über einen ständig eingeloggten Datenbankclient Verbindung zur Datenbank aufnehmen kann. Da aber für jedes aufgerufene Programm ein neuer Prozeß gestartet wird, kann dies bei größerer Belastung zu Performanzproblemen führen. Ein anderer Ansatz ist die Nutzung einer entsprechenden API des Webservers, über die eine Anbindung an eine Datenbank realisiert werden kann. Der Nachteil vieler dieser APIs ist, daß sie oft spezifisch für einen einzelnen Webserver angeboten werden.
Für die Realisierung von OMNIS/2 haben wir uns für Java-Servlets entschieden. Dadurch

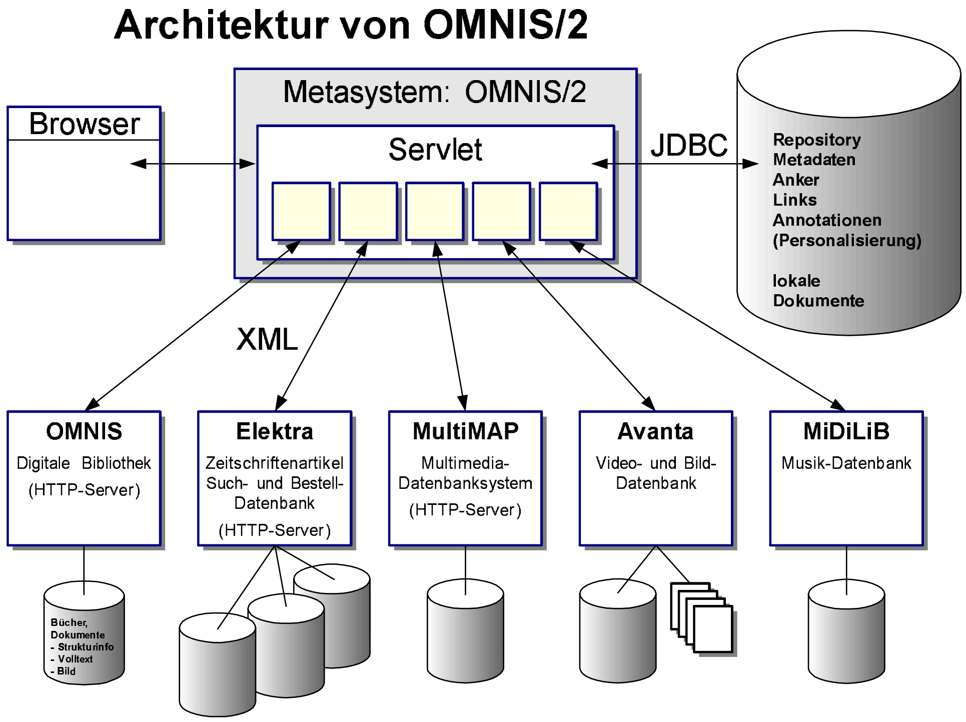

Abbildung 2: Architektur von OMNIS/2

können die Vorteile der Realisierung in Java (große Mächtigkeit, Plattformunabhängigkeit, relativ schnelle Implementierung) mit den Vorteilen einer serverseitigen Anbindung (Unterstützung auch von Thin-Clients) kombiniert werden. Für interaktive, multimediale Anzeigen werden im Browser zusätzlich zu HTML Java-Applets benötigt. Diese können jedoch jetzt im Gegensatz zur MultiMAP-Implementierung deutlich schlanker ausfallen. Auch die Kommunikation wird einfacher, da die Java-Servlets und Applets am HTTP-Protokoll vorbei, direkt miteinander kommunizieren können, ohne jeweils neue Prozesse (wie beim CGI-Aufruf) zu erzeugen. Java-Code kann damit direkt nachgeladen und Java-Objekte direkt übergeben werden.

Die konzeptionelle Architektur (vertikale Schichtung):
Im folgenden wird die konzeptionelle Architektur von OMNIS/2 näher betrachtet. Sie ist in Abbildung 2 vertikal dargestellt. Eine wichtige Randbedingung des OMNIS/2-Systems ist, nicht in die darunterliegenden, eingebundenen Bibliothekssysteme und Dokumente hineinzuschreiben, auch nicht zur Kennzeichnung der Links, respektive Anker oder der Annotationsstellen. Dementsprechend wurde eine Drei-Schichten-Architektur analog dem Dexter-Modell bzw. dem Amsterdam-Modell gewählt. Die eingebundenen Bibliothekssysteme entsprechen dem Within-Component-Layer, also der Dokumentenschicht. In Abbildung 2 sind beispielhaft einige dafür eingetragen. Sie können im Gegensatz zum Amsterdam-Modell selbst wiederum entfernt und verteilt liegen. Die gesamte Komposition, Verlinkung und Annotation findet in der OMNIS/2-Metaschicht statt, die selbst wiederum aus einem Applikationserver (implementiert in Java-Servlets) und einer JDBC-Kopplung zur OMNIS/2-Datenbank besteht. Diese Schicht entspricht damit dem Storage-Layer des Dexter- und des Amsterdam-Modells. Die dritte Schicht ist die Repräsentationsschicht im Browser.

Die technische Architektur (horizontale Schichtung):
In der mehr technischen Sichtweise der OMNIS/2-Architektur, in der Abbildung 2 oben horizontal dargestellt, folgen wir bei der Realisierung der "three tier architecture" zur Anbindung von Datenbanken ans Web. Wir verwenden dabei eine serverseitige Anbindung der OMNIS/2-Datenbank unter Verwendung von Servlets und einer JDBC-Kopplung. Dabei wird ein HTTP-Server mit Hilfe der Servlets zu einem vollständigen Applikationsserver, also dem vollständigen OMNIS/2-Kernsystem, ausgebaut. Browser und HTTP-Server kommunizieren über einen Standard-HTTP-Port, bzw. über den direkten Austausch von Java-Klassen, sofern Applets zur Anzeige nötig sind. Vom Browser kommende Benutzeranfragen werden im Servlet verarbeitet, das für den Zugriff auf die OMNIS/2-Metadatenbank eine JDBC-Kopplung zum relationalen Datenbanksystem TransBase offen hält. Bei der Suche in einer der angeschlossenen, externen Bibliotheken muß die Suchanfrage in das entsprechende URL-Format übersetzt und versandt werden. Externe Bibliotheken können verteilt und entfernt liegen, so daß die Anfrage hier ein zweites Mal über das Internet geht. Bei einer Linkverfolgung in eine angebundene Bibliothek muß aus den Informationen in der OMNIS/2-Metadatenbank erst die entsprechende Query im entsprechenden URL-Format des Zielsystems vom Servlet generiert werden. Die XML-Antwort wird mittels eines XML-Parsers geparst und um die Zusatzinformationen aus der OMNIS/2-Metadatenbank (Verlinkung, Annotationen etc.) für die Präsentation ergänzt. Dazu wird entsprechend den Ergebnisdokumenten eine zweite Anfrage an die OMNIS/2-Metadatenbank gestellt. Diese ist immer nötig. Abschließend wird das gesamte Dokument für die Ausgabe in HTML aufgebaut. Erste Versuche mit XML-Browsern (so daß die Umwandlung nach HTML überflüssig wäre) wurden unternommen, ergaben aber für die multimedialen, interaktiven Teile, insbesondere der Autorenkomponente, noch keine zufriedenstellenden Ergebnisse.

4 Entwicklung eines Demonstrators

Nach der Grobspezifikation und der Architekturfestlegung wurde ein erster kleiner Demonstrator entwickelt, um die praktische Einsatzfähigkeit der Konzepte zu validieren. Um auch die externe XML-Kopplung zu testen, banden wir darin das Fakultätsbibliothekssystem der Informatik der TU München mit 800 000 Dokumenten als Legacy-Anwendung an. Um auf den Datenbestand dieses OMNIS-Systems der ersten Generation zugreifen zu können, mußte darin unterhalb der Präsentationschicht eine XML-Ausgabe integriert werden. Diese einzige notwendige Erweiterung im Legacy-System war überraschend schnell und leicht möglich.
Der Aufstieg von XML parallel zur Projektlaufzeit von OMNIS/2 wirkte sich sehr positiv auf das Projekt aus. Insbesondere entstanden in der Zwischenzeit eine Reihe leistungsfähiger Werkzeuge, die wir nutzen konnten. Der in OMNIS/2 eingesetzte XML-Parser *xml4j* von IBM Alphaworks verkürzte die Entwicklungszeit ganz erheblich und ermöglicht eine schnelle und effziente Aufbereitung der Ergebnisse des angebundenen Bibliothekssystems der TU München.
Die von OMNIS gelieferten und vom Servlet geparsten XML-Daten werden im Servlet um die Ankerinformation ergänzt und für die Darstellung in handelsüblichen Browsern von XML in HTML übersetzt.
Als Webserver, der die Servlet-API anbindet, kam der bekannte *Apache* Webserver zum Einsatz. Die Vorteile dieses Webservers liegen in der hohen Stabilität, der relativ leichten

Erweiterungsmöglichkeiten, da der Quellcode offen liegt und der guten Dokumentation. Die eigentliche Umsetzung der Servlet-API Spezifikation im Webserver erfolgt durch die Erweiterung *Apache Jserv*, die die Spezifikation durch eine Java Servlet Engine implementiert und in den Server einkompiliert wird.
Mit OMNIS haben wir damit ein typisches textbasiertes Bibliothekssysteme eingebunden. Mit der Anbindung an MultiMAP, die als nächstes angegangen wird, erhalten wir zusätzlich alle Fragestellungen der Recherche und Einbindung allgemeiner Multimedia-Datenbanken, die insbesondere selbst bereits eine fortschrittliches Linkkonzept beinhalten. Im nächsten Förderzeitraum soll dann unser System um Einbindungsmöglichkeiten unterschiedlicher medialer Bibliothekssysteme aus dem SPP ausgebaut werden (AVAnTA als Video- und Bilddatenbank (Uni Bremen), MiDiLiB als Musikbibliothek (Uni Bonn), sowie evtl. Paddle als geographische Umweltbibliothek (FAW Ulm)). Geplant ist weiterhin die Einbindung eines OPAC-Systems um auch diese Bibliothekskatalogsysteme bei Bedarf einbinden zu können.

5 Feinspezifikation

5.1 Erstellung eines E/R-Schemas und Überführen in ein Datenbankschema

Die Erstellung des E/R-Diagramms, das dann in das relationale Schema überführt wurde, stellt einen kritischen Kernpunkt in der OMNIS/2-Entwicklung da und nahm einige Zeit und Iterationsstufen in Anspruch. Das Datenbankschema einschließlich seiner Indexstruktur entscheidet über die möglichen Zugriffsstrukturen und deren Effizienz. Hauptziel war, daß insbesondere die Anfragen bzgl. der Linkverfolgung und der Volltextsuche effizient bearbeitet werden können. Das Ergebnis ist in Abbildung 3 dargestellt. Die unterstrichenen Attribute im E/R-Diagramm bezeichnen wie üblich die Schlüsselattribute, wobei wir für die SQL-Query-Erstellung davon ausgehen, daß darauf ein B-Baum als Indexstruktur liegt. Sowohl das E/R-Diagramm als auch das Relationenschema gliedern sich hierbei in verschiedene Bereiche. Die Bereiche umfassen die Komponenten, die Verlinkung, die Volltextsuche und die Benutzerverwaltung.
Als Komponente wird in diesem Zusammenhang ein speicherbares Hypermedia-Objekt bezeichnet. Zur Verlinkung ist anzumerken, daß in der Modellierung n:m-Links möglich sind. Sie werden jedoch aus Effizienzgründen in n$*$m 1:1-Links gespeichert und sind durch eine gemeinsame link_UID gekennzeichnet. Die Volltextsuche erstreckt sich auf alle Volltexte, alle Namen und die Inhalte aller in der Metadatenbank gespeicherten Dokumente. Mit Hilfe der Benutzerverwaltung wird einerseits eine vereinfachte Zugangskontrolle mittels Namen und Paßwort realisiert, andererseits wird ermöglicht, daß ein Benutzer verschiedene Attribute der Präsentation global beeinflussen kann und daß damit auch eine einheitliche Präsentation für gleichartige Medientypen möglich wird. Dieses Konzept wird in Anlehnung an das Amsterdam Hypermedia-Konzept als *Channel*-Konzept bezeichnet.
Bei der Demonstratorentwicklung war u.A. die Realisierung einer effizienten Linkverfolgung ein wichtiger Forschungspunkt. Links sind eigene Entitäten und genügen damit, sofern es sich nicht um externe Links handelt, der referenziellen Integrität. n:m-Links, bidirektionale Links, temporale Links und sich überschneidende Links werden unterstützt. Das Dexter-

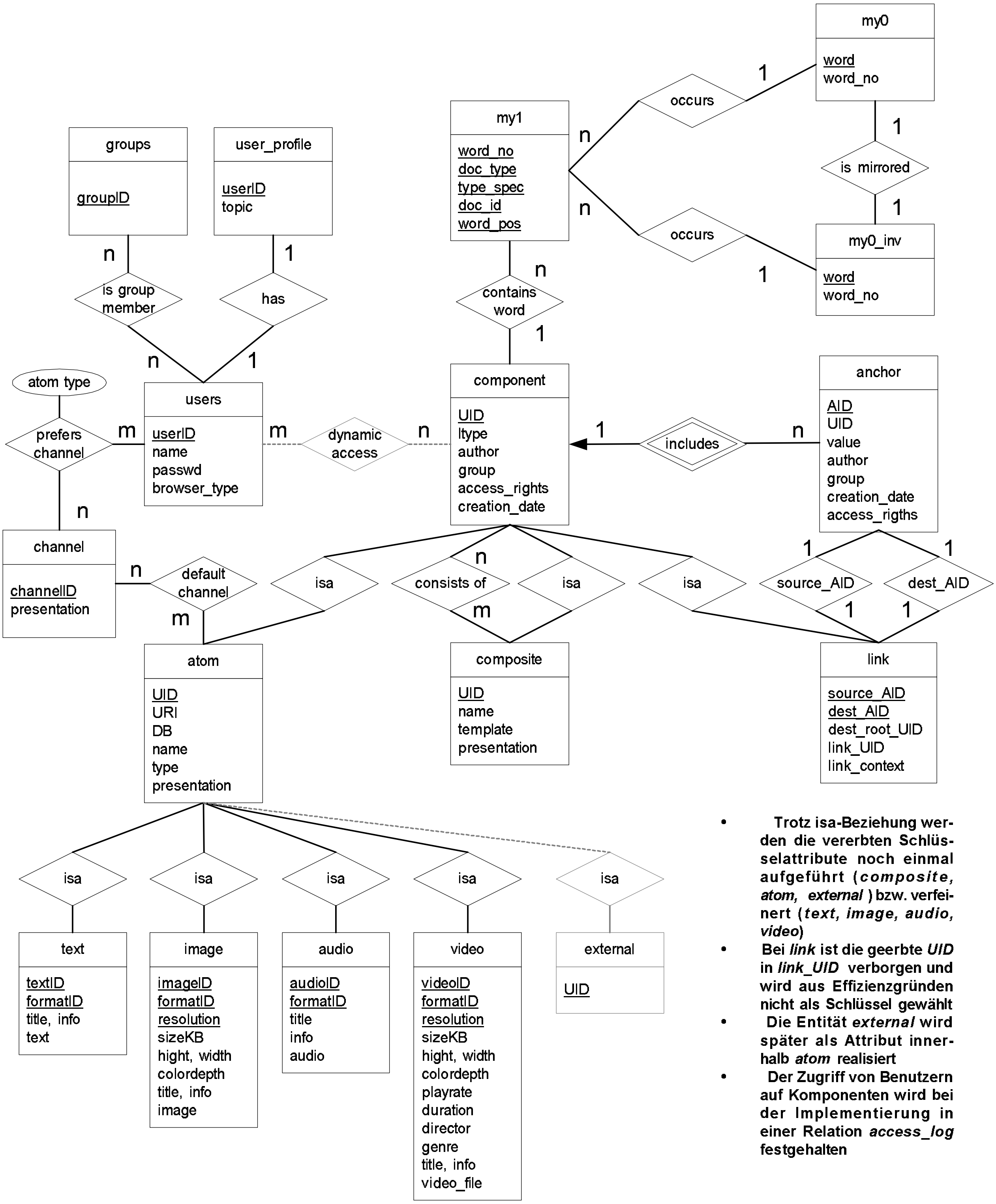

Abbildung 3: E/R Diagramm von OMNIS/2

und das Amsterdam-Modell sind Referenzmodelle im Sinn von Postulaten. Während dem Fortgang des Projekts wurden diese Modelle auf relationale und objektorientierte Datenbanksysteme abgebildet und dafür Komplexitätsuntersuchungen für den Zeitaufwand bei der Linkverfolgung und eine Reihe an Optimierungen durchgeführt. Insgesamt konnte die Gesamtkomplexität der Linkverfolgung von 3 quadratischen Summanden und 5 linearen Summanden auf $O(\log(n) + n \cdot \log(m))$ wobei $n = |\text{links}|$ und $m = |\text{anchors}|$ gedrückt werden [4]. Damit wird dieser Ansatz effizient operabel.

5.2 Erstellung einer Anfragekomponente

Da die Verlinkung der Dokumente nachträglich über die Metadatenbank in das System eingebracht wird, d.h. der eigentliche Datenbestand der anzubindenden Bibliotheken (OMNIS, etc.) bereits vorhanden ist, erfordert dies ein Werkzeug mit dem diese Verlinkung sowohl erstellt als auch dargestellt werden kann. Zu diesem Zweck wurde eine erweiterte Anfragekomponente spezifiziert und bereits teilweise implementiert.
Die Anfragekomponente stellt dem Benutzer die graphische WWW-Oberfläche zur Benutzung von OMNIS/2 zur Verfügung. Für die Darstellung werden die entsprechenden Seiten im Servlet mit Daten aus der Metadatenbank angereichert, da Links von dem eigentlichen Daten getrennt sind. Technisch kommt dabei eine Art Server-Parsed HTML zum Einsatz. Die Templates für die darzustellenden Seiten werden auf dem Webserver vorgehalten. Falls notwendig werden die Seiten zusätzlich durch Java-Applets ergänzt, um die notwendige Funktionalität zu erhalten, falls HTML dazu nicht ausreicht. Die Spezifikation der Eigenschaften der Anfragekomponente umfaßt:

- Volltextsuche
- Erweiterte Volltextsuche kombiniert mit Benutzerinteressen
- Linkverfolgung
- Kommentarerstellung
- Linkerstellung und -bearbeitung
- Dokumenterstellung und -bearbeitung
- Erstellung und Bearbeitung der Benutzerinteressen
- persönliche Channel-Einstellungen
- Hilfeangebot

Zur schnellen Erstellung der benötigten HTML-Masken konnten ausgereifte Editoren eingesetzt werden, in unserem speziellen Fall FrontPage 3.0 der Firma Microsoft.
Die Anfragekomponente arbeitet über das Servlet stark mit der Metadatenbank zusammen. Die vollständige Entwicklung der Anfragekomponente ist im aktuellen Stand des Projekts noch nicht abgeschlossen und soll im Rahmen eines umfassenden Autorensystems in der zweiten Projektphase fortgesetzt werden.

5.3 Integritätssicherung bei der Speicherung von XML-Multimedia-Dokumenten in OMNIS/2

5.3.1 Motivation

In OMNIS/2 soll XML als Austauschformat für die Schnittstelle zu allen existierenden Bibliothekssystemen genutzt werden. Die Schnittstelle zum TU-Bibliothekssystem OMNIS ist bereits operabel und dient als Referenzanbindung. Als Vorteil gegenüber einer Verwendung von HTML als Austauschformat können über XML Strukturinformationen mitübertragen werden und so als Metadaten genutzt werden. Außerdem wird XML in OMNIS/2 bei der Eingabe und Speicherung benutzerdefinierter Komponenten für die Template- und Composite-Verwaltung verwendet.
Über die Autorenkomponente kann eine interne Änderung von XML-Strukturen erfolgen, wie etwa bei der Modifikation einer Subkomponente innerhalb eines Composites. Deshalb ist es notwendig, zur Konsistenzerhaltung bei jeder Änderung Integritätsbedingungen zu überwachen.
Die effiziente Repräsentation und Überwachung der Integritätsbedingungen bei der Speicherung von XML-Multimedia-Dokumenten in Datenbanksystemen ist ein schwieriges und in der Literatur noch nicht gelöstes Problem.

Im folgenden werden zunächst grundsätzliche Speicherungsformen bei der Verwaltung von XML-Dokumenten in relationalen Datenbanksystemen (RDBS) untersucht (5.3.2). Danach wird das Problem der Integritätssicherung bei der Speicherung von XML behandelt (5.3.3). Hier werden zunächst Möglichkeiten der Integritätssicherung von XML-Dokumenten unabhängig von der Speicherungsform der Dokumente beschrieben. Anschließend werden die Möglichkeiten der Integritätssicherung bei der Speicherung in RDBS hinsichtlich bestimmter Speicherungsformen für XML in RDBS betrachtet.

5.3.2 Grundsätzliche Möglichkeiten bei der Speicherung in RDBS

Grundsätzlich gibt es folgende Möglichkeiten ein XML-Dokument in einem RDBS zu speichern:

1. Speicherung als Ganzes in einem Attribut einer Relation ohne weitere Aufspaltung:
   - Speicherung in einem *String*
   - Speicherung in einem *BLOB*, bzw. *CLOB*
2. Aufspaltung des Dokuments auf mehrere Relationen:
   - *Meta-Relationen*: alle Dokumente werden in denselben Relationen abgelegt, z.B. in den Relationen *element* und *attribute*. Bei dieser Speicherungsform gibt es einen festen Satz von Relationen, in denen alle XML-Dokumente gespeichert werden, egal ob eine DTD für sie existiert und von welchem Dokumententyp sie sind.

     Es wurde ein E/R-Diagramm für XML-Dokumente erstellt, in dem alle üblichen Deklarationsarten berücksichtigt werden, die innerhalb eines XML-Dokuments vorkommen können. Daraus wurden zu Testzwecken die entsprechenden Relationen erzeugt.

- *Spezielle Relationen*: für jeden Dokumenttyp gibt es spezielle Relationen, z.B. eine Relation pro Elementtyp.

  Diese Speicherungsform basiert auf folgender grundsätzlicher Abbildung zwischen einem XML-Dokument und dem E/R-Modell:

  – XML-Element → Entity
  – XML-Attribut → Attribute
  – Verschachtelung der XML-Elemente → Relationships

Bei der Untersuchung der Speicherungsformen zeigte sich, daß die Frage nach der bestmöglichsten Speicherungsform nur abhängig von den Anforderungen der jeweiligen Anwendung zu beantworten ist. Eine Bewertung führte zu folgenden Ergebnissen:

Speicherung in einem String oder einem BLOB/CLOB. Die Speicherung als String ist aufgrund der allgemeinen Längenbeschränkung für Strings in DBSen ungeeignet. Die Speicherung als BLOB/CLOB ist nur für Anwendungen geeignet, die das DBS eher als Dokumentenarchivierung verwenden und für die Zugriffe und Änderungen auf Teilobjekte und Integritätssicherung eine geringe Bedeutung besitzen.

Speicherung in Meta-Relationen. Die Speicherung in Meta-Relationen ist eher für Anwendungen geeignet, die eine große Menge von XML-Dokumenten unbekannten Typs (i.e. ohne DTD) speichern müssen oder XML-Dokumente mit stark unterschiedlichen DTDs. Dabei liegt der Schwerpunkt eher auf einer effizienten Zugriffs- und Ablagemöglichkeit für XML-Dokumente, als auf deren Integritätssicherung.

Speicherung in speziellen Relationen. Die Speicherung in speziellen Relationen ist gut geeignet für Anwendungen, die XML-Dokumente speichern sollen, deren Dokumenttyp über eine DTD bekannt ist. Hier kann die Speicherung optimiert werden und weitgehend Integritätssicherung auf Datenbankebene durchgeführt werden.

5.3.3 Integritätssicherung bei der Speicherung von XML

Bei der Frage der Integritätssicherung bei der Speicherung von XML-Dokumenten muß man zwei grundsätzliche Formen der Formulierung und Realisierung von Integritätsbedingungen unterscheiden:

1. Die Integritätssicherung erfolgt unabhängig von der Speicherungsform der Dokumente.

2. Die Integritätssicherung erfolgt nach den Möglichkeiten, die in einer bestimmten Speicherungsform gegeben sind.

Integritätssicherung durch XML-Parser: Eine Reihe von Integritätsbedingungen kann innerhalb einer DTD formuliert werden und durch einen XML-Parser überprüft werden. Dabei werden folgende Integritätsbedingungen unterschieden:

1. *Wohlgeformtheits-Integritätsbedingungen* werden von einem *wohlgeformten Dokument* nach der XML-Spezifikation erfüllt.

2. *DTD-Integritätsbedingungen* werden von einem *validen Dokument* nach der XML-Spezifikation bezüglich einer bestimmten DTD erfüllt.

Alternativen zu DTDs werden als notwendig betrachtet, da über DTDs viele Integritätsbedingungen nicht ausdrückbar sind, wie z.B. genaue Datentypen für Attribute und Elemente, komplexere Inhaltsmodelle, Wiederverwendung von Elementdefinitionen.
Die XML Schema Working Group hat seit 1998 vier verschiedene Vorschläge für Sprachen zur Diskussion gestellt, die sogenannte *XML-Schemas* beschreiben können. Innerhalb des Projekts wurde der erste Vorschlag des W3C *XML-Data* untersucht, inwieweit dieser zur Integritätssicherung der XML-Templates in OMNIS/2 verwendet werden könnte.

Integritätssicherung bei der Speicherung von XML in RDBS Weiterhin wurden die verschiedenen Speicherungsformen für XML-Dokumente in RDBS hinsichtlich ihrer Eignung zur Integritätssicherung untersucht. Als wichtiges Ziel wurde erkannt, daß verhindert werden sollte bei Update-Operationen das Dokument zur Integritätssicherung vollständig aus der DB zu holen und es evtl. aus mehreren Teilen zusammensetzen zu müssen. Falls alle Integritätsbedingungen durch das DBS gesichert werden können, ist eine erneute Zusammensetzung erst beim Retrieval des Dokuments durch Benutzer, bzw. Applikationen notwendig.
Für die verschiedenen Speicherungsformen in RDBS zeigten sich folgende Ergebnisse:

- Speicherung als Ganzes (String oder BLOB/CLOB)

  Die Integritätssicherung muß hier vollständig auf Anwendungsseite erfolgen und damit außerhalb der Datenbank.

- Aufspaltung in verschiedene Relationen

  - Speicherung in Meta-Relationen

    Die inhaltliche Integritätssicherung erfolgt hier vorwiegend auf der Anwendungsseite. Die strukturelle Integritätssicherung kann aber zum großen Teil auf *check*-Klauseln in SQL abgebildet werden und damit innerhalb der Datenbank laufen.

  - Speicherung in speziellen Relationen

    Die Integritätssicherung kann hier i.a. vollständig durch Zugriffe auf die Dokumententeile in der Datenbank erfolgen. Damit kann bei Updates die Extraktion des Gesamtdokuments zur Integritätssicherung vermieden werden.

Im Projekt wurden für die Speicherung der XML-Dokumente eine Reihe an teils sehr komplexen Integritätsbedingungen aufgestellt, in datenbankseitig und anwendungsseitig zu überprüfende Integritätsbedingungen eingeteilt und dafür jeweils verschiedene Implementationsvorschläge erstellt. Die datenbankseitig zu überprüfenden Integritätsbedingungen wurden soweit möglich in SQL-*check*-Klauseln implementiert, die anwendungsseitigen in Java. Dabei zeigte sich, daß ein Teil dieser Integritätsbedingungen nicht effizient genug online verarbeitet werden kann. Daraus aufbauende Optimierungen und effizientere Lösungen haben wir uns als Teil der Forschungsfragestellungen für die zweite Projektphase vorgenommen.

6 Vorarbeiten zur Personalisierung

Das Ziel der Personalisierung ist es, Daten, die für den Benutzer besonders relevant sind, aus einem großen Informationsvolumen auszuwählen und diese sortiert nach ihrer Wichtigkeit oder Relevanz dem Benutzer anzuzeigen. Die Hauptbestandteile eines Personalisierungssystems sind dazu Benutzer- und Objektprofile, das Benutzer-Feedback und die Informationsfilterung (vgl. Abbildung 4).

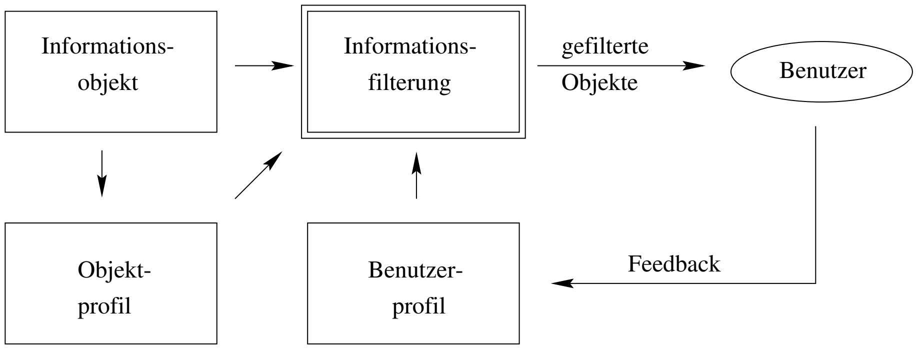

Abbildung 4: Grundstruktur eines adaptiven Personalisierungsverfahrens

Das von uns entwickelte GRAS-Verfahren (Gaussian Rating Adaption Scheme) ist für die Personalisierung von Hypermedia-Dokumenten konzipiert und beschränkt sich nicht auf Textdokumente. GRAS verwendet Benutzer- und Objektprofile. Jedes Profil besteht aus *mehreren* Themenprofilen. Jedes Themenprofil beschreibt das Interesse eines Benutzers oder den Inhalt eines Objektes in einem speziellen Themenbereich. Einfache Rating-Schemata verwenden zur Beschreibung des Benutzerinteresses nur einen Bewertungswert zu einem Thema. Dies ist aber in vielen Fällen nicht adäquat.
In GRAS haben wir ein Verfahren entwickelt, das für die Darstellung der Themenprofile Gaußkurven verwendet. Eine Gauß-Kurve wird durch zwei Parameter μ und σ charakterisiert. Mit einem Wert beschreiben wir den Fokus oder Mittelwert (μ) des Benutzerinteresses auf einer Skala von "hat hohes Interesse an diesem Thema" bis "mag dieses Thema überhaupt nicht". Der zweite Wert drückt die Breite (σ) des Benutzerinteresses aus. Abbildung 5a zeigt zwei Gaußkurven mit verschiedenen μ- und σ-Werten. μ verschiebt die Glockenkurve in der x-Achse. σ macht sie schlanker (niedriger σ-Wert) oder breiter.
Die gestrichelte Kurve kann als Interessenprofil einer sehr stark interessierten Person interpretiert werden. Ein hoher positiver μ-Wert, kombiniert mit einem niedrigen σ-Wert steht

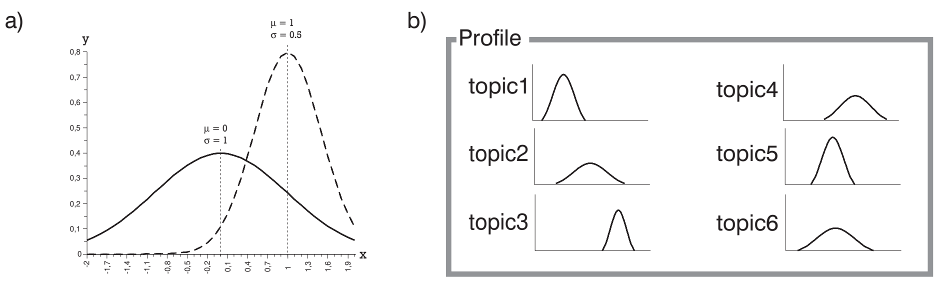

Abbildung 5: a) Beispiel für zwei verschiedene Gaußkurven; b) Beispiel eines Profils, das aus mehreren Themenprofilen besteht.

für eine hohe Signifikanz und bedeutet ein hohes Interesse an Objekten, deren Inhalte das spezielle Thema genau treffen. Die andere Kurve stellt das Interessenprofil eines Benutzers dar, der einem Thema weder sehr stark interessiert, noch völlig abgeneigt gegenübersteht. Deshalb ist der μ-Wert null und der σ-Wert eins, was für keine besondere Signifikanz steht. Abbildung 5b zeigt, wie mehrere, durch Gaußkurven dargestellte Themenprofile ein Benutzer- oder Objektprofil bilden.
Ein Maß dafür, wie stark ein Benutzer bezüglich eines Themas an einem Objekt interessiert ist, liefert die Schnittfläche der beiden Gaußkurven, die zum Benutzerprofil und zum Objektprofil dieses Themas gehören. Detaillierte Analysen zeigen, daß damit auch negierte Aussagen modelliert und korrekt behandelt werden [4]. Die Profilüberdeckung liefert ein n-Tupel an Themen-Überlappwerten je Objekt. Zur Bildung eines Gesamt-Rankings können darauf verschiedene Entscheidungsmodelle verwendet werden: das additive Modell, das konjunktive Modell, das disjunktive Modell oder ein von uns vorgeschlagenes durchschnittsbildendes Modell.
Das GRAS-Verfahren wurde um einen adaptiven Profilgenerierungs- und anpassungsalgorithmus erweitert, so daß sich durch Rückkopplungen die Objektprofile und die Benutzerprofile von selbst mit der Zeit korrekt einschwingen. Bei sehr großen Datenmengen können damit die Objekte auch mit der Standard-Gaußkurve (=indifferent) vorinitialisiert werden. Das ist ein großer Vorteil gegenüber Verfahren, die eine explizite Themenzuordnung für jedes Dokument benötigen. Gleichzeitig konnte in GRAS damit nicht nur inhaltsbasierte Personaliserung und Filterung sondern auch eine soziologische Filterung realisiert werden. Letztere bildet sozusagen "den guten Tip" nach.

7 Kooperationen

Im Projekt wurde eine erfolgreiche Kooperation mit dem SPP-Teilprojekt AVAnTA (Prof. Herzog, Uni Bremen) durchgeführt.
Im Bremer Projekt wird ein inhaltsbasiertes Videoanalysesystem entwickelt. Dabei werden Farb-, Textur-, Shot- und Cluster-Informationen generiert. In der Kooperation sollte untersucht werden, ob die Recherche auf den Analyseergebnissen anstatt über Graphgrammatiken, nicht günstiger über unsere Volltextindizierungstechniken realisiert werden kann. Dazu entwickelten wir für das Bremer Teilprojekt ein optimiertes DB-Schema zur Ablage und Verwaltung der Videoanalyseergebnisse einschließlich Vorschlägen für hochoptimierte SQL-Anfragen darauf. Gleichzeitig wurde eine Anbindung der Videoanalysedatenbank AVAnTA an OMNIS/2 spezifiziert. Ein Umsetzen dieser Spezifikation wird OMNIS/2 um ein Videoanalysesystem erweitern und damit über OMNIS/2 auch inhaltsbasierte Suchen in einer Videobibliothek ermöglichen.
Es fanden zusätzlich zu den SPP-Kolloquien vier gegenseitige Projekttreffen statt. Diese intensive Zusammenarbeit konnte mit einer gemeinsam betreuten Diplomarbeit aus München, die teils in München und teils in Bremen bearbeitet wurde, ausgebaut werden [8]. In der kommenden Förderphase soll die Zusammenarbeit fortgesetzt werden, damit die Anbindung fertig implementiert werden kann.

8 Ausblick

OMNIS/2 wird weiterhin von der Deutschen Forschungsgemeinschaft im Rahmen des Schwerpunktprogramms “Verteilte Verarbeitung und Vermittlung digitaler Dokumente” gefördert. Insbesondere soll in der weiteren Projektlaufzeit der entstandene Demonstrator zu einem vollen Prototypen ausgebaut werden und erste externe Systeme aus dem SPP eingebunden werden. In der weiteren Projektlaufzeit soll die Anbindung von MultiMAP-Applikationen als Beispiele für externe multimediale Dokumentendatenbanksysteme untersucht und realisiert werden.

9 Liste der Publikationen aus dem OMNIS/2-Projekt

Publikationen

[1] Specht G., *Complexity Analysis of Link Navigation in Dexter Based Hypermedia Database Systems*, Informatica, 8(1), Vilnius, Lithuania 1997, pp. 23-42.

[2] Schmidt A., Specht G., *Java, XML und Servlets zur Integration datenbankbasierter Applikationen im Web*, Proc. Java-Informations-Tage (JIT '98), 12.-13.11.1998 in Frankfurt, Springer-Verlag, Informatik aktuell, 1998, pp. 259-268

[3] Specht G., *Architekturen von Multimedia-Datenbanksystemen zur Speicherung von Bildern und Videos*, in: Luth N. (Hrsg.): Inhaltsbezogene Suche von Bildern und Videosequenzen in digitalen multimedialen Archiven, Workshop der 22. Jahrestagung für Künstliche Intelligenz (KI '98), 15.-17.9.1998 in Bremen, Bremen 1998, pp. 7-25

[4] Specht G., *Multimedia-Datenbanksysteme: Modellierung - Architektur - Retrieval*, Habilitationsschrift, Technische Universität München, 1998, 270 S.

[5] Bayer R., et. al., *Integration of Digital Services for Libraries*, Eingeladener Hauptvortrag, Int. Conf. Digital Libraries: Advanced methods and technologies, digital collections. 18.-22. Oktober, 1999, Sankt Petersburg, Russland, 6 S.

Posterpräsentation

[6] Specht G., Zirkel M., *Introducing Nested and Overlapping Links in HTML-Texts*, Int. Conference INET'98, 21.-24.7.1998 in Genf, Posterprogramm

Diplomarbeiten

[7] Lutzenberger H., *Evaluierung der Einsatzmöglichkeit von XML*, Diplomarbeit, Institut für Informatik, TU München, 1998

[8] Zimmer Y., *Integrating the Content-Based Video Retrieval System AVAnTA in the Digital Library System OMNIS/2*, Diplomarbeit, Institut für Informatik, TU München, 1999

[9] Beinhofer E., *Entwurf und Implementierung einer Anfragekomponente an das OMNIS/2-System*, Diplomarbeit, Institut für Informatik, TU München, 1999

[10] Böhm V., *Integritätsmechanismen bei der Speicherung von XML-Multimedia-Dokumenten im Projekt OMNIS/2*, Diplomarbeit, Institut für Informatik, TU München, 1999