Zweitveröffentlichung

Berthold, Henrike; Binkowski, Frank; Henrich, Andreas; u. a.

Architektur Multimedialer Informationssysteme

Datum der Zweitveröffentlichung: 17.02.2025

Akzeptiertes Manuskript (Postprint), Zeitschriftenartikel

Persistenter Identifikator: urn:nbn:de:bvb:473-irb-1064063

Erstveröffentlichung

Berthold, Henrike; Binkowski, Frank; Henrich, Andreas; u. a. (2002): Architektur Multimedialer Informationssysteme, in: Informatik, Forschung und Entwicklung : Organ der Fachbereiche Softwaretechnik, Datenbanken und Informationssysteme sowie Mensch-Computer-Interaktion der Gesellschaft für Informatik e.V., Berlin ; Heidelberg ; New York: Springer, Jg. 17, Nr. 2, S. 77–89, doi: 10.1007/s00450-002-0101-5.

Verlagshinweis

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/s00450-002-0101-5.

Rechtehinweis

Dieses Werk ist durch das Urheberrecht und/oder die Angabe einer Lizenz geschützt. Es steht Ihnen frei, dieses Werk auf jede Art und Weise zu nutzen, die durch die für Sie geltende Gesetzgebung zum Urheberrecht und/oder durch die Lizenz erlaubt ist. Für andere Verwendungszwecke müssen Sie die Erlaubnis der Rechteinhaberinnen und Rechteinhaber einholen.

Für dieses Dokument gilt das deutsche Urheberrecht.

Architektur Multimedialer Informationssysteme

Henrike Berthold[1], Frank Binkowski[1], Andreas Henrich[2], Silvia Hollfelder[3], Wolfgang Lindner[4], Ulrich Marder[5], Klaus Meyer-Wegener[6], Günter Robbert[2]

[1] TU Dresden, Fakultät Informatik, Institut für Systemarchitektur, Professur für Datenbanken, 01062 Dresden (e-mails: henrike.berthold,frank.binkowski@inf.tu-dresden.de)
[2] Universität Bayreuth, Fakultät für Mathematik und Physik, Fachbereich Informatik, Lehrstuhl für Angewandte InformatikI, 95440 Bayreuth (e-mail: {andreas.henrich,guenter.robbert}@uni-bayreuth.de)
[3] FhG, Institut für Integrierte Publikations- und Informationssysteme (IPSI), Dolivostr. 15, 64293 Darmstadt (e-mail: hollfeld@ipsi.fhg.de)
[4] Universität Rostock, Fachbereich Informatik, LS für Datenbank- und Informationssysteme, 18051 Rostock (e-mail: wl@informatik.uni-rostock.de)
[5] Universität Kaiserslautern, Fachbereich Informatik, AG Datenbanken und Informationssysteme, Postfach 3049, 67653 Kaiserslautern (e-mail: marder@informatik.uni-kl.de)
[6] Universität Erlangen-Nürnberg, Institut für Informatik VI, Martenstr. 3, 91058 Erlangen (e-mail: klaus.meyer-wegener@informatik.uni-erlangen.de)

1 Einführung

Die rasante Entwicklung des Worldwide Web eröffnet viele neue Anwendungsdomänen für Multimediale Informationssysteme (MMIS). Die traditionellen Systeme bzw. Ansätze sind jedoch oftmals den hohen Anforderungen einer Migration von einer abgeschlossenen, homogenen Anwendungswelt in eine offene, heterogene und hochgradig komplexe Umgebung, wie sie das WWW darstellt, nicht gewachsen. Zu den wesentlichen zu bewältigenden Problemen zählen hierbei die Bereitstellung verteilt gespeicherter multimedialer Daten, die Vereinigung bzw. Unterstützung unterschiedlicher Dokumentenmodelle, die Integration von Altsystemen, die Sicherstellung von Anwendungsneutralität sowie die dynamische Optimierung der Zugriffe auf Mediendaten. Die MMIS-Gruppe[1] arbeitet in ihren Projekten, die sich mit jeweils unterschiedlichen Aspekten verteilter MMIS befassen, an Einzellösungen für die genannten Probleme. Im Einzelnen sind dies die Projekte ACTIVA [1,16,21,22], Imos [7], MARS [20], Memo.real [28,30,36], MacWrap [28,29] und VirtualMedia [31,32] sowie das bereits abgeschlossene Projekt Dsmily [5,6]. Um diese Beiträge zu einem zukünftigen webbasierten MMIS koordiniert darstellen zu können, wurde eine *MMIS-Referenzarchitektur* entwickelt. Diese Referenzarchitektur stellt auf einem hohen Abstraktionsniveau die wesentlichen logischen Komponenten eines Web-basierten MMIS dar. Die Komponenten sind auf sechs Schichten verteilt: (von unten nach oben) Betriebssysteme, Speicherungssysteme, Integration, Datenmodell, Dienste und Anwendungen. Im Allgemeinen nutzen die Komponenten einer Schicht hauptsächlich die Schnittstellen bzw. Dienste der unmittelbar darunter liegenden Schicht. Aufruf- bzw. Kommunikationsbeziehungen zwischen weiter entfernten Komponenten sowie zwischen Komponenten innerhalb derselben Schicht sind jedoch ausdrücklich nicht ausgeschlossen. Derartige Details bleiben aber einer feineren, implementierungsnäheren Architekturspezifikation vorbehalten.

In Projekten der MMIS-Gruppe wird derzeit schwerpunktmäßig nur ein Teil der Komponenten der Referenzarchitektur untersucht bzw. entwickelt. Eine Beschreibung sowie Einordnung der einzelnen Projekte in die Architektur geben wir in Kapitel 3. Die darauf folgenden Kapitel 4 und 5 ergänzen diese Betrachtungen mit den Beschreibungen einer Reihe relevanter Fremdprojekte, wobei ersteres Forschungsprototypen und letzteres kommerziellen Entwicklungen gewidmet ist. Eine Gegenüberstellung aller betrachteten Projekte bezogen auf die eingeführte Referenzarchitektur geben wir abschließend in Kapitel 6. Zunächst soll aber im folgenden Kapitel die Referenzarchitektur etwas detaillierter vorgestellt werden.

[1] http://www6.informatik.uni-erlangen.de/mmis/

2 Die Architektur im Überblick

In den folgenden Abschnitten werden die sechs Schichten der Architektur von oben nach unten im einzelnen kurz charakterisiert. Eine tabellarische Gesamtübersicht wird in Tabelle 1 präsentiert (siehe Anhang).

2.1 Anwendungsschicht

Als interessant erachten wir hier in erster Linie Multimedia-Anwendungen, die von Natur aus auf einer globalen (Informa-

tions-) Infrastruktur beruhen. Anwendungsentwicklung und -ausführung sind eng verzahnt. Ein MMIS sollte beides gleichermaßen gut unterstützen, zumal der Übergang teilweise fließend ist, z.B. beim Authoring. Wir nehmen ferner an, dass verschiedene Anwendungen Mediendaten gemeinsam nutzen können, wobei alle notwendigen Synchronisationsmaßnahmen vom MMIS getroffen werden.

Im Folgenden gehen wir kurz auf einige anwendungsspezifischen Charakteristika ein: *Authoring-Anwendungen* sind starkinterak tiv, da der Anwender sowohl während einer Anwendung einzelne Medienobjekte analysiert und durchsucht (z. B. VCR) als auch mehrere Medienobjekte kombiniert (z. B. Erstellung eines neuen Videos aus mehreren Videoquellen). Des weiteren spielt das Transaktionsmanagement zur Konsistenzerhaltung eine wesentliche Rolle. Nutzer von *News on Demand-Anwendungen* erwarten eine effiziente, transparente Vorauswahl des Angebotes mehrerer, verteilter Inhaltsanbieter (Content-Provider). Diese Vorauswahl sollte den persönlichen Präferenzen (User Profiles) entsprechen. Normalerweise ist hier die Präsentationsdauer der einzelnen Medienobjekte (News Clips) relativ kurz. Zur Verbesserung der Systemleistung kann anwendungsspezifisches Wissen, wie die Beliebtheit und der Lebenszyklus einzelner News Clips, berücksichtigt werden. Unter *Point of Information* werden sämtliche Anwendungen verstanden, die vorstrukturierte Informationen vermitteln. Hierunter fallen beispielsweise Tourist-, Produkt- und Lern-Informationssysteme. Meist werden hypermediale Dokumente eingesetzt, durch die der Nutzer interaktiv navigieren kann. Ein wesentliches Merkmal dieser Anwendungen ist der Medien-Mix (Audio, Video, Bild, Text). Diese Dokumente werden mit Hilfe von Präsentationssprachen (z. B. HyTime [37], MHEG [35], SMIL [4]) im Vorfeld definiert (Pre-Orchestrated Documents). Bei *Digital Libraries-Anwendungen* liegt der Schwerpunkt auf der Unterstützung geeigneter Such- und Retrieval-Funktionen, da der Benutzer sein Informationsbedürfnis meist aus sehr großen, heterogenen Datenbeständen decken möchte. Auch diese Anwendungen sind im Allgemeinen starkinterak tiv, da oft eine Vorauswahl mit Hilfe von Such- oder Retrievalfunktionen getroffen und im nächsten Schritt diese Vorauswahl interaktiv analysiert wird.

2.2 Diensteschicht

Aufgrund der Ausrichtung der MMIS-Architektur auf (auch im geographischen Sinne) globale multimediale Kommunikation kann und soll die Heterogenität eines solchen Systems nicht völlig vor den Anwendungen verborgen werden. Aus diesem Grund gibt es neben globalen Diensten, die eine monolithische Sicht auf das MMIS erlauben, auch lokale Dienste, die normalerweise auf einzelne, sichtbare Subsysteme beschränkt sind. Zu den globalen Diensten gehören

- höhere bzw. globale Suchdienste,
- Datenanalysedienste,
- Benutzerverwaltung,
- Integrationsdienst(e) für mobile Komponenten, sowie
- Metacomputing, d. h. spontane Zusammenschaltung der Ressourcen autonomer Subsysteme zur Erfüllung komplexer Anwendungsanfragen wie bspw. Transformation von Medienobjekten.

Über lokale Dienste können Anwendungen die Serviceleistungen einzelner Subsysteme des MMIS direkt in Anspruch nehmen. Zu diesen Diensten gehören

- dauerhafte, konsistenzerhaltende Verwaltung multimedialer Dokumente,
- Suche nach Dokumenten,
- Präsentation (Playout) multimedialer Dokumente,
- Erzeugung und Veränderung multimedialer Dokumente sowie
- anwendungsseitige Definition von Dokumentenmodellen (Schemadefinition).

Jeder Dienst kann prinzipiell eine zweistufige Anwendungsschnittstelle anbieten. Die oberste Stufe ist eine GUI-Komponente, die einfach in die Anwendung (u. U. dynamisch) integriert wird. Diese Lösung ist besonders für die Suchdienste und den Playout-Dienst vorteilhaft, kann aber auch für die meisten anderen Dienste sinnvoll sein (Ausnahmen: Transaktionen und Metacomputing). Die untere Schnittstellenstufe besteht aus Funktionen mit einer festgelegten Anwendungssemantik(Dienst-API), die z. B. von (Applikations-) Serverkomponenten realisiert werden können. Die Dienste selbst basieren auf den generischen Zugriffsmethoden des MMIS-Datenmodells. Diese Schnittstelle könnte den Anwendungen theoretisch ebenfalls zugänglich gemacht werden. Allerdings würde das bedeuten, dass neben 3-tier- bzw. n-tier- auch (klassische) 2-tier-Applikationen unterstützt würden, was z. B. Probleme hinsichtlich der Skalierbarkeit des MMIS und der Realisierung eines konsistenten Sicherheitskonzepts verursacht.

2.3 Datenmodellschicht

Im MMIS werden standardisierte Datenmodelle wie das relationale Datenmodell (SQL), das objekt-relationale Datenmodell (SQL:1999) oder das von der ODMG definierte objektorientierte Datenmodell (mit OQL) verwendet, wo immer es sinnvoll erscheint. Keines dieser Modelle ist jedoch optimal für die Dokumentenverwaltung oder gar die Verwaltung multimedialer Dokumente geeignet. Daher führen wir folgende Ergänzungen ein:

Das *Dokumenten-Metamodell* ist ein Datenmodell für Dokumentenmodelle als Basis für die Dokumentenverwaltung und den Schemadefinitionsdienst. Natürlich gibt es hierfür geeignete Standards, z. B. aus dem XML-Umfeld (XML-DTD, XML-Schema, RDF, ...). Viele neuere Dokumentenmodelle wie bspw. SMIL entstanden bereits auf XML-Basis, während ältere Modelle wie HyTime oder MHEG auf dem komplexeren, für Web-basierte MMIS allerdings weniger gut geeigneten SGML-Standard aufbauen. Interessant erscheint jedoch auch die Idee eines Multimedia-Metamodells, wodurch sich bspw. Aspekte der Modellierung von Zeit (Synchronisation) und Raum (Layout) sowie der Annotierung multimedialer Dokumente verallgemeinern bzw. vereinheitlichen lassen. Aktuelle Ansätze wie SMIL (Synchronisation, Layout) oder MPEG-7 [34] (Annotierung), die sich im Übrigen auch eher als Dokumentenmodelle verstehen, lassen diesen Schritt noch vermissen.

Medienspezifische Abstrakte Datentypen (MADTs) sind ein Datenmodell für Medienobjekte, das die Semantik der verschiedenen Medienarten (z. B. Text, Bild, Audio und Video)

für die Dienste und Anwendungen festlegt. Ein Medienobjekt kann sowohl monomedial als auch multimedial (z. B. Video mit Ton) sein. Um auf den Inhalt von Medienobjekten zuzugreifen (z. B. durch den Playout-Dienst), steht eine spezielle MADT-Schnittstelle zur Verfügung. Die Struktur von Medienobjekten kann entweder als singulär (Text, Bild) oder als eindimensional (Audio, Video) charakterisiert werden. Letzteres bedeutet, dass ein oder mehrere Medien zusammen mit einem konstanten räumlichen Layout an eine (einzige) Zeitlinie gekoppelt sind. Komplexere, i. Allg. hierarchische oder netzwerkartige (Hypermedia-) Strukturen sind also durch das nicht fest vorgegebene Dokumentenmodell bereitzustellen. Aus dessen Sicht sind Medienobjekte die atomaren Bestandteile eines Dokuments.

Suchoperationen für multimediale Dokumente unterstützen in Standarddatenmodellen gewöhnlich nur die sog. Boolesche Suche. Diese Funktionalität wird insbesondere für die Suche nach inhaltsbeschreibenden Metadaten (Annotationen) benötigt. Für die Suche nach bzw. in multimedialen Dokumenten sind jedoch auch unscharfe Suchtechniken notwendig (z. B. stochastisches Information Retrieval). Hierfür ist die Anfragesprache (SQL oder OQL) um spezielle Suchfunktionen (z. B. die Bestimmung eines nächsten Nachbarn gemäß des Vektorraummodells) zu erweitern. Diese Funktionen können wiederum auf Operationen des Dokumentenmodells und der MADT-Schnittstelle aufbauen.

2.4 Integrationsschicht

Die Hauptaufgabe der Integrationsschicht ist das weitgehende Verbergen der Heterogenität und der physischen Verteilung des MMIS. Im Einzelnen unterscheiden wir folgende Integrationsdienste:

Über *vereinheitlichte Schnittstellen* können unterschiedliche Speicherungssysteme, die äquivalente Dienste erbringen, in gleicher Weise angesprochen werden. Konzeptionell kann dies z. B. mit Wrappern oder mit Konnektoren realisiert werden. Technologisch wird hierbei meist auf Standard-Middleware wie CORBA, EJB oder COM+ zurückgegriffen. Es ist aber ebenso möglich, hierfür die Erweiterungsmechanismen eines objekt-relationalen Datenbanksystems (ORDBS) auszunutzen.

Die *Anfrageverarbeitung* übernimmt die Übersetzung und die Zerlegung von Anfragen in Teilanfragen, die an die einzelnen Speicherungssysteme gesendet werden können. Soweit möglich bzw. sinnvoll, erfolgt eine globale Optimierung der Anfragen. Nach Eingang der Antworten zu den Teilanfragen muss eine Zusammenführung und Aufbereitung der Anfrageergebnisse vorgenommen werden.

Das *Materialisierungs-, Replikations- und Versionsmanagement* entscheidet, welche physischen Repräsentationen von Medienobjekten auf welchen Speicherungssystemen erzeugt (oder gelöscht) werden, und verwaltet diese Information. Das Materialisierungsmanagement ist im Wesentlichen volltransparent für die Anwendungen (d. h. mit Ausnahme von Administrations- und Entwicklungswerkzeugen). Das Replikationsmanagement ist halbtransparent, d. h., den Anwendungen ist freigestellt, dies zu steuern, profitieren aber auch von einer reinen Nutzung des Caching-Dienstes. Das Versionsmanagement muss hingegen durchgängig von den höheren Schichten (Datenmodell und Dienste) unterstützt werden, damit die Anwendungen es sinnvoll einsetzen können.

Die Aufgabe des *Ressourcenmanagements* ist die Verwaltung, Reservierung und Zuordnung von Ressourcen (z. B. Verarbeitungskapazität, Bandbreite von I/O-Systemen und Netzwerken), die für Zugriffe auf Medienobjekte zur Verfügung stehen. Eine Teilfunktion des Ressourcenmanagements ist die Zulassungskontrolle (Admission Control) zur Gewährleistung spezifizierter Dienstgüte (Quality of Service).

Das *Transaktionsmanagement* schließlich sorgt für die Steuerung globaler, verteilter Transaktionen durch ein Zwei-Phasen-Commit-Protokoll (2PC). Dabei ist auch zu berücksichtigen, dass einzelne Speicherungssysteme u. U. keine Transaktionsunterstützung bieten.

2.5 Speicherungssystemschicht

Aus funktionaler Sicht unterscheiden wir drei Arten von Speicherungssystemen:

Dokumenten- und Metadatenserver speichern multimediale Dokumente und alle hierfür zusätzlich benötigten (für die Anwendungen nicht sichtbaren) Verwaltungsdaten.

Medienobjektserver speichern Medienobjekte und synchronisieren bzw. optimieren deren Auslieferung an die Clients. Es kann mehrere verschiedenartige Medienserver geben. Darunter sollten sich auch solche befinden, die ein Protokoll für die Auslieferung kontinuierlicher Medien in Echtzeit unterstützen.

Indexserver speichern Merkmale von Dokumenten (oder einzelnen Medienobjekten) in einer für die verwendeten IR-Methoden optimierten Form (z. B. spezielle Baumstrukturen oder invertierte Listen).

Physisch können dankder darüber liegenden Integrationsschicht sehr verschiedene Systeme zum Einsatz kommen. Das schließt insbesondere auch bereits existierende *Datenbanken* ein, die nicht speziell für das MMIS entworfen wurden. Diese könnten sowohl als Medienobjektserver als auch als Dokumentenserver integriert werden. Im letzteren Fall müsste das Dokumentenmodell des Servers mit Hilfe des Dokumenten-Metamodells nachgebildet und eine geeignete Abbildungsfunktion (bspw. ein Wrapper) in der Integrationsschicht bereitgestellt werden.

Natürlich kann die Integrationsschicht nicht alle potenziellen Defizite eines konkreten Speicherungssystems ausgleichen. Das betrifft z. B. fehlende oder mangelhafte Echtzeitfähigkeit (von Medienservern) oder fehlende Unterstützung verteilter Transaktionen.

2.6 Betriebssystemschicht

Grundsätzlich kann das MMIS auf einem oder mehreren der heute weit verbreiteten universellen Betriebssysteme wie UNIX (und seine Derivate) oder Windows NT aufgesetzt werden. Einige Teile des MMIS, insbesondere Medienobjektserver, Ressourcenmanagement und Playout-Dienst, könnten jedoch mit Hilfe spezieller Echtzeit-Betriebssystemfunktionen eine bessere Dienstgüte, höheren Durchsatz und eine bessere Ressourcennutzung liefern. Einige der wünschenswerten Funktionen, z. B. Clustern von Servern mit Lastverteilungs- und Failover-Automatikso wie

Unterstützung von Spezialhardware für die digitale Signalverarbeitung durch geeignete (z. B. mehrschichtige) Treibermodelle, werden von modernen Betriebssystemen bereits ansatzweise unterstützt. Es fehlen jedoch oftmals noch Funktionen für den für Multimediasysteme typischen gemischten Realtime/Timesharing-Betrieb (Realtime-Prozess/Thread-Scheduling, optionale feste Ressourcenzuteilung).

Generell ist zu sagen, dass nach dem heutigen Stand der Techniksolche Echtzeitfunktionen vor allem Prozesse mit streng periodischer Charakteristik, z. B. Abspielen kontinuierlicher Medien, gut unterstützen können. Der MMIS-Playout-Dienst und die ihn unterstützenden Komponenten sind allerdings auf permanente Benutzerinteraktion ausgelegt und erzeugen daher günstigstenfalls partiell periodische Betriebssystemprozesse. Für eine optimale Betriebssystemunterstützung erscheint daher eine enge(re) Verzahnung des MMIS mit dem Betriebssystem auch auf semantischer Ebene unvermeidlich.

3 Eigene Projekte

Im Folgenden beschreiben wir die Forschungsprojekte der MMIS-Gruppe, die unabhängig voneinander durchgeführt werden, aber einen gemeinsamen Rahmen suchen. Neben einer Beschreibung des jeweiligen Ziels bzw. Ansatzes werden auch Bezüge zur Referenzarchitektur geknüpft, die verdeutlichen, welchen Beitrag zum gesamten MMIS das Projekt leistet. Ein tabellarischer Überblick ̈uber alle Projekte findet sich in der Tabelle 2 (siehe Anhang).

3.1 MARS: Retrieval-Dienst für multimediale Lernumgebungen

Im Rahmen des MARS-Projekts [20] wird ein Retrieval-Dienst für multimediale Lernumgebungen entwickelt, der die Suche nach wiederverwendbaren Teilen von Lerninhalten auf verschiedenen Arten von Dokumenten bzw. Bestandteilen von Dokumenten mit unterschiedlicher Granularität ermöglicht. Hierbei werden neben der attributbasierten Suche und der Volltextsuche zusätzlich die Suche hinsichtlich der Dokumentstruktur sowie die inhaltsbasierte Suche auf Medienobjekten unterstützt. Alle für die Suche relevanten Daten werden in einem Repository, basierend auf dem objektorientierten DBS H-PCTE, verwaltet. Als Anfragesprache kommt die eng an OQL angelehnte Sprache POQL^{MM} zum Einsatz. Um die syntaktischen Details von POQL^{MM} vor Endbenutzern zu verbergen, wird zur Zeit eine graphische Benutzeroberfläche implementiert.

Als Betriebssystemplattform wird im MARS-Projekt auf Datenbankseite ausschließlich UNIX verwendet, die Speicherungsschicht besteht aus dem DBS H-PCTE, in dem die Dokumente zusammen mit ihren Registrierungs- und Beschreibungsdaten abgelegt werden. Für die Verwaltung von Daten für die inhaltsbasierte Suche werden zusätzlich spezielle Indexstrukturen, basierend auf Bäumen, eingesetzt. Hauptbestandteil der Integrationsschicht ist die Anfragebearbeitung. Als Datenmodell wird das mächtige Datenmodell von PCTE [25] übernommen (reguläre Pfadausdrücke, Linkkonzept mit verschiedenen Kategorien von bidirektionalen Links). Zusätzlich werden spezielle Suchoperationen bereitgestellt, welche die inhaltsbasierte Suche auf multimedialen Daten ermöglichen.

In der Diensteschicht bietet MARS die Verwaltung von multimedialen Lerninhalten sowie die attribut-, struktur- und inhaltsbasierte Suche nach wiederverwendbaren Teilen von Dokumenten an.

Der Anwendungsbereich des MARS-Projekts ist bislang auf das Authoring von multimedialen Lerninhalten festgelegt, wobei aber durchaus auch andere Anwendungsgebiete wie Digitale Bibliotheken oder Zeitungsarchive denkbar sind.

3.2 Imos: Föderiertes Multimedia-DBS

Im Imos-Projekt [42] wird ein Föderiertes Multimedia-DBS (FMDBS) entworfen und teilweise implementiert. Ein FMDBS föderiert drei Arten von Komponentensystemen: traditionelle Datenbanken, Medienobjektserver und Indexserver.

Die Leistungen des FMDBS werden in einer Föderationsschicht erbracht, die Teile der Integrations-, Datenmodell- und Diensteschicht realisiert. Diese Schicht bietet den Clients ein globales Schema, das die Daten der Komponentensysteme sowie zusätzliche Daten, z. B. Beziehungen zwischen Mediendaten und strukturierten Daten, enthält. Es werden keine Kopien der Daten angelegt. Lediglich die Information über die Konstruktion der globalen Daten aus den Komponentensystemen sowie die zusätzlichen Daten werden in der Föderationsschicht gespeichert. Die Medienobjekte sind in diesem globalen Schema nur durch ihre Metadaten und ihren Zugriffsidentifikator repräsentiert.

Die Föderationsschicht stellt außerdem einen spezialisierten Client zur Verfügung, der die Echtzeitverbindungen zu den Medienservern über eine Multimedia-Middleware aufbaut und Anwendungen eine Schnittstelle zum Zugriff auf Medienobjekte bietet [28]. Außerdem realisiert die Föderationsschicht Datenbankfunktionalitäten wie Transaktionen und Benutzerverwaltung.

Im Imos-Projekt wird das Gesamtsystem entworfen. Der Aufbau des globalen Schemas und die Anfragebearbeitung werden detailliert untersucht.

Ein FMDBS erbringt keine Funktionalität auf der Anwendungsschicht. Es stellt sich den Anwendungen als ein System dar. Wenn man es als Teil eines komplexeren Systems betrachtet, bietet es die folgenden lokalen Dienste auf der Diensteschicht: inhaltsbasierte Suche nach Medienobjekten, Operationen auf Medienobjekten (Playout, Editing) sowie Schemadefinition (-integration).

Mediensuchoperationen und Dokumente werden mit den Mitteln des Standard-Datenmodells der ODMG modelliert (Datenmodellschicht). Medienobjektoperationen werden über eine Zugriffsschnittstelle gesondert zur Verfügung gestellt.

Die verteilte Anfragebearbeitung ist der Integrationsschicht zugeordnet. OQL-Anfragen an das globale Schema werden in Anfragen an die Komponentensysteme zerlegt. Die Komponentensysteme werden über Wrapper vereinheitlicht. Das FMDBS-Transaktionsmanagement basiert auf vorhandenen Lösungen.

Die FMDBS-Daten werden hauptsächlich in den Komponentensystemen gespeichert. Das umfasst alle drei Systeme, die in der Speicherungssystemschicht genannt sind. Als Dokumenten- und Metadatenserver fungieren traditionelle DBS.

Die Anfrageverarbeitung in der Föderationsschicht, die der wichtigste Forschungsschwerpunkt ist, kann auf einem Standard-Betriebssystem laufen (Betriebssystemschicht).

3.3 Dsmily: Verteiltes Multimedia Information Retrieval

Das Projekt Dsmily (Distributed Scalable Multimedia Information Retrieval System) hat sich von 1995 bis 1999 mit einem Konzept für verteiltes Information Retrieval befasst, das auch mit heterogenen Informationsquellen umgehen kann [5]. Basierend auf dem Prinzip des probabilistischen Retrieval wurde eine Erweiterung vorgeschlagen, die unterschiedliche Indexierungen in den beteiligten Servern zulässt und trotzdem das Optimalitätskriterium einhält. Das System kann hierarchisch strukturiert werden, also Zwischenknoten für Gruppen von Servern einführen, so dass es besser skaliert als zweistufige Systeme. Eine gegebene Anfrage wird in der Serverhierarchie weiter nach unten gereicht, in jedem Knoten auf dem Weg zu den Blättern ausgewertet und mit einem Retrieval Status Value (RSV) versehen. Diese RSVs werden in einer komplexen Formel zusammengefasst zu einem globalen RSV, der das globale Ranking des Dokuments bestimmt. Weiterhin gibt es ein sog. Auswahlkriterium, nach dem entschieden werden kann, bestimmte untergeordnete Server nicht zu fragen, wenn von vorneherein abgeschätzt werden kann, dass sie keinen Beitrag zum globalen Ergebnis liefern können. Das spart in jedem Fall Ressourcen, u. U. sogar Geld, wenn der zu befragende Server kostenpflichtig ist. Das Verfahren kann auf multimediale Dokumente ausgedehnt werden, wenn man die sog. Passagen eines Dokuments indexiert und separat recherchiert. Die Konzepte wurden in umfangreichen Dokumentenbeständen aus der TREC-Sammlung validiert [6].

Dsmily zielt primär auf die Diensteschicht und hier die globalen Suchdienste, muss aber, um dort die Funktion erbringen zu können, auch Anpassungen in anderen Schichten vornehmen. So werden bei den lokalen Diensten und im Datenmodell Suchoperationen vorausgesetzt, die ein Ranking durchführen können. In der Integrationsschicht realisiert Dsmily eine spezifische Form der Anfrageverarbeitung, nämlich die Verteilung von Teilanfragen mit Auswahl der Server und die Integration der von dort erhaltenen Ranking-Listen. In der Speicherungssystemschicht führt Dsmily eigene Indexserver ein, die mit den anderen Servern kooperieren können und sie um IR-Funktionalität erweitern, falls sie diese nicht selbst anbieten können.

3.4 VirtualMedia: Datenmodell für Medienobjekte in verteilten MMIS

Im VirtualMedia Projekt wird ein Datenmodell für Medienobjekte entwickelt, das es erlaubt, die potenzielle Leistungsfähigkeit von Peer-to-Peer Metacomputing [33] für die Verarbeitung von Mediendaten in einem verteilten MMIS auszunutzen [31]. Eines der Hauptprobleme hierbei ist, dass man zum Entwicklungszeitpunkt weder weiß,

- über welche speziellen Ressourcen für die Medienverarbeitung das MMIS zur Laufzeit verfügt, noch,
- in welchem Speicherungsformat die zu verarbeitenden Mediendaten vorliegen werden.

Der in VirtualMedia verfolgte Ansatz besteht folglich darin, von diesen beiden (und noch einigen anderen) Aspekten zu abstrahieren. Diese Abstraktion bezeichnen wir als Transformationsunabhängigkeit [32]. Auf diese Weise wird die Semantikv on Medienobjekten vollständig von technischen Aspekten wie Medienformaten, Materialisierung und Optimierung von Verarbeitungsschritten getrennt.

Zur Beschreibung der Semantiksteht den Anwendungen eine deskriptive Sprache (VirtualMedia Markup Language, VMML) zur Verfügung. Ein VMML-Dokument wird von VirtualMedia in einen gerichteten, azyklischen Anfragegraphen übersetzt. In diesem Graphen repräsentieren die Startknoten (virtuelle) Medienobjektproduzenten, die Endknoten Medienobjektkonsumenten, die Zwischenknoten (virtuelle) Medienobjektfilter und die Kanten die Medienobjekte selbst. Der VirtualMedia-Algorithmus verknüpft diesen Graphen mit intern in einer Metadatenbankv erwalteter Information über die Materialisierung von Medienobjekten (Materialisierungsgraphen) sowie die Verarbeitungsressourcen des MMIS (Filterdatenbank). Der daraus resultierende Plan ist anschließend innerhalb der vom MMIS bereitgestellten Metacomputing-Umgebung ausführbar.

Im Rahmen der Referenzarchitektur konzentriert sich VirtualMedia primär auf den Aspekt der medienspezifischen abstrakten Datentypen (MADT) innerhalb der Datenmodellschicht. An zweiter Stelle sind noch die Spezifikation von Metadatenschemata, die Integration von Medienservern (beides Speicherungssystemschicht) sowie die Spezifikation eines Metacomputing-Dienstes (Diensteschicht) zu nennen.

3.5 ACTIVA: Zulassungskontrolle für interaktive Multimediaanwendungen

ACTIVA (Admission Control for Interactive Multimedia Applications) zielt explizit auf das Ressourcenmanagement von interaktiven Anwendungen, wie beispielweise Video Browsing in Digitalen Bibliotheken [21] oder das Navigieren in einer Point of Information-Applikation. Diese Anwendungen sind nicht nur durch häufige VCR-Interaktivität, sondern auch durch häufige Medienwechsel (z. B. Medien-Mix eines Point of Information) gekennzeichnet. Eine wesentliche Anforderung an derartig interaktive Anwendungen ist das Erreichen von geringen Latenzzeiten, d. h. geringe Wartezeiten beim Wiederaufnehmen einer Präsentation nach einer Interaktion. Daher werden hier, im Gegensatz zu den konventionellen Verfahren, gesamte interaktive Sitzungen verwaltet. Diese Vorgehensweise impliziert, dass die Zulassungskontrolle genau einmal für eine gesamte Sitzung und nicht für alle einzelnen Medienobjekte, die während einer Sitzung präsentiert werden, erfolgen muss. Das Hauptproblem liegt in der Vorhersage der zukünftigen Ressourcenverbräuche, die starkv om interaktiven Benutzerverhalten determiniert werden. Im Rahmen dieses Ansatzes werden sowohl Vorhersagen, die auf Beobachtungen der Vergangenheit basieren (observation based), als auch stochastische Vorhersagen ermöglicht. Im ersten Fall wird das beobachtete Systemverhalten als Indikator für die Zukunft verwendet. Die stochastischen Vorhersagen basieren auf Benutzer-Interaktionsmodellen, die als zeitkontinuierliche Markov-Ketten (Continuous Time Markov Chains) modelliert werden. Dieses Benutzer-Interaktionsmodell wird analysiert und

für die Zulassungskontrolle ausgewertet [16]. Zur Spezifikation des Benutzer-Interaktionsmodells verwenden wir anwendungsspezifisches Wissen, wie z. B. für Video Browsing [1] oder Electronic Shopping [22]. Weitere Aufgaben des Ressourcenmanagements, wie beispielsweise effizientes Layout für Daten (z. B. Data Striping in DiskArrays), das effiziente Puffern am Server oder am Client sowohl zur Erhöhung der Systemleistung als auch zur Gewährleistung von Dienstgüte und Netzwerkaspekte werden im Rahmen von ACTIVA nicht berücksichtigt. In der MMIS-Referenzarchitektur greift ACTIVA primär auf die Funktionalität von Medienobjektservern und Dokumenten- und Metadatenbanken zu. Dabei wird davon ausgegangen, dass die Systemressourcen auf der Speicherungssystemebene spezifiziert und der Integrationsebene zur Verfügung gestellt werden, da diese Information für die Zulassungskontrolle benötigt wird. Eine weitere Aufgabe der Medienserver ist die Umsetzung der Medienanfragen. Diese umfasst die Lieferung von Medienobjekten für eine Anwendung. Die Metadatenbankwird für Zugriffe auf meist technische Metadaten (z. B. Bandbreite eines Medienobjektes) benötigt.

3.6 MacWrap: ORDBS-basiertes Multimedia-DBS

Das Ziel des MacWrap-Projekts [28,29] ist die Entwicklung eines neuen Konzepts für den Aufbau eines Multimedia-DBS, welches statt einer Middleware-Komponente ein ORDBS als „Föderierungsebene“ für verschiedene Subsysteme nutzt. Die Idee besteht darin, das DBS selbst, basierend auf der Erweiterungstechnologie, um verschiedene Server, und damit um gewünschte Funktionalität, zu erweitern. Dabei soll untersucht werden, inwieweit sich bereits vorhandene Middleware-Techologien im Rahmen der Erweiterungstechnologie ausnutzen lassen. Als Architektur wird die sog. Multimedia-Compensator-Wrapper-Architektur (MacWrap) vorgeschlagen. Benutzerdefinierte Datentypen und Funktionen sollen dabei auf externe Server zurückgreifen können, ohne dass dies dem Anwender des ORDBS sichtbar wird. Beispiele für externe Server sind Medienserver und Datenbanken, obwohl das Konzept auch darüber hinaus Gültigkeit hat.

Das Konzept ist weitgehend anwendungsneutral, orientiert sich aber an Digitalen Bibliotheken. Bei den Diensten wird eindeutig die Dokumentenverwaltung in den Vordergrund gestellt. Andere Dienste müssen sich aus den eingebundenen Systemen ergeben. Das Datenmodell basiert auf dem SQL:1999-Standard. So entspricht der Struktur- und Operationenteil (mit Ausnahme der Typkonstruktoren, die aus einem früheren Entwurf zum Standard übernommen wurden) im Wesentlichen dem gegenwärtigen Standard. Anders verhält es sich jedoch bei den höheren Konzepten. Da die in SQL:1999 enthaltenen Aspekte hier nicht ausreichen, sollen sie um orthogonale UDTs, Services und erweiterte ECA-Regeln ergänzt werden. Zentral für das Konzept sind die Integration der Datenquellen, hier besonders das Wrapping, sowie die dynamische Schemaerweiterung um benutzerdefinierte Datentypen und Funktionen der Subsysteme. Die Anfrageverarbeitung spielt ebenfalls eine Rolle. Die Speicherungssysteme werden nicht selbst entwickelt, sondern nur genutzt und eingebunden. Daher bleibt auch die Wahl des Betriebssystems offen, das ORDBS selbst läuft unter einem Standard-Betriebssystem.

3.7 Memo.real: Echtzeitfähiges Medienobjekt-Speicherungssystem

Im Rahmen des Sonderforschungsbereichs 358 „Automatisierter Systementwurf“ befasst sich das Teilprojekt G3 „Memo.real“ mit dem Entwurf von echtzeitfähigen Medienobjekt-Speicherungssystemen [36]. Die Herausforderung liegt darin, Echtzeitfähigkeit mit Datenunabhängigkeit zu verbinden. Wenn die Anwendungen vom intern verwendeten Speicherungsformat entkoppelt werden, muss die Möglichkeit eröffnet werden, in Echtzeit zu konvertieren. Die Kooperation mit der Professur für Betriebssysteme und mehreren Professuren in der Elektrotechnik im Rahmen des genannten SFB lässt dies möglich erscheinen. Die Professur für Betriebssysteme entwickelt ein Echtzeit-Betriebssystem namens DROPS [18]. Die für den Medienserver genutzten Konverter laufen dann als Prozesse oder Threads unter DROPS. Dazu muss ihr Ressourcen-Bedarf aber vorab mitgeteilt werden, was eine genaue Analyse der Konverter und eine Beschreibung ihres Verhaltens erfordert. Diese Arbeiten laufen derzeit. Als ein Mittel der Beschreibung werden schwankungsbeschränkte Ströme verwendet, die den Datenaustausch zwischen Dateisystem, Konverter und Anwendung charakterisieren können. Falls mehrere Konverter hintereinander geschaltet werden müssen, sollten auch zwischen ihnen schwankungsbeschränkte Ströme realisiert werden. Das Modell wurde um den Umfang der auszutauschenden Daten und das Format des gesamten Datenobjekts [30] erweitert. Damit lassen sich schon notwendige Bedingungen für die Ausführbarkeit einer Kette von Konvertern formulieren. Ziel muss es natürlich sein, auch hinreichende Bedingungen zu formulieren. Mit der Konverterbeschreibung soll dann eine Planungsrechnung durchgeführt werden, bevor die Kette zur Ausführung an DROPS gegeben wird.

All dies geht noch von existierenden Konvertern aus und versucht diese einzubinden. Die beteiligten Teilprojekte aus der Elektrotechnik haben aber durchaus das Potenzial, zugeschnittene Hardware für neue Konverter oder zumindest die kritischen Teile davon zu entwerfen und auch zu produzieren. Dies eröffnet noch weitergehende Möglichkeiten. Die Konverterbeschreibung wird hier ebenfalls benötigt, nur ist sie nicht mehr einfach Reflexion des Vorhandenen, sondern Blaupause für das Neue.

In der Referenzarchitektur ordnet sich Memo.real in die Speicherungssystemschicht ein, weil eben ein Medienobjektserver entworfen wird. Seine Funktionalität wird sich auch in dem medienspezifischen Datenmodell der Datenmodellschicht niederschlagen. Anderen Diensten wie Präsentation und Ressourcenmanagement liefert Memo.real wichtige Unterstützung, ohne sie jedoch vollständig zu realisieren.

4 Ausgewählte Fremdprojekte

Die im Folgenden dargestellten Fremdprojekte wurden nach ihrer Nähe zu den eigenen Projekten ausgewählt. Angesichts der Fülle weltweiter Aktivitäten in diesem Gebiet kann es sich nur um eine kleine Auswahl handeln. Es besteht immer das Risiko, wichtige neue Entwicklungen zu übersehen. Wir hoffen, dennoch eine repräsentative Auswahl getroffen zu haben.

4.1 Garlic

Im Garlic-Projekt des Forschungsinstitutes der IBM in Almaden [10] wird eine Datenbank-Middleware entwickelt, die Datenbanken und Indexserver koppeln kann. Jedes Garlic-Komponentensystem hat einen Wrapper, der das Komponenten-Schema in das Garlic-Datenmodell überführt.

Die Garlic-Middleware bietet den Clients ein globales objektorientiertes Datenbankschema, auf das mittels einer objektorientierten SQL-artigen Sprache zugegriffen werden kann. In diesem Projekt wurde speziell untersucht, wie Indexserver eingebunden werden können. Das wurde am Beispiel des Bildretrievalsystems QBIC, welches ebenfalls von der IBM entwickelt wurde, auch demonstriert [11].

Jede Komponente in Garlic verfügt über Daten, die es über sein Schema dem Garlic-System zur Verfügung stellt. Die Schemata der Komponentensysteme bilden die Basis des globalen Schemas. Beziehungen zwischen den Daten unterschiedlicher Quellen werden über sog. Komplexe Objekte realisiert.

Besonderes Augenmerkwird im Garlic-Projekt auf die Optimierung von Anfragen in einem solchen heterogenen System gelegt [17,40].

Ein Garlic-System leistet auf der Diensteschicht die Schemadefinition (-Integration) und die Suche nach Medienobjekten. Mediensuchoperationen werden mit den Mitteln des Standard-Datenmodells der ODMG modelliert (Datenmodellschicht). Bezüglich der Integrationsschicht liegt der Fokus auf der Anfrageverarbeitung und dem Wrapping. Die zu integrierenden Systeme der Speicherungsschicht sind Metadatenserver (Datenbanken) und Indexserver. Garlic basiert auf einem Standard-Betriebssystem (Betriebssystemschicht).

4.2 HERON

Im interdisziplinär angelegten HERON-Projekt [26] zwischen der Informatikund der Bibliothekder Universität Augsburg werden Recherchewerkzeuge für Bildinhalte weiterentwickelt und in den Arbeitsplatz von Kunsthistorikern integriert. Als Anwendungsgebiet und Testfeld wurde für das HERON-Projekt die Heraldik ausgewählt, die im kunsthistorischen Bereich zur Identifizierung und Datierung von Dokumenten bzw. Kunstwerken eine sehr bedeutende Rolle als Hilfswissenschaft einnimmt. Ziel des HERON-Projekts ist es, den Zeitaufwand bei der Recherche nach Wappen durch den Einsatz eines DBS, welches zum einen die Verwaltung der umfangreichen Wappenbestände übernimmt und zum anderen die inhaltsbasierte Suche nach Wappen ermöglicht, zu verringern. Für die inhaltsbasierte Suche werden schon in anderen Bildretrievalsystemen eingesetzte Techniken, wie die automatische Bildsegmentierung und die Extraktion von Features, wie Farbe, Form oder Textur, an den Anwendungsbereich angepasst und in das Recherchewerkzeug integriert [44]. Da die Wappenbände zahlreiche textuelle und genealogische Informationen zu den verzeichneten Wappen enthalten, welche für die Recherche von großer Bedeutung sind, werden diese Informationen ebenfalls in das DBS übernommen.

Aus Sicht der MMIS-Referenzarchitektur wird die Betriebssystemschicht im HERON-Projekt durch das Betriebssystem UNIX gebildet, auf der Ebene der Speicherungsschicht wird ein ORDBS eingesetzt. Die Anfrageverarbeitung und JDBC als Wrapping-Mechanismus sind die wesentlichen Bestandteile der Integrationsschicht. Hinsichtlich der Datenmodellschicht wird im HERON-Projekt das objekt-relationale Datenmodell von DB2 eingesetzt. Die von der Diensteschicht angebotenen Leistungen sind die Verwaltung von umfangreichen Wappenbeständen sowie die attribut- und inhaltsbasierte Suche nach Wappen im Datenbestand. Der Anwendungsbereich des HERON-Projekts ist zur Zeit auf die Heraldikbeschr änkt.

4.3 AMOS

Ziel des Systems AMOS (Active Media Object Store) war die Erweiterung eines objektorientierten bzw. objekt-relationalen [23] DBS um Funktionalitäten zur Unterstützung der Speicherung, der Anfrage und Präsentation interaktiver multimedialer Dokumente [2]. Im Gegensatz zu ACTIVA erfolgte das Management der Ressourcen nach dem „Best Effort"-Prinzip, d. h., alle Anfragen werden ohne explizite Zulassungserlaubnis bedient. Um auch bei Systemengpässen kontinuierliche Präsentationen zu ermöglichen, wurden Adaptionsmechanismen für Server und Client entwickelt [43]. Das System AMOS bietet im Rahmen der MMIS-Referenzarchitektur die vollständige Funktionalität der Diensteschicht an: Transaktionsmanagement (bei kontinuierlichen Medien jedoch bedingt über Einfüge-Operationen des DBS gesteuert), Schemadefinition über objektorientiertes oder objekt-relationales Schema, Dokumentenverwaltung [9], Playout und Editing.

4.4 Cardio-OP Multimedia Repository

Im Verbundprojekt Cardio-OP [27] wird ein MMIS entwickelt, das speziell auf die Anforderungen an medizinische Informationssysteme (hier mit dem Schwerpunkt Herzchirurgie) abgestimmt ist. Das System soll sowohl die Funktion eines umfassenden Nachschlagewerks für praktizierende Ärzte sowie Patienten erfüllen als auch die Erstellung und Nutzung multimedialen Lehrmaterials für Vorlesungen und zum Selbststudium unterstützen.

Das MMIS bietet eine inhaltsbasierte, medientypübergreifende Suche nach Dokumenten und Dokumentfragmenten. Letzteres ist die Voraussetzung für ein ebenfalls neu entwickeltes Authoring-Werkzeug, das die Wiederverwendung bereits vorhandener Dokumentfragmente fördert. Der Präsentationsdienst (Playout) ist in der Lage, die Dokumente dynamisch an das jeweilige Präsentationsmedium sowie spezielle Benutzerprofile anzupassen. So kann ein und dasselbe Dokument in jeweils optimaler Weise bspw. im Hörsaal oder auf dem Heim-PC eines Studenten präsentiert werden. Darüber hinaus unterstützt das MMIS auch Offline-Präsentationen, z. B. auf CD-ROM oder in Printmedien.

Dementsprechend unterstützt das in Cardio-OP eingesetzte Dokumentenmodell ZyX [8], anders als vergleichbare Standards wie MHEG oder SMIL, besonders die Aspekte Wiederverwendung und Präsentationsneutralität. Dies wird durch Konzepte wie die explizite Modellierung von Dokumentfragmenten, die Unterstützung von Dokumentkomposition „on-the-fly", sowie Adaption und Personalisierung von Dokumenten erreicht. Das Mediendatenmodell basiert hingegen auf kommerziellen Produkten [46] wie dem Excalibur Text

DataBlade, dem Excalibur Image DataBlade und dem Video Foundation DataBlade, bei denen es sich um Erweiterungen des Informix Dynamic Server handelt (vgl. Abschnitt 5.2). Ein wiederum selbst entwickeltes „Media Integration Blade“ stellt einheitliche Zugriffsmethoden für die verschiedenen Medientypen bereit:

- Lokalisierung: Ein sog. „Uniform Locator“ kapselt die Referenz auf den Speicherort, wobei auch verschiedene externe Servertypen unterstützt werden.
- Metadatenverwaltung: Gemeinsamkeiten technischer Art zwischen den Basistypen werden durch eine einfache Klassenhierarchie dargestellt.
- Annotierung: Unabhängig von konkreten Medienformaten können sowohl diskrete als auch kontinuierliche Annotationen zu den Medienobjekten verwaltet werden.
- Suchoperationen: Es werden medientypunabhängige Anfragen unterstützt, die sich auf die Metadaten und die Annotationen beziehen können.

5 Kommerzielle Systeme

Der Fokus dieses Kapitels liegt auf dem, was die beiden marktführenden DBS-Hersteller IBM und Oracle derzeit leisten. Nach der Übernahme von Informix hat IBM erwartungsgemäß ein deutliches Übergewicht, wenn man die Summe der gebotenen Funktionalität betrachtet. Es bleibt jedoch abzuwarten, ob wirklich beide DBS, das „hauseigene“ DB2 und der Informix Dynamic Server, langfristig weiterentwickelt werden.

5.1 IBM DB2

Das ORDBS DB2 [12] unterstützt durch den Einsatz spezieller Zusatzmodule, sogenannte Extender, die Verwaltung und die inhaltsbasierte Suche auf unterschiedlichen Medientypen wie z. B. Text, Bild, Audio und Video. Für jeden genannten Medientyp wurden gleichnamige Extender implementiert und in DB2 integriert, die jedoch im Funktionsumfang sehr unterschiedlich ausfallen. Positiv hervorzuheben ist hierbei, dass insbesondere die Image-, Audio-, und Video-Extender vielfältige Möglichkeiten zur Speicherung und Formatkonvertierung für die verschiedenen Medientypen anbieten. Hierbei können diese neben einer internen Speicherung in einem DB2-eigenen Format auch extern, d. h. außerhalb des DBS abgelegt werden. In diesem Fall wird im DBS lediglich eine Referenz gespeichert, die den Zugriff auf die Daten realisiert.

Der Image-Extender integriert die im IBM-Projekt QBIC [38] entwickelten Methoden für die inhaltsbasierte Suche auf Bilddaten in DB2. Hierbei wird die Suche anhand von Farbmerkmalen, Formeigenschaften und Texturen unterstützt. Leider ist der Extender als Blackbox realisiert, so dass der Anwendungsentwickler keinen direkten Zugriff auf die Bilddaten bekommt und somit eine Erweiterung der schon vorhandenen Retrievalmöglichkeiten um eigene Methoden, z. B. für die Bildsegmentierung, erheblich erschwert wird. Diese Aussage gilt auch für alle anderen Extender, da sie ebenfalls als in sich geschlossene Komponenten realisiert sind.

Im Vergleich zum Image-Extender ist die vom Audio- und Video-Extender angebotene Retrievalfunktionalität derzeit nur als sehr rudimentär anzusehen. So bietet beispielsweise der Audio-Extender neben der Formatkonvertierung nur Möglichkeiten zur attributbasierten Suche, wie z. B. nach der Länge einer Audiodatei oder dem Audioformat. Etwas mehr Funktionalität hingegen stellt der Video-Extender durch seine integrierte Erkennung von Szenenübergängen in Videodaten zu Verfügung. Dadurch ist es möglich, neben der attributbasierten Suche und der Formatkonvertierung eine automatische Segmentierung von Videos in Videosequenzen vorzunehmen.

Für die inhaltsbasierte Suche auf Textdaten wurde der Text-Extender entwickelt, welcher neben der reinen Volltextsuche auch das approximative String-Matching und die Suche mittels regulärer Ausdrücke unterstützt. Darüber hinaus bietet IBM weitere Extender für spezielle Datentypen bzw. Einsatzbereiche an. So lassen sich beispielsweise mittels des XML-Extenders XML-Dokumente oder mit Hilfe des Spatial-Extenders geographische Daten in die Datenbank übernehmen und verwalten.

5.2 Informix Dynamic Server (IBM)

Im Gegensatz zur IBM-Eigenentwicklung DB2 wartet der mit Informix aufgekaufte Dynamic Server mit einem sehr offenen Erweiterungskonzept auf. Mit umfangreichen Programmierschnittstellen (DataBlade Developers Kit) werden Drittherstellern und Endbenutzer eingeladen, neue Datentypen, Methoden und sogar Zugriffstrukturen (z. B. R-Baum) in das DBS zu integrieren [13]. Infolgedessen bietet Informix selbst nur wenige grundlegende Multimedia-Erweiterungen an: das Excalibur Text Search DataBlade, das Excalibur Image Datablade, und das Video Foundation DataBlade.

Das Excalibur Text Search DataBlade [15] bietet im Wesentlichen Funktionen für die Volltextsuche an. Die Texte können sowohl innerhalb des DBS verwaltet werden als auch extern in verschiedenen proprietären Formaten (z. B. MS Word; die benötigten Zugriffsfunktionen stellt Informix bereit). Die zentrale Rolle bei der Textsuche spielt die UDF „etx_contains“, die z. B. wie folgt zur Formulierung einer unscharfen Anfrage mit Ergebnis-Ranking einsetzbar ist:

```
SELECT rc.score, id, description
       FROM videos
WHERE etx_contains(description,
  Row('multimedia', 'PATTERN_TRANS
     & PATTERN_SUBS'),
  rc # etx_ReturnType)
ORDER BY 1;
```

Der erste Parameter der etx_contains-Funktion ist der zu durchsuchende Text (hier eine Spalte der Tabelle videos). Der zweite definiert das zu suchende Textmuster und der dritte (optionale) Parameter ist ein OUT-Parameter, der einen RSV (Score) für jeden durchsuchten Text zurückliefert. Dieser ist um so höher, je besser der Text die Anfrage erfüllt, und kann daher zur Sortierung der Texte nach Relevanz herangezogen werden. Um die Auswertung der etx_contains-Prädikate effizient durchführen zu können, erstellt und verwaltet das DataBlade spezielle Volltextindexstrukturen, die mit einer entsprechenden „Create index“-Anweisung angelegt werden müssen.

Auch das Excalibur Image DataBlade [14] konzentriert sich auf die Unterstützung der unscharfen Suche, in diesem

Fall nach Bildern. Wenn ein Bild in der Datenbankgespeichert wird, extrahiert eine sog. „feature extractor function“ charakteristische Merkmale aus dem Bild und bildet daraus einen Merkmalsvektor, welcher dann anstelle des eigentlichen Bildes indexiert wird. Für eine Anfrage muss ein Suchbild bereitgestellt werden, für das ebenfalls ein Merkmalsvektor berechnet wird. Dieser wird mit den gespeicherten Merkmalsvektoren verglichen, was im Ergebnis immer eine Rangfolge der ähnlichsten Bilder ergibt.

Das DataBlade verwendet ein eigenes Rasterbildformat, das auch den Anwendungen als externes Speicherformat angeboten wird. Für einige andere Formate wie GIF, JPEG, TIFF etc. sind Konverter vorhanden. Außer zum Schreiben, Lesen und Konvertieren gibt es nur wenige Funktionen (Abfrage einiger Parameter wie Bildhöhe, -breite, Speicherformat etc. sowie Skalierung der Bildgröße).

Das Video Foundation DataBlade (VFDB) [24] stellt einige Basisdatentypen und -funktionen für den Medientyp Video zur Verfügung. Obwohl es auch direkt durch Anwendungen genutzt werden kann, ist es in erster Linie als Ausgangspunkt für Entwickler von Video-DataBlades mit weitergehender Funktionalität gedacht. Die vom VFDB bereitgestellte Funktionalität umfasst im Wesentlichen Folgendes:

- Formatunabhängige Adressierung: Punkte in der Timeline eines Videos können unabhängig vom Speicherungsformat (z. B. MPEG oder AVI) beschrieben werden, beispielsweise im SMPTE-Format oder in Form von Frame-Nummern. Die Verwendung solcher Timecodes erleichtert die Verknüpfung von beschreibenden Metadaten (z. B. Szenenbeschreibungen oder Szenenwechselindexierung) mit den Videorohdaten, vor allem, wenn die Rohdaten in verschiedenen Speicherformaten vorliegen (z. B. digital als MPEG-Datei und analog auf VHS-Band).
- Geräteunabhängige Speicherungsschnittstelle: Die Videorohdaten können auf den verschiedensten Speichermedien abgelegt werden, z. B. Dateisysteme, Videorecorder oder Video-Streaming-Server. Das VFDB definiert für die Nutzung dieser Speichermedien eine einheitliche, Gerätedetails verbergende Schnittstelle, das sog. Virtual Storage Interface (VSI).
- Metadatenschema: Es werden einige Tabellen vordefiniert, die Metadaten zu Videos aufnehmen können, darunter die wichtigsten physischen Attribute (Länge, Format usw.) sowie das Video oder einzelne Szenen beschreibende Daten, die verschiedenen Typs sein können (Text, Grafik, Audio, Video) und zu Gruppen mit semantischen Gemeinsamkeiten gebündelt werden können (Stratifikation). Das Schema ist nach Bedarf erweiterbar.

Als Beispiele für auf dem VFDB basierende Erweiterungen sind das Continuous Long Field DataBlade [23] und das MPEG DataBlade [19], beide entwickelt am FhG IPSI, zu nennen.

5.3 Oracle

Wie andere Datenbankhersteller bietet auch Oracle eine Multimedia-Erweiterung mit der Bezeichnung interMedia [39] für ihre aktuelle Datenbankversion Oracle9i™ an. Diese Erweiterung stellt neue Objektdatentypen und zugehörige Methoden zur Verfügung, die es ermöglichen, Audio-, Video- und Bilddaten in der Datenbankzu speichern, zu indexieren, zu manipulieren, zu verarbeiten und nach ihnen zu suchen [45].

Positiv hervorzuheben ist die nahtlose Integration von interMedia in das DBS von Oracle. Hierbei nutzt interMedia das objekt-relationale Datenmodell von Oracle und stellt dem Anwendungsprogrammierer die Objektdatentypen Image, Audio und Video sowohl als Java- als auch als C++-Klassen bereit. Auf diese Weise können die Objektdatentypen von interMedia einfach um neue Methoden erweitert bzw. neue Datentypen durch Vererbung erzeugt werden.

Die Erweiterbarkeit der interMedia-Architektur wird durch sog. „Plugins“ gewährleistet. Diese können neben zusätzlichen Formaten, Kompressionsverfahren, Methoden, speziellen Indexstrukturen und Verarbeitungsalgorithmen auch weitere externe Datenquellen unterstützen.

Die Stärken von interMedia liegen deutlich auf der Seite von Java, da neben den schon erwähnten Java-Klassen auch eine erweiterte JDBC/SQLJ-Schnittstelle für Anfragen an das DBS angeboten wird, welche vielfältige Methoden zum Speichern und Laden von interMedia-Objektdatentypen bereithält. Hierbei können die Rohdaten der Medienobjekte wahlweise in der Datenbankals BLOBs (unter Transaktionskontrolle) oder außerhalb der DB (ohne Transaktionskontrolle), z. B. auf einem Media-Server, gespeichert werden. Im letzteren Fall wird wiederum lediglich ein Zeiger in der DB gespeichert. Die entsprechenden Metadaten, Attribute und Methoden für die verwalteten Objekte werden in beiden Fällen im DBS von Oracle abgelegt.

Für die Medientypen Bild, Audio und Video wird eine Konvertierung von/in unterschiedliche Formate sowie eine Kompression „on demand“ bereitgestellt. Des Weiteren unterstützt interMedia die Zusammenarbeit mit einem „streaming server“ wie etwa RealNetworks™ Server und ermöglicht somit, Video- bzw. Audiodateien in Echtzeit zu verarbeiten. Jedoch unterstützt interMedia bislang nur die attributbasierte Suche nach Audio und Video. Inhaltsbasiertes Retrieval von Bildern wird basierend auf Signaturen, die Farb-, Form- und Texturmerkmale von Bildern beschreiben, unterstützt.

5.4 CueVideo (IBM)

Das 1997 begonnene Forschungsprojekt CueVideo, das am Almaden Research Laboratory der IBM durchgeführt wird, hat die Ermöglichung einer effektiven Suche in den wachsenden Beständen audiovisueller Daten zum Ziel. Die Hauptzielrichtung besteht dabei nach [3] darin, ein System zum automatischen Analysieren und Zusammenfassen von Lehr- und Lernvideosammlungen zu entwickeln und damit die Suche und das Browsen darin zu erleichtern. Dabei soll auf der einen Seite eine hohe Leistungsfähigkeit erreicht werden, auf der anderen Seite sollen aber auch Funktionen entwickelt werden, wie sie bei textbasierten Internet-Suchmaschinen üblich sind. Das System besteht aus zwei voneinander unabhängigen Teilen: einem Offline-Indexierungsprozess und einem Online-Such- und Browse-System.

Der Offline-Prozess verarbeitet alle üblichen Videoformate und nutzt zur Indexierung der Videos eine Kombination von automatisierter visueller und auditiver Analyse. Das beinhaltet u. a. die Trennung der Sprache von Musikund anderen Hintergrundgeräuschen, die Erkennung kontinuierlicher Sprache mit ViaVoice™ und eine Videosegmentierung. Er

erzeugt verschiedene Vorschau-Videos einschließlich des Storyboards (als Folge von Schlüsselbildern), einer Animation und einer Slideshow dieses Storyboards und des sog. Moving Storyboard (einer Storyboard-Animation unter Einbeziehung der Originaltonspur). Durch die Verwendung einer Zeitskalen-Modifikation ist CueVideo in der Lage, die Wiedergabe von videobasierten Inhalten unter Beibehaltung der natürlichen Tonhöhen zu beschleunigen oder zu verlangsamen. Das wird z. B. im Moving Storyboard verwendet.

Der Online-Video-Server besteht im Wesentlichen aus einem sprachbasierten Suchsystem, einer Anfrageverarbeitung, einem Prozess, der die gefundenen Ergebnisse, insbesondere deren Vorschau-Videos, präsentiert, und einem Streaming-Multimedia-Server. Anfragen können als Texte eingegeben oder auch gesprochen werden. Das System ermittelt die Stellen in den Videos, in denen diese Texte vorkommen. Es besitzt ein grafisches Nutzerinterface, das mit jedem Internet-Browser auf unterschiedlichen Plattformen verwendet werden kann und das Standard-Plugins wie z. B. RealNetworks™ oder QuickTime™ einsetzt [41].

6 Zusammenfassung

Wir haben eine Referenzarchitektur für (gerade erst im Entstehen begriffene) global verteilte multimediale Informationssysteme eingeführt. Des Weiteren wurden eine Reihe eigener sowie fremder Projekte vorgestellt, die einen Beitrag zur Realisierung dieser Architektur leisten. Die Tabelle 2 im Anhang ist als Übersicht und Orientierungshilfe gedacht. Sie beinhaltet eine grobe Einordnung der verschiedenen Projekte in Bezug auf die Referenzarchitektur und zeigt, wo die jeweiligen Forschungs- bzw. Entwicklungsschwerpunkte liegen. Dabei bedeuten die Symbole in der Tabelle:

- ■ Forschungs- bzw. Entwicklungsschwerpunkt,
- □ Komponente, die am Rande betrachtet oder deren Existenz vorausgesetzt wird.

Dementsprechend bedeutet ein leeres Feld, dass die entsprechende Komponente (fast) gar nicht betrachtet und auch nicht vorausgesetzt wird.

Bei den Fremdprojekten beruht die Einordnung allein auf der jeweils angeführten Literatur und kann daher unvollständig sein. Wir haben ebenfalls die kommerziellen Systeme in die Tabelle aufgenommen, wobei prinzipiell die gleichen Vorbehalte wie bei den Fremdprojekten gelten.

7 Ausblick

Globale, Web-basierte multimediale Informationssysteme stehen erst am Anfang ihrer Entwicklung. Doch was sie inhärent sind, nämlich *verteilt*, charakterisiert auch am besten den derzeitigen Stand dieser Entwicklung: Viele Forschungsprojekte fokussieren auf Teilaspekte von MMIS, tun sich jedoch schwer, sich selbst in Beziehung zu anderen Aktivitäten bzw. zum „großen Ganzen“ zu setzen. Diese Arbeit versucht, mit der vorgestellten Referenzarchitektur einen Rahmen zu schaffen, der es in Zukunft erleichtern soll, Forschungsprojekte und Systeme im Bereich der MMIS einzuordnen sowie Schnittstellen und Überschneidungen mit verwandten Ansätzen zu identifizieren und zu dokumentieren.

Natürlich muss auch die Referenzarchitektur weiterentwickelt werden. Weder erheben wir den Anspruch, mit der vorliegenden Version alle relevanten Aspekte erfasst zu haben, noch konnten alle relevanten Projekte in diese Übersicht aufgenommen werden. Die Auswahl der Projekte ist vor allem durch zwei Merkmale gekennzeichnet: einerseits wurde der Schwerpunkt auf Aktivitäten im deutschsprachigen Raum und andererseits auf den Datenbanken- bzw. Information-Retrieval-Hintergrund der Autoren gelegt. Neben einer verstärkten Konzentration auf internationale Projekte wäre daher auch eine Verschiebung des Blickwinkels auf andere „Communities“, bspw. Netzwerke, Content Management oder KI, interessant. Dies gilt insbesondere im Hinblickauf die (fast) leeren Zeilen in Tabelle 2. Diese bedeuten, dass den entsprechenden Aspekten/Komponenten in der Datenbanken- bzw. IR-Community bisher eher wenig Aufmerksamkeit zuteil wird. Es bedeutet jedoch keinesfalls, dass auf diesen Gebieten nicht geforscht wird, sondern vielmehr, dass der „Blick ̈uber den Tellerrand“ noch mehr geschärft werden muss.

Danksagungen. Diese Arbeit wird teilweise durch folgende Projekte der Deutschen Forschungsgemeinschaft (DFG) gefördert:

- Sonderforschungsbereich 358 „Automatisierter Systementwurf“,
- Sonderforschungsbereich 501 „Entwicklung großer Systeme mit generischen Methoden“,
- Graduiertenkolleg 191 „Werkzeuge zum effektiven Einsatz paralleler und verteilter Rechnersysteme“,
- Graduiertenkolleg 466 „Verarbeitung, Verwaltung, Darstellung und Transfer multimedialer Daten – technische Grundlagen und gesellschaftliche Implikationen“.

Literatur

1. Aberer, K., Hollfelder, S.: Resource Prediction and Admission Control for Interactive Video Browsing Scenarios using Application Semantics. In: Proc. 8th IFIP Conference on Data Semantics (DS-8) - Semantic Issues in Multimedia Systems, Rotorua, Neuseeland, 5.–8. Jan. 1999, pp 27–46
2. Aberer, K., Thimm, H., Neuhold, E. J.: Multimedia Database Management Systems. In: Handbookof Multimedia Computing, Borko Furht (Ed.), Boca Raton, Florida, CRC Press 1998, Kap. 28, pp 579–604
3. Amir, A., Ponceleon, D., Blanchard, B. et al.: Using audio timescale modification for video browsing. In: 33rd Hawaii Intl. Conf. on System Sciences (HICSS-33), Maui (Hawaii), 4.–7. Jan. 2000
4. Ayars, J. et al. (Eds.): Synchronized Multimedia Integration Language (SMIL 2.0) Specification. W3C Recommendation, 7. August 2001, http://www.w3.org/TR/2001/REC-smil20-20010807/
5. Baumgarten, C.: A Probabilistic Solution to the Selection and Fusion Problem in Distributed Information Retrieval. In: Proc. ACM SIGIR Conf. on Research and Development in Information Retrieval 1999
6. Baumgarten, C.: Probabilistic Information Retrieval in a Distributed Heterogeneous Environment. Dresden, Technische Universität, Fakultät Informatik, Diss. 1999
7. Berthold, H.: A Federated Multimedia Database System. In: Proc. EDBT 2000 Ph. D. Workshop, Konstanz, 31. März – 1. April 2000

8. Boll, S., Klas, W.: ZyX - A Semantic Model for Multimedia Documents and Presentations. In: Proc. 8th IFIP Conference on Data Semantics (DS-8) - Semantic Issues in Multimedia Systems, Rotorua, Neuseeland, 5.–8. Jan. 1999
9. Boll, S., Klas, W., Löhr, M.: Integrated Database Services for Multimedia Presentations. In: S. M. Chung (Ed.): Multimedia Information Storage and Management, Dordrecht: Kluwer 1996
10. Carey, M. J. et al.: Towards Heterogeneous Multimedia Information Systems: The Garlic Approach. In: Proc. of the 5th Intl. Workshop on Research Issues in Data Engineering (RIDE): Distributed Object Management 1995
11. Cody, W. F. et al.: Querying multimedia data from multiple repositories by content: The Garlic project. In: Proc. of IFIP 2.6 Third Working Conference on Visual Database Systems (VDB-3) 1995
12. DB2 Universal Database, IBM Corp., 2001, URL http://www.ibm.com/software/data/db2/library/
13. Developing DataBlade Modules for Informix Internet Foundation.2000. Informix Corp. 1999, URL http://www.informix.com/informix/whitepapers/devdbm_wp.pdf
14. Excalibur Image DataBlade Module. User's Guide Version 1.1. Informix Press, Juli 1997
15. Excalibur Text Search DataBlade Module. User's Guide Version 1.1. Informix Press, Juli 1997
16. Friedrich, M., Hollfelder, S., Aberer, K.: Stochastic Resource Prediction and Admission for Interactive Sessions on Multimedia Servers. In: Proc. of ACM Multimedia, Los Angeles, Okt./Nov. 2000, pp 117–126
17. Haas, L. M. et al.: Optimizing queries across diverse data sources. In: Proc. 23rd Intl. Conf. on Very Large Data Bases (VLDB) 1997
18. Härtig, H., Baumgartl, R., Borriss, M. et al.: DROPS - OS Support for Distributed Multimedia Applications. In: Proc. 8th ACM SIGOPS European Workshop, Sintra, Portugal, 7.–10. Sept. 1998
19. Malsy, M., Hollfelder, S., Steinmetz, A., Aberer, K.: Unterstützung des strukturierten Zugriffs auf MPEG Videos in einem Multimedia Datenbankmanagementsystem. In: Proc. GI-Workshop „Inhaltsbezogene Suche von Bildern und Videosequenzen in digitalen multimedialen Archiven", KI '98, Bremen, 16.-17. September 1998, pp 27–34
20. Henrich, A., Robbert, G.: MARS: A Retrieval Service for Multimedia Authoring Environments. In: Proc. Challenges of ADBIS-DASFAA Symposium on Advances in Databases and Information Systems, Prag, Tschechische Rep., 5.–8. Sept., 2000, pp 88–98
21. Hollfelder, S., Everts, A., Thiel, U.: Designing for Semantic Access: A Video Browsing System. In: Multimedia Tools and Application, 11(3), August 2000, pp 281–293
22. Hollfelder, S., Oria, V., Özsu, M. T.: Mining User Behavior for Resource Prediction in Interactive Electronic Malls. In: Proc. of IEEE Intl. Conf. on Multimedia and Expo (ICME) 2000, New York, Juli 2000, pp 863–866
23. Hollfelder, S., Schmidt, F., Hemmje, M., Aberer, K.: Transparent Integration of Continuous Media Support into a Multimedia DBMS. In: Özsu, Dogac, Ulusoy (Eds.): Proc. 3rd Biennial World Conf. on Integrated Design and Process Technology (IDPT), Issues and Applications of Database Technology (IADT'98), Berlin 1998, Vol. 2, pp 192–199
24. Informix Video Foundation DataBlade Module. User's Guide Version 1.1. Informix Press, Juni 1997
25. Kelter, H.: A high-performance object management system for system development environments. In: Proc. of the 16th Annual Intl. Computer Software and Applications Conf. (COMPSAC'92), Chicago, USA 1992, pp 45–50
26. Kießling, W., Erber-Urch, K., Balke, W.-T., Birke, T., Wagner, M.: The HERON Project – Multimedia Database Support for History and Human Sciences. In: Dassow, J., Kruse, R. (Eds.): Informatik'98: Informatikzwischen Bild und Sprache, Berlin Heidelberg New York: Springer 1998, pp 309–318
27. Klas, W., Greiner, C., Friedl, R.: Cardio-OP: Gallery of Cardiac Surgery. In: Proc. Intl. Conf. on Multimedia Computing and Systems '99 (ICMCS'99), Florenz, Italien, 7.–11. Juni 1999
28. Lindner, W., Berthold, H., Binkowski, F., Heuer, A., Meyer-Wegener, K.: Enabling Hypermedia Videos in Multimedia Database Systems Coupled with Realtime Media Servers. In: Proc. Intl. Database Engineering & Applications Symposium (IDEAS 2000), Yokohama, Japan, 18.–20. Sept., 2000, pp 327–336
29. Lindner, W., Heuer, A.: Das MacWrap Project - Nutzung eines Objekt-Relationalen DBMS für ein offenes Multimedia-Datenbanksystem. In: Proc. 12. GI-Workshop „Grundlagen von Datenbanken", Plön, 13.–16. Juni, 2000, pp 61–65
30. Märcz, A., Binkowski, F., Meyer-Wegener, K.: Modellierung von Mediendatenkonvertern für die zeitgerechte Ausgabe in Medienservern. TU Dresden, Fakultät Informatik, September 2000
31. Marder, U.: On Realizing Transformation Independence in Open, Distributed Multimedia Information Systems. In: Proc. 9. GI-Fachtagung „Datenbanksysteme in Büro, Technikund Wissenschaft" BTW'2001 (Oldenburg, 7.–9. März 2001), Heuer, A. (ed.), Heidelberg: Springer, März 2001, pp 424–433
32. Marder, U.: Medienspezifische Datentypen für objekt-relationale DBMS: Abstraktionen und Konzepte. In: Proc. 8. GI-Fachtagung „Datenbanksysteme in Büro, Technikund Wissenschaft" BTW '99 (Freiburg, 1.–3. März 1999), Buchmann, A. P. (ed.), Springer-Verlag, Berlin 1999, pp 210–231
33. Marder, U., Kovse, J.: Multimedia Metacomputing. In: Proc. 7th Workshop on Multimedia Information Systems (MIS'01), Capri, Italien, 7.–9. Nov. 2001, pp 173–182
34. Martinez, J. M. (Ed.): Overview of the MPEG-7 Standard. März 2001, URL http://www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm
35. Meyer-Boudnik, T., Effelsberg, W.: MHEG Explained. In: IEEE Multimedia 2 (1), 26–38 (1995)
36. Meyer-Wegener, K.: Entwurf von echtzeitfähigen Medienobjekt-Speicherungssystemen. Teilprojekt G3 des Sonderforschungsbereichs 358, Dresden, Technische Universität, Fakultät Informatik, URL http://wwwdb.inf.tu-dresden.de/forschung/mmis/Memo.real/, Projektstart 1999
37. Newcomb, S. R., Kipp, H. A., Newcomb, V. T.: HyTime: The hypermedia/time-based document structuring language. In: Comm. ACM 34 (11), 67–83 (1991)
38. Niblack, W., Barber, R., Equitz, W. et al.: The QBIC Project: Querying Images by Content, Using Color, Texture, and Shape, In: SPIE Proc. Vol. 1908, San Jose, USA 1993, pp 173–187
39. Oracle9i^{TM} interMedia Data Sheet, January 2001, URL www.oracle.com
40. Roth, M. T., Ozcan, F., Haas, L. M.: Cost models do matter: Providing cost information for diverse data sources in a federated system. In: Proc. 25th Intl. Conf. on Very Large Databases (VLDB) 1999, pp 599–610
41. Srinivasan, S., Ponceleon, D., Amir, A., Blanchard, B., Petkovic, D.: Engineering the Web for Multimedia. In: Murugesan, S., Deshpande, Y. (Eds.): WebEngineering 2000, LNCS 2016. Berlin: Springer, 2001, pp 77–89

42. Süß, H.: A Flexible Architecture for the Integration of Media Servers and Databases. In: Jajodia, Özsu, Dogac (Eds.): Proc. 4th Intl. Workshop on Multimedia Information Systems (MIS'98), Istanbul, Türkei, 24.–26. Sept. 1998, LNCS 1508, pp 174–184
43. Thimm, H.: Optimal Quality of Service under Dynamic Resource Constraints in Distributed Multimedia Database Systems. Sankt Augustin, GMD-Forschungszentrum InformationstechnikGmbH, GMD Research Series, Nr. 10, Zugl.: Dissertation Techn. Univ. Darmstadt, Juni 1998
44. Vogel, J., Balke, W.-T., Kießling, W.: (Semi-)Automatic Segmentation in Historic Collections of Heraldic Images. In: Proc. 15th Intl. Conf. on Pattern Recognition, ICPR 2000, Barcelona, Spanien, Sept. 2000
45. Ward, R: Oracle9i^{TM} interMedia Audio, Image, and Video User's Guide and Reference, Oracle Corporation 2001
46. Westermann, U., Klas, W.: Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets. In: Proc. 1st Intl. Workshop on Multimedia Intelligent Storage and Retrieval Management, MISRM '99, Orlando, Florida, 30. Okt. 1999

Anhang

Tabelle 1. Die MMIS-Referenzarchitektur im Überblick

Schicht		Komponenten		Merkmale
Anwendungen		Authoring		starke Interaktivität, Generierung neuer Dokumente
		News on Demand		Vorauswahl, persönliche Präferenzen
		Point of Information		vorstrukturierte Information, Medien-Mix
		Digitale Bibliothek		Schwerpunkt auf Suche, sehr große Datenbestände, Interaktivität
Dienste	globale	Suchdienste		Verzeichnisse, Information-Broking, verteiltes Information Retrieval
		Analyse		Multimedia Mining
		Benutzerverwaltung		Authentifizierung, Autorisierung, Profiling, Billing
		Integrationsdienst für Mobile Komponenten		Datenabgleich und Management, Hoarding, Isolation-only Transaktionen, Server-Push
		Metacomputing		Verteilte/parallele Medienobjektverarbeitung
	lokale	Dokumentverwaltung		Persistenz, Konsistenz (transaktional)
		Suche nach Dokumenten		Fakten-Retrieval, Information Retrieval (auch kombiniert)
		Präsentation (Playout)		VCR-Fkt. (Play, FF, ...)
		Editing/Transformation		Manipulation von Dokumenten
		Schemadefinition		Definition von Dokumentenmodellen
Datenmodell		Dokumenten-Metamodell		z. B. XML oder SGML
		Medienspezifisches Datenmodell		MADTs für Text, Bild, Audio, Video
		Suchoperationen		Suche nach 'ähnlichen' Dokumenten
		(Erweiterte) Standard-Datenmodelle		Relationenmodell, ODMG, ...
Integration		Vereinheitlichte Schnittstellen	Wrapper	Verbergen von Heterogenität, Integration von Altsystemen
			Konnektoren	
		Anfrageverarbeitung		z. B. Zerlegung in Teilanfragen, Vereinigung von Teilergebnissen
		Materialisierungsmanagement		Sorgt für Ortstransparenz, kann auch Versions- und Replikatverwaltung (Caching) beinhalten
		Ressourcenmanagement		Verwaltung, Zulassungskontrolle, Reservierung
		Transaktionsmanagement		Koordination verteilter Transaktionen (2PC)
Speicherungssysteme		Dokumenten- und Metadatenserver		Multimediale Dokumente
		Medienobjektserver		atomare Medienobjekte, (optional) Streaming und Filterung
		Datenbanken		(semi-)strukturierte Daten
		Indexserver		Dokumentmerkmale, (optional) Thesauri und Ontologien
Betriebssysteme		Realtime/Timesharing-Systeme		optimiert für 'weiche' Echtzeit, Unterstützung für schwankungsbeschränkte Ströme, z. B. DROPS
		Standard-Betriebssysteme		Unix, Windows, ...

Tabelle 2. Einordnung der betrachteten Projekte und Systeme in die MMIS-Referenzarchitektur

Komponenten / Projekte	MARS	Imos	Dsmily	VirtualMedia	ACTIVA	MacWrap	Memo.real	Garlic	HERON	AMOS	Cardio-OP	DB2	Informix	Oracle	CueVideo
Authoring	□										■				
News on Demand					□										
Point of Information					□										
Digitale Bibliothek						□			■		■				□
Suchdienste			■												■
Analyse												□		□	■
Benutzerverwaltung					□									□	
Integr. mobile Komponenten															
Metacomputing				□		□									
Dokumentverwaltung	■								□	■	■			□	□
Suche nach Dokumenten	■	■	■					■	■	□[2]	■	■	□		■
Präsentation (Playout)	□	□		□	■	□	□		■	■	■			□	■
Editing/Transformation	□	□		□			□		□	■	■	□		□	
Schemadefinition		■				□		□		□[3]		□	□	□	
Dokumenten-Metamodell		□													
Medienspezifisches Datenmodell	□	□	□	■	□	□	■		□	■	■				
Suchoperationen	■	■	■					■	■		■	□	■		■
(Erweiterte) Standard-Datenmodelle	□	■	□			■		■	□	□	■	■	■	■	
Vereinheitlichte Schnittstellen – Wrapper	□	□	□			■		■			□	□	□	□	
Vereinheitlichte Schnittstellen – Konnektoren															
Anfrageverarbeitung	□	■	■			■		■	■		■	□	■	□	■
Materialisierungsmanagement				■					■			□	□	□	
Ressourcenmanagement				□	■	□	□			■					
Transaktionsmanagement		□				□						□	□	□	
Dokumenten- und Metadatenserver	■	□		□	□	■		□	□	□	□				□
Medienobjektserver	□	□		□	□	□	■			□	□	□	□	□	□
Datenbanken						□						■	■	■	
Indexserver	■	□	■					□							□
Realtime/Timesharing-Systeme							□								
Standard-Betriebssysteme	□	□	□			□		□	□	□	□	□	□	□	□

[2] Nur Fakten-Retrieval
[3] Bzgl. Standard-Datenmodell