

# TUM

INSTITUT FÜR INFORMATIK

Die konzeptionelle und funktionelle  
Weiterentwicklung des multimedialen digitalen  
Meta-Bibliothekssystems OMNIS/2

Michael G. Bauer, Günther Specht



TUM-I0109

Dezember 01

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM-INFO-12-I0109-0/1.-FI

Alle Rechte vorbehalten

Nachdruck auch auszugsweise verboten

©2001

Druck:            Institut für Informatik der  
                  Technischen Universität München

# Die konzeptionelle und funktionelle Weiterentwicklung des multimedialen digitalen Meta-Bibliothekssystems OMNIS/2

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

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Zielsetzung</b>	<b>3</b>
1.1	Allgemeine Projektbeschreibung . . . . .	3
1.2	Stand am Ende der 1. Förderphase . . . . .	3
1.3	Zusammenfassung der Projektarbeit der 2. Förderphase . . . . .	4
1.4	Gliederung des Arbeitsberichts . . . . .	5
<b>2</b>	<b>Weiterentwicklung des Demonstrators zum operativen Forschungsproto-</b>	<b>5</b>
	<b>typen</b>	
2.1	Entwurf eines Austauschformats zur Anbindung externer Systeme . . . . .	6
2.2	Nachträgliche Verlinkung der XML Dokumente mittels XSLT . . . . .	6
2.3	Das Dokumentenmodell . . . . .	8
2.4	Das Ankerkonzept . . . . .	9
2.5	Rechtekonzept für Dokumente und Links . . . . .	10
2.6	Anbindung der darunterliegenden Bibliothekssysteme . . . . .	10
2.7	Optimierung der Datenbankanbindung . . . . .	12
<b>3</b>	<b>Kooperationen</b>	<b>13</b>
<b>4</b>	<b>Ausblick</b>	<b>13</b>
<b>5</b>	<b>Liste der Publikationen aus diesem Projekt</b>	<b>13</b>
<b>A</b>	<b>Anhang: DTD des internen OMNIS/2-Formats</b>	<b>14</b>



# 1 Einleitung und Zielsetzung

## 1.1 Allgemeine Projektbeschreibung

Im Projekt “OMNIS/2: Integration von multimedialen Datenbanksystemen und Bibliotheksrecherchesystemen” des DFG-Schwerpunktprogramms (SPP) “Verteilte Verarbeitung und Vermittlung digitaler Dokumente” (V3D2)<sup>1</sup> 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 entsprechend 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 sollen mehrere, verschiedenartige Systeme gleichzeitig unter einer Oberfläche eingebunden werden können. 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 2. Förderphase (1999-2001). Ein Bericht über die 1. Förderphase liegt in [10] vor.

## 1.2 Stand am Ende der 1. Förderphase

Die erste Förderphase (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 so spezifiziert, daß nicht nur wie ursprünglich geplant das Bibliothekssystem der Fakultät für Informatik sondern auch externe Bibliothekssysteme transparent angebunden werden können.

Der in der 1. Förderphase entwickelte Demonstrator implementierte bereits viele Grundzüge des in der 2. Phase verbesserten Prototypen. Im Demonstrator war bereits das Volltext-Bibliothekssystem der Fakultätsbibliothek der TU München angebunden. Damit hatte das System Zugang zu ca. 80.000 Dokumenten dieses Systems. Es war weiterhin möglich auf vorhandene Textdokumente in der sog. Metadatenbank des Systems zuzugreifen, wobei diese Textdokumente bereits nachträglich verlinkt werden konnten.

Abbildung 1 zeigt die Architektur von OMNIS/2, die seit der ersten Förderphase gültig ist. Die Abbildung weist horizontal die bekannte Three-Tier-Architektur auf. In der vertikalen Sichtweise auf das System ist deutlich die Konzeption als Metasystem zu erkennen. Die

---

<sup>1</sup><http://www.cg.cs.tu-bs.de/v3d2/>

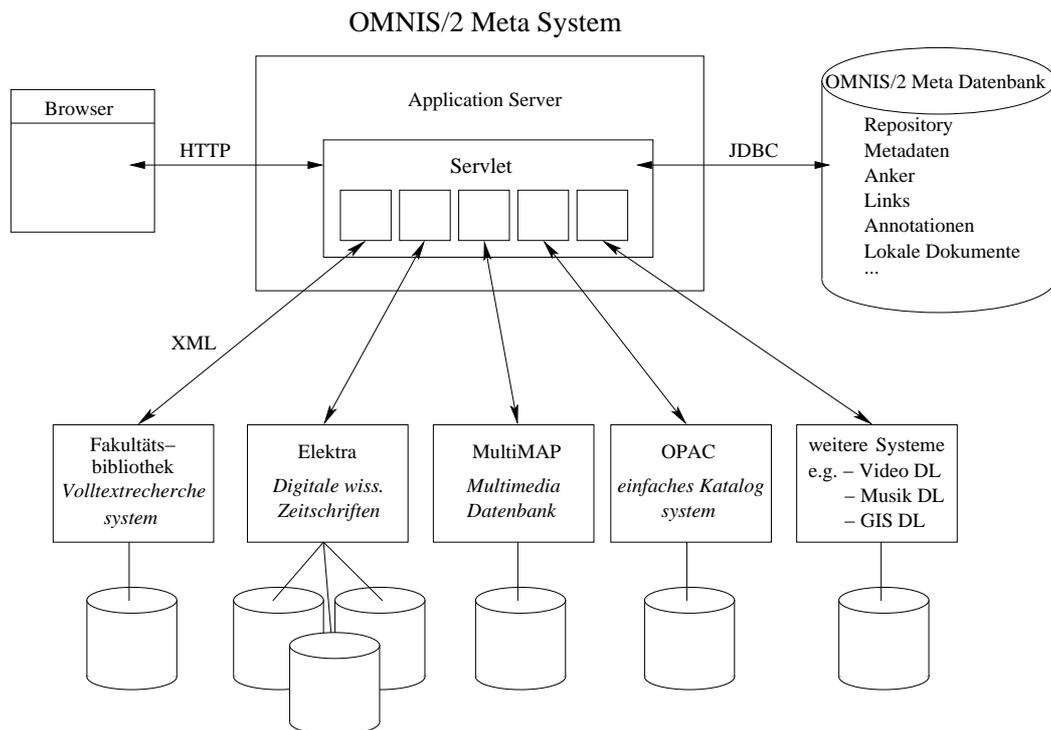


Abbildung 1: Architektur von OMNIS/2

obere Schicht bildet das als Metasystem konzipierte Kernsystem OMNIS/2, bestehend aus der eigentlichen Anwendung im Webserver und der Datenbank (sog. Meta-Datenbank), die sämtliche Information zu Ankern, Links und Metadaten speichert. Darunter liegen die bestehenden digitalen Bibliothekssysteme auf denen OMNIS/2 aufbaut und die es um die gewünschten Features erweitert.

### 1.3 Zusammenfassung der Projektarbeit der 2. Förderphase

In der zweiten Förderphase wurden alle Teile des in der 1. Förderphase entwickelten Demonstrators überarbeitet und konsequent weiterentwickelt. Dies geschah unter Beachtung der in der 1. Phase entwickelten Spezifikation.

Insbesondere wurden

- der Demonstrator zum operativen Forschungsprototypen weiterentwickelt,
- ein Austauschformat zur Anbindung externer Systeme entworfen,
- die nachträgliche Verlinkung der XML-Dokumenten mittels XSLT untersucht,
- ein Dokumentenmodell entwickelt und dieses auf Design Patterns abgebildet,
- das Ankerkonzept überarbeitet,
- ein Rechtekonzept für Dokumente und Links erstellt,
- ein OPAC-System eingebunden,

- Performanzuntersuchungen zur Speicherung der XML-Dokumente in XML-Datenbanksystemen unternommen [3], sowie
- die Datenbankanbindung optimiert.

Darüberhinaus wurden die Kooperationen mit den SPP-Teilprojekten AVAnTA (Prof. Herzog, Uni Bremen) und Paddle (Dr. Tochtermann, FAW Ulm) weitergeführt und vertieft. Insbesondere wurden dabei die Anbindungsmöglichkeiten der Systeme an OMNIS/2 für die 3. Förderphase vorbereitet.

In der zweiten Förderphase des Projekts entstanden acht Konferenzveröffentlichungen (jeweils eine Veröffentlichung auf den ECDL Konferenzen 2000 (Lissabon) und 2001 (Darmstadt), auf der DEXA 2000 in Greenwich, UK, auf der IDEAS 2000 in Yokohama, Japan, auf der WAIM 2000 in Shanghai, eine Veröffentlichung auf dem internationalen Workshop DLib 2001 in München, sowie zwei Veröffentlichungen auf den Net.ObjectDays 2001), zusätzlich zwei Technical Reports und zwei Diplomarbeiten.

## 1.4 Gliederung des Arbeitsberichts

Der folgende Arbeitsbericht ist in fünf größere Teile untergliedert. Wir beginnen mit der Weiterentwicklung des Demonstrators zum operativen Forschungsprototypen in Kapitel 2. Wir beschreiben hier das von uns entwickelte Austauschformat für die existierenden digitalen Bibliothekssysteme (2.1), anschließend unseren neu entwickelten Ansatz zur nachträglichen Verlinkung mittels XSLT (2.2). Danach folgt eine Beschreibung des Dokumentenmodells von OMNIS/2 (2.3), des Ankerkonzepts (2.4) und des Rechtekonzepts für Dokumente und Links (2.5). Am Beispiel des OPACs des Bibliotheksverbunds Bayern (BVB) beschreiben wir die Anbindung externer Bibliothekssysteme (2.6). Der erste Teil schließt mit einem kurzen Abriss über die Optimierung der Datenbankanbindung (2.7). Der zweite Teil beschäftigt sich mit den Kooperationen im SPP (Kap. 3). Dann folgt ein Ausblick auf den geplanten Projektfortgang in Kapitel 4. Der Bericht endet mit einer Veröffentlichungsliste und der DTD des zuvor beschriebenen Austauschformats im Anhang.

## 2 Weiterentwicklung des Demonstrators zum operativen Forschungsprototypen

Der in der ersten Phase entwickelte Demonstrator zeigte einen zum Beginn der 2. Förderphase unbefriedigenden Zustand. Zwar wurden die Konzepte und die Spezifikation der ersten Phase umgesetzt und implementiert, jedoch wurde sehr schnell klar, daß die Ziele der 2. Phase damit nur schwer zu erreichen sein werden. Wir entschlossen uns daher zu einer Anpassung der Implementierung des Demonstrators, wobei einerseits die Fähigkeiten erhalten bleiben sollten, andererseits aber die Möglichkeit einfacherer Erweiterungen gegeben sein sollte. Insbesondere wurden dabei das Dokumentenmodell und das Ankerkonzept deutlich überarbeitet (siehe unten). Notwendig und beantragt war auch die Spezifikation und Implementation eines generischen Austauschformats zur Anbindung externer Systeme.

## 2.1 Entwurf eines Austauschformats zur Anbindung externer Systeme

Ziel der Entwicklung des Austauschformats war es, die Voraussetzungen dafür zu schaffen, beliebige weitere Systeme konsistent und mit kalkulierbarem Aufwand anbinden zu können. Der Kern davon ist ein einheitliches internes Datenformat auf das alle Daten aus den externen Systemen abgebildet werden. Alle weiteren Schritte gehen dann von einem einheitlichen Format aus. An das interne Format stellten wir die folgende Anforderungen: Das Format sollte XML basiert sein, da bereits die angebotenen Systeme XML liefern. Das Format sollte mächtig genug sein um alle anschließenden internen Schritte (u.a. Verlinkung, Annotation) darauf zu ermöglichen. Es sollte flexibel sein um neue Systeme ohne Änderungen am internen Format anbinden zu können. Es sollte trotzdem aber noch leicht erweiterbar sein, falls sehr exotische Systeme angebotenen werden sollen, die vom bisherigen Format nicht abgedeckt werden.

Beim Entwurf des internen Formats haben wir auf das DUBLIN CORE METADATA ELEMENT SET (DCMES) der DUBLIN CORE METADATA INITIATIVE (DCMI) zurückgegriffen. Bei DCMES handelt sich um einen fachübergreifend einsetzbaren Metadatenstandard für „Resource Discovery“. Dieser Ansatz adressiert genau die Problematik bei der Anbindung externer Systeme an OMNIS/2, bei der es ebenfalls um die Auf- und Wiederfindung von elektronischen Ressourcen vielfältiger Art geht. Das von uns entwickelte interne Format von OMNIS/2 basiert auf den von der Initiative vorgeschlagenen 15 Metadatenfeldern. Erweiternd haben wir das interne Format so konzipiert, daß zusätzlich noch Informationen über Inhaltsdaten angegeben werden können. Die vollständige DTD des internen OMNIS/2-Formats ist in Anhang A aufgelistet.

## 2.2 Nachträgliche Verlinkung der XML Dokumente mittels XSLT

Um die Vorteile von OMNIS/2 nutzen zu können, müssen Benutzer auf alle Dokumente über OMNIS/2 zugreifen. Dies kann entweder über die erweiterten Suchfunktionen von OMNIS/2 geschehen (wobei hier in den darunterliegenden Systemen gesucht wird und die Ergebnisse nachträglich verlinkt werden) oder indem Links zu Dokumenten verfolgt werden, die wiederum von OMNIS/2 annotiert, bzw. nachträglich verlinkt werden. Ein Zugriff auf Dokumente über Linkverfolgung läuft wie folgt ab. Im ersten Schritt setzt das OMNIS/2 System eine Anfrage an die lokale Metadatenbank ab um festzustellen in welchem externen System das gewünschte Dokument liegt. Dies setzt voraus, daß jedes Dokument einen persistenten, eindeutigen Identifikator (UID) in dem darunterliegenden System besitzt. Für Dokumente aus der lokalen Datenbank kann dies leicht sichergestellt werden, bei anderen digitalen Bibliothekssystemen ist dies eine Voraussetzung, damit sie an OMNIS/2 angebotenen werden können. Nachdem diese Anfrage beantwortet wurde, wird die Metadatenbank ein zweites Mal angefragt um zu diesem Dokument Ankerinformation zu Links und Annotationen abzufragen. Wir verwenden hierbei Anker im Sinne des Amsterdam Hypermedia Modells und betrachten sie als eine Abstraktion von Linkquellen und Linkzielen. Neben ihrer exakten Position benötigen Anker ebenfalls eindeutige Identifikatoren (AIDs und UIDs) so daß sie eindeutig einem Dokument zugeordnet werden können. Nach dieser zweiten Anfrage wird der Hauptteil des nachträglichen Verlinkens durchgeführt. Dabei werden die Links zu den Dokumenten hinzugefügt und dem Benutzer präsentiert.

Ein wesentliches Ergebnis ist, daß XSLT, obwohl zu anderen Zwecken entwickelt, sich durch

Erweiterungen auch zum Zusammenfügen von XML Dokumenten und der Linkinformation eignet. Erstmals wurde dies in OMNIS/2 realisiert. Das W3C empfiehlt XSLT um XML Dokumente in andere XML Dokumente, HTML Dokumente oder ASCII Text zu transformieren. Die Manipulationen der XML Dokumente werden von einem XSLT Prozessor vorgenommen, der dazu ein XSLT Stylesheet abarbeitet. XSLT Prozessoren sind für eine Reihe von Programmiersprachen verfügbar (C++, Java, Perl, etc.). XSLT selbst bietet allerdings keine Möglichkeit Datenbankoperationen durchzuführen. Dies muß über die Programmiersprache geschehen mit der der XSLT Prozessor verwendet wird. Dabei können Parameter mit dem Stylesheet ausgetauscht werden. Diese Möglichkeit ist der Kernpunkt unserer Idee, wie man nachträgliches Verlinken für digitale Bibliotheken implementieren kann.

Wir haben drei Möglichkeiten der nachträglichen Verlinkung mittels XSLT untersucht und alle Möglichkeit evolutionär implementiert.

Die erste Technik bezeichnen wir als *einfaches Verlinken*. Hierbei wird in einem ersten Schritt mittels JDBC die Linkinformation aus der Metadatenbank abgefragt. Aus dem Zielanker wird nun die vollständige URL gebildet, die das verknüpfte Zieldokument referenziert (da dieses z.B. nähere Informationen enthält). Diese URL ist der Parameter, der an das XSLT-Stylesheet übergeben wird. Mit Hilfe des übergebenen Parameters erzeugt dann das XSLT-Stylesheet einen gültigen Link an der gewünschten Position.

Die zweite von uns untersuchte Technik ist die *nachträgliche Verlinkung von Teilstrings*, da die zuvor vorgestellte Technik nur die Verlinkung von vollständigen Tags ermöglicht. Bei dieser Technik muß nicht nur der Tag identifiziert werden, der verlinkt werden soll, sondern auch die exakte Position des Links. Die Anker müssen daher die Position des Substrings im Taginhalt bezeichnen. Im allgemeinen reicht dafür eine einfache Adressierung mit Integerwerten, die den Start und die Länge des Ankers, bzw. bei Multimedia-Objekten die Koordinaten des Ankers oder den Anfangspunkt und die Dauer bezeichnet.

Die von uns entwickelte dritte Technik läßt sich mit *dynamischer Verlinkung* bezeichnen: In vielen Fällen ist es nötig die Daten auch aus dem Stylesheet in das Javaprogramm zu übergeben (bis jetzt wurde nur der umgekehrte Fall beschrieben). Dies ist erforderlich um mehrere Tags oder Dokumentenlisten, wie sie z.B. nach Suchanfragen geliefert werden, zu verlinken. Außerdem ist es wünschenswert nur Teile eines XML Dokuments zu transformieren und dann die Werte diverser Parameter für das restliche Dokument zu ändern.

Die meisten XSLT-Prozessoren bieten sog. *Extensions* an, die es erlauben Programmiersprachen in XSLT zu verwenden. Wir verwenden dabei den XSLT-Prozessor Xalan aus dem Apache-Projekt, bei dem es möglich ist *Extensions* in Java zu verwenden. Indem am Beginn des Stylesheets ein neuer Namensraum festgelegt wird, wird es möglich Java Methoden in der gleichen Art und Weise aufzurufen, wie XSLT Funktionen. Da die Java Methoden mit beliebigen Parametern aufgerufen werden können, können Teile des XML-Dokuments an die Java Methoden übergeben werden, diese können die Daten entsprechend bearbeiten (z.B. Datenbankabfragen durchführen) und die Daten an das XSLT-Stylesheet für weitere Bearbeitungen zurückgeben. Dies erlaubt uns sehr viel flexibler mit XSLT zu arbeiten als zuvor und die gewünschten Ergebnisse zu erreichen.

Ein Papier über diese von uns entwickelte Technik wurde auf der ECDL 2001 in Darmstadt angenommen [1].

## 2.3 Das Dokumentenmodell

OMNIS/2 muß als Metasystem für existierende digitale Bibliothekssysteme viele verschiedene Dokumententypen unterstützen. Konzeptionell wird im System zwischen *externen* und *internen* Dokumenten unterschieden, je nachdem ob die Dokumente aus den darunterliegenden digitalen Bibliothekssystemen stammen oder aus der relationalen Metadatenbank. Weiterhin werden die Dokumente nach ihrer Struktur unterschieden. Das sind einerseits eigenständige Dokumente (sog. Atome) und andererseits zusammengesetzte Dokumente (sog. Composites). Composites können aus Atomen oder Composites bestehen und bilden damit eine hierarchische Struktur (genauer einen DAG). Diese von uns gewählte Unterteilung folgt dem Amsterdam Hypermedia Modell.

Atome sind entweder von einem einzelnen Typ oder können auch komplexerer Natur sein, wenn sie in dieser Form als ganzes in einem externen System zugänglich sind. In der aktuellen Version unseres Systems verwenden wir Text, Image, Audio, Video und Extern als eigenständige Dokumentenatome. Die externen Dokumente lassen sich weiter unterscheiden aufgrund der verschiedenen Dokumententypen, die aus den externen digitalen Bibliothekssystemen stammen. Diese Unterteilung impliziert bereits eine starke Kapselung der dokumentenspezifischen Merkmale eines Typs und die Möglichkeit alle Typen über eine gemeinsame Schnittstelle anzusprechen.

Diesen Überlegungen folgend entschieden wir uns für einen objektorientierten Entwurf. In dem von uns entwickelten Dokumentenmodell betrachten wir jedes Dokument als Objekt, das zur Laufzeit eigenständig existieren kann. Jedes Objekt wird identifiziert anhand eines persistenten eindeutigen Identifikators (UID). Die Objekte können sich auf Anforderung selbst aus der Datenbank aufbauen, zusätzlich tragen sie alle Information (Anker, Annotationen) über sich selbst und können sich selbst darstellen, d.h. sie können ihren Inhalt in einer Form präsentieren, der weiterbearbeitet oder angezeigt werden kann). Dies kann dazu genutzt werden, daß sich die Dokumente zusätzlich auch noch in anderer Form darstellen können (z.B. für das von uns geplante Autorensystem). Composite Dokumente stellen sich dar, indem sie den Aufruf an ihre Kinddokumente weitergeben (die wiederum die Fähigkeit haben sich selbst darzustellen).

Bei der Implementierung stützen wir uns auf das sog. Composite Design Pattern. Mittels dieses Design Patterns können Hierarchien nachgebildet werden und es bietet sowohl für Composites, als auch für Atome eine einheitliche Schnittstelle. Bild 2 zeigt das Design Pattern als UML-Diagramm.

In dem von uns entwickelten Prototypen stellen sich die Dokumente in XML dar. Wir setzen auf dieser Darstellung mittels XSLT auf und haben gleichzeitig die oben beschriebene Möglichkeit die Dokumente nachzubearbeiten, d.h. zu annotieren und zu verlinken. Die Ähnlichkeit der hierarchischen Struktur von XML zu unserer hierarchischen Dokumentenstruktur unterstützte uns dabei.

Die Entscheidung für dieses Konzept beschleunigte die weitere Entwicklung signifikant und bietet auch die Möglichkeit der einfachen Erweiterung.

Ein Papier über unser Dokumentenmodell wurde auf dem Workshop DLib2001 in München angenommen [4].

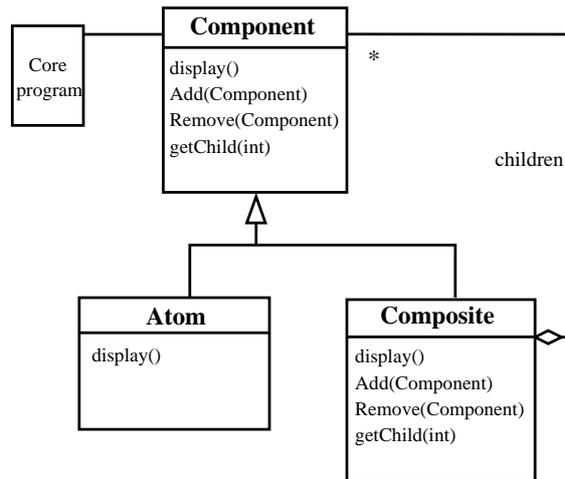


Abbildung 2: Composite Design Pattern (UML)

## 2.4 Das Ankerkonzept

Dokumente sind im System OMNIS/2 eigenständige Objekte, die alle nötige Information über sich selbst tragen. Zu diesen Informationen gehören auch Anker für Annotationen und Verlinkung. Wir betrachten Anker im Sinne der Definition des Amsterdam Hypermedia Modells als Abstraktion von Linkquellen und Linkzielen. Links als eigenständige Entitäten verbinden Quellanker und Zielanker. Das Anker- und Linkkonzept von OMNIS/2 ist mächtiger als das Linkkonzept im WorldWideWeb. Das System unterstützt 1:n Links, bidirektionale Links und überlappende Links.

Wir entschieden uns zur Realisierung des Ankermodells für einen ähnlichen Ansatz wie für das Dokumentenmodell in OMNIS/2. Im Gegensatz zu Dokumenten gibt es allerdings keinen Unterschied zwischen *externen* und *internen* Ankern, da alle Anker in der Metadatenbank gespeichert werden.

Für jeden Dokumententyp werden Anker mit entsprechender Ausprägung benötigt, da unterschiedliche Positionsangaben in unterschiedlichen Positionangaben in unterschiedlichen Medientypen nötig sind.

Zusätzlich zur exakten Positionsbeschreibung tragen die Anker einen eindeutigen Identifikator (AID), damit eine eindeutige Dokumentzuordnung möglich ist.

Wir haben uns bei der Konzeption des Anker- und Linkkonzepts entschieden, die Anker nicht als Bestandteil der objektorientierten Dokumentenhierarchie (als besonderer Dokumententyp) zu betrachten. Das Amsterdam Hypermedia Model würde eine solche Konstruktion zulassen, was zu Konstruktionen wie Links auf Links führen würde. Eigene Untersuchungen im Vorgängerprojekt MultiMAP haben gezeigt, daß diese zusätzliche Möglichkeit weder von Autoren noch von Anwendern genutzt wird. Wir entschieden uns eine eigene Ankerhierarchie zu entwerfen und Anker als selbständige Objekte zu konzipieren. Dies führt (ähnlich wie bei Dokumenten) dazu, daß sich Anker selbst aus der Datenbank aufbauen können und sich unabhängig von den Dokumenten speichern können. Abbildung 3 zeigt einen Teil unseres leicht erweiterbaren Ankerkonzepts in UML. Zur Vereinfachung werden nur drei Ankertypen dargestellt.

Auch über das sich selbst aufbauende Ankerkonzept konnte ein Papier veröffentlicht werden (Proc. of NetObjectDays 2001 in Erfurt) [2].

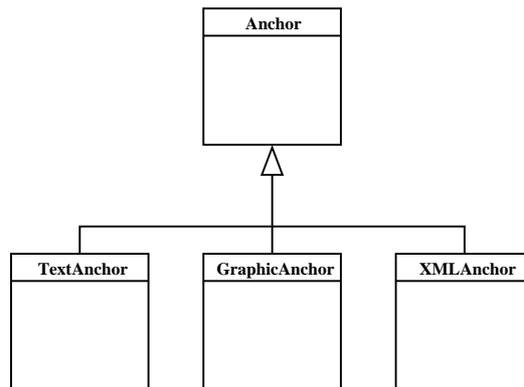


Abbildung 3: Klassendiagramm für Anker (UML)

## 2.5 Rechtekonzept für Dokumente und Links

Der Zugang zum OMNIS/2 System stützt sich auf ein Benutzerkonzept ab, d.h. Benutzer müssen sich explizit mit Namen und Paßwort anmelden um Zugang zum System zu erlangen. Ab diesem Zeitpunkt befinden sich Benutzer in einer Sitzung (Session) und können über mehrere Aktionen hinweg eindeutig identifiziert werden bis sie sich explizit ausloggen und damit die Sitzung beenden. Das Sessionmanagement ist über Cookies realisiert. Diese Entscheidung fiel, da ein Sessionmanagement mit Cookies sehr einfach durch die von uns verwendete Servlet-Engine unterstützt wird.

Das Sessionmanagement erlaubt uns darüberhinaus ein Rechtekonzept für das gesamte OMNIS/2 System einzuführen, sowohl für die Dokumente als auch für die Anker. Die Benutzer werden dabei Teil einer Benutzergruppe und können selbst festlegen, ob von ihnen erstellten Dokumente und Anker entweder für sie selbst, für ihre Benutzergruppe oder für alle lesbar oder schreibbar sein sollen.

## 2.6 Anbindung der darunterliegenden Bibliothekssysteme

Bereits in der ersten Förderphase wurde mit dem OMNIS System ein Bibliothekssystem durch XML an OMNIS/2 angebunden. In der zweiten Förderphase war unser Ziel zu zeigen, daß auch OPACs als weitverbreitete Systeme einfach anzubinden sind. Der von uns gewählte OPAC des Bibliotheksverbands Bayern (BVB) verzeichnet ca. 10 Millionen Titelsätze und bietet damit einen sehr wertvollen Ansatzpunkt für OMNIS/2. Da es für OPACs im allgemeinen keine standardisierte Schnittstelle gibt, haben wir uns entschieden den OPAC des BVB über den Information Broker ElektraII anzusprechen. ElektraII ist ein Information Broker, der mehrere digitale Bibliotheken als Datenquelle nutzt (derzeit bereits mehr als 30 Systeme) und einheitliche Schnittstellen für die Arbeit mit diesen Systemen bietet. Zum Zugriff auf den Elektra Broker stehen verschiedene Schnittstellen bereit, unter anderem ein XML-Client, der auf HTTP-Anfragen XML-Dokumente als Antwort liefert. ElektraII wurde an unserem Lehrstuhl entwickelt und ist seit Mai 2000 ein kommerzielles Produkt der Firma Sisis<sup>2</sup>.

Parallel zur Entwicklung des oben beschriebenen Austauschformats erfolgte eine Umstrukturierung der Suche in den darunterliegenden digitalen Bibliothekssystemen. Unsere Absicht bei dieser Entwicklung war die Suche so zu kapseln, daß die Kommunikation mit

<sup>2</sup><http://www.sisis.de>

externen Bibliothekssystemen über definierte Schnittstellen aber unter Verwendung eines einheitlichen Formats erfolgt. Zusätzlich kann mit dem internen Format sichergestellt werden, daß Ergebnisse einheitlich verarbeitet werden.

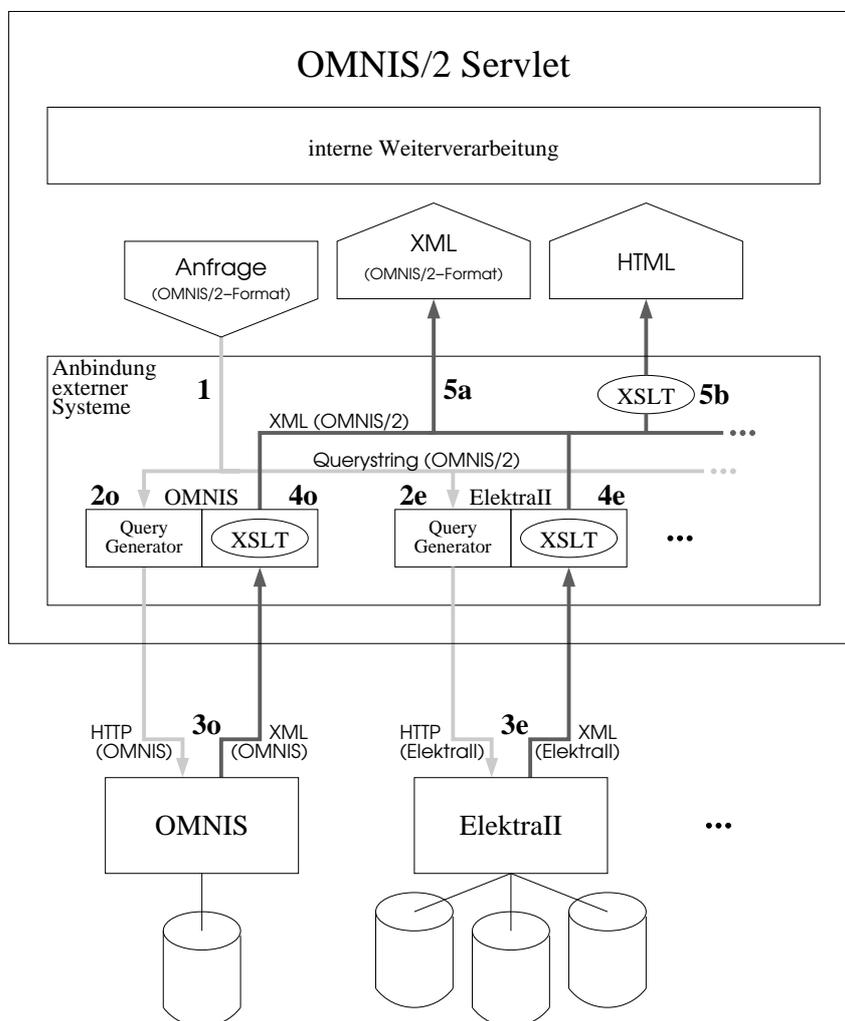


Abbildung 4: Interne Abläufe bei Anfragen aus dem Servlet nach Dokumenten aus externen Systemen

Abbildung 4 zeigt die Architektur der Anbindung externer Systeme. Der in der Grafik dargestellte Ablauf ist im folgenden anhand der einzelnen Schritte ausführlich beschrieben (die Zusätze *o*, bzw. *e* bezeichnen dabei die Schritte für Omnis, bzw. das ElektraII System).

1. Auslieferung der Anfrage an die zuständigen Komponenten: Der erste Schritt nach der Eingabe einer Anfrage ist die Auslieferung der Anfrage an die für die entsprechenden angebotenen Systeme zuständigen Komponenten der Anbindung. Eine Anfrage kann dabei sowohl an ein spezielles externes System gerichtet sein als auch an beliebig viele Systeme gleichzeitig.
2. Query-Generierung: Schritt 2 (2o bzw. 2e) stellt die Query-Generierung dar, die die Anfrage, die für alle Systeme im einheitlichen OMNIS/2-Anfrageformat vorliegt, in Querystrings im speziellen Format des entsprechenden Systems umwandelt. Bei den

beiden bisher angebotenen Systemen werden die Querystrings dabei in HTTP-Requests enkodiert.

3. Bearbeitung der Anfrage durch das externe System: Schritt 3 (3o bzw. 3e) stellt die Bearbeitung der Anfrage durch das externe System und die Rücklieferung der Antwort an OMNIS/2 dar. In beiden Fällen werden dabei XML-Dokumente geliefert.
4. Umwandlung der Anfrage in das interne OMNIS/2-Format: Schritt 4 (4o bzw. 4e) stellt die Umwandlung des XML-Dokuments aus dem Format des externen Systems in das interne OMNIS/2-Format dar. Für diese Aufgabe wird ein XSLT-Prozessor eingesetzt, der das Dokument mittels eines Stylesheets in das neue Format umwandelt.
5. Aufsammlung der Ergebnisse und Rücklieferung an die anfragende Komponente des Servlets: Schritt 5 beinhaltet die Aufsammlung der Ergebnisse aus den einzelnen Systemen, falls die Anfrage an mehr als eine Datenbank gerichtet war. Wird eine Rücklieferung der Ergebnisse im internen Format gewünscht, wird eine Liste mit den Ergebnisdokumenten an die anfragende Komponente des Servlets geliefert (Fall 5a). Fall 5b zeigt die Vorgehensweise, falls eine Rücklieferung im HTML-Format gewünscht wird: Die einzelnen Ergebnisdokumente im internen Format werden nochmals von einem XSLT-Prozessor verarbeitet und in HTML-Code umgewandelt, bevor die Liste mit den Antworten zurückgeliefert wird.

Diese Technik läßt sich auch für die weiteren Anbindungen einsetzen. Mit der hier realisierten Anbindung des OPAC des Bibliotheksverbunds Bayern mit 10 Millionen Titelsätzen zusätzlich zur bestehenden Anbindung der Fakultätsbibliothek Mathematik-Informatik der TU München wurde eine kritische Masse für eine gewinnbringende Nutzung des OMNIS/2 Systems erreicht.

## 2.7 Optimierung der Datenbankanbindung

In der ersten Förderphase wurde für die Architektur eine Realisierung als Java-Servlet festgelegt. Diese Entscheidung hatte großen Einfluß auf die Wahl der Datenbankanbindung. In der zweiten Förderphase wurde die Datenbankanbindung als sog. Pool-Lösung realisiert. Dabei öffnet das Servlet bei der Initialisierung sofort eine festgelegte Anzahl an Verbindungen (in unserem Fall fünf Verbindungen) und verwaltet diese in einem Pool. Das bedeutet, daß bei einer Datenbankoperation zuerst eine Verbindung aus dem Pool angefordert werden muß und sich somit die Zahl der verfügbaren Verbindungen im Pool reduziert. Nach Abschluß der Operation legt das Programm seine angeforderte Verbindung in den Pool zurück. Die Verwaltung als Pool bietet große Geschwindigkeitsvorteile, da nicht immer neue Verbindungen erzeugt werden müssen. Außerdem ist es möglich die Anzahl der offenen Verbindungen für spezielle Anforderungen (z.B. technische Beschränkungen der JDBC-Treiber, Lizenzbedingungen) zu begrenzen.

Bei der Implementierung des Poolkonzepts konzipierten wir den Pool mittels des Singleton Design Patterns. Damit wird sichergestellt, daß im ganzen System nur exakt ein Pool an Verbindungen existiert.

### 3 Kooperationen

Bereits seit der ersten Förderphase bestehen enge Kooperationsbeziehungen zu den Projekten AVAnTA (Prof. Herzog, Uni Bremen) und Paddle (Dr. Tochtermann, FAW Ulm). Im Bremer Projekt wird ein inhaltsbasiertes Videoanalyzesystem 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. Dies soll in der dritten Förderphase implementiert werden.

Das Teilprojekt Paddle der FAW Ulm entwickelt eine benutzerdefinierbare Umgebung für digitale geographische Bibliotheken. In mehreren Projekttreffen in der 2. Förderphase wurden Gemeinsamkeiten identifiziert. Es wurde begonnen eine Anbindung der Datensätze aus Paddle zu erarbeiten.

### 4 Ausblick

Im weiteren Projektfortgang soll insbesondere der entstandene Prototyp zu einem vollständigen System fertiggestellt werden. Darüberhinaus eignet sich OMNIS/2 hervorragend als Grundbaustein innerhalb der inzwischen entstandenen OpenArchives Initiative<sup>3</sup>. Während diese erst erste Spezifikationen verabschiedet hat, haben wir mit OMNIS/2 wesentliche Teile davon bereits implementiert, so daß sich mit den noch nötigen Erweiterungen schnell ein fertiger OpenArchives Service Provider erstellen ließe. Damit könnte OMNIS/2 zusätzlich in einem zweiten, völlig neuen Bibliothekskontext eingesetzt werden.

### 5 Liste der Publikationen aus diesem Projekt

#### Publikationen in der 2. Förderphase 1999-2001

- [1] Bauer M.G., Specht G., *Enhancing Digital Library Documents by A Posteriori Cross Linking Using XSLT*, Proc. of the 5th European Conf. on Research and Advanced Technology for Digital Libraries (ECDL 2001), Darmstadt, Springer, LNCS, 2001, pp. 95-102.
- [2] Bauer M.G., Specht G., *The Anchor and Linking Concept of a Meta System for Existing Digital Libraries*, 7th MIK-Workshop 2001, Proc. of Netobjectdays 2001, Erfurt, pp. 260-265.
- [3] Schindler R., Specht G., *Performanz von XML-Datenbanksystemen*, 7th MIK-Workshop 2001, Proc. of Netobjectdays 2001, Erfurt, pp. 254-259.

---

<sup>3</sup><http://www.openarchives.org>

- [4] Bauer M.G., Specht G., *The Object Oriented Document Model of a Meta System for Existing Digital Libraries*, First International Workshop on Digital Libraries 2001, Proc. of DEXA 2001, München, IEEE Press, pp. 933-936.
- [5] Specht G., Bauer M.G., *OMNIS/2: A Multimedia Meta System for existing Digital Libraries*, Proc. of the 4th European Conf. on Research and Advanced Technology for Digital Libraries (ECDL 2000), Lissabon, Springer, LNCS 1923, 2000, pp. 180-189.
- [6] Specht G., Kahabka T., *Information Filtering and Personalisation in Databases using Gaussian Curves*, Proc. of the IEEE 4th International Database Engineering and Applications Symposium (IDEAS 2000), Yokohama, Japan, IEEE Computer Society Press, pp. 16-24.
- [7] Specht G., Zoller P., *Modelling Temporal Aspects in Hypermedia Applications*, ACM First Int. Conf. on Web-Age Information Management (WAIM 2000), Shanghai, China, Springer, LNCS 1846, 2000, pp. 259-270.
- [8] Specht G., v.Harsdorf P., Kahabka T., Peterander F., *Awareness in interactive Database Applications*, Proc. of the 11th Int. Conf. on Database and Expert System Applications (DEXA 2000), Greenwich UK, Springer, LNCS 1873, pp. 203-211.
- [9] Specht G., Bauer M.G., *Multimedia Beitrag: OMNIS/2 auf der "V3D2 Multimedia CD-ROM, Verteilte Verarbeitung und Vermittlung digitaler Dokumente"* Digitale Bibliotheksinitiative der Deutschen Forschungsgemeinschaft, Version 1.0, 22.9.2000, Editor: Institut für Computergraphik, TU Braunschweig.
- [10] Specht G., Bauer M.G., *Systemarchitektur und Funktionalität des multimedialen digitalen Meta-Bibliothekssystems OMNIS/2*, Technischer Bericht, TUM-I9924, Technische Universität München, 1999, 18 S.

## Diplomarbeiten

- [1] Krammer A., *Anbindung multimedialer digitaler Bibliothekssysteme über XML an das Metasystem OMNIS/2*, Diplomarbeit, Institut für Informatik, TU München, 2001.
- [2] Schmatloch A., *Nutzung von XML zum universellen Datenaustausch zwischen Applikationen*, Diplomarbeit, Fakultät für Informatik und Automatisierung, TU Ilmenau, 2000.

Publikationen aus der 1. Förderphase siehe Technischer Bericht TUM-I9924 [10]  
(5 referierte Publikationen).

## A Anhang: DTD des internen OMNIS/2-Formats

```

1      <!-- omnis2_ns.dtd -->
2      <!-- XML Document Type Definition of the internal OMNIS/2 XML-Format -->
3      <!-- Requires Namespace-Prefix "o2:" -->
4
5      <!-- Root-Element -->
```

```

6      <!ELEMENT o2:SearchResult      (o2:numberOfHits,
7                                     o2:fromSystem,
8                                     o2:nextResultsURL?,
9                                     o2:items?)>
10
11     <!-- Total number of hits for the Query,
12           maybe larger than the number of elements returned
13           in this document (this resultpage) -->
14     <!ELEMENT o2:numberOfHits      (#PCDATA)>
15
16     <!-- Name of the system this data comes from -->
17     <!ELEMENT o2:fromSystem        (#PCDATA)>
18
19     <!-- URL to call the next result page -->
20     <!ELEMENT o2:nextResultsURL    (#PCDATA)>
21
22     <!-- List of results -->
23     <!ELEMENT o2:items             (o2:item+)>
24
25     <!-- Single result from the resultlist -->
26     <!ELEMENT o2:item              (o2:meta,
27                                     o2:content?)>
28
29     <!-- Metadata -->
30     <!ELEMENT o2:meta              (o2:dc)>
31
32     <!-- Standard Dublin Core Metadatenatz (DCMES),
33           see http://purl.org/DC/documents/rec-dces-19990702.htm -->
34     <!ELEMENT o2:dc ((o2:title | o2:creator | o2:subject | o2:description |
35                     o2:publisher | o2:contributor | o2:date | o2:type |
36                     o2:format | o2:identifier | o2:source | o2:language |
37                     o2:relation | o2:coverage | o2:rights )*)>
38
39     <!ELEMENT o2:title              (#PCDATA)>
40     <!ELEMENT o2:creator            (#PCDATA)>
41     <!ELEMENT o2:subject            (#PCDATA)>
42     <!ELEMENT o2:description        (#PCDATA)>
43     <!ELEMENT o2:publisher          (#PCDATA)>
44     <!ELEMENT o2:contributor        (#PCDATA)>
45     <!ELEMENT o2:date               (#PCDATA)>
46     <!ELEMENT o2:type               (#PCDATA)>
47     <!ELEMENT o2:format             (#PCDATA)>
48     <!ELEMENT o2:identifier         (#PCDATA)>
49     <!ELEMENT o2:source             (#PCDATA)>
50     <!ELEMENT o2:language           (#PCDATA)>
51     <!ELEMENT o2:relation           (#PCDATA)>
52     <!ELEMENT o2:coverage           (#PCDATA)>
53     <!ELEMENT o2:rights             (#PCDATA)>
54
55     <!-- Attributes to save qualified Dublin Core (qDC). The following

```

```

56         attributes are allowed for each element:
57         refinement: Refinement of the element meaning
58         encoding: Encoding Scheme used for the element -->
59
60     <!ATTLIST o2:title           refinement    CDATA    #IMPLIED>
61     <!ATTLIST o2:title           encoding      CDATA    #IMPLIED>
62
63     <!ATTLIST o2:creator         refinement    CDATA    #IMPLIED>
64     <!ATTLIST o2:creator         encoding      CDATA    #IMPLIED>
65
66     <!ATTLIST o2:subject         refinement    CDATA    #IMPLIED>
67     <!ATTLIST o2:subject         encoding      CDATA    #IMPLIED>
68
69     <!ATTLIST o2:description     refinement    CDATA    #IMPLIED>
70     <!ATTLIST o2:description     encoding      CDATA    #IMPLIED>
71
72     <!ATTLIST o2:publisher       refinement    CDATA    #IMPLIED>
73     <!ATTLIST o2:publisher       encoding      CDATA    #IMPLIED>
74
75     <!ATTLIST o2:contributor     refinement    CDATA    #IMPLIED>
76     <!ATTLIST o2:contributor     encoding      CDATA    #IMPLIED>
77
78     <!ATTLIST o2:date            refinement    CDATA    #IMPLIED>
79     <!ATTLIST o2:date            encoding      CDATA    #IMPLIED>
80
81     <!ATTLIST o2:type            refinement    CDATA    #IMPLIED>
82     <!ATTLIST o2:type            encoding      CDATA    #IMPLIED>
83
84     <!ATTLIST o2:format          refinement    CDATA    #IMPLIED>
85     <!ATTLIST o2:format          encoding      CDATA    #IMPLIED>
86
87     <!ATTLIST o2:identifier       refinement    CDATA    #IMPLIED>
88     <!ATTLIST o2:identifier       encoding      CDATA    #IMPLIED>
89
90     <!ATTLIST o2:source          refinement    CDATA    #IMPLIED>
91     <!ATTLIST o2:source          encoding      CDATA    #IMPLIED>
92
93     <!ATTLIST o2:language        refinement    CDATA    #IMPLIED>
94     <!ATTLIST o2:language        encoding      CDATA    #IMPLIED>
95
96     <!ATTLIST o2:relation        refinement    CDATA    #IMPLIED>
97     <!ATTLIST o2:relation        encoding      CDATA    #IMPLIED>
98
99     <!ATTLIST o2:coverage        refinement    CDATA    #IMPLIED>
100    <!ATTLIST o2:coverage        encoding      CDATA    #IMPLIED>
101
102    <!ATTLIST o2:rights          refinement    CDATA    #IMPLIED>
103    <!ATTLIST o2:rights          encoding      CDATA    #IMPLIED>
104
105    <!-- Supplied content: -->

```

```

106 <!ELEMENT o2:content          (o2:inline*,
107                               o2:link*)>
108
109 <!-- Content-Data supplied 'inline'.
110      possible formats: XML, Text conforming to the XML-rules,
111      unparsed Text in a CDATA-Section. -->
112 <!ELEMENT o2:inline          (#PCDATA)>
113
114 <!-- Attribute for the element inline:
115      type: optional attribute for the type of the data -->
116 <!ATTLIST o2:inline          type          CDATA   #IMPLIED>
117
118 <!-- Link to externally available content: -->
119 <!ELEMENT o2:link            (#PCDATA)>
120
121 <!-- Attributes for the element link:
122      type: optional attribute for the type of the data
123      url:  required attribute for the URL of the link target -->
124 <!ATTLIST o2:link            type          CDATA   #IMPLIED>
125 <!ATTLIST o2:link            url          CDATA   #REQUIRED>

```