

The Anchor and Linking Concept of a Meta System for Existing Digital Libraries

Michael G. Bauer
Institut für Informatik, TU München
Orleansstraße 34, D-81667 München, Germany
bauermi@in.tum.de

Günther Specht
Fachgebiet Datenbanken und Informationssysteme
Fakultät für Informatik und Automatisierung, TU Ilmenau
Postfach 100565, D-98684 Ilmenau, Germany
gspecht@prakinf.tu-ilmenau.de

Abstract.

In this paper we describe the anchor and linking concept of a meta system for existing digital library systems. We show how we manage anchors and links, which are stored separately from the actual documents in a meta system, in an object oriented manner.

Our approach is already implemented in the system OMNIS/2, which is an advanced meta system for existing digital library systems and enhances existing digital library systems or retrieval systems by additional storing and indexing of user-defined multimedia documents, automatic and personal linking concepts, annotations, filtering and personalisation.

1 Introduction

Nowadays many digital library systems exist which store a variety of information. This information is usually available and stored in many different media types. The systems as a whole emerged into established tools and the users in a simplified view received systems with powerful retrieval capabilities, but still miss features that would improve their ability to work with documents in digital library systems as it is common with books printed on paper (i.e. adding references, marking pages and annotating text). Some of these ideas were approached by some systems over the last decade, but a cross platform solution was always out of scope. This led us to the development of the OMNIS/2 system¹ which is a

¹OMNIS/2 is funded by DFG (German Research Foundation) within the research initiative "V3D2" ("Distributed Processing and Delivery of Digital Documents") and is part of the Global Inventory Project (an initiative of the G7-countries).

meta system for various existing digital libraries [6]. It equips users with a tool that enables them to search several systems at once and especially to benefit from links between documents of different digital library systems (cross linking of documents). Users are able to add links by themselves by using an authoring tool that is part of OMNIS/2 and we will soon offer links that are generated automatically whenever possible (e.g. for bibliographic references or keywords). These features are often asked for by research groups who work with specialized library systems (e.g. VD17, a digital library of all German printings of the 17th century, a former project) [3]. Merging links and documents does not require a modification of the source documents and can even be done without harvesting, which would collect all documents and store the documents in an own huge database. This is achieved by a technique which adds the links at run-time using XSLT [7] right before the documents are presented to the user. We call this technique a posteriori cross linking [1]. In addition it is possible to annotate external documents without a write permission for the library systems which hold the annotated documents. The annotations are not limited to text and can consist of multimedia documents. Users are able to use a personalisation feature to create their own view on the documents and to “work” with digital library systems by themselves. In its current version the system has already access to some 80000 documents of the digital library system of the Faculty for Computer Science at TU München.

Besides the document model, which we already published [2], the anchor concept is a very important component of the OMNIS/2 system. In combination with the document model it provides the above mentioned features (as a posteriori crosslinking, annotations, personalisation) and ensures the separation of anchors and documents.

2 The Object Oriented Document Model

For the sake of completeness we repeat the concept of the object oriented document model in a few words.

The document model of OMNIS/2 distinguishes four different types of documents. Categorizing the documents according to source, we distinguish internal and external documents. We call the documents from the underlying digital library systems external documents and refer to the documents uploaded by the users, i.e. userdefined documents and annotations (texts, graphics, etc.) as internal documents. In addition we also categorize our documents according to structure. The document model therefore distinguishes another two types of documents, composites and atoms, thereby following the Dexter Hypertext Model [5]. Composites either consist of one or more atoms, one or more composites or both. A composite can not exist for itself, but always has to contain at least one atom. The four types are not disjunctive, consequently composites are internal documents and atoms are either local or external.

This results in a hierarchical document structure (see Fig. 1 for an example). This hierarchy is actually a DAG (directed acyclic graph), thus atoms and

composites can be shared within the same hierarchy and can occur in several different levels.

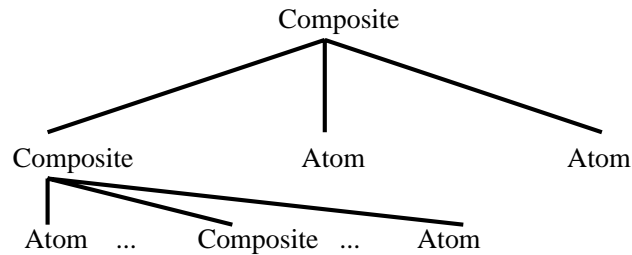


Fig 1. Example of a hierarchical document structure

In the current system we consider text, images, audio, video and external (i.e. the external documents) as valid document types. The external documents are further divided into the various document types originating from the integrated external digital library systems.

This basic framework leads to an object oriented design for the specification and implementation of the document model. In our model every document is seen as an object and can exist for its own at runtime. Every object is identified by a persistent unique identifier (UID). The document objects have the ability to create themselves from the database on request. In addition they carry all information about themselves (i.e. anchors, annotations) and can display themselves for the presentation to the user.

A very suitable method to implement our document model uses the composite design pattern (a structural design pattern) [4]. This design pattern enables us to represent part-whole hierarchies using objects and also provides a uniform interface to both composites and atoms.

3 Anchors and Links within OMNIS/2

In our context anchors follow the definition of the Dexter Hypertext Model [5] and in general are an abstraction of link sources and link destinations, while links connect source anchors and destination anchors. The anchoring mechanism is required to address and to refer to locations within the content of an individual document. The anchor and linking concept of OMNIS/2 is quite powerful. It supports 1:n links, bidirectional links and overlapping links. This link concept is apparently more powerful than the one currently used in the WorldWideWeb and requires appropriate management of anchors of course. As explained above already it is a very strong additional requirement that the anchors are separated from the actual documents (which makes the OMNIS/2 system special), as there is usually no access to the underlying systems.

We decided to use a similar approach to anchors as we used for the document model. The major difference though is that we store all anchors in our relational meta database, so there are no *external* anchors. For each type of document the

anchors have to have the matching characteristics (i.e. for text documents there are text anchors). This encapsulation is necessary as for every document type the position description of the anchor can be completely different. Text anchors can be easily identified by simply using position and length attributes. For anchors in pictures or graphics we decided to use simple geometric descriptions (e.g. polygons, circles with center and radius, etc.). In XML documents the anchors describe a position within an XML document. Using XPATH-style syntax to address parts of the document is very flexible and very suitable for our requirements.

Besides their exact position the anchors also need unique identifiers (AIDs) so that they can be associated with a document.

Strictly following the ideas of our object oriented document model it would have been possible to make the anchors part of the object hierarchy and to treat them as special occurrences of documents. This would have enabled us to create structures which contain links on links (next to links on documents). We decided against this possibility although it would have been possible following the Amsterdam Hypermedia Model. We consider this feature as not important for a real world application and will not implement it. The anchors form their own object hierarchy which can be seen in figure 2. As for documents we again put great effort into creating stand-alone objects. This results in anchors that create themselves from the database and have the ability to store themselves independently from the documents.

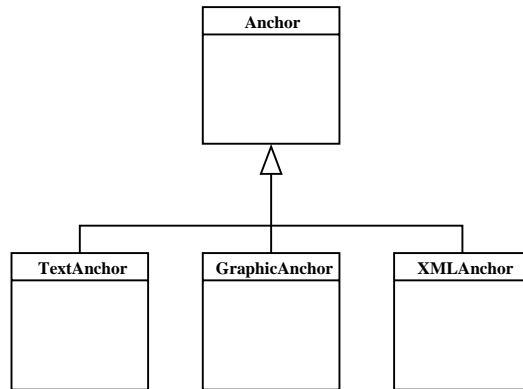


Fig 2. Class Diagram for anchors (UML)

Combining these different features we get a flexible concept which allows us to enrich documents by using 1:n links, bidirectional links and overlapping links.

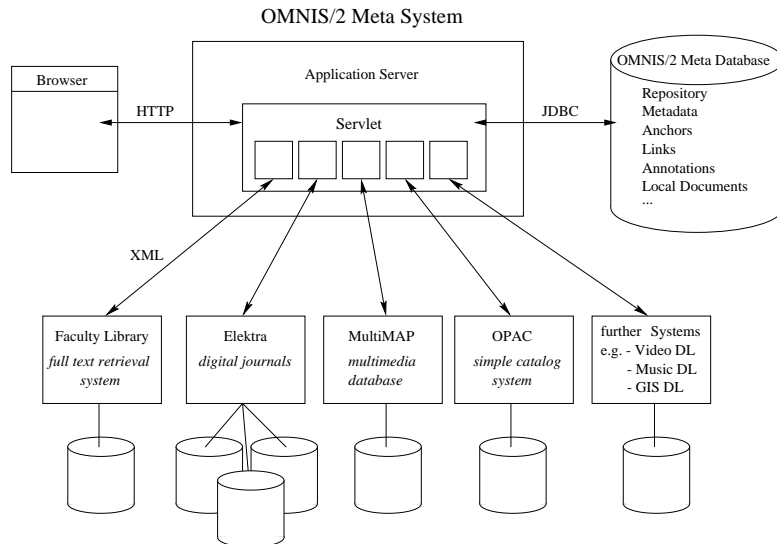


Fig 3. Architecture of OMNIS/2

4 Architecture of OMNIS/2

The above mentioned object oriented document model (section 2) in conjunction with the anchor and linking concept is already implemented and used in our meta system OMNIS/2. In the following we briefly describe its architecture.

OMNIS/2 is separated basically into two layers (see Fig. 3). We use established digital library systems as data providers and treat them as large containers with powerful query languages. We assume that the systems use XML to make their data available to the outside world as it is common for modern systems nowadays. If no XML interface is available it would also be possible to access the digital library systems through a tight coupling by using certain ports the systems provide. Our system sits on top of the established digital library systems as a meta system and acts as a service provider to the users, who access OMNIS/2 through a common browser. We decided to implement the meta system as a Java servlet in the Apache webserver. The meta system stores and manages all of the linking information and also the annotations, i.e. multimedia documents that users created by themselves, in its own relational database. With referential integrity dangling links are avoided within the OMNIS/2 system if referenced user-defined documents are removed. The system itself is not only a meta system for digital library systems, but it can also be used as a stand-alone multimedia digital library system as all features are available as well for user-defined documents.

5 Summary

The separation of content (i.e. documents) and anchors is an important feature for annotating and linking documents from existing digital library systems. We presented our anchor and linking concept which fits into a previously developed object oriented document model while preserving flexibility and extensibility.

Additionally we presented the meta system OMNIS/2 which provides the environment in which the anchor and linking concept is successfully used in conjunction with the object oriented document model.

References

- [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), Springer, LNCS, 2001.
- [2] 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, IEEE Press.
- [3] Dörr M., Haddouti H., Wiesener S., *The German National Bibliography 1601-1700: Digital Images in a Cooperative Cataloging Project*, Proc. of ADL'97, Washington DC, IEEE Computer Society, 1997, pp. 50-55.
- [4] Gamma E., Helm R., Johnson R., Vlissides J., *Design Patterns - Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- [5] Halasz F., Schwartz M., *The Dexter Hypertext Reference Model*, Comm. of the ACM, 37(2), Feb. 1994, pp. 30-39.
- [6] 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), Springer, LNCS 1923, 2000, pp. 180-189.
- [7] *XSL Transformations (XSLT), Version 1.0*, <http://www.w3.org/TR/xslt>