Journal of Universal Computer Science, vol. 1, no. 5 (1995), 269-291
submitted: 31/3/95, accepted: 23/5/95, appeared: 28/5/95, *©Springer Pub. Co.*

HOME: an Environment for Hypermedia Objects

Erik Duval
(Departement Computerwetenschappen, Katholieke Universiteit Leuven, Belgium
Erik.Duval@cs.kuleuven.ac.be)

Henk Olivié
(Departement Computerwetenschappen, Katholieke Universiteit Leuven, Belgium
olivie@cs.kuleuven.ac.be)

Piers O'Hanlon,
(Audio-Visual Service, University College London, England
piers@livenet.ac.uk)

David G. Jameson
(Audio-Visual Service, University College London, England
g.jameson@ucl.ac.uk)

Abstract: In this paper, we present HOME, a new environment for distributed hypermedia. We mainly concentrate on the server side, and provide access to World-Wide Web clients through a gateway mechanism. Data and metadata are strictly separated in the distributed HOME server. The architecture is based on a layered approach with separate layers for raw data, multimedia characteristics and hypermedia structure. We briefly present some of the implementation aspects and emphasise distinctive characteristics of HOME. We conclude with a comparison with related research and our plans for the future.
Key Words: HOME, distributed hypermedia, networked multimedia, image store, navigation, query
Category: H.5.1, I.7.2

1 Introduction

The explosion of interest from the general public in Gopher, WAIS the World-Wide Web (W.W.W.) [Berners-Lee, et al. 94] [Cailliau 95] and other Networked Information Discovery and Retrieval (NIDR) systems [Obraczka, et al. 93] indicates that there is a widespread demand for the functionality of such systems. As has been pointed out before [Andrews, et al. 94], the current generation of NIDR systems suffers from a number of problems.

We have been working for some time on a networked hypermedia system, that is in some sense similar in scope to the W.W.W. and Gopher, but that is based on a rather different design and data model. This paper will present our system, called HOME (for Hypermedia Object Management Environment).

2 Background

The authors have collaborated on the development of a networked hypermedia system in a number of European projects, funded within the framework of the DELTA (*D*eveloping *E*uropean *L*earning through *T*echnological *A*dvance) program [Duval, Olivié 94] [Jameson, et al. 93] [Beckwith, et al. 93]:

- The aim of the project CAPTIVE ('*C*ollaborative *A*uthoring, *P*roduction and *T*ransmission of *I*nteractive *V*ideo for *E*ducation', 1989 to 1991) was to develop an infrastructure for collaborative development of multimedia educational resources. An image store and a database *about* such resources were designed, implemented and integrated in a European telecommunication infrastructure, based on Direct Broadcast Satellites (DBS), an analogue video network, Integrated Services Digital Networks (ISDN) and the Internet.
- The MTS project ('*M*ultimedia *T*ele*S*chools', 1992 to 1994) involved large-scale experiments with delivery of interactive sessions, using a combination of DBS transmission from a conventional studio and ISDN feedback from students located throughout Europe. In the ACT project ('*A*dvanced *C*ommunication *T*echnology', also 1992 to 1994) technology was developed for large-scale field experiments in the MTS project.

Underlying most of our efforts in these projects is the fact that educational resources are almost never re-used and that dissemination of these resources is often problematic. In order to promote re-use, authors should be able to find out what is available, where it is located, whether it is suited and how it can be obtained. In order to support re-use, we have developed a general-purpose solution based on database technology, a hypermedia data model and a telecommunication infrastructure, as will be explained in the remainder of this paper.

3 Overview

HOME is based on a client-server architecture, where the server takes care of data management. The client interacts with the end user, initiates requests to the server and displays the result. The HOME server is itself distributed, as will be explained further on.

The next section deals with the reasons for and the consequences of separating data and metadata management. Section 5 and 6 detail the two layers of HOME that are responsible for data and metadata management of multimedia objects respectively. Section 7 presents an additional layer that superimposes a hypermedia structure on the multimedia objects. Section 8 mentions some relevant aspects of our current implementation. Section 9 lists the more distinctive features of HOME, section 10 compares our approach with related work and section 11 briefly mentions some of our plans for the future.

4 Separation of Data and Metadata

A fundamental design decision of our distributed hypermedia system is the separation of 'raw' multimedia data (audio, still images, video, etc., see section 5) and metadata (data describing characteristics of multimedia objects, see section 6). We have taken this decision because 'raw' data and metadata are rather different, so that the tools and techniques used to store and process them differ as well [Little, Venkatesh 94] [Bucci, et al. 94] [Bowman, et al. 94]:

- Compared with metadata that describe e.g. the author of a multimedia object, or its content, the object itself is much *larger* in size.

- Whereas metadata are well structured, this is mostly not the case with 'raw' multimedia data: a bitmap e.g. is (nothing but) a large sequence of bits. Video encoding standards contain little or *no structuring* information (cut points, camera positions), and this information is embedded. Moreover, current image processing and video computing techniques are not able to automatically make this structure explicit, especially in the case of more abstract characteristics [Tonomura, et al. 94] [Jain, Hampapur 94].
- Raw multimedia data are often *time-dependent* [Hardman, et al. 94]. This must be modelled in a multimedia database and affects communication [Little, Venkatesh 94], because the synchronisation between different data streams must be preserved.

Most current database management systems can cope well with metadata of multimedia objects. The raw data themselves must be stored in either external files or so-called Binary Large OBjects ("BLOB's"). In neither of these cases are they integrated very well in the database.

An advantage of separating data and metadata is that a user can examine the contents of the database without having to retrieve the typically very voluminous data objects themselves [Little, Venkatesh 94]. Interaction with a system designed in this way typically proceeds in two phases:

- *location phase*: querying the metadata, the user identifies the relevant data; the end result of this phase is a set of locators of raw data; the locator may also include the protocol to be used to retrieve the data, as in the case of W.W.W. Universal Resource Locators [Berners-Lee, et al. 94] [Cailliau 95];
- *communication phase*: parameters for quality of service are negotiated and delivery of the raw object(s) takes place.

5 Raw Data Layer

5.1 Introduction

The raw data layer contains the *raw data objects*: we use this term to refer to raw multimedia data, i.e. still images, audio clips, video sequences, etc.

In HOME, this layer integrates all raw objects that are accessible through services on the Internet. This is important for interoperability (see section 9) as it enables us to refer to all such objects on the metadata level. For this purpose, we rely on the Universal Resource Locator (U.R.L.) mechanism (see section 6).

The next section presents in some detail a special-purpose data store we have developed for storage of still images and short video sequences. It illustrates the flexibility of our approach that enables us to integrate a wide diversity of multimedia data repositories within our framework.

5.2 Image Store

We have developed an *image store* for storage of still images and video sequences. This store has been integrated in the raw data layer, as will be explained in this section.

5.2.1 Storage

Images and sound are stored in analogue form on laser disc.

- A Write-Once Read Many times (WORM) disc holds up to 36,000 frames per side, of still or moving video.
- A pre-recorded analogue Laservision disc (such as used in the Bristol Biomedical disc, see below) may contain up to 55,000 per side.

These capacities compare well with current digital storage, offering efficient, high quality and reasonably fast accesses, with an average seek time of 0.5 sec.

Once one ventures into the Video On Demand servers [Little, Venkatesh 94], digital systems perform better though the costs are higher. We are currently testing digital storage units and plan to migrate to such systems.

5.2.2 Access

Direct access to the image store is possible over analogue links such as UCL's internal Live Interactive Video Educational Network (LIVE-NET), DBS, and video conference links through the use of Codec's over ISDN or ATM (see figure 1). The Image store is currently being used on the INSURRECT project which is involved with teaching surgery within the UK over the SuperJANET ATM video network.

Frames may be *digitised on-the-fly* and delivered to the network. The server will accept requests in the HyperText Transfer Protocol (HTTP) making it accessible within the W.W.W. (see also section 7). Once a request for digitisation is received, it is parsed by a Common Gateway Interface (CGI) compliant client to retrieve the frame number of the image, its desired size, and encoding [Gleeson, Westaway 95]. The frame number may be checked against an access list so as to provide access to certain sections of the disc. If the request is valid, the frame is accessed on the disc and subsequently digitised. The laser disc is controlled through its serial port which is accessible, along with a digitising card on a networked Sun workstation.

The digital image is converted to the desired size so the user may view 'thumb nail' versions of the images before retrieving an entire image. This also keeps the network load to a minimum whilst users browse the image base. Finally the image is delivered in the requested coding - such as GIF or PPM, which allows the image to be included as an in-lined image in a HTML document.

Further developments are underway to deliver sections of the moving video by conversion into a suitable moving image format such as MPEG or H.261[ITU-T Recommendation 03/93]. Dependent upon network access these may either be delivered in real-time or as one data file. For real-time access the demands on the system are much greater and would require substantial further development. However as interest in Video On Demand services grows, relevant work is being done in both the commercial and academic environments.

By using a hybrid analogue-digital approach we can exploit the benefits of analogue storage media in the digital world. Laser discs offer large storage potential and allow the use of analogue technologies for transfer and manipulation. At UCL we currently utilise our video network LIVE-NET for teaching which is a hybrid of digital and analogue links. All switching is done in the analogue

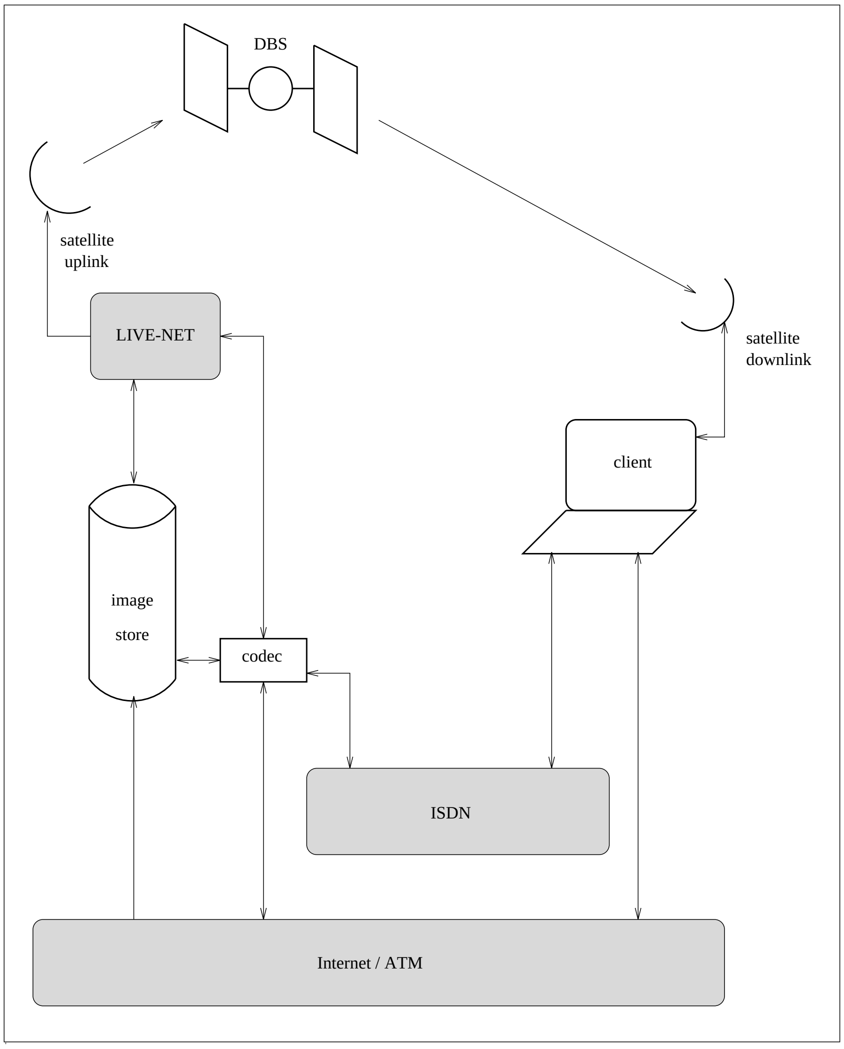

Figure 1: telecommunication between different components

domain on LIVE-NET, though once we go onto the SuperJANET video network digital Multi-point Control Units (MCU) are used to automatically switch video streams dependent upon operational mode.

6 Multimedia Layer

The multimedia layer deals with characteristics of the raw data objects considered in isolation. The data model of this layer is based on an object-oriented approach. Classes can be defined for different kinds of raw data objects. Figure 2 represents the currently implemented class hierarchy:

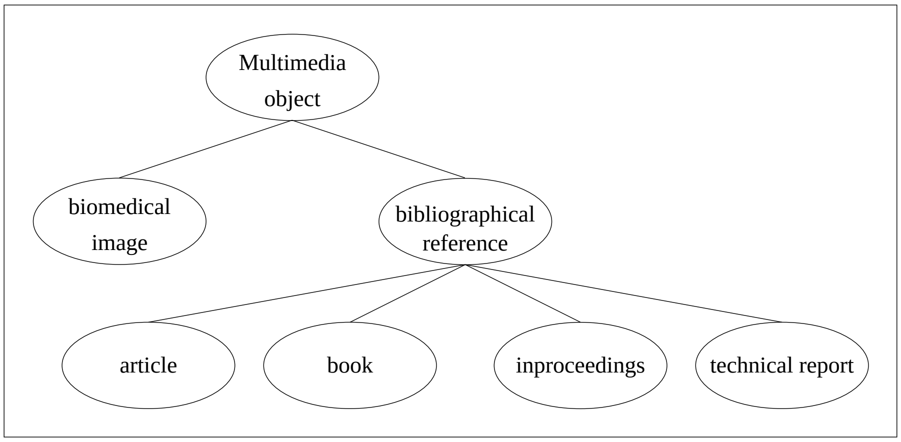

Figure 2: Class hierarchy of the Multimedia Metadata Layer

– The *multimedia object* class groups characteristics common to all raw data objects. An important attribute of this class is the unique name used to refer to an object (see also section 9). Other attributes of this class include the author, the creation date, the person who last modified the object, and the date of last modification.
  The raw data content of a multimedia object is referred to by the mechanism known as Universal Resource Locators (U.R.L.'s). This mechanism stems from the W.W.W. [Berners-Lee, et al. 94] [Cailliau 95] and indicates both the protocol to be used to retrieve the object (e.g. ftp, gopher, http) as well as the location of the object itself (basically the Internet host and a filepath on that host).
  It is important to emphasise the flexibility of the U.R.L. approach: in fact, we believe this is one of the main contributions of the World-Wide Web. Through U.R.L.'s, references can be made to documents accessible over the file transfer protocol, telnet, gopher, etc., on any computer connected to the Internet. These documents can also be generated upon request, using the Common Gateway Interface (CGI [Gleeson, Westaway 95]) mechanism. Moreover, the addressing scheme is easily extensible with new protocols.
  Our terminology is somewhat unusual in that normally the term multimedia object refers to a structured composite of several single media. In fact, HOME uses a black box approach and doesn't address issues such as e.g. synchronisation between different media components of a multimedia object [Buford 94].

- *Biomedical images* constitute a subset of the images on our image store (see section 5.2). Currently, this set includes ca. 5900 images, whose specific characteristics (species, sex, stain, etc.) are modelled by this subclass.
- Another subclass of multimedia objects is the class of *bibliographical references*, with further subclasses for articles in journals, books, papers in conference proceedings and technical reports. Attributes for these classes include e.g. author, title, publisher, etc. This class hierarchy is based on the Bibtex system that is part of the LaTex environment [Lamport 86].

The multimedia layer of our server essentially supports creation, deletion and modification of multimedia objects. Object retrieval based on search criteria is also possible: this facility enables for instance an end user to look for biomedical images donated by a particular person, concerning a particular species, etc. This kind of search facility is very important in a large-scale environment that may include hundreds of thousands of multimedia objects, where simple browsing techniques break down because of the overwhelming amount of available data. (Anyone who ever experienced the frustration of not finding a W.W.W. document he visited some time before will appreciate this point.)

7 The Hypermedia layer

As explained in the previous section and illustrated by figure 3 (where higher layers refer to objects at lower layers), the multimedia layer manages metadata of isolated objects stored at the raw data layer. Metadata concerning interrelationships between different objects are dealt with by the hypermedia layer.

The most simple hypermedia data model is the basic node-link paradigm: information is organised in chunks that are called 'nodes' and that can be interrelated by 'links' [Nielsen 90]. We have developed a new hypermedia data model, based on a set-oriented approach, strongly influenced by the HM data model [Andrews, et al. 95] [Maurer, et al. 94]. The data structuring facilities of our hypermedia data model are presented in section 7.1. Section 7.2 and 7.3 detail navigational and query facilities. Section 7.4 explains how query and navigational access can be combined.

7.1 Data Structuring

The basic data structure of the Sets-Of-Sets (SOS) model is a set, a somewhat modified version of the traditional mathematical concept. A set is identified by its unique *name*. In the SOS model, there are two classes of sets:

- A *singleton* is a set with exactly one element, a multimedia object (see section 6). For every multimedia object created at the multimedia layer, a corresponding singleton (with that object as its single element) is automatically created at the hypermedia layer. In this way, all objects from the multimedia layer are accessible in the hypermedia layer.
  A typical example might be:
  S = {map of the K.U.Leuven campus}.
- A *multiset* is a set with an arbitrary number of elements, that are sets themselves. The elements of a multiset are ordered. A typical multiset is e.g.:

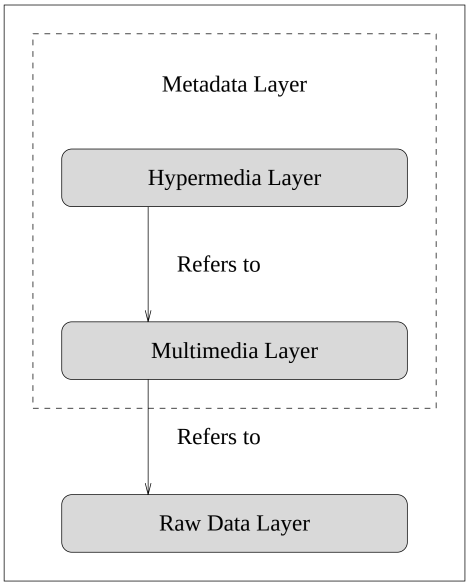

Figure 3: The different layers of HOME

M = { {map of the K.U.Leuven campus},
{ {texts about the history of the K.U.Leuven},
{historical images}},
{administrative information},
{presentation of research at the K.U.Leuven}}.

A multiset can be *extensionally defined*. In that case, membership is defined by manually adding or removing elements to or from the set. Alternatively, a multiset can be *intensionally defined*, when a set of search criteria defines membership to the multiset. In the latter case, the multiset is called a *dynamic multiset* in HOME. This is similar to the notion of a computed composite in [Halasz 88].

It is important to note that set membership does not need to be hierarchical, but can contain loops: recursive relationships are explicitly allowed. Consider e.g. figure 4, where the rounded squares represent sets and the connections indicate membership.

The set on our research unit ('Indigo') contains a set on projects we are involved in and another set with elements for each of the members of our unit. One of the projects is 'HOME', presented in this paper. The multiset on HOME has an element on the people working on HOME. For each of the members of the research unit, there is a set 'Projects' that contains information on all the projects a particular member is involved in. Clearly, there are several cycles in this structure.

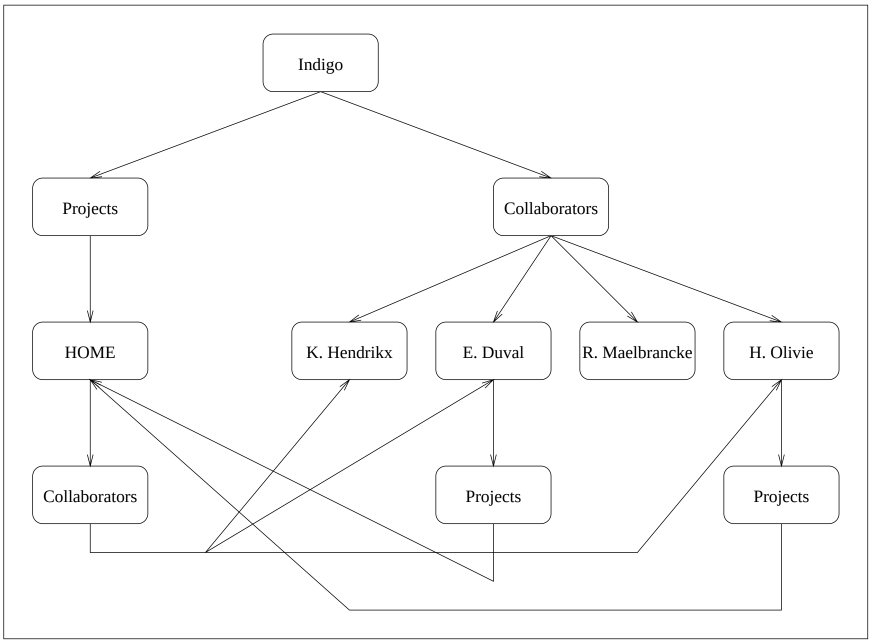

Figure 4: Recursive Membership: an Example

As sketched above, the SOS model is a rather simple hypermedia data model, where information is structured in a (not strictly) hierarchically manner. The net result is rather similar to data structuring in e.g. Hyper-G [Kappe, Maurer 93] [Kappe, et al. 93] or Gopher. We are currently working on a more advanced data model.

7.2 Navigation

In this section, we will present the currently implemented user interface to HOME. When the user first accesses HOME, a list of root sets is presented, as in figure 5. The user can get back to this screen, by clicking on the 'Home' icon, always present in the upper left corner of the screen.

In order to zoom in on a multiset (e.g. 'Hypertext and Hypermedia' in figure 5), the user clicks on the folder icon. The result is shown in figure 6.

The user can now zoom out of the 'Hypertext and Hypermedia' multiset by clicking on the opened folder icon of figure 6. This brings him back to figure 5. If, however, the user zooms further in on 'Hypermedia Systems', again by clicking on the folder icon, then the screen looks like figure 7.

Now, the current set is 'Hypermedia Systems', with a number of multisets as its elements, and a singleton ('On Second Generation Hypermedia Systems (...)'). Clicking on the arrow icon in front of the singleton visualises the multimedia object that is the single element of this singleton, which leads to figure 8.

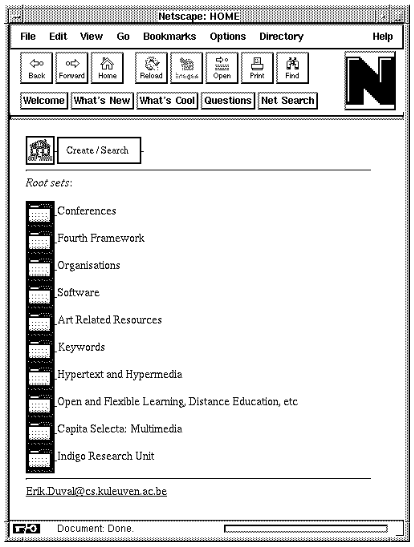

Figure 5: Root Sets

The 'arrow up' icon on figure 8 enables the user to zoom out of the current singleton. This takes him back to the screen of figure 7.

7.3 Query

The 'Create/Search' button at the top of every screen allows an end user to create new sets or to query the HOME system. The screen on figure 9 represents a typical query screen for a multimedia object, in this case a paper in a conference proceedings.

When querying the system, the user can provide a search pattern for any of the attributes. In figure 9, the user searches for all papers in proceedings where the pattern 'Duval' is present in the list of authors. Figure 10 shows the result of the query.

The different singletons that belong to the query result (which is, in fact, a dynamically generated multiset) are listed, with arrow icons to enable the end user to visualise them.

7.4 Combination of Navigation and Query

Navigation and query access can be combined: imagine e.g. that an end user obtains figure 10 as a result of the query in figure 9. When a user visualises, say, 'A Home for Networked Hypermedia (...)' (the third item in the list of figure 10) and then zooms out of that singleton, then the result looks like figure 11.

Navigating from one of the singletons that belong to the result of a query, end users can thus find material related to the query result, but not part of that

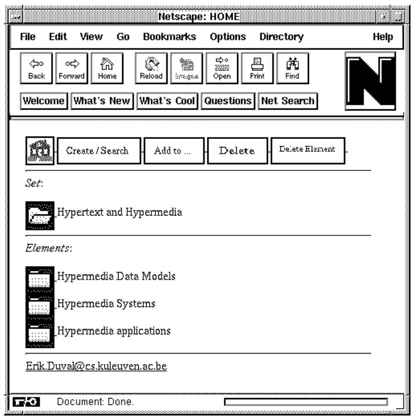

Figure 6: Zooming in on 'Hypertext and Hypermedia'

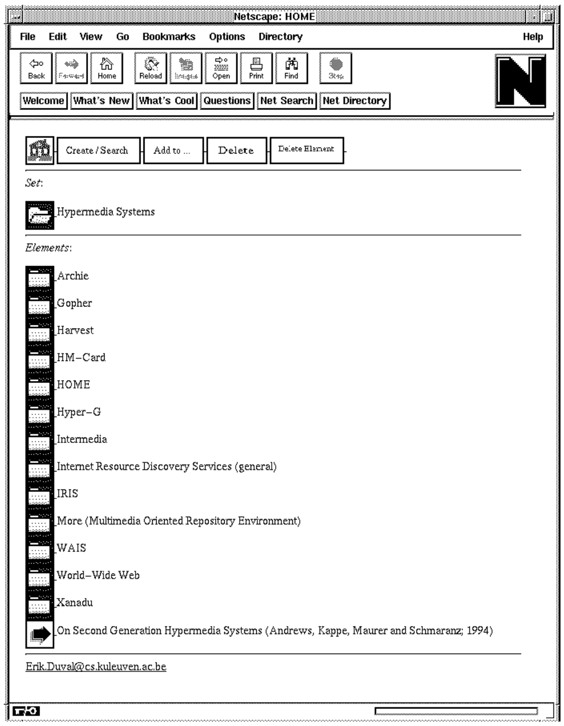

Figure 7: Zooming in on 'Hypermedia Systems'

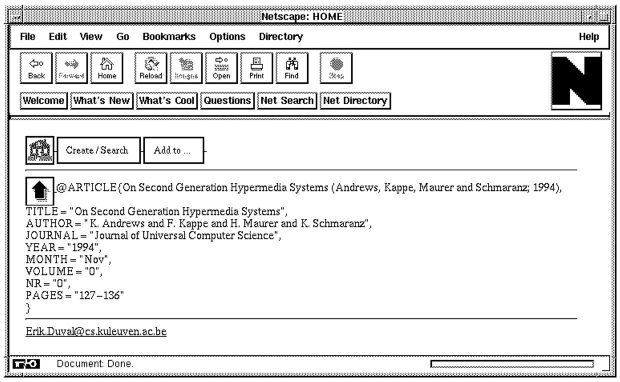

Figure 8: Visualising 'On Second Generation Hypermedia Systems (...)'

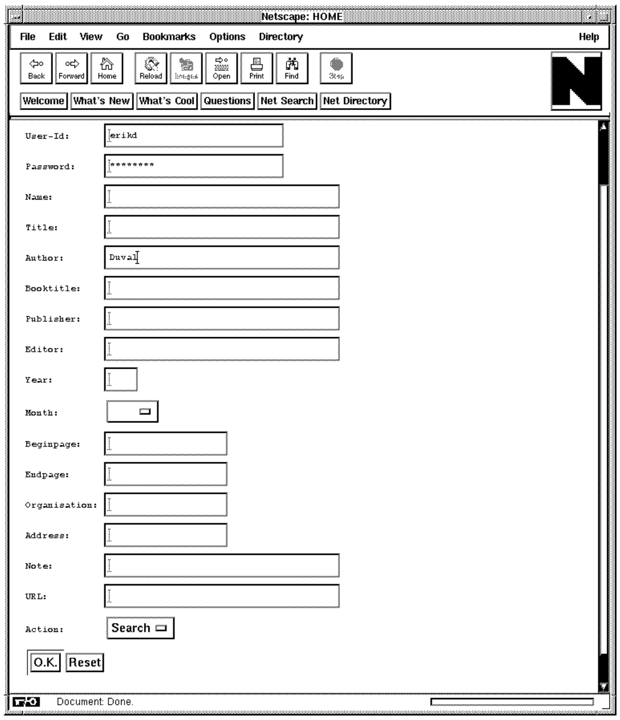

Figure 9: A query screen for proceedings papers

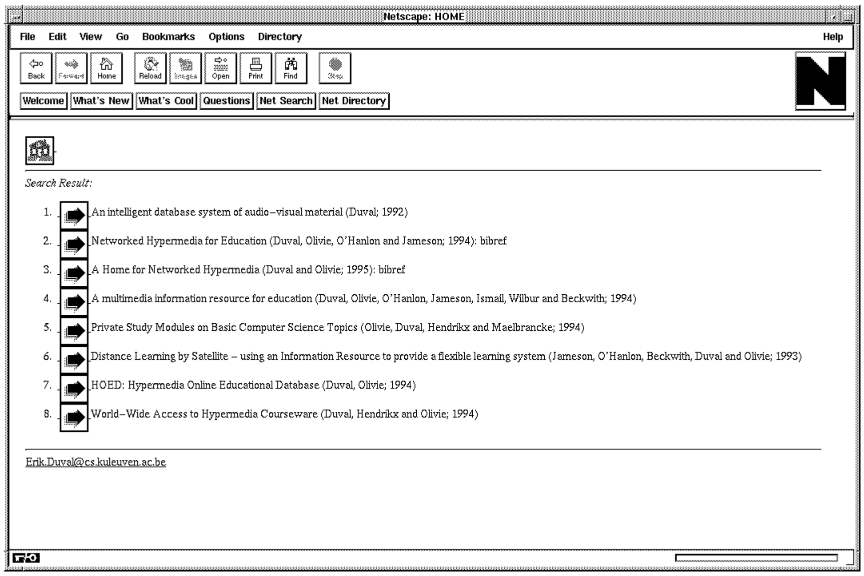

Figure 10: Result from a query

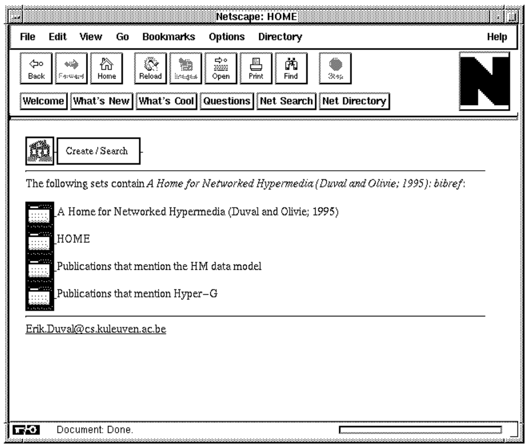

Figure 11: Zooming out of a query result

result itself. Very often, this material will also be of interest to the querying user. This approach can contribute to overcoming a well-known problem in information retrieval: it is often difficult for end users to formulate exact search criteria that identify all and nothing but relevant material. The approach sketched above requires the end user to identify only some of the relevant material. Local navigation (by zooming out of the relevant singletons) enables him to retrieve the other relevant material afterwards, as similar material will normally be linked with the material identified in the query.

8 Some Implementation Aspects

Figure 12 indicates how we have integrated our distributed server in the World-Wide Web. We have set up W.W.W. servers in London and Leuven. W.W.W. clients can access these servers, using the HyperText Transfer Protocol (HTTP). Through the Common Gateway Interface (CGI [Gleeson, Westaway 95]) protocol, the W.W.W. servers interact with the HOME server and the image store. The two latter components deliver the result, package it in the W.W.W. HyperText Mark-up Language (HTML) for document definition and send it to the W.W.W. server, that in turn ships it to the W.W.W. client that originally submitted the request.

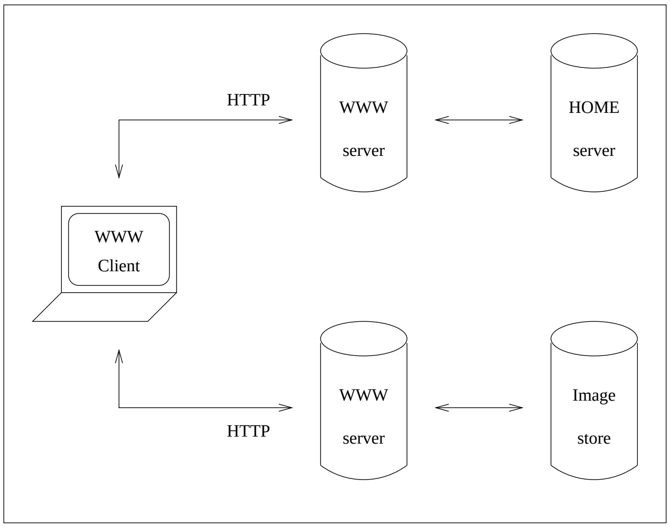

Figure 12: Interaction between W.W.W. client and server components

The main reason why we have integrated our system into the W.W.W., is the availability of client and server software for a wide variety of hard- and software platforms. Section 9 will detail the added value HOME provides when compared with the W.W.W.

The HOME server is implemented in Tcl [Ousterhout 94]. This interpreted scripting language offers considerable functionality, ease of use and flexibility. Tcl is certainly a well suited language for rapid prototyping and early experimentation. Up to now, we have not had any problems with a possible decrease in efficiency, as the connection and transmission times typically far outweigh the time it takes to process requests.

All data describing characteristics of the multimedia objects and their interrelationships are stored in a commercially available Relational DataBase Management System (RDBMS), in our case Oracle, but any full-fledged RDBMS will do, as we only need the typical functionality all comparable products support. Interaction between the Oracle DBMS and the Tcl language is implemented in Oratcl, an extension to the Tcl language.

It is important to note that we have deliberately chosen not to use the SQL language *for client server interaction* [Khoshafian, et al. 93]. (As mentioned at the start of this section, we do rely on SQL for internal server functionality.) There are two reasons for this decision:

- SQL would provide clients with too much flexibility and would enable them to create inconsistent server states.
- Using SQL for client server interaction would make clients more complex as more application logic would need to be replicated in every client.

The Tcl Application Programmer's Interface that we have designed provides a higher level of abstraction and enables us to overcome the difficulties mentioned above.

9 Distinctive Characteristics

In this section, we list some of the more distinctive characteristics of our approach. The next section will compare HOME with related research.

9.1 Separation of Structure and Content

The *separation* of the multimedia and hypermedia layer results in a separation of structure and content, or, in traditional hypermedia terminology, a separation of links and nodes. This is different from the W.W.W. approach, where links are embedded in documents. [Hall, et al. 95] refers to the problem that embedded links cannot be changed without revising the document itself as *link fossilisation.* Separation offers many advantages, especially in a large-scale, networked, multiuser context [Duval, Olivié 94] [Kappe, et al. 93]:

- *Read-only* material can be linked to and from:
  - Documents on e.g. CD-ROM or stored on videodisk (and digitised upon request, as in the case of the image store, see section 5.2) can be linked to and from. In HOME, such material can act as content of a singleton. The latter can in turn be an element of a multiset, etc.

- In traditional hypertext terminology, users can link to or from any document. In SOS terminology, users can insert sets created by others as elements in their own multisets, without modifying the set originally created by somebody else. This last point is important as it guarantees *document integrity* to the original author, while still allowing anyone to superimpose personal structures on the content.

– Using traditional terminology, *bi-directional links*, or a *back-link capability* [Engelbart 90] can more easily be supported if structure is separated from content, because link information can be queried independently (i.e. without a need to scan all the documents), to identify documents that refer to a particular document. In the SOS context, this corresponds to the fact that one cannot only zoom in on one of the elements of a set, but that it is also possible to zoom out of a set, obtaining a list of sets that contain the current set.

  - This makes it easy to remove *dangling references* when a document is deleted. When a set is destroyed, the structure of all multisets it previously belonged to can be updated accordingly. In traditional hypermedia terminology, when a document is destroyed, all links referring to that document can also be deleted.
  - The back-link capability is also important for the generation of local overview maps. Such maps indicate the links to as well as from a particular node. Local maps can help to avoid the disorientation problem known as "lost in hyperspace", although they offer only limited relief for navigation problems in large information spaces. In that case, they can be integrated in a larger overview through so-called "fisheye views" [Sarkar, Brown 94]. In HOME, a graphical overview of the is-an-element-of relationship could be used to provide this sense of context to the user. Such a facility requires the back-link capability.

– *Different structures* can be superimposed *on the same content*: a particular set (e.g. representing an employee) can be linked to some sets (e.g. representing the projects he works on) in the context of one set (e.g. representing the workload of each employee) and to other sets (e.g. the employees he supervises) in the context of another set (e.g. representing the hierarchical structure of the company) - it is difficult to envisage how this could be achieved in e.g. an HTML encoding in the W.W.W.

9.2 Identity rather than Location

As the multimedia layer enforces object identification, the *location* of a multimedia object can easily be modified: this requires only an update of the corresponding attribute. Using identifiers contrasts favourably with the W.W.W. approach of referring to objects by location, because the latter approach implies that all references to a document must be updated as well when the location changes. This is clearly impractical in large-scale hypermedia environments, as daily practice with the W.W.W. illustrates all too well in the form of broken links. The process whereby, in a document, gradually more and more links to locations become invalid, as the destinations change location, is called *link decay* in [Hall, et al. 95].

9.3 Interoperability

Interoperability is an important concept in an open distributed environment. This notion refers to the idea that different information retrieval and discovery [Obraczka, et al. 93] systems should work well together, e.g. that clients of one such system should be able to query servers of another system.

- Our environment interoperates rather well with existing tools and services: any object referred to by a U.R.L. can be incorporated at the multimedia metadata layer and subsequently a structure can be defined over that object at the hypermedia metadata layer. This approach makes our environment interoperable with FTP, WAIS, Gopher and W.W.W. As the U.R.L. mechanism is open and extensible, it can be expected that new services would also fit in this approach.
- As explained above, W.W.W. clients can access our server, through a gateway mechanism that translates W.W.W. requests and packages the result from our server in W.W.W. format (i.e. HTML). This integrates our environment quite seamlessly in the W.W.W. A similar approach could be followed for integrating HOME in other similar environments, e.g. Hyper-G [Kappe, Maurer 93] [Kappe, et al. 93].
- A third form of interoperability results from the fact that documents composed with external application programs can be interlinked with other documents in HOME. Thus we can avoid the problem of *dead ends* which arises in the World-Wide Web because only native HTML documents can have embedded links [Hall, et al. 95].

9.4 Database rather than File Management Systems

We believe it is important to exploit existing *database technology* when designing multimedia or hypermedia databases, rather than directly relying on the file system services offered by the operating system (as in the W.W.W.), because of the extra functionality that comes with databases [Bucci, et al. 94]:

- ACID properties (*a*tomicity, *c*onsistency, *i*solation and *d*urability of transactions): Essentially, these properties guarantee that a transaction, as the atomic unity of client-server interaction either commits in its entirety or not at all, that its effect is the same as if it would have been submitted in a stand-alone system and that this effect will last over time [Elmasri, Navathe 89].
- *concurrency control, access control, accounting* features: All operations have parameters `userid` and `password`. They will only be executed if the user has sufficient privileges to do so. This will be guaranteed by the DBMS.
- Provisions for *back-up and recovery* are more elaborate in a DBMS than in a traditional file system.
- A *query engine* can be used to:
  - define *dynamic links*: Conventional links are defined in an extensional way, based on the identity of the participating nodes. Dynamic links are defined intensionally: a prescript identifies all nodes the link points to (or from). The prescript is resolved at run-time, when the link is activated [Halasz 88]. Dynamic links are particularly useful if the data involved change frequently over time. Dynamic definition of membership is the HOME concept that corresponds with dynamic links.

- provide *query based access*: As an alternative to navigation, query based access is especially important in a large-scale environment, where the simple browsing paradigm breaks down because of information overload. As explained in section 7.4, it is often quite convenient to issue a query first, in order to locate some of the relevant information. Subsequently, local navigation within or around the search results supports exploration of relevant material either included in or related to the query result.

9.5 Relational Database Technology

Although we had originally anticipated that HOME would benefit from the added value of object oriented database technology, we have completed the implementation phase, using a *relational* DBMS. In retrospect, the choice between a relational and an object oriented DBMS seems to be less important than expected [Bucci, et al. 94]:

- The most important reason is that the DBMS is responsible for management of *meta*-data only: this implies that no novel features for dealing with multimedia content are required.
- An object oriented DBMS would provide more built-in support for the class hierarchy of figure 2. Implementing such support in a language like Tcl [Ousterhout 94] on top of a relational DBMS proved to be a rather moderate effort.
- The core functionality of the SQL query language is sufficiently standardised to allow for a design and implementation that is reasonably independent of the particular RDBMS (Oracle in our case).

9.6 Structured Objects

A rather unique feature of HOME is the *integration of structured objects* (see section 6): subclasses can be defined for objects that share a particular set of characteristics. This feature enables end users to interlink structured data (bibliographical references, co-ordinates of people, etc.) with unstructured multimedia objects (located through the U.R.L. mechanism [Berners-Lee, et al. 94] [Cailliau 95]).

10 Related Work

10.1 Hypermedia Data Models

Our hypermedia data model is strongly influenced by the HM model. In HM [Andrews, et al. 95] [Maurer, et al. 94], the basic unit for structuring information is an S-collection: this is either a primitive node (a name and a multimedia content) or a structure (containing a head, a set of S-collections and a set of links). The most distinctive differences between our SOS model and the HM-model are:

- In HM, a navigable *topology* can be defined between members of an S-collection.

- In HM, when an S-collection is accessed, its content (or the content of its head) is automatically visualised.
- In HM, an S-collection can have an associated multimedia content. The object need not be 'packaged' as a singleton, as in the SOS model.
- [Maurer, et al. 94] mentions that 'extending the functionality of Zoom_Out to give access to any S-collection of which the current collection is a member at first seemed attractive, but on reflection the possibility of users zooming out into a completely different context appeared to promote more confusion than understanding'. This is exactly the opposite of our finding (see e.g. section 7.4, where an elegant combination of navigational and query access is based on the idea rejected in the quote above). Hence, the SOS model defines zooming out as providing access to all sets that the current set is a member of.

A number of other data models are similar to both the SOS and the HM model, because they are also set based [Eichmann, et al. 94] [Garzotto, et al. 94]. As we are currently developing a new data model, we will not further elaborate on this issue here, nor on the relationship between set based and more traditionally oriented data models for hypermedia. These subjects will be dealt with in a forthcoming paper.

10.2 Distributed Hypermedia Systems

- Above, we have repeatedly compared our approach with the *World-Wide Web* [Berners-Lee, et al. 94] [Cailliau 95]. Although we have cited a number of well-known problems with 'the Web', it is appropriate to emphasise here that the Web made at least two important contributions to the development of distributed hypermedia systems - besides, of course, the exposure of the very notion of distributed hypermedia to the public at large:
  - The U.R.L. addressing scheme (see also section 6) accommodates references to documents on any Internet computer, through an extensible set of protocols.
  - The HyperText Transfer Protocol (HTTP) supports requests for documents, referred to by a U.R.L.
- *Hyper-G* is a large-scale hypermedia system under development at the University of Graz [Kappe, Maurer 93] [Kappe, et al. 93]. It is similar in scope to the World-Wide Web, but doesn't suffer from some of W.W.W.'s problems - most notably: links and nodes are stored separately in Hyper-G servers and query based access to documents is supported. Both these characteristics also hold for HOME. The architecture of Hyper-G is different from our layered approach and consists of a full text server, a link server an a document server. The latter two roughly correspond with our hypermedia and multimedia management layer respectively.
- The *Microcosm* design is based on a set of processes. A document viewer can send messages to a chain of filters. These can either block a message, pass it on or change it before passing it on. Finally, the messages arrive at a link dispatcher that displays information to the user (e.g. a list of links that the user can follow) [Hill, Hall 94] [Hall, et al. 95]. In an effort to adhere to the 'open hypertext' approach, third party applications can be used to create, edit and view documents, in a more or less integrated way. Recently, the

Microcosm team has investigated integration of this link service approach in the World-Wide Web [Hall, et al. 95] and started a new project, called *Multicosm*, to investigate large-scale distribution of Microcosm [Mul 94].
- *MORE* (Multimedia Oriented Repository Environment) deals with information for software re-use [Eichmann, et al. 94]. Just like HOME, MORE relies solely on the World-Wide Web for end user access. HTML documents are all generated dynamically, based on data stored in a relational database. It is interesting to note that the MORE team, targeting software re-use, came up with a solution very similar to the one we developed for courseware re-use.

11 Future

- We are currently working on the integration of support for *Computer Supported Collaborative Work* (CSCW) in HOME. More concretely, we are currently concentrating on the functionality required for interaction between student and staff and among students, in a set-up where HOME is used as a Campus-Wide Information System (CWIS). Proper support for CSCW requires:
  - *event notification*: in order to warn users when something happens to shared information;
  - *long transactions*: in traditional database context, a transaction lasts no longer than tenths of seconds; in a collaborative environment, this can be substantially longer (hours or days).

  The hypermedia paradigm lends itself well to asynchronous communication [Andrews, et al. 94]. A real-time component must be integrated within this framework, so that e.g. a student can contact a member of the staff in case of urgent problems. We will therefore elaborate a gateway between electronic mail systems (the Unix mail program in the first place) and HOME. This must enable end users to edit, send and receive email from within HOME.
- As mentioned above, the HOME server is implemented in the Tcl language [Ousterhout 94]. Tcl includes facilities for safe inter-process communication and is integrated with Tk, a user interface toolkit. We are therefore considering to develop specialised HOME client software in Tcl/Tk, bypassing the World-Wide Web environment completely (though some form of interoperability should probably be preserved), which would provide us with a more flexible implementation environment. As newer versions of e.g. the Mosaic W.W.W. client can communicate with Tcl applications, interesting new possibilities arise. These are currently under investigation.
- In Hyper-G [Kappe, Maurer 93] [Kappe, et al. 93], a search can be constrained to a particular (set of) collection(s). This sort of *search scope control* is extremely important in a large-scale networked environment as the number of data objects that satisfy the user's search criteria can be very large. In HOME, a search is currently performed over the largest relevant domain, i.e. the set of all data objects that belong to a particular class (e.g. biomedical images), because the search constraints are expressed in terms of attributes and these are only defined for data objects that belong to the relevant class. On the other hand, HOME is more flexible than Hyper-G in this respect, as it allows arbitrary complex search conditions over an arbitrary large set of attributes defined for a particular data object class.

- The present *user interface* supports creation of hypermedia data. Subsequent modification (apart from deletion) is not supported and should be. User interface issues related to large information spaces should also be elaborated further, based on techniques such as fisheye views [Sarkar, Brown 94], 3D visualisation [Robertson, et al. 93] and treemaps [Shneiderman 92].
- We are currently planning a number of *projects* for development of HOME servers. These include a bibliographical database, a hypermedia server on live art, an annotated slide collection on architecture, a distributed European network of topical courseware databases and an information resource on job opportunities.
- *Active* and *deductive database* technology could be used to model (interaction with) the end user in the server. This would make it possible to design and implement customisation and adaptation of the information presented to the end user.

12 Conclusion

In this paper, we have presented HOME, an environment for hypermedia management. Based upon a rigorous separation of data and metadata, three different layers are discerned in HOME: a storage layer dealing with raw data management, a multimedia layer that takes care of metadata concerning individual objects and a hypermedia layer where interrelationships between objects are defined. The latter layer relies on the 'Sets Of Sets' model that structures data in singletons and multisets. We are currently working on a more advanced data model that will replace the SOS model in a future version of HOME.

Acknowledgements

The financial support of the European Commission, provided within the framework of the DELTA program for the projects CAPTIVE, MTS and ACT is gratefully acknowledged. We are also indebted to the Belgian National Fund for Scientific Research, for its partial funding.

We also wish to thank Richard Beckwith, Jane Williams, Nermin Ismail, Koen Hendrikx, Rudi Maelbrancke and Nick Scherbakov for their helpful comments - they were invaluable to us. Finally, the suggestions and comments by the anonymous reviewers have helped us to improve the overall quality of this paper.

References

[Andrews, et al. 94] K. Andrews, F. Kappe, H. Maurer, and K. Schmaranz. On second generation hypermedia systems. *Journal of Universal Computer Science*, 0:127–136, Nov 1994.

[Andrews, et al. 95] K. Andrews, A. Nedoumov, and N. Scherbakov. Embedding courseware into the internet: Problems and solutions. In *ED-Media 95: World Conference on Educational Multimedia and Hypermedia (June 18-21, 1995, Graz, Austria)*, Jun 1995. To be published.

[Beckwith, et al. 93] R. C. Beckwith, D. G. Jameson, P. O' Hanlon, and E. Duval. Interactive satellite teaching and conferencing using an image server. In *Proceedings of the ESA Olympus Conference, April, 20-22, 1993, Seville*, 1993. To be published.

[Berners-Lee, et al. 94] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret. The world-wide web. *Communications of the ACM*, 37:76–82, Aug 1994.

[Bowman, et al. 94] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber, and M.F. Schwartz. Harvest: A scalable, customizable discovery and access system. Technical Report CU-CS-732-94, Department of Computer Science, University of Colorado-Boulder, August 1994.

[Bucci, et al. 94] G. Bucci, R. Detti, V. Pasqui, and S. Nativi. Sharing multimedia data over a client-server network. *IEEE Multimedia*, 1(3):44–55, 1994.

[Buford 94] John F. Koegel Buford. *Multimedia Systems*. SIGGRAPH Series. ACM Press, 1994.

[Cailliau 95] R. Cailliau. About www. *Journal of Universal Computer Science*, 1:221–230, Apr 1995.

[Duval, Olivié 94] E. Duval and H. Olivié. HOED: Hypermedia online educational database. In T. Ottman and I. Tomek, editors, *Educational Multimedia and Hypermedia Annual, 1994. Proceedings of ED-MEDIA 94 - World Conference on Educational Multimedia and Hypermedia, Vancouver, BC, Canada; June 25-30, 1994*, pages 178–183, 1994.

[Eichmann, et al. 94] Eichmann, McGregor, and Danley. Integrating structured databases into the web: The more system. In R. Cailliau, O. Nierstrasz, and M. Ruggier, editors, *Proceedings of the First International WWW Conference - Geneva, Switzerland, 25-27 May 1994*, pages 369–378, May 1994. Available at http://rbse.jsc.nasa.gov/eichmann/www94/MORE/MORE.html.

[Elmasri, Navathe 89] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Benjamin/Cummings, 1989.

[Engelbart 90] D. C. Engelbart. Knowledge-domain interoperability and an open hyperdocument system. In *ACM CSCW 90 Proceedings*, pages 143–156, Oct 1990.

[Garzotto, et al. 94] Franca Garzotto, Luca Mainetti, and Paolo Paolini. Adding multimedia collections to the dexter model. In *Proceedings of ECHT 94: European Conference on Hypermedia technology, Edinburgh, Schotland; September, 18-23, 1994*, pages 70–80, September 1994.

[Gleeson, Westaway 95] Martin Gleeson and Tina Westaway. Beyond hypertext: Using the www for interactive applications. In R. S. Debreceny and A. E. Ellis, editors, *Innovation and Diversity - The World Wide Web in Australia. AusWeb 95 - Proceedings of the First Australian World Wide Web Conference, Ballina, Australia, 30 April - 2 May 1995.* Norsearch Publishing, Apr 1995. Available from http://www.its.unimelb.edu.au:801/papers/AW04-04/.

[Halasz 88] Frank G. Halasz. Reflections on notecards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31(7):836–852, July 1988.

[Hall, et al. 95] Wendy Hall, Leslie Carr, and David De Roure. Linking the world wide web and microcosm. In *New Directions in Software Development 95: The World-Wide Web (University of Wolverhampton; 8 March 1995)*, Mar 1995. Available from http://scitsc.wlv.ac.uk:80/ndisd/hall.ps.

[Hardman, et al. 94] L. Hardman, D. C. A. Bulterman, and G. van Rossum. The amsterdam hypermedia model - adding time and context to the dexter model. *Communications of the ACM*, 37:50–62, Feb 1994.

[Hill, Hall 94] Gary Hill and Wendy Hall. Extending the microcosm model to a distributed environment. In *Proceedings of ECHT 94: European Conference on*

Hypermedia technology, Edinburgh, Schotland; September, 18-23, 1994, pages 32–40, September 1994.

[Jain, Hampapur 94] Ramesh Jain and Arun Hampapur. Metadata in video databases. *ACM SIGMOD Record*, 23:27–33, Dec 1994.

[Jameson, et al. 93] D. G. Jameson, P. O'Hanlon, R. Beckwith, E. Duval, and H. Olivié. Distance learning by satellite - using an information resource to provi de a flexible learning system. In G. Davies and B. Samways, editors, *Teleteaching, Proceedings of the IFIP TC3 Third Teleteaching Conference, Teleteaching 93, Trondheim, Norway, 20-25 August 1993*, pages 459–467. International Federation for Information Processing, Elsevier Science, 1993.

[Kappe, et al. 93] Frank Kappe, Hermann Maurer, and Nick Scherbakov. Hyper-g. a universal hypermedia system. *Journal of Educational Multimedia and Hypermedia*, 2:39–66, 1993.

[Kappe, Maurer 93] F. Kappe and H. Maurer. Hyper-g: A large universal hypermedia system and some spin-offs. *ACM Computer Graphics*, May 1993.

[Khoshafian, et al. 93] S. Khoshafian, A. Chan, A. Wong, and H. K. T. Wong. *Client/Server SQL Applications*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 1993.

[Lamport 86] Leslie Lamport. *Latex: A Document Preparation System*. Addison-Wesley, May 1986.

[Little, Venkatesh 94] Thomas D. C. Little and Dinesh Venkatesh. Prospects for interactive video-on-demand. *IEEE Multimedia*, 1:14–24, 1994.

[Maurer, et al. 94] H. Maurer, N. Scherbakov, K. Andrews, and P. Srinivasan. Object-oriented modelling of hyperstructure: overcoming the static link deficiency. *Information and Software Technology*, 36:315–322, 1994.

[Mul 94] The multicosm project: Towards a scaleable distributed multimedia information environment. Dec 1994. Available at: http://vim.ecs.soton.ac.uk/multicosm.html.

[Nielsen 90] J. Nielsen. *Hypertext and Hypermedia*. Academic Press, 1990.

[Obraczka, et al. 93] K. Obraczka, P. B. Danzig, and S. Li. Internet resource discovery services. *IEEE Computer*, 26(9):8–22, September 1993.

[Ousterhout 94] J. K. Ousterhout. *Tcl and the Tk toolkit*. Addison-Wesley, 1994.

[Robertson, et al. 93] G. G. Robertson, S. K. Stuart, and J. D. Mackinlay. Information visualization using 3d interactive animation. *Communications of the ACM*, 36(4):57–71, April 1993.

[Sarkar, Brown 94] Manojit Sarkar and Marc H. Brown. Graphical fisheye views. *Communications of the ACM*, 37:73–84, Dec 1994.

[Shneiderman 92] Ben Shneiderman. *Designing the User Interface: Srategies for Effective Human-Computer Interaction*. Addison-Wesley, 1992.

[Tonomura, et al. 94] Yoshinobu Tonomura, Akihito Akutsu, Yukinobu Taniguchi, and Gen Suzuki. Structured video computing. *IEEE Multimedia*, 1:34–43, 1994.