ED 388 240 IR 017 389

AUTHOR	Duval, E.; Olivie, H.
TITLE	HOED: Hypermedia Online Educational Database.
PUB DATE	94
NOTE	7p.; In: Educational Multimedia and Hypermedia, 1994. Proceedings of ED-MEDIA 94--World Conference on Educational Multimedia and Hypermedia (Vancouver, British Columbia, Canada, June 25-30, 1994); see IR 017 359.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Computer Science Education; Courseware; Database Management Systems; *Databases; *Educational Resources; Foreign Countries; *Hypermedia; Information Retrieval
IDENTIFIERS	*Client Server Computing Systems; World Wide Web

ABSTRACT

This paper presents HOED, a distributed hypermedia client-server system for educational resources. The aim of HOED is to provide a library facility for hyperdocuments that is accessible via the world wide web. Its main application domain is education. The HOED database not only holds the educational resources themselves, but also data describing characteristics of these resources. Distinctive features of the HOED server are the separation of data regarding content (nodes) and structure (links), and use of a database management system (DBMS) rather than files for data management. The use of the HOED infrastructure for the development of private study modules on basic computer science topics and for reuse of courseware components is explained. Finally, HOED is briefly compared to related research. (Contains 11 references.) (Author)

HOED: Hypermedia Online Educational Database

E. DUVAL, H. OLIVIÉ
Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200 A, B-3001 Leuven (Heverlee), Belgium
Email: {Erik.Duval,olivie}@cs.kuleuven.ac.be

Abstract: This paper presents HOED, a distributed hypermedia client-server system for educational resources. The aim of HOED is to provide a library facility for hyperdocuments [Engelbart, 1990]. Its main application domain is education. The HOED database not only holds the educational resources themselves, but also data describing characteristics of these resources. Distinctive features of the HOED server are the separation of data regarding content (nodes) and structure (links), and use of a DataBase Management System (DBMS) rather than files for data management. We will explain how we make use of the HOED infrastructure for the development of private study modules on basic computer science topics and for re-use of courseware components. We will also briefly compare our work with related research.

1 Introduction

The work presented here deals with a *distributed hypermedia* system [CACM 1994]. It is related to and influenced by numerous fields of computer science, including information retrieval, databases, Computer Supported Collaborative Work (CSCW) and Computer Aided Learning (CAL).
In [Duval & Olivié, 1993], we presented a distributed *multimedia database* and explained how it can be used to manage multimedia and hypermedia educational resources. We will describe here how our work has evolved since then into a *client-server distributed hypermedia database*, called HOED (sections 2). The HOED server and its distinctive characteristics are discussed in section 3; the client environment is described in section 4. Two application areas are covered in section 5. After a comparison with related work in section 6, we conclude with a description of the current status and our plans for the future.

2 Architecture

The design of HOED is based on a client-server approach.

- The *server* takes care of data management and is itself distributed. A *hypermedia database* is responsible for management of the structural aspects. This hypermedia database is implemented on top of a *multimedia database* that takes care of content management, based on a mechanism that includes a type indicator (e.g. string, date, GIF image, etc.) with each multimedia value. Analogue and digital *raw data stores* offer physical storage facilities for the multimedia data.
- The *client* interacts with the end user or application program, initiates requests to the server, and displays the query results. In this way, the client hides details regarding the distributed nature of the server, the communication protocol and the database query language.

The design and implementation of the multimedia database and raw data stores has been covered in [Duval & Olivié, 1993]. The next section will present the server in more detail. Section 4 covers the client environment.

3 HOED Server

3.1 Data Model

The structural unit of data in HOED is the *node*. The primary function of a node is to contain multimedia content: text, computer software, audio or video streams, etc. In contrast, the primary function of a *link* is to contain structure (see also section 3.2).

An unlimited number of arbitrary (attribute,value) pairs can describe characteristics of nodes, links, or the participation of nodes in links. At any time, new (attribute,value) pairs can be added to the database. Some attributes are predefined, e.g. 'Type' that types nodes, links and participations of nodes in links. For particular types of these constructs, additional attributes can be predefined:

- We have e.g. defined a specific type of nodes for images on an analogue video disk about dentistry, produced at Bristol University (see also section 5). A predefined set of attributes describes the common properties of the material presented on these images: breed, sex, species, stain, etc.
- Specific types of links model specific types of spatio-temporal aggregation. 'Sequential aggregation' e.g. models a composite node as an ordered sequence of components.

3.2 Separation of Structure and Content

In conventional hypermedia systems (exemplified by e.g. Hypercard and Toolbook), content and structure are inextricably intertwined: links are embedded in document nodes. In HOED, links and nodes are separated for several reasons:

- *Maintenance* [Kappe, Maurer & Scherbakov, 1993] is accomodated by separation of structure and content. If e.g. a node is removed from the hypermedia database, then all links to the node that is being removed can also be removed, in order to avoid dangling references. This does not require a modification of any of the remaining nodes.
- Nodes and links can be *created, modified and deleted independently*: no conversion process is required to import new nodes, as the link information does not need to be imported into the document. This facilitates integration of separately developed tools: node editors e.g. don't need to provide link editing facilities. A new node can readily be integrated in the overall hypermedia network: new links can be added to existing nodes, so that these refer to the new node, without requiring a modification to the actual content of these nodes. An important consequence is that read-only documents (e.g. stored on a CD-ROM) can be linked to and from.
- When links are embedded in nodes, it is very difficult to support a *back-link capability* that identifies the links to a node [Engelbart, 1990]. A search through all nodes in the universe in order to identify those that contain a link to the node in question will be very time consuming. Incorporation of the links to a node in that node itself requires an update of the node whenever a new link to it is created or an old link is deleted. When links are stored separately, then the hypermedia database can simply be searched for all links pointing to a particular node.

3.3 Data Management

Some systems that do separate structure and content rely directly on a file system for data management purposes [Wiil, 1993]. Using a DataBase Management System (DBMS) for this purpose offers many advantages:

- The DBMS can take care of consistency, concurrency and access control, backup and recovery, data integrity and distribution.
- The query facility of the DBMS can be used to support:
  - *query based access*: This is especially relevant in a large scale networked hypermedia context, where navigation alone will no longer be appropriate if information overload is to be avoided [Duval & Olivié, 1992].

- *Intensional links* [Kappe, Maurer & Scherbakov, 1993]: Conventional links list explicitly the identity of the target node(s) of a l'nk. Intensional links are defined by a set of search conditions, and refer to all nodes satisfying the conditions at the time of evaluation. They are therefore dynamic, in the sense that they can refer to different sets of nodes at different moments in time.
- *link filtering*: Search criteria can be defined to identify those links that are relevant to a particular group of users, e.g. in terms of user profiles that indicate their areas of interest, their level of expertise, etc. This is especially relevant in an educational context: e.g. once a student has obtained a sufficient grade in a particular test, more and more of a global hypermedia structure of that domain may be revealed to him.

4 HOED Client

We envision different kinds of clients accessing the HOED server. *Node editors* support creation, modification or deletion of multimedia node content. Typical node editors include CAD tools, authoring systems, text editors, etc. *Link editors* can be used to create links between nodes. A *browser* enables end users to explore the HOED information space, following links between related nodes. A *searcher* enables developers to specify search constraints that identify the resources they are looking for. The constraints can be composed of arbitrary complex combinations of simple constraints, using the boolean connectors AND, OR and NOT.
We are currently developing two sets of client components:

- A first set of components is under development in Tcl/Tk [Ousterhout, 1990], a user interface management system available from Berkeley. Tcl/Tk provides facilities for communication between applications, so that the different client components will be able to exchange messages. As indicated by the bottom arrow in figure 1, the client components interact with the HOED server using SQL, the standard query language for interaction with an RDBMS.

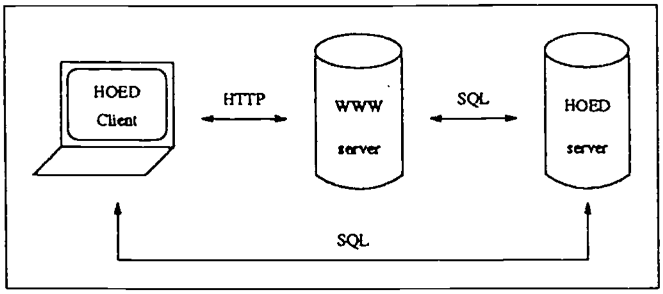

Figure 1: Gateway to the World-Wide Web

- Because of the enormous amount of documents available through the WWW, its rapidly increasing popularity and the free distribution of WWW software (both on client and server side, for many different platforms), we are developing a second set of client components, based on forms stored on a World-Wide Web (WWW) server (see also section 6) that interacts with the HOED server through a gateway.

  We rely on the Common Gateway Interface that defines a protocol for interaction between a WWW server and a non-WWW server. In this set-up, users access the HOED server through a WWW client (e.g. Mosaic or Cello) that interacts with the WWW server, using the HyperText Transfer Protocol (HTTP), as indicated on figure 1. The WWW server composes an SQL query for the HOED server, sends the query to that machine and presents to the end user the results that he receives. In this way, the WWW server is used as a "user interface server", as it holds the definition of the forms that define the screen lay-out and functionality during interaction. The more basic hypermedia storage and retrieval facilities are provided by the HOED server.

5 Application Areas

5.1 Basic Computer Science Modules

Since September 1993, we have started the development of a modular private study course on basic computer science topics for university students that do not need an in depth education in this field [Olivié & Duval, 1993]. The modular design of the course enables us to take into account the diverse background of the students and the peculiarities of the curricula in different faculties of the university. The modules will be stored as collections of linked multimedia nodes in HOED. Access to the modules can proceed in different ways:

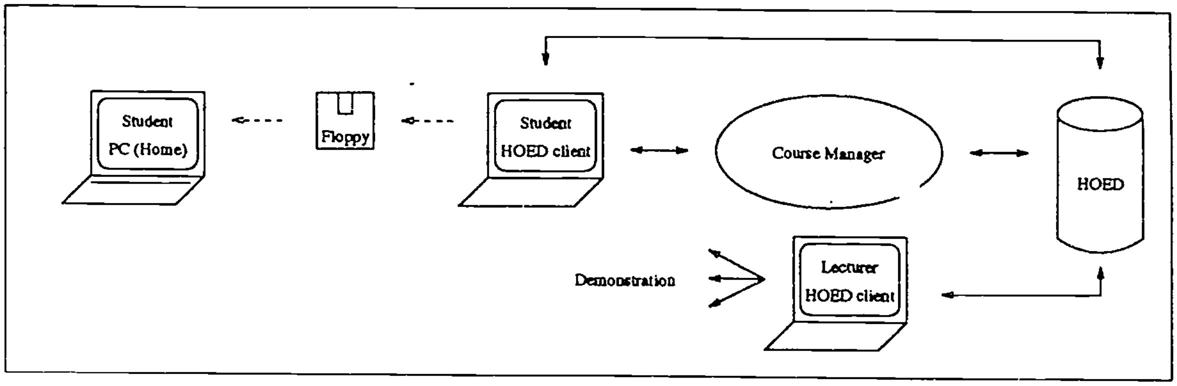

Figure 2: Basic Computer Science modules: Overview of Interactions

- A *course manager* is currently under development. This software component will support student evaluation and guidance. Pre-tests, based on intensional links (section 3.2) in terms of a student model, can refer a student to the most suited module.
- Bypassing the course manager, students can also navigate more freely through the learning resources in HOED, just as they can e.g. make use of the university *library*. This will enable them to explore the educational material, to pursue their own interests, or in order to find additional background material.
- A lecturer can retrieve (part of) a module from HOED in order to use it for demonstration purposes.

Both students and lecturers can access HOED over KULnet, the K.U.Leuven university computer network. Students are allowed to copy some of the educational material on floppy disk, so that they can use it on their own PC at home. This greatly increases availability of the resources for the students, the more so as a survey we carried out in 1993 among first year students in Engineering, Mathematics, Physics and Informatics showed that 82% of these students had access to a PC at home.

5.2 Re-Use of Courseware Components

In the CAPTIVE project ('89-'91) [Duval, 1992], aimed at 'Collaborative Authoring, Production and Transmission of Interactive Video for Education', we experimented with re-use of existing courseware components when developing new educational material. Video footage originally shot for a linear video e.g. was later on integrated in an interactive video. When trying to re-use existing resources, developers of new material face a number of problems. HOED provides at least a partial solution to three of these problems:

- Developers need to be able to find out *what* is available and whether they can use it for their own purposes. For the latter question, different considerations relate to the technical characteristics of the existing resources, copyright and pricing issues, the content presented in a (component of a) module, etc. This search can be supported by the query and navigation tools covered in section 4.

  Not only developers of new material can benefit from this facility: students or lecturers can also use it to access relevant material. In fact, providing this kind of access is the first aim of our collaboration with a team at *Bristol University*. In a first phase of this collaboration, ca. 250 images on a dentistry laser video disk have been described. The descriptions (including the location of the images on the disk) have been stored in the HOED server. When a relevant node has been identified, users can see the corresponding image, using a local tool for access to the video disk.

- Once the relevant resources have been identified, they can easily be accessed, using the client/server communication protocol. Developers need not be aware of *where* the material is physically located. In order to further refine the results of a query search, the nodes retrieved from the server can be used as a starting point for local navigation. in order to explore related material.
- Integration of existing resources is accomodated by the separation of node and link information: links can be created, modified and deleted *independently*. (see section 3.2).

The resulting mechanism for sharing of (parts of) nodes resembles Ted Nelson's idea of *transclusions*, or virtual copies [Nelson, 1993]. As described by Nelson, a mechanism can be established to compensate an author with a relatively small fee, every time one of his nodes is accessed. This approach may solve at least partially the copyright issue that often complicates re-use of independently developed resources.

6 Related Work

HOED can be compared with a number of recent client-server systems for networked information management over the internet [Obraczka, Danzig, & Li, 1993]:

- *Wide Area Information Servers* (WAIS) provide full text searching functionality with relevance feedback, i.e. the user can indicate relevant documents in an intermediate result, and WAIS locates similar documents that are likely to be relevant as well.
- *Gopher* provides access to information through a hierarchical menu system. Selection of a menu entry causes a sub-menu or document to be retrieved and displayed.
- The *World-Wide Web* (WWW) is an attempt at creating a world wide hypermedia system of linked documents. Links cannot only point to documents, but also to internet services such as remote login or file transfer.
- *Hyper-G* [Kappe, Maurer & Scherbakov, 1993] is a research project at the University of Graz.

WAIS and Gopher belong to computing paradigms (full-text search and hierarchical menu access respectively) that are fundamentally different from the hypermedia approach. The main disadvantage of the WWW is that links are embedded in documents (see section 3.2. As mentioned in section 4, we do however use the WWW interactive forms facility to support a set of client components that access the HOED server through a gateway. Both the overall aim and design of Hyper-G are more similar to ours.

7 Current Status and Plans for the Future

At the time of writing (March 1994), development of the HOED server is well advanced: although still at a prototype stage, it is now operational.The original conceptual data schema (developed in the Enhanced Entity Relationship model) was translated into a corresponding relational database schema, which was implemented, using a commercially available RDBMS. We have also developed prototype client components, including a simple query facility that enables end users to retrieve information about nodes, based on (boolean combinations of) simple search constraints regarding the characteristics of nodes, links or the participation of the former in the latter. Our plans for the immediate future focus on:

- As mentioned in section 5, we are working on different applications that will make use of the HOED environment. The development of *basic computer science modules* started in September 1993. Experiments for *access to local resources*, in collaboration with the university of Bristol, started in the summer of 1993. We are also further developing more client components.
- We will integrate within the HOED server facilities for Computer Supported Cooperative Work (CSCW) and collaborative learning. In a first phase, we will provide an annotation facility. Lecturers can use this facility to collaborate on course material and make their contributions accessible for lecturers and students in other institutions. Students will also be able to annotate their courses, e.g. as an aid while working through the material.
- The HOED server currently supports the basic hypermedia data model (links, nodes and webs). We are investigating how more advanced *data models* [CACM 1994] can be supported on top of the current model. In fact, modeling hypermedia data is not a trivial matter, because of the complex structure and multimedia content of hypermedia objects. This is therefore an appropriate case study for different data modeling paradigms. We have not elaborated on this issue here, but this is one of the reasons why we originally became interested in the HOED project.

8 Conclusion

Distributed, large scale multi-user hypermedia environments provide many exciting challenges in the area of data modeling, user interface development, communication protocols, information retrieval, etc. We are often excited about the *rapid developments* in these fields and the new applications they lead to.
The opportunity to experiment with novel ways to deliver education is much more than just a practical outcome of these technical developments. The prospect this offers for innovation of the *educational practice* is quite a powerful stimulus for continuing our efforts.

References

[CACM 1994] Special issue on hypermedia. *Communications of the ACM*, 37(2), February 1994.

[Duval & Olivié, 1992] Towards the integration of a query mechanism and navigation for retrieval of data on multimedia documents. *ACM SIGIR FORUM*, 26(2):8–25, Fall 1992.

[Duval & Olivié, 1993] A distributed management system for multimedia and hypermedia educational resources. *Proceedings of ED-MEDIA 93*, pages 161–168. AACE.

[Duval, 1992] An 'intelligent' database system of audio-visual material. *Proceedings of the DELTA Conference on Research and Development*, pages 231–241. Kluwer Academic Publishers

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

[Kappe, Maurer & Scherbakov, 1993] Hyper-G – a universal hypermedia system. *Journal of Educational Multimedia and Hypermedia*, 2(1):39–66, 1993.

[Nelson, 1993] *Literary Machines*. Mindful Press, 1993.

[Olivié & Duval, 1993] Basic computer science modules. In *CBLIS 93, International Conference on Computer Based Learning in Science*, pages 20–28, 1993.

[Obraczka, Danzig, & Li, 1993] Internet resource discovery services. *IEEE Computer*, 26(9):8–22, September 1993.

[Ousterhout, 1990] An X11 toolkit based on the tcl language. In *Proceedings of the 1991 Winter Usenix Conference*.

[Wiil, 1993] Experiences with hyperbase: A hypertext database supporting collaborative work. *SIGMOD Record*, 22(4):19–25, December 1993.