REQUIREMENTS ANALYSIS AND DESIGN FOR A FLEXIBLE LEARNING-ON-DEMAND SYSTEM

Ketil Lund, Pål Halvorsen, Vera Goebel‡, and Thomas Plagemann

UniK - Center for Technology at Kjeller
University of Oslo
Postbox 70
N-2027 Kjeller
Norway
{ketillu, paalh, plageman, goebel}@unik.no

ABSTRACT

Interactive distance learning (IDL) is an evolving paradigm of instruction and learning that attempts to overcome both distance and time constraints found in traditional classroom learning. The electronic classrooms at two sites of the University of Oslo and two further sites in Norway overcome separation in space by exchanging digital audio, video, and whiteboard information using the national academic ATM network. However, a limitation of these classrooms, respectively this type of application, is that it is limited to synchronous IDL, i.e., the students must be present in one of the classrooms during the lecture. To overcome this limitation we currently develop a Learning-on-Demand (LoD) system to provide asynchronous learning capabilities to the electronic classroom. With such a system, students will be able to follow lectures from networked PCs whenever they want. Based on today's electronic classroom, we analyze the requirements for such an LoD system, describe its design, and present the current state of our work.

INTRODUCTION

Interactive distance learning (IDL) is an evolving paradigm of instruction and learning that attempts to overcome both distance and time constraints found in traditional classroom learning. According to (Noam 1995), a large shift in the role of the university is about to happen: "many of the physical mega universities ... are not sustainable, at least not in their present duplicative variations." He predicts that in "ten years from now a significant share of conventional mass education will be offered commercially and electronically".

Learning-on-Demand (LoD) is one valuable approach for IDL and it extends conventional educational programs, as it introduces the idea of learning by doing. This creates an environment where learners have the necessary tools (computational or otherwise) to explore different information spaces, and obtain contextualized and relevant information. From a technical point of view, LoD is possible in the near future because of the fast technological development making it possible to handle the vast amount of necessary data and to make it available all around the world via the various existing communication technologies (ranging from Gigabit ATM networks to GSM). Thus, LoD becomes important in learning processes, because it makes the right information available at the right time.

At UniK (Center for Technology at Kjeller), we have used electronic classrooms for IDL since 1993, teaching regular graduate level courses. These classrooms overcome geographical separations of the different classrooms by exchanging video, audio, and whiteboard data. However, they only support synchronous IDL, i.e., only live, on-line teaching is possible where students and teacher must physically be present in one of the classrooms during the time the lecture is given.

In order to additionally support asynchronous IDL, i.e., separation in time and space, we are developing an LoD system, enabling students to interactively retrieve and play back information and (parts of) stored lectures whenever they want. The LoD system will store all data from the electronic classroom in a multimedia database system (MMDBS). Data of each type (e.g., audio, video, and transparencies) as well as meta-data like synchronization information and annotations, are independently stored to allow users to retrieve only those data the user is interested in. User needs play a central role in the design of the LoD system, because they determine at run-time which data should be retrieved and presented, and in which quality the data should be presented. Thus, Quality of Service[1] (QoS) specification, and especially the support of QoS within the LoD system, are important issues. In the context of networking and operating systems, QoS is since several years a well-studied

‡ Vera Goebel is now at the University of Oslo

[1] QoS determines the quantitative and qualitative requirements which a system should satisfy in order to achieve the desired quality.

research topic. However, in the context of MMDBS, QoS support has only very recently been recognized as an important area of research (Goebel et al. 1999), (Halvorsen et al. 1998), (Walpole et al. 1999).

The technological solutions needed to realize such an LoD system can be grouped around two fundamental questions:

1. *How to organize the data, i.e., modeling, distribution, storage, etc.*? So far, we have implemented a data model for the lecture data, and we have designed a buffer preloading and replacement mechanism, a continuous media file system, and a scheme for data placement on disks.
2. *How to make the data available to the users*? In our LoD system, the lecture data will be available to the users through an MMDBS. Thus, an appropriate client interface, synchronized playback, network support, and query processing and optimization are needed. In our work at UniK, we address all these issues.

It is the aim of this paper to identify the technical requirements of such an LoD system and to describe its design. The entire discussion is focused on MMDBS aspects of the LoD system, because of today's lack of solutions for QoS support in MMDBSs. The remainder of this paper is organized as follows: First, we describe our synchronous IDL system, because it determines the type and amount of data that must be handled by the LoD system. Next, we present the visions and goals for our LoD system, in addition to a requirements description. We then describe the system, and finally, we summarize with the conclusions and future work.

SYNCHRONOUS IDL

Distance education refers to all types of studies in which students are separated by space and/or time. The electronic classrooms (Bakke et al. 1994) overcome separation in space by exchanging digital audio, video, and electronic whiteboard information between two or more sites. Since 1993, the electronic classrooms have been regularly used for teaching graduate level courses as well as for research on QoS support in distributed multimedia systems (Plagemann & Goebel 1998).

During a lecture, at least two electronic classrooms are connected. Teacher and students can freely interact with each other regardless of whether they are in the same or in different classrooms. This interactivity is achieved through the three main parts of each electronic classroom: *electronic whiteboard*, *audio system*, and *video system*. All participants can see each other, can talk to each other, and may use the shared whiteboard to write, draw, and present prepared material from each site. The electronic whiteboard, audio, and video system in turn consist of several components. In Table 1, these components are briefly described and denoted with a component identifier (CID). The following description of the electronic classroom refers to the components via their CID. Figure 1 illustrates the basic layout of an electronic classroom, and Figure 2 shows two pictures from a local and a remote classroom that are taken during a lecture. In addition to the ordinary classroom structure that is visible in these pictures, i.e., student and teacher area, a technical back room is located behind the classroom.

The *electronic whiteboard* is a synonym for a collection of software and hardware elements to display and edit lecture notes and transparencies that are written in Hypertext Markup Language (HTML). The whiteboard itself **(1)** is a 100" semi-transparent screen that is used together with a video projector **(10)** and a mirror **(9)** as a second monitor of an HP 725/50 workstation **(8)** in the back room. A light-pen **(3)** is the input device for the whiteboard. A distributed application has been developed that can be characterized as a World-Wide Web (WWW) browser with editing and scanning features. When a WWW page is displayed, lecturer and students in all connected classrooms can concurrently write, draw, and erase comments on it by using the light-pen. Thus, floor control is achieved through the social protocol - as in an ordinary classroom - and is not enforced by the system. Furthermore, a scanner **(7)** can be used to scan and display on the fly new material, like a page from a book, on the shared whiteboard. The entire application can be managed from the workstation in the classroom **(7)**.

The *audio system* includes a set of microphones mounted on the ceiling. They are evenly distributed in order to capture the voice of all the participants and to identify the location of the loudest audio source in the classroom. Furthermore, the teacher is equipped with a wireless microphone. To generate a digital audio stream, two codecs are available: the audio codec from the workstation **(8)** and the audio part of the H.261 codec. Thus, one of the following three coding schemes can be selected: 8 bit 8 Khz PCM coding (64 Kbit/s), 8 bit 16 Khz PCM coding (128 Kbit/s), and 16 bit 16 Khz linear coding (256 Kbit/s). Speakers are mounted on the ceiling to reproduce the audio stream from the remote site.

The *video system* comprises three cameras, a video switch, a set of monitors, and a H.261 coding/decoding device (codec) to generate a compressed digital video stream. One camera **(5)** is automatically following and focusing on the lecturer. The other two cameras **(4)** capture all events happening in the two partly overlapping parts of the student area in the classroom. The audio system detects the location of the loudest audio source in the classroom, i.e., a student or the teacher that is talking. A video switch selects the camera that covers this location in order to produce the outgoing video signal. Two control monitors **(6)** are placed in the back of each classroom. The upper monitor displays the incoming video stream, i.e., pictures from the remote classroom, and the lower monitor displays the outgoing video stream, i.e., video information from the local classroom. Thus, the teacher can see the students in the remote classroom and can check the outgoing video information while facing the local students. The students in turn can see the remote classroom on a second large screen which is also assembled out of a whiteboard **(2)**, a video projector **(10)** that is connected to the output of the H.261 codec, and a mirror **(9)** in the back room.

VISION AND GOAL

Despite the distributed nature of the electronic classrooms, they support only synchronous IDL. Additionally, there are only a few electronic classrooms in Norway today. Thus, students have to meet in one of the classrooms at the time the lecture is given. To change this situation and to support asynchronous IDL, we are building an adaptive distributed multimedia system for LoD. This system is based on an MMDBS with QoS support which is currently being developed in the OMODIS project (Plagemann & Goebel 1998), (Goebel et al. 1996), (Goebel et al. 1999), (Goebel et al. 1998), (Halvorsen et al. 1998). We plan to store recorded lectures and additional information relevant for the courses in the MMDBS. The data can be retrieved later by the students and teachers. There will also be support for editing of the stored lectures.

In our LoD system, each user can have an individual on-line access to the MMDBS. To find and present the requested data, flexible query and playback facilities must be provided. There must be several possibilities to find lectures, e.g., using the date of the lecture: `give me the lecture from 17/3/99`, using a contents description: `give me all lectures in which the teacher discussed MPEG4`, or using the name of the teacher: `give me all lectures from course Modern Operating Systems taught by Vera Goebel`. Furthermore, it should be possible to search for particular parts of a lecture, for example by content description as shown above or by reference to particular transparencies: `give me the part of lecture X where the teacher explained transparency Y`. Another type of query can be used to collect handouts: `give me a pdf file that includes the handouts from lecture X to lecture Y`.

The students must have the possibility to select only some parts of the different multimedia data, e.g., a student doing only a quick repetition may select just the sound, in order to hear the teachers explanation: `give me audio from lecture X` (neither video nor transparencies are presented). Generally, user requirements and special user needs, e.g., deaf or blind students, can be better supported by selecting only the appropriate media types. The user should also be able to play back the different multimedia data types with a scalable quality, e.g., due to limited resources or the user's requirements: `give me full quality audio and low quality video from lecture X` (e.g., reduced frame rate). To provide this functionality, the user interface must support the most usual type of queries in a very intuitive form and in addition, provide the possibility to write ad hoc queries to the MMDBS using the provided query language.

When the desired lecture data is located, the user should be able to interactively play back the lecture using ordinary VCR operations, e.g., play, fast forward, rewind, and pause. The data streams will be transmitted using IPv4 and IPv6 over various types of transmission networks like ATM, ISDN, analog modem lines, or GSM.

During a live lecture, there exist temporal and spatial relationships between the media types, and the presentation of the stored lecture should be fully synchronized according to these relationships, independent of whether all elements are retrieved or only a subset.

For the students, there are two main advantages with such an LoD system:

1. *Independence of where and when the lecture is taking place*. The students can follow a lecture at the time and place that suit them best, provided there is a computer with a network connection available.
2. *Opportunity to repeat topics*. The students can search for certain topics and have them repeated. A possibility of assembling sequences from several lectures treating the same topic, can help the student to get a broader understanding of the topic.

From the teachers point of view, we envision the following scenario: During the lecture, the "raw" data is stored on disk. After the lecture, the teacher edits the recorded lecture, e.g., removes unnecessary parts, changes the audio in some parts (for instance to improve an explanation), or adds a second soundtrack in another language. When the editing is finished, the lecture is stored in the database and made available for the students. Editing of existing lectures is done by checking the lecture out of the database, updating it, and checking (storing) it into the database again. Hence, the main advantages for the teacher are the possibilities of improving the lecture before it is stored in the MMDBS and editing/updating of previously stored lectures.

Based on the described functionality, an optimal, technical solution is using an MMDBS with QoS support running on a real-time operating system. To the best of our knowledge, there are no systems today offering this functionality. Existing systems can be classified into the following categories:

- *Synchronous IDL*: The majority of the existing IDL systems like (Eckert et al. 1997), (Grebner 1997), (Maly et al. 1996), (Stiller et al. 1997), only support synchronous IDL. Consequently, they are not relevant in this context.
- *Interactive, asynchronous IDL*: There are a few commercial LoD systems available supporting asynchronous IDL, e.g., IBM's Higher Education

Program[2], LearningSpace from Lotus[3], and Lucent Technologies[4], but most of these systems offer only a set of web pages which can be accessed over the Internet. The contents of these web pages can be compared to on-line manuals and tutorials containing training programs, installation and maintenance information, and help for troubleshooting. As they do not support continuous, time dependent data types like audio and video, their functionality is limited compared to our LoD system.

- *MMDBS*: There is a lot of on-going work addressing multimedia support in DBSs, e.g., the OASYS projects at GMD Darmstadt[5], the MUDS project at the University of Maryland[6], the object-oriented SGML/HyTime compliant multimedia DBS from the University of Alberta (Özsu et al. 1997), the multimedia DBS activities at Case Western Reserve University[7], and at the University of Buffalo[8]. However, there are no approaches which fully support QoS and real-time aspects in MMDBSs.
- *Multimedia systems with QoS support based on plain file systems*: There is a lot of ongoing work in the area of storage and presentation of continuous, time-dependent data like audio and video. In the MARS project at Washington University, St. Louis (Buddhikot 1998), off-the-shelf components and enhancements to the 4.4 NetBSD UNIX operating system, e.g., zero-copy data path and guaranteed CPU access, have been used to build a multimedia-on-demand server. The multimedia whiteboard described in (Stiller et al. 1997), allows an interactive, synchronized playback of recorded video and audio to the clients' workstations. The LAVA project[9] focuses on IDL by transmitting stored video data. In (Walpole et al. 1999), a QoS model for multimedia databases is introduced. This model covers the specification of user-level QoS preferences and their relationships to QoS control at the resource-management level. However, it is only intended for data stored in file systems, and it does not cover database management systems. Since all these systems use only plain file systems, they do not support functionality like query processing and transaction management. A file system leaves formatting and management of the large amounts of data to the application. Using a DBS provides mechanisms like data independence (data abstraction), high-level access through queries, controlled multi-user access (concurrency control), and fault tolerance (transaction management, recovery). Furthermore, using a file system, natural relationships for a synchronized playback of multimedia data must be modeled and stored separately whereas in a DBS this information can be modeled explicitly as part of the data itself. One of the strengths of a DBS is the ability to manage data in a uniform way on the behalf of multiple applications. A DBS hides data organization and distribution from the application and enables data sharing in a transparent manner (Özsu et al. 1997).

A different approach to IDL is reported in the iTeach project[10] from Darmstadt University of Technology. A multibook, i.e., an electronic textbook, is used for a more experimental learning process. The user is guided through a lecture describing a topic with the help of text, pictures, and animations where the chapters are generated on the fly based on a user profile and the actions the user performs. The iTeach approach is complementary to our LoD system. Therefore, in order to support asynchronous playback of lectures and interactive studying, an optimal solution might be a combination of our system and the electronic, multimedia textbook from the iTeach project.

REQUIREMENTS

The LoD system involves a lot of interaction between user and system, allowing the user both to submit queries and control the playback. As the data of a lecture consists of several multimedia data types, support for synchronous playback of continuous, time-dependent data streams is needed. This also means that support for QoS is important, and as far as possible, user and application requirements must be considered in the MMDBS, operating systems, transport protocols, and network. Users can specify their requirements in each query or the system will use a particular user profile as default. The requested QoS might also be renegotiated according to available resources.

Playback of stored lectures to a large number of concurrent users requires an efficient retrieval and processing of the independent multimedia data types. The storage manager must optimize data layout on disk to increase disk I/O. The different data elements are stored separately to support presentation of only the requested data elements. Table 2 summarizes the coding used in

[2] IBM's Higher Education Program, On-Demand Learning sector: http://www.hied.ibm.com/odl

[3] LearningSpace, Lotus, http://www.lotus.com/home.nsf/welcome/learnspace

[4] Lucent Technologies, Center for Excellence in Distance Learning (CEDL): http://www.lucent.com/cedl

[5] GMD Darmstadt, The OASYS projects, http://www.darmstadt.gmd.de/oasys/projects/index.html

[6] University of Maryland, Department of Computer Science, the MUDS (Multimedia Database System) project, http://www.cs.umd.edu/users/vs/mm/index.html

[7] Case Western Reserve University, Department of Computer Engineering and Science, Database Systems Research Group, http://erciyes.ces.cwru.edu/mmedia.html

[8] University at Buffalo, New York, Department of Computer Science and Engineering, Database and Multimedia Group, http://www.cs.buffalo.edu/pub/WWW/DBGROUP/index.html

[9] LAVA project: http://www.nr.no/lava/lava-u

[10] Interactive Multimedia Teaching (iTeach): http://www.kom.e-technik.tu-darmstadt.de/Research/mmbook

today's electronic classroom and the respective bandwidth and throughput requirements of the different multimedia data types, e.g., transparencies, video frames, audio samples, based on measurements from a live lecture (Sæthre 1996). However, the H.261 coding is designed for live data transmission and is not appropriate for storing data (Steinmetz & Nahrstedt 1995). Therefore, in our scenario, we use Motion-JPEG to perform a worst-case analysis. Each video frame is about 60 KBytes large, i.e., we have a bandwidth requirement (both for I/O and network) of about 14.4 Mbit/s per stream for a 30 frames/second video and a disk storage requirement of about 9.7 GBytes for a video of a 90 minutes lecture. The bandwidth requirement of PCM coded audio is 256 Kbits/s per stream, and the need for storage space is approximately 172.8 MBytes. In total, the aggregate bandwidth and storage requirements for a lecture (video and audio) are about 14.7 Mbits/s (per user) and 9.9 GBytes respectively[11]. Using a parallel disk system, the storage space is not a big problem, but depending on the number of concurrent users, both I/O and network bandwidth, buffer space, and CPU resources are important issues. Without proper storage and buffer management in addition to efficient disk and CPU scheduling, these components are potential bottlenecks.

Not only must these potential bottlenecks be dealt with, there are also requirements arising from available resources and the users' preferences. Below, we list some of these requirements resulting from the functionality described in Section 3:

- Since users may access the system through different networks (e.g., ATM or GSM), the system must be able to transmit the same multimedia data type with different bandwidth requirements, e.g., by varying the frame rate and/or the frame resolution of a video stream. In addition, each client specifies QoS requirements according to available resources, e.g., low resolution video due to little available network bandwidth, and personal preferences.
- The system should support resource reservation and monitoring to provide QoS guarantees for each individual user. Consequently, admission control is necessary in order to avoid system overload.
- To support flexible query facilities, we need to be able to perform advanced queries on structured data. Consequently, we need a detailed and flexible data model. Since the queries normally will be performed on meta-data describing the audio and video streams, the data model must have extensive support for meta-data.
- Synchronization between multimedia data elements with temporal relationships (inter-object synchronization) and within each multimedia data type (intra-object synchronization) must be supported. Thus, we have real-time requirements of data delivery and delay jitter must be minimized.
- As the system should support a high degree of interaction, it is important that delays and response times are kept as low as possible.

[11] Motion-JPEG and PCM are used for the requirement analysis for video and audio respectively. However, less bandwidth and storage demanding codecs like MPEG might be used instead

DESIGN

In this Section, we present the design of the LoD system we are developing. First, we present the overall architecture, before we describe the requirements and current status of each component in more detail.

OVERALL ARCHITECTURE

The application at the client side has to enable users to interactively browse through lectures, to use predefined queries, to formulate queries, and to specify QoS requirements as part of queries and as part of a general user profile. Furthermore, the client has to support QoS and synchronized playback.

On the server side, a multimedia database management system (MMDBMS), running on a real-time operating system, constitutes the LoD server. This MMDBMS will consist of a data model, a query processor and optimizer, a transaction manager, and a storage manager. The storage manager consists of a file system and index structures, buffer management, and mechanisms for optimized data placement (physical storage management).

All transparencies and scanned images that are used in lectures, the interactions with the whiteboard, as well as video and audio streams from the different electronic classrooms have to be separately stored in digital format in the MMDBS. Users can retrieve the information via network, and it is presented in a synchronized manner using the multimedia enabled client application.

Figure 3 shows the architecture of the final prototype. However, the first prototype will be implemented using COTS (commercial off-the-shelf) technology. On the server side, we will use ObjectStore as DBS, while on the client side, we intend to use a public domain multimedia enabled browser. The client and server will communicate using TCP, UDP and RTP over IPv4 and IPv6 (combined with RSVP). The realization of the LoD system as shown in Figure 3 will take place in the form of stepwise extensions and refinements of the COTS prototype.

DATA MODEL

To support flexible queries on the stored multimedia data, a detailed and flexible data model is necessary. It is important that the data model supports separate modeling of the different multimedia data types, and synchronization between these data types. Consequently, the data model must support both meta-data and temporal information.

We have developed a data model called TOOMM (Temporal Object-Oriented MultiMedia data model) (Goebel et al. 1999), and implemented it in Java as a layer on top of ObjectStore. TOOMM integrates temporal and spatial concepts into an object-oriented data model in addition to modeling and handling of advanced multimedia-related meta-data. Objects in TOOMM comprise the properties of traditional object-oriented data models and three different time dimensions: valid time, transaction time, and *play time*. The play time dimension places units of multimedia data, such as frames or audio samples, into a temporal structure for multimedia presentations. In addition, TOOMM is based on two principles: separation of multimedia data from its presentation specification and separation of multimedia data from its temporal and spatial information, and meta-data.

We have also extended TOOMM for better handling of non-temporal data, e.g., the transparencies. This extension is based on SGML and HyTime. We are currently working on the integration of this part.

QUERY PROCESSING AND OPTIMIZATION

The main problem of query processing on multimedia data is the heterogeneity of the data types. As a result, it is not possible to work out one single processing strategy that is optimal for all data types. Instead, different strategies must be used for each data type, and these strategies must then be combined while processing the data.

It is also important that queries can be easily formulated for all multimedia data types. This involves both predefined queries, and ad hoc queries written in a declarative query language. Since the queries will be made on multimedia data such as video and audio, i.e., content-based queries, it is necessary that the query language has mechanisms to support this kind of queries. Such content-based queries are usually supported through textual descriptions of the data, e.g., text describing the content of a scene in a video. The immediate result of such a query on meta-data is the location of the actual data, which is then played back to the user who issued the query.

Query by example is also a desirable feature in an LoD system. A student may ask for lectures on topics that are similar to a certain topic. Again the query is made on meta-data, selecting topics that are similar to the one presented in the query.

STORAGE MANAGEMENT

As shown in Section 4, the bandwidth requirements of our LoD system are relatively high, and not only the network, but also the storage system must be able to meet these requirements. Because the LoD system has a high degree of interaction, there is also a need for low response times. However, high bandwidth and low response times are considered contradicting optimization goals, and a good compromise must be found. We are working on several issues within storage management:

- In order to ensure QoS during retrieval of data from the disks, more advanced disk-scheduling algorithms than FIFO must be used. For this reason, we are working on a multimedia file system for the DBS which supports QoS for multiple concurrent users. In addition, we have designed a modification of the UNIX I-Node principle, that support contiguous allocation of disk blocks (Wang et al. 1999).
- Traditional buffer replacement algorithms are not suitable for multimedia data, and may lead to excessive I/O between main memory and disk. It is necessary to have algorithms tailored for multimedia data, and at UniK, we have developed a new buffer preloading and replacement algorithm for disk I/O, called Q-L/MRP (Halvorsen et al. 1998). The main feature of this mechanism is QoS support using a dynamic prefetching daemon. This daemon is able to dynamically adapt to changes in network and disk I/O load. Furthermore, QoS requirements from the users, like frame rate, are mapped to requirements for the buffer mechanism. To further increase the system performance, we are working on reducing the data copying operations between different protection domains, i.e., shortening the data path between the storage system and the host-network interface.
- Data placement describes the physical layout of data on the disks. It is an important issue in fulfilling the special needs of LoD (i.e., both high bandwidth and low response times). At UniK, we have developed a new scheme for placement of data in a disk array, that can meet these special needs. This scheme uses a disk array, which is divided into several groups with each group storing one multimedia data type. In this way, the efficiency of the disks is improved, as each disk receives fewer concurrent requests. The separate storage of the different data types also helps supporting the playback of individual multimedia data types (Lund 1997).

TRANSACTION MANAGEMENT

The majority of operations against the LoD server are read requests as students play back lectures. Only when the teacher stores a new lecture, or edits an old one, write requests are issued. Thus, the concurrency control requirements can be somewhat relaxed. Still, there is a need for transaction management, in order to ensure that data is stored correctly and provide support for concurrency control during writes:

- A new lecture is not made available until it is completely stored in the DBS.
- An existing lecture, which is to be edited, is checked out of the DBS. Students who access this lecture while it is checked out are notified that the lecture is being edited, but they are still allowed to access the

old version of the lecture, until the edited version is checked in.

- A lecture cannot be checked out more than once at a time, i.e., if one teacher has checked out a lecture for editing, another teacher must wait until the first teacher checks the lecture back in before he or she can check out the same lecture.

CONCLUSIONS AND FUTURE WORK

In this paper, we have described the components and the functionality of the LoD system under development at the University of Oslo, together with the electronic classroom, which serves as the basis of this work. We have presented our goals and visions of the LoD system, and we have analyzed the requirements that must be met to be able to achieve a satisfying QoS.

The LoD system will be subject to interdisciplinary research, looking into both technical and human factor related issues. Some of the human factor related questions we would like to investigate are:

- How sensitive are users to QoS, and what are the typical values users regard as acceptable/good?
- What are the typical queries users will submit?
- Will users concentrate on certain data elements, or will they always request all elements if resources permit?
- Will students use the system to learn at home?
- Will the system allow employees to follow courses?
- Will students use the system only if they missed class, or will they use it also for other reasons, e.g., for examination preparation?
- What is the impact of the LoD system on learning efficiency? Has QoS an impact on learning efficiency?
- How does the modeling/presentation of the multimedia data influence learning efficiency and the user acceptance of the LoD system?

However, it is not possible to do all at once, rather it will happen as stepwise extensions and refinements. At present, we are working on the first prototype of the system, and consequently, the current work is concentrated on the technical issues.

We consider an LoD system to be an improvement over traditional distance education since the degree of interaction is much higher, and the students get a feeling of being present in the classroom. Therefore, we believe that LoD systems are a valuable extension to traditional classroom teaching, especially if courses are taught by experts, i.e., there are no other persons that could teach the course, and travelling to the students is too expensive (with respect to time and/or money).

REFERENCES

Bakke, J. W., Hestnes, B., Martinsen, H., 1994, "Distance Education in the Electronic Classroom", Televerkets Forskningsinstitutt, report TF R 20/94

Buddhikot, M. M., 1998, "Project MARS: High Performance, Web Based Multimedia-on-Demand (MOD) Services and Servers", *PhD Thesis*, Sever Institute of Technology, Department of Computer Science, Washington University, Saint Louis, Missouri, USA

Eckert, A., Geyer, W., Effelsberg, W., 1997, "A Distance Learning System for Higher Education Based on Telecommunications and Multimedia", *Proc. ED-MEDIA/ED-TELECOM'97*, Calgary, Canada

Goebel, V., Eini, I., Lund, K., Plagemann, T., 1999, "Design, Implementation, and Application of TOOMM: A Temporal Object-Oriented Multimedia Data Model", *Proceedings of 8th IFIP 2.6 Working Conference on Database Semantics (DS-8) "Semantic Issues in Multimedia Systems"* (Kluwer), Rotorua, New Zealand, pp. 145 - 168

Goebel, V., Halvorsen, P., Lund, K., Mirzadeh, N., Plagemann, T., 1998, "Towards QoS Support in Multimedia Database Management Systems", Poster, ACM Multimedia 98, Bristol, UK

Goebel, V., Plagemann, T., Berre, A.-J., Nygaard, M., 1996, "OMODIS - Object-Oriented Modeling and Database Support for Distributed Systems", *Norsk Informatikk Konferanse (NIK'96)*, Alta, Norway, pp. 7-18

Grebner, R., 1997, "Use of Instructional Material in Universal Teleteaching Environments", *Computer Networks and ISDN Systems*, Vol. 29. No. 15, pp. 1787-1789

Halvorsen, P., Goebel, V., Plagemann, T., 1998, "Q-L/MRP: A Buffer Management Mechanism for QoS Support in a Multimedia DBMS", *Proceedings of 1998 IEEE International Workshop on Multimedia Database Management Systems (IW-MMDBMS 98)*, Dayton, Ohio, USA, pp. 162 – 171

Lund, K., 1997, “Techniques for Data Placement in Multimedia Database Systems” (in Norwegian), *Master Thesis*, University of Oslo, UniK

Maly, K., Wild, C., Overstreet, C. M., Abdel-Wahab, H., Gupta, A., Youssef, A., Stoica, E., Talla, R., Prabu, A., 1996, "Virtual Classrooms and Interactive Remote Instruction International", *Journal of Innovations in Education"*, Vol. 34, No. 1, pp. 44-51

Noam, E.M., 1995, "Electronics and the Dim Future of the University", *Science Magazine*, http://www.simslatham.com/stoneage.html

Plagemann, T., Goebel, V., 1998, "Analysis of Quality-of-Service in a Wide-Area Interactive Distance Learning System", *Telecommunication Systems Journal, Balzer*

Steinmetz, R., Nahrstedt, K., 1995, "Multimedia: Computing, Communications and Applications", Prentice Hall

Stiller, B., Walter, T., Hänni, H., 1997, "Télépoly: Synchronous Interactive Teleteaching over ATM between ETH Zürich and EPF Lausanne", in: *Proc. LearnTec 97*, Germany

Sæthre, K. A., 1996, "Distributed Multimedia Applications and Quality-of-Service" (in Norwegian), *Master Thesis*, University of Oslo, UniK

Walpole, J., Krasic, C., Liu, L., Maier, D., Pu, C., McNamee, D., Steere, D., 1999, "Quality of Service Semantics for Multimedia Dabase Systems", *Proceedings of 8th IFIP 2.6 Working Conference on Database Semantics (DS-8) "Semantic Issues in Multimedia Systems*" (Kluwer), New Zealand, pp. 393 – 412

Wang, C., Goebel, V., Plagemann, T., 1999, “Techniques to increase disc access locality in the Minorca multimedia file system”, Proceedings of the conference (part 2) on ACM multimedia '99, pp. 147-150

Özsu, M. T., Iglinsky, P., Szafron, D., El-Medani, S., Junghanns, M., 1997, "An Object-Oriented SGML/HyTime Compilant Multimedia Database Management System", in: *Proc. of the fifth ACM International Multimedia Conference (ACM MM'97),* Seattle, Washingston, USA, pp. 239 – 249

FIGURES AND TABLES

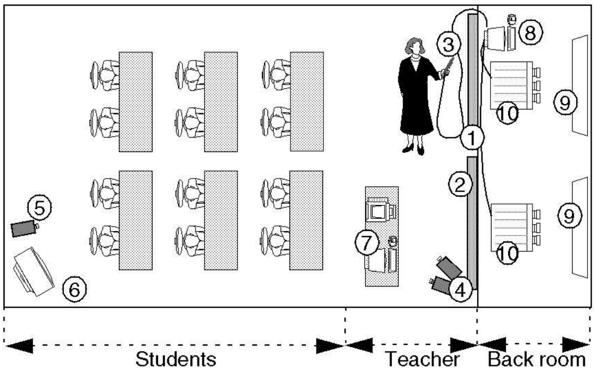

Fig. 1 Structure of an electronic classroom

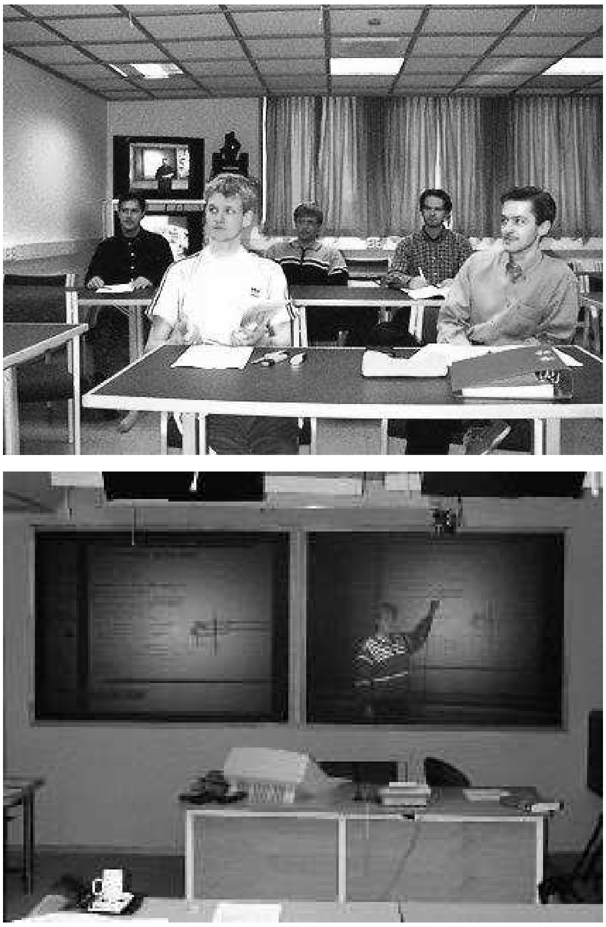

Fig. 2 Pictures from the electronic classroom during a lecture

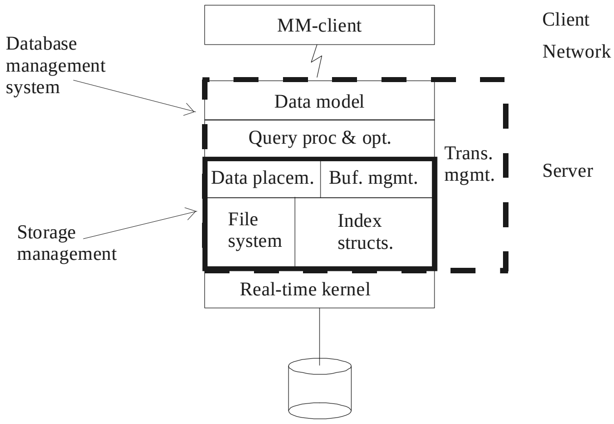

Fig. 3 Overall architecture of our LoD system

Table 1 Components of the electronic classroom

CID	Name	Description
(1) **(2)**	Whiteboards	100" large semi-transparent boards used to display screen information from the back room workstation and the output of the H.261 codec
(3)	Light-pen	Input device for the electronic whiteboard
(4)	Student cameras	Two cameras pointing at two partly overlapping parts of the student area
(5)	Teacher camera	Camera that is automatically following and focusing on the teacher
(6)	Control monitors	Two TV sets in the back of the classroom allow the teacher to see the incoming and outgoing video stream while facing the students.
(7)	Classroom workstation and scanner	Enables the teacher to control the application; the scanner can be used to scan material during a lecture and present it on the whiteboards
(8)	Back room workstation	Running the application, i.e., handles all input from the electronic whiteboard and performs all interactions with the peer application in the connected classroom(s)
(9)	Mirrors	Reduce space requirements in the back room, because they double the distance between video projector and whiteboard
(10)	Video projectors	Connected to the monitor output of the workstation in the back room and to the H.261 Codec respectively, and they project the input via a mirror onto the whiteboard

Table 2 Workload of the electronic classroom

Media Type	**Coding**	**Bandwidth Requirement**
Video	H.261	Max: 1950 Kbit/s; Min 13 Kbit/s; Avg: 522 Kbit/s
Audio	PCM	256 Kbit/s
Electronic Whiteboard	HTML	1600 Kbit/transparency