



A MODULAR TRAINING SYSTEM FOR EDUCATION IN THE WWW ENVIRONMENT



U. Schroeder, B. Tritsch, A. Knierriem-Jasnoch

Abstract

This paper presents the architecture and WWW-independent implementation of a Modular Training System for networked workstations and personal computers. The emphasis of this system is on telecommunication-centred interactive multimedia technology, extending training beyond individual classrooms. The media representation in a client/server environment, the person-to-person communication needs of such a system, and the dynamic flow control of educational courses are addressed. This training system is compared to the existing WWW environment and its extensions for educational uses, giving hints for new functionality to be incorporated into the individual WWW components. **Keywords:** *Computer-based training, modular training system, training and education on the Web*

Introduction

The increasing need to improve training and retraining in modern society is widely recognized. Using appropriate systems based on existing telecommunication services and information technology, this need can already be fulfilled in a certain degree of quality. Before the appearance of the *World Wide Web*, large production costs allowed only large companies to afford to develop adequate computer-based learning systems to train their technical staff. But the need for small- and medium-size enterprises to upgrade the knowledge of their employees is becoming more and more vital in order to meet a rapidly moving market. Moreover, many people need individual educational systems they can use from their homes, because education becomes an important factor in the work market. Responsibility for education is shifting from the employer towards the individual. This enhances the need for educational systems, which can also be used during spare time from the home PC.

Consequently, the WWW seems to be the adequate training platform for learning activities and many projects exploit its functionality for educational uses (see: Yahoo: Online Teaching and

Learning, The World Lecture Hall, and Education within The Virtual Library). Despite the integration of the hypermedia and wide-area network functionality, the WWW itself does not provide a sufficient educational framework to meet the requirements derived from the statements above. Therefore, the HyperScript project adds methods for systematic courseware development based on integration of learning objectives and their evaluation [7], and uses existing WWW technology like CGI scripts and CCI control to integrate sophisticated and specialized tools for programming education [6]. With this, we set up a learning environment incorporated into a practical software engineering course [1], where student groups also utilize the WWW for software documentation and document management. Moreover it is used to support their group communication. Other requirements include easy-to-use and robust training systems that allow the users to be trained at the right time, with the adequate learning material being accessible via network, with potential help of a human tutor, and within their sometimes limited hardware and software environment. The courseware must be designed according to a network-based system architecture and being adaptable to different user groups and

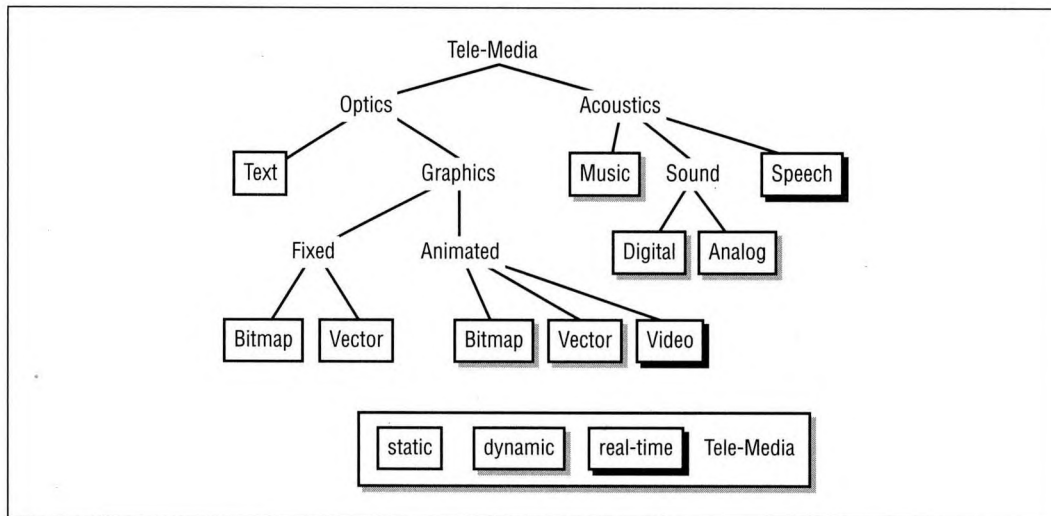


Figure 1: The Tele-Media objects

pedagogical aspects. Even before the WWW became popular, these reasons lead to the design and the implementation of the *Modular Training System* (MTS). Its major modules are a cross-platform learning front-end, a pedagogical back-end, a dedicated networking module to connect front-end and back-end over LAN or WAN, and a human communication component.

On the first glance, the goals of the WWW and the MTS differ: while the WWW is an information system also allowing distance learning to some extent, the MTS is an education and training system also allowing information access to some extent. For this reason, we want to look at some details of the MTS architecture and its components. This leads to a better understanding of the current state of the WWW with respect to teaching environments. Moreover we state how MTS features can be mapped onto the Web technology in order to better suite educational needs.

The Architecture of the Modular Training System

Educational software in a distributed, wide area environment requires a new architecture of the

computer-based training system. This architecture is component-oriented and is strictly following a client/server approach. It is derived from a reference model describing Tele-Communicating Multimedia Objects (*Tele-Media*) and their relations to transmission, synchronization, and real-time issues [9]. The Tele-Media model provides the mechanisms for the delivery and the presentation of the used multimedia objects. These Tele-Media objects represent learning material circulating through the different components of the distributed Modular Training System (see Figure 1).

In the following paragraphs, the individual components of the MTS architecture are introduced briefly. The *Generic Learning Support* (GLS) is a client application which creates the learning front-end. It presents the multimedia learning courses and allows the user to interact with the learning system. The *Course Domain* and the *Course Operating System* represent the pedagogical back-end. The course domain provides modular- and platform-independent course material of three levels. A dedicated server application, the *Course Interpreter* (CI), is the main component of the Course Operating System and supports course configuration and course execution.

The *Tele-Media Network Module* provides a network protocol and API functionalities, and enables the GLS to communicate with the CI. Finally, the *Human Communication Component* provides online audio and video tools for person-to-person communication on heterogeneous networks and platforms supporting active student-teacher communication.

We believe that a strict separation between these components is the key to courseware systems. Within the course domain, virtual components and methods can be used within the design of courses. This virtual material is mapped onto libraries of real learning materials residing in the course domain. Finally, within the Generic Learning Support, the real learning material is loaded and mapped onto the physical platform available to the student. Interfaces to standard tools for data access, transfer, presentation, and interaction are supported by the Tele-Media Network Module, enabling inter-operability between the platforms. Thus all higher-level development and education can be performed in parallel, increasing the portability of both learning tools and courses.

The Modular Training System was designed to run on heterogeneous multimedia platforms (Sun Sparc, Silicon Graphics Indigo, IBM-compatible PCs). To convey complex facts, the learner's understanding of the course material must be reinforced as much as possible and embedded multimedia is seen as the appropriate enabling technology to allow this reinforcement [2]. Therefore, device independence of the learning material data, interoperability between the different platforms, and portability of course controls were essential.

The Modular Training System was partially developed and evaluated at the House of Computer Graphics in Darmstadt, Germany within different projects on a European-wide scale. The results showed that the MTS allows sophisticated learning and teaching scenarios on distributed computer systems [10]. The MTS user front-end (the GLS) is based on the presentation of embedded

multimedia objects rather than using external viewers. This allows authors or designers to set up the layout of a course and to present formatted multimedia materials in a way that goes beyond the capabilities of a current WWW front-end.

The MTS Components

In the following sections, all MTS components are introduced in more detail. This information will later be used to discuss the functionalities of the corresponding WWW components.

The Generic Learning Support

The learning front-end GLS (see Figure 2) is implemented platform dependent, supporting different graphical user interfaces such as OSF/Motif or MS-Windows. This implementation provides an optimized presentation performance for multimedia data and a "native look and feel." The GLS functionality covers both learning-dependent and learning-independent features such as formatted text (ASCII, RTF), raster images (TIFF, GIF, PCX, BMP, PPM), audio/video/animation objects (WAVE, AVI, MPEG, FLI), vector primitives (line, polyline, rectangle, arc, ellipse), menus, child windows, hot zones for hyperlinks, timers and a number of dialog objects (e.g., buttons, multiple choice tests, single- and multiple-line text inputs, sliders, selection boxes, different button boxes, browsing, and navigation tools).

All visible objects can be placed at any location within the main window or a child window of the course setup defined by its author. Clipping and scaling functionalities allow precise placement and size adjustment down to the scale of a pixel. By referencing RTF files, embedded formulas and tables may be handled. By using transparent text backgrounds, all ASCII and RTF strings may be placed on top of other visible object without completely hiding their graphical contents behind the strings. Since the GLS allows the exact placement of all visible objects it also supports individual manipulations and incremen-

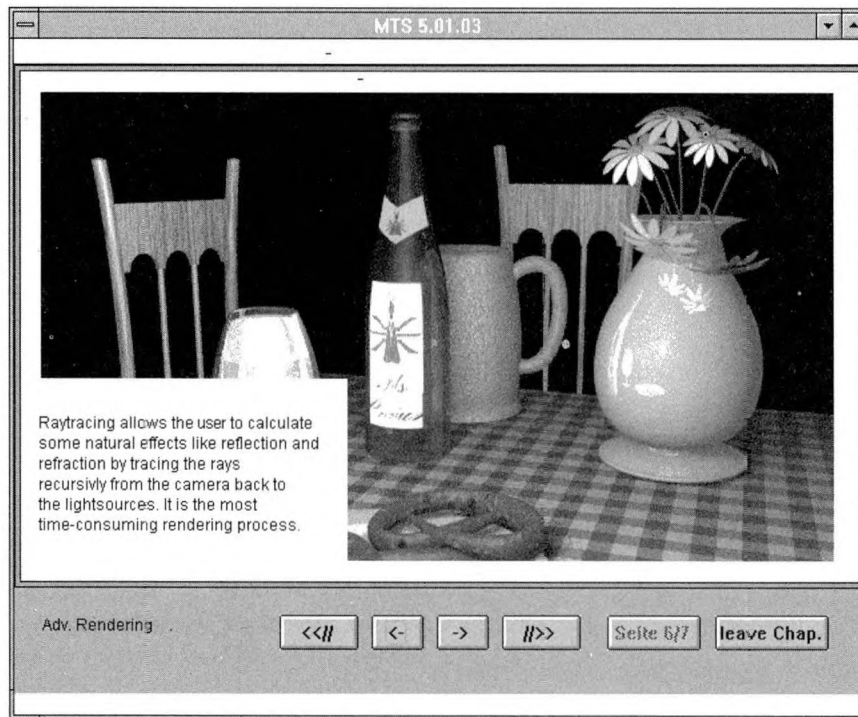


Figure 2: A screenshot during an MTS learning session using a book metaphor

tal redraws of these objects during their entire lifetime.

The Course Domain

The traditional meaning of the term *course* has been extended for the MTS course model. While in most traditional CBT courses, fixed sequences or at most conditional branching to predetermined sequences exist, the new MTS course model allows one to traverse through a domain of course material. The course material is dynamically selected with respect to the learner's profile, the equipment of the learning environment, and the learning progress. The course domain provides modular course material of three levels, covering curriculum control, interaction specification and basic multimedia learning material.

There are course modules (course nodes) being responsible for the goal-oriented specification of the course contents and for controlling the

course structure. Each course node handles the representation of the subject on the client side. For this purpose, different mechanisms are used: appropriate knowledge presentation, user interaction (e.g., by means of simulators), and profiling of the learner's progress. Therefore, a Course Node refers at least to the three corresponding function units. Additionally, a course node may be composed of course (sub-) nodes in order to represent the functional hierarchy of learning material.

Function units operate event driven and refer to Material Objects representing either basic learning contents (Learning Material Objects or LMOs) or user-interface items like buttons, menus, etc. (Intrinsic Material Objects). A course domain is composed of the sum of course modules, while a specific "course" is a goal-oriented and learner-dynamic view of the domain.

Links between course modules may be either direct or virtual. The latter are specifications of demands and allowances to the desired module (computational hyperlinks). Virtual referencing is a mechanism for the computation of target-material alternatives and for presentation preferences during run-time: an audio file and a video sequence is chosen if the technical equipment is sufficient. Otherwise, text is used instead of audio and an image sequence instead of video.

LMOs describe the characteristics of the module (contents and requirements) and reference to the file(s) containing the basic multimedia learning material. Standard formats for the latter are: for text, ASCII and RTF; for images, TIFF, GIF, PCX, BMP, and PPM; for audio/video/animation, WAVE, AVI, MPEG, and FLI.

Course nodes and Function Units are represented by scripts in the *Course Description Language* (CDL). Each script consists of three sections: the *Characteristic Section* holds information on the modules' capabilities (e.g., the subject it treats) and its requirements (e.g., learner's experience, GLS support, etc.); the *Declaration Section* includes either direct or virtual references. In the second case, a mapping process is responsible for the selection of modules with the capabilities/requirements matching best with the demands/allowances specified in the references; the *Body Section* describes the presentation sequence of the referenced modules and their behavior on incoming events (e.g., user interaction).

The *Course Interpreter* is a server process translating CDL scripts by resolving all virtual references and launching the presentation of the material objects on a GLS platform. Moreover, the interpreter is responsible for the handling of the GLS responses generated by user interactions, timers, errors, or system-specific events. In combination with the course domain, the Course Interpreter is the "brain" of the Modular Training System.

Within the course domain, archiving of and access to the multimedia course modules are pro-

vided by an underlying distributed database mechanism. Conceptually, this mechanism can either be realized by a standard network-file system or by using a distributed multimedia database system. The first solution is feasible for in-house training activities on a LAN; the second solution is rather adequate for learning scenarios in a WAN environment.

The Tele-Media Network Module

The Tele-Media Network Module, connecting client (GLS) and server (CI), provides the tele-control of multimedia objects over networks. It introduces a platform-independent and object-oriented network protocol addressed by an Application Programmer Interface (API). The API intends to hide the communication protocol between client and server. Therefore, it offers the application programmer a transparent way of sending and receiving the protocol messages by just using API function calls (see Figure 3). These API functions cover the complete spectrum of the MTS functionality as described above including the feedback from active interaction objects such as buttons, menus, dialogs, sliders, hot zones, etc. The advantage of separating the networking from the application and only allowing their communication over a common API is evident: Portability is made easy! If the Modular Training System is installed within environments using different kinds of networks, portability is only a matter of exchanging the Tele-Media Network Module.

The complete protocol-based communication between client and server is event oriented. This means that it is highly asynchronous and may create and process interrupts (events) at any time, also in parallel to a user. Therefore, the GLS is controlled by two instances simultaneously: the *user*, interacting with the user interface. This may create protocol messages that are transmitted to the server immediately and cause a reaction of the server; the *server*, sending protocol messages asynchronously to the GLS in order to control the course flow, can synchronize the presentation of

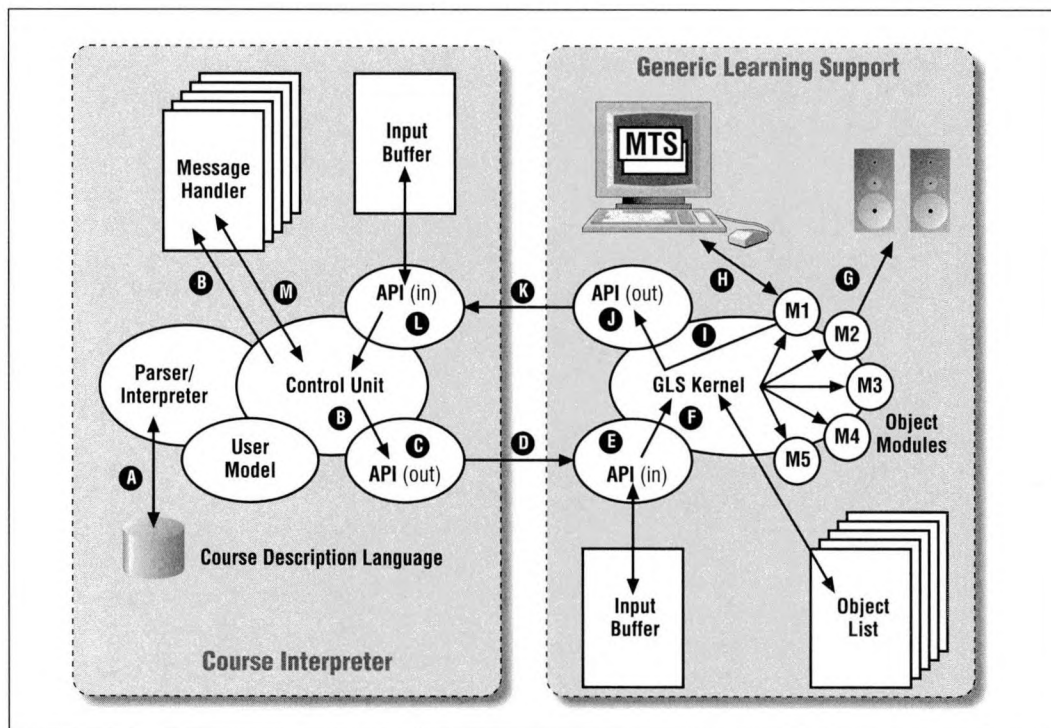


Figure 3: The message mechanism of the MTS: Interpretation of a course script (A); initialization of an object both on CI and GLS side (B to F); presentation of the object (G to H); handling of user interaction (H to L); decision of the system reaction on user input (M), which may start the message sequence again.

time-dependent multimedia objects and react on user input.

Human Communication Components

A particularly challenging feature of the Modular Training System was that of the integration of real-time video and audio, focusing on the communication aspects during learning sessions. This can be communication between two students facing the same problems during their course, or teacher-to-student(s) communication, where the teacher (or course author, tutor) has access to the Course Interpreter. The latter allows the teacher to modify the presentation logic at students' requests, or to adapt to a new situation.

Different multimedia platforms provide different types of audio devices and services. Nevertheless, all of them offer a common set of functionalities, such as record, store and playback. Online audio exchange for communication purposes was realized based upon these basic functionalities. In order to provide cross-platform usability, a common, intermediate exchange format was defined and a number of online bi-directional converters supporting different platform-dependent audio formats were implemented (see Figure 4).

Video communication for learning environments requires a fast codec algorithm that can be performed in real-time on either low-end hardware or by software. Because the real-time issues for video as a communication channel were more

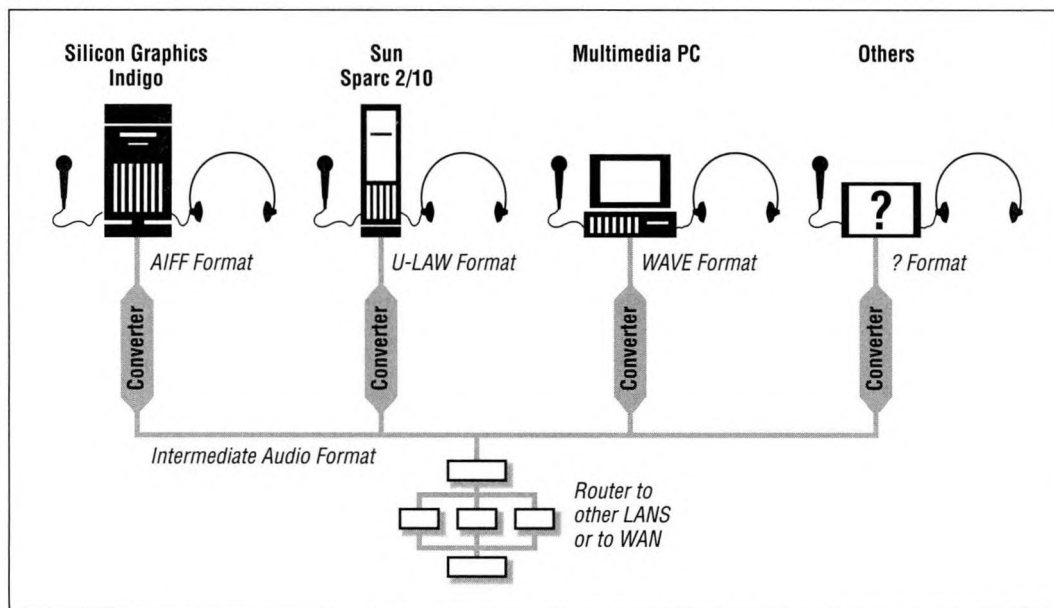


Figure 4: The audio communication mechanism using online converters for cross-platform usability

important than high-quality pixel and color resolution, we decided to use the JPEG (Joint Photographic Expert Group) codec standard for our video sequences, referred to as M-JPEG (Motion-JPEG) [5]. Within the MTS, a multipoint audio/video communication environment allows real-time discussions between groups of up to six teachers and students. MTS control elements are embedded in the GLS application.

Within the WWW, online communication is supported by using the M-Bone technology which is driven by the same idea of having people communicate in a more natural way. A big difference between M-Bone and the MTS Human Communication Component is that the first is used to broadcast real-time audio and video sequences to a big audience. In contrast, the latter connects small groups of teachers and students to allow vivid and more private discussions between the group members. Because this component of the MTS is not described here in more detail, we refer to [8] for further information.

From MTS to WWW: A Comparison

How can the described Modular Training System be compared to existing WWW technology tailored for educational uses? Most of the MTS components can be mapped directly to programs in the common WWW environment. However, some of the MTS functionalities show shortcomings of the WWW when MTS is used as a learning platform. In order to support the extended pedagogical or technical aspects and requirements described in the chapters above, the WWW needs additional sophisticated tools. In the following sections, we first compare MTS and WWW components and then discuss differences in the underlying concepts.

WWW Clients

The GLS directly maps to a WWW browser like NCSA Mosaic, which is also implemented on all major platforms and would be used as learning front-end in an educational set up. Mosaic is an

excellent WWW viewer and has been the driving force in the success of WWW. We have used it in the HyperScript approach without major drawbacks. Compared to GLS, one constraint regarding the formats accepted for inline images (GIF) and text formats (HTML, ASCII), along with an absence of inlined audio and video or animation. For these types of images, external viewers are launched. The disadvantages of this approach is the loss of uniformity in user interfaces and the fact that hypertext functionality is lost for these images. With GLS, one can define hotzones on graphics and other intrinsics. With HTML, you can do the same with clickable images. The disadvantage is that the URL to which the link points is not displayed in the browser's status bar as opposed to normal anchors within text areas.

The GLS intrinsics for menus, buttons, and text input correspond to menus, buttons and text area in interactive forms. Additionally, the GLS allows inlined images with different formats to be either clipped or scaled, while WWW clients can only clip inlined images or preformatted text areas. For formatted text with user-defined font type, size and color, the GLS supports clipping and panning operations which preserve the size of the selected fonts.

Additional visual elements like scrollbars, learning-specific dialogs, and browsing tools enable MTS course authors to produce advanced educational and highly interactive course frameworks. Moreover, a set of vector elements provides the graphical functionality for layout improvements that go far beyond the simple line primitives of the WWW browsers. Timers on both the GLS and CI sides allow exact timing and synchronization tasks for all multimedia objects. An example for an interactive course scenario using some of the features described above is that a learner can be asked to solve a problem by placing graphical objects within a given time, before a solution dependent on the learner's progress is presented automatically.

Most of the "intelligence" for repaint operations and the handling of object and element states is

embedded within the GLS. Therefore, the processing and supervision of these operations is kept away from the network and server as much as possible. This makes the MTS fast and reduces the semantical network overhead of the communication between client and server component.

Course Domain

There is no specific course domain in WWW. The only structuring mechanism for sets of documents is the facility to incorporate direct links to other documents, implementing menus of hierarchical components. The MTS allows for a didactic concept by providing three layered module types: course nodes for course flow control, Function Units for interaction specification, and basic learning material for knowledge presentation. The WWW only provides the basic learning material.

In addition, the MTS distinguishes between three pedagogical types of Function Units:

- *Presentation*: Conveying knowledge
- *Exploration*: Making use of simulation programs allowing explorative learning by experiments
- *Test units*: Integrating evaluation forms to compute the learner's success and thus influencing the sequence of the course

Using WWW technology, the second and third types are implemented through the CGI interface: *external simulation* programs can be launched on the server machine again loosing hypertext functionality. One possibility of integration with the presented course material can be achieved by NCSA Mosaic's CCI "remote control," when the simulation program is used to control the WWW viewer.

Test units are one important feature of online courses as suggested by HyperScript [7]: interactive forms, representing evaluation forms, are linked to course nodes. The learner can always take an evaluation to find out about his/her success or failure of reaching the unit's learning

objective. Upon submission, the forms start separate evaluation programs through CGI.

Within the MTS, test results are used to update the user profile, thus directly influencing the context and subject of the corresponding module. In WWW, user profiling, evaluation of technical requirements, and other characteristics must be implemented separately on top of the system. In the HyperScript project, user profiling uses the server's log data and results of evaluation forms and answers given directly by the learner, who can also readjust evaluation results. It is planned to incorporate specifications of module requirements, which ought to be met before loading a module. In a later phase, general indexing should be used so that modules can be computed using WAIS and other Internet searching techniques.

Furthermore, within the MTS the course structure, the interaction and the evaluation all are expressed in CDL. While the MTS offers an integrated, monolithic approach, the WWW is more flexible and offers a general programming interface (cgi scripts in Perl, Bourne Shell, C).

To implement computational hyperlinks (MTS technique for late module binding) one would create a set of tools at server side, which are activated through the CGI interface and compute the document to be presented on the fly. This technique is used in many educational projects [3]. One major difference between the compared systems is that the MTS has predefined attributes (module characteristics) which can be evaluated while mapping virtual references to a document to be presented.

Protocols

For educational purposes, the WWW's uni-directional and stateless HTTP protocol is one of the often-mentioned shortcomings. For this reason, extensions to WWW [4] start a separate process on the server machine to log the user's action and compute the result of links followed by the user. Within MTS, this approach is easily imple-

mented by conditional hyperlinks (virtual references) and the bi-directional protocol. Moreover, the protocol may transport object and element states from client to server and commands to change these states from server to client.

While the network protocol between back-end and front-end of the WWW is based on the exchange of files containing either HTML commands or multimedia data, the MTS uses a learning-specific, bi-directional, and asynchronous communication protocol between client and server component. Each platform and network-independent message defined in this protocol may be addressed by a dedicated API function. This structure implements an easy-to-use interface to the communication processes between the different MTS components.

The API functions allow access to all course material objects including their initialization, activation, deactivation, destruction, and state manipulation (like position, size, color, etc.) at any time. Moreover all user interactions are handled by specific API functions, which send them asynchronously to the server.

The Underlying Concepts

The first obvious difference between MTS and WWW is the concept of layout and formatting: While the WWW is based on a mark up language (HTML), leaving the final layout to be computed by the used WWW browser, the MTS presentation can be defined exactly by the course author. This includes positions, sizes, colors, adjustment, and scaling of objects through CDL statements. The advantage of the MTS approach is the possibility of incremental redrawing, for instance when only a few attributes of learning objects are changed due to user interaction or timer events. With WWW, the complete document would have to be transmitted and redrawn by the browser. Because of the absolute formatting, MTS offers two modes of presentation for inlined graphics, video, and text areas: graphics can be either clipped (as in WWW) or scaled, while text can be

clipped or panned (as opposed to WWWs line wrapping).

For educational purposes, incremental updating of presentation attributes of single objects can be used in many ways:

- activation or deactivation of hotzones and menu items or buttons, depending on the user's profile or learning context (notion of HyperScript views)
- change of background colors or bitmaps for buttons or anchors representing sub-modules, which have been visited and evaluated.

The latter allows an easy implementation of the HyperScript proposal of visualization of learning success which needs sophisticated tools on top of WWW (graphical presentation of the document structures with node coloring representing untouched, visited, successfully evaluated, and failed evaluations). The prize of the MTS approach of absolute formatting is the description effort every author has, when producing CBT material. In WWW, only few markup tags are needed, presentation being on the WWW clients responsibility.

The other major difference in concepts consists of a general higher-level education functionality of MTS. Pedagogical useful elements are predefined, which need to be constructed using HTML's basic features. For instance, there are units of radio buttons allowing 1-of-n, or m-of-n choices, or predefined dialog forms with answers (yes, no, undecided). Additional elements are sliders, which are best used for a learner's self-assessment, which is proposed in the HyperScript method.

Conclusion

The multimedia desktop workstation is the communication and learning center of the future. Towards the end of this century, information and training material must be accessible in many forms. Personal and co-operative multimedia on heterogeneous, networked PCs and workstations

provides interactive learning sessions among distributed users. We are now at the beginning of a new era of information management and distribution which could well change our society as dramatically as telephone technology only a few decades ago; the center of this new technology is not only the man-machine interface, but also the human-human interface.

The Modular Training System (MTS) gives valuable hints for the usage of the WWW for distance education. By implementing and evaluating the MTS during the last four years in Finland, France, Germany, Greece, and Portugal, an adequate environment for teaching and training on digital networks has been developed. This experience may be easily mapped onto the development and implementation of features for CBT on the WWW. Therefore, educational WWW projects should take advantage of the results achieved by the MTS. This can be a simple consideration on what can be adopted from the MTS when building tools on top of the WWW or may give some new directions for future extensions to the WWW.

From our point of view, the major implications for CBT on the WWW derived from the experiences with the MTS are the following:

- Add layout information, learning-specific elements, additional pedagogical mechanisms, and maintenance features for the provided learning material to the browser component (the front-end)
- Extend the protocol in a way that it can handle asynchronous and bi-directional messages containing object states
- Allow the server to react on user interactions in a flexible way by providing the adequate server components

Courses designed following the MTS concept are inherently adaptable to available environments, user groups, and pedagogic aspects. The modularity of the course material opens a wide range of flexibility and reusability in different courses

and on different platforms, and supports different tasks and responsibilities within the Modular Training environment. In particular, the integration of tests and the facility to control the course flow dynamically and in relation to the test results offer a new dimension of computer-based training. In the future, CBT on the WWW should also be able to take advantage of all these features by adopting the extended MTS mechanisms described in this paper. ■

References

1. M. Brunner, B. Kühnapfel, U. Schroeder, *New Media in Software Engineering Education*, Technical Report No. PI-R 3/95, Software Engineering Group, Department of Computer Science, Technical University of Darmstadt, 1995.
2. J.L. Encarnação, B. Tritsch, Ch. Hornung, *DEDICATED - Learning on Networked Multimedia Platforms*, Visualization in Scientific Computing: Uses in University Education, Elsevier Science Publisher B.V., 1993.
3. B. Ibrahim, *World-Wide Algorithm Animation*, Computer Networks and ISDN Systems, Vol 27, No. 2, Nov. 1994, pp. 255 - 265.
4. B. Ibrahim, S.D. Franklin, *Advanced educational uses of the World Wide Web*, Computer Networks and ISDN Systems, Vol 27, No. 6, April 1995, pp. 871 - 879.
5. Joint Photographic Expert Group ISO/IEC, *JPEG Technical Specification*, JTC1/SC2/WG8, CCITT SGVIII, 1989.
6. U. Schroeder, M. Brunner, *Utilizing WWW and Mosaic for Computer Science Education*, Proceedings of the First International Workshop on Training and Teaching on the Web, First WWW conference, CERN, May 1994.
7. U. Schroeder, *HyperScript - Innovative educational use of WWW*, Workshop proceedings at 3rd International WWW conference, Darmstadt, 1995.
8. B. Tritsch, A.S. Vieira, Ch. Hornung, *Video and Audio Communication over LAN*, Local Area Network Applications: Leveraging the LAN, Elsevier Science Publisher B.V., 1993, pp. 183-196.
9. B. Tritsch, *The Tele-Media Learning Architecture - A Distributed Multimedia Teaching Environment for Heterogeneous Computer Platforms in Digital Networks*, Dissertation in preparation.
10. B. Tritsch, A. Knierriem-Jasnoch, *A Modular Training System for Distributed Platforms in SMEs*,

Proceedings of the ED-Media 95 Conference, Graz, Austria, 1995.

About the Authors

Ulrik Schroeder

[<http://www.informatik.th-darmstadt.de/~uli/>]
University of Darmstadt

Ulrik Schroeder has worked as assistant professor for software engineering at the University of Darmstadt since 1995. He received his diploma in computer science in 1986 and a PhD degree in 1994, both at the University of Darmstadt, Germany. His major work areas and research interests include software engineering, object-oriented design methods, consistency control, and computer-based training and education.

Bernhard Tritsch

[<http://www.igd.fhg.de/~tritsch/>]

Fraunhofer Institute for Computer Graphics
Bernhard Tritsch received his diploma in physics at the University of Freiburg, Germany. Before joining the Fraunhofer Institute for Computer Graphics in Darmstadt in 1991, he worked at the Royal Nepal Academy for Science and Technology (RONAST) in Kathmandu, Nepal, the European Research Center for High-Energy Physics (CERN) in Geneva, Switzerland, and the Fraunhofer Institute for Solar Energy Systems in Freiburg, Germany. His major work areas and research interests include multimedia systems, cross-platform multimedia applications, and computer-supported communication.

Anette Knierriem-Jasnoch

[<http://www.igd.fhg.de/~anette/>]

Fraunhofer Institute for Computer Graphics
Anette Knierriem-Jasnoch received her diploma in mathematics at the University of Darmstadt, Germany in 1989. Before joining the Fraunhofer Institute for Computer Graphics in Darmstadt in 1994, she worked at the Interactive Graphics Systems Group in Darmstadt, Germany. Her major work areas and research interests include computer-based training and education, and open and distributed learning environments.