Computer Based Learning with Distributed Multimedia Systems

Max Mühlhäuser
Telematics Group, University of Kaiserslautern
Erwin-Schrödinger-Str., 6750 Kaiserslautern
Phone [+49] (631) 205-2992, Fax -2803, email max@informatik.uni-kl.de

Abstract
This contribution will discuss possible influences of advances in information technology on computer based learning and instruction. A joint project of Universities and Digital Equipment, called Nestor, will be described which takes advantage of such advanced information systems. Emphasis is put on the use of a combined object-oriented / hypermedia approach, on multimedia and cooperative-work extensions, and on process modeling based on hypermedia navigation.

1. MOTIVATION

The information technology of the nineties is marked by its merge with telecommunications on one hand and with consumer electronics technology on the other hand. This leads to a new technology of distributed multimedia systems, where people - usually at physically distributed locations - work together using cooperative multimedia workstations.

As a technical base, imagine for example workstations with a broadband ISDN connection, exchanging digitized compressed video and audio data, storing these data on recordable CD-ROMs, displaying them on HDTV monitors. As cooperative multimedia peripherals, imagine e.g. several computer-synchronized electronic blackboards at which physically separated teams cooperate with one another and with the computer (note that these are just example scenarios).

There will be a demand to use such cooperative multimedia workstations in an advanced computer based learning / instruction (CBI) scenario where individuals or teams of authors write multimedia learning material, individuals or teams of learners try to acquire knowledge - very often right at their working places -, supported by (possibly remote) tutors on-line.

While on the hardware side technology is evolving quite well and will enable such advanced CBI with distributed multimedia systems, on the software side very much of the enabling system technology is lacking. There is a high demand for generic, re-usable, and conceptually well-founded support in the areas

• multimedia,
• cooperation, and
• instructional design.

In all three areas, software engineering is a painful and tedious task today, hardly supported by formal languages, tools, or network / operating systems.

The most promising developments in software technology and CBI are object-oriented techniques and hypermedia; both domains, however, need to be exploited much further in order to enable advanced cooperative multimedia CBI.

This contribution will report on advanced software techniques and concepts suitable for the aforementioned context. The discussion will be based on work carried out in the Nestor project which is undertaken jointly by German Universities (Karlsruhe, Kaiserslautern, Freiburg) and the Digital Equipment Corporation CEC Karlsruhe research center.

2. PROJECT NESTOR

Nestor is a project on an Instructional Tool Environment which integrates tools (e.g., storyboard and video editor, hypermedia browser, ...), information (e.g., learning objectives, concept maps, course contents, ...), and processes (authoring processes and instructional strategies, cf. section 6) of the instructional design domain and supports the whole life-cycle of CBI material from early design until learning. A common understanding of all project members is reached by a comprehensive environment architecture which is constantly updated. Due to the size of the problem space, Nestor has to restrict itself to the most relevant elements of this architecture: these elements are investigated in one of the two Nestor tracks:

• in the research track, where more long term research oriented work is carried out, or
• in the development track, where comparatively straightforward - but to date unavailable - elements are implemented as prototypes.

The basic ‘ingredients’ which form the starting point of Nestor are:

• the hypermedia and the object-oriented paradigms,

• a scenario of cooperative multimedia workstations, and

• a set of tools, information, and processes as mentioned above, for which process modelling and control has to be supplied.

The hypermedia and object-oriented paradigms have many elements in common. The challenge for Nestor was therefore to combine these two elements into a single concept and to use it as a base for the whole environment. This leads to the following working areas in Nestor:

1. Object-Oriented / Hypermedia (OOHM) Base System,

2. OOHM-Extensions regarding:

   2 a. Multimedia support (better integration and ease of use),

   2 b. Generic Cooperation support, and

   2 c. Process modelling on the basis of OOHM navigation (see below).

We will use these four areas for structuring the remainder document.
3. OOHM BASE SYSTEM

Looking at recent proceedings and readers about hypermedia, one can realize an increasing discrepancy between the large number of hypertext / hypermedia systems available and the slow advance of a theoretical basis for this field. Standardization bodies have taken care of the field, but common understanding about hypertext comprises little more than the concept of nodes and links and no formal base for the description of hypertext systems or hypertexts.

As a consequence, there are - just as an example - systems which allow static links only, others allow dynamic (computed) links only; all known systems allow the user to work on concrete instances of hypertexts only ("a single node or link at a time"), supporting no reuse, no consistency checks, etc.

The Nestor OOHM base system derives great benefit from the combination of the two paradigms, e.g.:

- strong (object-oriented) typing of nodes and links (node and link ‘classes’) including multiple inheritance, polymorphism and hierarchical structuring, as one of the approaches to re-use, consistency checks, etc.;
- static and dynamic nodes (static nodes carrying e.g., ‘single media’ information, dynamic ones carrying e.g., animations), static and dynamic links (where ‘dynamic’ refers to the computation both of the link destination and of the actions to be carried out at link traversal);
- n-ary links, e.g., carrying out the synchronization of multiple ‘single media’ nodes connected through them.

The ‘dual approach’ by Dürr and Lang [1] brings further advantages:

- Since the interpretation of objects as nodes or links is carried out via method invocation and no dedicated object classes ‘node’ and ‘link’ are required, any existing object-oriented system can immediately be subjected to a default hypermedia interpretation (and version control) without change.
- Optionally, any object (class) can decide to carry out the hypermedia interpretation by itself (i.e., not to leave it to the hypermedia browser); this leads to a very high degree of flexibility and modularity.
- Persistent storage is supported either by mapping to a conventional (relational or object-oriented) DBMS or by means of a dedicated object-oriented concurrent access control mechanism which supports distributed cooperative access to persistent hypermedia data.

Section 6 will introduce the PreScript formalism as another generic OOHM concept.

4. MULTIMEDIA SUPPORT

While the first generation of multimedia workstations basically delivered media device control functionality ("the computer got a handle on the multimedia device, not on the data"), at present more and more people agree that broad use of multimedia systems requires the availability of multimedia data as ‘normal’ digital data (of type ‘video’, ‘audio’, etc.) in the com-
puter, subject to compression, processing and interpretation, transmission, etc. But such a new generation of digital integrated multimedia systems raises a number of new problems in software technology, out of which Nestor investigates with emphasis

in the research track:

- network-, location- and device transparent multimedia support, cf. [2],
- a synchronization model and concept for the integration of different media,

and in the development track:

- processing of time-dependent media (audio, video) on standard workstations (replay of digitized video on standard workstations, audio devices for busses according to the SCSI standard).

Especially for the first two items, the OOHM base concept has proven to be very helpful: the object-oriented approach helps in providing transparency, and the n-ary dynamic link approach is very useful for the synchronization model.

5. COOPERATION SUPPORT

The CSCW (computer supported cooperative work) paradigm has found increasing interest in the past years, but - again - a look at the literature shows a lack of general concepts: most contributions are specific to a certain application domain (joint editing, conferencing, group decision support, ...); the few existing more general approaches do not cover a broad enough range of possible cooperation patterns. The Group Interaction Environment (GroupIE), developed as part of Nestor, models and supports a broad range of generic CSCW 'building blocks'. The developer (in our case, the authoring tool developer or the CBI author, respectively) can choose from these building blocks for implementing a specific CSCW support structure (to be adhered to by teams of authors or by teams of learners, respectively; cf. [3]).

CSCW support is specifically promising in the hypertext/hypermedia field as there is a high demand for sophisticated 'multi-user' support: most hypertext systems, despite their attempt to support huge - i.e., shared - information spaces, are of rather 'single user' nature.

On the technical side, modern window-based workstations are much harder to use in a cooperative environment than traditional terminals since the complex window systems (e.g., X-Windows™) assume a one-to-one instantaneous connection between the application and the user; neither 'shared' i/o to/from an application by different users nor 'canned' output to a file - which could be replayed, annotated, distributed etc. - are foreseen. These topics are addressed in the 'sharedX' and 'Xrecorder' projects in the Nestor development track, respectively (cf. [4]).

6. PROCESS MODELLING - HYPERMEDIA NAVIGATION SUPPORT

A user's interaction with a hypermedia system (selecting and viewing nodes, choosing and following links to subsequent nodes, etc.) is depicted as 'navigation'. Assuming the Nestor
OOHM system as a basis, both 'instructional strategies' and 'authoring processes' can be regarded as 'navigation prescriptions':

- **Instructional strategies** are models and rules for the organization, sequencing, and display of CBI contents in relation to didactic concepts and to models of the learner (see [5] for some example strategies like 'component display theory', 'progressive differentiation', 'elaboration theory', etc.). In the hypermedia context these strategies determine or restrict the *path* of the learner through the hypermedia network. Depending on the chosen strategy, this path can be *unrestricted* ('exploratory learning'), *suggestively or decisively restricted* ('guided learning') or fully *decisive* (a negative example for this option is the 'presentation-type' CBI material used in the late sixties and early seventies, a positive example may be a 'game driven' CBI system which computes the path as part of the game strategy).

- **Authoring processes** are defined as rules (prerequisites, constraints) for the use of instructional tools and information (cf. section 1) related to the history of use, timelines, etc. Since the OOHM base system can be used to represent both the information and the tools (as dynamic nodes), authoring processes as well are manifested as navigation (of the author) through an OOHM network.

In particular, Nestor requires process modelling (for instructional strategies and authoring processes) via hypermedia navigation support to be

- **Human and machine readable**: today, instructional strategies are mainly stored in books and in expert's heads; authoring processes are defined by intuition or - in larger instructional design related organizations - described verbally. At most, specific instances of processes (application of a specific instructional strategy in a specific piece of CBI material or support for a specific authoring process in an instructional toolset, respectively) are represented in machine-readable form - but then, they are hardcoded and intertwined with other software components.

  The Nestor approach is to describe such processes as PreScripts, i.e. as navigation rules, in a highlevel (partly graphical) language which can be used by humans (in order to define, use, and communicate processes) and by the computer (for controlling / supporting these processes).

- **Re-Usable**: an instructional strategy, once formalized as a PreScript, should be allowed to be used for different content domains, and a content domain should be allowed to be accessed using different instructional strategies. Different authoring processes should be supported in the same instructional environment.

**The PreScript approach**: according to the above, Nestor PreScripts are navigation rules which are defined for a certain 'family of OOHM networks'. This 'family' is described via a 'generic OOHM network' (a constructive approach is used, similar to that of 'graph grammars' [6], supporting nonterminal and terminal symbols for nodes and links, recursions, optional and infinite repetitions, etc. Accordingly, a PreScript consists of two parts:

- a *construction* part which describes the 'generic OOHM network', and
- a *navigation* part which describes the navigation rules, related to the construction part.
A specific *mapping process* relates a PreScript to a specific instance of an OOHM network, i.e. determines which nonterminal and terminal symbols of the PreScript relate to which nodes and links in the OOHM network.

- The OOHM network may be constructed from scratch; in this case, the construction part of the PreScript may be used to drive a kind of 'syntax-directed (graphical) OOHM editor'; the mapping is thereby given implicitly.

- An existing OOHM network may be adjusted (by adding some nodes or links or by inheriting additional types) so that it accords to the construction rules defined in the PreScript. At present, the mapping has to occur by hand in this case.

- In a future version, the *construction* rules may partly be executed at runtime ('navigation time') only (along with the navigation rules), allowing the manipulation of the OOHM network by the navigating user; this feature is especially important in the context of authoring processes.

**7. CONCLUSION**

A CBI vision was sketched in the framework of future cooperative multimedia workstations; critical software support functions were identified, desirable for tool developers, authors, and learners in order to exploit this vision. The Nestor approaches to these requirements were briefly described. Solid formal frameworks for OOHM techniques, for cooperation and multimedia support, and for navigation support (i.e., process modelling and control) have been found to be the most critical ones in order to move CBI technology from its current inefficient 'instance based mode' towards re-use and ease-of-use and in order to help distributed multimedia systems find their way to a widespread use in CBI.

**8. REFERENCES**


