INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • Phone (510) 643-9153 • FAX (510) 643-7684

Collaboration Support in Networked Distance Learning

Bernd Krämer[1] Lutz Wegner[2]

TR-097-024

May 1997

Abstract: Learning is basically a social process. Experiences with *Computer Supported Learning* (CAL) over the last thirty years have shown that technology cannot substitute for some of the essential elements of this process, e.g. personal communication, face-to-face collaboration, positive and negative reinforcement through fellow students, etc. Today, with local and wide area networks becoming a reality, there seems to be a chance to simulate some elements of this learning process by a suitable combination of synchronous and asynchronous collaboration techniques. In particular, this paper proposes ways of supporting this interaction within a consistent, representation-independent complex object model. To map the model onto affordable technologies we borrow structures and methods from both database research and current multi-media course development. The arguments for the suitability of our approach, keeping in mind that distance learning remains a necessity in many circumstances, are supported with a number of examples.

---

[1] FernUniversität Hagen, 58084 Hagen, Germany, Bernd.Kraemer@fernuni-hagen.de

[2] Universität Gh Kassel, 34109 Kassel, Germany, wegner@db.informatik.uni-kassel.de

1 Introduction

Computer Aided Instruction (CAI) is the attempt to utilize the medium computer in a sensible way in education. Sensible in this context means that we are far from assuming that the computer (networked or not) becomes the sole holder and provider of knowledge and that it offers the primary (or most effective) interface for the knowledge acquisition process. As Engbing, Keil-Slawik, and Selke [6] correctly point out, this would be much like assuming one can lock a student for a year into solitary confinement in a room equipped with reference and textbook material and expect him to re-surface as a well-educated member of society.

However, computers in education, in particular networked computers, offer some unique forms of interaction, e.g. animation, hypertext navigation, on-line queries and searches of current data world wide. They permit making annotations and to quickly and cost-effectively download documents and programs for personal use. They permit certain groups of society, like handicapped, immobile or otherwise constrained persons, access to education which would otherwise be hard to obtain[1]. Therefore, looking into ways to improve distance education with the networked computer is not a luxury, but a necessity, and must be seen as a long-term activity.

As such computers in education have been a topic for now thirty years. Progress and encouragement in this field have not been along a straight path but rather has seen euphoric ups and depressed lows. These waves were in parts caused by technological pushes as shown in Figure 1 below.

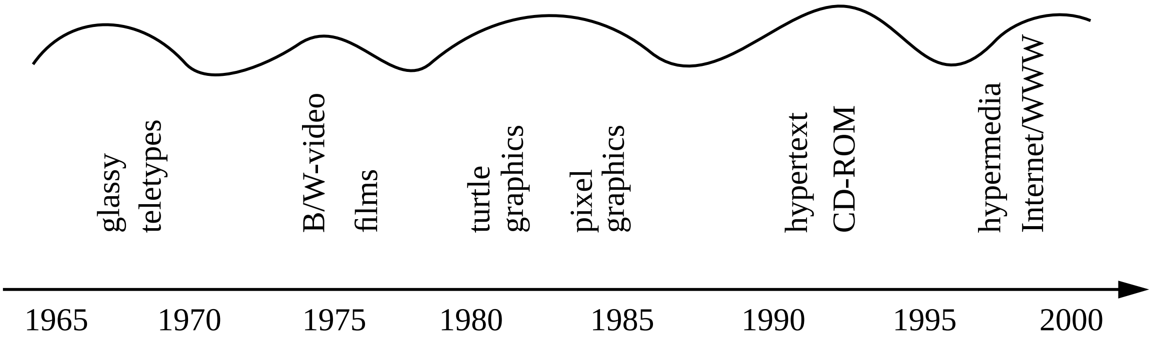


Figure 1: Base technology and time of widespread market introduction as far as of importance for modern education (samples only)

Figure 1 also includes some non-computer technologies like the use of video for lectures which

---

[1] It is highly debatable whether the increased use of networked computers in learning and research actually removes or whether it builds barriers between e.g. rich and poor sections of the population or rich and poor nations. Fact is, however, that nearly 70% of the students at FernUniversität Hagen work full or part-time between their study.

one of the authors of this paper experienced when being a student in Karlsuhe, Germany, in the late Sixties/early Seventies. It also shows the excitement that came with the educational packages for the first Mac and later the PC with modest CGA/EGA graphics. In particular the COSTOC project from the mid-Eighties must be mentioned here which was initiated by Hermann Maurer, best known these days for his Hyper-G and HyperWave developments [14].

Hypertext [16, 5] a term coined in 1965 by Ted Nelson, inventor of the Xanadu Project [15], brought another wave, when authored software was distributed as "canned" information, which was also usable in museums, exhibitions and other places accessible by the public. Hypertext together with the affordable form of distribution on CD-ROM is and will remain a component of educational software.

The fifth and last wave, on which we are presently surfing, is the World Wide Web (WWW), an extension of the hypertext idea but with on-line, world-wide interconnection within the Internet and a seamless integration of multiple types of media. Teaching units for download may be corrected up to the last minute, they can be augmented on the spot with other offered material, they permit queries, may contain animation via Java applets, include feedback from students via forms, depict simulation results, and feature use of remote execution of interactive tools.

At the same time we see experiments with broadband transmissions (e.g. with the MBone connections developed here in Berkeley) between universities and their use in teleteaching. These scenarios include transmitted video of lectures and students asking questions, a shared whiteboard and class notes distributed electronically. Indeed, authors like Gemmell and Bell [7] even argue that telepresentations are a matured technology.

However, they acknowledge that there is a large variety in delivery schemes. On the one extreme end we have large nation-wide distribution of units with strong TV and radio broadcast elements with basically no concurrent back channel as well established in the Open Learning Australia (OLA) post-secondary distance education institutions [17]. In the middle are schemes where an audience shares a room but lecturers are only telepresent. These are now quite common in industrial applications, e.g. technical training and sales initiatives.

Not yet very common and, according to [7], "currently difficult is giving presentations to a large number of desktops (and including back-channels makes it even more difficult). "Clearly, for these teleteaching developments, the interactively and interconnections of the network are the central factors. Whether this wave will be followed by yet another backlash, caused e.g. by poorly maintained servers leading to many missing links, poor transmission speeds, cooler assessments of the involved costs, the appearance of trash and pornography on the net, etc., is an open question. Already the next wave, namely virtual reality environments, are lurking behind the corner where teachers take us by the hand and guide us through the libraries of the world, allowing educational travel through time and space, permitting access to unlimited knowledge just as envisioned by Vannvar Bush's Memex [4] in 1945.

2 The Need for Collaboration

One of the persistent misconceptions about CAI is that it can replace traditional in-class teaching. The source of this believe is sometimes marketing hype by those wanting to sell teachware or promoting their projects. Quite often it is fostered by the hope of governments which want to cut down on schools and teacher hiring.

Evidence for this perception can be found even today, as e.g. Hardaway and Will [11] promote *portable educational material* (courses as multimedia packages on CD-ROM) as a way to "reduce actual classroom sessions to optimize overall time management [11, p. 95]."

To be fair, these suggestions are often made out of a false sense of helpfulness, e.g. when our CAI-Lab is overcrowded and we offer students in Kassel to take home a copy of the course and to study it in front of their own PC at their own pace. Surprisingly enough, students insist on sitting in pairs in front of our aged PC's in an overheated, darkened room where exactly the same material will be presented over the next 5 to 7 days (see Figure 2).

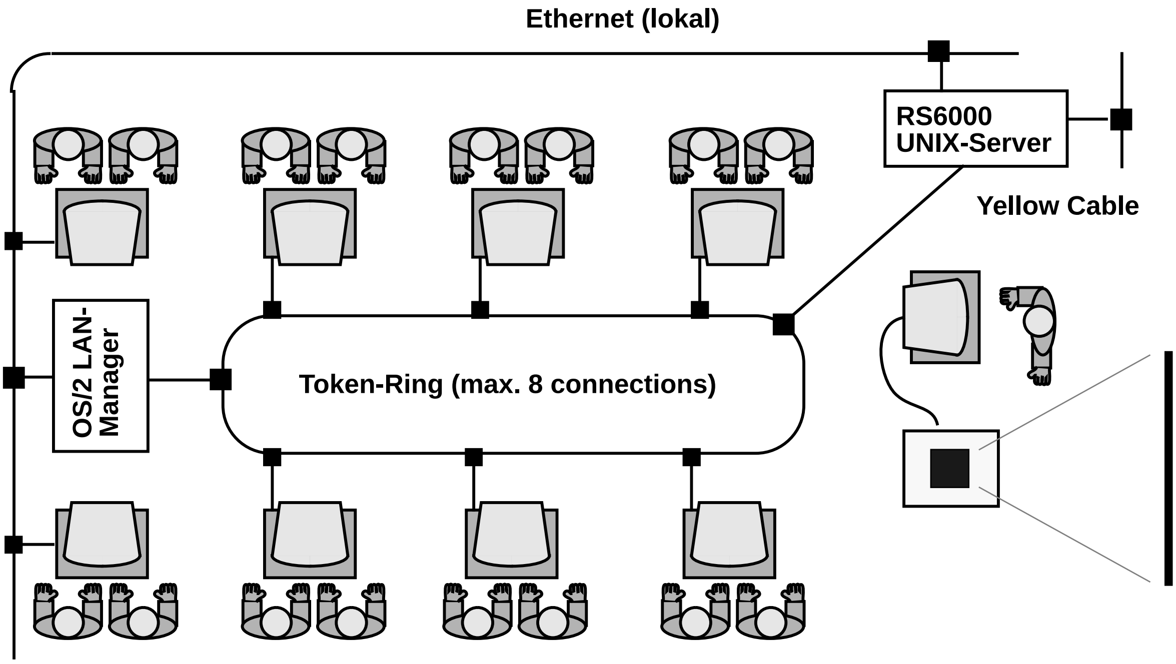


Figure 2: Principle design of CAI-Lab at Kassel University

In the particular context of this *Introduction-to-UNIX*-course, students realize that they want

- use of the UNIX system for trying out their material
- participate in on- going discussions about what they are just learning
- want to ask the instructor if something is not understood

- be able to learn about corrections and extensions of the course material which are made spontaneously
- get a feed-back (positive and negative enforcement) of their progress and whether they have understood the material[1]

All of these reactions have been reported from other sites of CAI experiments as well and stress the need for collaboration. To be more precise, we see a need for synchronous and even spontaneous collaboration which means being able to react and interact with the instructor, teaching assistants going around between the terminals, and other students. There is also a need for asynchronous collaboration, like worksheets and assignments which are handed out and which all students should work on alone or in small groups of two or three students and which are discussed at a different time later on using a whiteboard for displaying results. Last, but not least, there is the final written exam which everybody has to write at the same time under the same conditions, which turns out to be an essential element of the motivation process.

All in all this empirical evidence concurs with the trend towards computer mediated collaborative learning [1, 18] and the establishment of learning theatres. Simulating the atmosphere and possibilities of a learning theatre in a distance learning environment will therefore be one of the challenges in our development below.

Computer mediated collaborative learning follows the same time, same place (STSP) format of classical classroom teaching [11]. It is different from lab scenarios where multimedia (or in the simpler case, video tapes of actual lectures) are presented in a different time, different place (DTDP) format. For this form, Hardaway and Will [11, p. 94] envision that “students would visit the lab and replay the lecture as many times as they like during available operating hours.”

Again, empirical evidence from the aforementioned video lectures in Karlsruhe in the late Sixties clearly indicate that students don’t want this choice. One conclusion would then be that what works well for the entertainment industry, e.g. the multi-billion video rental market and multimedia games, does not necessarily carry over to the educational field.

Another aspect not mentioned yet is the authorship of course material. The small experiences with the COSTOC-material in Kassel and the much more extensive empirical data from Hagen suggest that best results are achieved when instructors use self-authored courses. Of course, this is not necessarily a computer-related phenomenon since many teachers will start out with statements “I don’t like A’s text completely and B’s doesn’t suit me either, so I combined the best of both with my own stuff thrown in where appropriate.[2]” Also, many instructors will

---

[1] Although there are over 100 multiple choice questions with answers built into the material, this cannot replace the feed-back a student gets from seeing (or hearing) others struggle with the same tricky question, respectively **not** struggle with questions.

[2] The same teachers end up their classes with a statement like this: “maybe it wasn’t that wise to mix all these texts; at least I could have saved myself a lot of time by just sticking to the script of either A or B!”

introduce so-called corrections to course material (paper- or computer-based) even though many turn out to be falsifications, at least from what we have seen with corrections done by others to our own teachware.

Therefore, extensibility and revisions of course material on short notice should be another feature in networked distance learning which must include some type of version control and also touches upon copyright and authorization issues.

3 Distance Learning in the Electronic Age

Up to recently, distance teaching in Europe was mainly based on specialized text books designed to fit the needs of time and location independent learning on the basis of self-instructional material. A one-semester course typically consists of six to eight course units that are sent to the students via snail bi-weekly together with homework assignments for the topics introduced in the previous course unit. Each unit typically includes

- a short introduction,
- major study objectives,
- the very course contents with interspersed self-assessment problems, definitions, examples and graphical illustrations,
- a list of related literature with annotations,
- solutions to the self-assessment problems contained in the main body of the unit, and
- an index of technical terms.

A glossary is usually provided with the first course unit. At FernUniversität the different elements of a unit are color-coded and the units follow an agreed presentation style. Audio cassettes, videos and regular tv broadcasts provide additional information to this core teaching material.

Within a period of two weeks, students mail their solutions to homework assignments to the responsible teacher where they are corrected and are sent back to the student together with the master solution. It is obvious that this time-consuming feedback process suffers from spontaneity and personal interaction. Other communication channels such as telephone, facsimile or electronic mail may overcome some of these deficiencies but are far from satisfactory.

To enable person-to-person tutoring, FernUniversität is supported by more than 70 study centers spread throughout Germany and a few neighboring countries including Austria, Hungary and Switzerland. Compared with on-campus tutoring, one of the drawbacks of study centers is the relative independence of course authors and tutors and their scarce interaction.

In addition, the geographical distribution of our students aggravates team work, peer-learning, experimentation with technical equipment and various forms of social interaction typical for a campus situation.

Distance education traditionally suffered from a lack of communication facilities which could bridge the physical separation of students as well as the distance between teachers and students. Social networking among peer students, team work in distributed project groups, peer-learning, and personal interaction among teachers and students were constrained to conversation over the telephone, information exchange by snail and electronic mail, and rare personal contacts.

The Internet enhanced with suitable cooperation technology allows us to diminish these deficiencies through techniques for sharing design objects, technical instruments and equipment, for developing solutions cooperatively or clarifying difficult passages of course materials over a distance.

Collaboration in distance learning shares many features with *groupware*, also called *computer supported collaborative work* (CSCW), in particular collaborative editing as described e.g. by Baecker et al. [2] or by Kaiser and Whitehead in connection with efforts like Distributed Authoring and Versioning on the World Wide Web (WEBDAV) [12]. However, collaborative editing as against collaborative distance learning does not have the strong, codified feed-back from the readership which is essential in teaching and occurs in forms of handing out worksheets, submitting assignments, grading of assignments, working out solutions to exercises, designing and grading of exams on the side of the teacher.

Another major difference is that the topic of collaboration, i.e. the course in the case of the distance learning application, is a highly structured, finished multi-media document which can be distributed beforehand, e.g. by CD-ROM or over slow lines during nighttime. Yet another difference is the variety of material which is joined into the learning material:

- the “official” course material
- shared (public) corrections and annotations
- pending corrections
- private annotations
- selectively made public annotations (e.g. notes made available to teaching assistants)
- assignment worksheets
- handed-in assignments (transfer of ownership from student to teaching assistant).

Our main goal for collaboration in distance learning is then to support communication through gesturing at a shared drawing, image, piece of text, animation or simulation. Proposed changes should be made easily visible, evaluated and further developed in give-and-take discussions over technical media. This will make it necessary for course constructors and participants to identify knowledge references and establish on the fly new links into fine-grain knowledge components such as part of an image, a frame or a scene in a video, a snapshot of a simulation, a particular formula in the course, etc.

Apart from these more synchronous features, there should be an audit trail and a way to turn gestures into persistent markers and annotations which allow one to trace a development of a topic and permit participants joining the collaboration later to catch up on the discussion.

Whiteboards and chat tools are of lesser interest here because they assume a poorly structured material as typically encountered in initial brain-storming sessions. As pointed out before, we assume a well-structured course material to start with. Also, whiteboards and chat tools suffer from insufficient concurrence control, where one person might, for example, start to sketch an architectural design of a software solution, while, all of a sudden, someone else changes the context by scrolling the page on zooming into some detail of another object presented.

Video conferencing is also not a suitable technology for the context of distance education although we recognize that video conferencing helps socializing and coordinating audio contributions which would help get teacher-student and student-student collaboration going. However, given the poor bandwidth for students working from their homes, the presentation of a still image of the acting person or a strongly compressed talking head transmission [7] may serve as a substitute for the time being.

4 Sharing a Focus

For the following discussion we assume that students, teaching assistants (TAs) and instructor are connected by (low bandwidth) communication lines with PC's and/or workstations as communication devices. We also assume that the network is wide area (with the possible exception of instructor-TA connections) and includes audio communication, but not video.

A possible scenario would involve two students A and B, a TA T, and instructor I in an argument over some details within a course unit, say on Software Engineering (SE) within the Electrical Engineering (EE) faculty of Hagen, from where the following example is taken.

Example 1 Production Cell

Consider an abstracted production cell with feed and delivery belts, a rotary table, a robot with two arms, and a press. The example is taken from [13]. Figure 3 below shows a schematic view of the cell and Figure 4 below a VRLM animation of the same cell.

The argument could be about the safety requirements which say that the press may not operate while arm 2 of the robot is pointing towards the press and is also extended (into the press).

Existing collaboration schemes as developed for temporal media in [9] and discussed in general in [3], suggest posting a transparent white board (actually a glass board) over the objects in discussion and to permit concurrent, shared annotations with pens on this surface. However, this touches only the representation of the object and does not take into account their structure and semantics.

We start out by noting that each of the course elements (figures, graphs, chapters, formulas, animations) and in fact the course itself has a structure, most of it induced by a whole-parts

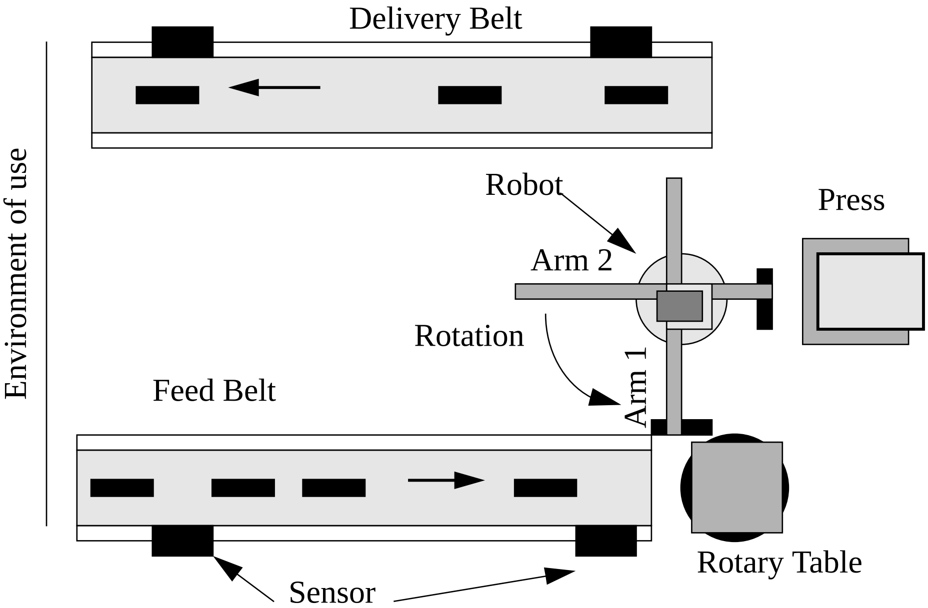


Figure 3: Bird’s eye view of production cell (from [13])

relation forming a tree (see Figure 5). In general, since there will be links to other structured objects and since there can be object sharing, the course material effectively is a (directed) graph.

Next we want to be able to position a focus on any complex or atomic (character, number, pixel) object and to share this focus with other users which are concurrently looking at the

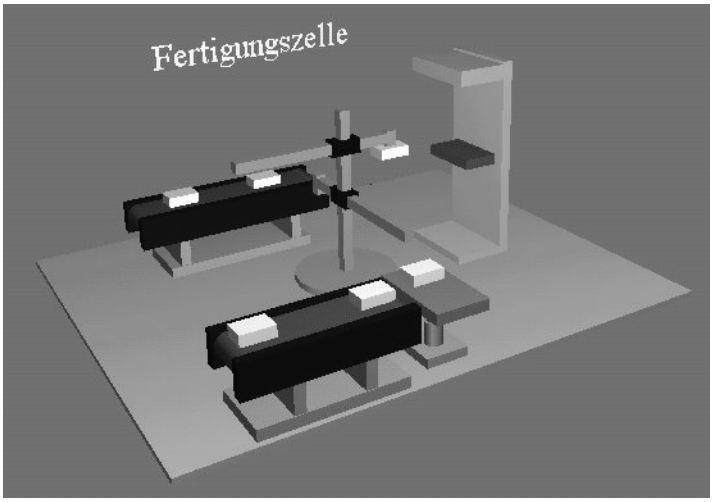


Figure 4: VRML view of production cell

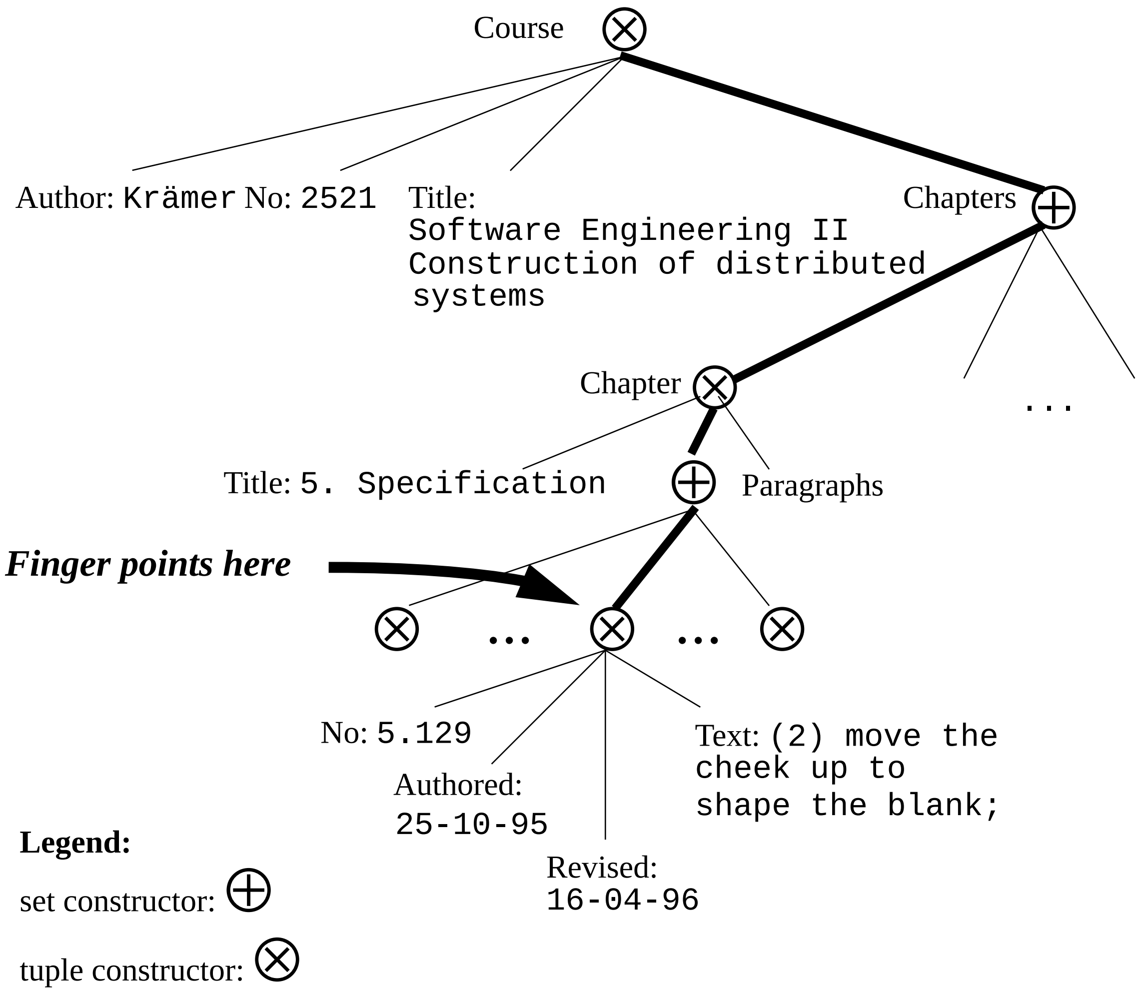


Figure 5: Instances of complex objects forming a course—here shown as a tree in IFO-like notation

same context. Other suitable terms for focus would be cursor or telepointer. Since all three terms are usually associated with windowing systems, we will introduce a new term, namely *finger*, with the convention that at any time a finger points to a single object.

Within graph notation, a finger corresponds to a node in the graph. On the implementation level, a finger corresponds to the (invariant) address of the node, in OODB-speak to the object identifier (OID) of the object. In the strict hierarchical model, it can also be seen as the path from the root of the object tree to the sub-object to which a finger currently points. In Figure 5 we have depicted this path with more solid lines.

When made persistent, fingers act as links or bookmarks. There can be several fingers concurrently within one object and we will later distinguish between private and public fingers. However, each user may have one active finger at each time. Through this active finger objects can be manipulated and there is a small set of navigational operations to move the finger within the tree. Input from the keyboard and/or the mouse is directed towards this active finger[1].

Since we aim at a certain degree of freedom from representation and at uniformity of navigation and manipulation, fingers have no predefined graphical representation. Rather, the convention is that they must be recognizable within the context by some form of highlighting. Since we also want to distinguish fingers belonging to different users, color will be of great importance.

Fingers were first introduced in [19, 20, 21, 22] for the visual database editor ESCHER which is based on a nested relational model. There, finger movements and visualization are closely tied to nested tables but examples below show that the paradigm fits any tree-structured data and can be adopted to multiple visualizations as well.

Example 2 Course Objects

Consider again the production cell with a finger on one paragraph from a course describing the press operation. Assume that each paragraph in the course has an id, a creation and a revision date. Three representations are given:

- a tree representation
- a tabular visualization
- a pure textual visualization.

In the Figures 5-7 we assume that a finger is sitting on the tuple-object which makes up paragraph 5.129 within chapter 5, etc. In Figure 5, this is indicated by the more prominent edges within the tree. Note that a finger then becomes a stack of addresses (of nodes) which corresponds to the technical view in ESCHER.

Figure 6 shows the same instance as a nested table with a suitable schema displayed above the table. Note that the finger is now a shaded area and that a second finger sits within the schema on the corresponding attribute. We also assume that in the tabular visualization, longer texts are shown as icons (here shown as ❒). Finally, Figure 7 might be a more text-oriented representation with the tuple shown in a different arrangement and with the finger location indicated through inverse video.

Fingers may be moved with the following four basic operations which have several synonymous names:

- In, Enter, Push
- Out, Escape, Pop
- Next, Successor, →, ↑

---

[1] This should also explain why we prefer the term finger over the term cursor: in a windowing system we will also have a cursor which can be moved by e.g. the mouse and this cursor can be used to redirect and manipulate the fingers which point to objects.

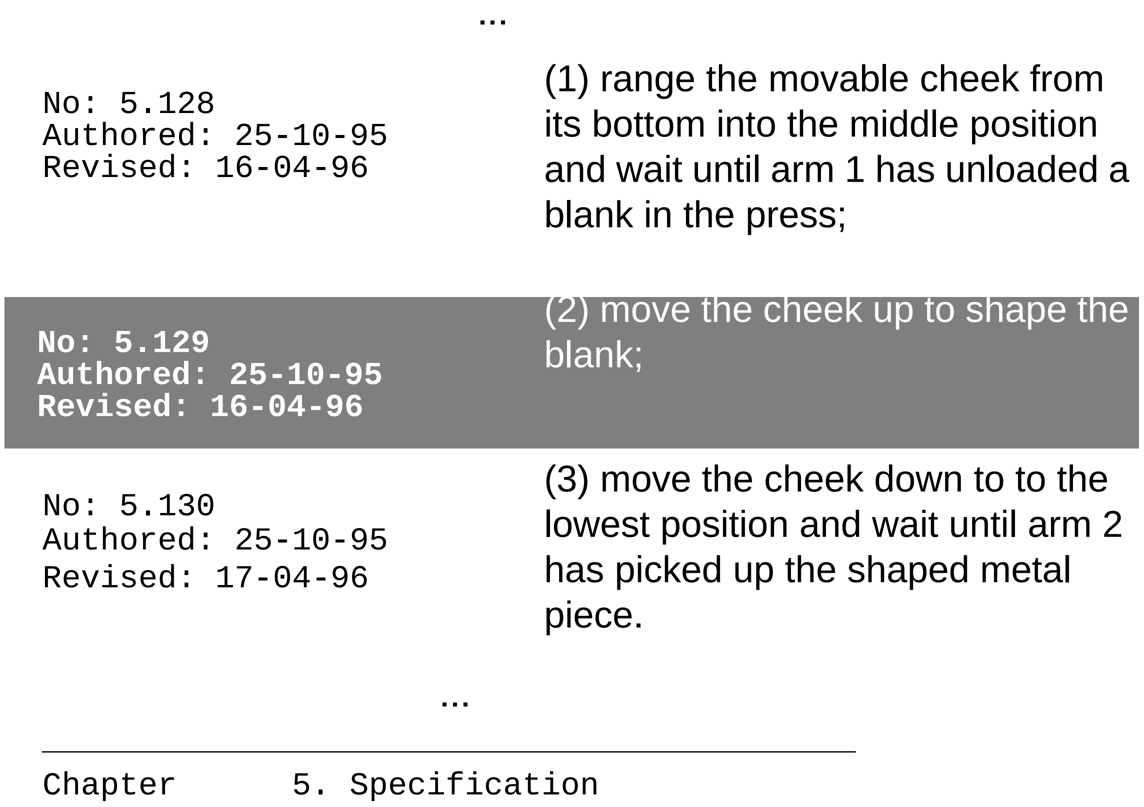
{ } Chapters				
Title	{ } Paragraphs			
	No	Authored	Revised	Text
0. Course Requirements	0.001 ... 0.029	02-05-95 ... 10-05-95	23-02-96 ... -/-	❐ ... ❐
1. Introduction	1.001 ... 1.130	11-05-95 ... 26-02-95	-/- ... 18-04-97	❐ ... ❐
...				
4. Production Cell Description				
5. Specification	... 5.128 5.129 5.130 ...	... 25-10-95 25-10-95 25-10-95 ...	... 16-04-96 16-04-96 17-04-96 ...	... ❐ ❑ ❐ ...
6. Synchronization				
...				

Figure 6: Nested table view with actual texts projected out

- Back, Predecessor, ←, ↓

Going into a complex object means moving a finger to its first constituent sub-object; in a tuple to the first attribute, in a set or list to the first element, in a paragraph to the first word, in a production cell to the first sub-component (say the feeder belt by agreement). Because it corresponds to putting an address on top of the stack which represents the path to the object, it is also known as the push operation.

Equivalently, pop means moving out (escaping) to the enclosing object. Next and back move from one sibling to the next, provided one hasn't reached the end, respectively start, of a collection yet. Readers might be surprised that both "arrowup" and "arrowright" denote the same



...

No: 5.128
Authored: 25-10-95
Revised: 16-04-96

(1) range the movable cheek from its bottom into the middle position and wait until arm 1 has unloaded a blank in the press;

No: 5.129
Authored: 25-10-95
Revised: 16-04-96

(2) move the cheek up to shape the blank;

No: 5.130
Authored: 25-10-95
Revised: 17-04-96

(3) move the cheek down to to the lowest position and wait until arm 2 has picked up the shaped metal piece.

...

Chapter 5. Specification

Figure 7: Textual representation with finger on a paragraph

operation: remember that we talk about logical, tree-like representations of objects which are independent from any visual (physical) representations. Thus, in Figure 6 “next” implies moving the finger to the next paragraph which is the line below. Would the finger have rested on, say, the “authored” date, next would be the “revised” date which is to the right in Figure 6 but down in Figure 7. Thus both keys on the keyboard map to the same logical operation.

Alternatively, fingers may be placed with cursor movements and clicks with the mouse. A mouse click always positions the finger on the atomic element whose representation (rectangular bounding box) currently encompasses the cursor coordinates. A mouse drag with button 1 depressed makes the finger follow the drag (in real time, at least in our table representation) to the smallest complex object which has starting and current end position within its bounding box. When the mouse button is released, the finger rests on this area. Clearly, this requires feed-back from the display (mouse coordinates) to the internal object representation and might not be possible in all circumstances.

5 Navigational Course Material

This brings us to the question of what is required of objects and their visualization. Clearly, course elements are created and maintained with a variety of tools and are stored in an abundance of formats. In many cases there was a grouping of elements present at object creation time, but the stored result does not contain the grouping and ordering anymore. Figure 8 below is such an example. It shows a Petri net for the press-robot synchronization. The net was created with a Petri net tool that maintains the logical structure of the net consisting of places (represented as circles), transitions (represented as boxes), a flow relationship (represented by arcs), a marking of places with tokens, and a labeling of places and transitions. In the course material, the net ends up as a pixmap object suitable for printing or for screen presentation with standard browsers. To make it accessible for collaboration, e.g. for a joint exploration of the net's dynamic behavior using token movements, it would need to be "opened" again.

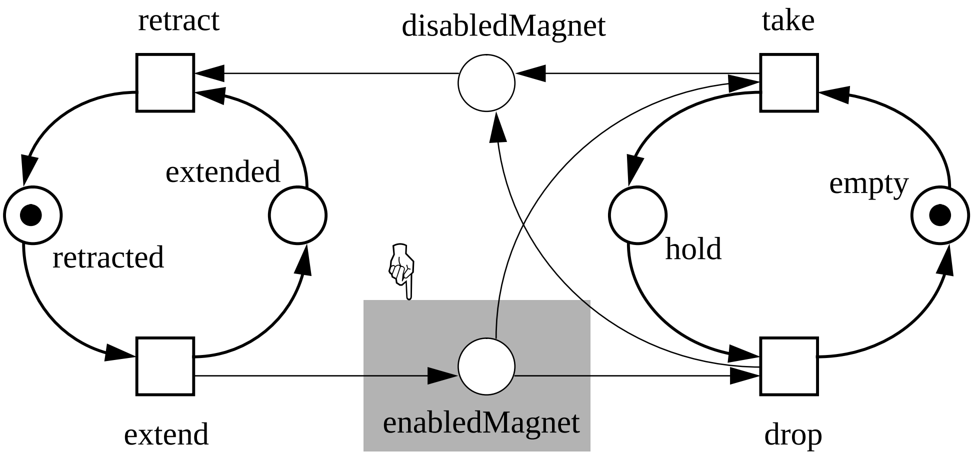


Figure 8: Petri net graph representing magnet-expander synchronization

We thus must aim at minimal requirements and target for "penetrable" as opposed to sealed documents:

- objects know their structure and can provide identities (addresses) of their nodes
- objects don't know their structure but a shadow object can be created and stored which recreates this structure
- objects don't have a stored structure and don't warrant creating a stored structure; however, there is a filter which understands the structure and can create it, or relevant portions thereof, on the fly.

Hypertext systems, CAD-objects stored in object-oriented or object-relational databases would satisfy the first criterion. Addresses would be URLs, values of attributes which have key property, Record Identifiers (RIDs) or Object Identifiers (OIDs).

Some graph formats including some PostScript tools have the ability to recognize substructures like polygons and polylines, frames, etc. A Structure can be extracted and stored as a graph in a shadow document with links into the actual graph. Clickable maps might also fit into this area. The shadow documents give object identities to the collaboration software that wants to position fingers on certain objects but maintains an internal mapping to the recognized subobjects.

The third case would be little structured data, e.g. pixel pictures or text streams. However, these data may have a well-defined structure in some context, e.g. words within a document with white spaces as separators. Although it would be too costly to store an id with each such miniature object, a filter can easily identify each of them and positioning fingers on any of them is not a problem.

The preceding points concerned object identity and positioning of fingers onto those objects. Having a finger on an object does not necessarily imply that you have a high-lightening in its representation. Here it is necessary to intervene the display process and to force it to generate markings which produce the desired effect, e.g.

- invert text
- change font or style
- change scale (point width) of objects
- change from solid to dotted lines
- change color (foreground background)
- put handles on corners of boxes
- draw bounding box.

In case of some hypertext systems it might even be feasible to insert markers temporarily into the actual source text which cause a display change. Most often, low level X-functions will permit high-lighting the correct passages. In the case of the VRML object shown in Figure 4, we would add a generic highlighting function to the original VRLM code to support finger presentations. If everything fails, using a tinted or otherwise marked glass board over the displayed object could be used although this would violate our display independence principle.

6 Concurrency and Collaboration

Within the framework sketched above we can now achieve *event broadcasting* as against *display broadcasting*. In [3] these terms are coined in connection with collaboration transparency achievable in JAVA although it is not clear to us why it wouldn't support collaboration awareness as well.

The main point, in any case, is the fact that finger movements translate into very small messages which can be easily synchronized and transmitted to all clients, here students and teaching assistants engaged in a discussion. Fingers then operate on data previously distributed, say during the night over low bandwidth lines, or off-line via CD-ROM. Together with finger movements updates, patches, annotations and work-sheets are distributed which alter the tree in a limited way and which require little bandwidth. Gemmell and Bell [7] point this out in their request for efficient support for animation and special effects in slide shows: “It doesn’t take many bits to define a polygon and say ‘fly from (x1, y1) to (x2, y2).’ It can take a lot of bits to retransmit the polygon for every point along its path.”

New objects can be introduced by any collaborator by placing a finger on the object in a clipboard. Usual cut-and-paste operations will transfer this object into the object tree at a location marked by a destination finger. Given type compatibility, the object is transferred and replaces the destination object.

This also opens questions of transmitting and storing metadata, e.g. type information. This is also introduced in [12] as a topic to consider in collaborative authoring and versioning and for which no generally agreed protocol exists today. In [10] it appears as course templates which are document grammars that define the structure of courses. These templates are needed to achieve homogeneous course components which opens the possibility of exchanging these building blocks in an electronic educational market. Course components are in turn built from such varied primitive course elements as text, images, video, math models, audio, on-line help, programs, 3D models.

Having metadata such as templates or types rises interesting questions in connection with inconsistencies with the object tree when not all collaborators have the same version, say because patches haven’t reached every client.

Finally, there is the need to distinguish between private and public fingers. This concerns ownership and who has the right to control a public finger. For public fingers there must be some form of locking and/or token passing implemented in some form of *floor control*.

Ownership/Control

		under own control	lockable public control
Visibility	own screen only	privat non-shared	non applicable
	all screens	privat shared	public shared

Figure 9: Ownership versus visibility in collaboration

Ownership must be separated from visibility which could be shared (a finger is visible to all collaborators) or non-shared. Figure 9 shows a possible taxonomy where public non-visible fingers are an unlikely choice unless one wants to consider secret (e.g. administrative) cursors in documents.

Similarly, floor control must be augmented by *session control*, which allows e.g. joining and leaving a collaborative seminar. In our proposal, joining a session corresponds to *forking off* a finger from an existing one (e.g. the instructor’s public cursor) which puts the new participant right into the focus of the current session.

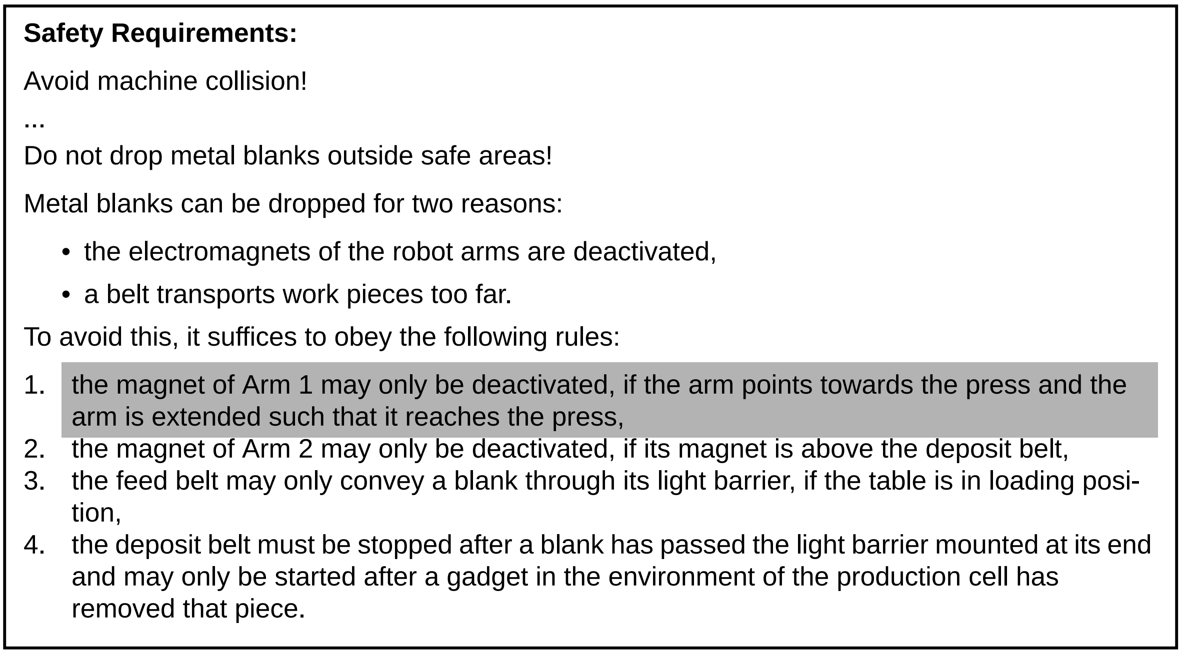
Other aspects, like lock discovery mechanisms, check-in/check-out mechanisms, name space management, version control, queuing of requests, roles, sub-groups, private discussions, speed-up, slow-down, repeat, moderator selection, back and side channels for communication, have been mentioned e.g. in [7, 8] but are skipped here for lack of space.

We close this Section with a scenario for the example above. The scenario uses the techniques just introduced.

Example 3 Collaboration Scenario

Mary, John and Barbara have teamed up to form a learning for the course „Distributed Software Engineering“. Although they are living tens and hundreds of miles apart from each other, they decided to form a distributed project team to solve homework assignments jointly and help each other in understanding the course content if difficulties arise.

The problem our three students are currently concerned with is the development of a design solution for the safe control of the behavior of a robot used as a transport component of a production cell. The behavioral design of the robot controller is supposed to be delivered in terms of Petri nets. Mary has produced a preliminary design (the net in Figure 8) which she wants to discuss with her peers. Hence she initiates an online team meeting. As the common work context she has been setting up her design solution and the course unit informally introducing the case study.



Safety Requirements:

Avoid machine collision!

...
Do not drop metal blanks outside safe areas!

Metal blanks can be dropped for two reasons:

- the electromagnets of the robot arms are deactivated,
- a belt transports work pieces too far.

To avoid this, it suffices to obey the following rules:

1. the magnet of Arm 1 may only be deactivated, if the arm points towards the press and the arm is extended such that it reaches the press,
2. the magnet of Arm 2 may only be deactivated, if its magnet is above the deposit belt,
3. the feed belt may only convey a blank through its light barrier, if the table is in loading position,
4. the deposit belt must be stopped after a blank has passed the light barrier mounted at its end and may only be started after a gadget in the environment of the production cell has removed that piece.

Figure 10: Informal description of safety requirements

John’s first reaction is that he has difficulties to understand the essentials of Mary’s proposal finding it overly complex. To clarify the issue, Mary sets a finger on that part of the informal problem description in course Unit 3, section „Safety Requirements“ stating the safety condition (see Figure 10) she was considering when designing the Petri net. In addition she also points to „enabledMagnet“ in the net and explains: “This place serves exactly the purpose to ensure that pieces held by this arm can only be dropped if the arm is extended because it is only set after transition *extend* has occurred“ (while saying this she moves the finger to the left lower transition of the net). In this moment Barabara jumps in and says: „Do you remember the specification of the safety constraints we developed four weeks ago to formalize the safety requirements?“. She opens another window and puts a new finger on the third and fourth predicate in the premise of the formula presented in this window (see Figure 11). She continues: „These two subexpressions correspond to the case in which the places „out“, „hold“ and „enabledMagnet“ are marked and transition „drop“ is enabled. I think Mary’s draft is fine, we just need to work on other two conditions she’s not considered yet because they represent state information maintained by the controllers of the press and the robot motor.“

...

$$\mathrm{drop}(\mathrm{Arm1}) \Rightarrow \mathrm{empty}(\mathrm{Press}) \wedge \mathrm{pos}(\mathrm{A1P}) \wedge \mathrm{out}(\mathrm{A1}) \wedge \mathrm{hold}(\mathrm{Arm1})$$

...

Figure 11: Formal description of safety requirement 1 in Fig. 10

John: „Ah, now I got it. Let’s keep these relationships in our records so that we can easily reconstruct them.“ He pushes a key combination to materialize the pointers (fingers) as links in the database.

7 Conclusion

Distance education remains a necessity world-wide either for mainly geographical reasons as in Australia or for social-economic reasons as in Europe and the States. Unfortunately, distance education and computer supported learning in general have traditionally suffered from the inability to create the social process of learning together in a group which constitutes a central aspect of knowledge acquisition. Now, with the advent of networked distance learning, with collaboration schemes and support for cooperative activities, like discussions and coordinated browsing, togetherness can be simulated to some extend.

After having reviewed some of the history of CAL and recent movements towards class theatres and telepresence, we have introduced the special requirements for distance education and illustrated its current elements with an example drawn from an engineering course at FernUniversität Hagen. Next we identified requirements which were needed to have shared pointing to objects and stressed the need to have cursors, here called fingers, positioned on logical objects rather than on their display representations.

Examples of possible visualizations are given and questions of control and shared visibility are mentioned. In particular, event broadcasting seems possible which is important in the context of small bandwidth which would not permit display broadcasting over large area networks.

What remains to be seen is how many of the existing interactive applications like Mathematica, Maple, Corel Draw, PHIGS, XGL, etc. eventually “open up” for group cooperation and whether they can be tied into collaborative teleteaching material. More likely, current presentation tools, like the MBONE wb tool, and other Web-technologies will grow into the collaboration direction. Even today, the just mentioned event broadcasting could be simulated by Java applets which can establish a socket link to the server which delivered them.

Our own research goes into much the same direction and focuses on maintaining a common state which is stored in a complex object database and has distributed table representations. Although these have been implemented and operate for a limited number of multimedia types, much of the research presented here is still in the discussion stage.

References

[1] M. Alavi. Computer mediated collaborative learning: An empirical evaluation, MIS Quart. 18:2 (Jun. 1994) 159-173.

[2] Ronald M. Baecker, Dimitrios Nastos, Ilona R. Posner, and Kelly L. Mawby, The User-centred Iterative Design of Collaborative Writing Software, Proc. InterCHI’93, 399-405 (reprinted in Readings in Human-Computer Interaction, R. Baecker, J. Grudin, W.A.S. Buxton, and S. Greenberg eds, Morgan-Kaufmann, 1995)

[3] J. Begole, C. Struble, and C. Shaffer. Leveraging JAVA Applets: Toward Collaboration Transparency in JAVA, IEEE Internet Computing, March-April 1997, pp. 57-64

[4] V. Bush. As We May Think, Atlantic Monthly (July 1945) 101-108.

[5] E. J. Conklin. Hypertext: An Introduction and Survey, IEEE Computer 20:9 (Sept. 1987) 17-41.

[6] Dieter Engbrink, Reinhard Keil-Slawik, and Harald Selke: Teach and learn with interactive media (in German), Heinz Nixdorf Institute, Paderborn, Report No. 45, Dec. 1995

[7] D. James Gemmell and C. Gordon Bell: Noncollaborative Telepresentations Come of Age, Comm. ACM 40:4 (April 1997) 79-89

[8] Werner Geyer and Volker Hilt: A Model for Collaborative Services in Distributed Learning Environments, Proc. Interactive Distributed Multimedia Systems and Services (IDMS ‘97), Darmstadt, Germany, Sept. 1997 (preprint)

[9] T.C. Nicholas Graham and Tore Urnes. Integrating Support for Temporal Media into an Architecture for Graphical User Interfaces, Proc. ACM ICSE ‘97, Boston, MA USA, pp. 172-182

[10] Matti Hämäläinen, Andrew B. Whinston, and Svetlana Vishik. Electronic Markets for Learning: Educational Brokerages on the Internet, Comm. ACM 39:6 (June 1996) 51-58

[11] Don Hardaway and Richard P. Will: Digital Multimedia Offers Key to Educational Reform, Comm. ACM 40:4 (April 1997) 90-96

[12] Gail Kaiser and Jim Whitehead. Distributed Authoring and Versioning, IEEE Internet Computing, March-April 1997, pp. 76-77

[13] Bernd J. Krämer. A Case Study in Developing Complex Safety Critical Systems. In HICSS ‘97, Maui, January 1997.

[14] H. Maurer. Hypermedia Systems as Internet Tools, in: Computer Science Today — Recent Trends and Developments, Jan van Leeuwen (Ed.), Springer LNCS 1000, Springer-Verlag, Berlin-Heidelberg (1995) 608-624.

[15] T. H. Nelson. Replacing the Printed Word: A Complete Literary System, IFIP Proc. (Oct. 1980) 1013-1023.

[16] T. H. Nelson. Getting It Out of Our System, Information Retrieval: A Critical Review, G. Schlechter, ed. Thompson Books, Wash. D. C. 1967.

[17] Open Learning Australia 1997 Handbook, see http://www.ola.edu.au/

[18] B. Shneiderman, M. Alavi, K. Norman, E. Y. Borkowski. Windows of Opportunity in Electronic Classrooms, Comm. ACM, 38: 11 (Nov. 1995) 19-24.

[19] Jens Thamm, Sven Thelemann, and Lutz Wegner: Visual Information Systems - A Database Perspective, Proc. DMS ‘96 Third Pacific Workshop on Distributed Multimedia Systems, Hong Kong Univ. of Science and Technology, Clear Water Bay, Kowloon, June 25 - 28, 1996 (David Du and Olivia R. Liu Sheng eds.) Knowledge Systems Institute, Skokie, IL, pp. 274-285

[20] L. Wegner, S. Thelemann, S. Wilke, and R. Lievaart: QBE-like Queries and Multimedia Extensions in a Nested Relational DBMS, Proc. Int. Conf. on Visual Information Systems, Melbourne, Australia, 5-6 February 1996, C. Leung (Ed.), pp. 437-446

[21] L. Wegner, M. Paul, J. Thamm, and S. Thelemann: A Visual Interface for Synchronous Collaboration and Negotiated Transactions, Proc. Advanced Visual Interfaces (AVI’96), Gubbio, Italy, May 27-29, 1996, T.Catarci, M.F.Costabile, S.Levialdi, G.Santucci (Eds), ACM Press, pp.156-165

[22] Dagmar Wilke, Lutz Wegner, and Jens Thamm: Database-driven GUI Programming, Proc. International Conference on Visual Information Systems (Visual ‘97), La Jolla, CA., 15-17 Dec. 1997 (in print)