

TALE - Teaching Adaptive Lessons Electronically Professional Distance Education Using Hyper-G Technology

**Dieter Freismuth, Denis Helic, Georg Meszaros, Klaus Schmaranz,
Bernhard Zwantschko
Graz University of Technology, Austria
{dfreis, dhelic, gmes, kschmar, bzwan}@iicm.edu**

1. Motivation

A lot of different approaches to distance education can be found since multimedia and the Web were introduced. High level multimedia courseware is normally shipped on CD-ROM because of its size. Furthermore most of the high level CD-ROM productions come with specialized course software. This approach is highly platform dependent because the software is mostly either written for MS-Windows based PCs or Macintosh systems, seldomly both, and Unix is very often simply neglected. Authoring of high level multimedia productions is difficult and expensive and very often cannot be done without low level programming.

On the other hand courseware on the Web is mostly made up of HTML pages with few multimedia add-ons to make it less bandwidth consuming. The advantage of the Web based approach is that standard browsers can be used and users are not forced to leave their well known environments.

Both approaches have one in common: The interactive component is missing. In this sense electronic courseware up to now is not better than standard lectures, just the opposite is true! Taking the standard classroom paradigm a lecture consists of a teacher that interacts with the students in the sense that questions can be asked and unclear points can be discussed.

Following this classroom paradigm an electronic lecture should consist of documents, explanation and interactive discussion. Electronic documents (text or multimedia) in this context correspond to the transparencies in a "real" lecture.

To make electronic courseware more valuable than a standard lecture we have to go a step further: an online lecture session should be recorded similar to making a video tape

of a "real" lecture. Then it can also be used offline. After some adaptations by the teacher it can be taken as the basis for a new lecture.

Following this approach one would get self expanding and adaptive lectures without any additional effort. Clearly it does not make any sense to record a video tape and distribute this tape electronically because of the size of video. Besides this an electronic video is difficult to edit and modify. There exist other methods to get the same, or partially even better results using modern techniques like whiteboards. The bandwidth problem in today's Internet can also be solved using modern replication techniques without losing the interactive component.

Another glimpse at a "real" lecture shows that student--teacher interaction is not the only interactive component: students talk to each other from time to time to clarify misunderstandings or ask other students questions that they do not want to ask the lecturer in public. Also this feature must be included in electronic teaching systems. Students must be able to get in contact with other students electronically. In this case the discussion should not disturb the lecture and should also not be part of the recording.

In the following sections we will introduce TALE - a system that fully utilizes modern server and communication techniques to provide interactive expandable and adaptive multimedia courseware on the Web. TALE keeps all the benefits of non-electronic lectures and adds new features without changing the lecturing paradigm. Authoring of lectures is no longer a time consuming process but it is done automatically during an electronic lecture. All kinds of interaction between students and teacher as well as between students and other students are an integral part of the system. Special replication mechanisms are used to make TALE usable even across networks with low transmission bandwidth. Last but not least TALE is platform independent and can be used with every standard Web browser.

2. System Requirements

The first and most important requirement is derived from the fact that lecturers as well as students are very likely not computer scientists. For this reason the system has to be as simple to use as possible both for authoring courseware as well as for distance learning. This implies the use of standard browsers such as Netscape or Internet Explorer instead of forcing people to deal with completely new software environments. Due to the quick take-off of Java [JavaSoft] over the last year it is nowadays possible to add special functionality to standard browsers by writing applets that run on the client side.

Distance education over Internet also means that data transfer rates can be extremely low. On the other hand good courseware normally contains a lot of explanatory multimedia add-ons that are rather big. Having in mind that distance education also includes online tutoring, standard mirroring mechanisms are not sufficient to solve this problem. With these methods it is either possible to access the mirrored courseware or to take part in

online tutorials. The use of URLs and the lack of intelligent replication mechanisms in today's Web makes it impossible to mix both. A mechanism is needed to replicate space consuming parts of the courseware to a local server or to a CD-ROM and use this material without losing connection to the distant tutoring server during online tutorials.

As has been mentioned above online tutoring is one of the methods that can be used in distance education. In this context online tutoring is a synchronous process. The nature of this process implies that students should be able to ask questions and discuss unclear points. It can happen that students cannot immediately follow the lecturer. For this reason also asynchronous access to such a session has to be provided. Asynchronous does not only mean to access the courseware but a whole recorded session including questions and answers. Also for students learning asynchronously the possibility to ask questions should not be lost. These questions of asynchronous attendees do not disturb the online session but are kept on the server to be answered later. Although not part of the session the lecturer can decide to make asynchronous discussion part of a lesson and add it to the session for later use. The result of this process are self extending lessons that become more and more complete over time.

Always have in mind that students and teachers normally have no visual contact during a session and very often also no direct audio connection due to slow transmission speed. To support discussion under these conditions it is absolutely necessary to have whiteboard-like mechanisms that support textual chat, overlaid pointing and drawing on courseware material without consuming too much bandwidth.

Once a session is recorded it can also be used by the lecturer himself for the next online tutorial. Before using the material it must be possible to edit the session, reorder it, add additional information or remove outdated parts. While teaching such a recorded session the lecturer must have the possibility to stop it for questions, discussion or additional explanations. The whole new session including previous recordings should again be recorded as a session of its own to provide asynchronous access and offline teaching.

Sofar we have been talking about sessions that include a lecturer. All recorded lectures can be accessed by a student or a group of students without a lecturer. In the case of a group of students working with the same recorded lecture they must be able to find out who else is working with it and get in contact with the others to ask questions and discuss unclear points. It must also be possible to leave a question for the lecturer if something remains unclear which will be answered later.

Not all students want everybody to see their questions or remarks. On the other hand it is also not desirable for lecturers to have courses on a server with too many questions attached to them. This is the point where user access rights come into play. Supporting user access rights everybody can decide who should be able to read the comments. In this case students can also go as far as to leave a question or remark only for a group of other students but not for the lecturer himself. More than this: students can attach their own private remarks to courses without anybody being able to see them.

Another important fact that lies in the nature of learning has to be considered: not all the students have the same skills and not all the students are the same types of learners. Some students are visual types, others can deal better with audio, some like examples, others like textual descriptions better. It would be desirable if the students could choose the type of material they like to make learning easier for them. Additionally a nice feature would be to make the material skill dependent. This means students that are very much into a certain knowledge area only get high-level information, others get courseware that is more comprehensive and also includes more or less trivial explanations. The possibility to switch back and forth between different types of material and skill must be provided.

3. System Architecture

Considering the requirements above the whole TALE system has to be very adaptive to dynamics on the Internet, dynamics during online tutoring sessions, updates in courseware and material by annotations and discussions and last but not least adaptive to the skills and special likings of students.

Three different parts of the system with different characteristics can be identified:

- **Material:** The static part of lectures. The term material includes all prepared documents used to illustrate lectures. Mostly it will be textual material with multimedia add-ons. Multimedia add-ons can be of any type from images over video or sound to 3D scenes and more. Although material is static its use is highly adaptive to the students' skills and learning techniques. For this reason an intelligent mechanism is needed to provide alternate views and switch back and forth between different representations of the material used.
- **Course Flow:** The dynamic part of lectures. This includes the order and (adaptive) timing of material to be shown along with the lecture itself. Also discussion using chat and whiteboard are part of the course flow. Additionally the course flow itself is annotable and extendable over time depending on the lecturer. The course flow itself is independent on the adaptive use of material addressed above.
- **Tools:** All necessary Java applets that make standard browsers able to use time synchronized material as well as messaging, chat and whiteboard features. Supplementary applets needed for authoring and adapting the courseware.

The courseware server (that can in fact be split across multiple physical servers) now contains the whole material used for lectures including alternate views of the material. Furthermore it contains all the tools needed for viewing and authoring lectures and all the communication tools. Last and most important it contains all lectures represented by course flow structures that use certain material and tools of the courseware server.

A course flow can be considered to be made up of hyperlinks to courseware material with additional type and timing information. This principle is comparable to what is called transclusion in the Xanadu system [Nelson 93]. At this point we do not distinguish

between already prepared courses and online tutorials because both follow the same principle. The only difference is that an online tutorial can be recorded to produce packaged courseware. According to [Ottmann, Bacher 95] we call this authoring on the fly. In fact existing courseware or also selected parts of different courseware packages together with add-ons can be used to prepare new lectures.

Having a look at the highly adaptive and dynamic behaviour of the system it is necessary to have a courseware server that automates context switches and consistent dynamic hyperlinking. Also features like session oriented behaviour as well as user and group access control must be integral parts of the system. The strongest requirement of all is full compatibility with standard Web Browsers to make courseware accessible for the widest possible range of users.

We decided to use Hyper-G ([Andrews et al. 95], [Maurer 96]) as a basis for the whole system since it is nowadays the only available Web compliant server supporting all the document management and session oriented features that are needed without additional programming:

- **Alternate Views:** Hyper-G with its structured information space supports the concept of document clusters. Several different views of the same information (e.g. text, multimedia, multilingual) can be clustered together and the user can switch back and forth between the alternate documents on the fly. No additional hyperlinks have to be provided.
- **Dynamic Hyperlinking:** Integral part of Hyper-G is the concept of document sequences which is in fact the overall characteristic of an electronic lecture. A sequence in this context means a number of documents that are ordered, in this case the order is time-sequential. On the other hand the number of documents is not constant but depends on the students' skills and access rights. Besides this the documents themselves can also be document clusters as described above. Again students can change their preferences on the fly and get more or less explanatory material. Using standard hardcoded hyperlinks instead of Hyper-G's dynamic sequencing mechanisms would make it absolutely impossible to achieve this behaviour.
- **Session Orientation:** Hyper-G's protocol supports user sessions and user tracking, even using standard http protocol. This allows students and teachers to have a look who is attending the current session and get in contact with others. It also allows to trace the way of students through additional material in the database and get useful feedback about certain courses. If for example all students search for additional information when visiting a specific document it becomes clear that the document is difficult to understand and should be adopted.
- **User Access Rights:** Integrated user access rights in the Hyper-G server allow it to have private annotations and add-ons. It is also possible to write remarks that can only be read by certain users or usergroups. This allows students to have their own learning groups.
- **Stable Hyperlinks:** Hyperlinks in Hyper-G are represented by URNs (Universal Resource Names) instead of URLs (Universal Resource Locators) and stored in a

- separate link database. The benefit of this concept is it to be able to move documents around in the server or even from one server to another without producing dangling links. It is also possible to attach additional links to documents without having write access to these documents.
- **Typed Hyperlinks:** Links in the database also carry link types. This makes it possible to distinguish between standard referential hyperlinks, annotations, questions, answers and more. The different types of hyperlinks can be represented using different icons so that the user knows beforehand which kind of information is found behind a hyperlink. As an example if a student is looking for additional information this student will certainly not want to follow a link to a question but to an explanatory add-on. Knowing in advance which link leads to the explanation the student can avoid to get uninteresting documents.
 - **Bidirectional Hyperlinks:** Hyperlinks in Hyper-G can be followed in both directions from source to destination and also from their destination back to the source. Together with the link type mentioned above this allows it to comment documents by simply letting a link of type comment or annotation point to the commented document. By following this comment link backwards the reader can access the comment.
 - **Document Replication:** Besides standard caching algorithms Hyper-G supports a special document replication mechanism [Kappe 95]. Document replication is similar to mirroring with the difference that the mirrored documents know about their origin. Let us consider users connected to a replication server but accessing data on the original courseware server during an online tutorial. It is not necessary for the users to know for which documents replicas exist. Whenever they try to download a document from the remote server they will automatically get the local copy instead. This mechanism also works in combination with CD-ROMs.

On this basis an extendable Java toolset was developed implementing authoring on the fly for courseware preparation as well as discussion and whiteboard functionality. To allow easy implementation of new functionality like video conferencing the toolset is based on what we call an Open Messaging Architecture [Freismuth et al. 96]. Open messaging means that all the tools use messages to communicate but the message format itself is not fixed. This allows to implement new protocols and tools that fit into the system without having to care about existing applets. Unknown messages are simply ignored by the applets similar to the Smalltalk approach. The main difference is that all messages can be delivered locally as well as across the network making it easy to write programs that can for example be remote controlled. Beyond that messages can also be broadcast to several other open messaging applets on the network which is extremely useful for online tutoring and multi conferencing. It is not even necessary for an applet to know who is listening because this is handled by the open messaging protocol kernel internally.

4. Authoring and Using Courseware

As has been mentioned earlier courseware consists of courseware material and the course flow. The term authoring on the fly in this context means to compose a course flow document on the fly during a lecture using mostly static course material. Mostly static is meant in the sense that all the documents used in a lecture are statically residing on the main course server and its replication servers and only few new course material is produced during the session. The new course material produced during the session are explanations by the lecturer and contributions by the students during discussions.

Depending on the environment in which the system is operated new material will mostly be of type text or audio, only in ultra high bandwidth environments it could be video which is not very likely at the moment.

An online lecture is started at a certain time and all the students wanting to take part either connect directly to the tutorial server (the one the lecturer is connected to) or indirectly via their local replication server. Technically spoken a low bandwidth message broadcasting connection between the lecturer and the students is established. Due to the Java implementation of the Open Messaging Architecture it is not necessary to install any software, neither for the students nor for the lecturer, the applets are automatically downloaded from the server when needed. However it is highly recommend to keep the applets locally on the client side to save time.

Every action of the lecturer is broadcast in form of a small message to the students. Documents are not transmitted using the messaging protocol but URNs are broadcast instead and the according documents are requested by the clients. Using this approach takes full advantage of document replication since the students download the course material from their local replication server instead of the remote course server. Only documents that are not available as replicas are transmitted over long distance. So far the system is in what we call a "follow me" mode.

In addition to broadcasting the actions to the students these actions are recorded on the main course server for later use. Also the students themselves can store private recordings of the session if they have the necessary user access rights on their server.

Certainly a simple "follow me" mode is not enough to make up a good lecture. For this reason the lecturer has the possibility to draw on documents used during a session. Drawings are treated as overlays and also broadcast to the students synchronously. Also the drawings are recorded including timestamps to be used for following sessions or offline courseware material.

The only thing that is missing now to make up a fully fledged hypermedia course are the lecturer's explanations. Depending on the bandwidth the lecturer can choose to use either text or sound for explanations. Video will not make too much sense as long as bandwidth on Internet is not sufficient. As can be expected also the explanations are recorded for later use. We call this an "enriched follow me" mode.

If a student now has a question or comment it is possible to signal this to the lecturer who in turn can decide to accept (and answer) the question or to skip it. In case of accepting the question it is broadcast to the other students along with the answer and is also recorded. There is also the possibility to ask questions that are not broadcast which allows two or more students to discuss a topic without disturbing the lecture. During a discussion the students can also utilize the "follow me" or even the "enriched follow me" mode as well the drawing facilities to express themselves more clearly.

Students that do not have enough knowledge to follow an online session at standard speed also have the choice to switch to asynchronous mode. In this case they follow the recording of the session at their chosen speed even if the session is not over yet. Switching to asynchronous mode also means to lose the possibility for online dialog and discussion which would disturb the lecture. Although not online it is possible to post questions that can be answered by the lecturer at a later stage.

When the lecture is over a complete recording of this session including discussion is available for later use. This recording can be immediately replayed by everybody who has read access rights to it. The lecturer also has the possibility to edit the session, add additional information, skip parts of it or enrich it by additional comments.

The whole mechanism of recording and editing a session can also be used without online students as a comfortable and quick way to prepare new offline course material.

Using recorded courseware offline students have the same possibilities as if they were taking part in an online tutorial, only the lecturer is missing. Even the "follow me" mode can be used. Students can get in contact with others that are online too and they can discuss unclear points. They can add comments or questions to the lecture. They can even record their discussions and add them to the lecture as a comment. Students' comments do not really become a part of the courseware material, but others will get the possibility to follow a hyperlink to the additional material. If the material is interesting enough the lecturer can decide to edit the courseware and add parts of the commentary material. Taking advantage of user access rights the students can even choose to discuss certain topics in closed user groups.

5. References

[Andrews et al. 95] Andrews K., Kappe F., Maurer H. and Schmaranz K.: "On Second Generation Network Hypermedia Systems", Proc. ED-MEDIA '95, (1995), 69--74.

[Freismuth et al. 96] Freismuth D., Helic D., Meszaros G., Schmaranz K., Zwantschko B.: "DINO - Distributed Interactive Network Objects - The Java Approach", submitted to ED-Media '97.

[JavaSoft] JavaSoft Inc.: "Java Home Page", <http://www.javasoft.com>.

[Kappe 95] Kappe F.: "A Scalable Architecture for Maintaining Referential Integrity in Distributed Information Systems", J.UCS 1/2 (1995), 84--104.

[Maurer 96] Maurer H. ed.: "HyperWave - The Next Generation Web Solution", Addison-Wesley, (1996).

[Nelson 93] Nelson T.: "Literary Machines", Mindful Press (1993).

[Ottmann, Bacher 95] Ottmann T., Bacher C.: "Authoring on the Fly", J.UCS 1/10 (1995), 706--717.