DOCUMENT RESUME

ED 466 141 IR 021 211

AUTHOR Brennecke, Andreas; Selke, Harald
TITLE The HyperSkript Authoring Environment--An Integrated Approach for Producing, Maintaining, and Using Multimedia Lecture Material.
PUB DATE 2001-06-00
NOTE 7p.; In: ED-MEDIA 2001 World Conference on Educational Multimedia, Hypermedia & Telecommunications. Proceedings (13th, Tampere, Finland, June 25-30, 2001); see IR 021 194.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Authoring Aids (Programming); Computer Mediated Communication; Computer Software Development; *Cooperative Planning; Courseware; Foreign Countries; *Hypermedia; Instructional Innovation; Instructional Materials; *Lecture Method; *Material Development; *Multimedia Materials
IDENTIFIERS Germany

ABSTRACT

Based on a technical infrastructure that supports face-to-face university teaching, an environment that enables small groups of lecturers to develop and maintain lecture material cooperatively was developed. In order to allow for a flexible use, only a few formal workshops are imposed on the users while cooperation is supported by easy-to-use mechanisms. The conventional group of authors who produce for unknown learners is replaced by a group of authors who are at the same time teachers that use the material that has been produced jointly. They use the material in their own lectures and improve it based on their teaching experiences-in nearly the same way as they would do with their conventional lecture notes. The distributed multimedia lecture material, or HyperSkript, allows fore more flexibility in the use without raising the expenditure of producing the material significantly. The functionality of an existing Web server is extended to support and integrate individual, distributed, and cooperative processes that are involved in the production and maintenance of multimedia lecture material. (AEF)

The HyperSkript Authoring Environment—An Integrated Approach for Producing, Maintaining, and Using Multimedia Lecture Material

Andreas Brennecke, Harald Selke
Heinz Nixdorf Institut, University of Paderborn
Fürstenallee 11, 33102 Paderborn, Germany
{anbr|hase}@uni-paderborn.de

Abstract: Based on a technical infrastructure that supports face-to-face university teaching, we have developed an environment that enables small groups of lecturers to develop and maintain lecture material co-operatively. In order to allow for a flexible use, only few formal workflows are imposed on the users while co-operation is supported by easy-to-use mechanisms. We have extended the functionality of an existing web server to support and integrate individual, distributed, and co-operative processes that are involved in the production and maintenance of multimedia lecture material.

Introduction

Traditionally, multimedia material that is being used in university teaching has been pre-produced by a single author or a team of authors. Students may then work with this material independent of place and time. The main goals of using technology are on the one hand to produce material that fosters learning by providing interactive techniques, animations, and simulations. On the other hand, communication tools are used to improve the contact between lecturers and students. The authoring tools with which the multimedia material is produced usually impose a strict separation of production and reception. Communication channels, if they are established at all, only serve as means for the distribution of material and information.

In this contribution we present an approach that transcends the one-way street of creation and reception. Instead, it promotes the distributed, yet integrated production, maintenance, and use of multimedia material. The conventional group of authors who produce for unknown learners is replaced by a group of authors who are at the same time teachers that use the material which has been produced jointly. They use the material in their own lectures and improve it based on their teaching experiences—in nearly the same way as they would do with conventional lecture notes. Thus, in addition to enhance lecture material multimedially, an integrated environment is provided that enables lecturers and students to work with own as well as pre-produced material. Our distributed multimedia lecture material (or HyperSkript, for short) allows for more flexibility in the use without raising the expenditure of producing the material significantly.

In order to allow for a most flexible use of the HyperSkript, authors may largely use the tools they are accustomed to, which is to say that standard technologies are employed to the largest possible extent. In addition to the authoring environment described in this contribution, a teaching environment and a number of supplementary tools have been developed some of which support the production of multimedia material—like interactive animations or the audio annotations described in (Grimm & Hoff-Holtmanns 1999)—while others simplify the use of the material by allowing users to create their own views on the documents. Semantic maps for example enable users to create a graphical overview on a specific subject. They act as a navigation tool. Individual copies of the maps can be modified by the learners and adopted to their actual knowledge (see Klemme et al. 1998 and Hampel & Selke 1999). The overall structure of the HyperSkript is shown in Fig. 1.

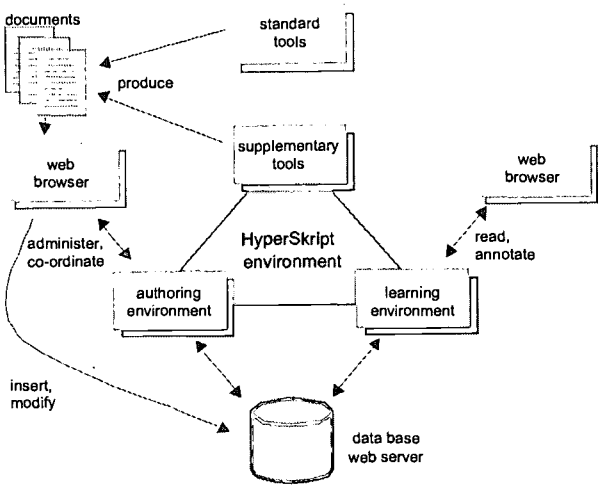

Figure 1: The HyperSkript environment extends standard web technology by providing functions for the co-operative production of documents (authoring environment), functions for the active use of material (learning environment), and supplementary tools for the production of multimedia documents. The authoring as well as the learning environment are accessed via standard web browsers.

Requirements for the authoring environment

The experiences made in several years of using new media in university teaching (as described in Brennecke & Keil-Slawik 1995, Brennecke et al. 1997, e.g.) served as a starting point for the implementation of a new authoring environment. After having established an infrastructure that enabled users to work with electronic material at all locations where learning takes place (at home or in the library as well as in the rooms where tutorials and lectures are held), the production and maintenance of this web based material should be supported efficiently. Additionally, there was the idea of co-operating with a lecturer from another university in a way that allowed to create and maintain the material for lectures and use it at different universities while still teaching students in an individual way.

The HyperSkript contains all documents necessary for a course on a specific subject. This material is produced by several authors and integrates documents that all authors have agreed upon as being relevant for the context as well as documents that reflect their particular competencies and preferences. In that way, the material may be used in different courses while still allowing to present individual lecturing approaches. To allow this, the lecturers create different views for each single course by selecting the appropriate material. The material within the document base remains unmodified but may be supplemented by additional documents.

Co-operative development and maintenance are also important factors in rationalising the process of producing multimedia material which is a crucial aspect when it comes to working with such material on an every-day basis. The material needs to be structured in a way that it can be easily adapted or extended to integrate new results from research or simply modify the course to better meet the students' goals. The latter goes along with the requirement that not only the lecturers but also the students may individualise the material according to their needs and preferences. Thus, the students should also be able to work with the documents within the HyperSkript while leaving the original material in a consistent state.

As a result, the lecture material is designed to incorporate a high-quality core of material that has been produced co-operatively. This core is then surrounded by individual, sometimes short-lived, material like additional documents for a particular course, complementary notes, personal annotations of the students, or audio-annotations of a lecture. The different user groups (authors and students of different courses) thus need to be presented different views onto the material. The material should be structured in a modular way to allow for easier maintenance on the one hand and multiple use in different contexts on the other hand.

HyperSkript Architecture

Like in a textbook written by a group of authors, the authors have to co-operate on the question which contents to include in the HyperSkript and on the way these are broken down into modules as well as the multimedia and pedagogic design. The topic modules may consist of documents of various types ranging from images and texts to sets of slides, animations, or even CBTs. However, to guarantee that the material may be used in flexible ways and easily combined with other modules, each module should be self-contained and contain all documents necessary for dealing with that particular topic. These modules are stored in the so-called compound layer of the HyperSkript which is being produced and maintained co-operatively by the participating authors. For each module a "main author" may be specified who is responsible for editing the content of that module. However, a module may only be released, i.e. given read access by the students, when all authors have agreed on it. Each module may also contain meta-data, in order to generate data according to the LOM standard.

The material in the compound layer may be complemented by background material like, e.g., journal and conference papers, images, or movie clips illustrating a relevant aspect. As this material is "objective" in the sense that it does uring approach but rather documents that have been produced in a different context, the authors need not agree on the content. The authors may thus decide to distribute the maintenance of these documents—stored in the so-called base layer—in order to reduce the amount of work for each single author. The base layer is therefore sub-divided into areas, each of which is administered by one responsible author who alone may change or update content in this area. The other participating authors are notified of all relevant changes in any area of the base layer.

Finally, the topic modules and the background material have to be integrated with course-specific documents which finally make up the particular course—usually held by one of the authors. The assembly is done within the so-called course layer which is administered individually by the respective author according to the context in which the course is given. The architecture is shown in Fig. 2. The HyperSkript authoring environment aims at efficiently supporting the authors in creating, maintaining and updating the shared documents within the compound layer co-operatively and within the base layer in a distributed manner.

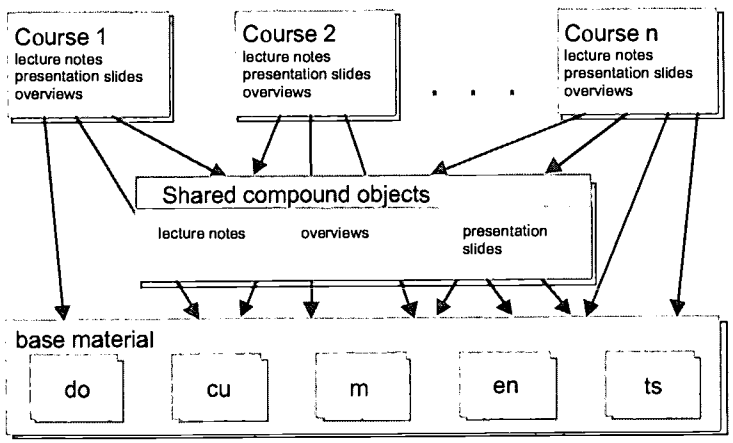

Figure 2: A HyperSkript consists of three layers which reflect different kinds of co-operation.

As opposed to systems that support co-operation in large distributed organisations which aim at the co-ordination of work processes by the implementation of formal workflows, a HyperSkript is produced by a small team of authors who co-operate voluntarily to achieve a common goal: the efficient production and maintenance of lecture notes that make use of multimedia and are of a high quality. In that sense, producing a HyperSkript is comparable to a group of lecturers and their assistants editing a textbook for university teaching. Therefore, the HyperSkript authoring environment does not implement any formal procedures but rather guarantees that the processes of creating and updating the material are transparent to all participating authors. For small groups of authors who trust each others, an optimistic "relaxed mode" allows the authors, e.g., to agree on updating a particular piece of text while in a meeting or even when talking to each others on the phone. Within the authoring environment one of the authors may then sign the agreement in the name of another author. However, it will be stored who actually signed in the name of

which author. Where such informal co-operation seems inappropriate, a "strict mode" requires each of the authors to log in to the authoring environment and sign the agreement themselves.

In the co-operation, the authors may use software they are accustomed to to the largest possible extent. They may use any kind of synchronous or asynchronous communication; the authoring environment simply allows to easily access these external tools or the necessary contact information. While the production of material is external to the authoring environment, it allows the authors to efficiently manage the documents, guaranteeing that students always see a consistent version of the lecture notes while the authors may create new versions of topic modules in a special editing area. All documents—including those in the course layer which may also be produced by the students—need to be managed within the system.

Using the learning environment (cf. Bader et al. 1999 and Meier & Holl 2000), the students work with the material in a way very different from the way the authors edit the material with the authoring environment. They access the HyperSkript from an entry point in the course layer which the lecturer supplies. The learning environment then allows each student individually to restructure the documents according to their own preferences, complement them with their own documents, and annotate existing documents. Students may also form groups and create their own private workspaces. While they have read access to all material of the HyperSkript which the lecturer decides to present, the internal HyperSkript architecture with its three layers is invisible to them.

HyperSkript Realisation

The HyperSkript authoring environment has been implemented on top of the Hyperwave Information Server—a specialised web server—which was chosen because of its strong orientation towards document management, its support for asynchronous co-operation and its extensibility on the server side as well as the client side. The latter allows for different user interfaces reflecting the special needs of authors and readers respectively while managing all documents within a single database. The mechanisms needed have been implemented using the Hyperwave Application Programmer's Interface in Server Side JavaScript.

In order to achieve an optimal integration into the authors' workplace, all functions of the authoring environment can be accessed with web clients of the fourth generation. Documents can be integrated into the environment using any of the Hyperwave clients, which again means using a web browser or one of the extensions to MS Windows programmes like the Virtual Folders which allow to access a Hyperwave Server from the Windows Explorer in a way similar to accessing a file system.

Within the authoring environment, documents of different kinds can be maintained and edited co-operatively. In addition to atomic documents, like texts, images, or audio clips, complex objects consisting of several basic documents like multimedia applications or hypertexts are supported. The authoring environment only implements the mechanisms related to co-operative document management (finding new or modified documents, releasing documents, creating draft versions etc.). The editing of the individual documents is done using the usual tools like HTML editors or office suites.

HyperSkript layer	type of work	material
course layer	individual	course-specific
compound layer	co-operative	transferable
base layer	distributed	universal

Table 1: The layers within the HyperSkript architecture reflect different types of working and different levels of co-operation.

As depicted in Tab. 1, the HyperSkript layers each require different mechanisms to support the co-operation needs. Presenting and editing the material within the *course layer* is done *individually* and thus does not need any kind of co-operation within the authoring environment. However, it needs to be guaranteed that students always see the latest released version of any document. In addition to basic mechanisms such as access control and locking mechanisms to prevent simultaneous write accesses, both other layers require specific kinds of co-operation support. When editing documents of the *compound layer*, a draft version must be created which can then be worked upon. The older version remains visible to the readers of the HyperSkript as long as this draft has not been signed for

release by all team members in a *co-operative* process. In a similar way, new topic modules are always created in a special area which is invisible to the readers. Because production and administration are *distributed* among the authors in the *base layer* by division of labour, there is basically the need for notification as described above.

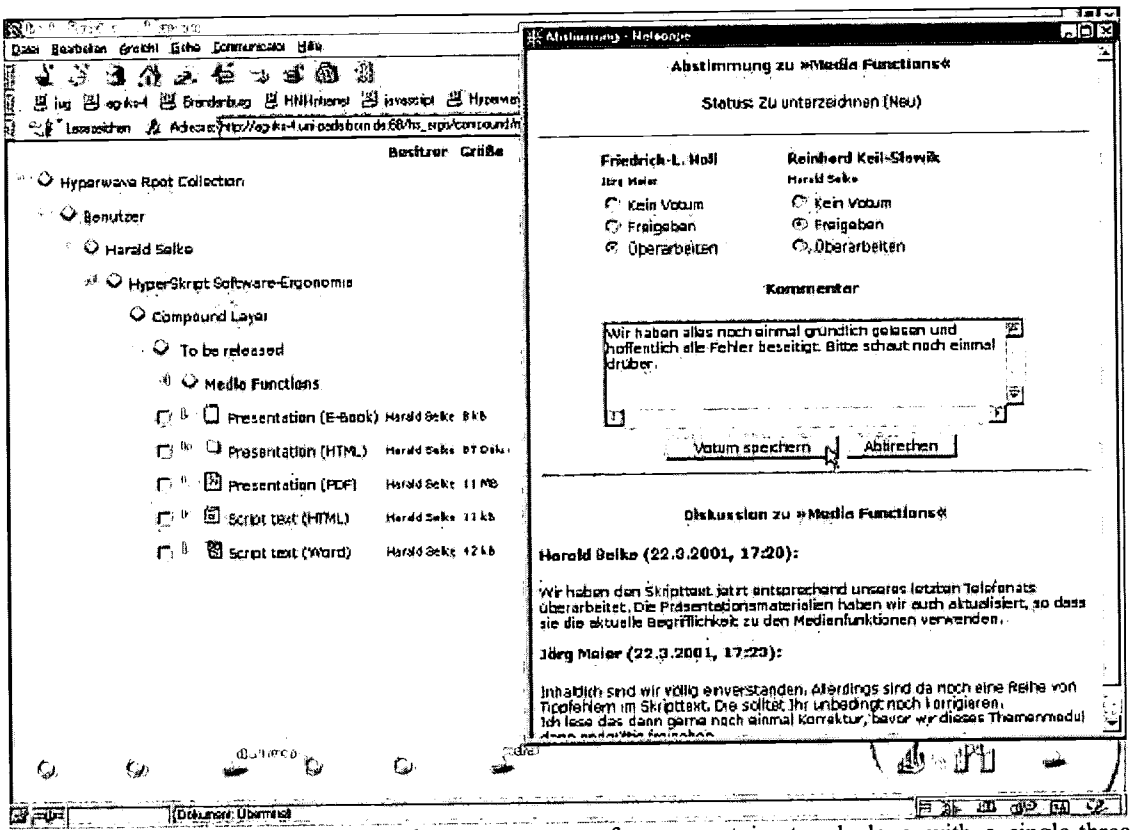

Figure 3: Discussing a topic module. The current state of agreement is stored along with a single-threaded discussion. An author can agree or disagree using the radio buttons in the top part of the dialog.

In order to efficiently support the different aspects of co-operation, certain awareness information is needed. According to Gutwin and Greenberg (Gutwin & Greenberg 1997), the following questions are crucial:

- Who is currently working with the system? A list of the team members who are currently logged in is visible to the authors. These team members may then be contacted easily by synchronous communication means such as a chat, a phone call, or a video conference. Given appropriate conferencing software, synchronous editing using a shared application is also possible. The HyperSkript authoring environment itself allows to specify any such contact information within a special user object. It does not implement any tools for synchronous communication but rather allows to easily access them via external tools from within the environment.
- Who is currently working on which documents? An author might want to contact another team member who is working on a particular document synchronously, e.g. because both are working on documents that belong together. This kind of awareness information is currently not supported by the HyperSkript authoring environment.
- Which documents have been modified or newly created by other authors? This vital information can either be distributed by notifications to the other team members—e.g. by sending an e-mail with a list of new or modified documents—or be gathered by each of the team members individually by using an appropriate search query. Currently, the HyperSkript authoring environment only supports a manual notification mechanism in addition to a flag shown next to edited or new documents. Because we expect that there will be many minor changes—such as the correction of typos—we decided not to rely on an automatic notification mechanism but rather log important events and leave it to the authors to send this log to other team members when appropriate. However, a search mechanism will be provided in the next version. In contrast to ordinary workflow systems within our "relaxed mode" each author has to decide which modification may be important enough to inform the other

authors respectively which one have to be agreed by the others. (The correction of orthographic mistakes typically requires no reaction from the co-authors e.g.)

- Which state is a document currently in? At any point, the team members need to be provided with versioning and editing information. For that purpose, draft (or experimental) versions carry a certain attribute and are stored in designated areas of the data base. For new and revised topic modules of the compound, a voting mechanism is supplied where each author can state their agreement or disagreement on including the document in the current version into the HyperSkript. An author can decide to release a topic module for which he is responsible within the compound layer at any time. However, before it is actually released, all other authors have to agree on releasing that version. In addition to the current vote (who has agreed, who has disagreed, who has not cast a vote yet), there is a basic type of discussion forum that allows to make comments and suggestions for improvement. As soon as all authors have agreed on releasing the module, it is integrated into the HyperSkript and thus becomes accessible for the readers. The discussion is stored in an archive; older versions or drafts that have not been released may be archived. Fig. 3 shows the discussion of a topic module within our HyperSkript authoring environment.

Conclusions

The HyperSkript authoring environment supports teams of university lecturers who wish to co-operate in the production of multimedia lecture notes emphasising the need for developing and maintaining documents on a long-term perspective. A web based document management server has been extended to support co-operation processes in distributed authoring. Important features include awareness information, notification and voting mechanisms, creation of draft versions and easy-to-use release mechanisms. By integrating these new functions into an environment of standard software that is used in the every-day work processes of the lecturers, the HyperSkript approach seems to be a suitable tool for individual, co-operative, and distributed production, maintenance, and use of multimedia lecture material.

The authoring environment described in this contribution is integrated into the HyperSkript project in which an additional learning environment and supplemental tools have been developed. A prototypical script for courses on software ergonomics and design of multimedia systems has been produced, improved continually over two years by two authors, and used at two universities in different courses. Evaluation so far shows that students rate the material as becoming better over time. However, this good result may also have been influenced by other improvements in the accessibility of the material and requires a more detailed evaluation.

References

Bader, O., Holl, F.-L., Meier, J. (1999). HyperSkript—Designing a new Learning Environment. In: *Proceedings of HCI International'99—8th International Conference on Human-Computer Interaction*. IRB-Verlag, Stuttgart, 239-240.

Brennecke, A., Keil-Slawik, R. (1995). Notes on the "Alltagspraxis" of Hypermedia Design. In: H. Maurer (ed.): *Educational Multimedia and Hypermedia, Proceedings of ED-MEDIA 95*. Association for the Advancement of Computing in Education, Charlottesville, VA., 115-120.

Brennecke, A., Schwolle, U., Selke, H. (1997). The Evolution of an Electronic Teaching and Learning Environment. In: Müldner, T., Reeves, T. C.: *Proceedings of ED-MEDIA/ED-TELECOM 97*. Association for the Advancement of Computing in Education, Charlottesville, VA., 98-105.

Grimm, R., Hoff-Holtmanns, M. (1999). Evaluating a Simple Realization of Combining Audio and Textual Data in Educational Material—Making Sense of Nonsense. In: Collis, B., Oliver, R. (ed.): *Proceedings of ED-MEDIA 99*. Association for the Advancement of Computing in Education, Charlottesville, VA., 1390-1391.

Gutwin, C., Greenberg, S. (1997). Workspace Awareness. Position paper for the ACM CHI'97 Workshop on Awareness in Collaborative Systems, Atlanta, Georgia, March 22-27 1997.

Hampel, Th., Selke, H. (1999). Customizing the Web—Two Tools for individual and collaborative use of hypermedia course material. In: Collis, B., Oliver, R.: *Proceedings of ED-MEDIA 99*. Association for the Advancement of Computing in Education, Charlottesville, VA., 634–639.

Klemme, M., Kuhnert, R., Selke, H. (1998). Semantic Spaces. In: K. Höök, A. Munro, D. Benyon (eds.): Workshop on Personalised and Social Navigation in Information Space. SICS Technical Report T98:02, Kista, Sweden, March 1998, 109–118.

Meier, J., Holl, F.-L. (2000). HyperSkript: eine multimediale Intranet-Lernumgebung. In: Uellner, S., Wulf, V. (eds.): Vernetztes Lernen mit digitalen Medien. Proceedings of the first conference „Computergestütztes Kooperatives Lernen (D-CSCL 2000)". Physica-Verlag, Heidelberg, 103-116.

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

NOTICE

Reproduction Basis

☒ This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").