

Linking by Interacting: a Paradigm for Authoring Hypertext

Maria da Graça Pimentel

Departamento de Computação
SCE-ICMC
Universidade de São Paulo
São Carlos, SP 13560-970, Brazil
E-mail: mgp@icmc.sc.usp.br

Gregory D. Abowd

College of Computing & GVU
Center
Georgia Institute of Technology
Atlanta, GA 30332, USA
E-mail: abowd@cc.gatech.edu

Yoshihide Ishiguro

Human Media Research Lab
NEC Corporation
Takayama-cho, Ikoma
Nara 630-0101, Japan
E-mail: ishiguro@hml.cl.nec.co.jp

ABSTRACT

Authoring hypertext structures has always been a difficult task. The cognitive overhead problem, well known from authors when creating structured hyperdocuments, has been tackled in a variety of forms. In this paper we advocate that capture-based systems should support flexible hypertext structures generated by *linking by interacting* operations. These include *linking by capturing* operations — which capture user-interactivity during a live session — and *linking by augmenting* operations — which allow the captured contents to be extended with complementary operations available outside the live session. We demonstrate how such combination leads to the generation of flexible hypertext structures by presenting our implementation in the educational domain.

KEYWORDS: Authoring hypertext, User-hypertext interaction, Linking, Capturing, Presenting, Augmenting.

INTRODUCTION

Authoring hypermedia structures is acknowledged by many authors as being a hard and expensive task. This is true whether or not the structures are static — by static structures we mean that, although the hyperdocuments allow users some level of interaction to decide when to play or visit their information, the hypertext structure and contents are fixed. The approach we propose is the instrumentation of an environment where the live experiences themselves can be used to author hypermedia documents corresponding to flexible and extensible records of those experiences, allowing the generation of an associated hypermedia document at almost no cost. The captured information can be associated to a truly hypertext structure — in opposition, for instance, to a linear structure resulting from simply capturing audio and video information.

The capture-based approach exploits the fact that everyday experiences are rich in information, which, when captured and made available as hypermedia documents, provides a valuable source of information. It also assumes that humans need support in capturing the salient memories from live experiences that must be called upon at a later time. There are many situations in our everyday lives that satisfy both of these criteria. Two environments that have been exploited are meeting rooms and classrooms, with several automated capture systems reported in the literature (see [4] for a review).

One problem with using the capture approach in isolation is that there is no easy way to integrate the contents presented during the live session with information produced outside the session. As far as the captured material is concerned, information prepared before the session will be integrated in the resulting document only if it is in fact presented during the session. This can leave out important material created in advance and meant to be accessed before or after the session takes place. Moreover, material produced as a follow up of a session will not be made part of the hypermedia material associated to the session. This problem is not tackled by current capture-based systems — in most cases, static documents are produced as a result of capturing. In such systems, the information generated is not meant to be modified nor extended in any way. In fact, when static information is produced, the user is restricted to interact with the original presentation.

As a solution to those limitations, we propose that capture-based systems support flexible hypertext structures generated by *linking by interacting* operations. These operations will be detailed in the context of an application in the educational domain that demonstrates our approach.

However, the provision of linking by interacting operations makes sense only when the resulting flexible hypertext structure can be exploited. We demonstrate the appropriateness of our approach by illustrating how the hypertext structure can be used to generate alternative views of the overall information and by discussing the hypermedia information generated in real courses supported by our application.

The remaining sections of this paper are organized as follows. First, we discuss some related work. We then introduce the Classroom 2000 project —the platform we have used to implement our approach. Next, we describe in detail the *linking by interacting operations* we have implemented. We then present some of the services we have implemented that exploit the flexible hypertext structure, followed by a discussion of data gathered from real courses. We conclude by summarizing the benefits of our approach and highlighting some future work.

RELATED WORK

Facilitating the task of authoring hypertext structures has taken a few distinct approaches. The most traditional approach includes models and tools for the authoring of hyperdocuments to be done as a dedicated and exclusive task (e.g. [19] [27]). One alternative for such manual effort is to provide more or less automatic services to generate the hypertext structure by exploring information retrieval techniques (e.g. [5]). Another alternative is to restrict the structure to guided-tours [33] or paths [35] [13] used in isolation or combined with other techniques such as information retrieval [15].

The authoring of less formally structured spatial hypertexts has also been investigated and supported [23]. In this case, the aim is to support complex authoring such as information analysis where traditional node-link bases structures are not necessary and may be even disrupting. Extensive research in this area include its application in the generation of information workspaces in general [22] [31], the authoring of web-based presentation [30] and of paths in the educational domain [29].

Shahraray and Gibbon's approach supports the automatic generation of hypertext documents by processing the contents and structure of video programs that are accompanied by closed caption information [28]. Also exploiting existing information, Auffret et al. present a model processing archive audio-visual documents to support the authoring of associated hypermedia documents [6].

As far as capture-based systems are concerned, no reported work attempts to exploit the underlying hypertext structure generated in order to allow the provision of flexible and extensible hypermedia documents. Such provision has been advocated by many hypertext researchers — for instance to support changing information [18] and annotation [21] as well as reuse [14]. The seminal capture work on the Tivoli system at Xerox PARC [24] [25] was used to support a single scribe summarizing the results of a single technical meeting. Though the system was used over an extensive period of time, the information produced did not allow extensions. A capture system developed at Microsoft Research [8] attempts to provide support for annotations during and after the live event, but does not try to incorporate activities or comments that occur before the presentation. They also do not address implicit linking between different presentations.

A more comprehensive approach has been taken in the design of DOLPHIN, a groupware application built to support co-located or distributed meetings [32]. The capture session allows users to extend a prepared agenda and to record handwritten information that, as in Tivoli, could be manipulated during the meeting by editing operations that included gesture recognition. Support to integrating information produced before the sessions is done by allowing the users to import text, images and prepared hypermedia information. Support to integrating information produced after the sessions is done by allowing the captured contents to be imported to a collaborative hypermedia system [17]. Although providing potential support for activities before, during and after the capture session, the authors do not report on its use.

Chiu presents recent work on their system NoteLook, which allows a media-enabled conference room to support capture, annotation, indexing of the resulting documents [12]. Implemented the context of providing paper-like surfaces for reading and browsing hypertext, Xlibris allows the generation of dynamic links by capturing user interaction with the reading stand-alone device [26]. Xlibris implements a hypertext structure that allows the classification and organization of the annotations, as well as the searching for related information in the device.

Work has also been reported in the educational domain — capturing lecture experiences is desirable since students are likely to need to revisit the information as originally presented. In this domain, work includes the provision of more or less automated support to the capture of audio and video of the live sessions in order to have the presentation available for later review. This is the case of the ADEPT Project at Stanford [20] and the AutoAuditorium Project [9]. In such implementations, however, the information generated is mostly a linear video-based document.

Some implementations capture more than audio and video information associated to each session — as a way to provide a more comprehensive record of the captured experience. The Cornell Lecture Browser¹ captures images of the slides presented during the session and automatically generates table of contents that provides synchronized indexing into the slides and the video information.

The *Authoring on the Fly* system [7] provides an elaborate capture infrastructure that implements the linking by capturing paradigm — the implementation, which emphasizes the support to broadcasting the presentation live, supports capture of a fair number of applications running on Unix-only environments. Alongside with DOLPHIN and the Classroom 2000 infrastructure (see next section), this seems to be the only implementation that truly supports an underlying hypertext structure for the documents generated.

It is important to observe that, as reported, those capture-based systems generate, in general, static hypermedia

¹www.cs.cornell.edu/zeno/Projects/lecture%20browser/Default.html

documents as a result of the capturing process. Our work aims at overcoming this limitation by linking with augmenting operations.

THE CLASSROOM 2000 INFRASTRUCTURE

The Classroom 2000 project exploits ubiquitous computing technology to allow the capturing of a varied of interactions of an instructor with an instrumented environment. Given its application domain, the aim of the project is to allow students to take on a more active and enriching role in the classroom by capturing details of a live session and making them widely available. Since accurate details are made available, students are free from taking detailed notes and have more time to create their own personalized notes and actively participate in the classroom.

The Classroom 2000 software infrastructure has been used to support activities in several classrooms and meeting rooms with varied hardware equipment. The most comprehensive hardware installation is found in the original prototype classroom, depicted in Figure 1, which has been in operation since January 1997. In this room, the hardware infrastructure includes an electronic whiteboard, two ceiling-mounted projectors, one camera and several microphones embedded in the ceiling, all operating in unison through a suite of client-server software modules.

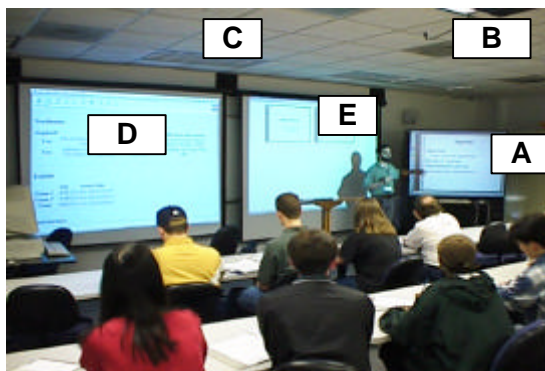


Figure 1: The Classroom 2000 classroom is instrumented to capture of the information presented on the electronic whiteboard (A), captured by a video-camera in the back of the room (B) and by microphones embedded in the ceiling (C), as well as web browsing projected (D). Information from previous slides is presented simultaneously in other projected screen (E)

The supporting software infrastructure, referred to as the Zen* system, performs the tasks of capturing and synchronizing streams of information during each live session. These tasks include controlling the capture of the media streams (audio and video) and orchestrating the generation of an associated document when the session is finished. As a result of the overall processing, a few minutes after the instructor concludes a lecture a multimedia-enhanced web hyperdocument has been automatically authored and is made available to students.

The document presented in background of Figure 2 is a

sample of the hyperdocuments automatically generated with the Classroom 2000 infrastructure. In this case, the frame on the right hand-side presents the lecture as a sequence of discrete slides as presented or created during the lecture. The frame on the left presents a timeline decorated with relevant activity as it happened during the lecture, either a new slide being visited on the electronic whiteboard or a Web page being visited on a separate display inside the classroom. The frame on the bottom left of the window presents the video (and audio) recording of the lecture, played back as a RealNetworks™ stream. The window on the foreground corresponds to a web page visited during the lecture — it was open by selecting its title used to decorate the timeline.

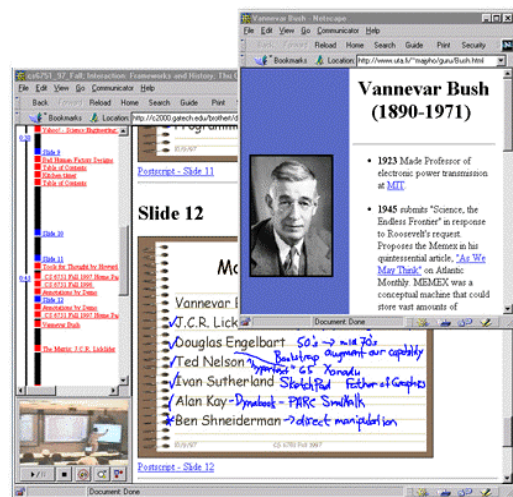


Figure 2: Web browsers are used for presenting the captured notes that link presented material from the electronic whiteboard and Web pages visited in class with streaming digital audio/video recorded during class

Underlying that hyperdocument there is a hypertext structure automatically extracted from capturing the interaction of the instructor with the instrumented environment. Therefore, a fairly elaborated hyperdocument is generated after each captured session at almost no cost. We discuss in detail this hypertext structure next.

LINKING BY INTERACTING

Given the aim of facilitating the generation of associated hypermedia documents, we advocate that capture-based systems support flexible hypertext structures generated by *linking by interacting* operations that are divided in *linking by capturing* and *linking by augmenting* operations:

- A first requirement is to have operations able to capture user-interaction within an instrumented environment. The information, as captured, must correspond to nodes, anchors and links which, in turn, can be turned into a complex hypertext structure and associated contents — this information is used to automatically generate hypermedia documents that corresponds to the live session. We call this process *linking by capturing*.

- The hypermedia document generated through this capture process is static in its very nature: the capture begins at the start of a live session and ends when it finishes. To overcome this limitation, we propose that operations must be offered to allow for the captured contents to be extended with complementary operations that support user interactivity outside the live session — which we call *linking by augmenting* operations.

We have implemented both linking by capturing and linking by augmenting operations in the context of the Classroom 2000 project, as described next.

Linking by Capturing

The Classroom 2000 environment is instrumented so that several streams of information are captured during the live session. In general terms, each stream is captured as an individual sequence of events that are time stamped. Processing each stream according to a time axis allows the automatic generation of time-based linear structures that, combined, correspond to flexible hypertext structures associated with documents containing those streams. The continuous media streams captured are audio and video streams; the discrete streams include slides and web browsing — all corresponding to nodes in our hypertext structure.

The instructor can interact with the electronic whiteboard by writing and drawing contents or marking over existing contents (quite common when prepared slides are used). The instructor can also navigate among the existing slides — forward and backward — as needed. Therefore, a single slide can be visited many times during a session and each time new annotations can be handwritten on it — or else existing annotations can be removed.

Another resource exploited by many instructors is web navigation. By capturing such navigation, material presented in the electronic whiteboard can be made complementary to that presented in web pages or vice-versa.

The *linking by capturing operations*, defined according to the usual requirements demanded by users in the underlying application domain (teachers and students), required the four node structures: *slide as a node*, *web page as a node*, *captured media stream as a node* and *timeline as a node*.

Slide as a Node. An instructor may or not use prepared slides, which are slides having background images associated with them. New slides can be inserted anytime within a sequence of prepared slides. When prepared slides are not used, the instructor writes on a blank screen and generates as many slides as needed — each slide being associated to one screen's worth of information. Once the session is finished, a *slide node* is automatically created as an image corresponding to each slide created by the instructor (images for slide nodes 10 and 11 are presented on the right frame of Figure 3).

Web Page as a Node. The instructor can use computers in the classroom to present the contents of web pages and this

facility has proven to be useful in a number of different courses. In several courses, web pages are used as the main vehicle for presenting contents — and the slides in the electronic whiteboard are used to write additional information discussed during the lecture. On the other hand, when the whiteboard is the main resource used to present contents, web pages are often used for presenting reports or projects prepared by the students or other complementary material such as readings.

Captured Media Stream as a Node. The whole lecture is captured as a single audio and/or video stream. The good quality of the audio capture in the classroom allows not only the recording of the presentation made by the instructor but also the questions asked by the students. As presented below, interactions of the instructor with the electronic whiteboard (*linking by capturing ink*) or web browsers (*linking by web browsing*) are captured and associated to links into a captured media stream node. The audio and video media stream nodes are presented as RealNetworks™ streams (Fig. 3-g).

Timeline as a Node. The live session itself is associated to a one-dimensional node structure representing the duration of the lecture, which we call a *timeline* (Fig. 3-d). This node is exploited by several of the *linking by capturing* operations.

Exploiting the node structures, five *linking by capturing operations* automatically generate flexible hypertext structure: *linking by capturing visits to a slide*, *linking by capturing ink*, *linking by capturing web browsing* and *linking by indexing into a media stream*.

Linking by Capturing Visits to a Slide. During a live session, each time the instructor presents a slide long enough a visit is registered to its slide node. In the document shown in Figure 3, each slide node is presented once on the frame on the right — according to the sequence established by the instant the slide node was first visited. The *linking by capturing visits to a slide* operation creates links that are embedded as anchors placed on the top of each slide node and on their relative offset to the timeline node. The links on the top of the slide node index into the appropriate offset in the continuous media stream node (Fig. 3-a). The links associated to the timeline index into the corresponding slide nodes in the frame in the right (Fig. 3-e).

Linking by Capturing Ink. In each slide node, this operation creates links that are associated with the writings and markings (i.e., all ink marks) made by the instructor. Each link points into the audio and video streams — at the offset corresponding to when the content was written (Fig. 3-b,c). A clustering algorithm is applied so that ink strokes that are close to each other correspond to a single link [2].

Linking by Capturing Web Browsing. As the instructor navigates through the Web during the lecture, the address of each page presented is captured. These addresses are used to create anchors that, containing the title of the page, link into page itself. In the interface shown in Figure 3, these anchors are positioned in the appropriated offset in the timeline node

as when the page was visited (Fig. 3-f) — and may cause the page to be presented in a new window (as in Figure 2).

Linking by Indexing into a Media Stream. In order to provide dedicated linking into media streams, the timeline node as presented in the left frame of Figure 3 is built as a sequence of anchors that index into the audio or video streams in 30 second intervals (Fig. 3-d). Therefore, in such interface clicking on any point of the timeline will launch the audio or video at that point in the lecture (Fig. 3-g).

Linking by Augmenting

Two types of augmenting operations have been implemented. The first allows the information captured during the live experience to be interactively augmented. The second allows the users interactively to integrate information produced outside the captured session into the body of hypermedia information. It is important to observe that the *linking by augmenting* services can be implemented as interactive operations because the underlying hypertext structured is stored independently of the contents of the captured streams. Three operations have been implemented: *augmenting slide nodes with textual information*, *augmenting slide nodes with annotations*, and *augmenting a lecture with material produced outside the live session*.

Augmenting Slide Nodes with Textual Information. Much of the information captured during the classroom experience can be enhanced in order to provide a richer material to students and to instructors. When prepared slides are used, a tool provides a service that uploads the prepared slides into

the database. During this process, the same tool also extracts and stores the textual content of the slides in the database. As a result, the textual information within each slide — which have been interactively authored when the slide was first created — is automatically associated with the node. As far as the hypertext structure is concerned, a *new textual node* is created to contain the textual information, and a bi-directional link is established between this *textual node* and the corresponding slide node.

The content of a textual node is useful for a number of services. First, the title of the slide can be used in the presentation interface — in the example in Figure 3-e the title is used to label the timeline node at the times of the visits to each slide. Second, it can be used in search services. Third, it can be used in automatic linking services.

A complementary service ought to be offered to allow a textual version of the information written during the live session to be associated with each slide node. Current handwriting recognizers do not perform well when handwritten information is combined with gestures and drawings. For this reason, we provide an editing interface (Fig. 4) that allows the instructor to interactively provide the transcription of the contents of handwriting associated with the slide nodes that is stored in a *handwriting node*. From an authoring hypertext perspective, when providing this textual information the instructor is creating both the node to hold the information and the bi-directional linking between this handwriting node and its associated slide node.

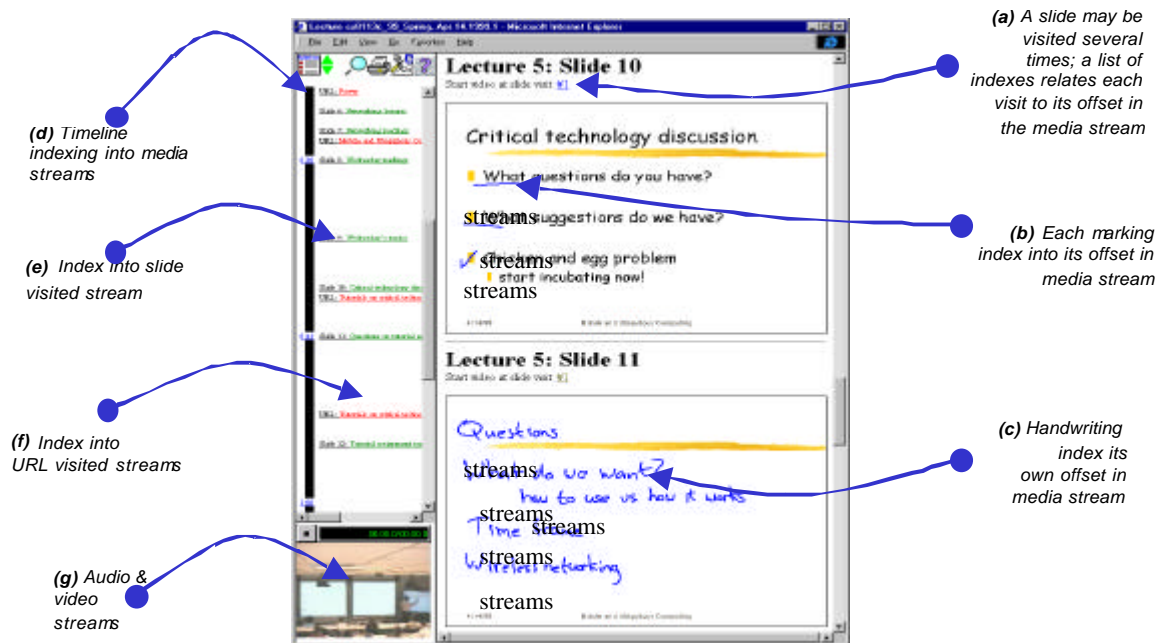


Figure 3: Intra-session linking by capturing classroom activity — handwriting, slide transitions and web browsing offer synchronization information for indexing into the audio and video streams

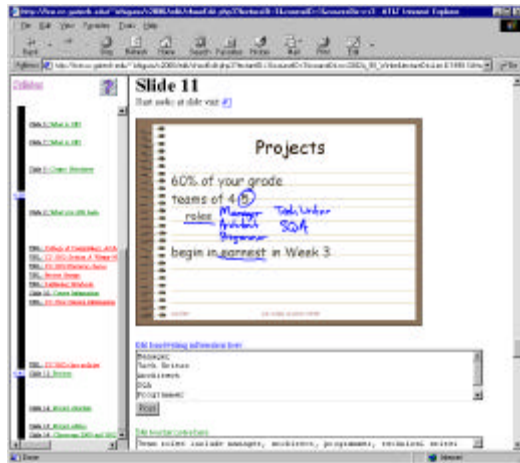


Figure 4: The instructor can interactively provide the translation of the handwriting

Augmenting a Slide with Annotations. For a number of reasons, instructors may want to add extra content to the slide after the live session. We provide another editing interface that allows the instructor to interactively attach comments to any particular slide. This information can also be used for automatic searching and linking services — we illustrate in Figure 5 a presentation interface where the comments added by the instructor are shown. Again from the authoring hypertext perspective, the instructor is creating both a node to hold the comment and a bi-directional link between this node and its corresponding slide node.

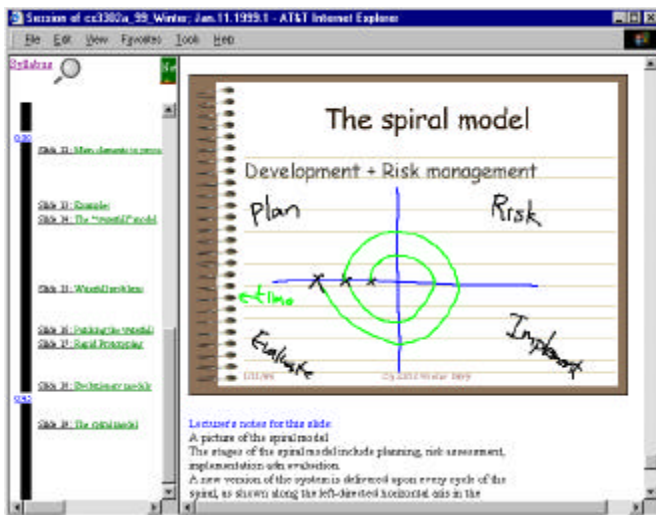


Figure 5: Presentation of an Annotation added by the instructor after the live session

The provision of such service ultimately allows the instructor to turn the overall material in a dynamic hypermedia document, since annotations can be added and removed as necessary. However, it is also desirable that material created prior to the live session, as a preparation to the session itself, be integrated into the hyperdocument. Moreover, material produced as a follow up of a session

should also be made part of the hypermedia material associated to it. Such services cannot be appropriately provided by textual annotations, and we present our solution next.

Augmenting a Lecture with Material Produced Outside the Live Session. We aim to allow users interactively to integrate information produced outside the captured session with the body of hypermedia information. In order to accomplish this, we provide interfaces that allow any user to interactively associate single slides or full lectures to web pages managed by a specialized service called a CoWeb.

The CoWeb is an informal and unstructured web repository which allows collaborative authoring of web-based material. This service has been used both in educational applications and traditional computer-supported collaborative work [16].

Any structure can be built by using the capability of creating internal and external links provided. In the CoWeb, new pages can be created and linked by anyone, and permission is granted for anyone to edit and create links in already existing pages. An especially powerful feature is that the CoWeb is persistent. Pages exist over time, so that new additions or edits can be easily identified. This simple infrastructure has supported a wide variety of collaborative activities, from group writing to telementoring.

Our goal was to exploit the CoWeb infrastructure so that individual or group contributions could be integrated with the hypermedia document associated with one captured session. Moreover, integration should be done by the user when interacting with the hypermedia document.

Our current approach for achieving such integration is as follows. When browsing through the notes corresponding to a lecture, a user can create links to new or ongoing discussion pages in the CoWeb. These links can be created via a simple form interface, as shown in Figure 6 — the corresponding anchor is created in the hypermedia document and its link is created as a result of a request to the CoWeb service. When the CoWeb service receives the request, it creates a new discussion page if necessary (in the case of a new discussion topic). Next, it adds to the discussion page a thumbnail image of the slide that originated the link, so as to provide an anchor back to the captured lecture, as illustrated at the bottom of Figure 7.

EXPLOITING THE APPROACH

We have implemented the *linking by interacting* operations described above to support the generation of flexible hypertext structures. In order to exploit the resulting structure to demonstrate the appropriateness of our approach, we discuss the generation of alternative views of the associated hypermedia document.

Each of the interfaces presented in Figures 2 to 6 above correspond to possible views of the hyperdocument produced as a result of the *linking by interacting* operations. All these views are automatically generated on the fly, and are made available for the instructor or the students

according to their preferences. As a course evolves, the quantity of captured information also grows and its structure may become quite complex. This rich hypertext structure can be exploited to present richer, alternative services and associated views, such as those presented next.

Course views. In order to provide support at course level, the hypermedia information for each course includes general information with respect to when and where the live session occurs. This allows both the automatic inference of when the individual sessions are due to occur and the automatic updating of overall information as each session is concluded. As a result, a collection of captured lectures for a single course is presented as an automatically generated syllabus organized by lecture dates, such as that illustrated on the left of Figure 8.

Each syllabus allows the student to access a given lecture with either audio or video augmentation and in a way that can accommodate either a fast or slow network connection — as well as other interface alternatives. In the example on the left of Figure 8, the options are selected as radio buttons presented on top of the list of lectures.

Search service & views. We provide a service to search the contents of the hyperdocuments for the specific terms in all the information available for the course — which includes the nodes containing the instructor’s handwriting and annotations, the content of prepared slides, the title and content of Web pages visited and of voice transcripts when these are available. Figure 9 illustrates a first level view of search results, showing a summary of results over a single course — thumbnails of slides matching the query are anchors that link to their context in the lecture. These results are divided by lecture and further details can be presented in other views.

EXPERIENCE WITH LINKING BY INTERACTING

Classroom 2000¹, supporting most of the *linking by capturing* operations described, has been an ongoing research effort for over 3 years with over 100 classes taught by more than 24 instructors. The extension of the environment to support *the linking by augmenting* approach has been in experimental use for one year. As a part of our evaluation of the system we distribute a questionnaire at the end of each course taught. To date we have data from nearly 700 students ranging from undergraduate to graduate in courses from Math, Engineering, and Computer Science. Overall, we find that students typically depend on the notes provided, and while they don't feel like it's a substitute for attending class, they are relieved to have the system in place on the occasions when a class must be missed.

Surveyed, most instructors admitted that effective use of the capture infrastructure incurs a cost for the instructor, including the time needed to prepare Web links to present during class (a similar task to building Walden's Paths [13]). However, they also indicated that they would continue to use the system because they feel that it benefits their

students. A number of instructors have indicated that they view previously presented lectures by themselves or other teacher when preparing similar lectures.

We have discussed elsewhere features associated to the automatic capture and processing of the underlying multimedia information [2] [11] and with the instrumentation of the environment to make it easy to use [10]. We review the lessons learned from implementing and supporting such an infrastructure over the period of three years in [1].

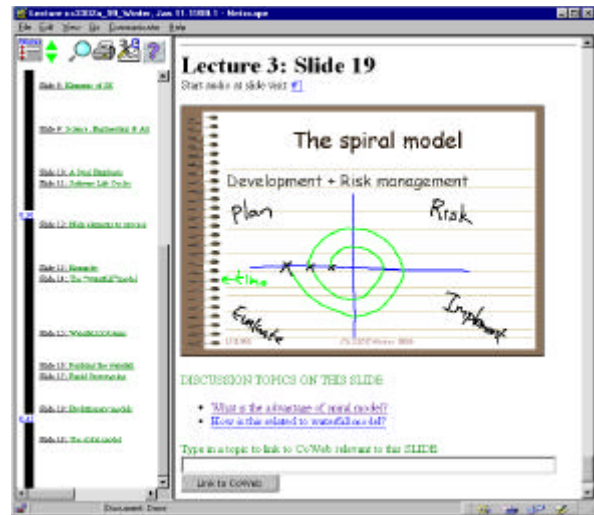


Figure 6: Interface for creating a link to the CoWeb



Figure 7: Page in the CoWeb linked with a slide through a thumbnail image in the hypermedia document of a lecture

¹ see www.cc.gatech.edu/fce/c2000



Figure 8: A syllabus can be generated on the fly to provide view of the collection of captured lectures for a course

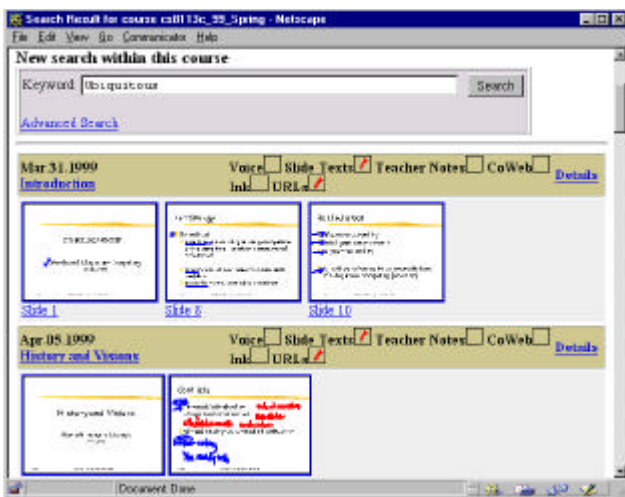


Figure 9: Results from search service provided

Linking by Capturing: a sample

In order to illustrate the volume of hypertext structure and contents that can be created by the use of the linking by capturing operations, a summary of the use of the *linking by capturing operations* in the Fall 1999 semester at Georgia Tech is as follows:

- The system supported 13 courses and meetings involving 10 different instructors, two instrumented classrooms and one instrumented meeting room.
- Eight courses captured slide streams only, two courses captured slide and audio streams and another 5 courses captured slides, audio and video streams.
- The courses had, in average, 27 lectures and 278 visits to slides. Five of the courses used prepared slides.

- Web pages were visited in 11 courses, with an average of 130 Web pages per course.
- An average of over 13000 ink links were created per course — an average of 490 per lecture.

Linking by Interacting: an experiment

The *linking by augmenting* operations were defined to support the generation of flexible and dynamic hypertext structures from the structure provided by the *linking by capturing operations*. We report the results achieved with a graduate seminar course where all the *linking by interacting* operations were available. This computer science course, taught by one of the authors and another co-instructor, had 31 graduate and senior undergraduate students. The course had 90-minute lectures that used the Classroom 2000 infrastructure twice a week. In total, there were 18 lectures over a 10-week period.

How the infrastructure was used. The course consisted of a mix of traditional lecture and discussions based on 69 outside readings — which were listed off a separate web page from the course Home Page and were mostly available on-line for reading.

Use of the CoWeb was required in the class: it was the medium used for instructor and students to deliver essays, summaries and projects. For each reading assigned in class, one or more students were assigned responsibility for preparing a summary of the paper and posting it to the CoWeb prior to the lecture. Students and instructors were free to read and comment upon the reading summary. The papers were usually discussed in class and students were then responsible for updating their summary based on lecture discussion.

Student groups were responsible for two external projects in

the class, and the project reports were to be posted to the CoWeb as well. Occasionally, projects were presented in class and all were commented upon by the instructors. Students were encouraged to comment on projects of other groups, but little commentary was provided via the CoWeb.

Resulting hypermedia information. The use of the instrumented classroom through the *linking by capturing* operations generated 27 hours of audio and video information, associated to 307 slides and 216 references to visits to web pages — and average of 17 slides and 12 web visits per lecture. Seven of the lectures had a prepared lecture and the remaining 11 ones used a blank whiteboard. Most slides contain handwriting or markings that were turned into anchors linking into the media streams — an average of 1264 ink links per lecture.

After each lecture one instructor, using *linking by augmenting* operations, manually transcribed the handwriting on the electronic whiteboard (205 of 317 slides), added titles to slides that did not have one already (101 of 317 slides) and provided additional commentary on some slides (27 of 317). Each of those were turned into nodes and linked to the appropriate place in the underlying hypertext structure for the course — the interaction with the operations provided again have hidden the instructor from this authoring process.

Use of the CoWeb was required in the course, so a lot of activity was created in its discussion space, totaling 303 independent pages created by students and instructors. Using the *linking by augmenting* operation that insert links in the captured lecture hypertext structure, 66 links were established between captured lectures and the CoWeb space, for an average of 3.6 links per lecture. That is not as high a value as it had been expected, but not surprising for such a novelty. An elaborate discussion on the use of the environment as an educational tool is found in [3].

CONCLUSION

We have presented a novel approach where linking by interacting operations lead the automatic generation of extensible hypertext structures. When compared with related work, the novelty of the work reported here is in:

- The amount of automation supported by the linking by interacting operations in general, and the link by capturing operations in particular.
- The integrated support to activities performed inside and outside the capture sessions.
- The overall approach of exploiting user interaction as a seamless approach to generate flexible hypertext documents.

Our preliminary experience with supporting the extensible hypertext structure shows that there are very interesting technical questions to be addressed in providing effective support to the authoring of this dynamic repository of hypermedia documents. We will continue this type of

experimentation in the near future. We have started experimenting with allowing the generation of automatic links between the material corresponding to the material generated during and outside the captured experience. A major goal, however, is to provide extended capabilities for reusing captured material.

We are currently working on an important addition to our current infrastructure, corresponding to the support of both synchronous and asynchronous collaborative tools — which brings up new interesting research issues. In terms of support to synchronous collaborative work, we have done some initial experimentation with integrating information captured from devices used by students during the lecture [34] — we plan to extend this experience to information created outside the instrumented environment.

ACKNOWLEDGMENTS

Gregory Abowd's work on Classroom 2000 is supported by grants from the National Science Foundation (IRI-9703384, EIA-9806822 and CISE-9818305) and from equipment grants from Sun Microsystems and Hewlett-Packard. Maria da Graça Pimentel is supported by FAPESP-Brazil, grant PD-98/05984-7.

REFERENCES

1. Abowd, G.D. (1999) Classroom 2000: An experiment with the instrumentation of a living educational environment. IBM Systems Journal, 38:4, pp. 508-530.
2. Abowd, G.D., Atkeson, C.G., Brotherton, J.A., Enqvist, T., Gully, P.A., & Lemon, J. (1998) Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. Proc. CHI'98, ACM, pp. 440-447, 1998.
3. Abowd, G.D., M. Pimentel, B. Kerimbaev, Y. Ishiguro, and M. Guzdial. (1999) Anchoring Discussions in Lecture: An Approach to Collaboratively Extending Classroom Digital Media. Proceedings of Computer Supported Collaborative Learning (CSCL'99).
4. Abowd, G.D. and Mynatt, E.M. (1999) Ubiquitous Computing: Past, present and future. ACM Trans. CHI. Special issue on HCI in the New Millenium (to appear).
5. Allan, J. (1996) Automatic Hypertext Link Typing. Proc Hypertext'96, pg 42-52.
6. Auffret, G.; Carrive, J; Olivier C.; Dechilly, T., Ronfard, R.; Bachimont, B. (1999) Audiovisual-based hypermedia authoring: using structured representations for efficient access to AV documents. Proc. ACM Hypertext '99, pg 169-178
7. Bacher, C; Müller. R. (1998) Generalized Replay of Multi-Streamed Authored Documents, Proceedings of ED-Media '98, Freiburg, June 1998
8. Barger, D., A. Gupta, E. Sanocki and J. Grudin. (1998) Annotations for Streaming Video on the Web:

- System Design and Usage Studies. Microsoft Research Technical Report MSR-TR-98-60, Redmond, WA.
9. Bianchi, M.H. (1998) AutoAuditorium: A Fully Automatic, Multi-Camera System to Televisе Auditorium Presentations. 1998 Joint DARPA/NIST Smart Spaces Technology Workshop, 30-31 July 1998, NIST, Gaithersburg, MD.
 10. Brotherton, J.A. & Abowd, G.D. (1998) Rooms Take Note: Room Takes Notes! In Working Papers of AAAI '98 Spring Symposium, March, 1998.
 11. Brotherton, J.A, Bhalodia, J. & Abowd, G.D. (1998) Automated Capture, Integration, and Visualization of Multiple Media Streams. Proc. IEEE Conference on Multimedia and Computing Systems, July, pp. 54–63.
 12. Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L. (1999) Meeting Capture in a Media Enriched Conference Room. Proceedings of. CoBuild '99.
 13. Furuta, R.; Shipman, F.M.; Marshall, C.C; Brenner, D.; Hsieh, H. (1997) Hypertext paths and the World-Wide Web: experiences with Walden's Paths. Proc. ACM Hypertext'97, pg 167-176.
 14. Garzotto, F; Mainetti, L.; Paolini, P (1996) Information Reuse in Hypermedia Applications. Proc. ACM Hypertext'96, pg 93-104.
 15. Guinan, C; Smeaton, A.F. (1992) Information retrieval from hypertext using dynamically planned guided tours. Proc. ACM Hypertext'92, pg 122-130.
 16. Guzdial, M. (1999). Supporting Learners as Users. The Journal of Computer Documentation, 23(2), 3-13.
 17. Haake, J.M; Neuwirth, C. M.; Streitz, N.A . (1994) Coexistence and Transformation of Informal and Formal Structures: Requirements for More Flexible Hypermedia Systems. Proc. ECTH'94, pg 1-12.
 18. Halasz, F.G. (1988) Reflections on Notecards: seven issues for the next generation of hypermedia systems. Communications of the ACM 31:7, pg 836-852.
 19. Hardman, L; van G. Rossum; Jansen, J; Mullender, S. (1994) CMIFed: a transportable hypermediaauthoring system. Proc ACM Multimedia, pg 471-472.
 20. Harris, D& DiPaolo, A. (1996) Advancing Asynchronous Distance Education. IEEE Trans. Education. August 1996.
 21. Marshall, C.C. (1998) Toward an ecology of hypertext annotation. Proc. ACM Hypertext'98, pg 40-49.
 22. Marshall, C.C; Shipman, F.M. (1997) Spatial Hypertext and the practice of information triage. Proc. ACM Hypertext'97, pg 124-133.
 23. Marshall, C.C; Shipman, F.M. (1995) Spatial Hypertext: Designing for Change. Communications of the ACM 38:8, pg 88-97.
 24. Minneman, S., S. Harrison, B. Janseen, G. Kurtenback, T. Moran, I. Smith, and B. van Melle (1995). A Confederation of Tools for Capturing and Accessing Collaborative Activity. ACM Multimedia' 95.
 25. Moran, T. P., L. Palen, S. Harrison, P. Chiu, D. Kimber, S. Minneman, W. van Melle, and P. Zelweger (1997). Salvaging Multimedia Meeting Records. Proc. CHI'97.
 26. Price, M.N.; Golovchinsky, G.; Schilit, B.N. (1998) Linking by inking: trailblazing in a paper-like hypertext. Proc Hypertext'98, pg 30-39.
 27. Robertson; J. Merkus, E.; Ginige (1994) A. The Hypermedia Authoring Research Toolkit (HART). Proc. ECTH'94, pg 177-185.
 28. Shahraray B. & Gibbon D.C. Automated authoring of hypermedia documents of video programs. Proc. ACM Multimedia '95, pg 401-409, 1995.
 29. Shipman, F.M.; Furuta, R.; Brenner, D.; Chung, C.; Hsieh, H. (1997) Using paths in the Classroom: experiences and adaptations. Proc. ACM Hypertext'98, pg 267-276.
 30. Shipman, F.M.; Furuta, R.; Marshall, C.C (1997) Generating Web-based presentations in spatial hypertext. Proc. International Conference on Intelligent User Interfaces , 1997, pg 71-78.
 31. Shipman, F.M.; Marshall, C.C; LeMere, M. (1999) Beyond location: hypertext workspaces and non-linear views; Proc. Hypertext'99. pg 121-130
 32. Streitz, N.A T.; Geibler, J.; Haake, J.M.; Hol, J. (1994) DOLPHIN: Integrated Meeting Support across Local and Remote Desktop Environments and LiveBoards. In Proceedings of CSCW'94, pg 345-358.
 33. Trigg, R.H. (1999) Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment. ACM Trans. Inf. Systems 06:04, pg 398-414.
 34. Truong, K.N., Abowd, G.D., and Brotherton, J.A. (1999) Personalizing the Capture of Public Experiences. Proc. User Interface and Software Technology Symposium (UIST'99). pg 121-130.
 35. Zellweger, P. (1989) Scripted Documents: A Hypermedia Path Mechanism, Proc. Hypertext'89, pp. 1-14.