UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

ISSN 0103-2569

Interaction Alternatives for Linking Everyday Presentations

Alessandra Alaniz Macedo
José Antonio Camacho-Guerrero
Renan G. Cattelan
Valter R. Inácio Jr.
Maria da Graça Campos Pimentel

Nº 230

RELATÓRIOS TÉCNICOS

São Carlos – SP
Abr./2004

Interaction Alternatives for Linking Everyday Presentations

Alessandra Alaniz Macedo, José Antonio Camacho-Guerrero
Renan G. Cattelan, Valter R. Inácio Jr.
Maria da Graça Campos Pimentel

[1]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - São Carlos/SP - Brazil

{ale,jcamacho,renan,inacio,mgp}@icmc.usp.br

Abstract. *Live presentations such as meetings or lectures can be captured in instrumented environments towards producing hyperdocuments corresponding to the information presented in the live session. However, a given presentation is usually related to other captured sessions as well as to other documents available to the overall group. Therefore, there is a demand for operations that allow the linking of the related documents both before and after a live presentation takes place. Moreover, the availability of such operations during the presentation provides a unique opportunity for the identification of links in the live context itself.*

On the one hand, it is interesting that the creation of links occur automatically given the amount and diversity of information that can be related as the number of presentations grows. On the other hand, the evaluation of the possibilities of links identified by linking algorithms can be a burden when the user does not have the focus of attention on the associated task.

We propose the that well-known operations — searching and recommending — be used to identify links in a growing repository of documents associated with the presentations when the user has the focus of attention on the underling contents. This means that the operations should be made available before, after and during the presentation — and in any case the user should be able to decide when any relevant results should be attached as annotations to the repository of associated documents. In this paper, we present the model and associated implementation that we have built to attend those requirements. The activation of the operations can be done automatically by voice and handwriting on pen-based input surfaces — desired input methods during the live presentation — as well as by traditional typing and Web browsing — desired input methods before and after a live session.

Keywords: Authoring, Linking, Capturing, Presenting, Augmenting, Information Retrieval.

Contents

List of Figures

1. Introduction

A problem faced by people interacting with computational technology is that they must adapt to those technologies. The demand for more natural user computer interaction motivated the development of a research area known as ubiquitous computing [Weiser, 1991]. Ubiquitous computing proposes the seamless integration of hardware and software technologies into the environment in order to aid humans in their everyday activities without changing the way they usually perform those activities.

One of the research themes in ubiquitous computing is that of capture and access applications, which exploit the fact that everyday experiences are rich in information that may be captured and made available as hyperdocuments that can be reviewed by users at a later time [Abowd et al., 2000] [Pimentel et al., 2000]. Capture and access applications have been built to meeting (e.g. TeamSpace [Richter et al., 2001] and NoteLook [Chiu et al., 2000]) and educational settings (e.g. Classroom 2000 [Abowd, 1999] and Cornell Lecture Browser [Mukhopadhya and Smith, 1999]).

Arguing that one of the most important features in capture and access applications is the degree of automation provided, Abowd et al. suggest to support the underlying activities in four phases as in the movie industry [Abowd et al., 1996]: (i) *pre-production*: preparation of a session; (ii) *live capture*: capture of relevant data streams; (iii) *post-production*: integration of the captured data into hyperdocuments; (iv) *access*: presentation of the hyperdocuments to users. The pre-production, the live capture and the access phases take place respectively *before*, *during* and *after* a live presentation. The post-production phase also occurs *after* a live presentation, being is a phase where additional processing can be performed.

Pimentel et al. have observed that, since the information presented in live sessions is usually related to that captured in previous sessions, as well as to other documents, there is a demand for operations that allow both manual and automatic linking of related documents both *before* and *after* a live presentation takes place [Pimentel et al., 2001].

In this paper we argue that automatic linking services should be available as authoring aids *before*, *during* and *after* live sessions since in all those moments users have the focus of attention on the underlying information. We propose that interaction alternatives based on searching and recommendation operations be offered to users towards identifying links in those complementary moments. We present our implementation that was carried out by extending our previous results on building Web-based linking [Macedo et al., 2002] and recommendation [Macedo et al., 2003]services.

Considering the ubiquitous computing domain, different interaction alternatives should be offered to users depending on the interface style exploited for capturing information. In this paper, we present the model and associated implementation where the activation of the linking operations can be done automatically by voice and handwriting recognition, and demonstrate their use during a live presentation, as well as by traditional typing and Web browsing, which we demonstrate in use before and after a live session.

The remaining sections of this paper are organized as follows: we first present previous works that have supported our current proposal; we then describe our linking operations; next, we discuss related work; last we summarize the benefits of our approach and comment on future work.

2. Background: Starting points

In order to investigate the creation of hyperlinks using documents that represent live experiences captured by means of ubiquitous applications, we have exploited and extended previous work that we summarize in this section.

2.1. INCA: Network Abstractions

Capture and access applications are usually composed by a confederation of devices and software systems, presenting a inherent distributed nature. Truong & Abowd developed an INfrastructure for Capture and Access Applications (INCA) in order to provide network abstractions demanded in the implementation of those applications [Truong and Abowd, 2004].

INCA provides architectural support for building applications by means of a set of key design concerns including *Capture* (part of the system is responsible for the capture of information as streams of data) and *Access* (part of the system provides access to multiple integrated streams of information that are gathered as a response to context-based queries). For each design concern, INCA implements a module and an interface that developers must instantiate and implement when building applications. The basic entities are the capture and access modules, which behave like producers and consumers in a publish-subscribe approach. In order to automatically control the presence of capture and access modules, INCA provides a *Registry* that manages publishing and subscribing operations done by application modules.

2.2. xINCA: Functionality Abstractions

Besides architectural aspects, a large number of capture and access applications present recurrent functionalities. A typical example occurs in the educational domain: most of the applications developed for this domain needs software to control interactions in a whiteboard. Other recurrent functionalities are also present in capture and access applications: examples include the capturing of audio and video streams as well as the browsing history of the user (weblog). These functionalities can appear in any combination and suggest a component-based approach. With this in mind, we developed an eXtended INfrastructure for Capture and Access Applications (xINCA toolkit [Cattelan et al., 2003]) as a layer on top of INCA.

xINCA provides software components that captures user-interactions with electronic whiteboards, text generated during chat sessions, URLs visited during Web browsing as well as streams of audio and video recorded during live capture sessions. xINCA components implement instances of INCA modules (capture modules, access modules or both capture and access modules) and are registered at runtime in a INCA *Registry* entity. Figure 1 depicts the communication flow between components in two simultaneous sessions (Session #1 and Session #2) that are used by three applications (App #1, App #2 e App #3). In this example, App #3 only receives (accesses) the strokes and images presented (captured) in the whiteboard component of App #1. Also, App #2 access the information produced by a user typing in the chat component of App #1 at the same

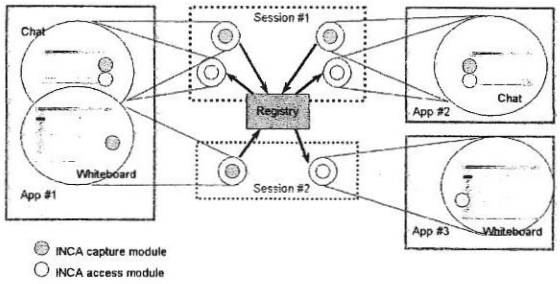

Figure 1: Communication flow between xINCA components in two simultaneous sessions

time that captures the information typed in its own chat component and sends it to the chat component of App #1.

2.3. LinkDigger Service

Macedo et al. investigated mechanisms allowing the automatic identification of relationships in textual homogenous Web-based repositories by using Latent Semantic Indexing (LSI), which is normally used to overcome problems related to the use of lexical based analysis [Furnas et al., 1988]. The similarities defined by LSI are based on closeness of terms in a semantic space built according to co-occurrence of all terms in collection of documents manipulated instead of matching. Their LinkDigger [Macedo et al., 2002] was then extended to implement a recommendation service [Macedo et al., 2003].

Next, we show LinkDigger being used to relate information extracted from all phases of a captured and access application in a system built with INCA and xINCA.

3. Linking Live Presentations

Figure 2 presents an architecture comprising a capture and access application for which we provide linking supporting tools before, during and after a live presentation.

The information captured in the live capture phase is integrated to that provided in the pre-production and access phases and stored in an XML database. The pre-production phase is supported by AutorE [Pimentel et al., 2003], a tool that allows the manual preparation of sessions (by allowing the creation of sessions and the association of meta-data and prepared slides, for instance). The same AutorE tool supports, in the access phase, a user manually extending the captured information with annotations and links. The live-capture phase is supported by iClass [1], a live session capture system build to record several pieces of information produced during a session, including strokes handwritten on an electronic whiteboard on top of prepared slides, audio or video streams captured from a microphone or camera, as well as the URLs from Web pages visited during the session.

The architecture shows the participation of the LinkDigger linking service in all phases: we have extended the original iClass implementation by integrating LinkDigger, which allowed us to provide searching, linking and recommendation operations in all phases — the resulting links being stored in WLS, an OHS linking service [Bulcão Neto et al., 2002].

[1] iClass was inspired in the Classroom 2000 system [Abowd, 1999].

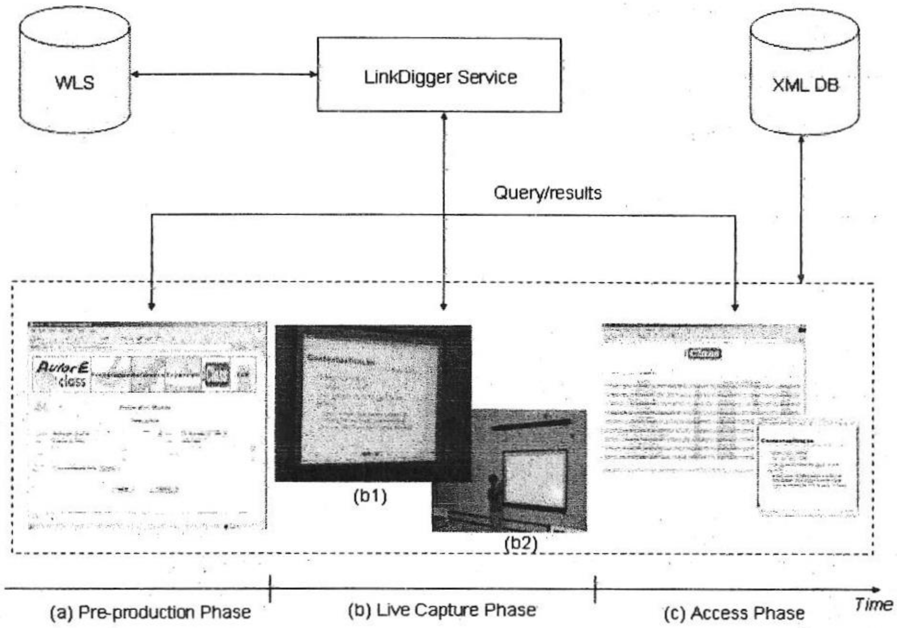

Figure 2: A linking service integrated in all phases of a capture and access application.

Given the requirements for ubiquitous applications, our implementation exploits voice and handwriting recognition, as well as the transparent extraction of the original text from prepared slides, to activate the identification of links among hyperdocuments produced in all phases. However, considering that in many cases users will be using traditional desktop computers, we also support linking in the pre-production and access phases. Moreover, considering that when user the system in the various phases the the user has the focus of attention on the underlying information, we support that the user decide whether any relevant links suggested by the service should be attached the associated repository.

3.1. Linking processing in the pre-production phase

The pre-production phase is related to the session setup. Different applications deal with different contexts and thus manage the preparation of material to be exploited in a live presentation. Our architecture allows the management of such high level of abstraction through an authoring system, called AutorE.

The AutorE system implements functions to support the process of authoring multimedia information in terms of preparation, reuse, extension and reference [Pimentel et al., 2003]. This system makes these functions available to capture systems. For each function, AutorE has a specific module. So far, only the preparation module was integrated to the linking service. By using the preparation module, presenters can prepare a new session for capture by reusing material from other sessions or by adding brand new material in the forms of metadata or prepared slides. In Figure 3, the top Web interface is a typical interface of the preparation module.

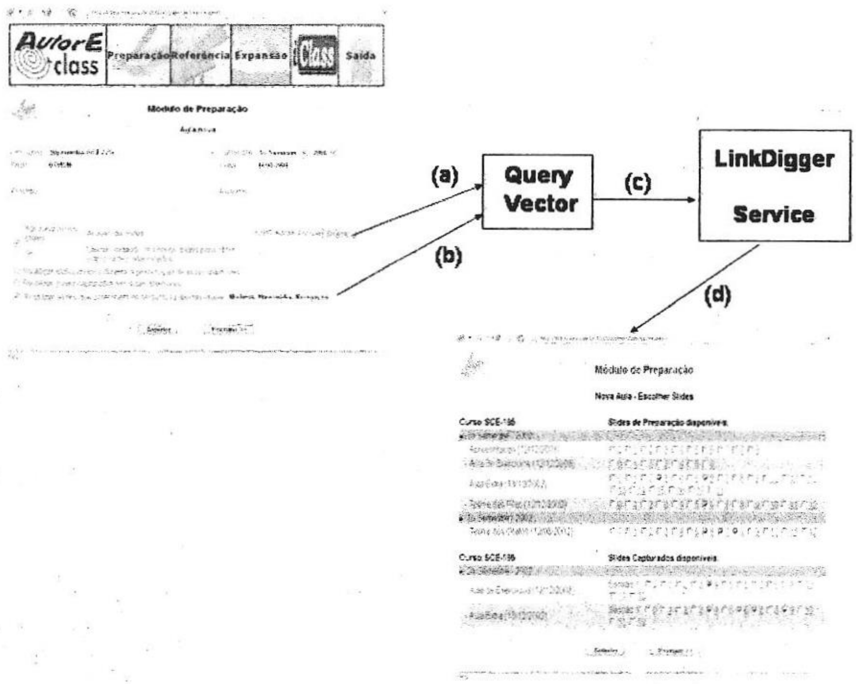

Figure 3: Linking processing in the pre-production phase allows reuse of information when authors prepare new sessions. (a) A query vector is created with text automatically extracted from slides and metadata from the session (its title, for instance). (b) The query vector can be expanded by keywords provided by users. (c) The query vector is sent to the linking service. (d) The resulting links are shown as recommendations in the preparation interface.

According to our proposal, the preparation module was extended to queries that are activated automatically by LinkDigger. This means that, when preparing a session, users may be presented automatically with suggestions of related material presented in other sessions. This is achieved as follows: as the user inputs metadata or prepared slides to the session, the corresponding text (typed by the user in forms for adding metadata or extracted automatically from the slides) are used by LinkDigger to query the existing repository — the result being a list of pages that correspond to other sessions. The suggestions are presented in the lower part of the original AutorE interface presented in Figure 3. It is important to observe that such infrastructure can also exploited by a user who can explicitly query the service for related information.

The query vector that is automatically sent to the LinkDigger Service (Figure 3(a)) can be extended by users (Figure 3(b)). Once the query vector is generated, it is sent to the LinkDigger Service (Figure 3(c)). This vector is processed by the linking service, which identifies documents semantically related to the vector. Results from the linking processing are sent to the interface (Figure 3(d)), which allows the user to selected which references should be used in the presentation being prepared.

It is important to observe that any other authoring application can be used in the pre-production phase since it sent a query vector to the LinkDigger API.

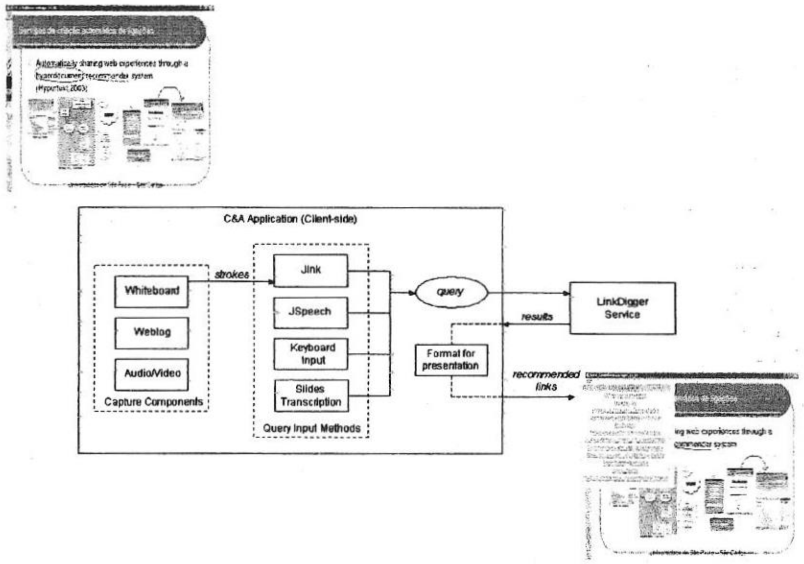

Figure 4: Linking processing in the live capture phase. Strokes captured in the whiteboard component are used to form a query vector along with text automatic extracted from prepared slides. The activation of the query is done via a voice command. The resulting recommendations are shown in a popup window.

3.2. Linking processing in the live capture phase

The live capture phase presented in Figure 2 shows a live session in progress with support of the iClass system: the prepared slides are presented on and electronic whiteboard via a Java applet (Figure 2(b1)); a presenter may write on top of the slides and may wear a microphone to capture an audio stream for the whole session (Figure 2(b2)) — all functionalities being provided by capture components (whiteboard, video, audio, chat, weblog, etc) from the xINCA toolkit [Cattelan et al., 2003].

As far as the automatic generation of links during the live capture phase is concerned, the LinkDigger service automatically identify relationships among the information manipulated in the live presentations and that previously stored in the XML database from previous sessions according to the process indicated in Figure 4. First, the capture application (iClass) composes a vector query with information from some xINCA components and, next, sends the vector to the linking service. Texts from the slides presented during the session is used to transparently built the query vector. However, considering such ubiquitous computing environment, it is important that information from the handwritten annotations as well as from the capture audio stream may be provided as interaction alternatives to built the query vector.

In order to recognize on-line handwritten characters, we developed the jInk API according to the method proposed by Chan & Yeung [Chan and Yeung, 1998]. By this method, the recognition process is divided into three stages: structure extraction and reconstruction, elastic structural matching and postprocessing. In the first stage, the coordinates obtained by the application that carries out the capture of strokes are converted in a set of segments and their directions. These segments are then manipulated by a variety of operations, such as smoothing and normalization, in order to reduce the effect of noise and small deviations in the stroke. After this, the primitive elements that compose the structure of the stroke such as lines, curves and loops, are identified. In this same stage it is performed the reconstruction, where these primitives are combined to eliminate broken strokes. At the next stage, it is performed the elastic structural matching, where the primitives obtained are compared with a pre-defined set of character models. If the primitives doesn't match with none of the models, then successive transformations are applied in the structure of the stroke, until it is recognized or rejected by the system. In the third and last stage, additional processing is performed over the recognized strokes in order to remove ambiguities and increase the accuracy of the recognizer. In an evaluation carried out by Chang & Yeung using multiple users, the recognition rates for digits, lowercase letters and uppercase letters were respectively 98.6%, 98.5% and 97.4%.

Our implementation, jInk, provides a graphical interface to be used by users interested in adjusting the character models to their own writing styles in order to obtain a more accurate recognizer. The jInk API was incorporated into the iClass system so that, besides digits and letters, jInk can also recognize symbols and gestures; gestures can be used to activate some operations in the application (not exploited in the current version).

In an another step towards exploiting the ubiquitous computing environment supported by iClass, we wanted to allow users to use voice commands in order to activate the linking processing. The voice recognition is carried out by the ViaVoice™ software, exploiting a implementation of the Java Speech API (JSAPI) [Sun Microsystems, 1998] developed by IBM. JSAPI specification defines an access interface to the functionalities of recognition and voice synthesis provided by engines like ViaVoice. This interface allows that Java developers utilize such functionalities without knowledge about recognition methods and their implementation since the processing is carried out by the engine.

Therefore, a user can use a voice command to activate the composition of a query vector with information extracted by xINCA components. Issued the command, the query is sent to the LinkDigger service (this can also be activated manually via a button on the interface). As a result of the LinkDigger processing, hyperlinks are defined on-the-fly and presented as recommendations in a small pop-up window, as shown in Figure 4.

After generating the hyperlinks, LinkDigger provides a button in the interface for users to indicate whether or not a given link should be added as annotation to the document generated automatically for the session — that is, only those will be part of the documents from all links suggested

3.3. Linking processing in the access phase

At the end of a session, the XML information corresponding the a session is used to generated several alternatives of hyperdocuments for users to review the session. A Web interface (Figure 5, window on top) presents a hierarchical structure of years and semesters that gives access to a list of corresponding sessions that can be reviewed in several formats (Figure 5, window on the bottom) — the sessions can be associated with lectures from courses or meetings from projects, for instance.

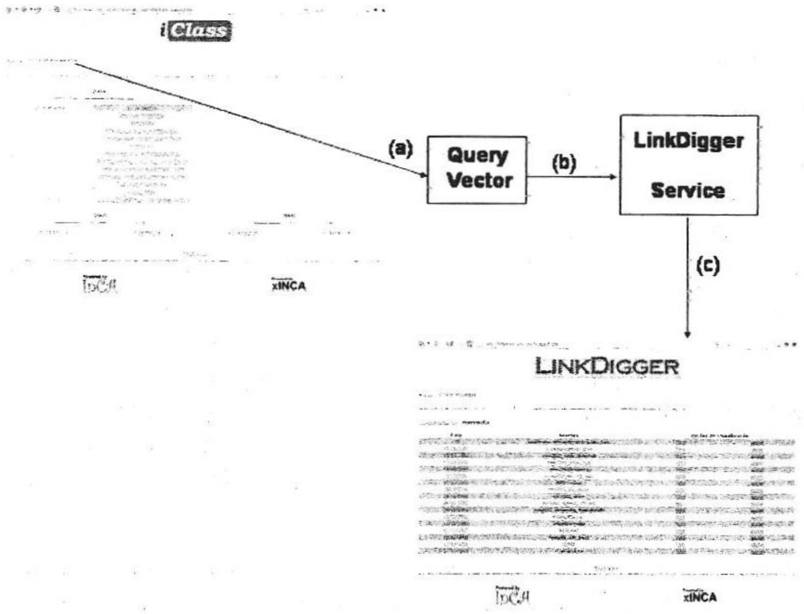

Figure 5: Linking processing in the access phase: (a) keywords provided by users are used as a query vector; (b) the query vector is sent to the linking service; (c) semantic hyperlinks defined between the query vector and the captured material is presented in the access interface.

Among the formats supported in the current version are HTML, XHTML, SMIL and an applet that plays back the session by animating the corresponding strokes (on top of their images) synchronously with the audio stream. While interacting with all those interfaces, users can have recommendations being automatically issued or can make explicit queries that can be used in a query vector submitted to LinkDigger.

Figure 5(a) illustrates that users are able to provide keywords to be used as a query vector. Once this vector has been defined, a requisition is made to the linking service (Figure 5(b)). The vector is processed by LinkDigger and the results are sent to the presentation interface (Figure 5(c)).

3.4. Handling Privacy

The fact that presentation information is continuously captured and shared leads to important issues in terms of privacy. In some situations, users may want to have their material preserved. We defined a mechanism where users specify if they want their material to be used to define hyperlinks and recommendations to other people. At the pre-production phase, users can tell the system to do not create hyperlinks with the material used to produce the documents corresponding to that session in all phases.

To preserve the spirit of sharing, the default option is to be able to share the information produced in all phases.

4. Related Work

Although the Web has empowered us with the ability to distribute and share digital information, it does not bridge the physical real-world and digital documents [Grønbæk et al., 2003]. Facilitating the task of discovering intrinsic reference between live experiences captured by ubiquitous computing infrastructure is a multifaceted approach. As far as ubiquitous hypertext and ubiquitous capture systems are concerned, no reported work reports operations that allow the automatic linking of semantic related documents by means of interaction with electronic devices.

In terms of interfaces, researches have developed interfaces emphasizing the creation, visualization and navigation of relationships and content delivery to mobile devices [Carter et al., 2004] [Romero and Correia, 2003] [Sinclair et al., 2002] [Weal et al., 2003]. Chen has built a multimedia interface for presenting the tracking of communicative actions in a multiparty videoconference [Chen, 2003].

More dedicated to ubiquitous infrastructures than hypertext interfaces, investigations in terms of authoring from capture have been developed. Systems emphasizing location tracking are examples of automatic authoring of documents that can be presented as hyperdocuments. The Digital Family Portrait provides family members with continuous access to a hypertext that represents the amount of activities carried out in a house by a relative that lives elsewhere [Mynatt et al., 2001]. Tracking people's routes in order to identify and locate them [Orr and Abowd, 2000], finding lost objects [Peters et al., 2001] and recording actions to support short-term memory [Tran and Mynatt, 2002] [Trevor et al., 2004] are scenarios which could exploit hypertext to present information recorded from location trackings. Some researches have already investigated prototypes to combine principles from augmented reality and hypermedia to support organizing and managing digital and physical material in terms of spatial relationships [Grønbæk et al., 2003] [Grønbæk et al., 2002].

Also interested in ubiquitous infrastructure there are researches investigating automatic authoring tools in terms of capture of live experiences, as it is the case of Dolphin, a groupware application built to support the capture of audio, video and user-interaction with electronic whiteboards in face-to-face or distributed meetings — generating documents that can be later edited collaboratively [Streitz et al., 1994]. Chiu et al. present the NoteLook system to allow capture, annotation, indexing of the resulting documents during a media-enabled conference room [Chiu et al., 1999]. Chiu also present an algorithm to match images in order to define multimedia links in paper-like interfaces [Chiu et al., 2000]. Also in the context of providing paper-like surfaces for reading and browsing hypertext, Xlibris allows the generation of dynamic links by capturing user interaction with a reading stand-alone device [Price et al., 1998]. 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. In the Tivoli system, information 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 [Haake et al., 1994]. The Authoring on the Fly system provides an elaborate capture infrastructure which emphasizes the support to broadcasting the live presentation [Bacher and Müller, 1998]. Other examples of capture and access applications in the meeting environment are TeamSpace [Richter et al., 2001] and Flatland [Mynatt et al., 1999].

5. Conclusion

Our implementation and its underlying model make use of search and recommendation operations to provide interaction alternatives to linking sessions captured in ubiquitous environments.

The architecture proposed depicts how a linking service can be integrated into a capture and access application in order to provide recommendations in the form of hyperlinks among the documents that are generated automatically as a result of a live session. Moreover, the hyperlinks can be made part of those documents when the user interacts with the information and has the focus of attention in the associated contents (before, during or after the live session).

Our current efforts include using the jInk API to allow gestures to activate linking during the capture phase, as we current achieve with voice.

Considering the whole lifetime of a document, we can generalize our approach so that it can be used to support that other sources of information be anchors of links — the contents of URLs visited during a session, information added in the access phase by means of explicit annotations, or even the contents of chat sessions carried out while users access documents of the repository are examples.

We can also allow users to be able to combine different capture components to create hyperlinks — such as linking only information from the text extracted from slides. Another possible extension it to create links that consider specific attributes to relate sessions — such linking only sessions from a given person.

6. ACKNOWLEDGMENTS

Alessandra Alaniz Macedo and Renan Cattelan are respectively PhD and MsC candidates supported by FAPESP (99/11527-0 and 02/05677-4). Jose Antonio Camacho Guerrero is supported by CNPq (382341/2003-1) and Maria Pimentel is supported by FAPESP and CNPq.

References

Abowd, G. D. (1999). Classroom 2000: an experience with the instrumentation of a living educational environment. *IBM Systems Journal*, 38:508 – 530.

Abowd, G. D., Atkeson, C. G., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N., and Tani, M. (1996). Teaching and learning as multimedia authoring: The Classroom 2000 Project. In *Proceedings of the ACM International Multimedia Conference and Exhibition*, pages 18 – 22, Boston, MA, USA.

Abowd, G. D., Harvel, L. D., and Brotherton, J. A. (2000). Building a digital library of captured educational experiences. In *Proceedings of the Kyoto International Conference on Digital Libraries*, pages 395 – 402, Kyoto, Japan.

Bacher, C. and Müller, R. (1998). Generalized replay of multi-streamed authored documents. In *Proceedings of ED-Media*, Freiburg.

Bulcão Neto, R. F., Izeki, C. A., Pimentel, M. G. C., Pontin, R. P. M., and Truong, K. N. (2002). An open linking service supporting the authoring of web documents. In *Proceedings of the ACM Document Engineering*, pages 66 – 73, Virginia, USA.

Carter, S., Churchill, E., Denoue, L., Helfman, J., and Nelson, L. (2004). Digital graffiti: Public annotation of multimedia content (to appear). In *Proceedings of the CHI 2004*, page 4p., Vienna, Austria. ACM Press.

Cattelan, R. G., Baldochi Jr, L. A., and Pimentel, M. G. C. (2003). Processing and storage middleware support for capture and access applications. In *Poster paper presented at the ACM/IFIP/USENIX International Middleware Conference*, Rio de Janeiro, RJ, Brazil.

Chan, K. F. and Yeung, D. Y. (1998). A simple yet robust structural approach for recognizing on-line handwritten alphanumeric characters. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pages 229 – 238.

Chen, M. (2003). Visualizaing the pulse of a classroom. In *Proceedings of the eleventh ACM international Conference on Multimedia*, pages 555 – 561, Berkeley, California, USA. ACM Press.

Chiu, P., Foote, J., Girgensohn, A., and Boreczky, J. (2000). Automatically linking multimedia meeting documents by image matching. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 244 – 245, San Antonio, Texas, USA. ACM.

Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L. (1999). Meeting capture in a media enriched conference room. In *Proceedings of CoBuild.*

Furnas, G. W., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R. A., Streeter, L. A., and Lochbaum, K. E. (1988). Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the Eleventh International Conference on Research & Development in Information Retrieval*, pages 465 – 480.

Grønbæk, K., Kristensen, J. F., Øbrbæk, P., and Eriksen, M. A. (2003). "Physical hypermedia": organising collections of mixed physical and digital material. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 10 – 19, Nottingham, UK. ACM Press.

Grønbæk, K., Vestergaard, P. P., and Øbrbæk, P. (2002). Towards geo-spatial hypermedia: Concepts and prototype implementation. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 117 – 126, College Park, Maryland, USA. ACM Press.

Haake, J. M., Neuwirth, C. M., and Streitz, N. A. (1994). Coexistence and transformation of informal and formal structures: Requirements for more flexible. In *Proceedings of the ECTH*, pages 1 – 12.

Macedo, A. A., Pimentel, M. G. C., and Guerrero, J. A. C. (2002). An infrastructure for open latent semantic linking. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 107 – 116. ACM Press.

Macedo, A. A., Truong, K. N., Camacho-Guerrero, J. A., and Pimentel, M. G. C. (2003). Automatically sharing Web experiences through a hyperdocument recommender system. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 48 – 56. ACM Press.

Mukhopadhya, S. and Smith, B. (1999). Passive capture and structuring of lectures. In *In Proceedings of the ACM Conference on Multimedia*, pages 477 – 487.

Mynatt, E., Rowan, J., Craighill, S., and Jacobs, A. (2001). Digital family portraits: Providing peace of mind for extended family members. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 333 – 340, Seattle, Washington, USA. New York: ACM Press.

Mynatt, E. D., Igarashi, T., Edwards, W. K., and LaMarca, A. (1999). Flatland: New dimensions in office whiteboards. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 346 – 353.

Orr, R. J. and Abowd, G. D. (2000). The smart floor: A mechanism for natural user identification and tracking. In *Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, pages 275 – 276, The Hague, Netherlands.

Peters, R. E., Pak, R., Abowd, G. D., Fisk, A. D., and Rogers, W. A. (2001). Finding lost objects: Informing the design of ubiquitous computing services for the home. Technical Report GIT-GVU-04-01, Georgia Tech., Atlanta, Georgia, USA.

Pimentel, M. G. C., Abowd, G. D., and Ishiguro, Y. (2000). Linking by interacting: a paradigm for authoring hypertext. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 39 – 48, San Antonio, TX, USA.

Pimentel, M. G. C., Ishiguro, Y., Kerimbaev, B., Abowd, G. D., and Guzdial, M. (2001). Supporting long-term educational activities through dynamic Web interfaces. *Interacting With Computers Journal*, 13:353–374.

Pimentel, M. G. C., Sante, D. G., Bulcão Neto, R. F., Izeki, C. A., and Fortes, R. P. M. (2003). Preparing and extending capture-based documents. In *Proceedings of the International Information and Telecommunication Technologies Symposium*, pages 1 – 8, Florianopolis, SC, Brazil.

Price, M. N., Golovchinsky, G., and Schilit, B. N. (1998). Linking by inking: trailblazing in a paper-like hypertext. In *Proceedings of the ACM Conference on Hypermedia and Hypertext*, pages 30 – 39.

Richter, H., Abowd, G. D., Geyer, W., Fuchs, L., Daijavad, S., and Poltrock, S. (2001). Integrating meeting capture within a collaborative team environment. In *Proceedings of the International Conference on Ubiquitous Computing*, pages 123 – 138.

Romero, L. and Correia, N. (2003). Hyperreal: a hypermedia model for mixed reality. In *Proceedings of the fourteenth ACM conference on Hypertext and Hypermedia*, pages 2 – 9, Nottingham, UK. ACM Press.

Sinclair, P. A. S., Martinez, K., Millard, D. E., and Weal, M. J. (2002). Links in the palm of your hand: Tangible hypermedia using augmented reality. In *Proceedings of the 13th on Hypertext and Hypermedia*, pages 127 – 136. ACM Press.

Streitz, N., Geibler, J., Haake, J., and Hol, J. (1994). DOLPHIN: integrated meeting support across local and remote desktop environments and liveboards. In *Proceedings of the Conference on Computer Supported Cooperative Work*, pages 345 – 358.

Sun Microsystems (1998). Java Speech API Specification.

Tran, Q. and Mynatt, E. (2002). Cook's collage: Two exploratory designs. In *Position paper for the Technologies for Families workshop at CHI 2002.*

Trevor, J., Hilbert, D., Billsus, D., Vaughan, J., and Tran, Q. (2004). Contextual contact retrieval. In *International Conference on Intelligent User Interfaces (IUI 2004)*, page 3p., Madeira, Funchal, Portugal. ACM Press.

Truong, K. N. and Abowd, G. D. (2004). Inca: A software infrastructure to facilitate the construction and evolution of ubiquitous capture & access applications. In *To Appear In the Proceedings of Pervasive 2004: The Second International Conference on Pervasive Computing*, page 18p., Linz,Vienna,Austria.

Weal, M. J., Michaelides, D. T., Thompson, M. K., and DeRoure, D. C. (2003). The ambient wood journals: replaying the experience. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 20 – 27, Nottingham, UK. ACM Press.

Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 265(3):94 – 104.