# Multimodal Interactions for Linking Everyday Presentations in a Ubiquitous Computing Infrastructure

Alessandra A. Macedo
FFCLRP - Univ. de São Paulo
Ribeirão Preto,SP - Brazil
ale.alaniz@usp.br

José Camacho-Guerrero
Innolution Ltda
Ribeirão Preto, SP - Brazil
jcamacho_jr@yahoo.com

Renan G. Cattelan
Univ. Federal de Uberlândia
Uberlândia,MG - Brazil
renan@facom.ufu.br

Valter Inácio Jr.
IBm Brazil Ltda
Campinas, SP - Brazil
valterinacio@gmail.com

Maria G. C. Pimentel
ICMC - Univ. de São Paulo
São Carlos,SP - Brazil
mgp@icmc.usp.br

## ABSTRACT

Live presentations can be captured in instrumented places generating documents with the talk. A presentation is normally related to others as well as to documents. Therefore, linking of the related documents, before and after a live presentation takes place, can be useful when lecturers are interesting in knowing more about his talk. Moreover, the availability of linking during the presentation provides a unique opportunity for linking in the live context itself. On the one hand, it is interesting that the creation of links occurs automatically given the amount and diversity of information. On the other hand, the evaluation of the possibilities of links identified by algorithms can be a burden when the user does not have the focus of attention on the associated task. We propose that well-known Web interacting operations be used to identify links. The operations should be made available *before*, *during* and *after* live presentation and in any case the user should attach relevant results as annotations. We present the model and associated implementation. The activation of the operations can be done automatically by multimodal interactions, typing and browsing.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Information Storage and Retrieval; H.5.2 [**User Interfaces**]: Interaction Styles

## General Terms

Design, Experimentation

## Keywords

Multimodal Interaction, Information Retrieval, Ubiquitous Computing, Capture and Access Applications, Hypermedia

## 1. INTRODUCTION

A problem faced by people interacting with computing technology is that they must adapt themselves to those technologies. The demand for more natural user-computer interaction motivated the built of ubiquitous computing [30]. Ubiquitous computing proposes the seamless integration of technologies into a physical environment so as to aid humans in their everyday activities, without changing the way that they perform those activities. Multimodal interactions consist on using of user inputs, such as voice, gestures and pen interaction, to interact with computational technology. Cross-modality interactions are welcome efforts to provide natural user interaction required by ubiquitous computing.

In ubiquitous computing, capture and access applications exploit the fact that everyday experiences are rich in information that may be captured and made available as Web documents, which in turn can be reviewed later by users [3] [17]. Users of capture and access applications can benefit from the use of multimodal interfaces because they may control capture of content exploiting non-traditional inputs and access of captured material. The degree of automation is increased as suggested by [1] before, during and after underlying activities to be captured.

Key design dimensions in the construction of capture and access applications include defining when and where the capture occurs, methods for capturing and annotating, number of devices, techniques for reviewing access information and the length of time the capture information persists [27]. Moreover, it has been observed that, rather than try to capture everything, system design should focus on the psychological basis of human memory [23]. Such systems, in essence, tackle limitations of traditional pen-and-paper or audio-and-video capture [13], for instance in meeting environments [31].

Investigating some of these issues, we propose the use of multimodal interactions to linking information captured from everyday presentations supported by ubiquitous computing infrastructure in more natural way. Our proposal intends facilitating the generation of associated documents as extra information to users who can not lose the focus of attention on the underlying presentation. Our proposal suggests multimodal interactions in three categories according to user inputs: *linking by typing*, *linking by inserting* and *linking by capturing*. *Linking by typing* occurs when users assign metadata or keywords as anchor to linking with stored

information. *Linking by inserting* is related to the input of files to be processed and converted to an index used to linking with other information. Finally *linking by capturing* is defined when interactions via voice, gesture and hadwritting are captured and used to be linked with information. These types of linking by multimodal interaction are illustrated in an educational context.

Exploiting our proposal, ubiquitous environments could became more interactive and intelligent. Using voice, gesture, typewriting or handwritting, users are able to know extra information related to the information being captured without any cognitive overhead, just interacting with the environment. The provided multi-modality may allow end-users to interact with technology in ways that are most suitable. Presenters will be able to request information by voice, for instance, when they are writting on the whiteboard and need to access information - hands-free. The information can then come back to them as recommendation.

The remaining sections are organized as follows: we first we discuss related work; then we present previous work; next we describe multimodal interactions aiming to activate linking operations; last we present experiments, benefits of our approach and comments on future work.

## 2. RELATED WORK

On the Internet, users are able to easily access digital information, but they are overhead when try to discovery intrinsic relationships between information. The task of discovering intrinsic links is an open multifaceted approach including researches from Information Retrieval, Natural Language Processing, Hypermedia, etc. Nowadays attempts are focusing on natural ways of interaction including efforts from Ubiquitous Computing, Human Computer Interaction, etc.

Synchronization of captured information by means of ubiquitous environments usually occurs according to temporal indexing or user interaction with electronic capture devices. Indexing by synchronization relates information considering temporal variables instead of textual and structural comparisons. The Cornell Lecture Browser captures images of the slides presented during the session and automatically generates a table of contents that provides synchronized indexing into the slides and the video information [14]. DOLPHIN, a groupware application, supports co-located or distributed meetings [24]. 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 including gesture recognition [12]. DOLPHIN does not synchronize audio and video elements. Tivoli integrates generated information after session by means of capturing the content to be imported by a collaborative hypermedia system. In any case, models for the captured information are a must [5].

The Authoring on the Fly (AoF) system provides an elaborate capture infrastructure that implements capture, transmission and retrieval of distributed live sessions [15]. Alongside with DOLPHIN and iClass, this seems to be the only implementation that truly supports an underlying hypertext structure for the documents generated. After capture sessions, AoF processes all captured material and generates automatically hyperdocuments which allow reproduction of sessions by means its synchronization model. It is also possible to edit material after a capture session.

The NoteLook system allows capture, annotation, indexing of the resulting documents during a media-enabled conference room. Chiu et al. alsonpresent an algorithm to match images in order to define multimedia links in paper-like interfaces [8]. Also in the context of providing paper-like surfaces for reading and browsing hypertext, Xlibris supports the generation of dynamic links by capturing user interaction with a reading stand-alone device, implementing a hypertext structure that allows the classification and organization of annotations, as well as the searching for related information in the device [21]. Also in meetings domain, the work by Kalnikaité et al. exploits mobile touch screen devices to allow users to add explicit capture marks [13]. Concerning retrieval, the automatic processing of audio and video has also been investigated [22]. KSU-FIS Lecture Recording System exploits a set of cameras and microphones to capture activities carried out and classroom conducted by instructors in the Kyushu Sangyo University in Japan [28].

All presented systems were investigated in order to study underlying models which support retrieval of related information. The majority presented markup streams of media items captured as anchors of hyperlinks to the media [14] [1] [15]. Some allows the addition of textual information and search by means of users queries [19]. Considering multimodal interactions, few scientific works were studied. A voice-based web browser called BrowserVox deals with different Brazilian-Portuguese websites using automatic speech recognition and text-to-speech for multimodal interactions (mouse or voice), with audiovisual feedback [16]. Watanabe et al. describe an approach to develop Web multimodal applications using X+V and widgets [29].

## 3. BACKGROUND

Given the aim of facilitating the access to associated documents by means of defining automatically hyperlinks, we advocate that it is possible in capture and access domain. INfrastructure for Capture and Access Applications (INCA) implements network abstractions demanded in the implementation of those applications [26]. It provides architectural support for building applications by means of four key design concerns: (a) *Capture* (part of the system is responsible for the capture of information as streams of data); (b) *Storage* (part of the system is responsible for the storage of attribute-tagged information); (c) *Transduction* (part of the system converts information into different formats); and (d) *Access* (part of the system provides access to multiple integrated streams as multimedia information).

For each design concern, INCA implements a module and associated interface which 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. To control the presence of capture and access modules, INCA also provides a *Registry* that manages publishing and subscribing operations carried out by application modules.

In INCA, the term *access* is associated with the fact that an *access* module has *synchronous access* to the DataObject handled by a *capture* component. In the context of capture and access applications, the term *access* involves both *synchronous* and *asynchronous* accesses.

Exploiting INCA, *iClass* is a capture and access application, inspired in *eClass* [2], that records information produced during a lecture, including strokes and slides from

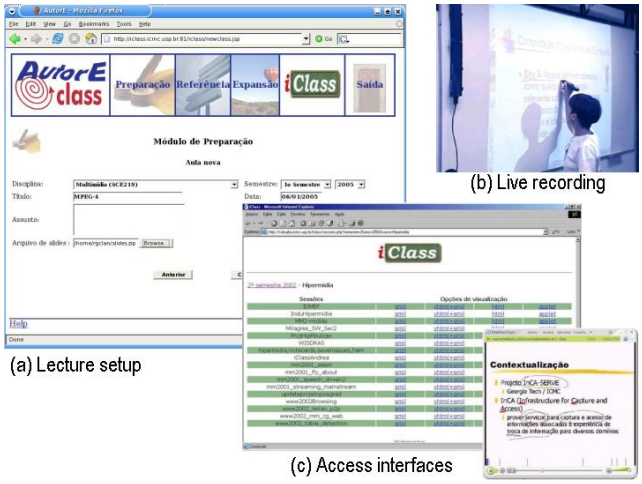


**Figure 1: (a) Web form: lecture setup. (b) Live recording. (c) Access interfaces**

an electronic whiteboard, audio, video and Web browsing. Concluded the session, an XML document integrating the different captured media is produced and stored for access.

*iClass* follows the phases suggested in the literature for structuring capture and access [2] [18]: *pre-production*, *live recording*, *post-production*, *access* and *extension*. The *pre-production* phase is related to the session setup using an authoring tool called AutorE [20]. AutoE 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 supports, in the access phase, a user manually extending the captured information with annotations and links. Figure 1(a) shows a Web form used for lecture setup (course name, lecture title, slides file, term and date/time information) which has portions set automatically since contextual information such as *who*, *when*, *where* and *what* is combined with user and time/location information.

During lecture startup, the instructor selects the support required for that particular lecture involving whiteboard, audio, etc. Once the session starts, INCA capture components are instantiated and run on a computer attached to an electronic whiteboard or a tablet pc. A storage component is also instantiated and is responsible for contacting a server informing which capture components are running on the client. Next, the storage service will receive information from capture components running in the client. The *live recording* and *post-production* phases happen simultaneously using a callback mechanism.

Outside a live session, a hierarchical structure of years, terms, courses and lectures is presented in dynamically-built Web pages. Figure 1 (c)(top) lists the lectures recorded for a particular course in a given term: the title of the lecture appears on the left, and the remaining columns lead to options of visualization — SMIL, XHTML+SMIL, XHTML and a Java applet that playback slides in a stroke-by-stroke mode in synchrony with audio or video, if captured. Figure 1 (c)(right) shows a SMIL document corresponding to a lecture being presented with RealOne® Player: slides and electronic ink are presented synchronously with audio.

# 4. LINKING BY MULTIMODAL INTERACTION

Considering the demand for (i) natural interactions in ubiquitous computing applications and (ii) discovery of intrinsic relationships between information in many areas, we propose multimodal interactions that are divided in *linking by typing*, *linking by inserting* and *linking by capturing*. As a result, these operations generate extra information as recommendation to users.

Our proposal was instantiate to support a capture and access application in an educational context, the iClass system, in all capture and access phases: before, during and after a capture session. Figure 2 illustrates inputs of multimodal interactions for each phase aiming to provide hyperlinks as extension of the material being manipulated. Before a capture session, a user adds slides and metadata to setup his/her session (see Figure 2(1.1)). This information is sent to a linking level to be linked (see Figure 2(a) and (e)). During a session, as information is captured using xINCA (see Figure 2(2.1)), this is transcribed (or not) (see Figure 2(2.2)) to formulate a query vector sent to the linking level (see Figure 2(b), (d) and (e)). After the end of a session (see Figure 2(1.3)), captured repository is searchable and extensible by user queries (see Figure 2(c) and (e)). The product of all phases are query vectors sent to the linking level which is responsible for the following linking processing (see Figure 2(3)):

**Collecting:** information captured in each phase of a capture and access application. Each captured session by the ubiquitous infrastructure generates a document.

**Indexing:** significant words (excluding stopwords) are extracted from each document (a captured session). Text operations can be applied to facilitate the representation of these documents and allow the logical view from that of a full text to that of a set of index terms. So optional stemming is carried out, what reduces each word to its linguistic root based on the language detected. LinkDigger[1]'s current implementation supports documents in English, Spanish and Portuguese. Other text operation can be assigned such as identification of nouns groups eliminating adjectives, adverbs and verbs and compression in terms of text operations.

**Generation of a Terms-Documents Matrix:** the terms by documents matrix is generated, where rows and columns represent index terms and documents, respectively. The Terms-Documents Matrix is called matrix $X$. Not all terms are equally useful for representing the document contents: less frequent terms allow identifying a narrower set of documents. The importance of the index terms is represented by weights associated to them. Every term is given an appropriate weight as its term frequency ($tf$) relative to the frequency of the same term in the whole documents collection, the inverse document frequency ($idf$) [4]. The weight as zero indicates that a term does not belong to the document. Non-binary weights provide consideration for partial matches. These term weights are used by LinkDigger to compute a degree of similarity between pairs of document and query vectors.

[1]Using Latent Semantic Indexing (LSI) [11], we built LinkDigger to create semantic hyperlinks between textual and documents images repositories.

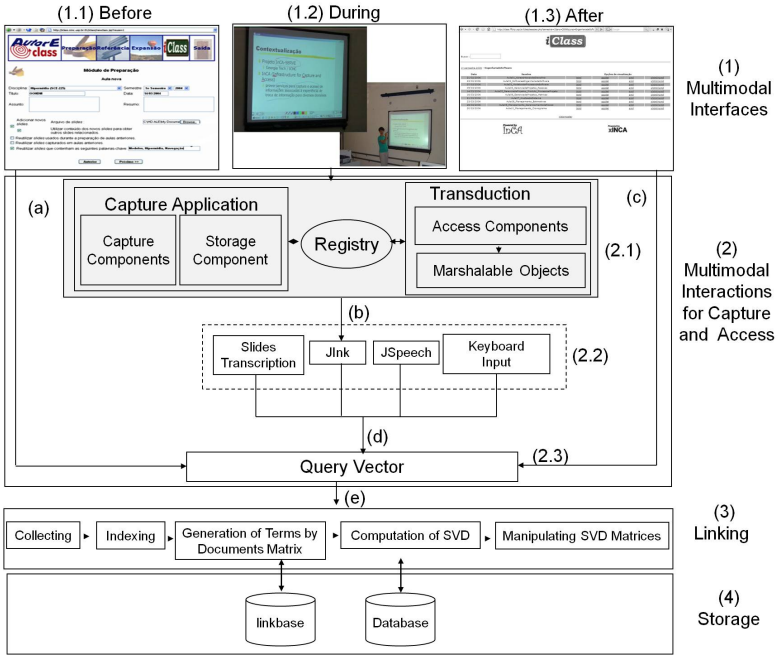


**Figure 2: A multi-level architecture supporting linking from information captured by an educational capture and access system. Information from multimodal interactions as input of a linking service integrated in all phases of a capture and access application.**

**Computation of SVD:** the matrix $X$ is decomposed into three component matrices $T$, $S$ and $D'$ using Single Value Decomposition (SVD), which is part of the LSI technique [11]. By convention the diagonal elements of $S$ are constructed to be all positive and ordered in decreasing magnitude, so the first $k$ largest may be kept and the remaining smaller ones are discarded. We extended SVD to automatically select entries larger than 70% of sum of all entries of $S$. Our value of $k$ considers a lost of 30% of information. The tuning of $k$ is still investigated in literature [9].

**Manipulating SVD Matrices:** the reduced dimensionality solution generates a vector of $k$ real values to represent each document. SVD provides reduced rank-k approximation for the column and row space of a term-document matrix $X$ for any value of $k$. The outcome is the reduced matrix $\hat{X}$, which is obtained by multiplying the three reduced component matrices. The reduced matrix is stored by LinkDigger to be compared with queries vectors, composed by multimodal interactions, in order to provide hyperlinks as suggestions.

Our implementation is instantiated by LinkDigger which, once a day, prepares the existing repository of captured material to be linked to information from multimodal interactions. Next we present how multimodal interactions are exploited to generate the query vector to search previous captured material. These multimodal interactions are generated according to users's needs submitted together with calls from tools supporting a capture and access processing during all phases of a capture and access application: pre-production, live capture and access of captured information.

## 4.1 Pre-production phase: linking by typing and inserting

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 the authoring system AutorE. By using AutorE, presenters prepare a new session for ubiquitous capture by reusing material from other sessions or by adding brand new material in the forms of metadata or prepared slides. In Figure 2(1.1), the top Web interface is a typical interface of the preparation module.

The preparation module was extended by multimodal interactions which activate the LinkDigger service. When preparing a session, users may receive suggestions of related material used in other sessions. So presenters are able to reuse and embed those material in their session. This is achieved by the following multimodal interactions: linking by typing and linking by inserting.

*Linking by typing* occurs (i) when users input metadata in some fields (Lecture, Title, Subject, Term, Date and Abstract) of the preparation interface and (ii) when users type explicitly keywords in a specific field of search. *Linking by inserting* is the result of inclusion of prepared slides to the session setup. These two operation sent information to AutorE system (see Figure 3). Upon user's request, AutorE prepares the received information to be sent to LinkDigger (see Figure 3(1)). It converts into text information typed by the user in forms for adding metadata and keywords, or information extracted automatically from the slides. Pre-processed words from files or typed information compose a query vector created to be automatically sent to LinkDigger (see Figure 3(2)). Afterwards, the vector is processed

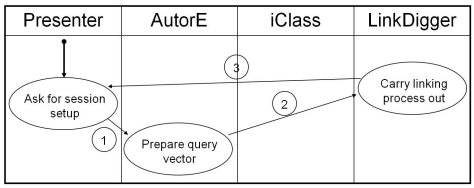


**Figure 3: Upon linking by inserting and typing, users receive recommendation of related material to be added to the presentations being prepared.**

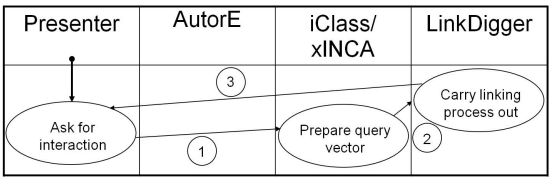


**Figure 4: Upon linking by capturing, strokes captured by whiteboard component are used to compose a query vector along with text automatic extracted from prepared slides. The activation of the query is done via a voice. The resulting recommendations are shown in a popup window.**

by the linking service, which identifies documents semantically related to the vector and shows them to users (see Figure 3(3)). Once a day, LinkDigger carries out its processing. The outcome is a list of relationships (presented as lecture pages) corresponding to other sessions that are used as recommendation. This outcome is sent to the lower part of the original AutorE interface.

It is important to observe that any other authoring application can be used in the pre-production phase. This application just needs to send a query vector to the LinkDigger API. The proposal of relating material while a session is been setup intends to enrich information to be presented without changing user's focus of attention on the underlying preparation. The authors defend the use of this kind of approach once people have become daily more demanding and need to be contextualized with more complete information to become competitive.

## 4.2 Live capture phase: linking by capturing

The live capture phase presented in Figure 2(1.2) shows a live session in progress with support of the iClass system: the prepared slides are presented on 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 (Extended Infrastructure for Capture and Access Applications) toolkit presented previously.

*Linking by capturing* occurs from capturing of *textual information from slides* and *handwritting over slides* by the lecturer during live presentations. *Texts from the slides* presented during the session are used to transparently built the query vector sent to LinkDigger. However, considering such ubiquitous environment, it is important that information from the handwritten annotations and from the capture audio stream may be provided as interaction alternatives to also build the query vector.

Considering a query vector composed by texts from slides and recognized handwritten characters[2], the LinkDigger service automatically identifies relationships among this vector and information in an 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 (see Figure 4(1)) and, next, sends the vector to the linking service (see Figure 4(2)). As a result of the LinkDigger processing, hyperlinks are defined on-the-fly and presented as recommendations to the presenter (see Figure 4(3)).


**Figure 5: iClass presents related material to be selected and used to extend material being captured.**

In terms of multimodal interaction, we also allowed users to use voice 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) [25] developed by IBM. JSAPI specification defines an access interface to the functionalities of recognition and voice synthesis provided by engines like ViaVoice. Therefore, a user can use a voice command to activate the composition of a query vector with information extracted by xINCA. Issued the command, the query is sent to the LinkDigger (this can also be activated manually via a button on the interface) which presents related documents in a small pop-up window, as shown in Figure 5. The pop-up window allow to select each recommended material 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. Only selected material will be part of the documents from all links suggested. The authors believe that in some years this kind of approach will be embedded in many capture and access environments in order to assist people with different goals in terms of related information.

## 4.3 Access phase: linking by typing

At the end of a session, the XML information corresponding a session is used to generate several alternatives of hy-

[2]In order to recognize on-line *handwritten characters from annotations*, we developed the jInk API [10] according to the method proposed in [7]. jInk, provides a graphical interface incorporated into the iClass system to be used by users to adjust the character models to their own writing styles to obtain a more accurate recognizer. Besides digits and letters, jInk can also recognize symbols and gestures. Further details can be found in [10].

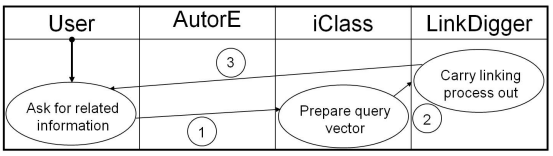


**Figure 6: Linking processing in the access phase: (a) keywords provided by users are used as a query vector; (b) this vector is sent to the linking service; (c) semantic hyperlinks defined between the query vector and the captured material are presented.**

perdocuments for users to review the session. A Web interface 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 1(c), access interface on the top) — the sessions can be associated with lectures from courses or meetings from projects (Figure 1(c), access interface on the bottom), for instance.

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 synchronously with the audio stream. Multimodal interactions occur while users are interacting with all those interfaces. They access recommendations being automatically issued or they can express linking by typing. *Linking typing* occurs when users compose explicitly queries that can be used in a query vector submitted to LinkDigger using keywords they input.

Figure 6(1) 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 6(2)). The vector is processed by LinkDigger and the results are sent to the presentation interface (Figure 6(3)). Providing end-users with related information about captured material it will benefit people interested in knowing more information when they access the documents presenting captured information.

# 5. EXPERIMENTS

Here we discuss experiments designed to verify the feasibility and utility of our proposal applied before, during and after a capture session has taken place in an educational scenario. Extending the iClass system, our service related 278 lectures from 10 courses taught between 2004 and 2006.

For our purposes, each lecture is a textual document represented as a vector. The set of representative keywords (index terms) of each lecture was reduced since terms as conjunctions, prepositions and pronouns (stopwords) were eliminated. A matching process between words from lectures and stopwords dictionaries was done looking for index terms. After excluding stopwords and lectures with less than 10 words, our service manipulated a matrix $X$ with 205 lectures (columns) and 13006 distinct words (lines). The semantic feature of the semantic relationships generated by our service is achieved through the Latent Semantic Indexing (LSI) technique [11]. LSI organizes text objects into a semantic structure appropriate for matching, what overcomes usual problems of lexical approaches such as polysemy and synonymy. Considering LSI, the matrix $X$ was decomposed into three component matrices $T$, $S$ and $D'$ using Single Value Decomposition (SVD), which is part of the LSI theory. The matrix $S$ is a diagonal matrix with non-zero entries (called singular values) along a central diagonal. We considered the $k$ most important dimension as 87 (automatically calculated considering 30% of loss) during the reduction of the singular matrix $S$ are selected, and all other factors are discarded: only the 87 highest values in the singular matrix $S$ were selected. We extended SVD to automatically calculate the value of $k$ given a lost of 30%. The reduced dimensionality solution generates a vector of $k$ real values to represent each document. SVD provides reduced rank-k approximation for the column and row space of a term-document matrix $X$ for any value of $k$. The outcome is the semantic reduced matrix $\hat{X}$, which is obtained by multiplying the three reduced component matrices.

In order to obtain the similarity level between documents, the calculation of similarity based on cosine measure over each pair of document vectors extracted from the matrix $\hat{X}$. We have considered maximum cosine as 1 and minimum as 0.8. Cosine values farther to zero mean the highest similarity levels between a pair of documents. The outcome is a list of relationships between each pair of lectures that can be used for search and recommendation purposes. Our approach related lectures from courses offered by staffs from three different Computer Science departments of three different institutes. Results from each group of experiments were qualitatively analyzed by a faculty with a strong Computer Science background. Each link was evaluated in a high/medium/low/no relevance scale.

## 5.1 Before a session

Using the interface presented in Figure 3(a), a professor, interested in teaching about models for development of software in an HCI (Human Computer Interaction) course, exploited our proposal. She uploaded her set of pre-prepared slides and tried to look for related material. As a result, our implementation built the query vector with metadata, typed keywords and words from the inserted file. The query vector was sent to LinkDigger to relate it with previous captured lectures. From 278 lectures, 34 were presented to the professor as recommendation of related content. The professor may select material which should be reused in her presentation being prepared. From 34 recommendations, 9 were highly related to the lectures that was being prepared (high relevance), 13 with medium relevance, none with low relevance, and 12 with no relevance. Unsatisfied with the results, she carried out a second attempt.

She had already inserted metadata and pre-prepared slides but now she also selected the field "search by keywords" on the interface for setup lectures to be captured. So LinkDigger could compose the query vector with "process models and software development keywords", metadata and words from pre-prepared slides. 46 lectures were presented as material semantically related to the information typed and inserted by the professor. So she was able to select material which should be reused in her presentation being prepared. The 46 returned lectures were distributed as following: 28 lectures from 3 Software Engineering Courses (SE), 11 from 2 Object-Oriented Programming Courses (OOP), 4 from 1 Computer Networks (NC) Course, 1 from 1 Human-Computer Interaction Course, 1 from 1 Parallel Computing (PC) Course and 1 from 1 Operating System (OS) Course. From all presented lectures, 14 links were rated with high relevance to the lecture being prepared, 15 with medium

relevance, 11 with low relevance (lectures related to UML modeling) and 6 with no relevance (lectures related to nc, PC and OS Courses. The high relevant lectures were directly related to theory and examples of process models taught in the Software Engineering Courses. She finally selected the following 4 lectures to enrich her material: Formal Methods, RAD Model, HCI Patterns and Software Process Models.

In another opportunity, a speaker who was preparing a meeting about Codding Standards exploited our proposal. As the HCI professor, he uploaded pre-prepared slides and selected to reuse content from previous captured material just by checking reuse of previous captured lectures (he could type specific keywords to allow the selection of related lectures to be reused). 14 captured lectures were defined as semantic related material to his presentation: 9 lectures taught in OOP courses and 5 in SE courses. The speaker reused just 6 lectures: 5 lectures are about OO paradigm and Java, and 1 lecture, given in a SE course, is about programming and examples of language programming. Indeed, from 14 presented lectures, 6 lectures were semantically related to slides being uploaded (rated with high relevance).

## 5.2 During a session

Two live lectures were planned to experiment our proposal during capture sessions. One experimentation occurred in a HCI (Human-Computer Interaction) course and another was carried out in an OOP (Object-Oriented Programming) course. With a live session in progress with support of the iClass system, a professor was explaining about different types of HCI models presented in the HCI literature. During that presentation, the presenter asked for extra-material using a specific voice command, "see related material" in order to activate the linking processing and receive suggestions to be added to that lecture. Towards creating links, our service is considering the whole content of slides visited.

To relate the HCI model lecture to others, the set of lectures was used. The three following Software Engineering (ES) lectures, given in different terms and years, were presented as related to the HCI model lecture: Part I - Software Process Model (99.31%), Part II - Software Process Model (cosine as 99.53%) and Part I Software Process Model (97.80%). The two first lectures were taught in 2005 and the last one in 2004. Actually, all ES lectures suggested were semantically related to the HCI model lecture. Some HCI and ES models are the same and others have identical phases.

In a second attempt, our proposal was experimented in an Objet-Oriented Analysis and Project (OOAP) lecture taught in the OOP course. The 2 following lectures were returned to the professor as related material: UML Sequence Diagrams (cosine as 99.97%) and Software Process Model (99.85%). The first lecture was taught in OOP courses and the second in a Software Engineering. Both lectures were semantically related to the OOAP lecture. Sequence Diagrams are designed in Object-Oriented paradigm and some process models give support to this kind of project.

## 5.3 After a session

The linking process after capture session generated 338 links between all lectures. A qualitative analysis was carried out and the results are as follows: 113 links were rated with high relevance, 89 with medium relevance, 116 with low relevance and 20 with no relevance. Analyzing the same links generated by the same linking process with the same relevance scale, an undergraduate student presented the following results: 121 links were rated with high relevance, 90 with medium relevance, 120 with low relevance and 7 with no relevance.

We could observe that many highly relevant links (30 links) were found among Software Engineering lectures being taught in different years. It was expected once it has more textual information than the others. Most medium relevance links were created from lectures with many exercises to lectures with different goals: exercise practice, projects presentation and traditional lectures. As far as the contents were concerned, those lectures had more differences than similarities. Low relevance links related lectures where projects or seminars were presented. Most no relevance links related sessions where demonstrations or tests were carried out. Many lectures introducing the different courses were related despite those lectures have different content. We believe it occurs mainly because those lectures had some obvious similar keywords: introduction, basic course information, exams, teacher, bibliography and others.

The majority of links are related to Software Engineering, Algorithm and Data Structures and Theory of Computing. We believe this majority of links among those lectures is resulting of the fact those lectures have the major portion of the textual information indexed. For example, an introductory Algorithm and Data Structures lecture presenting binary number and Abstract Data Type (ADT) was related to a Computer Programming presenting implementation in language C of arrays and matrices as ADT. Search and hashing algorithms from different lectures were related.

The experiments have been chosen because they manipulate homogeneous information. The qualitative results confirm that this proposal is able to identify important relationships. It is also relevant to observe that our service has been experimented considering recall and precision measures [6]. Recall is the fraction of known relevant documents which were effectively retrieved. Precision is the fraction of retrieved documents which are known to be relevant.

# 6. 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.

Future efforts possibly will include to allow gestures to activate linking during the capture phase, as we current achieve with voice. Our work could be augmented with accessibility and usability guidelines.

The fact that presentation information is continuously captured and shared leads to important issues in terms of privacy. In some situations, users may want 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 cor-

responding to that session. To preserve the spirit of sharing, the default option is to be able to share the information produced in all phases. 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 is to create links considering specific attributes to relate sessions — such linking only sessions from a given person.

# 7. ACKNOWLEDGE

Authors would like to thank FAPESP, CNPq and CAPES.

# 8. REFERENCES

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

[2] G. D. Abowd. Software engineering issues for ubiquitous computing. In *Proceedings of the 21st International Conference on Software Engineering*, pages 75–83, 1999.

[3] G. D. Abowd and E. D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Trans. Comput.-Hum. Interact.*, 7(1):29–58, 2000.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1ª edition, Janeiro 1999.

[5] C. Braga. From access control policies to an aspect-based infrastructure: A metamodel-based approach. In *MoDELS Workshops, LNCS 5241*, pages 243–256, 2008.

[6] J. A. Camacho-Guerrero, A. A. Macedo, and M. G. C. Pimentel. A look at some issues during textual linking of homogeneous Web repositories. In *Proceedings of ACM Document Engineering*, 74–83 2004.

[7] K. Chan and D. Yeung. A simple yet robust structural approach for recognizing online handwritten alphanumeric characters. In *Proc. of the 6th Int. Workshop on Frontiers in Handwritting Recognition*, pages 229–238, 1998.

[8] P. Chiu, J. Foote, A. Girgensohn, and J. Boreczky. Automatically linking multimedia meeting documents by image matching. In *Proceedings of the eleventh ACM on Hypertext and hypermedia table of contents*, pages 244 – 245, San Antonio, Texas, USA, 2000. ACM.

[9] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391 – 407, 1990.

[10] V. dos Reis Inácio Junior. Um framework para desenvolvimento de interfaces multimodais em aplicações de computação ubíqua. Master's thesis, USP - ICMC, 2007.

[11] G. Furnas, S. Deerwester, S. Dumais, T. Landauer, R. Harshman, L. Streeter, and K. Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th International Conference on Research & Development in Information Retrieval*, pages 465–480, 1988.

[12] J. Haake, C. Neuwirth, and N. Streitz. Coexistence and transformation of informal and formal structures: Requirements for more flexible. In *Proceedings ECTH94*, pages 1 – 12, 1994.

[13] V. Kalnikaité, A. Sellen, S. Whittaker, and D. Kirk. Now let me see where i was: understanding how lifelogs mediate memory. In *Proc. ACM CHI'2010*, pages 2045–2054, 2010.

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

[15] R. Muller and T. Ottmann. The "Authoring on the Fly" system for automated recording and replay of (tele)presentations. *Multimedia Systems Journal*, 8(3):158 – 176, May 2000.

[16] E. Munzlinger and C. H. Q. Forster. Desenvolvimento e avaliaç ao de um sistema multimodal e multiusuário de navegaç ao web. In *WebMedia '08: Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*, pages 29–32, New York, NY, USA, 2008. ACM.

[17] M. Pimentel, G. Abowd, and Y. Ishiguro. Linking by interacting: a paradigm for authoring hypertext. In *Proceedings ACM on Hypertext and Hypermedia*, pages 39 – 48, 2000.

[18] M. Pimentel, Y. Ishiguro, B. Kerimbaev, G. Abowd, and M. Guzdial. Supporting educational activities through dynamic web interfaces. *Interacting with Computers*, 13(3):353–374, 2001.

[19] M. Pimentel, A. Macedo, and G. Abowd. Linking homogeneous Web-based repositories. In *Proceedings of International Workshop on Information Integration on the Web*, pages 35 – 42, Rio de Janeiro/Brazil, 2001. URL: http://www.cos.ufrj.br/wiiw/schedule.html.

[20] M. Pimentel, D. Sante, R. B. Neto, C. Izeki, and R. Fortes. Preparing and extending capture-based documents. In *Proceedings of International Information and Telecommunication Technologies Symposium*, pages 1 – 8, Florianopolis,SC,Brazil, 2003.

[21] M. Price, G. Golovchinsky, and B. N. Schilit. Linking by inking: trailblazing in a paper-like hypertext. In *Proc. of ACM Conference on Hypertext*, pages 30–39, 1998.

[22] A. Ranjan, J. P. Birnholtz, and R. Balakrishnan. Improving meeting capture by applying television production principles with audio and motion detection. In *ACM CHI'08*, pages 227–236, 2008.

[23] A. J. Sellen and S. Whittaker. Beyond total capture: a constructive critique of lifelogging. *Commun. ACM*, 53(5):70–77, 2010.

[24] N. Streitz, J. Geibler, J. Haake, and J. Hol. DOLPHIN: integrated meeting support across local and remote desktop environments and liveboards. In *Proc. Conf. on Computer Supported Cooperative Work*, pages 345 – 358, 1994.

[25] Sun Microsystems. Java Speech API Specification, 1998.

[26] K. N. Truong and G. D. Abowd. Inca: A software infrastructure to facilitate the construction and evolution of ubiquitous capture & access applications. In *Proceedings of Pervasive 2004: The Second International Conference on Pervasive Computing*, pages 140–157, Linz,Vienna,Austria, April 2004.

[27] K. N. Truong and G. R. Hayes. Ubiquitous computing for capture and access. *Foundations and Trends in Human-Computer Interaction*, 2(2):95–171, 2009.

[28] K. Ushijima. Construction and practice of a new educational environment with lecture recording system. In *Proc. of the International Conference on Advanced Information Networking and Application (AINA)*, pages 12–15, Fukuoka, Japan, 2004. IEEE Computer Society.

[29] W. M. Watanabe, A. T. Neto, A. G. da Silva Filho, and R. P. de Mattos Fortes. Desenvolvimento de componentes de interfaces multimodais ricas para a web utilizando x+v e dojo widgets. In *WebMedia '08: Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*, pages 97–100, New York, NY, USA, 2008. ACM.

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

[31] Z. Yu and Y. Nakamura. Smart meeting systems: A survey of state-of-the-art and open issues. *ACM Comput. Surv.*, 42(2):1–20, 2010.