Sixth Annual IEEE International Conference on Pervasive Computing and Communications

ArCoMo—An Artefact-based Collaborative Mobile Learning Environment

Christian Hoff, Ulf Wehling, and Steffen Rothkugel
Faculty of Sciences, Technology, and Communication (FSTC)
University of Luxembourg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
{christian.hoff, ulf.wehling, steffen.rothkugel}@uni.lu

Abstract

In this paper we propose Arcomo, a mobile learning environment designed for hybrid wireless networks. We introduce three key concepts used in the environment: spaces, anchor points, and artefacts. Spaces are used to uniquely identify, store, share, and disseminate learning materials, annotations, and examples among mobile users. Anchor points are defined by users on documents in spaces. Every anchor point specifies a location in a document, which can be a word, a sentence, a paragraph, an image, or even the whole document. Artefacts are utilized to describe relationships between anchor points together with comments on them. This way, they can map the current working context of a student that typically contains several open documents and a relationship between them. The main contribution is a general annotation concept which supports different file formats and is applicable for collaborative mobile users.

1 Introduction

Students at universities are using computers already since a couple of years to write papers or lecture notes, to do their exercises, to search for additional information in the Internet or books in the library, and a lot more. Also communication with fellow students via e-mail or instant messaging as well as sharing files among them is an important task of computers. In recent years, there is a change from using stationary computers at home, in public computer rooms at the university or library, or Internet cafes to mobile computers like laptops or tablet PCs. Equipped with technologies like Bluetooth or Wi-Fi in ad hoc mode, they are able to establish spontaneous groups and exchange data in the lecture room, in the cafeteria, or in the park. Other communication technologies like GPRS, UMTS, or Wi-Fi access points allow staying "online" anytime and anywhere. We refer to this combination of mobile ad hoc networks (MANETs) and infrastructure networks as hybrid wireless networks.

The paper-and-pencil annotation paradigm is a fundamental technique in traditional learning. Highlighting sentences in a book, writing down some hints on a page in an article, or adding a post-it note with a reference to another source is daily business during literature research and learning. The concept is also present with respect to electronic documents, but with less success, however. Today's electronic annotation systems are not as mature as their traditional paper-and-pencil counterpart [8, 7]. Most annotation systems are limited to a single document format. Sharing of annotations is only possible in very specific settings, like for instance commenting PDF documents with Adobe Acrobat [16]. Sharing different types of documents—or files in general—should be easily possible while retaining annotations. Either single documents might be annotated, or multiple interrelated documents might be connected creating more complex annotations. Explicit support for mobile users is a major objective of the envisaged environment.

Along the lines of Norman [18], we strive for a learner-centered approach. In fact, we literally implement Norman's vision of *cognitive artifacts*, i.e. "anything invented by humans for the purpose of improving thought or action". Landauer [15] in turn distinguishes between two phases when using computer technology: *automation* and *augmentation*. It is particularly the latter that fosters human performance in achieving cognitively complex tasks. That is where our concept of artefacts manifests itself.

In this paper we propose a mobile learning environment called Arcomo that exploits the communication possibilities of hybrid wireless networks. The environment offers a general annotation concept that supports different file formats and is suitable for collaborative mobile users. Section 2 gives a brief overview over some existing annotation systems. Section 3 provides insight into the envisaged mobile learning scenario. Section 4 introduces the concept of *spaces*, which are used to logically group documents (or

0-7695-3113-X/08 $25.00 © 2008 IEEE
DOI 10.1109/PERCOM.2008.99

arbitrary data in general) together and to share them with others. *Anchor points* and *artefacts*, which are the basis for our general annotation concept, are described in the Sections 5 and 6. Section 7 finally concludes this paper.

2 Related Work

Although the technologies employed to build spontaneous learning groups and to share new information among the members are available, most e-learning environments lack support for such dynamic and mobile settings. Furthermore, creating, sharing, and accessing annotations on learning materials has an important influence on the learning process [7, 8]. Today, there are no prevalent annotation services even though a variety of sophisticated annotation systems exist on the market since a couple of years. Brust *et al.* [7] traced this observation back to several problems in common annotation systems. These systems are commonly designed to support only a single document format, e.g. PDF, DOC, or HTML. Since people also work on several documents concurrently while researching a particular aspect, the main focus can switch from one document to another. However, this kind of lateral reading cannot be captured in current annotation systems.

Nevertheless, a large number of annotation systems or services exist enabling users to annotate documents or even reply to existing annotations. Most of the existing annotation systems are designed for the Web, i.e. for annotating HTML documents. Two such Web-based systems enabling users to annotate HTML documents are ComMentor [19] and Annotea [14]. Both systems also provide support for sharing annotations with others. A more detailed overview over existing Web-based annotation systems can be found in [11]. Other applications enabling users to comment or annotate documents include Adobe Acrobat (PDF) and Microsoft Word [20] (DOC/DOCX). However, all these systems are restricted to one special document type and do not allow to interlink and annotate different document types.

The Xanadu Project [17] uses a model that identifies documents, versions, links, and even single words or characters through unique IDs. The text of the documents as well as each version (for instance an updated paragraph) resides in the overall address space. New documents create new text fields in the address space, but also reuse existing parts when referencing another document. Finally, each document is represented as a list of reference pointers to different text parts in the global address space. The CosmicBook [3] system, respectively the CosmicReader, is an implementation of this model with a graphical representation of such reference pointers across multiple documents and application windows. However, it is restricted to hyperlinks between the proposed document format. Thus, it does also not support different document types.

3 Mobile Learning Application Scenario

The following example illustrates briefly our vision of how students will work in such a mobile environment, sharing documents as well as annotations on them, and creating relationships among information items. Lisa and John are both studying computer science at the university. In the current semester they are both attending to the Operating Systems course. They are already registered in the mobile learning environment and have subscribed to the learning material repository of that course. This global repository resides on a server in the Internet and hosts all kinds of learning materials. Whenever their laptops have a connection to that global server, they automatically receive new lecture slides, handouts, exercise sheets, sample programs and a lot more of learning materials. The received documents are automatically replicated to the local repositories, which reside on the user's devices. In the recent lecture the professor introduced the concept of semaphores to synchronize threads. In the lab session to that course students have to solve the dining philosophers problem using the C programming language. John found during some Web research an HTML page with an animation illustrating the dining philosophers problem. He saved the HTML page on his laptop and interlinked it with the exercise sheet, the animation, and the slide covering semaphores of the recent lecture. He also added some notes with comments and ratings describing the usefulness of the referenced documents for their current task. As he decides to share this documents with fellow students, he also adds the documents to his local repository. In the afternoon Lisa and John meet each other in the cafeteria. The stored HTML page with the animation as well as John's notes are transferred from John's to Lisa's computer using the ad hoc Wi-Fi connection of their laptops. Looking at the animation, Lisa remembers the C course she visited previous semester. In this course they used system calls for I/O operations and she already created some notes interlinking a reference guide on system calls in the C programming language with that course. She browses her laptop for these notes and adds them to the current notes on semaphores and the dining philosopher problem. John considers this as very useful information and they decide to make the notes available for other students. Depending on the availability of the global server, they will be stored in the global repository and visible for other students.

4 Spaces

As aforementioned, our mobile learning environment is based on a global and many local repositories. Since users typically have different organizing principles for their documents we introduced the concept of *spaces* [22]. Basically, a space is a collection of related documents, replicated across different machines, like laptops or servers. Every document within a space is uniquely identified through a Uniform Resource Identifier (URI), which is independent of its physical location. The advantage is that the users are not tied to any prescribed structure when sharing their documents. The documents within a space are only logically grouped together like it is done in virtual folders. Physically, they can be distributed across the whole local disk as well as across machine boundaries. Also moving or renaming a document on the local disk later will not break the link between the document and the space. Users subscribe to spaces in order to share documents. Whenever users who have subscribed to the same space meet each other, locally available documents unknown to the other device are exchanged automatically. The same applies to users who are connected to a server that hosts files residing in that space.

The *space* URI scheme introduced in [22] is used to identify documents within a space, independent of their physical location. Applied to the mobile learning application scenario, a concrete example is the following. Assume the professor teaching the Operating Systems course has created a space with the name `OS` on the host `mlearning.uni.lu`. Students attending this course can subscribe to that space by using the URI `space://mlearning.uni.lu/OS` together with a unique username. The professor himself has subscribed to that space with the username `bob@uni.lu`. Assume that the current lecture is about semaphores and the professor adds the corresponding presentation to that space. Since the file name of the lecture slides on the professor's device is `semaphores.pdf`, they are mapped under this name. Then the complete URI to uniquely identify the lecture slides is `space://mlearning.uni.lu/OS/semaphores.pdf?owner=bob@uni.lu`. However, such URIs are automatically generated on the local device each time a user adds a file to a space.

The mapping between a space URI and the physical location of the corresponding document on the local disk is done in a platform specific way. Under Mac OS X Spotlight comments are employed to hold the space URI of a given document. Spotlight itself—which also searches the Spotlight comments—is then employed to resolve a space URI locally, i.e. to find a file specified through a URI on the local disk. As Spotlight indexes files the search is very fast. Whenever changes in the file system are made the Spotlight comments of the affected files are preserved. Thus, moving or renaming a file locally will not result in breaking the link between the document and the associated space. A similar mechanism for the Microsoft Windows platform is not yet implemented in the current prototype such that moving and renaming files later on results in broken links. However, the implementation of such a mechanism is subject of future work.

One popular and already widely supported technology to retrieve relevant content is Really Simple Syndication (RSS). Users subscribe to RSS feeds in order to get notified about newly available information concerning different topics in an automated manner. Moreover, RSS can also enhance student research [9]. Technically, RSS feeds are plain XML files and can contain any kind of data. Thus, they are well suited to exchange information about locally available documents among the subscribers. Every registered user of our mobile learning environment provides an RSS feed containing a list of space URIs, which represents the locally available documents. The same applies to the servers hosting spaces, which also provide RSS feeds. Every incoming feed is parsed by the receiving device, which compares the received list of space URIs with the locally available documents. Unknown respectively new documents are requested from the corresponding device.

Since every user in the Arcomo environment can act as information provider as well as subscriber, the communication among the devices must be coordinated. To organize the communication the current prototype uses the clustering algorithm proposed in [4], which is especially designed for hybrid wireless networks. This algorithm selects dedicated devices—called clusterheads—within the local one-hop neighborhood. The selection is based on different device properties, like battery power, wireless networking technologies, and more. Whenever the local content of a space changes the mobile device sends the corresponding RSS feed to its clusterhead. The clusterhead is then responsible for conciliating the received RSS feeds and requesting respectively disseminating the corresponding documents among all reachable subscribers. Depending on connectivity the clusterhead is also in charge of pushing new documents to the servers as well as to retrieve new documents from there.

The current prototype responsible for the dissemination is implemented in Java on top of the network simulator JANE [10]. This simulator allows to develop applications and test them in simulation, hybrid, and platform mode. In hybrid mode graphical user interfaces are attached to the simulation. This way, real users can interact with selected devices hosted by the simulation. In platform mode the application is executed on real hardware without any simulation running in the background. The devices currently use Wi-Fi as communication technology.

5 Anchor Points

When students read books, lecture slides, handouts, or other learning materials they want to mark distinct elements in these materials. The reasons for that can be manifold. Maybe because the sentence is very important for the overall understanding or on the other extreme the sentence is totally unclear and must be further investigated. Besides this first step of marking elements, students want to add some annotations or even to create relationships to other learning materials and resources. A more elaborate discussion on annotations can be found in [7].

To accomplish the first step—namely highlighting elements—we introduced the concept of *anchor points* (APs). An AP describes a distinct section of a document, e.g. a word, a sentence, a paragraph, a picture, or even the whole document. They are used to mark a certain aspect in a learning material or user document the student is interested in. APs are managed by the applications responsible for the specific file format themselves. The applications provide functionality to create and retrieve APs and to store them in the original files or metadata of these files. Later these APs can be accessed from other applications allowing to open the documents in their corresponding application showing the specific AP.

We already developed a first prototype that allows us to add and open APs in MS Office documents. Nowadays XML file formats become more and more popular. Due to the DOM representation of XML file formats, it is easy to mark elements in the DOM tree as APs and to integrate them into the original file format. In the case of MS Office we developed an extension with an integrated user interface to create and store APs in the new MS Office XML file formats. In [21], we also performed some investigation on obtaining context information and supporting APs in OpenOffice, source code (Eclipse) and HTML (Firefox) files.

Currently we aim at identifying a generic API that applications should provide to support APs. Our vision for the future is that software vendors implement this simple API, enabling the applications to handle APs inherently—just like most applications allow copy-and-paste or drag-and-drop.

6 Artefacts

Although there are conceptually as well as technologically highly sophisticated annotation systems, we have to put in question why today there is no prevalent annotation system. Furthermore, although most learning materials are available in digital form, studies show that for annotation purposes, people still prefer pen and paper [5]. Fundamental problems [7] in annotation systems lead to this situation. Designed for a single media format only, they force learners to use different annotation systems for various learning materials. Furthermore, people open several documents and are reading multiple, complementary sources at the same time. But annotation systems lack capturing this lateral reading and thinking.

To overcome these limitations in existing annotation systems the concept of *artefacts* is introduced. Artefacts are directly built on APs and allow to add annotations and to create relationships between different resources. Basically, artefacts capture the context a user is currently working or thinking in. The context includes open documents or more precisely the currently visible parts of the documents, the relationships between these documents, as well as comments on the type or idea behind this relationship. A detailed discussion on exploiting context information can be found in [21].

Technically, artefacts are small XML documents connecting APs enriched with comments. Note that a comment itself can also be a document or another artefact. This way, comments are not restricted to plain text only. Like user documents and learning materials, artefacts are stored in the local repository and disseminated depending on the access rights to other local repositories or to the global server.

Another important aspect is a reasonable visualization of artefacts and the artefact space. A straightforward visualization is shown in Figure 1. Documents referenced in the artefact are shown in their original application. Other windows on the users current desktop not related to the visualized artefact are hidden behind a semi-transparent curtain. The active applications are scaled and rearranged to fit onto the screen. The original constellation of the users desktop is shown in the red frame (Figure 1). The viewport of these applications is set to the linked APs, which are highlighted and visually connected to the artefact window. The artefact window itself shows the annotation describing the context the user was working in. Besides this straightforward visualization, we are also thinking about more sophisticated ones. For example, during browsing a document the list of artefacts referencing the currently visible anchor point should be presented to the user. Hovering with the mouse over this list, previews of the referenced documents are displayed. The whole artefact can be displayed by zooming and arranging the application windows of the corresponding document similar to the Exposé feature of Mac OS X. Links between anchor points are drawn in different colors depending on their usefulness and ratings. The application windows can also be visualized in 3D showing more important documents in the foreground and less important ones in the background.

Artefacts also build a good starting point for Collaborative Filtering (CF). CF algorithms use a similarity matrix S to calculate the usefulness of items for a distinct user. Each entry $s_{i,j}$ in S describes the similarity between item i and item j. Based on these similarity values different techniques (e.g. weighted sum or regression) are applied to calculate a prediction of the usefulness of an item for the active user. Since artefacts in a space describe weighted relations between the documents in that space, CF techniques can be utilized to recommend documents to other users interested

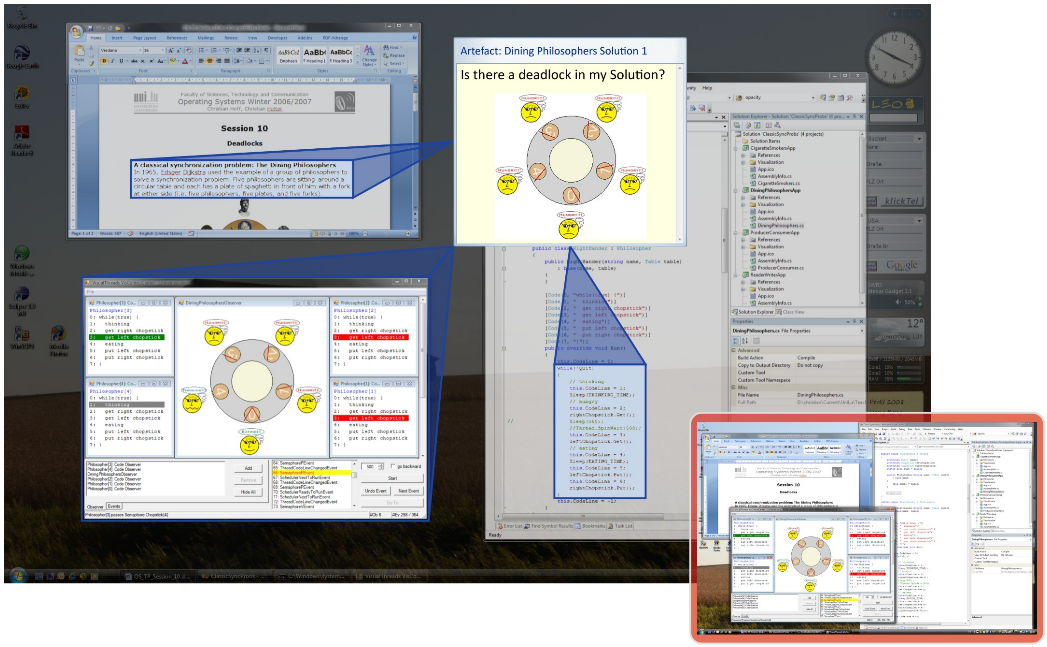

Figure 1. Visualization of an artefact with scaled and arranged application windows showing highlighted anchor points. Red frame: original application windows (unscaled and unarranged).

in the same topic. The relations between documents (rated pairs of APs) are aggregated in the similarity matrix S and used to suggest documents to users. We described the CF approach more detailed in [12, 13].

Widely used open source Learning Management Systems (LMSs) like ATutor [1] or Moodle [2] as well as commercial ones like Blackboard's Academic Suite [6] that integrates the former WebCT environment are *reactive* systems. They wait for user interaction and react to corresponding requests. New LMSs are following a *proactive* approach to support the students in the learning process. Rule-based systems [23] direct students to distinct activities based on predefined rules and user experiences. Introducing CF techniques into LMSs allow to evoke proactive behavior by e.g. recommending learning materials the user should have a look next. Furthermore, this information can be used to pre-fetch documents the user is supposed access in the future with a high probability.

Besides this "indirect" collaboration by providing recommendations or document pre-fetching, artefacts also foster direct cooperation among mobile learners. Since artefacts describe a context the user is working in, they can be used to raise open questions, to sketch problems, but also hints or solutions. As already depicted in the introducing application scenario, Lisa and John have to solve the dining philosophers problem in the C programming language. There are already a lot of learning materials and resources in the corresponding space. For instance, lecture slides describing semaphores, an exercise sheet with the problem definition, a C reference guide on how to use system calls to access semaphores, and a lot more. They also created an artefact interlinking all this useful information together with comments and hints for a solution. This allows them to restore the recent working context very easily. Now, Lisa implements a first version of the dining philosopher using the C programming language, but she observes a deadlock during execution. She creates a new artefact on top of the artefact describing the problem definition and references also the source code, or to be more precise the passage in the source code that is responsible for the deadlock in her opinion. The artefact is enriched with some comments describing the deadlock and a snapshot of the program execution leading to it. In the evening Lisa and John meet each other in the cafeteria of the dormitory. The repositories are automatically synced through the ad hoc Wi-Fi connection of their notebooks and John restores the context the deadlock occurred by double clicking the new artefact. They immediately start discussing and the failure is found quickly. John modifies the code and adds a new artefact referring to the artefact describing the deadlock and the modified program. He also adds some annotations explaining the failure in the previous version and his solution. As soon as the repositories are in sync again, Lisa and other students in the same learning group can access this information to avoid this failure.

7 Conclusion

In this paper we presented Arcomo, a mobile learning environment for hybrid wireless networks. The environment is based on the concepts of spaces, anchor points, and artefacts. To summarize our vision, the interaction between the three concepts is described briefly. Users subscribe to spaces in order to share learning materials, user documents, as well as artefacts on them with other users in a transparent fashion, independent of their physical location. Anchor points allow users to mark different elements in a document. This can be for instance a sentence or paragraph in the document, which is important for understanding a particular aspect. Anchor points that are created by different users on the same document are also synchronized automatically between the different repositories. The concept of artefacts finally enables users to interlink anchor points from different source documents and to comment or rate them. A visualization of such an artefact, interlinking and commenting different document types, is shown in Figure 1.

References

[1] ATutor—A Web-based Learning Content Management System (LCMS). http://www.atutor.ca.

[2] Moodle—A free, Open Source course management system (CMS). http://www.moodle.org.

[3] Ted Nelson's CosmicBook. http://xanadu.com/cosmicbook/.

[4] A. Andronache, M. R. Brust, and S. Rothkugel. Multimedia content distribution in hybrid wireless networks using weighted clustering. In *WMuNeP '06: Proceedings of the 2nd ACM international workshop on Wireless multimedia networking and performance modeling*, pages 1–10, New York, NY, USA, 2006. ACM Press.

[5] D. Bargeron and T. Moscovich. Reflowing digital ink annotations. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 385–393, New York, NY, USA, 2003. ACM Press.

[6] Blackboard Company. Blackboard Academic Suite. http://www.blackboard.com.

[7] M. R. Brust and S. Rothkugel. On Anomalies in Annotation Systems. *aict*, 00:3, 2007.

[8] G. Cabanac, M. Chevalier, C. Chrisment, and C. Julien. Collective Annotation: Perspectives for Information Retrieval Improvement. In *RIAO 2007: Proceedings of the 8th Conference on Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, 2007.

[9] S. J. Cold. Using Really Simple Syndication (RSS) to enhance student research. *SIGITE Newsl.*, 3(1):6–9, 2006.

[10] D. Görgen, H. Frey, and C. Hiedels. JANE-The Java Ad Hoc Network Development Environment. *anss*, 0:163–176, 2007.

[11] R. M. Heck, S. M. Luebke, and C. H. Obermark. A Survey of Web Annotation Systems. http://www.math.grin.edu/~rebelsky/Blazers/Annotations/Summer1999/Papers/survey_paper.html, 1999.

[12] C. Hoff, P. Gratz, and S. Rothkugel. Collaborative filtering in artefact-based m-learning environments. In *International Workshop on Mobile and Ubiquitous Learning Environments (MULE 2007) in conjunction with 15th International Conference on Computers in Education (ICCE 2007)*, November 2007.

[13] C. Hoff, P. Gratz, and S. Rothkugel. Context-sensitive prediction in artefact-based m-learning environments. In G. Richards, editor, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2007*, pages 2629–6021, Quebec City, Canada, October 2007. AACE.

[14] J. Kahan and M.-R. Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 623–632, New York, NY, USA, 2001. ACM Press.

[15] T. K. Landauer. *The Trouble with Computers: Usefulness, Usability, and Productivity*. The MIT Press, Cambridge, MA, USA, June 1996.

[16] R. Mendoza and K. Peterson. From tools to tasks: discoverability and adobe acrobat 6.0. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 912–927, New York, NY, USA, 2004. ACM Press.

[17] T. H. Nelson. Xanalogical structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use. *ACM Comput. Surv.*, 31(4es):33, 1999.

[18] D. A. Norman. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison Wesley Publishing Company, Reading, MA, USA, May 1993.

[19] M. Röscheisen and C. Mogensen. Interaction design for shared world-wide web annotations. In *CHI '95: Conference companion on Human factors in computing systems*, pages 328–329, New York, NY, USA, 1995. ACM Press.

[20] G. B. Shelly, T. J. Cashman, and M. E. Vermaat. *Microsoft Office Word 2003: Complete Concepts and Techniques*. CourseCard Edition: Course Technology Press, march 2005.

[21] U. Wehling, C. Hoff, S. Rothkugel, and M. R. Brust. Exploiting Context Information for Computer-based Annotation Systems. In *INTED 2007: Proceedings of the International Technology, Education and Development Conference*, March 2007.

[22] U. Wehling and S. Rothkugel. Spaces–A Replication Scheme for Sharing Tagged Files in Hybrid Wireless Networks. In *UBICOMM 2007: Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. IEEE Computer Society, November 2007.

[23] D. Zampunieris. Implementation of a proactive learning management system. In T. Reeves and S. Yamashita, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006*, pages 3145–3151, Honolulu, Hawaii, USA, October 2006. AACE.