UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

MOBILE ELDIT: CHALLENGES IN THE TRANSITION FROM AN E-LEARNING TO AN M-LEARNING SYSTEM

Anna Trifonova, Judith Knapp, Marco Ronchetti and Johann Gamper

January 2004

Technical Report # DIT-04-009

Mobile ELDIT: Challenges in the Transition from an e-Learning to an m-Learning System

Anna Trifonova*, Judith Knapp+, Marco Ronchetti*, Johann Gamper°

* University of Trento – Sommarive 14 – 38050 Povo (Trento) – Italy
Tel. +39-0461882033 Fax +39-0461882093 E-mail {Marco.Ronchetti, Anna.Trifonova}@dit.unitn.it
+ European Academy of Bolzano – Drususallee 1 – 39100 Bozen – Italy
Tel. +39-0471055092; Fax +39-0471055089; E-mail Judith.Knapp@eurac.edu
° Free University of Bolzano – Dominikanerplatz 3– 39100 Bozen – Italy
Tel. +39-0471315666; Fax +39-0471315649; E-mail Johann.Gamper@unibz.it

Abstract: This paper presents 'Mobile ELDIT' (m-ELDIT), a system under development which goal is to offer access from PDAs to the learning materials of ELDIT – an adaptable language learning platform. The ELDIT system, which is developed at the European Academy of Bolzano, consists of a learner's dictionary and a text corpus. Exercises, a tandem system and a tutor module are planned. The system works online and the adaptation of the content is done dynamically according to interactions with the learner. We analyze the requirements for adaptation and transformation of the data, the user interfaces and also the architecture of the 'Mobile ELDIT'. We discuss the necessity of specific user modeling in order to provide both online and offline access to the learning materials from mobile devices.

Introduction

Mobile learning is a field that recently has attracted the interest of lots of researchers in the learning domain (Trifonova & Ronchetti 2003b). Basically m-learning is considered any form of studying, teaching or learning that is delivered through a mobile device or in a mobile environment. In general by mobile device we mean PDAs and digital cell phones, but more generally we might think of any device that is small, autonomous and unobtrusive enough to accompany us in every moment and can be used for educational purposes.

We analyzed different ways to apply mobile devices in education (Trifonova & Ronchetti 2003a) and discussed that a mobile learning system should have three main functionalities – "Context Discovery", "Mobile Content Management and Presentation Adaptation" and "Packaging and Synchronization". First of all the "Context Discovery" should discover context information which is important for the learning situation, like the devices' capabilities and limitations (software and hardware) or other information about the infrastructure; user/device location and relevant environmental information; temporal information; preferences and etc. This data should be used by the "Mobile Content Management and Presentation Adaptation" for adapting the content for the specific device and user needs. Finally the "Packaging and Synchronization" should take care of selecting the content that will be needed by the user during the offline usage of the system and upload it. It is best that the entire process is done automatically, thus is important that the user activities are tracked and feedback to the algorithm to improve its work.

In our current work we want to develop a mobile version of an existing online language learning system, called ELDIT, which is designed to satisfy the specific needs of the bilingual region of South Tyrol in Italy. ELDIT is currently adaptable and will be adaptive to user behavior, needs and preferences. As mobile devices (PDAs, smart-phones, etc.) are becoming more popular a useful possibility would be to access the system from such devices. In order to provide such functionality we started to build the Mobile ELDIT (or m-ELDIT) system. Beside the adaptation of the content for the specific needs of mobile devices the main difference of m-learning and e-learning is the connectivity. While the content of ELDIT is generated on the fly upon a user request, thus requiring always 'on' connection, mobile devices often have periods of disconnection, either intentionally (when the connection is too expensive) or not (when no infrastructure is provided). Facing this problem Mobile ELDIT should be able to hoard the content needed for the offline usage.

Hoarding is a technique for selecting a set of documents to be uploaded and used when disconnected. Related terms are caching and prefetching, though they are more often used when considering online conditions and web performance. Caching is a technique for keeping content that has been requested by one user available on the nearest server for a certain amount of time so other requestors can access it faster. Prefetching on the other hand is a technique which tries to guess what will be needed to the client in the near future, cache it and this way improve the clients' experience. Different schemes of caching and prefetching are proposed and the goal is to reduce network traffic, to minimize access latency, bottlenecks, servers' workload, etc. in the WWW world. Although the goal of hoarding content for offline use is little shifted from the one of Web caching, some of the techniques can be reused. However, while in the online case one can balance between the accuracy of the cached set and the added traffic, in the situation we consider a much higher accuracy is required, and, as an additional constraint, the memory is limited.

The rest of the paper is organized as follows: Section 2 gives a description of ELDIT; Section 3 gives more detailed description of the problems we are facing for supporting the use of the system from personal digital assistants (PDAs) and an overview of the architecture that we consider; in Section 4 we discuss the user modeling that is needed for adaptation and automatically selecting content for offline usage. Related work, Conclusions and References follow.

The ELDIT System

South Tyrol is a bilingual (German and Italian) province located in the north-east of Italy. Although both languages, German and Italian, are official languages, only few people consider themselves truly bilingual. Citizens are entitled to use their mother tongue in dealings with the public administration including judicial authorities. Therefore, passing the so-called exam in bilingualism is a prerequisite for employment in the public sector.

The main scope of the ELDIT project (http://www.eurac.edu/eldit) is to create an innovative electronic language learning system for the population of South Tyrol to prepare for the exams in bilingualism. However, the system has been designed in a very general way, such that everybody interested in learning the German or Italian languages can profit from it.

Figure 1 shows the main modules of the ELDIT system. Based on the material for preparation of the exams in bilingualism, we have developed an electronic learner's dictionary and a text corpus. The dictionary is especially designed to reduce the burden of vocabulary acquisition in foreign language learning. The text corpus contains all the texts of the exams in bilingualism. Each word is annotated with lemma and part-of-speech and is linked to the corresponding dictionary entry, which facilitates a quick dictionary access for unknown words. Furthermore, we will implement simple quizzes that can be generated automatically out of the existing data set, a tandem module for collaboration, and an adaptive tutor which guides the learner through an individual vocabulary acquisition process which alternates between vocabulary learning and applying the vocabulary on a suitable text.

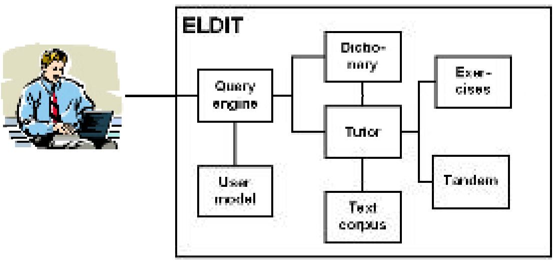
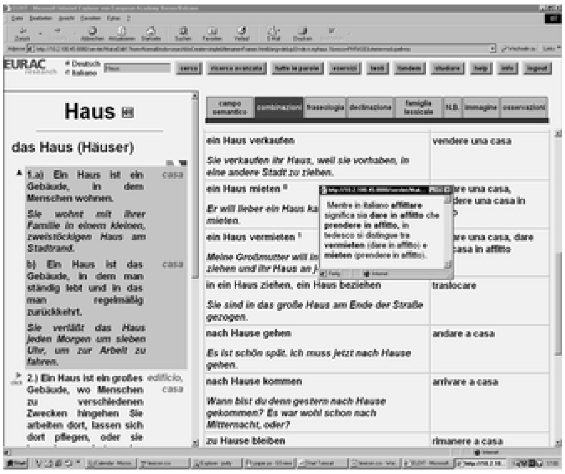


Figure 1: ELDIT Architecture (main modules)



Haus

das Haus (Häuser)

1.a) Ein Haus ist ein Gebäude, in dem Menschen wohnen. — casa

Sie wohnt mit ihrer Familie in einem kleinen, zweistöckigen Haus am Stadtrand.

b) Ein Haus ist das Gebäude, in dem man ständig lebt und in das man regelmäßig zurückkehrt. — casa

Sie verläßt das Haus jeden Morgen um sieben Uhr, um zur Arbeit zu fahren.

2.) Ein Haus ist ein großes Gebäude, wo Menschen zu verschiedenen Zwecken hingehen. Sie arbeiten dort, lassen sich dort pflegen, oder sie — edificio, casa

ein Haus verkaufen *Sie verkaufen ihr Haus, weil sie vorhaben, in eine andere Stadt zu ziehen.*	vendere una casa
ein Haus mieten *Er will lieber ein Haus ka… mieten.*	…re una casa, …dere una casa in …
ein Haus vermieten *Meine Großmutter will i… ziehen und ihr Haus an j…*	…re una casa, dare …casa in affitto
in ein Haus ziehen, ein Haus beziehen *Sie sind in das große Haus am Ende der Straße gezogen.*	traslocare
nach Hause gehen *Es ist schön spät. Ich muss jetzt nach Hause gehen.*	andare a casa
nach Hause kommen *Wann bist du denn gestern nach Hause gekommen? Es war wohl schon nach Mitternacht, oder?*	arrivare a casa
zu Hause bleiben	rimanere a casa

Mentre in italiano **affittare** significa sia **dare in affitto** che **prendere in affitto**, in tedesco si distingue tra **vermieten** (dare in affitto) e **mieten** (prendere in affitto).

Figure 2: Screenshot of the dictionary entry of the German word "Haus"

A dictionary entry is presented to the user in two frames (see figure 2). The left-hand frame shows the lemma of the word and a list of different word meanings, each of which is described by a definition, an example sentence, and an optional translation equivalent in the other language. The right-hand frame is organized in several tabs and shows additional, semantic and syntactic information such as word combinations, related words, linguistic difficulties, etc. The linguistic difficulties are also indicated by a kind of footnote numbers and shown in a small window on the place where they occur.

One of the main design guidelines in the ELDIT project was to consider pedagogical and psycholinguistic learner demands. In order to respond to these learner demands we need to store rather detailed information about the language learning material and resources (Gamper & Knapp, 2003). Our learning material shows some important characteristics which differ from traditional systems: the data are semi-structured and highly interlinked and have to be annotated at a very fine-grained level of detail. In fact, we have to encode information at the level of single words and even below. This level of detail is needed in order to support the language learner as much as possible, and at the same time it allows reusing the learning material for several purposes.

Last but not least, very fine analyses about user preferences and user behavior can be carried out. In the ELDIT system adaptable and adaptive features are distinguished. Adaptable features allow for the manual, a-priori customization of the system. Adaptive features cover the aspect that the system adapts automatically to the user, based on assumptions about the user as well as on observations about the user's interaction with the system. Currently, only adaptable features are implemented, but not yet enabled; the implementation of the adaptive features is future work.

Adopting a rapid prototyping approach, we were seeking for a simple, yet expressive language to implement our data model, which at the same time is robust to frequent changes and facilitates the knowledge engineering process. XML shares these properties and turned out to be a good choice for the implementation (Gamper & Knapp, 2003).

m-ELDIT: The Problem and the Proposed Architecture

Design Goals and Functionalities

The objective of the Mobile ELDIT project is the development of a mobile version of the ELDIT system introduced in the previous section. In the study we assume that the m-ELDIT user is a self-motivated learner, preparing for the bilingual exam, who doesn't need any supervisory control of studying process. As mentioned earlier ELDIT could be used by the students both as learners' dictionary and for preparing for the bilingual examination. Commercial dictionaries for mobile devices are available on the market. The development of such a dictionary itself is not an interesting research topic. The only issue that is open here, considering the ELDIT dictionary, is the compression of the data so that it can fit in the device memory because the ELDIT dictionary itself contains big quantity of data which is hard to store on the nowadays still limited device's memory. This is because of the fine-granularity of the data, where the annotations are at the level of single words and even lower, in order to support the adaptive generation of the pages, scalability and robustness to frequent changes and updates. Our assumption is that only part of the whole ELDIT content can fit into the memory and the amount of memory may vary on different devices. Our system aims to automatically select and upload the content that will be needed by the user during the next period of disconnected usage of the system. The content will be adapted to the device characteristics in advance. The decision on the hoarding set (what content should be uploaded) must be more precise then in the general hoarding or the web prefetching case (see related work section), thus we need more efficient prediction of the user future needs. Predicting what words an arbitrary user will search for in a dictionary is in practice an unfeasible task. In this sense we find it suitable to provide access to mobile users only to text modules of ELDIT and the dictionary entries related with them (words). The prediction can benefit from some preferences of the learning domain, which are:

- The search space is much more limited than in the whole web case
- Semantic information can be available through the metadata
- Behavior of generic users can be analyzed so as to extract most likely paths to be followed
- Behavior of the particular user (preference, learning style, etc.) could contribute finding an optimal strategy.

As mentioned earlier for offering different services to mobile users, including access of learning content, the system should support three main functionalities – "Context Discovery", "Mobile Content Management and Presentation Adaptation" and "Packaging and Synchronization". The content accessed from mobile devices should be especially designed or automatically adapted for the limited device capabilities. The presentation of learning materials is an important issue and should be carefully designed. If the content will be accessed through a standard web-browser on the PDA then it should not contain incompatible elements, for example scripts.

The Architecture

In our system we have decided to introduce a separate server (we call it m-ELDIT server) which will take care of the three functionalities needed to support mobile users' access (see figure 3). As discussed the ELDIT data (both word entries and texts) are XML files. The presentation logic is separated from the data itself, thus different presentations can be generated according to the needs. The special formatting could be done either on the client device, either on an intermediate server (often called transcoding server) or inside the LMS. The mobile devices have much more limited processing power and memory limitations and also the battery power should be considered, thus a probably better choice is that the server does this task. The client request for a page (figure 4) gives the context information needed to generate the pages, i.e. the HTTP version, the device display resolution, support of colors, etc. Using the HTTP request information the 'Content Redesign Engine' should produce from the ELDIT XML files the proper presentation pages.

The 'User Behavior Analyzing Engine' is responsible for retrieving useful information about the users' learning styles and preferences that will be used later by the 'Hoarding Engine' in finding the hoarding set. Although ELDIT has its own user models collected during the usage of the system they do not contain all the information needed for the mobile scenario. For example by analyzing the web server log files we can discover similarities between learning paths of different users, but also the differences. The user models will be discussed in the next section.

The 'Hoarding Engine' is the main module through which the mobile user interacts with mobile ELDIT. Based on the user request's context data it should ask the 'Content Redesign Engine' for properly formatted pages that should be pushed to the device's cache memory. The hoarding algorithms should take as input the output from the 'User Behavior Analyzing Engine' (i.e. the user models with the similarities and the differences of the particular user with the common users' behavior and the current user preferences and learning history) and additional information about the learning content itself. Then it should predict which path the user is most likely to follow and assign weights to the learning objects depending on how important they are for the next user session. The objects (the redesigned pages in our case) with higher weights should be uploaded to the device first; afterwards the materials with smaller weights should be uploaded until the device's available cache is filled. The module should be able to analyze how successfully the previous hoarding was done and try to improve further prediction. Possible methodologies for evaluation are discussed further in the paper.

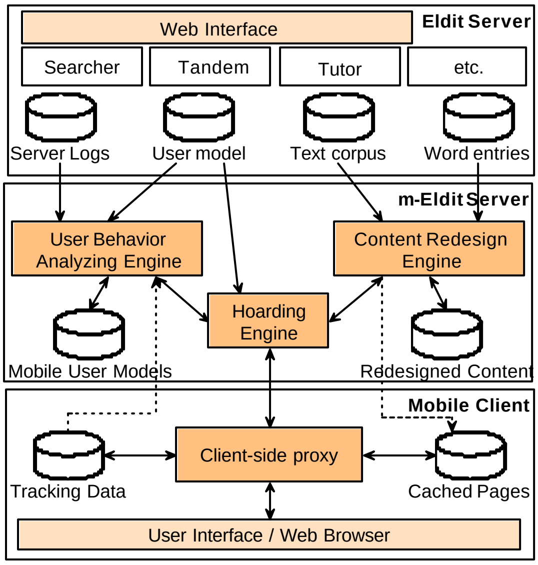


Figure 3: m-ELDIT architecture

```
GET
http://www.science.unitn.it/~foxy/mELDIT/Texts/General.056.html
HTTP/1.1
Accept: */*
UA-OS: Windows CE (POCKET PC) - Version 3.0
UA-color: color16
UA-pixels: 240x320
UA-CPU: ARM SA1110
UA-Voice: FALSE
UA-Language: JavaScript
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/2.0 (compatible; MSIE 3.02;
Windows CE; PPC; 240x320)
Host: www.science.unitn.it
Proxy-Connection: Keep-Alive
```

Figure 4: HTTP Request from a mobile device (iPAQ Pocket PC)

In addition a small proxy on the client device is responsible for receiving the browser requests and retrieving the content from the server or from the local store ('cached pages' in the figure above) when there is no connection available at the moment. The client-side proxy could also seamlessly upload the content that will be used in the future, based on the prediction done in the 'Hoarding Engine'. Uploading might be done on a special user request, where the user might also be given an option of setting different parameters, e.g. provisioned disconnection time, expected duration of time in which the system will be used offline, preferred by the user topic and etc. Different other options could be foreseen, like the proxy might be aware of the "cost" of the connection and behave in different ways according to that, i.e. synchronizing the cache when the 'cheap' connection is available (internet through LAN or cradle) and using only the cached content whenever possible on 'expensive' connections. It should take care not to return errors if a requested page is not in the cache, but some meaningful message, e.g. "The page is not available at the moment. It can be provided on the next synchronization". Another functionality of the proxy should be the tracking of the user activities and feedback to the mobile LMS. Thus the mobile user models should be aware of the user needs and adapt accordingly. The mobile LMS should be responsible for calculating and updating the user models, which will differ from the user models in a standard LMS.

User Modeling

The hoarding algorithm in our system has the role of deciding which texts the user would prefer to study during the next offline session(s) and upload them with the associated words. Every text consists of a small set of all the words in ELDIT. Even considering only the words that construct the text not all of them will be requested by the user. Moreover depending on the word the user might be interested in seeing only the translation of the word and example or might want more detailed information of it (as shown on figure 2).

User behavior observation and capturing of common access patterns will help predicting the objects (LO) to be needed next with a certain confidence. Both the comparative analysis on all the users of the system and the analysis of particular user behavior can be very important.

A possible modeling criterion might be the level of language knowledge of the user (i.e. beginner, intermediate, proficiency, etc.). In this way an analysis on what words are never viewed by users with similar language skills as the currently observed user would lead to excluding those words from the hoarding set and thus reducing the memory used. And also vice versa – if a word (used in a text) is viewed very often by users classified in the same category as the current user the algorithm should include it in the hoarding set with big priority.

One of the factors that might be considered in the user learning style analyses is the learner interest in the current material. The ELDIT text corpus contains texts about different subjects both for the Italian and German languages (e.g. Food and drinks; Economic and industry; Tourism, countries and cities; Art; Family; etc.). A useful measure for this might be the time spent on reading a concrete text page. If the student is interested he/she will probably spend more time on reading and reviewing unknown words. In such a way the users with common interests can be grouped.

Individual user models should store information about the learner and his/her behavior. The individual user models in ELDIT are a separate XML file (for every user) that contain information about user identification - login and password; log-in and log-out times; number of words reviewed; total number of clicks during the session; number of clicks onto the tabs (currently collocations, idiomatic expressions, derivations and compound words, and linguistic characteristics explained in footnotes). Recording and using information about user's native language, language interests, proficiency, learning style and other information is planned. Some of the data currently collected in the ELDIT user profiles might be useful for the analyses together with the web server log files, e.g. log-in and log-out times might help us connect sequences of requested pages (recorded in the log files) with particular users.

We should point out that the learning style of the user may change, depending on the task but it might also develop over time for the same task. Thus the individual students' learning styles should be handled in a flexible way. The recommendations should be taken as 'current' and dynamic changes should appear in the user model. The students might have different interests while exploring the same system (ELDIT) in mobile context, and while using a PDA for accessing the content they might develop different learning style. Thus the mobile LMS should keep separate user models.

Methodology of Evaluation of the System

There are different aspects in which such a system should be evaluated. One is connected with the presentation of such learning materials on PDAs and how convenient and handy users find such a system. In this aspect the evaluation might be done by questionnaire or oral interrogation with the users. Another aspect is the accuracy of the offline use supporting sub-system. As the decisions for the automated hoarding are taken based on analyses of server log files we can use the techniques applied in machine learning evaluations. Typically it consist of dividing a data set into a training set and a test set, using the former to learn the model, and the latter to evaluate the model's performance. This methodology has been applied to the predictive user models developed to date (Zukerman & Albrecht, 2001).

An often used metric in the evaluation of caching proxies is the hit ratio. Hit ratio is calculated by dividing the number of hits by the total number of uploaded predictions (cache size). In hoarding systems a more often used measure is the miss ratio - a percentage of accesses for which the cache is ineffective. Kuenning & Popek (1997) defined a 'miss cost' as a main difference in the evaluation of a caching and a hoarding system. In caching/prefetching systems the misses in the prediction reflect as a time penalty as the missing content should be retrieved from the web. This defers from the mobile case where with unavailable internet connection a miss in the hoard might be fatal. In order to quantify this measure Kuenning & Popek (1997) demand a user rating on every miss, using 4 different impact values:

0. The computer is completely unusable as a result of a miss. No future work can be done
1. The current task cannot proceed. Work can continue on a less desirable task
2. The current work will proceed, but the activity on that task will change in some way as a result of the hoard miss
3. The hoard miss will cause little or no trouble

They also define the 'time to first miss' - a simple count between the start of the disconnected operation and the first hoard miss. This evaluation criterion can be used only on real-use of a system (and its hoard part). It is strongly connected with the hoarding size. Another measurement is the 'miss-free hoard size', defined as the minimum amount of disc space that a particular hoarding system would require to allow a complete disconnection period to take place without any misses.

Related Work

There is a lot of ongoing work in the area of content adaptation for mobile devices and device independent representation of web content. In this context different approaches are proposed for describing device capabilities (HTTP Request Header, CC/PP, UAPROF, etc.). Also different architectural approaches are developed for using the information of devices' capabilities and adapting the content accordingly. The adaptation could be server-based (XML/XSLT, Cocoon, Axkit), proxy-based (AvantGo, Palm Web Clipping) and client-based (XHTML/CSS). A comprehensive review of the current device independence technologies and activities can be found in Butler (2001) or on the W3 Consortium web site (www.w3.org). Adapting the content through transcoding servers or proxies is one of the most often used techniques. Different transcoding techniques can be applied: for simply translating from one presentation language to another (e.g. WAP-HTML-WAP), for reducing the contents size, for satisfying bandwidth or screen capabilities of the devices, to adapt the structure of the content in a more appropriate way and etc. Joshi (2000) argues about the appropriateness of direct client-server architecture for mobile hosts accessing multimedia web content and the use of proxy-based architectures. He proposes a combined system as a better solution. As the content negotiation is one of the hot topics nowadays the author supposes that in the future more and more servers will take care for providing an adapted content for the concrete device. Meanwhile a proxy is needed to negotiate the content on the account of the device and/or to transcode the multimedia content if the server doesn't do it. The prototype system that is discussed there concentrates on the transcoding of images and video and supposes that the server will provide alternative versions for active components (e.g. CGI alternative for JavaScript functionalities), thus they don't provide a solution for transcoding this type of content. Moreover the author mentions that after this automated transcoding the resulted pages might be sometimes visually unpleasing. Although the author mentions the problem of synchronicity, which is important when mobile hosts are used, he does not consider an offline browsing. It is assumed that the disconnections are temporal and the user will not work offline but the system should be able to support the continuation of the work after reconnection from the same point where the disconnection appeared.

One of the things that drastically differ m-learning from e-learning is the offline usage. We believe that delivering content for offline usage is an important issue as still mobile devices are often disconnected because of the lack of access in certain places but also because of the high prices in most of the cases, thus our intention is to support both online and offline access to data. Operation in a disconnected or intermittent connection situation is a common problem in mobile computing. A disconnection should be distinguished from connection failure as it is either intentional or is foreseeable, thus some preparation can be done by the interested applications. Nevertheless only some of the transcoding proxies take care also for caching web pages for offline usage (e.g. AvantGo).

A similar problem to our one (off-line access to data) is treated in the offline browsing of web content. A quick review of the available offline browser utilities (like www.avantgo.com, www.httrack.com, www.webstripper.net, etc.) shows that generally during the online periods the user selects sites that should be uploaded for later off-line usage and entire sites are dumped to the local storage or the user specifies the depth of the links to be cached. In situation where mobile devices are considered the memory limitations should be taken into account and thus it is preferable that only the content that will be needed during the next usage of the system is downloaded.

The preliminary downloading of the data that will be needed in the future, called caching, perfetching or hoarding is often considered in the Internet world. Wang (1999) has presented a survey of the state-of-the-art techniques and elements of Web caching systems. These techniques include Prediction-by-Partial-Matching, analyses of users' access patterns, provided by the servers, prediction of the user's future Web accesses by analyzing his or her past Web accesses, etc. Although some of these techniques are useful for predicting the content needed also in m-learning domain still they aim at different goals – reduction of bandwidth consumption, of access latency, server workload and etc. They explore the case of the Web where the search space is much bigger, the users are numerous and have different interests thus the prediction accuracy is quite low comparing to what is needed in our scenario, but could be compensated by the fact that the intemet connection is permanent. The learning scenario has characteristics that expose some additional information to be considered and thereby possibility to improve the existing solutions. In our opinion a more efficient strategy can be defined for hoarding by taking advantage of the peculiarity of the m-learning scenario and the semantic data that is kept in the LMS.

The idea of hoarding for disconnected devices in distributed file systems has been first described by Kistler & Satyanarayanan (1992), although they do not consider mobile devices. They propose the Coda File System to explore the use of caching of data not for improving performance but for increasing the availability. They propose architecture for hoarding and for keeping the coherence of the utilized files. The initial system was based on client-server architecture which tracks the local file modifications and saves a 'Client Modification Log'. The project has lately evolved into UbiData project (Helal et al. 2002, Zhang et al. 2003) and the direction it took is in double-middleware architecture for ubiquitous data (file) access. They introduce incremental hoarding, where the idea is to use a version control system to maintain object differences and they also study the automatic data selection problem. A metadata server is included to store the 'users' mobile profile' which keeps a list of user files that are considered 'interesting'. They define a "hybrid priority" metric for choosing the hoarding set of files. The "hybrid priority" is calculated by taking into account the recency of use, the frequency of access and the active periods of the file use and the algorithm also considers upper space limit of memory. The reported effectiveness of their filtering algorithm is more than 84% (Zhang et al. 2003).

Facing the hoarding problem for mobile computing in disconnected mode an interesting solution has been proposed in SEER (Kuenning & Popek 1997; Kuenning et al. 1997). The authors were also inspired by the work on the Coda system but go in different direction. They defined a new measure, "semantic distance", between individual files by observing the user activities and propose an algorithm for automatic hoarding of projects for mobile computers. With "semantic distance" the author tries to quantify the user's intuition about the relationship between files in the same project. For this different measuring criteria are used – " temporal semantic distance", "sequence-based semantic distance", "lifetime semantic distance", directory membership, filename conventions and hot links, and are combined to assign weights to documents and take decisions for hoarding them in an automatic way (automatic periodic hoarding). The approach met some unpredictable behavior in the real-world system, which appeared because of the way the operating systems and some often used programs work (like the "find" operation under Unix). Recent experimentations with the same system (Kuenning et al. 2002) showed surprising findings – the complex clustering methods that are used in the system work in most of the cases worse then a LRU (least recently used) algorithm enhanced with some heuristics. This shows us that the research field is still open for work.

Conclusions

In the paper we had presented a mobile language learning system, called Mobile ELDIT (m-ELDIT), which aims to support people on the move, interested in taking the bilingualism exams in South Tyrol. In a previous paper (Trifonova & Ronchetti 2003b) we have discussed that mobile learning in general applies best to activities with certain characteristics: (1) short, for filling the gaps of waiting time; (2) simple and with added value; (3) delivered just in time/place. Language learning fits very well in this frame. We are building our mobile learning system on top of an existing adaptive language learning platform, thus the users will have the possibility to work both from their desktop PC and while away from their homes and offices on their PDAs. We have pointed out that m-ELDIT should take care for redesigning the requested materials for the specific characteristics of the device used and also predict and upload the content that will be used during the offline periods. We have proposed an architecture for the system, which consists of a small module on the client device (client-side proxy) and three modules on the mobile server side – "Content Redesign Engine", "User Behavior Analyzing Engine" and "Hoarding Engine". The hoarding algorithm should take advantage of the user models from the ELDIT system, but also separate analyses are needed for the access patterns to the system from mobile devices.

References

Butler M. (2001). Current Technologies for Device Independence: *HP Labs Technical Reports HPL-2001-83*

Gamper J., Knapp J. (2003). A Data Model and its Implementation for a Web-Based Language Learning System. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*

Helal A., Khushraj A., Zhang J. (2002). Incremental Hoarding and Reintegration in Mobile Environments. In *Proceedings of Symposium on Applications and the Internet (SAINT)*

Joshi A. (2000). On proxy agents, mobility, and web access. *Mobile Networks and Applications,* 5 (2000) p.233–241

Kistler J., Satyanarayanan M. (1992). Disconnected Operation in the Coda File System, *ACM Transactions on Computer Systems*, Vol. 10, No. 1

Kuenning G., Popek G. (1997). Automated Hoarding for Mobile Computers. In *Proceedings of the 16th ACM Symposium on Operating Systems Principles (SOSP-16),* San-Malo, France, pp.264-275

Kuenning G., Reiher P., Ma W., Popek G. (2002) Simplifying Automated Hoarding Methods. In *Proceedings of 5th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (WiM'02),* Atlanta, Georgia, USA, pp.15-21

Kuenning G., Reiher P., Popek G. (1997). Experience with an Automated Hoarding System. *Personal Technologies*, 1, 3, September 1997, pp.145-155

Trifonova A., Ronchetti M. (2003a). A General Architecture for M-Learning. In *Proceedings of the Second International Conference on Multimedia and ICTs in Education (m-ICTE 2003),* Badajoz, Spain

Trifonova A., Ronchetti M. (2003b). Where is Mobile Learning Going?. In *Proceedings of E-Learn 2003 Conference*, Phoenix, Arizona, USA

Wang J. (1999). A Survey of Web Caching Schemes for the Internet. *ACM Computer Communication Review*, 25(9)

Zhang J, Helal A., Hammer J. (2003). UbiData: Ubiquitous Mobile File Service. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, Melbourne, Florida

Zukerman, I., Albrecht D. W. (2001). Predictive Statistical Models for User Modeling. *User Modeling and User-Adapted Interaction,* 11, pp. 5-18