Hoarding Content in an M-Learning System

Anna Trifonova and Marco Ronchetti

Department of Information and Communication Technologies, University of Trento
Via Sommarive 14 – 38050 Povo (Trento) – Italy
Tel. +39-0461882033; Fax +39-0461882093;
E-mail: {Anna.Trifonova, Marco.Ronchetti}@dit.unitn.it

Abstract: The advances in mobile technologies predispose the support of learners' and teachers' activities on the move. As the nowadays' infrastructure still does not provide cheap 'always on' connection at every place and every moment, a mobile learning system should provide also support for offline usage of learning material. The process of automatically selecting and pre-fetching content to be used during the disconnected periods is called hoarding. Considering a situation where the mobile device's limited memory can not hold all available materials, the aim of the hoarding is to predict what the user will really need and use. A high accuracy of the set selection is necessary, as in the learning scenario every miss can lead to student's misunderstanding of the studied subject. Hoarding is still not well explored. We are currently developing a mobile version of an existing language learning online platform. Our Mobile ELDIT system strongly needs a hoarding subsystem. We have developed a basic hoarding algorithm, whose preliminary outcomes we report and we want to study the parameters that could help a hoarding algorithm improvement in order to cover the peculiarity in m-learning scenario. Our goal is to provide an efficient strategy, taking into account additional parameters automatically extracted by the system.

Introduction

E-learning is growing very fast and many Universities and companies are supporting in some way an e-learning solution. It is now clear that the WWW is a very successful educational medium. On the other hand the rush in the field of wireless and mobile technologies creates opportunity for new field of research - 'mobile learning'. This domain can include a wide variety of applications and new teaching and learning techniques (Trifonova et al. 03). In their tries of finding the best way to apply mobile devices in education people are experimenting with different fields. Courses modules were created throughout different projects for people with numeracy and literacy problems, for kids, university students, for teachers, for studying computer science subjects, psychology or language learning.

We analyzed different ways to apply mobile devices for educational purposes. This led us to classifying services that are specific and should be provided by a general m-learning platform (Trifonova et al. 04) and later we concentrated on one of these services as a concrete problem to solve. Namely this is the hoarding of content for offline usage.

Hoarding is a technique for selecting a set of documents to be uploaded and used *when disconnected*. Related terms are caching and pre-fetching, though they are used when considering the general WWW case (e.g. 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. Pre-fetching on the other hand is a technique which tries to guess what will be needed to the client in the near future, download it in advance and cache it to improve the clients' experience. Different schemes of caching and pre-fetching were proposed in the WWW world (see Wang 99, Barish 00, Hassanein et al. 02). The goals were to reduce network traffic and to minimize access latency, bottlenecks, servers' workload and etc. Though the goal of hoarding is somehow different, some of the techniques used in the web context can be reused, although hoarding requires a much higher accuracy and presents the added limitation of the memory availability. The learning scenario has characteristics that expose some additional information to be considered and thereby possibility to improve the existing solutions. Further in the paper we discuss the general hoarding process and give some preliminary outcomes of our work up to now.

Mobile ELDIT and its Need for Hoarding

At the University of Trento we are developing a mobile version of an existing online language learning system, ELDIT (http://www.eurac.edu/Eldit). The ELDIT platform (Gamper & Knapp 03) is mainly developed to help people of the South Tyrol region in Italy to prepare for the exams of bilingualism, needed by everybody who wants to work in the public sector. Nevertheless it can be used by anybody interested in learning German or Italian language. Thus the system is to be used by a self-motivated learner, who doesn't need any supervisory control of studying process. ELDIT contains a big quantity of learning materials, including learner's dictionary and text corpus – total of about 1GB of raw data.

For the mobile ELDIT we wanted to give to the users the possibility to access the data of ELDIT from mobile device (PDAs). These devices commonly have a small touch-screen, no keyboard and a very limited memory (32-64MB). Our Mobile ELDIT offers to the users access to ELDIT text corpus - about 800 texts and associated words in both German and Italian languages. Each text has about 150 words and few comprehension questions that the user should answer in the other language. Nouns, verbs and adjectives are linked to word entries with rich explanations, translations and additional data. The texts are divided into two difficulty levels and are split into thematic groups. More technically the Mobile ELDIT consists of different modules. One of them takes care for redesigning the content into proper for the small screen and the limited web browser format. Another module, which we'll discuss in details further in the paper, takes care for predicting the learner's future needs and for the packaging of the study material for offline usage. As mentioned before this process is called hoarding and is needed because the mobile device can not hold all the data of the e-learning platform. The device has intermittent connection, sometimes because of the lack of infrastructure, sometimes because of the connection costs, yet we want the user to be able to access part of the study material during the offline periods. As we want to free the user from annoying procedures of pre-fetching content manually we need to decide automatically what the user will need. We suppose that the device will be regularly synchronized with the main system, thus that new portion of the material can be made available on the next usage. We have developed the basic hoarding system and we are collecting usage data for experimenting with different parameters of the hoarding algorithm, in order to improve its work.

The Hoarding Process

Generally the pages in a web based material (and in particular on any web-based learning source) are highly interlinked (as shown on Figure 1). On the other hand the user browsing path over a web-based material can be viewed as a much simplified hierarchy structure (a sub-tree or directed graph). The user follows the links (the edges on the picture) from one page to another, or can go back to a previously viewed page (see Figure 2).

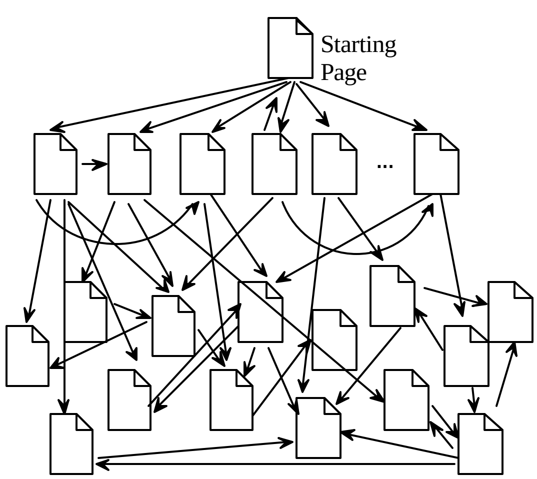

Figure 1: *Web-based material structure*

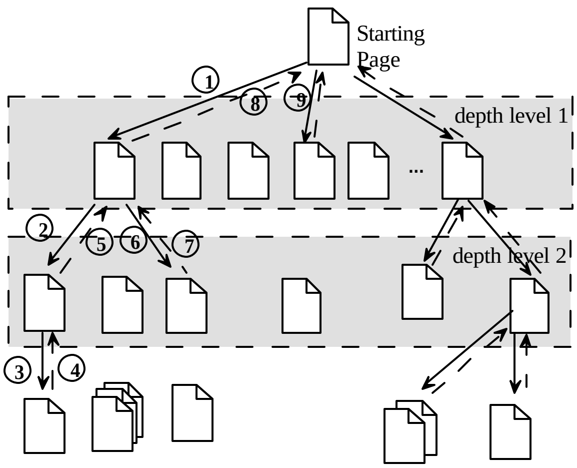

Figure 2: *Browsing path over a web-based material*

The hoarding process consists of few steps:

1) Decide (predict) what material the learner will start from;
2) From the materials that are linked to this 'start point' generate a 'candidate set;
3) Predict which will be the learning sequence or set, that is of interest for the user at next depth level and discard the items that are not of interest from the set (this step is called pruning) – loop until the deepest level;
4) Set priorities to the materials in the hoarding set, based on their predicted priority of importance for the next session (higher if the probability the object will be used soon is high);
5) Pre-fetch, starting from the beginning of the ordered list, putting on the device those objects with bigger priority, until available memory is filled in.

We can see that the algorithm will strongly depend on system's knowledge about the user. This knowledge includes user's learning style, personal learning habits and abilities, the level of expertise in the studying field and topic. This knowledge about the user can be acquired in different ways – by direct assessment of the user, by questionnaires and quizzes, but also by observing and analyzing the user behavior during the studying with the system, thus automatically discovering user's learning style, preferences, acquired knowledge and etc.

For making the things clear we can consider two separate engines. One will deal with observing the user and creating user models and the other for the hoarding. We call the first one 'User Behavior Analyzing Engine' and it is discussed in (Trifonova et al. 05b). The hoarding algorithm 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 (domain knowledge).

Some questions appear on this stage:

- What is the best starting point for the user's next session?
- What is a 'session' in the mobile learning scenario?
- How can we predict the most probable learning path (sequence of LO)?
- How do we create (formalize) a useful user model?
- What are the important parameters of the user behavior which have influence on the prediction?
- How do we use different parameters of the user model for predicting and/or pruning and do these different parameters have different significance for the prediction and/or pruning process?
- How do we formalize the domain knowledge?
- Do different domain knowledge parameters have different significance for the prediction and/or pruning?
- How do we measure the successfulness of the automatic hoarding and how do we improve the work of the algorithms, considering these measures?

The rest of the paper attempts to answer some of these questions and gives only initial guidelines for others.

Creating the Candidate Set

As mentioned earlier one of the steps of the hoarding algorithm is to construct the 'candidate' set of objects, related (linked) to the starting point or to other objects that were predicted to be viewed. When using a web-based material the user clicks on the links of one page to go to another one and can either continue to browse further or can go back to a previously viewed page. The links between the pages give us the structure of the web site (a learning material in particular), thus we can extract the relation between the Learning Objects (LO), for example by parsing the pages.

```
for (every LO) {
 create a row;
 for (i=1, number_of_LO, i++) {
  if current_LO contains link to LOi
    set celli = 1;
  else set celli=0;
 }
}
```

***Listing 1**: Creating links between LO matrix*

	LO_1	LO_2	LO_3	…	LO_n
LO_1	x	1	0		1
LO_2	0	x	1		1
LO_3	1	1	x		0
…				x	
LO_n	1	0	1		x

***Table 1**: The data matrix – links between LO*

The links might be either bi-directional or not. We can build a LO correlation table in the way shown in Listing 1.

In Table 1 we can see that LO_1 contains link to LO_2 and to LO_n , but not to LO_3. The link is bi-directional for LO_2 and LO_3. In this way we can easily observe the set of LO that the user will be possibly requesting if he/she decides to browse deeper in the site, i.e. to go one level of depth further. Those are the objects directly linked to a particular object. From the table above (Table 1) we can easily construct the 'candidate' set of LO for every next level of hoarding. Later this candidate set will be pruned (its size can be decreased by dropping some of the objects that are not likely to be requested). Based on observations on the users (either the particular user or all other users) the system can estimate the average (or max.) browsing depth and session length, which will be the stop point for creating the candidate set.

Pruning

Pruning is the step where the candidate set should be freed from the items that the user will not need. In some cases it might be possible to predict which will be the learning sequence or set, that is of interest for the user at next depth level and discard the items that are not of interest from the set. In other cases it might be more meaningful to predict what is *not* of interest directly. Most often these will be the parts of the learning material that the user is familiar with. The system can try to detect the user expertise level on the study topic (by questionnaire for example) and to narrow the hoarding set using some domain knowledge.

In the context of hoarding the content on the first user access the system should upload as much as possible data trying to satisfy all user's requests (as shown on Figure 3). On every next step the hoarding algorithm will try to tighten the hoarding set, while keeping a small error rate. The goal of the hoarding algorithm is to maximize the 'hit rate' and at the same time to minimize the 'miss rate'. In other words the ideal situation is to achieve hit_rate=100% and miss_rate=0%, which would mean that the hoarding set contains *all* and *only* the items needed by the user during her/his next studying session.

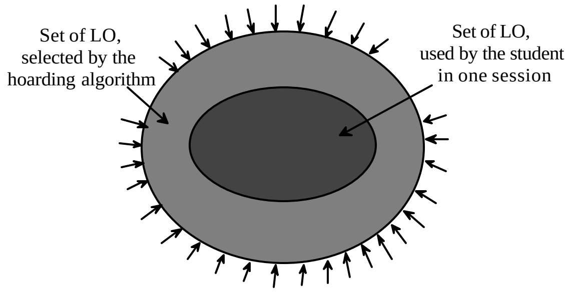

Figure 3: *The hoarding starting step*

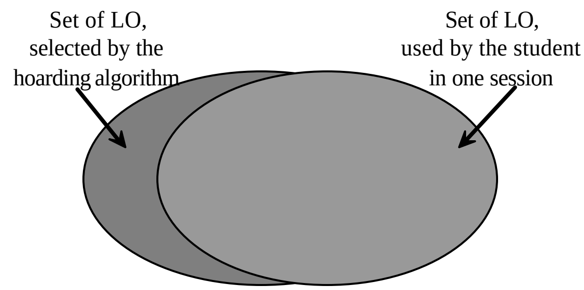

Figure 4: *The expected picture*

Though the ideal picture is to have exact overlapping between the predicted set and the requested from the learner set in a real system such a situation is almost impossible to reach. Most probably we will have some (desirably big) overlap between the cached by the hoarding algorithm set and the one really requested by the learner (see Figure 4). Reference for different metrics to measure the goodness of the hoarding generally in mobile computing systems can be found in Kuenning & Popec '97.

User Modeling

In the literature one can find lots of different ways to model a user and/or her behavior depending on the application and its needs. In the context of hoarding we recognize two groups of characteristics that will be used differently by the algorithm. We schematically call the first 'user behavior' and the second 'user knowledge'. Depending on the mobile learning system it is possible that not all the parameters can be discovered or that they might be discovered through different techniques. The data about the user might be obtained by (any combination of) questionnaires, tests and quizzes or automatically by tracking the user and analyzing the log files. The process for retrieving automatically the user patterns consists of few steps, shown in the figure below. The first step is the preparation of the data for analysis For this step the log files should be preprocessed and integrated into a database. Afterwards different data mining algorithms can be applied to extract interesting relations.

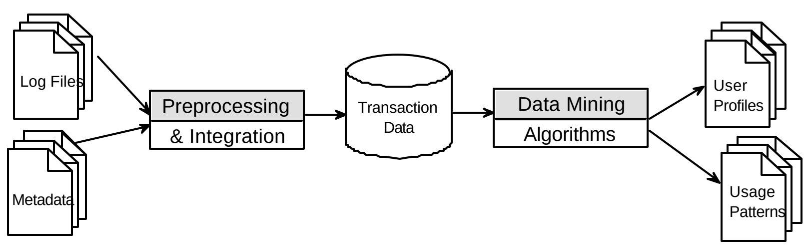

Figure 5: *Architecture for deriving user profiles*

The *user behavior* can be described in terms of browsing styles (e.g. consecutive, random, interest driven, etc.), preferred type of educational media (e.g. prefers video to combination of text and pictures) etc. Based on the user behavior we can group the learners and analyze the similarities and differences between the groups and between the members of the same group. This should help us to predict what will be needed, i.e. this data will be used to fill-in the hoarding set.

On the other hand the *user knowledge* profile should consist of everything that the system knows about what the user already knows. Examples are the system awareness of the user's competence in a certain subject (i.e. beginner, intermediate, advanced) or a list of all the topics already covered by the user previously. In contrast of the *user behavior* the profile of the *user knowledge* will be used for pruning the entries from the hoarding set, i.e. for excluding objects in order to decrease the size of the hoard.

We can distinguish static data about the user and dynamically changing data. The static data is for example the user age, gender, mother tongue and etc. On the other hand the dynamic data is our current knowledge about the evolving user parameters and should be reviewed in certain periods of time. For example the user browsing pattern might change drastically few days before an exam date, thus the hoarding system should be able to quickly recognize such changes and react accordingly.

First Hoarding Results

We did some initial experiments in our Mobile ELDIT system with the aim to explore a basic hoarding system and see if the hoarding will really work. In our experiments the usage data was collected in the following way: as a first step the users were given a set of ELDIT texts and were asked to study them at their convenience. Whenever the users felt that they were 'finished' with this portion of texts they were given another set. Part of the users were preparing for the bilingualism exam, others were just studying the language, without aiming at passing the exam. Only in certain cases the users were given the option to choose the texts they would like to read. Some interesting and important outcomes from those experiments, not directly related with the hoarding process, can be found in (Trifonova et al. 05a).

For obtaining the first hoarding results we observed only one user at a time. We were selecting all the words that were accessible from a text and then we did pruning, based on what we think the user already knows. We assumed that the user knew all the words that were present in a previous text, and whose links were not followed. So on every next iteration the hoarding set was smaller (see Figure 6). On the figure the blue dots show the real user requests, the red line shows how the hoard decrease with every next text that the user was reading. The yellow triangle show the miss rate (calculated as the percentage of accesses for which the cache was ineffective) if it is not zero.

First of all, the graphic on Figure 6 shows that hoarding process itself works. We have used a rather simple rule for pruning – what we consider that the user knows. As a next step we should refine the algorithm and to improve it in two directions:

1) To make the hoard decrease faster and
2) To assure that the algorithm works more precise.

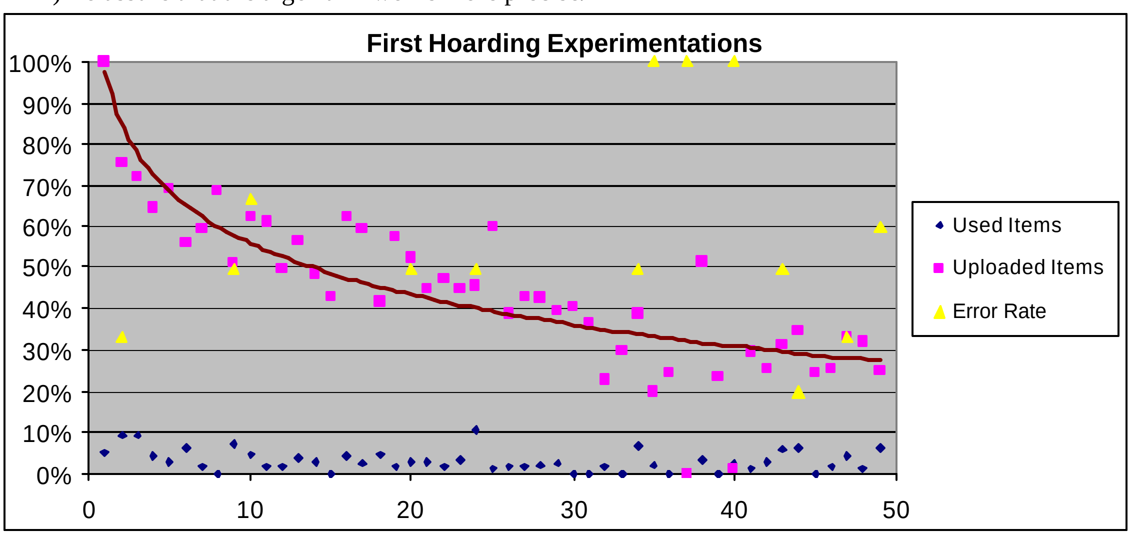

Figure 6: *Example data showing the decreasing of the hoarding set*

To make the hoard decrease faster (i.e. in fewer steps and with bigger values) we should combine the knowledge gathered from other users' usage data. This means that we have to analyze the similarity between the users. For similar users (for example similar in their proficiency on the studied subject) we can guess that a user knows certain word, based on our awareness that the other similar user (or users in a group) knows it. In contrast with this first experiment, where we considered a word familiar for the learner only when he already had the possibility to see, it in the second step we will try to guess in advance.

One the other hand one can see that in some cases we have a big (sometimes 100%) miss rate. We have mentioned earlier that for the mobile learning scenario the accuracy is very important. This was also proven by a questionnaire that our first users filled-in, where almost 100% mention hoarding misses as the most disturbing problem of the mobile system. A miss in the hoarding might lead to termination of the study process or even worse to misunderstanding of the material. We have to assure that the algorithm works more precise.

One of the main reasons for errors of the hoarding algorithm is the simplicity of the pruning rule that we have used in the current experiment. In the cases that we have obtained 100% miss rate the reason was that the user had requested a text and without reading it pressed the back button and continued with other material. This misled the algorithm to 'conclude' that the user knows all the words that were provided in the text. This led to excluding those words from further including in the hoarding set. Later on the user requested the same text again, this time really reading it. As the algorithm already had decided that all words are known to the user and excluded them, every word requested was missing. This problem can be solved by monitoring the time spent on a page, so as to be able to infer whether the page was actually read. On Fig. 7 one can see that in Mobile ELDIT a common time needed for reading a text is more then 3 minutes, so if the time spent on a text is less then 180 sec. the user most probably did not really read it.

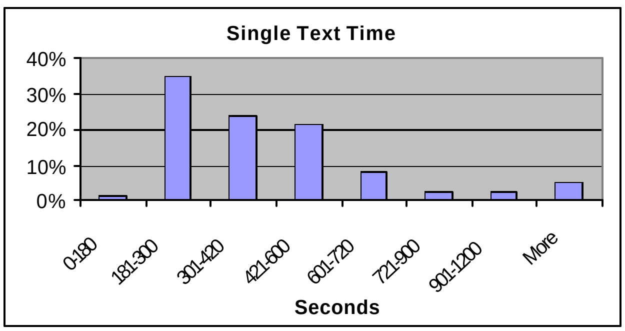

Figure 7: *Time spent on single text*

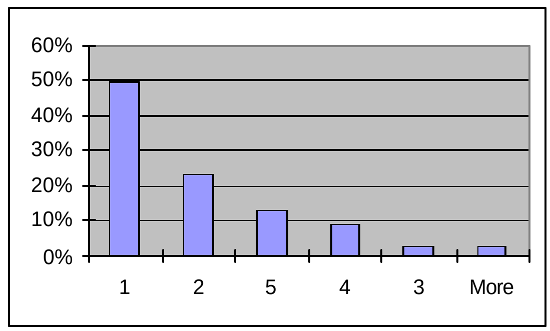

Figure 8: *Number of sessions in one day*

Another observation considering the errors of the hoarding algorithm is the existence of LOs that are critical for a certain set. In our case often the critical LOs are the verbs in the texts. In our system every verb from a text is connected to its infinitive entry, thus even if different conjugations exist they share the same 'link'. Because of this when the user reads a text it is very possible to see a form of a verb that he/she is familiar with. This hoarding algorithm puts this verb in the 'User_Knowledge' set and on every next step excludes this verb from the hoarding set. It is also possible (and even occurs often) that a certain conjugation is particularly difficult and is not familiar to the users. This leads to a request, that can not be satisfied - a hoarding-miss. Another example is the existence of different derivations of the same word that are also connected to the main entry. Derivations are also critical points in the pruning process. In our later experiments we have introduced the idea of 'Critical set', where we keep track of LOs that are particularly difficult in certain texts. We hope that the use of 'Critical' Sets will have a strong influence on the algorithm accuracy.

Another interesting and important observation in our first experiments is that the term session in the context of hoarding should be redefined.

In the Internet world a session is defined as "a continuous period of time during which a user's browser is viewing Web pages or a Web application within the same server or domain" (source - MSDN Library, *http://msdn.microsoft.com/library/default.asp?url=/library/en-us/passport25/NET_Passport_Common_Documentation/glossary.asp*). It consists of a series of transactions or clicks on the web pages links made by a single user. There are different criteria to decide whether a session is over or not. The most commonly used one is the inactivity period of the user, where if the user activity stops for a certain period of time (generally 30 minutes) on the resumption of the activity by the same user a new session is considered started.

On the other hand for hoarding in a mobile system the importance falls on the time between two possibilities of the user to synchronize with the main server. In this sense we find more useful to define a hoarding-session as *the time between two synchronizations* of the mobile device with the main online system. The default session length might be one day (a daily-session), as often the synchronization is done once per day. During the system usage other session length might be observed and explicitly set for every user. On figure 8 we show the number of sessions (defined as in the general Internet case as series of user clicks with inactivity period less then 30 min.) that take place in the same day. One can see that in about half of the cases the users were using the system more than once in one day, which confirms the necessity of specific definition of the session length in the 'disconnected' mobile scenario.

One of the particularities that mobile learning offers is that mobile devices are certainly personal devices (used only by their owners) it is possible to securely identify the user. Also the users of a learning platform are comparably smaller than the WWW users and thus would be possible to adapt the hoarding algorithm work to every user by analyzing previous successfulness.

Current and Future Work

As we mentioned in the previous section there are few directions for improving the basic hoarding algorithm we have presented. A first direction is to improve the pruning in terms of faster and stronger tightening of the hoarding set. The second direction is in the one of increasing the accuracy, by carefully analyzing the reasons for the pruning errors. On the next step we are making the algorithm more sophisticated by adding simultaneous analysis of more than one user at a time – the outcomes of the 'User Analyzing Engine'. This includes experiments with different data mining algorithms and their parameters for finding similarities between users, correlations between objects, often requested sequences and etc. We have also added the idea of 'Critical Sets' of objects. In our opinion one of the most important things for successful hoarding is the analysis of the users' behavior and the extraction of valid rules for pruning and prioritizing the objects of the hoarding set. For this we need a lot of tracking information that we are currently gathering.

A working demo version of the Mobile ELDIT system which does not contain the hoarding sub-system is available for free download on http://www.science.unitn.it/~foxy/MobileEldit.php.

Conclusions

In this paper we described the hoarding problem for a mobile user *without* Internet connection. The problem is how to support the study process on a mobile device when it is impossible to load in its limited memory all the data that comprise the full knowledge base.

We have outlined a general algorithm, and we have posed a number of questions that need to be answered in order to solve the problem. Though our work is still in progress we have done some important deductions that can also be a starting point for other developers in the field.

We have attempted to give some first answers to these questions. We have drawn attention to particularities of the mobile learning scenario that differ from scenarios considered previously:

- Sessions – we have emphasized that there is a difference between a 'session' in the Internet word and what should be considered a session in this particular scenario – hoarding content for mobile learning.
- Tuning of the algorithm – because of the certainty in identifying the user, as the mobile devices are very personal, the algorithm should adapt its work to the needs of every user
- Effective pruning – one of the important pruning criteria is the analysis of what the user could but did not access in his/her previous sessions. While this technique can not be used in the general case of internet caching and pre-fetching it is a great source of information for pruning in the mobile learning case.

References

Barish G., Obraczka K. (2000). World Wide Web Caching: Trends and Techniques. *IEEE Communications.*May 2000.

Gamper J., Knapp J., (2003). A Data Model and its Implementation for a Web-Based Language Learning System. *Proc. of Twelfth International World Wide Web Conference (WWW2003),* Budapest, Hungary, May 20-24, 2003

Hassanein, H. Zhengang Liang Martin, P. (2002). Performance comparison of alternative Web caching techniques. *Proceedings of Seventh International Symposium on Computers and Communications (ISCC 2002)*.

Kuenning G. H., Popek G. J., (1997). Automated Hoarding for Mobile Computers, *Proc. of 16th ACM Symposium on Operating Systems Principles*, St. Malo, France, pp.264-275.

Trifonova A., Ronchetti M. (2003). Where is Mobile Learning Going?. *In Proc. The World Conference on E-learning in Corporate, Government, Healthcare, & Higher Education (E-Learn 2003)*, Phoenix, Arizona, USA, November 7-11, 2003.

Trifonova A., Ronchetti M. (2004). A General Architecture to Support Mobility in Learning. In *proc. of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, August 30 - September 1, 2004, Joensuu, Finland.

Trifonova A., Ronchetti M. (2005a). Prepare for Bilingualism Exam with a PDA in your hands. *Proceedings of the International Conference on "Methods and Technologies for Learning" (ICMTL 2005)*, March 9 - 11, 2005, Palermo, Italy.

Trifonova A., Ronchetti M. (2005b). User Behavior Observations for Supporting Offline Delivering of Learning Materials in a Mobile System. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-Media 2005)*, June 27-July 2, 2005, Montreal, Canada

Wang, J. (1999). A Survey of Web Caching Schemes for the Internet. ACM *Computer Communication Review, 25 (9)*, 1999, Page(s): 36-46