

---

## Hoarding content for mobile learning

---

Anna Trifonova\* and Marco Ronchetti

Department of Information and Communication Technologies

University of Trento, Italy

38050 via Sommarive 14, Povo (TN), Italy

Fax: +39-0461-882093

E-mail: Anna.Trifonova@dit.unitn.it

E-mail: Marco.Ronchetti@dit.unitn.it

\*Corresponding author

**Abstract:** M-learning is a rapidly expanding domain recently. Provoked by the fast advances of mobile technologies, different applications and systems are developed continuously. Here we address the hoarding problem, which is weakly explored before but is a particularly important issue in the mobile domain, and a solution should be included in every system with a large quantity of data. Hoarding is the process of automatically selecting learning content, which is to be prepared and prefetched on the mobile device's local memory for the following offline session. We describe the hoarding problem and the strategy to solve it with the goal of providing an efficient hoarding solution.

**Keywords:** mobile learning; hoarding; offline access to learning content; disconnected operation; caching; prefetching; general approach techniques.

**Reference** to this paper should be made as follows: Trifonova, A. and Ronchetti, M. (2006) 'Hoarding content for mobile learning', *Int. J. Mobile Communications*, Vol. 4, No. 4, pp.459-476.

**Biographical notes:** Anna Trifonova graduated in 1999 from the New Bulgarian University (Sofia, Bulgaria), with a major in Information Systems and Technologies – applications in business and office. Currently, she is a PhD student at the International Graduate School of Information and Communication Technologies at the University of Trento, Italy. Her research topic is 'Mobile learning: wireless and mobile technologies in education'. Her scientific interests and publications are mainly in the mobile learning domain. She has published more than ten articles in international peer-reviewed conferences and workshops on this topic since 2003.

Marco Ronchetti graduated with a degree in Physics in 1979. After a post-doctoral position at the IBM-T.J.Watson Research Center, USA, and after complying with the Military Service duty, he became a researcher in theoretical physics (Statistical Physics-Condensed Matter) at the University of Trento. In the mid-1990s his interest drifted towards computer science, initially in the web and software engineering areas, and more recently on e-learning. In 2001 he became a Rector's Delegate for the web and the online services, and the department's delegate for e-learning. He has 30 publications in physics and more than 40 in computer science.

## 1 Introduction

Wireless and mobile technologies have been developing very fast over the last few years. New devices and technological solutions appear on the market with great speed, and the research and development communities are trying to find the best possible ways to use them. Small, relatively inexpensive devices like Personal Digital Assistant (PDA) and smartphones enable computational and data access whilst on the move. As a consequence, mobile applications are appearing in different fields, like commerce (Andreou *et al.*, 2005), healthcare (Liang *et al.*, 2003; Lin and Vassar, 2004), tourism (Garzotto *et al.*, 2004), *etc.*

In the learning domain a whole new field is opening, called mobile learning or, in short, m-learning (Trifonova and Ronchetti, 2003). It has been considered as the next step in distance learning, which will be an integral part of any form of educational process in the future. Mobile learning can cover a wide range of applications, educational fields, pedagogical approaches, and technological solutions. The common criterion for entering in the mobile learning domain is to use a mobile computational device in some teaching and/or studying activities or education supporting services. As the m-learning domain is explored only in recent years, many new research topics are emerging in various areas, including technological issues, pedagogical and methodological ones, problems related to content, and user interface adaptation, *etc.*

The problem we focus on and what we describe in this paper is the one of supporting the access to web-based learning content from a PDA device during its periods of disconnection. Such offline periods may appear for different reasons – intentional (*e.g.*, the available connection is too expensive for the user) or unintentional (*e.g.*, lack of infrastructure at a given time and location). During offline periods the user can only access materials located on the device's local memory. Mobile systems typically have a relatively small amount of memory, which is often not enough to store all the needed study materials. In such a case, a decision should be taken on which part of the material has to be cached. Often we cannot count on the user's own judgement of what he/she will need and prefetch. Rather, in our opinion, some sort of automatic prefetching would be desirable. The process of automatic selection and caching of material to be used during offline periods is called 'hoarding'.

The paper is organised as follows: in the next section we describe the research context and the motivations for our work. Afterwards, we present in more details the hoarding process and give description of how the different steps should be taken. A brief review of related works is followed by conclusions and references.

## 2 The research context

### 2.1 A real-world system

After analysing various suggestions from researchers in the field of mobile learning (Trifonova and Ronchetti, 2003), we designed and developed a system according to such findings, so that it should be useful and engaging to work with. The system, called Mobile ELDIT, is a version of ELDIT (Gamper and Knapp, 2003), a system for online learning of the German and Italian languages. Mobile ELDIT allows accessing a subset of the ELDIT learning materials from mobile devices (namely PDAs).

ELDIT is designed according to the principle of separation between data and their presentation. The data are XML-formatted<sup>1</sup> and the learning content is very low-granulated. The system consists of two main streams – dictionary and texts corpus. The text corpus is the part that is later adapted to be used via mobile devices. Every text is made of about 150 words and additional comprehension questions. Words are connected to their entry in the dictionary. On the other hand, each word entry contains explanations, translations, and examples on different senses. It also provides additional information, like idiomatic expressions, derivations from the word, *etc.* The online system contains more than 600MB of raw data, which is an order of magnitude larger than the typically available PDA memory. Moreover, such data are continuously growing as the ELDIT system evolves and the data are being enriched over time. It is therefore obvious that the mobile version strongly needs a hoarding subsystem.

Mobile ELDIT system contains a server part and an on-device part. The server side has the important functionality of adapting the content to the PDA by rendering it into proper format for the device screen and displaying limitations. We have decided to use the XML-formatted data to generate web (HTML) content, which is displayed to the user via a web browser on the mobile device. This decision was taken mainly because we would like to keep the users' perception of the mobile version as close as possible to the online desktop ELDIT, and also because web browsing is very familiar to almost every user; therefore, it is not necessary to learn yet another user interface, and new users can easily and quickly start using the mobile system. The second functionality on the server side is the one of analysing the user behaviour and preparing content for the offline periods – the hoarding subsystem described in more details further in the paper. The device-side part of mobile ELDIT is a caching proxy, which has to respond to browser requests during disconnected periods by providing the pages that are already in the cache. The cache is filled-in on connection with the predicted by the hoarding subsystem set of Learning Objects (LO). (Note: in the paper we use the term learning object for referring to learning units and more concrete separate HTML pages in the mobile ELDIT system. Nevertheless, it might be any digital chunk of learning content that is in some way connected to the other parts.) Another important responsibility of the client-side-proxy is to keep track (in log files) of users' requests of material. Basically, this is a list of all requested LO together with time information. These log files are the main source of information for analysing the user behaviour for the hoarding purposes.

## 2.2 *A generalised solution*

Our bottom-up approach to hoarding starts from the special case of a real-world system, and is based on a set of general principles described in Trifonova and Ronchetti (2004). Our ultimate goal, however, is to provide a general strategy that can be used also in different systems. Ideas on the possible approaches, guidelines on what algorithms are appropriate in what cases, and analysis on how different parameters that emerge from our work should be tuned by researchers and developers of mobile learning solutions who also want to automatically decide what part of the learning content will the user need in the next offline period and do the caching.

### 3 The hoarding process

Hoarding in the learning context is the process for automatically choosing what part of the overall learning content should be prepared and made available for the next offline period of a learner equipped with a mobile device. We can split the hoarding process into few steps that we will discuss further in more details:

- 1 Predict the entry point of the current user for his/her next offline learning session. We call it the ‘starting point’.
- 2 Create a ‘candidate for caching’ set. This set should contain related documents (objects) that the user might access from the starting point we have selected.
- 3 Prune the set – the objects that probably will not be needed by the user should be excluded from the candidate set, thus making it smaller. This should be done based on user behaviour observations and domain knowledge.
- 4 Find the priority to all objects still in the hoarding set after pruning. Using all the knowledge available about the user and the current learning domain, every object left in the hoarding set should be assigned a priority value. The priority should mean how important the object is for the next user session and should be higher if we suppose that there is a higher probability that an object will be used sooner.
- 5 Sort the objects based on their priority, and produce an ordered list of objects.
- 6 Cache, starting from the beginning of the list (thus putting in the device cache those objects with higher priority) and continue with the ones with smaller weights until available memory is filled in.

An effective hoarding system will highly depend on the system’s knowledge about the specific user for which materials are to be prepared. Thus the hoarding process should be split into two parts:

- 1 the first interaction with the system, when no knowledge is available about the concrete user
- 2 every next (after the first) access, when the system has some knowledge about the user and continuously gathers more on every iteration.

This system’s knowledge includes user preferences, learning style, personal learning abilities, the level of expertise in the studied field and topic. It can be acquired in different ways – by direct assessment of the user, by questionnaires and quizzes, but also by observing and analysing the user behaviour during his/her usage of the system, thus automatically discovering user’s learning style, preferences, acquired knowledge, *etc.* We should point out that our current work is mainly focused on this last mode – automatic gathering of important data for hoarding knowledge about the learner.

#### 3.1 Measure the quality

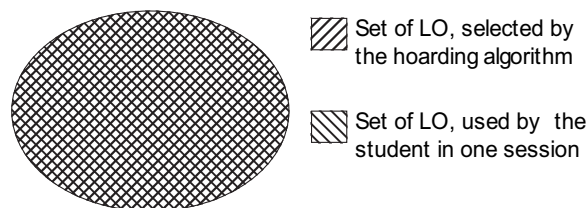
An important point is to measure the quality of the hoarding and to try to improve it continuously. An often used metric in the evaluation of caching proxies is the *hit ratio*. Hit ratio is calculated by dividing the number of hits (*i.e.*, found LOs) by the total number of uploaded predictions (cache size). It is a good measure for hoarding systems, though a

better measure is the *miss ratio* – a percentage of accesses for which the cache is ineffective. Kuenning and 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 since the missing content should be retrieved from the web. This differs from the mobile case where with unavailable internet connection a miss in the hoard might be fatal. In order to quantify this measure it is possible to demand a user rating on every miss, using few different impact values. In some cases of the learning scenario this technique has little sense, because it might be doubtful if we can trust the user's opinion about his/her own knowledge and expertise and most probably every requested learning material is in fact important for the study process. In Kuenning and Popek (1997) miss cost is also defined as *time to first miss* measure – a simple count between the start of the disconnected operation and the first hoard miss. Note that this evaluation criterion can be used only on real-use of a system (and its hoard part). It is also strongly related to the hoarding size. Another possible 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.

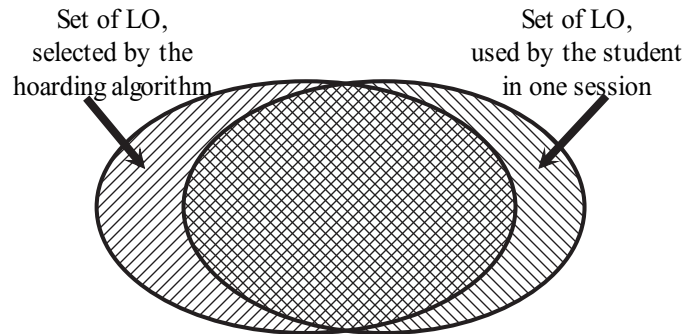
The two important measurements that can be used by the hoarding for improving its work on every step are the 'hit rate' and the 'miss rate'. A low hit rate means that the hoarding was somehow ineffective because much unneeded stuff have been cached. The user is never *directly* aware of a low hit rate, but he/she is strongly affected by a high miss rate, since it measures the system's failure to respond to the user's requests. Of course the two measures are somehow interrelated: wrong priorities might lead to include some unneeded stuff in place of some useful one, therefore adversely affecting both measures.

The goal of the algorithm is to maximise the 'hit rate' and at the same time to minimise the 'miss rate'. The ideal situation is to achieve  $\text{hit\_rate}=100\%$  and  $\text{miss\_rate}=0\%$ , which would mean that the hoarding set contains *all and only* the items that the user needs during his/her study session as shown in Figure 1. Of course, a hit rate lower than 100% would be acceptable as long as the miss rate remains at 0: it would only imply a sub-optimal usage of the available resources (*i.e.*, a waste in memory) without affecting the perceived system performance.

**Figure 1** The ideal hoarding set



Though the ideal picture (Figure 1 above) is to select all and only those items that will be used by the user, it is obvious that in a real system such an ideal situation is almost impossible to reach. Most probably we will have some (desirably big) overlapping between those cached by the hoarding algorithm LO and those LO really requested by the learner (see Figure 2).

**Figure 2** The expected picture

As mentioned before, the hoarding subsystem should be able to analyse how successful was the previous hoarding and improve its further predictions. For this we should be able to check which parameters or combinations of parameters of the user model and/or domain knowledge have bigger impact on the goodness of the algorithm.

By analysing the goodness of the prediction of the hoarding algorithm we can try to tune its work. For example, if a user indicates an LO miss as fatal, the algorithm should check why this LO was not cached, *e.g.*, if this entry was pruned or was given a small priority, and later the 'rules' for pruning and/or prioritising should be reconsidered accordingly. This is actually one of the particularities that mobile learning offers. As mobile devices are certainly personal devices (used only by their owners) it is possible to securely identify the user.

### 3.2 Definition of session in the mobile learning context

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' (MSDN Library). It is a series of transactions or clicks on the web pages links made by a single user. There are different criteria to decide if a session is over or not. The most commonly used one is the inactivity period of the user: resumption of the activity by the same user after a timeout has occurred is considered as the start of a new session.

On the other hand, for hoarding in a mobile system the importance falls on the time between two possibilities of the user to synchronise with the main server. In this sense, we find more useful to define a session as the time between two synchronisations of the mobile device with the main online system. The default session length might be one day, as synchronisation is generally done once per day, but during the system usage other session length might be observed and explicitly set for every user.

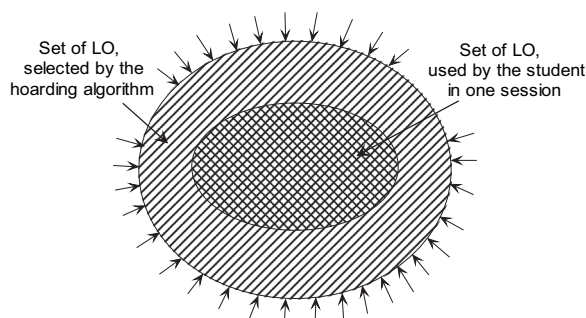
Further in the paper we will use the term 'single session' to indicate the first definition above, whilst we shall use 'session' to indicate the hoarding-related meaning. We will also speak of 'daily session' to mean all the activities that have taken place in a calendar day.

### 3.3 The first access to the system

Earlier we mentioned that the hoarding process differs on the first access of a user to the mobile system. This happens because we do not ‘know’ this concrete user and his/her particularities. Nevertheless, most of the steps of the hoarding should exist, although they will be a little changed. We still have to ‘predict’ the starting point, to generate a candidate set and to try to sort the objects in this set, but in this first access of the user the hoarding subsystem should calculate and use some default values, extracted by analysing the behaviour of all previous users of the system.

Let us start with the learner’s entry point. Generally, a learning material is created by the educator with a certain sequence in mind. Thus, based on the additional knowledge about the learning material structure, the system can be aware of the most possible starting point of the student’s first session. This might be an index page or a list of all lectures of the course. Based on the observations on all previous users, the system can be aware of frequently used sequences of material used on first request and can also estimate the average or maximum depth, in which the students browse during their first session. Still it might be that users have very different behaviour. In the context of caching the content on the first user access, the system should hoard as much data as possible in trying to satisfy all the user’s requests, as shown in Figure 3. In a system like m-ELDIT this means to deliver only a limited amount of basic data (texts) and much auxiliary material (dictionary entries).

**Figure 3** The hoarding starting step



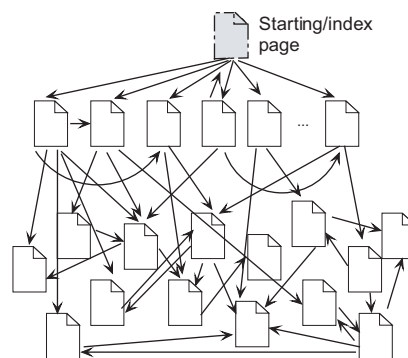
Later, the system can try to detect the user expertise level on the study topic (by questionnaire for example) and to narrow the hoarding set using some domain knowledge; *e.g.*, if certain materials should be proposed to beginning users, whilst current user is advanced, the material should be excluded from the hoarding set. An initial evaluation of the user knowledge could be provided by the educator assessment, though it is out of the scope of this paper. As we are mainly interested in automatically extracting important knowledge about the user we would like to look at the tracking data of what the learner accessed. In some cases it might be possible to consider (with certain confidence) that a portion of the material which the student reviewed is mastered, or we can do some mining based on how long the user needed to review this concrete portion. In other cases it might be very important to look also at what the learner could access, but decided not to view.

It is important to consider that on the first interaction the user is commonly unfamiliar with what can be done, what interactions are allowed, what will be received on different actions, *etc.* This means that user actions might be based on his/her curiosity, rather than driven by his/her knowledge or by the content. This leads to the assumption that the mining on the data gathered by the system on the first user knowledge should be more attentive, and extracted rules might be unreliable.

### 3.4 Predict the starting point

As mentioned in the previous section, the web-based learning material provided by some educator will generally be structured in some manner and will have a certain starting point or index page (shown in Figure 4). This is the starting point of the learner for his/her first learning session with the system. It can be also often a starting point of every following session, especially if this index page contains an ordered reference of other materials, like lectures sequences, exercises, *etc.*

**Figure 4** Web-based material structure



A possible approach for predicting the starting point of user sessions is to keep statistics on what the starting point of a session is, considering what the end-point of the previous session was.

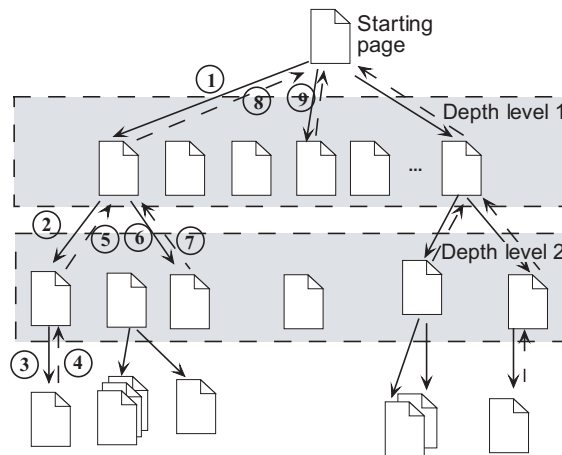
Our initial experiments on mobile ELDIT show that after the first learning session (which we consider almost unpredictable, as the rules that could be extracted by analysing it are unreliable) the users generally show a very ‘coherent’ behaviour – if a list of materials is presented to a user he/she almost always starts from the first item of the list, then goes to the second, to the third, and so on. It is also valid for the sessions – the user continues from the point where he/she finished last time. This rule is rarely changed and if it happens, it is based on some specific interest of the user. For example, we were giving a list of texts that were thematically grouped and the users were generally browsing starting from the beginning of the list. In infrequent cases when a certain topic was especially interesting to the user, he/she skipped the previous subjects and read directly what was of interest and later returned to what was skipped. We cannot be sure that in every kind of learning material the users will show the same sequential behaviour. We however believe that supposing a continuous user browsing is a good starting point for hoarding predictions whenever instructivism approach is applicable. It should also be mentioned that in our scenario the material (the texts) were just listed and not specifically

ordered, which gives the user the freedom to navigate as preferred. Still the users show this consecutive browsing behaviour. Often, in contrast with our system, for reading certain material there might be a precondition given by the educator for the user to be already familiar with the previous topics. Depending on the mobile learning system, additional information about those preconditions can be known to the system and considered in the predictions. Evaluation of the user competence on a subject will surely contribute.

### 3.5 Generate 'candidate' set

As mentioned earlier, one of the steps of the hoarding algorithm is to construct the 'candidate' set of learning objects to be hoard. 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 (Figure 5). This means that the candidate set should contain the objects linked to the starting point, *i.e.*, the objects that the user might decide to visit. Further, it should also contain the objects that are linked to those objects that the user will access and so on. The construction of the candidate set should be up to the depth level that is generally reached by the user. For the first access, this value can be taken as the average (or the maximum) depth of all previous first sessions.

Figure 5 Browsing path



The links between the pages give us the structure of the website (a learning material in particular), thus we can extract the links between the LO by parsing the pages and keep this data in a format that is more useful for computations. These links might be either bi-directional or not. We can build a table that represents these links in the way shown on Listing 1.

**Listing 1** Creating the LO links table

```

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;
  }
}

```

An example table that can be a result from this algorithm is shown in Table 1. On the first row one can see that  $LO_1$  contains link to  $LO_2$  and to  $LO_n$ , but not to  $LO_3$ . There is a bi-directional link between  $LO_2$  and  $LO_3$  (see row 2 col. 3 and row 3 col. 2). In this way we can easily observe the set of objects 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 would be the objects directly linked to a particular object. From this table we can easily construct the ‘candidate’ set for every next step/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).

**Table 1** Links between LO

	$LO_1$	$LO_2$	$LO_3$	...	$LO_n$
$LO_1$	x	1	0		1
$LO_2$	0	x	1		1
$LO_3$	1	1	x		0
...				x	
$LO_n$	1	0	1		x

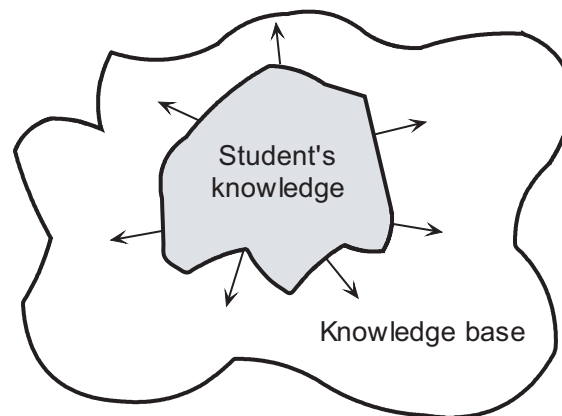
The generation of the candidate set should start from the starting point, predicted for the next offline session. It should generate a ‘candidate set’ of the LOs connected to this point, and pruning of those candidates should follow afterwards. When the pruning of this depth-level-1 candidate set is finished, a candidate set should be generated for every LO that is still in the set, thus going one level deeper. Again, pruning should be done on the newly generated candidate set and the cycling procedure should stop when the estimated user’s browsing depth is reached.

### 3.6 Pruning

Pruning is the step when the hoarding system decides if an LO is probable to be seen by the user or not; in the latter case, the hoarding system excludes this material from the hoard. This should be considered the most important (together with prioritising) step and at the same time the most fragile one in the hoarding process. Alternative to the pruning might be a prediction of the exact path that the user will be following, but in a real system (unless a very strict following of the learning sequence is required by the educator) this would be almost impossible.

Pruning should be done of LO that are not interesting for the user or the user already knows/has mastered. One possible schema is to determine the user knowledge by assessing him/her at the beginning of the learning with the system. The user knowledge is always a subset of what is provided by the system knowledge base (see Figure 6). By a well-defined questionnaire, the system might determine with a good accuracy the user knowledge set.

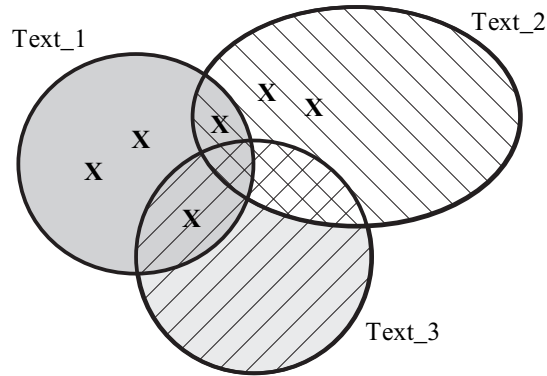
**Figure 6** User knowledge as a subset of the knowledge base



If the system does not provide any initial assessment, then the user knowledge set is empty at the beginning. Nevertheless, the goal of any educational tool is to increase the student's knowledge over the provided knowledge base, so in general the set representing the user knowledge should be dynamic – continuously growing. If some particular exception is not determined, then the system should prune the LOs that are in the knowledge base of the student. At this point, it is already clear that it is very important to correctly determine what subset of all knowledge base is the student's knowledge.

In our Mobile ELDIT system we decided not to test the user knowledge at the beginning but rather to try to automatically gather this information by analysing the user browsing behaviour. We did pruning of the LO based on our supposition that the user knows a certain LO.

It was previously mentioned that our data are very low-granulated – up to a word entry, which are the LOs in this case – thus we have some overlapping in the data that can be accessed from different locations (*e.g.*, the same words will be presented in more than one text). On Figure 7 we schematically show the LO sets of three texts. The X-symbols show the words that the user requested to see when reading the text in which the X belongs. At the first user access we did not have any knowledge about his/her language skills thus no pruning was done. If the user was reading on the first session Text\_1 and it was predicted that for the second offline session Text\_2 should be prepared, then we can prune the LO that the user had the possibility to access the last time, but decided not to do so. Thus, we subtract from the whole set of LO for Text\_2 the words that we consider the user knows.

**Figure 7** Overlapping of LO accessed from different locations

It should be pointed out that it is possible that the user only opens a page with a text but doesn't really read it. In such a case, the LOs that were contained in the set will be wrongly considered as known by the user. Thus, this elementary rule might be too simple and might lead to big hoarding miss rate. If used with a combination of other rules, the accuracy of pruning should be noticeably higher. For example, one can look at the time needed for reading certain page: if the time was below a given threshold, the material can be considered as not read.

### 3.7 Prioritising

Setting the priorities to the LOs that are still in the hoarding set after the pruning process is also a very important step. This is because a pruned set might still be bigger than the available mobile device memory and only part of it will fit in. The priorities in the hoarding context should mean how important the object is for the next user session and should be higher if we suppose that there is a higher probability that an object will be used sooner. In this sense, the predicted 'starting point' of the user's next offline session should be always assigned a maximum priority.

For prioritising the LO we can analyse the accesses done previously by all the users and extract interesting and important knowledge. Aggregated data, like the correlation between the objects, based on their contemporary usage in other users' sessions, is one thing that can be easily discovered, and is very helpful. For example, a well-known association rules (see Hand *et al.*, 2001) discovering can be applied to determine from all previous learning sessions the relations between LOs that are 'very strong', *i.e.*, associations discovered with confidence near to one and big enough support value. Note that it is expected that not a lot of such associations will be found, as the common scenario is to have big variety of LOs and also big diversity of students' knowledge, interests, and learning preferences. The rules extracted in this way will be of the following type:  $LO_i \Rightarrow LO_j; \text{conf}=0.99 \text{ sup}>0.5$  which we can read as 'Almost every time when the  $LO_i$  was viewed by some user,  $LO_j$  was also viewed in the same session. Example can be that  $LO_i$  is a problem given by the educator to the students to practice the comprehension of certain material studied and  $LO_j$  – the solution given also by the lecturer and linked at the end of the lecture'.

**Table 2** Example of sessions and requested LO

	$LO_1$	$LO_2$	$LO_3$	$LO_4$	$LO_5$	$LO_6$
Session <sub>1</sub>	0	0	0	1	1	1
Session <sub>2</sub>	1	1	1	0	0	0
Session <sub>3</sub>	0	0	0	1	1	1
Session <sub>4</sub>	0	0	1	0	1	1
Session <sub>5</sub>	1	1	1	0	0	0
Session <sub>6</sub>	1	0	1	0	0	0
Session <sub>7</sub>	1	0	0	0	1	1

For the example, we can preprocess our tracking data (the user’s clicks recorded on the mobile device) into the data shown in Table 2. Every row represents a single session (not taking into account to which particular user belongs). Every cell 1 means that  $LO_i$  was viewed during Session<sub>j</sub> not taking care of the sequencing. For this data association rules, algorithm will discover with confidence=1 the following relations:

$$LO_2 \Rightarrow LO_1; LO_2 \Rightarrow LO_3; LO_4 \Rightarrow LO_5; LO_5 \Rightarrow LO_6; LO_6 \Rightarrow LO_5$$

Association rules can be discovered also in more limited number of sessions (not all at a time). For example, one might search for correlated objects only in the sessions of users that were classified in the same group of content interest or field expertise. Considering again the example data in Table 2 if we apply a clustering algorithm (like k-means) (see again (Hand *et al.*, 2001)), the algorithm will produce two clusters. Applying association rules only to the sessions in the same cluster, we get some additional associations. The clusters and discovered associations are as follows:

**Table 3** Associations found for clusters of sessions

Cluster	Instances	Additional associations
Cluster <sub>0</sub>	Session <sub>1</sub> Session <sub>3</sub> Session <sub>4</sub> Session <sub>7</sub>	$LO_1 \Rightarrow LO_5$ $LO_3 \Rightarrow LO_5$ $LO_3 \Rightarrow LO_6$ $LO_4 \Rightarrow LO_6$
Cluster <sub>1</sub>	Session <sub>2</sub> Session <sub>5</sub> Session <sub>6</sub>	$LO_1 \Rightarrow LO_3$ $LO_3 \Rightarrow LO_1$
		...

The above associations (like  $LO_1 \Rightarrow LO_5$ ) show that if  $LO_1$  is to be selected for the hoarding set, there is big probability that the user will also be accessing the object  $LO_5$  during the same session. Moreover, associations of the type  $LO_5 = 1$  and  $LO_6 = 1 \Rightarrow LO_2 = 0$  can also be discovered, showing that if the user will be viewing objects  $LO_5$  and  $LO_6$  it is most probable that the object  $LO_2$  will not be viewed, thus in some cases we can increase the set priority of certain material; in other cases, we can set it to much lower level, which will sometimes lead also to exclusion from the hoarding set.

Note that for the example above we considered only associations with confidence = 1 and any support greater than 0. In real situations the best values for these parameters should be experimentally discovered. Generally, the confidence value of the discovered associations can help also in placing the items of the 'candidate set' in an ordered list.

Also, other data mining and/or machine learning algorithms should be considered and tested to see their appropriateness for the hoarding process and how they can be combined best.

When no other rules can be applied and there are too many LOs with the same priority that are predicted to be uploaded and the memory space is limited, a random choice should be applied.

#### 4 User modelling

There are different ways to model user behaviour depending on the application and its needs. In the context of hoarding we recognise two groups of characteristics that will be used differently in the hoarding process. We schematically call the first 'user behaviour' and the second 'user knowledge'. Additionally, there could be another group of characteristics that we call 'user preferences', which are not substantial for the hoarding, thus we do not discuss them here. 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 analysing the log files. The process for retrieving automatically the information about the user should consist of few steps, like preparation of the data for analysing where the log files are preprocessed and integrated into a database, and afterwards application of different knowledge extraction algorithms to find interesting relations. In more details we discuss the user behaviour observation elsewhere (Trifonova and Ronchetti, 2005).

The user behaviour 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), speed of read/study (fast, medium, slow), *etc.* Based on the user behaviour we can group the learners and do mining based on the similarities and differences between the groups and between the members of the same group (shown in previous sections). This should help us mainly to predict what will be needed, *i.e.*, this data will be used to fill in the hoarding set or in prioritising the LO.

On the other hand, the user knowledge profile should consist of everything that the system knows about what the user already knows. Example is 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. Users can also be grouped based on their knowledge, but in contrast to the user behaviour the profile of the user knowledge will be mainly 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 include, for example, the user age, gender, mother tongue, *etc.* On the other hand, the dynamic data are our current knowledge about the user parameters that are changeable over time and should be reviewed in certain periods of time. For example, the user browsing pattern might change drastically a few days before an exam date, thus the hoarding system should be able to quickly recognise such changes and react accordingly.

## 5 Related work

A lot of work has been going on in the 'mobile' area. New architectures and guidelines are proposed to satisfy the needs for mobility in different fields, like wireless business and commerce (Andreou *et al.*, 2005; Chen and Nath, 2004), healthcare (Lin and Vassar, 2004; Cocosila *et al.*, 2004), tourism (Garzotto *et al.*, 2004), and learning (Trifonova and Ronchetti, 2003).

What is deficient in most of the proposed architectures and systems is that they consider either only online access to the content and services (see for example Chen *et al.*, 2005) or they are designed especially for small content data that all fit into the device memory (example: mobile learning courses at <http://www.hotlavasoftware.com/>). The point to consider is that in some scenarios (like the learning one) the content that is to be delivered can be quite large. Only some transcoding proxies take care also for caching web pages for offline usage (*e.g.*, AvantGo). We think that delivering content for offline usage is an important issue as mobile devices are often disconnected because of the lack of access in certain places or because of the high prices in most of the cases, thus our intention is to support both online and offline access to data.

A problem similar to the one we face (offline access to data) is treated in the offline browsing of web content. A review of the available offline browser utilities (like [www.avantgo.com](http://www.avantgo.com), [www.htrack.com](http://www.htrack.com), [www.webstripper.net](http://www.webstripper.net), *etc.*) shows that generally during the online periods the user selects sites that should be uploaded for later offline 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 and the hoarding set is large, the memory limitations make such an approach unfeasible.

The caching problem has been studied in the general case for the internet. Wang (1999) presents 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 analysing his/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 a different goal – reduction of bandwidth consumption, of access latency, server workload, *etc.* They explore the case of the web where the search space is much bigger and 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 internet connection is permanent.

The idea of hoarding for disconnected devices in distributed file systems has been first described in Kistler and Satyanarayanan (1992). Although they do not consider mobile devices in the sense of PDAs they propose the Coda File System to explore the usage 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 utilised 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) and the direction taken 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 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 usage. 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 of disconnected operation, an interesting solution has been proposed in SEER system (Kuenning and Popek, 1997). The authors were also inspired by the work on 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 authors try 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. These criteria 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 behaviour in the real-world system, which appeared because of the way the operating systems and some frequently used programmes work (like the ‘find’ operation under Unix). Recent experimentations with the same system (Kuenning *et al.*, 2002) showed surprising finding – the complex clustering methods that are used in the system work, in most of the cases, worse than a Least Recently Used (LRU) algorithm enhanced with some heuristics. This shows us that the research field is still open for work.

## 6 Conclusion

This paper describes the hoarding problem for a mobile learner without internet connection. The problem is how to support work on a mobile device when it is impossible to load in its memory all the data that comprise the full knowledge base.

We have outlined a general hoarding strategy and we gave details on possible approaches on every step of the described process. Though our work is still in progress we have shown that in a real-world mobile learning system, hoarding might be a very important part. We have done some important deductions that can give also a starting point for other developers in the field. We have drawn attention to particularities of the mobile learning scenario that differ from scenarios considered previously:

- Sessions – we have emphasised that there is a difference between a ‘session’ in the internet world and what should be considered a session in this particular scenario – hoarding content for mobile learning.
- Effective pruning and prioritising – we have drawn the attention on the importance of these steps and also proposed pruning and prioritising criteria that were not used for hoarding before, like considering what the user could but did not access in his/her previous sessions. Whilst this technique cannot be fully used in the general case of internet caching and prefetching, it might be a source of information for pruning in the m-learning case.

The main contribution of this paper is the drawing of the attention of the researchers and developers in the mobile learning domain to the importance of the hoarding problem. People are going around this problem (in different domains) for years, saying that mobile devices' characteristics are continuously growing and, soon, fast internet connection will always be available. Problems, however, will still exist! First, we cannot assume that learners will equip themselves with the top technologies. Second, the always growing need for 'more space' can be seen also with desktop PCs. Once more space is available, you start using it and you need more. As it is true for the compression technologies, that it will always be needed, it will be the same for mobile devices and hoarding. Once we can put on a device memory all the text, we will want to put video also; once we can put video, we will want higher quality, *etc.* Thus, hoarding should be considered whenever we want to develop an efficient real-world mobile learning system.

## References

- Andreou, A.S., Leonidou, C., Chrysostomou, C., Pitsillides, A., Samaras, G., Schizas, C.N. and Mavromous, S.M. (2005) 'Key issues for the design and development of mobile commerce services and applications', *International Journal of Mobile Communications*, Vol. 3, No. 3, pp.303–323.
- Chen, L. and Nath, R. (2004) 'A framework for mobile business applications', *International Journal of Mobile Communications*, Vol. 2, No. 4, pp.368–381.
- Chen, M., Zhang, D. and Zhou, L. (2005) 'Providing web services to mobile users: the architecture design of an m-service portal', *International Journal of Mobile Communications*, Vol. 3, No. 1, pp.1–18.
- Cocosila, M., Coursaris, C. and Yuan, Y. (2004) 'M-healthcare for patient self-management: a case for diabetics', *International Journal of Electronic Healthcare*, Vol. 1, No. 2, pp.221–241.
- Gamper, J. and Knapp, J. (2003) 'A data model and its implementation for a web-based language learning system', *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*, Budapest, Hungary, 20–24 May.
- Garzotto, F., Paolini, P., Speroni, M., Proll, B., Retschitzegger, W. and Schwinger, W. (2004) 'Ubiquitous access to cultural tourism portals', *Proceedings 15th International Workshop on Database and Expert Systems Applications*, 30 August–3 September, pp.67–72.
- Hand, D.J., Mannila, H. and Smyth, P. (2001) 'Principles of data mining', *Massachusetts Institute of Technologies, 2001*, ISBN 0-262-08290-X.
- Helal, A., Khushraj, A. and Zhang, J. (2002) 'Incremental hoarding and reintegration in mobile environments', *Proceedings in Symposium on Applications and the Internet (SAINT)*, pp.8–12.
- Kistler, J. and Satyanarayanan, M. (1992) 'Disconnected operation in the coda file system', *ACM Transactions on Computer Systems*, Vol. 10, No. 1, February, pp.3–25.
- Kuenning, G.H. and Popek, G.J. (1997) 'Automated hoarding for mobile computers', *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, St. Malo, France, October.
- Kuenning, G.H., Reiher, P., Ma, W. and Popek, G.J. (2002) 'Simplifying automated hoarding methods', *5th International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (WiM'02)*, Atlanta, Georgia, USA, September.
- Liang, H., Xue, Y. and Byrd, T.A. (2003) 'PDA usage in healthcare professionals: testing an extended technology acceptance model', *International Journal of Mobile Communications*, Vol. 1, No. 4, pp.372–389.
- Lin, B. and Vassar, J.A. (2004) 'Mobile healthcare computing devices for enterprise-wide patient data delivery', *International Journal of Mobile Communications*, Vol. 2, No. 4, pp.343–353.

- Trifonova, A. and Ronchetti, M. (2003) 'Where is mobile learning going?', *Proceedings of the World Conference on E-learning in Corporate, Government, Healthcare, and Higher Education (E-Learn 2003)*, Phoenix, Arizona, USA, 7–11 November.
- Trifonova, A. and Ronchetti, M. (2004) 'A general architecture to support mobility in learning', *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004 "Crafting Learning within Context")*, 30 August – 1 September, Joensuu, Finland.
- Trifonova, A. and Ronchetti, M. (2005) 'User behavior observations for offline delivering of learning materials in a mobile system', *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-Media 2005)*, 27 June–2 July, Montreal, Canada.
- Wang, J. (1999) 'A survey of web caching schemes for the internet', *ACM Computer Communication Review*, Vol. 25, No. 9, pp.36–46.
- Zhang, J., Helal, A. and Hammer, J. (2003) 'UbiData: ubiquitous mobile file service', *Proceedings of the ACM Symposium on Applied Computing (SAC)*, Melbourne, Florida, March.

## Note

- 1 see [www.w3.org/XML/](http://www.w3.org/XML/)