User Behavior Observations for Supporting Offline Delivering of Learning Materials in a Mobile 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
{Anna.Trifonova, Marco.Ronchetti}@dit.unitn.it

Abstract: In the context of e-learning and especially in adaptive learning systems an integral part of any system is the user analysis and behavior observation. We look at this important part from another angle - the one of mobile learning and more specifically the automatic delivery of learning material during users' offline periods. This process, called hoarding, depends strongly on system's awareness of the user specific needs, problems, behavior characteristics and preferences. As our goal is not to adapt the material to be presented, nor to influence on user's learning sequence, but to predict what part of the material the user will need for his/her next learning session, the process of user modeling and the parameters that interest us differ. Here we discuss these differences, the problems that we meet and possible ways to solve them.

Introduction

Distance and e-learning are both widely discussed in the last decades. More recently adaptation and personalization are very often addressed in the educational context and are applied in different parts of learning systems, e.g. adaptation in the learning material personalization or adaptation of the navigational path of the individual learner. In all cases adaptation and personalization are done based on the system's understanding of and knowledge about the concrete user. In the recent years one more term also comes to focus in the learning domain – mobile learning. Mobile learning or m-learning is discussed as the step further in distance learning, an integral part to the future educational systems. It is becoming obvious that the small mobile devices, like mobile phones and PDAs will soon be in everyone's hands and their usage have to be explored for the educational purposes too.

This paper focuses on the issues related to analyzing user behavior in order to support offline access to learning material from a mobile device. First the motivation of this work is discussed in more details in the following section. We show why the offline delivery is an important issue and what the questions that have to be answered with user behavior analysis are. Afterwards we discuss the user modeling and the attributes that a model should support. Later on the process of extraction the user model automatically is shown, together with few possible approaches. The following section is dedicated to some important issues, namely creation and updating of user models which should be considered in order to formalize the model. Discussion and future work are followed by conclusions and references.

Motivation

Mobile devices could be used in education in lots of different ways (Trifonova et al. 03) and the research in the field is quite dispersed. This led us to classifying services that are specific and should be provided by a general m-learning platform. The proposed architecture (Trifonova et al. 04) consists of three separate modules that interact between themselves in order to support learning through mobile devices. Namely these modules are "Context Discovery", "Mobile Content Adaptation" and "Support of Disconnected Operation". These three modules should

cooperate and thus allow the user to access specially adapted learning materials and other learning related services through mobile devices, both online and offline.
In the literature one can see that years pass, but the supposition that 'very soon' every device will have always on connection has still not become true. In the fall of 2000 Clark Quinn says (Quinn 00):

> "The vision of mobile computing is that of *portable* (even wearable) computation: rich interactivity, *total connectivity*, and powerful processing - a small device that is *always networked*, allowing easy input through pens and/or speech or even a keyboard when necessary (though it may be something completely different like a chord keyboard), and the ability to see high resolution images and hear quality sound."

Though we also agree that this will sooner or later become true the current situation is not like we would like it to be. The devices had really became mobile in the sense of light, small and powerful for impressively short period of time and though there are quite a lot of technological ways to connect to the internet, through WAP, GPRS, Wi-Fi and etc. still users have long periods of disconnection. These periods might be intentional or not - because of the lack of proper infrastructure or because the connection has high costs. Nevertheless our vision is that it is important that the user has access to his data and to materials that he/she wants to study even when the connection is not presented.

The automatic support of offline delivery of learning content is called hoarding. The hoarding process should adjust to every user specific needs and will strongly depend on system's knowledge about the user. This knowledge might include user's learning style, natural learning habits and abilities, the level of expertise in the studying 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 analyzing the user behavior during the studying with the system, thus automatically discovering user's learning style, preferences, acquired knowledge and etc.

The hoarding process consists of automatic prediction of what learning material will the user need in its next learning session and pre-fetching (caching) of the predicted materials in the device local memory. For making the things more clear for the hoarding process we consider two separate engines. One will deal with observing the user and creating user models and the other for the pre-fetching itself. We call the first one "User Behavior Analyzing Engine" and this paper will concentrate on it. On the other hand 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) and decide what learning material to be cached. The hoarding process is described in more details elsewhere (Trifonova et al. 05).

The ways to model a user and/or her behavior depends on the application and its needs. Usually the student behavior is observed and analyzed with the goal to improve the teaching process effectiveness. In contrast of what is commonly done in Intelligent Tutoring Systems (ITS) or generally adaptive educational systems (AES) we do not aim at adapting the didactic strategies, i.e. changing the presented material or its structure. We need to analyze the student behavior in order to predict what the user will be doing in the forthcoming learning session. In our scenario the following questions appear and we have to try to answer them:

- What is a learner session in the mobile scenario?
- What is the starting point in the learning material set or sequence for the user's next session?
- How can we predict the learning path or sequence?
- How do we create (formalize) a useful user model?
- What are the important parameters of the user behavior that have influence on the hoarding predictions?
- How do we use different parameters of the user model for predicting and/or pruning in the hoarding process and do these different parameters have different significance for the prediction and/or pruning process?

User Modeling Issues for Hoarding Purposes

As mentioned earlier user modeling is widely explored in Adaptive System (AS) that typical should have at least three main modules that schematically can be called "Domain Knowledge" module, "User Modeling" module and

“Adaptation” module, also called “Intelligent tutor”. Commonly the “Intelligent tutor” is a complex module, which is the heart of the adaptive system and holds all the logic and rules for the adaptation process. Intelligent tutoring systems tend to recognize individual user needs and as a consequence to adapt the sequence or the type of learning material. This process is based on the user model created and sustained by the system and in general the modeling criteria are specific for each system.

In the context of hoarding we suggest the separation of the user modeling parameters into three groups of characteristics that will be used differently by the hoarding algorithm. We schematically call them ‘user behavior’, ‘user knowledge’ and ‘user preferences’. Depending on the mobile learning system it is possible that not all the parameters that will be discussed here will be needed. Furthermore 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. Nevertheless in the system that we consider (Trifonova et al. 04a), called Mobile ELDIT, we would like to explore the possibility to obtain the data only by analyzing the user log files and we’ll concentrate our discussion on techniques that could be used in this case.

The *user behavior* can be described in terms of browsing styles, e.g. consecutive, random, interest (subject)-driven, etc.; preferred type of educational media (e.g. the user prefers video to combination of text and pictures); quick browser or deep-explorer and 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 contain the system knowledge about what the user already knows and what he/she doesn’t know. 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 or concepts that the user already knows. In some cases the user knowledge might be a set of items (concepts, learning objects (LO) and etc.) with its corresponding estimated values for the user familiarity. 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.

Finally the *user preferences* should represent the system awareness of the user’s preferred presentation or interaction styles. The user preferences together with the other user characteristics will be used by the system to do adaptation of the presented material but this sort of adaptation is out of the scope of our current work.

Automatic Extraction of Attributes

The common way to determine the user characteristics and knowledge is by assessing him/her through questionnaires, quizzes and tests and letting the user manually set his/her own preferences. Our research interest falls on the automatic discovery of these attributes. It is obvious that the user knowledge of a concept determined by assessing him/her and checking the tests results can give some commonly precise quantified measure of the learner understanding and advances in the subject. On the other hand by analyzing only the user’s interaction with certain system might give less precise approximation, but makes the life easier.

The process for retrieving automatically the user patterns (shown on Figure 1) consists of 1) preparation of the data for analyzing and 2) applying different algorithms for automatically extracting interesting knowledge.

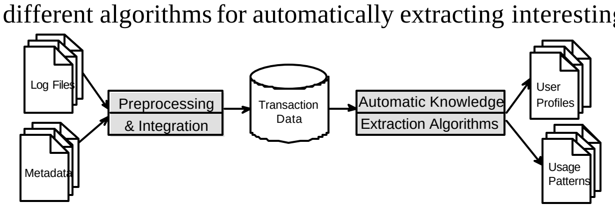

Figure 1: *The process of extracting knowledge about the student*

By “data” we mean the log files where the user interaction with the system is saved, but also it can be any additional data about the learning material itself, about expected user behavior and etc.
One of the advantages that mobile learning gives us comparing with e-learning is the possibility to easy distinguish one user from other. In e-learning environments the problem of having multiple users using the same computer or the fact that often the users are behind a proxy server is generally solved by asking for username and password the user on every session. In a mobile learning system one can have the advantage that the mobile devices - cell phones and PDAs are very personal devices, generally used only by one person, thus the problem of identifying the user, which often appears in web based systems, is not applicable here. The log files might be collected on the server, but in this case the offline periods will not be covered. So a better solution is that the tracking is done on the mobile device, the tracking data is stored locally and when connection is available is transferred on the server.

The preprocessing of the raw data is the process where the log files and all other available data should be parsed and integrated into a database or other suitable format to perform knowledge extraction algorithms. Generally the preprocessing is one of the most resource consuming processes, but in the context of analyzing user behavior in a mobile learning system this part might be on the server and will most likely be performed during the offline periods of the user. This means that in our context this is not a critical point, e.g. even quite slow speed of the preprocessing and extraction of the data will not lead to user’s impression of a slow system.

As one can see on the figure above with the help of different algorithms for knowledge extraction we expect to get two types of data – on one side are the different typical usage patterns that we need to extract out of all available data set and on the other side is the understanding and categorization of every concrete user. It should be possible to automatically extract knowledge for all three groups of user modeling parameters discussed in the previous section, e.g. *user behavior*, *user knowledge* and *user preferences*.

The following subsections describe some of the techniques that can be used either for extracting typical patterns or to classify the users or to define similarity between them.

Data mining

Data mining in general incorporates different techniques that try to discover and describe structural patterns in data in order to explain that data and/or make predictions from it. Data mining includes clustering and classification, finding association rules and etc (Hand et al. 01). The rules or relationships found with data mining are often called models or patterns. Data mining algorithms will help in finding aggregate common usage patterns from all the tracking data that is collected in the mobile system.

Usually the data set consists of an $n \times p$ data matrix which might be a simplification, idealization or abstraction of the real data. Some data might be very sparse (e.g. containing lots of empty fields), which is very often the case of web mining. In our context (the hoarding, see Trifonova et al. 05) we will have a quite large and sparse matrix, where the rows will correspond to the individual users or user’s session and the columns will represent a particular learning object (LO). For example the cell might represent if the particular user had visited the LO or not.

Some clustering and classification algorithms can be applied to separate the users into different groups. Users with similar browsing activities or with the same interests and needs of learning material will be grouped and later the prediction of their future activities for the hoarding can be done by taking into account activities of the members of the same group. This is one of the techniques that are often used by recommendation systems, like online stores.

Imagine the following data:

	LO_1	**LO_2**	**LO_3**	**LO_4**	**LO_5**	**LO_6**
$Session_1$	0	0	0	1	1	1

Session$_2$	1	1	1	0	0	0
Session$_3$	0	0	0	1	1	1
Session$_4$	0	0	1	0	1	1
Session$_5$	1	1	1	0	0	0
Session$_6$	1	0	1	0	0	0
Session$_7$	1	0	0	0	1	1

* Where 1 means that LO$_i$ was viewed during Session$_j$ not taking care of the sequencing.

Based on the above example data we can try to discover Association Rules applying Apriori Algorithm (see Hand et al. 01). The following associations are found, with confidence=1:

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

We can also use clustering to automatically split the users into two groups, according to what pages or LO they viewed. A clustering algorithm would produce the following two clusters:

Cluster	**Instances**	**Centroids**
cluster0	4 (57%) Session$_1$; Session$_3$ Session$_4$; Session$_7$	0 0 0 0 1 1
cluster1	3 (43%) Session$_2$; Session$_5$ Session$_6$	1 1 1 0 0 0

* The best number of clusters to be generated should be experimentally found.

Based on the centroids, shown in the table above, 'Aggregate Profiles' for every cluster can be created (Mobasher et al. 00). The Aggregate Profile is a table that gives the significance weight of every LO for the current cluster. It can be calculated as the sum of all occurrences of the LO throughout the sessions in the cluster divided on the total number of the sessions in the cluster. Once the clusters are created the user similarity to this cluster can be calculated. New users might be classified to the cluster with the strongest similarity.

Association rules can be discovered also in more limited number of sessions (not all at a time), for example search for correlated objects only in the sessions of users that ware classified in the same cluster (group). Applying association rules only to the sessions in the same cluster as shown above we get some additional associations. The clusters and discovered associations are as follows:

Cluster	**Instances**	**Associations**
cluster$_0$	Session$_1$ Session$_3$ Session$_4$ Session$_7$	$LO_1 \Rightarrow LO_5$ $LO_3 \Rightarrow LO_5$ $LO_3 \Rightarrow LO_6$ $LO_4 \Rightarrow LO_6$
cluster$_1$	Session$_2$ Session$_5$ Session$_6$	$LO_1 \Rightarrow LO_3$ $LO_3 \Rightarrow LO_1$

The above associations (like LO1 $\Rightarrow$ LO5) show that if the object LO1 is to be selected for the hoarding set there is big probability that the user will also be accessing the object LO5 during the same session. Moreover associations of the type LO5=1 & LO6=1 $\Rightarrow$ LO2=0 can also be discovered, showing that if the user will be viewing objects LO5 and LO6 it is most probable that the object LO2 will not be viewed, thus can be excluded from the hoarding set or at least its hoarding priority can be set to much lower level.

For the example above we considered only associations with confidence=1 and any support greater than 0, but in a real situations the best values for these parameters should be discovered experimentally.

Collaborative Filtering

Another approach, which is also used in recommendation systems, is called collaborative filtering. In contrast to the clustering approach in collaborative filtering the algorithm searches for few users that are most similar to each other and later all decisions are made based on those few users. A common scenario is to present every user as a vector of N numbers. Every number represents one of all possible items (in our case learning material chunks) and if the user requested or rated this item or not. Generally the similarity is measured as the cosine of the angle between the vectors of the users.

$$similarity\left(\vec{A}, \vec{B}\right) = \cos\left(\vec{A}, \vec{B}\right) = \frac{\vec{A} \bullet \vec{B}}{\left\|\vec{A}\right\| * \left\|\vec{B}\right\|}$$

Sequential patterns

Basically the user browsing path over a web-based material (and in particular on any web-based learning source) can be viewed as a hierarchy structure (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 (Figure 2). Thus based on the knowledge about the learning material structure, the system can be aware of the possible sequences the user is able to follow and analyze his/her choices.

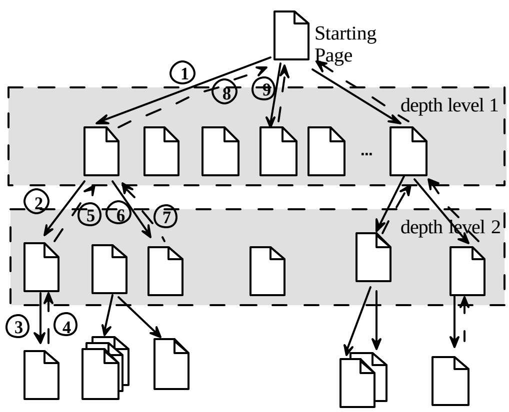

Figure 2: *Browsing path over a web-based material*

We can use the aggregated (e.g. often followed) sequences in order to predict next user steps. For example lots of students have very consecutive learning style, i.e. they prefer to study consecutively the material provided by the educator in the provided order, rarely skipping something. Other students might prefer to jump from one topic on other and later go back, leaded by their interest.

Excluding data from user sets

In the context of hoarding the user analysis is needed for two purposes – for selecting what material the user will be needing but also for "knowing" what part the user does not need. In this sense one can extract lots of useful information, knowing the structure of the content, from analyzing what the user was able to review but decided not to do it. For example imagine a language learning system where the learner reads a text in the target language. The words of the texts are linked to the dictionary entries of those words, so the user can quickly click and consult the dictionary or other related information. Once in a while the user clicks on some of the words that is unknown to him/her and goes back to the text to continue reading. What the system is recording into a log file is the clicks that the student did, thus the system knows what the student did not know at that time. Later the system should use those logs and identify with a certain confidence the learner's competence on this corpus, thematic subject or word set. What is not recorded in the logs is that the student read lots of words, but as long as he was familiar with them did not click on the links. In some cases, like in this scenario, this data might be even more interesting and important. One of the peculiarities of the learning scenario, in contrast of the web, is that the knowledge base is much smaller and its structure will not change this dynamically. This leads to the fact that the knowledge base can be also preprocessed and used in the process of automatic knowledge extraction. The structure of the learning material and the awareness of the system of the relations between the learning objects give the opportunity to know what the user had the possibility to see, but decided not to.

Maintaining User Models

The maintaining of user models includes creation, initializing, saving (adding or updating) data to and restoring parameters from the model. Saving is triggered on user interactions, while restoring is done when particular part of the model is needed by the system for taking decisions what should be done next.

On the first user interaction with the system a user model should be created. It should be initialized and later updated during the student interaction. A possible solution is to initialize the user model after presenting to the user a short pretest and analyzing the outcomes. With a well-defined pretest it is possible to capture the user abilities and knowledge on different concepts treated by the system. Without such an initial test the users should be treated equally. The case where no initial knowledge about the user is available is called ‘cold start problem’.

One direction to help solving this so called ‘cold start’ problem is by providing specific knowledge about the learning material through metadata. The structure, like common ‘initial point’, provisioned student learning path, or connections between individual LO will be good for the initialization of the model, but are specific for every application. Also the provisioned audience might be indicated in advance, so default values of needed parameters can be set up.

We can distinguish static data about the user and dynamically changing data. The static data is for example the user gender, mother tongue and etc. On the other hand the dynamic data is the system’s current knowledge about the changeable over time 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.

It should be pointed 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 automatic extraction of user parameters and updating of the user models and patterns should be done regularly, depending on the expected dynamism of the learners’ behavior. Though generally the process of automatic extraction of attributes, described in the previous section, is quite time and computational power consuming in the context of hoarding and support of offline delivery of data it is not a critical point. The data can be processed on the server and is not mandatory to be done during the user’s online period.

Another point to consider is that the user model and the learning history of the user should be clearly distinguished. The user model keeps the extracted knowledge about the user, while the history can be found either in the log files or in some more comfortable (e.g. transactional) format after the preprocessing is done, as shown before.

Discussion and Future Work

- Evaluation
  Though our system is in the educational domain it is of no use to evaluate it in the sense of increased examination results, because we don’t tend to make changes neither in the learning material itself, nor in the navigational path of the student. It makes more sense to measure the goodness of the system in the terms of machine learning and

knowledge extraction algorithms. Anyway different versions of the system should be prepared to test different parameters influence on the goodness. In more details we analyze the hoarding elsewhere (Trifonova et al. 05).

- Open user model
  In lots of systems is discussed the appropriateness of open user model, where the user is allowed to see its own user model in a human readable way and probably edit it. As our goal is not to influence on user learning but to predict it will be interesting to see if open models are applicable in the hoarding context.
- Pedagogical and psychological issues
  Up to now our approach is very technically oriented and concentrated on the hoarding process, i.e. predicting the user steps without influencing on the browsing sequence. An interesting question though is if it should be combined with adaptation of the learning material. Should we limit the navigation once we have predicted the most probable learner path? How? What do we loose?

Conclusions

User behavior observation is often considered in adaptive e-learning systems. The goal is to model the user so that to improve the study results by influencing on user learning path or by adapting the material presentation. Our work is in the mobile learning domain and more specifically the automatic delivery of learning material. This process, called hoarding also depends on system's awareness of the user specific needs, problems, behavior characteristics and preferences. We need to predict what part of the material the user will need for his/her next learning session. Thus the process of user modeling and the parameters that interest us differ. Here we discussed these differences, the problems that we meet and possible ways to solve them.

References

Hand, D., Mannila, H., Smyth, P. (2001). *Principles of Data Mining*, Massachusetts Institute of Technologies, 2001.

Mobasher, B., Dai, H., Luo, T., Sung, Y., Nakagawa, M., & Wiltshire, J. (2000). Discovery of Aggregate Usage Profiles for Web Personalization. *Proc. of the Web Mining for E-Commerce Workshop (WebKDD'2000),* Boston

Quinn, C. (2000). Mobile, Wireless, In-Your-Pocket Learning. *LiNE Zine "Learning in the new economy"*, available online (12.11.2004): www.linezine.com/2.1/features/cqmmwiyp.htm

Trifonova A., Knapp J., Ronchetti M., Gamper J. (2004). Mobile ELDIT: Transition from an e-Learning to an m-Learning System. *Proc. of ED-MEDIA 2004*, June 21-26, 2004, Lugano, Switzerland

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. (2005). Hoarding Content for M-Learning Context. In *Proc. of Workshop on Pervasive eLearning (PerEL 05), in conjunction with 3rd IEEE Int. Conf. on Pervasive Computing and Communications*, Kauai Island, Hawaii