# DEVELOPING A MOBILE DISTANCE LEARNING SYSTEM

Petr Rogov, Nikolay Borisov

*Abstract: This article considers the basic problems of client-server electronic learning systems based on mobile platforms. Such questions as relational learning course model and student's transitions prediction through the learning course items are considered. Besides, technical questions of electronic learning system "E-Learning Suite" realization and questions of developing portable applications using .NET Framework are discussed.*

*ACM Classification Keywords: K.3.1 Computers and education: Computer Uses in Education – Distance learning*

## Introduction

Wireless connection is increasingly used all over the world especially in the field of business. Users with wireless access to information can work faster and more effectively than their colleagues dealing with cable networks. The keen interest in wireless data transfer technologies is stipulated for extensive application of mobile devices.

Wide spread of portable computing and its penetrating to various fields of human activities make developers think about use of new mobile platforms. It is obvious that mobile market development exerts great influence on the software based on client-server technology. In such cases, porting the client part of an application to the mobile platform becomes the aim to be achieved.

This tendency has also touched the field of electronic learning. As a rule, the e-learning software is based on client-server technologies. Thus, using the portable computer connected to the wireless data transfer environment the user receives an opportunity of planning the educational process. The term *distance learning* means the client's remoteness from the learning content server. Therefore, software developers should provide the student with access to e-learning server irrespective of his current geographical position.

## Developing a Mobile Electronic Learning System

One of the most complicated problems of wireless networks is the absence of stable, guaranteed connection with a network server. Moving from one access point to another the client periodically becomes disconnected from a network. Thus, the e-learning software developer faces two problems:

1. The client-server system based on client's hardwired connection to server cannot normally function in the wireless environment;
2. The program module working on the client side should provide access to a learning content even if connection to the server is not temporary available.

Whereas the first task is only technical and the solution consists in the choice of a proper data transfer protocol and a way of connecting to the server, the second is rather complicated problem, both technically and algorithmically. We shall consider this problem in more detail.

Let us assume that to move from one access point to another student spends time not exceeding 2 hours. For this time, the typical student can study 1-2 lessons with the average information density. We also know that during studying this material the student can face questions, which were discussed in the previous lessons and return to them to restudy. Thus, to provide a full access to the learning content in conditions of no connection to the server the client application should predict sequence of student's transitions to the following lessons as well as probable jumps to the previous lessons. After the forecast is made the program should download the content of all these lessons from the server and keep it in a local storage. Since that moment the part of the learning course will be locally accessible, the user can easily continue studying in conditions of no connection to the server.

The electronic distance learning system E-Learning Suite (http://www.elsuite.net) is developed for solution of problems in question. Client to server connection is carried out using the HTTPS protocol, which provides a high degree of security of the data to be transmitted, and does not demand a permanent connection to the server. The prediction of the student's transitions from one lesson to another and the necessary data caching are carried out due to relational model of the learning course.

Conditionally, the student's process of study can be presented as a cyclic alternation of two stages: study of a theoretical material and test tasks fulfillment. Thus, we can divide all the learning course items into two groups: lessons and tests. Study units can be arranged in any order and quantity. Then it is necessary to construct a learning course model in which all the items will be connected in a single system. Such connections are sure to be the relations between lessons and tests. Relations can be also divided into two groups: relations between items of the same type and relations between different items. Thus, the electronic learning course can be presented as a complex relational data structure in which the items are located linearly, but the access to them and transition sequence is entirely defined by the established relations. In the context of such relational structure each item can be considered as separate object of the learning course. Hence, the learning course model becomes not only relational but also objective. Each course item possesses a set of properties. Their values are assigned by the relations connecting this item to others. For example, one of the main properties of the lesson is the ability to study it (i.e. whether a student has an opportunity to study the specified lesson now). In a classical sequential learning process model the study of the next lesson is allowed only if all the previous lessons are have been learnt. In the relational model it is not a prerequisite and can be set by the learning course author under his discretion.

One of the most important tasks during the automated learning is the tracking of the student's transitions within the set of the learning course items. For example, in the case of failure to pass a test it is advisable to restudy the appropriate material. In order to understand the restudy mechanism and links between learning course items it is required to describe structures of the lesson and test taken separately.

Each lesson in a learning course represents a set of separate topics combined by one common notion. Topics of the lesson constitute a logically completed consideration of any aspect of this concept. Thus, topics can form a complex hierarchical structure, i.e. each topic can include an unlimited quantity of subtopics. Number of topics in a lesson is not limited. Tests represent the sets of questions on various notions. Thus, each notion has its own questions set, the number of questions being unlimited. A teacher defines number of questions on each notion, included in a test. Therefore, the result of the test is not only a general mark, but also a differentiated score on each notion. During the study process this information can be used both for planning the further learning and for revising the material. For example, the test contains questions on three notions. By results of the test the student has perfectly studied two of them and has not studied the third. In that case there is no need to make the student learn anew the material studied, but it is quite enough to force him to revise the material he fails to master.

In the suggested learning course model transition sequence through the items, the restudy mechanism and links between items are realized simply enough. As it was mentioned above, each course item (including individual topics in the lesson) is an object with a set of properties. The most important lesson properties are flags, defining inclusion of the given lesson in the parent item and ability of transition to this item. Values of these flags are calculated as conjunction of elementary conditions. The developer of the learning course sets these conditions and their number is not limited. Such conditions can be, for example, the result of the specified test or the studying status of the lesson in question. The set of conditions that can be used to construct such expressions depends on specific system implementation.

Use of such links between course items allows speaking about the adaptive learning system. Such a system allows dynamic learning course generation according to the user's knowledge level. Being informed of the user's knowledge level on each notion the system can determine whether to include or exclude some topics (or lessons) of the learning course. The described behavior completely is adjusted by the course developer and weakly depends on specific system implementation.

The learning course model under consideration allows to predict student's transitions between the items because this sequence is strictly set by the course developer. In addition, there is an ability to predict backward transitions sequence because it is possible to analyze existing links to the previous items for each topic and lesson. Therefore, the suggested learning course model allows to predict user's behavior with the fine precision and to cache the required data.

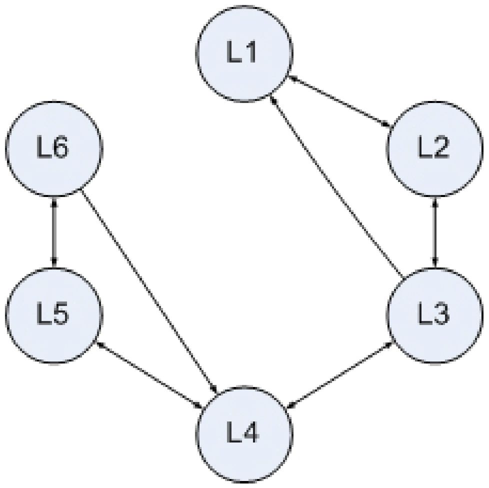

Figure 1 – Learning course model

Figure 1 schematically represents a learning course model. Arrows shows transition ability between lessons. One can see that the student cannot proceed from lesson 1 (L1) to lesson 3 (L3) without studying lesson 2 (L2). However, having reached lesson 3 (L3), the student can return to lesson 1 (L1) for material revision.

One more problem connected with absence of the fixed connection to the server is learning statistics accumulation. If the connection is available, this information is collected in real time on the system's server. But if connection is not available, local saving of this information is necessary. Then, with the advent of the server-based connection, it is transferred to the central storage. In that case, danger of transmitted data falsification arises, so, the information stored in the local files has to be encrypted. The encryption algorithm with the open key is expected to provide a high security level.

Working with the described relational learning course model, the transition prediction algorithm comes to analysis of the links between lessons and tests. Caching of the information consists in downloading the necessary lessons from the server and saving them on a local device. Considering Figure 1 let the student study lesson 4 (L4) at the moment of loss the connection to the server. In that case, according to a course structure, lesson 5 (L5) is the next to study. Hence, it should be cached first. Lesson 4 (L4) is also double linked with lesson 3 that means the student can return to lesson 3 for material revision. The next step will be the loading of lessons of the second and the following levels, counting from the current one. That is lesson 6 (L6) and 2. Lesson 1 should be the last to cache. It is necessary to note that the caching of the following lessons has the greater priority than the caching of the previous ones as the majority of students will go to the next lesson rather than to previous one.

Considering the question of development of a software system for portable computers we should not forget about the limited capabilities of the given class devices. The majority of pocket computers have limited operating and long-term memory. Low-speed data transfer channels are commonly used in wireless networks. All these limitations impose restrictions on a client part of the system.

Here it is possible to point out two basic problems: network traffic minimization and data caching optimization. Data caching optimization means disk space usage minimization. Traffic minimization is achieved by compressing the data transmitted through the network. Cached data is also stored in the packed form and unpacked only when required. Automated data caching is preferably for more powerful computers working in networks with unlimited traffic. If device features are strongly limited or the network traffic is chargeable (for example, working on the Internet) then data caching process should be supervised and adjusted manually. The adjustable parameters are:

- disk space quota used for the data cache;
- the maximum size of data downloading from the server in one session (or in the specified time interval);
- data packing method;
- manual choice of the learning course items which should be stored locally.

To increase client application usability the data caching process can be automatic, semi-automatic or manual. In an automatic mode the application independently, in a background thread, downloads all the necessary information from a server and stores it on a local device. In a semi-automatic mode the application performs caching automatically, but does not break the limits on disk space and the network traffic. In a manual mode the application offers the user only the list of learning course items which should be downloaded. In this case, the decision on loading and storing is made by the user.

It is necessary to specify which existing mobile platforms are available as we consider a mobile application suite development. First of all, various solutions of Microsoft Company in the field of mobile devices: Windows CE, Windows Mobile and PocketPC, supporting .NET Compact Framework. The Microsoft .NET Framework has been chosen as a programming platform being one of the most modern tool for mobile development.

The opportunity to use third-party libraries similar to .NET Framework is considered for system portability maintenance not only between various systems of Windows family, but also Linux, etc. The Mono Project (www.mono-project.com) is an example of such library.

Application suite architecture is also promoting portable system development. The system logic is concentrated in separate independent modules and the graphic user interface only displays the data being processed. Thus, it reduces the work necessary to port the application to a new platform.

## Conclusion

A large number of works all over the world is devoted to automated learning and knowledge control systems development. Nowadays this question is becoming more urgent because of wide spread of computer networks and mobile devices.

The automated electronic learning system users can be university students who, in addition to existing forms of correspondent learning, can receive as well more comfortable, electronic way to access learning material. Besides, employees of the large companies who needs to keep their professional skills on high level. Another application field is organization of job-seekers automated learning and knowledge control.

Using modern data transmission technologies, it is possible to speak about construction of large-scale electronic learning systems with a number of servers and various client types.

## Bibliography

[Atanov, 1993] G.A. Atanov, N.N. Martynovitch, V.V. Tokiy. The Program of the Physics Course as a Student Model. // Proceedings of the International Conference on Computer Technologies in Education ICCTE'93 – Kiev, Ukraine – 1993 – P. 138-139.

[Atanov, 1997] G.A. Atanov, G.V. Kandrashin. Frame Structure of Intelligent Tutoring System Based on the Activity Approach. // Proceedings of the 8th World Conference on Artificial Intelligence in Education. – Kobe, Japan. – 1997.

[Dillenbourg, 1992] E. Dillenbourg, J. Self. Framework for Learner Modeling. // Interactive Learning Environments. – 1992. – Vol 2. – Issue 2. – P. 111-137.

[Prosise, 2002] J. Prosise. Programming Microsoft .NET. Microsoft Press, 2002.

## Authors' Information

**Petr Rogov** – Tver State Technical University, Prospekt Lenina, 25, 170023, Tver, Russia; e-mail: rogov@elsuite.net

**Nikolay Borisov** – Tver State Technical University, Prospekt Lenina, 25, 170023, Tver, Russia; e-mail: nborisov@tstu.tver.ru