Authoring Fuzzy-based Tutoring Systems for M-Learning

Ramón Zatarain, M. L. Barrón-Estrada, Guillermo A. Sandoval-Sánchez and L. Vega-Zazueta, Carlos A. Reyes-García *

Instituto Tecnologico de Culiacan, Juan de Dios Batiz s/n Col. Guadalupe, C.P. 88220
Culiacán, México, tel. +52 667 7131796
rzatarain@itculiacan.edu.mx
* Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, Puebla, 72840, México
kargaxxi@inaoep.mx

Abstract. This paper is about an authoring tool that is being implemented to produce intelligent tutoring systems or any kind of m-learning application for different mobile devices like Cell Phones, Smart Phones and PDAs. The system is divided into three main components: The Editor of courses, which is an IDE (Integrated Development Environment) where the author create, open and save the user application; An export module to produce applications for different kinds of mobile devices and for different kinds of LMS (Learning Management Systems); and the intelligent module which adds adaptation features to the application produced by the user. Right now, the system has been tested by producing different application for Cell Phones and PDAs. We show some example of those applications.

1 Introduction

Whereas the presence of e-Learning is nowadays considered something common, it is not the same when we talk about the use of m-Learning. This is due to a couple of factors: first; we have that this technology was barely used, and second its use in the area of the education is still more recent. M-Learning is any service or facility that provides the student general electronic information and educative content that helps the process of knowledge acquisition without concerning the place or the moment [3].

E-Learning (electronic learning) has been accepted during the last two decades by several factors; one of them is the existence of students that by lack of time can not attend regular classes at an educational center; another one is that in certain regions, for some students, it is really hard to reach geographically speaking. During the time of life of e-Learning one of the bigger supports it has received has been from the Internet. The Web is a very rich atmosphere for educative investigation: the Web can provide several types of learning materials and given its powerful systems of data collection these can be added easily to web-based learning materials [4].

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 385–394

Received 19/06/07
Accepted 31/08/07
Final version 20/09/07

On the other hand, technology has changed much in the last years, and, along with this advance, the costs of the devices have diminished. Computers have become mobile, and with it a new paradigm arrives to us: "anytime, anywhere computing" [1]. This becomes patent in different areas like commerce, entertainment, communications and industry of services, operations of businesses, learning and training [2]. In the field of learning, the benefits of this new mobility are expected to means a more efficient instruction and improvement in the results of the education process [1].

In the field of computer-based learning there exists the ITS (Intelligent Tutoring Systems). The ITS systems are software programs behaving in many aspects in a similar way to human tutors [9] and they try to imitate their activities. They are used in diverse areas, such as mathematics, physics and education of languages [10]. The popularity of ITS is growing day with day due to diverse factors like increasing the performance of the students, expanding the cognitive development, and reducing the time that the student requires to acquire new knowledge [11].

The task of creating courses for mobile devices is not a simple work, and requires technical knowledge of the computer science field (design, programming, etc.). That is why even when there are specialists in topics of different fields of knowledge willing to create mobile courses this is not possible given their lack of experience in the task of programming of this type of devices.

In order to deal with this new challenge of producing adaptive learning applications, we implemented an author tool where a non-programmer instructor can more easily produce Intelligent Tutoring System for mobile devices.

The arrangement of the paper is as follows: In Section 2, we describe the software architecture of the tool. In Section 3, we discuss the implementation of the software. Tests and results are presented in Section 4. Comparison to related work is given in section 5 and conclusions and future work are discussed in Section 6.

2 Multitutor Architecture

The application is divided in two parts; the first one is a desktop program, in which tasks such as creation, edition and publishing of the courses are made, while the other is an interpreter able to visualize the course material on the target mobile (see figure 1).

Figure 1. Desktop application and interpreter

Therefore we have a version of the desktop application that always makes the same tasks, and tends to need a smaller amount of changes by its static nature, and a mobile application, able to adapt to the capacities of the mobiles and type of learning.

In the architecture of the desktop application a model of layers is used, which has three levels; being the first of them the *user layer* where the graphical interfaces appear, and where the user has a direct interaction with the author tool; the second level called *business layer* is the one that contains the logic of the application which makes all the tasks that the user asks to the system through the components that are part of the previous layer; whereas in the lowest level is placed the *data layer*,. in which the components that carry out the task of saving information in storage media are found.

This architecture benefits the system design since it allows us to have a better organized application, with cleaner code, which tends to be free of errors and easy to debug since each of the main components (user interface , system logic and data persistence) are separated in modules.

On figure 2 we can observe the layer architecture of the desktop application. We show only a few components that are part of each level.

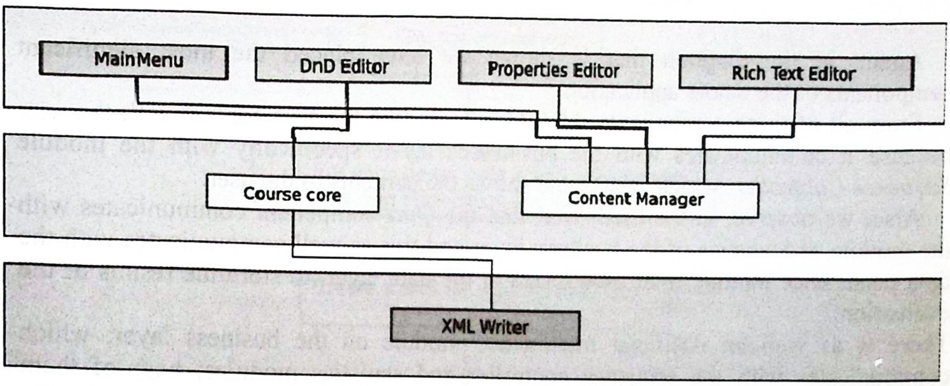

Figure 2. Desktop application architecture

We can see component *MainMenu*, which is used as the starting point in the application. From that menu, the application calls the course editor.

We also have the *DnDEditor*, which shows the course structure to the user, allowing him/her to edit the course using drag and drop actions.

On the business layer we have component *course core* used to keep communication with the *DnDEditor* component at the user level, and from where changes made by the course author are reflected. In other words, this module contains the course information.

XMLWriter is a component of the data layer which allows accomplishing the task of saving the course in storage media. Also, this module is the one that cooperates on the needed tasks to publish the courses.

As we can observe on figure 2, each of the layers can only communicate with the contiguous layer, respecting in this way the layer model, and favoring encapsulation.

With regard to the interpreter for the mobile device we can see the architecture in figure 3. It has been chosen, like in the desktop application architecture, a model of layers. This architecture has been chosen for being fast to implement, easy to debug and to favor its maintainability, being this a critical point in the application.

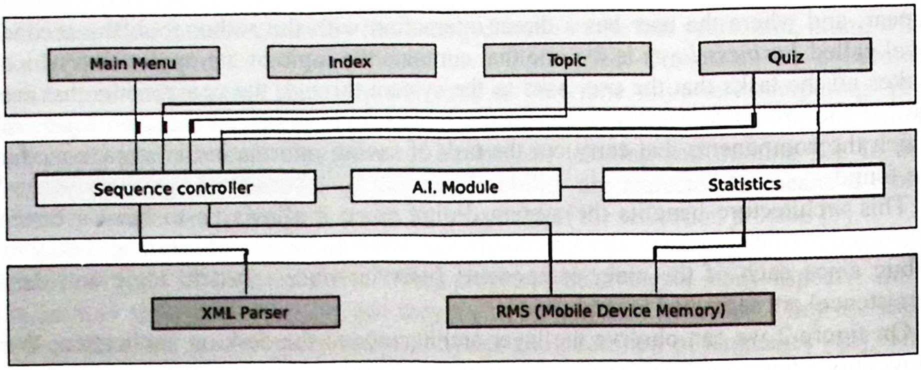

Figure 3. Mobile interpreter architecture

Again, in the diagram that is shown we have placed the most significant components of the whole application.

From all of these components, *MainMenu* acts like the first contact with the user because it communicates with the businesses layer, specifically with the module *Sequence Controller*, which selects and shows the contents to the user.

Also, we observe, in the user layer, that the *Quiz* component communicates with the module of Statistics of the business layer and this as well communicates with the data persistence module (that uses RMS) in the data layer, to store the results of the evaluation.

There is as well an Artificial Intelligence module on the business layer, which communicates with the sequence controller and statistics modules; both of them provide the intelligent module with information about the user interaction with the course; with that information this module performs several tasks (using fuzzy logic techniques) to determine which information is shown and how it is going to be shown to the user according to the type of learning.

3 Implementation

Multitutor Móvil is a multiplatform independent implementation which uses Java and XML technologies. Java has been chosen as the language for the desktop application to take advantage of the multiplatform support, which helps the application to work on a wide range of computers without requiring neither complete nor partial reprogramming.

In order to implement the persistence of data, XML has been used, because of its flexibility for generating future extensions to the project; besides XML provides standard ways to communicate with other applications.

We need to mention that when a user decides to initiate the authoring of a new course; a user interface is shown from which the structure of the course must be created, as well as the existing relations between each one of the objects that compose it. This task is made through an action of drag & drop.

When the user finishes the edition of the course he/she continues with the exporting process. This task at user level is carried out with only selecting the export option at the project menu.

For implementing those operations, the first step is finding the root of the course (which internally is represented like a tree). Once it has been located the program begins to traverse the tree and to extract the contents and properties of each of the nodes (which are elements of the course). Each of the elements is written on one or several XML files which reflect the information and the attributes of them.

When the process of writing all of the XML files has finished, it proceeds to create a package, by compressing the information into a *JAR* file and adding the mobile device interpreter to it. Figure 4 illustrates the process made by the exporting course algorithm.

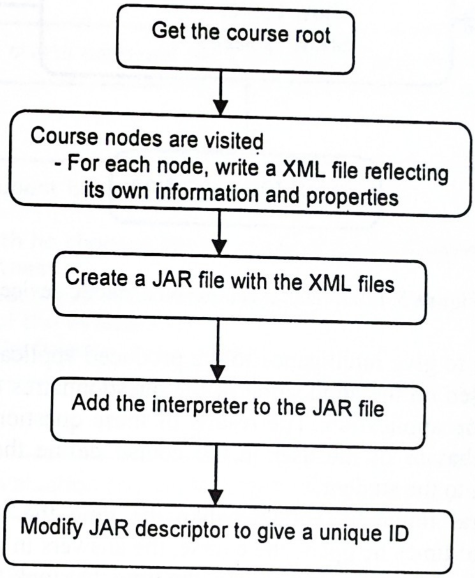

Figure 4. Exporting course algorithm

After exporting a course we need to copy it to the mobile device memory for its installation. The process of copying is made in different ways depending on the mobile device and the operating system running in the desktop computer. Installing the software provided by the manufacturer of the mobile device is usually enough to accomplish the communication with the computer.

Once the program is in the mobile device, the next step is to install it. When the process is over it will be possible to locate it in the list of applications that the device recognizes as installed in its memory.

For executing the course the interpreter takes control of this process by reading and showing the contents of the XML files (see figure 5).

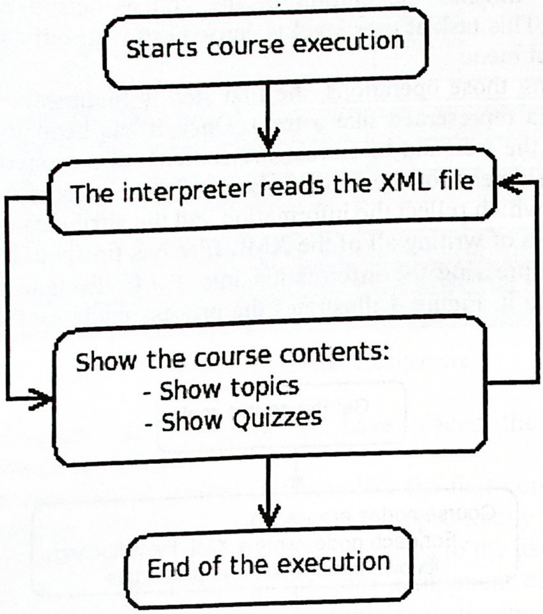

Figure 5. Interpreter execution on a mobile device

There is a module to give intelligence to the produced applications. The operation of that module is based on the application of the questionnaires that previously were defined in the desktop application. The results of these questionnaires, added to an evaluation of the behavior of the user in the course define the way in which the information is shown to the student.

We defines several linguistic variables like the time the user spends in each subject, the number of times he opens the course, the answers in questionnaires (if the correct answer was the first or second choice, the time that took the user to answer it, etc.). Fuzzy values are given to each of the evaluated points or linguistic variables. We assign weighted values to each variable involved on the choice of the course path to be followed. In other words, our student model is like a two dimensional array of linguistic variables, corresponding to different combinations of intelligences (logical/mathematical, linguistic/verbal, visual/spatial, etc.) and learning goals, where we stores the weighting factor that indicates the students preferred learning characteristics. The information stored in the student model includes the navigation history, the time spent on each learning unit, the answers given to questions and the different sections covered in the learning units.

For example, a student will receive a first quiz with 18 questions. Each question has its weighted value for each type of intelligence where the weight is assigned depending on the answer. For each question two possible answers exist that are designed to be opposite among them. In case that the user does not respond a question we will not compute this question.

Our formula for computing the type of intelligence to a given user is:

If (user shows a linguistic answer in question$_m$) then
LinguisticVar = LinguisticVar + (((LinguisticVar * value of question$_m$))*100)/ # (of linguistic answer number)

Later, the student is evaluated again but now according to different variables like spent time in a subject or unit, path the student choose, number of times he spent in a given subject, and other quizzes he/she takes at the end of a subject or unit. An important point here is that the author (an instructor) decides the elements of evaluation, the normal time in a topic, and other variables that are used for building the most preferred type of learning of any student.

The course of action in which the course adapts to the user intelligence depending of different attributes obtained during the learning process is illustrated in figure 6.

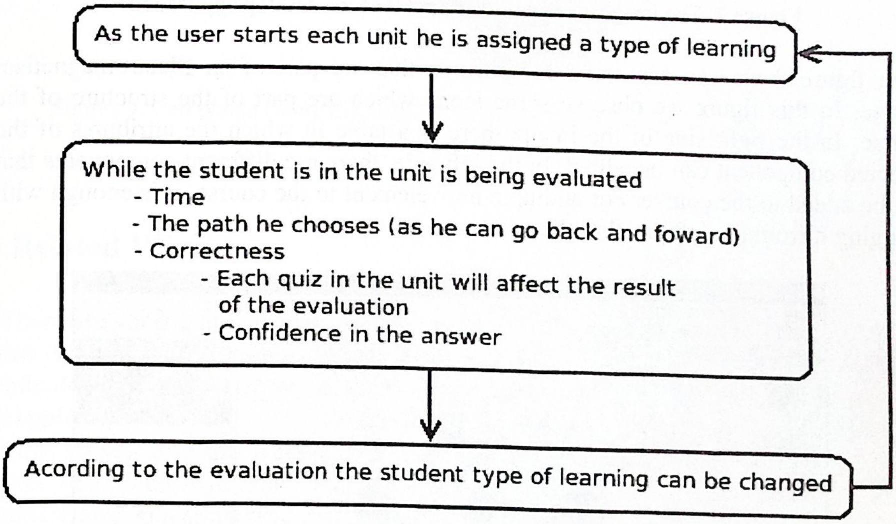

Figure 6. Learning type selection algorithm

4 Test and Results

In this section, we present some figures that illustrate the main interfaces of the Multitutor author tool. Beside we give an example of an application created with the tool. As we can see, Figure 7 shows the interface of the main menu. We can see several options, such as the creation of new courses, or opening some of the existing ones for its edition or publication.

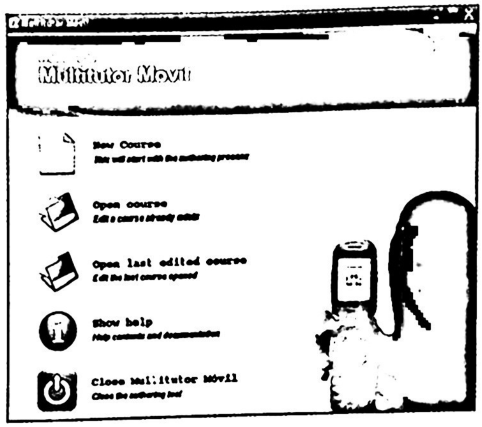

Figure 7. The image shows the entry point of the desktop application.

In figure 8 we can see some components that are part of an Electromagnetism course. In this figure we observe some icons which are part of the structure of the course. In the right side of the image there is a table in which the attributes of the selected component can be edited. In the left side, there are different components that can be added to the course. For adding a new element to the course, it is enough with dragging it from the left panel to the edition panel.

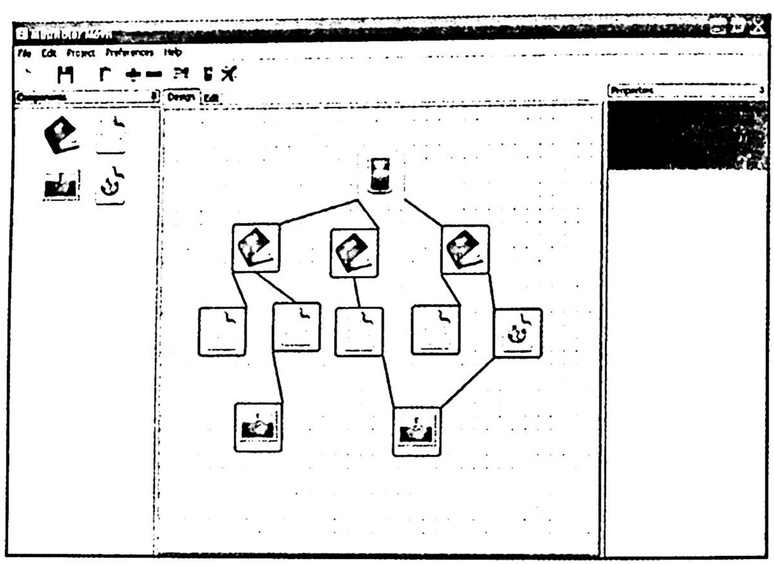

Figure 8. The structure of an electromagnetism course

A course on electromagnetism was created and distributed with students of the Instituto Tecnológico de Culiacán. Students used the course as a source of information for reviewing the topics of the course. The course was tested and successfully ran in different devices, such as PDA's, PocketsPC's and cell phones owned by the students and instructors of the institution.

Figure 9 show part of the course in different mobile devices.

Figure 9. Cell phone (Sony Ericsson k750) running an electromagnetism course

5 Related Work

There are several author tools used to create mobile applications. *MyLearning* [12] which is developed by *m-learning.org* is one of them. *MyLearning* is used for creating mobile device courses. However, the variety of mobile devices which are the target of this application are PocketPC's, which are a type of mobile devices that have technical features greater than the average, and represent a much smaller p of penetration in the market compared with the cellular. On the other side, *Hot Lava* [13], another commercial application oriented to m-learning, has as main disadvantage that the market of the produced applications does not include cell phones. In addition, both programs do not have the capacities of adaptation to the form of learning of the user.

6 Conclusions and Future Work

Multitutor is an author tool that allows an instructor to produce intelligent applications for mobile devices. The author can produce mobile an application from a desktop computer. The application can include different components like text, images, and questionnaires as well as learning features according to the type of intelligence of the student. One feature of the tool is simplicity, because it allows a fast creation of mobile applications. The software has been implemented in Java version 1.6 and XML. So, we can use Multitutor in any device where a J2VM is installed.

Right now we still are working and testing the intelligent tutoring module to each of the produced courses.

References

1. Luis de Marcos, José Ramón Hilera, José Antonio Gutiérrez, Carmen Pagés, José Javier. Martínez. Implementing Learning Objects Repositories for Mobile Devices. Department of Computer Science, University of Alcalá. Alcalá de Henares (Madrid). Spain.
2. Eilif Trondsen. Global Perspective on mLearning. SRI Business Intelligence. San Francisco. Febrero del 2005.
3. Lehner, F. and Nosekabel, H. The Role of Mobile Devices in E-learning - First Experience with a E-learning Environment. IEEE International Workshop on Wireless and Mobile Technologies in Education. Los Alamitos, USA: IEEE Computer Society. 2002
4. Harri Ketamo, mLearning for kindergarten's mathematics teaching. Tampere University of Technology, information technology, Pori. IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'02). 2002
5. Connolly Thomas and Stansfield Mark. Using Games-Based eLearning Technologies in Overcoming Difficulties in Teaching Information Systems. University of Paisley, Paisley, Scotland, UK. Journal of Information Technology Education. Volume 5, 2006
6. Quinn, C. M-Learning. Mobile, Wireless, in-your-pocket Learning, http://www.linezine.com/2.1/features/cqmmwiyp.htm Marzo del 2005.
7. Houser, C., Thornton, P., Kluge, D., Mobile Learning: Cell Phones and PDAs for Education. Internartional Conference on Computers in Education ICCE'02. London, UK. 2002
8. Shepherd, C. M is for Maybe. http://www.fastrak-consulting.co.uk/tactix/features/mLearning.htm Marzo del 2005.
9. Edward R. Sykes, Franya Franek. Web-Based Architecture of an Intelligent Tutoring System for Remote Students Learning to Program Java.
10. Woolf, B., P., Beck, J., Eliot, C., & Stern, M. Growth and maturity of intelligent tutoring systems: A status report. Smart machines in education. Cambridge, MA: MIT Press.
11. Graesser A. C., Person, N. K., & Harter, D. (2001). Teaching tactics and dialog in autotutor. International Journal of Artificial Intelligence in Education, 12, 12-23.
12. Attewell, J (2005). Mobile technologies and learning: A technology update and mlearning project summary. London: Learning and Skills Development Agency. Available online at www.m-learning.org/reports.shtml. June 15, 2007.
13. McConatha, D., Praul, M. Mobile Learning in the Classroom: An Empirical Assessment of a New Tool for Students and Teachers, http://www.hotlavasoftware.com. June 15, 2007.