Memorization and training activities in mobile devices

Ainara Urrutia**, Naiara Maya**, Jõsune Gereka**, Ohian Odrioza**,
Jon A. Elorriaga*, Ana Arruarte*

*University of the Basque Country (UPV/EHU)
**Gipuzkoako Ikastolen Elkarte
E-mail: a.arruarte@ehu.es

Abstract:
This paper presents IRAKAS, an m-learning system that provides support for the whole cycle of memorization and training activities in a wide range of domains. The paper is focused on the development of learning materials.

1. Introduction

Memorization is a learning strategy that allows an individual to recall important information verbatim. Although memorization draws upon one of the most fundamental human faculties, it is one of the least exercised techniques in contemporary education. In order to train this ability several memorization and training exercises must be carried out. Those exercises involve working with numbers, letters, syllables, words, signs, drawings, sentences and texts. Solving these activities demands a great effort of attention and concentration. New technologies allow the creation of m-learning systems for doing this kind of exercises.

Information and communication technologies and mobile devices are increasingly recognized as strategic tools that have the potential to enable global access to educational materials [1]. Quinn [2] defined m-learning as “learning through mobile computational devices”. Chabra et al. [3] perceived m-learning as “the ability to receive learning anytime, anywhere and on any device”. M-learning is the term given to the delivery of learning by means of mobile devices [4].

Along this paper the project called IRAKAS is presented. IRAKAS is a multilingual system that provides support for the whole cycle of memorization and training activities in a wide range of domains. First, it allows instructional designer to author those activities using a web application. As output this application produces the DU. IRAKAS Trainer is an application that allows students to solve the memorization and training activities of a DU in a mobile device. As output it creates the SD. In this project a prototype of a mobile device has been developed. In addition IRAKAS Trainer can run in a traditional PDA.

IRAKAS Inspector is a web application to evaluate the learning process of each student. This application must load the DUs that are going to be used. It takes the SDs as input and visualizes graphically the information about the students’ learning progress as well as student groups’ progress. It also allows teachers to change the system data.

2. IRAKAS System Architecture

The architecture of IRAKAS framework is shown in Figure 1. It is composed of the activities manager system called IRAKAS Designer, a mobile module called IRAKAS Trainer and finally the follow-up system called IRAKAS Inspector. Physically there is one IRAKAS Designer server and several IRAKAS Inspector servers, one for each school.

Data is transferred from one module to another using an external memory device. Current version of IRAKAS based training does not need internet connection. The exchanged information is organized in Didactic Units (DUs) and Student Data (SD) stored in XML files. A DU represents a collection of exercises for a domain organized in a pedagogic way. A SD represents the information about the evolution of the learner and more general information.

IRAKAS Designer is a web application that provides the instructional designers with an environment for the creation of different types of memorization and training activities. As output this application produces the DUs.

IRAKAS Trainer is an application that allows students to solve the memorization and training activities of a DU in a mobile device. As output it creates the SD. In this project a prototype of a mobile device has been developed. In addition IRAKAS Trainer can run in a traditional PDA.

IRAKAS Inspector is a web application to evaluate the learning process of each student. This application must load the DUs that are going to be used. It takes the SDs as input and visualizes graphically the information about the students’ learning progress as well as student groups’ progress. It also allows teachers to change the system
parameters to personalize the students training by means of tuning the activity sequencing.

3. IRAKAS Designer

IRAKAS Designer is an activity manager application that allows development, organization and export of exercises in DUs. The created activities are stored in a database managed by MySql. PHP programming language has been used for the development of the application. Next the functional modules of IRAKAS Designer are described (see figure 2).

3.1. Domain Acquisition

The main function of the domain acquisition module is to help the user in the creation of the structure of each DU and to offer the necessary tools to fill it with exercises (see figure 3).

DU structure is hierarchically organized in six levels. Each learning domain is attached to a specific Area that will be the root of the unit (for example, geography). Each area is made up of Content Blocks (e.g. rivers, mountains, …). The content blocks are composed of different Modules (e.g. rivers of Europe, rivers of Africa, rivers of America, …). Each module contains Multilevel Packages, those include Levels organized by difficulty. Finally, Levels include Exercises without any order.

IRAKAS Designer offers several exercise types to complete DU. They are classified into two different groups of exercises: static exercises and generative exercises. In static exercises the content designer specifies the whole exercise definition parameters meanwhile in generative exercises only some exercise creation conditions are defined. In other words, the program that runs in the mobile device is able to generate exercises in real time taking into account the conditions that are defined by IRAKAS Designer. Among others exercise types that IRAKAS includes multiple choices, fill-in-table, word completion, ordering exercises, crosswords: wordsearch puzzle, sudoku, matching columns, classification …

All the exercise types share common attributes such as feedback (the message that the learner will receive when the exercise is corrected), question wording and help (clues to solve the exercise). On the other hand, there are other fields related to each exercise type. For example, in the case of mathematical domains, a generative exercise can have the following characteristics: the operation, the maximum and minimum value, the specific value and others.

Figure 2. Architecture of IRAKAS Designer

Figure 3. Creating a multilevel pack
In addition to exercise type, activities sequencing is important too. In order to guarantee a proper learning environment that respects each student learning rhythm and capacity, some sequencing parameters have been defined in the multilevel packages. Those parameters include: the repetition factor of correctly and incorrectly solved exercises, the percentage of exercises the student has to do at least at each level and the percentage of exercise solved correctly per level to allow the student to start with the following level. Thus, the teacher can define different learning rhythm for each student. Those parameters can be defined and changed by the teacher by means of IRAKAS Inspector and are stored in the file which stores the student data and exchanged with IRAKAS Trainer.

In generative exercises some parameters are not fixed so that IRAKAS Trainer chooses values randomly while sequencing the exercises in the training session. In these cases the same exercise can be repeated several times, and every time the parameters will have different values (for example exercises with numbers that are chosen randomly).

In addition, IRAKAS Designer allows including images in some exercise types. When creating a multilevel package there is the possibility of choosing the same image for all the exercises of that package. Nevertheless it is possible to change/remove/add a different image to each level or each exercise of this multilevel package.

The tool also offers the possibility of managing previously created exercises. For example, it is possible to change or copy exercises from one level to another one and also to create new exercises from existing ones.

3.2. Domain control

In order to achieve correctly structured DUs IRAKAS Designer provides a module to validate these units. The following controls are performed: (1) The exercise type is defined at the multilevel package, therefore the control module checks that there is no exercise of other types in the package. (2) It is verified that the stored information is suitable and enough for the correct execution in IRAKAS Trainer. (3) It is verified that the structure of the unit is correctly created and, that exists at least one exercise of the corresponding type in each level.

3.3. Didactic Unit Generation

Once the DU is correctly created this information is processed by the module in charge of translating and exporting these data from the database so that IRAKAS Trainer can understand and use them. The resulting DU will be stored in the external memory device and finally, it will be loaded in the mobile device (Figure 4) and also in IRAKAS Inspector.

3.4. Administration

This module allows the user to administrate the database and the users of the system.

5. Conclusions

The project IRAKAS presented in this paper has its focus on the creation of a learning system which provides to create memorization and training activities, to dispatch them in a mobile device, and to evaluate the student progress.

Although the whole system is not yet evaluated in a real school-wide context each component has been already individually tested. IRAKAS Designer has been tested in the context of language and mathematics. Concretely, an instructional designer has created a real DU composed of more than 1,400 exercises to train orthography in Basque language. Those activities have been simulated in two mobile devices: Zaurus SL and IRAKAS Trainer and the results managed in IRAKAS Inspector successfully.

Acknowledgments: This work is funded by the University of the Basque Country (UE06/19), the CICYT (TIN2006-14968-C02-01) and the Gipuzkoa Council in collaboration with the European Union.

6. References