

A Mobile App for Adaptive Test in Intelligent Tutoring System Based on Competences

Miguel BADARACCO ^{a,1}, Jun LIU ^{b,2}, and Luis MARTINEZ ^{c,3}

^aNational University of Formosa. Faculty of Economics and Business Administration.
Formosa, Rep. Argentina

^bSchool of Computing and Mathematics, University of Ulster, Newtownabbey BT37
0QB, Northern Ireland, UK

^cUniversity of Jaén Spain. Department of Computer Science, Campus Las Lagunillas,
s/n, E-23071 Jaén, Spain

Abstract. An Intelligent Tutor System (ITS) aims to customize teaching processes dynamically according to student's profile and activities by means of artificial intelligence techniques. The use of Competency based Education as pedagogical model has developed the ITS based on Competences (ITS-C). The architecture of an ITS-C uses a modular structure that optimizes the computational requirements for its implementation. Such modular architecture facilitates its use with ubiquitous devices. Therefore ITS-C or several of its processes may be ubiquitously offered to teachers and students. In this contribution a mobile app for adaptive test in ITS-C is introduced.

Keywords: Intelligent Tutoring Systems, Competency Based Education, Mobile Learning, Computerized Adaptive Tests.

Introduction

An ITS provides direct customized instruction or feedback to students in their learning processes by means of Artificial Intelligence (AI) techniques, being mainly applied to knowledge representation, managing an instruction strategy as an expert both in the teaching and pedagogical issues in order to diagnose properly the student learning status at any time. To fulfill its objective, an ITS is organized by an architecture composed by a domain model (what is taught?), student model (who is taught?), diagnosis of the student [1], instructional model (how is it taught?) [2] and the interface (man-machine interaction) [1], [3] (see Figure 1).

In [4] is presented an architecture for ITS based on Competency-based Education (CBE), which is characterized by a modular structure to represent the domain model and the student model. The updating process (student's diagnosis) based on Computerized Adaptive Test (M-CAT) [5] is an important module that is used during the student's learning process in which it is necessary to choose suitable items, modeling experts' knowledge regarding the usefulness of items. These features allow

¹ Tel.: +54 03717 451702; e-mail: itfor@arnet.com.ar

² Tel.: +44 28 90368919; e-mail: j.liu@ulster.ac.uk

³ Tel.: +34 953 211902; Fax: +34 953 212472; e-mail: luis.martinez@ujaen.es

optimizing performance and computational requirements of this module during its implementation. Therefore, due to the fact that adaptive tests during the learning process are common and frequent, it seems a potential process to be carried out in an ubiquitous way by means of mobile devices.

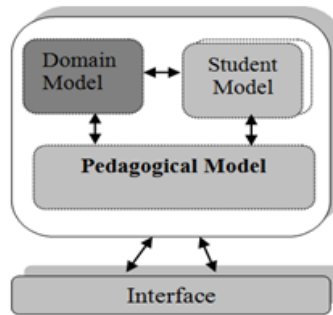


Figure 1. Generic architecture of ITS

This contribution presents an application developed on a mobile platform (Android) that allows deploying the architecture of an ITS's-C and implementing the diagnosis by Computerized Adaptive Tests of modular and distributed way using tablets and smartphones.

The paper is organized as follows; Section 2 reviews the ITS-C architecture and structure. Section 3 presents a Fuzzy Computerized Adaptive Tests for mobile devices (FM-CAT), its architecture and functionality. Finally some concluding remarks are pointed out.

1. Intelligent Tutoring Systems Based on Competency-based Education. Architecture and Diagnosis Process

An ITS-C extends ITS by linking the latter and the pedagogical model based on Competency-based Education (CBE) using the architecture showed in Figure 2 [4]. The domain model, student model are reviewed and then a further detailed revision of the diagnosis of an ITS-C is presented to facilitate the understanding of the proposal introduced in section 3.

1.1 Domain Model of ITS-C

The representation of the *domain model* in an ITS-C is based on the descriptors utilized in CBE [4] that reflect good professional practices to guide the development of the competency associated with an occupational role or profile [6], [7], [8]. Such a set of descriptors are:

- *Competency unit (cu)*: It is a main function that describes and groups the different activities concerning the role or profile chosen.

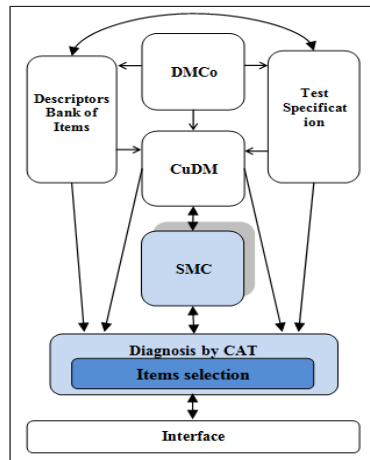


Figure 2. ITS-C Architecture

- *Competency element (ce)*: It is the disaggregation of a main function (*cu*) that aims to specify some critical activities. A function (*cu*) can be specified by one or more competency elements (*ce*), according to its complexity or variety.
- *Evidence of performance (evd)*: It checks if a process is performed according to best practices.
- *Evidence of product (evp)*: It is a descriptor of tangible evidence in the results level, when the best practices have been used.
- *Evidence of knowledge (evk)*: It is a descriptor about scientific-technologic knowledge that allows the user understands, reflects and justifies competent performance.

Therefore the *domain model* contains the expert's competences profile about a knowledge domain, hence for an ITS-C it will consist of four components briefly detailed below, further description see [4]:

- A domain model of competency (DMCo)*: It is represented by a semantic network whose nodes are competence units (*cu*), competence elements (*ce*), descriptors (*evd*, *evp*, *evk*) and their relations.
- A curriculum domain model (CuDM)*: It deploys the DMCo according to a teaching strategy that defines the competences associated to a professional profile to perform a training proposal in different situations. The CuDM based on the CBE takes a modular structure, in which each module (M_i) contains competency elements (*ce*) belonging to the DMCo.
- A set of descriptors*: The descriptors associated with the *ce* of the didactic modules are *evd*, *evp*, and *evk*, that belong to a bank of items.
- Test specifications*: They are provided by the teachers and associated with the diagnosis process considering the scope of application and the rules that the system should follow to propose adaptive tests according to the student's necessities of learning.

1.2 Student Model of ITS-C

In an ITS-C the *student model of competence (SMC)* stores student’s information, whose data are updated through a diagnosis process. For the representation of the student’s knowledge and learning process, the SMC uses an overlay model in the semantic network of the CuDM [9].

In such a semantic network the nodes *evp*, *evd* and *evk* store a probability distribution $P(\theta_{evp} = k | \vec{u}_i)$, $P(\theta_{evd} = k | \vec{u}_i)$, and $P(\theta_{evk} = k | \vec{u}_i)$ regarding the student’s level of competency *k* in the corresponding node, *k* can take values from 1 to the maximum number of level of competency on which the student is evaluated. Being θ the student’s level of technical-scientific knowledge about a descriptor for a response pattern \vec{u}_i obtained from the responses provided by the student in the test *T* (See Figure 3) during the diagnosis process.

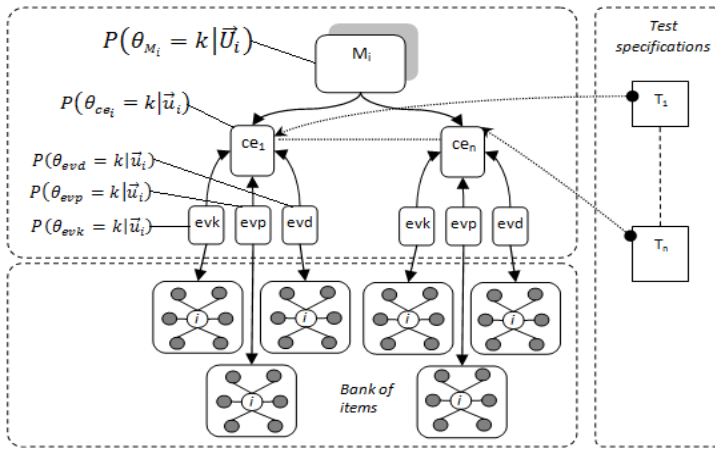


Figure 3. Module of Student Model of Competence

1.3 Diagnosis for ITS-C based on CAT

The diagnosis process estimates and updates the level of competency achieved by the student in the nodes of the SMC. To carry out the diagnosis of an ITS-C, it was adapted and extended the Computerized Adaptive Test (CAT) based on the Item Response Theory (IRT) [10]; [9]. In CAT systems the relationship between student outcomes in the test and its response to a certain item can be described by a monotone increasing function called the Item Characteristic Curve (ICC). The ICC of an ITS-C coincides with the correct response option of the characteristic curve of option (CCO). Its main components are:

- *A response model associated to the items:* It describes the student’s expected performance according to his/her estimated knowledge. An ITS-C uses a *discrete* and *non-parametric* response model based on the Item Response Theory (IRT) [11] able to evaluate multiple choice answers [10].
- *Bank of Items:* Each item I_i is associated to its descriptors (evd, evp or evk) and each option of I_i corresponds to a characteristic curve of option (CCO) obtained by a calibration process based on the Ramsay algorithm [12]. Each

CCO is represented by a probability distribution, $P(\vec{u}_i|\theta_0)$, where each component represents the probability that the student selects the response pattern \vec{u}_i , given her level of competence θ .

To develop a test the teachers must provide *test specifications* considering the scope of application and the student’s necessities of learning, namely:

- i) *Initial level of Knowledge*: The initial knowledge estimation is crucial because determines the length of the CAT for each student. It may be estimated by using different models based on previous information.
- ii) *Criterion for selecting descriptor (evp, evd or evk)*: The algorithm selects the descriptor that has the level of knowledge associated with lower probability [10]; [9]:

$$\min(\theta_{ev}) = \min(\text{MAP}(P(\theta_{ev}|\vec{u}_n))) \tag{1}$$

- iii) *Criterion for selecting items*: The adaptive CAT mechanism uses different methods to select the items for the test, a common method is the maximum information [13]; [10] that selects the item which maximizes the information in the provisional distribution of student’s knowledge. The information function for the item, I_j , is calculated as follows:

$$PI_j(\theta_i) = \frac{(P'_j(\theta_i))^2}{P_j(\theta_i)(1 - P_j(\theta_i))} \tag{2}$$

Being θ_i the knowledge level of the student i , $P_j(\theta_i)$ the value of the CCO for the student’s level, and $P'_j(\theta_i)$ the function derived from the CCO at that point. Other selection criteria were proposed in [13]; [10].

In [9] the item selection process uses experts’ knowledge, linguistically modeled, to characterize the usefulness of an item. Whereby the selected item is the maximum usefulness $\max(X_i^k)$.

- iv) *Stop criterion*: The test should stop when the student achieves a level of knowledge fixed a priori, though there are other criteria.

During the management of a test, the student's knowledge is estimated every time that he/she answers a question, by updating the student's knowledge distribution [13], as:

$$P(\theta_{ev}|\vec{u}_1, \dots, \vec{u}_i) = \begin{cases} |P(\theta_{ev}|\vec{u}_1, \dots, \vec{u}_{i-1})P(\vec{u}_i|\theta_o)| & \text{if } Q_i \text{ assesses } evd_j, \\ evk_j \text{ or } evp_j. & \\ P(\theta_{ev}|\vec{u}_1, \dots, \vec{u}_{i-1}) & \text{in other case.} \end{cases} \tag{3}$$

Being $P(\theta_{ev}|\vec{u}_1, \dots, \vec{u}_{i-1})$ the a priori student’s knowledge estimation on *evd*, *evp* or *evk*, and $P(\vec{u}_i|\theta_o)$ the CCO for the option of the response pattern.

After the updating process the system estimates the level corresponding to the distribution by using one out of two choices introduced in the CAT [13]; [10]:

- i) *Expectation a posteriori (EAP)*:

$$\theta_{ev} = EAP(P(\theta_{ev}|\vec{u}_n)) = \sum_{k=1}^n kP(\theta_{ev} = k|\vec{u}_n). \tag{4}$$

Being k the knowledge level.

ii) *Maximum a posteriori (MAP)*:

$$\theta_{ev} = MAP(P(\theta_{ev}|\vec{u}_n)) = \max P(\theta_{ev} = k|\vec{u}_n) \quad (5)$$

The competency level θ_{ce} is computed as:

$$\theta_{ce} = |k_1P(\theta_{evd} = k_1|\vec{u}_n) + k_2P(\theta_{evp} = k_2|\vec{u}_n) + k_3P(\theta_{evk} = k_3|\vec{u}_n)|, \quad (6)$$

being $P(\theta_{evd} = k_1|\vec{u}_n)$, $P(\theta_{evp} = k_2|\vec{u}_n)$ and $P(\theta_{evk} = k_3|\vec{u}_n)$ the probability regarding the descriptors evp , evd and evk and k_1 , k_2 and k_3 the competency levels respectively.

The competency level θ_M for the node M is computed by using the values of the nodes ce .

$$\theta_M = \sum_{j=1}^n |k_jP(\theta_{ce_j} = k_j|\vec{u}_n)|, \quad (7)$$

where $P(\theta_{ce_j} = k_j|\vec{u}_n)$ is the probability regarding the node ce_j and the competency level, k_j .

As it was pointed out in subsection 1.2, for the representation of the student's knowledge and learning process, the SMC uses an overlay model in the semantic network of the CuDM [9] whose structure is organized modularly, whereby an ITS-C might implement a competency domain gradually assimilating each module as "learning capsules". In 1.3 was discussed the diagnosis based on CAT which improves the efficiency, adaptability and evaluation length which results in lower computational requirements.

Therefore, with the exponential growth of the use of mobile devices in all social levels and the potential ubiquitous access to educational resources and tools open up new possibilities for educational innovation to democratize access to learning intelligently and independently of time and space. So in the following section it is presented an application developed on a mobile platform (Android) that allows deploying the architecture of an ITS's-C and implementing the diagnosis by Computerized Adaptive Tests of modular and distributed way using tablets and smartphones.

2. A Mobile Adaptive Test for ITS-C

In this section are presented the features of the Mobile Computerized Adaptive Test (M-CAT) application: firstly it is showed its architecture, technologies and profiling of the users used for its implementation. Eventually, it is described the functionality and performance of the M-CAT.

2.1 M-CAT application characteristics

First we will briefly refer *mobile learning* and *ubiquitous learning*, supported on these concepts; we will discuss the idea of generic architecture to adopt in the development and implementation of M-CAT.

Mobile Learning happens when the student is away from his usual place of learning. One consequence of the use of mobile devices is that learning can take place in places of interest [14], [15], but also that it is possible to redefine and provide experience in formal environments. In this way, mobile learning is linked to the paradigms “anytime” and “anywhere”.

Ubiquitous Learning facilitates that every place can result in a learning space for the user that intelligently connects both objects, people, devices and even other learning spaces, and which may appear learning opportunities tailored to the context and personalized to user. One way to think about the difference between mobile learning and ubiquitous learning is that while mobile learning takes computers out of the classroom to the world, in the ubiquitous learning the world becomes the classroom and on the computer [16].

Ubiquitous Learning Environment based on ITS-C is understood as a virtual space-oriented and aimed at the acquisition of knowledge (independent of time and space) through the deployment of an ITS-C and the implementation of some components. Both students and teachers have access to the services offered by various mobile devices.

Defined the underlying concepts, then we describe briefly the generic architecture which supports our proposal.

2.1.1 M-CAT. Generic architecture

The generic architecture is organized in three layers or levels that operate on a technological infrastructure (Figure 4). The components and levels are: a) Technology infrastructure. b) Deployment level. c) Implementation and distribution level. d) Interaction level.

Each level offers a number of benefits and services according to the user's profile.

The *Technology infrastructure* of M-CAT was developed to run on mobile devices (smartphones and tablets) on the Android platform, the main features of the architecture are:

- The structure of the Android operating system consists of applications running on a Java framework for object-oriented applications on the core Java libraries on a Dalvik virtual machine with runtime compilation. Libraries written in C include a GUI manager (surface manager), an OpenCore framework, a relational database SQLite, a programming interface API OpenGL ES 2.0 3D graphics, a WebKit rendering engine, SGL graphics engine, SSL and Bionic C standard library.
- Most applications are written in Java, there is not a Java virtual machine on the platform. The Java bytecode is executed, but is first compiled into a Dalvik executable and run on Dalvik Virtual Machine Dalvik. Dalvik is a specialized virtual machine designed specifically for Android.

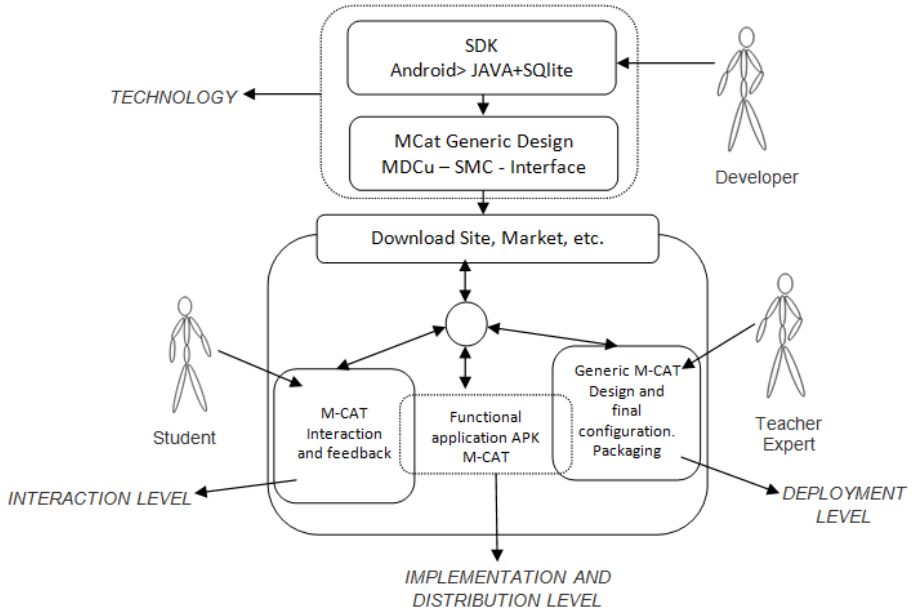


Figure 4. Architecture and technology used in M-CAT.

- We have used the Java language and for persistent storage of data we have implemented a SQLite database.
- The development of generic design of M-CAT was made on Android using the SDK. The result is an APK that can be downloaded by the expert teacher for parameterization and distribution (Fig. 4). Or a project package that can be worked through by Android Development Tools (ADT) plugin for the Eclipse integrated development environment (IDE).
- The Android application development does not require complex programming languages to learn. All that is needed is an acceptable knowledge of Java and access the software development kit (SDK) provided by Google which can be downloaded free. All applications are compressed into File Application Package (APK package is a variant of the JAR format Java and is used for distributing and installing bundled components for the Android platform), which can be easily installed from anywhere in the file browser most devices.

Deployment level: it facilitates to the expert teacher designing M-CAT on some domain of knowledge, including the construction of the item bank and operating parameters. On this level, the expert teacher is responsible for designing parameterize M-CAT, provides all the information necessary for implementation:

- Parameters of the M-CAT, after downloading the M-CAT the teacher has two alternatives of parameterization:
 - From a mobile device: using the functionality of the interfaces provided by the application (Figure 5), which have to enter data for the modules (quantity, description), evaluation criteria and ending, etc. Finally, should configure the item bank, item quantity,

description, answer choices, etc. As well as to assess the usefulness of each items in accordance the criteria established in the model.

- The other possibility is to work directly on the project package (through integrated development environment that includes ADT, such as Eclipse) and incorporate directly the SQLite database respecting the prescribed format for M-CAT.

The next step consists in the generation of the FM-CAT (APK application) for distribution.

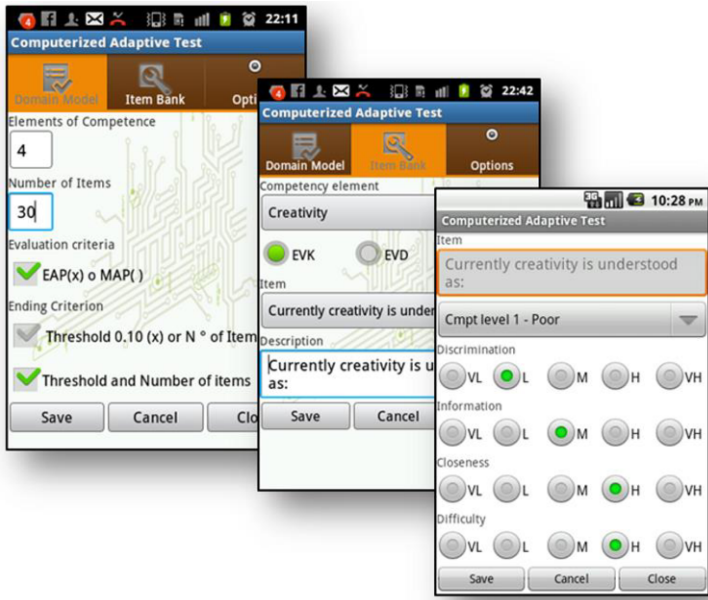


Figure 5. M-CAT -Teacher interface features.

Implementation and distribution level: After defining the specifications for the deployment of ITS-C, the expert teacher generated mobile client application for the students.

- Packaging and distribution: then distribute it for students use.
 - According the specifications set by the teacher, the packaging can build the application ready to be installed on a device, this application will consist of a client application and the deploy data in the ITS-C.
 - The layout provides the application. It can be done via Android Market, download site or any means of distribution.

Interaction level: This level chases two main aims, first to allow students access on their mobile devices to M-CAT able to assess competencies modularly with feedback to fingertips, second to facilitate teacher to receive feedback from students and analyze information of the interactions:

- Student. It is the ultimate receiver of the M-CAT, download and install the application, perform the proposed test and interaction (Figure 6). He is admitted to:
 - Check status of student model.
 - Select competency element to assess.
 - Set up system options.
 - Perform the M-CAT.
- Teacher. Receives feedback information and has the possibility to analyze the process and outcomes of the experience.
 - Receives and processes information feedback.
 - Check status of student model of each student.
 - Generate statistics and reports.
 - Executes processes adjustment for future versions.



Figure 6. M-CAT –Student interface features.

2.2 Performance

Here an illustrative example that deploys the modular architecture of an ITS-C and implements the diagnostic by a Mobile Computerized Adaptive Test (M-CAT) developed by using an Android platform.

Let us suppose that is being evaluating a student by an M-CAT in evidence node ev_k , whether the probability distribution of competence levels and the correspondent characteristic curves are defined. Therefore, the process is carried out as follows:

It is selected item with $\max(X_i^k)$ and the item is showed to the student (Figure 7).

According to the answer provided by the student the updating process is carried out (See section 1.3). And again the ITS-C provides a new item to the CAT that the student receives in her/his M-CAT app.

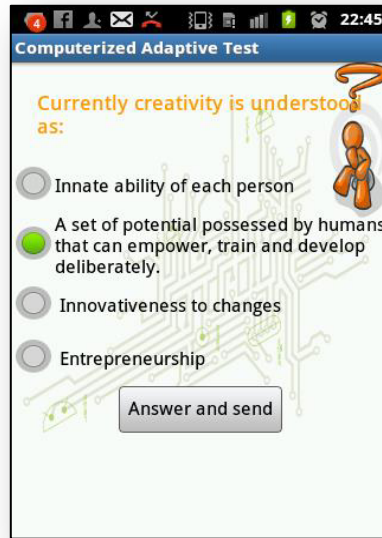


Figure 7. Question within a CAT implemented in an Android application for mobile devices.

3. Conclusions

We have presented an application developed on a mobile platform (Android) that allows deploying the architecture of an ITS's-C and implementing the diagnosis by Computerized Adaptive Tests of modular and distributed way using tablets and smartphones. The modular architecture of the domain model on an ITS-C and the diagnosis based on M-CAT has proven efficiency and lower computational requirements. Although the application does not cover all modules of the ITS-C, its current version can be a useful tool to assist with a potential ubiquitous access to educational resources and open up new possibilities for educational innovation to democratize access to learning intelligently and independently of time and space.

References

- [1] E. Wenger, "Artificial intelligence and tutoring systems," ed: Morgan Kaufmann Publishers, Inc., 1987.
- [2] J. Bourdeau and M. Grandbastien, "Modeling Tutoring Knowledge," in *Advances in Intelligent Tutoring Systems*. vol. 308, R. Nkambou, et al., Eds., ed: Springer Berlin / Heidelberg, 2010, pp. 123-143.
- [3] R. Nkambou, "Modeling the Domain: An Introduction to the Expert Module," in *Advances in Intelligent Tutoring Systems*. vol. 308, R. Nkambou, et al., Eds., ed: Springer Berlin / Heidelberg, 2010, pp. 15-32.
- [4] M. Badaracco and L. Martínez, "An Intelligent Tutoring System Architecture for Competency-Based Learning," in *Knowledge-Based and Intelligent Information and Engineering Systems*. vol. 6882, A. König, et al., Eds., ed: Springer Berlin / Heidelberg, 2011, pp. 124-133.
- [5] M. Badaracco and L. Martínez, "A fuzzy linguistic algorithm for adaptive test in Intelligent Tutoring System based on competences," *Expert Systems with Applications*, vol. 40, pp. 3073-3086, 2012.
- [6] A. Catalano, et al., Eds., *Curriculum design based on standards of occupational competence* (Concepts and methodological guidelines. Buenos Aires: CINTERFOR OIT, 2004, p.^pp. Pages.

- [7] E. a. G. Zalba, N (2006, 20/02/2008). *An approach to education based on competencies*. Available: http://www.me.gov.ar/spu/guia_tematica/CPRES/cpres-comision.html
- [8] E. S. Europe TUNING. (2000, 20/09/2009). *Una introducción a la contribución de las universidades al proceso de Bolonia*. Available: http://tuning.unideusto.org/tuningeu/images/stories/Template/General_Brochure_Spanish_version.pdf
- [9] M. Badaracco and L. Martínez, "Architecture and Diagnosis for Intelligent Tutoring System based on Competency-based Education," in *Technical Report: CIATIC-Centro de Estudios Avanzados en TIC, UJA.*, ed. Jaén, España., 2012.
- [10] E. Guzmán, et al., "Adaptive testing for hierarchical student models," *User Modeling and User-Adapted Interaction*, vol. 17, pp. 119-157, 2007.
- [11] W. J. van der Linden and R. K. Hambleton, Eds., *Handbook of modern item response theory*. New York: Springer Verlag., 1997, p.^pp. Pages.
- [12] J. O. Ramsay, "Kernel smoothing approaches to nonparametric item characteristic curve estimation," *Psychometrika*, vol. 56, pp. 611-630, 1991.
- [13] R. J. Owen, "A bayesian approach to tailored testing (Research Report No. 69)," in *Educational Testing Service*, W. J. v. d. Linden, Ed., ed. Princeton, 1969, pp. 69-92.
- [14] E. Soloway, et al., "Science in the Palms of Their Hands," *Communications of the ACM*, vol. 42, p. 21, 1999.
- [15] Y. Rogers, et al., "Ambient wood: designing new forms of digital augmentation for learning outdoors," in *Proceedings of the 2004 conference on Interaction design and children: building a community 2004*, pp. 3-10.
- [16] D. Ley, "Ubiquitous Computing," *Emerging Technologies for Learning*, vol. 2, pp. 64-79, 2007.