Personalized Construction of Self-Evaluation Tests

María T. París-Requeiro
Computer Science Department
University of A Coruña
A Coruña, Spain
mparis@udc.es

Mariano J. Cabrero-Canosa
Computer Science Department
University of A Coruña
A Coruña, Spain
mariano.cabrero@udc.es

Abstract—**The European Higher Education Area, an agreement by 29 countries to unite and harmonise qualifications and Universities' rapprochement to the real demands of the labour market, will make a significant change in the traditional model of teaching tools to carry out more personalised monitoring of the student's work, leading to the possibility of continuous evaluation. The suitable use of Information and Communication Technologies (ICT) can make a contribution to improving the quality of teaching and learning. In this context, a self-evaluation platform is developed using the technology of Intelligent Agents. This system can be adaptable as it adjusts the various self-evaluation tests to the student's level of knowledge. Each student has a profile and, depending on this, timing and interaction is set by the agents.**

Keywords-component: e-learning, user profile, self-evaluation

I. INTRODUCTION

In June 1999, the Education Ministers from 29 European countries met in the Italian city of Bologna to approve the declaration for the convergence process towards the European Higher Education Area (EHEA). 2010 was set as a final deadline to finalize this process which will allow the unification of fundamental questions related to Higher Education studies in the European Union.

The basic aims set by the EHEA are the following:

- Create a system of university qualifications which are compatible in all Europe, divided into two cycles (graduate and postgraduate).
- Use the same university credit system in all European countries, based on the student's efforts.
- Promotes tudent and lecturer mobility in Higher Education establishment.
- Design a new teaching model focused on the student who becomes the protagonist of his own learning.
- Establish a high quality evaluation system based on the student's continuous work.

To achieve these aims, one of the most significant changes in the EHEA is the new vision of the concept of learning. The traditional University system focused on teaching (by the lecturer) will become a system which is focused on the student's learning, through the use of more active teaching methodologies, more personalised monitoring of the student's work by the lecturer, and more involvement and student autonomy in the process of teaching and learning.

Another significant change of the EHEA concerns the process of evaluation. Traditional teaching methods measure the student's learning by using objective processes – both written and oral – which cannot evaluate the student's continuous effort and have no clearly formative objective. In this new educational scenario, the student's continuous evaluation and the absence of a teacher are the main axes of the formative process. The lecturer will assist and guide, designing various activities focused on acquiring the desired level of competence. One technique which has formative characteristics is a self-evaluation test. However, this type of assessment is not very useful as it cannot adapt to different student's profiles. Most software tools built to date which incorporate this type of assessment are not adapted to the student's individual characteristics nor do they allow the extraction of information on student behaviour when sitting the assessment.

II. SOFTWARE IN EDUCATION: BACKGROUND

We are familiar with software used in education which is a combination of tools used didactically to facilitate and improve the process of teaching and learning [1]. Numerous systems have been developed but the results obtained were not those hoped for. The reason is that most do not have clearly formative characteristics and allow any kind of programmed activity to be carried out by the lecturer, despite the fact that the student has not acquired the necessary knowledge to do so in optimum conditions. Moreover, they consider that one particular student's level of knowledge will be the same as the others in the group, regardless of work developed and personal circumstances.

The main aim of these applications should be enable the development of initiative and the student's autonomous learning through different tools which will allow him to check his own work, take advantage of his potential capacity for learning and let him choose the tasks to do, how to do them and the level of depth. Moreover, they should facilitate constructive learning by tutoring the student's actions providing an explanation of the mistakes committed and offering opportune help and support. Finally, they should give the learner the mechanisms to be able to plan, regulate and evaluate his learning [2].

A brief historical summary of the application of Information Technology to teaching is showed next, starting with the most basic systems up to Intelligent Tutor Systems (ITS) devised in the '80s. We will later show some proposals of the current perspective supported in instructive and

978-1-4244-6571-2/10/$26.00 ©2010 IEEE

constructive didactic approaches, noting the contribution of Intelligent Educational Systems.

The first teaching systems under the name of *Computer Aided Instruction* (CAI) appeared between the '50s and start of the '70s. They provide information to the student in the form of three categories: (a) Lineal Programs, whereby all students receive the same knowledge and in the same order. The student's particular aptitudes are not taken into account. (b) Ramified Programs, which offer all students the same knowledge, but the order depends on their answers. These do not take into account the student's aptitude and the system acts in the same way when given the same answers. (c) Adaptive Systems whereby all students receive the knowledge adapted to their needs, both the difficulty of the problem and the detail with which they must answer. The problem with these systems is that they are not valid for all subjects and only accept a single answer to a problem when in fact there may well be several.

In the '80s, these types of systems evolved towards *Intelligent Tutor Systems* (ITS), which combine techniques of *Artificial Intelligence (*AI*)*, psychological models of the student and the expert, and theories of education. An ITS is an expert system in a subject which adapts the information to the student's needs. As the student's learning process occurs step by step, continuous updating will be necessary of the information stored in the ITS.

In the '90s, the application of AI techniques to the development of these systems led to *Intelligent Educational Systems (*IES). Unlike previous models, IES do not claim to substitute a classic system of teaching and learning, but are an alternative complement to improve the quality of teaching. Depending on the student's own learning, different types of IES are distinguished: (a) Intelligent Training Systems, based on an instructional focus in which the lecturer provides continuous student feedback. Its main disadvantage is the student's passive role which can lead to the loss of motivation. (b) Adaptive Hypermedia Systems (AHS), based on a constructive focus, whereby the student chooses the route of his learning from those programmed by the lecturer. As a consequence of this amount of freedom, the student can lose direction and not achieve his aims. Despite being opposing, both options are valid and necessary and can be complementary.

Currently, given the expansion and familiarisation of the Web, under the name of e-learning systems, there are various software solutions designed mainly for supporting un-staffed teaching and learning, although they serve as a support resource for traditional teaching i.e. platforms and software systems which permit communication and interaction between lecturers and students, access and sharing of contents, materials and resources, the application of co-operative strategies of learning etc., which support (to a large extent) the student's formative process. The main inconvenience is the lack of adaptation to the type of student involved in this interaction. Any e-learning system should adapt the form, quantity and difficulty of content to the student's qualification to motivate both his progress and how he reacts when faced with obstacles. Thus one must program systems which can build a dynamic student's profile which summarises his abilities and aptitudes as regards a concrete topic.

III. CREATING A STUDENT'S PROFILE

A student's profile could be set up by uniting a piece of data which reflects the student's competencies as regards concepts, procedures and aptitudes for a subject. Such information can be obtained easily from evaluating various objective assessments, such as examinations or tests and from the lecturer's subjective evaluations such as the learner's participation in the classroom or in tutorials. This information, clearly symbolical, could be used to personalize any type of student evaluation assessment, adapting it to the level of acquired knowledge and aptitude. This applies in the same form as the design of self-evaluation assessments.

A computational model of a student's profile which is dynamically adaptable and up-to-date can be set up by evaluating various self-evaluation tests and analysing how this is confronted and how to solve the problem [3]. As a general rule, traditional teaching is divided into various sessions or seminars with a teacher, and these are accompanied by complementary activities which aim to strengthen, consolidate and amplify the fundamental concepts presented in each session. Taking this in to consideration, a student's profile would be made up of two components: (a) a particular component, which is obtained from the student's knowledge and aptitude for a concrete topic; and (b) a general component, which is the calculation of all the particular components of the student's profile.

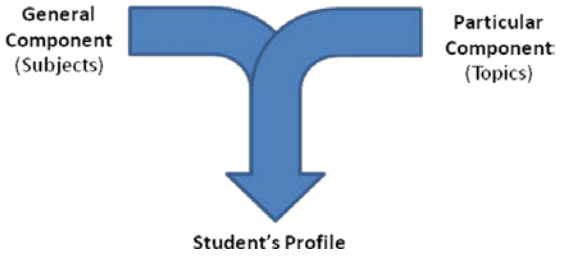

Figure 1. The components of the student's profile

The rationale behind considering this double component stems from the fact that the student may be very able in a concrete topic (as he has been successful in tests) whereas he lacks knowledge in other areas. Considering purely the general component of his profile, his knowledge would be low and consequently, further tests would not be difficult. Thus challenges would not increase and he could become demotivated. In the same way, if the successful result of a test raises the general component of his profile considerably, later tests would be more challenging even when the student has not shown a high level of competence. Thus, the general component of a student's profile measures his general competence in the subject and the particular component measures his level of knowledge and aptitude in each topic (Fig. 1). The former is updated when the student logout the system and its value is calculated as the average value of all profiles in each topic. The latter is updated after answering any test belonging to a given topic. The score of a test is a linguistic

label representing the number of correct/incorrect questions answered and the student's behaviour whilst sitting the test. Table 1 shows how the student's current profile is updated by this score.

TABLE I. UPDATING STUDENT'S CURRENT PROFILE BY A TEST SCORE

Current profile	Score of self-evaluation test				
	Very high	High	Medium	Low	Very Low
High	High	High	High	Medium	Low
Medium	High	High	Medium	Low	Low
Low	High	Medium	Low	Low	Low

To obtain an initial student's profile, one can consider the mandatory realization of a number of non adapted tests. This initial profile would be constantly modified depending on results obtained in adapted test: correct/incorrect questions, consuming time to solve it, time to answer each question. This type of test would be set up automatically by selecting questions whose level of difficulty suits the actual student's profile: depending on his particular level of knowledge and errors committed when doing previous tests on the same topic. Once the test is corrected, the system shows the corresponding feedback and updates the student's profile (Fig. 2).

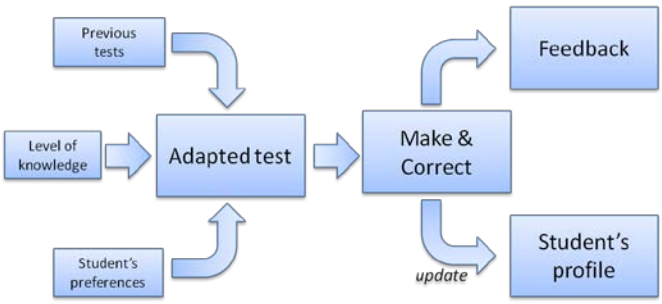

Figure 2. Three steps to personalize the construction of self-evaluation tests.

IV. COMPUTATIONAL MODEL OF THE SELF-EVALUATION PROCESS

As we have seen, in the new scenario created for the treaty of Bologna, evaluation is a process which continuously measures the student's effort. We have mentioned the use of self-evaluation assessments, adapted to the student's level qualification, as a means of evaluating acquired knowledge and help study. In this sense, we have developed a self-evaluation software tool, based on Intelligent Agents technology, which can automatically generate a test based on a personalised profile [3].

The multi-agent system developed uses a set of agents to manage the self-evaluation process, from the moment when the system is accessed, passing through the process of generating the test, to the moment when results are given. Fig. 3 shows the organisation of agents which carry out these tasks and which are described below:

- Interface Agents: interact directly with the User. These are classified as Student interface Agent and Generic Interface Agent.
- Intermediate Agents: carry out the tasks requested through the user interface. They are classified as Student Agent, Authentication Agent, Corrector Agent, Adaptor Agent, and Monitoring Agent.
- Information Agents: the Database Agent accesses the information stored in databases.

The main functionalities of the agents are detailed below:

- The *Interface Agent* allows the student's interaction with the tool. Two types can be distinguished: (a) *Generic Interface Agent,* for students who have not been authenticated, its aim is to negotiate access to the system for a User who is not authenticated, (b) Student *Interface Agent*, for authenticated users, its aim is to allow the student to do a self-evaluation test, inform him of the result, show the mistakes and give the feedback to improve his level.
- The *Student Agent* maintains the student's profile during the interaction with the system. Its aims are to inform and design the student's profile.
- The *Authentication Agent* controls a student's access to the tool and ensures he is identified until he has finished the interaction. It must check if the student is authenticated or not. When the *Authentication Agent* authorises access, a *Student Personal Agent* is created.
- The *Correction Agent* corrects self-evaluation tests. For this, it analyses and compares the information received from each of the student's answers, and the information stored in the database. It must correct and obtain the test result.
- The *Adaptor Agent* generates self-evaluation tests adapted to the student's profile. It endeavours to choose a host of questions and create the self-evaluation test.
- The *Monitoring Agent* supervises the student's activity when he does the self-evaluation test. One of its aims is to obtain the parameters of monitoring which depend on the difficulty and complexity of the topic of the test, i.e. the maximum time to do the test, the time for each question, etc. Another aim is to measure these parameters and give information on the student's behaviour whilst sitting the test.
- The *Database Agent* manages and centralises the access to information which is stored in the database. It must provide information on the User or on the test which will be created: questions available, configuration of the test and parameters to measure.

Each agent is assigned a set of sub-task and interacts with others agents to achieve his goals.

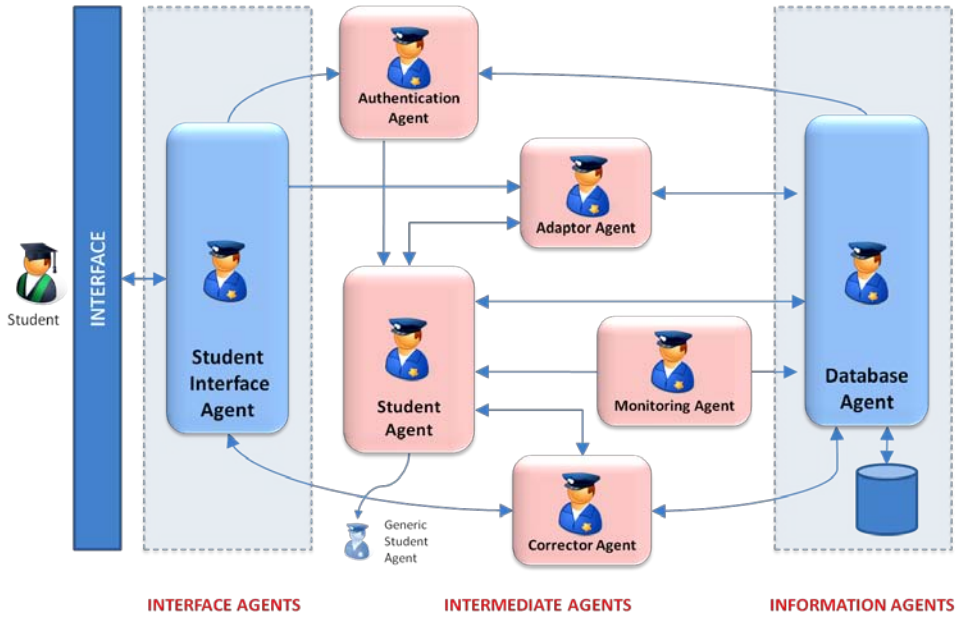

Figure 3. Organization of Multi-agent system.

V. IMPLEMENTATION OF THE SELF-EVALUATION TOOL

To implement the self-evaluation tool, a Web application has been developed whereby the agents' platform is situated. The global architecture of the system is composed of a Web client (a browser with which the student interacts), a Web server, and a database, as the multi-agent system is an extra component of this architecture as shown in Fig. 4. Through the Web interface, students interact transparently with the multi-agent system. The server collects information generated by interactions of the multi-agent system and database, from agents and from students. It processes and presents it in the form of dynamic Web pages.

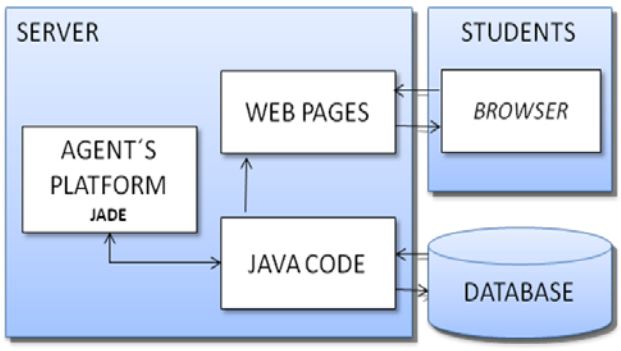

Figure 4. Global Architecture

Fig. 5 shows the interactions between agents to solve the request to obtain the result of a self-evaluation test, and how updating the particular component of student's profile:

- Ask for correction: the *Student Interface Agent* receives the request and sends it to the *Corrector Agent*.
- Ask for information on the questions: the *Corrector Agent* asks the *Database Agent* for the data necessary to correct the test, and when the test is corrected, the *Corrector Agent* sends the results to the *Database Agent* so that these are stored in the database.
- Carry out a correction: the *Corrector Agent* sends the test results to the *Student Agent*, charged with maintaining the particular component of student's profile belongs to a current topic. Also sends them to the *Interface Student Agent*, charged with showing the mistakes and giving the feedback to improve level of student.

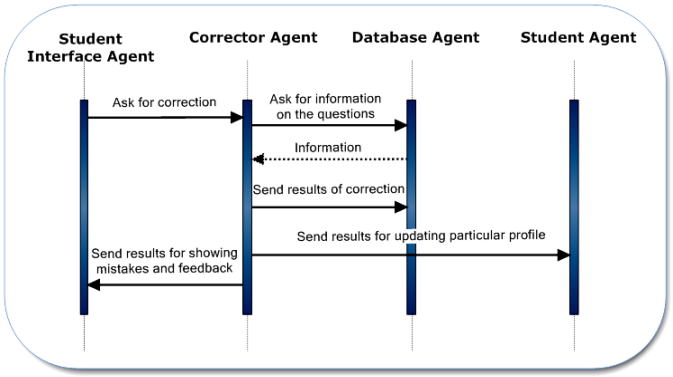

Figure 5. Interaction diagram to carry out a correction

Fig. 6 shows the interactions between agents when the student wishes to exit to the system and the general component of student's profile is updated:

- Request to exit: the *Student Interface Agent* receives the request to exit the tool and sends it to the *Student Agent*.
- Collect particular components of profile: the *Student Agent* ask the *Database Agent* for information about the particular profile in each topic.
- Update general profile: the *Student Agent* compute a new general profile from particular profiles and send it to the *Database Agent.*

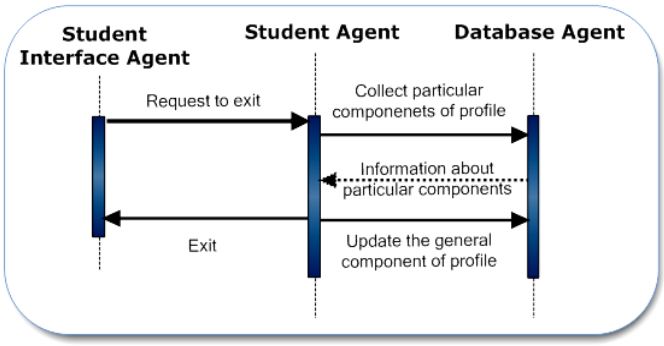

Figure 6. Interaction diagram to update the student's profile

The implementation of this architecture implies the integration of different technologies. Firstly, the multi-agent system is modelled by the IDK, tool of INGENIAS [4], an agent methodology which extends MESSAGE and establishes how a Multi-agent System has to be modelled and integrated with the "best practices" of engineering. The tool uses the Agents' platform JADE compliant with the FIPA standard.

Secondly, the Web application is developed in J2EE. For this, Apache technology known as STRUTS is used, following the Model-View-Controller pattern of architecture.

Finally, information on the students and the process of self-evaluation is stored and managed in a database implemented with MySQL.

VI. CONCLUSIONS

The EHEA proposes a teaching model which is student focused, as the evaluation will undergo a substantial change as will the student's level of knowledge, effort and continuous work. To carry out the changes considered by the EHEA, one must provide all the necessary help, not only to the students, but also to the lecturer.

Self-evaluation is a process which starts with an assessment in the form of a test and ends with information on errors committed. This type of assessment is beneficial both for the student and lecturer. For the student, a test result is an objective evaluation of the level of knowledge, understanding, mastery and progress reached in the subject, which allows him to direct his learning. In turn, the lecturer can gather significant information on the degree of satisfaction of the initially set aims, which will evidently depend on teaching strategies and resources.

To resolve some of these needs, a self-evaluation tool has been developed which allows the student to evaluate his learning process, helping him to check and consolidate his acquired knowledge and motivating him in his search for further knowledge. This tool can be adapted for each student, which satisfies a series of objectives. Firstly, it gives the student a flexible and dynamic way to evaluate his level of knowledge and know where he must improve. Secondly, it involves and motivates the student in his own process of learning. Thirdly, and lastly, it facilitates the continuous monitoring and evaluation of students by the lecturer, thus alerting him to competencies which will be more difficult to acquire.

The tool uses a Multi-agent System to build a student's profile based on the results of the self-evaluation test. Moreover, it records student interaction with the tool, generate adapted tests, and choose questions (and level of difficulty) which will be part of the test.

Therefore, by using this tool, the student will be able to control, verify and promote learning through the self-evaluation tests adapted to his profile and from the information of feedback generated by the agents once the test is corrected.

ACKNOWLEDGMENT

This work has been supported by Project 2007/000134-0 of Xunta de Galicia, Spain.

REFERENCES

[1] O. Candelario, and M. Beltrán, "Educational software: its possibilities and use in the process of teaching and learning of English", http://www.monografias.com (last access: November 2008).

[2] P. Marqués, " Design and evaluation of educational programs", http://www.xtec.es (last access: February 2009).

[3] M. París, "Plataforma de agentes para la autoevaluacón personalizada de alumnos", Master's thesis, 2007. Universidade da Coruña, Spain.

[4] J. Pavón, and J. Gómez, " INGENIAS", http://grasia.fdi.ucm.es (last access: February 2009).

[5] W. Giovanni, F. Garijo, J. Gómez, J. Pavón, P. Kearny, and P. Massonet, "The Message Methodology", in Methodologies and Software Engineering for Agent Systems, Springer, pp. 177-194, 2006.

[6] R. Pressman, "Software Engineering", MacGraw-Hill, 1982.