

AN AGENT FRAMEWORK FOR PERSONALISED STUDENT SELF-EVALUATION

María T. París and Mariano J. Cabrero
Department of Computer Science, University of A Coruña
Campus de Elviña s/n, 15071 A Coruña, Spain

Keywords: e-Learning, Multi-agent systems, User profile.

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. The lecturer will have to adapt his methods, techniques and teaching tools to carry out more personalised monitoring of the student's work, leading to the possibility of continuous evaluation. The suitable use of 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.

1 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 finalise this process. Among other things, it brings new teaching and evaluation models based on the student's continuous work. In this situation, it will be the student himself who is the protagonist of his own learning by using, at the right time and place, the contents and resources provided specifically for him by the lecturer. With this methodology, it is far easier to adapt and personalise teaching to the student's concrete needs and capacities.

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 students' 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. Thus, the lecturer must be given new software tools to allow him to evaluate the student's continuous work in a personalised way.

2 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 personalise any type of student evaluation assessment, adapting it to the level of acquired knowledge and aptitude.

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 (París, 2007). Taking this into consideration, a student's profile would be made up of two components: (1) a particular component, which is obtained from the student's knowledge and aptitude for a concrete topic; and (2) a general component, which is the calculation of all the particular 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. 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.

To obtain an initial student's profile in each area, one can consider the mandatory realisation of a number of non adapted tests. This initial profile would be constantly modified depending on results obtained in adapted tests. This type of test would be set up automatically by selecting questions whose level of difficulty suits the student's profile: depending on his particular level of knowledge and errors committed when doing previous tests on the same topic.

3 ARCHITECTURE

To facilitate the evaluation process task, we have mentioned the use of self-evaluation assessments as a means of evaluating acquired knowledge and helping study. In order to be really useful, these assessments must adapt the difficulty of the questions to the student's level of qualification. This

solution has been implemented in a self-evaluation software tool which can automatically generate a test and a personalised profile (París, 2007). Due to the complexity to handle symbolical knowledge, the possibility to break down the global task into small sub-tasks, the distributed vision of the problem solving process over Internet, and the consequent reduction of development and maintenance costs, we considered a distributed solution using agent technology.

Table 1: 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

The multi-agent system developed uses a host 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. Figure 1 show the organization of agents which carry out these tasks.

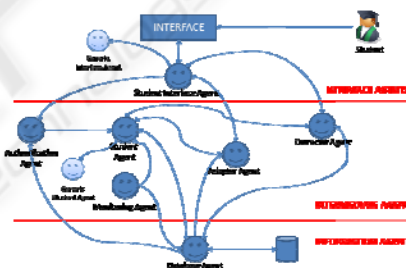


Figure 1: Organization of the Multi-agent system.

3.1 Description of Agents

The Interface Agents allow the student's interaction with the tool. Two types can be distinguished: *Generic Interface Agent*, for students who have not been authenticated, and *Student Interface Agent*, for authenticated users.

The Intermediate Agents carry out the tasks requested through the interface. They are classified as follows:

- *Student Agent*: maintains the student's profile during the interaction with the system. Its aims are to inform and design the student's profile.
- *Authentication Agent*: controls a student's access to the tool and ensures he is identified until he has finished the interaction. When the *Authentication Agent* authorises access, a *Student Agent* is created.

- *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.
- *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.
- *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 Information Agent, or *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.

3.2 Interactions between Agents

In order to satisfy the functionality of the self-evaluation tool, main interactions established between agents are defined below:

3.2.1 Asking for Access

The following interactions are established when the student wishes to access to the system:

- Request to access: the *Generic Interface Agent* receives the request to access the tool and sends it to the *Authentication Agent*.
- Check to access: the *Authentication Agent* ask the *Database Agent* for information about the user. A *Student Interface Agent* and *Student Agent* are enforced in order to interact directly with the registered user.

3.2.2 Creating Self-evaluation Test

The following interactions are established when the student wants to do a test:

- Request to create a test: The *Student Interface Agent* receives the user request and sends it to the *Adaptor Agent* and the *Monitoring Agent*.
- Obtain test characteristics: the *Adaptor Agent* interacts with the *Database Agent* and the

Student Agent to obtain the characteristics of the self-evaluation test (number of mandatory concepts, maximum time to answer each question, level of test, etc.). These characteristics depend on the chosen topic and the student's profile.

- Obtain test questions: the *Adaptor Agent* asks the *Database Agent* for the questions to create the test in line with previously obtained characteristics.

3.2.3 Correcting Self-evaluation Test

The following interactions are set up to obtain the result of a self-evaluation test:

- 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 (Figure 2): 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.

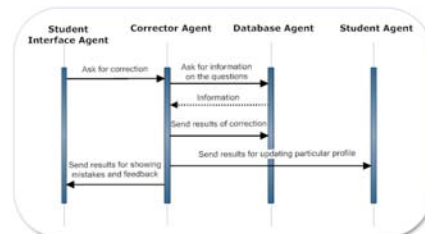


Figure 2: Interaction diagram to carry out a correction.

3.2.4 Monitoring Self-evaluation Test

This interaction is established to obtain data on how the student completes the test:

- Consultation of time taken: the *Monitoring Agent* sends the information on the way the test is done to the *Database Agent*.

3.2.5 Asking for Logout

The following interactions are established when the student wishes to exit to the system (Figure 3):

- 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 the new general profile from particular profiles and send it to the *Database Agent* (Figure 3).

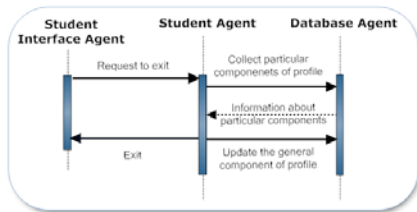


Figure 3: Interaction diagram to update student's profile.

4 IMPLEMENTATION

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 Figure 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 it and presents it in the form of dynamic Web pages.

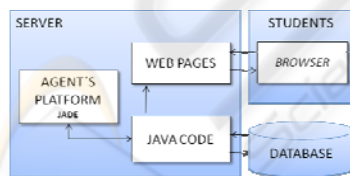


Figure 4: Global architecture.

The implementation of this architecture implies the integration of different technologies. Firstly, the multi-agent system is modelled with the IDK tool of INGENIAS (Pavón Mestras & Gómez Sanz, 2002). This tool uses the Agents' platform JADE (JADE, 2009) compliant with the FIPA standard (FIPA, 2009). Secondly, the Web application is developed in J2EE. Finally, information on the students and the process of self-evaluation is stored and managed in a database implemented with MySQL.

5 CONCLUSIONS

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.

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. The tool can be adapted for each student through the use of Intelligent Agents technology. The agents build a student's profile based on the results of the self-evaluation test. Moreover, they register student interaction with the tool, generate adapted tests, and choose questions (and level of difficulty) which will be part of the test.

By using this tool, the student will be able to control, verify and improve 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.

ACKNOWLEDGEMENTS

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

REFERENCES

- JADE: *Java Agent Development Framework*. (2009). Retrieved from <http://jade.tilab.com/>
- París, M. T. (2007). *Plataforma de agentes para la autoevaluación personalizada de alumnos*. Master's thesis. Universidade da Coruña. Spain.
- Pavón, J., & Gómez-Sanz, J. (2003). Agent oriented software engineering with ingenias. In Marik, V., Müller, J., and Pechoucek, M. (Eds.), *3rd International Central and Eastern European Conference on Multi-Agent Systems, LNAI, vol. 2691* (pp. 394-403). Springer-Verlag.
- FIPA: *The foundation for Intelligent Physical Agents*. (2009). Retrieved from <http://www.fipa.org/>