# Using a Simulated Student to Repair Difficulties in Collaborative Learning

Aurora Vizcaino and Benedict du Boulay

Grupo ORETO, Escuela Superior de Informatica, SPAIN

and

School of Cognitive and Computing Sciences, University of Sussex , UK

Email: aurora.vizcaino@uclm.es and bend@cogs.susx.ac.uk

**Abstract**

*We describe the use of a simulated student in a synchronous but distributed collaborative learning environment in the domain of programming. The role of the simulated student is to detect and repair difficulties in collaborative learning amongst the human students, for example when a human student is too passive or when the students start chatting about off-topic conversations. The simulated student intervenes by posting messages in the shared "chat" window, just like the human students and was believed to be another human student by them. The paper describes the rules by which the simulated student operates and briefly outlines an evaluation of the system with university first year programming students. The system proved to be successful both in detecting a range of difficulties and in intervening effectively.*

## 1. Introduction

Computer Supported Collaborative Learning offers students many advantages. Learners can exchange ideas and reflect upon other points of view. Adding a simulated student to the group enables all kinds of pedagogically beneficial interactions to be covertly staged within the group itself: thought provoking questions can be asked, taciturn students can be prodded to speak, bad ideas can be questioned, small slips can be caught before they have serious consequences, attention can be directed away from areas that are already mastered and towards areas where students are ripe to learn [15].

The interest in pedagogical agents began about ten years ago, when researchers began to explore new types of interactions between computers and students [8]. Agents have played different roles, such as learning companions (see, for example [1]), teachers [9], advisors [6] and students [11].

Integration-Kid [2] was the first system built as a learning companion system. A learning companion is an artificial student who interacts with the human student and learns under the guidance of the computer teacher. Thus, the learning companion performs the learning task at about the same level as the student, and both the student and the companion exchange ideas while being taught by the computer teacher [3]. EduAgents system [5] incorporates two types of teaching agent as well as companion agents. One type of teacher agent is a behavioristic one, while the other type has a more constructivist approach to teaching. The learning companion agents enrich the learning situation by taking an active part in the session.

Currently there exist different systems that use agents in collaborative environments. For example, the EPSILON (Encouraging Positive Social Interaction while Learning ON-Line) project is an initiative to develop an agent that can intelligently and adaptively provide pedagogical support to students who learn collaboratively [13]. The project repairs segments of interaction during which one team member has not shared new knowledge with the group [14]. In [7] is described a collaborative learning support system that detects an appropriate situation for a learner to join in a collaborative learning session. Constantino-González has developed COLER [4]. This is a Web-based collaborative learning environment in which students can solve data-base-modelling problems while working synchronously in small groups at a distance. Her work seeks to facilitate effective collaborative learning interactions. A coach has been implemented as a personal assistant to each client. The coaches are pedagogical agents that encourage students to discuss and participate in collaborative problem solving. The contents of this paper are organised as follows.

The next section describes the roles of a Simulated Student that have been designed in order to increase students' motivation and to improve their learning. Section three describes HabiPro a collaborative system where the Simulated was added. Section four outlines the

| Situation | Role | Strategy |
|---|---|---|
| **problem-solving** | | |
| Students do not have enough knowledge so they don't know how to work. | • The SS gives hints or explains the exercises. | Proposing clues, or solutions but always with the goal of fostering students' reflection. |
| Students always try wrong solutions. (perhaps they are trying to guess the solution) | • The SS explains why that solution cannot work.<br>• The SS tries to motivate the students (if it occurs that students are bored or tired). | To accustom the students to think about the advantages and disadvantages of a proposal. |
| Students have different points of view about the solution, and they propose different or even opposing answers. | • The SS helps the students to reflect on the different proposals.<br>• The SS encourages the student who proposes the solution to explain it. | To teach respect for different ideas and to think about their advantages or disadvantages.<br>Learning by listening and learning by teaching. |
| Students propose correct solutions. | • The SS checks that students really understand the solutions and that they did not arrive at it by chance.<br>• The SS proposes a wrong solution to create doubt. | Checking gain of knowledge |
| **off-topic conversations** | | |
| Students talk about other topics for a long time. | • The SS suggests continuing with the problems and asks questions or proposes solutions. | Drawing students' attention back towards the problems. |
| **passive students** | | |
| Student with deficient knowledge. | • The SS asks other students to explain the exercises. It can check if a gain of knowledge has arisen.<br>• The SS investigates what topics the student demonstrates more knowledge about and invites her/him to explain these topics.<br>• The SS checks if it is appropriate to lower the level of difficulty of the exercises. | Learning by teaching.<br>Learning from an explanation.<br>Adaptation of the level of difficulty of the exercises to students' knowledge level. |
| Student with adequate knowledge. | • The SS motivates and invites the passive student to intervene.<br>• The SS suggest turn taking protocols. | To motivate students to participate.<br>Reinforce self-cofindence. |
| Hyperactive student. | • The SS moderates the hyperactive participation and encourages the rest of the students to participate.<br>• The SS suggests using turn protocols. | To guarantee equitable participation. |

**Table 1. SS Interventions to help students**

.

Simulated Student's evaluation. Finally conclusions are presented

## 2. A Simulated Student which Fosters Collaboration and Learning

This section describes an implemented Simulated Student which detects and avoids situations that hamper collaboration or learning in a CSCL environment. The Simulated Student controls the students' interventions, analyses them and intervenes in order to encourage students to participate, checks students' knowledge and helps them to solve the exercises when they cannot to find the solution. The model was designed for synchronous and distributed collaboration, thus enabling students to collaborate at the same time although they are in different geographical places.

The Simulated Student in our model acts like a normal human student as much as possible, even to the point of proposing wrong solutions. So it tries to make the human students' learning process as natural as possible, allowing them to think of possible solutions and reflect on all proposals. When a negative situation takes place, the Simulated Student acts in a special role as a "responsible student" trying to prevent any decrease in collaboration or motivation. The Simulated Student model has three main components: a set of individual Student Models (SMs), the Group Model (GM) and the Simulated Student Behaviour Model (SSBM). The SSBM uses the information stored in the Group Model [10] and in the Student Models to decide when and how the Simulated Student should intervene. The model concentrates on three negative situations: unproductive problem-solving, off-topic conversations and individual passivity.

In this system extended off-topic conversations were treated as potentially a negative situation for this collaborative application. The simulated student only acted when the number of off-topic sentences exceeded a particular threshold. Short off-topic conversations may be a valuable source of cohesion within the group. Nevertheless, as Sipusic et al. [12] claim, general interaction among participants in a collaborative learning group is beneficial with the exception when the discourse is mostly off-topic and detracts from the time and effort devoted to learning. A more forgiving and rounded view of off-topic conversations is easily accommodated within the model described here, either by switching off the detection method altogether or by setting its threshold for intervention much higher.

Table 1 summarises situations that can occur in collaborative learning and the role of the Simulated Student and the pedagogic strategy used to control them. An agent playing this role and operating this strategy was added to HabiPro, a collaborative system which is described in the following section.

## 3. Habipro

HabiPro, from the Spanish "Habitos de Programación" (Programming Habits), is a collaborative, distributed, synchronous system designed to develop good programming habits in students. It provides different spaces of work (see Figure 1). One of them is an unstructured chat window (right window) that permits communication among students. The Simulated Student (Alumno3 in Figure1) also uses the chat window to communicate with the real students. The bigger window on the left displays the problems to be solved. Below this problem area, we can see the answer windows, one per student. In these windows each student writes her proposal. Having one answer window per student permits the learners to know who has proposed each solution. They can use the chat window to decide which solution they think is the correct one, and when they reach an agreement they can check whether the solution is really correct.

## 4. Evaluation

The main goal of the evaluation was to observe how the Simulated Student reacted when faced with negative situations and how the behaviour of the Simulated Student affected the other students' learning (for more details, see [16]). Forty-four students enrolled in the first course of Introduction to Programming in the first year of the Computers Science degree in Ciudad Real took part in the experiment. Students had to solve problems using HabiPro in two sessions. In the first session one group of eleven pairs of students used a version of HabiPro with the Simulated Student and the other group of eleven pairs of students used a version without the Simulated Student. In the second session the students used the version of HabiPro that they had not used in the first session. When a pair of human students used the system with the Simulated Student, they thought that they were in fact working as a group of three human students.

### 4.1 Did The Simulated Student Detect when Students Needed Help to Solve the Exercises?

One role of the Simulated Student was to help the students solve the exercises when the learners did not have enough level of knowledge or they were lost. When this happened the Simulated Student gave clues, hints or proposed solutions close to the real one. Results from the experiment showed that the Simulated Student always intervened when it was necessary and when students proposed a wrong solution the Simulated Student acted by suggesting a solution or asking a question related to the solution. The interventions of the Simulated Student helped students to solve the problem in 93.8% of the

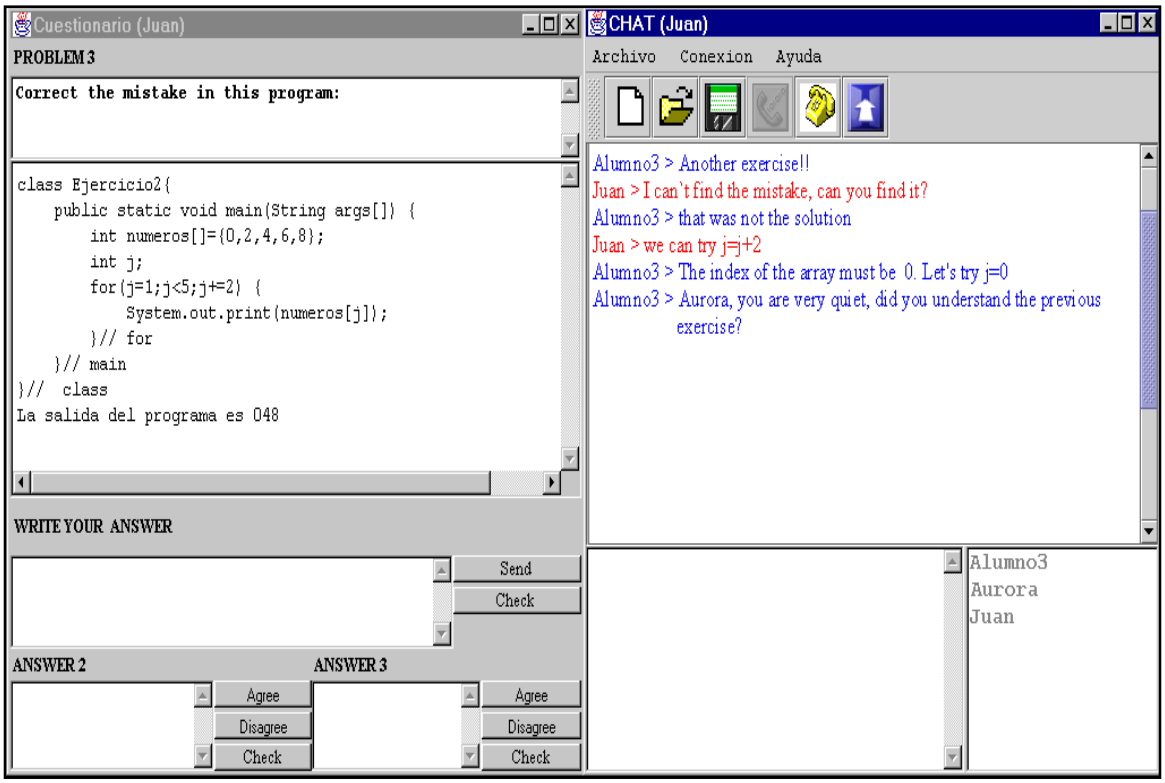


**Figure 1. Current HabiPro interface**

cases, 61 times out of 65, although students ignored the Simulated Student's advice 6.15% of the time**.**

### 4.2 Did the Simulated Student Detect Off-topic Conversations?

Students had off-topic conversations fourteen times. The Simulated Student detected twelve of these situations and headed them off in eleven cases. So its interventions was successful in 91% of the cases. A single intervention was enough to stop the off-topic conversations in each case. Analysis of the log files showed that the Simulated Student made one unnecessary intervention. We also analysed how many times students had off-topic conversations when they used the version without the Simulated Student. The logs showed that double the number of off-topic conversations occurred.

### 4.3 Did the Simulated Student Detect Passive Students?

Without the Simulated Student learners exhibited passive behaviour nine times and in all cases except one the passive student repeated his/her behaviour. When the students worked with the Simulated Student version, the agent always detected passive behaviour when it took place. On all occasions its intervention caused the passive student to take part in solving the exercises. In fact, from the logs we observed that after the Simulated Student's intervention the passive student usually proposed a solution to the problem. However, one student repeated the passive behaviour, so two interventions were necessary to encourage that student to participate. The Simulated Student acted unnecessarily once when it supposed that one student was passive but in reality the students were working as they should have been doing.

## 5 Conclusions

We have implemented and evaluated a Simulated Student that operates in a synchronous, distributed CSCL environment. It has shown itself capable of intervening effectively to deal with unproductive problem-solving, off-topic conversations and passive participants. The data obtained from the experiment indicated some situations where the Simulated Student acted unnecessary or could not amend the situation. For example The Simulated Student did not detect one off-topic conversation despite the number of sentences being higher than the threshold

because each consisted of very short interventions. The Simulated Student will be improved in order to correct the problems detected.

## References


[1] Chan, T.-W., and Baskin, A.B., (1990). Learning Companion Systems. In Intelligent Tutoring Systems: *At the Crossroads of Artificial Intelligence and Education.* Frasson, C. And Gauthier, G. (Eds.). Chapter 1.

[2] Chan, T.-W. (1991). Integration-kid: a learning companion system. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, Morgan Kaufmamm Publishers, Inc., 1094-1099.

[3] Chan, T.-W., and Chou, C.-Y.(1995). Simulating a Learning Companion in Reciprocal Tutoring Systems. In *Conference on Computer Support for Collaborative Learning* (CSCL'95). Lawrence Erlbaum Associates, Inc.

[4] Constantino-González, M. A., and Suthers, D. D., (2001). Coaching Collaboration by Comparing Solutions and Tracking Participating. Proc. of the *1st European Conference in Computer Supported Collaborative Learning EURO-CSCL*, Maastricht, the Netherlands, 22-24 March, 2001, Dillenbourg, P., Eurelings, A., Hakkarainen, K. (Eds.), pp 704-705.

[5] Hietala, P.,(1996) A prototype for a social learning system with intelligent agents. In proceedings *European Conference on Artificial Intelligence in Education.* Fundaçao Calastre Gulbenkien, Lisboa, Portugal, September. http//www.uta.fi/ph/papers/EuroAIED96/EuroAIED96.html

[6] Hietala, P., and Niemirepo, T. (1998). The competence of learning companion agents. *International Journal of Artificial Intelligence in Education*, 9, 178-192.

[7] Inaba, A., Supnithi, I., Ikeda, M., Mizoguchi, R., and Toyoda, J. (2000). How Can We Form Effective Collaborative Learning Groups? In Proceedings of *5th International Conference Intelligent Tutoring Systems 2000.* Montreal, Canada. June.

[8] Johnson, W. L., (1999) Pedagogical Agents. http://www.isi.edu/isc/carte/pedagogical_agents.html.

[9] Kasai, T. and Okamoto, T.(1998) The development of an intelligent learning environment for supporting meta-cognition enhancement, In Proceedings of *World Conference on Educational Multimedia and Hypermedia* (ED-MEDIA98), pp.673-678, 1998.

[10] Paiva, A., (1997). Learner Modelling for Collaborative Learning Environments. In du Boulay and Mizoguchi (Eds.), *Proceedings of AI-ED 97*. Kobe, Japan, IOS Press, 215-230.

[11] Ramírez Uresti, J. A. (2000) Should I Teach my Computer Peer? Some issues in teaching a learning companion. In the *Fifth International Conference on Intelligent Tutoring Systems* (ITS'2000). Montreal, Canada, pp103-112.

[12] Sipusic, M.J., Pannoni, R., Smith, R.B., Dutra, J.,Gibbons, J.F., and Sutherland, R.W. (1999). Virtual Collaborative Learning. Technical Report TR-99-72. Sun Microsystems.

[13] Soller, A, Cho, K.S. and Lesgold, A. (2000). Adaptive Support for Collaboration Learning on the Internet. Poster presented at *ITS 2000 Workshop on Adaptive and Intelligent Web-based Systems*, Montreal, Canada, pp 30-46.

[14] Soller, A., Lesgold, A. (2000). Knowledge Acquisition for Adaptive Collaborative Learning Environments. *AAI Fall Symposium: Learning How to Do Things*. Cape Cod, MA, pp 251-262.

[15] VanLehn, K., Ohlsson, S., and Nason, R. (1994). Applications of Simulated Students: An Exploration. *Journal of Artificial Intelligence in Education*, 5(2), 135-175.

[16] Vizcaíno, A. (2001) Enhancing Collaborative Learning Using a Simulated Student. PhD thesis. Universidad de Castilla-La Mancha. ISBN:84-8427-176-5. ProQuest Information and Learning.