# A Simulated Student Can Improve Collaborative Learning

**Aurora Vizcaíno,** *Grupo Alarcos. Escuela Superior de Informática, Paseo de la Universidad,4 13071 Ciudad Real (Spain)*
*aurora.vizcaino@uclm.es*

**Abstract.** This paper describes a Simulated Student architecture designed to detect and avoid three situations that decrease the benefits of learning in collaboration. These are off-topic conversations, students with passive behaviour and problems related to students' learning. In order to check the efficiency of the model in a real case a Simulated Student, which has the features described in the model, was added to a collaborative, synchronous system for learning programming. This paper describes the experiment undertaken to test the efficiency of the Simulated Student at correcting the three negative situations. The experiment showed that in the majority of the situations the Simulated Student worked correctly, thus proving that the model proposed is adequate in the avoidance of negative situations.

**Keywords.** Collaborative learning, simulated student, off-topic conversation, passive students

## INTRODUCTION

Computer Supported Collaborative Learning offers students many advantages. Learners exchange ideas and reflect upon other points of view. However, when students work in a group certain situations may occur that hamper collaboration or learning. For example, when there is a passive student in a group, the group's performance is usually lower than when all the students participate actively and the passive student possibly learns less than the rest of the group, as many researchers indicate, the information that is received but is not used during the learning process, is difficult to remember when needed (Schank and Kass, 1996). This paper describes the use of a Simulated Student as a group member to monitor students' behaviour in order to detect and avoid negative situations that decrease the benefits of the collaborative learning.

The interest in pedagogical agents began about ten years ago, when researchers began to explore new types of interactions between computers and students (Johnson, 1999). Agents have played different roles, such as learning companions (see for example Chan and Baskin 1990), teachers (Kasai and Okamoto 1998), advisors (Hietala and Niemirepo, 1998) and students (Ramírez, 2000). This paper focuses on a type of agent that simulates a student's behaviour and interacts with others in a group as if it were a real student. This type of agent will be called a "Simulated Student" in the rest of the paper.

Simulated Students have only recently become available, even though the idea is not new (Doak and Keith, 1986). Simulated Students improve traditional teaching methods where students work together in pairs or small groups. The following points indicate some of the advantages of using them in tutoring systems.

- Teachers can practice the art of tutoring by teaching a Simulated Student (VanLehn, Ohlsson and Nason, 1994).
- Students can learn in collaboration with a Simulated Student. Because the Simulated Student can be simultaneously an expert and a co-learner, it can scaffold and guide the human's learning in subtle ways (see for example, Okamoto and Kasai, 1999).
- Having a Simulated Student as part of the group enables all kinds of pedagogically beneficial interactions to be staged from 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 (VanhLenh, Ohlsson and Nason, 1994).
- A Simulated Student usually has one thing that a real student can never have: expert knowledge related to the problem to be solved. The lack of such expertise in a group composed only of human students dooms it to be less effective than one including a Simulated Student, in principle at least (Webb, 1989).

Simulated Students might reduce the incidence of negative situations which hamper the benefits of collaborative learning taking place. A negative situation occurs, for instance, when a deficit in social grounding takes place, produced because there are not enough students to play all the roles needed. A Simulated Student could solve this problem playing the roles that are not being played by anybody else, and even changing these roles depending on the situation (Singley, Fairweather and Swerling, 1999).

A Simulated Student can also monitor the group's interactions, detect miscommunications, and correct misunderstandings. Another very important advantage of using Simulated Students in a collaborative environment is that each group may have its own Simulated Student (it is very difficult to have a human teacher monitoring each group because normally schools do have not enough teachers to do this). Besides, a Simulated Student is available at any time, so students do not need to worry if the teacher is busy or unavailable.

This paper is organised as follows; the next section explains different uses of agents in intelligent tutoring systems and describes the advantages of using a Simulated Student in collaborative learning environments. It also outlines the features and limitations of several systems that use agents or Simulated Students. The following section presents the Simulated Student model that we have designed with the goal of improving the collaborative learning systems. Then, a collaborative tool where the Simulated Student was tested is described. After that, the experiment carried out to evaluate the efficiency of the model is described. Finally the results obtained from the experiment are discussed.

## AGENTS, ARTIFICIAL COMPANIONS AND SIMULATED STUDENTS

Various systems have been developed involving either an agent, an artificial companion, a virtual student or a Simulated Student. The name used is not the important thing. The most important thing is that these new partners have been introduced with the idea of helping students in the difficult task of learning.

## Improving Learning and Fostering Reflection

Integration-Kid (Chan, 1991) 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 (Chan and Chou, 1995). EduAgents system Hietala (1996) 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.

Steve is an advanced prototype designed to interact with students in networked immersive virtual environments, and has been applied to naval training tasks such as operating the engines aboard US Navy surface ships (Johnson et al., 1998). Steve is an example of an animated agent; this kind of agent is not dealt with in this paper. People Power is a system with a co-learner (computerised learner) which takes the role of a collaborator (Dillenbourg and Self, 1992). This system investigates how the co-learner and the students co-generate reflective social dialogue through argumentation or negotiation.

These examples used Simulated Students in order to help students to solve problems, or to investigate social collaboration between a computer and a human, in the case of later systems, but none of them deals with the topic of avoiding negative learning situations that may arise in collaborative environments. More recent work is researching how agents might detect and avoid undesirable student's behaviour during collaborative learning.

## Detecting and Avoiding Negative Collaborative Situations

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 (Soller, Cho and Lesgold, 2000). The project investigates the case of "a failure knowledge sharing episode". This is defined as a segment of interaction during which one team member has not shared new knowledge with the group. An episode is considered effective if one or more students learn the new knowledge and ineffective otherwise (Soller and Lesgold, 2000). Although this project endeavours in the future to deal with other situations that decrease students’ communication it is currently focused on knowledge sharing.

Another system that attempts to correct negative situations is presented by Okamoto and his colleague (Okamoto, 1996; Inaba and Okamoto, 1997). They describe a system that can diagnose and coordinate flows of discussion through the students’ conversations. The system has a chat area where students discuss different solutions to the problems. Each time a student writes a sentence s/he has to say what type of sentence it is and select the appropriate button in the intention menu (there are ten categories). We consider that this strategy may hamper or make more difficult the students' learning because they have to think about two different things: the problem to be solved and the kind of sentence that they have to select. This could distract the students' attention.

A similar application is presented by Inaba et al.(2000). She describes a collaborative learning support system that detects an appropriate situation for a learner to join in a collaborative learning session. The system has different agents to support collaborative learning dynamically. When the program detects a good moment for a learner to shift from individual learning mode to collaborative learning mode, it forms a learning group each of whose members is assigned a reasonable learning goal and a social role which are consistent with the goal for the whole group.

This system does not control if, when the member joins the collaboration session, she or he plays their role correctly. Although it may be convenient for the first student to work in collaboration, this may not be the case for the second student. The system should also check whether any negative situations take place when students are working in the group.

Constantino-González has developed COLER (Constantino-González and Suthers, 2001). This is a Web-based collaborative learning environment in which students can solve database 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. Having one coach per student implies that students' behaviour is more monitored however this may provoke students to collaborate more with their coach than with their partners, producing the opposite effect than the coach intends.

## A SIMULATED STUDENT WHICH FOSTERS COLLABORATION AND LEARNING

This section describes a model for a 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 or helps them to solve the exercises when they cannot 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 desired behaviour of the Simulated Student influences the configuration of the model; in the model outlined the Simulated Student has a similar status to the human students (see Figure 1). This is an important feature, which creates several advantages such as favouring a more comfortable environment for collaboration or encouraging the students to reflect. Students feel more at ease working with companions who have a similar level of knowledge to themselves. Dillenbourg (1999) claims that collaboration is more likely to occur between people of a similar status than between a boss and his/her employee, or between a teacher and pupils. In addition, Goodman et al., (1998) carried out an experiment that demonstrated that the interaction between students is greater than interaction with a teacher. When students receive advice from a peer, students usually reflect on it and consider the proposal but if the proposal is from a teacher, students do not generally query it.

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 model shows (see Figure 1) the Simulated Student and the human students using the same methods of communication. This makes it different from previous models where the human students perceived the agent as an animated figure which used another window to communicate with the students (Johnson et al., 1998; Goodman et al., 1998). To use the same techniques of communication for both real and Simulated Student helps students to consider the Simulated Student as a partner. In the previous cases, students could see the Simulated Student as an "assistant" that the system offers.

## Description of the Model

The Simulated Student model has three main components: the individual Student Model (SM), the Group Model (GM) and the Simulated Student Behaviour Model (SSBM). The Simulated Student model also has two complementary modules: the Information Manager and the Interface. The Interface is the means of communication between the (real and simulated) students. The Information Manager module classifies the information obtained from students and stores it in the student models and group model. The SSBM uses the information stored in the Group Model and in the Student Models to decide when and how the Simulated Student should intervene. All the three main components are described in detail in the next few paragraphs.

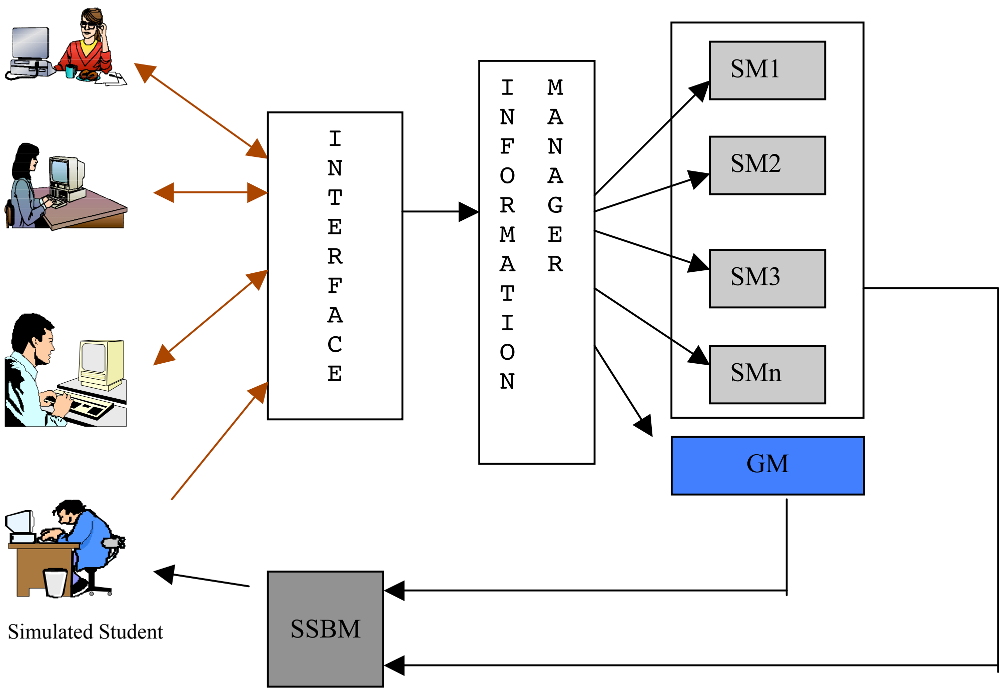


Fig. 1. Overview of the model showing flows of information.

***The Student Model***

A generic student model is formed from a set of entities where each entity expresses information about the learner. The information that a student model contains depends on the goals of the system. In our case the goals are to detect and correct situations that decrease motivation and communication in collaborative learning.

If the student model is going to work in a collaborative application further features must be considered such as individual student's goals and opinions about their partners (Paiva, 1997). The student model that we have designed contains features that enable the system to reason about the students' collaborative behaviour. These features are:

Frequency of interaction: One challenge of groupware applications is to provide a collective and equitable benefit, as Grudin (1994) claims there is often a disparity between who does the work and who gets the benefit from it. Equitable and regular participation increases the amount of information available to the group, enhancing group decision making. Improving each student's frequency of interaction increases the likelihood that all group members will learn the subject matter, and decreases the likelihood that only a few students will understand the material, leaving the others behind (Soller et al., 1998). Because of this, controlling the frequency with which a student interacts with the system and with his/her groupmates is very important. The frequency of interaction is a critical factor for detecting passive students.

Type of interaction: The students' interaction may be of different types such as talking via a chat window (this can also be of different types, e.g.: proposal, question or an explanation) or solving exercises in a shared problem-solving window. Knowing the type of interaction helps to characterise the student's role. In collaborative environments different roles arise, and these roles directly affect cognitive processes (Burton, Brna and Treasure-Jones, 1997).

Level of knowledge: The knowledge that a student has is a factor that a system should take into account since it would have to adapt its exercises, explanations and, in general, the processing of learning to the student's knowledge.

Personal beliefs: In collaborative situations the learner's beliefs are not only about the domain but also about the other learners. One student's belief about another can produce an increase or decrease in their Zone of Proximal Development (Luckin and du Boulay, 1999). If a student thinks that her partner has more knowledge about a topic than her, she expects to learn more by working with this person than studying alone. Gracile (Ayala and Yano, 1995) is a system that uses mediating agents to exchange information about the students' skills and knowledge, trying to maximise the ZPD.

Mistakes: The detection of individual mistakes is very important to determine individual misconceptions. If one student makes fewer mistakes at the end of the session than she did at the beginning, this might indicate that learning has taken place.

### *The Group Model*

The group model is defined as a way of capturing those features that identify the group as a whole (Paiva, 1997). Different opinions exist about how to model a group. Paiva (1997) claims the group is something more than the sum of its parts. For her the group model must be constructed using as a basis the actions and beliefs with which the group is in agreement.

The group model used here is based on Paiva's proposal since the amount of information stored in the entities proposed by her model is enough for the Simulated Student to be able to detect the negative situations and act efficiently. The entities are:

- Group knowledge: Beliefs that the group has. These are inferred from the group's actions (actions that are carried out because all group members were in agreement).
- Group mistakes: The mistakes diagnosed from the group's actions are group mistakes.
- Differences: The differences between the students are an important factor to consider. An example of difference is the conflict where one student supports theory P and another learner believes the theory not P. On many occasions it is convenient to use the differences between students to trigger possible discussion. This strategy is used by COLER (Constantino-Gonzalez and Suthers, 2000). A coach identifies important differences between students and encourages students to discuss these differences in ways expected to lead to learning.
- Preferences: To know what type of exercises or what kind of assistance students prefer permits that the application adapt itself to the group of users.

### *The SSBM*

The Simulated Student Behaviour Model (SSBM) is the most important component in the model. This component uses the information from the student models and the group model to decide how it must act. The architecture of the SSBM is displayed in Figure 2.

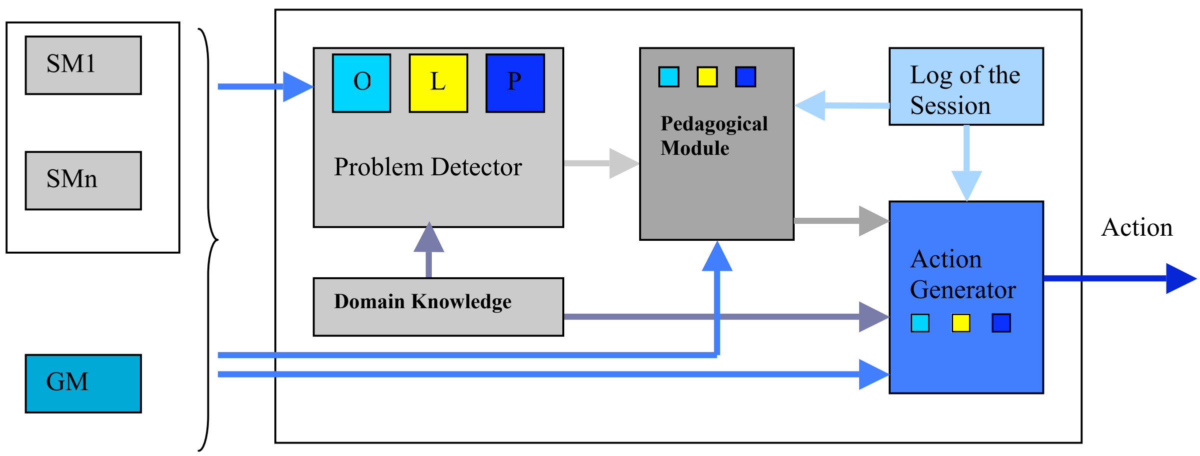


Fig. 2. SSMB architecture showing flows of information.

The SSBM is formed of five components, the Problem Detector, the Pedagogical Module, the Log of the Session, the Domain Knowledge and the Action Generator.

Domain Knowledge: As its name indicates, Domain Knowledge contains information about the subject to be learnt. This information is necessary in order to know how much knowledge students must have at each moment and also to adapt the Simulated Student's actions. In our case the Domain Knowledge was modelled representing the programming concepts in a database. Each concept is related to a number that indicates how relevant this concept is. Moreover, for each exercise there are a set of possible solutions (correct and incorrect) with a score to indicate how far the students are from the solution.

Problem Detector: Through the information received from the student models, the group model and the Knowledge Domain, the Problem Detector checks whether negative situations are taking place. This module, as Figure 2 shows, is formed of three sub-components labelled “O”, “L” and “P”. The first one is in charge of control if Off-topic conversations arise. The second component monitors the group's Learning process; this module will be explained in the following section. The last one checks each student's Participation in order to detect passive students. It is possible to modify these components individually or add new ones to avoid further different negative situations. This is an advantage that this model has, thanks to its modularity.

The Pedagogical Module: This indicates what action the Simulated Student should carry out in order to avoid the problematic situation that the Problem Detector identified. Several factors are taken into account by this module before it chooses an action. Apart from the nature of the problem detected, these factors are the individual and group features, and the Log of the Session. The Log of the Session stores all the interventions, even those of the Simulated Student. Having a record of interventions enables the system to know previous Simulated Student's answers and makes it possible not repeat answers or actions. The Log of the Session is also useful for analysing the (simulated or real) students' behaviour.

Examples of types of action that the pedagogical system can trigger are to motivate the students or to reinforce students' learning. The actions are classified into three groups, one group per negative situation; the small squares inside the pedagogical module represent this.

For each type of action that the Pedagogical Module chooses, there exists a set of possible roles that the Simulated Student can play. Depending on the student models, the group model, the Domain Knowledge, and the Log of Session, the Action Generator chooses which role the Simulated Student should play. For example, if the Problem Detector detects a passive student, the pedagogical module can advise that the student be invited to collaborate. The Action Generator decides exactly how the Simulated Student should invite the student, perhaps with a direct invitation or with a question, etc. Per each situation the Simulated Student has a set of sentences that it can use to communicate with the students. The number of sentences is rather big in order to avoid repeating the same sentence.

*The Learning Problem Detector*

The "Learning Problem Detector" (Figure 3) monitors the students' progress to decide when the Simulated Student should intervene. For instance, if the students propose a correct solution, the Simulated Student can ask about the solution in order to check whether the students really understand the solution or if they have just answered it correctly by chance. The group and individual knowledge indicate what topics the students understand individually and at a global level. Both the individual's mistakes and the group's mistakes suggest which topics the students do not comprehend. The preferences of the group are another parameter to be taken into account, since a group may always fail in the same kind of exercise because they do not know how to approach it appropriately, even though they understand the topic which the exercise is asking about.

The Learning Controller sub-module checks whether the students have problems with the topic in hand or whether the students have reached an appropriate level of knowledge (the Domain Knowledge indicates what degree of knowledge students should have at each moment). When irregularities are found in the learning process the Irregularity Selector investigates what is producing the anomaly. This information is passed to the Pedagogical Module which decides what pedagogical support the Simulated Student should offer.

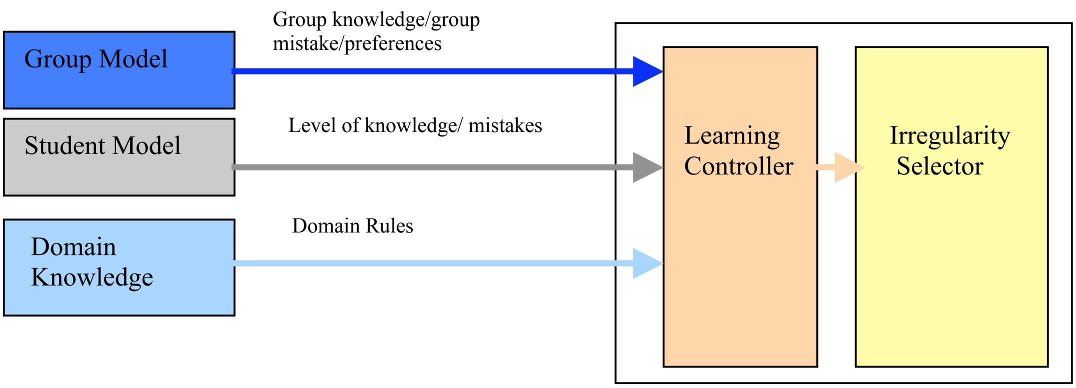


Fig. 3. Learning Problem Detector.

The model also has two more modules: the Off-topic Conversations Detector and the Passive Behaviour Detector. The first is designed to check whether students are communicating about the problem in hand or about some other topic, football, for example. The second monitors the degree of interaction of each member of the group. Further details can be found in Vizcaíno (2001b).

**Roles of the Simulated Student**

The previous section outlined the Simulated Student model and the information it uses in order to act. This section describes how a Simulated Student should behave when a problem related to the learning process arises.

Table 1
SS Interventions to help students' learning

| Situation | Role | Strategy |
|---|---|---|
| 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<br>(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. |

On some occasions groups waste a lot of time trying to solve a problem in an incorrect way. The fact that students try different ways is a good pedagogical technique because students learn from their experiences. A central part of the learning process occurs when students attempt to apply the instructional material to solve problems for themselves (Anzai and Simon, 1979; Anderson, 1983). Important learning progress may occur when students encounter obstacles, work around them, and explain to themselves what worked and what did not (Anzai and Simon, 1979; Ohlsson and Rees, 1991). However, this type of learning has potential cognitive and motivational pitfalls. Students trying to solve problems sometimes expend much time and effort pursuing blind alleys because of errors or poor strategies. Of course, in some cases students may learn something valuable while searching for a solution. In many cases, however, such episodes leave students confused and frustrated. So if a group does not obtain feedback after spending a lot of time working on a task, members may lose motivation, and even abandon the activity or begin to talk about other topics causing some group members to feel that they are wasting their time.

The Simulated Student avoids these negative effects by monitoring the students' knowledge and their learning process. When the Simulated Student detects that learners are not close to finding a solution it gives clues or explanations and even, if it is necessary, indicates the correct answer.

The effective presence of a Simulated Student in collaborative applications also avoids the Group Think Effect, which is another negative situation that arises in collaborative environments.

The Group Think Effect is produced when the group accepts an idea from one member for social reasons or because it is easier to do so. If a Simulated Student asks why they accept a proposal or proposes wrong ideas with the goal of producing doubt, the Group Think Effect should decrease. Table 1 summarises situations that can take place in a collaborative learning process. The role of the Simulated Student and the pedagogic strategy used to control the problem are also shown.

Table 2
SS Intervention to avoid off-topic conversations

| Situation | Role | Strategy |
|---|---|---|
| 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. |

As was mentioned in the previous section, the model was also designed to monitor students' communication in order to detect off-topic conversations and passive students. Table 2 shows the Simulated Student's role when it detects an off-topic conversation.

Table 3 indicates when the Simulated Student intervenes in order to encourage to a student who does not take part in solving the exercises to participate. Before intervening the Simulated Student checks why the student has passive behaviour. Perhaps the student has not enough knowledge or perhaps the opposite is taken place: the student has knowledge but s/he is a shy person. Depending on the problem, the Simulated Student uses different strategies.

Table 3
SS Interventions to avoid passive students

| Situation | Role | Strategy |
|---|---|---|
| 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 suggests turn-taking protocols. | To motivate students to participate.<br>Reinforce self-confidence. |
| Hyperactive student. | • The SS moderates the hyperactive participation and encourages the rest of the students to participate.<br>• The SS suggests using turn-taking protocols. | To guarantee equitable participation. |

**HabiPro, a System to Develop Good Programming Habits**

HabiPro, from the Spanish "Habitos de Programación" (Programming Habits), is a collaborative, distributed, synchronous system designed to develop good programming habits in students.

Two different versions of HabiPro were designed, one without a Simulated Student and the second one with a Simulated Student. To develop the second version the model outlined earlier was used. Both versions of HabiPro provide different spaces of work (see Figure 4). One of them is an unstructured chat window (right window) that permits communication among students. The Simulated Student (Alumno3 in Figure 4) also uses the chat window to communicate with the real students. The reason why the chat interaction is unstructured is to allow the students free communication and to avoid the situation of them having to think about the type of intervention that they are going to make as well as the exercises.

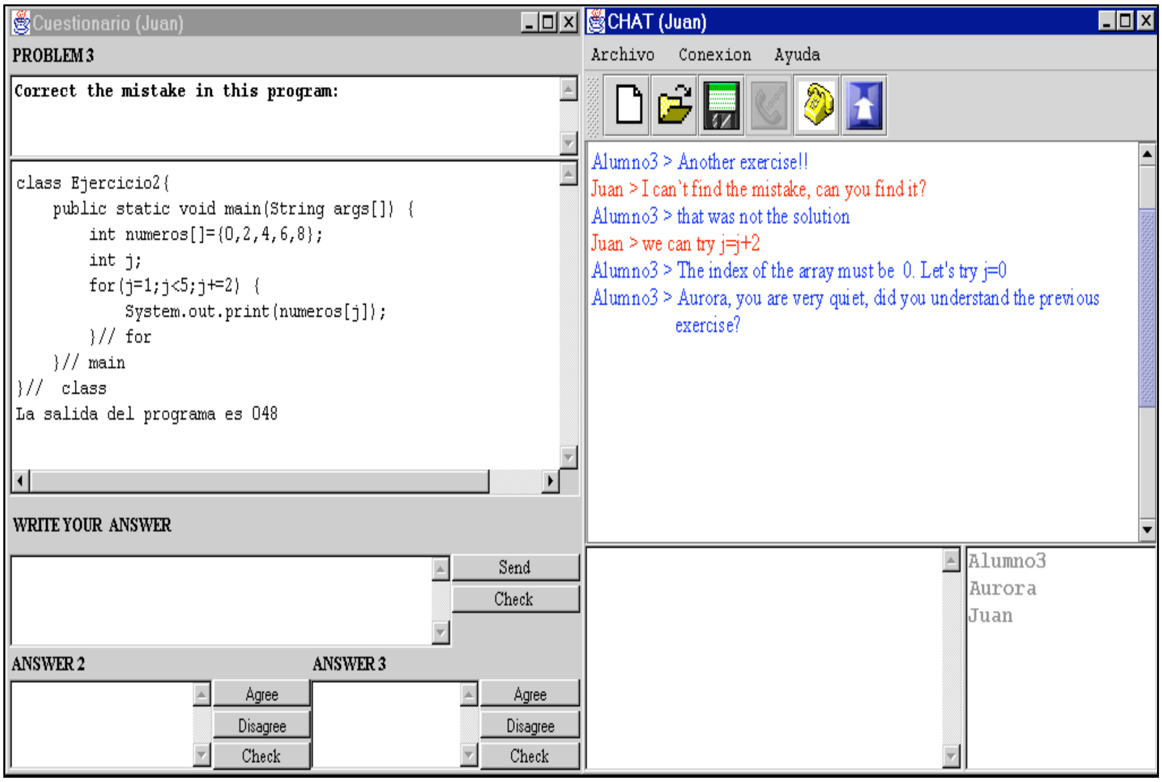


Fig. 4. HabiPro interface.

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.

Next to each answer window there are three buttons; one to check whether the proposal written in that window is correct, another to explain why the student agrees with that solution, and the last to explain why they do not agree. When students press the agree or disagree button the sentence "I agree/disagree because...." appears in the chat window and the students have to complete the sentence. The goal of these buttons was to facilitate the use of the chat window, so students would have to write less. Controlling who presses these buttons could be a method with which to monitor the types of student interactions. However, experience has shown us that students did not use these buttons very often, so it was not a useful indicator.

### *Detecting Learning Problems in HabiPro*

The Simulated Student monitors the students' learning in order to check whether learning is taking place or to detect learning problems. For example students may not understand a topic or, after several attempts to search for the solution, they are still not close to the correct one. The following paragraphs describe the behaviour of the Simulated Student to ensure correct learning.

The Simulated Student controls both the individual student's knowledge and the group's knowledge through the solutions that they write down in the answer windows. The module called the Learning Problem Detector checks how many times students try a solution and whether it is close to the correct one or not. The Simulated Student should act when this module detects that students do not have enough knowledge or they do not understand the problem since they cannot find the solution after an adequate number of attempts. Normally the interventions of the Simulated Student consist of giving a clue that helps the student to delimit the question or to reflect on the problem.

The number of trials that the Learning Controller considers adequate per exercise depends on the difficulty of the exercise and on the students' level of knowledge.

When a problem is detected the Simulated Student gives hints as a human student might do, using words that students would use and sometimes does not show much confidence in its proposal. So, students reflect upon these ideas and reject them if they disagree. If the clue was given in the manner of a teacher using technical language and with more authority, perhaps the advantage of working with a Simulated Student would disappear.

In the case of students not understanding the clue proposed by the Simulated Student and continuing to try incorrect answers, the Simulated Student generally suggests the solution and an explanation of it, with the goal of students understanding the solution.

Besides helping students to find the solution to the problem, the Simulated Student also makes other types of interventions related to motivating students' learning or checking that learning has taken place. For instance, when students find the solution at their first attempt, the Simulated Student congratulates them with the aim of maintaining their motivation. This is positive reinforcement.

One disadvantage of working collaboratively in a group might be that a student who knows the solution writes it in the answer window without explaining to her groupmates why the solution works. In this way perhaps the groupmates do not learn why the solution is correct. The Simulated Student tries to avoid this situation by checking whether students understand their solutions. For instance, the Simulated Student asks why a solution was proposed and why it is better than the others. In this way it can also verify whether students found the solution by chance.

### *Detecting off-topic Conversations in HabiPro*

In order to detect whether off-topic conversations are taking place, the Simulated Student uses the Conversation Processor which contains different databases. One is a general database, which contains words related to problem solving in programming. Another is a specific database that possesses words related to the specific exercises, one database per exercise. The final database, called the playful database, contains words related to conversations that young students might have in their free time (football, women, men..). Of course, this database can be modified depending on the environment where HabiPro is used.

When students write a sentence in the chat window, the Conversation Processor checks whether the conversation contains keywords stored in the specific database. If no keyword is found, the second step is to check if the student's conversation contains words belonging to the general database. In the case of none of the words in the conversation matching one or more words of the specific and general database, the Simulated Student supposes that students are talking about other topics. If this situation occurs once or twice the Simulated Student does not intervene because a short off-topic conversation usually helps students feel more comfortable in the group or to relax briefly. On the other hand, if the conversation lasts a while it might be negative, so this is when the Simulated Student acts. The Situation Controller module is in charge of deciding whether the Simulated Student does or does not have to intervene. The Situation Controller tests the duration of the off-topic conversation, the group's knowledge and the group's preferences in order to determine whether an off-topic conversation is taking place and whether the Simulated Student should act.

The goal of having a playful database is that the Simulated Student can know what students are talking about. It can then finish that conversation with a sentence related to the off-topic conversation, with the objective of appearing to understand said conversation. An example of intervention is "I don't like football. Let's finish this exercise". In the case of it not being possible to know what students are talking about because none of the words in the conversation match the words in the playful database, the Simulated Student tries to draw the students' attention back towards the exercise by suggesting a possible solution. We performed an experiment with teachers imitating the role of a Simulated Student. The experiment showed that the most efficient interventions to close off-topic conversations were those that did not give an option to continue the dialogue, but only proposed a possible solution to the problem and ignored the students' comments about other topics (Vizcaíno and Prieto, 2000). One example of the Simulated Student's interventions could be: "I think the solution is 13, don't you?".

### *Detecting Passive Students in HabiPro*

In order to detect passive behaviour the Simulated Student uses a module called Detector of Passivity to check the frequency with which each student intervenes by proposing solutions in the answer window. If this parameter is inferior to a established threshold, (the thresholds may be the average of interventions or may be indicated by a teacher), the Simulated Student suspects that a student might be passive.

However, a low frequency of intervention in the answer window is not a clear indicator of passivity. For this reason, the Simulated Student also checks the frequency and density of participation in the chat window and the students' level of knowledge. These indicators help to detect:

- Whether the student's participation is also poor in the chat window. In this case, if student's participation density is also low, the student is considered passive.
- Whether, although the learner participates little in writing in the answer window, when s/he does participate in proposing ideas, his/her density of interaction must be equal to or higher than the average of the group density. In this case, the student would not be considered passive, perhaps the student needs more time to reflect than the rest.
- Whether the student participates in the chat window with only sentences such as: yes, no, maybe (low density of interaction). In this case the student would be considered passive.

When the Detector of Passivity, after studying all the indicators, considers that the learner is passive, a second module called Selector of Passivity has to investigate what type of passivity is taking place. This sub-module checks the number of solutions that have been proposed by the student and the index of mistakes and successes obtained. The study of these parameters in conjunction with the comparison of the specific (features of the student) and global (features of the group) parameters such as density, frequency of interventions or knowledge helps the Simulated Student to conclude why the passivity might have occurred.

The Simulated Student has different types of intervention, depending on the kind of passivity detected. The different roles were summarised in Table 3.

## EVALUATION

HabiPro allowed us to design different experiments in order to test the model's efficiency. This section describes a recent experiment and presents the results obtained. For more information about HabiPro see Vizcaíno et al. (2000) and Vizcaíno (2001a,b).

The main goal of the experiment was to observe how the Simulated Student reacted when faced with certain negative situations and how the behaviour of the Simulated Student affected the other students' learning. Another objective was to evaluate the students' assessment of the Simulated Student's interventions. Given these needs, the aims of the experiment described here were to explore:

1. The efficiency of the Simulated Student in detecting problems in the learning process and its efficacy in solving the problem.
2. The efficiency of the Simulated Student in detecting and avoiding off-topic conversations.
3. The efficiency of the Simulated Student in detecting and assisting passive students.
4. The effect of the Simulated Student on other students' learning.
5. Students' assessment of the Simulated Student.
6. Students' opinion about their learning when using HabiPro.

**Hypotheses**

- H0: (Null hypothesis): Adding a Simulated Student to HabiPro does not affect the students' behaviour and learning.
- H1: By adding a Simulated Student to HabiPro students could solve more exercises than using the version without the Simulated Student.
- H2: By adding a Simulated Student to HabiPro the duration of off-topic conversations could be decreased.
- H3: By adding a Simulated Student to HabiPro passive students could be helped to became active.

**Design of the Experiment**

To test the hypotheses we designed an experiment in which students had to solve problems using HabiPro in two sessions. In the first session one group of students used a version of HabiPro with the Simulated Student and the other group 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. The experiment is a within-subjects design.

**Subjects**

Forty-four students enrolled in the first course of Introduction to Programming in the first year of the Computer Science degree in Ciudad Real (Spain) took part in the experiment. The students chosen had to have one factor in common: the same teachers in their theory and practice classes to avoid the possibility of some students knowing more because their teacher had explained better or faster, and thus a similar level of knowledge was expected. Those students who were repeating the course were not permitted to take part in the experiment because they might have had more knowledge than the students who had just started to learn programming. The experiment was run as part of their normal laboratory activities. It was carried out in the same laboratory in which they attended their normal practical classes and at the same time that they were used to attending them.

Students were randomly divided into the two sub-groups, Team A and Team B each of 22 students. One subgroup started the experiment working with the version of HabiPro containing the Simulated Student and the other subgroup with the version without Simulated Student. The sub-groups were also randomly divided into couples. So we had two subgroups each of eleven pairs.

**Procedure**

Each couple taking part in the experiment attended two sessions about one week apart. The sessions each lasted approximately one hour. Each pair had to solve programming problems using a different version of HabiPro in each session. So, the eleven couples that used the version without the Simulated Student in the first session used the version with the Simulated Student in the second session, and vice versa. Each student worked from a separate computer and they communicated with each other using the chat window: that means that members of a couple could be in different geographical locations.

All the students took a pre-test one week before the first session. The test contained programming exercises that they had to solve. The goal of the pre-test was to indicate the number of correct exercises that students could solve alone and without HabiPro.

When students used the version of HabiPro containing the Simulated Student they did not know that the third member of their group was the Simulated Student. In this way we tried to avoid the situation of students considering the simulated student's proposal as being more appropriate or, on the other hand, ignoring it because it came from a Simulated Student.

The two versions of HabiPro, with and without the Simulated Student, contained nineteen exercises. Both versions recorded the students' actions and their participation in the chat window.

The importance of maintaining anonymity was explained to the student. Students connected to HabiPro with a false name with the goal that their personal beliefs did not influence the collaborative process.

When the first session finished students were individually invited to fill in a questionnaire where they gave their opinion about the participation and collaboration of their partners, their assessment of HabiPro and recommendations or suggestions for improvement.

The second session was carried out the next time that the students had lab practice, in other words, one week later. Due to the fact that the students had to use the opposite version of the one that they used the first time, a small introduction to the new version was needed. In this case, Team A was introduced to HabiPro without the Simulated Student and vice versa.

When the second session finished, students individually filled in a questionnaire which was similar to the one in the first session, the only difference being that Team A had to give their opinion about two partners (the human student and the virtual one) in the first session, and in the second session only about one partner, the human student. The opposite applied to Team B.

**Materials**

Apart from HabiPro, students were presented with a pre-test, and with a questionnaire. A description of them follows.

*Tests*

The pre-test contained 10 exercises similar to the exercises provided in HabiPro.

*Questionnaire*

There were two types of questionnaire depending on the version of HabiPro used. The questionnaires were designed to measure the subjects' perceptions of their partners and of HabiPro as a learning tool.

**Results**

This section presents the results obtained from the experiment. The first sub-section shows how the Simulated Student influenced the other students' learning. The second sub-section is centred on the Simulated Student's interventions to stop off-topic conversations and the last one focuses on checking the Simulated Student's efficiency at encouraging passive students to participate.

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

One role of the Simulated Student is to help the students to solve the exercises when the learners do not have enough level of knowledge or they are lost. When this happens the Simulated Student gives clues, hints or proposes solutions close to the real one. In this section the degree of success of the Simulated Student in playing the role of adviser is analysed. Before analysing the results obtained when students used the version with the Simulated Student we are going to see with what frequency students needed help to solve the problems when they worked without the Simulated Student. The first row of Table 4 indicates each couple's identification number (C). The second row indicates the number of times that each couple needed three or more attempts to solve the problem (Nu), this means that the couple proposed two wrong solutions. The third row shows the number of exercises that students solved (Exe).

Table 4
Number of times that students had problems in solving the exercises

| C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Nu | 3 | 4 | 2 | 2 | 2 | 3 | 4 | 2 | 1 | 2 | 3 |
| Exe | 7 | 14 | 7 | 14 | 9 | 9 | 11 | 8 | 9 | 8 | 9 |

| C | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | Tot |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nu | 1 | 3 | 2 | 3 | 5 | 4 | 3 | 2 | 3 | 2 | 3 | 59 |
| Exe | 13 | 12 | 10 | 10 | 19 | 11 | 11 | 10 | 9 | 11 | 9 | 230 |

Table 4 shows that in 59 situations students did not solve the exercises in the first two trials. These results will be commented on later. Table 5 shows the results obtained when students used the version with the Simulated Student. The logs stored when students worked with this version were analysed in order to answer the following questions:

- How many times did the Simulated Student detect that students needed assistance to solve the exercises?
- Did the Simulated Student's intervention help students to solve the exercises?
- Did students always consider the Simulated Student's advice?
- How many interventions by the Simulated Student were necessary to solve the problem?
- Did the Simulated Student act when it was inappropriate to do so?

The first column in Table 5 indicates each pair's number. The second column, called "number of times that students needed help", indicates how many times a pair had "problems" in solving the exercise. By having problems, we mean that the couple proposed two wrong solutions. This information was obtained from the stored logs. These contained all the answers written in the answer window (even the incorrect ones), all the conversation in the chat window and all the Simulated Student's interventions. The third column indicates how many times the Simulated Student detected the situation. The fourth column called "students solved the problem" shows how many times the Simulated Student's intervention seemed to help the students to solve the exercise. The fifth column indicates how many times the students ignored the Simulated Student's proposal. The sixth column indicates how many times the Simulated Student intervened in order to help students. The seventh column indicates how many times the Simulated Student acted unnecessarily, in other words, when the Simulated Student considered that its help was necessary although it was not. The last column shows the number of exercises that students solved.

By comparing Table 4 and 5 we can see that students had more problems in solving the exercises when they used the version with the Simulated Student than in the other case. This fact might be because the exercises were more difficult and longer in the version with the Simulated Students.

The logs of the version without the Simulated Student showed that in 68% of the cases students found the solution at the third or fourth attempt, and most times they needed to consult the help offered by the system. However, in the rest of the cases the students, instead of reflecting upon the problem, started to talk about other topics. The following conversation is a typical example of a conversation that took place when students did not find the solution after several attempts.

Table 5
Number of Times that Students Needed Help and SS's Interventions

| **Couple** | **Number of times that students needed help** | **Detected** | **Students solved the problem** | **Students ignored the help offered** | **SS. interventions** | **SS intervened unnecessarily** | **Exercises solved** |
|---|---|---|---|---|---|---|---|
| 1 | 4 | 4 | 4 | 0 | 4 | 0 | 10 |
| 2 | 3 | 3 | 2 | 1 | 3 | 0 | 16 |
| 3 | 1 | 1 | 1 | 0 | 1 | 0 | 8 |
| 4 | 1 | 1 | 1 | 0 | 1 | 0 | 15 |
| 5 | 2 | 2 | 1 | 1 | 2 | 0 | 14 |
| 6 | 2 | 2 | 2 | 0 | 2 | 0 | 13 |
| 7 | 4 | 4 | 3 | 1 | 4 | 0 | 13 |
| 8 | 3 | 3 | 3 | 0 | 4 | 1 | 11 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| 10 | 2 | 2 | 2 | 0 | 3 | 1 | 12 |
| 11 | 5 | 5 | 4 | 1 | 5 | 0 | 11 |
| 12 | 2 | 2 | 2 | 0 | 2 | 0 | 13 |
| 13 | 4 | 4 | 4 | 0 | 4 | 0 | 8 |
| 14 | 2 | 2 | 2 | 0 | 2 | 0 | 11 |
| 15 | 2 | 2 | 2 | 0 | 0 | 0 | 10 |
| 16 | 5 | 5 | 5 | 0 | 6 | 1 | 12 |
| 17 | 5 | 5 | 5 | 0 | 5 | 0 | 11 |
| 18 | 4 | 4 | 4 | 0 | 4 | 0 | 12 |
| 19 | 2 | 2 | 2 | 0 | 2 | 0 | 10 |
| 20 | 3 | 3 | 3 | 0 | 4 | 1 | 9 |
| 21 | 4 | 4 | 4 | 0 | 4 | 0 | 10 |
| 22 | 5 | 5 | 5 | 0 | 5 | 0 | 10 |
| **Sum** | **65** | **65** | **61** | **4** | **69** | **4** | **251** |

Student1: The clues indicate that our solution is wrong.
Student2: I have no idea what the answer is.

Student1: Are you from Ciudad Real?
Student2: No, I am from Puertollano and what about you?

Students wasted a lot of time talking about other topics. This might be the reason why the students solved less exercises even though the problems were easier than in the version with the Simulated Student.

Now the data obtained in the case that the students used the Simulated Student version is analysed. The results show that the Simulated Student always intervened when it was necessary (100% successful), the logs indicated that when students proposed a wrong solution the Simulated Student acted by suggesting a solution or asking a question related to the solution. The intervention of the Simulated Student helped students to solve the problem in 93.8% of the cases, 61 times out of 65. However students ignored the Simulated Student's advice 6.15% of the times, hence in these cases the Simulated Student intervention's was not efficient.

From these results it is possible to deduce that one intervention from the Simulated Student was enough to help the students to solve the problem. Table 5 shows more interventions (69) because of the 4 times that the Simulated Student intervened unnecessarily. In the discussion section the possible reasons why the Simulated Student acted when it was not necessary will be analysed.

The data obtained from the experiment supports the hypothesis that the Simulated Student helps students to solve problems, because although students did not know how to attack the problem in many situations, the Simulated Student's interventions helped students to find the solution. However, in 63.63% of the cases the students solved more exercises correctly than when they used the version without the Simulated Student. So, in an indirect way, the hypothesis H1 is supported. However, more statistical work was also performed to see if the difference in the number of exercises that students solved with and without the Simulated Student was significant. Therefore, Table 6 shows the number of exercises that each pair of students solved correctly. WSS stands for "Without Simulated Student", SS stands for "Simulated Student".

In order to know whether there are differences between the number of exercises solved in the first session and the number of exercises solved in the second session a test of normality was carried out. This test is useful to check whether both variables have or do not have a normal distribution. Depending on their distribution parametric or nonparametric tests should be used.

Due to the sample being less than 50 the data of the Shapiro-Wilk test should be considered. This test indicates, with $p=0.010$, that none of the variables has a normal distribution. Therefore a nonparametric test should be applied in order to see whether significant differences exist between the results of the first and second sessions. Two nonparametric tests are used: the sign test and the Wilcoxon signed-rank test. The McNemar test cannot be used because the variables are not dichotomous.

For each pair of observations, the sign test only uses the direction of the differences (positive or negative), while the Wilcoxon signed-rank test begins by ranking the differences without considering the signs, restoring the sign to each rank, and finally summing the ranks separately for the positive and negative differences. This test offers more information about the data than the sign test. The following tables show the results obtained after applying both test to number of exercises of Session 1, and number of exercises Session 2.

Equivalent results are obtained with both statistics. However due to the fact that the size of the sample is smaller than 50, the Wilcoxon test is taken more into account. The Wilcoxon test indicates that the Asymptotic Significance is 0.787. This means that the probability of making an error if the null hypothesis was rejected is very high, for this reason the null hypothesis should be accepted. The null hypothesis is that the distributions are equal.

Table 6
Number of exercises solved in each session

| Pair | Version used in the first session | Number of Exercises solved in 1st Session | Number of Exercises solved in 2nd Session |
|---|---|---|---|
| 1 | WSS | 7 | 13 |
| 2 | WSS | 14 | 12 |
| 3 | WSS | 7 | 10 |
| 4 | WSS | 14 | 19 |
| 5 | WSS | 9 | 10 |
| 6 | WSS | 9 | 10 |
| 7 | WSS | 7 | 11 |
| 8 | WSS | 8 | 11 |
| 9 | WSS | 9 | 10 |
| 10 | WSS | 8 | 9 |
| 11 | WSS | 9 | 11 |
| 12 | SS | 9 | 13 |
| 13 | SS | 8 | 8 |
| 14 | SS | 16 | 11 |
| 15 | SS | 15 | 10 |
| 16 | SS | 14 | 12 |
| 17 | SS | 13 | 11 |
| 18 | SS | 13 | 12 |
| 19 | SS | 11 | 10 |
| 20 | SS | 12 | 9 |
| 21 | SS | 12 | 10 |
| 22 | SS | 11 | 7 |

Table 7
Test of Normality

| | Kolmogorov-Smirnov | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|---|
| | Statistic | df | Sig. | Statistic | df | Sig. |
| **Exercises Session 1** | .225 | 44 | .000 | .898 | 44 | .010 |
| **Exercises Session 2** | .204 | 44 | .000 | .815 | 44 | .010 |

Table 8
Descriptive Statistics

| | N | Mean | Std. Deviation | Min. | Max. |
|---|---|---|---|---|---|
| **Exercises Session 1** | 44 | 10.68 | 2.82 | 7 | 16 |
| **Exercises Session 2** | 44 | 10.86 | 2.31 | 7 | 19 |

Table 9
Ranks
Wilcoxon Signed-Rank Test

| | | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| **Exercises Session 2-Exercises Session 1** | **Negative Rank** | 20(a) | 21.50 | 430.00 |
| | **Positive Rank** | 22(b) | 21.50 | 473.00 |
| | **Equals** | 2(c) | | |
| | **Total** | 44 | | |
| a Exercises Session 2 < Exercises Session 1 | | | | |
| b Exercises Session 2 > Exercises Session 1 | | | | |
| c Exercises Session 1 = Exercises Session 2 | | | | |

Table 10
Contrast Statistic (b)

| | Exercises Session 2 - Exercises Session 1 |
|---|---|
| **Z** | -.270(a) |
| **Asymptotic Significance (2-tailed)** | .787 |
| a. Based on negative ranks. | |
| b Wilcoxon signed-rank test | |

Table 11
Frequencies
Sign Test

| | | N |
|---|---|---|
| **Exercises Session 2 - Exercises Session 1** | **Negative Differences (a)** | 20 |
| | **Positive Differences (b)** | 22 |
| | **Equals(c)** | 2 |
| | **Total** | 44 |
| a Exercises Session 2 < Exercises Session 1 | | |
| b Exercises Session 2 > Exercises Session 1 | | |
| c Exercises Session 1 = Exercises Session 2 | | |

Table 12
Contrast Statistic (a)

| | Exercises Session 2 – Exercises Session 1 |
|---|---|
| **Z** | -.154 |
| **Asymptotic Significance. (2-tailed)** | .877 |
| a Sign Test | |

The next step is to check whether students solved more exercises with a particular version of HabiPro. In order to study this fact, the variable "version" is considered as a factor, its value indicates whether the students started to work with or without the Simulated Student. Thus, it is necessary to divide the initial sample into two sub-samples: students who used the version with the Simulated Student in the first session, students who used the version without the Simulated Student in the first session and the same division for the second session. The conditions of normality and variance are studied in both sub-samples. Table 13 shows the results.

The results of the Shapiro-Wilk test are different for each value of the factor "Version". The test indicates that the null hypothesis should be rejected (with a p value of 0.000) for the value "Without the Simulated Student", the null hypothesis is that the variable has a normal distribution. However, the test indicates a normal distribution in the case of the version having a value of "With Simulated Student". So independent of the session, the distribution is normal when the version with the Simulated Student is used and the distribution cannot be considered normal when students used the version without the Simulated Student.

Table 13
Test of Normality

| | | Kolmogorov-Smirnov | | | Shapiro-Wilk | | |
|---|---|---|---|---|---|---|---|
| | ***Version*** | **Statistic** | **df** | **Sig.** | **Statistic** | **df** | **Sig.** |
| **Exercises Session 1** | **With Simulated Student** | .125 | 22 | .200 | .953 | 22 | .414 |
| | **Without Simulated Student** | .348 | 22 | .000 | .733 | 22 | .010 |
| **Exercises Session 2** | **With Simulated Student** | .165 | 22 | .121 | .943 | 22 | .296 |
| | **Without Simulated Student** | .295 | 22 | .000 | .682 | 22 | .010 |

The Levene Statistic shows that the level of significance is too big to refuse the null hypothesis (H0: there is homogeneity of variance), hence the samples have a homogeneity of variance in both sessions.

The results obtained imply that a parametric study should be performed when the Simulated Student is used. In contrast, if the version of HabiPro without the Simulated Student is used a nonparametric study might be made.

The next study starts analysing the first sub-sample where the value of "version" is "with Simulated Student" in the first session and, of course, "without Simulated Student" in the second session. The following Table 15 shows the average number of exercises solved in each session.

Table 14
Test of Homogeneity of Variance

| | | Levene Statistic | df1 | df2 | Sig. |
|---|---|---|---|---|---|
| **Exercises Session 1** | **Based on Mean** | .031 | 1 | 42 | .861 |
| | **Based on Median** | .136 | 1 | 42 | .714 |
| | **Based on Median and with Adjusted df** | .136 | 1 | 39.894 | .714 |
| | **Based on Trimmed Mean** | .135 | 1 | 42 | .715 |
| **Exercises Session 2** | **Based on Mean** | .588 | 1 | 42 | .447 |
| | **Based on Median** | .121 | 1 | 42 | .730 |
| | **Based on Median and with Adjusted df** | .121 | 1 | 30.788 | .731 |
| | **Based on Trimmed Mean** | .217 | 1 | 42 | .644 |

Table 15
Statistics to measure the relationship among variables

| | | Mean | N | Std. Deviation. | Std. Error Mean |
|---|---|---|---|---|---|
| **Sub-Sample 1** | **Exercises Session 1** | 12.18 | 22 | 2.34 | .50 |
| | **Exercises Session 2** | 10.27 | 22 | 1.75 | .37 |

Table 16
Paired Samples Test

| | | Paired Differences | | | | | t | df | Sig. (2-tailed) |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean | Std. Deviation | Std. Error Difference | 95% Confidence Interval of the difference | | | | |
| | | | | | Lower | Upper | | | |
| **Sub-sample 1** | **Exercises Session 1 - Exercises Session 2** | 1.91 | 2.49 | .53 | .81 | 3.01 | 3.601 | 21 | .002 |

We can see that there is a difference between means: students solved more exercises in the first session with the Simulated Student than in the second session without it. The next step is to test whether the difference is significant. The statistic used to compare means is the T-Student test which indicates that there is a significant difference between means. This argues that students do tend to solve more exercises using the version of HabiPro with the Simulated Student.

Let's see what happens with the second sub-sample, when students did not use the Simulated Student version in the first session but they did use it in the second session. The following table indicates that there is difference in the means.

Table 17
Descriptive Statistics

| | N | Mean | Std. Desviation | Min. | Max. |
|---|---|---|---|---|---|
| **Exercises Session 1** | 22 | 9.18 | 2.46 | 7 | 14 |
| **Exercises Session 2** | 22 | 11.45 | 2.67 | 9 | 19 |

Since the sample has a difference of means, the next phase will be to check whether the difference is significant. Because the sample, in this case, does not have a normal distribution, the nonparametric tests, the sign test and the Wilcoxon signed-rank tests, are again used.

Table 18
Ranks
Wilcoxon signed-rank test

| | | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| **Exercises Session 2 - Exercises Session 1** | **Negative Rank** | 2(a) | 10.50 | 21.00 |
| | **Positive Rank** | 20(b) | 11.60 | 232.00 |
| | **Equals** | 0(c) | | |
| | **Total** | 22 | | |
| a Exercises Session 2 < Exercises Session 1 | | | | |
| b Exercises Session 2 > Exercises Session 1 | | | | |
| c Exercises Session 1 = Exercises Session 2 | | | | |

Table 19
Contrast Statistic (b)

| | Exercises Session 2 - Exercises Session 1 |
|---|---|
| **Z** | -3.450(a) |
| **Asymptotic Significance (2-tailed)** | .001 |
| a Based on negative ranks. | |
| b Wilcoxon signed-rank test | |

Table 20
Frequencies
Sign Test

| | | N |
|---|---|---|
| **Exercises Session 2 - Exercises Session 1** | **Negative Differences (a)** | 2 |
| | **Positive Differences(b)** | 20 |
| | **Equals(c)** | 0 |
| | **Total** | 22 |
| a Exercises Session 2 < Exercises Session 1 | | |
| b Exercises Session 2 > Exercises Session 1 | | |
| c Exercises Session 1 = Exercises Session 2 | | |

Table 21
Contrast Statistic (b)

| | Exercises Session 2 - Exercises Session 1 |
|---|---|
| **Exact Significance (2-tailed)** | .000(a) |
| a Binomial distribution has been used | |
| b Sign test | |

Therefore, the test shows that the difference between means is also significant in the case of students starting to solve the problems without the Simulated Student. From observing the contrast tests, the Wilcoxon or the signs tests it can be deduced that students solved less exercises in the first session without the Simulated Student than in the second session with it. So, taking into account the results obtained we can conclude that students solved more exercises when they worked with the Simulated Student independently if they used this version in the first or second session.

***Did the Simulated Student Detect Off-topic Conversations?***

The second goal of the Simulated Student was to detect and terminate off-topic conversations. To evaluate the Simulated Student's efficiency in playing this role, the following questions were taken into account:

- Did the Simulated Student detect off-topic conversations?
- How many times did the Simulated Student avoid this situation?
- How many times did the Simulated Student intervene in order to stop the conversations?
- Did the Simulated Student act when students were not participating in off-topic conversations?

Each column of the following table answers one of these questions. As in the previous cases the first column indicates each couple number. The second column indicates the number of times that off-topic conversations took place, the third column shows how many times the Simulated Student detected it, the fourth column expresses how many times the problem was solved, the fifth column indicates the number of times that the Simulated Student intervened and the last column counts the number of times that the Simulated Student intervened unnecessarily.

The last row shows that students had off-topic conversations fourteen times. The Simulated Student detected twelve of these situations and terminated them in eleven cases. So its interventions were successful in 91% of the cases. One intervention was enough to stop the off-topic conversations in each case. Table 22 shows that there were thirteen interventions, one more than the number of situations detected. This is because of an unnecessary intervention. The results indicate that in the majority of the cases (85.7%) off-topic conversations were detected and stopped. This fact supports the third hypothesis (H2: By adding a Simulated Student to HabiPro the duration of off-topic conversations could be decreased).

Table 22
Results related to off-topic conversations

| | **Couple** | **Number of occurrences** | **Detected** | **The problem was solved** | **SS's interventions** | **SS intervened unnecessarily** |
|---|---|---|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 0 |
| | 2 | 1 | 1 | 1 | 1 | 0 |
| | 3 | 0 | 0 | 0 | 0 | 0 |
| | 4 | 0 | 0 | 0 | 0 | 0 |
| | 5 | 1 | 1 | 1 | 1 | 0 |
| | 6 | 1 | 1 | 1 | 1 | 0 |
| | 7 | 3 | 2 | 2 | 2 | 0 |
| | 8 | 1 | 1 | 0 | 1 | 0 |
| | 9 | 1 | 1 | 1 | 1 | 0 |
| | 10 | 1 | 1 | 1 | 1 | 0 |
| | 11 | 1 | 0 | 0 | 0 | 0 |
| | 12 | 0 | 0 | 0 | 0 | 0 |
| | 13 | 0 | 0 | 0 | 0 | 1 |
| | 14 | 0 | 0 | 0 | 0 | 0 |
| | 15 | 0 | 0 | 0 | 0 | 0 |
| | 16 | 1 | 1 | 1 | 1 | 0 |
| | 17 | 1 | 1 | 1 | 1 | 0 |
| | 18 | 0 | 0 | 0 | 0 | 0 |
| | 19 | 0 | 0 | 0 | 0 | 0 |
| | 20 | 0 | 0 | 0 | 0 | 0 |
| | 21 | 2 | 2 | 2 | 2 | 0 |
| | 22 | 0 | 0 | 0 | 0 | 0 |
| **Sum** | | **14** | **12** | **11** | **13** | **1** |

We also analysed how many times students had off-topic conversations when they used the version without the Simulated Student. Table 23 shows that double the number of off-topic conversations occurred. A conversation was counted as off-topic when the conversation was about topics not related to the exercises and they had four or more interventions and the length of the sentences was greater than 10 words.

The reasons why there was double the amount of off-topic conversation could be because:

- As has been already explained, sometimes when students did not solve the exercises they started to talk about other topics.
- Nobody controlled whether off-topic conversations took place.

The first row of Table 23 indicates each couple number (C), the second row shows the number of times that off-topic conversations were detected (Nu). The fact that in this case there were double the number of off-topic conversations than when students used the version with the Simulated Student supports Hypothesis 2.

***Did the Simulated Student Detect Passive Students?***

This section presents an evaluation of the role of the Simulated Student in detecting passive students. As was done in the previous section first of all we analysed with what frequency passive students arose in the version without the Simulated Student. Table 24 indicates each couple number, in the first row. The last row shows how many times passive behaviour was detected per couple.

Table 23
Number of Times of Off-topic Conversation. Version Without Simulated Student

| C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Nu | 1 | 2 | 1 | 0 | 2 | 2 | 2 | 3 | 1 | 2 | 3 |

| C | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | Tot |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nu | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 2 | 1 | 28 |

Table 24
Number of times that there were passive students. Version without SS

| C | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nu | 0 | 3 | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 2 |

| C | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | Tot |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Nu | 2 | 0 | 2 | 2 | 3 | 0 | 0 | 1 | 0 | 0 | 19 |

Table 24 shows that nine students showed passive behaviour and in all cases except one (couple number 20) the passive student repeated his/her behaviour. By studying the logs where passive students were detected it was discovered that in seven cases the passive students' partner was a student who proposed correct solutions very often. Perhaps the passive student did not participate because s/he did not want to demonstrate that his/her level of knowledge was less or because it was more comfortable do so.

In the other two cases the members of the group had a similar level of knowledge. However one student took less part in solving the exercises perhaps because of shyness or for other personal reasons.

The partners who worked with a student who participated little often asked the passive student question such as: "are you still alive?" or "are you there?". Although students answered the questions they seldom improved their participation.

The study carried out to analyse the Simulated Student's efficiency in detecting and encouraging passive students was similar to that presented in the previous section. So, similar questions were studied:

- How many times did a student show passive behaviour?
- How many times did the Simulated Student detect it?
- How many times did the Simulated Student solve the problem?
- How many interventions by the Simulated Student were necessary to solve the problem?
- Did the Simulated Student act when there was no passive student?
- How many times did the passive student relapse?

The following table shows the results obtained after studying the logs. The first column indicates each pair's number. The second shows the number of times that students demonstrated passive behaviour. We considered that a student was passive when s/he did not propose a solution during two consecutive exercises and the student did not take part in the chat or his/her participation was very poor, for instance solely with sentences such as: ok, yes, no. The third column represents the number of times that the Simulated Student detected a passive student. The fourth column indicates how many times the Simulated Student intervened. The fifth column expresses how many times the Simulated Student intervened unnecessarily because the student was not in fact passive. The last column shows how many times the passive student repeated passive behaviour sometime later in that session.

The last row of Table 25 summarises the results. In this case one less (eight instead of nine) passive student situations was detected than in the case of using the version without the Simulated Student. This issue was not expected since we thought that working with three students it was easier for students to be passive than in the case of working just with another student. By comparing Table 24 and Table 25 we can observe that the passive students repeated the passive behaviour more often when they worked with the version without the Simulated Student, perhaps because they were not motivated enough or because the Simulated Student helped to avoid students repeating a passive behaviour.

The data in Table 25 shows that the Simulated Student 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, as the last column indicates, one student repeated the passive behaviour, so two interventions were necessary to encourage that student to participate. Usually just one intervention was enough. This situation may have been due to the fact that the duration of the sessions was not very long. Perhaps if the session had lasted longer, more interventions by the Simulated Student would have been necessary.

Table 25
Results related to the passive students' behaviour

| | Couple | Number of times that it occurred | Detected | The problem was solved | SS's. interventions | SS intervened unnecessarily | The passive student repeated his/her behaviour |
|---|---|---|---|---|---|---|---|
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 4 | 1 | 1 | 1 | 1 | 0 | 0 |
| | 5 | 0 | 0 | 0 | 1 | 1 | 0 |
| | 6 | 2 | 2 | 2 | 2 | 0 | 1 |
| | 7 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 8 | 1 | 1 | 1 | 1 | 0 | 0 |
| | 9 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 10 | 1 | 1 | 1 | 1 | 0 | 0 |
| | 11 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 12 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 13 | 1 | 1 | 1 | 1 | 0 | 0 |
| | 14 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 15 | 1 | 1 | 1 | 1 | 0 | 0 |
| | 16 | 1 | 1 | 1 | 1 | 0 | 0 |
| | 17 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 18 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 19 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 20 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 21 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 22 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Sum** | | **8** | **8** | **8** | **9** | **1** | **1** |

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. In the discussion section the possible reasons that led the Simulated Student towards that conclusion will be analysed. The results explained in this section support the hypothesis H3 that a Simulated Student decreases passive behaviour.

### *Other Interesting Results*

In this section some curious results obtained from the questionnaire that students filled in are presented. In one question students were invited to give their opinion about their partners' assistance. Students could express their opinion using five categories (1. Not very useful, 2. A little useful, 3. Useful, 4. Quite useful, 5. Very useful). Figure 5 shows the subjects' opinion about the assistance offered by the Simulated Student.

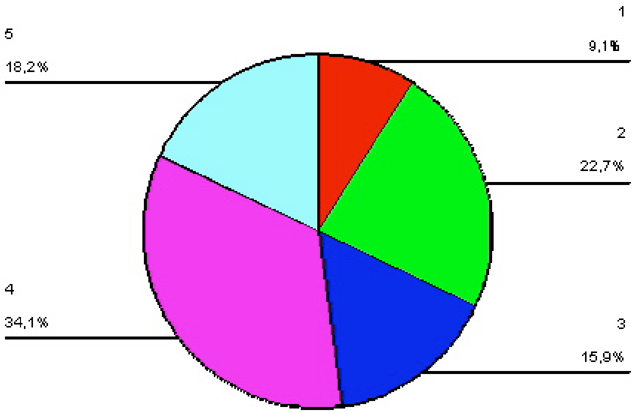


Fig. 5. Students' assessment of the assistance offered by the Simulated Student (1=Not very useful, 5=Very Useful).

Subjects were also asked to comment on their impressions of HabiPro as a method of learning. The system was evaluated in five categories: 1. Very bad, 2. Bad, 3. Normal, 4. Good, 5. Excellent. All these categories had scales from 1 to 5, with 5 being the highest mark. Figure 6 shows the results obtained as percentages.

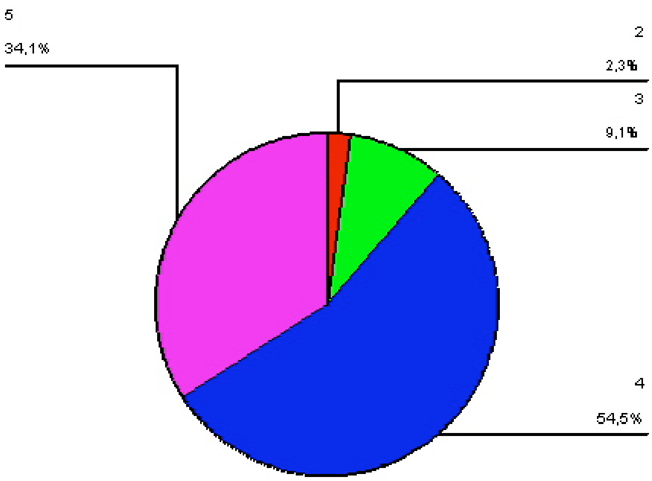


Fig. 6. Students' reaction towards HabiPro (2=bad, 5=excellent).

It is a fact that not everybody likes to work in a group. Some people prefer working or studying alone. We are aware of this reality and for this reason we thought it convenient to add to the questionnaire a question that indicated whether students considered that working with more students could improve their learning. So, they were asked: "Do you think that you would have learnt more alone?". This time there were only two possible answers: Yes or No. Figure 7 displays the answers.

The figure speaks for itself. Most of students believed that working with a partner helped them to learn. This is an important issue that testifies that students valued the help offered by their partners (even though one of them is a Simulated Student).

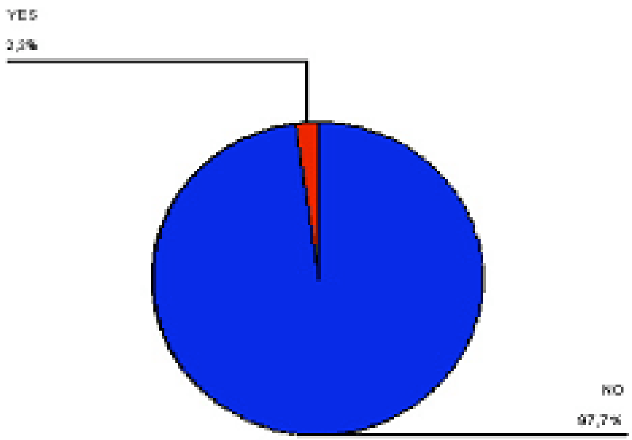


Fig. 7. Results to the question: "Do you think that you would have learnt more alone?"

## DISCUSSION

This section analyses the results presented in the previous section. The discussion starts by analysing the role of the Simulated Student in helping students to solve problems. Next the Simulated Student's unnecessary intervention is studied and finally the role of the Simulated Student in controlling the off-topic conversations is examined.

### The Simulated Student's Role in Helping to Solve the Exercises

From the experiment it was observed that the Simulated Student always intervened when students could not solve the exercises. An example of the Simulated Student's intervention is shown in the following conversation. In this case the Simulated Student is proposing a solution. It does not impose its idea, leaving the students free to check the proposal or to ignore it. The Simulated Student is Student3.

Student1: I don't know how the "for" works.

Student2: Yes, I see that, we have tried a lot of possible solutions and none of them are correct.

Student3: I think that the index of the array must be 0. Let's try j=0.
Student1: Yes!!! Now I remember that the index of an array starts with 0 in Java.

On the other hand, although the Simulated Student intervened when it was necessary, it also acted four times when it was unnecessary. When and why did this occur? The Simulated Student has no natural language capability hence it cannot understand the chat conversation. The Simulated Student uses the information from the answer windows and the number of times that students check a solution in order to decide when to act. So it may occur that the Simulated Student proposes a solution that the another student has just written in the chat window. This would be an unnecessary intervention and this is what in fact happened on the four occasions that the agent intervened inadequately. An example of this situation is shown by the following conversation:

Student1: The solution that we have tried doesn't work.
Student2: I think that the semicolon that is in the first line is not necessary.
Student3: I have found the mistake!!! It is the semicolon in the first line.

Students might think that Student3's behaviour was strange because it proposed the same idea that had already been mentioned in another way. However, in most cases students thought that Student3 wrote the sentence in the chat window at the same time as Student2, but Student2's intervention simply arrived earlier. Everybody who has used a chat application connecting two or more people at the same time knows that such a chat conversation is not as logical as an oral conversation, since, except in the applications that use a turn talking protocol, chat users are not aware whether the others are writing at the same time, and neither of them knows in which order interventions will arrive. So, in this case the expectations of working in a chat application helped us to mask a possible defect of the Simulated Student. However, we have to admit that in one of the cases where the Simulated Student suggested something that had already been proposed, one of the students answered:

Student2: Student3, what are you thinking of? I have already proposed that solution.

Of course the Simulated Student did not answer, and so perhaps a possible off-topic conversation was avoided, since students continued solving the exercises normally.

Table 5 also showed that on four occasions students ignored the Simulated Student's proposal. As was previously explained, students are not obliged to obey the Simulated Student and they can choose between checking what it proposes or not doing so. In future work we will investigate whether there is some relationship between the manner in which the Simulated Student proposes a solution and the probability that students accept or refuse it. The fact that the four times that suggestions were ignored occurred in different exercises might indicate that there is no particular relationship between the two factors.

### The Simulated Student's Role in Detecting Passive Students

From the results obtained and summarised in Table 25 it can be seen that there were eight cases of passive students. All of them were detected and the Simulated Student's intervention caused the passive student to participate in all the cases, although one student repeated his/her passive behaviour.

The Simulated Student intervened once unnecessarily. The log of the conversation was analysed in order to find out why this occurred. The log indicated that the student who was considered passive did not write anything in his answer window for two exercises and the density of interaction in the chat was low. However his interventions in the chat were useful, although not very long. Due to the short density of interventions and to the small amount of participation in the answer window the Simulated Student considered that was passive. Part of the conversation is presented:

Student2: What is wrong in the problem?
Student1: j<5;
Student2: OK, I am going to try it. *(Student2 writes j<5 in the answer window)*
Student1: OK
Student2: Student1, you were right.
Student1: Thanks
Student3: Propose something Student1!, you normally hit the solution.

The conversation shows that the Student1 participated in the chat window. It was even he who proposed the solution, but his interventions were very short. This fact joined to the fact that Student2 wrote the solution in the answer window led the Simulated Student to think that the Student1 was a passive student.

**The Simulated Student's Role in Detecting Off-topic Conversations**

In both previous cases the number of times that a circumstance occurred was equal to the number of times that it was detected. This means that the Simulated Student was a 100% successful. However in off-topic conversations 85.7% of cases were detected. Analysing the logs it was observed that the cases not detected were those where the students' interventions were very short. For instance

Student1: Another exercise!!
Student2: I am tired
Student1: It is bad time
Student2: Yes, siesta time
Student1: Is this the eleventh one?
Student2: I think so

The Simulated Student was not programmed to check whether sentences formed of less than five words were off-topic. It is not very usual for off-topic conversations to have just four of five words, besides this does not mean a great waste of time. The Simulated Student acted as was arranged. If it were necessary to avoid this situation it would be easy by just adjusting the number of words or checking all the interventions without taking into account the length of the sentence.

The Simulated Student's intervention was useful in 91% of the cases (11 out of 12). The example below shows how the Simulated Student acted in order to stop conversations.

Student2: Who are you?
Student1: I am Ana and you?
Student2: My name is Jose and what is your name Student3?
Student3: Let's finish the exercises and we can talk later
Student3: I think that one variable is not declared.
Student2: Yes, the "X" is not declared!!

The Simulated Student detected that students were talking about other topics and by proposing a solution tried to divert the students' attention back towards the exercises.

In one case the off-topic conversation problem was not solved since students ignored the Simulated Student's intervention and continuing talking about other topics. However the fact that the Simulated Student did not intervene again indicates that the duration of the conversation was not very long. If it had lasted longer the Simulated Student would have intervened a second time. It would have been interesting to see what would have occurred after a second intervention.

In the case of off-topic conversations there was one unnecessary intervention. The reason for this was that although students were talking about the problem they did not use words related to Java or to solving the exercises and they sometimes used slang expressions. Because of this none of the words was found in the Simulated Student's database. This fact led the Simulated Student to think that an off-topic conversation was taking place. The following conversation and the Simulated Student's unfortunate intervention are shown.

Student1: This looks really hard
Student2: I hadn't got a clue about the last one.
Student2: But, d'you know what they're asking?
Student1: Yeh, something like the last one
Student3: Stop the chat and let's think a little!!

Students had to face a new exercise and Student2 did not understand what the exercise was asking for. Student3 did not find keywords so it decided to act. If the Simulated Student had been able to process natural language this intervention would probably not have occurred.

## FUTURE WORK

The data obtained from the experiment indicated some situations where the Simulated Student acted unnecessarily or could not amend the situation. For example the Simulated Student did not detect one off-topic conversation because it consisted of very short interventions. The Simulated Student will be improved in order to correct the problems detected. So, in order to improve the performance of the Simulated Student detecting passive students all the interventions will be checked independently of the length of the sentence. We will also add more words to the Simulated Student's database to widen the possibilities of detecting off-topic conversations.

To prevent the Simulated Student from repeating something that has already been said we are exploring the use of techniques developed in the field of natural language processing. We are working in this issue with CICESE, a Mexican research centre, which is in the process of analysing the conversations obtained from our experiments, in order to adapt the syntactic analyser for Spanish that he has developed to help the Simulated Student achieve a better understanding of the conversations (Ibarra, Favela and López, 2000).

## ACKNOWLEDGMENTS

I would like to thank Ben du Boulay for his collaboration in this research. Without his help this work would never have come into existence. I also want to thank the referees for their useful comments which helped me to improve the paper.

## REFERENCES

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anzai, Y., & Simon, H. A. (1979). The theory of learning by doing. *Psychological Review*, 86, 124-140.

Ayala, G., & Yano, Y. (1995). Interacting with a Mediator Agent in Collaborative Learning Environments. In Y. Anzai, K. Ogawa & H. Mori (Eds.) *Symbiosis of Human and Artefact: Future Computing and Design for Human-Computer Interaction*, (pp: 895-900). Advances in Human Factors/Ergonomic. Elsevier Science Publishers.

Burton, M., Brna, P., & Treasure-Jones, T. (1997). Splitting the Collaborative Atom: How to Support Learning about Collaboration. In du Boulay, B. & Mizoguchi, R. (Eds.) *Artificial Intelligence in Education: Knowledge and Media in Learning System* (pp. 135-142). Amsterdam: IOS.

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

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

Chan, T.-W., & 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.

Constantino-González, M. A., & Suthers, D. D., (2000). A Coached Collaborative Learning Environment for Entity-Relationship Modeling. In G. Gauthier, C. Frasson & K. VanLehn (Eds.) *Proceedings of ITS 2000* (pp: 324-332). Berlin: Springer.

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

Dillenbourg, P., & Self, J.(1992). A Computational Approach to Socially Distributed Cognition. *European Journal of Psychology of Education*, 3(4), 353-372.

Dillenbourg, P. (1999). Introduction: What Do You Mean By "Collaborative Learning"? In P. Dillenbourg (Ed.) *Collaborative Learning Cognitive and Computational Approaches*. Elsevier Science.

Doak, E.D., & Keith, M. (1986). Simulation in teacher education: The knowledge base and the process. *Tennessee Education*, 16(2), 14-17.

Goodman, B., Soller, A., Linton, F., & Gaimari, R. (1998). Encouraging Student Reflection and Articulation using a Learning Companion. *International Journal of Artificial Intelligence in Education*, 9, 237-255.

Grudin, J. (1994). Groupware and Social Dynamics: Eight Challenges for Developers. *Communications of the ACM*, 37(1), 92-105.

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

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

Ibarra, M.A., Favela, J., & López, A. (2000). Syntatic-conceptual Analysis of Sentences in Spanish Using a Restricted Lexicon for Disambiguation. In O. Cairo, L. Sucar, & F. Cantu (Eds.) *MICAI2000: Advances in Artificial Intelligence* (pp 538-547). LNAI1793. Berlin: Springer.

Inaba, A., & Okamoto, T. (1997). The Intelligent Discussion Coordinating System for Effective Collaborative Learning. *Workshop Artificial Intelligence in Education*, pp 26-33.

Inaba, A., Supnithi, I., Ikeda, M., Mizoguchi, R., and Toyoda, J. (2000). How Can We Form Effective Collaborative Learning Groups?. In G. Gauthier, C. Frasson & K. VanLehn (Eds.) *Proceedings of 5th International Conference Intelligent Tutoring Systems 2000*. Berlin: Springer.

Johnson, W.L., Richel, J., Stiles, R., & Munro, A. (1998). STEVE: Integrating pedagogical agents into virtual environments. *SIGART Bulletin* 8, 16-21.

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

Kasai, T., & 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).

Luckin, R., & du Boulay, B. (1999). Ecolab: The Development and Evaluation of Vygotskian Design Framework. *International Journal of Artificial Intelligence and Education*, 10(2), 198-220.

Ohlsson, S., & Rees, E. (1991). The function of conceptual understanding in the learning of arithmetic procedures. *Cognition and Instruction*, 8, 103-179.

Okamoto, T. (1996). The intelligent Discussion Supporting System embedded Coordinator Agent under Distributed Environment. In *Proceedings of JCKBSE'96 Second Joint Conference on Knowledge-Based Software Engineering*, Bulgaria, September.

Okamoto, T., & Kasai, T. (1999). Students' Strategies in Collaborative Learning Environment with Multi Companions Agents. In L. Baggott, & J. Nichol (Eds.) *Intelligent Computer and Communication Technology: Teaching & Learning for the 21st Century* (pp. 45-53). PEG99, Ninth International PEG Conference, School of Education, University of Exeter, England.

Paiva, A. (1997). Learner Modelling for Collaborative Learning Environments. In B. du Boulay and R. Mizoguchi (Eds.) *Proceedings of AI-ED 97* (pp 215-230) Amsterdam: IOS Press.

Ramírez Uresti, J. A. (2000). Should I Teach my Computer Peer? Some issues in teaching a learning companion. In G. Gauthier, C. Frasson & K. VanLehn (Eds.) *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems* (ITS'2000) (pp.103-112). LNCS 1839. Berlin: Springer.

Schank, R., & Kass, A. (1996). A Good-Based Scenario for High School Students. *Communications of the ACM* 39(4). 28.

Singley, M., Fairweather, P., & Swerling, S. (1999). Team tutoring systems: Reifying roles in problem solving. In *Proceedings of Computer-Support for Collaborative Learning (CSCL '99)*, Stanford, California, 538-548.

Soller, A., Goodman, B., Linton, F., & Gaimari, R. (1998). Promoting Effective Peer Interaction in an Intelligent Collaborative Learning System. In *Proceedings of the 4th International Conference on Intelligent Tutoring Systems* (ITS 98) (pp 186-195). Berlin: Springer.

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

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

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

Vizcaíno, A., & Prieto, M. (2000). Examining the Effectiveness of New Technology in High School. In *Proceedings of the International Symposium of Technologies of Information and Communication in Engineering and Industry*. TICE'2000 (pp355-360). Troyes, France, October 2000.

Vizcaíno, A., Contreras, J., Favela, J., & Prieto, M. (2000). An Adaptive, Collaborative Environment to Develop Good Habits in Programming. In G. Gauthier, C. Frasson & K. VanLehn (Eds.) *Proceedings of the 5th International Conference on Intelligent Tutoring Systems* (pp. 262-271). Berlin: Springer.

Vizcaíno, A. (2001a). Can a Simulated Student Avoid Negative Situations in Collaborative Environments?. In P. Dillenbourg, A. Eurelings, K. Hakkaraine (Eds.), *Proceedings of the First European Conference on Computer-Supported Collaborative Learning* (pp 704-705). Maastricht, the Netherlands, March 2001.

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

Webb, N. (1989). Peer Interaction and Learning in Small Groups. *International Journal of Educational Research,* 13, 21-40.