The LECOBA learning companion system: Expertise, motivation and teaching

Article *in* International Journal of Continuing Engineering Education and Life-Long Learning · January 2001

DOI: 10.1504/IJCEELL.2001.000395

1 author:

Jorge Adolfo Ramirez Uresti
Tecnológico de Monterrey
36 PUBLICATIONS **304** CITATIONS

SEE PROFILE

The LECOBA learning companion system: expertise, motivation and teaching
by Jorge A. Ramirez Uresti
International Journal of Continuing Engineering Education and Life-Long Learning (IJCEELL), Vol. 11, No. 3, 2001

Abstract: A Learning Companion System (LCS) is a variation of an Intelligent Tutoring System (ITS) where besides the tutor and the student a third agent is added a Learning Companion (LC). The exact nature of the role of the learning companion is one of the most important issues of these systems. This paper describes an LCS for Boolean Algebra (LECOBA) implemented to explore the role of the companion as a student of the human student (Learning by Teaching). To implement such a system, issues such as the motivation of the student to interact with the companion and the LC's knowledge of the domain had to be dealt with. LECOBA provides companions with two types of expertise weak and strong, and two types of motivation Motivated and Free. The evaluation of the system suggested that subjects who faced the weak learning companion in the motivated condition showed a trend of greatest learning gain, though the differences were not significant.

The LECOBA Learning Companion System: Expertise, Motivation and Teaching.

Jorge A. **RAMIREZ URESTI**

School of Cognitive and Computing Sciences
University of Sussex
BRIGHTON, BN1 9QH, U.K.
juresti@itesm.mx
jorgeru@cogs.susx.ac.uk
http://www.cogs.susx.ac.uk/users/jorgeru/

Abstract

A Learning Companion System (LCS) is a variation of an Intelligent Tutoring System (ITS) where besides the tutor and the student a third agent is added: a Learning Companion (LC). The exact nature of the role of the learning companion is one of the most important issues of these systems. This paper describes a LCS for Boolean Algebra (LECOBA) implemented to explore the role of the companion as a student of the human student (Learning by Teaching). To implement such a system, issues such as the motivation of the student to interact with the companion and the LC's knowledge of the domain had to be dealt with. LECOBA provides companions with two types of expertise: weak and strong, and two types of motivation: Motivated and Free. The evaluation of the system suggested that subjects who faced the weak learning companion in the Motivated condition had a trend of most learning gain, though the differences were not significant.

Keywords: Intelligent Tutoring Systems, Learning by Teaching, Learning Companion Systems, Pedagogical Agents.

Bibliographical Notes: Jorge A. Ramirez Uresti received a B.Sc. in Computer Systems Engineering (1991) and a M.Sc. in Computer Science (honours, 1994) from the ITESM-CEM, Mexico. He worked as a Research Assistant in the Centre of Technology and Information at ITESM-CEM from 1992 to 1995. In 1995, he started reading for a D.Phil. in Computer Science and Artificial Intelligence at COGS, University of Sussex, UK. Since January of 2000 he has been working at the Department of Computer Science of the ITESM-CEM, Apdo. Postal 50, Modulo de Servicio Postal, Atizapan, Edo. Mexico, MEXICO 52926. juresti@campus.cem.itesm.mx

1 Introduction

An Intelligent Tutoring System (ITS) can be seen as a system with two agents: a tutor and a student (Figure 1). A criticism of such systems is that they are inherently based on one-to-one interactions between a student and a tutor and cannot encompass the richer learning possibilities opened up by involving more than one learner.

Learning Companion Systems (LCSs) were first introduced by Chan and Baskin [1]. These are systems which attempt to model groups of learners as in a classroom. A LCS consists of at least three agents. The tutor and the student remain the same as in an ITS. The new addition is an agent called the Learning Companion (LC), or just the companion (Figure 1). The role of this new agent is to be a peer of the human student. In principle this companion could be helpful to the student in a number of ways. For example, the companion could be a role model for the student [2, 3]; both students could collaborate or compete as equals [4]; the companion could be a source of advice [5]; the companion could be a student of the human student [6].

[Figure 1 about here please]

This paper describes work being done in the latter category: a learning companion as a student of the human student. A LCS for Boolean Algebra has been built to explore the hypothesis that a less capable learning companion would be helpful to students in their learning by encouraging them to teach it. This hypothesis has given rise to three issues when designing and implementing a LCS. First, a companion with less expertise is proposed as helpful but, how capable should it be in order to be of pedagogical value to the user? Second, is there a necessity to motivate the user to collaborate with a less capable companion? And finally, if the role of the companion is to be a student of the user, how can it be taught by her?

2 Teaching the LC

An important issue in LCSs is the role of the companion in the system. In particular its role as a student of the user has just begun to be explored [6, 7]. The results of the experiments conducted so far have not been encouraging. Students did not benefit, as it was expected, by a learning by teaching interaction with the companion.

However, there is much evidence to suggest that Learning by Teaching can be a facilitator for learning. Students who teach other students learn more and better (see for example [8, 9, 10, 11, 12]). A student who needs to teach other people will have to revise, clarify, organise and reflect on her own knowledge in order to be able to teach, i.e. the student will need to master the knowledge. A learning companion with less knowledge than the student should in principle be helpful for the student to learn by teaching. Therefore, we suggest that *a less capable learning companion would be helpful to students in their learning by encouraging them to teach it*.

3 LECOBA

A LEarning COmpanion system for Boolean Algebra (LECOBA) has been developed to explore the hypothesis that a less capable learning companion would be helpful to students in their learning by encouraging them to teach it. The system consists of three agents: a tutor, a learning companion and a student.

A screen dump of LECOBA is shown in Figure 2. The figure shows the system at the moment when the students are beginning to work on a problem. The windows shown here are: 1) the tutor's window at the top left corner, 2) the LC's window at the top middle of the screen, 3) the student's window at the top right corner, 4) a tool for the student to solve problems ("Simplification Tool") at the bottom left corner, and 5) a window for the student to give suggestions to the companion ("Student - Suggestion") near the middle of the screen.

[Figure 2 about here please]

The interface of LECOBA was implemented in Java 1.1. This interface communicates via sockets with several Prolog programs that control all the events in the system. In particular one of the programs is a powerful simplification module which allows simplification of most Binary Boolean Algebra expressions. The simplification is based on the order in which rules are applied to expressions. Thanks to this order the simplification module can emulate various simplification approaches people use in different scenarios. This module is used to implement the two types of companions present in LECOBA (weak and strong) and the tutor. Finally, the student model (SM) was also developed in Prolog. It is based on the SMART student modelling paradigm by Shute [13]. The implementation is such that at every step in the problem resolution, the SM uses all the information available to the system: the current expression being simplified, all the possible rules that can be applied, the rule the tutor would choose and the rule the student chooses.

3.1 The Tutor

The tutor's utterances are displayed in the tutor's window. The tutor teaches the laws and theorems (rules) of Binary Boolean Algebra, how to use them, and when. Emphasis is on teaching when each rule is best used to simplify a Boolean expression. Even though it is a complex problem to decide which rule is best to use at any given moment, the system gives general guidelines which could be used successfully with many problems. The goal is to introduce students to Boolean Algebra and to the basis of Boolean simplification.

The tutor is in charge of teaching both learners --- the companion and the human student. Because the main objective of the system is to explore the interaction between the companion and the student, the tutor is designed to be as unobtrusive as possible. Its tasks are to teach concepts to students, to give examples using those concepts, to select problems for the students, and to comment on the students' performance. Only when the tutor is doing one of these four activities does it come into contact with the students. First, the tutor teaches a concept followed by some application examples. Then, it gives one problem to both students. Once it has given the problem, it 'disappears' from the student and the companion's sight and its task is to monitor them while solving this problem. In this way, the interaction can focus on that between the student and the companion. Finally, after they have finished solving the problem, the tutor comments on their performance. This cycle continues until the end of the curriculum.

3.2 The Learning Companion

The learning companion is the most important part of the system. It is implemented using simulation techniques. This means that it does not actually learn while the tutor is teaching. The companion's knowledge is secretly selected by the tutor depending on the problem and necessities of the moment. Actually, the tutor is controlling the companion throughout the complete interaction. Effectively, the companion and the tutor communicate with each other privately without letting the student know. For example, when both learners are working on a problem the tutor could tell the

companion to ask a question to the student or to give a suggestion which is expected to help in the resolution of the problem.

The role of the LC is not only being a student of the human student. It is also a collaborator. The companion is capable of helping the student when she is working on a problem. It may give suggestions and justifications, exemplify how its suggestions could be applied and give its opinion on previous moves by the student. On the other hand, when the companion is the one solving a problem, it is able to ask the student for agreement and to express its lack of knowledge of a concept. In this way, by being doubtful or accepting its ignorance, the LC might encourage the student to teach it. However, there may be occasions in which the student refuses to help the LC. In such cases, the companion will try to solve the problem by itself.

3.3 The Student

The student is in control of the interaction between her and the companion. Her window (shown in Figure 2) displays a series of messages from which she can choose one to send to the companion. In the figure the student has just chosen a message which says "I suggest you to use..." so the suggestion window has just opened --- this window gives the student a choice of the rules studied so far to suggest to the companion.

When the tutor presents a problem to the learners (student and companion), it makes it clear that they both have to work on it as a team. There is flexibility for the division of labour within the team. It is up to the student to decide who will actually solve the problem and if she will interact with the companion or not. The student is free to choose not to interact with the companion if she wants.

3.4 Teaching Window

LECOBA allows students to teach the learning companion via a teaching window --- this window is presented in Figure 3.

[Figure 3 about here please]

The teaching window presents to the student the companion's understanding of the domain at a specific moment during the interaction --- i.e. its 'student model'. The objective is to let the student see exactly what the companion knows when trying to solve a problem. In Figure 3, the companion's knowledge is represented by a series of buttons and menus --- each button with its corresponding menu. The first four buttons and menus are enabled. These represent the Boolean rules that they have studied so far. Rules are ordered in the priority that the companion will try to apply them to a Boolean expression. For instance, rule 'OR0X' is labelled as '1st', this means that this rule is the first one which the companion will try to use when solving a problem. Rule 'AN00' is the last one the companion would consider. The order in which the companion uses rules can be changed by clicking on their corresponding buttons and then swapping position with another rule. The menus allow the student to tell the companion how to use a rule. She can tell it to use rules in a specific mode or not to use them at all.

To teach the companion, students must change the companion's knowledge to make it more suitable for the task. Students can enable, disable, change the order and modify the way in which rules are used by the companion.

The teaching window can be seen as a reflection tool for the student. It encourages her to reflect on her own learning before deciding what to teach to the companion. When teaching the companion the student will need to modify its knowledge in the way she thinks it is better to solve the current problem. To select the companion's new knowledge, the student will need to understand why the LC is using that particular rule order. In order to try to understand the companion's knowledge, she will first need to think about her own knowledge of the domain, i.e. what knowledge does she use to simplify expressions and why. In summary, the student will need to revise, clarify, organise and reflect on her own knowledge before she can teach the learning companion.

3.5 Example of an Interaction

When the student starts an interaction with LECOBA she is presented with all the windows displayed in Figure 2 except the "Student - Suggestion" window (near the middle of the screen). The tutor starts by greeting the student and gives her an introduction to the system she is about to use. Immediately thereafter, the tutor begins teaching in its own window the concepts of the first topic of the curriculum. Once the student has studied all the concepts of the current topic, the tutor gives her up to three examples of how to use the rules just seen to solve a simplification problem. After the examples, the tutor presents the student and the learning companion with a problem for them to solve together. At this moment of the interaction the student can decide who will solve the problem: either herself or the LC.

In Figure 2 the student decided that the LC would solve the problem. In its window, the companion acknowledged the student's selection and informed her that it was ready to start solving the problem. The description of the problem to solve is displayed in the tutor's window (top left corner). The LC then informed the student that it wanted to apply rule 'AN1X' as the first step in the resolution. The student has now several choices to communicate with the companion: to give it a suggestion, to teach it, to ask it for a justification of its selection, to tell it that she thinks the problem has been solved or to agree with it.

Teaching and making suggestions to the companion are the main actions which allow the student to "discuss" with the LC while it is solving a problem. For example, in Figure 2 the student decided to give a suggestion to the companion. The student can now decide to suggest any rule in the "Student - Suggestion" window --- the suggestion can be a correct or an incorrect selection. The companion would receive that selection and reply based on its expertise. The reply would allow the student to have an idea of the companion's knowledge of the domain. The student could also have selected to teach the companion. This is a direct way of knowing the companion's expertise. Selecting to teach it would open the teaching window (see Figure 3). As discussed before, this window presents the companion's current understanding of the domain. The student could then compare her knowledge of the domain with the companion's and decide if teaching it and what to teach it, i.e. to explain to it what she thinks is best to simplify expressions. After being taught, the companion would let the student know if it "understood" what she explained to her. In summary, by making suggestions to the companion and by teaching it, the student can discuss with it the best approach to solve the problem assigned to them. Through this discussion with the companion, the student has the opportunity to practice the concepts studied and to reflect in her understanding of them. This reflection should facilitate the student's learning of Binary Boolean Algebra.

The student can also select to solve the problem by herself. In such a case the companion is present to collaborate with her; it can give spontaneous suggestions to the student and the student can request a suggestion from the LC. If the companion spontaneously makes a suggestion, the student can decide to ignore it if she wishes. However, by asking the companion for a suggestion and by

hearing the spontaneous ones, the student is able to discuss with it the best move for the next step. This discussion is aimed at improving the student's understanding of the domain.

Finally, regardless of who solved the problem, after all the discussion between the student and the learning companion, when the learners have decided that the problem is solved, the tutor gives them feedback on their problem resolution. The tutor comments on those steps which were not solved correctly and presents the correct way of solving them. These comments are designed to allow the student to see how her own solution differs from the tutor's solution for comparison and learning from her mistakes. The tutor would next decide the learner's new activity (solve another problem, study more examples, study the current topic again or advance to the next topic) and the cycle of interaction repeats itself again.

4 Expertise: Weak or Strong?

From all the issues surrounding LCS, one of the most important is perhaps the question of the expertise level that the learning companion should have in order to be of educational value to the student interacting with it. Most of the systems implemented so far have dealt in one way or another with this issue. However, only Hietala and Niemirepo [14, 5] have designed their system in order to explicitly study the expertise level of the companion. Their interest was to select the companion's expertise to maintain the student's motivation to collaborate with the LC. They classified companions as *weak* or *strong* based on their expertise. For them, a weak LC is one that has minimal expertise whereas a strong LC has almost an expert-like expertise. Their results showed that, in general, students preferred strong companions, specially when tasks got harder.

Subjects in Hietala and Niemirepo's system did find the learning companion they wanted. But, was it the best learning companion to learn with?

Students, in general, were more comfortable with a strong companion at the end of the interaction, when tasks got harder. This was a very effective way to complete the task on hand, by asking someone who knows more for the answer. However, excessive use of strong LCs may make students not pay attention to the teaching they are receiving, as in the systems developed by Chan and Chou [6]. If students end up by letting the learning companion do all the work and having a passive attitude towards their learning, there would be few benefits in having an LC, quite the opposite, there would be disadvantages instead of advantages.

On the other hand, subjects in the experiment used weak learning companions at the beginning of the interaction. Most probably this was because companions were not labelled with their expertise level --- students did not know which type of companion they were using. This inclarity was deliberately introduced into the system to force students to test all the companions until they had found the one they preferred. The use of weak learning companions was therefore due to a search for the best companion for a given individual, the one preferred. Although, it must be said, that some students did prefer weak learning companions, but this was not the most common choice. So, in general, weak companions were used mainly in the search for the preferred companion but were not good enough to collaborate with for serious tasks. However, interacting with a weak companion could potentially benefit more the student's learning than a strong companion could. A weak companion may allow a student to explain and teach to it. Seen in this perspective, a weak learning companion would give many benefits to a collaboration with a LC.

In contrast with Hietala and Niemirepo's findings, and using their terminology, it is argued in this work that weak learning companions may be more beneficial to the learning of the students, weak LCs may allow them to learn by teaching [15].

LECOBA has been designed to allow the use of weak and strong companions. Only one type of companion, weak or strong, is present during an interaction of a particular student with the system and cannot be changed. The selection of the expertise of the LC, weak or strong, is done before a user starts interaction with the system. Users, therefore, do not know what type of companion they are working with; they might discover it during the course of the interaction. This feature of LECOBA allowed more investigation into the expertise level that LCs should posses.

5 Motivation

Motivating students to use the learning companion(s) available in a system is an important issue which has not been studied enough. We cannot assume that students will take advantage of the LCs just because they happen to be present in the system.

Hietala and Niemirepo's [5] work explored this issue. Their interest was to select the companion's expertise to maintain the student's *motivation* to collaborate with the LC. Their result was that strong companions, in general, kept student's motivation 'alive for the collaboration'.

However, when what is intended is to motivate students to collaborate with a particular kind of learning companion, which might be either weak or strong, Hietala and Niemirepo's motivational factor is, in fact, part of the LC's desired characteristics. This is not to say that expertise as a motivational factor is of no use. It is quite the opposite, expertise has been shown to be a good motivational factor, but for some cases it is not possible to use expertise as a motivational factor.

At this point, one may well be thinking why is it necessary to motivate students to collaborate with learning companions? To answer this question consider the case of a student using a system with a weak learning companion. A weak learning companion has poor knowledge of the domain so its comments, suggestions, etc. are often incorrect. The student interacting with this LC may at the beginning interact with the LC just out of curiosity. However, once the student realises that the LC does not possess a good knowledge of the domain, she may well decide not to use the LC anymore, or worse, to stop using the system, wasting the benefits of having a companion.

A similar case can be envisaged of an interaction with a strong companion. Once students realise that a strong companion knows a lot of the domain, they can decide to let the companion do all the work --- as in Hietala and Niemirepo's research where strong LCs are found to be the ones preferred by students solving difficult tasks. In this case motivating students to use LCs is not the issue, the issue is to motivate students to work more by themselves.

Given the nature of the experiment (see next section), in LECOBA the degree of the companion's expertise could not itself be the motivating factor. Therefore, another form of motivation was required. Students interacting with a weak LC need to make an extra effort (specially to teach it), so they may need to be motivated to do so. Students interacting with a strong LC may need to be motivated to work more by themselves and not to direct the companion to do all the work. LECOBA motivates students in two different ways: 1) Using scores (Motivated mode) and 2) Reminding students that collaboration with the LC is beneficial for them (Free mode).

Scores will help to explore if students collaborate more with the companion when they are pressured to do so. Figure 2 shows the scores mechanism in LECOBA. In each one of the tutor's, companion's and student's windows there is a score, whose value ranges from 0% to 100%. The scores in the companion's and student's windows are based on the performance of each respective learner. Scores are designed to challenge the student to interact more with the companion or to work

more by herself. The challenge is to obtain the maximum score in the 'Total Score' in the tutor's window. This score is determined not only by the student's performance, but also by the companion's performance. The companion's performance can be improved mainly by teaching the companion. The student's performance improves as a mixture of her own work and her involvement in the problem resolution when the LC is working. While the 'Total Score' has not reached its maximum value, students will continue studying the same topic in the curriculum.

The second way to motivate students is just by reminding them that interacting with the learning companion is beneficial for their learning --- Free mode. Students are allowed to advance in the curriculum just by solving three consecutive correct problems. This type of motivation permitted to observe if students do get bored with companions, specially of weak ones, and if when working with strong LCs, students tend to direct the companion to do all the work.

6 Evaluation

LECOBA was tested under four different conditions to explore the effects of expertise and motivation in the student's learning. These conditions are summarised in Table 1.

Experimental Conditions		
Motivation	*Expertise*	
	Weak	Strong
Motivated	CONDITION 1 Student gets a lot of opportunities to teach the LC. *Student learns more than in the other conditions.*	CONDITION 2 Student gets few opportunities to teach the LC. *Less learning than in Condition 1.*
Free	CONDITION 3 Student ignores opportunities to teach the LC. *Student gets bored and may stop interaction with the LC.*	CONDITION 4 Student lets the LC do all the work. *Student hampers her own learning.*

Table 1: Experimental conditions under LECOBA. Hypotheses are shown in italics.

For each of the conditions in Table 1 students were expected to learn Binary Boolean Algebra. However, each condition was expected to influence the student's learning to different degrees. Condition 1 (Weak/Motivated) was expected to be the best one of the four conditions for helping students learn the most. In this situation a student would be strongly motivated to interact with the companion and thus have many opportunities to learn by teaching the LC.

6.1 Subjects and Design

The experiment was a 2x2 factor, between-subjects design. There were two independent variables: expertise (weak and strong) and motivation (Motivated and Free), and one main dependent variable: learning gain. Subjects were assigned randomly to each one of the four conditions.

Thirty two (32) undergraduate engineering students in their first and second years at the University of Sussex volunteered to take part in the experiment. They were paid for their participation.

Students in their first year had attended a course, where Binary Boolean Algebra was taught, the term before the experiments took place. LECOBA provided them with an opportunity to use a tutoring system to revise before the examination in that same term. Students in their second year had taken the same course, as first year students, in their first year. LECOBA allowed them to review Boolean Algebra before they attended a course where previous knowledge of this domain was needed.

Subjects attended two sessions of one hour each. A pre-test of 15 minutes was administered at the first session. This was followed by a demo of the system. During the rest of the first session and the beginning of the second session subjects had the opportunity to work with the system. The time of the interaction with LECOBA was set to 30 minutes per session. Finally, at the end of the second session students took a post-test of 15 minutes.

6.2 Results

6.2.1 Learning Gains

The experiments with LECOBA showed that subjects who faced the weak learning companion and who were strongly motivated to interact with it (Condition 1, Weak/Motivated) had a trend of most learning gain, though the differences were not significant. On the other hand, subjects who interacted with a strong companion and who were mildly motivated (Condition 4, Strong/Free) had a trend of worst learning, though again the differences were not significant.

6.2.2 Motivation

The result of the research shows that the strong motivation of the Motivated condition encouraged subjects to work much harder than subjects on the Free condition. Subjects facing the weak companion interacted frequently with it in the Motivated condition. In the Free condition they decreased their interaction with the weak LC to the point that at the end of the time for the interaction, the collaboration with it was minimal.

On the other hand, as expected, subjects with the strong companion worked more by themselves in the Motivated condition. In the Free motivation, contrary to expectations, subjects facing the strong companion also solved more problems than it. The expectation was that students would let the LC do all the work once they realised that it was capable. However, this result is deceiving as even though these subjects solved more problems than the companion, the logs of interaction showed that these were the subjects who asked most for suggestions to the strong LC. Therefore, there was a trend that these subjects did not select the strong companion to solve problems but preferred to work by themselves and then used the companion as a tool that would provide solutions to problems.

6.2.3 Teaching the LC

Teaching the weak companion was performed significantly more often in the Motivated condition than in the Free condition. Subjects in the Free condition acknowledged the benefits of teaching the LC but, even so, they did not teach it much. On the other hand, as expected, subjects facing the strong companion taught it very little in both the Motivated and the Free conditions. Besides, a small positive correlation was found between teaching the learning companion and the learning gain but, unfortunately, it was not significant. Finally, the experiment showed that the majority of subjects who felt that they had learnt as a result of teaching the companion recognised that teaching allowed them to reflect when they were learning.

7 Conclusion

The results of this research suggest that, depending on the role of the companion, there may be more learning benefits for students learning with a LCS if they face a weak companion than if they collaborate with a strong companion. However, there was some evidence to support the argument that subjects with mild motivation may interact with the weak companion just out of curiosity or because of the tutor's remarks, but once they realise that its expertise is poor, they are very likely to stop interacting with it. In consequence, and adding the fact that teaching was significantly more in the Motivated condition, it was demonstrated that students should be strongly motivated to teach the weak learning companion. Otherwise, with mild motivation, even if students recognise that teaching is helpful for their learning, they will not teach enough to benefit.

As for the strong companion, the results showed that students with a strong motivational condition work more by themselves than subjects in the Free condition. However, it could not be proven that subjects with mild motivation instruct the companion to perform all the work. Nevertheless, the results indicate that the suggestions of the strong companion might foster a passive attitude in the student and as a consequence hamper her learning. This behaviour may hinder the student's learning as she would not have to think so deeply while interacting with the system because most of the answers will be provided by the strong companion

Finally, an evaluation of the role of the companion as a student of the human student remains. Teaching the LC was not an action which permitted a passive attitude. Evidence suggests that it was so demanding when facing the weak companion that the high effort required might have been a reason for subjects in the Free condition to markedly decrease their teaching. But more importantly, the results give also some support to the claim that subjects who are strongly motivated to teach the weak companion benefit most from the teaching interaction. The benefits related to teaching could not have occurred with the strong companion as it does not lend itself for teaching.

Acknowledgements

The author wishes to thank Benedict du Boulay and the anonymous reviewers for commenting on a draft of this paper, and *Consejo Nacional de Ciencia y Tecnología* (CONACYT) of México for their financial support.

References

[1] Tak-Wai Chan and Arthur B. Baskin. “Studying with the prince” the computer as a learning companion. In Proceedings of ITS-88: Intelligent Tutoring Systems, pages 194-200, 1988.

[2] David Gilmore and John Self. The application of machine learning to intelligent tutoring systems. In John Self, editor, Artificial Intelligence and Human Learning, Intelligent Computer-Aided Instruction, chapter 11, pages 179-196. Chapman and Hall Computing, 1988.

[3] Geoff Cumming and John Self. Collaborative intelligent educational systems. In Dick Bierman, Joost Breuker and Jacobijn Sandberg, editors, Artificial Intelligence and Education. Proceedings of the 4th International Conference on AI and Education, volume 3 of *Frontiers in Artificial Intelligence and Applications*, pages 73-80. IOS Press, Amsterdam, Netherlands,1989.

[4] Tak-Wai Chan and Arthur B. Baskin. Learning companions systems. In Claude Frasson and Gilles Gauthier, editors, Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education, chapter 1, pages 6-33. Ablex Publishing Corporation, Norwood, New Jersey, 1990.

[5] Pentti Hietala and Timo Niemirepo. The competence of learning companion agents. International Journal of Artificial Intelligence in Education, 9:178-192, 1998.

[6] Tak-Wai Chan and Chih-Yueh Chou. Exploring the design of computer supports for reciprocal tutoring. International Journal of Artificial Intelligence in Education, 8:1-29, 1997.

[7] Y. Jun. Linear kid: A mathematical software designed as a computer-based peer tutoring system. In Electronic Proceedings of the Second Asian Technology Conference in Mathematics (ATCM´97), 1997. http://www.cs.runet.edu/atcm/EPATCM97/ATCMP008/paper.html.

[8] David Berliner. Being the teacher helps students learn. Instructor, 98(9):12-13, 1989.

[9] Sinclair Goodlad and Beverly Hirst. Peer Tutoring: A Guide to Learning by Teaching. Kogan Page, London, 1989.

[10] Donald Michie, Andrew Paterson and Jean Hayes-Michie. Learning by teaching. In 2nd Scandinavian Conference on Artificial Intelligence 89, pages 307-331. IOS Press, Tampere, Finland, 1989.

[11] David Nichols. Issues in designing learning by teaching systems. In P. Brusilovsky, S. Dikareva, J. Greer and V. Petrushin, editors, Proceedings of the East-West International Conference on Computer Technologies in Education (EW-ED'94), volume 1, pages 176-181, 1994.

[12] Srinivas Palthepu, Jim Greer and Gordon McCalla. Learning by teaching. In Lawrence Birnbaum, editor, Proceedings of the 1991 International Conference on the Learning Sciences, pages 357-363. Association for the Advancement of Computing in Education (AACE), 1991.

[13] Valerie J. Shute. SMART: Student modeling approach for responsive tutoring. User Modeling and User-Adapted Interaction, 5(1):1-44, 1995.

[14] Pentti Hietala and Timo Niemirepo. Collaboration with software agents: what if the learning companion agent makes errors? In Benedict du Boulay and Riichiro Mizoguchi, editors, Artificial Intelligence in Education: Knowledge and Media in Learning Systems, volume 39 of *Frontiers in Artificial Intelligence and Applications*, pages 159-166. IOS Press, 1997.

[15] Jorge A. Ramírez Uresti. Teaching a learning companion. In Gerardo Ayala, editor, Proceedings of the International Workshop "Current Trends and Applications of Artificial Intelligence in Education". The Fourth World Congress on Expert Systems, pages 83-89. ITESM, Monterrey, México, 1998.

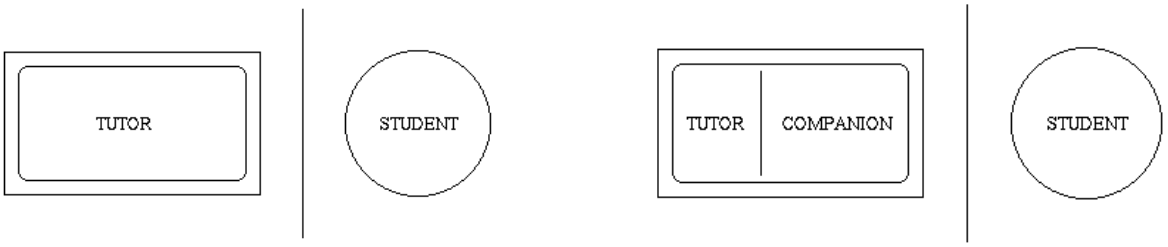

Figure 1: ITS vs. LCS

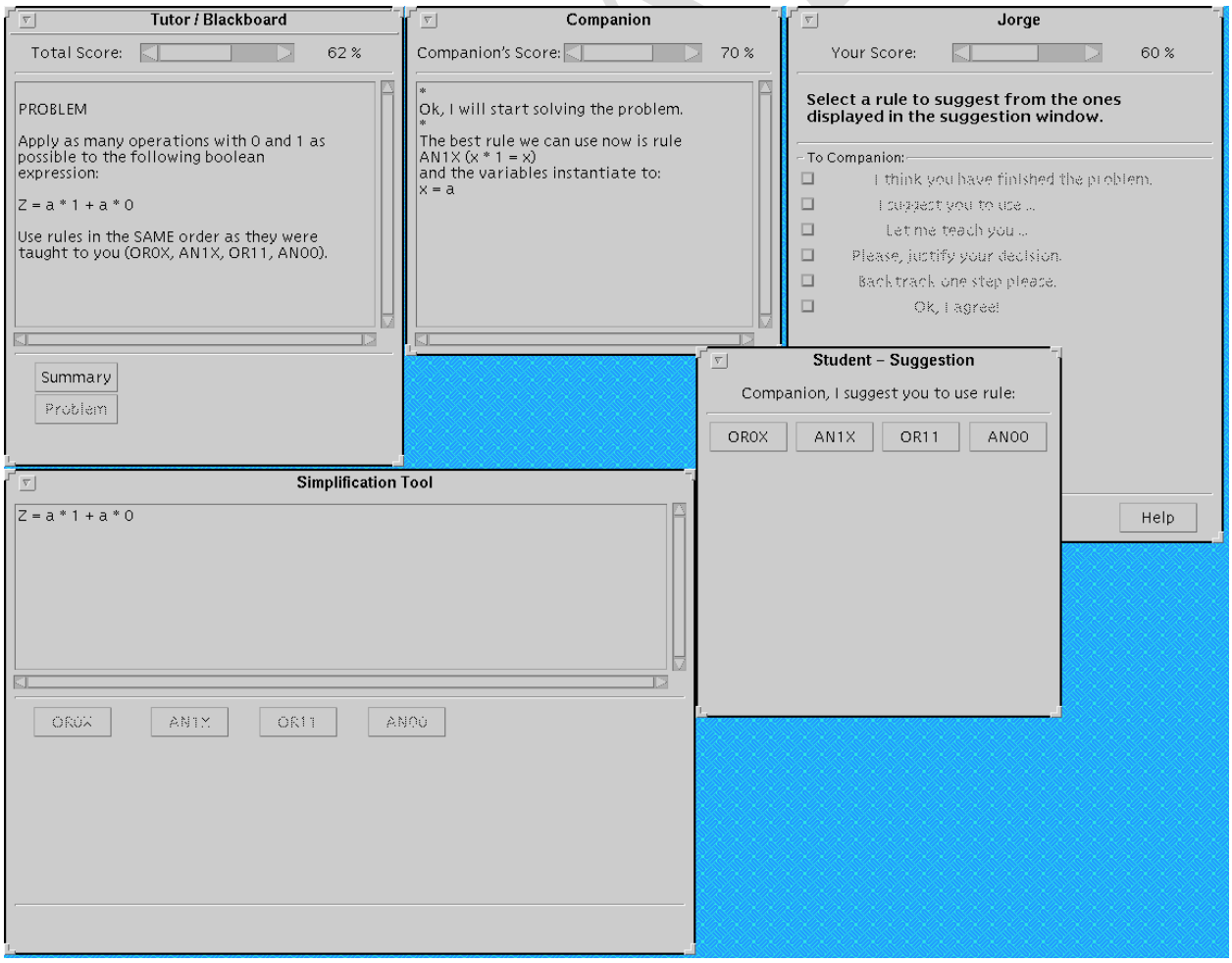

Figure 2: LECOBA. Students are solving a problem, the learning companion is solving the problem, and the user is about to give a suggestion to the companion.

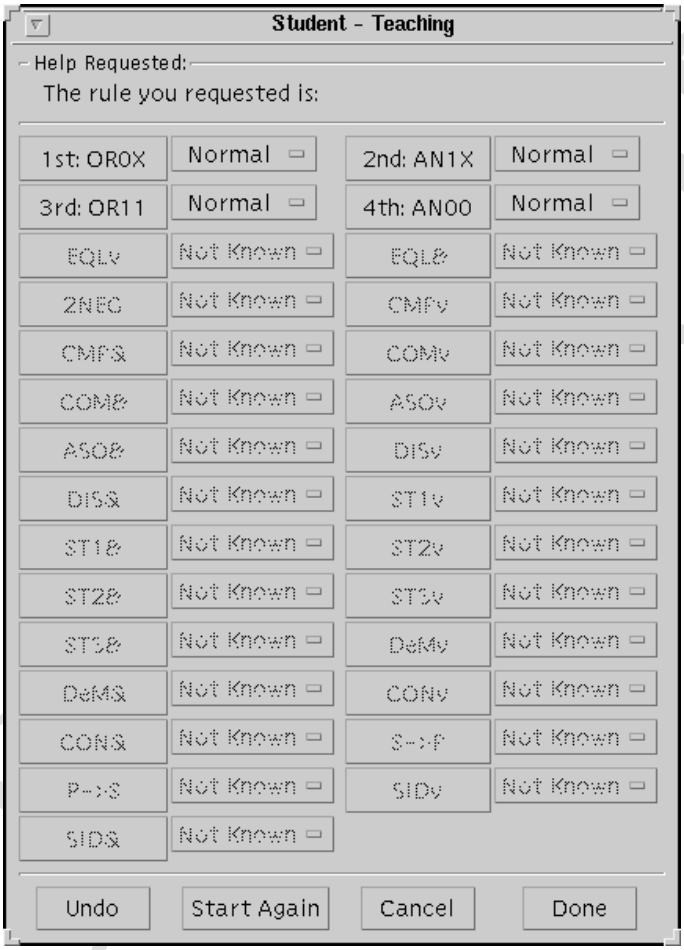

Figure 3: Teaching Window in LECOBA.