In Halloran, J. & Retkowsky, F. (Eds.). *The tenth White House papers. Graduate research in the cognitive and computing sciences at Sussex* . CSRP No. 478 . School of Cognitive and Computing Sciences, University of Sussex. Brighton, U.K. 1998.

Teaching a Learning Companion

Jorge A. *Ramírez Uresti* *

School of Cognitive and Computing Sciences
University of Sussex
Brighton, BN1 9QH
United Kingdom
jorgeru@cogs.susx.ac.uk
Fax: +44 (1273) 671320

Learning Companion Systems (LCS) are a variation of Intelligent Tutoring Systems (ITS). In an LCS, besides the traditional tutor and student, a new agent is introduced: the Learning Companion (LC). The issues of the expertise and behaviour that such a companion agent should have, if it is going to be of any use to the student, are very important in these systems. This paper explores the hypothesis that a less capable learning companion is helpful to a human student by encouraging her to teach the LC.

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 (LCS) were first introduced by Chan and Baskin [4] influenced by the idea of collaborative partners [7, 6]. These are systems which attempt to model groups of learners as in a classroom. An 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 should be helpful to the student in a number of ways. For example, the companion could be a role model for the student; both students could collaborate and compete as equals [5]; the companion could be a source of advice [9]; the companion could be a student of the human student.

figure18

Figure 1: ITS vs. LCS

This paper describes work in this final category. A LCS is being built to explore the hypothesis that a less capable learning companion is helpful to a human student when learning. The student will be encouraged to learn by teaching this kind of companion. The question of how to motivate the student to interact with the LC is addressed. Finally, a possible way in which the student would be able to teach the companion is discussed.

Expertise and Behaviour

Work in LCSs has increased in the last few years but much remains to be done to explore the full capabilities and possibilities of these systems. Self and his colleagues proposed ITSs which offered collaboration with the student rather than instruction [7, 6]. But it was Integration-Kid [3] the first system built as an LCS that introduced the idea of a LC. The LC in this system was capable of collaborating and competing with the student. Integration-Kid proved the feasibility of LCSs and demonstrated that they stimulated learning interactions which are not possible with a tutor only. But, more importantly, it raised the issues of the expertise and behaviour that such a companion agent should have if it is going to be of value to the student.

In Integration-Kid the companion had a knowledge level which is average to students in its domain. With this expertise the companion was thought to display sub-optimal performance which was expected to motivate the student by showing her that this is normal when learning. Integration-Kid behaved as a collaborator or as a competitor. These two behaviours were intended to help the student when solving a problem and to make her reflect when presented with different solutions. Hietala and Niemirepo [10] have explored the effects of different degrees of companion's expertise. Their work used companions with low expertise (weak) and with high expertise (strong). Their interest was to select the companion's expertise to maintain the student's motivation to collaborate with the LC. Their experiments took into consideration the students' general learning capabilities and their personality traits to observe the effect and acceptance of the LC. In general, they found that students preferred strong companions specially when tasks got harder. The LCs in this work were there to give advice to the student and to collaborate with her. Both of the above LCSs try to help the student in her learning activities. They encourage collaboration and suggestions from the LC to help the student reflect on her knowledge. Competition gives the student the opportunity to see a different approach to do the same task and may stimulate her to work harder.

Another helpful way for a student to learn when interacting with a peer is *to teach her peer*. Research has shown that students who teach other students learn more and better [1, 8, 11]. A student who needs to teach other people will have to revise, clarify, organize, 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 weak LC would help the student to learn better by encouraging her to teach it*. Although Hietala and Niemirepo found that strong companions were more used, the companions in their environment had `fixed' knowledge -- 2 strong and 2 weak companions. The student was able to select one of those 4 companions but she was not able to modify the companion's knowledge. Therefore, it was natural that when subjects were faced with more difficult tasks they preferred, in general, to get help from a more knowledgeable companion.

Implementation

An LCS is being developed to explore the idea of teaching a weak LC. The domain of the system is Binary Boolean Algebra. The tutor will teach the laws and theorems (rules) of this domain, how to use them, and when. Emphasis will be 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 will give 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.

As mentioned before, the tutor will be the one 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 being designed to be as unintrusive 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 will it be in contact with the students. Once it has given a problem to them, it will `disappear' from the students sight and its task will be to monitor them while solving the problem. In this way, the interaction could focus on that between the student and the companion. After they have finished solving the problem, the tutor will comment on their performance. This cycle continues until the end of the curriculum.

The learning companion is the most important part of the system. It will be implemented using simulation techniques. This means that it will not actually learn while the tutor is teaching. The companion's knowledge will be secretly selected by the tutor depending on the problem and necessities of the moment. Actually, the tutor will be controlling the companion throughout the complete interaction. It could be seen as if the companion and the tutor were communicating with each other 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 student will be in control of the interaction between her and the companion. The tutor will present the problem making it clear that they 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 it and if she will interact with the companion or not. The student will be able to choose not to interact with the companion if she wants. This freedom of interaction generates the problem of, how to motivate the student to interact with the LC -- this issue is discussed below. If the student decides to interact with the companion, she will have the opportunity to teach it. But, how will the student teach the LC? A suggestion about how to do this is presented below.

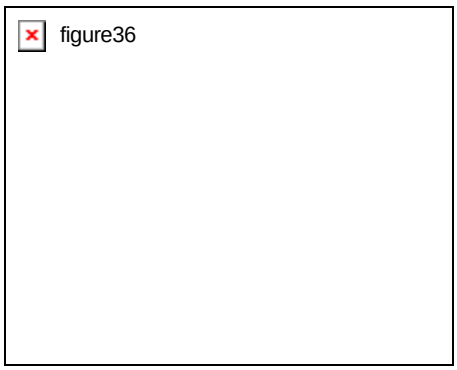

Figure 2: Tutor controlling LC.

Figure 2 presents the general agent architecture of the system being implemented. The solid arrows between the agents represent a communication between two of the agents that is seen by the third agent. The dashed arrow represents a private communication -- in this case between the tutor and the companion.

Motivating the student

Interaction between the student and the companion is essential in any LCS. If at any time the student decides that she does not want to interact with the companion anymore, then either 1) the LCS becomes an ITS -- if the student is able to continue working without interacting with the companion, or 2) the interaction with the LCS is terminated -- if the student must use the LC. Both cases are detrimental to the objective of LCS so care should be taken with the motivation of the student while using the system.

In the implementation described before, the student is not told to interact with the companion. She will have the possibility to decide if she wants to make use of the companion or not. Even in the case where she decides to interact with the companion, there is a risk that after a while the student could get bored with a weak companion. A weak companion will give wrong suggestions and solutions some of the time. It will also appear to be very keen on being taught by the student by asking questions which show that it does not know very much. These actions could make the student see the LC as an annoyance, more work to do. She might get bored with it and decide not to use it anymore. Of course, the student may realize the possibilities for reflective learning provided by the companion and may not need further motivation to work with it.

Our system will have the flexibility of working in two interaction modes: motivated and free. These modes will help to observe if the student gets bored with a weak LC and if pressure on the student to interact with the companion is useful. In free interaction the student will only be told that using the LC is beneficial for her to learn -- specially if she teaches the student. In motivated interaction the score mechanism in figure 3 will be used.

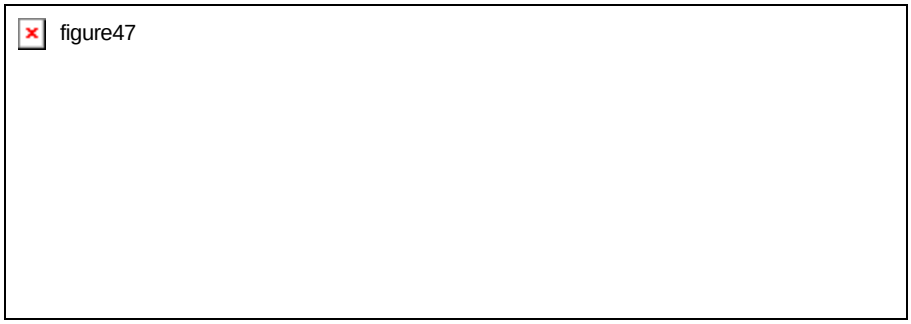

Figure 3: Motivation mechanism.

The aim of the scores is to motivate the student by giving her a challenge. The student will be presented with the three scores in figure 3. The objective is that the student finds a way to obtain enough points to advance a level in the curriculum. This will depend on the total score which is a combined score for both the student and the companion. Tmin marks the minimum number of points to advance a level. The other two scores are the scores for the Student and the Companion respectively. The minimum mark in these scores indicates the minimum number of points each student needs to be able to advance a level. The maximum mark is the maximum points that will be given at a particular level. By working on problems the solver(s) will get some points, whether solutions are correct or not. The key factor is that the companion's points will have more impact in the total score. This way of allocating points is

intended to prevent the student advancing a level without having used the companion enough. It is expected that once the student realizes that the points obtained by the companion are better for the total score, the student will use more and more the LC. Once the student is interacting with the companion she will have the possibility to teach it. The scores should provide enough motivation for the student to make her want to interact with the companion in order to advance a level.

Teaching the LC

In the present LCS implementation the main activity between the student and the companion will be teaching. The weak companion will behave in a way which will encourage the student to teach him. For example, the companion could ask the student if she agrees with him or not in using a simplification rule. Very often during the interaction the companion's suggestions will be wrong, so as a way to disagree with him the student could decide to teach him. The companion could also ask the student directly to teach him. The option of teaching the companion will always be available when the LC is working on a problem.

To teach the LC the student will use a window such as the one in figure 4. This window is based on the idea of inspectable student models [2] in that it represents the knowledge of the companion at a particular moment. The window provides the student with a series of buttons and menus which let her communicate with the companion. It would be better if the student could discuss directly with the companion using some form of natural language, but unfortunately the present state of the art in this area is not enough to support the kind of dialogues which would be needed when teaching.

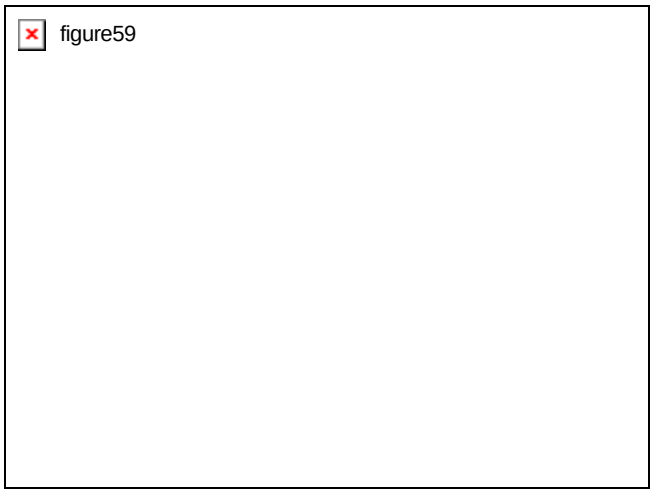

Figure 4: Teaching window.

The teaching window presents the knowledge of the companion at a specific moment during the interaction. The objective is to let the student see exactly what the companion knows when trying to solve a problem. In figure 4, the companion's knowledge is represented by a list of simplification rules labeled `Rule Order'. This list contains, in order of priority, all the rules that the companion knows how to use. In the figure, the companion tries to apply rule `r1' first. If he can not apply this rule, he will try to use the next rule, `r2'. He continues trying rules until one can be used.

When the student is teaching the companion, she will need to modify the list of rules in the way she thinks it is better to solve the current problem. In order for her to select the companion's new rule order she will need to understand why the LC is using that particular order. To understand this she will first need to think about her own knowledge of the domain, i.e. what order does she use to simplify

expressions and why. In summary, the student will need to revise, clarify, organize and reflect on her knowledge before she can teach the LC -- by changing the order of the rules.

The menus and buttons in the window will let the student modify the rule order in which the companion is using the rules. Once she has taught the companion, it will use the rules as it has been told. The student can then appreciate if what she taught the companion is appropriate or not. The student can continue teaching the companion as much as she likes. In a sense, by teaching the companion, the student will have the possibility to experiment in the domain and learn more about it.

The issue of exactly which rules the companion should be primed with secretly by the tutor remains to be solved. One idea is to make the companion's knowledge track that of the student, but some steps behind. In this case the student would effectively be reflecting on her own past performance.

Conclusion

LCS are systems which try to resemble human social environments. The learning companion adds a new dimension to the interaction between computer and users. It is claimed that the benefits of having such a companion are similar to the benefits of having a human peer. However, research must be done in order to understand the actual benefits of a computer companion. In particular, the effect on the student of the companion's knowledge level and behaviour should be studied.

This paper describes a system being implemented that proposes to use a learning companion which has less expertise than the student. The aim is that the student would be able to reflect on her own knowledge by teaching the learning companion. A proposal to motivate the student to interact with the LC is offered. It is essentially challenging the student by giving her a goal in the form of scores. To teach the learning companion a teaching window is proposed which is based on the concept of inspectable student models. This window should help the student to reflect on her own knowledge.

Acknowledgements

The author would like to thank the Consejo Nacional de Ciencia y Tecnología (CONACYT) of México for supporting this research.

References

1 David Berliner. Being the teacher helps students learn. *Instructor*, 98(9):12-13, May 1989.

2 Susan Bull and Elspeth Broady. Spontaneous peer tutoring from sharing student models. 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 143-150. International Society for Artificial Intelligence in Education, IOS Press, 1997. Proceedings of AI-ED 97 World Conference on Artificial Intelligence in Education.

3 Tak-Wai Chan. Integration kid: A learning companion system. In J. Mylopolous and R. Reiter, editors, *IJCAI-91, Proceedings of the Twelfth International Conference on Artificial Intelligence*, volume 2, pages 1094-1099. Morgan Kaufmann Publishers, Inc., August 1991.

4 Tak-Wai Chan and Arthur B. Baskin. ``studying with the prince" the computer as a learning companion. In *ITS-88: Inteligent Tutoring Systems*, pages 194-200, Montreal, Canada, June 1988.

5 Tak-Wai Chan and Arthur B. Baskin. Learning companion 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.

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

7 David Gilmore and John Self. The aplication 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.

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

9 Pentti Hietala and Timo Niemirepo. Studying learner-computer interaction in agent-based social learning environments. In Paul Brna, Ana Paiva, and John Self, editors, *European Conference on Artificial Intelligence in Education*, pages 386-392, Lisbon, Portugal, September 1996. Colibri. Proceedings of Euro AI-ED96.

10 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. International Society for Artificial Intelligence in Education, IOS Press, 1997. Proceedings of AI-ED 97 World Conference on Artificial Intelligence in Education.

11 Srinivas Palthepu, Jim Greer, and Gordon McCalla. Learning by teaching. In Lawrence Birnbaum, editor, *The International Conference on the Learning Systems*, pages 357-363. Association for the Advancement of Computing in Educatio (AACE), 1991. Proceedings of the 1991 conference.