USING SELF-EXPLANATIONS WITH INFORMATION AND COMMUNICATION TECHNOLOGIES TO IMPROVE DISTANCE LEARNING IN COMPUTER SCIENCE

NICOLE TOURIGNY, LAURENCE CAPUS AND ANDRÉ GAMACHE

LSI - ERICAE, Département d'informatique et de génie logiciel, Faculté des sciences et de génie, Pavillon Pouliot, Université Laval, Cité universitaire, Québec, Canada, G1K 7P4
E-mail: {capus, tourigny, gamache}@ift.ulaval.ca

The curriculum of the computer science at our university has the following objectives: 1) acquire knowledge on fundamentals, and methods in computer science and to discover edges of this field to related ones; 2) develop aptitudes and skills needed for gathering and making appropriate abstraction based on data related to a specific problem or situation, elaborate critical analysis and interpretation in order that relevant solutions can evolved from; 3) do efficient team working; 4) structure an enlightened opinion on basic issues in the computer science. In a way that effective, precise, logical and pragmatic solutions can be stated taking into account the requirements of the field.

The final goal is to train students in order to be highly trained in the field of computer science. It means that they should acquire sufficient knowledge in various computer domains (algorithm design, data structures, databases, programming languages, telecommunication, artificial intelligence, etc.) and develop general and specific solving abilities to cope with real and complex problems. It is of prime importance since computer science professionals are always confronted with understanding some new domains, and thus searching for new solutions. It is all the more important for students to become and stay competent because they are challenged on international level.

A crucial question in this context is how to make sure or help students become competent in computer science. A first step is to have pertinent information accessible. This is generally achieved with books, notes, Web sites seen as electronic books, etc. In our department, computer tools like WebCT™ and InterWise™ are used in many courses to make information available to students. However, the mere access to the information does not guarantee the development of student's intellectual and technological skills. A current trend in education promotes an active pedagogy based on the constructivist approach (Jonassen 1991; Jonassen *et al.* 1999). Several studies showed that students who were proactive in acquiring their knowledge learned more and better succeeded than those who were impassive in the intellectual process (Chi 2000). In other words, students really need to be involved in some activity in order to integrate knowledge and use it to solve problems. In fact, information becomes knowledge when it leads to actions (Schreiber *et al*. 2000), which means developing pertinent solutions and using them to solve problems.

However, it is not sufficient to provide students with some intellectual activities to insure skill development. Effectively, one of the lessons learned from our past experiences is that some form of supervision and feedback is needed because most students are not good to self-assess and correct their problem solving process, even when they are provided solutions. Feedback can be given under various forms. For example, in a classroom, teacher is doing so by asking questions and correcting student exercises. But it can be feasible for a teacher to correct all exercises done by each student. So, what can do?

We propose to use the information and communication technologies to answer this issue, using improved functionalities implemented by new technologies. We think that it is important to propose some environment to help students to be active in their education. Furthermore, based on our past experiences, we think that students can learn more if they are involved with peers. This is why we propose that students could work alone or in interaction with peers, in the process of problem solving. A Web environment seems a good means to support students in their learning: 1) a Web environment generally offers several types of tools like communication tools, evaluation of student's knowledge, course management; 2) it makes the learning materials accessible at any time (Capus *et al*. 2002). Systems like WebCT™ and Learning Space™ are two well-known examples of such environments. However, in those Web platforms, students cannot directly participate to solve exercises (Cronjé 2001). They can exchange with other students by means of forums and chats, but there are no tools to organize and well managed the students' solving process interactions centred on the pedagogical material. It is up to the teacher to design and implement such tools.

To answer this need, we first developed SPEAC in our laboratory. This system is a Web environment based on an active pedagogical approach. SPEAC was designed for a university course on artificial intelligence, which is compulsory for the bachelor program of computer science and optional for some other programs. SPEAC contains examples of solved problems. Studying by means of examples is an efficient learning strategy (VanLehn 1998). Furthermore, students are invited to self-explain solved exercises provided and controlled by SPEAC and promoting the use of self-explanation that occurs when one explains something to oneself. The rationale behind SPEAC is based on previous results obtained by Chi *et al*. (1989): those who self-explain more succeed better than those who only read examples. SPEAC allows students to comment explanations made by peers or ask peers to help her/him construct an explanation. This way, students can collaborate to understand how to solve some problems. The teacher can next comment the explanations made by students. In other words, SPEAC allows feedback from peers and from the teacher. This feedback function allows students to validate their knowledge and eventually correct their errors.

We next developed a second generation of SPEAC, called SPHINX, which emphases user-friendly interaction. It also allows student to modify her/his interface preferences (colours, password, etc.). This second generation, SPHINX, also offers more functions for learning support than SPEAC. SPHINX allows student to look at, study and explain an example. But SPHINX can also suggest a student to explain more examples if SPHINX found that a student has so far not explained any examples or not enough. When a student is logged on, SPHINX stores the information related to the actions made by the student (for instance, time, type of action: click, open, close, etc., object of the action: window, example, explanation, etc.). From this information, SPHINX builds a kind of map of course concepts studied by the student. This map is updated according the following actions of the student. Then, SPHINX advises to the student to make another actions in order to enrich her/his map of concepts. In other words, SPHINX can give student some feedback based on information it tracks from student.

SPEAC was implemented and tested in 2002-2003 with around 260 senior students, whereas SPHINX was implemented and tested in 2003-2004 with about 200 students. These two systems are interactive educational tools that allow some kind of interaction between the users of a teaching/learning activity. SPEAC and SPHINX offer mechanisms to use efficiently pedagogical material. They also allow learning of problem solving process in a non-traditional pedagogical approach where teachers and students can cooperate like partners (Lankard Brown 1998). This is why we chose to develop a Web environment to let students learn by means of examples, self-explanations, explanations, collaboration and feedback. Students outside classroom between lectures can use SPEAC and SPHINX. It means that students can access a pedagogical environment anytime and obtain some feedback for their work. Feedback can be give by peers that comment other explanations, by teacher that comments student’s explanations, and by SPHINX that can encourage students to study and explain more exercises.

Evaluations of these systems were made with questionnaires and interviews with students and teachers about their appreciation of the environment they used (SPEAC or SPHINX). Most students declared that they have liked this additional pedagogical support, and they should have used it much more. In fact, students used SPEAC or SPHINX more to consult exercises than to explain them within the tool. The students that used the SPHINX environment the most often obtained a better average compare to those using it only a little than the one of the class. They also generally appreciated the feedback they received. For teachers, SPEAC and SPHINX offer a means to know more about students’ progress between lectures, give feedback and adjust their pedagogical interventions in the classroom if needed. As future work, we plan to test this kind of environment in other contexts, for example to support teaching/learning situations located in different places.

ACKNOWLEDGEMENTS

The authors would thank IRIS (Institute of Robotics and Intelligent Systems, Ottawa, Canada) and CRSNG (Natural Sciences and Engineering Research Council of Canada) for their financial support, as well as teachers and students for their collaboration and participation.

REFERENCES

Capus L., Potvin B. and Tourigny N. (2003). An e-learning framework to extend lecture courses, *SSGRR 2003s, International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, e-Medicine, and Mobile Technologies on the Internet*, July 28 – August 3, L'Aquila, Italy (available on CD-Rom).

Capus L., Potvin B. and Tourigny N. (2002). A Web Environment based on Self-Explanation to Help Students Learn Solving Process in an Artificial Intelligence Course. *ICTE 2002, International Conference on Information and Communication Technologies in Education*, November 20-23, Badajoz, Spain, Educational Technology, Vol. I, A. Méndez Vilas, J.A. Mesa Gonzàlez, I. Solo de Zaldìvar M. (Eds.), 98-102.

Chi M.T.H. (2000). Self-Explaining Expository Texts: The Dual Processes of Generating Inferences and Repairing Mental Models, *Advances in Instructional Psychology*, Lawrence Erlbaum Associates, Mahwah, NJ, 161-238.

Chi M.T.H., Bassok M., Lewis M.W., Reimann P. and Glaser R. (1989). Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems, *Cognitive Science*, 13, 145-182.

Cronjé J.C. (2001). Metaphors and models in Internet-based learning. *Computers & Education*, 37. 241-256.

Jonassen, D. (1991). Thinking technology: Context is everything. *Educational Technology*, 31(6). 35-37.

Jonassen D., Peck K. & Wilson B. (1999). *Learning With Technology. A Constructivist Perspective*. Prentice-Hall, Upper Saddle River (N.J.).

Lankard Brown B. (1998) Using Problem-solving Approaches in Vocational Education, Practice Applications Brief, *ERIC/ACVE, Educational Resources Information Center:/Clearinghouse on Adult, Career, and Vocational Education*. Available from: http://www.ericacve.org/docs/probb-pab/htm

Schreiber G., Akkermans H., Anjewierden A., de Hoog R., Shadbolt N., Van de Velde W. and Wielinga B. (2000). *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press, ISBN 0262193000.

VanLehn K. (1998). Analogy Events: How Examples are used During Problem Solving, *Cognitive Science*, 22(3), 347-388.