Capus, L, Curvat, F., Leclair, O., & Tourigny, N. (2006). A Web environment to encourage students to do exercises outside the classroom: A case study. *Educational Technology & Society*, 9 (3), 173-181.

A Web environment to encourage students to do exercises outside the classroom: A case study

Laurence Capus, Frédéric Curvat, Olivier Leclair and Nicole Tourigny
ERICAE, Département d'informatique et de génie logiciel, Université Laval, Canada, G1K 7P4
laurence.capus@ift.ulaval.ca
nicole.tourigny@ift.ulaval.ca

ABSTRACT

For the past five years, our students have been passing less and less time preparing for lectures and exams. To encourage them to do more exercises, a pedagogical activity was offered outside the classroom. With the goal of making students more active during the problem-solving process, an innovative online environment, Sphinx, was developed. Sphinx proposes a set of exercises with their solutions, and invites students to explain them. Sphinx also gives students the opportunity to exchange ideas about an exercise and its solution, and gives the teacher the ability to observe if students take part in the problem-solving process. The originality of Sphinx lies in its ability to support students between lectures with a common learning strategy: learning-by-example. An experiment was conducted with 137 students during the 2003 fall session. The solved exercises were consulted most often before the two exams. However, only a few students participated actively in the experiment by intensively using the Sphinx environment; they obtained a better average score than the rest of the class. Thus, participation and collaboration had a positive effect on the students' marks. Tools should be added to motivate more students to take part in the explaining and collaborating process.

Keywords

E-learning, Self-explanation, Learning-by-example, Collaborative learning, Case study

Introduction

For the past five years, we have observed that our students generally have been passing less and less time preparing for lectures. Only a few have been reading theory and doing exercises. We also noted that they were not prepared for their exams. We decided to offer a pedagogical activity outside the classroom to encourage them to do more exercises.

Usually, the teachers on our campus use WebCT® or Learning Space® as online environments to support distance learning. Such environments exploit information and communication technologies to offer relevant tools: e-mail, discussion forum, chat, etc. However, they do not include specific tools for helping students learn the problem-solving process by solving exercises within these environments (Cronjé, 2001). For courses given in the classroom, teachers generally use Web sites with classic elements such as electronic slides, descriptions of assigned projects (or class projects), marks, exercises to be solved, etc. However, these elements are static. If students want to ask questions, they have to send a message to their teacher. This can be time consuming for the teacher and is not flexible for the students.

Since the 2002 fall session, we have proposed an online environment – more interactive than those usually employed on our campus – with examples of solved exercises that students are invited to self-explain (Capus et al., 2003). This environment has been improved in response to the various users' comments (students and teaching team). Our teaching method is original because we combine classroom instruction with an online environment to support learning outside the classroom. The purpose of the online environment was to make students more active during the problem-solving process and to motivate them to work more outside the classroom by giving them feedback within a virtual classroom between lecture sessions. It also allows the teacher to follow the progress of each student using the environment.

In this paper, we describe a case study conducted with 137 students during the 2003 fall session in a university course on artificial intelligence. Generally, the students consulted exercises, but they did not often explain them. The group of students who used the Sphinx environment most frequently obtained a better average score than that of the class as a whole. The objective of this paper is to show how the functional architecture of the Sphinx environment supports learning and teaching outside the classroom.

Next section describes how the Sphinx environment supports learning outside the classroom. This is followed by description of the experiment we conducted, as well as an analysis and discussion of the results obtained.

How to encourage students to do exercises outside the classroom?

To encourage students to do exercises outside the classroom, we developed Sphinx, an online environment that presents a set of exercises with their solutions, available at anytime on any platform. Sphinx allows students access to worked examples and derived problems and uses a collaborative model for learner advancement in problem-solving skills. It also allows the teacher to observe if students take part in solving exercises. This section describes the pedagogical principles on which Sphinx is based, the needs that Sphinx should help satisfy to support learning and teaching efficiently, the functionalities added to fulfil these needs, and finally the Sphinx's architecture.

The pedagogical principles of the Sphinx environment

The Sphinx environment combines learning-by-example and collaborative learning. Several authors, like VanLehn (1998), mentioned that study by means of examples is a common and natural way of learning. Sphinx also integrates self-explanation, because research has shown that the best problem-solvers self-explain more than other students (Chi, 2000; Chi et al., 1989). VanLehn and Jones (1993) conducted an experiment on the ability of students to solve problems after studying examples. Their hypothesis was that good problem-solvers are able to encode knowledge contained in an example and then explain implicit information related to this example; thus, they should obtain a better score when they solve new problems. Their results showed that the ones who learned the most had given the greatest number of explanations while studying examples.

To encourage students to self-explain a chosen problem in Sphinx, they are invited to write their explanations within the environment in order to formalize them more systematically. The collaboration allows students to learn with peers (Oliver and Omari, 2001). All students can look at the explanations and comment on them or can respond to another student's questions. The teacher waits for a moment before commenting or answering in order to let the collaboration process take place. Global feedback can also be given in the classroom so teachers can avoid having to intervene frequently to answer students' questions. This aspect is valuable because it is important that the use of the environment should not increase the teacher's tasks.

Students can find different kinds of examples in the Sphinx environment – questions/answers, problems with several solving steps, etc. – with different levels of difficulty. In fact, the teacher adds these examples according to their relevance to the course. Generally, the examples in the Sphinx environment are taken from an exercise book used in the course during previous terms. Table 1 presents an example from the Sphinx environment on resolution refutation proof, based on the "dead dog" problem (Luger, 2005). The solution is voluntarily a simple list of steps without explanations. The students have to reason through the problem in order to understand how the solution was obtained.

Table 1. An example from the Sphinx environment

Problem	With the following premises: P1: $\neg$dog(X) $\vee$ animal(X) P2: dog(fido) P3: $\neg$animal(Y) $\vee$ die(Y) Peter has to prove that Fido will die.
Solution	S1: $\neg$dog(X) $\vee$ animal(X) S2: dog(fido) S3: $\neg$animal(X) $\vee$ die(X) S4: $\neg$die(fido) S5: $\neg$animal(fido) S6: $\neg$dog(fido) S7: {} Peter proved that Fido will die.

Students can use self-explanation or explanation to make this reasoning. A self-explanation can contain an inference that refers to a piece of knowledge; in this case, the self-explanation specifies the application conditions of an action, the consequences of an action, the relation between an action and a goal or the relation to principles illustrated by the example (Chi et al., 1989; Khan and Yip, 1995). For the example contained in Table 1, a good explanation for step S4 can be: "To make a resolution refutation proof, add the negation of what is to be proved (Fido will die), in clause form, to the set of axioms.” In other words, the expected explanations correspond to the justifications for the reasoning used to solve the problem. A self-explanation can also be a paraphrase, a self-check or even be a textual fragment with no sense, but in all cases it is good for learning (Chi et al., 1989). Consequently, the most important goal is to motivate the students to explain and discuss solutions. A good explanation can then be obtained at the end of a discussion between users (the students and eventually the teaching team).

How does the Sphinx environment support learning?

A recent faculty study showed that students worked only about one hour per week per course instead of the six hours advised by teachers. We noted our students were more insecure about their exams and the class average had decreased. We needed a way of encouraging our students to work more outside the classroom. The Sphinx environment is a tool for attracting the students into a learning community, and then creating enthusiasm among the students. This environment also allows the teacher to observe the work of each student logged on, and to get some information about her/his difficulties. However, the development of an educational environment is not easy. It is rather a complex task because of the required knowledge from various fields: user interface engineering, cognitive sciences, educational sciences, and knowledge engineering (Tchounikine, 2002). After a preliminary experiment in 2002 (Capus et al., 2003), we collected users’ comments to improve the online environment. Then, we identified special needs and added particular functionalities to fulfil these needs in order to support the learning process in the Sphinx environment.

Needs to be considered

As for any software, the user interface should be appropriate to the users' needs. It is doubly important in an educational context where the main objective is to help the student acquire new knowledge. Thus, the interface should not hinder this knowledge process by discouraging the user with its ineffectiveness or problems of coherency. The graphic interface should be ergonomic and should provide all the information necessary to the user, who should be able to personalize it. A user model must be provided to save user preferences about media types, colour parameters and the position of the interface. It is also important to provide simple contextual help that is available when needed.

More particularly, several means of facilitating learning can be proposed based on the principle of help in real time. We suppose that help given to the learner during the use of the environment effectively encourages self-explanation. For each category of explanation, it is important to give the user effective and adapted advice. Then, adapted semantics can be employed to allow the student to clarify her/his thought. To help the learner structure her/his thought, one can highlight special argumentative keywords contained in her/his explanation. This emphasis on the logic structure of the sentence should improve the re-reading. A way of helping the learner to construct her/his explanation is to begin with a description of the action implemented in the step of the problem-solving process to be explained. The action is characterized by an action verb, a complement and parameters. A list of words from an educational dictionary (Legendre, 1993) can be proposed for each part of the action. An immediate automatic retroaction is then possible if the right response is stored in the environment. It is important to construct a learner model (Baker, 2000) in order to advise the student on the examples to be studied and the concepts to be revised. The profile of the learner should integrate her/his knowledge about the domain. To build such a model, one can use various methods: based on the course, based on results, or by self-assessment (Delestre, 2000).

Functionalities to support the learning process

We added some particular functionalities to the online environment in order to meet the above needs. These functionalities should more efficiently support the learning process.

Different navigation modes

Students can navigate among the examples, as they prefer. Examples are classified according to the structure of the course, which is represented by a tree in the user interface. Each week of the course (or leaf of the tree) proposes a list of examples. Our objective is to propose another way of navigating to help students that are dependent on field. This is an important tool (Triantafillou et al., 2003) for advising them during the revision phase. A list of keywords, not visible to users, is combined with each example. These keywords are used to find examples related to predefined subjects. To guide the student, the example links are coloured. We defined three categories of links, e.g. three colours: for the ‘not visited links’, ‘visited links’ and ‘links to be visited’ (belonging to the set of suggested links). This principle is simple, and similar to the one used on all Web sites. Thus, students can navigate freely or by using a more personalized revision mode.

Information about messages

We based this functionality on the principles of discussion forums. For each example, the user interface indicates the number of messages posted. This functionality allows the student to quickly see if comments were made on her/his explanation or if new messages were added since her/his last logon.

Modifications of individual preferences

The student can also modify her/his individual preferences concerning nickname, password, email, colours and links, etc. This functionality allows the student to personalize her/his own interface making the user interface become more familiar.

Saving users’ actions

Users’ actions are saved in a special folder in text files. These files have a specific format, which is easy to translate by parsing. Each line of the files is a record composed of a series of values corresponding to a relevant event in a user’s action. Each record contains the event type, for instance ‘write an explanation’, ‘read an explanation’, or ‘display an example’. If the event is ‘write explanation’, the time to write an explanation and the number of significant words in the written explanation are added to the record. For each event type, other information is saved such as the number of positive (e.g. ‘I agree with this explanation’) and negative (e.g. ‘I disagree with this explanation’) students’ messages as well as the number of positive and negative teacher’s messages. The list of keywords for the example consulted and its difficulty level, as well as the number of words in the written explanation are also added to the record. All this information allows the environment to update the learner model.

Learner model

The learner model indicates the knowledge level of each student and is quite simple. A network of relevant concepts studied during the course describes this model. The concepts can take the following values: ‘not seen’, ‘seen’, ‘known’, ‘learned’. At the beginning of the session, each concept has the value ‘not seen’. According to the students’ actions, the learner’s model is updated by modifying the value of each concept implied by the actions. In fact, the information gathered by the preceding functionality allows the environment to modify the new value of a concept. Assessment of learner’ knowledge using only this information is not accurate because other elements are needed. For instance, students can learn concepts in the classroom or with their handbook. For the time being, the learner model does not take this outside learning into account. However, we have found a way of making it work for our context. Sphinx uses a reference table specifying the number of points for each of the user’s actions according to its importance and its *a priori* impact on learning. For instance, if a student explains an example, the number of points is greater than if this example is only read. The rationale behind this approach is that explaining an example is more efficient than only reading it to understand a concept. The revision navigation mode uses the learner model to propose links to relevant examples. Indeed, the environment suggests to students that they revise specific examples that integrate concepts not sufficiently understood. It automatically gives messages to students advising them on what they should do. The creation of these messages is also based on the learner model.

Other functionalities, such as browse bars allowing forward and backward movements within the environment improve the user interface. The content of an example can be printed. The home page of the Sphinx environment displays information about its utility and its use. A help menu is available to introduce the environment and give details on its use. A contextual help offers the student advice and tips.

Architecture of the Sphinx environment

Figure 1 presents the architecture of the Sphinx environment in a simplified way. The student enters the environment by using the Student Module. This is a Web interface that allows the student to have access to the examples of solved exercises, and to all explanations and discussions. The student can modify her/his preferences for this interface. The Feedback Module is activated when the student chooses words for describing the action made in a step of the problem-solving process. It also allows the student to know if her/his response was right or wrong.

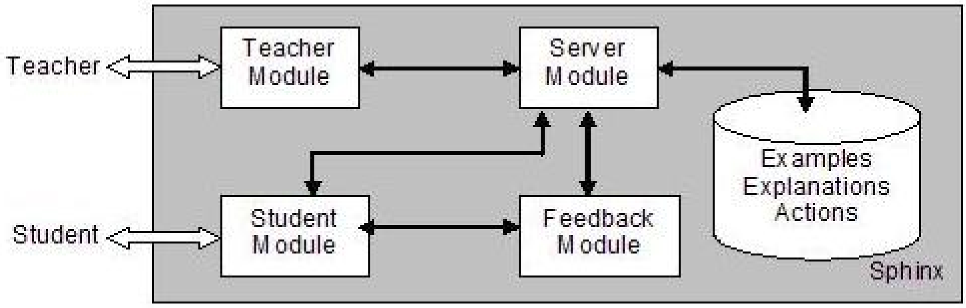

Figure 1. A simplified architecture of the proposed environment

The Server Module is the central point of the environment. It manages access to and the saving of data/knowledge (examples of solved exercises, users' explanations and actions, and user preferences). This is the link between the different modules: Teacher Module, Student Module, and Feedback Module. The teacher has access to the environment by using the Teacher Module, which is an acquisition module for adding examples of solved exercises. This aspect is often forgotten in educational environments, but it is very important for facilitating the use by teachers, because it eliminates the need for the teacher to adapt to the programming language used to build the system. The teacher can also have access to the environment by using the Student Module and can interact with all the students by observing explanations and discussions, or by adding new explanations or comments.

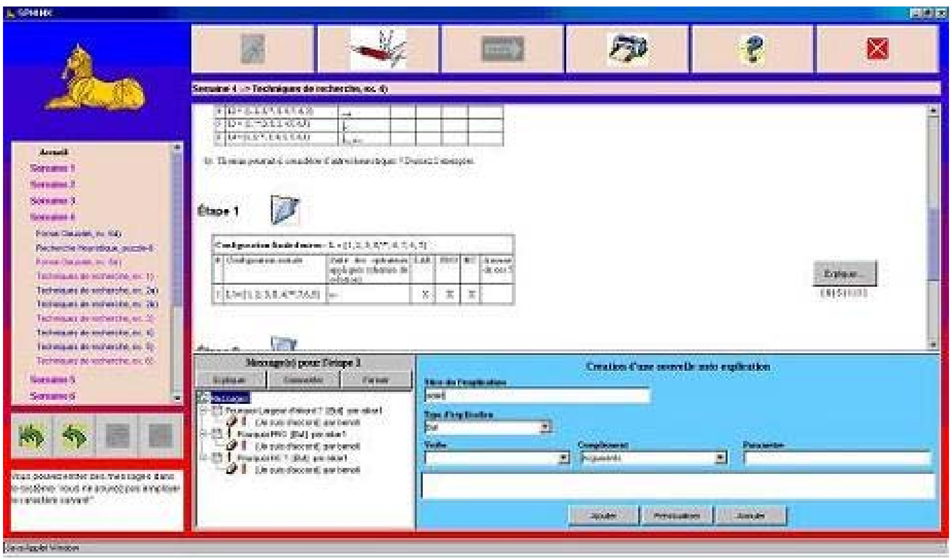

Figure 2. A snapshot of the user interface

The online environment, Sphinx, performs on any platform; both Windows and Linux are available for our students. It does not limit the number of connections. The interface is developed in the Java language, mySQL was chosen as a database system, and data and knowledge are represented in an XML format. All users have access to the environment by any Web browser that supports the Java Servlet technology. To enter the environment, they have to validate their login and password. Next, they can use the general user interface. Other particular interfaces are proposed to observe examples of solved exercises, to write messages, change preferences or consult the help menu. The database system allows access to users' messages and user data (preferences, history, etc.). The users' actions are saved in text files in a special folder, the track folder. This function is important because these files allow us to evaluate how the students used the environment. To conclude this section, Figure 2 presents a snapshot of the user interface.

The left-hand column proposes the list of examples for each week. The browser arrows and a help message are shown below this. Tools like printing, modifying personal features, help, etc., are at the top of this snapshot. The main window shows an example with details on each step of the solving process. For each step, an 'explain' button (at right of each solving step) allows the user to post a message by using the grey shaded window at the bottom right of the window. The numbers below the 'explain' button indicate the number of messages: total, not read, new, replied. All the discussions about this step can be seen at the left of the blue shaded window, structured as a discussion forum.

Experiment, results and discussion

During the entire 2003 fall session, this environment was made available as additional pedagogical material to 137 students in our department's course on artificial intelligence. Most of the students were in their second year of university studies, and this course is compulsory for their bachelor program. The term is fifteen weeks long, with two exams, one at the middle of the term and the other at the end of the term, usually in the fifteenth week. The students had to use a login and password, thus, only the students enrolled in this course could log onto the environment. Each week, the teacher added between five and ten examples corresponding to the content seen in classroom; the examples were listed according to their difficulty level. Based on the text files saved in the track folder, we evaluated the participation and collaboration of the students. We also compared the students' marks in order to find a correlation between the use of the environment and the results obtained. Finally, we questioned the students on the use of the new functionalities such as different navigation modes, information about messages and modifications of individual preferences. The purpose of our evaluation was to determine if the Sphinx environment was effective in increasing students' marks and if its functionalities were useful or not.

Participation

We recorded 506 student logons, an average of 3.7 per student. In fact, only 113 students out of the 137 used Sphinx. If we consider only the students who actively participated in the experiment, the average number of logons per student is 4.47. The average logon time is about one hour. We noted that most of the students who entered the environment did not add any explanation or message. However, they consulted solved exercises, explanations and messages of the other students. Why did the students participate so little in the discussions? We have no sure answers at the present time. One might hypothesize that they did not know how to explain the solutions to problems or they did not want to participate in discussions because they were afraid of making mistakes. Further investigation is needed to clarify this point. However, it seems that the students found it useful to consult the information available in the Sphinx environment.

We also noted that the Sphinx environment was not used regularly during the session. Figure 3 shows the number of logons per week. Most of students logged onto the environment during exam week (Weeks #7 and #14), so we can suppose that the Sphinx environment was useful for students as a revision tool before each exam.

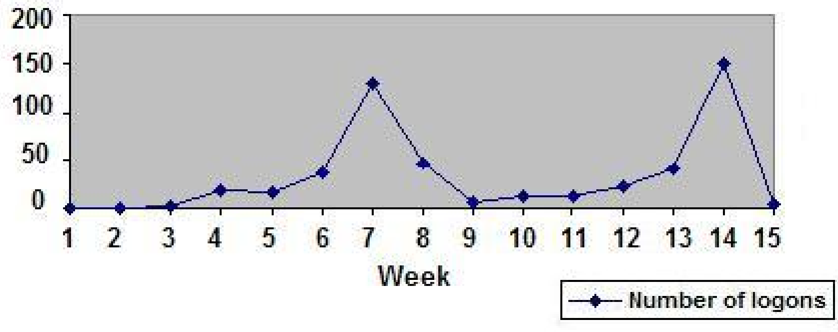

Figure 3. Number of logons per week

Collaboration

Table 2 corresponds to three extracts of discussions (translated into English) posted by students during this experiment. Discussion #1 is about the best path found in a search tree by using the alpha-beta technique. The student Jflal7 explained why the node R was chosen. This was a good explanation and another student agreed with it. The students' messages were not only explanations. Sometimes, students asked questions on the solved exercises because they did not understand them. For instance, in discussion #2, the student MaTur2 asked a question about a natural language processing program that builds sentences with a set of words and grammar rules. This program had no semantic verification process. Then, MaTur2 asked why a certain sentence was not given as a solution. During this discussion, another student Gucot59 explained to MaTur2 that the solution proposed in the exercise was not complete and it missed twelve other sentences. From the teacher viewpoint, collaboration is very important because students can exchange ideas and learn from each other. This discussion illustrates how collaboration can be exploited in a learning context.

Table 2. Three extracts of discussions between students

Discussion	*Title of message*	*Nickname*	*Content of message*
#1	Reason	Jflal7	The value R is 2 and the one of Q is 1. R is chosen because one must choose the maximum.
	Correct!	Stleg19	It's correct!
#2	And the cupboard?	MaTur2	Should this sentence be here: A cupboard builds a joiner?
	Partial Solution	Gucot59	I think this program produces 12 other different sentences.
#3	A lot of negatives!	Jofou33	Is it possible to delete the negatives in the sub clauses?

In discussion #3, the student Jofou33 wondered why many negatives were used in an exercise on knowledge representation using conceptual graphs. The other students did not give any response. In this case, there was no collaboration or one can suppose that the other students did not know how to answer the question. Because we think it is important to give correct feedback to our students, a teaching assistant posted a response (not given in the table) to explain the representation.

Students' marks

We compared the marks of the whole class to those of the students who used the Sphinx environment most frequently. We think that to be useful in the learning process, the student should actively use Sphinx. If a student only enters Sphinx and prints some examples, s/he is not getting any more than s/he would get from a book, for instance. Thus, we considered only the students who added messages, logged onto the environment for more than four hours and participated in several discussions. When we analysed the logon time, this group of students clearly stood out from the others. Their average score was 70.84% whereas the average score of the class was 67.44%. One might think that the best students used the Sphinx environment most frequently. However, upon looking at the students' general marks, we noted that this hypothesis was wrong, thus indicating that Sphinx may be having a positive effect on the students' marks. This increase is an encouraging result and we will continue improving the Sphinx environment to help students learn and succeed.

Users' comments

At the end of the session, the students filled out a questionnaire. The analysis of the questionnaires showed that 63% of the students used the new functionalities (different navigation modes, information about messages and modifications of individual preferences), 61% wrote that Sphinx had helped them and 70% appreciated the possibility of revising concepts with which they had difficulties. We can conclude that the Sphinx environment is useful to students especially because of the different modes of navigation. The degree of satisfaction concerning the messages is lower. Only 48% appreciated this functionality. The students have different learning styles. Some prefer to be directed, others do not. For the moment, these messages are quite basic and need to be more precise and personalized, and perhaps should even use a pedagogical agent. These two facts could explain the low degree of satisfaction.

The teaching team thinks that the new functionalities of the Sphinx environment improve the learning context and should help students use this environment more easily and more often. According to the team, the messages could be improved by explaining why they are given. It would help the students to better understand what is expected from them. The teaching team considers that encouraging messages should be given even when participation is satisfactory. Finally, the team would like to have another functionality with which to propose specific advice, to underline the importance of an example, for instance.

Benefits of the Sphinx environment

Through using the Sphinx environment we learned that participation and collaboration can have a positive effect on the students' marks. If we consider only the students who really participated in the experiment by adding several messages, logging onto the environment for more than four hours and taking part in several discussions, we can conclude that the use of the Sphinx environment improved the learning process. Moreover, the results on learning-by-example showed the importance of the explaining process. Thus, in the next phase in the development of Sphinx we will add elements that hopefully will encourage more students to take part in the processes of explaining and collaborating.

Future work

We are planning a new experiment to measure improvement in each individual's learning. In the present version of Sphinx, at the beginning of the session, each concept in the learner model has the value 'not seen'. According to her/his actions, her/his student's model is updated by modifying the values of the concepts implied by the actions. At the end of the session, each learner model can be consulted in order to evaluate the knowledge level of concepts. However, such an evaluation is not really exact because our students can use pedagogical material other than the Sphinx environment to learn. If a student does not explain specific examples, this does not always mean that s/he does not know the concept illustrated by these examples. S/he can have learned it in classroom, by doing exercises, or by some other means. The improvement in learning can be verifiable from a global viewpoint but it is more difficult to evaluate on an individual basis, because the Sphinx environment is only an online educational support for classroom instruction. Students can decide to use or not use it. Above all, we want to encourage students to work more outside the classroom. We are now working on a better representation of the learner model, which would allow us to measure individual learning improvement more efficiently. We would also like to adapt the Sphinx environment to distance learning (Tourigny et al., 2005), an instruction mode that our university is becoming more and more interested in.

Conclusion

This paper presented a case study on an online environment, Sphinx, available on a continuous basis to 137 students during the 2003 fall session. This environment is a tool for helping students outside the classroom and for encouraging them to do exercises. A set of solved exercises was proposed each week during the course. A student could consult an exercise and its solution. She/he could try to do the exercise, to self-explain a step of the solving process for her/himself or add an explanation in the environment. She/he could share her/his explanation with the other students or comment on an explanation of another student. Or she/he could simply read the discussion of an example. Sphinx is based on self-explanation (Chi, 2000; Chi et al., 1989) and collaboration (Oliver and Omari, 2001), two activities that allow students to build and/or revise their domain knowledge and become more active when they learn. We noted that the students who used the Sphinx environment most frequently, obtained a better average score. Even if it is difficult to interpret this result, it encourages us to continue using the Sphinx environment and we can show this result to our future students in order to encourage them to use Sphinx more frequently. We are improving our methodology for studying individual learning effectiveness and are planning a bigger experiment to evaluate how and why students learned with the Sphinx environment.

The analysis of this experiment also allowed us to identify some aspects of the environment that should be improved. Particular attention will be paid to ways of motivating the students to take part in the explaining and collaborating process. We continue to improve the present version of the Sphinx environment and our final goal is to build a shell adaptable to any course and any kind of training. Teachers should be able to easily store examples of their course and students should be able to consult examples at anytime, explain them, comment on explanations of peers, and receive feedback within a virtual classroom between lecture sessions.

Acknowledgements

The authors would like to thank the Institute for Robotics and Intelligent Systems (IRIS-Precarn Inc.) for its financial support. The authors would also like to thank the students, Jalil Emmanuel, Maxime Garceau-Brodeur, Huu Loc Nguyen and Benoît Potvin, for their participation in the development of the Sphinx environment as well as the students registered in the course IFT-17586 Artificial Intelligence I for their collaboration. Finally, the authors would like to thank the two anonymous reviewers for their constructive comments.

References

Baker, M. (2000). The roles of models in Artificial Intelligence and Education research: a prospective view. *International Journal of Artificial Intelligence in Education, 11*, 122-143.

Capus, L., Potvin, B., & Tourigny, N. (2003). An e-learning framework to extend lecture courses. *Paper presented at the 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.

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

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & 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* (3-4), 241-256.

Delestre, N. (2000). La construction automatique de cours hypermédia adaptés à l'apprenant par agencement de briques élémentaires, In Bourigault, D., Charlet, J., Kassel, G. & Zacklad M. (Eds.), *Ingénierie des connaissances 2000*, Paris: Eyrolles, 35-46.

Khan, T., & Yip, Y. J. (1995). CBTII - Case-Based computer-aided instruction: survey of principles, applications and issues. *The Knowledge Engineering Review, 10* (3), 235-268.

Legendre, R. (1993). *Dictionnaire actuel de l'éducation*, Second Edition, Montréal: Guérin.

Luger, G. F. (2005). *Artificial Intelligence, Structures and Strategies for Complex Problem Solving* (5th Ed.), New York: Addison Wesley.

Oliver, R., & Omari, A. (2001). Exploring Student Responses to Collaborating and Learning in a Web-Based Environment. *Journal of Computer Assisted Learning, 17* (1), 34-47.

Tchounikine, P. (2002). Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain. *Revue I3, 2* (1), 59-95.

Tourigny, N., Capus, L., & Gamache, A. (2005). Using Self-Explanations with Information and Communication Technologies to Improve Distance Learning in Computer Science. In Tiginyanu, I. M. & Sontea, V. (Eds.), *Proceedings of the 4th International Conference on Microelectronics and Computer Science*, Chisinau, Moldova, September 15-17, Vol. II, 461-464.

Triantafillou, E., Pomportsis, A., & Demetriadis, S. (2003). The design and the formative evaluation of an adaptive educational system based on cognitive styles. *Computers & Education, 41* (1), 87-103.

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

Van Lehn, K., & Jones, R.M. (1993). Learning by explaining examples to oneself: A computational model. In Chipman & Meyrowitz (Eds.), *Foundations of Knowledge Acquisition: Cognitive models of complex learning*, Boston, MA: Kluwer, 25-82.