Miao, Y. , Holst, S.L., Haake, J.M., & Steinmetz, R. (2000). PBL-Protocols: Guiding and Controlling Problem Based Learning Processes in Vitual Learning Environments. In B. Fishman & S. O'Connor-Divelbiss (Eds.), *Fourth International Conference of the Learning Sciences* (pp. 232-237). Mahwah, NJ: Erlbaum.

PBL-protocols: Guiding and Controlling Problem Based Learning Processes in Virtual Learning Environments

Yongwu Miao, Shirley J. Holst, Joerg M. Haake, Ralf Steinmetz
GMD - German National Research Center for Information Technology,
IPSI – Integrated Publication and Information Systems Institute
Dolivostrasse 15, D-64293, Darmstadt, Germany,
Tel: +49-6151-869 918,
E-mail: {miao, holst, haake, steinmetz}@darmstadt.gmd.de

Abstract In this paper we describe our approach for supporting the use of the problem-based learning (PBL) method in virtual learning environments. We identify two key difficulties. Firstly, neither learners nor tutors, who are used to more traditional methods of teaching, know exactly what to expect from PBL or how to behave appropriately within their new roles. Secondly, these problems are compounded by having to interact within a socially unfamiliar computer-based learning environment. We frame these difficulties in terms of *schema theory* which suggests that people behave and know what to expect from situations according to mentally represented *scripts* describing the typical course of events. We apply computational descriptions of these scripts, i.e. *PBL-protocols*, which guide and control social interaction in virtual PBL groups according to (a) the phase of the learning process and (b) the specific roles of tutor, learner and expert; three roles which are typically distinguished in the problem-based learning method. The PBL-protocols restrict the type of contributions allowed for each role type for each phase of the PBL learning process. We describe the implementation of the PBL-protocols in the CROCODILE virtual learning environment, and discuss results and implications of an early pilot study.

Keywords distributed learning environments, scaffolding

Introduction

Problem-based learning is an increasingly popular instructional method requiring learners to actively gather and apply knowledge in order to solve ill-structured real-world problems. Contrary to traditional instructional methods, where the teacher organises and imparts information to the students, problem-based learning is guided by tutors who take a facilitative role, encouraging students to engage in active and meaningful learning. The principles embodied in PBL make it a learning method that promotes a number of factors known to improve learning (Woods, 1994). Namely, active engagement in the learning material, cooperation among self-directed students, prompt feedback, tailoring of the learning process to fit student's learning preferences, and involvement of students in self- and peer- assessment.

However, inspite of these advantages, some barriers have to be overcome in order to implement PBL successfully. Teachers and learners who are unfamiliar with PBL tend to be reluctant to change their traditional roles (Jones, Valdez, Norakowski & Rasmussen, 1994; Bridges, 1992). Teachers used to teaching through lectures and discussions lack the skills of a facilitator in guiding learners to discover information for themselves. As a facilitator they can give hints, provide resources and ask searching questions, but they must withhold information that they would previously have simply given to the students. Learners are also slow to adjust to the PBL method, and to the change in their role from passively receiving information to actively engaging in a problem-solving process. Additional problems arise when the PBL method is applied in virtual learning environments where participants are distributed and weak communication channels make group interactions difficult. It is hard to make and keep track of progress towards learning goals efficiently.

In order to support problem-based learning in virtual learning environments, we argue that computational mechanisms can be designed to help in overcoming these difficulties. In this paper, we present our idea to guide and control problem-based learning processes in virtual learning environments by using computational descriptions of collaborative problem-based learning strategies (i.e. *PBL-protocols*). Firstly, we present the concept behind *learning protocols*. Then we apply the concept of learning protocols in support of collaborative problem-based learning. We then describe how the resulting PBL-protocols are implemented and tested in our prototype system.

Theory and Background Concepts

We frame the problems that arise when the PBL method is applied in virtual learning environments in terms of *schema theory* (Schank & Abelson, 1977). According to schema theory, people know how to behave and what to expect in particular situations, because they use *scripts*. Scripts are mental structures representing the person's knowledge about objects, people, or situations. They are derived from prior knowledge and experience, and set up expectations about what is probable and appropriate in relation to particular social contexts. The most commonly cited example is the restaurant script, in which the process of eating at a restaurant is divided into 'scenes' (e.g. entering, ordering, eating and leaving a restaurant). *Role scripts* embody knowledge about how people in a particular role (e.g. waiter, or customer) are expected to behave in each scene. We argue that users of PBL initially have inappropriate scripts regarding how to behave and what to expect from others as they do PBL in the virtual learning environment. We have designed a computational mechanism which forces the teacher and learners to behave within the constraints of the their new roles in the PBL process. We call this mechanism a *PBL-protocol,* which is a special type of *learning protocol.*

The concept of a learning protocol has been established previously (Pfister, Wessner, Beck-Wilson, Miao & Steinmetz, 1998; Wessner, Pfister & Miao, 1999). A learning protocol is a script written in computational form. It represents how learners, and also tutors, are expected to behave during the learning process. More than just a representation, the learning protocol actually forces the learners and tutors to behave appropriately by restricting which behaviours are allowed. Like the restaurant script, the learning protocol divides the learning process into a number of distinct phases (or 'scenes') over time. These phases may be sequentially ordered or networked, and learners can progress through some or all of the phases as they complete their task. Each participant of the learning process will be categorised into specific *protocol roles* when they begin to use an instance of the learning protocol, e.g. 'learner' or 'tutor' and they can contribute to the construction of shared knowledge according their prescribed role. Associated with the phase of the learning protocol, is a set of *behavior rules* which specify who is permitted to perform which operation on which type of object within that phase. Once a phase of the learning protocol is completed, the participants can decide to which phase they now want to transfer. They must choose from those phases that are successor phases of the current phase in the learning protocol.

PBL-protocols

We have applied the notion of learning protocols to problem-based learning. Within the literature on problem-based learning it is clear that the learning process is well structured and can be divided into a number of distinct phases (URL; Stepien, Gallagher & Workman, 1993). This makes problem-based learning an ideal application domain for learning protocols. The literature on PBL describes a varying number of phases and sequencing for PBL depending on the conditions in which it is carried out. We take a typical example, consisting of the six phases illustrated in figure 1. Notably, different learning protocols for PBL may also be desirable depending on the size and structure of the learning group or according to the knowledge, skills, interests and learning styles of the individual members. These factors will lead to alternative strategies being adopted to perform problem-based learning. Hence, a family of learning protocols can be defined for PBL.

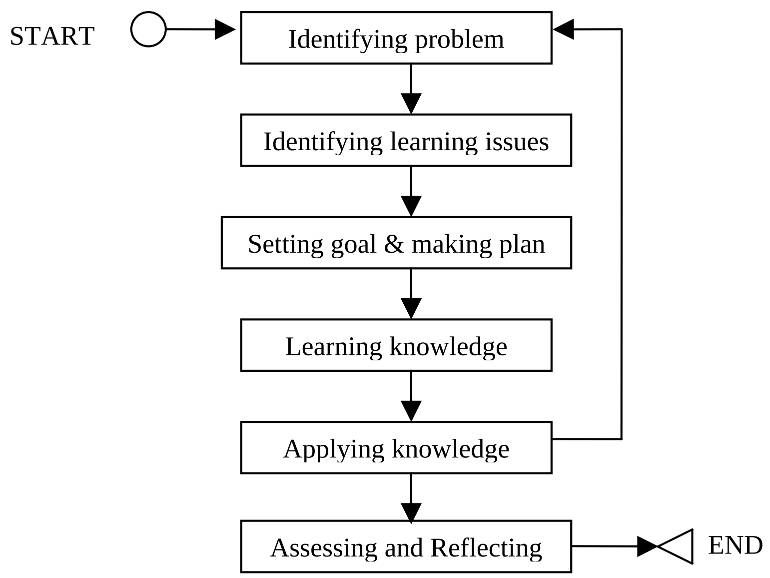

Figure 1. An example of a PBL-protocol; arrows connect PBL phases

We design our PBL-protocols specifically to support problem-based learning in a virtual learning environment where learners collaborate together to build a shared hyperdocument to represent their shared knowledge. The hyperdocument consists of a set of connected pages. The PBL-net is the page at the topmost level and serves as the entry point for the whole hyperdocument. The PBL-net consists of a set of nodes and links. Each node of the PBL-net serves as a hyperlink to another page of the hyperdocument. Each of these pages can in turn contain further untyped nodes that serve as hyperlinks, together with information in the form of text, graphics, and images. The nodes and links of the PBL-net itself are all typed and labelled according to the type of information to be represented (e.g. problem, learning issue, hypothesis, and solution). This will vary according to the phase of the learning process, since the information which learners must deal with changes as they progress through the task. For more details of how the hyperdocument is constructed, see (Miao, Holst, Holmer, Fleschutz & Zentel, 2000).

We can establish three basic roles in problem-based learning scenarios – tutor, learner and expert. The PBL-protocol constrains behaviour for each of these roles by determining which types of nodes and links can be manipulated by each role. Learners can manipulate nodes and links which represent the problem itself (i.e. nodes such as: problem, learning issue, resource, principle, evidence, hypothesis, solution; and links such as: is_a_sub_problem_of, is_a_sub_issue_of, is_a_prerequisite_for, is_contrary_to, derive_from). Tutors and experts, on the other hand, cannot always directly engage with all aspects of the construction of the shared knowledge. Rather they take the specialised role of a facilitator by asking questions, giving hints, providing comments and recommending resources. The PBL-protocol prevents the tutor and expert from adding directly to certains types of information content belonging to the shared hyperdocument. For example, they are not able to create problem, learning issue, hypothesis or solution nodes but are only allowed to create nodes and links which make indirect (facilitative) contributions to the shared representation. The actual node types that can be created by tutors and experts are: comment, hint, question, resource, source. The link types that can be created are, for example, comment_on.

During the first phase of PBL, 'identifying the problem', the tutor introduces the problem, perhaps with the help of a tutor or expert. Both tutor and expert can contribute to the shared representation by creating 'source' nodes. Only the student can actually define the problem using 'problem' nodes and 'is_a_sub_of' links in order to show how the problem decomposes into sub problems. The result is a network of nodes at the top-level of the hyperdocument that carry content pages. During this phase, the tutor can create 'hint' and 'question' nodes, giving indirect help in how to define the problem. The students can respond to the tutor using 'answer' nodes, use 'comment' nodes to comment on the contributions of others, and can ask their own questions. When the tutor decides that the problem has been identified satisfactorily, then he /she takes the learning group to the next phase 'identifying learning issues'. Here, learners can extend the hyperdocument by identifying 'learning issue' nodes and indicating their relationship to the problem nodes by means of appropriate links. The learners can also express the extent to which they have knowledge about each learning issue node, and on the basis of this information they can decide what information needs to be collected in the next 'learning phase'. The tutor can once more only indirectly contribute to the shared representation.

In the remaining 4 phases, the PBL-protocol will continue to guide and control which nodes and links can operated on by whom. The learning group decides at which point they want to transfer to the next phase, and the shared hyperdocument as it stands so far will be used as the basis on which to establish learning goals, and a learning plan. Notably, learners are able, if desired, to return to earlier phases of the PBL-protocol if the need arises. For example, at the phase of 'applying knowledge' the learners may not be satisfied with their proposed hypotheses or solutions. At this point they may identify a gap in their understanding and return to define new learning issues, or even to redefine the problem itself.

Implementation and Pilot Study

We have implemented the PBL-protocols in a prototype system, called CROCODILE. The concept behind CROCODILE has been described in previously (Pfister, Wessner, Beck-Wilson, Miao & Steinmetz, 1998; Wessner, Pfister & Miao, 1999; Miao, Fleschutz & Zentel, 1999). In figure 2, we illustrate the three windows that the learners see. The lower window is the PBL hyperdocument editor in which learners can build their shared representation. The upper left background window shows the virtual room in which learners are located in the CROCODILE learning environment. The upper right hand window is the protocol control panel.

A user can initiate a learning protocol instance by clicking on the "protocol" icon (see figure 2) in the hyperdocument editor window. A dialog window will pop up and the user can inititate a learning protocol by selecting from a list of pre-defined PBL-protocols. Currently, we have prepared three alternative PBL-protocols for problem based learning which are tailored to fit various descriptions of PBL which are found in the literature. After choosing a PBL-protocol, the protocol control panel appears. Information about the initiated protocol instance, such as the name of selected PBL-protocol, the name of the current phase, and the assigned role membership, is presented. Roles are also assigned within the protocol control panel.

To make a contribution to the shared hyperdocument, the user can create nodes by dragging the 'create nodes' icon with the mouse (see figure 2) and dropping it in an appropriate place. If more than one node type is allowed for their role, then a selection box appears, from which the user chooses from the node types available for their role. The user then types in the statement that describes the content of the node. Links are created by a draw-line gesture going from the start to the destination node. A selection box once more allows them to choose the link type. To move between phases of the PBL-protocol, the user can click on the 'next' button within the protocol control panel. If there is only one possible succeeding phase, then the switch will be made automatically. If there are two or more branches within the protocol, then the user will be presented with a choice in the form of a dialogue. The user can select one alternative, at which point the selected phase will be executed.

We carried out an initial internal pilot study with 5 users. We used the problem-based learning approach to tackle a research topic within our group. Four people within our group acted as learners and one took the role of the tutor. The role-appropriate restrictions described above did support the users to pay attention to different types of information at appropriate stages of the PBL process. However, there was a notable lack in the system of any explanation as to why restrictions on node and link types were being made. We therefore propose to develop a help system that explains to learners and tutors what types of information they should be representing in the shared hyperdocument at each phase of PBL. With this understanding, and with repeated use of the PBL-protocol, learners, tutors and experts should come to know what to expect and how they should behave during each phase. The protocol will then have supported users to create their own internal scripts for PBL behaviour.

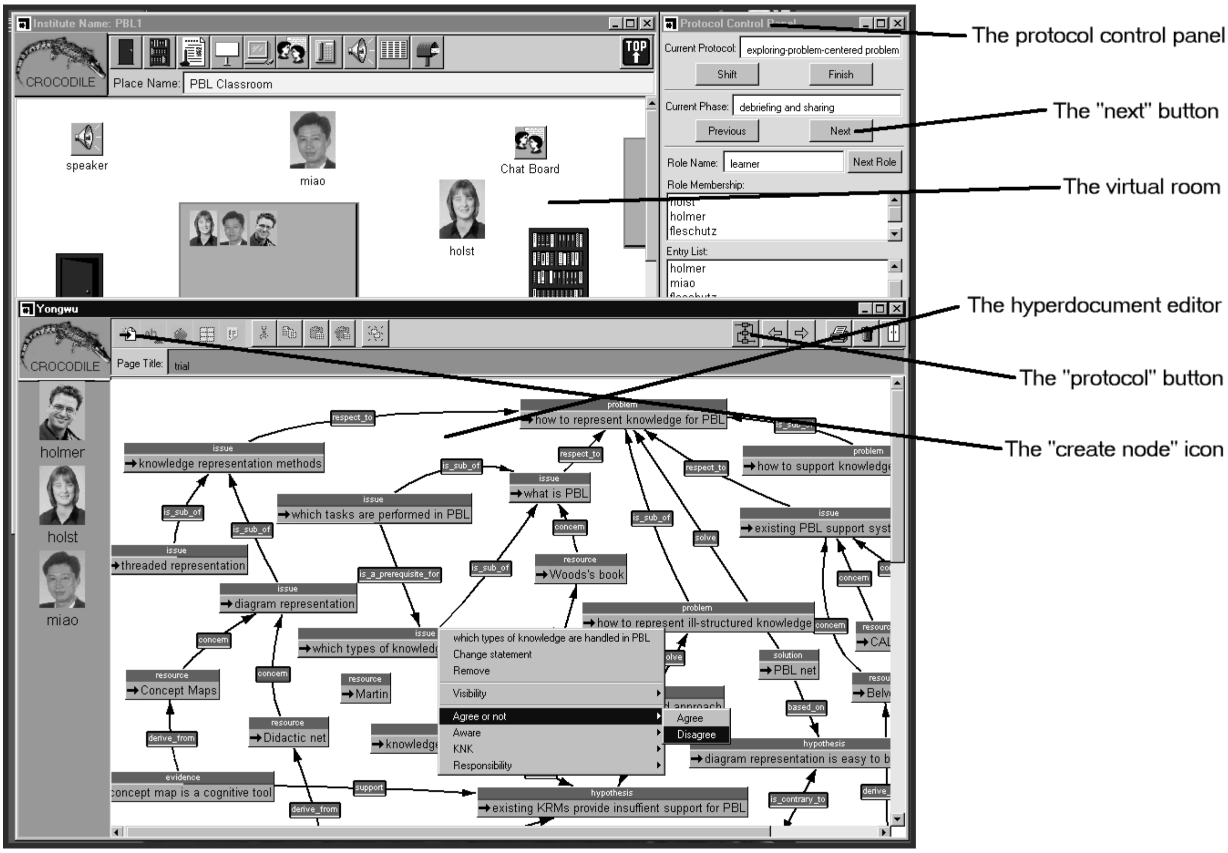

Figure 2. PBL-protocol control panel, virtual room and hyperdocument editor of CROCODILE.

Summary and Discussion

In this paper, we described the theoretical background for the use of learning protocols to support collaborative learning processes. We developed PBL-protocols to help to overcome the difficulties encountered by users of the PBL method within virtual learning environments. The resulting PBL-protocols restrict behaviour to fit within pre-defined roles, and to guide users from one stage of the learning process to the next. We have begun to implement PBL-protocols in the CROCODILE environment, and have run an initial pilot study.

In terms of related work, other systems that support PBL do not deal with the question of how to support users to behave within their predefined role types through the different phases of the learning process. The Collaboratory Notebook (O'Neill, 1994; Edelson, O'Neill, Gomez & D'Amico, 1995), Web-SMILE (Guzdial, Hmelo, Hübscher, Nagel, Newstetter, Puntembakar, Shabo, Turns, & Kolodner, 1997), CALE (Mahling, Sorrows & Skogseid, 1995), CSILE (Scardamalia, Bereiter & Lamon, 1994) and Belvedere (Suthers, Toth & Weiner, 1997) are each systems which store the contributions of all users, whether teachers or learners, in a shared database. Any user can contribute to the database at any time, and can retrieve the contributions of others at any time in order to read them. These systems did classify the contributions of users into different types. Only Belvedere and Web-SMILE distinguish between the different phases of the learning process. They also provide guidance for users to perform the focal task in each phase. However, unlike CROCODILE, they do not support the whole group to move through phases of their task together. Rather, the users of these systems can work in different phases of the task at the same time. In contrast, CROCODILE provides explicit support for the whole group to move through the task in a synchronised way. Finally, unlike CROCODILE, none of the above systems restrict user actions according to the members' respective roles as learner, tutor or expert.

In the future we plan to evaluate the use of CROCODILE to support problem-based learning by comparing the performance of problem-based learning groups with and without the support of the PBL-protocols. We will also explore how to tailor the PBL-protocols according both to the characteristics of different learning groups and the nature of problems to be solved. We plan to implement the help system as described above.

References

Bridges, E. M. (1992). Problem based learning for administrators. Eugene, OR: ERIC Clearinghouse on Educational Management. (ERIC Document Reproduction Service No. ED 347 617)

Edelson, D., O'Neill, K., Gomez, L., D'Amico, L. (1995). A design for effective support fo inquiry and collaboration. In: *Proceedings of CSCL'95* pp. 107-111. Mahwah, MJ: Lawrence Erlbaum.

Guzdial, M., Hmelo, C, Hübscher, R., Nagel, K., Newstetter, W., Puntembakar, S., Shabo, A., Turns, J., and Kolodner J.L., (1997). Integrating and Guiding Collaboration: Lessons learned in computer-supported collaboration learning research at Georgia Tech. In R. Hall, N. Miyake, & N. Enyedy (Eds.), In *Proceedings of Computer-Supported Collaborative Learning '97*, pp. 91-100. Toronto, Ontario, Canada.

Jones, B., Valdez, G., Norakowski, J., & Rasmussen, C. (1994), Designing Learning and Technology for Educational Reform. North Central Regional Educational Laboratory.

Mahling, D. E., Sorrows, B. B., and Skogseid, I. (1995) A Collaborative Environment for Semi-Structured Medical Problem Based Learning. *Proceedings of CSCL'95*,

Miao, Y., Fleschutz, J. M., and Zentel, P. (1999). Enriching Learning Contexts to Support Communities of Practice. In: *Proceedings of CSCL'99*, pp. 391-397. Stanford, USA, December 12-15, 1999.

Miao, Y., Holst, S., Holmer, T., Fleschutz, J. M., and Zentel, P. (2000). An Activity-Oriented Approach to Visually Structured Knowledge Representation for Problem-Based Learning in Virtual Learning Environments. To appear in the proceedings of COOP'2000.

O'Neill, D. K. (1994). The Collaboratory Notebook: A Networked Knowledge-Building Environment for Project Learning. In T. Ottmann & I. Tomek (Eds.), *Educational Multimedia and Hypermedia*, 1994. (pp. 416-423). Charlottesville, VA: AACE.

Pfister, H. R., Wessner, M., Beck-Wilson, J., Miao. Y, and Steinmetz, R. (1998). Rooms, protocols, and nets: metaphors for computer-supported cooperative learning of distributed groups. *Proceedings of ICLS-98*, pp. 242-248, Dec. 16-19, 1998. Georgia Tech, Atlanta.

Scardamalia, M., Bereiter, C., and Lamon, M. (1994). The CSILE project: Trying to bring the classroom into World 3. In K. McGilly (ed.), Classroom lessons - Integrating cognitive theory and classroom practice (pp. 201-228). Cambridge, MA: MIT Press.

Schank, R. C., and Abelson, R. P. (1977). Scripts, plans, goals, and understanding. Hillsdale, NJ: Erlbaum.

Stepien, W. J., Gallagher, S. A., and Workman, D. (1993). Problem-Based Learning for traditional and interdisciplinary classrooms. Journal for the Education of the Gifted, Vol. 4, pp. 338-345.

Suthers, D., Toth, E., and Weiner, A. (1997). An Integrated Approach to Implementing Collaborative Inquiry in the Classroom. In R.Hall, N.Miyake, & N. Enyedy (Eds.), In *Proceedings of Computer-Supported Collaborative Learning '97*, pp. 272-279, December 10-14, 1997. Toronto, Ontario, Canada.

URL: http://www.rcc.ryerson.ca/learnontario/idnm/mod2/mod2-5/mod2-5.htm.

Wessner, M., Pfister, H. R., and Miao, Y. (1999). Using Learning Protocols to Structure Computer-Supported Cooperative Learning. *Proceedings of the ED-MEDIA'99*, pp. 471-476, Seattle, Washington, June 19-24, 1999.

Woods, D. (1994) Problem Based Learning: How to Get the Most from PBL, McMaster University, 1994.

Acknowledgements

We would like to thank our colleagues in CONCERT (at GMD-IPSI) for valuable discussions of the ideas in this paper, and for their feedback on previous drafts.