Supporting Self-Directed Learning Process in a Virtual Collaborative Problem Based Learning Environment

Yongwu Miao
Technische Hochschule Darmstadt, miao@darmstadt.gmd.de

Follow this and additional works at: http://aisel.aisnet.org/amcis2000

Recommended Citation
http://aisel.aisnet.org/amcis2000/349

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Supporting Self-directed Learning Processes in a Virtual Collaborative Problem Based Learning Environment

Yongwu Miao
GMD - German National Research Center for Information Technology
IPSI - Integrated Publication and Information Systems Institute
Dolivostrasse 15, D-64293, Darmstadt, Germany
Email: miao@darmstadt.gmd.de

Abstract

The PBL process is a self-directed learning process. In the light of self-directed learning theory, the author argues that computational mechanisms can be used to support self-directed learning for PBL groups in virtual learning environments. In this paper, an approach to the support of self-directed learning is proposed. This approach supports self-directed learning in a virtual PBL environment in two ways. Firstly, the shared workspaces together with their accompanying tools and documents allow students to actively engage in collaborative, creative, and unstructured PBL activities. Secondly, the process support tool helps students to organize, execute, and coordinate structured parts of PBL processes. This approach has been adopted to develop our prototype system CROCODILE.

Keywords

Problem based learning, virtual learning environment, self-directed learning, learning plan, coordination

Introduction

Problem-based learning (PBL) is an innovative and increasingly popular instructional method. According to (Jones et al., 1994), students doing PBL take charge of their own learning. Guided by tutors, they define their own problems and learning goals. They come to understand how specific actions relate to their goals and learn how to coordinate their efforts towards achieving these goals. Successful, engaged learners continually develop and refine their learning and problem-solving strategies. The PBL process is a typical self-directed learning process.

For the self-directed learning theorists, self-directed learning is an instructional process that centers on assessing learning needs, securing learning resources, planning and implementing learning activities, and evaluating learning. The learners assume primary responsibility for the whole process (Brockett and Hiemstra, 1991). Knowles, as the founder of self-directed learning theory, suggested that mechanisms could be provided that would help learners to determine their needs, to make learning plan, and to evaluate their progress (Knowles et al., 1984). Such a mechanism might be in the form of learning contracts that contain information on learning goals, anticipated learning resources and strategies, together with a projected time line, and ideas for how to evaluate or validate learning achievements (Knowles 1986). In the light of self-directed learning theory, the author proposes that such mechanisms to support self-directed learning can be designed and developed as shared workspaces and process support tools in computer-based learning environments. By means of such computational mechanisms, geographically distributed learners are supported to actively conduct collaborative PBL activities, to organize and to coordinate their self-directed learning processes.

A Scenario of PBL

The scenario described in this section is developed from a real learning scenario, in which students learned about an issue of environmental concern (Center for Problem-Based Learning, 1998). The PBL course lasted for four days. A dozen students with various backgrounds were involved in the course. The paragraphs below briefly describe this scenario in detail.

At the beginning of the course, students sign on to the course, divide into small groups and go to their respective group rooms to identify and get to know members of their group. Next, they go to an auditorium where a speaker gives them an introduction to the PBL process.

In the same auditorium, four guest speakers talk with students about the discovery of deformed frogs in the local...
area. They challenge the students to investigate the status of the frog population and encourage them to take a proactive stand regarding this environmental concern. The tutor then coaches the students to identify and understand the problem. After discussion, students identify the problem: “what is the cause of deformity of the frogs and how could we prevent it from spreading?” They decompose the problem into sub-problems such as “what are the possible implications for humans?” Consequently, students pose a preliminary problem statement and draw a web structure to represent its many facets.

Next, students identify major issues that are connected to the problems. The identified issues are, for example, the frog habitat, the various types of deformities in frogs, wetlands, watersheds, the effects of pollution on a natural habitat, and so on. An issue can be decomposed into several sub-issues, for example, the issue of frog habitat includes frog food, living environment, etc. The tutor coaches the students to identify what they know and what they need to know by asking questions, commenting, and giving hints. Consequently, students post a first version of their KNK (Know and Need to Know) charts.

As the next step, students prioritize the needs to know (according to importance) and identify the prerequisite relations among them. They define their objectives in terms of learning goals and sub-goals and then design a series of coordinated learning activities that will achieve them. They identify and allocate resources (participants, learning materials, and rooms) such that each student knows what is expected of him / her, where and at what time. In other words, students make a learning plan.

Then, according to the learning plan, students (individually or in teams) collect information from selected articles, books, videos, web sites, and other resources. Other students may be allocated to develop a set of questions for interviews, and then (in teams or individually) they interview people such as the frog experts or other relevant individuals by phone or face-to-face. They take notes and recordings in notebooks. Some students form teams to perform science activities including habitat exploration, soil and water testing, population counts, etc. This will help them to understand the environmental factors that affect the frog. Each team then writes report either collaboratively or by delegating certain individuals to do it. After finishing the assigned activities, they return to their group rooms to share their information. They must communicate certain information to other students who need it to perform subsequent activities. While carrying out these activities, students frequently report to the tutor about their progress. If some unexpected events occur, the tutor will help their students to modify their learning plan in order to fit these changes. At a result of this whole process, students collect the necessary information and share it with their group members according to the learning plan.

Next, according to the learning plan, students divide into teams to discuss the problem. They refine the problem statement, and propose tentative hypotheses and solutions. Then the tutor asks students to communicate (orally and/or in writing) their findings, hypotheses, and solutions in a discussion room. They debate various perspectives on solving the problem. This debate is based on the evidence they have collected and on the principles that they have learned. The students evaluate the reliability of their findings and consider whether their hypotheses and solutions are satisfactory. When they realize that they can not yet propose satisfactory hypotheses and solutions and do not have the necessary knowledge to improve them, then they can iterate through the process again. At this point they re-identify learning issues, revise the learning plan, collect new information, and re-apply the newly acquired knowledge to solve the problem. (In the above scenario, students repeated the process twice). Finally, the students negotiate an agreed set of hypotheses and acceptable solutions.

**The Characteristics of PBL and Requirements for the Support of Self-directed Learning**

This section defines characteristics of the PBL process on the basis of above scenario, and then identifies two key requirements for the support of self-directed learning in virtual PBL environments.

PBL is a collaborative process that encompasses multiple activities. Each activity is associated with a particular goal and is carried out in a specific location. In this location, students can interact with each other and with the learning environment. The PBL participants often divide up the work between them, such that a sub-group is responsible for performing a particular activity. Some activities are performed in synchronous sessions and others are carried out in asynchronous sessions. In each session, it is impossible to predict precisely what actions make up the activity, to know in what order they are performed, or to predict who will manipulate which artifact as the next action. When students are collaboratively generating hypotheses, proposing solutions, evaluating the reliability of their findings, or considering the reliability of the hypotheses and solutions, then various ideas will be contributed by various people in an order that cannot be predicted in advance. Therefore, the system should be able to support students in carrying out such creative and unstructured collaborative activities.

A PBL process is often a long-term task (ranging from a couple of days to a semester) that consists of multiple interrelated activities. These activities may be executed sequentially or in parallel. For example, some learning activities have to be performed after certain prerequisite knowledge has been acquired. Data analyzing activity has to follow the data collecting activity, and is then followed by the activity of drafting a report. Other activities, such as
collecting information by interviewing experts or searching on the Web, can be carried out in parallel. A student may participate in multiple activities and one activity may be performed by many people. These activities may be carried out in the same room or in different rooms. In addition, certain artifacts that were produced in one activity session will used for subsequent activities as well. The learning plan is firstly used to specify the course of the various activities and relations between them. Later, when students implement these activities, they use the learning plan to coordinate their work. Termination of one activity may result in the initiation of subsequent activities at which point some artifacts will need to be transferred from one room to another. In addition, if some unexpected events occur, the students will have to modify their learning plan to accommodate to these changes. Therefore, the system should support users to create, modify, monitor, and implement learning plan for coordinating activities.

In the subsequent two sections, an approach to support self-directed learning in a virtual PBL environment is proposed, where the two requirements identified above (support for creative and unstructured collaborative activities, and support for coordination of interrelated activities) can be met.

Shared Workspaces

In order to support PBL in a computer-based learning environment, we have developed a virtual PBL environment, called CROCODILE (an abbreviation for CReative Open COoperative DIstributed Learning Environment). It is implemented in VisualWorks Smalltalk and available on Windows’95, ’98, NT and Solaris.

CROCODILE adopts a virtual institute metaphor to structure the shared workspace. A virtual institute consists of agent, place, tool, and document components. An agent can be an actor or a group. An actor is a computational representation of a user. A group consists of other agents that may be actors and other groups. A place is a computational space in which actors can present themselves to others and in which activities can take place. The various places in the virtual institute are connected by doors and form a hierarchical structure. The root place of a virtual institute is a campus that contains a set of functional buildings (e.g., administrational building, dormitory, library, and instructional buildings), which, in turn, contain various types of rooms (e.g., homes and public rooms). Actors can navigate from one place to another through the doors. A Tool (e.g., whiteboard, bookshelf, message-box, calendar, etc) provides certain system functions and is available in a place. A Document is a logical unit of information that will be handled (e.g., stored, moved, open, or destroyed) as a whole. Documents can be connected by hyperlinks. Each document consists of information items in the form of text, table, graphics, images and even hyperlinks to other documents. A document can be stored in a bookshelf and can be transferred from one place to another by using a message-box. Documents can be opened on whiteboards. An opened document displayed on a whiteboard can be edited by means of the edit function that is provided by the whiteboard.

A place with actors, tools, and documents forms a learning context. Figure 1 gives an example of a shared workspace (a public room). As illustrated in Figure 1, this room contains a chat-board, a speaker, a calendar, a message-box, a bookshelf, a phone, and a whiteboard. As indicated by the pictures of actors, five users are currently working in this room. When a user selects a tool by clicking on its icon, the corresponding tool window will open. For example, when the user clicks on the chat-board icon or the speaker icon, a chat-board tool or an audio tool will open. Users in this room can use the chat-board tool or the audio tool for unstructured discussions. The user can also look at documents stored in the bookshelf or in the message-box by clicking on the corresponding icon. The documents inside it will be listed in a pop-up window. The user can open a document by selecting it. In this room, two documents are put on the desk. As indicated by the document icons, a user is working on one of the documents. The other document is still closed. In this situation, any user can open it. The user of a document can put it back to the bookshelf by dragging it and dropping it on the bookshelf icon. If the user drags the document icon and drops it on the message-box icon, the system will ask the user where to send this document by offering a list of places. If the user chooses one, the system will send it to the message-box in the selected place. More information about the virtual institute metaphor can be found in (Miao et al., 1999).

Figure 1: A Public Room

If the user wants to share his document with others, he can drag and drop it onto a whiteboard icon. The document editor associated with the whiteboard will treat this document as the currently edited document. The pictures of users displayed on the whiteboard icon indicate who is currently working on the whiteboard. When a user clicks on
the icon of the whiteboard, the document editor that is associated with the whiteboard icon will open on his/her screen (see Figure 2).

By means of the document editor, users can jointly construct their shared knowledge as a hyperdocument. The document editor allows users to create typed nodes (such as “problem”, “issue”, “resource”, “evidence”, “principle”, “hypothesis”, “solution”, etc) and typed arrows (such as “concern”, “derived_from”, “support”, “based_on”, etc). The typed nodes and typed arrows form a diagram, called a PBL-net. An example of a PBL-net is illustrated in Figure 2. Note that each typed node in the PBL-net is associated with a document. By using this tool, students can collaboratively define their problems, identify learning issues, collect information, generate hypotheses and solutions, and etc. More information about the construction of shared knowledge can be found in (Miao et al., 2000).

Figure 2: Document Editor

Process Support

As described above, a PBL process consists of a set of interrelated activities that are performed in synchronous sessions or in asynchronous sessions by multiple participants. That is, at the abstract level of such a process, it is possible to create a process description (e.g., learning plan) that represents the structured steps (activities) and articulates the goals, participants, and handled artifacts of each step. It is also possible to edit this in an ad-hoc way to reflect dynamically changing situations. However, each step of such a process is an unstructured process with little certainty about the work procedure. In order to support collaborative processes with such characteristics, we have developed concepts of session and session-based collaborative process (Miao and Haake, 1999). The term session is defined as such an unstructured process that is executed in a synchronous or asynchronous collaboration mode in a shared workspace by a group of people in order to achieve a goal. The notion of a session-based collaborative process denotes the overall work process that consists of a set of coordinated sessions. An approach to the support of general session-based collaborative processes in computer-based collaboration environments has been developed as well (Miao and Haake, 1999). In order to support self-directed learning in our virtual PBL environment, this approach is adjusted and improved to meet the requirements identified above. In order to support the steps of a process, as described in the last section, virtual places are provided, where participants can actively interact with each other and handle documents by using tools available in the virtual places. The paragraphs below discuss mechanisms to support coordination of the work steps of a PBL process.

Support for the Definition of Learning Plan

A process definition tool is developed in our virtual PBL environment, whose primary users are students. By using this tool, a PBL process can be specified as a hypertext document called a PBL-plan. Figure 3 illustrates the window of the process definition tool. In the upper area of the window, there is a system logo, a button bar with generic functions for handling PBL-plans, and a text input field showing the name of the edited learning process. Below there is a palette of the components of our visual process model language (left hand side) and the content pane (right hand side) displaying a process definition.

As illustrated in Figure 3, the PBL-plan can be specified as a set of connected sub-processes and sessions. Each sub-process is represented as a process node, which may be decomposed into further sub-processes (also represented as further process nodes) forming a hierarchical structure. Each PBL-plan or sub-process has a state attribute. The possible values of a plan state are “created”, “defined”, “active”, and “finished”. The session (represented as a session node) is the elementary process that can not be decomposed. Each session has a state attribute as well. The possible values of a session state are “created”, “defined”, “active”, “suspended”, and “finished”. A session is assigned to some agents who are responsible for achieving the goal of the session. A place is arranged for a session, where the session takes place. Other important attributes are goal, scheduled start time, estimated duration, collaboration mode, and active/terminated conditions.

The Connection Node is used to represent the temporal control points in plans. There are six types of connection nodes: Start, End, AndJoin, OrJoin, AndSplit, and OrSplit. A start node is the entry point of a PBL-plan or a sub-process, while an end node is the exit point of a PBL-plan or a sub-process. A PBL-plan or a sub-process has only one start node and one end node. The split node is used to fork sessions in such a way that either all of the subsequent sessions will be executed in parallel or one of them will be executed and others will be suspended. The join node provides a simple mechanism to synchronize sessions. A
join node specifies when the execution can continue: either all of the preceding sessions must be finished or at least one of them needs to be finished. The arrows connect process/session nodes and connection nodes and represent temporal relations between process node, session nodes, and connection nodes in the process definition.

An Artifact node with arrows represents a document flow from the session where the document is produced to other sessions where the document is consumed. Each artifact has an attribute to represent the current state of the referred document. The possible values of the artifact state are “created”, “defined”, “inEditing”, and “finished”. The artifact input and artifact output are used to transfer documents across sub-processes.

In order to create a node, users can click a button in the palette and position it in the content pane of the window. A corresponding type of node will be created and the state of the node is automatically assigned as “created”. Users can create an arrow by a draw-line gesture going from the source node to the destination node.

In order to specify the properties of a node, users can click on the node to be defined and a corresponding window will pop up. Users can assign values to the node in this window. For example, in order to specify a session node, users have to select a place and a set of participants from the lists provided by the window. They should describe the goal of the session, determine the scheduled start time and estimated duration, and specify the active/terminated conditions of the session. Once all attributes of the session node have been specified, then the state of the session node becomes “defined”. It is important to note that when clicking on a process node, the content pane of the window will show the content of the process node, which in turn contains nodes and arrows that form the sub-process.

Figure 3: Process Definition Tool

Figure 3 illustrates an example of a process definition. It is a sub-process of a PBL-plan that consists of a session for collaboratively customizing materials, a lecture session, and two practice sessions. They are connected by temporal arrows (such as “finish-trigger” and “AND-split”) and by named artifact arrows (such as “course materials). After a process model is defined, it can be stored in a process definition base and can be instantiated and reused. Next, we will discuss the run-time features of the PBL-plan.

Support for the Execution of Learning Plan

Rather than only being a static representation of a learning procedure, the PBL-plan can be executed automatically to coordinate activities. These activities may be carried out by different actors or sub-groups, either synchronously or asynchronously, in the same virtual place or in different virtual places.

In a virtual institute, multiple learning plans may exist and be executed at any one point in time, because it is allowed multiple PBL groups to carry out PBL activities concurrently. It is possible that a learner participates in more than one PBL activity and a place can be used by more than one session. The system provides two ways for learners to execute and get information about learning plans. Firstly, learners can execute and monitor a learning process by using the process definition tool, in which information about all sessions, sub-processes, and their relations are organized and displayed as a hierarchical diagram. Secondly, as mentioned above, there is a calendar in each home and public room. The calendar in a home lists all sessions, of which the owner of the home is a participant. The calendar in a public room shows all sessions, which take place in the public room. The calendar can also be used to schedule isolated sessions.

Normally, the organizers of a PBL course define the top level of a PBL-plan and are responsible for starting it. If the state of a process is “defined”, learners can enact the process manually by clicking the start node and then the state of the process changes into “active”. Consequently, all sub-processes and sessions connected with the start node will be enacted. An enacted sub-process will, in turn, enact the start node of that sub-process. When a session is enacted, the state of the session turns into “active” or “suspended” according to the definition of the session. If the active-condition of the session has not been met, the state of the session becomes “suspended”. For example, if the necessary artifacts have not been delivered or not all participants are present in the place where the session is due to take place. When the state of a session becomes “active”, the state of the artifacts used in this session turns into “inEditing”.

If the collaboration mode of a session is defined as “synchronous session” and the current state of the session is “suspended”, when a new event occurs (e.g. a participant moves into the virtual place of the session) the system will evaluate the active-condition of the session. This may result
in a change of the state of the session from “suspended” to “active”.

In an active session, participants of the session will collaboratively work in the virtual place of the session. Because the activities performed in a session have little certainty and constitute an unstructured process, the system does not provide active support, but allows participants to use tools available in the virtual place at will. How they communicate with each other and collaboratively construct their shared knowledge has been described in the last section.

If participants of a session achieve the goal of the current session, then they can terminate the session by using the process definition tool or the calendar in the place of the session. Then, the state of the session becomes “finished”. If the finished session is not the last session in the plan, then the subsequent sessions will be enacted according to the definition of the process. When a document is produced in a session, users can change the state of the artifact into “finished”. Consequently, the document will be transferred automatically from the place where the document is produced to the place where the document is to be used next according to the definition of the process. The process of transferring documents is realized by using message-boxes. The event of delivering a document may trigger the sessions that are waiting to use the document according to the definition of active-condition of the sessions.

As a learning plan is executed, the state changes of the process/session nodes can be observed in the window of process definition tool (different colors) and in calendars (by text). Notes that it is possible to define and modify unexecuted part of a process definition, even after the process starts to be executed.

Related Work

We compare related work from two viewpoints. One is to compare with workflow systems. The other is to compare with existing PBL support systems.

Comparison with Workflow Systems

Many products of workflow management systems are already on the market with a varied set of features and with various degrees of process support. The Workflow Management Coalition (WFMC) was founded to define standards for terminology and interfaces of workflow management systems (Home page of the Workflow Management Coalition). The approach described in this paper follows the framework proposed by the WFMC. However, session-based collaborative processes like PBL have some distinguishing characteristics and existing workflow management systems can not support such collaborative processes to the same extent.

In session-based collaborative processes like PBL, activities are carried out in a synchronous or asynchronous session. The participants of the session collaboratively work to achieve the common goal by exploiting the tools and documents available in the shared workspace. In existing workflow systems, an activity (or a work step) is defined as a process element that is performed manually (without system support) or by invoking a specific application tool. The shared workspace for performing activities at each step is not explicitly provided in workflow systems. The unstructured activities carried out collaboratively by a group of people can not be supported by these systems.

In PBL, participants of a PBL course often work as a whole from the beginning to the end. The work procedure is defined collaboratively by the group and executed by the same group within the work processes, although sometime they work on different tasks individually or in sub-groups. Participants with different roles (e.g., teacher and learner) can collaboratively perform activities at the same step. In workflow systems, a participant with a certain role works only at the role-related steps and does not care about the work at other steps. For each activity, exactly one role is defined. Even if multiple performers with the same role are engaged in the same step, they deal with different work items individually.

Comparison with Existing PBL Support Systems

Existing PBL support systems provide limited process support. Some systems like CNB (Edelson et al., 1995; O’Neill, 1994) and CSILE (Scardamalia et al., 1994) support unstructured, self-directed learning activities simply by providing shared information spaces like public forums. Participants can access information in the shared information spaces at any time. However, they can not support any structured work procedure. Some systems like Belvedere (Suthers et al., 1997), and Web-SMILE (Guzdial et al., 1997) provide a predefined work procedure in the form of a diagram. Such a procedure description provides guidance for learners to perform PBL activities. However, this procedure description is fixed and can not be changed by learners to fit their needs. More importantly, this procedure description can not be enacted. Some systems like CALE (Mahling et al, 1995) and Web-SMILE (Guzdial et al., 1997) allow learners to identify actions. However, the identified action items are isolated from each other and information for enacting them is missing. Therefore, these systems can support common understanding about work procedure, but can not support the coordination of actions. Furthermore, none of these systems provide shared workspaces where rich forms of social interaction and social construction of knowledge can be supported.
Conclusions

The theoretical basis for this work lies in self-directed learning theory. Through an analysis of a PBL scenario, technological requirements for support of self-directed learning were identified. An approach to supporting self-directed learning in PBL processes was proposed. The author found that self-directed learning could be supported from two aspects in virtual learning environments. Firstly, the shared workspaces with tools and document allow students to actively engage in collaborative, creative, unstructured PBL activities. Secondly, the process support tool can support PBL groups to define, monitor, modify, and execute a learning plan.

The prototype system has been tested and used in our group. Five people used the problem-based learning approach that is supported in CROCODILE to tackle a research topic of interest in our research group. One person took the role of the tutor. The users were able to use the system intuitively to navigate in the virtual institute. They used the tools available in the virtual rooms to support collaborative learning about their research problem. They defined their research problem, their learning goals and resources, and allocated tasks. The trial raised a number of questions, such as how to manage the size of the PBL-net that is created so as to maintain a good overview. Most of our work to date has been focused on demonstrating the feasibility of implementing and using the system. In the next step, we will evaluate the system in real world settings.

Acknowledgements

The author would like to thank his colleagues in the CONCERT department at GMD-IPSI for their help. Special thanks go to Shirley Holst for the great job she has done in revising the paper.

References


