Awareness and Collaboration in the iHelp Courses Content Management System

Christopher Brooks, Rupi Panesar, Jim Greer

Advanced Research in Intelligent Educational Systems (ARIES)
Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9 Canada
cab938@mail.usask.ca, rup785@mail.usask.ca, greer@cs.usask.ca

Abstract. Traditional learning content management systems have minimal support for awareness among learners, and tend to support only loosely coupled collaboration features. This paper shows how we have integrated user model-based awareness features and collaboration features into our learning content management system for both learners and instructors.

1 Introduction and Motivation

Most popular learning content management systems provide poor collaboration support for learners. These systems, initially aimed at providing simple mechanism for the publishing and consumption of educational content, typically provide only loosely coupled collaboration facilities (e.g. discussion forums, chat systems, etc.). The absence of tighter integration of these facilities with course content impedes the potential of learners to effectively collaborate and learn. In particular, distance education learners are at a severe disadvantage as they are unable to see the subtle societal cues that are prevalent in face-to-face teaching scenarios.

One of our goals is to apply both Social Development Theory [15] and Activity Theory [4] to e-learning environments by encouraging learner collaboration in and around the artefacts of learning. The ability to collaborate, along with the awareness of when and with whom to collaborate, provides a potential to increase learner performance and satisfaction. Further, by extending this awareness to other facilitators of learning (e.g. instructors, tutorial assistants, markers, etc.) we believe that we can provide these facilitators with a means to better scaffold learner interaction.

This paper is structured is follows; the following section will outline the architecture of our web-based learning content management systems, focusing on how learner tracking is employed to create detailed learner models. Section three describes three of our current collaboration tools and indicate how they support learning based on activity theory. Section four provides an overview of some of the features we currently employ to increase awareness of learner activities within the learning environment, and outline a prototype visualization for supporting awareness for instructors over class interactions. Finally, section five will conclude the paper with a look at other research avenues we are investigating with this system.

2 iHelp Courses

The iHelp suite of tools[1] was born out of early work in tracking and modeling learners to provide just-in-time expertise location [13]. User models were built up from various observed user interactions (e.g. learners posting questions in a discussion forum), as well as self-declared learning attributes (e.g. learning style, background knowledge, etc.), and intelligent agents were used to match learners with compatible peer helpers on demand.

Since then a number of new directions have been pursued to increase the breadth of this learning environment. The iHelp suite now supports both asynchronous and synchronous discussions, the ability to present packages of standards-based learning objects (e.g. using the IMS Content Packaging format [10]), as well as a shared document annotation tool. One of the focuses of the iHelp environment is to be light weight on the client side yet retain its ability to track what learners are doing. Principle in making this a possibility is the use of asynchronous JavaScript (AJAX) to record user interactions such as when they have viewed a message, posted a message, viewed learning content, or answered the items in a quiz. This information is then stored in the generic tracking model for learner activities and is used to generate derived attributes (e.g. dwell time on a particular learning object).

In addition to this built-in user modeling functionality, iHelp Courses provides a simple API based on the SCORM run time environment [3]. This API allows content packages to store specialized learner characteristics, and helps to support the rapid development of user model-based learning content without modifying the deployed content management system. While this feature is in its infancy, it has been used successfully to create a simple learning-object driven recommender system.

The iHelp Courses system has been deployed and made available to over 1,000 learners in the past two years, and is the primary method of instruction for two fully online courses that teach introductory computer science for non-majors. The asynchronous discussion and synchronous chat systems are available as stand alone tools (named iHelp Discussions and iHelp Chat respectively), and are used by nearly 1,000 students every academic term, resulting in over 5,000 discussion messages being posted and 12,000 chat messages exchanged.

---

[1] See http://ihelp.usask.ca for more information about these tools.

3 Collaboration

One of the principle tenets of activity theory is that the achievement of learning outcomes arises from the interactions between the learner, the artefacts in the environment, the tools to manipulate those artefacts, and the learning community. Mwanza and Engeström indicate that learning is:

> "…driven by genuine developmental needs in human practices and institutions, manifested in disturbances, breakdowns, problems, and episodes of questioning the existing practice." [14].

From this we draw our belief that by providing tools to explicitly link the collaboration of learners with learning artefacts (e.g. content, assignments, interactive simulations) we should be able to positively influence the learning process. This influence could come in many different forms including a decrease of learner attrition rates from online courses, and increase in learner satisfaction in online courses, an increase in learners meeting learning objectives, or an increase in sense of "belonging" in the online community.

Using Activity Theory as an underlying principle in large scale learning systems is not in and of itself novel. Bourguin and Derycke [16] describe a high level architecture for associating CSCL widgets with different learning tasks. Learners can then choose specific tasks to investigate, and get a customized learning environment in which to interact with their peers. We instead take a bottom-up artefact centred view of the problem, and aim to attach collaboration and task features around specific kinds of learning content.

The first approach we have taken is to couple our discussion and chat tools directly with instructional content. This coupling does not occur only at the package level, and instructors are given much latitude in how they would like associations to be made. Further, we aim to make the collaboration as ambient as possible and minimal overhead is needed from the user to initiate collaborations. This is explained in more detail in section 3.1.

The second approach we have taken is to create a groupware application that allows learners to share and collaboratively annotate text (computer program source code in our experiences, though we intend to broaden this to other text artefacts such as student essays). This workspace must be explicitly invoked and accepted by both users, and gives learners the ability to highlight and annotate directly on the shared artefact. This is elaborated in section 3.2.

3.1 Learning Object-based Collaboration

The iHelp system provides two different discussion interfaces. The first is an asynchronous system, where messages are posted in forums and message threads are hierarchal in nature. Messages can be posted in plain text or a subset of HTML and can contain any number of attachments. The second interface is a synchronous chat system which allows learners to exist in many different channels at once, and operates much like Internet Relay Chat (IRC). Learners can also initiate new private channels for private chats with one another. All learners have the ability to create up to eight pseudonyms for protecting their identity during interaction and, depending on the preferences set by the facilitator, can post to certain discussion forums anonymously.

E-learning standards describe a content package as a collection of hierarchal activities which learners typically traverse using an inorder traversal[2]. Taking a pragmatic approach to defining a learning object, we allow facilitators to create new discussion forums or chat rooms and associate them with any subtree of activities within the content package. As learners traverse the content they are automatically moved into either the chat room or the discussion forum associated with that content. Learners can switch back and forth between chat rooms and discussion forums using a tabbed interface, and new chat rooms automatically open up in the background so current collaborations that might be happening are not lost. Finally, collaboration can be cancelled or terminated by clicking on a “Show/Hide” on the interface and minimizing the chat or discussion forums. Fig. 1 shows an example of this collaborative interface.

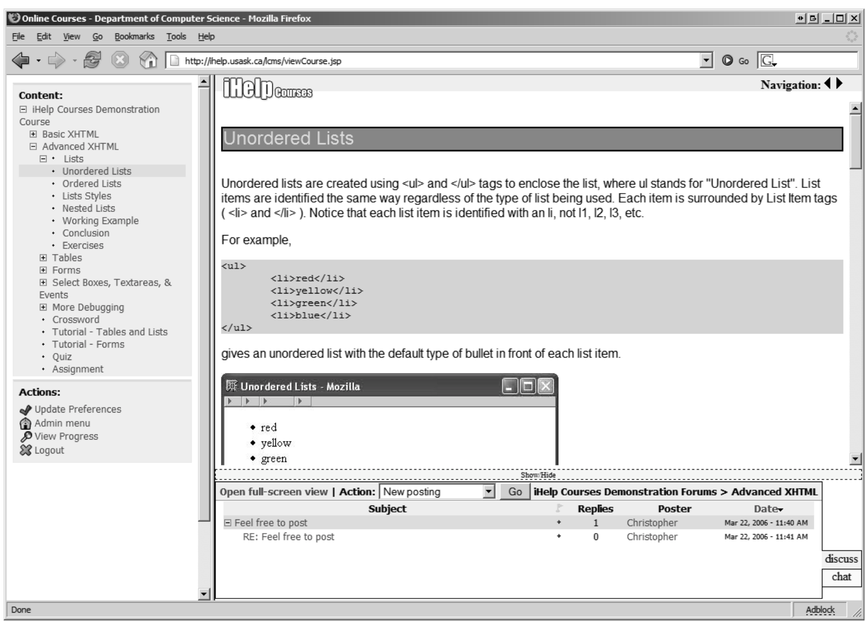

Fig. 1. Learning Object-based Collaboration tools. The left menu shows a single content package with hierarchically arranged activities. Each activity links directly to a file within the content package, which is rendered in the top right frame of the browser. The bottom portion of the right frame holds collaborative tools (with the discussion forum interface shown here),

---

[2] Strictly speaking this traversal can be modified by overlaying a set of sequencing rules to the activity hierarchy and applying a student tracking model. The interested reader is directed to [8] for more information.

which can be toggled between asynchronous or synchronous discussions using the tabs on the right.

3.2 Collaborative Annotation Environment

To help the novices in learning programming concepts in online courses and to enhance the collaboration among them, we have integrated a collaborative annotation tool for text documents within our existing content management system. While this tool is still in the prototype stage, it is quite stable and has been deployed and used in a number of online and blended learning courses over the last eight months.

While not yet commonplace, a number of collaborative editors have been built elsewhere to encourage learners to share text artefacts. CAROUSEL a web based system allows students to share algorithmic representations with their peers. Students can create their algorithmic representation using any utility or software, and can upload those files using the system. These representations are then available for other students to view, evaluate and discuss through discussion forums [9]. GREWP supports the learning of programming by allowing students to edit the same code simultaneously. This application is a stand alone application, and also provides means to record and replay the interactions that have occurred between participants [8].

In comparison to these and many of the other collaborative editors out there, our tool has two main goals: to allow learners to collaborate actively with one another around the object of interest (program source code in our case), and to integrate seamlessly into our content management system to reduce the complexity of starting the tool for novice users.

Students working in this tool can share their artefacts with an instructor, teaching assistant or peer helper by pasting the code contained in their local 'editing' workspace into the shared code-annotation workspace. These changes are then immediately displayed for the other collaborator(s). Either collaborator can then annotate the document, highlight and point to certain areas, and chat in the associated chat tool.

One general finding of CSCW is that group awareness is very important to successful collaboration. This means that collaborators ought to be aware of the actions of each other and of everything else that concerns the common project [12].. Support for awareness in groupware interfaces has been shown to improve groupware usability [6,7]. When collaborators can easily gather information to answer questions like who is collaborating, what is the collaborator doing and where is he or she working, they are able to better organize their actions, anticipate one another's actions, and better assist one another [5]. Our collaborative code annotation tool takes advantage of the iHelp chat window to support informal awareness of presence between participants. This visible list of participants along with the unique color assigned to each collaborator's interactions is explicitly displayed in the interface. Colour makes all users in the group aware of the activities of a particular user. If a user wants to refer to (say) a particular line of code, he/she has to simply select a line of text and the selected line will blink and be highlighted in his/her assigned color, on every other participant's browser screen. The system's representations of other users' activities are implicit. By the mechanism of annotation, users can comment to each other about the text artefact. Replay mode allows collaborators to review each other's activities.

The current implementation of the iHelp collaborative annotation tool enables facilitators and peers to reference and add annotations to the specific lines of code in a software program. Comments attached to a a particular line of code are made immediately visible to the other participating user(s). The annotation mode is a powerful feature as annotations can not only help students to remember, clarify, comprehend, and think about the key points, but they can also facilitate students to engage in discussion [2]. Fig. 2 shows a screenshot of the collaborative annotation tool's interface.

The iHelp collaborative annotation tool is integrated directly within the context sensitive chat system. It can be invoked by clicking on another learners name within the chat room, and indicating that a shared session should begin. Once accepted, a new window opens for each of the users, who then can share a document as they see fit. The ease of starting the application makes it suitable for both computer savvy users and those learners who are new to this kind of environment.

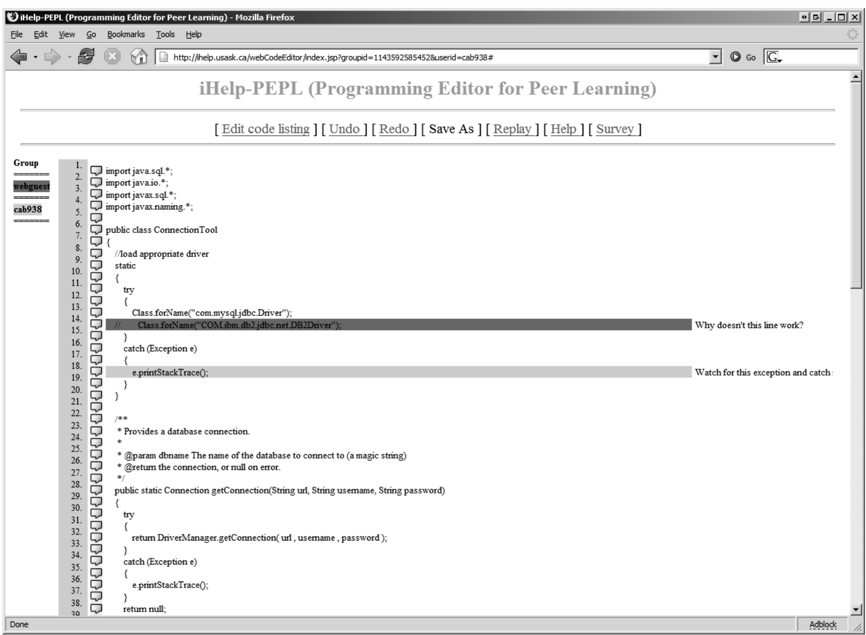

Fig. 2. Collaborative editor interface. The list of participants in this collaborative setting are given on the left, and are colour coded. Each time someone clicks on a line in the interface a bar the same colour as their name flashes to indicate what line has been chosen. Annotations can be added by double clicking on a line, and show up on the right hand side (long annotations are truncated until the mouse hovers over them). The entire session can be replayed using the links at the top of the window.

The prototype is written in JavaScript, DHTML and HTML with a conceptually simple front end. It makes extensive use of asynchronous JavaScript (e.g. AJAX) and DHTML to realize the interactive markup effects and offers capabilities not available in existing systems. The shared-code workspace is basically a framed area, where the code to be shared will be displayed in the browser window of each user in the collaborative group. The shared-code area follows the principle of relaxed WYSIWIS for uniform look. By downloading the most recent version from the shared workspace into their local editing workspace, the students can save the copy of the document locally.

4 Awareness

While mostly absent from content management systems, awareness has been a key field of study for many in the area of Human Computer Interaction. Gutwin et al. break the concept of awareness up into four different types: social awareness, task awareness, concept awareness, and workspace awareness [12]. Of these types, workspace awareness is perhaps most relevant for learning content management systems. Defined as "up-to-the-minute knowledge about other students' interactions with the shared workspace" [12], workspace awareness can be exploited to increase the social interconnections between students, motivate the exploration of material, and provide feedback to the instructor of the course. This section outlines how we have provided awareness in the iHelp suite with the goal of increasing collaboration as well as supporting instructors in evaluation of how the course is proceeding.

4.1 Peer Awareness

Awareness within the iHelp suite focuses almost exclusively on encouraging and supporting collaboration between learners. One of the responses of instructors who used the learning object-based collaboration features was that students tended to browse "away" from the conversation. The instructor would then have to chase down the student by following the chat rooms he or she had gone into. At the same time, learners indicated that they didn't know who was in the environment for collaboration, unless they checked through much of the content first. In many cases this meant that collaboration would start by chance, and that it would only continue until someone began looking at different content.

We implemented a number of features to address this problem, and two of these features were specifically concerned with increasing learner awareness that they were really in a shared environment. The first was to annotate each activity with an indicator of what collaboration is available. This was done by appending a small icon of a person to the end of the title if there were two or three people viewing that piece of content, or appending two such person icons iff there were more than four people viewing that piece of content. In addition, a fractional number was placed after the title of the activity where the numerator indicated the number of asynchronous forum messages on that topic that have been read by the user, and the denominator represented the number of total messages available to read. Figure 3 demonstrates these awareness features.

The second feature was related specifically to the chat channels. To prevent disorientation, new chat rooms open in the background, so current discussions are not disturbed. Awareness of what is happening in background channels is given by changing the title of the channel to indicate the number of messages in that channel that have not yet been read.

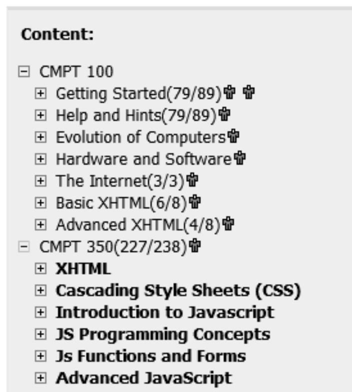

Fig. 3. Peer awareness in the navigation component of iHelp Courses. This figure shows two courses, *CMPT 100* and *CMPT 350*. In the first course the instructor has chosen to add individual discussion modules and chat rooms to each module. The first module, *Getting Started*, contains 89 discussion forum messages, 79 of which have been unread by the current learner. There are currently four or more learners reading content in that module as is indicated by the presence of two small person icons. The setup of the second course is somewhat different – here the instructor has chosen to implement one larger forum and chat room for all modules in the course. As the learners progress through individual modules their collaboration tools will not change.

4.2 Class Awareness

In addition to ambient awareness within the learning environment, we are working on providing explicit visualizations of learner interaction for instructors (and eventually for learners as well. Most e-learning systems tend to provide only rough statistics on student activity, and scale poorly to large classes. In our initial exploration we have applied sociograms to learner models where the nodes in the sociogram represent individual learners, and the directed edges between nodes indicate some form of interaction (either a reading of or a reply to an asynchronous forum posting). In large classes (e.g. those having more than 100 students), this becomes unwieldy. We further refined this sociciogram by breaking learners into three different groups:

- Participants: Those individuals who have written messages, either on their own or as replies to other messages. A participant is connected to another participant in a directed fashion if the first has replied to the second.
- Lurkers: Those individuals who have read postings but have not written any. Lurkers have no edges between themselves and other nodes, but are situated either closer to the centre of the sociogram if they have read many postings, or closer to the outside edge of the sociogram if they have read few postings.
- Delinquents: Those individuals who have neither read nor written a posting.

Each group of learners was put into their own sociogram that aligned nodes along the exterior of a circle. The different sociograms were then layered on top of one another such that the delinquents were farthest from the centre of the screen, the participants were closest to the centre of the screen, and the lurkers were in between (fig. 4). This corresponded well both with the perceived participation rate of individuals, as well as with the sizes of the different classes of individuals in blended courses. Nodes were further colour coded to represent the different roles of the people involved; red for potential facilitators, and light grey for learners.

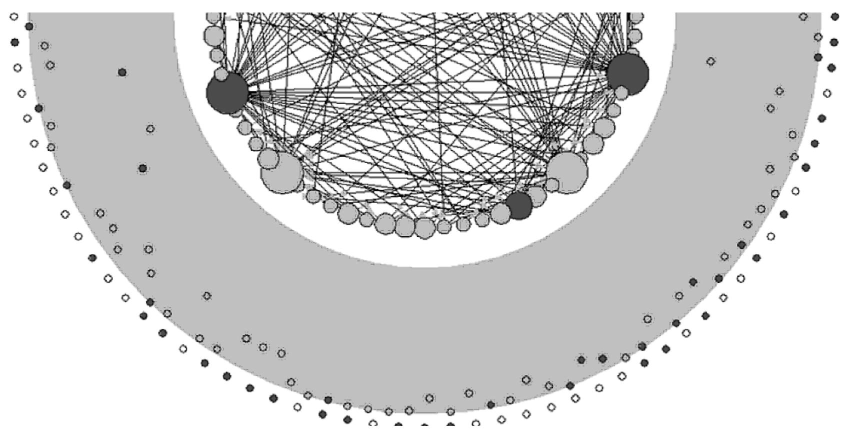

Fig. 4. Portion of a sociogram for an introductory computer science forum. Dark nodes indicate facilitators, while lighter coloured nodes indicate learners. The inner circle is made up of participants, four of which are very important to the community (as shown by having a larger node size). A casual observation of this network indicates that, while some learners write a fair bit (many interconnected nodes in the middle), there are lots of learners who haven't ever read anything (the outer ring of delinquents), and many lurkers who read very little (as they tend to be closer to the outside of the middle ring instead of the inside of that ring). Note that the ring of delinquents includes a disproportionately high number of facilitators as our currently deployment gives access to this forum to most staff and faculty in the department.

5 Conclusions and Future Directions

The iHelp suite aims to be a standards-based research oriented e-learning system where detailed user models can be built. This paper has outlined two specific aspects of this system, collaboration and awareness to support this collaboration. Firmly inspired by activity theory, and the thought that collaboration best happens around the artefacts of discourse, we believe these features have made our environment more suitable for distance and blended learners. Empirically testing such an assertion is difficult, but feedback from instructors and students who have used these tools have been extremely encouraging.

Approximately 40% of the learners who used collaborative annotation tool, filled-in the feedback surveys and responded very positively that it helped them to work collaboratively. For example, one learner quoted "*It enabled students to interact with someone who knew the topic and could answer any kind of questions*". All of the survey respondents reported that their first preference will be to use this collaborative annotation tool for working out on their programming problems and then instant chat or discussion forum.

To date there has been limited use of the user models underlying this system. However, we do have a number of plans to enrich the awareness aspects of the system using more details in the user models. For instance, we are working on a system to motivate student interactions in the environment by allowing them to compare aspects of their activity to that of other learners (including the derived "average" learner, or perhaps an instructor created "ideal learner"). We still believe in grounding this comparison in the artefacts of learning, and are working on ways of changing the course content navigation window to support this kind of user model awareness [1].

Our implementation of class awareness is still in its early stages. We have observed that instructors are both interested in this kind of visualization, as well as willing to change their pedagogical style based on the knowledge they gain of classroom dynamics. We are planning studies to both qualitatively and quantitatively analyse this visualization. In particular, we believe that we can classify the health of class discussions (and individual learners in those discussions) based on graph characteristics of our awareness sociograms. We are also investigating the motivational effects of opening this visualization up to learners and student assistants, where they are able to see clearly how they fit in with the rest of the class.

Acknowledgements. This work has been conducted with support from Science and Engineering Canada (NSERC) and its Learning Object Repositories Research Network (LORNET). Special thanks to Lori Kettel, Robin Dunlop, Ryan Silk, Erik Fredrickson, and Jian Liu for their help in building the technology our research is based upon.

References

1. Brooks, C., Hansen, C., Greer, J. (2006) Social Awareness in the iHelp Courses Learning Content Management System. Workshop on Social Navigation and Community-Based Adaptation Technologies, Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006). In Submission.
2. Chong, N. S. and Sakauchi, M.(2001). Creating and sharing web notes via a standard browser. In Proceedings of the 2001 ACM Symposium on Applied Computing (Las Vegas, Nevada, United States). SAC '01. ACM Press, New York, NY, 99-104.
3. Dodds, P. (ed). (2001) Sharable Content Object Reference Model (SCORM) Version 1.2, The SCORM Run-Time Environment, Advanced Distributed Learning.
4. Engeström, Y., Miettinen, R., and Punamäki, R. (eds.) (1999) Perspectives on Activity Theory. Cambridge, MA: Cambridge University Press.
5. Gutwin C., Stark, G., and Greenburg, S. (1995) Support for Workspace Awareness in Educational Groupware. Pro ACM Conference on Computer Supported Collaborative Learning, pp 147-156, Indiana University, Bloomington, Indiana, USA October 17-20, Distributed LEA Press.
6. Gutwin, C. and Greenberg, S. (1998). Effects of Awareness Support on Groupware Usability. Proc. ACM CHI '98, 511-518.
7. Gutwin, C., Roseman, M., and Greenberg, S. (1996) A Usability Study of Awareness Widgets in a Shared Workspace Groupware System. Proc. ACM CSCW '96, 258-267.
8. Hickey, T. J., Langton, J., & Alterman, R. (2005). Enhancing CS programming lab courses using collaborative editors. Journal of Computing Sciences in Colleges, Volume 20 Issue 3, 157-167.
9. Hübscher-Younger, T. and Narayanan, N. H. 2003. Constructive and collaborative learning of algorithms. In Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education (Reno, Navada, USA, February 19 - 23, 2003). SIGCSE '03. ACM Press, New York, NY, 6-10.
10. IMS Content Packaging Information Model, Version 1.1.4, IMS Global Learning Consortium Inc., 2004
11. IMS Simple Sequencing Information and Behavior Model, Version 1.2, IMS Global Learning Consortium Inc., 2003.
12. Koch, M. (1994). Design Issues and Model for a Distributed Multi-User Editor. Computer Supported Coop. Work Volume 3, Issue 3-4 , 359-378.
13. McCalla, G., Vassileva, J., Greer, J., and Bull, S. (2000) Active Learner Modelling. In proceedings of the conference on Intelligent Tutoring Systems 2000. Montreal, Canada.
14. Mwanza, D., and Engestroöm, Y. Pedagogical Adeptness in the Design of E-learning Environments: Experience from the Lab@Future Project. E-Learn 2003.
15. Vygotsky, L.S. (1978) Mind in Society. Cambridge, MA: Harvard University Press.
16. Bourguin, G., and Derycke, A. (2001) Integrating the CSCL Activities into Virtual Campuses: Foundations on a new Infrastructure for Distributed Collective Activities. In Proceeding of Euro-CSCL 2001.