Computational Support for Collaborative Learning through Generative Problem Solving

N. Hari Narayanan, Cindy E. Hmelo, Valery Petrushin,
Wendy C. Newstetter, Mark Guzdial, and Janet L. Kolodner

EduTech Institute, College of Computing, Georgia Institute of Technology

Abstract

In this paper we present a vision of computer-supported collaborative learning through solving generative problems - problems that promote open-ended inquiry and have multiple solutions. This vision stems from a novel and evolving approach to collaborative learning that we are developing at the EduTech Institute. This approach is based on the following premises: that learning is facilitated by generative problem solving, collaborative work and use of multiple cases; that learning and skill acquisition need to be, and can be, scaffolded through software; and that a computer environment which integrates a shared and structured electronic workspace with a full variety of functionalities can effectively support all of the above. We describe this approach and the architecture of the corresponding computer environment. This environment is designed to serve three critical functions: provide a shared workspace for students, facilitate inter- and intra-group collaborative work, and make available the tools and resources that students need for problem solving and learning. The software components of the environment that have already been implemented are described. In the final section we frame ongoing and planned research and development efforts in terms of the characteristics desired of such an environment and ways of assessing its impact.

Keywords — case-based methods of instruction, educational groupware, instructional strategies and approaches.

1. Introduction

Too often, classroom instruction provides students with many bits of knowledge that they are never able to assemble and apply in productive ways, particularly outside the classroom walls. One reason for this is the focus of traditional schooling on learning isolated facts in compartmentalized disciplines. Not surprisingly, this knowledge often cannot be transferred to real-world problems. Theories of constructivism and situated cognition suggest that for learning to be useful the learner needs to be actively involved in constructing new knowledge within meaningful contexts, not merely absorbing it. Furthermore, learning is enhanced by group-oriented collaborative work, reflection and articulation. These are therefore the central premises of a multidisciplinary approach to structuring learning within the context of case-based instruction that we are developing at the EduTech Institute. This approach is called Multiple Case-Based Approach to Generative Environments for Learning (McBAGEL).

Three factors distinguish this approach: (1) The use of generative problems to promote learning. Generative problems are those that motivate open-ended inquiry, whose solutions require synthesis, which have multiple solutions, and which, therefore, promote the generation, evaluation and combination of ideas in the course of problem solving. The type of generative problems that we use are design problems. (2) The use of multiple cases provided by computer-based case libraries as knowledge sources to aid problem solving. (3) The emphasis on software-scaffolded and group-oriented collaborative work in and out of the classroom.

We are designing a computer-based learning environment that we expect students to use as a workspace for conducting work as part of this approach. The architecture of this environment and its components is the main topic of this paper. However, since the environment's role is to support collaborative learning in the context of generative problem solving, a discussion of the approach and the educational philosophy behind it precedes the description of the computer environment. Then, software components of the environment that have already been designed, implemented and used in classrooms of Georgia Tech are described. In the final section we frame ongoing and planned research and development efforts in terms of the characteristics desired of such an environment and ways of assessing its impact.

2. Educational Framework

Our approach is based on a synthesis of ideas on learning and problem solving from the fields of education, cognitive psychology and artificial intelligence. This approach is based on the following five central tenets:

(1) Learning is enhanced by problem solving. Learning is more effective when it occurs through activities associated with solving generative problems (e.g., identifying and formulating the problem, generating alternatives, evaluating, decision making, reflecting, and articulating) rather than through transmission models of instruction. Design, by its very nature, is a generative activity. Therefore, *design-oriented problems* are particularly effective for technical domains like engineering and architecture and may well provide effective anchors for math and science learning.

(2) Collaborative work is central to learning. Students are expected to solve problems and do assignments in groups. Group-oriented work, in and out of the classroom, is important both in facilitating learning and in preparing students for today's multidisciplinary team-oriented workplaces. As students work in collaborative groups, they are forced to articulate and reflect upon their thinking, leading to an appreciation of the importance of distributed cognition [14] as well as enhancing learning and subsequent transfer [3]. Collaborative work allows students to successfully tackle problems more complex than what any one group member could do alone.

(3) Access to multiple cases will facilitate flexible learning. Providing students with access to multiple cases that contain information-rich and contextualized descriptions of specific situations set within the broader context of a course can significantly impact learning and transfer. The availability and use of multiple cases during problem solving facilitates learning new knowledge, and supports the adaptation and transfer of previous solutions to the current problem [11]. It is expected that by revisiting design skills through numerous cases, flexible transfer of these skills will be supported [20]. Intelligent computer-based case libraries can provide students with not only such access but also means of flexibly navigating among cases and parts of cases.

(4) Learning and the acquisition of problem-solving skills need to be scaffolded. The experiences implementing effective problem-based learning environments teach us that solving real-world problems requires scaffolding, i.e., help from facilitators, knowledgeable experts, and the learning environment [12, 18]. The goals of scaffolding are to enable students to carry out a reasoning process or achieve a goal that they would not be able to do without help, and to facilitate learning to achieve the goal without support. The scaffolding of different skills can be provided through software, by appropriately utilizing multimedia and tools such as collaboration software, simulation and visualization programs, decision-support systems and smart case-libraries.

(5) A shared electronic workspace that seamlessly integrates a full variety of functionalities for the above will enhance learning. This workspace will tie together tools that students will use while solving problems, collaborating, and perusing multiple cases. It is also an ideal vehicle for providing adaptive software-realized scaffolding of various skills. Finally, it will encourage both synchronous and asynchronous collaborative work among students. Such an integrated yet flexible computer-based learning environment that the students use as a "professional workspace" is a central component of our approach.

We want to situate classroom learning in information-rich contexts that afford opportunities for problem formulation, exploration and discovery. Students will work on problems for extended periods of time, reflecting and articulating on both the process and the product. Case libraries will provide them with both relevant data and specific solution strategies in the domain of instruction, all within the context of complex and realistic real-world problems. The problems students have to solve and the cases that are made available to them serve as anchors for learning. Collaborative, reflective and articulative activities, aided by the tools and cases provided by the computer-based learning environment, should improve the students' knowledge, problem solving skills, and self-directed learning skills. Cases, being rich knowledge structures that explicate both conceptual and strategic knowledge, will allow the students to master concepts, principles and strategies in the course of attempting to solve problems. The collaborative nature of student activities should facilitate the construction of new knowledge since it encourages articulation and intra-group communication. Our approach is designed in particular to address the following three issues.

Cognitive Flexibility and Transfer. Consideration of a single case leads to inflexibility of the acquired knowledge and strategies [22]. Rather than having students focus on a single case, our intention is to have students revisit ideas from multiple cases both through the design problems that students work on and the design cases in the case libraries. We believe that by having students analyze multiple cases, and by having them reflect on how these cases are similar and different to the problems they are solving, more flexible knowledge should be constructed. The cognitive flexibility theory [20] supports this prediction.

Collaboration. Collaboration is a key piece of our approach. Research on collaborative learning shows that learning while solving problems in groups facilitates the learning of articulation skills, makes learning more effective for all group members, and allows students to successfully tackle problems more complex than any one group member could individually solve [3, 14, 17]. Moreover, the collaborative discussion that occurs is important for student learning because it activates prior knowledge, thus facilitating the processing of new information [2, 19]. On the other hand, Blumenfeld et al. [1] suggest that students may have more motivation to learn but make less use of learning and metacognitive strategies. In addition students may not have the skills to benefit from collaborative work. Therefore it is important to help students to collaborate well together in order to make collaborative learning work well. Aspects of our approach - the division of the student body into small groups, the complexity of the design problems that the groups will tackle, and the use of collaboration software to scaffold communication and cooperative work - are all intended to overcome these limitations and enhance the benefits of group-oriented learning.

Reflective Articulation. Two important aspects of our approach are articulation and reflection. There are several forms of reflective articulation including generating analogies [10], predicting outcomes of events or processes [21], developing questions about the learning materials [10], and self-explanations [4]. Studies suggest that reflective articulation can enhance retention, elucidate the coherence of current understanding of the problem being solved, develop self-directed learning skills, and provide a mechanism for abstracting knowledge from the content in which it was learned, thus facilitating transfer. It is very important to provide two levels of articulation - individual and group - in a collaborative learning environment. Also, it is not just articulation by itself that is important, but it is the specific kinds of articulations that engenders reflection - *reflective articulations* - that lead to enhanced understanding. The goal of reflection is to analyze and evaluate one's knowledge, learning and problem solving strategies. Several researchers have demonstrated the importance of articulation and reflection in learning. Pirolli and Recker [15] suggest that reflection on problem solutions that focuses on understanding the abstract relationships between problems is related to improved learning. Lin [13] has found that reflection on problem-solving processes leads to enhanced transfer and that technology can be used to scaffold appropriate kinds of reflections. One way that reflection can be enhanced is through the articulation of meta-cognitive knowledge and skills that typically occurs in collaborative discourse.

3. Computer Support for Collaborative Learning in McBAGEL

Figure 1 is a schematic diagram of the architecture of a software environment that we are developing to complement the McBAGEL approach in classrooms. This environment provides an external memory for keeping track of problem specifications, important facts and constraints, ideas about how to deal with the specifications, and learning requirements. The main screen provides several fields for keeping track of multiple sources of information, design alternatives, and further actions to be taken. Space is provided to record the facts and constraints that are important, to record ideas about how to deal with the specifications, and to keep track of what else needs to be learned, what information needs to be collected, and what actions need to be taken. Together, these windows allow the student to see where s/he is now, where s/he has been, and where s/he is going. This screen can be used as an individual workspace or as a shared workspace for the group. The main screen also provides access to other resources and tools that students need to solve the design problems: case libraries and other information resources; tools for simulation, visualization, decision making etc.; a tool for inter- and intra-group communication, collaboration, and multimedia document sharing; and a set of basic tools such as document processing programs, drawing/painting programs and spreadsheets.

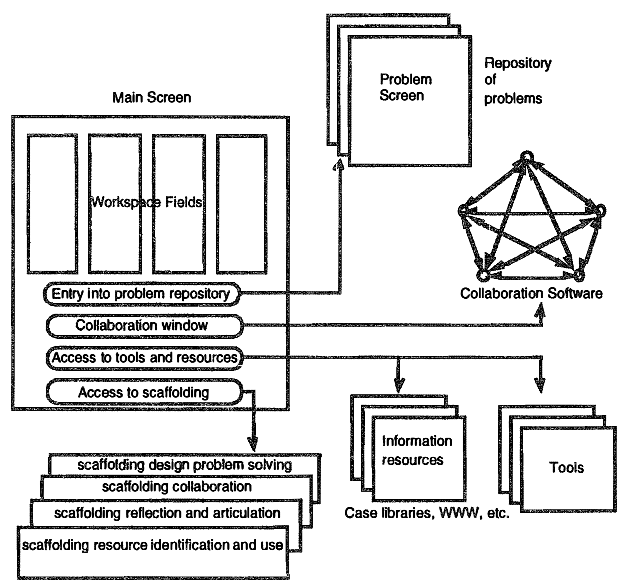

Figure 1. Software Architecture

The problem screen provides easy access to the evolving problem description. This screen begins with a minimal description of the design problem presented to the students. Details emerge as they inquire about additional information about constraints, material resources and functional issues regarding the design.

The collaboration window allows students to enter into a collaboration environment that provides much more than mere communication facilities. It will provide an ability to enter into structured discussions on different topics pertaining to the class and the problem at hand as well as to share multimedia resources with other members of the group and class. A user will be able to browse through past and ongoing discussions which are presented in a structured format to allow easy topic-based, time-based or author-based browsing, and to contribute to those discussions by constructing and sending different types of messages. This collaboration facility will be made available to not only students, but also to teachers. It will provide teachers with a means to collaborate in conducting a course and to share experiences and learn from each other. It can also be a vehicle for student assessment based on their collaborative interactions.

In addition to providing a work environment, this system makes available scaffolding to help novices with design, collaboration and reflection. Design scaffolding will vary as a function of the design stage students are working on. For example when the students are working on problem formulation, the software will provide coaching to help them understand what is involved in this stage: e.g., identifying the problem, formulating the problem, partitioning/decomposing the problem, and framing the problem. The collaboration software will provide procedural facilitation to aid in the development of collaboration skills. Reflection will be facilitated through the articulation that occurs during collaborative problem solving and learning activities.

In summary, this environment will provide means to organize and manage projects from the students' perspective (e.g., the main screen provides for explicitly listing organizational and learning issues) and the teachers' perspective (e.g., tracking student progress and keeping records of student work). In addition, we envision that the environment will be used for research purposes (e.g., archiving data such as the inter- and intra-group communications and resource sharing that took place during a course for later assessment, collecting data to be used for student/group modeling in order to devise better course- and student-specific on-line scaffolding and coaching methods, etc.). An initial prototype of this environment has been developed with Hypercard on the Macintosh platform, but it has not yet been tested in a classroom. Borrowing from the metaphor of the white board workspace of problem-based learning found in medical schools, this prototype provides an electronic workspace that is split into four regions. It also allows easy access to other tools and resources. Figures 2 and 3 show the workspace and problem screens of this prototype.

Here is a brief scenario to illustrate how we imagine the students will use this environment. Students, who will be working in small groups, enter the environment at the main window shown above, which represents their shared electronic workspace. They are provided with relevant information on the design problem they need to solve via the button "new problem". In this case, it is to design an archery stadium for the Olympics. As students are initially formulating and understanding the problem, they will be encouraged to identify data relevant to the problem from the information they have been provided with, and to articulate this by recording those in the "facts" space. Similarly, as they consider alternative solutions, they will make use of the "ideas" space. The problem-based learning methodology that this environment embodies explicitly prepares students for self-directed learning by requiring them to identify their knowledge deficiencies in the "need to learn" space and the actions they plan to take to remedy those deficiencies in the "action plan" space. Several buttons are found on the bottom of the screen that provide access to different tools that they will need to solve the problem. "Stage" is a pull-down menu which acts as a gateway to various kinds of software-realized scaffolding tailored to different stages of problem solving.

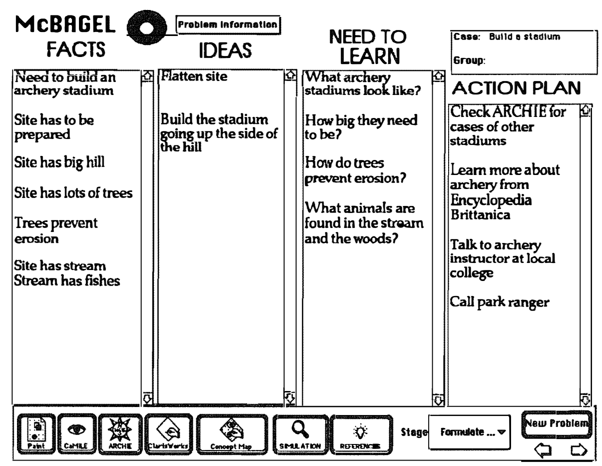

Figure 2. McBAGEL Workspace Screen

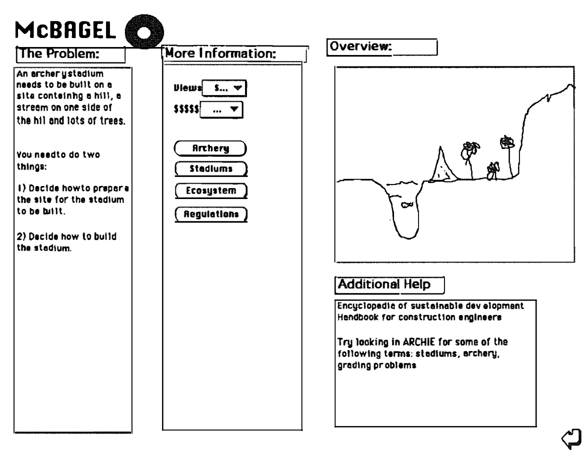

Figure 3. McBAGEL Problem Screen

While the structure of this environment is still evolving, some of its components have already been designed, implemented and individually fielded in classrooms. In the following two sections we elaborate on these implemented components and describe future directions for our research.

4. Implemented Components

• *Case Libraries:* Research on case-based reasoning [11] provides guidelines for indexing and making available resources needed while problems are being solved, especially case materials. Case libraries organize cases in ways that make it easy to access their most interesting parts, understand their implications, and recognize the range of problems needing solving and the range of solution methods available. Case studies are structured in terms of overviews, problems, stories, and responses. Each story discusses some problem that arose in designing some artifact, the way that the problem was addressed, and the outcomes that resulted. To make it easy for users to extract from stories their important points, stories are presented with illustrative graphics, and several kinds of contextual information is associated with each story. Students can examine the full artifact that some story is associated with, can see a general description of the problem the story addresses, a general description of the kind of solution it provides, and can ask to follow links to other stories that illustrate a similar problem or solution. The stories help students discover which issues they should be considering during design and help them to anticipate the results of carrying out their proposed designs. We have developed a number of case libraries in support of design problem solving.

• *Case Library Authoring Tool:* DesignMUSE [5] is a case library authoring tool that has been developed to allow easy construction of case libraries. During the 1995 Winter Quarter it was used to create a library of environmental cases for use in our sustainable technology classes. Thus, while existing case libraries act as intelligent information resources, this authoring tool will allow students to construct their own case libraries to record the design problems that they solve. Both the authoring tool and the case libraries are built on Common Lisp for Macintoshes.

• *CaMILE:* Our collaboration software CaMILE [8], based in principle on CSILE [18], integrates information-gathering tools, communication tools, and applications into a collaborative environment. CaMILE provides a discussion environment into which the full range of text, graphics, spreadsheets, video, and so on that reside locally or on the Internet can be incorporated. It is designed to meet two goals. First, it serves as a collaboration and information indexing tool. Discussions are structured and annotated with links to material anywhere across the network. Second, it serves as a design support tool. Discussions about design problems can be annotated with links to actual ongoing designs. The discussion trace can then serve as a design rationale and a case study of a design. It allows students to collaborate in learning and problem solving by providing a facility for structured inter-group and intra-group communications that are archived, and by providing a way to share multimedia documents easily among collaborators. Like CSILE, CaMILE scaffolds collaboration through procedural facilitation. While electronic mail merely allows team members to share ideas, CaMILE helps them to organize their ideas into coherent arguments, relate their ideas to one another, and use resources across the network to support their arguments. CaMILE was built with Hypercard on Macintosh computers.

• *Exploratory Simulations:* We have developed a range of exploratory simulations [16] that enable students to learn through simulated experience. Key to these simulations are tight integration with real world problems and activities, and flexible specification of simulation choices to allow for creative and sophisticated simulation problem solution. These simulations have been constructed using the Smalltalk language.

5. Future Directions

Learning from case libraries: As students are solving problems, several kinds of resources are needed to help them. Clearly, they need access to documentation, of the kind found in books and encyclopedias. But another significant but often overlooked resource is codified prior experience: e.g., cases that describe solutions to similar problems. Our approach to supporting learning from prior experience is to make on-line case libraries available from within the software environment. Cases help with understanding a problem better, suggesting solutions and parts of solutions, and evaluating proposed solutions, thereby helping a student to know where to focus his/her attention. Our research on case libraries will proceed in two directions. One is generating content: creating the kinds of cases with which to populate these libraries in order to have maximum impact on learning. The other concerns issues of information organization, presentation and navigation. How can cases be organized and presented in ways that make it easy to access their most interesting parts, understand their implications, and recognize the range of problems and the range of solutions available? While the existing case libraries provide one answer to this question (another, for example, is provided by [9]), we are currently revisiting this issue from the perspective of students, who are novice practitioners. From this perspective we believe that additional capabilities such as access to definitions of the terms used by experts, access to explanations of what experts find it appropriate to focus on, guidance in choosing what to focus on next, and allowing students to extend the libraries (or create new ones) are also required.

Supporting collaborative problem solving and learning: Support for group communication and sharing will be provided by facilitating collaborative work through the software environment. CaMILE was used during the past two quarters in a junior-level design foundations course, taught in mechanical engineering (ME 3110; Creative Decisions in Design). We have collected data on the system's usage and its effects, and are in the process of analyzing this data. A World Wide Web version, WebCaMILE, is also under development. We plan to link case libraries and WebCaMILE so that students engaged in a design activity might use WebCaMILE to discuss and exchange case-study materials. Cases provide the kinds of information that a student might point to as justification for some argument presented to others, as a potential alternative to a design decision, or as a rebuttal to someone else's design decision. Tailoring CaMILE's procedural facilitation to reflect more closely the content and nature of the problems students will be solving and investigating new ways of scaffolding collaboration are other topics of ongoing research.

Software-realized scaffolding: Of particular importance in making this integrated software environment work for students is providing software-realized scaffolding to support student use of the environment for learning. We have identified several specific areas in which we can provide facilitation.

• Scaffolding collaborative design and problem-solving: Our environment will provide scaffolding for design and problem-solving using several techniques:

– By structuring the kinds of entries which can be made in a group discussion, e.g., new theories or ideas, alternatives, comments, rebuttals, and questions. When a student chooses one of these kinds of entries, an editor opens for their comments and a prompting window opens with suggestions for useful entries to make, e.g., for a rebuttal, suggestions might include "The strengths of this idea are..." and "But the key weakness is...". This scaffolding guides the discussion in useful directions defining the kinds of entries to be made, asking students to choose one before entering an item into the discussion, and suggesting appropriate things to say.

– By providing agents to actively review student work and suggest better ways to design and solve problems. For example, agents may identify where connections might be made between efforts, where additional resources exist that might aid an effort, and where efforts may be going astray [6].

– By providing menus of glossaries of relevant vocabulary and their definitions.

– By providing means of visualization and making explicit the design process.

• Scaffolding reflection and learning: We want to support two kinds of reflection in the environment because we believe that reflection can significantly facilitate learning.

– Reflection-in-action: The students' articulations in the discussion, the declaration of item type, and the linking of cases to discussion are all forms of reflection-in-action. These are kinds of reflection which are integral to the design process and which support both the execution of a good design process and the learning about that process. Reflection-in-action helps to make strategies explicit and learnable, develops an expanded repertoire of strategies, and improves student understanding and control of the design process.

– Reflection-as-summary: Student summarization at the end of a design process is an important learning activity for students and an important resource for future groups of students. Our plan is for students to summarize their group design projects such that summaries from one class become cases in the library for the next class. Thus, students summarize not just for their own benefit but to help a future audience.

• Scaffolding resource identification and use: Case libraries support student exploration by providing multiple indices into cases. Students might begin by looking at one case of interest and then explore related cases by a number of different dimensions, or begin by browsing all cases related to a problem. Students can gain perspective on what problems they are facing, what the parameters of the problems are, and how these parameters are explored in the cases in the library from case overviews. We want case libraries to provide support for all these kinds of searching and browsing, but coupled with support that helps in applying the found information to the task at hand (e.g., linking cases that highlight an important alternative solution to the discussion on that alternative). In addition, we envision the use of visualization tools to aid in resource identification and use.

Integration: As many of the critical components of the software environment are being implemented and used in classrooms, the most significant task ahead of us is integrating the different pieces into a single environment. This integrated environment supporting the McBAGEL approach has to play several roles: facilitation of design problem solving and its constituent components, facilitation of learning, access to resources, and access to teachers and fellow learners. The software environment has to serve as both an electronic workspace and a learning environment providing help with a variety of intellectual activities as students collaborate on design projects. We see a need for this environment to promote reflection and summarization as well. Software-guided reflection is particularly important in facilitating skill transfer between different problem domains. The construction of such an environment on Macintosh computers is currently underway.

Assessment: The next step, slated to begin in Fall 1995, is to use and assess both the approach and the concomitant software environment in a series of design courses at Georgia Tech. We will use assessments to determine what kind of learning has occurred and how well students apply what they have learned. The goals of learning involve not merely acquiring a set of static facts to be recalled on a test but rather involve constructing a coherent understanding of a domain that can be flexibly transferred to new situations. The extent to which learning can be used in new situations (i.e., transfer) allows assessment of how flexibly the students have learned the content and are able to apply it to complex problems. Students' learning will be evaluated on mastery, near-transfer, and far-transfer problem-solving. Cognitive research suggests that because problem-based instruction is geared towards complex curricular objectives, assessments need to include open-ended questions in which students explain what approaches they have to a problem and its solution [7]. A variety of methods will be used to collect this data including interviews and paper-and-pencil short answer tests. This allows measurement of the products and processes of the students' learning. Some authentic performance assessments will also be devised. Students' presentations will be assessed to examine how they define the problems and justify their solutions as well as the quality of their solutions. Because transfer is not an all-or-none phenomenon, different types of transfer will be assessed and measures will be developed that assess this. We will use measures of knowledge, skills, planning, and qualitative understanding as students are asked to justify their solutions. This will assess the flexibility of the knowledge that the students construct. For example, because of the emphasis on problem solving, we would expect increased integration of the content they are learning into their problem-solving on transfer problems. Because students are using the collaborative environment and gaining experience and feedback in articulating their plans for problem-solving, we expect improvement in the students' planning skills as well.

6. Conclusions

Collaborative learning environments have the potential for helping students to construct usable knowledge and to learn strategies that prepare them for a lifetime of learning. To afford generative learning, such environments need to contain rich sources of information. In addition, opportunities for student collaboration, articulation and reflection must be provided to help students think deeply about the problems they are working on and to learn to go beyond the given problems. The McBAGEL approach is designed to meet these requirements. Providing computational support to this approach requires the design of a software architecture that integrates multiple tools and information resources with a structured electronic workspace. This paper describes our efforts on developing the theoretical and practical aspects of such an architecture. The focus of our current research is on refining and testing the components further, and on fully implementing the integrated environment. Future research will focus on deploying it in classrooms and conducting assessments of its impact on student learning.

Acknowledgments

Research reported here has been supported by ARPA under contract N00014-91-J-4092 monitored by ONR, ONR under contract N00014-92-J-1234, and by the Woodruff Foundation's support of the EduTech Institute.

References

1. Blumenfeld, P., Soloway, E., Marx, R., Krajcik, J., Guzdial, M., and Palincsar A. 1991. Motivation project-based learning: Sustaining the doing, supporting the learning. Educational Psychologist, Vol. 26, Nos. 3&4, pp. 369-398.

2. Bransford, J. D., and Johnson, M.K. 1972. Contextual prerequisites for understanding: Some investigations of comprehension and recall, Journal of Verbal Learning and Verbal Behavior, Vol. 11, pp. 717-726.

3. Brown, A. L., and Palincsar., A. S. 1989. Guided, cooperative learning and individual knowledge acquisition. In Knowing, learning, and instruction: Essays in honor of Robert Glaser, L. B. Resnick (Ed.). Erlbaum, Hillsdale, NJ , pp. 393-451.

4. Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., and Glaser, R. 1989. Self-explanations: How students study and use examples in learning to solve problems. Cognitive Science, Vol. 13, pp. 145-182.

5. Domeshek, E.A., and Kolodner, J.L. 1992. A case-based design aid for architecture. In Proceedings of the Second International Conference on Artificial Intelligence and Design.

6. Fischer, G., Lemke, A. C., Mastaglio, T., and Morch, A. I. 1991. The role of critiquing in cooperative problem solving. ACM Transactions on Information Systems, Vol. 9, No. 3, pp. 123-151.

7. Glaser, R., and Silver, E. 1994. Assessment, testing, and instruction: Retrospect and prospect. In Review of Research in Education, Vol. 20. Darling-Hammond (Ed.). American Educational Research Association, Washington D. C., pp. 393-419.

8. Guzdial, M., Rappin, N., and Carlson, D. 1995. Collaborative and multimedia interactive learning environment for engineering education. In Proceedings of the ACM Symposium on Applied Computing, pp. 5-9.

9. Hsi, S., and Agogino, A. M. 1994. The impact and instructional benefit of using multimedia case studies to teach engineering design. Journal of Educational Multimedia and Hypermedia, Vol. 3, Nos. 3/4, pp. 351-376.

10. King, A. 1992. Comparison of self-questioning, summarizing, and notetaking-review as strategies for learning from lectures. American Educational Research Journal, Vol. 29, pp. 303-323.

11. Kolodner, J. 1993. Case-Based Reasoning. Morgan Kaufmann, San Mateo, CA.

12. Koschmann, T. D., Myers, A. C., Feltovich, P. J., and Barrows, H. S. 1994. Using technology to assist in realizing effective learning and instruction: A principled approach to the use of computers in collaborative learning. Journal of the Learning Sciences, Vol. 3, pp. 225-262.

13. Lin, X. 1994. Far-transfer problem-solving in hypermedia environment: The role of self-regulated learning processes. Paper presented at the American Educational Research Association Annual Meeting, New Orleans.

14. Pea, R. D. 1994. Practices of distributed intelligence and designs for education. In Distributed Cognitions, G. Salomon and D. Perkins (Eds.). Cambridge University Press.

15. Pirolli, P., and Recker, M. 1994. Learning strategies and transfer in the domain of programming. Cognition and Instruction, Vol. 12, pp. 235-275.

16. Rappin, N., Guzdial, M., Ludovice, P., and Realff, M. 1995. DEVICE: Dynamic environment for visualizations in chemical engineering. In Proceedings of the AI-Education Conference, (In Press).

17. Riel, M. 1993. Global discourse via electronic networking. Paper presented at the Annual Meeting of the American Educational Research Association, Atlanta, GA.

18. Scardamalia, M., Bereiter, C., McLean, R. S., Swallow, J., and Woodruff, E. 1989. Computer-Supported Intentional Learning Environments. Journal of Educational Computing Research, Vol. 5, pp. 51-68.

19. Schmidt, H. G., DeGrave, W. S., DeVolder, M. L., Moust, J. H. C., and Patel, V. L. 1989. Explanatory models in the processing of science text: The role of prior knowledge activation through small group discussion, Journal of Educational Psychology, Vol. 81, pp. 610-619.

20. Spiro, R. J., Coulson, R. L., Feltovich, P. J., and Anderson, D. K. 1988. Cognitive flexibility: Advanced knowledge acquisition in ill-structured domains. In Proceedings of the Tenth Annual Conference of the Cognitive Science Society, Erlbaum, Hillsdale, NJ, pp. 375-383.

21. White, B. 1984. Designing computer activities to help physics student understand Newton's laws of motion. Cognition and Instruction, Vol. 1, pp. 69-108.

22. Williams, S. M., Bransford, J. D., Vye, N. J., Goldman, S. R., and Carlson, K. 1993. Positive and negative effects of specific knowledge on mathematical problem-solving. Paper presented at the American Educational Research Association Annual meeting, Atlanta GA.

Authors' Addresses

N. Hari Narayanan, Cindy E. Hmelo, Valery Petrushin, Wendy C. Newstetter, Mark Guzdial and Janet L. Kolodner: EduTech Institute, College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332-0280. {narayan, ceh, petr, wendy, guzdial, jlk}@cc.gatech.edu.