Integrating and Guiding Collaboration: Lessons Learned in Computer-Supported Collaborative Learning Research at Georgia Tech

Mark Guzdial,* Cindy Hmelo,† Roland Hübscher,* Kris Nagel,* Wendy Newstetter,* Sadhana Puntambekar,* Amnon Shabo,* Jennifer Turns,§ and Janet L. Kolodner*

** EduTech Institute, College of Computing Georgia Institute of Technology, † Learning Research and Development Center University of Pittsburgh, § EduTech Institute, Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology*

Directions of Progress: Integrating and Guiding

Research on computer-supported collaborative learning at the EduTech Institute at Georgia Tech is relatively new at less than five years old. In that short span, however, the team of researchers has developed eight different prototype software systems and evaluated them in a variety of settings with middle school, undergraduate, and graduate students. In this paper, we reflect on some of the overall lessons learned through this work.

We have developed an approach to instruction that involves students' learning through solving real-world problems. We help students manage the complexity of these problems in several ways. First, we use the methodology from problem-based learning (PBL) to help students structure their problem-solving. Second, we provide students with case libraries to help them learn from the experience of others. Third, we provide support for student reflection. Part of these ideas developed out of the close fit that we saw between the PBL educational methodology and the cognitive model derived from case-based reasoning [1]. Later work has focused on merging what we know about the design process with the PBL classroom methodology.

In PBL, students learn by solving authentic real-world problems and reflecting on their experiences. For example, engineering students learn how to design as they try to design kites and kiosks. Because these problems are complex, students work in groups, where they pool their knowledge and together grapple with the issues that must be considered. Facilitators guide student reflection on their problem-solving experiences, asking students to articulate both the concepts and skills they are learning, those they still need to learn more about and helping them identify the strategies needed for problem solving, collaboration, and self-directed learning.

Case-based reasoning (CBR) refers to reasoning based on previous experiences [2]. It might mean solving a new problem by adapting an old solution or merging pieces of several old solutions, interpreting a new situation in light of similar situations, or projecting the effects of a new situation by examining the effects of a similar old situation. Learning, in the CBR paradigm, means extending one's knowledge by incorporating new experiences into memory, by re-indexing old experiences to make them more accessible, and by abstracting out generalizations from experiences. Thus, a major issue CBR addresses is identifying old situations that are relevant to a new one. Reflection is one mechanism to help people index their experiences in meaningful ways.

---

* EduTech Institute, College of Computing Georgia Institute of Technology 801 Atlantic Drive Atlanta, GA 30332-0280, *{guzdial, roland, kris, wendy, sadhana, amnonh, jlk}@cc.gatech.edu*

† Learning Research and Development Center University of Pittsburgh, hmelo@pop.pitt.edu

§ EduTech Institute Center for Human-Machine Systems Research School of Industrial and Systems Engineering Georgia Institute of Technology, jennifer@isye.gatech.edu

CBR offers suggestions for practice that may enhance the effectiveness PBL [1,8]. Both point toward a constructivist mode of education, in which individuals learn from problem-solving. Each focuses on different but complementary aspects of the experience. The methodology of PBL asks students to solve problems and then reflect on what they have learned from the experience.

A graphical representation of the systems and lines of research appears in Figure 1. At a very high-level, we can identify three lines of research:

- • **Case-Based Reasoning Driven Tools**: One line of research builds upon the CBR-inspired PBL methodology described above. One of our earliest CSCL goals was to create environments that supported problem-based learning.

- • **Collaboration-Driven Tools**: A second line of research was inspired by the success of the OISE team lead by Marlene Scardamalia and Carl Bereiter with CSILE [3-5]. Our goal was to provide the same kind of knowledge-building environment in an undergraduate engineering context.

- • **Reflection and Process-Driven Tools**: A third and more mature line of research has arisen from the other two. We are now interested not just in problem-solving or collaboration, but in integrating different phases of the students' problem-solving and design process and providing scaffolding (guidance) which includes reflection, choosing between alternatives, and connecting learning goals with activities.

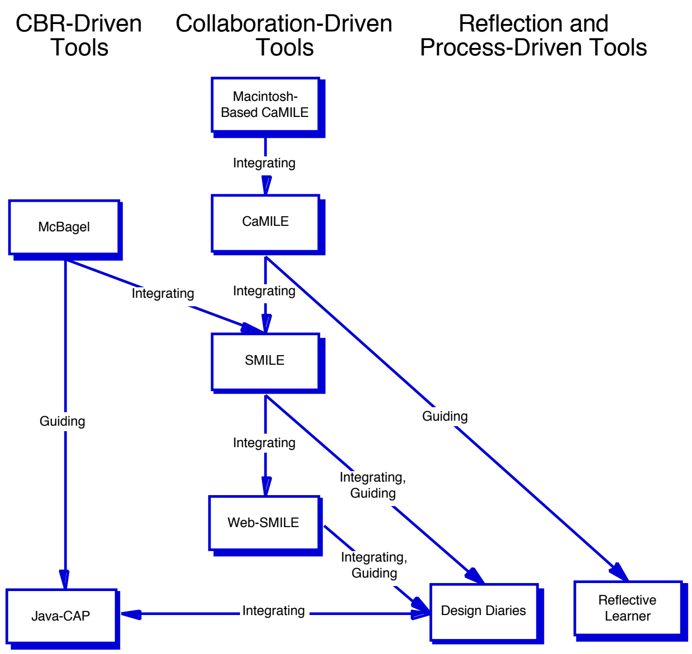

Figure 1: Tracing the Lines of Research and Lessons in CSCL Environments

At the highest level, there are two main lessons that we have learned from our work:

- • Integrate, integrate, integrate: Students don't want to use five different tools for ten different tasks. They might be willing to use five different tools if the connection between those tools makes sense and provides a real affordance on the ten different tasks—and those tasks are worthwhile and make sense! We have learned that it is too easy to create yet another technological solution for each task or problem that arises. It is more difficult but much more promising to broaden one's view of integration and to understand how a single approach can be realized in multiple mechanisms to meet a variety of performance and learning goals.

- • Students need guidance: It's a simple, almost obvious lesson that students need guidance, but it becomes complicated as one realizes the levels at which guidance is needed. For example, we have found that students can come up with ideas for design problems, and they can do research. However, they do not naturally do their research on the ideas for their design problems—those connections are often missing. Certainly, teachers can make those connections for students, but the software can help with making those connections, which is especially important when one teacher is spread across a whole classroom of students. We have found it very useful to suggest connections to students, to point out what it is useful to reflect upon and why, and to provide a space for that reflection.

In the following sections, we describe how each of these lessons has influenced the generations of systems-development and evaluation in each of these three lines of research.

I. Line of Process: CBR-Based Tools

A. McBAGEL and Early Case Libraries: Synchronous Collaboration and Information Resources for Problem-Based Learning

The McBAGEL (Multiple Case-Based Approach to GEnerative Environments for Learning) software arose out of the synthesis of PBL and CBR. In PBL, students use whiteboards to record their observations on the problem, their ideas for solution, the learning issues that need to be addressed to apply the ideas, and the students' action plans. The PBL whiteboards help scaffold the students' problem solving by communicating the PBL process as well as serving as an external memory aid [6-8]. One of the drawbacks of the traditional whiteboards is that they are ephemeral; students write on them and then the record of their deliberations is gone. Students are not able to go back to their previous experiences and consider how their earlier problem-solving might bear on their current situation. Borrowing the whiteboard workspaces of PBL, McBAGEL supports synchronous collaboration with an electronic workspace structured as a whiteboard (Figure 2). We envisioned students working in small groups around McBAGEL whiteboards, conducting their research and problem-solving activities, and using McBAGEL as a record-keeping aid. In addition, McBAGEL provided loose integration of software to other tools and resources of use in the students' problem-solving process.

One of the tools that students had access to through McBAGEL was an early version of the case libraries. In the middle school classrooms in which we trialed a design-based curriculum, we provided case libraries of designs and experiments that were rich in data relevant to their design activities. We reasoned that since novice designers were *unlikely* to have their own relevant experience, we would provide them with a set of expert designs to use in their efforts [1, 7].

The way that McBAGEL was used in a sixth grade life science classroom suggested that our model of collaborative activity and reflection needed to be refined. Just keeping records was not sufficient for the middle school students. They needed more help coming up with ideas, choosing between alternatives, and planning their projects. The students sometimes worked together in their small groups but they also added items individually at different times—suggesting the need to support asynchronous collaboration as well. Moreover, they needed to have a mechanism that would support publishing their individual whiteboards for discussion among the whole class. In addition, we needed to help (1) students reflect in ways that helped them look at the big picture and not just the activities they were currently working on, (2) provide scaffolding to help the students in their problem-solving and design, and (3) provide tools that would assist the students in generating ideas, choosing alternatives, and planning their projects.

The kind of scaffolding that was needed has taught us important lessons in how to design the later case library, JavaCAP (next section). The children did not find the case library very interesting. The children were not interested in the long sections of on-screen text describing design functionality, which suggested that the text needed to be kept short with accompanying graphical representations. The students also had difficulty understanding the relevance of the cases in the library. We decided that having students develop their own cases might help them better understand the cases. Creating a published case would probably be motivating for the students, and might help the students reflect on what they learned as they were working on a design problem.

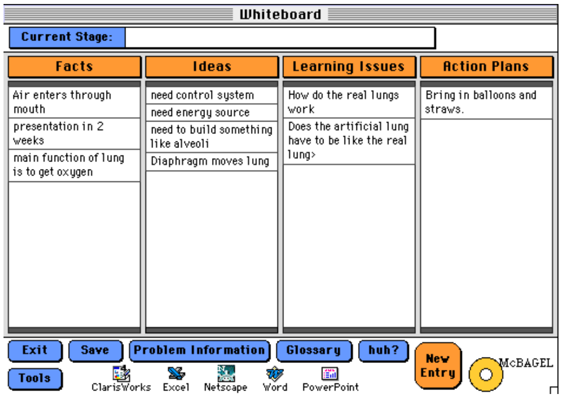

Figure 2: McBagel Prototype

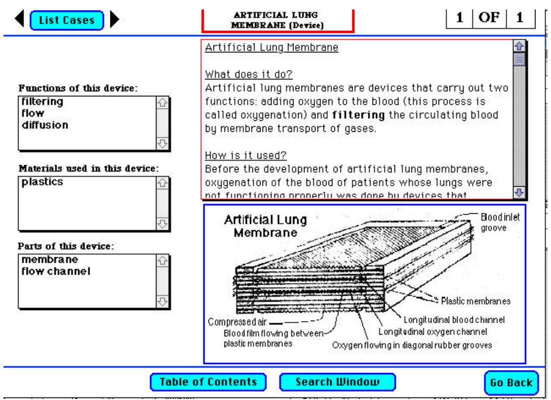

Figure 3: Early Case Library

B. SMILE: Integrating Problem-Based Learning with Collaboration Supports

Based on our experiences in the classroom with McBAGEL and CaMILE (described below in the Collaboration line of research), we designed a new learning environment, SMILE (Scaffolded Multi-user Integrated Learning Environment), integrating synchronous and asynchronous collaboration as well as scaffolding the learning process and the appropriate use of the tools.

Like McBAGEL, SMILE provides a shared space for record keeping, integrated access to support tools and workspaces, and a framework for attaching software-realized scaffolding. In both collaboration tools students create notes and organize them. In the synchronous tool McBAGEL, the notes are organized in the four columns. In the asynchronous tool CaMILE, the notes are structured according to the flow of discussion. Both kinds of workspaces play a role in student work, and in an integrated tool, these two workspaces should not be treated as supporting two completely distinct collaborations.

Instead of presenting two different, yet highly integrated tools to the students, we decided to provide a more uniform view. In both types of collaboration, students in groups or as individuals create and organize notes. Thus, SMILE's collaboration tools present themselves as spaces of notes that are organized according to the task: synchronous collaboration supports the PBL-style columns and asynchronous collaboration presents the notes hierarchically as in CaMILE.

Since the action column in McBAGEL's whiteboard is not really a part of the knowledge–integration process, it is removed from the synchronous collaboration tool in SMILE. Instead, a separate planner is provided in SMILE that integrates with process support (discussed further below). In a sense, we moved planning from the case analysis activity into the scaffolding of the whole activity.

C. JavaCAP: Guiding Students to Reflect and Record

JavaCAP (Java-based Case Authoring Program) evolved from case libraries developed for engineering education and from our early experience with middle school students. Having a case library written by the students for other students seemed like a good way to (1) promote student reflection, (2) help the students understand the value of cases, and (3) provide the lessons that other novice designers had learned.

In JavaCAP, we added an authoring section to encourage student reflection on their learning as they construct their own cases. As students enter a case through a Web browser, they need:

- • to summarize their experiences and what they learned from them;
- • to index their cases, which requires that they consider how others might use them;
- • to find interesting cases for comparison and connecting, which requires students concretely to describe and redescribe the problem they have been trying to solve.

Using the case library in both ways fosters backward and forward reasoning that promotes transfer [9]. The students are provided with prompts that ask them to summarize their experiences and the lessons learned from those experiences.

The scenes-from-a-play metaphor was chosen to provide students with a more familiar structure that would help them capture the sequencing of their experience. Students can upload multimedia elements to accompany their scene descriptions. At the end of the process they can publish their case on the Web.

The scenes are: the *Problem Presentations* scene; the *Alternatives Selection* scene; the *Solution* scene; and the *It's a* Wrap scene (Figure 4).

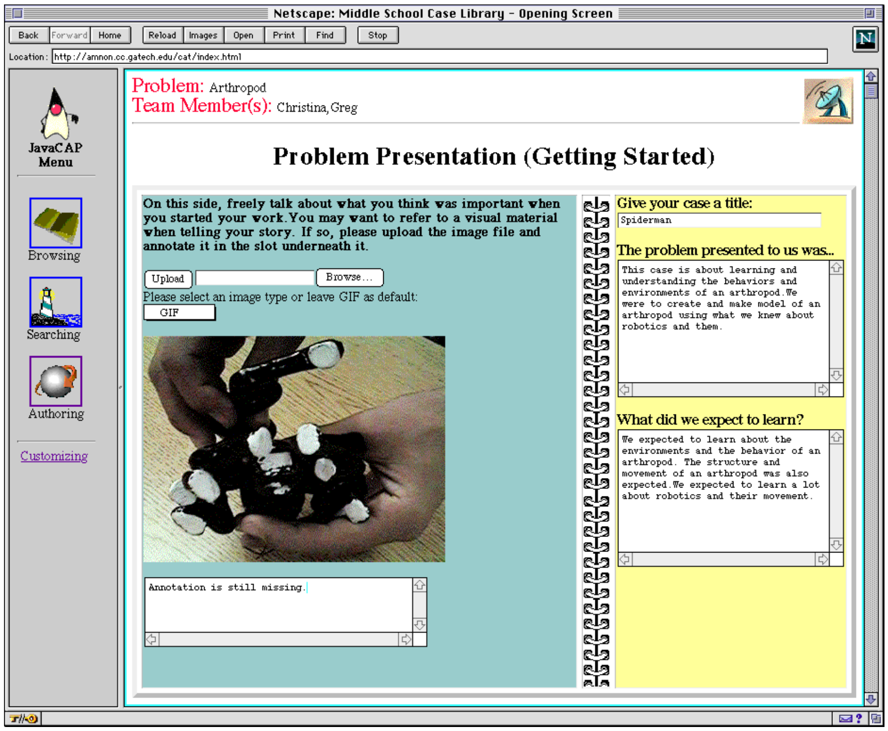

Figure 4: A scene page in JavaCAP where a student working by her own, filled out her point of view of her group's initial effort to cope with the Arthropod problem (within the EduTech Learning-by-Design curriculum for middle schools).

II. Line of Process: Collaboration-Driven Tools

A. Macintosh-Based CaMILE: Facilitating Collaboration on an Internet-Based Tool

CaMILE (Collaborative and Multimedia Interactive Learning Environment) was inspired by the work of Scardamalia and Bereiter in creating CSILE [3, 5]. CSILE facilitated the creation of a knowledge-building community, where students were given greater agency to drive the learning process [4]. CSILE scaffolded collaboration by providing a structure (asynchronous threaded discussions, where notes directly commenting upon another are displayed as related).

We and our collaborating engineering faculty wanted a CSILE-like tool to facilitate collaboration among engineering students, especially those working asynchronously on group design projects. Notes were threaded, and we chose note types and suggestions to be like those in CSILE and featured tight integration with multimedia. Students could link media (sound, video, spreadsheet documents, etc.) into the margins of their notes, providing a mechanism to share their design artifacts, such as sketches, simulations, and design analyses.

We tested CaMILE in several design classes [10, 11] with mixed results. While students remarked that they found CaMILE to be useful for some activities, there were a number of disappointments in its use.

- • It provided no affordance to the students because students saw each other frequently for other reasons.

- • Multimedia composition was rarely used. In general, students decomposed their problem to limit the number of artifacts that they needed to share [12].

- • It was slow. CaMILE was based on pre-Web networking technology, and a single discussion might take literally minutes to download.

- • Finally, being Macintosh-based was a definite disincentive to students who owned Windows-based computers.

B. WebCaMILE: Anchoring Collaboration in a Web-Based Context

Our revision to CaMILE was inspired by our findings that, when students collaborated face-to-face, they often worked around an artifact (such as a design report or a set of specifications) to ground or anchor their discussion. The new version of CaMILE, WebCaMILE, was designed to structure collaborations around anchoring artifacts, that is, *anchored collaboration*.

WebCaMILE maintained the scaffolding for collaboration from CaMILE. But several key characteristics changed:

- • WebCaMILE was implemented on the Web, which improved its speed and enabled its use across platforms.

- • Early versions of WebCaMILE did provide multimedia annotations, like in CaMILE, but when these still were never used, they were dropped.

- • Perhaps the most significant advantage of WebCaMILE over CaMILE is that individual notes (and their consequent threads of discussions) are directly addressable as links on the Web. This enables the creation of single-click access from a Web page anchor (e.g., a design report to discuss) to a thread for discussion of that anchor. We refer to this use of Web pages as anchors for discussion as anchored collaboration.

WebCaMILE has been much more successful than CaMILE was. Students are much more willing to use WebCaMILE to discuss anchors of interest. In comparison of anchored WebCaMILE discussions with unanchored newsgroup discussions in the same class, WebCaMILE discussion threads are longer, which suggests that WebCaMILE is supporting more sustained discussion than in a newsgroup [13].

C. SMILE: Integrating Process and Collaboration Support

A computer-based learning environment has only a chance of being useful in the classroom if the technology and the class room are well integrated with each other. We believe that the learning process is the best place to interface the software and the class room. Our goal and the main task of SMILE is to support the students' learning and designing process by scaffolding the process. Thus, we view SMILE mainly as a process scaffolding tool that employs tools like McBAGEL and CaMILE to support scaffolding.

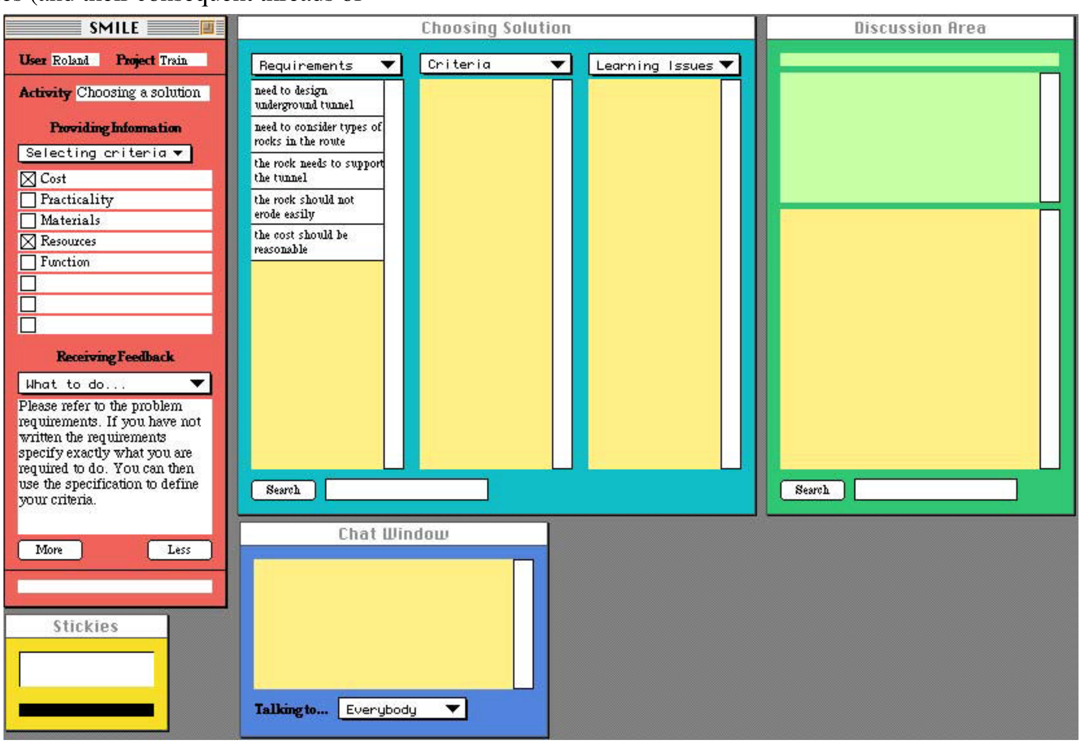

Figure 5: SMILE with Scaffolder (with Planning support), Whiteboards (like McBAGEL), and Discussion Area (like CaMILE) Visible

In designing SMILE (Figure 5), we put much greater emphasis into scaffolding the process and not just the separate activities. This kind of scaffolding supports the student in choosing the right next activity. Thus, an activity happens for a reason and within a context of other activities. By making the scaffolding more fine-grained and less abstract, it changes at lower levels into scaffolding activities and even actions.

D. Web-SMILE: Providing SMILE-like Functionality on the Web

Web-SMILE was developed as a prototype to test the approach of SMILE on a small scale in actual classroom use. It has become a tool of significant applicability within middle school classrooms where it has enabled the teacher to facilitate the collaborative process, even with large classes. Like SMILE, Web-SMILE integrates the McBAGEL whiteboards and CaMILE threaded discussions in one collaborative environment, but overall structured in terms of scaffolding of the problem solving process. Unique to Web-SMILE is the explicit use of the steps in the problem solving process to guide the use of the integrated tools. The lessons learned from this combination of tools will be used for the further development of SMILE.

Web-SMILE's seamless environment offers different interfaces to the common database of items recorded from either tool. This integration of tools is part of the process-oriented view of student support in Web-SMILE. Instead of prompting for a tool's use, the environment asks, “Where are you in solving the problem?” Based on the student selected step of the process, activities are suggested, with the tools accessible by a single click.

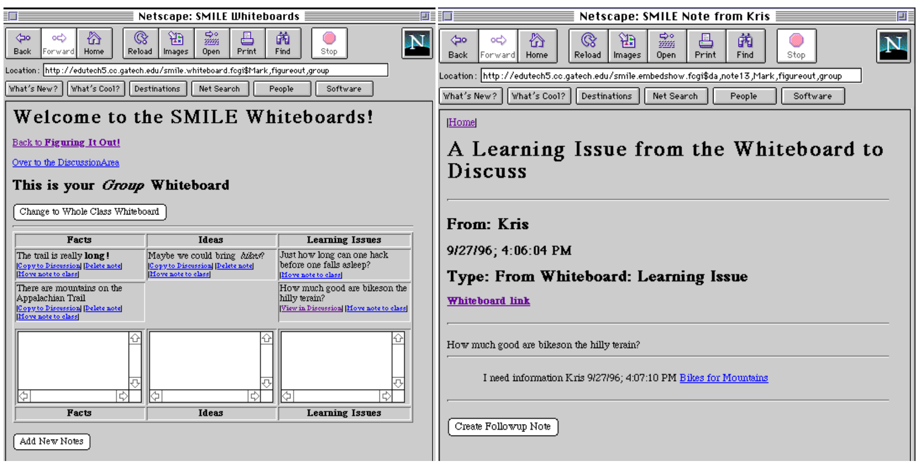

Figure 6: Web-SMILE’s Whiteboards (left) and linked Discussion Area (right)

III. Line of Process: Reflection and Process-Driven Tools

A. Reflective Learner: Guiding Students through the Process of Reflection

To focus student attention on collaboration as a vehicle for learning, we have experimented with scaffolding student reflection through writing of various kinds. We focused early efforts on unscaffolded learning essays, assigning students to write an essay reflecting on what they have learned through each major class assignment. We found that, with this activity, some students understand the goal of the activity, have no difficulties in carrying it out, and write quite insightful essays. Other students treat it more as a high school composition-type exercise than a real opportunity to think back on what has happened and how that might be an important learning lesson.

In an effort to bridge the discrepancy between the potential and realized value of this activity as it existed, we developed two supports for scaffolding the reflective writing: a writing framework, *the learning vignette*, and a piece of software, the *Reflective Learner*. In a learning vignette, students are not just supposed to reflect on what they have learned, but specifically to document their efforts at observing the goings-on around them as they collaborated, extracting high level principles from those observations, finding instances where such principles may have applied to past experiences and projecting where those principles might be applicable in future applications.

A major feature of the reflective learner is the support provided for developing each portion of the learning vignette—scaffolded editing (Figure 7). The questions guide the development of a written learning vignette. Students have the option of using the different screens as guidance in developing learning essays or in writing free form essays without prompts. Students submit these essays electronically. Subsequently, the instructor is able to embed comments directly in the essays, provide some summary comments, and assign an overall rating from a customizable list.

From the extensive use of the software, we have learned the following lessons:

- • Students can successfully write learning vignettes. With support, students can write meaningful and insightful learning vignettes. Additionally, students recognize that these writing assignments help them focus on what they are learning from their experiences.

- • The software does support the students with the difficult portions of the activity. Students have overwhelmingly reported that the software is easy to use and useful and that they make use of most of the scaffolding features. Further, the quality of the vignettes has improved over previous attempts to orchestrate the activity in classes without the software as has the students understanding of the role of the activity.

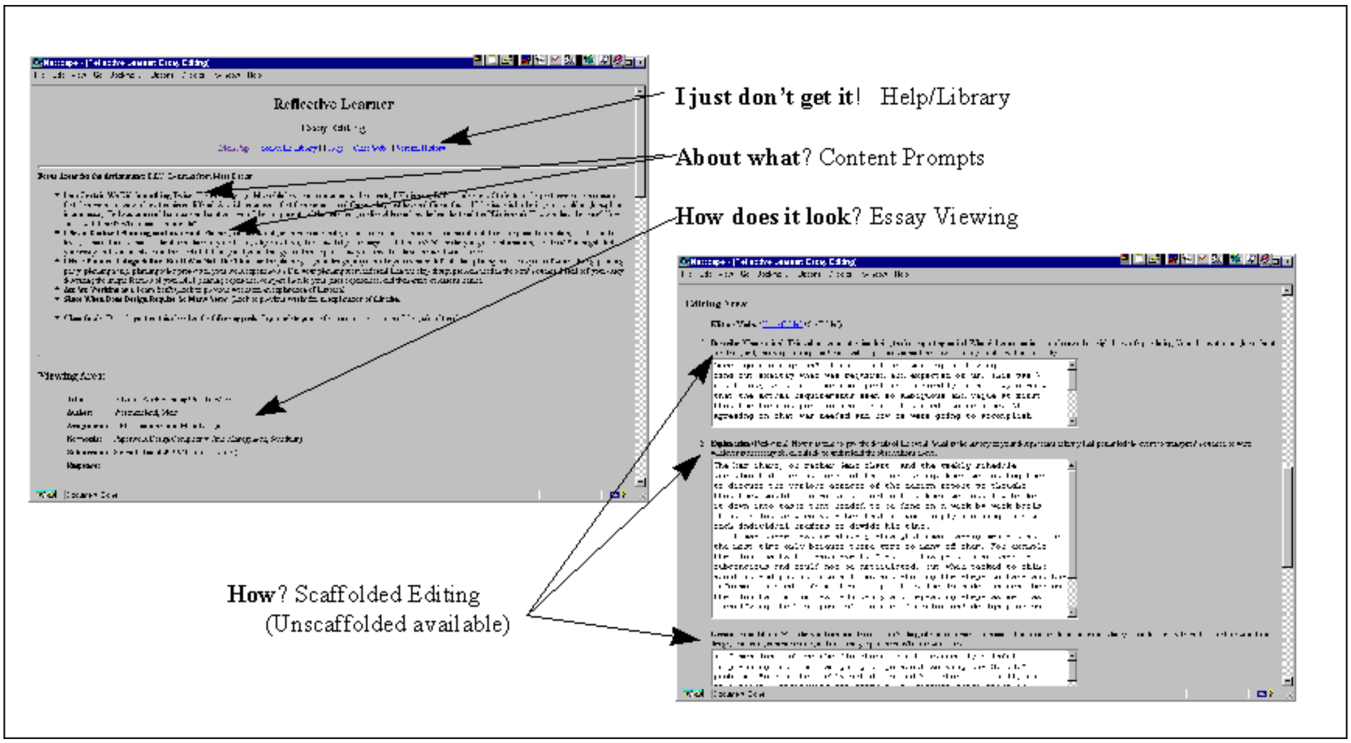

Figure 7. Editing a Learning Vignette in the Reflective Learner

B. Design Diaries: Guiding Students through Design Process Including Reflection

In our early implementations of our middle school curricula, we found that students needed explicit guidance in such activities as understanding the problem, carrying out the research, evaluating alternatives and in integrating the science with their design activities. Based on the findings from these studies, design journals were introduced which are a form of paper and pencil tool providing guidance with these activities

The design journals provide support for the students through the different stages of the design process. They contain prompts in the form of questions that help students to think about the processes or activities that they would engage in. They also aim to help students reflect on and articulate their design related activities. The journals prompt them to make several decisions along the project such as what they needed to learn in order to design, what materials they would use and why, how they would evaluate the possible solutions and whether they want to revise their models and if so, why and how.

The journals were used to support problems in earth life science curricula. We found that students needed at a more fine grained level than was provided in the journals. More importantly, they needed support to understand the links between the activities in the design process and to actively monitor their learning. Many students could not relate the research they did to their designs, they concentrated on the structural aspects of their designs and did not see the relations between the structure and the function. They tended to view the paper tool as worksheets and did not use them dynamically to guide their learning.

We are considering integrating the Design Diaries in a structure with SMILE and JavaCAP. In a software implementation, we can prompt for links between activities and components of their designs. Further, we can integrate the Design Diaries activity into the scaffolding structure of SMILE.

Conclusion

In the course of this work, we have learned several things about supporting collaborative learning on-line – integration and providing guidance. We can further decompose integration into four pieces:

1. about integrating software into the same environment,
2. about integrating software and classroom activities,
3. about thinking through the integration of lines of collaboration tools (e.g., JavaCAP is both a CBR tool and a reflection tool), and
4. integrating support for both group and individual work.

In a sense, "students need guidance" is almost a subset of the integration issue. When do students need what and for what? Integrating all of the pieces of problem-solving makes the pieces consistent and easily usable.

There are, of course, many lessons that we could have listed here. Design of good collaborative tools takes time, we have found it useful to explore across a wide area of possibilities, and students need input from a variety of agents and perspectives. An even longer story can be told about our efforts to integrate our tools and curriculum through teacher development efforts. Working in the ivory tower, or even always telling teachers what we were going to provide without getting input from them about what is needed and how they would use, it wouldn't have worked as well as what we did end up doing—and we wish that our collaborations with our teachers could have been stronger than they were!

In summary, our lesson is that successful collaborative problem-solving and learning is not easy nor does it happen by accident. In every line of research, our first solution was not at all the right solution. But through the mistakes of previous generations, we are finding mechanisms to support student learning in a collaborative setting.

Acknowledgments

The work at the EduTech Institute in CSCL has been supported by the Woodruff Foundation, the ARPA CAETI program under contract N66001-95-C-8608, the McDonnell Foundation, and the National Science Foundation under grant ESI-9553583.

References

[1] J. L. Kolodner, "Educational implications of analogy: A view from case-based reasoning," American Psychologist, vol. 52, pp. 57-66, 1997.

[2] J. L. Kolodner, Case Based Reasoning. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[3] M. Scardamalia, C. Bereiter, R. McLean, J. Swallow, and E. Woodruff, "Computer-supported intentional learning environments," Journal of Educational Computing Research, vol. 5, pp. 51-68, 1989.

[4] M. Scardamalia and C. Bereiter, "Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media," Journal of the Learning Sciences, vol. 1, pp. 37-68, 1991.

[5] M. Scardamalia, C. Bereiter, and M. Lamon, "The CSILE Project: Trying to bring the classroom into World 3," in Classroom Lessons: Integrating Cognitive Theory and Classroom Practice, K. McGilly, Ed. Cambridge, Mass.: MIT Press, 1994, pp. 201-228.

[6] M. Guzdial, J. L. Kolodner, C. Hmelo, H. Narayanan, D. Carlson, N. Rappin, R. Hübscher, J. Turns, and W. Newstetter, "Computer support for learning through complex problem-solving," Communications of the ACM, vol. 39, pp. 43-45, 1996.

[7] C. Hmelo, H. N. Narayanan, R. Hübscher, C. W. Newstetter, and J. L. Kolodner, "A multiple-case-based approach to generative environments for learning," VIVEK: A Quarterly in Artificial Intelligence, vol. 9, pp. 2-18, 1996.

[8] J. L. Kolodner, C. E. Hmelo, and N. H. Narayanan, "Problem-based learning meets case-

based reasoning," presented at International Conference on the Learning Sciences, Northwestern University, 1996.

[9] G. Salomon, D. Perkins, and T. Globerson, "Partners in cognition: Extending human intelligence with intelligent technologies," Educational Researcher, vol. 20, pp. 2-9, 1991.

[10] M. Guzdial, J. Turns, N. Rappin, and D. Carlson, "Collaborative support for learning in complex domains," in Computer Support for Collaborative Learning (CSCL '95), J. L. Schnase and E. L. Cunnius, Eds. Bloomington, IN: Lawrence Erlbaum Associates, 1995, pp. 157-160.

[11] C. E. Hmelo, J. A. Vanegas, M. Realff, B. Bras, J. Mulholland, T. Shikano, and M. Guzdial, "Technology support for collaboration in a problem-based curriculum for sustainable technology," in Computer Support for Collaborative Learning (CSCL '95), J. L. Schnase and E. L. Cunnius, Eds. Bloomington, IN: Lawrence Erlbaum Associates, 1995, pp. 169-172.

[12] W. C. Newstetter and C. E. Hmelo, "Distributing cognition or how they don't: An investigation of student collaborative learning," in Proceedings of ICLS'96, D. C. Edelson and E. A. Domeshek, Eds. Evanston, IL: AACE, 1996, pp. 462-467.

[13] M. Guzdial and J. Turns, "Technological Support for Anchored Collaboration," Draft, 1997