HANDBOOK OF RESEARCH ON EDUCATIONAL COMMUNICATIONS AND TECHNOLOGY

SECOND EDITION

A Project of the Association for Educational Communications and Technology

EDITED BY

DAVID H. JONASSEN

University of Missouri

LEA LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
2004 Mahwah, New Jersey London

Director, Editorial: Lane Akers
Assistant Editor: Lori Hawver
Cover Design: Kathryn Houghtaling Lacey
Textbook Production Manager: Paul Smolenski
Full-Service Compositor: TechBooks
Text and Cover Printer: Victor Graphics, Inc.

This book was typeset in 9/11 pt. ITC Garamond Roman, Bold, and Italic.
The heads were typeset in Novarese, Novarese Medium, and Novarese Bold Italic.

Lawrence Erlbaum Associates, Inc., Publishers
10 Industrial Avenue
Mahwah, New Jersey 07430
www.erlbaum.com

Library of Congress Cataloging-in-Publication Data

Handbook of research for educational communications and technology / edited by David H. Jonassen.—2nd ed.
p. cm.
"A project of the Association for Educational Communications and Technology."
ISBN 0-8058-4145-8
1. Educational technology—Research—Handbooks, manuals, etc. 2. Communication in education—Research—Handbooks, manuals, etc. 3. Telecommunication in education—Research—Handbooks, manuals, etc. 4. Instructional systems—Design—Research—Handbooks, manuals, etc. I. Jonassen, David H., 1947- II. Association for Educational Communications and Technology.

LB1028.3 .H355 2004
371.33'072—dc22

2003015730

Books published by Lawrence Erlbaum Associates are printed on acid-free paper, and their bindings are chosen for strength and durability.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

32

CASE-BASED LEARNING AIDS

Janet L. Kolodner, Jakita N. Owensby, and Mark Guzdial
Georgia Institute of Technology

32.1 WHAT IS A CASE-BASED LEARNING AID?

A case-based learning aid is a support that helps learners interpret, reflect on, and apply experiences—their own or those of someone else—in such a way that valuable learning takes place. Case-based learning aids have cases at their core. The creation and importance of case-based learning aids arose out of work done in two disciplines—work in computer science on case-based reasoning (CBR) and work in education on constructivist approaches to education.

CBR, inspired by people, was developed as a model for creating intelligent systems—systems that could reason by reference to their previous experiences. Such systems, it was conjectured, had the potential to behave more like real experts than could traditional expert systems. Reasoning based on experience would allow them to be more flexible and less brittle than rule-based systems, and with learning from experience built into their architectures, they would become more capable over time (Hammond, 1989; Kolodner & Simpson, 1989; Schank, 1982). Many experimental automated case-based reasoners have been created (see the lists, e.g., in Kolodner, 1993), and indeed, CBR has proven to be quite a useful technology. More interesting to education, however, are the implications CBR holds as a model of cognition—implications about what it means to be a learner and implications about learning and education.

CBR is a special kind of analogical reasoning. A previous experience might suggest a solution to a new problem or a way of interpreting a situation, may warn of a problem that will arise, or may allow the potential effects of a proposed solution to be predicted. CBR has as its core (a) analogy in the context of solving real-world problems and understanding real-world situations and (b) research methodology of computational modeling, aimed at deriving hypotheses about cognition. Whereas analogical reasoning focuses on analogy as a single reasoning method, put into play when a rule-based approach is failing, CBR sees analogical reasoning as the centerpiece of our ability to function as human beings. It posits that our most natural and powerful learning strategies are the automatic ones that situate learning in real-world experience. According to CBR's model, we naturally bring our previous experience and knowledge to bear in interpreting new situations we encounter; we naturally try to explain when things are not as expected (based on the predictions made by our previous experiences and knowledge); we naturally draw conclusions based on explanations and on similarities between situations; and once we draw conclusions, we naturally anticipate, at least a little bit, when this new thing we learned might be applicable. To be able to do all these things so automatically, we must also have some internal processes and representations that allow a new experience to call up similar ones from memory.

CBR also helps us understand how we might develop expertise and how an expert uses his or her own experiences and those of others to reason and learn. Consider, for example, an architect designing an office building. She calls on her experiences and those of others who have designed buildings that address similar needs to make decisions about how to proceed. She knows that many modern office buildings have atriums. Should this new building have an atrium? To answer that, she first looks at the reasons for including atriums in those buildings. In some, it was to provide light to inside offices; in others, to provide a friendly informal space to meet. Are those goals in the new design? They are, but she wonders whether the noise of a central meeting space might be problematic. She examines those buildings again, looking at the effects of the atriums on use of its offices. Indeed, some did cause too much noise, but others were quite successful. Why did some succeed and some fail? The architect looks to see the reasons for failures. Will they be present in the new building? If so, is there a way to avoid the failure by doing it another way (perhaps suggested by one of the successful atria), or should an atrium not be used?

CBR suggests the kinds of content we should extract from our experiences to be able to reuse effectively what we can learn from them, and the kinds of reflection that are effective for doing this, suggesting several critical processes that promote good transfer (Kolodner, 1993, 1997). In particular, CBR suggests five important facilitators for learning effectively from hands-on activities and vicarious experiences: (a) having the kinds of experiences that afford learning what needs to be learned; (b) interpreting those experiences so as to recognize what can be learned from them, to draw connections between their parts so as to transform them into useful cases, and to extract lessons that might be applied elsewhere; (c) anticipating their usefulness so as to be able to develop indexes for these cases that will allow their applicability to be recognized in the future; (d) experiencing failure of one's conceptions to work as expected, explaining those failures, and trying again (iteration); and (e) learning to use cases effectively to reason.

With respect to what the right kinds of experiences are, CBR suggests (a) that they be experiences that afford concrete, authentic, and timely feedback, so that learners have the opportunity to confront their conceptions and identify what they still need to learn; (b) that learners have the opportunity to move iteratively toward better and better development of the skills and concepts they are learning so as to experience them in a range of situations and under a variety of conditions; and (c) that they be experiences that allow cases to be compared and contrasted.

CBR's suggestions about promoting learning have informed three contributions to educational practice and the use of software tools for education.

- **Supports for reflection**: Prompts and other guidance for learners aimed at promoting productive reflection.
- **Case libraries as a resource**: Collections of cases and experiences that can act as external memory for a reasoner.
- **Engineering of the learning environment:** Effective sequencing of activities and facilitation of discussions so as to increase the frequency and impact of having the right kinds of experiences.

CBR's implications for supporting learning are in-line with those made by constructivist approaches to learning and the constructionist approach to education. All focus on promoting the kinds of thinking that will allow learners to construct productive mental models from concrete experiences. Constructionism goes on to say that experiences of actively constructing an artifact are particularly good for promoting such construction. Similarly, CBR begins by suggesting that we create environments that promote the kinds of hands-on experiences and active construction that will lead to good learning. But CBR goes farther. It provides a model of cognition (including processes and knowledge structures) that can be turned to for advice and predictions and that can be simulated on a computer as a test of ideas. This model, in turn, makes suggestions about how to orchestrate and facilitate students' experiences so that they can draw productive lessons from their experiences and makes suggestions about how to encourage transferable learning—so that lessons learned may be applied in new situations. CBR's cognitive model provides explanations of how learning happens and, from there, makes suggestions about how to ensure that active construction activities produce the results they afford.

32.2 CBR AS A MODEL OF COGNITION

CBR has been explored for many years in artificial intelligence as a way of creating more intelligent computer software. Several experimental case-based reasoners serve as the basis for CBR's cognitive model. The earliest case-based reasoner was CYRUS (Kolodner, 1983a, 1983b), a case library that knew about the life of statesman Cyrus Vance. When CYRUS was asked a question, it answered it by constructing a model of what the answer was likely to look like and then searching its memory for a matching case (a process of *reconstructing* the stories it held in its memory). Sometimes it did not find a case but, rather, answered questions by using this construction process to construct plausible stories. It was the first attempt to deal with retrieval and management of a case library. Early CBR systems, such as MEDIATOR (Kolodner & Simpson, 1989), CHEF (Hammond, 1989), and JULIA (Kolodner, 1993), showed us many of the processes involved in reasoning with cases. CHEF, which created recipes (plans for cooking), taught us much about the role of failure in learning and the role experience can play in helping us anticipate pitfalls as we are reasoning. A later system, called CELIA (Redmond, 1992), modeled the troubleshooting and learning of an apprentice mechanic. From CELIA we learned about the powerful role one's experiences can play before one has a full understanding of a domain and how important it is for a reasoner to have a variety of similar experiences so as to be able to extract the subtleties and nuances of the lessons it is learning and when each one applies. Still later reasoners, such as Creative-JULIA (Kolodner & Penberthy, 1990), IMPROVISOR (Kolodner & Wills, 1993), and ALEC (Simina & Kolodner, 1997; Simina, Kolodner, Ram, & Gorman, 1998) show us the role of CBR in creativity. The lesson from those models is that the quality of one's explorations before giving up on an idea, anticipation of the circumstances in which one might go back to it, immersion of oneself in an environment where one is likely to come upon such circumstances, and willingness to try, fail, and explain are all essential to reasoning that goes beyond the obvious.

CBR, as a cognitive model, values the concrete over the abstract (Kolodner, 1993). Whereas most traditional theories of cognition emphasize how general-purpose abstract operators are formed and applied, CBR makes concrete cases, representing experience, primary. CBR suggests that we think in terms of cases—interpretations of our experiences that we apply to new situations. To find the milk in a supermarket I've never been in, for example, I walk around the perimeter of the store until I reach the dairy section. Why? Because the dairy section of the supermarket I usually shop in is around its perimeter. When I throw a ball in the air, I expect it to come down because that's what I've always seen before. When I do strategic planning for my organization, I call on previous situations to suggest strategies and tactics and to warn of pitfalls. When I plan a dinner party, I consult menus I've served before as part of my planning; I may even serve the same meal I served another time if it worked well and different guests are invited this time.

Those schooled in traditional models of cognition will notice that CBR puts little explicit emphasis on abstract operators in the mind. There is no hierarchy of production rules, nor do we discuss networks of neuronlike components. Rather, we emphasize concrete experience in the form of stories that can be manipulated directly. CBR in many ways corresponds to our own introspection on how we think—in terms of stories and experiences. However, CBR does not exclude abstractions altogether. Rather, it places abstraction in roles that promote productive use of concrete experience: (a) for organizing similar cases in the case library so that one can choose one or a small number from the category from which to reason; (b) for creating indexing vocabulary; and (c) for managing partial matching—to allow the reasoner to recognize that two things that are similar but not identical are a close enough match. According to CBR's model, abstractions are extracted from concrete experience and formed as needed.

CBR explicitly integrates memory, learning, and reasoning. A reasoner, it says, is a being in the world that has goals. It seeks to navigate its world in such a way that its goals are successfully achieved. It has experiences, some of them successful and some not as successful, some pleasant and some not so pleasant, that allow it to learn about its environment and ways of using that environment to achieve its goals. As it has experiences, it seeks to learn the skills and concepts that will allow it to achieve its goals more productively in the future. It is engaged, therefore, in recording its experiences, interpreting its experiences to derive lessons useful to its future, anticipating when those lessons might be useful, and labeling its experiences appropriately so that it will be able to recognize the applicability of an experience in a later situation. A case-based reasoner is also engaged in noticing the similarities and differences between similar situations and experiences so that it can draw conclusions about its world and notice the subtle differences that suggest when each of the lessons it has learned is most appropriately applicable. Essential to its learning is failure—it needs to attempt to apply what it thinks is applicable and fail at that in order to know to focus its attentions on subtleties of which it had not previously been aware.

CBR suggests three components of cognition that we need to focus on: cases, case indexes, and the case processor.

Cases: Cases are interpretations of experiences. Cases have several subcomponents, just as stories do: their setting, the actors and their goals, a sequence of events, results, and explanations linking results to goals, and the means of achieving them. The better the interpretations of each of these pieces, and the better the explanations linking them to each other, the more useful a case will be when it is remembered later. For example, if we know that a plan carried out in a case failed, we can wonder whether it might fail again in a new similar situation, but we cannot make predictions. If, on the other hand, we know what caused the failure, we can check to see if the conditions that led to failure are present in the new situation. If they are, we can predict failure; if not, we might reuse the old plan.

The explanations that tie pieces of a case together allow us to derive lessons that can be learned from the case—its *lessons learned*. For example, if I unknowingly served fish to vegetarians, and they didn't eat, I might explain the failure as being due to my not having inquired about whether any of my guests were vegetarians or had special eating requirements. The lesson learned is that I should make those inquiries whenever I invite guests for dinner. On recall of a case, the lessons one has derived from it are available for application to the new situation, as are the explanations from which those lessons were derived. Lessons in a case can identify why things went wrong and why things worked and can help learners make predictions about the results of an experience given certain criteria and constraints. For maximum usefulness, cases should be interpreted with the goal of deriving lessons learned.

Cases can reside in one's memory, and the set of cases in one's memory is referred to as one's *case library* or *library of cases*. Cases in one's case library may be derived from one's own experiences or from the experiences of others. For example, one might read about someone else's experience and remember its lessons to apply in the future. In general, one's own cases will be more embellished, but the cases of others play a very important role in learning and reasoning, filling in where one's own experience is deficient.

Case indexes: A library is as good as the indexes and indexing scheme available for locating something on its shelves. So too with one's case library. We can find the right cases in our memories if we "indexed" them well when we entered them into the library and if the indexing scheme is defined well enough that we can recreate an index for an appropriate case when we are trying to locate something in memory. If reasoners cannot recognize a past experience as being applicable in a new situation, they will have no case to apply.

A good indexing scheme for case-based reasoners allows them to see a past situation as being relevant to the one now facing them. Thus, a case's *indexes* should allow us to find it at times when it might be productive to apply it. Good indexes are critical for *transfer*, the ability to apply knowledge or skills derived in one kind of situation in a situation that might be quite different.

The best indexing results from anticipating the circumstances when a lesson learned from a case might be useful and marking the case so that it will be recalled in such circumstances. For example, if I index the case where vegetarians didn't eat the fish I served under "serving fish as the main course at a dinner party," I will be reminded of that case each time I plan to serve fish at a dinner party. Remembering the case would remind me to apply the lesson it teaches: Ask guests if they have any special eating requirements. Or I might index the case more specifically under "having a dinner party," allowing me to be reminded that I ought to ask guests for their eating requirements even before I begin planning dinner.

It is important to keep in mind, though, that it is almost always impossible to identify every lesson an experience might teach and every situation in which it might be applicable. It is common to have an experience that one does not completely understand or appreciate until much later—sometimes because one is lacking the knowledge necessary to interpret it, sometimes because one is lacking the experience to know whether a result is positive or negative, sometimes for other reasons. We may recognize that our understanding is incomplete at the time of an experience, or we may come to realize that our understanding was incomplete only when attempting to use the case later and finding that its application led to poor results. Either way, indexing will be incomplete.

But incomplete indexing does not have to mean that cases are inaccessible IF the reasoner engages in *situation assessment* at the time that he, she, or it is trying to address a new situation. Situation assessment is a process of analyzing a new situation so as to understand it better. One attempts to infer unknown details of a new situation or to look at the situation from several perspectives. This interpretation process allows the reasoner to construct a better description of the new situation than he or she has available. Though the description is hypothetical, it plays a critical role in reasoning: The hypothetical interpretation of the new situation serves as an index that allows old cases to be recalled. One way to look at situation assessment is as a process of imagining, "If I'd encountered a situation like this in the past, what would it have looked like, and how would it have been described?"

Nor does a poor index at the time one encounters or experiences a situation mean that the situation can never be described well as a case or indexed well. Situation assessment allows a reasoner to remember a case that was not well indexed. If, after a case is recalled and used, the reasoner is better able to interpret it, he, she, or it might extract new lessons from the case or identify something critical about it and reinterpret the case and update the indexes associated with it at that time.

The case processor: A reasoner's case processor has a variety of responsibilities. This component needs to carry out the processing that results in understanding and indexing one's experiences, finding appropriate cases in memory, applying them in a new situation, and learning:

- interpreting a new situation in such a way that relevant cases can be located in the case library;
- deciding which of the old cases that is remembered is most applicable;
- applying the lessons learned from an old case to the new situation, for example, decomposing and recomposing pieces of old cases to create a new solution, adapting an old solution to fit a new situation, or choosing a strategy for moving forward;
- noticing results and explaining the reasons why some scheme did or did not work;
- structuring an experience as a case and choosing ways of indexing it; and
- when necessary, reinterpreting and reindexing an old case in light of new findings (e.g., derived by applying its lessons learned and finding that they did not work as expected).

Each of these components is important to productive use of cases for reasoning and learning. Together, they promote learning from cases, productive use of cases, reflection upon experiences so that they are indexed with future use in mind, and application of a lesson learned in one situation in another where it applies. One can find more detail about CBR and early case-based reasoners in Kolodner (1993), more detail about CBR as a cognitive model in Kolodner (1993, Chap. 4, 1997), and more detail about CBR's implications for learning and education in Kolodner (1997), Kolodner, Crismond, Gray, Holbrook, and Puntambekar (1998), Kolodner et al. (2003, in press), and Schank (1999).

32.3 IMPLICATIONS FOR EDUCATION

We can derive a variety of specific suggestions about promoting effective learning from the discussions of case libraries and CBR's cognitive model.

- CBR's focus on the role of failure in promoting learning suggests the importance of acquiring feedback on decisions made, in order to be able to identify holes in one's knowledge and to generate goals for additional learning. CBR's approach emphasizes the need for students actually to carry out and test their ideas, not just think about them.
- CBR's focus on explanation suggests that the learners should be pushed both to predict and to explain and that they should be helped to do both successfully. One cannot recognize a need to explain without first seeing a difference between what was expected and what happened. Thus prediction is important so that students can recognize holes in what they know.
- CBR's focus on indexing as the key to reuse of what is learned from experience suggests that, in addition to having experiences, students should reflect on and assess those experiences to extract both what might be learned from them and the circumstances in which those lessons might be appropriately applied, in order to index their experiences well for reuse.
- CBR's focus on iterative refinement suggests that learners should have the opportunity to try out their ideas in a variety of situations and to cycle through application of what they are learning, interpretation of feedback, and explanation and revision of conceptions several times—that we should not expect one application to promote accurate learning.
- CBR's focus on the role previous experience plays in reasoning suggests that learners should be encouraged to reuse their own previous experiences as they solve "school" problems. It also suggests that they might be helped along to solve more complex problems than they could by themselves by having access to the cases (experiences of others).

These suggestions have informed the creation of two approaches to sequencing activities for learning—Goal-based scenarios (Schank, Fano, Bell, & Jona, 1994) and Learning by Design (Kolodner et al., 1998, 2003, in press). They also suggest two roles for computers:

- Software might support student reflection, especially that involved in explaining their experiences, interpreting them to make them accessible and easily applicable, and anticipating the applicability of lessons that can be learned from them.

- Case libraries might serve as a resource to provide suggestions to learners as they are engaging in problem solving, explanation, or other reasoning.

That is, software can help students process their experiences to make them into cases that can be stored in their memories and later accessed and applied, and software can supply students with cases as resources that they can use to reason. A range of case-based learning aids has been designed with each of these functionalities in mind. Table 32.1 characterizes each of the case-based learning aids that will be described later in the chapter by the responsibilities that they take on. These case-based learning aids support the student as a case processor in taking on these responsibilities with the hope that, as the student interacts with the learning aid more and more, he or she will be able to begin taking on these responsibilities without the support of the case-based learning aid. Based on the needs of a particular learning environment, appropriate tools can be chosen or created that fulfill the needs of a learner.

We provide introductions to each of these kinds of case-based learning aids in this section, along with short examples, and in later sections, we provide detail on the two approaches to sequencing activities that CBR has informed and the design and use of the case-based learning aids introduced here.

32.3.1 CBR-Informed Supports for Reflection

It has been over 10 years since Alan Collins and John Seeley Brown (1988) first suggested that the computer could be used to support reflection. In that first conceptualization, the emphasis was on skills and process learning. Collins and Brown talked about capturing an expert's process, then allowing the student to compare his or her process to that of the expert. The computer's role was to record the expert's reasoning, making it available whenever it could be useful and to whoever needed it. In this way, the computer was supporting a kind of reflection that was difficult to do without a computer.

More recent supports for reflection have emphasized the use of design journals as a way of getting students to reflect on their plans and past experiences. In Idit Harel's (1991) Instructional Software Design Project, the only daily requirement for students was that they had to write down what they had done each day and what they planned to do the next. The hope was that they would articulate how they did things and what they were learning.

Collins and Brown's work has also been used as the basis for supporting reflection during reasoning or during project activity. KIE (Bell, Davis, & Linn, 1995) prompts students to think about evidence and its uses as they are creating a scientific argument. Reciprocal teaching (Palincsar & Brown, 1984) helps students to recognize the questions they need to ask themselves as they are trying to understand something they are reading. CSILE (Scardamalia, Bereiter, & Lamon, 1994) prompts students to think about their actions and their discussion as they are having knowledge-building conversations.

We know that reflection is an important component of learning, and each of these approaches helps students reflect in a way that that will help them learn a difficult-to-learn skill by suggesting important times for reflection and/or providing helpful prompts for reflection.

TABLE 32.1. Case-Based Learning Aids and the Responsibilities They Support

	Interpreting a New Situation	Deciding Which Old Case Is Most Applicable	Applying Lessons Learned from an Old Case to a New Situation	Noticing Results & Explaining Reasons Why Some Scheme Did or Did Not Work	Structuring an Experience as a Case & Choosing Ways of Indexing It	Reinterpreting & Reindexing an Old Case in Light of New Findings
Reflective Learner	X			X	X	
Archie-2 (as a resource)		X		X	X	
Archie-2 (as an authoring tool)	X		X	X	X	X
STABLE (as a resource)	X	X		X		X
Design Discussion Area	X			X	X	
Case Authoring Tool		X	X	X		
Case Application Suite	X	X	X	X		
JavaCAP/Storyboard Author	X			X	X	
Smile	X	X	X	X	X	

CBR allows us to go the next steps. Because it makes explicit the role of reflection in learning, it allows us to understand the kinds of reflection that are productive at different times and to understand what the results of those reflections ought to be. In particular, CBR tells us that reflection is critical for (a) interpreting an experience to connect its pieces together and extract what might be learned from it, (b) creating indexes, and (c) creating and evaluating solutions. In other words, CBR tells us that we should help learners understand their experiences in ways that will help them describe and index them well so as to be able to use them well later (Kolodner, Hmelo, & Narayanan, 1996) and that we should help them reuse their experiences productively and in ways that help them gain better understanding of the experiences they are using.

CBR-inspired support for reflection encourages students to think about (a) the kinds of problems they have faced in solving a problem or developing a skill or achieving a design challenge, (b) the kinds of solutions they constructed, and (c) the future situations in which the solutions might be used again, focusing particularly on how the lessons learned from an experience might be utilized in new ways. For example, Turns' Reflective Learner (Turns, Newstetter, Allen, & Mistree, 1997) helps students write "learning essays" about their design experiences. Puntambekar has described good results with paper-based, CBR-informed *design diaries* (Puntambekar, Nagel, Hubscher, Guzdial, & Kolodner, 1997; Puntambekar & Kolodner, 1998, submitted) in which students keep records of their project experiences.

Motivating students to reflect is a critical issue in learning, and the computer provides a motivation that children find compelling. For example, Shabo's JavaCAP (Shabo, Nagel, Guzdial, & Kolodner, 1997) and its successors, Kolodner and Nagel's (1999) Storyboard Author and Voida and Kolodner's (2002) Lessons Learned, help students summarize their project experiences and write them up as stories for publication in a permanently accessible case library for use by other students. The networked computer creates motivation for the students' reflection: Students enhance their own learning as they are trying to write summaries that can act as guides and supports to future students.

Kolodner and Nagel's (1999) Design Discussion Area uses the computer similarly to encourage reflection during hands-on activities. It provides a forum for students to share their ideas with others, to get advice and criticism of their own ideas from others, and to provide advice and criticism to others. Students write up the results of experiments they have done, ideas about achieving design challenges or solving problems they are working on, or what happened when they constructed and tested a design idea. They publish it for others to see. The computer prompts students to include relevant information in their write-ups. Publishing their materials makes the materials available to others to incorporate into their solutions. Reading the ideas of others gives them ideas. Commenting on others' ideas requires consideration of how the ideas of others work. Comments from others encourage deeper thought about the implications of their own ideas.

Owensby and Kolodner's (2002) Case Application Suite uses the computer to encourage interpretation, application, and assessment of old experiences and expert cases. Recognizing that an old experience or an expert case may be applicable, identifying which case to apply, and applying that case are skills that must be developed. The Case Application Suite scaffolds the examination and application of expert cases to the challenge the learners are trying to solve through the use of prompting, hints, examples, and chunking. Students can glean lessons learned (rules of thumb) from the experiences of the experts, and once their attempt at applying the lessons learned has been published, those experiences can serve as cases to be used by their peers.

There are several challenges to creating good CBR-informed supports for reflection.

- ***Motivating reflection:*** Reflection is hard to do and offers few extrinsic rewards. Motivating good reflection is a real challenge.
- ***Generating feedback:*** Computer-based supports for reflection can rarely respond intelligently about a students' reflection. In several of the tools listed, collaborative discussion areas are used to generate feedback on the students' reflections, but this kind of feedback will necessarily occur after the reflection is complete and is dependent on the quality of the discussants.
- ***Encouraging quality reflection:*** Reflection is hard to do but easy to "fake," that is, generating text that sounds reflective but really is not (Ng & Bereiter, 1995). Encouraging students to reflect about things that can lead to better learning is hard to prompt and structure.
- ***Not overdoing it:*** Periodic reflection while attempting to solve a problem or understand a situation is productive, as is summative reflection when one is finished. It is easy to identify times when reflection would be productive, but it is also easy to overdo it—to try to force reflection at times when it interferes with other reasoning or so often that it becomes a hated activity. We need to find that happy medium—a way of promoting reflection at productive times and without damaging a train of thought.

Computer tools can aid reflection, but the wanting to reflect, helping learners reflect better, and managing when to reflect have to be handled from elsewhere. Both sequencing approaches suggested by CBR (goal-based scenarios and Learning by Design) suggest pragmatic approaches to these issues. Other approaches (e.g., problem-based learning [Barrows, 1986], Project-Based Inquiry [Blumenfeld et. al, 1991]) also provide suggestions about managing these hard problems, and the kinds of reflective tools CBR suggests could, in principle, be easily inserted into any of those frameworks.

32.3.2 Supporting Learning with Case Libraries

The most common place where CBR has influenced the design of software tools to support learning is in the creation of case libraries. A case library offers two opportunities: the opportunity to learn from *others'* experiences and the opportunity to learn by sharing one's own experiences with others. Case libraries can offer a variety of kinds of information of value to learners.

- *Advice in the form of stories*: When we first think about case libraries, we normally think of stories—from experts, from peers, from people in unusual situations. Stories about success are valuable for the advice they give about how to proceed or what strategies to use. Stories about failure provide advice about what to avoid or issues on which to focus. Stories can also provide the basis for predicting what might happen if one tries out one's solution. Valuable stories are those that help a student understand a situation, the solution that was derived and why it was derived that way, and what happened as a result, as well as the explanations that tie those pieces together. Stories may be presented in a variety of media; the important thing is to present them in ways that make their points, or lessons that can be learned from them, most clear. Also important is that stories be indexed in ways that anticipate their use. That is, the indexer needs to think about the ways the case library will be used and the questions with which a user might come to the case library. He or she indexes stories so that it will be easy to find stories that address those questions (Kolodner, 1993).
- *Vicarious experience using a concept or skill:* We know that it takes several encounters with a concept or skill to learn it well (Redmond, 1992)—encounters that cover the range of applicability of the concept or skill allow the learner to see its varied uses and the other concepts or skills to which it is related, and to debug its applicability and refine its definition. But there usually is not time in school for students to experience actively the full range of applicability of a concept. Sharing experiences with other students or looking at the ways experts have applied concepts and skills can fill those gaps. In Learning by Design (Kolodner et al., 1998), such sharing is built into the system of activities students do in class in three ways—students engage in "gallery walks," sharing their design experiences with each other several times in the course of every design challenge in which they engage; students use DDA (Kolodner & Nagel, 1999) to write up their design experiences after in-class gallery walks to share across classes; and students write up what they have learned at the end of a unit (using StoryBoard Author), and the best are put it in an archive (Peer Publications) for students in following years. In all of these instances, students have the opportunity both to present their work and to engage in discussion with other students about it—they clarify for others, answer questions about why they did things a certain way, and then entertain suggestions about how to improve their designs.
- *The lay of the domain and guidance on focus*: An on-line case library's indexing system, if it is available for examination, can serve as an advanced organizer for students or even scaffolding for how students might think about their own cases (Spiro, Feltovich, Jacobson, & Coulson, 1991). For example, the system of indexes in Archie-2, which helped architectural students design public libraries, helped students develop an understanding of the issues that need to be addressed in designing libraries, the kinds of spaces libraries have, and the perspectives different kinds of library users might take on how well it functions. In this role, the case library's indexing system provides a view of the domain's major concepts and their relationships and guidance on what to focus on when designing or solving problems.
- *Strategies and procedures*: Sometimes what is most valuable about a story is not the solution itself, but the strategies employed or even just the starting point. For novices in a domain, the biggest problem is sometimes how to start (Guzdial, 1991)—What is the first thing to do or to try or to explore? In many models of design, simply defining the problem is the most challenging aspect (Schon, 1982). Cases that describe somebody's problem-solving or design process can show how others have defined problems and proceeded through to a solution.
- *How to use cases*: Learning about others' experiences in such a way that learners can reuse the lessons learned in novel situations is a complex metacognitive activity (Silver, Branca, & Adams, 1980). Cases that are about applying someone else's case can help students understand how experts reuse cases. Case libraries that prompt for the kind of analysis that is necessary in deciding whether a case is relevant and how to adapt it for reuse can help learners develop CBR skills.

The context in which case libraries are used is critical to their effectiveness. Case libraries have proven most useful as a resource that provides information as needed as students are engaged in constructive learning activities. In a project-based learning situation (Blumenfeld et al., 1991) a case library may provide guidance for getting started, for moving forward, and so on—if its cases answer the project-related issues that arise as students are working on a project. In a problem-based learning (Barrows, 1986) or learning-from-doing (Schank & Cleary, 1994) situation or in a learning-from-design situation (Kolodner, 1997), cases can provide those same benefits. But in a more traditional, lecture-based or fact-based classroom, cases may not be useful or may even be ignored by the students.

Common sense suggests that for cases to be a useful resource to students, the students must be engaged in an activity in which their impasses might be answered by cases in the case library. If the students are simply memorizing facts, then the challenges that the students will face (e.g., learning to memorize a particularly complicated fact) will not lead them to need or want to use a case library. However, if students are facing challenges that arise naturally in problem solving (e.g., "How do I model a situation like this?" or "What is a good starting point for this kind of problem?"), then a case library of relevant situations and problems can help them address those impasses.

Building case libraries can be as valuable educationally as using case libraries, as suggested above, sometimes even more valuable than simple use. Students building a case library explicitly have to deal with issues of identifying appropriate indexes, identifying strategies and process elements, and decomposing the case for others to use. By making these activities explicit, the intention is to induce learning goals in the student that are appropriate to generating transferable knowledge (Ram & Leake, 1995). The activity of building a case library is frequently motivating for students, as it is creating a public artifact whose purpose is to help future students. This is the same kind of motivating activity on which Harel and other constructionists have been building (Harel & Papert, 1990; Papert, 1991). Cognitively, the need to explain to others in a way that will allow them to understand requires reflecting on a situation, sorting out its complexities, making connections between its parts, and organizing what one has to say into coherent and memorable chunks. Storytelling can aid making sense and remembering (Schank, 1982, 1999).

Case libraries can be a particularly rich source for educational content and process. As a content resource, case libraries offer resources for students to study and to use in actual problem-solving activity. As a process resource, case libraries offer opportunities for students to articulate knowledge and reflect on their experiences in a way that other hands-on activities do not usually provide.

32.4 CBR'S SUGGESTIONS ABOUT ENGINEERING THE LEARNING ENVIRONMENT

CBR's suggestions have also been used to inform the design of learning environments that employ cases as a way to help students learn. Following is a discussion of two such learning environments—goal-based scenarios and Learning by Design.

32.4.1 Goal-Based Scenarios

One of the originators of CBR is Roger Schank. In his work on learning supports, he has been applying the lessons of CBR to creating a new kind of learning environment called a *goal-based scenario* (Schank et al., 1994).

Key to Schank's vision of learning is that motivation is a critical aspect of learning. Basing his claims on the cognitive model implied by CBR, he claims that unless students have a *reason* for wanting to learn or do something, nothing that anybody wants them to learn will make sense to them. Further, until a student fails (reaches an impasse) at something, Schank (1982) believes that they have no reason to question what they are doing and therefore no reason to want to learn anything new. For example, case libraries play a significant role in a goal-based scenerio, but setting up their *context* of use so that students will have a reason to want to use the case library and a context for understanding what it is offering is as important as creating the content of the case library itself.

A goal-based scenario is a learning environment that places students in a situation where they have to achieve some interesting goal that requires them to learn whatever is in the curriculum goals. In one goal-based scenario, for example, students play the role of advisors to the President in dealing with a hostage situation in a foreign land (Bareiss & Beckwith, 1993), in the process learning about several hostage-taking events that have happened in history and also learning some foreign policy. In another, students advise couples about their risk of having children with sickle-cell anemia (Schank et al., 1994), in the process learning about genetics in the context of sickle-cell disease. Using Broadcast News, students put together a news story, in the process learning both history and writing skills. Students learn about history or genetics or writing because they need to learn those things to achieve successfully the challenge set for them. The trick, of course, is to design challenges that both engage the students and focus them on the content and skills we want them to be learning.

Students engaged in a goal-based scenario are provided with a case library of videos of experts telling their stories, strategies, and perspectives that might help them with their task. When they reach an impasse in achieving their goal, they ask a question of the case library, and an appropriate video is retrieved and shown. Sometimes a story will suggest a topic they should learn more about or a skill they need to learn; other times it will tell how that expert dealt with some difficult issue the student is addressing. Students are in a situation where the case library is relevant for their impasses. Students engaged in a well-designed goal-based scenario take on goals that lead them to want to know and apply the recorded experiences of others.

Based on suggestions made by the case library, students move forward with their task—choosing a policy to recommend to the President, choosing a blood test, making a recommendation to a couple about whether or not they should have children, or deciding how to refer to a leader. In all goal-based scenarios that have been implemented on the computer, there are clear *right* answers to each small task they are working on, and the software can detect when the students have selected the *wrong* answer. The software informs students when they have failed at their task, and through use of cases in the case library, it helps them explain and recover from their failures and move forward successfully.

This second context for a case use—recognizing, explaining, and recovering from failure—suggests that case libraries used as part of a GBS need to index their cases in two ways—by content and also by their applicability helping a learner explain why his or her action failed and how to recover. A story told to the student after a failure can successfully lead to learning when the student is in a context where he or she needs that particular story to move forward.

Case libraries used in a goal-based scenario focus their indexing very tightly on the context in which a retrieved case will be used: On what task is the student working? What is his or her solution in progress? What difficulty is the student having? and On what poor answer has the student settled? When building a case library to be used as part of a goal-based scenario, case indexes are chosen by anticipating the situations in which a student will want to hear a story. By focusing indexing on the learner's goals, these case libraries can act as very powerful supports for learning.

Research papers by Schank and his students report more details of how the cases in a goal-based scenario should be organized and accessed (Bareiss & Osgood, 1993; Ferguson, Bareiss, Birnbaum, & Osgood, 1992; Schank, Berman, & Macpherson, 1999). Most critical to keep in mind is that the design of a goal-based scenario requires anticipating learner's goals when working on a challenge. This, in turn, requires anticipating the tasks students will carry out, the avenues of thought and strategies they will pursue, and the kinds of choices they will make. By using a students' tasks to promote goals students will pursue, the designer of a goal-based scenario can anticipate the kinds of impasses students will encounter and therefore the kinds of stories the case library needs to include and the ways those stories ought to be indexed for easy access.

32.4.2 Learning by Design

Like goal-based scenarios, Learning by Design (LBD) (Kolodner, 1997, Kolodner et al., 1998, 2003, in press) takes CBR's cognitive model seriously in the design of learning environments. But whereas the goal-based scenario approach focuses on designing computer programs that help a learner achieve an exciting challenge, LBD focuses on using CBR's model to suggest how to orchestrate a classroom environment. In addition to suggesting ways of integrating the computer into the classroom, LBD is explicit about teacher roles and about the sequencing of individual, small-group, and whole-class activities.

LBD curriculum units ask middle-school students (ages 12–14, grades 6–8) to achieve design challenges as compelling contexts for learning science concepts and skills. Design challenges provide opportunities for engaging in and learning complex cognitive, social, practical, and communication skills. For example, students design parachutes made from coffee filters to learn about air resistance and gravity and their relationship; miniature vehicles and their propulsion systems to learn about forces, motion, and Newton's laws; and ways of managing the erosion on barrier islands to learn about erosion, water currents, and the relationship between people and the environment. Construction and trial of real devices give students the motivation to want to learn, the opportunity to discover what they need to learn, the opportunity to experience uses of science, and the opportunity to test their conceptions and discover the bugs and holes in their knowledge. The teacher helps students reflect on their experiences in ways that help them extract and articulate and keep track of both the content and the skills they are learning.

CBR tells us that learning requires impasses and expectation failures—to show us what we do not know, to focus us on what we need to learn, and to motivate us to want to learn. This suggests an iterative approach to learning from experience—try to solve a problem or achieve a challenge, use the impasses and failures of expectation to show what needs to be learned, investigate in some way to learn more, and try again. But how can failures of expectation be engineered into students' activities? CBR suggests that the best learning experiences will be those that afford real feedback in a timely way. Designing, building, and testing working devices provide that kind of feedback. Based on these suggestions from CBR, LBD's curriculum units are centered on the design and construction of working devices or working models that illustrate physical phenomena or that measure phenomena (e.g., to get feedback about biological function).

CBR tells us that learning from experience requires reflecting on experiences in ways that will allow learners to derive well-articulated cases from their experiences and insert them well into their own memories. We also know that learning is most effective when learners have been able to identify what they need to learn—when they have had a chance to think about what they do know and how to apply that and then identified where the gaps are. LBD includes in its activities a system of classroom rituals that promotes such derivations. "Messing about" is guided play done in small groups that promotes making connections between a design challenge and what students already know. Playing with toy cars, for example, seeing which ones can go over hills and which ones cannot, gets students thinking about what it takes to get a vehicle over a hill and the different ways they have made things move. "Whiteboarding," borrowed from problem-based learning (Barrows, 1985), follows messing about and is a whole-class activity in which learners articulate together what they discovered during messing about and generate ideas about how to proceed and learning issues to pursue. "Poster sessions" are presentation venues where students present their investigation procedures and results to each other. "Pinup sessions," borrowed from the architecture design studio, give small groups the opportunity to share their plans with the whole class and hear other students' ideas. "Gallery walks," adapted from pinups, provide a venue for presenting one's designs in progress to the rest of the class. All three types of presentations require students to articulate what they are doing well enough for others to understand; they also provide students with ideas to build on in moving forward, a venue for getting feedback on their articulations (Are they communicating well?), for asking for advice and getting suggestions, and for vicarious experience applying the concepts and skills they are learning.

Using guidelines from case-based reasoning, LBD provides (a) libraries of cases for students to use as resources; (b) paper-and-pencil and software tools that allow students to keep track of their design experiences so that they can remember what they did and draw lessons from their experiences; (c) a system of classroom activities that help students make contact with their own previous experience and bring it to bear (messing about), help them anticipate what they need to learn more about (whiteboarding), and help them share their ideas with each other (poster sessions, gallery walks, and pinup sessions); (d) software tools that prompt students to explain their design decisions and design experiences to each other and get feedback from their peers; (e) software tools that prompt students to extract and articulate the content and skills they are learning from their experiences and write them up as stories to share with other students; (f) software tools that help students read the cases written by experts and extract from them the science and advice that can help them with their design challenge; and (g) teacher guidelines for facilitating reflective discussions and other activities in ways that help students to turn their experiences into cases—stored in their memories in ways that allow them to remember and apply them in later situations (e.g., helping them identify what they learned, how they learned it, under what conditions it might be applicable, and when such conditions might come up in the future). The tools LBD provides act as resources, help students create cases for others to use, help students keep track of what they have been doing; and help students reflect on their experiences and turn them into cases in their own memories. Each tool is used in the context of other classroom activities and discussions that support their use and is designed to enhance LBD activities. Some LBD teachers have integrated the use of software tools into their classrooms; some have not. More detail on LBD's software tools is provided in the next section.

32.5 EXAMPLES OF CBR-INFORMED LEARNING AIDS

CBR and case libraries have a rich research history, but educational applications of CBR are relatively new and still relatively few. We select a few projects and describe them below to provide concrete examples of how CBR can inform the creation of learning supports. They fit three categories: supports for reflection on and interpretation of one's own experience, support for use of case libraries, and hybrids that support both.

32.5.1 Supports for Reflection and Interpretation of One's Experiences

32.5.1.1 Reflective Learner. Students in undergraduate project-based design courses face a huge number of challenges as part of their learning. They have to do design at the same time that they are learning about design, using theory and engineering principles that they may have just learned a term before (Turns, Guzdial, Mistree, Allen, & Rosen, 1995a). Often, they are working in groups, so they have to deal with issues of collaborative work at the same time (Turns et al., 1995b).

What Turns discovered in her ethnographic studies of students in engineering design courses was that students often did not even know what they were supposed to be learning, why they were engaging in the activities they were being asked to engage in, and, worse yet, how to reflect on their activities in order to learn from them (Turns et al., 1997). She decided to build a support for learning that directly addressed the issue of reflection.

Her tool, Reflective Learner, supports students in producing "learning essays" about their experiences. The requirement for the students to write learning essays already existed in the engineering design class that she chose to study. However, the unsupported learning essays were not particularly satisfying to the teacher or students. Students still seemed confused about why they were doing what they were being asked to do.

Reflective Learner provides scaffolding in the form of prompts to help students write learning essays in a more effective manner. Its prompts are directly informed by CBR's suggestions about the reflection needed to be able to learn from and reuse one's experiences. It asks students

- to identify and describe a problem that they had encountered when undertaking the current phase of their design project,
- to describe their solution to the problem,
- to say what they had learned from the experience, and
- to anticipate the kinds of situations in which a similar solution might be useful.

Turns' interviews and discussions with students suggest that they found this activity useful and that it helped them to understand why they were doing what they were doing.

32.5.1.2 The Design Discussion Area (DDA) and Its Successor Tools in Smile. An important lesson learned from exploration of apprenticeship and case-based learning (Redmond, 1992) was that it takes several encounters with a concept or skill to learn it well. The first encounter allows the learner to build an impoverished picture of the concept or skill. Later encounters, in which that impoverished picture is applied and fails to work as expected, let learners know that their knowledge base is incomplete or incorrect, prompting engaged learners to want to revise their knowledge, cases, or indexing so that it works better. But school does not provide the time for students to have the full range of experiences that would allow them to build up a complete understanding. LBD's poster sessions, gallery walks, and pinup sessions, and their electronic extension, the DDA (Kolodner & Nagel, 1999), are designed to help students share their experiences with each other so that they can learn vicariously from each other's experiences.

For such learning to happen, students need to be able to present their design ideas coherently, and for students to learn science from their own experiences and those of others, they need to talk the talk of science as they are presenting their ideas and conversing with others. The DDA is designed with two learning goals in mind: (a) to help small groups of students present their design ideas and results to others coherently and using the right kinds of vocabulary and (b) to guide students in other work groups through conversations about those design ideas.

Figure 32.1 shows a design idea and short discussion about it along with the simple prompts the DDA provides to aid discussion. The DDA helps students articulate their design ideas by providing three kinds of scaffolding—a structuring of the writing area into well-organized chunks ("our solution idea," "functions it satisfies," and "how it will work" can be seen in Figure 32.1), hints for what belongs in each of those structured paragraphs, and examples to examine. The intention is that for each design idea or design experience they report on, students will report the design decisions they made, why they made those decisions, the evidence they used to come to that decision, and, if they have applied it, what happened, their explanation of why, and anything new they feel they need to learn. After small groups of students complete their reports and "publish" them in the case library, the DDA provides another set of prompts to help peers comment on published material. These prompts, shown on the right side of Fig. 32.2 similar to those in CSILE (Scardamalia & Bereiter, 1991), invite students to identify the kind of contribution they are making to a design discussion. We invited them to "praise," "wonder," and "suggest"; they can also make other kinds of comments if they specify the type.

When we used the DDA in the classroom of a very masterful teacher (Kolodner & Nagel, 1999), we found that when students used it before making presentations to the class, their presentations were of a higher caliber. But we also discovered that he was providing a great deal of scaffolding to students in addition to what we provided in the tool. In particular, for each of the kinds of presentations students made in the classroom (of experimental results, design ideas, and solutions in progress), the teacher was giving them different kinds of instructions about how to use

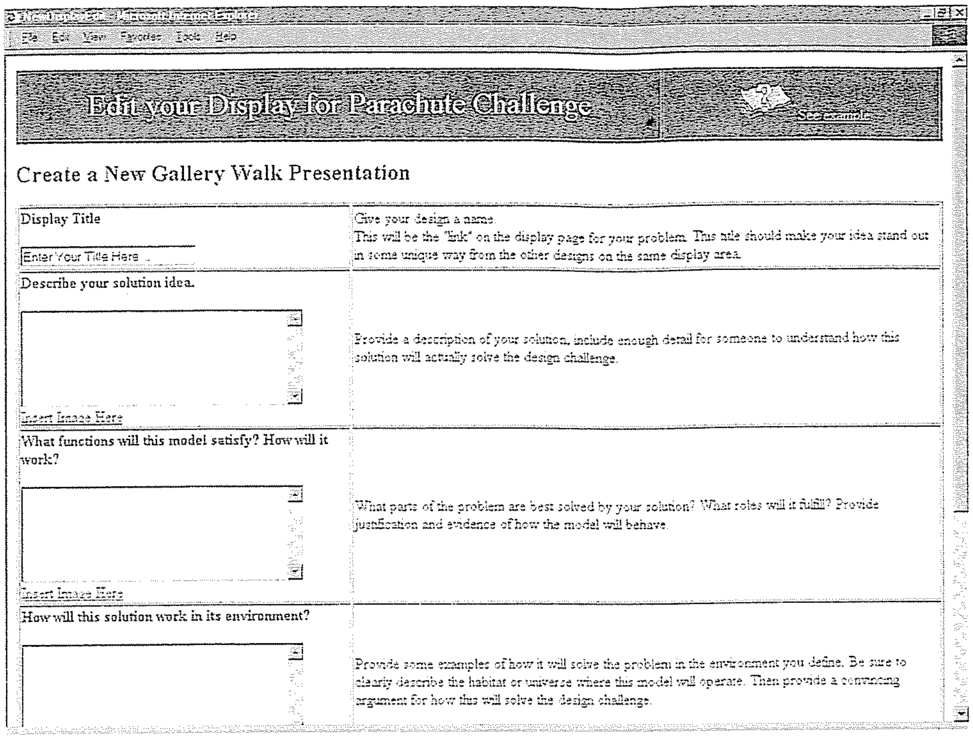

FIGURE 32.1. Design discussion area prompts.

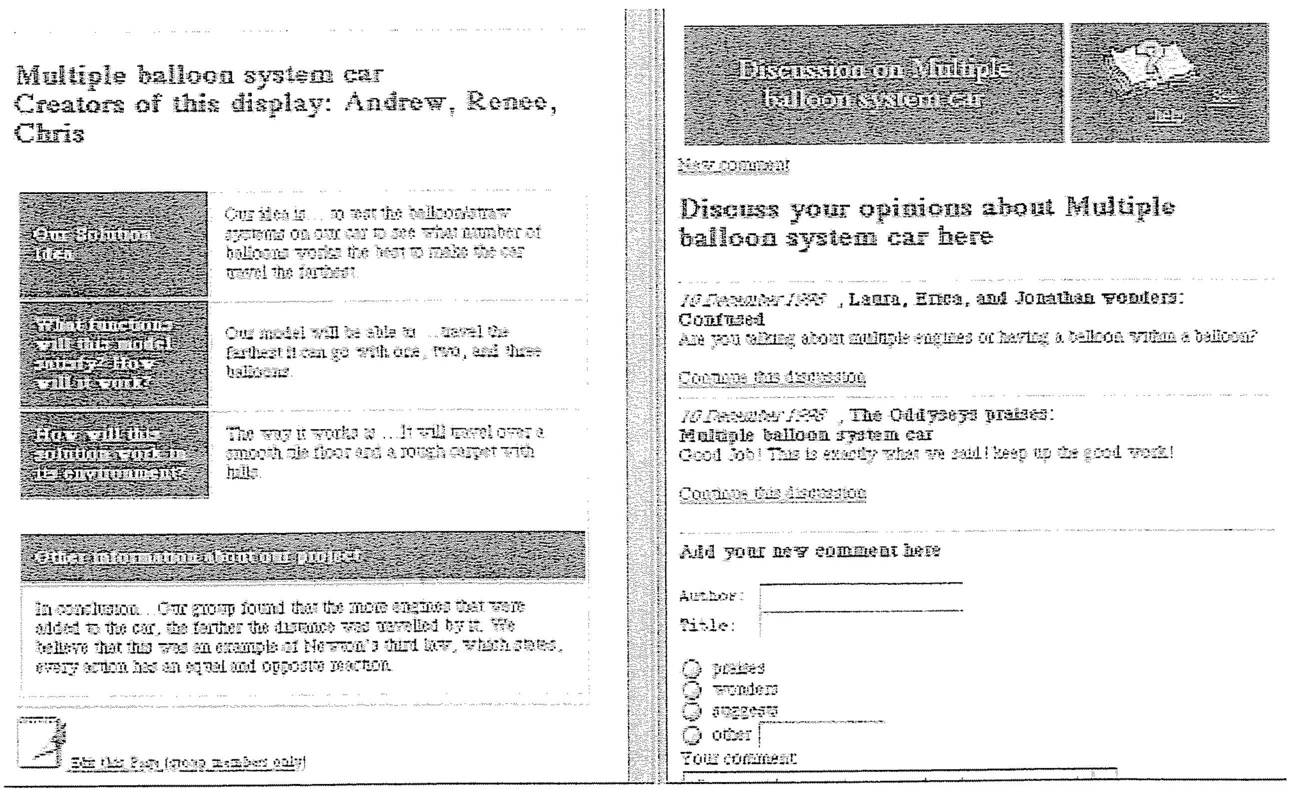

FIGURE 32.2. Design idea with discussion.

the tool. Our analysis of the situation led us to predict that if we rewrote the software, maintaining the same types of scaffolding but creating tools for each of the kinds of planning activities and presentations students engaged in, the software would be easier to use and provide better guidance. In response to these predictions, the DDA has grown to encompass several tools in Smile, each providing prompting specific to the kind of experience being reflected on and the kind of presentation that needs to be made. Figures 32.3–32.5 show a selection of those tools. In these screen shots, the left-hand side of the screen provides organizing structure to whatever task students are working on, and the right side holds hints, examples, and templates to help with completing the task.

An example of one of these tools is the Experiment Result Tool (Fig. 32.3). After students have conducted experiments and gathered data, this tool helps them analyze their results, with the aim of reflecting on their experimental methodology and understanding what lessons their results suggest. The Experiment Result Tool prompts them to do that—students record their data, compare their results to the predictions they made when planning their experiment, and create a rule of thumb based on their analysis. After using the tool to help them make sense of their results, they present their results to the class in a poster session. After discussion with their peers, they might edit their on-line write-up and publish it for others to see, comment on, and use.

The Pin-Up Tool helps students use the results of investigations to come up with their best solution to their project challenge. Students are asked to formulate design decisions and justify them with evidence—from experiments just performed, rules of thumb extracted, and science laws read about. We provide a template to help them line up their design decisions with their justifications. Students are asked to list their design decision, justify why they have chosen that

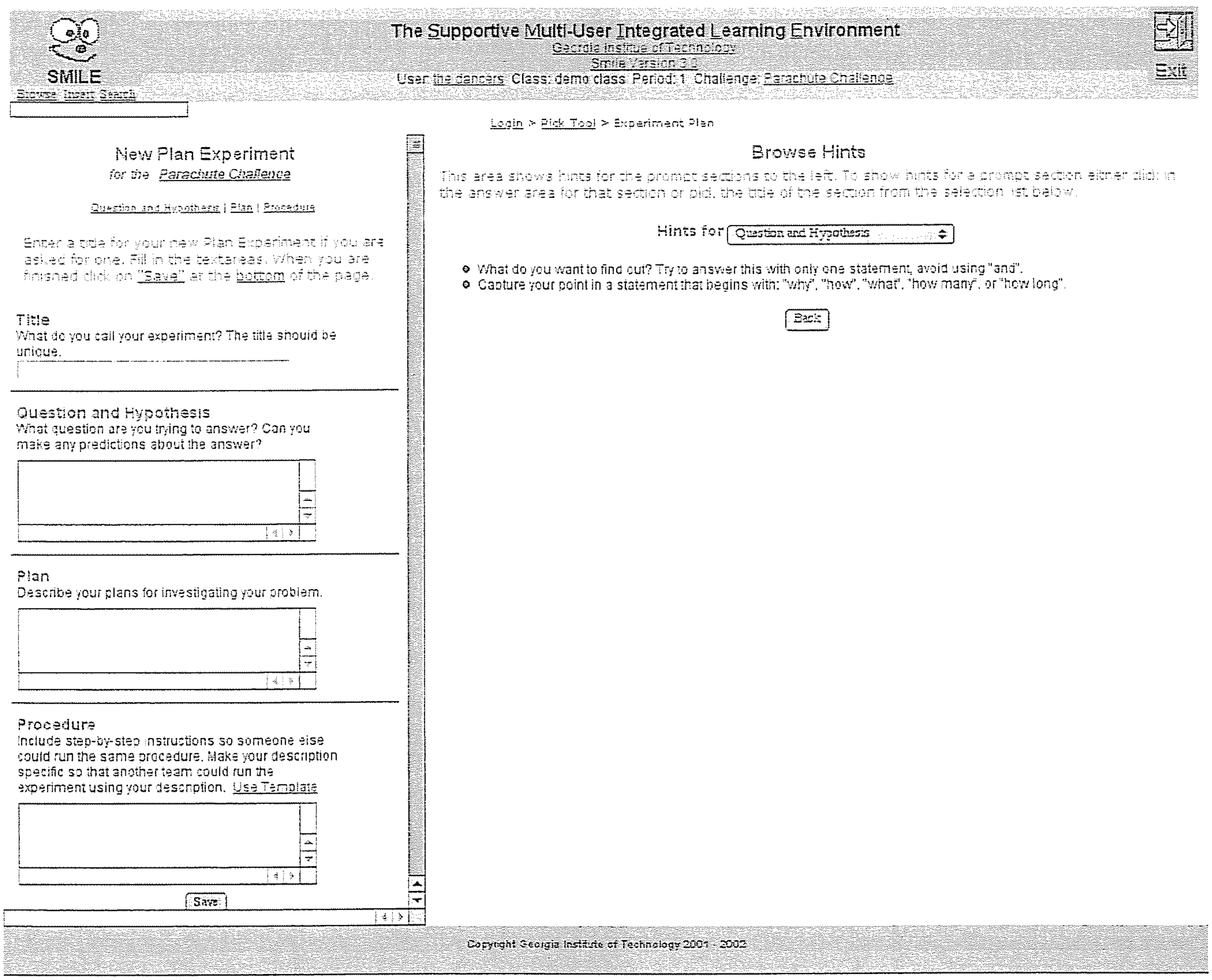

FIGURE 32.3. Reporting on an experimental result.

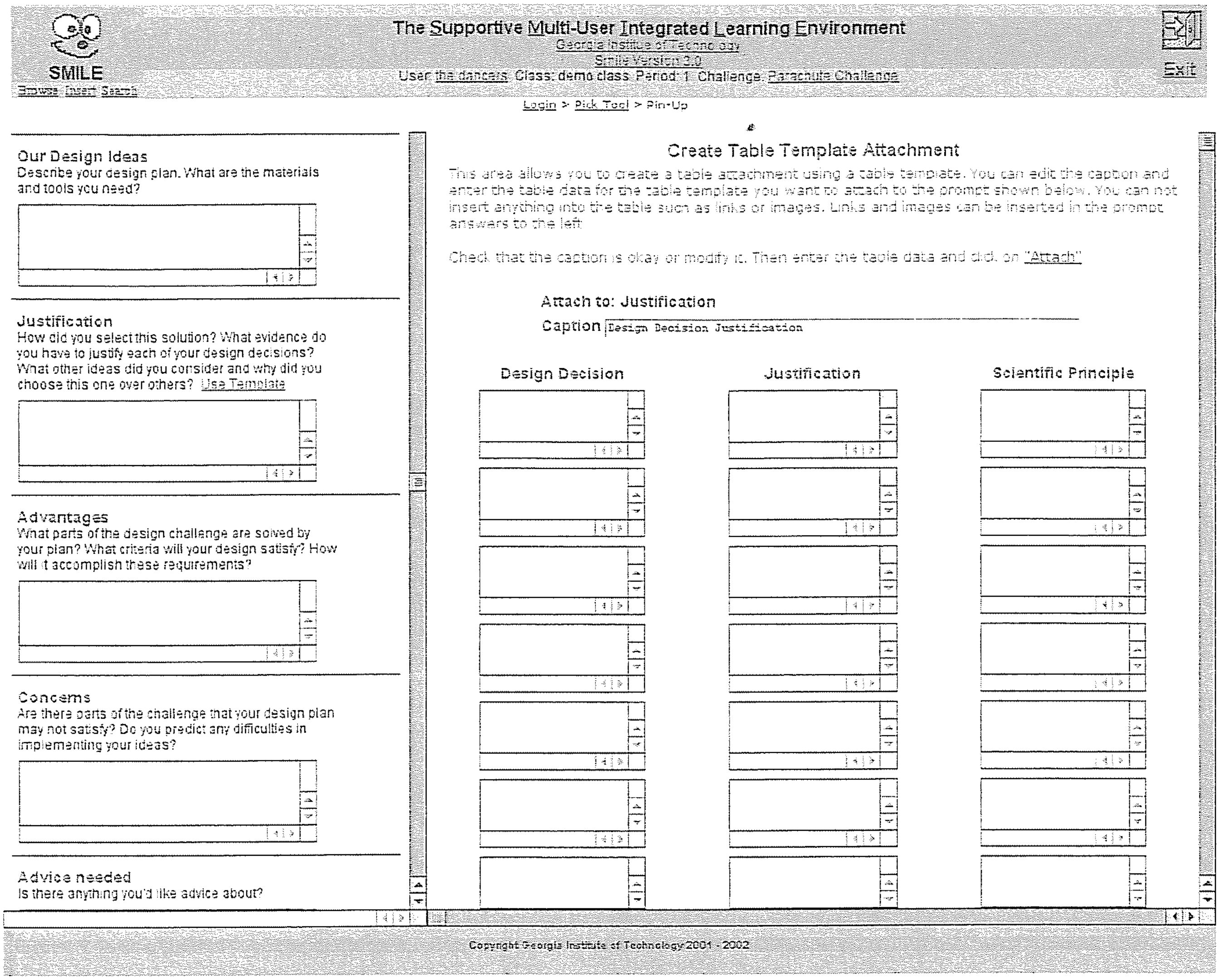

FIGURE 32.4. Coming up with design plans using the Pin-Up Tool.

decision, and provide a scientific principle that supports the decision.

The Gallery Walk Tool scaffolds students as they reflect on their design experiences and plan presentations of their solutions in progress for their peers. Their first time through, students have constructed a solution based on design decisions reported in their pinup presentation. But those solutions do not work exactly as they had thought. After trying out those ideas, this tool helps them look back on the decisions they made and articulate what happened differently than they had imagined. It then prompts them to explain, if they can, why their solution behaved differently than they had predicted. To facilitate this, the Gallery Walk Tool is linked to the Pin-Up Tool so that students can see their decisions and justifications as they are analyzing their results. Students can also edit their old decision and justification chart to show changes they will make in their next iteration. If students use the Gallery Walk Tool after each of their iterations, then at the end of their design challenge, they will have a full documentation that chronicles the decisions that were made at each iteration and why those decisions were made. This set can serve not only as a means of reflecting over the iterations of a design, but also as cases to be used by other students as they are engaging in the same challenge in the future.

After a team publishes its investigations, design ideas, and/or design experiences, their published artifact is available to other teams by clicking on its hyperlinked title in Smile's library. Looking at another team's idea will open two side-by-side windows: the presentation on the left and a comment window on the right. This anchored collaboration (Guzdial et al., 1997), similar to that in the original DDA, ties each student presentation to its own threaded discussion space. Other students may add a new comment or question for the team (a new thread) or insert a comment into an existing discussion. Scaffolding is quite

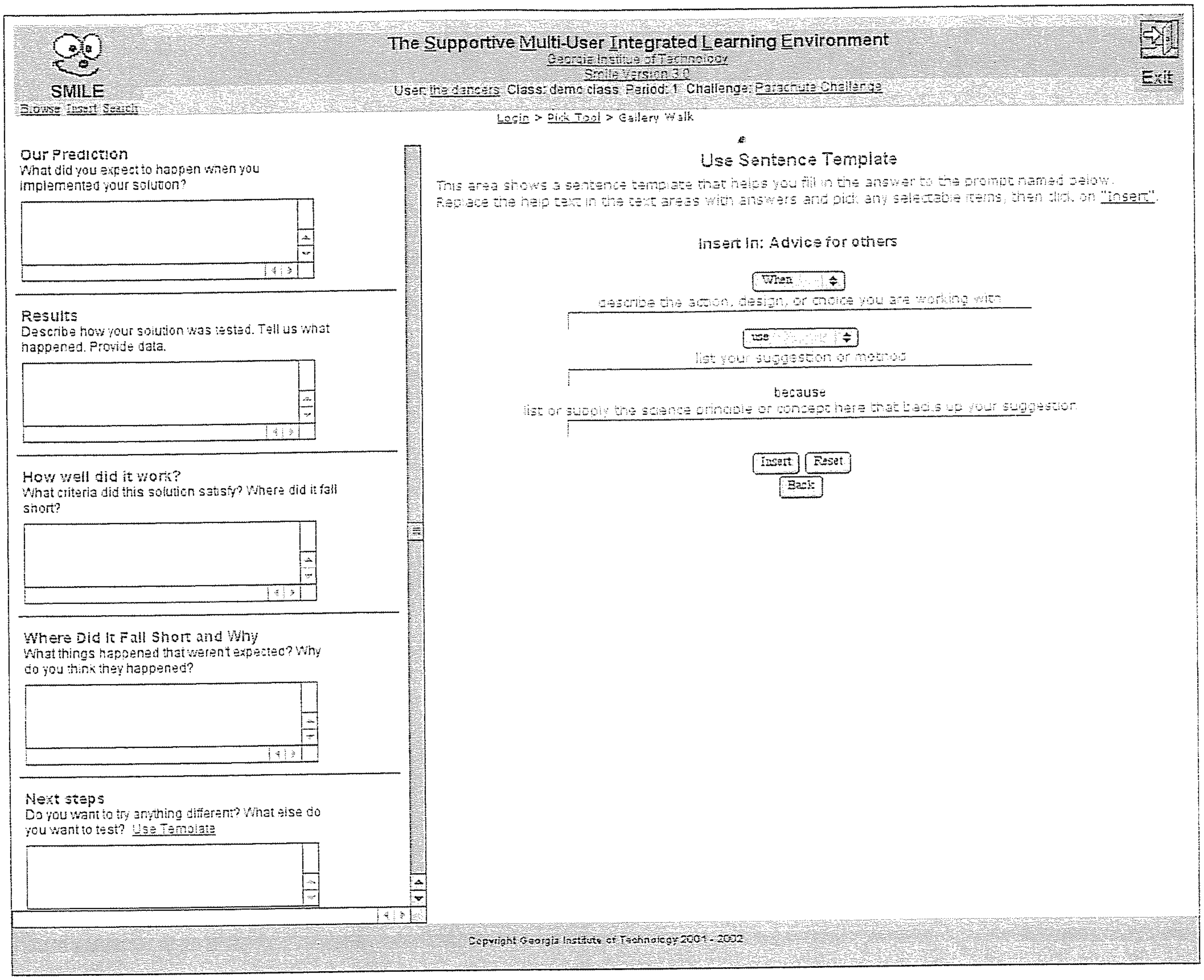

FIGURE 32.5. The Gallery Walk Tool.

minimal, so as not to get in the way, but serves two essential purposes: (a) It helps students differentiate between continuing an old discussion and beginning a new one; and (b) it makes suggestions to students about the kinds of comments they might want to make—"praises," "wonders," and "suggests" for new threads and "replies," "wonders," and "suggests" for continuing threads. As in the DDA, students can also add their own new types.

32.5.1.3 Case-Authoring Tool (CAT). Some design challenges do not lend themselves to exploration with real materials. It is hard, for example, to mess about with managing erosion in any way that gets across the complexities of managing erosion when winds and currents and tides are all interacting. For these kinds of situations, LBD has a different way for students to gain perspective on the challenge they are addressing—by looking at real-world cases that address those same sets of issues. For example, students working on the erosion problem read about the ravages of erosion on islands up and down the East Coast of American and around the world and the ways engineers have tried to control erosion and the problems that come with it. Those working on a tunneling problem read about cases where interesting tunnels have been built and what went into building them—e.g., the Chunnel, railroad tunnels through the Rockies, and the sewer system in New York. But reading expert cases is difficult, and knowing what might be learned from such a case can be difficult as well. CAT (Nagel & Kolodner, 1999) was designed to provide that guidance. It helps students divide their challenging task into manageable chunks and provides hints and examples for each. Figure 32.6. shows some of the help CAT gives students in articulating the solution the experts came up with. Three kinds of help are provided (as in the DDA): structuring of what they need to articulate into manageable chunks, hints for each of those chunks, and examples. CAT provides

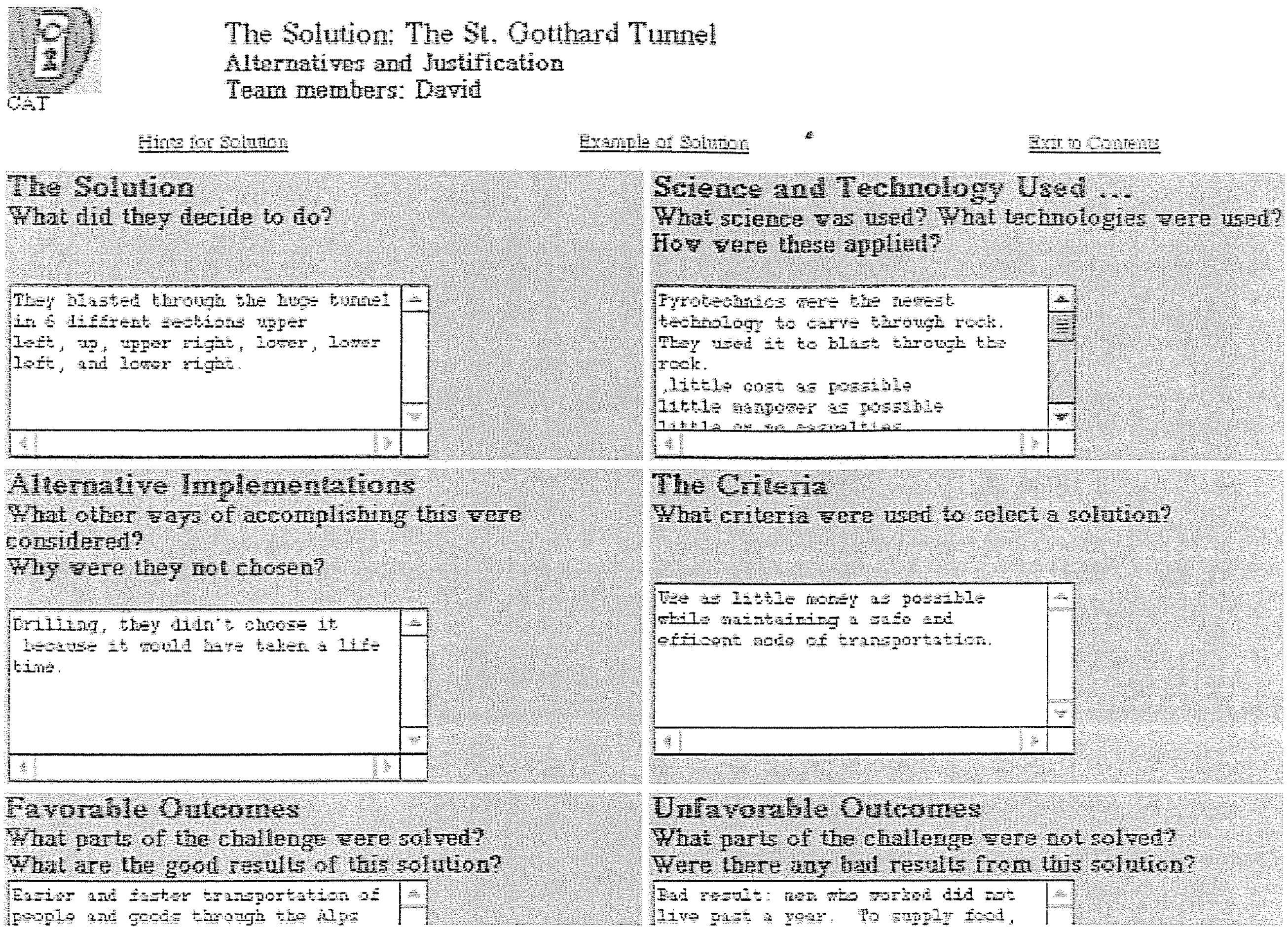

FIGURE 32.6. Case Authoring Tool's help with articulating expert solutions.

similar prompting to help students record the challenges the experts were up against and the issues they had to address and to record the results and how they affected the people and environment.

Our intention in designing CAT was that students would use it in small groups to read an article, extract what it says, and write that up for the rest of the class. We suggested that they first use CAT's prompts to skim the article they are reading and extract some of its important parts, then use the prompts to see where they should pay special attention in reading the article, and read those parts of it and write down what they have read. We suggested that they then do another iteration of rewriting their notes to compose a presentation of the case that others could use as a reference. As with the DDA, we wanted them to use CAT to help them read and interpret the expert case, present their case to the class, and then, after making small clarification changes in their on-line presentation, publish it as a resource. We did not, however, provide in this tool the kind of help students would need in applying a case to their new situation. We have designed CAT's successor, the Case Application Suite (Owensby & Kolodner, 2002), to provide both kinds of help and discuss it later as a case library tool.

32.5.1.4 JavaCAP and Its Descendants: StoryBoard Author and Lessons Learned. In LBD, students are asked to achieve a challenge, and along the way, they must investigate, analyze, and interpret component experiences. Although tools like the DDA and its successors help students interpret component experiences that are part of their full design project, they do not provide help with connecting those component experiences by pulling together the lessons learned from a full project experience. JavaCAP (Shabo et al., 1997), StoryBoard Author (Nagel & Kolodner, 1999), and Lessons Learned were designed to help students reflect on an entire project experience, summarize it and put it into perspective, extract from it what they have learned, and write that up in ways from which other students can learn. Our intention has been to provide guidelines in the software tool that will encourage students to look back over a long-term project experience to extract what they learned in productive ways.

JavaCAP (Fig. 32.7) was our first attempt at helping students reflect on their experiences and extract from them what they have learned. In JavaCAP, students began by describing their design problem and went on to describe the alternative solutions they came up with, why they chose the particular solution they

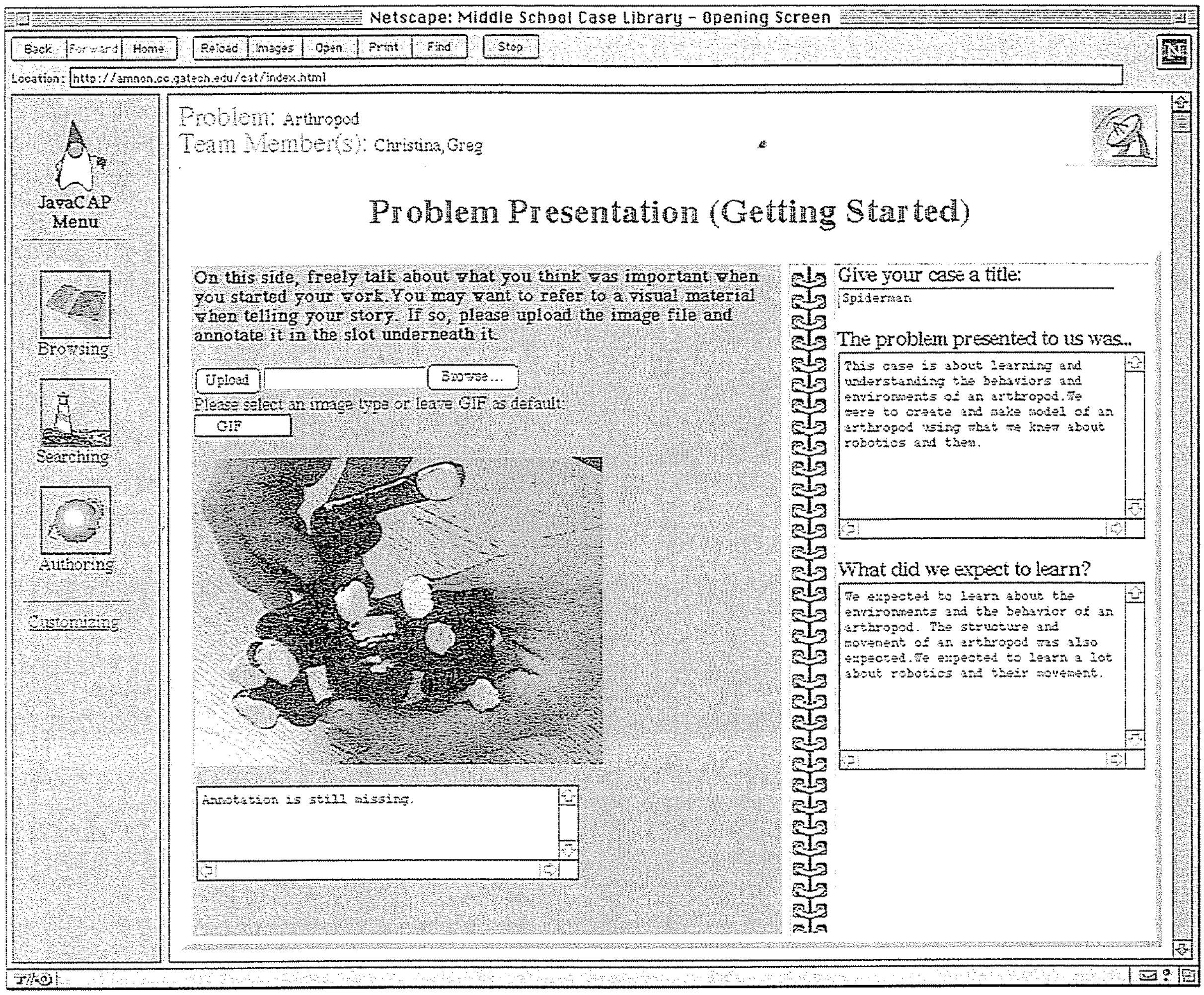

FIGURE 32.7. JavaCAP's scaffolding for presenting the problem.

did, and what they learned overall. Although students were able to articulate their design problem and describe their solutions, the scaffolding we provided for having them discuss what they had learned was quite limited. The open-ended prompting we provided led most students to write about the importance of research and the importance of the collaborative process, but our scaffolding was far too generic to support students in thinking deeply about the science they had used and its implications. As with the DDA, it has taken several iterations to get to scaffolding with the appropriate specificity.

StoryBoard Author (Fig. 32.8) was our next attempt to provide such scaffolding. More structured and specifically tailored prompting was added to help students put their experiences into perspective. Indeed, we took our cues from CAT, designed to help students read the cases of others, and in StoryBoard Author, we used the same kinds of prompts that we found they needed to understand the stories of others as guidance as they thought about and reported on their own experiences. Students were specifically prompted for reconstruction of their project experience in ways that would get at the details, and affect was used to help them remember the most interesting parts of their experience (e.g., What were you proud of?). In StoryBoard Author, students are asked to articulate the challenge they have been addressing, their solution to it and how they came to that solution, the science they applied in getting to the solution, and how well their solution works. Figure 32.8 shows prompts provided to students in StoryBoard Author to help them articulate a description of their project challenge.

To help students identify what they have learned, StoryBoard Author asks them to think back on the things that used to confuse them but do not anymore, the things that still confuse them, surprises they encountered, things that made them angry, and things that made them happy. It asks them to jot down short notes to themselves on the computer about these things, and it helps them sort each of those into one of three categories: science or technology concepts (e.g., gravity, inertia), science or technology skills (e.g., choosing variables, measuring), and project skills (e.g., collaboration, communication, planning). For each category, StoryBoard Author provides prompts and examples to help them tell the story of what they learned and how

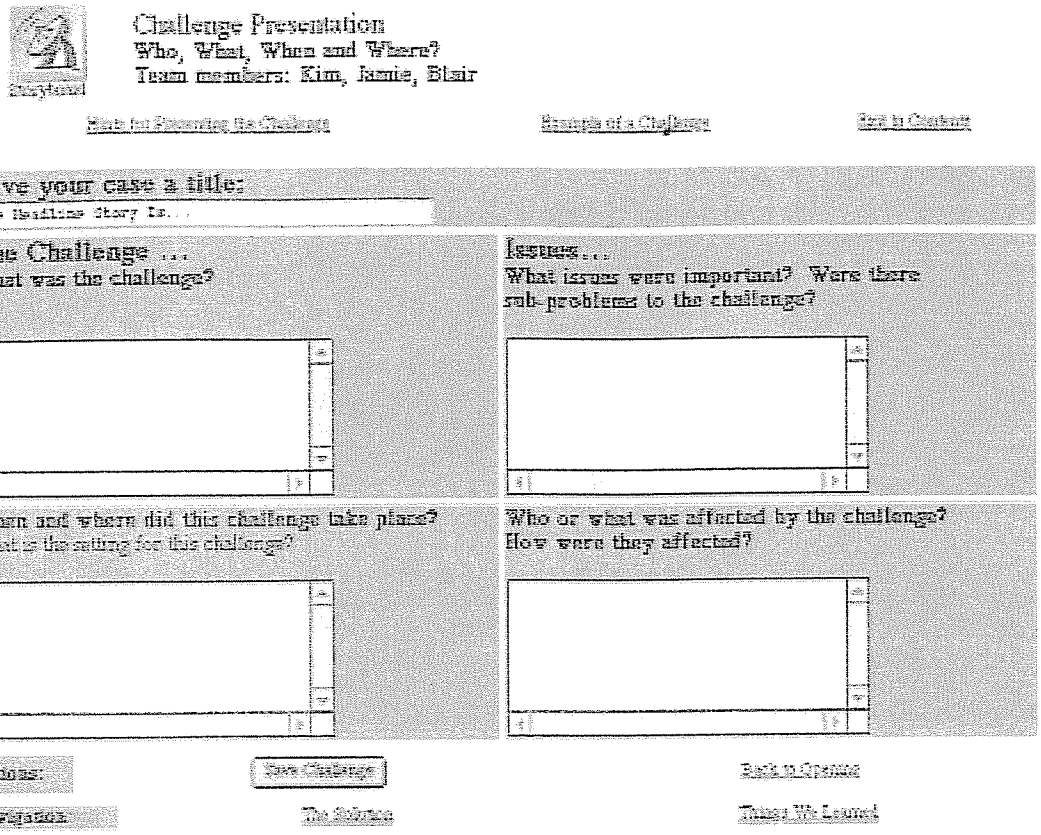

FIGURE 32.8. StoryBoard Author.

they learned it. Figure 32.9 shows our first attempts at helping students write stories about what they learned about science concepts.

The intention was that students would use StoryBoard Author to prepare presentations about their projects for their classmates. As with the DDA, the tool prompts for the kinds of things they should include in their presentations. After presentation to the class and discussion that helps them better articulate what they meant, they go back to software, revise their presentations, and publish them for others to learn from.

Experience with the different tools in Smile has allowed us to learn how to structure this tool better, and a new iteration, called Lessons Learned, will become part of Smile's tool suite. In addition to the functions initially included in StoryBoard Author, this tool will focus more on helping students write more technically—to be more specific about what they learned and to help them use scientific terminology and phraseology in their summaries.

32.5.1.5 Lessons Learned About Designing Case Based-Inspired Tools for Reflection. Earlier tools in each of the foregoing categories were less sophisticated in a variety of ways than later ones. Indeed, we have been able to extract several lessons about the design of these tools from experiences designing across the whole set.

All of our earlier versions of tools were far too general in the support they gave—either the full range of uses of the tool was not supported (as in the DDA and CAT) or the specific details of articulation students would have trouble with were not anticipated (as in JavaCAP and the DDA). Later versions of each tool that addressed these issues help students write more complete and more specific reports of their work, require less help from the teacher in getting to detailed reports, and raise the level of discussion in the classroom. We learned an important lesson about the design of scaffolding from this set of experiences:

Scaffolding the remembering and articulation of an experience. *Particular reasoning tasks students will engage in during case understanding and application should be identified and each scaffolded specifically according to its needs. Creating a suite of tools, each specific to a task, make both the pieces of that task and the task as a whole easier to grasp and manage.*

Two kinds of collaboration are needed in project-based classrooms for students to have the full range of productive discussions that allow them to connect their projects to the content they are learning (Puntambekar et al., 1997)—collaboration within groups and collaboration across groups. In our earlier versions of tools, we focused on providing anchored collaboration areas where discussions across groups could happen.

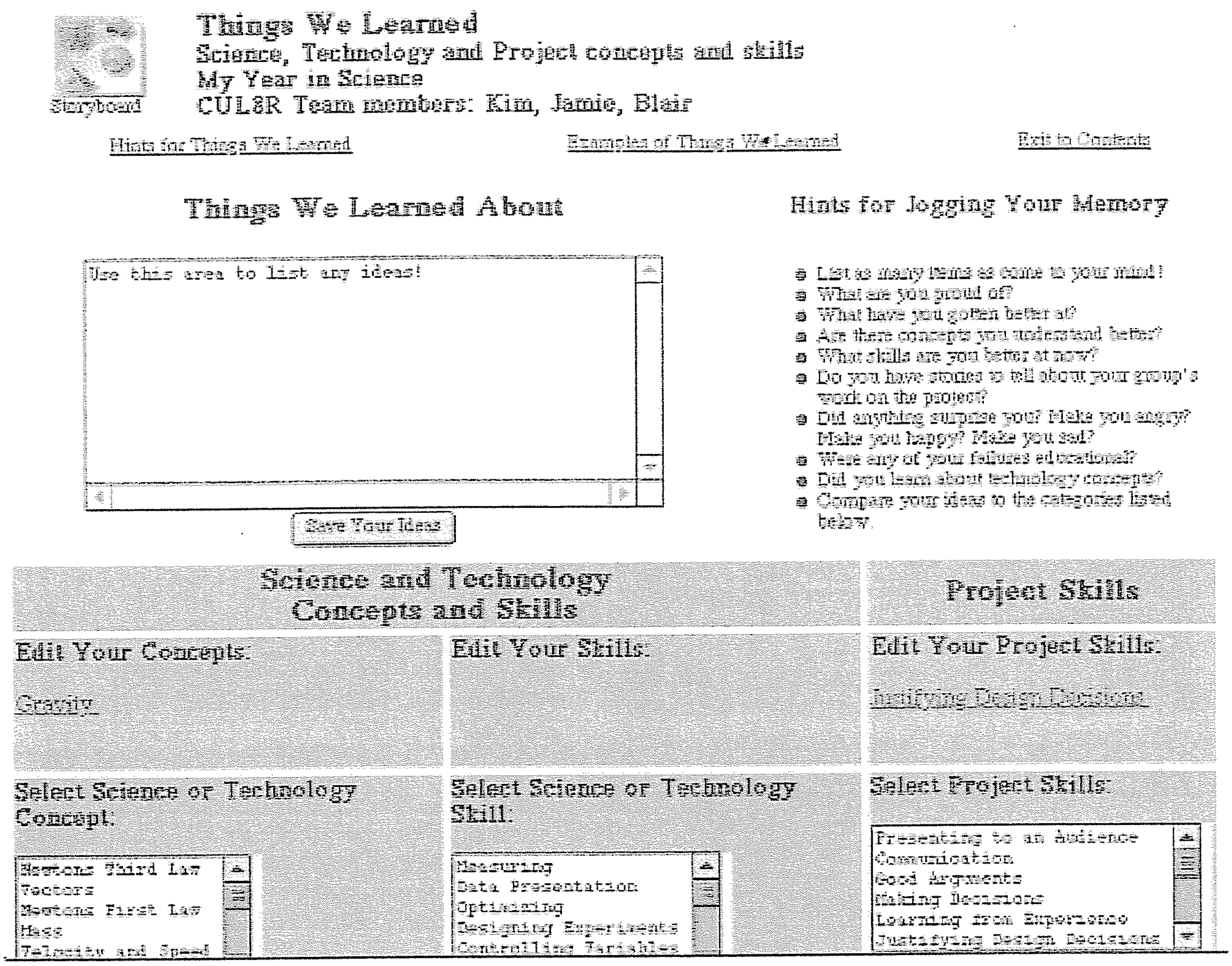

FIGURE 32.9. Encouraging students to write stories about science concepts.

We designed procedural facilitation as in CSILE (Scardamalia & Bereiter, 1991) to help students respond well to their peers. But the discussions they were able to have with their peers on line were limited (Kolodner & Nagel, 1999) because small groups had not been given the support they needed to be able to write coherent reports of their activities. From this, we learned a lesson about designing tools for collaboration.

Supporting collaborative discussions about experiences across groups. *Good collaboration across groups depends on small groups being able to articulate their ideas to others well. It is important in building collaboration tools for classrooms to support the articulation of things that will be shared across groups in order for good cross-group discussion to happen. It is often difficult for young students to articulate the stories of their experiences. When we revised the DDA to provide the specifics students needed to articulate their experiences (supporting their within-group collaboration better), the level of discussion between groups increased both in the software and in the classroom (Kolodner & Nagel, 1999).*

In our earliest tool (JavaCAP), based on advice about software-realized scaffolding (in EMILE; Guzdial, 1993, 1995), we provided the equivalent of "worksheets" for users to help them with their reflection and interpretation—pages that structured the entries they would make. But students had trouble sometimes knowing how detailed their entries should be. In later versions of our tools, we therefore added to the structuring two other kinds of scaffolding—hints to help them know our intentions of what they should be writing about in their entries, and examples to show the detail expected and to provide a template that we hoped would model what we expected. Later, when we found that some items we wanted students to think about and write about could be best expressed as lists or charts or other templates, we created specialized templates to help them frame their thinking (e.g., design decisions in the Pin-Up Tool, rules of thumb in the Experimental Results Tool). No one way of scaffolding would have worked by itself; it is the system of different scaffolds supporting each other that we think has provided success. Whereas we cannot predict the full range of systems of scaffolding that might be useful, we have learned a lesson about a system of scaffolds that seems appropriate for promoting good planning and good articulation of experiences.

The pragmatics of designing scaffolding systems. *To help learners report on their experiences and pull out what they have learned, four kinds of scaffolding working together in a system seems to work: structuring of the task they are carrying out in pieces of a manageable size, hints about what is expected for each, examples as models of the way to address each, and templates for those responses that themselves have a regular structure.*

32.5.2 Case Libraries as Resources

Case libraries are very important resources, providing models of case application to be reused as well as providing examples of successful and unsuccessful attempts at problem solving. However, applying cases to new situations is not always easy. Many times, students have difficulty recognizing that a case can be applied to a new situation; other times, students have difficulty figuring out how to adapt the case to meet their needs. Early case libraries like Archie-2 and Stable supplied cases and focused on structuring them so that learners could easily understand them. More recent case library tools, such as our own Case Application Suite, also try to help students with case application.

32.5.2.1 Archie-2 and Its Descendants. Archie-2 (Zimring, Do, Domeshek, & Kolodner, 1995) was created as a case-based design aid for professional architects. Its cases describe public buildings, focusing on libraries and courthouses. The intent was that as a designer was working on the design of a public building, he or she would consult Archie-2 periodically for advice. To get started, the architect would use Archie-2 much as architects use file cabinets, architectural journals, and the library—to find projects similar in intent to the new one and to see how others had handled the issues. The authors' intent was that an architect would browse Archie-2's library, looking briefly to see what issues other architects had addressed and how they had addressed them. An architect designing courthouses would browse the courthouses; one designing libraries would browse the libraries. Later, while addressing a particular issue (e.g., placement of the children's section in a library, lighting reading areas, access to management), the architect, they thought, would go back to Archie-2 again, this time focusing on that particular issue.

To ensure that such access could happen easily, they needed to structure cases for easy usability and accessibility. Usability was an issue because architectural cases are very large (whole public buildings). They cannot simply be presented to users in all of their complexity. Rather, users needed to be able to examine each case in parts. The big issues, then, became (a) how to divide a large complex case into easily usable parts, (b) how to provide a map of a case that would provide a big picture of the case and a map to its parts, and (c) how to provide access to a case's parts. They divided cases into parts, called *snippets* or *stories,* based on a physical and functional breakdown of the physical artifact coupled with an issue that was addressed with respect to that component and for which there was an interesting solution. The case library of public library cases, for example, had stories associated with it about placement of the children's space, lighting in the checkout area, way-finding, placement of bathrooms, and so on. Cases had tens of stories associated with them, each indexed by a relevant component of the artifact and the issue it addressed. To make it easy for users to navigate around these tens of stories a single case included, Archie-2's designers found that they had to provide several different maps of each case, as there were many ways of thinking about each.

Easy accessibility had several parts to it. (a) They wanted users to be able to ask for and then browse all cases of a kind (e.g., library, courthouse). (b) They wanted users to be able to ask for and then browse all snippets of cases that addressed the same issues (e.g., way-finding, placement of children's area). (c) From a case, they wanted users to be able to examine all stories that were about how a particular physical area or functional system was being handled. Figure 32.10 shows how that structuring looked to users. At the bottom is a spatial view of the Buckhead Library in Atlanta, with blue dots representing the spaces that had stories associated with them. The user has clicked on one dot, and a short summary of the story associated with that space in the library shows on the left. A complete version of that story shows in the top middle pane, with a general description of the problem that needed to be addressed to its left and a general description of the strategy it used to address that problem to its right. Users could see other instances of stories from other cases that addressed a similar problem by clicking in the Problem pane; they could see stories from other cases that enacted a similar kind of solution by clicking in the Response pane, and they could look at other stories about the Buckhead Library by clicking on another dot in the bottom Design frame. They also had other views of the Buckhead Library available to them that showed different ways of grouping the many Buckhead Library stories (e.g., according to functional subsystem in the building).

Though Archie-2 was designed for practicing architects, architecture faculty told its designers that they thought its cases would be useful to students working on design projects. Archie-2 was used by students in an architectural design studio who had the assignment of designing public libraries. Once they learned how to navigate Archie-2's case library, they found it quite useful. It suggested issues to focus on as well as suggestions. But, Archie's case library, as we had created it, was really useful only for assignments of library design or courthouse design (later prison design), and it was quite time-consuming to collect and format all the data necessary to build additional case libraries.

Luckily, another faculty member of the College of Architecture had an idea about how to build case libraries easily. A teacher of industrial design, he wanted to create a case library for learning about the design of simple mechanical appliances.

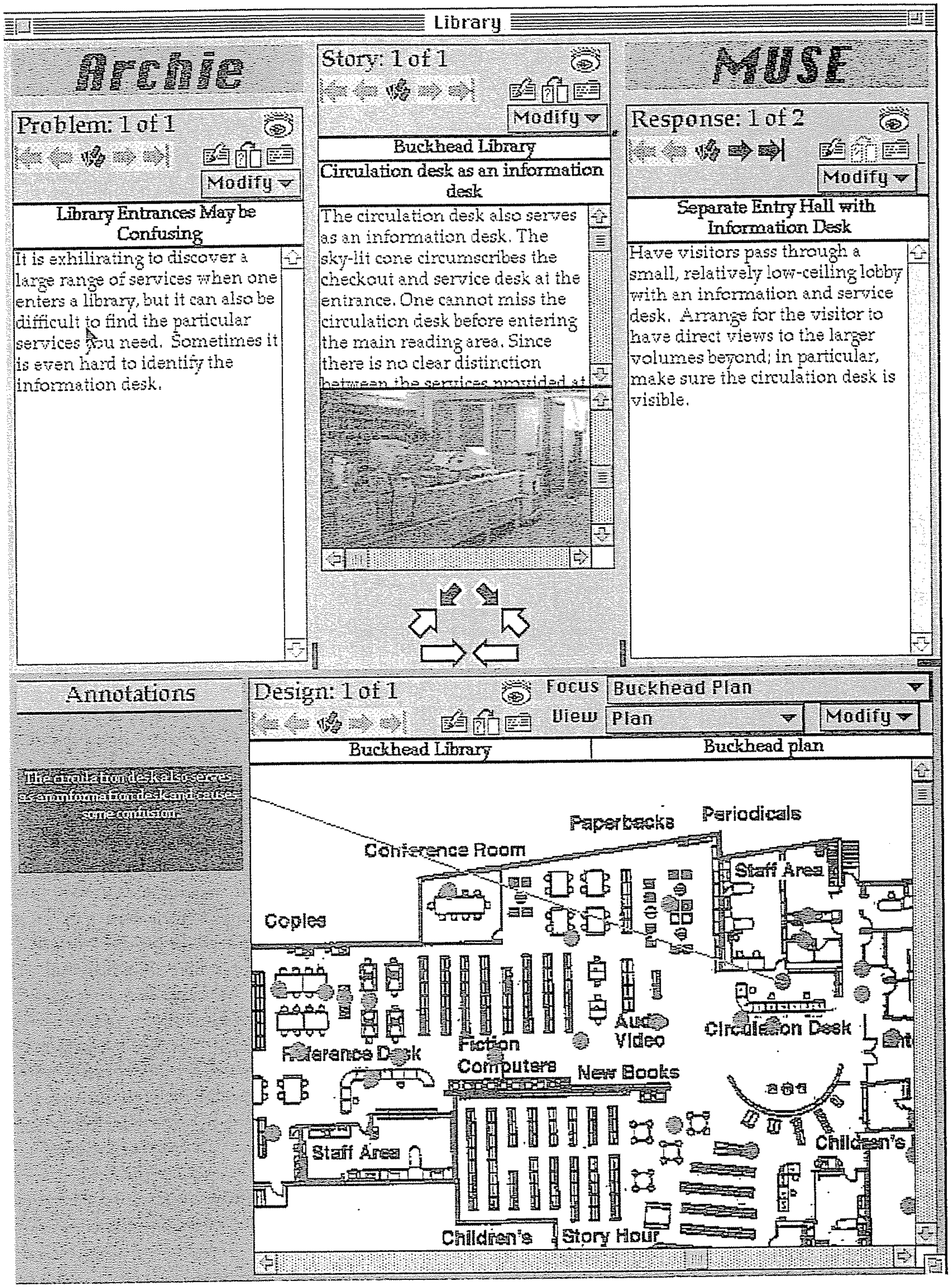

FIGURE 32.10. Archie-2.

He was teaching two classes—a lower-level (freshman) class where students were examining and evaluating such devices and a higher-level (junior) class where students were doing design. He had the students in the lower-level class record their descriptions and evaluations in a case library, using Archie-2's case-authoring tool, called DesignMuse (Domeshek & Kolodner, 1993). He was quite happy with the depth of what students in the lower-level course learned and also quite happy with the way students in the design course used the case library.

Since then, DesignMuse has been used to create libraries of skyscrapers and of airplane's hydrolic systems, and Archie-2 has been rewritten to be simpler to use. It has been used extensively in architecture studios at Georgia Tech (Zimring et al., 1995).

32.5.2.2 ***Stable.*** Complete goal-based scenarios are difficult to design in software if the learning goal for the student is a design challenge. There is no single correct solution to a design challenge, and even defining a space of correct solutions is very difficult in most design fields. The goal-based scenario approach of presenting a story at the point of failure becomes nearly impossible, because it is impossible to anticipate all failures and because failure is often nearly impossible to determine for sure.

One way around this is to build case library frameworks that are indexed by the general kinds of issues that arise in design tasks of some kind and by the kinds of failures and judgment errors that are known to come up frequently. This is essentially what was done in Archie-2—the case libraries about courthouses and public libraries were indexed by the kinds of architectural issues that arise in designing public buildings and the kinds of failures experts in the field have encountered. The case library cannot anticipate all errors that students might make, but it can provide reasonable guidance for design.

Stable (SmallTalk Apprenticeship-Based Learning Environment) is a descendant of Archie-2 designed to help students learn the skills involved in doing object-oriented design and programming. Whereas Archie-2 focused on helping students make design decisions, Stable goes the next steps in helping students learn design and programming skills. Stable uses a Web-based (hypermedia) collection of cases made from previous students' work. Students using Stable were learning object-oriented design and programming in a required computer science course. The problems that the students were asked to solve were related to the cases in Stable, at varying levels of relation. For example, students were asked to create a spreadsheet that accepted functions for cell entries, where a spreadsheet that did not accept generic functions was already in Stable. Students were asked to create a discrete event simulation of a subway system with multiple possible routes, where Stable contained several solutions to a simulation problem involving a bus system on a single basic route.

Since Stable's intent was to support skill learning, its was based on theories of apprenticeship learning (Collins, Brown, & Newman, 1989). In apprenticeship learning, a student attempts problems under the supervision and coaching of a master in the domain. The master uses a variety of methods to help the student learn. These methods are often referred to as *scaffolding*. For example, the master might model the process for the student but would be cautious about telling the student too much. Later, the master might ask leading questions to help the student focus. In successful apprenticeship learning, the master would answer questions but would not explicitly volunteer the rationale for his or her actions, to encourage students to generate the rationale themselves (Redmond, 1992). In this way, the master scaffolds or structures students' learning, encouraging them to think for themselves and solve problems on their own.

Stable was designed to provide a large amount of information but scaffolded in such a way that students were encouraged to think for themselves and request only the information they needed (Fig. 32.11).

- Each step of a design process was provided at three or more levels of detail, where the initial visit to a step was at the lowest level of detail (Fig. 32.12).
- Strategy information ("Why was this step done now or in this way?") was available, but not initially presented.
- Potential problems and solutions were presented, but mostly as links to previous steps. For example, a given step might say "A problem like this might occur" and "If it does, the cause probably occurred during this step," with a hyperlink provided to the previous step.
- Each step was linked to expert's observations on the case (e.g., "This is an example of a part–whole object relationship"), and the observations were also linked to other steps, to provide more concrete examples of an abstract observation.

it was successful in improving student performance and learning.

- Students were able to solve more complicated problems earlier in the term. We gave students a more complicated version of a problem that had been attempted in a previous term. Students did solve the problem (explicitly using Stable), and a coding of the STABLE-using students' problems showed that they were of a higher quality than the earlier problems.
- Students were able to solve design problems on a final exam better than students in previous years. STABLE-using students were asked to repair a faulty design. STABLE-using students did better on the repair task than previous students. We believe that STABLE-using students demonstrated this improved design repair skill due to their seeing more and more varied designs (e.g., multiple design solutions for the same problem) than previous students had.

Surprisingly, though, students expressed several complaints about Stable. Students were identifying cases that they wanted to compare and contrast with each other that were not already connected with each other by hyperlinks, and such comparisons were hard to do. For example, someone might become interested in how objects are created and want to look at several examples where objects were created. Or a student might be interested in how a user interface is created in an object-oriented program and, thus, want to compare how multiple cases implemented user interfaces. STABLE was designed to offer various levels of details *about a case*. It was not designed to offer much in the way of support for *comparing cases*, except through experts' observations.

The lesson learned from Stable was that a case library to support students engaged in design activities can facilitate student learning, be successful in supporting design, and be placed in a curricular setting that creates the relevant context that Schank has identified as being critical for successful learning from cases. However, Stable also showed that what students see as "relevant" is important to determine, and it may not always be evident. Several iterations of a tool are needed to ensure that all the

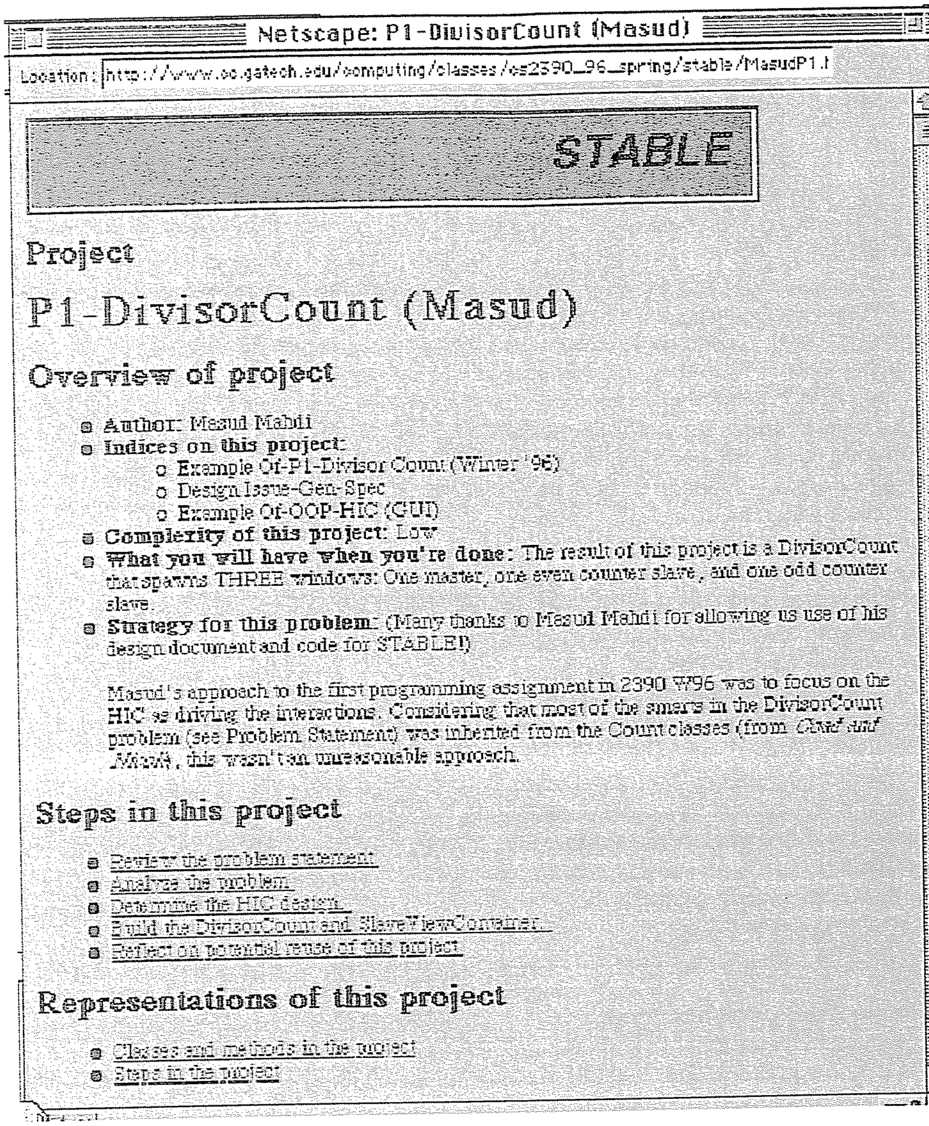

FIGURE 32.11. A STABLE project page, with steps and representation links visible.

capabilities that need to be in it for productive use are indeed included. There are open and interesting research questions on what relevance means in a case library context and how best to support it.

32.5.2.3 The Case Application Suite. The literature tells us that case application is difficult for novices. For example, given a standard physics problem, a novice problem solver will use the superficial characteristics of the problem (i.e., known and unknown variables) to search for a problem that has similar known and unknown variables and then attempt to apply the equation used to solve the previous problem to this current problem (Chi, Feltovich, & Glaser, 1981). A novice problem solver will most likely overlook the fact that there are categories of problems based on the structural characteristics they share. However, an expert in physics will recognize that the problem is of a certain category of problems (i.e., conservation of energy), drawing a diagram that represents the problem pictorially or by analyzing a diagram given. From there, the expert will recall a previous problem in that category that was successfully solved in the past, notice the strategy used to solve that problem, and either use that strategy as is or modify the strategy to accommodate the context of the problem. Important features of the problem, the relationship between those important features, and using that information to identify the type of problem they are trying to solve helps the expert problem solver identify which cases to recall from memory and which aspects of the case to apply. Without that understanding, application is impossible. Good case application requires several things (Owensby & Kolodner, 2002):

- an understanding of both the new situation and old ones thorough enough to recognize similarities between cases that might be applicable and the situation students have been presented with to which they wish to apply the case,

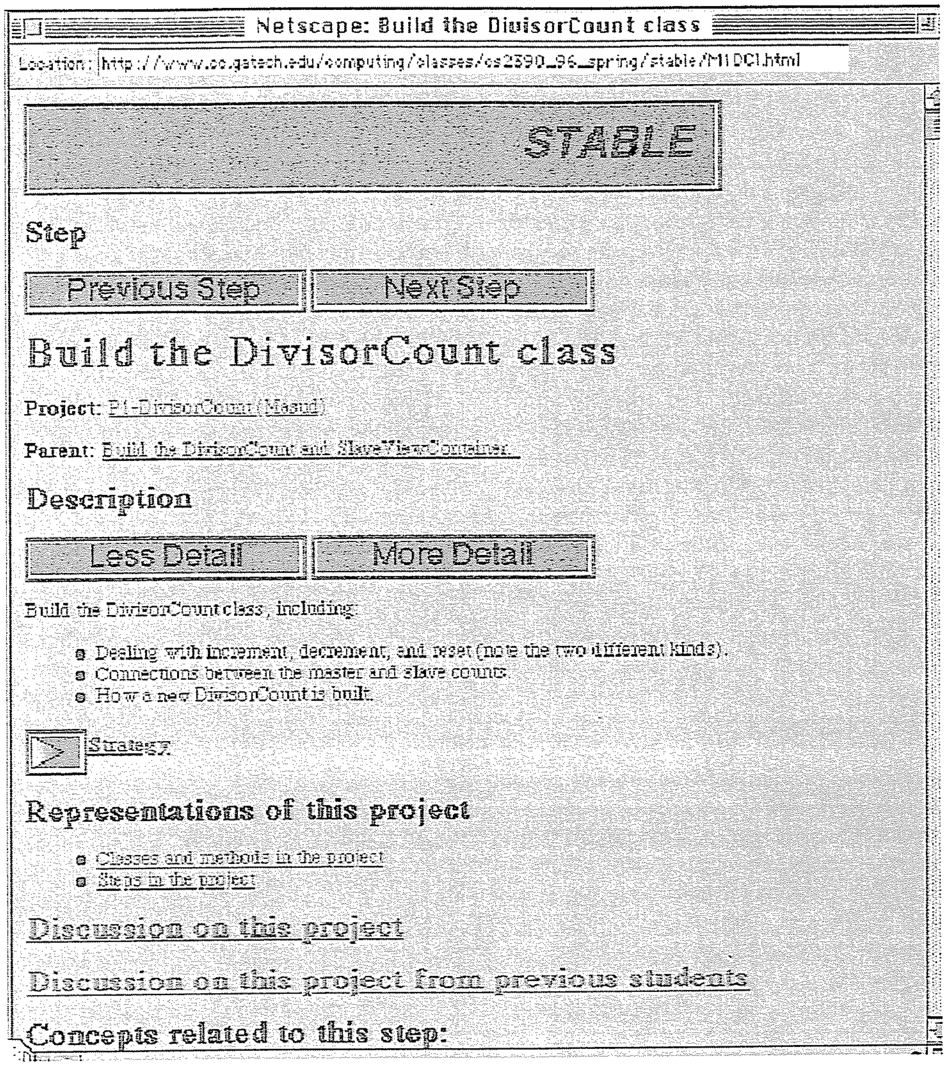

FIGURE 32.12. A Stable step page. Note the ability to increase or decrease the amount of detail on the step, as well as the link to strategy information.

- the ability to recognize what is known that might be applicable, and
- an available library of applicable cases that makes the job of remembering the right cases at the right times easier.

Keeping both the difficulties of students and the lessons learned from the earlier tool design in mind, we have turned CAT, (discussed earlier) into a suite of tools that supports three stages of case application: one for gaining an initial understanding of the case (Case Interpretation), one for thinking about how that understanding might apply in the new situation (Case Application), and one for predicting the success of the derived (Solution Assessment). Together, these tools make up the Case Application Suite. Its scaffolding aims to provide support to small groups as they engage in case interpretation and application, especially helping them to articulate and record their interpretations and the reasoning behind their case application suggestions. Hints, examples, and templates are designed to help students articulate appropriate content.

A big issue we had to address was how to help students apply an old situation to a new one. The first approach was to look to the analogical reasoning literature for advice on mapping between cases, but whereas there are several model constraints on such mapping (e.g., Gentner, 1983; Holyoak & Thagard, 1995), there was little in the way of articulation of step-by-step procedures for getting to such mapping, and such mappings seemed too hard for middle-schoolers.

Instead, a successful methodology for application that was derived for our physical science LBD units from CBR was used—the "design rule of thumb" as a representation of a lesson learned (Ryan, Camp, & Crismond, 2001). Rules of thumb in LBD's physical science classes are used to help students connect their design experiences and the process of designing to the science they are learning (e.g., To make a car go farther, make sure the wheels don't rub on the chassis because such rubbing adds friction [a negative force] to the system). Having students derive rules of thumb has resulted in students using scientific terminology and illustrating an understanding of scientific principles (Ryan et al., 2001). The Case Application Suite uses rules of thumb similarly as a vehicle for helping students make connections between expert cases they are reading and design challenges they are trying to achieve. The application process revolves around pulling out lessons learned from cases as rules of thumb, analyzing their applicability and applying them to a challenge, and then predicting the effects of the solution and assessing how well it meets the challenge. A rule-of-thumb template helps student articulate their rules of thumb. The intention is to scaffold students so that they can create detailed rules of thumb and use scientific principles to justify the lessons learned.

The Case Interpretation Tool (Fig. 32.13) scaffolds the examination and understanding of an expert case, focusing on sequencing, general understanding, highlighting of alternative solutions, the science used, and the rules of thumb that can be derived. It plays the same role as the previous Cat, but it is somewhat more streamlined. Figure 32.13 shows its structuring—with the case being interpreted on the left, the structuring prompts in the middle pane (and modeled after CAT), and hints and examples in the right-hand pane. Figure 32.14 shows the rule-of-thumb template.

Based on the rules of thumb that are created, the Case Application Tool (Fig. 32.15) helps students analyze those rules of thumb in light of their challenge and determine if those rules of thumb can be applied to their solution. Students are prompted to analyze a rule of thumb's applicability with respect to their design goals, issues and subissues, and criteria and constraints. Figure 32.15 shows this tool's general setup (with the

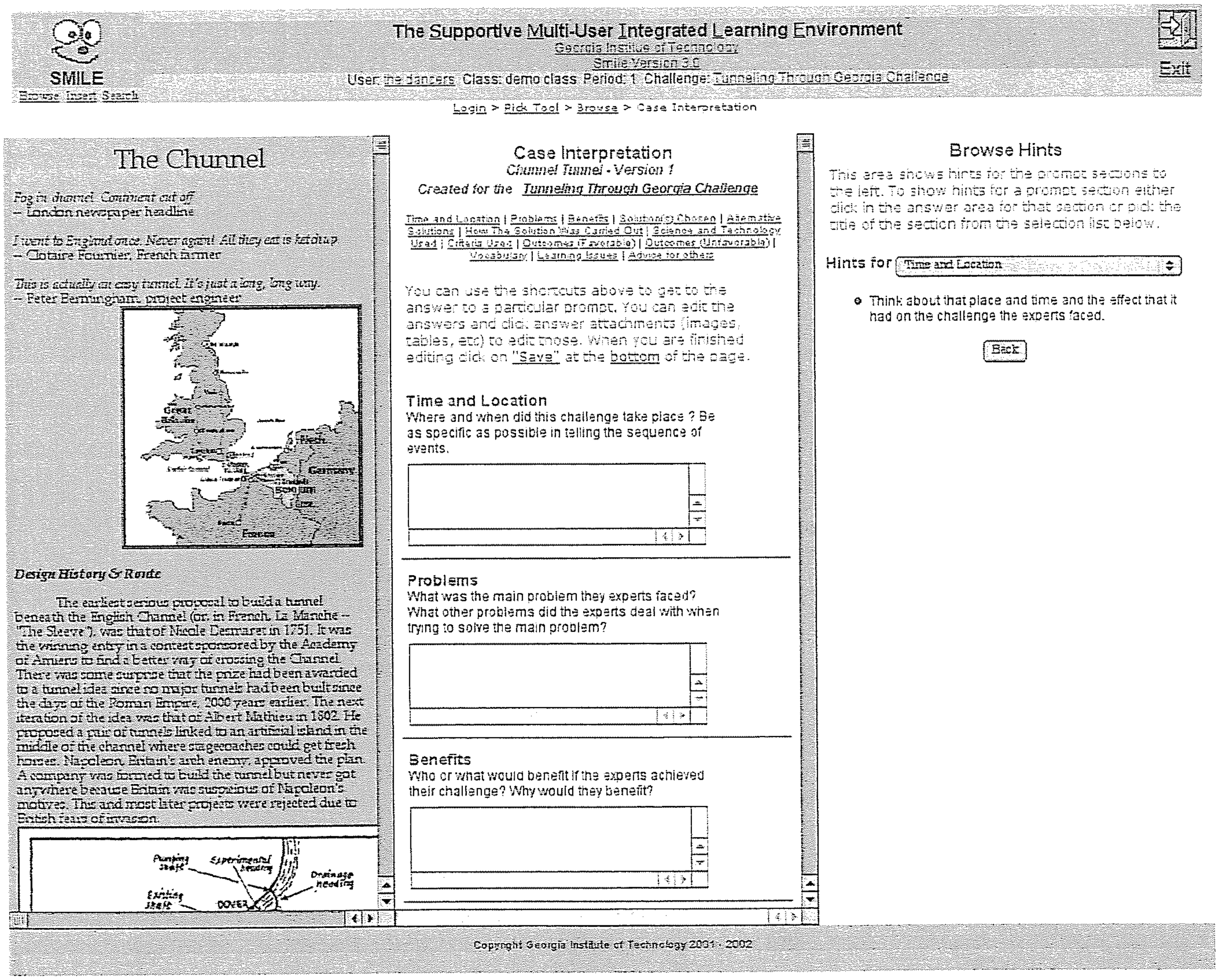

FIGURE 32.13. The Case Interpretation Tool.

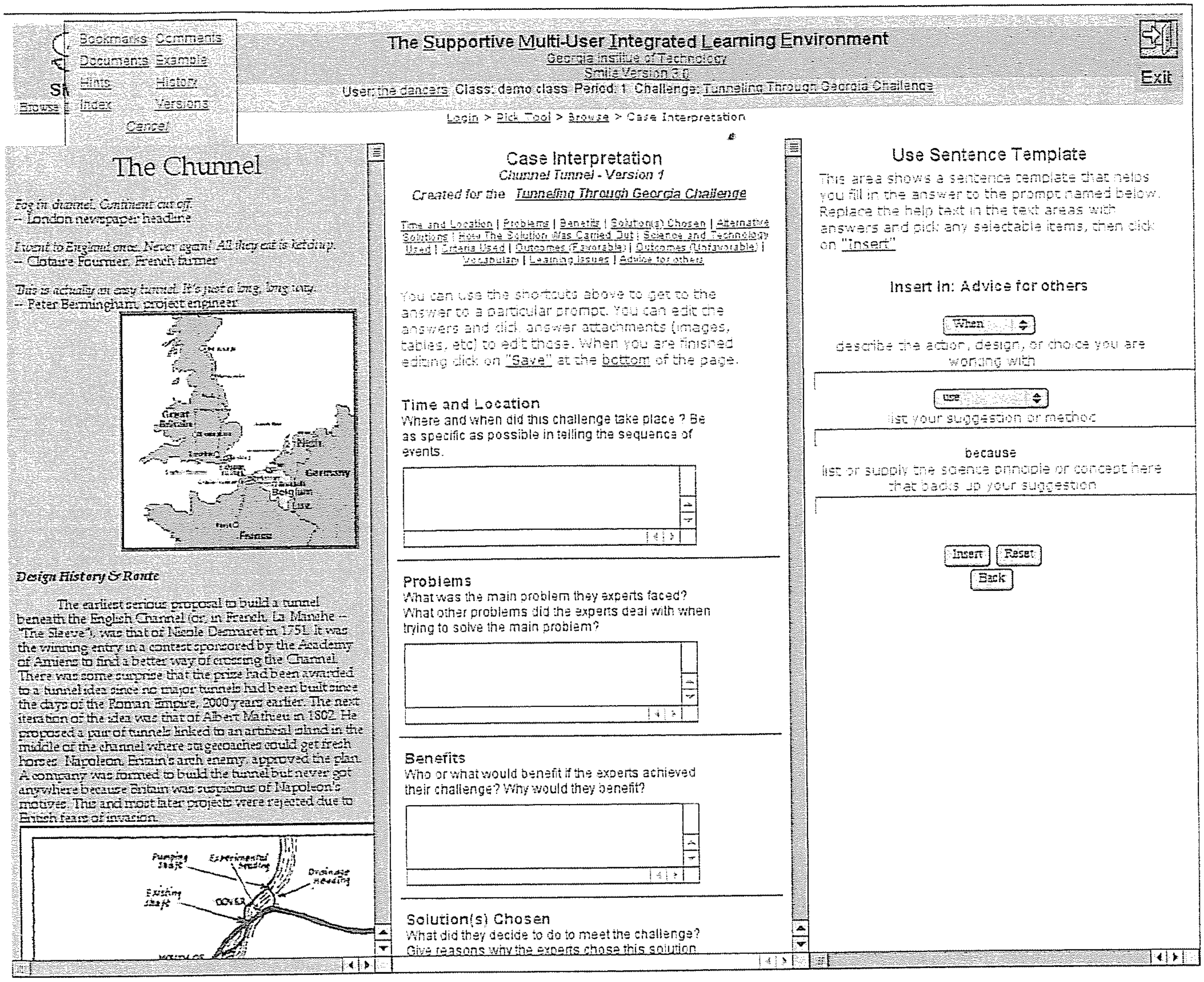

FIGURE 32.14. The Rule-of-thumb template (right frame).

case interpretation on the left, the structuring scaffolding in the middle, and hints and examples on the right). Figure 32.16 shows some of the scaffolding to help with application of a rule of thumb. Table 32.2 shows the full set of structuring prompts (middle para of each tool) across CAS' three tools.

When using the Case Application Tool, the students think about whether their solution can be improved using this rule of thumb and decide whether they should apply it. Once a solution or partial solution is derived, the results of applying a case or rule of thumb must be assessed, which is the goal of the Solution Assessment Tool (Fig. 32.17).

32.5.3 Hybrids

As part of the LBD project, researchers have created a variety of tools for helping students reason about their own experiences and reason about and apply the lessons learned from expert cases. As alluded to previously, development of early tools in that suite (e.g., the DDA) influenced the design of later tools in the suite (e.g., the tools in the Case Application Suite). Not only did researchers find that the kinds of scaffolding needed for each were similar (structuring, hints, examples, templates), but also they found that the use of the tools and their integration into classroom project-based activities seemed to be similar. Each provided tools that were good for small groups to work with to interpret their own or some expert experience, but students were not able to do a real quality job of those interpretations without also getting help from their peers and their teacher—usually as a result of a presentation and then a full-class discussion. Smile (Guzdial et. al., 1997; Nagel & Kolodner, 1999) was designed to pull together all of these functionalities across the variety of scaffolding tools. It includes in it refined versions of each of the original tools described previously, and as

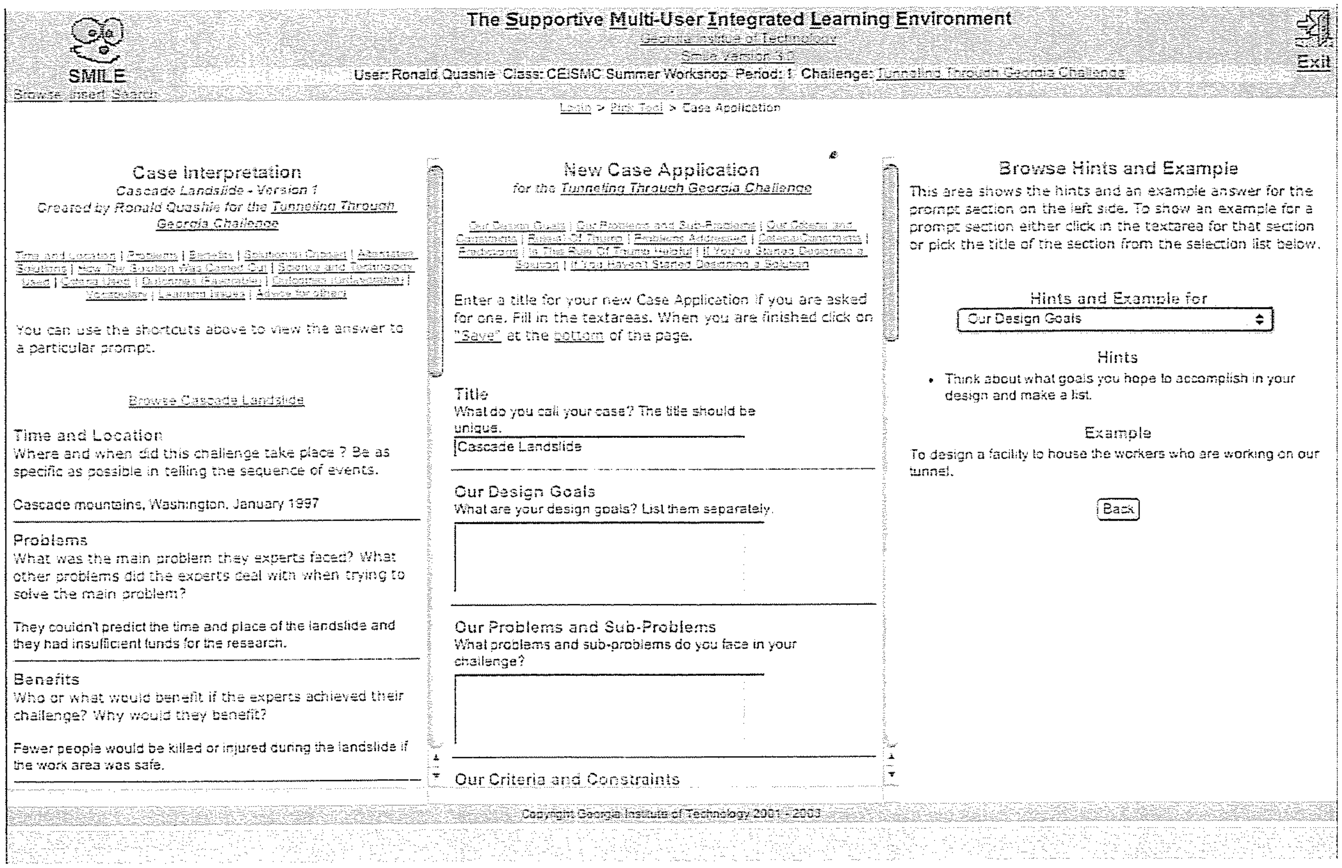

FIGURE 32.15. The Case Application Tool.

a whole, its tools promote the many kinds of reflection that CBR suggests are needed to learn from experience. Its suite of tools supports the full range of design discussions suggested earlier, helping students plan investigations and designs and interpret investigative results and behaviors of solutions in progress. Students use its tools while planning and to prepare for poster sessions, pinup sessions, and gallery walks. SMILE also includes the Case Application Suite (for guiding interpretation of expert cases), and it will include the redesigned StoryBoard Authoring tool, Lessons Learned (for summarizing over an extended project experience and extracting lessons that can be learned from it).

Each of Smile's tools helps students organize their thoughts and provides prompting in the form of hints and examples to help them make their presentations technical and complete. As discussed earlier, CBR informs on the content of that scaffolding—the structuring and prompts we provide are those needed to make connections among their goals, their plans, and what happened, to connect the disparate parts of their entire design experience, and to analyze their experiences so as to be able to index them as cases for future use. Students collaborate in groups as they use SMILE. Once they write up a presentation, they publish it in SMILE'S library for public access and comment, facilitating collaboration across groups and access by students to their peers' experiences as well as their own. Although SMILE was designed for LBD classrooms, it provides facilities appropriate for any project-based inquiry classroom, whether in science or in other subjects. In any inquiry environment, students design, run, and report on investigations, plan solutions to project challenges, make their solutions work, and extract out what they have learned. Smile's tools are designed to help students get more from those experiences by reflecting on them better.

32.6 RESULTS

We have already discussed many lessons learned about the design of tools for collaborative reflection. This section discusses some of the evaluation results collected from pilots and field tests of CBR approaches and the use of case-based learning aids and other lessons we can glean from them. Many of the findings discussed are preliminary, consisting of mostly formative assessment around issues of usability and general student performance; the most research has been done with respect to LBD. Nonetheless, the findings for these systems, with respect to the usefulness of and support provided by case-based learning aids, are promising.

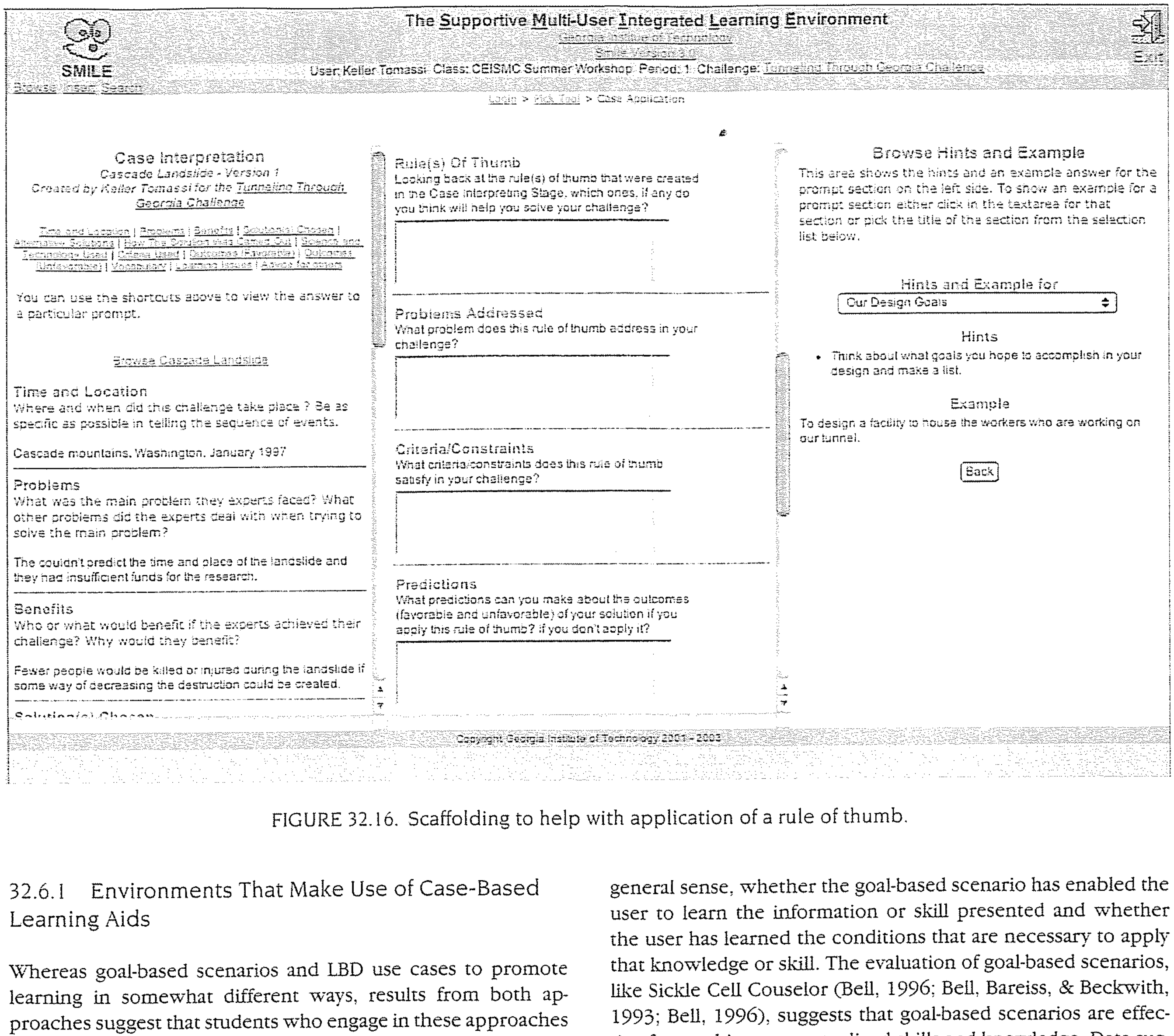

FIGURE 32.16. Scaffolding to help with application of a rule of thumb.

32.6.1 Environments That Make Use of Case-Based Learning Aids

Whereas goal-based scenarios and LBD use cases to promote learning in somewhat different ways, results from both approaches suggest that students who engage in these approaches have a better understanding of the domain and are more capable of applying that knowledge in useful ways than comparison groups.

32.6.1.1 Goal-Based Scenarios. Goal-based scenarios are designed around the idea that the best and most connected learning takes place when appropriately contextualized knowledge is learned in the context of actively pursuing a meaningful problem-solving goal that employs that contextualized knowledge. Because goal-based scenarios are designed around the task learners are working on and anticipating the learners' goals when working on the challenge, the case libraries used in goal-based scenarios are indexed based on where learners are within the task and what support they may need at that point. In particular, designers of goal-based scenarios want to know whether the indexing used is appropriate and helpful and, in a more general sense, whether the goal-based scenario has enabled the user to learn the information or skill presented and whether the user has learned the conditions that are necessary to apply that knowledge or skill. The evaluation of goal-based scenarios, like Sickle Cell Couselor (Bell, 1996; Bell, Bareiss, & Beckwith, 1993; Bell, 1996), suggests that goal-based scenarios are effective for teaching contextualized skills and knowledge. Data suggest that the counseling/role-playing aspect of goal-based scenarios effectively promotes understanding and learning the circumstances in which new knowledge is useful, compared with presenting comparable material without the counseling/role-playing aspect. The principal result is that embedding the target knowledge within the cover story promotes recall of relevant information (Bell et al., 1994).

32.6.1.2 Learning by Design. Our hypothesis is that learning environments that encourage the natural use of CBR to achieve challenges of real-world complexity and that are orchestrated in ways that promote repeated practice, promote articulation of the skills and practices being used, and explicitly encourage reuse of lessons learned from old experiences will promote transferable learning, of both content and skills. LBD was designed with this hypothesis in mind, and

TABLE 32.2. Case Application Suite Prompts

Case Interpretation Tool	Case Application Tool	Solution Assessment Tool
Time and Location Where and when did this challege take place? Be as specific as possible in telling the sequence of events. **Problems** What was the main problem they experts faced? What other problems did the experts deal with when trying to solve the main problem? **Benefits** Who or what would benefit if the experts achieved their challenge? Why would they benefit? **Solution(s) Chosen** What did they decide to do to meet the challenge? Give reasons why the experts chose this solution. **Alternative Solutions** Were other ways of meeting the challenge that were considered? Why were they not chosen? **How The Solution Was Carried Out** How did they put the solution into practice? What steps did they take to carry the solution out? **Science and Technology Used** What science and technology were used in choosing the solution? In putting the solution into practice? **Criteria Used** What criteria were used to select a solution? To select hoe the solution would be put into practice? **Outcomes (Favorable)** Were any of the outcomes favorable? What short-term effects did these outcomes have? Long-term? **Outcomes (Unfavorable)** Were any of the outcomes unfavorable? What short-term effects did these outcomes have? Long-term? **Learning Issues** Do you know everything you need to know about this case to move on? Do you have any questions about whether this case can help you solve your challenge? **Advice For Others** Do you have any 'rules of thumb' for others?	**Our Design Goals** What are your design goals? List them separately. **Our Problems and Sub-Problems** What problems and sub-problems do you face in your challenge? **Our Criteria and Constraints** What criteria and constraints are present in your challenge? How do they affect each of your design goals? Issues and sub-issues? **Rule(s) Of Thumb** Looking back at the rule(s) of thumb that were created in the Case Interpreting Stage, which ones, if any do you think will help you solve your challenge? **Problem Addressed** What problem does this rule of thumb address in your challenge? **Criteria/Constraints** What criteria/constraints does this rule of thumb satisfy in your challenge?... **Predictions** What predictions can you make about the outcomes (favorable and unfavorable) of your solution if you apply this rule of thumb? If you don't apply it? **Is This Rule Of Thumb Helpful** Is this rule of thumb helpful to use to design a solution for your challenge? Justify.	**Assess Solution (Design Goals)** Which specific design goals are successfully met by your new solution? Which are not? **Potential Problems That Were Seen Along The Way** Which specific issues and sub-issues are successfully met by your new solution? Which are not? **Assess Solution (Criteria and Constraints)** Were the criteria and constrains in your challenge taken into account by your new solution? How? **Things Overlooked** Were any criteria and constraints overlooked? How? **Next Steps** If design goals, issues/sub-issues, or criteria/constraints were not met, decide if your current solution covers enough to stand alone, whether it should be meshed with another solution to make a more complete solution, or if it should be abandoned.

analyses of LBD's effectiveness allow us both to evaluate that hypothesis and to gain an understanding of LBD's strengths and weaknesses.

Our design of LBD predicts three aspects of learning that stand to gain from the approach: (a) content knowledge in the target domain, (b) specific science process skills such as those involved in designing experiments, and (c) more general learning practices such as collaborative skills. Because LBD puts major focus on learning of science and collaboration practices, we have expected that LBD students would perform science and collaboration practices significantly better than non-LBD students. We have also expected LBD students to learn science content more deeply than their counterparts, but as it is notoriously difficult to show that based on multiple-choice tests, we did not know whether or not would find evidence for that.

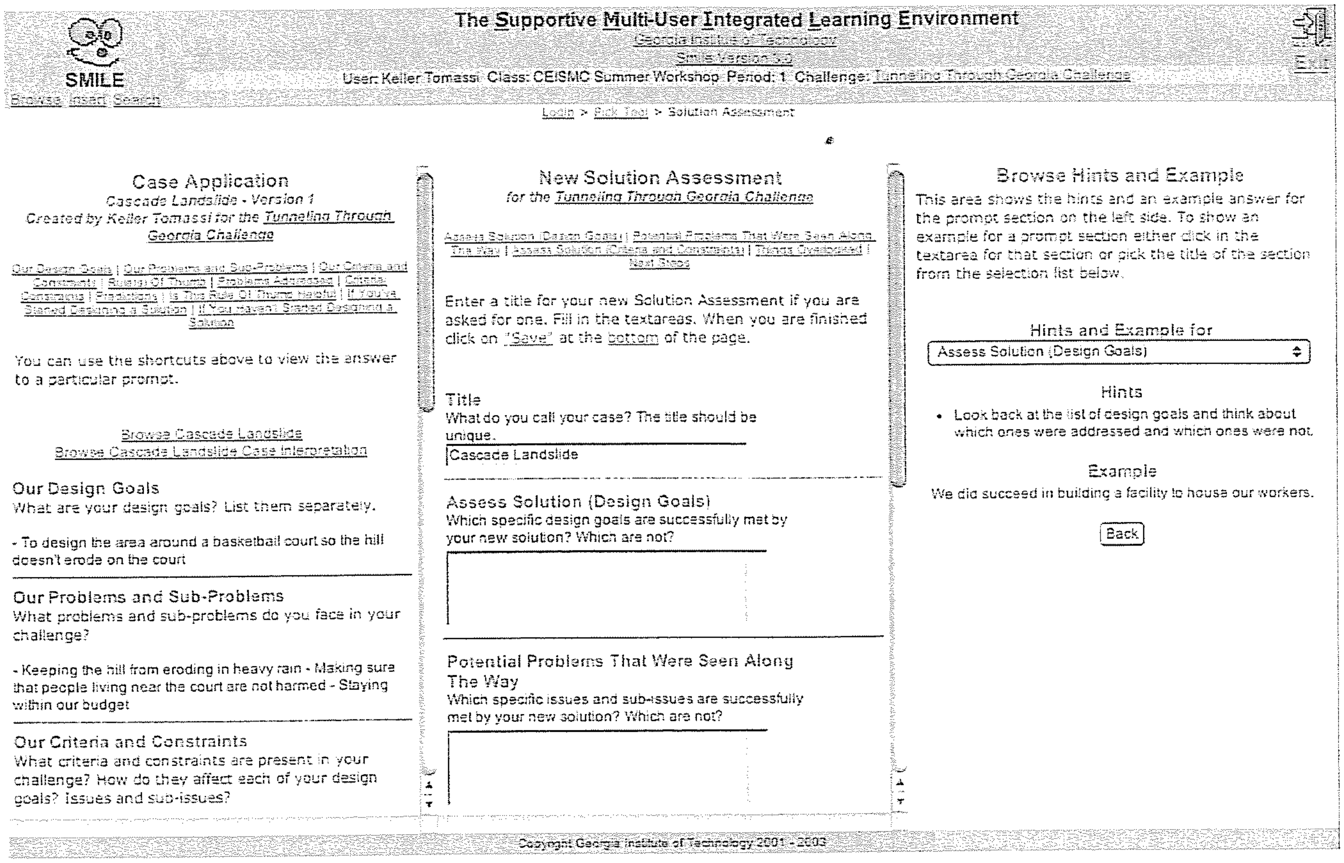

FIGURE 32.17. The Solution Assessment Tool.

We have carried out field tests in over a dozen classrooms and compared the knowledge and capabilities of students participating in LBD environments to those of students in matched comparison classes (with matched teachers). We have used two major strategies of formal assessment: (a) assessing content learning by comparing change from pre- to postcurriculum on written, mostly multiple-choice, exams, and (b) assessing students' application of science practices as they occur during data-gathering and analysis activities and during experimental design activities. Our results show that LBD students consistently learn science content as well as or better than comparison students, judged by standard-format multiple-choice tests. We are working on a deeper analysis of their learning, and preliminary results show greater understanding among LBD students (Kolodner et al., 2003), but it is early to make that claim. On the other hand, analysis of our performance data shows large, consistent differences between all LBD classes and their comparisons. While they are engaging in science activities, LBD students recall more of what they have learned than do comparison students, and they greatly outperform comparison students in their abilities to design experiments, plan for data gathering, and collaborate. Indeed, some of our mixed-achievement LBD classes outperform comparison honors students on these measures. We have found, too, through observation, that the skills acquired by LBD students often transfer to areas outside of their LBD experience and that LBD students connect with its activity structures, understand what each affords, and use them outside the LBD classroom to help them understand and investigate similar problems (Kolodner, Gray, & Fasse, 2003). See Holbrook, Gray, and Kolodner (2001), Kolodner et al., (2001), and Gray, Camp, Holbrook, and Kolodner (2001) for more detail.

32.6.2 Supports for Reflection and Interpretation of One's Experiences

In general, results for tools that support student reflection and interpretation of experiences show that although students can be scaffolded to reflect, interpret, and articulate their experiences better, designing scaffolding for middle schoolers (grades 6–8) that succeeds in having them carry out those tasks in a way that makes their reflections useful to others is tricky. We have presented the lessons we learned about how to do this in a preceding section; in this section we report on results of use of several of the tools.

32.6.2.1 Reflective Learner. Reflective Learner was designed to take a case-based approach to helping learners better interpret their experiences as they engaged in writing essays about what they had learned. The scaffolding in Reflective Learner represents CBR's suggestions at their simplest and most direct. Reflective Learner simply prompts college students to articulate their goals, how they went about achieving them, what was difficult, and what could be learned from that. Despite the simplicity of its prompts, analysis showed that students who used Reflective Learner wrote longer, more structured essays and received significantly higher grades than those who did not (Turns, 1997). Students report that the system was useful and easy to use, and although their initial use was motivated by the scaffolding provided for essay writing, their use continued because of the ability of the system to handle and manage their electronically submitted essays. Students were able to retain more of what they learned, and students became more responsible and active in their learning (Turns, 1997).

32.6.2.2 The Design Discussion Area and Its Successor Tools in Smile. The designers of the DDA and its successors have been trying to understand how to support teams of students as they engage in writing about their design efforts and discussing those efforts with each other and with the class. The first version of the DDA was trialed with 200 students in the classrooms of two masterful teachers, and although it was easy for students to use, integration of the tool into the classroom activities did not happen exactly the way the designers thought it would (Nagel & Kolodner, 1999). As a result of this, the scaffolding provided was not always the kind of support the students needed, and, thus, was not as useful as the designers had originally thought. After revising the DDA so that it specifically supported all three of LBD's presentation forums (poster sessions, pinup sessions, gallery walks), it was trialed again the following year, in the classrooms of one of those teachers. This time write-ups created with the system were far more complete and articulate, and the level of classroom discussion was far higher.

32.6.2.3 JavaCAP and Its Descendants: Storyboard Author and Lessons Learned. The designers of JavaCAP were interested in understanding how well the tool's use of narrative structure helped students reflect on and write about their experiences and whether those reflections were written in such a way that they could be used by others. Two studies of JavaCAP showed that it is usable and well understood by students and teachers, having the potential to be an excellent reflective tool, but that the authoring process needed to be scaffolded more specifically (Shabo et al., 1997). StoryBoard Author (Nagel & Kolodner, 1999) was an attempt to use what had been learned in designing other tools to provide such scaffolding. Unfortunately, teachers had students use only the "lessons learned" part of it, not taking the time to have students reconstruct their project experience before trying to remember what they had learned. Results were therefore disappointing; students were often unable to express what they had learned. On the other hand, this experience confirmed an original hypothesis in building this tool, that one cannot expect learners to remember what they have learned without asking them to reconstruct the experience as a whole. Studies conducted using several versions of Lessons Learned showed that helping students to recall the details of an experience in such a way that they can write about them in useful ways is extremely difficult. The role of affect helps to situate them within the context of the experience, but helping them to explain the lessons they learned in a way that connects the experience to the science remains a challenge (Kolodner & Voida, 2002).

32.6.3 Case Libraries

In general, results for tools that support student use of case libraries show that case libraries can be quite useful resources. They also show two somewhat surprising results. First, students were able to get as much as or more from building cases as from using them if a useful case structure was provided to them. Second, comparing and contrasting two cases to each other seems to be an important part of using cases. The model of applying one case to solve a new problem is overly simplistic, and our results show that case libraries should provide an option of looking at two cases at the same time.

32.6.3.1 Archie-2 and Its Descendants. Predictions in designing Archie-2 and its descendants were that its cases would offer resources that would help both with learning the content of the domain and with learning the how-to's of what architects do—in particular, the issues about which they worry. Its designers tried to anticipate the trajectories users would take in making their way through the system's cases and stories and tried to design navigation aids that would promote productive navigation. Its designers wondered, too, if student creation of cases would lead to deeper learning. Archie-2 and its authoring tool, Design Muse, were used in four classes—a junior-level architecture class where students were designing public libraries, a graduate level architecture class, a freshman-level industrial design class, and a junior-level industrial design class (Zimring et al., 1995). Although some users found the interface overly complex, in all three classes, users were happy with the navigation it allowed them to do and found its cases and case structures understandable. Students in the architecture class found that Archie's indexing scheme indeed helped them figure out what issues they needed to pay attention to in designing, and students in that class seemed to be able to design well using its cases. In the industrial design classes, freshman students used Design Muse to create cases, and students in the junior-level class used those cases to inform about designing. The teacher reported that he was pleased with the learning of both sets of students, but no formal evaluations were done. Most interesting, perhaps, was the graduate level architecture course. In this class, students who enaged in *building* cases seemed to be learning as much as or more than the students who were *using* the case library in their design work.

32.6.3.2 STABLE. The designers of STABLE were interested in whether use of STABLE's case library would help students learn design and programming skills better. The evaluation of STABLE was conducted to find out whether STABLE could improve student performance and learning. This evaluation focused on ease of use, student performance while using STABLE, and ability of students to use the knowledge and skills they acquired in the absence of STABLE. The evaluation showed that students using STABLE performed better while using STABLE, and they were able to employ what they learned in the absence of STABLE. Although students found STABLE sometimes difficult to use, overall they found the information in STABLE useful (Guzdial & Kehoe, 1998). This evaluation of STABLE also led to the finding that it is quite important to allow those using a case library to be able to compare and contrast cases side by side as they are working.

32.6.3.3 ***The Case Application Suite.*** The designers of the Case Application Suite are trying to understand how much and what kinds of scaffolding are needed to help students not only understand a second-hand expert experience, but also identify and use the lessons the experts learned in such a way that their solution to the challenge is the best it can be. During Fall 2003, the Case Application Suite was used in three LBD classrooms, one where the teacher had students interpret cases without much introduction and two where the teacher modeled case interpretation before students read their cases. In each classroom, some students used the tool, and some did not. Early analysis shows that, in general, the tools were easily usable, that students using the Case Interpretation Tool were able to extract rules of thumb far better than those who did not use the tool; that when the teacher had not modeled case interpretation well for students before using the tool, they were able to interpret a case well, and that even when the teacher did model case interpretation well, those students using the tool wrote more cogent summaries of their cases than those who did not use the tool. Analysis has not yet been done of the other tools in the suite, and statistics have not yet been run on analyses that have been completed.

32.7 CONCLUDING THOUGHTS

CBR makes a variety of suggestions about how to promote better learning. CBR suggests ways of making learning from hands-on activities more effective: (a) by making sure that students have the opportunity to apply iteratively what they are learning—getting real feedback about what they have done so far, being helped to explain what happened if it was not what was expected, and having an opportunity to try again and again until they are successful and come to a full understanding of what they are learning; (b) by making sure to include in the classroom rituals the kinds of discussions and activities that ask students to reflect on their experiences, extract what they are doing and learning, and articulate it for themselves or others; and (c) by making sure that students anticipate the kinds of future situations in which they will be able to apply what they are learning.

- CBR suggests resources that might be useful during learning—well-indexed libraries of expert cases and well-indexed libraries that hold the ideas and lessons learned by their peers.
- CBR suggests activities that can enhance learning in any setting—writing cases to share with others, reading the cases of experts, and preparing them for other students to learn from.
- CBR suggests ways of managing a student-centered problem-based, project-based, or design-based classroom so that students help each other move forward at about the same pace—gallery walks for sharing ideas keep everyone at about the same pace and archives of on-line cases allow those who can move forward at a faster pace to gain from the experiences of those who came before.
- CBR suggests ways of creating useful case libraries without an undue amount of up-front work by the teacher—seed a case library with several cases that model what is expected, and then have students add to that case library each year for students in the years to come.

This is a simple list. But we do not want readers to walk away thinking that CBR has all the answers, and if one simply does these things, learning will be enhanced. We hope that the discussions of the different systems and what makes them effective will help readers to understand that a great deal of planning and thought is needed to integrate these kinds of activities into a classroom in ways that work. We hope too that the discussions will provide some guidelines on how to get started.

ACKNOWLEDGMENTS

Writing of this chapter was supported in part by the National Science Foundation and the National Physical Science Consortium. Any opinions, findings, and conclusions or recommendations experessed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

References

Barrows, H. S. (1985). *How to design a problem-based curriculum for the preclinical years.* New York: Springer.

Bareiss, R., & Beckwith, R. (1993). *Advise the President: A hypermedia system for teaching contemporary American history*: Paper presented at the annual meeting of the American Educational Research Association.

Bareiss, R., & Osgood, R. (1993). Applying AI models to the design of exploratory hypermedia systems. *Proceedings of the ACM Conference on Hypertext*, pp. 94–105.

Barrows, H. S. (1986). A taxonomy of problem-based learning methods. *Medical Education, 20,* 481–486.

Bell, B. (1996). *A special-purpose architecture for the design of educational software* (Technical report 70). Institute for the Learning Sciences, Northwestern University.

Bell, B., Bareiss, R., and Beckwith, R. (1993). Sickle Cell Counselor: A prototype goal-based scenario for instruction in a museum environment. *Journal of the Learning Sciences, 3*(4), 347–386.

Bell, P., Davis, E., & Linn, M. C. (1995). The knowledge integration environment: Theory and design. In T. Koschmann (Ed.), *Proceedings of the Computer Support for Collaborative Learning 1995 Conference (CSCL'95)*. Bloomington, IN.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist, 26*(3, 4), 369–398.

Chi, M. T. H., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5*, 121–152.

Collins, A., & Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 1–18). New York: Springer.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453–494). Hillsdale, NJ: Lawrence Erlbaum and Associates.

Domeshek, E., & Kolodner, J. L. (1993). Using the points of large cases. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM), 7*(2), 87–96.

Ferguson, W., Bareiss, R., Birnbaum, L., & Osgood, R. (1992). ASK systems: An approach to the realization of story-based teachers. *Journal of the Learning Sciences, 2*(1), 95–134.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*, 155–170. (Reprinted in A. Collins & E. E. Smith (Eds.), *Readings in cognitive science: A perspective from psychology and artificial intelligence.* Palo Alto, CA: Kaufmann).

Gray, J., Camp, P. J., Holbrook, J. K., Fasse, B. B., & Kolodner, J. L. (2001). Science talk as a way to assess student transfer and learning: Implications for formative assessment. Paper presented at the Meeting of the American Educational Researchers Association, Seattle, WA, April 2001. http://www.cc.gatech. edu/projects/lbd/pubtopic.html#assess.

Guzdial, M. J. (1991). The need for education and technology: Examples from the GPCeditor. *Proceedings of the National Educational Computing Conference*, pp. 16–23. Phoenix, AZ.

Guzdial, M. J. (1993). *Emile: Software-realized scaffolding for science learners programming in mixed media.* Unpublished Ph.D. dissertation, University of Michigan.

Guzdial, M. (1995). "Software-realized scaffolding to facilitate programming for science learning." *Interactive Learning Environments, 4*(1). 1–44.

Guzdial, M., & Kehoe, C. (1998). Apprenticeship-based learning environments: A principled approach to providing software-realized scaffolding through hypermedia. *Journal of Interactive Learning Research, 9*(3/4).

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

Hammond, K. J. (1989). *Case-based planning: Viewing planning as a memory task*. Boston: Academic Press.

Harel, I. (1991). *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school.* Norwood, NJ: Ablex.

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments, 1*(1), 1–32.

Holyoak, K. J. and Thagard, P. (1995). *Mental Leaps*. Cambridge: The MIT Press.

Holbrook, J. K., Gray, J., Fasse, B. B., Camp, P. J., and Kolodner, J. L. (2001). *Assessment and Evaluation of the Learning by Design Physical Science Unit, 1999-2000: A Document in Progress.* http://www.cc.gatech.edu/projects/lbd/pubtopic.html#assess.

Kolodner, J. L. (1983a). Maintaining organization in a dynamic long-term memory. *Cognitive Science, 7*(4).

Kolodner, J. L. (1983b). Reconstructive memory: A computer model. *Cognitive Science, 7*(4).

Kolodner, J. (1993). *Case based reasoning*. San Mateo, CA: Morgan Kaufmann.

Kolodner, J. L. (1997). Educational implications of analogy: A view from case-based reasoning. *American Psychologist, 52*, 57–66.

Kolodner, J. L., & Nagel, K. (1999). The Design Discussion Area: A collaborative learning from problem-solving and design activities. *Proceedings of CSCL '99. Palo Alto, CA*, 300–307.

Kolodner, J. L., & Penberthy, L. (1990). *A case-based approach to creativity in problem-solving*. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 978–985). Mahwah, NJ: Lawrence Erlbaum Associates.

Kolodner, J. L., & Simpson, R. L. (1989). *The MEDIATOR: Analysis of an early case-based problem solver. Cognitive Science, 13*(4), 507–549.

Kolodner, J. L., & Wills, L. (1993, July). *Paying attention to the right thing: Issues of focus in case-based creative design.* Proceedings of the Ninth Annual Conference of the Cognitive Science Society.

Kolodner, J. L., Hmelo, C. E., & Narayanan, N. H. (1996). Problem-based learning meets case-based reasoning. In D. C. Edelson & E. A. Domeshek (Eds.), *Proceedings of the International Journal of the Learning Sciences (ICLS '96)*, Charlottesville, VA: AACE, pp. 188–195.

Kolodner, J. L., Crismond, D., Gray, J., Holbrook, J., & Puntambekar, S. (1998). Learning by design from theory to practice. In A. Bruckman, M. Guzdial, J. Kolodner, & A. Ram (Eds.), *Proceedings of International Conference of the Learning Sciences 1998* (pp. 16–22). Atlanta, GA.

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., Puntembakar (2003). Problem-Based Learning Meets Case-Based Reasoning in the Middle-School Science Classroom: Putting Learning-by-Design™ into Practice. *Journal of the Learning Sciences*, Vol.12, No.4, pp. 495–548.

Koloder, J. L., & Voida, A (2002). Articulating Lessons Learned: Scaffolding and Collaboration for Summative Reflection. *http://www.cc.gatech.edu/projects/lbd/pubtopic.html#software.*

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B. B., Gray, J., Holbrook, J., & Ryan, M. (in press). Promoting deep science learning through case-based reasoning: Rituals and practices in Learning by Design™ classrooms. In Seel, N. M. (Ed.), *Instructional Design: International Perspectives*. Mahwah, NJ: Lawrence Erlbaum Associates.

Nagel, K., & Kolodner, J. L. (1999). *SMILE: Supportive Multi-User Interactive Learning Environment.* (Poster presented at CSCL '99. Palo Alto, CA.) *http://www.cc.gatech.edu/projects/lbd/pubtopic.html#software.*

Ng, E., & Bereiter, C. (1995). Three levels of goal orientation in learning. In A. Ram & D. B. Leake (Eds.), *Goal-driven learning* (pp. 354–370). Cambridge, MA: MIT Press.

Owensby, J., & Kolodner, J. L. (2002, January). Case application suite: Promoting collaborative case application in Learning by

Design[TM] classrooms. In *Proceedings of the International Conference on Computer Support for Collaborative Learning, CSCL-2002,* pp. 505–506.

Palinscar, A., & Brown, A. (1984). Reciprocal teaching of comprehension-fostering and comprehension monitoring activities. *Cognition and Instruction, 1*(2), 117–175.

Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1–11). Norwood, NJ: Ablex.

Puntambekar, S., & Kolodner, J. L. (1998). The Design Diary: Development of a Tool to Support Students Learning Science by Design. *Proceedings International Conference of the Learning Sciences '98,* pp. 280–236.

Puntambekar, S., & Kolodner, J. L. (submitted). Distributed Scaffolding: Helping Students Learn from Design.

Puntambekar, S., Nagel, K., Hubscher, R., Guzdial, M., & Kolodner, J. L. (1997). Intragroup and intergroup: An exploration of learning with complementary collaboration tools. In R. Hall, N. Miyake, & N. Enyedy (Eds.), *Proceedings of Computer-Supported Collaborative Learning '97* (pp. 207–214). Toronto, Ontario.

Ram, A., & Leake, D. B. (1995). Learning, goals, and learning goals. In A. Ram & D. B. Leake (Eds.), *Goal-driven learning* (pp. 1–37). Cambridge, MA: MIT Press.

Redmond, M. (1992). *Learning by observing and understanding expert problem solving.* Unpublished Ph.D. Thesis, College of Computing, Georgia Institute of Technology.

Ryan, M., Camp, P., & Crismond, D. (2001). Design rules of thumb—Connecting science and design. Paper presented at AERA 2001, Seattle, WA.

Scardamalia, M., & Bereiter, C. (1991). Literate expertise. In K. A. Ericsson, & J. Smith (Eds.), *Toward a general theory of expertise: Prospects and limits* (pp. 172–194). Cambridge: Cambridge University Press.

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

Schank, R. C. (1982). *Dynamic memory* . Cambridge: Cambridge University Press.

Schank, R. C. (1999). *Dynamic memory revisited.* New York: Cambridge University Press.

Schank, R. C., & Cleary, C. (1994). *Engines for education.* Mahwah, NJ: Lawrence Erlbaum Associates. *http://www.ils.nwu.edu/~e_for_e.*

Schank, R. C., Fano, A., Bell, B., & Jona, M. (1994). The design of goal-based scenarios. *Journal of the Learning Sciences, 3*(4), 305–346.

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999). Learning by doing. In C. Reigeluth (Ed.), *Instructional design theories and models* (pp. 161–181) Mahwah, NJ: Lawrence Erlbaum Associates.

Schon, D. A. (1982). *The reflective practitioner: How professionals think in action.* New York: Basic Books.

Shabo, A., Nagel, K., Guzdial, M., & Kolodner, J. (1997). JavaCAP: A collaborative case authoring program on the WWW. In R. Hall, N. Miyake, & N. Enyedy (Eds.), *Proceedings of Computer-Supported Collaborative Learning '97* (pp. 241–249). Toronto, Ontario.

Silver, E. A., Branca, N. A., & Adams, V. M. (1980). Metacognition: The missing link in problem solving? In R. Karplus (Ed.), *Proceedings of the Fourth International Conference for the Psychology of Mathematics Education* (pp. 213–222). Berkeley: University of California.

Simina, M., & Kolodner, J. (1997). *Creative design: Reasoning and understanding. Proceedings of the International Conference on Case-Based Reasoning* (ICCBR97), Providence, RI.

Simina, M., Kolodner, J. L., Ram A., & Gorman, M. (1998, August). *Opportunistic enterprises in invention. Proceedings of the Twentieth Annual Conference of the Cognitive Science Society,* Madison, WI.

Spiro, R. J., Feltovich, P. J., Jacobson, M. J., & Coulson, R. L. (1991). Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. *Educational Technology, 31*(5), 24–33.

Turns, J. (1997). *learning essays and the reflective learner: Supporting assessment in engineering design education.* Unpublished Doctoral Dissertation. School of Industrial and Systems Engineering. Georgia Institute of Technology. Atlanta, GA.

Turns, J., Guzdial, M., Mistree, F., Allen, J. K., & Rosen, D. (1995a). *I wish I had understood this at the beginning: Dilemmas in research, teaching, and the introduction of technology in engineering design courses. Proceedings of the Frontiers in Education Conference,* Atlanta, GA.

Turns, J., Mistree, F., Rosen, D., Allen, J., Guzdial, M., & Carlson, D. (1995b, June). *A collaborative multimedia design learning simulator.* Paper presented at the ED-Media 95: World Conference on Educational Multimedia and HyperMedia, Graz, Austria.

Turns, J. A., Newstetter, W., Allen, J. K., & Mistree, F. (1997). *The reflective learner: supporting the writing of learning essays that support the learning of engineering design through experience. Proceedings of the 1997 American Society of Engineering Educators Conference.* Milwaukee, WI.

Zimring, C. M., Do. E., Domeshek, E., & Kolodner, J. (1995). Supporting case-study use in design education: A computational case-based design aid for architecture. In J. P. Mohsen (Ed.), *Computing in Engineering: Proceedings of the Second Congress.* New York: American Society of Civil Engineers.