Theory and Practice of Case-Based Learning Aids

Janet L. Kolodner and Mark Guzdial
Georgia Institute of Technology
EduTech Institute
College of Computing

Case-based reasoning, inspired by people, was developed as a model for creating intelligent systems – systems that could reason by reference to their previous experiences. Such systems, we said, 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 (Kolodner & Simpson, 1989). 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 (refs). More interesting to us, however, are the implications case-based reasoning holds as a model of cognition – implications about what it means to be a learner and implications about learning and education.

Case-based reasoning (CBR), as a cognitive model, values the concrete over the abstract (Kolodner, 1993). While most traditional theories of cognition emphasize how general-purpose abstract operators are formed and applied, case-based reasoning 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.

Case-based reasoning also helps us understand how we develop expertise and how an expert uses his/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?

As a theory of learning, case-based reasoning has much in common with constructivism: both claim that an individual builds his/her knowledge for him/herself from experience. Both

see learning as active; the learner plays an intentional role in deciding what to learn and in going about the activities of learning. Case-based reasoning also has much in common with constructionism (an approach to education): both approaches value learning from concrete experiences and the interpretations of the individual. But case-based reasoning goes further than both constructivism and constructionism; it defines 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. Like constructivism and constructionism, case-based reasoning has lessons for the teacher and for the designer of technological learning aids. Case-based reasoning 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 transfer so that lessons learned might be applied in more than one situation. It suggests help students might need so that they can turn their experiences into accessible and easily reusable cases in their memories.

In this chapter, we first review case-based reasoning as a model of cognition and describe its critical features. Then, we present its implications for supporting learning with and without technology. Finally, we review some examples from the research community of applying the lessons of case-based reasoning to promote learning.

I. Case-Based Reasoning as a Model of Cognition

Case-based reasoning (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 it had not previously been aware of.

Case-based reasoning 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 sub-components, 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. Upon 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.

Cases 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 might be derived from one's own experience 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 one has available for locating something in the library. 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 well-enough defined that we can recreate an index for an appropriate case when we are trying to locate something in memory. If the reasoner can't recognize a past experience as being applicable in a new situation, it will have no case to apply.

A good indexing scheme for a case-based reasoner allows the reasoner to see a past situation as being relevant to the one now facing it. Thus, a cases' 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 a bit 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 of 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 doesn't 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. One might recognize that his/her understanding is incomplete at the time of experience, or one may only come to realize that his/her

understanding was incomplete 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 he/she/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 different perspectives. This interpretation process allows the reasoner to construct a better description of the new situation than he/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 well-described as a case or well indexed. 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/it might extract new lessons from the case or identify something critical about it and re-interpret the case and update the indexes associated with it at that time.

The case processor: The case processor has a variety of responsibilities. It 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,
- re-interpreting and re-indexing an old case in light of new findings (e.g., derived by applying its lessons learned and finding that they didn't work as expected).

Each of these is important to productive use of cases for reasoning and learning.

Case-based reasoning has been explored for many years in artificial intelligence as a way of creating more intelligent computer software. A variety of the experimental case-based reasoners that the community has designed 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.[1] When CYRUS was asked a question, it

[1] Interestingly, CYRUS was created before we used the term "case-based reasoning." It was created as a way of exploring how to create a memory that a language understanding program could use well. It was

would answer 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 indexing and management of a case library. Early case-based reasoning systems, such as MEDIATOR (Kolodner & Simpson, 1989), CHEF, 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 how the role cases 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 cases 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 and ALEX show us the role of case-based reasoning 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, immersing oneself in an environment where one is likely to come upon such circumstances, and willingness to try, fail, and explain, are essential to reasoning that goes beyond the obvious.

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 neuron-like 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: (i) for organizing similar cases in the case library so that one can choose one or a small number from the category to reason from, (ii) for creation of indexing vocabulary, and (iii) for managing partial matching – to allow the reasoner to recognize that two things that are similar but not identical are a close enough match. Abstractions are extracted from concrete experience and formed as needed.

One can find more detail about case-based reasoning and early case-based reasoners in Kolodner (1993), more detail about CBR as a cognitive model from Kolodner (1993, chapter 4) and Kolodner (1997), and more detail about CBR's implications for learning and education from Kolodner (1997), Kolodner et al (1998), and Schank (in press).

II. CBR's Implications for Supporting Learning

CBR shares much with constructivism and constructionism. Both claim that what we learn is consciously constructed from our own concrete experiences. Constructionism goes on to say that experiences of actively constructing an artifact are particularly good for learning. Thus, CBR shares much with constructivism and constructionism in terms of its

---

only later that we came to the realization that such a memory had the broad implications suggested by work in CBR.

approach to supporting learning. Like both of those, it 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 looks to its cognitive model to provide explanations about how learning happens and, from there, begins to make suggestions about how to ensure that active construction has the results it affords. CBR suggests a form for what we store in memory about our experiences and the kinds of reflection that are effective for being able to reuse those experiences, suggesting several critical processes that promote good transfer.

In particular, case-based reasoning suggests five important facilitators for learning effectively from hands-on activities: (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 indices 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 (1) 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, and (2) that learners have the opportunity to iteratively move 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.

CBR's suggestions about promoting learning has informed two forms of learning supports:

- 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.

A. CBR-informed Supports for Reflection

It's been over ten years since Alan Collins and John Seely Brown first suggested that the computer could be used to support reflection (Collins & Brown, 1988). 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 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 Instructional Software Design Project (ISDP) (Harel, 1991), 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, and Linn, 1995) prompts students to think about evidence and its uses as they are creating a scientific argument. Reciprocal teaching (refs) helps students to recognize the questions they need to ask themselves as they are trying to understand something they are reading. CSILE (Scardamalia, Bereiter, and 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 looks to some difficult-to-learn skill and helps students reflect in a way that helps them learn the skill or looks to some important time for reflection and prompts students to reflect at that time.

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 (I) the kinds of problems they've faced in solving a problem or developing a skill or achieving a design challenge, (ii) the kinds of solutions they constructed, and (iii) 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, and Mistree, 1997) helps students write "learning essays" about their design experiences. Sadhana Puntambekar has described good results with paper-based, CBR-informed design journals (Puntambekar, Nagel, Hübscher, Guzdial, & Kolodner, 1997) in which students keep records of their design experiences.

Motivating students to reflect is a critical issue in learning, and the computer provides a motivation that children find compelling. For example, Amnon Shabo JAVACAP (Shabo, Nagel, Guzdial, & Kolodner, 1997) and its successor, Janet Kolodner's and Kris Nagel's STORYBOARD AUTHOR (1999), help students summarize their hands-on learning-from-design 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.

Nagel and Kolodner's DESIGN DISCUSSION (1999) 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've 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.

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 work such as Shabo's, collaborative discussion areas can 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, generate text which sounds reflective but really isn't (Ng & Bereiter, 1995). Encouraging students to reflect about things which 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's 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.

B. Supporting Learning with Case Libraries

The most common place where CBR has influenced learning tools is in the creation of case libraries. A case library offers the opportunity for students to learn from *others'* experiences. And, as implied above, a case library offers the opportunity to learn by sharing one's own experiences with others.

Case libraries can offer a variety of different 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 to focus on. 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 might 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 a user might come to the case library with. 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, the other concepts or skills it is related to, and to debug its applicability and refine its definition. But there usually isn't time in school for students to actively experience 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 they engage in; 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've 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, they 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 what to focus on**: An on-line case library's indexing system, if it is available for examination, can serve as an advanced organizer for the student or even scaffolding for how the student might think about her own cases (Spiro, Feltovich, Jacobson, & Coulson, 1991). For example, the system of indexes in ARCHIE, which helped architectural students design 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's 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's the first thing to do or to try or to explore? In many models of design, simply the definition of 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 re-use the lessons learned in novel situations is a complex meta-cognitive activity (Silver, Branca, & Adams, 1980). Cases that are about applying someone else's case can help students understand how experts re-use cases. Case libraries that prompt for the kinds of analysis that is necessary in deciding whether a case is relevant and how to adapt it for re-use can help learners develop case-based reasoning 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 (refs), 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) 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.

For cases to be a useful resource to students, the students must be engaged in an activity where 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 utilize 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's 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. One of the findings from one of the earliest case libraries explicitly designed for learning was that the graduate students who were *building* the case library seemed to be learning as much or more than the students who were *using* the case library in their design work (Zimring, Do, Domeshek, and Kolodner, 1995). Students building a case library explicitly have to deal with issues of identifying appropriate indices, identifying strategies and process elements, and decomposing the case for others to use. By making these activities explicit, we help to induce learning goals for the student that are appropriate to generating transferable knowledge (Ram & Leake, 1995). The activity of building a case library is frequently motivating for students since it is creating a public artifact whose purpose is to help future students. This is the same kind of motivating activity that Harel and other constructionists have been building upon (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. Story telling aids making sense and remembering (Schank, 1982).

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

III. Examples of CBR-informed Learning Supports

Case-based reasoning 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.

A. REFLECTIVE LEARNER

Students in project-based design courses face a huge number of challenges as part of their learning. They have to do design, while they're 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 didn't 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 upon their activities in order to learn from them (Turns, Newstetter, Allen, & Mistree, 1997). She decided to build a support for learning that directly addressed the issue of reflection.

Her tool, the 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.

The REFLECTIVE LEARNER provided scaffolding in the form of prompts to help students write learning essays in a more effective manner. Her prompts were informed by CBR. She explicitly asked 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 where a similar solution might be useful.

Her 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.

B. ARCHIE and descendents

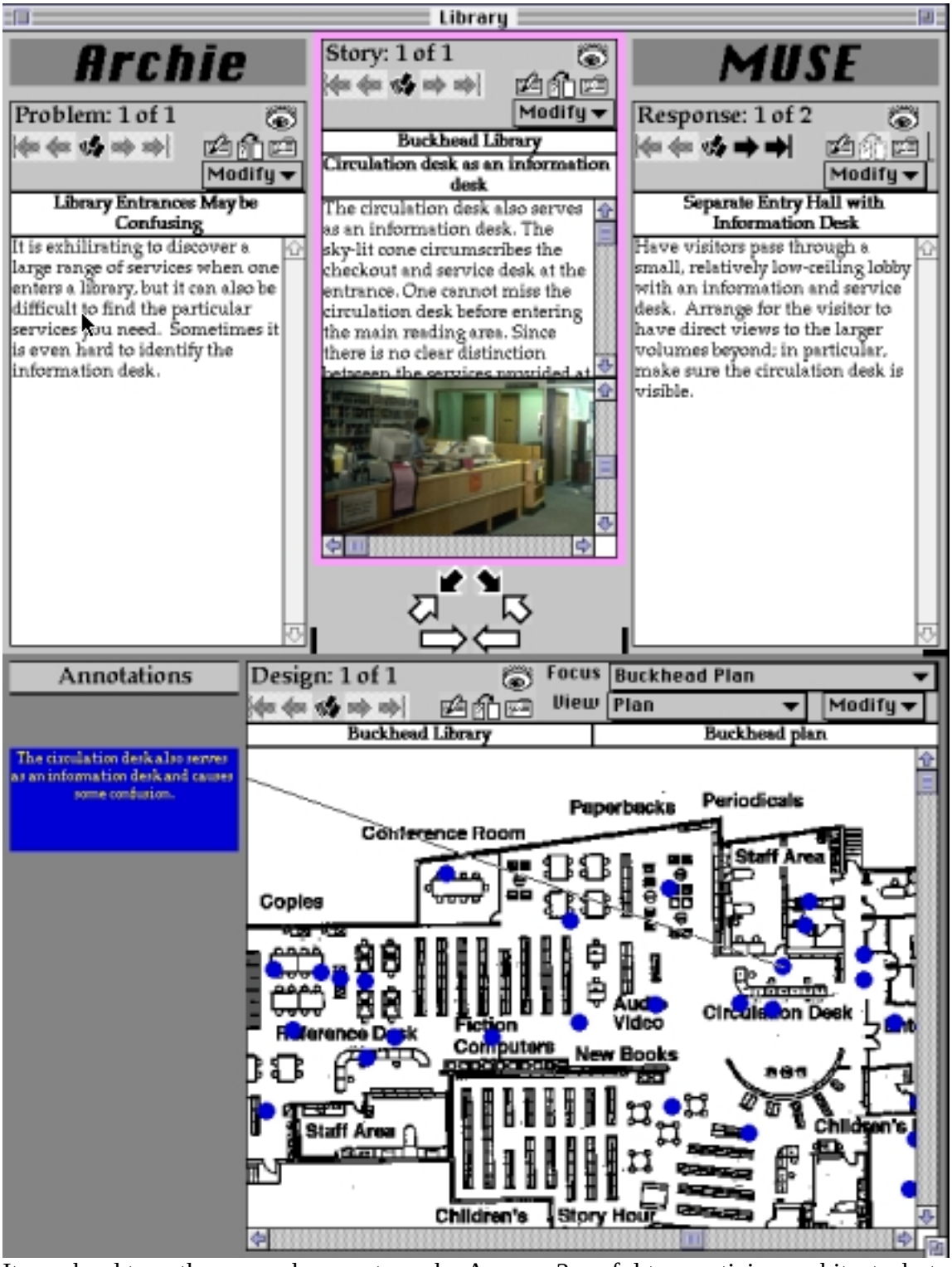
ARCHIE-2 (Zimring, Do, Domeshek, and Kolodner, 1995) was originally created as a case-based design aid for professional architects. Its cases describe public buildings, focusing on libraries and courthouses. The intent was that as a designer was working on the design of a public building, he or she would consult Archie periodically for advice. To get started, the architect would use Archie much as architects now 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. We thought an architect would browse Archie'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, we thought, would go back to ARCHIE-2 again, this time focusing on that particular issue.

To insure that such access could happen easily, we needed to structure cases for easy usability and accessibility. Usability was an issue because the cases we were dealing with were very large (whole public buildings). We couldn't simply present to users a case in all of

its complexity and expect them to be able to easily use it. Rather, we needed users to be able to examine the case in parts. The big issues, then, became (i) how to divide a large complex case into easily-usable parts, (ii) how to provide a map of a case that would provide a big picture of the case and a map to its parts, and (iii) how to provide access to a case's parts. We 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. One of our library cases, for example, had stories associated with it about placement of the children's space, lighting in the check-out 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. We found we had to provide several different maps of each case, as there were many different ways of thinking about a case.

Easy accessibility had several parts to it: (1) We wanted users to be able to ask for and then browse all cases of a kind (e.g., library, courthouse). (2) We 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). (3) From a case, we wanted users to be able to examine stories that were all about how a particular physical area or functional system was being handled.

Figure 1: Archie-2



It was hard to gather enough cases to make ARCHIE-2 useful to practicing architects, but architecture faculty told us that its cases would be useful to students working on design projects. We completed the case library of public libraries and made ARCHIE-2 available to students in an architectural design studio who had the assignment of designing public libraries. Indeed, 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 only useful 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 in 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 industrial design – in particular, the design of simple mechanical appliances. He was teaching two classes – a lower-level (freshman) one where students were examining and evaluating such devices and a higher-level (junior) one 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 (refs). 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.[2]

Since then, DESIGNMUSE has been used to create libraries of skyscrapers, and ARCHIE-2 has been rewritten to be far simpler to use. It has been used extensively in architecture studios at Georgia Tech (Zimring, Do, Domeshek, Koldoner, 1995).

C. Goal-Based Scenarios

One of the originators of case-based reasoning 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, Fano, Bell, & Jona, 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 case-based reasoning, 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 believes that they have no reason to question what they are doing and therefore no reason to want to learn anything new (Schank, 1982). 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, Fano, Bell, and Jona, 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.

---

[2] However, after he left Georgia Tech, nobody continued with the experiment, and we were unable to continue it further using the DESIGNMUSE tools. We did continue the experiment, however, in the context of STABLE (refs), to be described below.

Students learn about history or genetics or writing because they need to learn those things to successfully achieve the challenge set for them. The trick, of course, is to design challenges that both engage the students and focus them on whatever is the content and skills we want them to be learning.

The student engaged in a goal-based scenario is 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. The goal-based scenario inculcates in the students the goals that lead them to want and know how to use 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 couples about whether or not they should have children, or deciding how to refer to a leader. In all goal-based scenarios, there is 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 takes on as an additional role to clearly inform students when they have failed at their task.

This provides a second context for a case use: recognizing, explaining, and recovering from failure. A goal-based scenario indexes stories not only by content, but on their ability to explain why a student's action failed and how to recover. A story told to the student after a failure can successfully lead to learning because the student is in a context where he needs the story.

Case libraries used in a goal-based scenario focus their indexing very tightly on context in which a retrieved case will be used – what task is the student working on? what is his/her solution in progress? what difficulty is the student having? what poor answer has the student settled on? Indexes are chosen for cases in the case library 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 are more than simple case libraries; they can act as true 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). 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 student's 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.

D. STABLE

Goal-based scenarios are more difficult to build 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 more general case libraries 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 with frequency. This is essentially what we did with ARCHIE-2 – we designed a case library about courthouses and public libraries that were indexed by the kinds of architectural issues that arise in designing courthouses and libraries and the kinds of failures experts in the field have encountered. The case library can't anticipate all errors that students might make, but it can provide reasonable guidance for design.

STABLE (SmallTalk Apprenticeship-Based Learning Environment) is a descendent of ARCHIE-2 that is designed to help students learn the skills involved in doing object-oriented design and programming. While 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 in 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 rationale for his actions, in order to encourage the student to generate rationale himself (Redmond, 1992). In this way, the master scaffolds or structures the student's learning, encouraging him to think for himself and solve problems on his 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 only request the information that they needed.

- 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 least amount of detail.
- 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, in order to provide more concrete examples of an abstract observation.

Figure 2: A STABLE Project Page, with Steps and Representation Links Visible

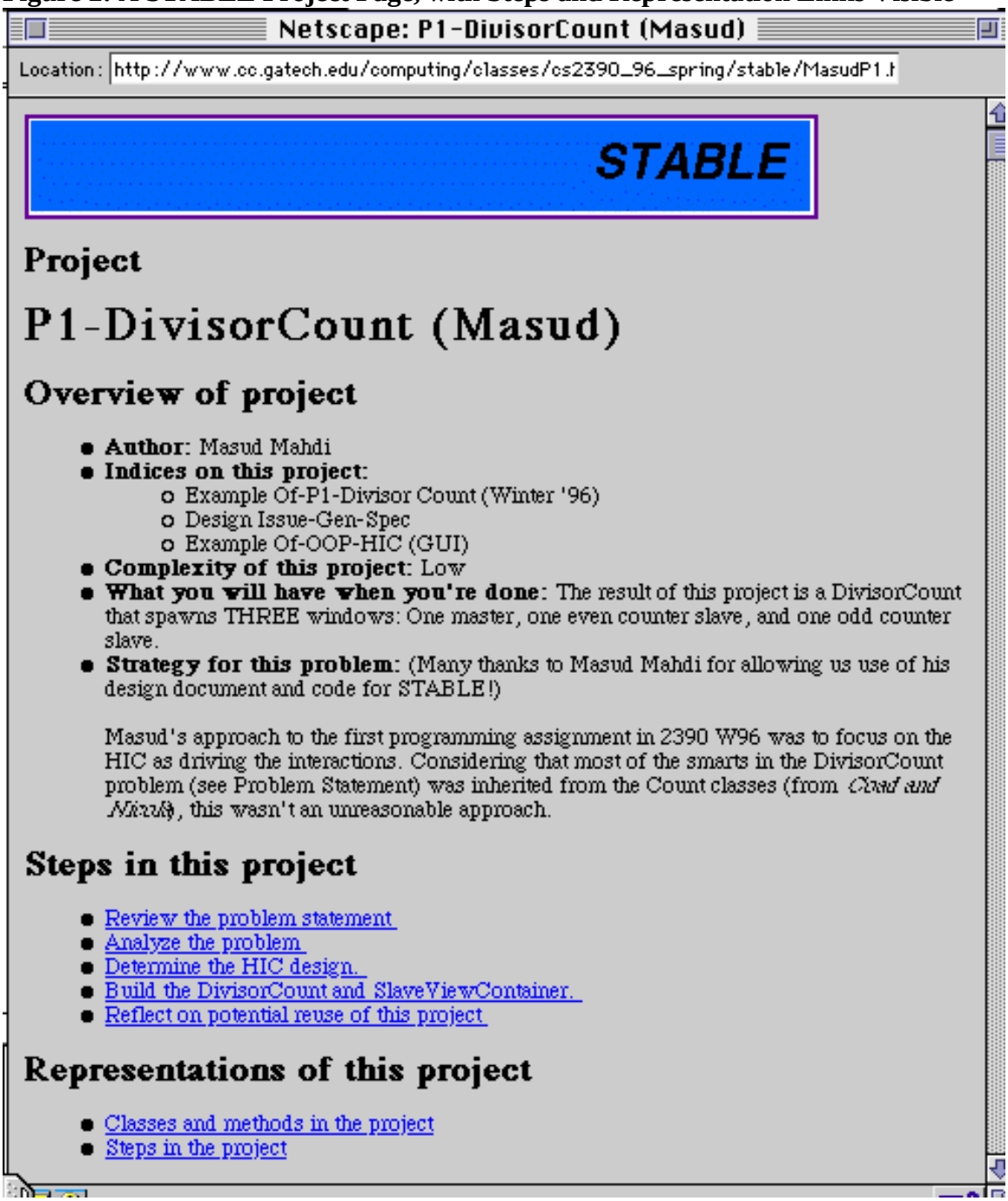
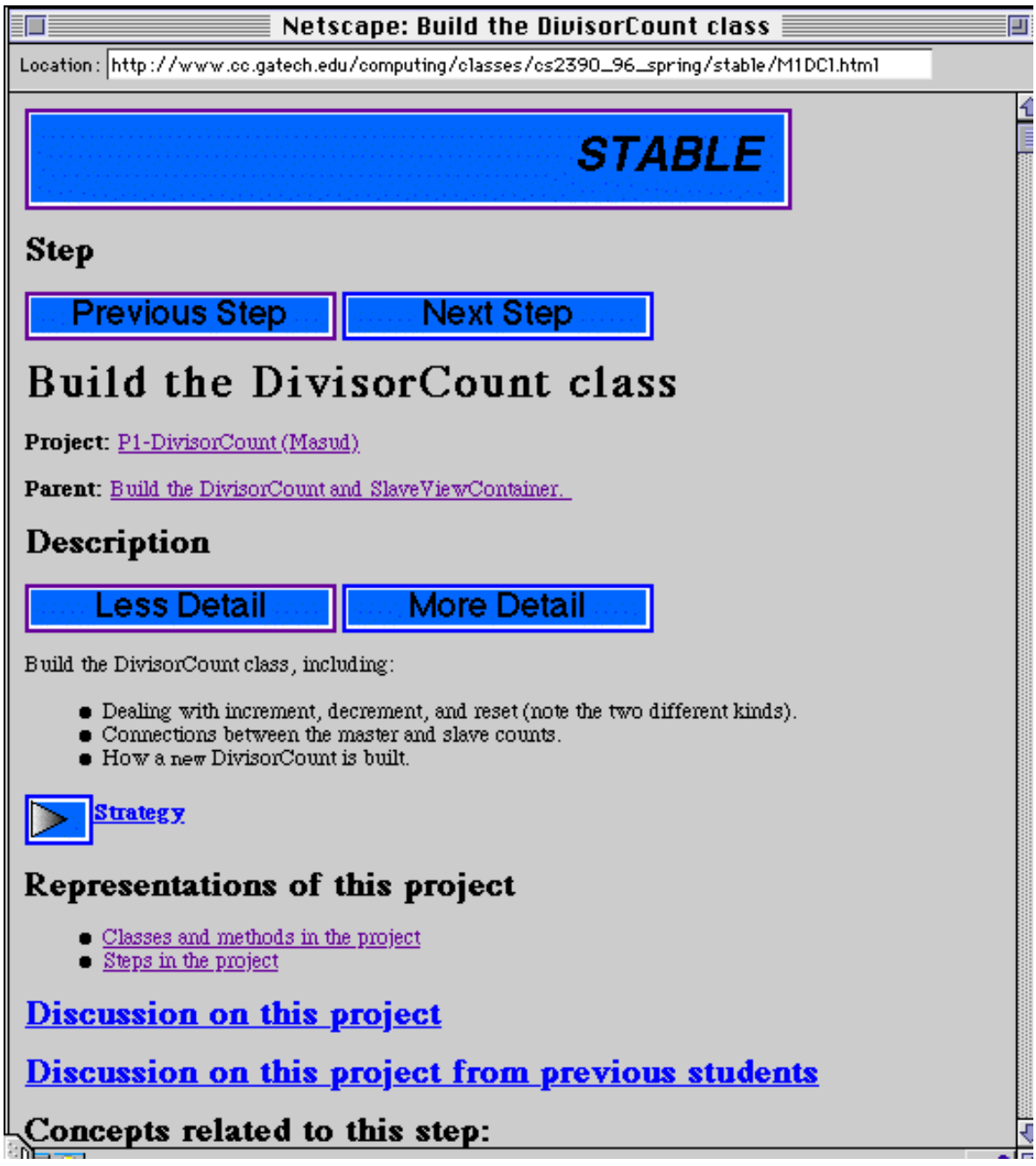


Figure 3: A STABLE Step Page. Note the ability to increase or decrease the amount of detail on the Step, as well as the link to Strategy information.



Evaluation of STABLE suggested that 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 higher quality than the earlier problems.
- Students were able to solve design problems on a final exam better than previous students on isomorphic design problems. STABLE-using students were asked to repair a faulty design. The STABLE-using students did better on the repair task than

previous students. We believe that students demonstrated this improved design repair skill due to the STABLE-using students 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. From interviews and observation of use, we found that students were identifying cases that they wanted to compare and contrast to each other that were not already connected to each other by hyperlinks, and such comparisons were hard to do. For example, students might become interested in how objects are created and want to look at several examples where objects were created. Or, students 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 which creates the relevant context that Schank has identified as critical for successful learning from cases. However, STABLE also showed that what students see as "relevant" is important to determine, and may not be always evident. Several iterations of a tool are needed to ensure that all the 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 to best support it.

E. Learning by Design and its supports for learning

Like Goal-Based Scenarios, Learning by Design (LBD) (Kolodner, 1997, Kolodner et al., 1998) takes case-based reasoning's cognitive model seriously in the design of learning environments. LBD curriculum units give students the opportunity to encounter 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 gives 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 skills they are learning.

Using guidelines from case-based reasoning, we provide (i) libraries of cases to students to use as resources, (ii) the kinds of 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, (iii) 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 ("gallery walks" and "pinups"); (iv) software tools that prompt students to explain their design decisions and design experiences to each other and get feedback from

their peers, (v) 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, (vi) 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 (vii) 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 we provide act as resources, help students create cases for others to use; help students keep track of what they've 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.

Design challenges as an approach to learning: Case-based reasoning tells us that learning requires impasses and expectation failures – to show us what we don't 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 to orchestrate failures of expectation? If one simply plans solutions, one gets no feedback to enable recognition of failures. CBR suggests that the best learning experiences will be those that afford real feedback in a timely way. Designing, building, and testing working devices provides that kind of feedback. Learning by Design'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).

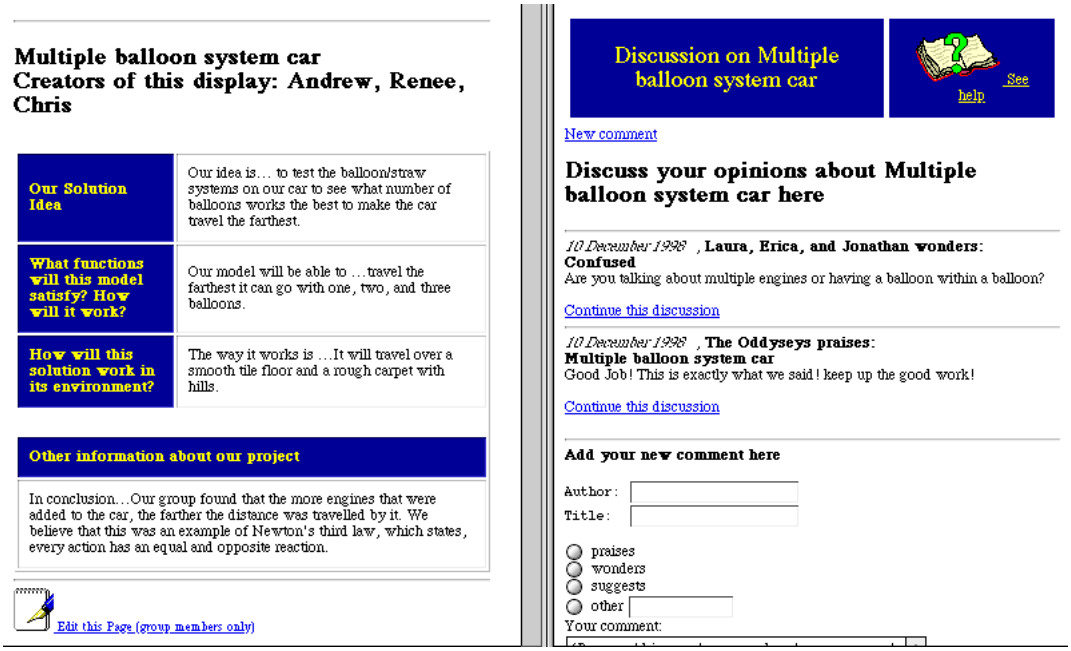
Classroom rituals that promote learning: CBR tells us that learning from experience requires reflecting on one's experiences in ways that will allow learners to derive well-articulated cases from their experiences and insert them well into their own memories. We also know that learning is most effective 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 can't, gets students thinking about what it takes to get a vehicle over a hill and the different ways they've 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. "Pin-ups," 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 pin-ups, provide a venue for presenting one's designs in progress to the rest of the class. Pin-ups and gallery walks require students to articulate what they are doing well enough for others to understand; they also provide a 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.

DESIGN DISCUSSION AREA (DDA): 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, lets a learner know that his/her knowledge base is incomplete or incorrect, prompting the engaged learner to want to revise his/her knowledge, cases, or indexing so that it works better. But school doesn't provide the time for students to have the full range of experiences that would allow them to build up a complete understanding. The gallery walk and pin-up, and their electronic extension, the DESIGN DISCUSSION AREA (DDA) (Kolodner & Nagel, 1999), are designed to help students share their experiences with each other so that they can vicariously learn from each other's experiences.

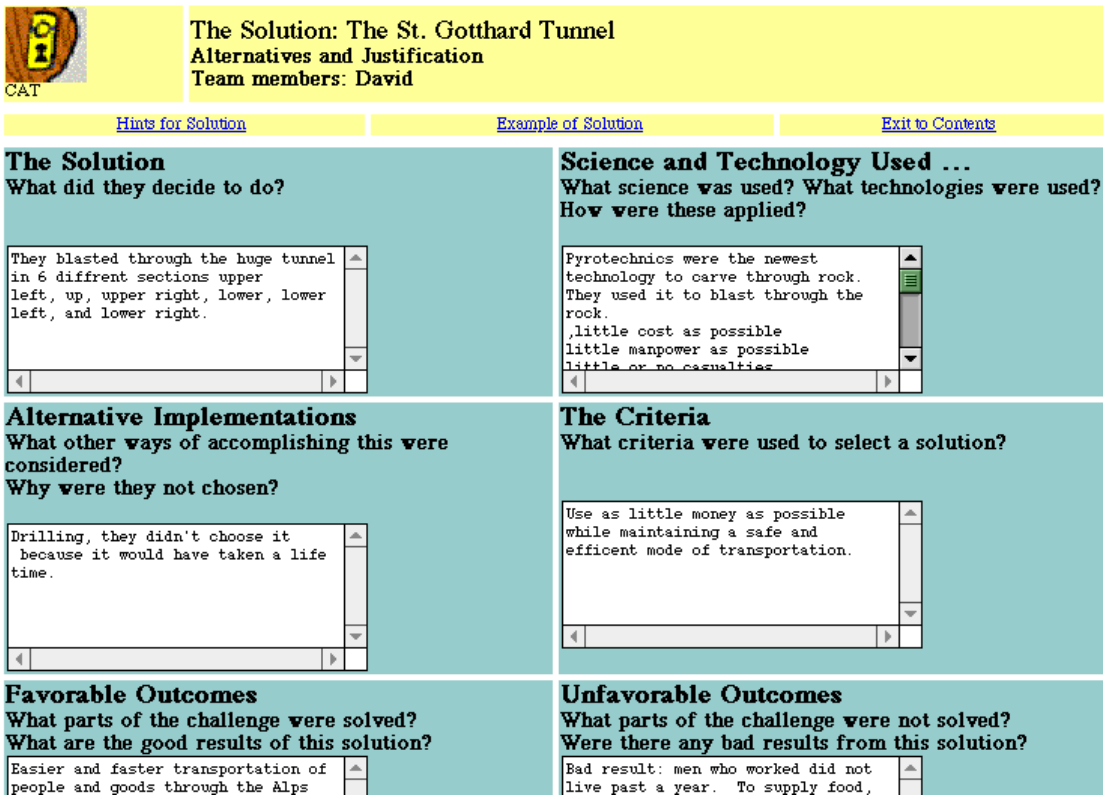
For such learning to happen, students need to be able to present their design ideas coherently, and in order for students to learn science from their own experiences and those of others, students need to talk to talk of science as they are presenting their ideas and conversing with others. DDA is designed with two learning goals in mind: (i) it helps small groups of students present their design ideas and results to others coherently and using the right kinds of vocabulary, and (ii) it guides students in other workgroups through conversations about those design ideas. Figure 4 shows a design idea and short discussion about it along with the simple prompts we provide to aid discussion. We help 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 the figure), 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, they tell about 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. DDA doesn't currently allow pictures to be added easily; we are working on that. Diagrams are certainly an important tool in articulating one's ideas.

Figure 4: Design idea with Discussion



CASE-AUTHORING TOOL (CAT): Some design challenges don't lend themselves to messing about with real materials. It is hard, for example, to mess about with erosion in any way that gets across the complexities of managing erosion when winds and currents and tides are all interacting with each other. For these kinds of situations, we have 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, 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. CASE-AUTHORING TOOL (CAT) (Nagel and Kolodner, 1999) provides that guidance. It helps students divide their challenging task into manageable chunks, and provides hints and examples for each. Figure 5 shows some of the help we give students in articulating the solution the experts came up with. We actually provide three kinds of help (as in DDA): structuring of what they need to articulate into manageable chunks, hints for each of those chunks, and examples. We provide 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 effected the people and environment.

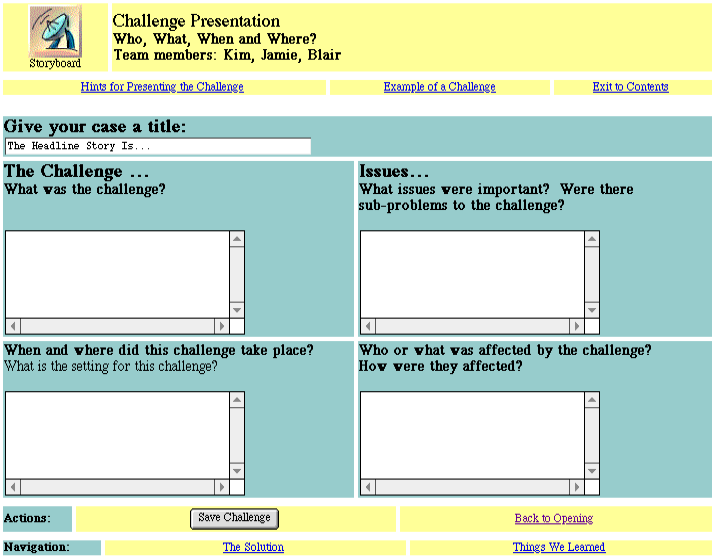
Figure 5: Case Authoring Tool



Our intention in using CAT is that students use CAT in small groups to read an article and extract what it says and to write that up for the rest of the class. We suggest they first use CAT's prompts to skim the article they are reading and extract some of its important parts, that they then use the prompts to see where they should pay special attention in reading the article and that they read those parts of it and write down what they've read. We suggest that they then do another iteration of rewriting their notes to compose a presentation of the case that others can use as a reference. They present the case to the class, and their writeup becomes a resource to the class as they all continue working on their design challenge. The case-authoring tool might also be used by the teacher to provide a set of cases to the students to use as they address challenges.

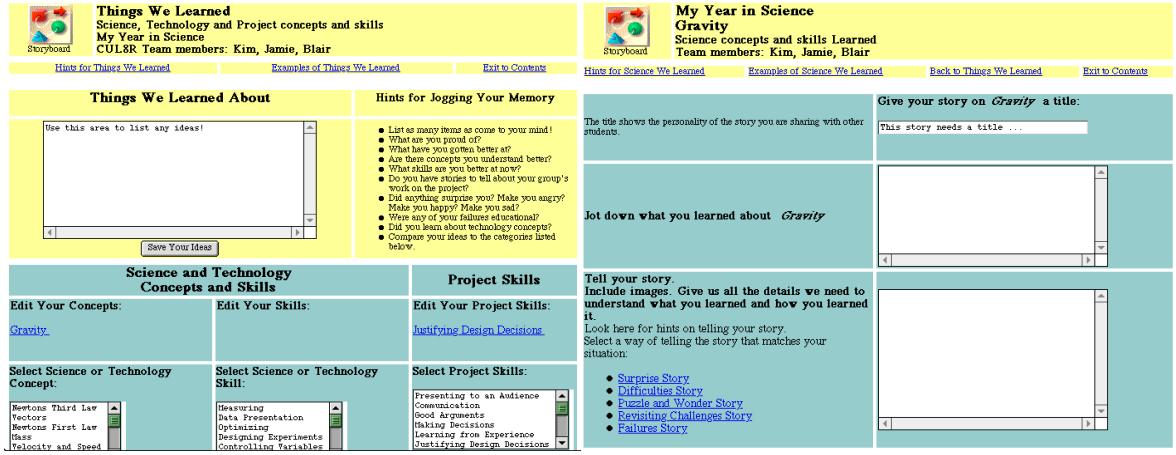
JAVACAP AND ITS DESCENDENT: STORYBOARD AUTHOR: JAVACAP (Shabo et al., 1997) and STORYBOARD AUTHOR (Nagel & Kolodner, 1999) are designed to help students reflect on an entire project experience, summarize it and put it into perspective, extract from it what they've learned, and write that up in ways that other students can learn from. We ask students first, to articulate the challenge they've 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 6 shows a description of a project challenge.

Figure 6: StoryBoard Author



To help students identify what they've learned, we ask them to think back on the things that used to confuse them but don't anymore, the things that still confuse them, surprises they encountered, things that made them angry, and things that made them happy. We ask them to jot down short notes to themselves on the computer about these things. We help 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, we provide prompts and examples to help them tell the story of what they learned and how they learned it. Figure 7 shows our first attempts at helping students write stories about what they learned about science concepts. This tool is in development and we are working on the other categories.

Figure 7: Encouraging Students to Write Stories about Science Concepts



The intention is that students will use STORYBOARD AUTHOR to prepare presentations about their projects for their classmates. As with DDA, the tool will prompt them 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 will go back to STORYBOARD AUTHOR and revise their presentations. Reports made using STORYBOARD

AUTHOR are available to others in the class and across classes for comment, ideas, and suggestions, as in DDA. At the completion of the project, the best reports and their lessons learned will be published in PEER PUBLICATIONS (Nagel and Kolodner, 1999) for classes in later years to learn from.

IV. Concluding Thoughts

Case-based reasoning makes a variety of suggestions about how to promote better learning.

- CBR suggests ways of making learning from hands-on activities more effective: (i) by making sure students have the opportunity to iteratively apply what they are learning -- getting real feedback about what they've 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; (ii) 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 (iii) by making sure 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 keeps everyone at about the same pace; 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's expected, and then have students each year add to that case library for students in the years to come.

A simple list. But we don't want readers to walk away thinking case-based reasoning has all the answers and if one simply does these things, learning will be enhanced. We hope 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 those discussions provide some guidelines on how to get started.

References

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

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.

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

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.

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.

Kolodner, J. (1993). *Case Based Reasoning* . San Mateo, CA: Morgan Kaufmann Publishers.

Kolodner, J. L., Hmelo, C. E., & Narayanan, N. H. (1996). *Problem-based learning meets case-based reasoning.* Paper presented at the International Conference on the Learning Sciences, Northwestern University.

Kolodner, Janet L. (1997). Educational Implications of Analogy: A View from Case-Based Reasoning. *American Psychologist.*

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, Georgia.

Kolodner, Janet L., Nagel, Kristine, (1999). The Design Discussion Area:

A Collaborative Learning Tool in Support of Learning from Problem-Solving and Design Activities, (submitted, CSCL, 1999).

Nagel, Kristine, Kolodner, Janet L. (1999). SMILE: Supportive Multi-User Interactive Learning Environment, (submitted, CSCL, 1999).

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.

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

Puntambekar, S., Nagel, K., Hübscher, R., Guzdial, M., & Kolodner, J. L. (1997). Intra-group 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, CANADA.

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 Unpublished Ph.D. Thesis, College of Computing, Georgia Institute of Technology.

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, Mass.: MIT Press.

Schank, R. C. (1982). *Dynamic Memory* . Cambridge; London; New York: Cambridge University Press.

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.

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, CANADA.

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

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., 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,). *A collaborative multimedia design learning simulator.* Paper presented at the ED-Media 95: World Conference on Educational Multimedia and HyperMedia, Graz, Austria, June 17-21.

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. and 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.