Case Application Suite: Scaffolding Use of Expert Cases in Middle-School Project-Based Inquiry Classrooms

Jakita N. Owensby & Janet L. Kolodner
801 Atlantic Drive, SE Atlanta, Georgia 30332
404-894-3285; 404-894-5041
{jowensby, jlk}@cc.gatech.edu

Abstract: In our research, we seek to understand how we can effectively help learners to interpret and apply expert cases through software-realized scaffolding designed to augment the teachers' modeling and coaching in a cognitive apprenticeship. We also seek to understand if exposure to the software's detailed scaffolding will result in better performance and learning of targeted skills, and if so, what those differences are. This paper presents results from a study we conducted in Fall 2002 where we sought to enhance the way students in project-based middle school classrooms used second-hand expert cases to reason. This enhancement was in the form of a software tool, the Case Application Suite, that was designed to scaffold students through some of the difficulties involved in applying expert cases. Its design is based on suggestions from the transfer and case-based reasoning literatures and the approach to education called cognitive apprenticeship. We used the Case Application Suite in Learning By Design™ classrooms where students learn science in the context of addressing interesting design challenges. In this paper, we discuss the design of the software and the study, present and analyze data regarding the software's effectiveness and student performance, and draw some conclusions about the effectiveness of the software and the feasibility of sharing scaffolding responsibilities between teacher, peers, and software.

Introduction

Learning from cases is a skill we exercise everyday, in both commonplace and technical settings. For example, we learn how to prepare a dish by reflecting on the outcomes of previous attempts at making the dish, noting what worked well, what did not work as expected, and refining our process based on the outcomes. However, in technical situations, especially for middle school learners, understanding and applying cases is more difficult (Owensby & Kolodner, 2002). We ask three questions in this study: How can we effectively help learners to interpret and apply expert cases? If learners have software to augment the teachers' modeling and coaching in a cognitive apprenticeship (Collins, Brown & Newman, 1989), will they more successfully perform and learn targeted skills, and if so, what will those differences be? Cognitive apprenticeship suggests that skills can be learned most effectively when they are modeled in their context of use by an expert followed by coaching and scaffolding as they carry out those tasks on their own. Research results from the transfer literature (see, e.g., Bransford, Brown & Cocking, 1999) support this approach, suggesting that learners learn most flexibly when they have opportunities to carry out skills in a variety of situations where those skills are appropriate and most easily when they have prompting available while they are carrying them out. We've designed a software tool, called the Case Application Suite (Owensby & Kolodner, 2002), to share responsibilities with the teacher as learners are learning to interpret and apply expert cases, providing modeling, coaching, and scaffolding for learners while working in small groups. The skill set it helps to teach is drawn from research on case-based reasoning (Kolodner, 1993).

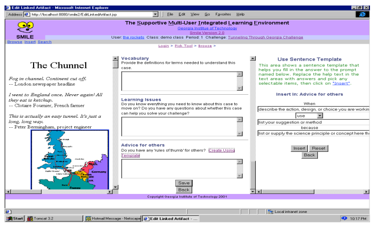
Our prediction is that learners in a cognitive apprenticeship who have just-in-time scaffolding available as they are carrying out skills in small-group settings will learn those skills more competently than those in the same cognitive apprenticeship environment who do not have that scaffolding available.

We test this hypothesis in the context of 8th-grade students learning the skills involved in interpreting and applying expert cases as they engage in Learning by Design's (Kolodner et al., 2003) *Tunneling* unit.

Background

Cognitive apprenticeship suggests that skills learning can be promoted if learners have the opportunity to see the skills they are trying to learn modeled in the context of use and then have opportunities to be coached through doing them themselves, within a context where those skills are needed and used repeatedly. It suggests three main roles for experts in skills learning: those of modeler, coach, and scaffolder. The modeler enacts skills in the context of their use, articulating steps in the task. Coaching and scaffolding involves prompting, hinting, and providing reminders as learners are carrying out those skills. The skill acquisition literature (Anderson, et. al., 1981; Bransford & Stein, 1994) stresses the importance of helping students not only understand how to carry out the skills needed for a task, but also to understand when it is appropriate to carry out those skills and in what sequence those skills should be carried out. That literature points out the importance of repeated practice, giving students numerous opportunities to engage in carrying out a task, and giving students feedback on their performance and help with debugging and refining their performance so that they can gain a better understanding of the skills needed for that task. A project-based inquiry approach to science education (Blumenfeld et al., 1991) can provide an excellent learning environment for enacting a cognitive apprenticeship. Our home-grown project-based inquiry approach, called Learning By Design (LBD; Kolodner, et al., 1998, 2003), has students learn science content and scientific and project reasoning in the context of design challenges. Important reasoning, including designing experiments, interpreting data, using data as evidence, and explaining scientifically, are modeled by the teacher, carried out by small groups over a variety of project challenges, and articulated to the rest of the class by each small group and then examined and discussed by the class in a variety of present-and-share venues. Building a working artifact was an important part of our early LBD units. When an artifact didn't work as well as expected, the opportunity to explain why promoted identifying issues that needed to be addressed and motivated a want to learn about those issues.

We've identified two important obstacles that, if overcome, could make learning from project-based inquiry more productive. First, students working in small groups are not always as rigorous and productive as we'd like. Without the help of the teacher as they are working, they are often unable to focus their attention appropriately on the reasoning they need to do. To address this issue, we designed a suite of software tools called SMILE (Kolodner & Nagel, 1999). SMILE's scaffolding fills in for the teacher as students are working in small groups, providing, for each reasoning skill they are enacting, organizational help, hints, and examples. Second, for most of earth science, it is impossible to design and build a working artifact. We have had to find a different way for them to identify issues they need to learn more about. Those units promote identifying issues by having learners read and use expert cases. We've inserted a suite of scaffolding tools into SMILE, called the Case-Application Suite (Owensby & Kolodner, 2002)(CAS), to help learners interpret and apply expert cases to their challenges. The literatures on cognitive apprenticeship and skills learning suggest that a variety of types of help are important to promoting skills learning, i.e., that a system of scaffolding and other help, each part designed to meet a specific need, is necessary to help support students as they are learning skills. CAS's system of scaffolds is designed with this suggestion in mind. CAS provides help with each of the reasoning tasks associated with case use, and it uses examples, prompting, hinting, and organizational charts to carry out scaffolding responsibilities. As for other skills, we train teachers to model the reasoning involved in interpreting and applying cases and to help students identify the ins and outs of their own reasoning about cases after sharing case interpretations with their peers. CAS was designed to play the role of coach as individuals or small groups of students are working to interpret an expert case and apply it to their challenge. Its questions prompt students to identify the lessons they can glean from a case, analyze their applicability to the new situation, consider alternatives and next steps, and assess the goodness of their proposals. Hints and examples are provided with each prompt to give students clues about the kinds of things they should consider and articulate in their reasoning. CAS includes three tools. The Case Interpretation Tool helps students identify problems the experts encountered in achieving their goals, solutions they attempted and why they chose those, what criteria and constraints informed those solutions, what results they accomplished and explanations of those, and any rules of thumb (lessons learned) that can be extracted from the experience (Figure 1). The Case Application Tool guides students through attempting to apply the rules of thumb gleaned from the case, prompting them to consider, for each rule of thumb, if it is applicable to their challenge and the different ways it could apply it to their solution. The Solution Assessment Tool helps students make predictions about the success of their proposed solutions, analyzing the impacts they expect their solution to make as well as where they expect their solution to fall short. Each tool in CAS has a left frame that displays the case students are analyzing and summaries they've written of that case; a middle frame with prompts (questions); and a right frame that displays hints, examples, comments, and templates. CAS's tools have five types of scaffolds. (1) The structure of the suite itself (3 tools) serves as a scaffold as each tool corresponds to a major step in the case application process. Together, they provide a high-level view of the processes involved in case application. (2) The prompts in each tool's center frame focus students' attention on important aspects of the case, e.g., the problems the experts faced, the solutions they created, the criteria and constraints that informed solutions, and rules of thumb that can be gleaned. (3) Hints are provided with each prompt to provide more specific help. (4) Examples are provided with each prompt to help students see what they need to be accomplishing. (5) Charts and templates serve as organizers to help students create and analyze the applicability of rules of thumb they have extracted. In addition to providing scaffolding to complement coaching provided by the teacher, CAS provides a second forum for students to share their interpretations and applications with each other. When integrated well into a classroom, the teacher models the stages of case use and coaches students, as a class, as they attempt to interpret or apply a case or assess a solution. Small groups then use the software as they carry out those processes without the teacher. Then groups present their cases to the class followed by a teacher-facilitated discussion about what can be learned from the full range of cases and the reasoning the most successful students engaged in to do their work. Then students edit their case reports based on discussions and post them for use by the entire class.


Edit Linked Artifact - Microsoft Internet Explorer
Address http://localhost:8080/smile2/EditLinkedArtifact.jsp
File Edit View Go Favorites Help
SMILE
Browse Insert Search
Login > Pick Tool > Browse >
Save
Back
Insert Reset
Back
Start Tomcat 3.2 Hotmail Message - Netscape Edit Linked Artifact - ...
Local intranet zone
10:17 PM

The Supportive Multi-User Integrated Learning Environment
Georgia Institue of Technology
Smile Version 2.0
User: the rockets Class: demo class Period: 1 Challenge: Tunneling Through Georgia Challenge

The Chunnel

Fog in channel. Continent cut off.
-- London newspaper headline

I went to England once. Never again! All they eat is ketchup.
-- Clotaire Fournier, French farmer

This is actually an easy tunnel. It's just a long, long way.
-- Peter Bermingham, project engineer

Vocabulary
Provide the definitions for terms needed to understand this case.

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? Create Using Template

Use Sentence Template
This area shows a sentence template that helps you fill in the answer to the prompt named below. Replace the help text in the text areas with answers and pick any selectable items, then click on "Insert".

Insert In: Advice for others
When
describe the action, design, or choice you are workin
use
list your suggestion or method
because
list or supply the science principle or concept here th

Copyright Georgia Institute of Technology 2001

Figure 1: Case Interpretation Tool with a Rule of Thumb Template in the Right Frame

Our Study

We are interested in learning how best to help students learn to interpret and apply cases to project challenges and in understanding the effects of adding to a cognitive apprenticeship software designed to augment the teacher's modeling and coaching. Our study collected data to answer three questions:

(1) How are students' abilities to interpret and apply cases to their project challenge affected by such scaffolding (effects with)? (2) To what extent would students' ability to apply cases in the absence of the tool be influenced by its use during a project (effects of)? (3) To what extent does the tool enable students to articulate the processes involved in case application?

To answer our questions, we piloted the Case Application Suite in the classrooms of two LBD teachers during the Fall 2002 semester, a 6th grade teacher (Mr. J) and an 8th grade teacher (Mrs. K) (Owensby & Kolodner, 2002) as part of the LBD *Tunneling* unit. In *Tunneling*, each group, consisting of three or four students, serves as a team hired to help design a tunnel for a transportation system that will run across the state of Georgia. Four tunnels need to be designed, each for a different geological area of the state –mountainous, a sandy region, and so on. They need to address several issues – at what depth to dig the tunnel, what methods to use for the digging, and what support systems are needed in the tunnel's infrastructure. Cases suggest which geological characteristics of the tunnel location they need to learn more about to address the challenge, introduce students to different kinds of tunneling technologies, and give them an appreciation of the complexity of tunnel design. Cases also provide examples of approaches that worked well that they can apply to their designs and approaches that did not work well, so they can avoid making the same mistakes. Planning to avoid those mistakes usually requires learning more about the geology of the tunnel region. We analyzed data from two of Mrs. K's classes. In each class, some student groups used the software (n=14 students; 4 groups), and the remainder used only the structural kind of scaffolding provided by the *My Case Summary* design diary page (n=33 students; 9 groups). *My Case Summary* design diary pages have a chart with four columns: *Case Summary, Problems that Arose, How Problems Were Managed, Ideas For Applying To Our Challenge*. We compared the capabilities of students who had the software available to those who did not, and we conducted individual student interviews.

Our Predictions

All students were exposed to the same teacher modeling and engaged repeatedly in case application, presenting their case interpretations to the class and receiving feedback from their peers and the teacher. Some groups (those who used the software) also had a detailed system of scaffolding available while working in small groups. The literatures suggest that the combination of teacher modeling and software coaching should result in students interpreting and applying expert cases to their challenge in more sophisticated ways. In particular, this prompting and hinting should result in students creating more useful and descriptive interpretations and applications of expert cases to their challenge. Because of the help they get while reasoning about cases, students exposed to CAS's scaffolding should also be more cognizant of their reasoning, making them better able to reason with cases when the tool is not available and better able to articulate the steps involved in reasoning about cases. In particular, given the tasks that CAS scaffolds, we predicted that students who had use of the software during small group work would be better at several things:

1. Making connections between problems and solutions in the expert case
2. Understanding criteria and constraints and identifying them in the expert case
3. Considering criteria and constraints when choosing whether it is appropriate to apply or abandon an expert solution
4. Making claims or arguments using the expert case as justification
5. Identifying relevant aspects of the case that can be applied to their challenge
6. Using the case to understand the context under which expert problems arose
7. Creating more descriptive and useful rules of thumb
8. Understanding and applying the process(es) that the experts used to implement a solution

The Environment of Use

Mrs. K's implementation of *Tunneling* was very much a cognitive apprenticeship approach. Earlier in the year, Mrs. K modeled case application skills for students as they learned about erosion, and she coached them in small groups as they applied erosion management methods to achieve the challenge of keeping a hill from eroding on a basketball court. At the beginning of *Tunneling*, Mrs. K coached the class through the reasoning involved in analyzing the Lotschberg Case with the help of a template similar to the My Case Summary design diary page. It had columns labeled *Facts, Problems, Solutions, Constraints, Ideas/Rules of Thumb, and Questions*. She put the chart on a large piece of yellow paper that she hung at the front of the classroom. She read the case with the students, and as they identified facts, problems, and solutions, she wrote them under their respective headings. Afterwards, she helped the class think about and state rules of thumb that could be derived from the case, and she wrote them on the paper. From those rules of thumb, she pushed the students to formulate questions (things they needed to learn more about in order to apply the rules of thumb) that she wrote under the Questions heading. At the end of this exercise, she informed the class that they would do the same thing with the cases that they would be assigned, and she assigned a case to each group and gave the student groups three periods to interpret their cases. Students using the software gathered around the computers; those not using the software sat around tables and read their cases together using her template. Each group made a poster and presented their case to the class. This was followed by discussion of each—both to make clear what each case could teach and to make clearer how to interpret cases well. Her implementation of case interpretation (including group presentations) lasted 6 class periods. A week and a half later, the class was ready to apply the expert cases to the solutions they were designing. The students who used the software gathered around the computers analyzing the rules of thumb they gleaned from cases they read, brainstorming all the ways that each rule of thumb could inform their solution, and determining which application made the most sense given their group's goals, criteria and constraints. Students who did not use the software used Mrs. K's template instead to analyze their challenge and design a solution—Mrs. K walked around providing help to these groups as they needed it. This analysis lasted 3 class periods. Immediately after the application phase, student groups wrote reports giving their recommendations and justifications for their sections of the tunnels. Mrs. K gave them 3 class periods to complete this. It is important to note that Mrs. K did not model application skills for students—she only helped them think about potential application of rules of thumb. This will be reflected in the results.

Data Sources and Analysis Procedures

We collected data from three sources. To understand how students' abilities to interpret and apply expert cases to their project challenge were effected by CAS's scaffolding, we observed and video-taped students using the software and we observed and video-taped class presentations and collected posters used to present their case analyses. To understand the effect of tool use on students' ability to apply cases to a new situation in the absence of the tool, we conducted a performance assessment. To understand the extent to which tool use enabled students to articulate the processes involved in case application, we interviewed students. In the performance assessment, called the Bald Head Island Challenge, each group played the role of a design team that has been hired by a construction company to investigate the feasibility of building two subdivisions on an island off the coast of Georgia. The performance assessment included five major tasks (1) finding and describing the risk factors involved in building subdivisions on the island, (2) identifying and describing possible ways to manage those risks, (3) creating rules of thumb that could be used in future projects, (4) designing a plan for testing management methods, designing, and constructing the subdivisions, and (5) making recommendations to the construction company about the feasibility of building the two subdivisions with the given constraints. Students were given a chart to fill in with five columns labeled *Describe the risk, Why is this a risk, What are ways to manage the risk, Pros, and Cons.* The chart was similar in structure to the My Case Summary design diary page and the template created by Mrs. K, and it provided some organization, but only a minimal level of scaffolding. Groups discussed their answers, and each group member was responsible for writing up his/her own answers, but we found that group members discussed their answers and then wrote up the exact same answer. We videoed group interactions in 5 groups during each of Mrs. K's two classes. We coded both group and individual performance. Two coders analyzed video-recorded group performance for the first three tasks on dimensions shown in Table 1 and for the last two tasks on dimensions shown in Table 2. A five-point Likert scale was used for each, with 1 representing no evidence of the quality to be rated and 5 representing the quality completely captures the nature of the episode. We analyzed written student performance for the first three tasks using a combination of these Likert scales and characterization elements. Characterization elements are used to solely to classify elements—they do not describe the quality of the element (e.g., Origin of Risk – This measure is used to categorize where the risk identified comes from. 1 – Risk identified is taken directly from the case. 2 – Risk identified is not mentioned in the case).

Results

Effects With the Tool in Place – Analysis of Video Presentations and Project Posters

The quality of the presentations in Mrs. K's class was very high among both the software and non-software groups. But examining their artifacts and videotapes of the presentations shows three differences. (1) The software groups connected satisfaction of criteria and addressing constraints with positive outcomes and not addressing constraints with negative outcomes. For example, in learning about the Queens Midtown Tunnel, which was built below the borough of Queens to connect Queens to Manhattan, one software group identified the following problem and solution: "They wanted to build [the tunnel] straight [through the city] but couldn't, so they continued it further underground in an S-shape under First Avenue and they took different core samples". This group noticed that the experts were not able to meet a criterion due to the constraint imposed by First Avenue being in the way which. made it impossible to build the tunnel in a straight line. Instead, they recognized that the experts had to address the constraint and change the design from a straight line to an S-shape in order to successfully complete the tunnel. They also noticed that this change in design meant the experts had to take new core samples to understand the composition of the new path before they began building the tunnel. A non-software group that interpreted the same case did not even mention the S-shape design used by the experts even though it was critical to the experts achieving their goal. (2) Software groups had more sophisticated causality to their rules of thumb. For example, the non-software groups' rules of thumb were in the form of simple imperative statements (e.g., "Control water problem", "Take core samples"), while the software groups' rules of thumb explain why (e.g., "Take core samples—they can save your life because if you hit the wrong kind of rock, you can get hurt", "You should always have an oxygen pass so the toxic fumes can get out"). (3) Software groups tended to include the process the experts used to implement the solution. For example, one software group noted that because the experts couldn't drill through the mountain like they wanted to, they had to build the tunnel in an S-shape around the mountain. Their non-software counterpart made no mention of this, even though the fact that an S-shape was chosen over other alternatives was an important facet of the solution.

Effects of Using the Tool – Performance Assessment

Tables 1 and 2 show results for the interpretation (the first three tasks) and application (the last two tasks) parts of the Performance Assessment. For the video-recorded data for Part 1, reliability was 89% and results show significantly better performance by software groups on four dimensions.

Table 1. Performance Assessment Results for Part 1 – Interpretation

Coding Characteristic (**bold** denotes significant difference p<0.05)	Software group	Standard Dev. (software group	Non-Software group	Standard Dev. (non-software group)
Recognizes that the case should be used to solve the challenge	**3.88**	**0.25**	**2.66**	**0.71**
Makes direct reference to the case to justify an argument or position	3.135	0.25	2.33	0.82
Able to identify expert problems	**3.00**	**0.00**	**2.42**	**0.61**
Able to identify expert "mistakes"	2.63	0.75	2.00	1.17
Able to identify relevant aspects of the case that can be applied to the challenge	**3.88**	**0.25**	**1.83**	**0.98**
Identifies risks based on prior experience with another LBD/software case	1.88	1.44	1.33	0.41
Able to identify criteria and constraints	3.38	1.11	1.58	0.66
Uses the case to understand the context of the risks	**2.88**	**0.25**	**1.67**	**0.61**
Identifies rules of thumb	1.00	0.00	1.00	0.00

- Software groups tended to describe expert problems on a finer-grained level than non-software (3.00 vs. 2.42, p<0.05). For example, non-software groups identified "sand" as a risk, while software groups identified the "incompatibility of the old sand and the beach with the new sand dug when the channel was deepened" as a risk, or expert problem. Being able to distinguish between those problems is important.
- Software groups tended to discuss whether a management method made sense for their challenge, analyzing how the management method would play out in their challenge and questioning each other about the feasibility of a proposed management method (3.88 vs. 1.83, p<0.05). Non-software groups discussed management methods only if they were far-fetched.
- Software groups used the case not only to identify the problems the experts encountered, but also to understand the context in which those problems arose (2.88 vs. 1.67, p<0.05). Non-software groups tended to look for keywords that they were familiar with when identifying problems and management methods. For example, one non-software student declared , "Oh!! I see erosion here—erosion is a problem," while in a similar incident, a member of a software group identified the problem as "shoreline eroding."
- Both software and non-software groups averaged 1 for identifying rules of thumb, probably because the whole class ran out of time before they had the opportunity to create rules of thumb.

Table 2. Performance Assessment Results for Part 2 – Application and Assessment

Coding Characteristic (**bold** denotes significant difference, p<0.05)	Software group	Standard Dev. (software group)	Non-Software group	Standard Dev. (non-software group)
Identifies issues or problems not explicitly stated in the case	2.88	0.25	2.00	1.02
Able to identify relevant aspects that can be applied to the challenge	2.5	1.08	1.666	1.03
Suggests incorporating a solution found in the case	2.5	0.58	1.916	0.92
Notices that a management method used by the experts cannot be applied as is but must be adapted	1.625	0.58	2.083	0.88
Notices that a solution used by the experts cannot be applied as is but must be adapted	2.375	0.95	2.333	0.68
Justifies use, modification, or abandonment of an expert solution based on criteria and constraints of group's challenge	2.75	0.25	2.25	0.76
Applies a solution used by the experts directly to their challenge	1.75	1.03	1.333	0.82
Suggests that an expert solution should be abandoned	1.25	0.50	1.25	0.61
Applies the case to the challenge using rules of thumb	1.00	0.00	1.00	0.00

For the video-recorded data for Part 2, reliability was 86%, and results show better performance by software groups on six dimensions but without statistical significance. Software groups tended to identify issues or problems not explicitly stated in the case that were important to their group's challenge, e.g., issues surrounding how people would move around on the island (roads, sidewalks, etc.) and how building equipment would be transported to the island. They tended to carry over the relevant aspects of the case that they identified in Part 1 to their plan, while non-software groups tended to neglect the risks they identified in Part 1, instead designing a plan from scratch. As a result of this, software groups were more likely to suggest that a solution from the case would be good to incorporate into their challenge solution. Software groups tended to justify the use, modification, or abandonment of an expert solution based on the criteria and constraints of the group's challenge. For example, one software group member suggested that the group build a sea wall out of an expensive material, but another group member suggested they consider a different material given the amount of money they had to work with. Both software and non-software groups averaged only 1 for applying rules of thumb. However, despite that, software groups were able to apply expert solutions to their challenge and justify those applications using the criteria and constraints of their challenge, suggesting that while they did not articulate the lessons they learned from the case, they had identified important lessons in the case.

Interestingly, non-software groups performed better on one dimension—noticing that management methods used by the experts cannot be applied as is but must be adapted. This is probably because most software groups tended to apply management methods directly, and to modify expert solutions more so than management methods. For example, most software groups decided to build houses in the middle of the island in low erosion zones rather than building on the coast like the experts did. So instead of worrying about how to manage erosion along the shore-line, they focused on taking advantage of the fact that there were sections of the island where little erosion occurred, and they decided to focus more on designing transportation systems to get residents to the beach or figuring out how to keep the shoreline from eroding the docks that they wanted to build for the residents' boats. Notice that in addition to lack of significant differences in software and non-software groups for application, performance overall is much lower in the second part of the performance assessment (application). We have two explanations for this. One explanation for this trend may be that interpretation involves a set of skills that the students have already developed, while application is less familiar. Most eighth graders who read at an eighth grade or higher level are asked to identify the important aspects of prose and talk about what they have learned in other classes (e.g., history), but rarely are they asked to apply what they've learned to create a solution. The other is that the teacher did not model application skills for students, so they were not able to observe the process of application as cognitive apprenticeship suggests is best for skills learning. Analysis of written work shows that individuals who participated in software groups tended to do several things better. First, they could identify more expert problems that were the result of experts implementing a solution. For example, students in one software group made the connection that the lack of new sand available to replace the old sand on the ocean floor was a direct result of the experts dredging the channel to allow larger boats to pass through. Both software and non-software groups identified problems that occurred in nature that the experts had no control over, like the natural movement of sand down the shoreline, but software groups also identified those problems that resulted from expert error. Second, students from software groups tended to identify pros and cons for each management method they identified. In fact, software groups tended to identify multiple pros and cons for each management method they identified. Non-software groups tended to identify either a pro, a con, or neither, but rarely both. Third, students from software groups tended to provide more correct justifications of risks. Non-software groups tended to either state risks with no justification or include incorrect justifications. For example, software groups' risks and justifications were "Some sand may not be stable enough to hold houses—people could get killed if the house shifts" or "The island will erode when you dredge the river—It will make the houses wash away." Examples of non-software groups' risks and justifications were "Excessive bugs—West Nile Virus" or "Flooding from hurricanes—Can only travel by boat."

Articulation of Reasoning Processes – Analysis of Interviews

Our individual interviews with students who used the software revealed that they thought it helped them organize their thoughts and keep track of where they were in the bigger task better than reading the case, pulling out important facts, and writing them up on their own (they had done that during a previous unit). Students told us that they used the hints a great deal, and they thought the hints generally provided helpful explanations of what they were being asked to articulate, though some hints were clearer than others (thus, uncovering a deficiency of the software). In addition, students thought the prompts helped them identify the important aspects of the case quickly and helped them formulate more thoughtful answers. Most interestingly, almost every student identified a process for case application that matched what was in the software: identify the main issues/problems in the case; identify solutions in the case; identify criteria and constraints in the problem to be solved; identify criteria and constraints in the case; match problems between problem to be solved and case; match criteria and constraints between problem to be solved and case; decide if the solution from the case can be used directly or if some parts need to be changed in order to use it. Students were able to identify this process when asked how they would use a case or example given to them in another class. Interestingly, though students identified this process for using cases to solve problems, when shown the artifacts they created from the Case Interpretation, Case Application and Solution Assessment Tools, students could only articulate what the Case Interpretation tool was designed to help them do (to help them understand the important points of the case and figure out the lessons they could learn). Only 1 of the 14 students was able to correctly identify the purpose of the Case Application Tool (to help them figure out whether a rule of thumb made sense to apply to their challenge and then to apply it). One explanation for this may be that the structuring and linking of the tools made the three step high-level process feel like one fluid process to the students, which could explain why they were able to explain the entire process, but could not segment the process into its three high-level steps.

Discussion

Students who were exposed to both teacher modeling and just in time scaffolding in the Case Application Suite performed better at interpreting and applying expert cases than students who did not. What is most interesting to us is the difference between results for interpretation and for application. Our results show that software vs. no software is not the issue. Instead, having exposure to modeling, coaching, detailed scaffolding, and multiple opportunities to carry out skills seems most important. Kids who were exposed to CAS's detailed scaffolding did tend to do better on application and they were able to articulate the processes involved in using a case or example to aid in problem solving. While these results are limited to case application, they make two suggestions about supporting students as they are learning other complex cognitive skills:

(1) A cognitive apprenticeship that includes just in time scaffolding during small group work to supplement teacher modeling and coaching seems more effective than teacher modeling and coaching without the extra scaffolding. Having the software available when working in small groups without the teacher seems to help students engage in more focused and detailed reasoning. If teacher and software share their approaches to carrying out the targeted skills, then software can be used fairly naturally as an agent with expertise working with small groups, allowing the teacher to successfully share scaffolding responsibilities as students learn complex cognitive skills.

(2) While we don't have conclusive evidence yet, it seems that the kind of help provided by CAS promotes internalizing the processes involved in carrying out complex skills. While learners couldn't tell us exactly what each of CAS's tools helps with, they had a good idea of the steps involved in the overall process of applying cases to new situations. We need to find out which parts of the scaffolding system are responsible for this.

During Fall, 2003, we are looking more in depth at the way students are acquiring and developing case application skills over time. We are seeking to describe the range of developmental trajectories in learning to interpret and apply expert cases. We also hope to understand more about how well our system of scaffolds supports students with learning other complex cognitive tasks.

References

Anderson, J.R., Greeno, J.G., Kline, P.K. & Neves, D.M. (1981). Acquisition of problem solving skill. In J.R. Anderson (Ed.) *Cognitive skills and their acquisition*. Hillsdale, NJ: Erlbaum, pp. 191-230.

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, Vol. 26*(Nos. 3 & 4), pp. 369-398.

Bransford, J.D., Brown, A.L. & Cocking, R.R. (1999). *How People Learn*. Washington, D.C. National Academy Press, pp. 41-66.

Bransford, J.D. & Stein, B.S. (1993). *The IDEAL problem solver* (2nd ed.). New York: Freeman, pp. 19-50.

Collins, A., Brown, J.S., & Newman, S.E. (1989). Cognitive Apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L.B. Resnick (Ed.) *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*. Hillsdale, NJ: Erlbaum, pp. 453-494.

Kolodner, J.L. (1997). Educational Implications of Analogy: A View from Case-Based Reasoning. In *American Psychologist*, Vol. 52, No. 1, pp. 57-66.

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

Kolodner, J.L., Crismond, D., Fasse, B. B., Gray, J., Holbrook, J., Puntembakar, S. (2003). Putting a Student-Centered Learning By Design™ Curriculum into Practice: Lessons Learned. *Journal of the Learning Sciences*, Vol. 12 No. 4.

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*. Atlanta, GA, pp. 16-22.

Kolodner, J.L. and Nagel, K. (1999). The Design Discussion Area: A Collaboration Learning Tool in Support of Learning from Problem-Solving and Design Activities. *Proceedings of CSCL '99*. Palo Alto, CA, 300-307.

Owensby, J.N. & Kolodner, J.L. (2002). Case Application Suite: Promoting Collaborative Case Application in Learning By Design™ Classrooms. *In Proceeding of the International Conference on Computer Supported Collaborative Learning*, CSCL-2002, Jan. 2002, pp. 505-506.