An environment for supporting independent, individualized learning and problem solving

Susan M. Williams, Ray Bariess, & Brian J. Reiser
School of Education and Social Policy
Institute for the Learning Sciences
2115 N. Campus Drive, Evanston, IL 60208
E-Mail: williams@ils.nwu.edu, bareiss@ils.nwu.edu, reiser@ils.nwu.edu

Abstract: We have been investigating ways of supporting middle-school students who are learning mathematics by constructing unique solutions to complex problems. This work addresses three needs in supporting independent, individualized learning and problem solving: realistic, complex contexts for learning, structured problem-solving activities that make complex problems more manageable for novices, and readily-available guidance for students during problem solving.

Learning is problem-based. Students are asked to design and layout a playground in order to learn to measure, draw to scale, and compute perimeter, area, and volume. The curriculum encourages an iterative approach to problem solving in which students create designs, give each other feedback, and revise their work. To support their problem solving, a software environment provides a simple model of the design task that aids students in identifying sub-goals, brainstorming questions that provide general procedural hints, and two structured multimedia databases: A student database is used to record students' designs, annotated with procedures used, new concepts learned, design rationale, and evaluations. A performance support database provides an archive of previously created designs with associated conceptual and procedural information.

Creating a learning environment in which students are actively engaged in constructing knowledge and pursuing their own learning goals involves more than adopting activities in which the students can participate. It also has far-reaching implications for the roles of teachers and students who must transform the autocracy of the traditional classroom into a partnership in which all participants share responsibility for teaching and learning.

Lampert (1995) has described this situation as similar to that of a tour director who is accustomed to taking tourists on a guided tour at each stop along the route. Imagine how difficult it would be for this director if each of the tourists is given a personal motor scooter and an opportunity to explore. There would be maps, tickets, and other materials to prepare and distribute before the tourists could begin their individual explorations. If problems or questions arose during the tour, the director would need to provide assistance to a widely separated group. The tourists themselves would take on increased responsibilities as they learned to read maps and recognize signs (perhaps in a foreign language), and became responsible for being back at the bus in time to continue their journey.

Lampert's analogy vividly communicates the dilemma of a teacher who attempts to transform his or her classroom: How can I anticipate which materials students will need? How will I know if the students have done everything they are supposed to do? How can I help one student without losing track of what the others are doing? Students who are accustomed to getting step-by-step instructions have similar questions: What is the problem we are being asked to solve? What are we supposed to do? Where can we get help?

We have been investigating ways to provide answers to these questions for middle-school students who are learning mathematics by constructing their own unique solutions to complex problems. This work addresses three needs in supporting independent, individualized learning and problem solving. First, creating appropriate, realistically complex contexts for learning helps students understand "What is the problem we are being asked to solve?" Second, structuring problem-solving activities makes them manageable for novices who ask "What are we supposed to do?" Finally, providing support for students during problem solving answers "Where can we get help?" This paper characterizes our approach to meeting each of these needs, and describes a curriculum and computer-based learning environment that synthesizes these components.

Creating realistic, complex contexts for learning

In problem-based curricula such as the one described herein, many of the initial questions asked by students are motivated by their attempts to understand the problem. Therefore, we emphasize the importance of selecting appropriate problems and presenting them in a way that facilitates comprehension. Our work is based, in part, on anchored instruction (Bransford, Sherwood, Hasselbring, Kinzer, & Williams, 1990), an approach known for its use of realistic complex contexts for learning. These contexts, called anchors, afford opportunities for students to learn a variety of concepts and procedures, how they are related, and the nature of the situations in which they can be useful.

The initial understanding of complex anchor problems are made simpler by the way they are presented: First, problems are presented in a video format because students can more easily form mental models of problem situations when they are presented in dynamic images rather than text. Second, problems are presented as stories, a format that is easily remembered. Third, most data required for problem solving are embedded in the stories. Students do not have to acquire this information from secondary sources. Fourth, the stories accurately represent real world situations, so that students can use prior knowledge during problem solving. Each story ends when the main character encounters a problem, and students are challenged to solve the problem using the information provided in the video. Learning begins with the goal of thoroughly understanding the situation described in the anchor and finding solutions to the problem that it presents.

In addition to supporting initial understanding by the way problem situations are presented, learning is also facilitated by certain kinds of curricular activities. Anchored instruction emphasizes taking multiple perspectives on a problem situation. The importance of this approach is supported by research indicating that, when learning by solving problems, students' initial understanding of the anchor problem is very inflexible, i.e., they understand a single solution to a single problem and have difficulty adapting this solution to solve very similar problems (Williams, Bransford, Vye, Goldman, & Carlson, 1992).

Besides adding curricular activities that encourage multiple perspectives, the problems themselves can encourage such activities. One type of problem that provides opportunities for taking multiple perspectives is a design problem, e.g., students might be asked to plan a trip or design a building that meets certain requirements and constraints. In contrast to typical school problems, the frequently ill-structured nature of design problems requires students to create their own interpretations of the problem. Each student can interpret the problem in a different way, and each definition is likely to suggest a different solution. Answers can be evaluated using many criteria, and many students can have correct answers without having the same answer. Students have the opportunity to compare their solutions with those of others in their class and to determine which aspects are required. In addition to comparing their answers to those of others, students must often create multiple designs themselves in order to optimize their solution.

We have selected the anchor, *Blueprint for Success*, from the problem solving series, *The Adventures of Jasper Woodbury*, (CTGV, 1993), as a context for our research. *Blueprint* is a design problem that introduces important geometry concepts in the context of a "career day" trip to an architectural firm. In the video, the main characters meet a businessman who is donating a lot to the city to be used as a playground. The challenge for these characters (and for the students) is to design this playground.

The *Blueprint* video provides much information about designing the playground, e.g., it requires a slide, swings, sandbox, and one other piece of student-chosen equipment. Additional scenes provide information about donated materials and safety factors, as well as how to create and communicate a design. Specifically, students are challenged to create scale drawings of each piece of equipment and a layout for the playground. They must ensure that their designs do not violate any of the requirements and constraints presented in the video. During the design process, students have opportunities to learn important skills and concepts, e.g., measurement, proportional reasoning, and creating and reading scale drawings. Solving this problem allows students to create designs that reflect their personal knowledge of playgrounds and affords an opportunity to be creative. At the same time that students have flexibility and personal freedom in problem solving, they learn that they must also use precise mathematics to create their designs and to communicate them to others.

Structuring learning and problem solving

If students are to complete a complex design task successfully, they need support during problem solving.

Determining which kinds of assistance are effective, and when and how they should be provided, is especially difficult when learning is organized around problem solving. Depending on their understanding of the problem, and the knowledge and skills that they bring to the process, students will have many different approaches and many different answers, even for a very simple problem. As problems increase in complexity, it quickly becomes impossible to anticipate students' errors and to supply them with feedback tailored to their specific strategies and solutions.

ASK Jasper is a computer-based, multimedia environment, designed to structure and support students' problem solving as they learn geometry while solving a complex design problem and communicating their design to others. In the ASK Jasper curriculum, students begin by viewing the *Blueprint* video, then use ASK Jasper to record their designs, while their teacher serves as a coach. The computer environment comprises a simple task model, a set of brainstorming questions that provide general procedural hints, and two structured multimedia databases. A student database is used to record students' designs, as well as their explanations of procedures used, new concepts learned, design rationale, and evaluations. A performance support database provides an archive of previously created designs with associated conceptual and procedural information.

Task model

Task modeling is a generally useful approach for studying learning and problem solving. A task analysis creates a model of problem solving by dividing a complex task into smaller sub-tasks. Such a model is an effective way to provide guidance when each learner is using a different approach because it focuses on breaking down the problem into necessary sub-goals rather than attempting to track or dictate the performance of the student. In the ASK Jasper environment, a hierarchy of sub-goals structures students' problem solving by helping them keep track of where they are in the problem space and provides cues when they don't know what to do next. It is up to the students to track their progress, i.e., which sub-goals they have achieved and what remains to be done.

For *Blueprint*, or for that matter any design problem, the top level of the task model includes the following steps: Understand the problem, design one or more components, combine the components, execute the design, and reflect on the success of the design and what has been learned. Each of these steps is broken down further, and problem solvers may cycle through these steps multiple times as understanding of the problem grows, and they revise their designs. The *Blueprint* model is designed to provide structure without being overly prescriptive. Students can start at many different points in the model. They must operationalize each of the sub-tasks before they can carry it out. They are free to do so in any way they choose and to skip steps that they feel are unnecessary.

Brainstorming questions

Sometimes, the general task model does not provide sufficient guidance for a student. Therefore, brainstorming questions are associated with each sub-task to provide hints for how to solve the *Blueprint* problem. Like the task model, these questions are not specifically related to the *Blueprint* problem and, with minor modifications, could be used with any design task. In general, the goals of the brainstorming questions are to reduce the complexity of the problem and to help students develop good strategies for problem solving.

The brainstorming questions do this in several ways. First, they suggest ways to break the tasks down into still smaller sub-tasks. For example, students are encouraged to summarize the problem according to a standard story schema: Who are the main characters, what problem did they encounter, *etc*. Second, the questions provide students with heuristic methods used by professional designers (e.g., Schon, 1983): What is the most important requirement for your design, what would a design look like that just satisfied that requirement (and ignored the rest), does this design satisfy all of the requirements, can the design be modified to satisfy the rest of the requirements, *etc*. Third, brainstorming questions help students by providing suggestions for evaluating their designs: Does your design satisfy the safety constraints, is there enough fence to go around your playground, *etc*. Finally, the brainstorming questions encourage reflection: How did you create your design, why did you do it this way, what math did you use, *etc*.

The student database

Creating and organizing design documents. As students go about the task of designing their playgrounds and communicating their designs to others, they generate documents of many kinds: drawings of playground components, evaluations by teachers and peers, explanations of procedures that were used, etc. The purpose of ASK Jasper's student database (Fig. 1) is to organize these materials in a way that corresponds to the task model, thus reinforcing the task model as an approach to problem solving (cf. ASK systems, Bareiss and Osgood, 1993). Students' entries in the database consist of text, graphic, and video materials for each of the sub-tasks in the model. Students can use a host of inexpensive and readily available computer-based graphics, video, and word processing tools to create their notes and designs, or they can create them on paper and input them via a scanner. The opportunity to "publish" their work in the performance support database, which is discussed in the next section, provides an authentic audience for their work.

Annotating designs. Students make entries into the database as they enter their designs, following the steps of the task model; however, the primary learning goal is not to create a solution to the *Blueprint* problem. Instead, it is for students to acquire the skills and concepts necessary to solve complex mathematical problems they will encounter in the future. The ASK Jasper database supports this process by encouraging students to elaborate their designs, not in an ad hoc manner, but according to a standard set of activities.

The first of these activities is reflection which helps students refine their understanding. Annotation categories called *Why* and *How To* encourage students to explain the rationale for their designs and the procedures they used to create them. *Learning Issues* allows student designers, their classmates, and teacher to add suggestions for what the designers might need to learn to improve their designs (Fig. 2). The *Learning Issues* category is especially important for focusing students on the knowledge-building purpose of their problem solving.

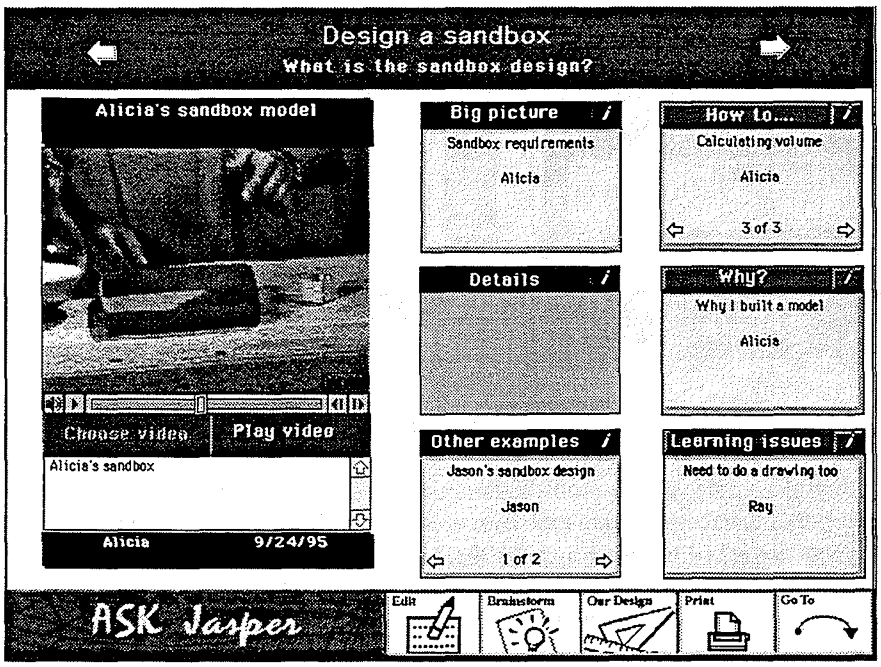

Fig. 1. Categories in the database organize student-created materials

The second activity supports generalization from the *Blueprint* problem by having students connect their designs with other information. Two categories, *Big Picture* and *Details*, help students to structure their database entries, e.g., a list of requirements and constraints might be linked to *Details* notes in the form of video clips of scenes from which the requirements were extracted, or a text story summary might have a *Big Picture* link to the *Blueprint* video. Another category, *Other Examples*, allows students to link their designs, procedures, and explanations to similar entries in the database. These related examples could be those of other students, alternate designs created by the same student, or revisions to correct mistakes in an earlier design. Understanding similarities and differences between examples is at the core of the generalization process. The ability to link notes within the database encourages students to make these connections.

Providing performance support

The task model, the brainstorming questions, and the student database provide general tools and suggest general techniques for solving design problems. However, they may not be specific enough to help students who are having difficulty. One of the benefits of anchoring students' learning in a specific problem is that it becomes possible to provide specialized help. This type of help is provided by a performance support database that provides specific examples of designs and other work products that are solutions to sub-tasks of the design problem. When designing ASK Jasper, we debated the wisdom of providing such examples, rather than providing examples by analogy to other design tasks. We decided to provide direct examples because we felt that less successful students, especially, would have difficulty in transferring the associated procedures and issues across domains.

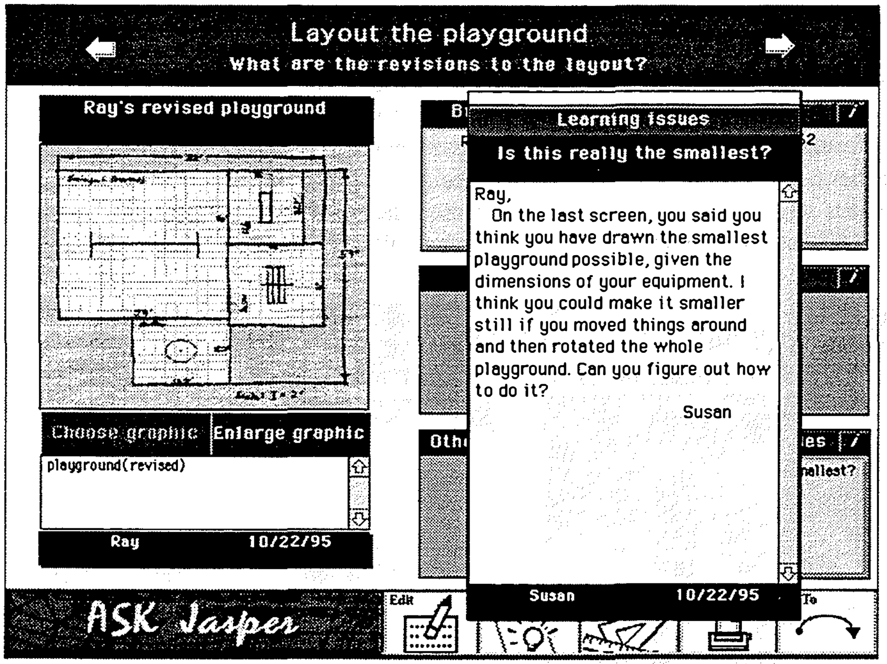

Fig. 2. *Learning Issues* notes encourage reflection and collaboration

The use of direct examples can be problematic, however, because students can simply copy the designs and other information rather than constructing their own solutions to the problem. This practice is discouraged in two ways. First, students are asked to create a rough draft of their own design before consulting the database. Second, the database is populated with examples that are substantially correct, but fail in one or more ways to meet the requirements and constraints of this problem. Students must recognize and modify these discrepancies if they are to reuse the design successfully. For example, one swingset design has dimensions that do not correspond in the three view drawings and also violates a primary safety constraint. A student is assisted in identifying these flaws by general comments associated with the design in the *Learning Issues* category of the database. Eventually, the performance support database will contain many example designs (linked via the *Other Examples* category) that span the range of naive conceptions of the design problem. Studying multiple designs may assist students in recognizing and avoiding problems in their own designs.

In addition to example designs, the performance support database contains examples of other documents the students are asked to produce, such as story summaries and lists of design constraints. Specific procedural guidance is associated with each of these documents via the *How To* category. For example, one procedural note describes how to make a ruler with graph paper to facilitate measuring to scale. Design rationale is provided via the *Why* category. For example, a note associated with a swingset design explains how it was modeled after actual swingsets that a student designer studied in his own playground. Problems with designs and other learning issues, raised by the designer or by other evaluators, appear in *Learning Issues*. As in the student database, *Big Picture* and *Details* help to relate a component of the solution to the overall context of the design problem. These categories also contain information from *Blueprint* to provide a context-sensitive reference source in which students can easily locate embedded data relevant to an aspect of their task.

The performance support database provides models, at several levels, to assist the student designer. Each individual note in the database is a model of a type of note that the student may want to include to create and communicate his or her design. Taken as a whole, a sequence of notes by a single designer constitutes a design episode that provides an overall model of the design process. However, it provides only one solution to the design problem. To encourage the student to consider multiple perspectives, a number of such design episodes are included in the database, which differ in approach, results, and as noted above, in misconceptions.

The addition of student designs to the performance support database is a publication process rather than an incremental and conversational process, as is typical in computer-supported collaborative learning environments (e.g., CSILE (Scardamalia, & Bereiter, 1992), CoVIS (Edelson, O'Neill, Gomez, & D'Amico, 1995), CaMILE (Guzdial, Rappin, & Carlson, 1995)). A student creates a design in the student database, subjects it to peer and to teacher evaluation, and revises it. When the design is satisfactory, it is added to the performance support database, and the student links it to other content, e.g., to design examples which he or she found useful during the design process.

Initial Results

A pilot test of the ASK Jasper curriculum and software was conducted with eight 6th-grade, public-school students of mixed ability in a pull-out program for a two week period. This preliminary evaluation focused on the usability and affordances of the problem and software. As expected, *Blueprint* proved to be an excellent problem for students of this age. Students found it interesting and had sufficient background knowledge to generate initial design ideas; however, their lack of knowledge of geometry and proportional reasoning caused them to make many errors in developing and communicating their ideas. Initially, they were reluctant to use the performance database for help with these problems either because they believed their designs were correct or because they didn't know what kind of help this resource could provide. After some initial encouragement, students began to consult the database regularly. The task model provided an effective organization and students appeared to have no difficulty finding the information they were seeking. Students were disappointed to find that the examples provided in the database needed modification to meet the requirements and constraints of the current problem. This attitude was more prevalent for less successful students. Students scripted and videotaped demonstrations of the target skills to use as annotations for their designs. They were very eager to have these demonstrations included in the database even though writing accurate scripts explaining the procedures required significant work beyond that required to create an acceptable design. These reflective activities appeared to be very effective in deepening students understanding.

After this pilot test, a three week classroom evaluation of the curriculum and software was conducted in two 5th grade math classes in suburban Nashville, Tennessee. Students in the two classes were grouped by their scores on the Tennessee Comprehensive Achievement Program. The mean percentile score for the high-achieving class (N=29) was 81. The less-successful class (N=26) had a mean of 41. Twenty-five percent of the students (all of them male and most of them in the less successful class) had been diagnosed with attention deficit hyperactivity disorder (ADHD) or recommended for testing for ADHD.

Two adjustments were made to the curriculum for this study. First, because of limited classroom time and computer resources, the curriculum was abridged: Students identified the goals, requirements, and constraints of the problem; designed and revised a swingset; and designed and revised a slide. Second, because our experimental design did not include an explicit control group, we required students to create each of the initial designs without access to the performance support database. This design allowed the initial swingset design to serve as a pretest and the second to serve as a posttest. The first slide design represented a near transfer task for the swingset design.

Analysis of the data is not complete at this time. Scoring of the initial and revised swingset designs reveals the following: Students' visual representations became less distorted between the initial design and the revision, i.e., virtually all students adopted the standard orthographic projection employed in examples in the performance support database. Their revised drawings included more details. (Interestingly, this increase was greatest for the ADHD students.) They labeled more dimensions in their drawings, and the labeling was more consistent with their scale drawings. The students became more cognizant of requirements and constraints on their designs and generally adhered to them in the revised designs.

Not all of the results of the intervention were positive. The variability of students' designs diminished between their first and second swingset drawings, possibly as a result of studying examples in the performance support database. Many students' initial designs were quite unique, but almost all of their revised designs were standard A-frame designs, similar to the examples in the performance support database. As the variety of examples in the database increases, we believe that this effect will decrease.

While our results are preliminary, our experience encourages us to believe that Blueprint for Success, our curriculum, and ASK Jasper, taken together, provide an authentic, rich, and motivating context for learning.

References

Bareiss, R. and Osgood, R. (1993). Applying AI models to the design of exploratory hypermedia systems, Hypertext '93 Proceedings, 94-105.

Bransford, J.D., Sherwood, R.D., Hasselbring, T.S., Kinzer, C.K., & Williams, S.M. (1990). Anchored instruction: Why we need it and how technology can help. In D. Nix & R. Spiro (Eds.), Cognition, education and multimedia: Explorations in high technology (pp. 115-142). Hillsdale, NJ: Lawrence Erlbaum & Associates, Inc.

Cognition and Technology Group at Vanderbilt (CTGV). (1993). The Adventures of Jasper Woodbury [Video]. Warren, NJ: Optical Data.

Edelson, D. C., O'Neill, D. K., Gomez, L. M., & D'Amico, L. (1995). A Design for Effective Support of Inquiry and Collaboration. In J.L. Schnase & E. L. Cunnius (Eds.) Proceedings of CSCL '95: The First International Conference on Computer Support for Collaborative Learning, October 17-20, 1995, Bloomington, Indiana. Hillsdale, NJ: Erlbaum.

Guzdial, M., Rappin, N., & Carlson, D. (1995). Collaborative and multimedia learning environment for engineering education. ACM Symposium on Applied Computing '95

Lampert, M. (1995). Managing the tensions in connecting students' inquiry with learning mathematics in school. In D.N. Perkins, J.L. Schwartz, M.M. West, & M.S. Wiske (Eds.), Software Goes to School. New York, Oxford University.

Scardamalia, M., & Bereiter, C. (1992). An Architecture for Collaborative Knowledge-Building. In E. Decorte, M. Linn, and H. Mandl (Ed.), Computer-based learning environments and problem solving, Springer-Verlag, Berlin.

Schon, D.A. (1983). The reflective practitioner : how professionals think in action. New York : Basic Books.

Williams, S.M., Bransford, J.D., Vye, N.J., Goldman, S.R., & Carlson, K. (1992). Positive and negative effects of specific knowledge on mathematical problem solving. Paper presented at the annual meeting of the American Educational Research Association, San Francisco.

Acknowledgements

This work was supported by a grant from the James S. McDonnell Foundation to the first author.