Information Technology in Childhood Education Annual (2002), 277-294

Design and Implementation of Mapping Software: Developing Technology and Geography Skills in Two Different Learning Communities

ROBERT S. FRIEDMAN
College of Computing Sciences
New Jersey Institute of Technology
Newark, NJ 07012, USA
friedman@njit.edu

JERRI DRAKES
Little Bytes, Co.
Newark, NJ 07102, USA

FADI P. DEEK
College of Computing Sciences
New Jersey Institute of Technology
Newark, NJ 07102, USA
Fadi.Deek@njit.edu

A software development collaboration project designed to maximize the skill sets and interests of school children and teachers, educational software technologists and researchers, and college undergraduates is presented. The work brings elementary school children with college seniors and technology consultants to implement a problem-solving methodology within a collaborative environment to design, develop, and implement a multimedia software application that enhances time and space orientation abilities of children and puts the programming, interface design, and multimedia systems capabilities of college students into action. This project-based learning offers students the opportunity to learn mapping skills, problem-solving techniques, and participatory design while planning and conducting virtual tours of their city.

Familiarity with geography, the science of space and place on the Earth's surface, helps the visualization and understanding of one's home and orients people and their relationships in relation to other cultures and environments. Learning about geography is the first step in understanding one's community and one's relationship to the world at large. As a community grows and changes, children must learn and master the ability to find their own way. Children become empowered when they are comfortable with their surroundings and prepared to explore the world beyond their quotidian boundaries. Maps help to show children where they are, where they've been, and where they can go, while creating a sense of belonging to a community, a history, and a path to the future. Without a secure sense of spatial orientation, one is directionless, and unconnected. The urban environment, in particular, is populated with many young students who cannot perceive beyond the boundaries of home.

In the early 1930s, Lucy Sprague Mitchell, the founder of the Board of Experimental Education, later known as Bank Street College, began the campaign to reveal the importance of teaching geography. She pioneered the concept of *human geography*, an approach to teaching geography that uses real world experiences as the foundation for the curriculum. This approach encouraged children to learn about the world not simply by immersion in facts, but by applying those facts during real world discovery (Mitchell, 1991). When is the right time to teach geography and how should it be taught? Is there a role for learning technologies in geography instruction, particularly when there is an added goal of local knowledge to general mapping skills and spatial orientation? While technology cannot be considered a panacea for educational reform (Kimmel & Deek, 1995), when properly used it can effectively improve and enhance instruction and learning experiences. This article describes a software development collaboration project designed to maximize the skill sets and interests of elementary school children and teachers, educational software technologists and researchers, and college undergraduates. Through the implementation of project-based learning, problem-solving methodologies, participatory design, and community involvement, software containing interactive video, calculation programs and spatial orientation tools offers 4th- through 6th-grade students in Newark, NJ, the opportunity to learn mapping skills while planning and conducting virtual tours of their city.

COGNITIVE DEVELOPMENT AND SPATIAL ORIENTATION

Learning theorists have articulated unique developmental predispositions for different kinds of learning. David Sobel (1998) stated that between ages five and seven, children start to move away from home and parents and explore the natural world. From ages seven to eleven, children are predisposed to merging with nature and making geographic sense of the world around them. They are ready to step out the box, their world measures beyond a mile or more, taking them out of their neighborhood and into the town and broader community.

With the passage of Goals 2000: Educate America Act in 1994, geography was officially recognized as a core curriculum subject in American schools. Also in 1994, the United States Department of Education released standards for teaching geography to K-12 students. In New Jersey, the Core Curriculum Content Standards descriptive statement on geography states:

> Thinking in spatial terms is essential to knowing and applying geography. It enables students to take an active questioning approach to the world around them and to ask what, where, when and why questions about people, places and environments and to formulate answers to critical questions about past, present, and future patterns of spatial organization and to anticipate the results of events in different locations. Thinking spatially, students learn to devise their own mental maps, which relationships and students' perceptions and attitudes about the area. Thinking spatially enables students to predict what might happen given specific conditions. Spatial concepts and generalizations are powerful tools for explaining the world at all levels, from local to global. They are the foundation for geographical understanding (1999).

Although computers cannot replace the human contact and feedback that only a teacher can provide, they are tools that can be used to significantly enhance students' educational experiences. Their expected impact in education is so meaningful that in a speech for a House of Representatives Panel on Technology and Education, MIT's Seymour Papert stated, "The presence of technology in society is a major factor in changing the entire learning environment" (Papert, 1997). Developing effective mathematical and spatial orientation software that is an integrated part of a curriculum based in reasoning presupposes an understanding of the role that computers are playing in today's classroom experience. Technology plays a major function in providing stimulating learning environments. In 1995, James Kulik and his colleagues at the University of Michigan conducted research on the use of computer based instruction software ("Meta Analysis Study"). Their basic finding was that using computer-assisted instruction software results in a substantial improvement in learning outcomes and speed, perhaps 20% or more on average. Such instruction works best, of course, in content areas where the computer can tell the difference between a student's right answer and wrong answer, for example, in mathematics or grammar exercises. Few other teaching methods have demonstrated such consistently strong results as this type of self-paced instruction.

Reeve's found that the use of cognitive tools such as interactive software applications engages students in "knowledge constructions rather than knowledge reproduction" (1998). He summarized the following principles as the foundation for using cognitive tools:

- Cognitive tools will have their greatest effectiveness when they are applied within constructivist learning environments.
- Cognitive tools empower learners to design their own representations of knowledge rather than absorbing representations preconceived by others.
- Cognitive tools can be used to support the deep reflective thinking that is necessary for meaningful learning.
- Cognitive tools have two kinds of important cognitive effects, those which are *with* the technology in terms of intellectual partnerships and those that are *of* the technology in terms of the cognitive residue that remains after the tools are used.
- Cognitive tools enable mindful, challenging learning rather than the effortless learning promised but rarely realized by other instructional innovations.
- The source of the tasks or problems to which cognitive tools are applied should be learners, guided by teachers and other resources in the learning environment.
- Ideally, tasks or problems for the application of cognitive tools will be situated in realistic contexts with results that are personally meaningful for learners.

Reeves cited Lehrer (1993) as finding that, "Cognitive tools empower learners to design their own representations of knowledge rather than absorbing knowledge representations preconceived by others." Through the process of participatory design (Druin, 1999), 10-year old students will have an equal, if not greater stake in the development, testing, refinement and use of the software as compared to the college students who are interacting with the youngsters at each stage and step of the software development process. For Reeves:

> The process requires learners to transform information into dimensional representations, determine what is important and what is not, segment information into nodes, link the information segments by semantic relationships, and decide how to represent ideas. This is a highly motivating process because authorship results in ownership of the ideas in the multimedia presentation.

Carver, Lehrer, Connell, and Ericksen (1992) determined that multimedia designers develop a variety of skills that extend beyond the use of specific software applications and programming knowledge. These include:

Project Management Skills

- Creating a timeline for the completion of the project.
- Allocating resources and time to different parts of the project.
- Assigning roles to team members.

Research Skills

- Determining the nature of the problem and how research should be organized.
- Posing thoughtful questions about structure, models, cases, values, and roles.
- Searching for information using text, electronic, and pictorial information sources.
- Developing new information with interviews, questionnaires and other survey methods.
- Analyzing and interpreting all the information collected to identify and interpret patterns.

Organization and Representation Skills

- Deciding how to segment and sequence information to make it understandable.
- Deciding how information will be represented (text, pictures, movies, audio, etc.).
- Deciding how the information will be organized (hierarchy, sequence) and how it will be linked.

Presentation Skills

- Mapping the design onto the presentation and implementing the ideas in multimedia.
- Attracting and maintaining the interests of the intended audiences.

Reflection Skills

- Evaluating the program and the process used to create it.
- Revising the design of the program using feedback. (quoted in Reeves, 1998).

TECHNOLOGY AND GEOGRAPHY SKILLS: BACKGROUND

Software companies have developed a wide range of software applications geared to geographic inquiry and map making. The current strategies used in mapping software programs permit students to be active mapmakers. *Neighborhood Map Machine*, *Trudy's Time and Place*, *Carmen Sandiego*, *Where are We*, *Map Makers Tool Kit*, *Geo Safari* – all introduce geography skills to students grades 4–8, geared to providing geographic discovery and knowledge. What these programs cannot offer, however, is any significant or concrete local context for the user.

Little Bytes, an educational technology consulting company based in Newark, NJ, piloted a technology-based workshop in 2000 designed to expose geography concepts to urban youth in grades 4-6. The eight-hour workshop, conducted at schools and youth centers, focused on the students' investigating and exploring the school neighborhood while learning geography facts. Students were prompted to create a map to provide newcomers with directions as to how to get to school. A classroom of 15 students, each viewing a self-paced Microsoft *PowerPoint* presentation with a link to *Neighborhood Map Machine*, performed classroom and computer related activities while gathering information about maps. For the first three sessions of eight, students became familiar with symbols and signs, used a compass and learned directions. Scale and distance, grid coordinates and the most popular "walk around the block" summed up the geography skills necessary to draw a neighborhood map on grid paper. Students recreated their drawings using *Neighborhood Map Machine*, printed their maps onto heat transfer paper and ironed them onto T-shirts bearing their school name. The highlights of the workshop included the creation 3D paper maps, an exercise conducted in the second session. This was a group activity, using poster board as landscape and card stock paper to create building structures to represent houses and various buildings as seen in their neighborhood. A walk around the block was a significant component in the program, a real world experience, allowing students to discover their own city and neighborhoods using the compass. The workshop was well received. It was not unusual to have students re-enroll each time it was offered at their youth center or school. These students have a strong desire to enhance their knowledge and perceptions of their community.

There were several lessons learned from the experience described. First, unless a student is fully engaged, the effectiveness of the learning process is limited. Some of the students had difficulty mapping the neighborhood after the "walk around the block" exercise. Although mapping is a fundamental concept, it is not readily grasped. The ability to illustrate a view from a specific perspective demonstrates a level of cognitive sophistication (Jonassen, Beissner, & Yacci, 1993; Stoyanov, 1997; Sobel, 1998). The progression of children's mapmaking skills exemplifies their cognitive development. Towards the end of what Piaget (1972) identified as the preoperational stage, children can draw simple maps; however, their perspective is rather pictorial, with their own home as the main object. As children mature to the concrete operational stage (Piaget, 1972), they move from pictorial maps of their homes to an elevated view of their communities. Today it is widely accepted that a child's intellectual ability is determined by a combination of heredity and environment. Children's intellectual development can be enhanced through environmental factors that provide stimulating learning materials and experiences.

Second, *Neighborhood Map Machine* is a software application with a great deal to offer, but it does not satisfactorily support the urban environment. Its symbols are suburban in their orientation, and many of its activities use rural landscapes as a backdrop to learn navigating skills. Third, children often do not have an opportunity to explore their neighborhoods. Urban lifestyles sometimes include dangers in neighborhoods and deplorable conditions of many buildings, causing parents to be reluctant to have their children explore on their own. Lacking funds for entertainment or travel confines the child and as a result they tend to stay at home.

If there is an innate congruence between mapping and mathematical skills of elementary school-age children, Piaget's work can foster instructional software design. In his essay "How Children Form Mathematical Concepts," Piaget stated that children develop many of their mathematical concepts independently and spontaneously, without having to be taught them explicitly. Indeed, he said, "when adults try to impose mathematical concepts on a child prematurely, his learning is merely verbal; true understanding of them comes only with his mental growth" (Piaget, 1953). Another significant element of Piaget's theory is that in the elementary years, math topics should be taught in as concrete and experiential manner as possible since children have not yet developed the ability to think abstractly. Therefore, actual objects should be used in class when illustrating a mathematical concept, and students should be encouraged to manipulate the objects extensively. Concepts should also be related to contexts with which children are already familiar (Slavin, 1991).

These learning theories and insights into technology's role in the classroom, coupled with a partnership among students and faculty at the New Jersey Institute of Technology (NJIT), Little Bytes, and St. Philips Academy, an independent primary school in Newark, serve as the foundation for a comprehensive multiyear program in multimedia learning systems that brings college seniors studying software engineering together with primary school students and teachers in an effort to provide community-based educational software designed to introduce the landmarks and cultural facilities of Newark to primary school children through the collaborative development of mapping skills instructional software. A major goal of the program is to build on the participatory design model of software design, articulated most clearly by Druin (1999), and a problem-solving methodology that has been successfully implemented at NJIT and in four Newark public high schools (Deek, 1997; Deek & Friedman, 2001). One of the goals of the collaboration is to test the hypothesis that integrating these two models will promote positive change in the academic climate of classrooms by incorporating teachers, students, and skilled college-level software engineers to create educational multimedia applications that accommodate the specific needs of the students, provide teachers with design-level access to appropriate instructional materials, educational technologies, and give youngsters hands-on experience in the design, development, testing, and use of computer software tools.

The Little Bytes workshop experience has opened the door to increased collaboration with community-based and civic organizations such as the Newark Museum, Newark Bears baseball team, New Jersey Performing Arts Center, and the Newark Police Department, all of whom contribute to the development of a mapping skills software application that is focused on Newark's neighborhoods and landmarks. NJIT seniors majoring in software engineering, multimedia information technology, information systems, and computer science are creating audio, video, animation, and interactive calculation tools so that children can create, on their desktop computers at school, virtual tours of their city's cultural and civic sights and attractions. The college students' educational experience is project-based, as it will move them out of the lecture hall and into the community, the design studio and the computer lab as they work with educational researchers, 4^{th}–6^{th}-grade students and educational technology consultants over the course of a semester to implement a software design that has been created collaboratively among St. Philip's Academy 4^{th}-grade school students, their teacher, and Little Bytes. Interaction with digitized seating charts, videos of routes between city landmarks, and interactive programs for creating virtual tours is the backdrop for learning geography skills while providing young students with knowledge and information about their city and what it has to offer. The application is intended to expand the user's boundaries, as they will be able to experience their neighborhood and their city in a broader sense. The software application consists of three major components. (a) A fun tutorial facilitated by an animated character, during which the user learns basic geography skills; (b) map making and exploration of Newark by way of video of routes between significant sites and virtual tours of those sites; and, (c) the use of geography skills to solve problems, such as designing tours of the city for visitors and building efficient routes between landmarks.

A fundamental element of this project is its methodology supporting a team-oriented approach to K-6 educational software design. It is based in a problem-solving methodology found effective in computing, mathematics, and composition courses at the precollege and college levels (Deek & Friedman, 2001), combined with an approach to software engineering known as participatory design (Druin, 1999). With this approach, children's interaction with technology expands beyond their end-user status and into the conceptual design, development, usability testing, and debugging phases as well. Moreover, rather than limit the connection between outside environments and the classroom to children's homes and families, local colleges, community groups, and cultural centers are brought together through the application of interactive video and the adoption of a multifaceted participatory design process for educational software. This approach supplies an alternative method of software design, development, and evaluation, one that, through input from these multiple perspectives, is more appropriate for the quickly changing and dynamic nature of educational and edutainment software, in that traditionally software evaluation is ad hoc to the software design and development process. Through participatory design, software engineering becomes more accessible even to primary school students through the use of visually oriented software applications, increasing the opportunities to team children, teachers, software engineering students, educational researchers, and software designers in the development of new applications, and making software evaluation, once the province of committees of teachers and administrators, a component to consider at each stage of development by all members of the design team.

Application development through participatory design has three main goals. First, to develop integrated learning environments that support visual and verbal literacy. Second, to encourage learners to construct their own paths to knowledge, and third, to develop methodologies that offer a better understanding of what children want and need when using technology. Druin has found that her test groups were able to find common ground, overcome communication problems, and generate helpful ideas. Having children as design partners permitted programmers to respond to and improve the parts of the software with which children had the most difficulty (1999, 2000). One goal of the involvement of teachers in the software development process is to link those factors impacting teacher practices and beliefs in instruction with software designers so the product instantiates the methodologies employed in the classroom. Sustained, ongoing, partnerships between software engineers and classroom teachers can be considered both professional development and inservice training for teachers, key components for reform in teaching and curriculum change. It has become accepted that long-term intensive professional development programs are necessary and that short inservice programs or workshops were not sufficient to produce sustained change (Guskey, 1986).

Druin (1999) recommended the incorporation of three techniques for participatory design: (a) contextual, which levels children and adults as they observe, take notes, and interact with each other; (b) participatory, which employs brainstorming activities with the goal of having all team members sketch out ideas; and (c) technology emersion, which is used to observe what children do with new, unfamiliar technology. This exercise in participatory design is a systematic attempt to bring about change in classroom practice of K-6 teachers, as it provides multifaceted teams of students and teachers immediate classroom experience as well as intensive professional development in the content and skills areas. The program provides a combined laboratory and classroom environment in which participants design, develop, test, and refine software tools intended to inculcate spatial awareness and mapping skills while also attaining positive results with students. Direct and successful work with children serves to enhance teacher efficacy. Simultaneously, teachers are able to reflect and receive feedback prior to returning to the classroom to implement the newly acquired skills and knowledge.

This model is consistent with other reports suggesting that a participatory design model is necessary for change in classroom practice and is a necessary prerequisite to achieve positive learning outcomes for students (Haney, Czerniak, & Lumpe, 1996). It is expected that changes observed in the students' mapping skills performance will include a greater understanding of spatial and directional concepts after using video-based, community-oriented software, inquiry-based learning approaches, and student-directed discussions.

PROBLEM SOLVING

In the traditional classroom, there is often more emphasis on solving a problem in a textbook rather than solving life-based problems. Such an approach to teaching reduces the likelihood of extrapolation and generalization outside of the classroom, and limits adequate development of the cognitive and metacognitive strategies needed by many students. Instructional emphasis should be given to mastery of concepts, relationships, and skills that are essential for the subsequent acquisition and functional generalization of math and spatial orientation skills (Woodward, 1991). Students should be guided in solving complex problems and should also be given sufficient opportunities to independently solve such problems. Through participatory design, there are extended interactive discussions among teachers, students, and software engineers. Once a development strategy is chosen, however, it becomes necessary to break down large tasks into more than one component and have the programmers develop primitive models to test based on the requirements of the component, the capabilities of the intended users and the pedagogical goal of the teacher. After the process is separated into smaller pieces or components, the software engineering students need to identify the programming languages, scripts, and tools that are needed to bring each component to fruition. Students working with graphical user interface design interact with the 4th-graders to determine engaging interactive elements and designs that are compatible with the software components as well as the skill sets of the intended users. The decomposition of problems to solve is the foundation of the SOLVEIT methodology (Deek, 1997).

Employing a consistent methodology that is common to all the participants and able to be documented can add to the sense of partnership among the participants as they share ways to solve problems specific to the areas that each encounter. Bringing theory into practice, as Filho (2001) suggested, demands that a process architecture be used when students are engaged in complex software engineering projects such as the mapping project described here. This architecture is a "conceptual framework for consistently incorporating, relating, and tailoring process elements into enactable processes, processes that include all the elements required for its performance by one or more agents" (p. 65.) SOLVEIT is such an architecture, and has been successfully employed in high school and college settings (Deek & Friedman, 2001). SOLVEIT is a computer environment that can contribute to an unambiguous presentation of important concepts and skills and the relationships among them (Woodward, 1991).

SOLVEIT is complementary to what Woodward found, the five stages of the instructional design process:

- Determining the concepts and skills that must be learned;
- Identifying the important relationships among concepts and skills;
- Organizing facts, concepts, and skills into logical hierarchies;
- Developing sets of instructional examples that unambiguously illustrate the range of concepts and skills that must be mastered; and
- Presenting the instructional examples to the student (Woodward, 1991).

The combination of a participatory design approach and SOLVEIT as a software engineering methodology will produce a highly accessible interactive software application. The SOLVEIT environment is a collection of tools supporting a six-step *process* for problem solving, software design and implementation. These steps are: (a) formulating the problem, (b) planning, (c) designing, (d) translating the solution, and finally, (e) testing and delivering the *product*. These steps have served as the fundamental architecture for programming (Deek, 1997) and writing instruction (Deek & Friedman, 2001).

The SOLVEIT problem-solving methodology builds on the four stages of problem solving defined by Polya (1945): (a) understanding the problem, (b) devising a plan, (c) carrying out the plan, and (d) looking back by integrating important elements of thinking and learning processes with the tasks of program development. Starting with the assumption that problems are generally solved by applying a sequence of transformations that change the given problem from an initial state to a goal state, and which taken together define the solution path (Simon, 1978; Mayer, 1983), the process begins with formulation of a general representation of a problem, and progresses to a more highly specified solution through a series of reformulations (Polya, 1945, 1962; Schoenfeld, 1979) or a transformation from the initial state to the goal state requiring explicit cognitive processing (Bloom, 1956). A well-defined problem contains three principal parts: (a) goal, (b) givens, and (c) unknowns, which are shaped by the process of problem-solving into a solution. A problem may also contain additional important elements that must be recognized and defined, such as conditions and constraints. Identifying the problem's goal, givens, unknowns, conditions, and constraints, based on the current representation of the problem, is the first step in problem-solving. One way to accomplish the problem-solving transformation is to redefine the problem into sub-problems and restate the goal in terms of sub-goals (Duncker, 1945; Newell & Simon, 1972; Wickelgren, 1974; Rubinstein, 1975; Mayer, 1983).

Two main kinds of problems are recognized in problem-solving research (Rubinstein, 1975): problems of analysis, also known as transformational problems, and problems of synthesis, which are the kind of problems frequently faced by engineers. Of course, some problems may have characteristics of both analysis and synthesis. Rubinstein (1975) defined a problem of analysis as one in which the solution consists of a series of transformations, or changes in the representation, of a given problem statement into a final solution. Greeno (1978) referred to these as problems of transformation (p. 245) where, given an initial state and a desired goal, a set of operators are defined and applied to produce the solution. Polya (1945) suggested using regressive reasoning or solution backwards (p. 142) to solve such problems. In this technique, one works backwards from what is required, developing and executing a plan that always keeps in mind the goal or result sought, until one eventually arrives back at the problem givens. In contrast, a problem of synthesis is one in which the major effort is in selecting and integrating known components to achieve a desired goal. While the problem as a whole may be new, the individual steps needed to solve the problem are not (Rubinstein, 1975). Greeno (1978) referred to these as problems of arrangement (p. 255): given some components, the task is to find a combination of components that meet the solution criteria. Polya (1945) referred to these as problems of construction (p. 23) and suggested solving them using progressive reasoning, which starts with what is already known, such as the plan of action. As an example, integrating the known kinds of components of a computer into a desired architecture requires selecting components with appropriate characteristics, for each type of component category, in such a way that the selected components can be integrated into a system that meets the requirements of the problem.

SOLVEIT first stage is problem formulation, during which students describe the problem, refine the problem, and extract the facts. Students verbalize the problem, as if they are doing the teaching, by explaining it to the rest of the class. The next stage is solution planning, when students generate possible solutions to the problem, select the best strategy to achieve the goal, breakdown the problem into components (if possible), and organize and associate the facts with the proper component. Again, students are encouraged to verbalize their thoughts on the solution process and asked to explain or defend their position(s) with regard to the proposed strategy for solving the problem. Solution design is the third step of the SOLVEIT methodology. It includes the sequencing and organizing of the components, specifying the data to be used for each component as well as the algorithmic logic and procedure. Students list each component part and verbalize the way that the problem will be solved. At this point the teacher encourages the students to talk it through and possibly "teach it back" to the software engineers. Solution translation, the next step in the process, includes composing the spatial and mathematical operations for each module, combining operations as needed, obtaining and displaying the final result. Solution testing, the fifth stage, is designed to ask whether the solution is complete, does the solution make sense, and if modification is needed, what is the nature of that modification? Ultimately, solution delivery, in which the process is reviewed, and the methodology and final answer are presented, situates the primary school students as both users and testers of the new software application.

DIGITAL TECHNOLOGY TO FACILITATE LEARNING-BY-DOING

The New Jersey Core Curriculum Standards finds that today's students are bored and discouraged, and they often come to believe that success is contingent on some innate ability as opposed to a skill that can be acquired. As the standards state, "Only in the United States do people believe that learning mathematics depends on special ability. In other countries, students, parents, and teachers all expect that most students can master mathematics if only they work hard enough." They also note that New Jersey students fail to see the relevance of quantitative subject such as mathematics in their daily lives beyond "shopkeeper mathematics." Therefore, the New Jersey Core Curriculum Content Standards developed a curriculum that has higher expectations of students, and goes beyond the acquisition of basic skills to include a variety of mathematical models. Additionally, it devotes more time to problem-solving and active learning. In keeping with these enhanced goals, the partnership and project described here seek to add interactive technology to the tools necessary to replace memorization and a skills and drills approach to classroom learning.

When a person has used a particular approach over time to solve a certain type of problem that becomes his or her "mental set" (Crider, Goethals, Kavanaugh, & Solomon, 1989). This indicates that concepts that are learned at a young age become a part of a person's thinking process, or "mental set," providing a richer conceptual vocabulary to use in later studies. The concept behind teaching any subject as reasoning is to transmit a real understanding of the subject's concepts to students. To have students discover for themselves an idea before encountering a mathematical symbol, for example, "unmasks" the concept behind the mathematical notation. This approach may encourage students to use their creativity; for only once a formula is *understood* can it be creatively applied. Furthermore, once students understand concepts they can create their own schemes. They will assimilate new methods and thinking patterns into ones with which they are already familiar. In terms of the software applications being developed, students will learn how to plan alternative routes to destinations when they encounter unexpected roadblocks, establish a variety of itineraries of landmarks and cultural centers for visitors to their city.

By setting roadblocks, students will find the process of learning mapping skills to be a reasoning process, one whereby the child discovers concepts experientially. Students are presented with tools and situations and through interacting with these tools the students come to discover foundational principles. In this process, the teacher takes the role of a guide, rather than that of an instructor. This is in concert with the educational theory of *constructionism,* an outgrowth of Piaget's theories, which states that students take an active role in constructing their own paradigms of understanding. Learning is largely a function of action (Kafai & Resnick, 1996).

NJIT students will lend digital technologies to this process by providing video of route segments to be called up in a graphical user interface in which students will create tours of their city and give directions to "lost" visitors. Through such interactivity, students will be situated in a context that corresponds closely to the way that children learn naturally. As Nicholas Negroponte, Mitchel Resnick, and Justine Cassell stated in *Creating a Learning Revolution,* computers can "enable children to become more active and independent learners, taking charge of their own learning through direct exploration, expression, and experience." This process can continue throughout their educational experience. As Negroponte believed, "Our goal is to develop digital technologies that enable children to continue to learn ever more advanced ideas by direct exploration and experimentation" (Negroponte, Resnick, & Cassell, 1997).

Software can also be made to simulate phenomena that would otherwise be impractical in the classroom experience. This will not only broaden the student's educational experience, but will also provide a more meaningful context within which to integrate knowledge that they learn. This can be used to great advantage in the elementary school years when children need to learn from contexts with which they are familiar. By assembling videos in which students navigate their city, explore museums, concert halls and baseball stadiums, they are accessing civic assets from their classrooms.

When students explore a given program at will, they are in effect designing their own lesson plan that is based on their own interests. The concepts that students learn through this process are therefore placed in contexts that the students will find more meaningful. When students are doing something they find interesting and enjoyable they can learn a great deal (Shank & Cleary, p. 98). Software that is designed with these aspects in mind can significantly enhance students' educational experiences by providing contexts that will give more meaning to children's explorations.

CONCLUSION

Described in this article are the underlying theories, past activities of participants, and current roles of three principle partners in an ongoing educational software development project. Elementary school children work with college seniors and technology consultants to implement a problem-solving methodology within a collaborative working environment to design, develop, and implement a multimedia software application that enhances time and space orientation abilities of children, puts the programming, interface design, and multimedia systems capabilities of college students into action, all while increasing the levels of interaction between and understanding of several Newark community groups, cultural centers, and the city's children. SOLVEIT, a problem-solving methodology, is applied to a requirements-driven software development process and is found to be congruent with Druin's participatory design methodology, resulting in each participant realizing the benefits of combining these methods as they, together, develop an educational software tool that offers each access to and a new view of their shared city.

References

Bloom, B.S. (Ed.) (1956). Taxonomy of educational objectives. *Handbook I: Cognitive domain*, New York: McKay.

Carver, S., Lehrer, R., Connell, T., & Ericksen, J. (1992). Learning by hypermedia design: Issues of assessment and implementation, *Educational Psychologist, 27* (3), 385-404.

Crider, A., Goethals, G., Kavanaugh, R., & Solomon, P. (1989). *Psychology, 3/e*. Boston: Addison, Wesley, Longman.

Deek, F.P., (1997). *An integrated environment for problem solving and program development*. Unpublished doctoral dissertation, New Jersey Institute of Technology.

Deek F.P. & Friedman, R. (2001). Computing and composition: Common skills, common process. *Journal of Computer Science Education-ISTE SIGCS, 15*(2). [Online]. Available: *http://www.iste.org/sigcs/community/jcseonline/2001/11/index.html*

Druin, A. (1999). Cooperative inquiry: Developing new technologies for children with children. *Proceedings of CHI'99*, pp. 592-599. New York: Association for Computing Machinery Press.

Druin, A. (2000). The role of children in the design of new technology. Submitted to *ACM Transactions on HumanComputer Interaction*. [Online]. Available: http://citeseer.nj.nec.com/druin99role.html

Duncker, K. (1945). *On problem solving.* Psychological Monographs 58, No. 5. (Whole # 270.) Washington, DC: American Psychological Association.

Filho, W. (2001). Requirements for an educational software development process. *Proceedings of the 6th annual conference on Innovation and technology in computer science education***,** pp. 65-68. New York: Association for Computing Machinery Press.

Guskey, T.R. (1986). Staff development and the process of teacher change, *Educational Researcher, 15*, 5-12.

Greeno, J.G. (1978). Natures of problem-solving abilities. In W.K. Estes (Ed.), *Handbook of learning and cognitive processes*, pp. 239-270. Hillsdale, NJ: Lawrence Erlbaum.

Haney, J., Czerniak, C., & Lumpe, A. (1996). Teacher beliefs and intentions regarding the implementation of science education reform strands. *Journal of Research in Science Teaching, 33*, 971-993.

Jonassen, D.H., Beissner, K., & Yacci, M.A. (1993). *Structural knowledge: Techniques for conveying, assessing, and acquiring structural knowledge*. Hillsdale, NJ: Lawrence Erlbaum.

Kafai, Y., & Resnick, M. (Eds.) (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Mahwah, NJ: Lawrence Erlbaum.

Kimmel, H., & Deek, F.P. (1995). Instructional technology: A tool or a panacea? *Journal of Science Education and Technology, 4*, 327-332.

Kulik, J. (1995). *Change, XXVII (2)* (March/April), 20-27.

Lehrer, R. (1993). Authors of knowledge: Patterns of hypermedia design. In S.P. Lajoie & S.J. Derry (Eds.), *Computers as cognitive tools* (pp. 197-227). Hillsdale, NJ: Lawrence Erlbaum.

Mayer, R.E. (1983). *Thinking, problem solving, cognition*. New York: W.H. Freeman.

Mitchell, L.S. (1991) *Young geographers: How they explore the world and how they map the world*. New York: Bank Street College.

Negroponte, N., Resnick, M., & Cassell, J. (1997). *Creating a learning revolution* [Online]. Available: http://www.unesco.org/education/educprog/lwf/doc/portfolio/opinion8.htm

New Jersey Department of Education (1999). *New Jersey core curriculum content standards*. Standard 6.7: All Students Will Acquire Geographical Understanding By Studying The World In Spatial Terms. Trenton, NJ: Author.

Newell, A., & Simon, H. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

Papert, S. (1997). *Remarks to a House of Representatives panel of technology and education* [Online]. Available: http://kids.www.media.mit.edu/projects/kids/sp-talk.html

Piaget, J. (1953). How children form mathematical concepts. *Scientific American*, *189*(5), 74-79.

Piaget, J. (1972). *The psychology of the child*. New York: Basic Books.

Polya, G. (1945). *How to solve it*. Princeton, NJ: Princeton University Press.

Polya, G. (1962). *Mathematical discovery: On understanding, learning and teaching problem solving*. New York: Wiley.

Reeves, T.C. (1998). *The impact of media and technology in schools* [Online]. Available: http://www.athensacademy.org/instruct/media_tech/reeves1.html

Rubinstein, M. (1975). *Patterns of problem solving*, Englewood Cliffs: NJ, Prentice Hall.

Schoenfeld, A.H. (1979, May). Explicit heuristic training as a variable in problem solving performance. *Journal for Research in Mathematics Education*, *10*, 173-187.

Shank, R., & Cleary, C. (1995). *Engines for education*. Hillsdale, NJ: Lawrence Erlbaum.

Slavin, R. (1991). *Educational Psychology, 3/e*. Englewood Cliffs: Prentice-Hall.

Simon, H. (1978). Information-processing theory of human problem solving. In W.K. Estes (Ed.), *Handbook of learning and cognitive processes*, Hillsdale, NJ: Lawrence Erlbaum.

Sobel, D. (1998). *Map making with children*. Portsmouth, NH: Heineman.

Stoyanov, S. (1997). Cognitive mapping as a learning method in hypermedia design, *Journal of Interactive Learning Research*, *8*(3/4), 304-323.

Wickelgren, W.A (1974). *How to solve problems*. San Francisco: W.H. Freeman.

Woodward, J. (1991). Procedural knowledge in mathematics: The role of the curriculum. *Journal of Learning Disabilities*, *24*, 242-251.