Using a Functional Approach to Change Preservice Teachers' Understanding of Mathematics Software

Terri L. Kurz
California State University, Bakersfield

James A. Middleton
Arizona State University

Abstract

This study examined the structure of two preservice teachers' understandings of educational software in mathematics using repertory grid techniques. Specifically, the study focused on how teacher educators can enable preservice teachers to discern the features of mathematics software, and develop pedagogical goals that reflect the affordances and constraints of available tools. Results showed a deepening of knowledge and a differentiation of knowledge following experiences of exploration, evaluation, and comparison of different types of software. Results imply that directing preservice teachers' attention to utilizing the functional characteristics of software for developing plans and for the design of instruction may be fruitful for more effective integration of software in their future teaching. However, participants' preexisting content knowledge and their pedagogical worldview tempered this effect. (Keywords: mathematics education, mathematics-based software, personal constructs theory, preservice teachers, technology integration.)

INTRODUCTION

Technology has tremendous potential for enhancing mathematics instruction (Connell, 1998; Roschelle, Pea, Hoadley, Gordin, & Means, 2000); it can be used to strengthen student learning, assist in developing mathematical concepts, and can enrich student learning in the areas of richer curricula, enhanced pedagogies and more effective organizational structures (Dede, 2000). However, technology has not reached its potential in preservice teacher instruction; newly graduated teachers often do not have the experience to use computers in the classroom or knowledge about available software (Gunter, 2001). In a study conducted by Smith and Shotsberger (2001), most preservice teachers identified technology as important in mathematics education to assist in the development of concepts, but those same people were uncomfortable discussing the specific uses of technology for instruction due to a lack of knowledge. Many preservice teachers feel that they are not prepared to teach with technology after they graduate (Carlson & Gooden, 1999). The question that develops: What kinds of experiences should preservice teachers have in regard to the integration of technology and mathematics?

An effective way to prepare teachers to use technology in mathematics is to prepare them to utilize technology for student use as a cognitive tool (Lajoie, 1993; Pea, 1986; Roschelle et al., 2000). A tool can be defined as a cultural artifact that "...predisposes our mind to perceive the world through the 'lens' of the capability of that tool," making it easier or harder to perform certain activities (Brouwer, 1996-1997, p. 190). Many technological devices can be used as tools including software, calculators, languages, and programs (Connell, 1998). Lajoie (1993) describes the benefits of using the computer as a tool for instruction in an educational setting. Technological tools: (1) support the cognitive process, (2) share the cognitive load, (3) allow more students to engage in mathematics that would otherwise be out of their reach, and (4) support testing of ideas and conjectures.

The intent of this investigation was to understand how preservice teachers' thinking changed concerning the integration of technology and mathematics in their future classrooms after experimenting, exploring, and evaluating five kinds of mathematics-based software. The question addressed in this research was: How can we enable preservice teachers (in the constrained conditions of typical university technology instruction) to meaningfully discern the features of tools for thinking in mathematics, and subsequently project pedagogical goals and tasks that potentially make more and better use of available tools?

TOOLS FOR MATHEMATICS INSTRUCTION

Five categories of software were developed with a focus on tool-based use in the mathematics classroom: Review and Practice, General, Specific, Environment and Communication (Kurz, Middleton, & Yanik, 2005). All of these categories can be used as part of a complete mathematics curriculum, with each type of software highlighting a different type of learning. When using software for the reinforcement of previously learned material, it falls into the Review and Practice category. The student does the same type of mathematics problems in a repeated manner, and no new conceptual material is introduced. General software is designed for use across a variety of mathematical domains; it has many different applications. The teacher must examine the area of mathematics that the software will be used in and develop lessons that promote the type of learning he or she will focus on. Software designed to emphasize learning in a particular area of mathematics is an example of Specific software. The focus is on the learning of a distinct mathematical topic, such as fractions, reflections, or order of operations. Software used as an Environment tool integrates different types of learning in a variety of subject areas. Environment software provides a virtual place for students to guide their mathematical learning. It takes students to a new place without requiring them to leave the classroom. Communication software is designed for sharing information between students and another party or parties including the instructor, other teachers, students, or professionals (in education or outside of the field). The idea is to increase the understanding of mathematical concepts and the ability to articulate mathematical arguments and concepts through discourse (see Kurz et al., 2005 for the complete framework).

UTILIZING SOFTWARE IN MATHEMATICS

Software programs, such as those described, are sometimes introduced to preservice teachers in their education courses, with little or no thought into the way the software is to be used in their subsequent careers. Sometimes, the software is simply shown for demonstration purposes, with no real thought given to the design and use of the software in the preservice teachers' future classroom (Drier, 2001).

Mathematics education courses at the college level can meet the needs of preservice teachers by showing them how to discriminate between available software. "The tools of technology alone are no more useful than pen and ink if the user is incapable of discriminating between useless information and quality information" (Blake, Tchoshanov, Della-Piana, Pacheco, & Brady, 2001, p. 256). A foundation of skills to discriminate between the constraints and affordances of software may help preservice teachers recognize the potential of mathematics-based programs.

Preservice teachers in their education and content-area courses can explore educational software. Deep exposure to software in mathematics will encourage students to critically analyze. With a broad knowledge of each software package's strengths and weaknesses, along with the ability to use various applications to achieve certain goals, students can be technologically successful (Oppong & Russell, 1998). Thorough familiarization and understanding of software is crucial for future implementation.

Preservice teachers also need hands-on experiences. Hands-on activities and positive experiences that address the learner at his or her level and learning style are important when learning with technology (Gunter, 2001). Delivery of information to preservice teachers is not sufficient; instead, preservice teachers need to appraise their beliefs about subject matter, learning and instruction in terms that relate to innovation (Marx, Blumenfeld, Krajcik & Soloway, 1998).

Finally, reflections on the curriculum are helpful to the student's technological and mathematical understanding. When students write down their reflections, the goal is to have them look at the material—whether a textbook, article, software or anything related to the course—and critically evaluate the material. A focus on positive and negative attributions, along with affordances and constraints, can offer insight. Reflections have been described as a powerful tool for teaching and learning about technology (Gunter, 2001). Reflections help the students focus on the dominant components of technology and how the components apply to student learning.

Tool-based software has been an area of focus in mathematics education for some time. However, there has not been research in the five major categories of tools described in this paper. The focus of this paper is to examine preservice teachers' changes in thinking, particularly their understanding of the affordances and constraints embodied by the five categories of software based on the framework of the course. Examination of the affordances and constraints through experience with, and analysis of, exemplars of the five categories of tools should have an affect on the thinking of preservice teachers. In short, this research attempts to identify how preservice teachers broaden in conceptualization of the kinds of software appropriate for integration into mathematics instruction, and deepen to be more specific regarding the fit of tool to task.

METHODS

Two cases were investigated, each of which described an individual preservice teacher's thinking toward the use of mathematics-based software for instruction. Within each case, the research question was answered using a variety of techniques. Utilizing the methods of Personal Constructs Theory (Kelly, 1955), pre-repertory grids and post-repertory grids were administered to each preservice teacher, classroom observations and transcripts were generated, and heuristic questions were administered and analyzed along with electronic conversations. The course was an intervention designed to present the preservice teachers with mathematics-based software in a meticulously designed way to help them draw distinctions between the five types of mathematics-based software that afford different qualities of teaching, learning and communication.

Students were enrolled in an upper-division course for preservice teachers designed as an introduction to mathematics-based software. The students met once a week for six weeks, two hours per session. An example of each category of software was explored each week in groups, with the exception of the communication software, which was used throughout the course. For specific and general software, students were given constructivist activities to experience as a learner; with review and practice, students played the interactive math game in a group; and with environment software, students solved the dilemma specified on the computer-based video. The instructor of the course would take the first 10 to 20 minutes of class to demonstrate the software tools available. After each software experience, students participated in a communication group in an online, threaded discussion and completed heuristic evaluations of the applications (Squires, 1997). In each of these discussions, the conversations were seeded by asking students to make distinctions between the different applications they had experienced, regarding the constraints and affordances each offered to mathematics instruction. Students were encouraged to analyze and comment on observations made by other student postings.

PERSONAL CONSTRUCTS THEORY (PCT)

Personal Constructs Theory (Kelly, 1955) uses the metaphor that all people are scientists and that each person interprets constructs based on the way he or she sees the world. A construct is a way of finding the similarities or differences between things, whether those things are people, events, items, or software. The constructs that each person develops are based on an organized system of thoughts; and they are linked and interrelated to one another (Beail, 1985). PCT can be used as an attempt to understand how a person thinks or feels about a particular person, idea or object based on his or her life experiences. It is unique in that it allows people to give meaning on their own terms and provisions as opposed to the majority of techniques in which the researcher determines the specific elements and constructs.

Kelly developed the repertory grid as a method to explore personal constructs. The repertory grid is a procedure designed to look at the configuration of personal meanings attributed to an area of inquiry (Feixas & Alvarez). The person under investigation gives the researcher "elements" and "constructs" based on his or her familiarity. Elements are the domain for the investigation; in this case, it was types of software. Constructs are interpretations based on how the person sees the elements. There are several different ways to investigate constructs. For this particular investigation, rating scales were used to complete the repertory grids. After the elements and constructs were elicited, a rating system was developed and used to evaluate perceptions. The subjects then rated the elements based on the constructs given.

Once elements and constructs were elicited, students were presented with a series of element-construct pairs. For each pair, students were asked to rate, on a scale of 1 to 5, the extent to which the construct defined the element. For example, pairing the element "Excel" with the construct "Can discover properties by doing activities on the program," Andrea was asked to rate how much "Can discover properties by doing activities on the program" defined the pedagogical features of "Excel." This was done for all pairwise combinations of elements and constructs. The result is a (number of elements) x (number of constructs) matrix of ratings that provided a reliable index of each student's perceptions of the pedagogical features of educational software to which they had been exposed.

Repertory grid techniques were used to measure how the perceptions of preservice teachers' thinking toward technology in mathematics changed as a result of exposure and evaluation of mathematics-based software. An individualized pre and post-repertory grid was administered to the preservice teachers in this study. The pre-repertory grid was administered at the beginning of the course, eliciting elements based on the answer to this question: "What are the types of software that can be used in an educational setting to teach mathematics?" Anchors were then added to the same question and it was probed again. The constructs were elicited by using pairwise comparisons of the elements. This same procedure was followed for the post-repertory grid.

ANALYSIS OF REPERTORY GRID

By examining change in both depth of constructs and their organization cognitively, it is possible to determine quantitative changes in teachers' conceptualizations due to the intervention of the course. Ward's Method of cluster analysis was applied to constructs in the repertory grids to determine inter-construct distances. Inverse scree tests were then applied to the agglomeration schedule of different cluster solutions to determine the number of significant vs. error clusters (Lathrop & Williams, 1987). Two types of clusters were examined: a clustering of the elements (software) and a clustering of the constructs. Both types of clusters were conducted pre and post-exposure, making four the total number of clusters for each preservice teacher. After the number of significant clusters was determined, the agglomeration schedule and dendrograms were analyzed to determine which elements or which constructs were placed in each cluster. Transcripts of observations, reflection responses, group discussions and heuristic responses were used to contextualize and describe the changes in thinking as they took place in the course. These analyses assisted in the creation of an overall model of how thinking changed in regard to the use of technology in the mathematics classroom for the two preservice teachers studied.

CLASSROOM OBSERVATIONS AND TRANSCRIPTS

During each session of software exploration, video cameras were placed behind the preservice teachers. Large mirrors were placed to the left of the monitor to capture facial expressions of the preservice teachers. With this design, it was possible to view the computer monitor along with the facial expressions and hand motions of the preservice teachers. Notes were taken immediately after each session. The videos were watched several times and complete transcripts of the entire software sessions were created. The classroom observations and notes focused on the interaction the students had with one another and the software. Analysis focused on experience, questions, and comments regarding the features of the software.

The observational data was also less structured in approach, with a focus on flexibility and a minimum of pre-structuring (Foster, 1996). We focused observations on instances where work or discussion was aimed at how features of different applications contribute to different learning outcomes of our participants' future students. Conversations based on how they viewed the use of the software, comments while experiencing the software, facial expressions and body language, and questions based on the software comprised the recorded transcripts and notes.

HEURISTIC QUESTIONS

How does one determine what software is effective or valuable in the mathematics classroom? Squires and Preece (1996) emphasize an approach that involves two central concerns: learning and usability. Software evaluation should focus on student learning, aligning with the way students learn along with usability; the interactions should be natural and intuitive. Accordingly, the software should be usable and be appropriate for the learning tasks to promote synergy. When this synergy occurs, there is a union that flows between the use of software and the learning process (Squires & Preece, 1996).

Squires and Preece (1996) developed a model of software evaluation. In this model, there is a distribution of cognition, which leads to students constructing their knowledge through interactions. There are two areas that should be understood when using this approach: The software should be evaluated from a constructivist perspective; and the software is dependent on the context in which it is presented. The model was designed for use with inservice teachers with a focus on increasing teacher awareness of learning and usability (Jones et al., 1999). This model takes a predictive approach to software evaluation; teachers try to determine the learning and usability afforded by the software as it may possibly relate to student learning.

For the purposes of this study, heuristics are more suited than checklists because they offer a depth not possible with simple checklist style questions; heuristics are dependent on the preservice teacher's examination of the software while engaged in mathematical experiences. The students received a list of the heuristic questions to be completed after each software experience. These questions were due the following week to provide students with ample time to reflect on the software. The answers to the 10 questions were analyzed using the qualitative techniques that represented a holistic description of the preservice teachers' examination of the software. The heuristic questions allowed the preservice teachers to reflect on what they thought about their experiences with each software program and how that experience might enhance their students' ability to learn mathematics (Squires, 1997).

ELECTRONIC COMMUNICATION AND REFLECTION

The electronic communication for this class took place utilizing Yahoo Groups (available at www.groups.yahoo.com). Each week, the students were asked to respond to an open-ended question, which focused on the features of the software and how the features support or hinder student learning. Upon completion of the course, the students were asked a reflection question that allowed them to sum up what they learned about mathematics-based software. This provided some insight into how they see the learning process as it relates to mathematics-based software. The question allowed for a comparison of changes in thinking, when compared to the first communication question.

RESULTS

Analysis of repertory grids indicated that following instruction, teachers' understanding had broadened to include more defining features, and became more organized with respect to the five general categories of mathematics tools presented. In addition, the ways in which examples of software were categorized changed in organization, indicating that teachers were using a pedagogical lens by the end of instruction.

Data were gathered on four preservice teachers. Two preservice teachers were selected for analysis, one in elementary and the other in secondary. While all of the preservice teachers experienced some change in thinking in relation to how they viewed learning in regards to mathematics-based software, only two cases were selected for complete analysis. The cases selected had the most prolific qualitative data to support the quantitative analysis. Additionally, a preservice teacher at the elementary level and at the secondary level was desired, so candidates with the most abundant qualitative data in each of these areas were selected.

Both individuals completed their assignments with detail and participated online with the threaded discussion on time. The preservice teachers were randomly assigned pseudonyms in alphabetical order. Andrea was planning on teaching mathematics at the middle school level; she majored in computer science before deciding that she wanted to teach mathematics. Andrea had a constructivist view of how students learn mathematics. Brittany had less experience in mathematics; her ultimate goal was to teach at the elementary school level.

One of the cases not selected for analysis was a preservice secondary mathematics teacher. She seemed very strong in her mathematical ability and participated with depth in class; however, her heuristic evaluations were lacking comprehensive details—often a sentence or two in response to the questions. The other candidate who did not get selected was an elementary preservice teacher. She was disorganized when completing online discussions and was an irregular participant. Some of her postings took place after the due date. At the end of the course, she apologized for being so unfocused this semester and said it was out of character.

ANDREA'S SOFTWARE CATEGORIZATION

Pre-Exposure

Andrea had some experience with mathematics-based software before the course began, and without anchors, was able to list the different types of software used to teach mathematics: Geometer's Sketchpad, Maple and Java applets. After anchors were given, she added five more: Excel, PowerPoint, word processing, communication groups, and the Internet. Her software categorization was based on only seven of these software types; communication groups was omitted because the data she gave was incomplete (she put a question mark when rating the constructs). Her first dendrogram can be seen in Figure 1. After conducting inverse scree tests to determine the significant number of clusters, two clusters were determined. Andrea had a cluster that included productivity software: PowerPoint and word processing. She also had an "other" category, which contained software that is used to teach mathematics in ways other than for presentation purposes.

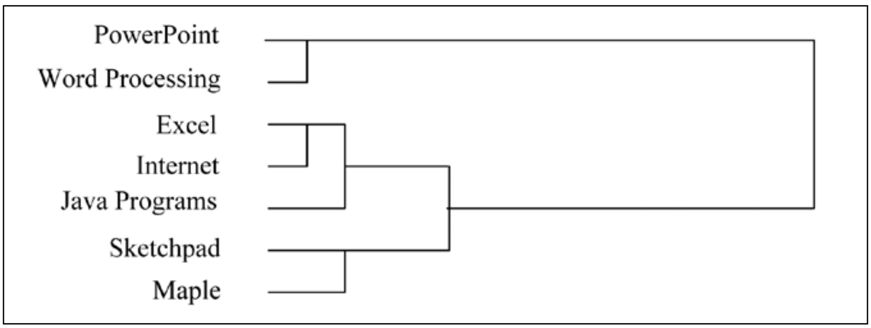

Figure 1: This is Andrea's Pre-Exposure Dendrogram of Software Types.

Post-Exposure

After exposure to and experience with the software, and heuristic questions and electronic communications as a guide, she developed a more complex categorization system of mathematics-based software (Figure 2). The inverse scree test determined three clusters. The first cluster consisted of production types of software including PowerPoint and word processing, along with other software that did not fit into her other categories: Maple, communication groups and the Internet. The second cluster consisted of general tool-based software: Excel, Geometer's Sketchpad and Java programs. These tools support open-ended discovery of mathematics. Her final category included Specific, Environment and Review, and Practice software: TesselMania, Jasper, and Math Blaster. This category is based on software that is designed to teach a specific area of mathematics, but there are not clear distinctions on how each program differs in regard to support for student learning.

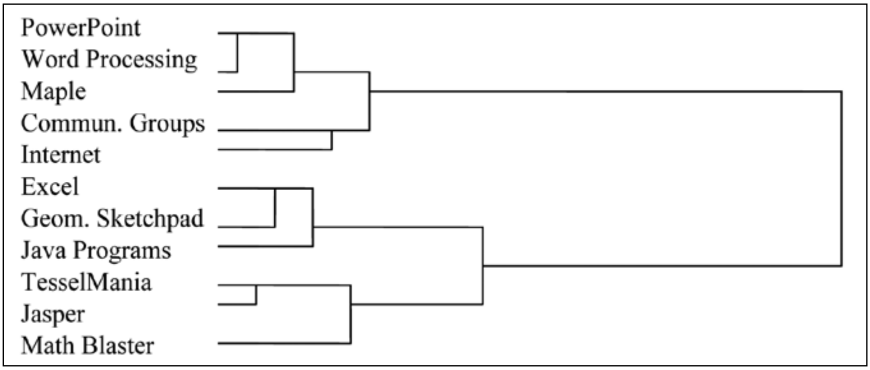

Figure 2: This is Andrea's Post-Exposure Dendrogram of Software Types.

ANDREA'S MATHEMATICS-BASED SOFTWARE CONSTRUCTS

Pre-Exposure

After completing the pairwise comparisons of software in the beginning of the course, Andrea determined 10 software constructs. Figure 3 is a dendrogram of her software constructs that demonstrates how closely related the constructs are to one another. (Due to space constraints, the constructs are simplified into shorter statements; the complete statements can be found in Table 1.) The constructs were more general in the beginning of the course. They referred to what the software can do (add, subtract, make a long list of numbers in seconds, solve equations, and do different kinds of graphs), without an emphasis on how the software supports learning.

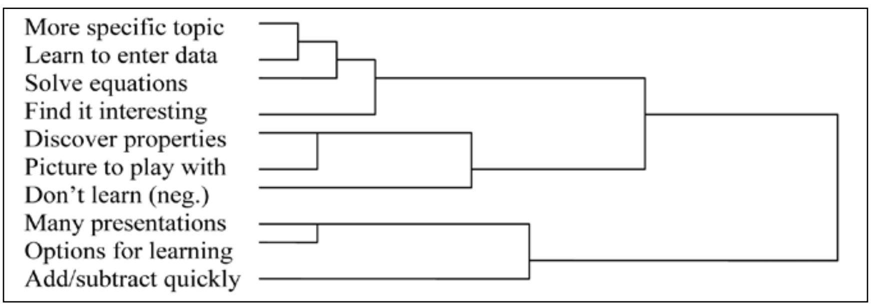

Figure 3: This is Andrea's Pre-Exposure Dendrogram of Software Constructs.

Post-Exposure

At the end of the course, her pairwise comparisons yielded 17 new constructs for a total of 27. Figure 4 provides a dendrogram of all of the constructs. These clusters indicate that Andrea developed a richer understanding of the ways in which software can benefit student learning in the mathematics classroom. Her new clusters were more developed and clear in focus. After the course, the constructs focused more on the support the software offers in terms of student learning and mathematical development (for example, it makes students really think about the problem and the answer, and students can explore and discover results), including the use of software as a tool to aid in student learning (general tool used to learn many topics).

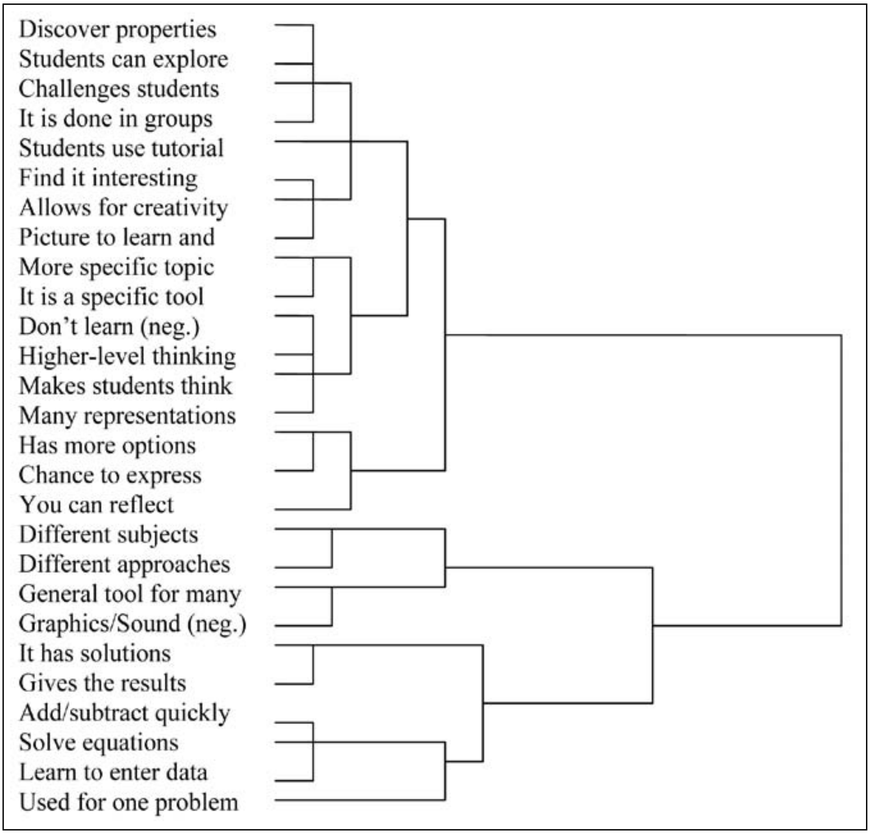

Figure 4. This is Andrea's Post-Exposure Dendrogram of Software Constructs.

The constructs created three distinct clusters. Table 1 provides a categorization of both the pre-exposure and post-exposure clusters, along with a classification of the clusters. The clusters show that Andrea developed more constructs in all three of the clusters. Within each cluster she also had some changes in the categorization of the constructs, for example "can be used to add, subtract, make a long list of numbers in seconds" started out in the Relates to Options within the Software cluster and moved to Focuses on Algorithms and Topics.

At the start of the course, all three clusters had about the same number of constructs. Focuses on Algorithms and Topics had the most constructs at four, while the others had three. After the course, Andrea's construct numbers within the clusters expanded. Her most thorough and detailed cluster was that of Emphasizes Discovery through Activities. This category had a total of 17 constructs (including the pre- and post-exposure constructs for all of the categories). Focuses on Algorithms and Topics contained six constructs, while Relates to Options within the Software had the least amount of constructs (four). The course design influenced the greatest change in Discovery through Activities. This change corresponds to the course design, which focused on utilizing a social constructivist approach to the learning of mathematics with software.

Table 1: Andrea's Cluster Membership of Mathematics Software Constructs

Clusters	Pre-Exposure Cluster Constructs	Post-Exposure Cluster Constructs
Cluster 1 Focuses on Algorithms and Topics	More specific topic; You learn how to enter a particular problem in the program; You can solve equations and do different kinds of graphs; Students find it interesting because they get to do something	Gives the results and shows you the steps; Can be used to add, subtract, make a long list of numbers in seconds; You can solve equations and do different kinds of graphs; It has solutions; You learn how to enter a particular problem in the program; Used for solving one problem
Cluster 2 Emphasizes Discovery through Activities	Can discover properties by doing activities on the program; There is a picture that you can play with to learn and discover something; Students don't learn much (negative)	Can discover properties by doing activities on the program; Students can explore and discover results; Challenges students; It is done in groups and students can learn from each other; Students can learn from a tutorial that lets students interact with the software; Students find it interesting because they get to do something; Allows students to be creative; There is a picture that you can play with to learn and discover something; More specific topic; It is a specific tool to help students learn; Students don't learn much (negative); Students think at higher levels; Makes students really think about the problem and answer; Provides different data representations so you have a better chance of reaching all the students; Has more options for students to learn; Gives you a chance to express not only what you learned but if you learned it right; You can reflect upon what you have learned
Cluster 3 Relates to Options within the Software	Provides different data representations so you have a better chance of reaching all of the students; Has more options for students to learn; Can be used to add, subtract, make a long list of numbers in seconds	You can use it for different subjects; There are different ways to approach the problem; General tool used to learn many topics; Graphics and sound overpowers the learning (negative)

Andrea's classroom interactions, electronic communication, heuristic answers, dendrograms and clusters show a clear picture of how her thinking changed in relation to software used to teach mathematics. She distinguished between the features of the software and described how these features fit into her model of student cognition. Her depth and breadth of the assignments and activities supported her knowledge acquisition and led to changes in thinking.

BRITTANY'S SOFTWARE CATEGORIZATION CONSTRUCTS

Pre-Exposure

At the beginning of the course, Brittany's exposure to mathematics-based software was quite limited. Without anchors, the only software she was able to identify was Number Munchers. After the anchors, she recalled four additional programs: KidPix, multiplication/division games, Microsoft Office and Moneymaker Pro. Two clusters were determined by the inverse scree test: A Review and Practice category that included multiplication/division games and Number Munchers, and a General category that consisted of Microsoft Office, Moneymaker Pro and KidPix (Figure 5).

Post-Exposure

After the course was completed, Brittany had a deeper understanding of the categorization of software. Her dendrogram consisted of 10 software types, and the inverse scree test resulted in four categories of software (Figure 6). Her first category contained Review and Practice software types: multiplication and division games, Number Munchers and Math Blaster. TesselMania and Jasper made up her second category of software designed to teach a distinct content in

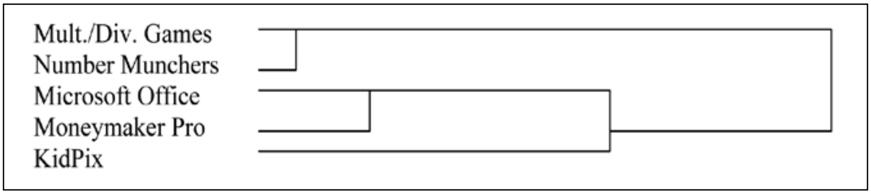

Figure 5: This is Brittany's Pre-Exposure Dendrogram of Software Types.

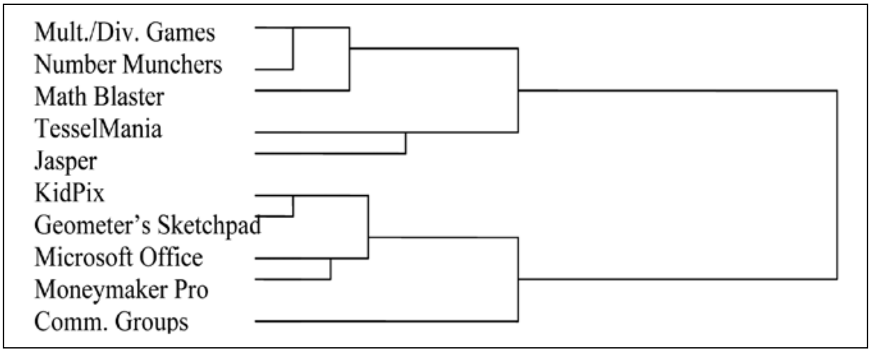

Figure 6: This is Brittany's Post-Exposure Dendrogram of Software Types.

mathematics. She had a third category (which appears to be a General category) that included KidPix, Geometer's Sketchpad, Microsoft Office, and Moneymaker Pro. Her final category consisted of one type of software: communication groups. Brittany developed a more detailed understanding of how different types of software support learning in mathematics. Her categories were clearly defined and she could ascertain the distinctions between them.

BRITTANY'S MATHEMATICS SOFTWARE CONSTRUCTS

Pre-Exposure

Brittany's pairwise comparisons yielded 13 software constructs as seen in her dendrogram of pre-exposure software constructs (Figure 7; the unabbreviated statements are located in Table 2). The dendrogram demonstrates how closely related the constructs are to one another. The constructs were well distributed into each cluster, with a focus on how software supported student learning, and how software can be used for review and practice issues. The also included the various features of software. There was no mention of software as a tool that supports learning through discovery or exploration.

Post-Exposure

After the course, Brittany's pairwise comparisons yielded 22 new constructs for a total of 35. Figure 8 is a dendrogram of all of the constructs (full statements can be found in Table 2). Her constructs started out general, emphasizing a linear approach to education along with the need for fun to motivate students (i.e., teaches essential math skills, increases motivation for learning, and focuses on memorization of facts). After the course, her constructs stayed within this framework, with some exceptions (i.e., students can see the mathematical concept visually, allows for hands on interaction with the software, and allows for more creative thinking activities).

Her clusters expanded in content after the exposure. Table 2 has the complete statements for the clusters of both the pre-exposure and post-exposure constructs. Cluster 1, "Describes how Software supports Student Learning,"

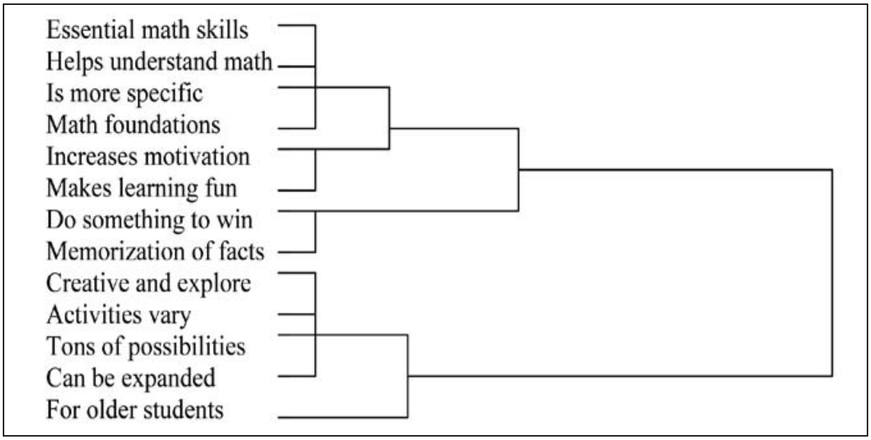

Figure 7: Brittany's Pre-Exposure Dendrogram of Software Constructs.

went from six constructs to 13. This category related to motivational aspects of software along with ways software supports student learning. Her Review and Practice constructs doubled from two to four. Her final cluster, "Emphasizes the Possibilities of the Software," had the largest gain—five constructs to 18. It is interesting to note that none of Brittany's constructs changed clusters after exposure. She also did not have any mention of tools or tool-based learning in any of her constructs.

Brittany's classroom interactions, electronic communication, heuristic answers, dendrograms, and clusters all indicate changes in her thinking. The elements and constructs of mathematics-based software expanded and developed into a model resembling the course structure; it connected with her beliefs about how students learn. Her course transcripts and heuristic responses dem-

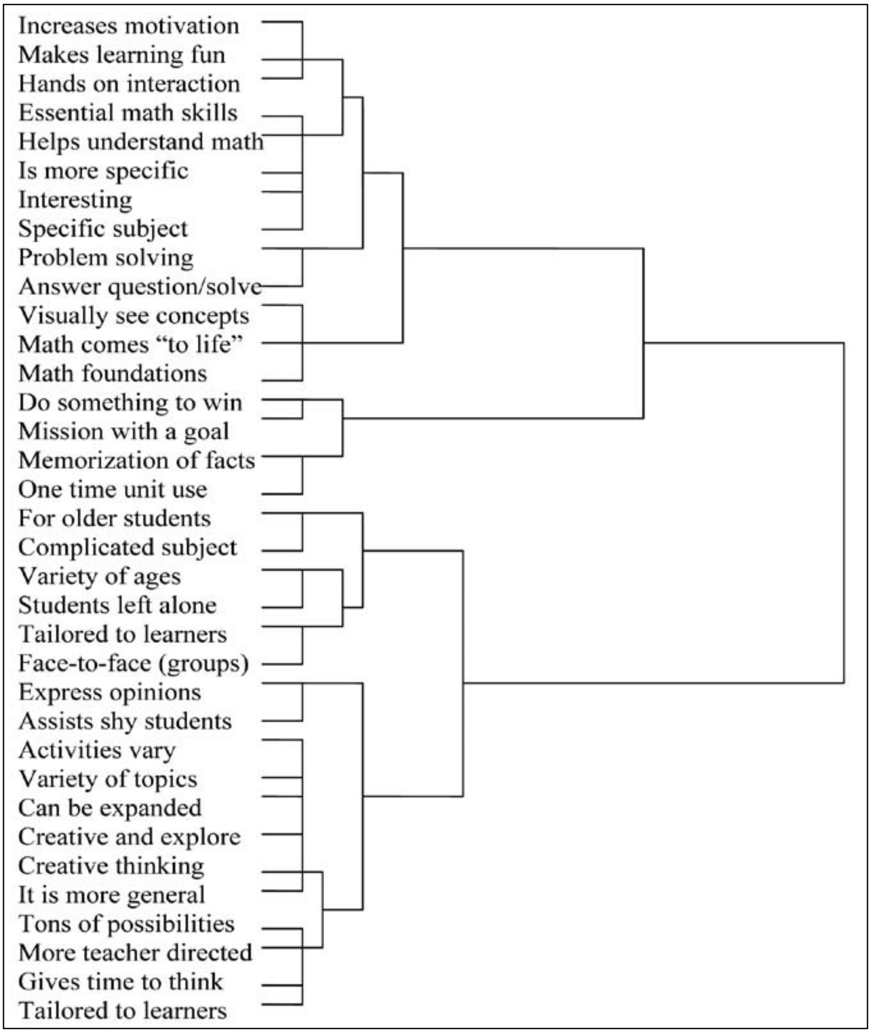

Figure 8. This is Brittany's Post-Exposure Dendrogram of Software Constructs.

onstrated a focus on completing tasks, not looking at the larger picture of mathematical development.

The interactions with their classmates and the software were different for the two preservice teachers. Andrea spent time on the activities and tried to understand the mathematical reasoning, both as a future teacher and from a student's

Table 2: Brittany's Cluster Membership of Mathematics Software Constructs

Clusters	Pre-Exposure Cluster Constructs	Post-Exposure Cluster Constructs
Cluster 1 Describes how Software supports Student Learning	Teaches essential math skills; Helps students understand math; Is more specific; Teaches mathematical foundations rather than concepts; Increases motivation for learning; Makes learning fun and exciting	Increases motivation for learning; Makes learning fun and exciting; Allows for hands on interaction with the software; Teaches essential math skills; Helps students understand math; Is more specific; Interesting; It thoroughly addresses a specific subject; Specific to problem solving; Allows students to answer questions and solve problems; Students can visually see the mathematical concept; Brings the math concepts "to life"; Teaches mathematical foundations rather than concepts
Cluster 2 Focuses on Review and Practice Aspects	Requires students to do something in order to win; Focuses on memorization of facts	Requires students to do something in order to win; It gives students a clear mission with a clear goal; Focuses on memorization of facts; Can only be used 1 time (unit) use
Cluster 3 Emphasizes the Possibilities of the Software	Allows students to be creative and explore; Activities vary, allowing for a wide range of learners; Tons of possibilities; Can be expanded; Specific to older students	Specific to older students; Has a more complicated subject matter; Works with a variety of ages; Students are left alone to solve the problem; Can be tailored to specific learners; Allows students to work face-to-face in groups; Allows students to express their opinions; Assists shy students that do not feel comfortable speaking up in class; Activities vary, allowing for a wide range of learners; Allows students to explore a variety of topics; Can be expanded; Allows students to be creative and explore; Allows for more creative thinking activities; It is more general; Tons of possibilities; More teacher directed; Gives students time to think about what was discussed and add to class discussions; Can be tailored to specific curriculum

perspective. She made comments about being the slow group and spending too much time on each activity. She did not move on until she was satisfied with what she was learning and confident in the mathematics behind the learning. Her conversations were rich in mathematics and confident in her learning of the subject. Brittany, on the other hand, appeared bored with the General and Specific software. With these two software types, her interactions were muffled and contained very few mathematically related terms or concepts. When experiencing Environment and Review and Practice tools, she was much more alert and entertained by the material; she looked as though she was having fun.

When looking at the electronic communications by the women, the thoroughness on the topics varied. Andrea had more detail, emphasized features, and how the software enables the students to learn. She emphasized using a discovery and exploration approach; she focused on this method to support objectives stressed in standardized tests and content-area knowledge. Her thoughts were quite focused and remained consistent throughout the course. Brittany did not stay as consistent in her communication correspondence. She agreed with what other students were saying and sometimes rehashed their comments, using the discovery and exploration examples used by other students with a less traditional approach. In other instances, she was more consistent with her approach to learning. For example, she focused on using software to make math fun and exciting. She liked software that was easy for students to understand and required little teacher preparation to use.

The heuristic questions demonstrated the association the features of the software had to the preservice teachers' view of cognition in mathematics. Their individual approaches to learning—those that they will employ as future teachers— were the basis for their answers. Andrea continued to emphasize using software as a tool for discovering mathematical concepts, while Brittany focused on completing tasks to reinforce mathematical skills already taught.

DISCUSSION

The curriculum model described, takes the approach of examining software use with a sense of application and relevance rather than for demonstration purposes without any consideration for future use in preservice teachers' classrooms (Drier, 2001). Included in the model was a basis for discriminating between the tools available to teach mathematics. (Blake, Tchoshanov, Della-Piana, Pacheco & Brady, 2001) This was supported through reflections (Gunter, 2001), electronic communication (Jonassen, Howland, Moore & Marra, 1999) and heuristic evaluations of software (Squires & Preece, 1996; Squires, 1999). The broad exposure to the software, along with investigations into the affordances and constraints of the features, allowed the preservice teachers to develop an understanding of the ways software can support their future students (Oppong & Russell, 1998).

The preservice teachers' perception of the features of the software changed. This was particularly apparent with the affordances and constraints each provided. Through their experiences in this course, both preservice teachers came up with many of the affordances and constraints described by researchers in the review of the literature in relation to the five software types (Kurz, Middleton, & Yanik, 2005). This is interesting, considering the students were in no way presented with any of the previous research findings, and that the preservice teachers' observations led the discussions—not the instructor's presentation. The descriptions of the affordances and constraints were much richer and more developed at the completion of the course as compared with the start of the course; an increase in features would be expected with experience and exposure. However, the importance lies in the preservice teachers having gained the ability to distinguish between the features and describe how these features support or hinder the learning process of their future students in only a few short sessions.

Additionally, both preservice teachers developed deeper understandings of and broader experience with mathematics-based software, and were able to expand their categorization of software after the course. Cluster analysis of software type revealed that the preservice teachers were able to fit the software they had been exposed to prior to this course into this model. For example, Brittany's dendrogram shows that she identified Math Blaster and Number Munchers as having similar features that utilize the same technique for mathematics instruction; both are Review and Practice Tools, and Brittany was able to characterize this. The same can be said of Andrea, who identified various types of General Software (pre-exposure) and placed them categorically together (post-exposure).

The clusters of the features of mathematics-based software expanded and developed after exposure for each of the participants. Each developed more detailed categories for the affordances and constraints each class of software embodied, but these details manifested differently depending on the view of teaching and learning mathematics each teacher brought with them. Andrea made adjustments in her clusters of the features of the software after the course and expanded the content. Brittany remained firm with her original clustering of the software features, but expanded the content of each cluster and added some explanatory detail. Participants' electronic communications, heuristic answers, and reflections showed they were able to determine implications for the features of the tools. The repertory grids, which displayed the similarities and differences pre and post-instruction, were the strongest indicators of this growth.

The heuristic technique for software evaluation worked well. Specifically, the questions provided insight into the ways in which preservice teachers viewed the use of the different categories of software to teach mathematics. The heuristic questions, however, only work to support the preservice teachers' preconceived notions of learning as evidenced by their electronic and heuristic responses and the reflection. Even though the preservice teachers were given social constructivist explanations of the questions as provided by Squires (1997), Squires and Preece (1996) and Pea (1986), they answered in their own terms in relation to their specific values of learning. This highlights the problems associated with time (too little) and experiences when attempting to go beyond feature identification and software classification to change an underlying epistemic system. Although the heuristic questions indicated no changes in philosophies of learning, they played a very important role in this research. The heuristic questions tied changes in preservice teachers' thinking to their philosophies of learning, which provided insight into why they appreciated one form of software over others. The questions also forced the teachers to make distinctions among software types when discussing their anticipated teaching role. This in turn formed the basis for their deepening of structure and broadening of software categories.

CONCLUSIONS

The major implication for this study is that even with short exposure, there was a great deepening of preservice teachers' knowledge in mathematics software utilization. Smith and Shotsberger (2001) indicate that most preservice teachers are in accord with using technology, but cannot discuss specific uses due to their lack of specific knowledge. This research indicates that even short-term experience can support preservice teachers' ability to concisely describe uses of technology with their future students. The model described by Lederman and Neiss (2000), which emphasized orienting preservice teachers to support their future students as users of technology, was supported through the course structure developed and tested in this study. This configuration allowed preservice teachers to discern the features of the software that emphasizes support of student learning with just a brief exposure.

The software investigations within this curriculum model were valuable, but may be more beneficial if students critically analyze how people learn mathematics. If a discussion and readings into theories of learning took place on a weekly basis, the preservice teachers could have made a richer evaluation of what it means to learn mathematics and how the software supports learning. Although the heuristic questions and electronic communication did touch on this issue, the questions did not offer the depth necessary to fully evaluate the learning of mathematics. When considering the cases, Andrea had a deeper background in mathematics and was exposed to constructivism in other courses; she was exposed to how children learn. Brittany, on the other hand, did not have the same depth of thought in that area. She did not mention theory of learning throughout her coursework as Andrea did. Additionally, Andrea had a strong background in both technology (as a former computer science major) and mathematics (a future mathematics teacher), while Brittany had only a brief exposure to both of these areas as a generalist in education. Their life experiences would account for the discrepancies in their explanations and notions of how children learn mathematics; analysis of learning theories in mathematics could possibly offer changes in thinking and growth in relation to how people learn with these technological tools.

This research provided instructors of preservice teachers with knowledge they can use to implement mathematics-based tools with their students. There is value in exploring the five tools; the tools offer all preservice teachers enrolled in the course a knowledge base of software to use no matter what their learning philosophy. Each preservice teacher has the opportunity to see some value in at least one of the tools when given the chance to find distinctions between them. When limiting exposure to one or two types of tools, the number of preservice teachers reached in the process is also limited. However, exposure to all five could also adversely affect the preservice teachers and their students in relation to how they use tools. The preservice teachers may only use the type of tool they deem acceptable in relation to their view of learning, thereby limiting the tools available to their students. For example, Brittany may only use Review and Practice software with her students, given that she regarded it as her favorite. If she chooses this type of software only, she is greatly limiting her students' ability to learn mathematics with instructional tools. Discussions and explorations into how people learn mathematics in relation to experience and exploration of the tools could alleviate this possible adverse concern.

These categories offer the potential to enhance the instruction of preservice teachers, inservice teachers, and students. Consequently, future investigations should focus on these areas on a larger scale. By performing this study in a middle school setting with a social constructivist mathematics teacher focusing on student learning and growth over a year, a greater knowledge of how these categories affect student learning and development in mathematics would result. Middle school students could investigate mathematical learning within these categories providing insight and feedback into the affordances and constraints as experienced by learners.

ACKNOWLEDGEMENTS

A grant from the United States Department of Education provided partial support for the activities reported in this manuscript (#P336B990064). The opinions expressed are solely those of the authors and do not reflect the opinions of the United States Department of Education.

Contributors

Terri L. Kurz, California State University, Bakersfield 9001 Stockdale Highway Bakersfield, CA 93311; tkurz@csub.edu. James A. Middleton, Arizona State University, PO Box 871011 Tempe, AZ 85287-1011; jimbo@asu.edu.

References

Beail, N. (Ed.). (1985). *Repertory grid technique and personal constructs: Applications in clinical educational settings.* Kent, Australia: Croom Helm.

Blake, S., Tchoshanov, M., Della-Piana, C., Pacheco, A., & Brady, T. (2001) Report on the partnership for excellence in teacher education: An NSF funded project. *Mathematics and Computer Education, 38*(3), 255–63.

Brouwer, P. (1996-1997). Hold on a minute here: What happened to critical thinking in the information age? *Journal of Educational Technology Systems, 25*(2), 189–197.

Carlson, R. & Gooden, J. (1999). Are teacher preparation programs modeling technology use for pre-service teachers? *ERS Spectrum, 17*(3), 11–15.

Connell, M. (1998). Technology in constructivist mathematics classrooms. *Journal of Computers in Mathematics and Science Teaching, 17*(4), 311–338.

Dede, C. (2000). Emerging influences of information technology on school curriculum. *Journal of Curriculum Studies, 32*(2), 281–303.

Drier, H. (2001). Teaching and learning mathematics with interactive spreadsheets. *School Science and Mathematics, 10*(4), 170–179.

Feixas, G. & Alvarez, J. *A manual for the repertory grid* [online]. Available: http://www.terapiacognitiva.net/record/pag/index.htm

Foster, P. (1996). Observational research. In R. Sapsford & V. Jupp (Eds.), *Data collection and analysis* (pp. 57–93). Thousand Oaks, CA: Sage Publications.

Gunter, G. (2001). Making a difference: Using emerging technologies and teaching strategies to restructure an undergraduate technology course for preservice teachers. *Education Media International, 38*(1), 13–20.

Jonassen, D., Howland, J., Moore, J., & Marra, R. (1999). *Learning to solve problems with technology: A constructivist perspective.* Columbus, Ohio: Merrill Prentice Hall.

Jones, A., Scanlon, E., Tosunoglu, E., Morris, E., Ross, S., Butcher, P., & Greenberg, J. (1999). Contexts for evaluation educational software. *Interacting with Computers, 11*(5), 499–516.

Kelly, G. (1955). *The psychology of personal constructs.* Norton: New York

Kurz, T.L., Middleton, J.A. & Yanik, H. (2005). A taxonomy of software for mathematics instruction. *Contemporary Issues in Technology and Teacher Education* [Online serial], *5*(2). Available: http://www.citejournal.org/vol5/iss2/mathematics/article1.cfm

Lajoie, S. (1993). Computing environments as cognitive tools for enhancing learning. In S. Lajoie & S. Derry (Eds.), *Computers as cognitive tools, (v. 1* pp. 261-288). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lathrop, R. & Williams, J. (1987). The reliability of inverse scree tests for cluster analysis. *Educational and Psychological Measurement, 47*(4), 953–959.

Lederman N. & Neiss, M. (2000). Technology for technology's sake or for the improvement of teaching and learning? *School Science and Mathematics, 100*(7), 346–348.

Marx, R., Blumenfeld, P., Krajcik, J. & Soloway, E. (1998). New technologies for teacher professional development. *Teaching and Teacher Education, 14*(1), 33–52.

Oppong N. & Russell, A. (1998). Using combinations of software to enhance pre-service teachers' critical thinking skills. *Mathematics and Computer Education, 32*(1), 37–43.

Pea, R. D. (1986). Cognitive technologies for mathematics education. In A. Schoenfeld (Ed.), *Cognitive science and mathematics education* (pp. 89–122). Hillsdale, NJ: Erlbaum.

Roschelle, J., Pea, R., Hoadley, C., Gordin, D., & Means, B. (2000). Changing how and what children learn in school with computer-based technologies. *The Future of Children*, 10(2), 76–101.

Smith, K. & Shotsberger, P. (2001). Web-based teacher education: Improving communication and professional knowledge in preservice and inservice teacher training. Available [online]. Eric Document #ED459161.

Squires, D. (1997). An heuristic approach to the evaluation of educational multimedia software [online]. Available: http://www.media.uwe.ac.uk/masoud/cal-97/papers/squires.htm

Squires, D. (1999). Usability and educational software design: Special issue of integrating with computers. *Interacting with Computers, 11*(5), 463–466.
Squires, D., & Preece, J. (1996). Usability and learning: Evaluating the potential of educational software. *Computers and Education, 27*(1), 15-22.