

PRESERVICE TEACHERS CONCEPTIONS OF MATHEMATICS-BASED SOFTWARE

Terri Kurz
Arizona State University
U.S.A.

James Middleton
Arizona State University
U.S.A.

H. Bahadir Yanik
Arizona State University
U.S.A.

Preservice teachers do not feel prepared to teach mathematics using technology (Smith & Shotsberger, 2001). To address this issue, we have developed a tool-based categorization of mathematics software and used it to instruct preservice teachers. Changes in thinking that occurred as a result of the course are analyzed primarily using repertory grid techniques supported by heuristic questions, reflections and Internet communication group correspondence.

INTRODUCTION

Technology has tremendous potential for enhancing mathematics instruction; it can be used to strengthen student learning and to assist in developing mathematical concepts. Technology 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 recent 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 were uncomfortable discussing the specific uses of technology for instruction due to lack of knowledge. Many preservice teachers feel that they are not prepared to teach using technology after they graduate (Carlson & Gooden, 1999). The question that then develops is what kinds of experiences preservice teachers should have with regards to the integration of technology and mathematics.

An effective way to prepare preservice teachers to use technology in mathematics is to prepare them to utilize technology for student use as a tool. 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, 1997, p. 190). For example, to solve multi-step algebraic equations, a pencil is a tool that is beneficial in assisting with solving the equation. Use of the pencil allows for the problem to progress, providing a record and visualization of the process. Some technological tools in mathematics are computer programs and software, calculators, and languages (like Logo) (Connell, 1998).

Lajoie (1993) describes the many benefits of using the computer as a tool for instruction in an educational setting. First, technological tools help to support the cognitive processes by reducing the memory load of a student and by encouraging

awareness of the problem-solving process. Second, the tools can share the cognitive load by reducing the time that students have to spend on computation. Third, the tools allow students to engage in mathematics that would otherwise be out of their reach, stretching the students' opportunities. And fourth, the technology tools support hypothesis testing by allowing students to easily test conjectures.

We have developed five general categories of software examined with the idea of tool-based use in the mathematics classroom. 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.

Review and Practice Tool

When software is used for reinforcement of previously learned material, the software falls into this category. The software is simply used to drill the student on a specific area of mathematics. The student does the same type of mathematics problems in a repeated manner. No new conceptual material is introduced. With review and practice software, the program controls materials, tasks and feedback in a highly directed manner.

Review and practice software is usually designed to be used individually with little teacher intervention. Students solve problems, asking for assistance from peers or the teacher when questions occur. An example of software in this category is *Pre-Algebra Math Blaster Mystery*. This program is designed to reinforce skills already learned in pre-algebra. There is an emphasis on computation, estimation, proportions, ratios and percents.

General Tool

When software is designed for use across a variety of mathematical domains, the software falls under this category. General software is designed for 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. General software can be used for a wide range of grade levels and mathematical subjects.

Geometer's Sketchpad is an example of software designed for general use. It is a dynamic geometry computer program that has gained respect for its potential at assisting students with the possibility of testing conjectures, emphasizing critical thinking and problem solving, through active manipulation of graphical objects and procedural scripts.

Specific Tool

Software designed to emphasize learning in a particular area of mathematics is an example of specific software. The focus with specific software is the learning of a distinct mathematical topic, such as fractions, reflections, order of operations, etc. This differs from the review and practice category in that the focus is on learning new

content, not reviewing a specific mathematical concept. *TesselMania* is an example of specific software.

TesselMania, while not a heavily researched application, has promise as a technology that emphasizes transformational geometry, specifically concepts of rotation, translation, and glide reflection. Because it is complex, including combinations of these concepts in the tessellation process, the program can support critical thinking and deep conceptual understanding.

Environments Tool

Software used as an environments tool integrates different types of learning in a variety of subject areas. The software provides an environment that is not normally possible in the classroom, and students make investigations into the given setting. Environments software provide a virtual place for students to guide their mathematical learning, taking students to a new place without requiring them to leave the classroom. Allowing student investigations into problem solving based on mathematical inquiry, sometimes the software is designed for cooperative investigation. The teacher does not present the software to the student rather, the teacher acts as a facilitator, assisting the students as requested and posing inquisitive questions or comments to keep students on track or to clarify.

The Jasper Project (Cognition and Technology Group at Vanderbilt, 1997) is an example of software that can be utilized as an environments tool. The Jasper Project is designed for use with students in grades 5 and up; there are a total of twelve scenarios utilizing real world examples, with an emphasis on either statistics and probability, distance/rate/time, geometry or algebra (CTGV, 1997). The videos include problem-solving environments that promote mathematical thinking through the scaffolding design of the software (Nicaise, 1997).

Communication Tool

Communication software is software that 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 understanding of mathematical concepts and ability to articulate mathematical arguments and concepts through discourse.

Groupware, videoconferencing, chats, electronic bulletin boards, e-mail and listserves are examples of software that has been developed for use as a communication tool (Jonassen, Howland, Moore & Marra, 1999). Students have an opportunity to go back and look at discussions that took place previously, and reformulate their thoughts if necessary. Groupware, videoconferencing, chats, electronic bulletin boards, e-mail and listserves are examples of software that have been developed for use as a communication tool (Jonassen, Howland, Moore & Marra, 1999).

Tool-based software has been an area of focus in mathematics education for some time. However, there has not been research in the 5 major categories of tools

described in this paper. The focus of this paper is to examine changes in thinking, particularly the understanding of the affordances and constraints embodied by the 5 classes of software, by preservice teachers. We expected that examination of these affordances and constraints through experience with, and analysis of, exemplars of the 5 classes of tools would have an affect on the thinking of preservice teachers. In short, this research attempts to identify *how* their thinking will alter as a result of this exposure to be more specific regarding the fit of tool to task, and broader in conceptualization of the kinds of software appropriate for integration into mathematics instruction.

METHOD

Two case studies were investigated: one a preservice mathematics teacher focusing on secondary education, the other a preservice teacher focusing on primary education. Utilizing the methods of Personal Constructs Theory (Kelly, 1955), pre-repitory grids and post-repitory grids were administered to each preservice teacher, classroom observations and transcripts were recorded, and heuristic questions were administered and analyzed in threaded electronic conversations.

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 a 6 weeks, 2 hours per session. Examples from one class of software (see above) was presented each week, with the exception of the communication software which was used throughout the course. After each software experience, students participated in a communication group and completed heuristic evaluations of the applications in an online threaded discussion (Squires, 1997). In each of these discussions, we seeded the conversations by asking students to make distinctions among the different applications they had experienced, regarding the constraints and affordances each offered to mathematics instruction.

To assess change in conceptualization of software tools, we analyzed pre- and post-repitory grids, which asked participants to compare different software they had experienced, and to proffer concepts that distinguished one from another in teaching mathematics. By examining change in both depth of constructs and their organization cognitively, it is possible to determine quantitatively, changes in teachers' conceptualizations due to the intervention of the course. Ward's Method of cluster analysis was applied to constructs in the repitory 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 versus error clusters (Lathrop & Williams, 1987). Transcripts of observations, reflection responses, communication group discussions and heuristic responses, were used to contextualize and describe the point in which changes in thinking took place in the course. These analyses assisted in the creation of an overall model of how thinking changed in regards the use of technology in the mathematics classroom for the two preservice teachers studied.

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 5 general classes of mathematics tools presented above. 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. We provide a brief discussion of one participant's constructs organization as an illustration:

Software Categorization

This preservice teacher, Andrea, classified the software she had had experience with in her own instruction into 2 distinct categories at the beginning of the course (Figure 1). She divided productivity software such as PowerPoint and word processing programs into a category and the rest into an "other" category, those softwares that were not solely productivity-oriented. The similarities and distinctions of the non-

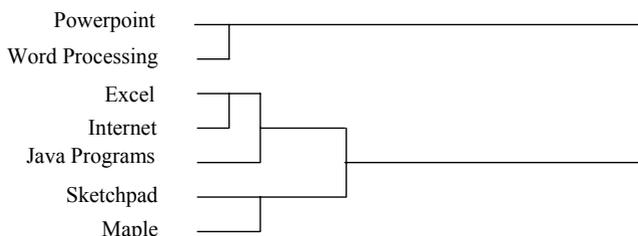


Figure 1: Dendrogram of Software Categories, Pre-Instruction

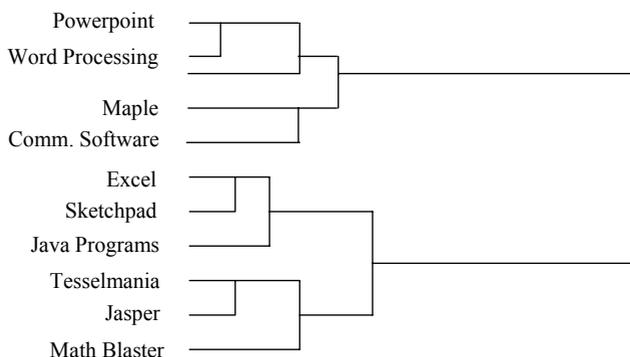


Figure 2: Dendrogram of Software Categories, Post-Instruction

presentation types of software are not clearly apparent to the student in this model.

After the completion of the course, the student has made clear distinctions between the types of software mirroring the similarities they have in supporting student learning (Figure 2). She maintains a category that relates to teacher-directed presentation types of activities. However, the second category is more oriented toward tool-based use of software to aid the general mathematical growth of students through the use of open-ended software exploration. The third category pertains to software used to assist the student in a specific mathematical area; it includes software that is supportive of review and practice techniques along with environments and specific softwares.

Constructs Development

After just a brief exposure to the software, this student was able to develop more defined clusters, increasing her constructs from 10 to 27 (Table 1). The constructs also more general in the beginning, referring to what the software can do (for example, add, subtract, make along list of numbers in seconds; you can solve equations and do different kinds of graphs). After the course, the constructs focused more on the support the software offers to student learning and mathematical development (for example, students can explore and discover results; makes students really think about the problem and the answer) including the use of software as a tool to aid in student learning (general tool used to learn many topics).

The elementary teacher had similar results. Her first comparison of software types included two categories: a review and practice category and a category of “others” also. After the course, she developed 4 categories: review and practice, communication software, a general category and a specific category. Her constructs increased from 13 to 35. She created 3 distinct clusters in the beginning, and after the course she still had 3 clusters, expanding the clusters with more defined and developed constructs.

DISCUSSION

Previous research indicates that the majority of preservice teachers believe that technology is important for its own sake (Quinn, 1998). However, university preparation in the field of technology use in the classroom is inadequate (Gunter, 2001; Smith and Shotsberger, 2001). When they graduate, preservice teachers are not ready to teach mathematics with technology, nor are they aware of the possibilities in learning the software can support.

Preservice teachers are ready to learn about the use of technology for use in the mathematics classroom. Exposure to the 5 tool-based categories along with follow up analysis in the form of heuristic questions, communication questions and reflections offer preservice teachers the opportunity to develop a stronger foundation of mathematics-based software knowledge. This base provides the preservice teacher

Clusters	Pre-Instruction Cluster Constructs	Post-Instruction 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; Students don't learn much (negative); Used for solving one problem
Cluster 2- Emphasis on 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; Makes students really think about the problem and answer; Students think at higher levels; Provides different data representations so you have a better chance of reaching all the students; More specific topic; It is a specific tool to help students learn; 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; 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)

Table 1: Cluster Membership of Mathematics Software Constructs

with new knowledge regarding the specific features of software that enable students to learn mathematics, and the fit of those features to particular goals of classroom instruction.

The findings of this study suggest that exposure of the 5 categories of mathematics-based software can lead to positive conceptual change. The thoughts of the preservice teachers became more developed and comprehensive after experiencing and reflecting on the affordances and constraints of tool-based mathematics software. This suggests that with time to experience and reflect, preservice teachers can alter their thoughts concerning the categories and features of software for use in the mathematics classroom.

References:

- Brouwer, P. (1996). 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.
- Cognition and Technology Group at Vanderbilt, (1997). *The Jasper project: Lessons in curriculum, instruction assessment and professional development*. Mahwah, NJ: Lawrence Erlbaum Associates.
- 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.
- Gunter, G. (2001). Making a difference: Using emerging technologies and teaching strategies to restructure an undergraduate technology course for pre-service 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.
- Kelly, G. (1955). *The Psychology of Personal Constructs*. Norton: New York
- Lajoie, S. (1993). Computing environments as cognitive tools for enhancing learning. In S. Lajoie & S. Derry (Eds.), *Computers as cognitive tools*, v. 1. 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.
- Nicaise, M. (1997). Computer-supported apprenticeships in math and science. *Journal of Computers in Mathematics and Science Teaching*, 16, 4, 443-465.
- Quinn, R. (1998) Technology: Preservice teachers' beliefs and the influence of a mathematics methods course. *The Clearing House* 71 (6), 375-377.
- Smith, K. & Shotsberger, P. (2001). Web-based teacher education: Improving communication and professional knowledge in preservice and inservice teacher training. Available [on-line]. Eric Document #ED459161.
- Squires, D. (1997). An heuristic approach to the evaluation of educational multimedia software [on-line]. Available: <http://www.media.uwe.ac.uk/masoud/cal-97/papers/squires.htm>