INTEGRATING SOFTWARE INTO MATHEMATICS EDUCATION

By

CAROLINE GRACE OBLACK

A THESIS

Presented to the College of Education
and the Honors College of the University of Oregon
in partial fulfillment of the requirements
for the degree of Bachelor of Science.

June 2000

Approved: [redacted]

Professor Mary Gleason, College of Education

An Abstract of the Thesis of

Caroline Grace Oblack for the degree of Bachelor of Science

in the College of Education to be taken June 2000

Title: INTEGRATING SOFTWARE INTO MATHEMATICS EDUCATION

As we move into the 21st century, a new educational curriculum will be needed in order to meet the needs of the future generation. Integrating technology throughout the curriculum will enable students to learn many skills effectively and more efficiently, especially those in the area of mathematics. In order to meet the national and state standards, students will need to understand how to use various types of technology from calculators to problem-solving simulations. Therefore, teachers need to understand the importance of integrating technology into their classrooms as well as how to utilize these tools throughout the existing curriculum.

Educational software is one of the easiest forms of technology to integrate into a curriculum. This paper will describe the importance of integrating educational software into an existing mathematics curriculum, how to select and evaluate different types of software, and how to integrate these new tools throughout the curriculum.

ACKNOWLEDGMENTS

The author expresses sincere appreciation to Professors Gleason and Moursund for their assistance in the preparation of this manuscript. In addition, special thanks are due to Professor Yoder, who assisted in developing the idea for this thesis due to her experience in technology and education. I also want to thank my husband and my parents for all there support throughout the writing process. Without them, I would never have been able to complete this task.

TABLE OF CONTENTS

Integrating Computer Software into the Teaching Curriculum

Computers were once only used by the elite, but now it is common to see computers in offices, households and schools around the nation. In many schools, computers and software have become useful tools in the classroom, but most classrooms have not yet completely integrated them into the curriculum.

What is considered integration? According to Dias (1999), integration of technology happens "when it is used in a seamless manner to support and extend curriculum objectives and to engage students in meaningful learning" (p. 11). This definition states that technology cannot be just a tool in the classroom where it receives little or occasional use. Computer software should be used regularly in the classroom to further exploration into existing curriculum.

Due to the limited use of technology that I have seen in the classrooms I have visited, I decided to find out why teachers may not be utilizing these resources. Research indicates that many teachers who have never used the computer have fears of breaking the mold that they are used to working in (Bradley & Russell, 1997), and therefore have not pursued integrating computer software into their curriculum. Others believe that integrating software, which takes time, may be a difficult feat to accomplish.

From my own observations, educational software is one of the easiest forms of technology to integrate into a curriculum. Educational software usually comes with a parent or teacher's manual that describes its content and how it works. However, in most cases it is up to the teacher to develop lessons and objectives for using this software in the classroom.

Integrating Software into the Mathematics Curriculum

Unlike the subjects of English and grammar, mathematics is an ever changing field. New discoveries in mathematics are being found every day, and many of these discoveries are being found with the assistance of computers.

Computers and software are changing math education by providing students the chance to learn many important mathematics skills better and faster than ever before (Moursund, 1986). Research indicates that students who receive computer-based instruction in their classes (Kulik, 1994):

- learn more than in classes without computer-based instruction
- learn information in less time
- enjoy the classes more
- develop more positive attitudes toward computers

Many teachers use drill and practice programs on the computer to "integrate" technology into the curriculum, but there are many more ways to utilize the use of various software programs to enhance not only basic skills, but problem solving and higher order thinking skills.

Many mathematical situations that people encounter in real life require them to use problem solving. Problem solving is "the process of searching for and finding solutions of problems, " (Wiebe, 1993, p. 13). In the past, teachers believed that if the students were taught how to compute, they would be able to also solve problems. However, James Wiebe believes "that having children do pages of computational problems mostly prepares them to do pages of computational problems, not necessarily to solve real-life mathematics problems" (p. 12). Since 1980, the National Council of Teachers of Mathematics have recommended that students are moved away from paper and pencil tasks toward developing problem-solving techniques and the ability to apply these techniques to real-life situations.

The question remains, how can software be used to assist students with learning mathematical skills ranging from computational to problem-solving in the classroom. The following ideas are ways for students to practice various mathematical skills using the available software programs and technology (Drier, Dawson, & Garofalo, 1999):

- Transport data from the Internet, other software, or other class data into spreadsheets, multimedia presentations, or view it from a projection screen.
- Make graphical representations of data.
- Make predictions and inferences using statistics.
- Simulate real-world events, such as population growth, or investment potential.

Some other ideas are using problem solving software to help enhance the students' problem solving skills and equation-writing software to enable students to see the values of various equations and view the graphical representations in a quick and simple format.

Drill and practice programs are somewhat useful, but utilizing software for these other parts of the curriculum will help the students reach a deeper understanding and appreciation of mathematics.

Software Selection

Choosing the correct software can be intimidating unless one knows what to look for. The first thing a teacher should ask is what does he want the technology to help him do. Does he want it to provide drill and practice, help students problem solve, or just graph some data that the students had collected?

Some other factors teachers should consider are quality, number of copies needed, source of best prices, and matching the software to curriculum goals (Roblyer &

Edwards, 1999). To find software that contains quality programming, teachers can read reviews and find recommendations, pilot the programs in their classrooms before buying it, and review the supporting materials on how the program operates. By doing one or a combination of these things, teachers can gain the knowledge as to whether the program is right for their students. Teachers should shop around for the best prices. Most software providers provide a discount with purchases of multiple copies, site licenses, or networked versions. In addition, teachers should compare programs that do somewhat the same thing to make sure that the program targets the correct age levels and the specific curriculum needs.

Another way to review software packages is to visit school districts that have already begun to implement computer technology into their schools. During these visitations, teachers can take some time and view the software packages that those schools are using and may be unfamiliar with. Teachers can also visit with the media or technology specialist on site to see which software programs align the best with the state and national standards, the curriculum, and how well the students' progress can be recorded and assessed (Recesso & Carll, 1999).

Hardware Selection

Recesso and Carll (1999) recommend that the software be chosen before the hardware, but in many cases this situation is not feasible. Many schools already have hardware whether it was purchased or donated. Before software is considered, therefore knowing the hardware's capabilities is important when choosing software. The software's requirements are usually listed on the packaging, and are usually available in both PC and Macintosh. Other concerns about hardware are (Recesso & Carll, 1999):

- Wiring, will the existing wiring support the new technology?
- Furniture, is there enough furniture and is it the appropriate size for those that are going to be using it?
- Is there enough space for printers?
- Can the computers be placed away from water, heaters, chalkboards, windows, etc.?
- Is the current networking system large enough or will it need to be expanded?
- Who will be supporting the hardware and software chosen?
- What trained personnel resources are available to provide on-site maintenance?
- What will be the initial cost?
- What costs will incur in order to maintain each workstation?

If these concerns are addressed during the hardware and software selection, then implementation will be more efficient and effective.

Software Evaluation

Most educational software products are hypermedia and/or multimedia products. Multimedia software incorporates text, sound, graphics, and/or video, whereas hypermedia software connects elements of a computer document (e.g. text, movies, pictures, and other graphics) through keywords (Roblyer & Edwards, 2000). For this reason, most educational software can be evaluated on its design using Koehler and Lehrer's (1998) design principles for hypermedia documents, except for the guideline referring to non-linearity, which does not apply to all multimedia software.

Make the Structure of the Domain Visible

By making domain or strand of knowledge obvious, the learner is more likely to achieve the objectives of the software. The structure of the links should be explicit and therefore help the learner make connections and find relationships within this strand of knowledge. Koehler and Lehrer (1998) state that links should be typed so that the learners can "infer the reason for connections" (p. 125), but for younger learners, icons or graphics might be a better choice.

Make Navigation Easy

When software does not follow a linear pattern, the learners are not required to follow a set learning sequence. If strong navigation tools are not in place, a learner can easily become lost and confused. In order to reduce this anxiety, software should follow some simple rules that the learner already knows. These rules can be found in most books. Though books tend to be linear, they still have important properties that can help the learner navigate through the book, such as a table of contents and an index. The navigation tools in software can help the learner weave through the different areas in the software program with ease and comfort. If he or she gets lost, these tools can help them find where they have been and where they want to go.

Provide Learners Opportunities to Learn by Example

"Examples are particularly important when properties to be learned cannot be explicitly stated," (Koehler & Lehrer, 1998, p. 126). When working with elementary children, this is especially true. Educational software should have a place to practice the new skills or demonstrations built into the program so that the learner is able to get the most information out of the experience.

Layer Annotations and Examples

An example of layering information is highlighting a word in a body of text while having the definition on the same page instead of on a separate one or by placing a link to a video within the text. By doing this, the learner is able to focus on the connection of the highlighted word or video instead of focusing their attention on finding the source on another page.

Lessons Learned

Educational software should offer positive feedback throughout the entire process. The software should be focused on helping the student learn and not punish them for making a mistake. For example, if a student is using a program that helps them practice their multiplication facts, the student should receive a congratulatory message if they get the correct answer. If the student should happen to make an error, the software should offer words of encouragement and help if needed. However, the software should not just give the correct answer when an error is made. Students need to be able to correct their own mistakes so that the maximum amount of learning can be achieved.

Uses Non-Linearity

Educational software should be designed in a web-like structure. A web-like structure just means that the program is not set up in a linear fashion. When reading a book, one page goes to the next and there is a required sequence to which the student must follow for the book to make sense. In a hypermedia document, the student gets to choose where they would like to go next within the program. One step is not dependent on the other. By using this technique, learners are able to look at the content from different perspectives, while still having the freedom of choice.

Moving Toward Integrating Software into the Curriculum

Teachers go through five stages when learning to integrate computers and software into the existing curriculum (Sandholtz, Ringstaff, & Dwyer, 1997). In order for teachers to make progress through these stages, teacher fears must also be addressed.

Teacher Fears of Technology and Software

Norma Carr-Ruffino (1985) identifies three main fears women have when they set goals: the fear of success, the fear of failure, and the fear of risk-taking. She explains that "becoming aware of your fears, is often a first step to overcoming" obstacles (p. 45). Though this book was written for and about women, men also can have these same fears, especially when trying something new.

According to Bradley and Russell (1997), the most common fears that teachers have about using technology include things such as "not knowing where to start" (p. 271), not understanding computer jargon, breaking the computer, looking silly, having students observe their ineptness, and the feeling of being overwhelmed. Bradley and Russell explain that these anxieties can be avoided if the teachers feel that they are being supported and encouraged to integrate technology and software into the curriculum. Without support and encouragement from their administrators and peers, the teachers fears of using computers is reinforced, therefore they are not able to perfect their computer skills as technology advances.

A close relationship between administration and teaching staff must be in place to enhance the use of technology, especially software, in the classroom. When teachers' superiors and peers are willing to support them in enhancing their technological skills and encourage them to use this newly found knowledge in the classroom, the teachers' level of anxiety can be greatly reduced. Therefore, teachers are more likely to experiment and discover new methods for teaching a skill or idea while also integrating technology and software into the curriculum.

Stages Teachers Go Through in Learning to Integrate Technology

ACOT Project

In 1985, Apple Computer began a project entitled the Apple Classrooms of Tomorrow (ACOT) project in order to enhance the learning process in technology. The ACOT project was set up to "investigate how routine use of technology by teachers and students would affect teaching and learning" (Sandholtz et al, p. 3). The goals of ACOT were to:

- install and operate computer-saturated classrooms as living laboratories in every grade (K-12)
- integrate state-of-the-art technologies into the instructional fabric of schooling
- bring about positive educational development and change
- study and understand the impact of total computer access on students, teachers, and instructional processes

The project began working in five schools in four states, but in following years expanded to include more. Each school district selected the volunteer teachers to participate, most having had few experiences with computers. ACOT equipped each classroom with computers, printers, modems, CD-ROM drives, software, etc. They also provided each teacher and student with two computers, one for home and one for school. The project teachers were trained on basic troubleshooting, telecommunications, and tool software such as spreadsheets, and if a more serious problem needed to be addressed, a coordinator based at each site would be there to help (Sandholtz et al., 1997).

Stages of Instructional Evolution

During the ACOT project, patterns of teaching and learning emerged over time to include the following stages: entry, adoption, adaptation, appropriation, and invention (Sandholtz et al., 1997).

Entry

The entry stage in thc ACOT program is considered the time where the teachers had very little experience with the technology were therefore unable to anticipate problems whether it be student misbehavior, the classroom environment, or software management. The project teachers during this stage had to start developing not only technological skills, but also new ways to manage behavior.

Adoption

Project teachers reached the adoption stage when they began to predict possible problems and therefore were able to precorrect the anticipated problems with prompts and other strategies. Teachers at this stage were also able to implement meaningful consequences for students who were not able to follow the rules. An example of this was implementing a group restriction of the computers if someone was caught illegally copying someone else's work.

Also during this stage, the technology in the classroom began to become more of an aid than a burden. With this experience, teachers "became more adept at avoiding problems, and, when problems did occur, narrowing down the source of the difficulty" (Sandholtz et al., 1997, p. 64).

Adaptation

During the adaptation stage, teachers began to use the computers and technology to their advantage. Not only would they use the computers to help manage behavior, they used it to monitor the work of students, to individualize instruction, and to improve record keeping. By using the computer to keep track of students' progress, it freed teachers time to focus on other things.

Appropriation

Sandholtz et al. (1997) describe appropriation not as a phase, but more as a change in attitude. "Appropriation is the point at which an individual comes to understand technology and use it effortlessly as a tool to accomplish real work," (p. 42). During this stage, teachers reported feeling more confident with their use of the computers and that it had "become a way of life," (p. 43). This stage leads to the last stage of invention.

Invention

When project teachers reached this last stage, they began experimenting with new instructional ideas such as interdisciplinary project-based instruction and team teaching. Students were able to explore and learn on their own. They began to work with each other to solve problems and achieve higher learning. ACOT reported that the most important change at this stage was that teachers were reflecting and questioning their old strategies of teaching and discovering why changes in their students had occurred.

During the ACOT project, most, but not all teachers reached this stage. Those that did viewed "learning as an active, creative, and socially interactive process," (Sandholtz et al., 1997, p. 47). Students had to construct knowledge for themselves rather than transfer it from the teacher.

Purposes for Integration

According to Roblyer and Edwards (2000), each strategy for integration can fall into a "group of general integration principles" (pp. 69-71). Some teachers use technology to meet their students learning needs. Others use the technology available to substitute for materials that may be lacking in their skills. Teachers can use any of the following strategies to meet specific needs in their classroom that they might not have been able to meet otherwise.

Integration Strategies Based on Teacher- Directed Models

Integration to remedy identified weaknesses

Students who have not learned the prerequisite skills are at a disadvantage when the rest of the class is ready to move on. For example, a student has not learned how to add two numbers together. The rest of the class is ready to try moving on to subtraction of small numbers; but without the prerequisite skill of addition, this student may be left behind. In this instance, tutorials and drill and practice software can be integrated into this student's curriculum to help him or her to succeed and move on to subtraction.

Integration to promote fluency or automaticity of prerequisite skills

Software programs such as drill and practice programs and tutorials can assist students in creating automaticity to their work. Basic math facts have to be memorized, but this process can take a lot of time and practice both in and out of the classroom. Drill and practice programs can help by providing a fun environment for the student to practice their facts in and help them achieve quicker success with memorizing their basic facts.

Integration to make learning efficient for highly motivated students

Motivation can come from within (intrinsic) or from an external source (extrinsic) such as the need to please parents. Many present educational methods are "criticized for failure to interest and motivate students" (Roblyer & Edwards, 2000, p. 69), because they do not address the needs and interests of the students. Therefore students that are extrinsically motivated may stay that way unless teachers can find a way to tap into those students' interests and encouraging them to learn more on their own. Technology, especially tutorial and simulation software, can help these students become intrinsically motivated about topics they enjoy.

Integration to optimize scarce materials and other resources

In many public schools, many resources are scarce, whether it be the right color of construction paper or not having enough teachers. Educational software can save paper and supplies by providing computer-based drill and practice instead of worksheets, tutorials in topics that ordinarily would require many supplies, and simulations instead of hands-on materials. For example, these computer-based activities help students explore problem solving and repeat experiments without needing physical manipulatives or expensive chemicals.

Integration to remove sequential boundaries

Many software programs have no set sequence. They are set up to let students explore areas in whatever order they are interested in. Educational software can assist students by letting them build and rebuild to create a finished product. For example, computer-assisted design (CAD) software can help students build a house within the computer environment and make changes without having to start completely from the beginning.

Integration Strategies Based on Constructivist Models

Integration to generate motivation to learn

Teachers whose students are considered at-risk may have a higher need with motivating their students to learn. The key to success with these students is discovering what interests and enthuses them. By integrating multimedia software into the curriculum, teachers can help students to develop their ideas and publish their findings in an environment that includes different types of media.

Integration to create and maintain creativity

Creativity is not the primary goal of education, but it is a highly desired trait. A person who is not creative can be educated, but most schools do not want to graduate students who are not able to think creatively. Software programs such as Logo, problem solving programs, and graphics programs can provide students with a non-threatening environment to explore their creativity.

Integration to enhance metacognition

Metacognition is the process a person uses to think about their own thinking. On many occasions, students do not think about why they are following a set of procedures, and therefore have difficulty generalizing that information to other situations. With the aid of software, such as problem solving programs and multimedia, teachers can create a better awareness about why a set of procedures work for a certain type of problem or situation. Logo is a great example of this because students can create a set of instructions about what they would like the turtle to draw or follow. Then when they test it, the students can visually see which part of the procedures worked and which ones did not. Therefore, this program helps students think about how they think and help them to become better problem solvers.

Integration to enhance transferring knowledge into problem solving

Many students when first learning how to solve problems have difficulty transferring the skills they have learned previously to solving the problem at hand. This is mainly because the skills taught are usually taught out of the context of problem solving. Many software programs available help students use various skills in a more visual way. These visual presentations enable students to transfer the knowledge needed into the problem solving process. Therefore students are creating mental models and learning to perform higher order skills.

Integration to enhance cooperative groups and provide opportunities for using multiple intelligences and ability grouping

Cooperative grouping in classrooms are becoming the norm. These activities help students of differing abilities share their talents with others in their class. It's true that teachers can provide cooperative activities without technology, but students enjoy working with creating group projects in programs such as Hyperstudio and slide show programs.

Integration to develop literacy and mastery in technological and visual skills

Integrating technology and software into the classroom to develop literacy in technology and visual skills prepares students for college classrooms and business offices. University professors and business professionals use software programs such as Microsoft PowerPoint and slide show programs to present information to their students or colleagues. Elementary teachers should use multimedia programs to help students create more visually complex book reports or experiment results.

Helpful Hints and Suggested Guidelines for Integrating Technology

After a teacher decides why they are integrating technology and software into their curriculum, they may need to know how to do it. Roblyer and Edwards (2000) have come up with a some guidelines that each teacher should follow when trying to integrate technology:

- Plan for a unit of instruction
- Be flexible with lesson plans and time to complete
- Match the activity, objectives, and what will be assessed
- Experiment and don't be afraid to make mistakes

When integrating technology, teachers need to plan on activities taking a little longer than expected, as well as being prepared for anything. Therefore the unit of instruction should cover a grading period such as a semester or even an entire school year. This will allow students to experiment, learn new skills and go into more depth with how certain software programs work and what their best uses are.

As most teachers know, time is a precious commodity. Therefore teachers need to plan for extra time in the event that the activity takes a little longer than expected. By being flexible, teachers will allow for more student-directed learning and the ability to follow-up with students' interests.

The last guideline is possibly the most important. As seen with the ACOT teachers, success is not always assured the first time, nor the second time. By experimenting teachers can learn from their mistakes and develop methods that are more efficient and effective.

Other effective integration strategies presented by Roblyer and Edwards (2000) include:

- Provide explanations about the task if students misunderstand what to do.
- Use the advanced organizers that accompany most software packages.
- Assess students as they learn to help individualize the content as necessary.
- Have students verbalize their thought processes.
- Challenge students to participate in more abstract discussions.
- Assist the groups by looking at their current problem-solving techniques and help them learn more efficient and effective ones.
- Provide training about working in groups before assigning projects involving multimedia projects.

Conclusion

As we enter the 21st century, a new curriculum must be created in order to meet the needs of the future generation. It will be necessary to integrate technology and software throughout this new curriculum in order to meet the national and state standards. In order for this integration to work, teachers must be supported by their administrators and peers and be willing to try new things. Training sessions will need to be provided for teachers to keep learning new techniques involving technology, as well as build on prior learning.

To integrate software into mathematics, teachers need to not only use the computers for drill and practice, but utilize the various software packages that are available for problem solving and other higher order thinking skills. Students need to be able to apply their computational skills to solve problems in real-life situations.

The future will rely heavily on technology. By providing the students of today with practice and providing them with new skills using the various software programs available, teachers will give their students a head start into the technological future which lies ahead.

References

Bradley, G., & Russell, G. (1997). Computer experience, school support, and computer anxieties. Educational Psychology, 17(3), 267-284.

Carr-Ruffino, N. (1985). The promotable woman: Becoming a successful manager. Belmont: Wadsworth Publishing Co.

Dias, L. B. (1999). Integrating technology. Learning and Leading with Technology, 27(3), 11-13, 21.

Drier, H. S., Dawson, K. M., & Garofalo, J. (1999). Not your typical math class. Educational Leadership, February, 21-25.

Koehler, M. J., & Lehrer, R. (1998). Designing a hypermedia tool for learning about children's mathematical cognition. Journal of Educational Computer Research, 18(2), 123-145.

Kulik, J. A. (1994). Meta-analytic studies of findings on computer-based instruction. In E. L. Baker, & H. F. O'Neil, Jr. (Eds.), Technology Assessment (pp. 9-33). Hillsdale: Lawrence Erlbaum Associates, Inc.

Moursund, D. (1988). Computers and problem solving: A workshop for teachers. Eugene: International Society for Technology in Education.

Recesso A., & Carll, J. (1999). Integrating technology into the K-12 educational setting. Educational Media and Technology Yearbook, 24, 2-10.

Roblyer, M.D., & Edwards, J. (2000). Integrating educational technology into teaching. Upper Saddle River: Merrill.

Sandholtz, J. H., Ringstaff, C., & Dwyer, D. C. (1997). Teaching with technology: Creating student-centered classrooms. New York: Teachers College Press.

Wiebe, J. (1993). Computer tools and problem solving in mathematics. Wilsonville: Franklin, Beedle & Associates, Inc.