DOCUMENT RESUME

ED 355 922 IR 016 018

AUTHOR	Overbaugh, Richard C.
TITLE	Critical Elements of Computer Literacy for Teachers.
PUB DATE	Mar 93
NOTE	25p.; Paper presented at the Annual Meeting of the National Society of Educators and Scholars (Evansville, IN, March, 1993).
PUB TYPE	Viewpoints (Opinion/Position Papers, Essays, etc.) (120) -- Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Computer Assisted Instruction; *Computer Literacy; Computer Managed Instruction; Computer Software; Definitions; *Educational Innovation; *Elementary School Teachers; Higher Education; Models; Music Education; Programing Languages; *Secondary School Teachers; Skill Development; *Teacher Education

ABSTRACT

A definition of computer literacy is developed that is broad enough to apply to educators in general, but which leaves room for specificity for particular situations and content areas. The following general domains that comprise computer literacy for all educators are addressed: (1) general computer operations; (2) software, including computer managed instruction, computer assisted instruction, and programing languages; (3) software issues; and (4) hardware. Bearing these in mind, three models for developing teacher computer literacy programs are discussed. The two-stage model suggests that the path to computer literacy should begin with utilization of computer management skills and then proceed to learning applications that teach content. A second model offers four stages through which new users of innovation pass and suggests that 2 to 3 years are necessary for new users to pass through these stages. A seven-stage model of introduction to innovation expands the number of concern levels in gaining computer literacy. The four domains of computer literacy are explored for specificity by applying them to music education as an example of an educational computing knowledge base. It is noted that, unless higher education integrates computers across the curricula, tomorrow's teachers will not be adequately prepared to teach students in a technologicallv intensive work environment. (Contains 35 references.) (SLD)

Critical Elements of Computer Literacy for Teachers

By

Richard C. Overbaugh
School of Education and Human Services
University of Southern Indiana
8600 University Boulevard
Evansville, Indiana 47712

Presented at the National Society of Educators and Scholars Conference

Evansville, Indiana
March, 1993

Introduction

The relatively recent infusion of technology has been welcomed by society from the workplace to entertainment and, since the public school system has long been accepted as a reflection of the society in which we live, schools are responsible for preparing students for effective and responsible lives in that society (Harrington, 1991). Therefore, schools are responsible for preparing technologically proficient, computer-using citizens. In order to fulfill this responsibility comes the need for adequate equipment and sufficiently knowledgeable teachers to thoroughly integrate computers into their curricula. Unfortunately, schools typically do not have the necessary resources to meet their needs and technology is no exception. However, progress is being made; Hasselbring (1991) estimates that the percentage of schools with at least one computer dedicated to instruction has increased from approximately 18 percent in 1981 to 95% in 1987 and that there are between 1.5 million and 2.1 million computers in the schools which equals about one computer for every 30 students which, although significant, is woefully inadequate. To illustrate typical high school students' lack of exposure to computers, a recent study involving over 500 undergraduate students entering a teacher-education program at a mid-western land grant university, showed that fully one-third had no computer experience at all and another one-fifth had experience only with content-area applications (Liu, Reed, & Phillips, 1990). But equipment procurement is only part of the problem; a part that perhaps is the easiest to solve. Just having computers in the schools is no guarantee that they will be used in a pedagogically sound manner or even used at all.

Even though computers have become an important part of education, and have been shown to be effective across all grade levels (Roblyer, Castine & King, 1988), most teachers simply don't have the technological skills necessary for effective technology utilization (Hasselbring, 1991; Nicklin, 1992; Miller, 1992; Ray, 1991. The Office of Technology Assessment (as cited in Hasselbring) reported that as recently as 1988, only one-third of K-12 teachers had even 10 hours instructional computing training. Furthermore, many graduates of teacher-education programs are less technologically proficient than their future students (Nicklin) and a substantial percentage of practicing teachers, who lack knowledge about computers, also lack the motivation to learn about them (Miller).

But before any judgments can be made about teachers' willingness or lack of willingness to become more knowledgeable about computer uses in schools, or become computer literate, some of the obstacles they face should be at least briefly examined. These obstacles can largely be classified as either equipment related or attitudinal. As indicated earlier, one computer for every 30 students is inadequate. This lack of equipment and difficulty scheduling time in computer labs understandably contributes to a disinclination to pursue computer use. Time is also one of the major problems; teachers' days and many evenings and weekends are already overloaded with meeting their present classroom demands. Learning about computers requires sizable chunks of time and most practicing teachers simply do not have enough to spare. Furthermore, large-scale computer integration into the classroom causes major changes in teaching methodology; changes that can potentially diminish teachers' self-confidence (Fishman & Duffy, 1992). Many also fear that computers might affect their classroom control and authority (Honey & Moeller (1990). However, to say that teachers are not willing to pursue computer literacy is irresponsible.

Hasselbring (1991) asserts that three conditions are necessary in order for teachers to accept technology. The first is well equipped facilities and physical environments that permit exploration and mastery of the technology. The second is computer training delivered regularly over *years* rather than short, disjointed workshops delivered sporadically. The third is constantly available technical support for hardware and software. Too often, boxes of computers and software are delivered and no-one is available to install them and answer the flood of inevitable questions. These three conditions address the technology and training aspects but ignore the personal, professional, and social aspects of teachers learning to use technology. Sheingold and Hadley (1990), maintain that, in addition to availability of equipment, teachers who are (a) motivated and committed to their students' learning (b) dedicated to their own professional development, and (c) have support and collegiality in their schools and school districts are much more likely to integrate computers successfully in their classrooms. Essentially, if teachers perceive classroom computing to be useful, and have adequate access to computers as well as technical support, they are willing to interrupt their routine to pursue instructional computing knowledge and skills.

Because computers have become pervasive in the workplace from traditional business climates to junkyard inventories, the need for computer literate educators to model and teach effective computer use cannot be ignored. So far, this paper has briefly highlighted some of the major issues related to teacher computer literacy in order to provide some background about that which is focused upon when teacher computer literacy is discussed. However, these issues are often bandied about without defining what comprises computer literacy. Such an omission is serious because problems are defined and solutions offered without a clearly defined end-state. Many believe that a definition of computer literacy is elusive because of different beliefs of what constitutes computer literacy that cover a broad spectrum from programming ability to the use of content-specific applications. I do not agree that computer literacy is an elusive term, nor do I believe that multiple interpretations of the terms should be blamed for problematic computer literacy programs for educators. Therefore, the purpose of this paper is to develop a definition of computer literacy broad enough to apply to educators in general, while leaving room for specificity for particular situations and content areas.

Critical Elements of Computer Literacy

The questions often asked by those responsible for preparing teachers to use computers effectively is: "What constitutes computer literacy?" and "How do we help teachers become computer literate?" But perhaps a more succinct phrasing is: "What knowledge do teachers need to effectively utilize computers in their classroom to help them with their job of teaching and what is the best way to provide that knowledge?" Recent interpretations of computer literacy emphasize learning how to use computers as tools for managerial tasks and to assist in teaching specific topics in education (Bozeman & Spuck, 1991; Sheingold, Martin & Endreweit, 1986). This paper attempts to address this question in three parts. First, four general domains that comprise computer literacy for all educators—(a) general computer operations, (b) software, including computer managed instruction (CMI), computer assisted instruction (CAI), and programming languages, (c) software issues, and (d) hardware are explored. Next, three models for developing teacher computer literacy programs will be discussed and, finally, the four domains will be explored for specificity by applying them to music education to provide an extensive example of an educational computing knowledge base.

General Computer Operations

Hardware Operation

Computer operations refers to the ability to operate at least one type of machine including simple operations such as turning it on, inserting a disk in the disk drive, and operating the printer. People should learn to use one type of machine in depth and then expand to other machine types because concepts formed from a deep understanding of one type of machine will transfer more easily to other machines. For example, extensive learning about and with an Apple IIe will make using an IBM or Macintosh later much easier because of transferable knowledge whereas trying to master all three somewhat simultaneously will likely lead to confusion and disillusionment.

File Manipulation

A basic skill needed by all computer users is the ability to manipulate files. This includes (a) moving files from disk to disk, (b) creating folders or directories, (c) moving files from one folder to another, or from one directory/level to another, (d) copying files, (e) making backup copies, (f) deleting files, and (g) renaming files.

Educational Software

Educational software can be divided into three broad categories: (a) computer assisted instruction (CAI), (b) computer managed instruction (CMI), and (c) programming languages. These categories can be thought to lie on a continuum from the least cognitively demanding (CAI) to the most cognitively demanding (programming languages).

Computer Managed Instruction

Computer managed instruction (CMI) includes all software designed to help teachers with administrative and secretarial chores such as word processors, spread sheets for budget work, data bases for inventories and student data, grade book programs, and test generators. CMI lies higher on the computer experience continuum than CAI; instead of reacting to the information presented by the computer as in CAI, CMI packages require instructions or procedures in order for the machine to know how to handle user input. Additionally, as discussed earlier, teachers usually consider CMI applications useful before CAI and, therefore, are more comfortable learning how to use CAI only after using CMI tools to help them with their administrative/secretarial type chores.

Another type of application that should be included in the computer managed instruction category is authoring languages and presentation software. Authoring languages are essentially programming languages that have been condensed and simplified so that non-programmers can write software. Examples of common authoring packages found in schools are *Hypercard* for the Macintosh, *Hyperstudio* for the Apple // series, and *Linkway* for DOS machines. Presentation software is similar to authoring software but is far less versatile. With presentation software, non-programmers can create multimedia presentations to help teach content via easy-to-use, icon-driven tools that can (a) access frames and series from a laser disc, (b) access portions of CD discs, (c) draw or access previously saved graphics, (d) create animations, and (e) access other programs.

Computer Assisted Instruction

Computer assisted instruction (CAI) software is comprised of four categories which also fall on a level-of-cognitive engagement and instructional complexity continuum: (a) drill and practice, (b) tutorials, (c) problem-solving applications, and (d) simulations. The first category, drill and practice, assumes that knowledge has already been taught and only serves as practice to promote automaticity and/or retention. The second category, tutorials, is similar to drill and practice but also includes the actual teaching components necesary to complete the task. Many tutorials provide a complete teaching experience—instruction, practice, and testing—and are therefore referred to as computer-based insruction (CBI). The third category, problem-solving software, includes any package designed to enhance problem-solving skills and are largely based on the four-step, scientific, problem-solving process: (a) define the problem, (b) develop a plan to solve the problem, (c) implement the plan, and (d) evaluate the effectiveness of the plan. The final category, simulations, are those programs that attempt to provide a near-to-real learning environment as is possible given system/learning environment restraints. When content is best learned in a real world environment, yet that environment is not available because of limitations such as cost, safety, or proximity, providing learning in a simulated environment is the next best. Simulations that incorporate laserdiscs are rapidly gaining popularity because of their ability to almost instantaneously access video footage to accompany text or simple graphics. The addition of video to simulations can potentially result in highly realistic learning environments (Overbaugh, 1992).

Computer managed instruction (CMI) packages can also be used as computer assisted instruction (CAI) tools, usually for problem solving. For example, students can use spread sheets to learn about money management or fiscal spending, data bases for collecting, organizing, and manipulating information about almost any topic and word processors for solving writing problems.

An interesting conundrum arises when CAI and CMI applications are compared with regard to their ease of use and their perceived utility to teachers. Even though CMI is more cognitively demanding than CAI, computers are important to teachers as tools to help with secretarial type tasks first (CMI) and as tools to help teach content second (CAI) (Bozeman & Spuck, 1991; Snyder, 1993; Woodrow, 1991). Therefore, the problem is: teachers are reluctant to use CAI before they become comfortable with computer applications that serve their personal needs (CMI), yet CAI applications are usually easier to learn and use. A logical solution is help teachers become proficient with word processing, a fairly easy to learn and extremely useful CMI tool, and some other applications such as test generators and gradebook programs. Once teachers can use such CMI tools easily, they realize the usefulness of the computer and become more willing to learn how to use the machine to help them teach content (CAI).

Programming Languages

A third software category is programming languages to understand how a computer functions or to teach higher-order thinking skills and enhance problem-solving ability. When microcomputers were introduced to education in the early 1970's, computer literacy advocates emphasised (a) programming skills and (b) drill and practice programs utilization (e.g., Kull & Archambault, Jr., 1984) but the emphasis has slowly shifted to learning how to (a) use computers as tools for secretarial/managerial chores (CMI) and (b) help teach content (CAI) (Bozeman & Spuck, 1991; Scheingold, Martin & Endreweit, 1986). However, many still feel that a computer literate teacher should know how to program. Therefore, programming will be discussed in relation to two aspects: (a) programming to understand how a computer functions and (b) programming to enhance higher-order thinking and problem-solving skills.

Programming to understand computer processing. Some feel that learning a programming language will help people understand how a computer works and thus gain valuable insight into what can be done with computers as tools. However, learning to program does not cause comprehension of how the machine actually works but, rather, simply illustrates the precise, meticulous manner in which instructions must be given to a computer to make it work as desired. The same sort of understanding can be obtained by using applications that require problem solving and careful "programming" to reach the desired result. An example is a spreadsheet that requires very careful planning and execution. When designing a spreadsheet, the user must determine the types of data to be entered in rows and columns, and what formulas will combine values from certain cells and place the result in another.

Reaching the same goal by two different methods immediately raises the question: Why use one rather than the other? The answer is twofold: First, learning how to use a spreadsheet is much faster than learning to program. To reach the same level of application sophistication with a programming language would require vast amounts of time. Second, learning to use a spreadsheet provides a useful skill that many users can transfer to the classroom. Therefore, a logical conclusion is that learning programming for a brief time, such as that to which most computer literacy programs are limited, will not result in any transference of skills regarding educational uses of the computer, and that the desired goal of teaching an appreciation of some aspects of the highly ordered internal processing of computers can be accomplished in a more efficient and useful manner via highly structured applications such as spreadsheets.

Programming to enhance problem solving ability. Programming is felt by many to enhance higher-order thinking skills and problem-solving by teaching the problem-solving process in a non-content oriented manner (Reed, 1987/88). When students learn to solve problems with a computer programming language, they engage in the four-step scientific problem-solving process: (a) specification—defining the problem by breaking it down into parts, (b) planning—devising a solution via programming commands, (c) coding—implementing the plan by coding a program, and (d) debugging—evaluating the solution by running the program to see if it works and debugging if there is a problem (Dalbey & Linn, 1986; Dalbey, Tourniaire & Linn, 1986; Kurland, Pea, Clement & Mawby, 1986). Some of the problems of early research programming-for-problem-solving research that failed to show problem-solving skills development are (a) brief treatments, (b) lack of control groups, (c) lack of measures tailored to the computer environment, (d) type of instruction, or whether or not students really learned a programming language, and (e) computer anxiety (Burton & Magliaro, 1987/88). Resolving these problems have resulted in increases in higher order thinking skills (Palumbo & Reed, 1987/88; Reed & Palumbo, 1987/88; Reed, Palumbo & Stolar, 1987/88), but transfer to other domains is still in question, although many believe it can happen.

Whether computer programming should be included in the computer-literate teachers' knowledge base depends on the purpose. If the purpose is to help teachers understand the inner working of the computer by examining the ordered manner in which lines of computer code are processed, then a more viable alternative is to use highly structured computer applications such as spreadsheets and databases that will serve double duty by transferring to other classroom uses. However, while prospective teachers do not necessarily need to learn to program, it is advisable for them to learn what are the common programming languages used in public education—Logo, BASIC, Pascal and C—and why they are taught, including the basic advantages and disadvantages of each.

Software Issues

Evaluation. Software evaluation is an imperative component of computer literacy. Teachers who wish to utilize computer based or computer assisted learning are faced with the problem of selecting quality programs that fulfill their students' needs. The necessity of an instrument to aid in the evaluation of available software is dictated by the two following reasons: First, in order to meet the high demand for educational software during the eighties, software companies flooded the market with proliferate amounts of marginal or poor software, rendering the task of selection a hit or, most likely, miss operation. Second, almost without exception, since schools operate on very constricted budgets, and because computer hardware and software are expensive, teachers have extremely limited buying power.

The educational software shopper is faced with a catalog of listings with short captions expounding each program's attributes. This approach is generally unsatisfactory and almost certainly inadequate for intelligent selection. The only way to be fairly certain a program is of good quality is to actually use and evaluate it, or consult someone reliable who has. Most software companies allow potential buyers to preview software for a limited time or return purchased software if not satisfied. There are many software evaluations available but, whatever one is chosen, there are a number of categories that should be included in order to thoroughly evaluate all educational aspects of prospective software:

1. General information—(a) bibliographical/biographical information such as the title, publisher and copyright stipulations, and (b) hardware requirements.
2. Documentation—(a) directions for using the software, statements of instructional purpose, objectives, auxiliary materials, and validity data, and (b) information about on-line directions.
3. Instruction—the instructional characteristics and methods of the software.
4. External control of the software—how much control the student and/or teacher has over the program.
5. Internal control of the software—how the program responds to student input.
6. Feedback and remediation—what type of feedback and remedial work is provided by the program.
7. Record keeping—what type of records are kept by the software and by what method, including instructions for access.
8. General content information—the type of instruction that is used and its factual integrity.
9. Subjective opinion of general content—the evaluator's opinions about the quality, appropriateness and values of the program.
10. Mechanics of the program—the quality of the sound, graphics and operation of the program.

Prices. Computer literate teachers should have some idea about the costs of various types of software. Although prices vary tremendously, the old saying, "You get what you pay for" is generally true and teachers should realize that certain genres are far more expensive than others. For example, many computer managed instruction packages and multimedia programs are very expensive whereas many content-specific educational programs are less expensive. Additionally, teachers should be aware of inexpensive software sources such as shareware and public domain software.

Copyright. No computer literacy program should be without instruction about copyright laws. Software theft is common because many otherwise honest people simply don't realize that copying a program for their own use, or making multiple copies of a single-user program for classroom use, is a form of stealing. Copyright instruction should include a segment detailing the many options available to educators such as multi-packs, lab-packs, and site licensing.

Hardware

The hardware domain includes (a) types of computers and their components, (b) basic computer architecture, and (c) general prices including popular peripherals.

Types of computers. There are three types of computers teachers are likely to encounter: (a) Apple II series, including the IIe and GS, (b) Macintosh, and (c) IBM and compatibles or DOS machines. In some respects, the Apple II or DOS machines may be the more appropriate choice if literacy across platforms is the goal because the Macintosh, with its extremely easy-to-use, intuitive graphic interface will not lead to much transfer if the user is likely to have to later use Apple II or DOS machines which require much more knowledge and expertise for efficient operation. However, the current trend is away from the more complex operating systems towards systems such as or similar to the Macintosh. The Apple II series, while still common in most schools is rarely chosen if Macintosh is an option. Furthermore, the distinction between the operating systems of the Macintosh and DOS machines is becoming blurred with the growing popularity of the Windows, and to a lesser extent OS2, operating environment. The Windows environment provides a graphic, mouse-driven operating system that, in many respects, is the same as the Macintosh environment. If this trend continues, all systems will likely be very similar in the future.

Computer architecture. A deep understanding of computer architecture is not needed, but teachers should know what some of the basic components are including: (a) RAM and the limitations imposed by the amount of RAM, (b) ROM, and (c) the basic function of the CPU.

Multimedia components. The two main multimedia components about which teachers should be aware are CD ROM players and laserdisc players. Other common peripherals include bar code readers, digital cameras, digital recorders, and indexing VCR's. Learning about these components will result from working with programs that utilize them, but some attention should be given to prices and the advantages/disadvantages of particular types. For example, Pioneer laserdisc players are the most common and, therefore, many of the level two laserdiscs as well as level three programs are written only for Pioneer players.

Local area networks (LANs). Network technology is actually a hardware *and* software concern that should be included in a literacy program. Networks are rapidly gaining popularity in schools, and teachers should be aware of the pros and cons associated with networked systems

Prices. Computer prices change rapidly, but teachers should know approximately how much various machines cost, and what peripherals and options are available separately and their cost. For example, even though Apple prices seem to be reasonable, extras purchased later (e.g., disk drives, color monitors, and printers) are extremely expensive when compared to IBM compatible equipment.

Critical Elements of Computer Literacy Summary

These four domains—(a) general computer operations, (b) educational software, (c) software issues, and (d) hardware—comprise an admittedly large amount of knowledge and skills. With the heavy course load already required for completing an undergraduate or graduate teacher education degree, many feel that time simply won't permit the addition of classes to cover such a comprehensive knowledge base. However, as mentioned earlier, since computers have become an integral part of our society and schools, professional educators are responsible for teaching with the technology in order for students to be prepared to use it in the workplace. The next section of this paper discusses some ideas for teacher preparation institutions to consider for implementing adequate computer literacy goals for teacher education students.

The Path to Computer Literacy

Becoming computer literate clearly will take more than one or two computer classes. However, depending on the present course offerings, a teacher preparation institution that dedicates itself to preparing computer literate teachers may not need to add any classes to the curriculum. One of the biggest mistakes seen throughout many schools is the idea that the job of teaching computer-based instruction skills to education students should be left to a computer teacher. This is a huge mistake. Effective computer use needs to be modeled by content area instructors across the curriculum or students will view computer use as something that occurs in a computer lab and is best left to the "computer people". If the education faculty effectively integrate computers into their methods classes, students will gain the necessary instructional computing skills without the need for additional classes, or only one or two. No matter how students are taught to be computer literate, they are learning about a new technology which results in definite stages through which they will pass. Three models will be discussed briefly as they can serve as a guide to curricula development or restructuring.

Models for Computer Literacy Development

Two-stage model. One of the most basic beliefs related to instructional computing is that computers must become personally meaningful before teachers will use them to help others (Bozeman & Spuck, 1991; Snyder, 1993). Therefore, the most important application of computers for practicing teachers and their students is as tools to help with administrative tasks (computer managed instruction) first, and to help teach content (computer assisted instruction) second (Bozeman & Spuck; Woodrow, 1991). With this in mind, the path to computer literacy for educators should begin with the utilization of computer management tools such as word processing, spread sheets, and grade book programs to develop personally meaningful skills and then proceed to learning about applications that help teach content (CAI).

Four-stage model. Kell, Harvey, and Drexler (1990) do not separate CMI and CAI uses as different stages but, rather, offer four stages through which new users of innovation pass: (a) awareness and preparation, (b) using the innovation mechanically and then smoothly, (c) refining the innovations' use and integrating it into their curricula, and (d) assessing its consequences for their students. They further state that two to three years are necessary for new users to pass through all four stages.

Seven-stage model. Either the two-stage or four-stage model can easily be subsumed by the seven ordered concerns identified by Hall, Rutherford and George (1988)—(a) awareness, (b) informational (c) personal, (d) management, (e) consequence, (f) collaboration, and (g) refocusing—that are generally followed when people are introduced to an innovation. These stages can be quantified with the Hall, Rutherford, and George stages of concern instrument which is based on the notion that, when introduced to an innovation, people tend to have very selforiented concerns and, as they learn more, become concerned with how to manage that innovation in their classroom and, finally, begin to explore how they can work with others to share their experiences and learn new ways to use previously learned ideas. The first concern stage is Awareness: I am not concerned about microcomputers. The second stage is Informational: I am concerned about learning more about microcomputers. The third stage is Personal: I am concerned about how using microcomputers will affect me personally in the classroom. The fourth stage is Management: I am concerned about the time needed to learn about microcomputers. The fifth stage is Consequence: I am concerned about the effects my use of microcomputers will have on my students. The sixth stage is Collaboration: I am concerned about working with others to learn more about microcomputers. The seventh stage is Refocusing: I am concerned about learning new ways to use what I already know about microcomputers.

Regardless of how complex one wishes to be with regard to the number of distinguishable concern levels, these progressive priorities can serve as a model for computer literacy development that suggests educators should (a) become aware of the potential uses of computers, (b) learn to use computers for managerial chores such as word processing, data basing, generating tests and keeping grades electronically (CMI) and (c) to use computers to assist their instructional endeavors (CAI) by integrating drill and practice, tutorials, problem-solving, and simulation packages with their regular classroom activities. This is the type of approach to computer literacy developed and implemented by Marietta College in Ohio for pre-service teachers which introduces computers to lower-level education majors and continues teaching progressive educational computing throughout the undergraduate career and into graduate classes (Golden, 1991). The Marietta College model involves five components: (a) Acceptance, (b) Understanding, (c) Application, (d) Evaluation, and (e) Design.

Computer Anxiety

A topic that cannot be ignored in any discussion of computer literacy development is the affective phenomenon of computer anxiety. Before teachers, who often experience more anxiety than their students (Cambre & Cook, 1985), can be expected to move through the expected development levels, they must be comfortable with the technology (Wissick, 1992). Computer anxiety is usually thought of as a temporary condition that can be overcome by learning in an environment structured to reduce that anxiety (Torris, 1985). Therefore, the first step to literacy should be to provide some computer-based activities structured to help teachers overcome their anxiousness. Research has shown various treatment lengths can effectively reduce computer anxiety from as much as 60 hours (Honeyman & White, 1987) and four weeks (Reed & Palumbo, 1987/88) to as little as 6 hours (Ayersman & Reed, 1993; Overbaugh & Reed, 1990; Overbaugh & Reed, in press). Another study showed that a brief instructional activity—six contact hours—of learning *with* computers reduced anxiety significantly more than learning *about* computers (Overbaugh, 1993). In other words, learners engaged in a task that simply uses the computer as a tool for learning, such as an easy-to-use simulation, more effectively reduces anxiety than an equivalent amount of time learning about the computer and its uses in education. Overcoming computer anxiety is certainly not the only goal for a literacy course (Woodrow, 1991), but should be the first step in order for students to learn about computers more efficiently.

Teaching With or About computers?

The Role of Declarative and Procedural Knowledge

Courses designed to lead to computer literate teachers should be structured in a manner that utilizes both procedural and declarative knowledge. Like many subjects, learning about computers can be uninteresting and useless of the students are unable to develop positive attitudes toward computers and formulate ideas of how they can use their new knowledge for their own or their students' benefit. Because the knowledge base for computer literacy is quite large, teaching everything in a hands-on, or procedural, manner is virtually impossible. Therefore, it is best to provide students with sufficient amounts of practical *hands-on experience* while also teaching *about* other uses and aspects of educational computing. A problem with this approach is that the vast amount of time required for students to learn by doing about computer fundamentals severely constrains the amount of general declarative knowledge that can be imparted in the same course. However, it is imperative to take the time required for personal interaction with the computer at the expense of the much larger body of knowledge that can be talked and read about but not actually experienced. Why? Because *learning by doing is more meaningful than learning about.* Many courses designed to raise computer literacy levels target novice users. Therefore, actually using the machine in a manner that is meaningful to them will build a stronger, more meaningful knowledge base that will serve as a starter structure into which related declarative knowledge can be placed more quickly, more easily and will be more retrievable than if students attempted to subsume and organize many raw, unrelated facts.

With this in mind, consider an introductory computer literacy course, which should be organized in a manner that attempts to strike a careful balance between procedural and declarative knowledge. Most beginners, although often anxious about computers, are eager to begin working with them, so a first step is to engage them in tasks that teach basic computer operation by using programs that introduce learners to computers in a friendly, non-threatening manner. Possibilities include introductory programs that are often supplied by computer manufacturers, followed by an easy-to-learn and use computer managed instruction program. A word processor is a good choice because many are very easy to use and the students can begin using them immediately in preparing classwork.

With hands-on experience and newly acquired practical skills, declarative knowledge can be quickly and easily imparted. After running an introductory program, and learning a simple word processor it becomes easy to talk about computer architecture, particularly RAM and its limitations, prices, types of printers, and other hardware. Different types and levels of word processors can be discussed, and the myriad uses for them at home and school can be introduce. Related software, such as spell checkers, grammar packages, writing process software, and writing styles templates can be introduced and discussed within the established framework. If this new knowledge is declarative, assimilation will be much easier than if it were introduced without having had related practical experience.

The same sort of progression can be followed for two other types of CMI software—data bases and spreadsheets. For example, a simple address and phone book data base can be learned, after which school related uses such as student and parent lists, room inventories, and equipment lists can be talked about. A simple spreadsheet could be developed such as a checkbook program which will lead to an easy understanding of how spreadsheets can be used to create, work with, and balance budgets—a task that is part of many teachers' jobs.

Learning about the computer and CMI type software may easily constitute the entire content of a computer literacy class. CAI software may also be a part or literacy classes but should be taught largely in content area courses so that proper CAI pedagogy and curricular integration is modeled. Regardless of the particular distribution of literacy components over the curriculum, students should be provided with a good working procedural knowledge base of practical and personal computer experiences, and related meaningful declarative knowledge. The procedural knowledge base will serve as a springboard for further learning, understanding, and utilization of instructional computing from newly acquired declarative knowledge.

Computer Literacy in a Content Area

In addition to general computer skills, teachers need to develop skills specific to their particular field. The next section of this paper will provide an example of some of the content specific applications that could be part of a music teacher's computer knowledge base.

Content Area Uses of Computer Managed Instruction (CMI) Software

To illustrate the use of CMI software in a specific content area, an instrumental music teacher will be used. In addition to using general CMI software for writing letters and lesson plans (word processor), balancing the budget (spreadsheet) and keeping track of student data and inventory (database), a content specific CMI program is field show design software. These programs help marching band directors write field shows by placing the students on the field in specified formations, automatically calculating spacing, and showing how various formations move from one position to another. The program allows easy experimentation and changes and when finished, provides printouts of field charts for teaching the show.

Content Area Uses of Computer Assisted Instruction (CAI) Software

The computer literate music teacher can take advantage of a wide variety of CAI packages to integrate with their regular classroom teaching or as auxiliary teaching tools for students who need or desire them. Examples are : (a) theory programs, (b) music construction sets, and (c) MIDI (Musical Instrument Digital Interface) software and hardware. Each of these will be discussed briefly.

Theory Software. Theory programs address aural theory or written theory. Aural theory programs teach common skills such as interval recognition, melodic dictation, chord recognition, and chord progression identification. This area is also the most severely limited by hardware: If the computer is incapable of producing a wide variety of tonalities in various timbres, the software will not be very useful. Machines such as the Apple IIGS and the newer Macintosh series have good sound capability but should still be interfaced with higher quality sound reproduction equipment.

Written theory is more text or graphics based than aural theory and therefore is better suited to many computers than aural theory. Written theory covers a wide spectrum beginning with simple note recognition skills to advanced composition analysis. Software addressing the lower end of the scale (e.g., note naming programs, fingering programs to help instrumentalists learn fingerings on their particular instruments, and music terminology) is useful for remedial work. More advanced written theory software can be integrated with theory classes or the software may be used as an adjunct to classes that do not have the time to cover material desired by some students. Examples are: (a) learning to build or analyze chords and chord progressions, (b) writing scales, and (c) transposing music for various instruments.

Until recently, hard drive storage capacity limitations or disk swapping prevented the development of complex applications that combine sound, graphics, and text into single programs. However, the large storage capacity of CD ROM has enabled software developers to begin to create high quality, multimedia applications integrating high fidelity sound, complex graphics and large amounts of text.

Music construction sets. Music construction sets is a generic term for programs designed to help students write music. The programs generally ask for certain parameters such as number of staffs needed, required range, and type of instrumentation, after which users can place notes of different values, dynamic marking, barlines, tempo markings, and text on the staff(s). Some of the more advanced programs will check the work for correct bar values and will highlight any bar with too many, or too few beats. Some packages will also play the composition, although the computer should be interfaced with a sound system.

The music writing capabilities of music construction sets can sometimes be accomplished with certain theory programs (discussed above) and through MIDI systems (discussed below) but are noteworthy as a single category because they are usually easier to use and generally have more capabilities for providing feedback and playing the compositions than theory programs. Music construction sets are separated from MIDI systems because they are less expensive, require less hardware and are usually easier to use.

MIDI. MIDI is an acronym for Musical Instrument Digital Interface which simply links musical instruments (usually, but not limited to, keyboards) to a computer by converting the instrument's output to a digital format that can be handled by a computer. A MIDI system is useful to instrumental teachers as both a CMI and CAI tool. A CMI use is replacing missing instrumental parts or creating new parts to accommodate incomplete instrumentation or players with unique needs. To do this, a musician need only play the part on the keyboard. The MIDI program converts the melodic line to digital information which the computer can use to create written output. The computer can also transpose the part to the correct key if the author chooses to work in concert key (the standard key, based on the piano). If desired, the MIDI software and keyboard can be used to create complete arrangements.

The same note transposing functions can play an integral role in other settings as a CAI type application. Students in theory classes or jazz groups can use the keyboard to experiment with various melodic lines for improvisation, theme development, themes and variations, or just simple experimentation with tonalities, timbres, and ideas.

A third use of MIDI software and hardware is to provide the sound of instruments that are missing from an ensemble. Since MIDI sounds are digital, a MIDI system can produce tones of various timbres in one of two ways. First, most common sounds are supplied with a MIDI compatible instrument on floppy disks. The desired timbre is chosen (e.g., piano, a myriad of percussion sounds, strings, brass and woodwinds) and the keyboard is played like any keyboard but the output sounds like the chosen instrument. A second method of sound production is employed should the desired timbre not be available: The MIDI system can record a single pitch of the desired sound—called sampling—and convert it to cover the range of the keyboard. If sampling is insufficient, complete sound editing is available by altering the frequencies and sound waves of each timbre.

MIDI systems can be extremely useful in many instrumental music settings. They can fill out weak ensembles, provide an easy way to experiment with music composition for students or teachers, and is a good tool for creating written sheet music. The technology is not overly expensive, especially if it is used as more than a fancy keyboard.

Music Teacher Computer Literacy Summary

To be computer literate, an instrumental music teacher should have a solid knowledge base of general information regarding at least one type of computer including prices, operation, useful peripherals and some architecture. The general knowledge should also include CMI applications such as a word processor, data base, and possibly a spreadsheet program that can be used for any subject by any teacher. This general knowledge can be taught in a basic computer literacy course designed to meet the needs of teachers.

Besides general knowledge, a computer literate teacher needs specific knowledge that can be applied to their particular situation such as theory programs, music construction sets, and MIDI hardware and software. This knowledge should be integrated with the general knowledge base but should be taught in content area methods classes which concentrate on specific techniques to help educators learn to teach effectively and efficiently.

Summary

The purpose of this paper was to identify many facets of computer literacy for teachers and suggest ways in which teacher-preparation institutions can thoroughly integrate computer based and computer assisted instruction throughout their curricula. The point that has to be emphasized once again is that to graduate computer literate teachers, instructional uses of computers must be modeled for teacher education candidates by their professors throughout the higher-education curriculum and not be limited to one or two "computer classes." If higher-education fails to thoroughly integrate computers across their curricula, tomorrow's teachers will not be adequately prepared to teach our students to be productive citizens in a technologically intensive work environment.

REFERENCES

Ayersman, D. & Reed, W.M. (1993). *The effect of three separate learning styles on computer anxiety: Assessed before and after an undergraduate BASIC programming course.* Paper presented at the 1993 Annual Meeting of the Eastern Educational Research Association, Clearwater Beach, Florida.

Burton, J.K. & Magliaro, S. (1987/88). Computer programming and generalized problem-solving skills: In search of direction. *Computers in the Schools, 4*(3/4), 63-90.

Boseman, W.D. & Spuck, D.W. (1991). Technological competence: Training educational leaders. *Journal of Research on Computing in Education, 23*(4), 514-529.

Dalbey, J. & Linn, M.C. (1986). Cognitive consequences of programming: Augmentations to BASIC instruction. *Journal of Educational Computing Research* 2(1), 75-93.

Dalbey J., Tourniaire, F. & Linn, M.C. (1986). Making programming instruction cognitively demanding: An intervention study. *Journal of Research in Science Teaching, 23*(5), 427-436.

Cambre, M. A. & Cook, D.L. (1985). Computer anxiety: Definitions, measurement, and correlates. *Journal of Educational Computing Research, 1*(1), 37-54.

Fishman, B.J. & Duffy, T.M. (1992). Classroom restructuring: What do teachers really need? *Educational Technology Research and Development, 40*(3), 95-111.

Golden, C.S. (1991). Hypermedia in teacher education: Integrating technology across the curriculum—the Marietta College model. in *Technology and teacher education annual,* (88-90). Greenville, NC: School of Education, East Carolina University.

Hall, G.E., George, A.A. & Rutherford, W.L. (1977). *Measuring the stages of concern abut the innovation: A manual for use of the stages of concerns questionnaire.* Austin: Research and Development Center for Teacher Education, The University of Texas.

Harrington, J. (1991). Normal style technology in teacher education: Technology and the education of teachers. *Computers in the Schools, 8*(1-3), 49-57.

Hasselbring, T.S. (1991). Improving education through technology: Barriers and recommendations. *Preventing School Failure, 35*(3), 33-37.

Honey, M. & Moeller, B. (1990). *Teachers' beliefs and technology integration: Different values, different understanding* (Report No. 6). New York: Bank Street College of Education.

Kell, D., Harvey, G. & Drexler, N.G. (1990, April). *Educational technology and the restructuring movement: Lessons from research on computers in the classrooms*. Report No. IR-014735). Paper presented at the Annual meeting of the American Educational Research Association, Boston, MA. (ERIC Document Reproduction Service No. ED 326 195).

Kull, J.A. & Archambault, F.X. Jr. (1984). A survey of teacher preparation in computer education. *Journal of Teacher Education, 35*(1), 16-19.

Kurland, D.M., Pea, R.D., Clement, C. & Mawby, R. (1986). A study of the development of programming and thinking skills in high school students. *Journal of Educational Computing Research, 2*(4), 429-455.

Liu, M., Reed, W. M. & Phillips, P. D. (1990). *Teacher Education Students and Computers: Gender, Major, Use, Occurrence, and Anxiety*. Paper presented at the annual convention of the American Educational Research Association, Boston.

Miller, M.J. (1992). The multimedia revolution needs teachers, too. *Newmedia, 2*(12), 12.

Nicklin, J. (1991, July 1). Teacher's use of computers stressed by education colleges. *Chronicle of Higher Education*, pp. 15-17, A17.

Overbaugh, R.C. (1993). *The effects of instructional content, brief instructional activities, and learning modality on teacher education students' computer anxiety*. Paper presented at the 1993 Annual Meeting of the Eastern Educational Research Association, Clearwater Beach, Florida.

Overbaugh, R.C. (1992). *The efficacy of interactive video for teaching basic techniques of classroom management to pre-service teachers*. Paper presented at the 1992 annual Meeting of the Eastern Educational Research Association, Hilton Head, North Carolina.

Overbaugh, R.C. (1990). *Research based guidelines for computer based instruction development.* Paper presented at the 1991 Annual Meeting of the Eastern Educational Research Association, Boston, Massachusetts, (ERIC Document Reproduction Service No. ED329 586).

Overbaugh, R.C. & Reed, W.M. (1990). *The effects of brief instructional activities on teacher education students' computer anxiety and performance.* Paper presented at the 1990 Annual Meeting of the Eastern Educational Research Association, Clearwater, Florida.

Overbaugh, R.C. & Reed, W.M. (in press). The effects of prior experience and instructional format on teacher education students' computer anxiety and performance. *Computers in the Schools.*

Palumbo, D.B. & Reed. W.M., (1987/88). Intensity of treatment and its relationship to programming problem solving. *Computers in the Schools, 4*(3/4), 119-128.

Ray, D. (1991). Technology and restructuring: New educational directions, Part 2. *The Computing Teacher, 18*(7), 8-12.

Reed, W.M. (1987/88). A philosophical case for teaching programming languages. *Computers in the Schools, 4*(3/4), 55-58.

Reed, W. M. & Palumbo, D. B. (1987/1988). The effect of the BASIC programming language on problem-solving skills and computer anxiety. *Computers in the Schools, 4*(3/4), 91-104.

Reed. W.M., Palumbo, D.B. & Stolar, A.A. (1987/88). The comparative effects of BASIC and logo instruction on problem-solving skills. *Computers in the Schools, 4*(3/4), 105-114.

Roblyer, M.D., Castine, W.H. & King, F.J. (1988). *Assessing the impact of computer-based instruction, a review of recent research.* New York: The Haworth Press.

Scheingold, K. & Hadley, M. (1990). *Accomplished teachers: Integrating computers into classroom practice.* New York: Bank Street College of Education.

Scheingold, K., Martin, L.M. & Endreweit, M.E. (1986). Preparing urban teachers for the technological future. In R.D. Pea and K. Sheingold (Eds.), *Mirrors of Minds: Patterns of Experience in Educational Computing.* Norwood, N.M.: Ablex Publishing Corporation.

Snyder, T. (1993). Keynote speech at the 1993 Annual Meeting of the Indiana Computer Educators (ICE) Conference, Indianapolis, IN.

Torris, D. (1985). *Suggested approaches to the measurement of computer anxiety.* Paper presented at the annual meeting of the Southeastern Psychological Association, New Orleans, (ERIC Document Reproduction Service No. ED254540).

Wissick, C. (1992). Teacher training in multimedia: Content enhancements and considerations for instruction. In R. Boone & K. Higgins (Eds.), *TAM topical Guide #1* (pp. 8-23). Reston, VA: The Council for Exceptional Children.

Woodrow, J.E.J. (1991). Teachers' perceptions of computer needs. *Journal of Research on Computing in Education, 23*(4), 475-496.