An Instructional System for Computer And Information Literacy

Jianping Zhang and Donald H. Cooley
Department of Computer Science
Utah State University
Logan UT 84322-4205

Abstract

This paper describes the development of a multimedia instructional system for computer and information literacy. This system consists of seven self-contained modules: Public Access Computer Labs & Electronic Mail; Computer Basics; Operating System Basics; Computer Networks and Information Superhighway; Document Preparation; Data Visualization and Spreadsheets; and Library Information Systems.

Introduction

Computers pervade every aspect of our lives. Millions of people worldwide interact with computers. Students use computers to write term papers. Travel agents use computers to plan a trip, and reserve airline tickets and hotels for travelers. Warehouses and department stores use computers for sales and inventory control. Children use computers to play games. Doctors, hospitals, and medical researchers use computers for diagnosis and treatment. Individuals use computers to place orders, read news, and send messages. Computers are at the forefront and heart of nearly all technologies. All of these uses imply that most of us depend on computers either in our work and/or in our daily life. Throughout our lives, we will be inevitably affected by computers. To retain control of our lives, we will need to understand computers and use them in a positive and productive way.

Government studies have shown that approximately 40 million Americans are functionally illiterate. Few would argue that in addition to this group, there are also many more Americans who are computer, information, and technology illiterate. The number of VCR's in American homes which continually blink "12:00" in their clock window is an indication of this technological illiteracy. A recent report in Newstrack stated that 36% of the 90% of the people who use computers in their work feel they are not computer literate (Fox, 1993).

At Utah State University, the importance of computer literacy to a student's career is well recognized. For this reason, computer literacy will no longer be viewed as a course requirement. Instead, it will be viewed as a college skill. Beginning Fall 1996, computer literacy will be required of all majors in the College of Science. In addition, the deans of all colleges in the university have endorsed this requirement for their colleges, and hence, the requirement should become university-wide in the near future.

Providing the necessary traditional classroom instruction for computer literacy for all students in the University would be an enormous undertaking for the Computer Science Department. It would require a large number of qualified instructors who are not currently available. Other disadvantages of classroom instruction are: 1) students have to attend at a fixed time; 2) the needs of students with different backgrounds cannot be met; 3) lab experiments separate from lectures; 4) students have no control on their learning pace. Our solution to this problem is to develop interactive, computer-based instructional technologies to assist and empower instructors. Computer-based instructional technology has demonstrated its ability to carry out numerous education and training functions effectively. This paper describes an instructional system developed for teaching computer and inforamtion literacy.

Course Content and Course Structure

The computer and information literacy course is based on a set of core knowledge modules. These knowledge modules will define computer literacy. By developing multiple modules, it will be possible to have more than a single or generic definition of computer and information literacy. By choosing different knowledge modules, departments can tailor the definition of computer and information literacy to meet the needs of their majors. Some knowledge modules will be common to all disciplines and hence will give breadth. Other modules will be specific to a discipline and hence will give depth. Each of the knowledge modules will be developed as a self-contained and self-paced learning entity. This sections discusses the course content for core modules and the structure of a module.

Course Content

Currently, seven modules are under development. They are Public Access Computer Labs & Electronic

Mail; Computer Basics; Operating System Basics; Computer Networks and Information Superhighway; Document Preparation; Data Visualization and Spreadsheets; and Library Information Systems.

Public Access Networks Labs. & Electronic Mail There are two learning objectives associated with this module. The first one is to give incoming students an overview of the University's public access computer network and laboratories. With video, students are shown how to access the labs, where they are located, what hardware and software are available, what time labs are open, etc. The second objective of the module is to teach students the basic computing skills in using different kinds of computers (e.g., MAC and IBM PC, UNIX), minimum editing skills, and how to send and receive electronic mails.

Computer Basics This module introduces students to critical compute terms and concepts. Specifically, topics covered in this module include: what is a computer; what a computer can do; overview of computer hardware, computer system software, and application software; what is information; and ethics of access and use of computers.

Operating System Basics This module includes the basic knowledge about operating systems: what is an operating system, what an operating system does, and how to use a particular operating system. Operating systems that may be taught in this module include command-line operating systems: DOS, UNIX, and VMS and visual operating systems: MAC and Microsoft Window, Windows 95.

Computer Networks And Information Superhighway There is a wealth of information available through Internet and on various bulletin boards. Prior to beginning this module, students have already learned basic e-mail techniques. In this module, students learn the basics of computer networks; both type and functionality. They would also learn about accessing information resources available through Internet.

Document Preparation This module teaches students about word processing. It first describes the basic editing processes used in creating memos, letters and other typical documents. It also describes some basic functions at the heart of a word processor. It then teaches the use of several most commonly used word processors

Data Visualization and Spreadsheets Spreadsheet programs represent a tremendous tool for the presentation, analysis, and visualization of data. In addition to their ability to present data in many different tabular forms, they also have powerful graphics capabilities. The goal of this module is to instruct students in some of the ways in which data can be presented for analysis through the use of at least one commonly available spreadsheet program

Library Information Systems The learning objective of this module would be to learn how to access the library's information using computers. Upon completion of this module, students should be able to search the material they need using some library information system. They should be able to search by author, title, and keywords. They would also learn the meaning of call numbers. With video, they will be shown where the books they need are located.

Course Structure

Upon entering a module, the student is given two options: *Test and Instruction. Test* is for self-evaluation and the student is free to take a test at any time. All test questions are randomly selected from a question pool. Based on the test results, a study plan is provided for the student. The plan tells the student which topics the student can skip over and which topics the student needs to spend more time on.

Instructions in a module consists of two sessions: concept session and tool session. The concept session teaches the student general concepts that are shared by all specific tools. The tool session teaches the student how to use specific tools. For example, the concept session of the Document Preparation module teaches concepts such as editing, open files, insertion, and deletion, and the tool session consists of two of the most popular document processing tools: WordPerfect and Microsoft word and teaches the student how to use WordPerfect or Microsoft Word to prepare documents. In particular, the tool session teaches the student how to use commands or functions such *as open a file* provided by a tool. It is believed that in such a modular one should not attempt to teach all of the capabilities of the associated tools. Rather, the approach is to teach the minimal functionality needed to use such a tool, and to give extensive instruction in the use of a tool’s help facilities. It is expected that students will then use the help tool and their basic knowledge of the tool to learn new functionality.

Both the concept and the tool sessions are organized as hierarchies of subtopics. At the bottom level (or leaf) is a list of instructional transactions, each of which teaches a concept or a command.

There are two sequences in which the student can study the course material. The student can learn all general concepts first, and then learn functions of one tool and then functions of another tool. The student can also learn one general concept and the corresponding functions in tools, and then another general concept and its corresponding functions in tools.

Teaching in Our Instructional System

This section discusses the teaching methods which we use in our instructional system.

Teaching A Command

Learning to use an application software tool involves learning of the use of a set of commands or functions. To actively involve a student in learning, teaching of a command consists of the five instructional activities: *Preview*, *Presentation*, *Demonstration*, *Practice*, and *Test.* (Merrill, et al., 1991).

Preview informs the student of the learning objective: what the student can do after he/she completes the learning of the command. In addition to the learning objective, the student is also informed of the prerequisites: what the student must know before he/she can start learning the command. The prerequisites are provided as hotwords so that the student can click them to review these prerequisites.

Presentation displays all the material for mastery of the command to be taught. For example, for the command *open files* in WordPerfect, presentation gives a text description of all steps of executing the file open command.

Demonstration uses examples to illustrate the material presented in *presentation*. For the example of the *open file* command, Demonstration shows how to open a specific file step by step.

Practice assigns some problems to the student and then the student solves the problems assigned on the simulated. Problems can be solved with or without guidance. With guidance, the system shows the student what to do at each step. Without guidance, the student solves problems on his/her own. Hints for solving a problem are always available upon the request of the student. Feedback is provided after each problem is solved.

Test checks the level of mastery the student has attained for the command and determines if the student passes the command or not. *Test* questions may be performance questions or concept questions. A performance question is similar to the problem in *practice*. Neither hints nor feedback is provided in *test*. Only the score is given to the student. Concept questions are multiple questions, true/false questions, matching questions, sequencing questions, and fill-in-blank questions. The student is informed if he/she passes or not.

Demonstration, *practice*, and *test* share a pool of problems. These problems are used differently by these activities. A problem consists of several subcomponents: description, solution, hints, and feedback. Description is the description of the problem and is shown to the student in all of these activities. Solution is the correct solution of the problem and may consist of a number of steps. In demonstration, a solution is presented to the student. In practice and test, the student's solution is compared with the stored solution to determine if it is correct. Hints are associated with each step of the solution and provided upon the student request. Feedback is associated with correct answers as well as errors and is provided for the student once a solution is entered.

Simulation

The most effective way to learn a software application is to watch demonstrations and practice. Simulation of a software system provides an environment for our instructional system to perform demonstrations and for the student to practice. A simulated software system consists of a simulated screen and a simple parser. The simulated screen could be exactly the same as the actual software system and hence the student may not know whether he/she is working on a simulated or an actual software system. The parser receives the input from the student and checks for correctness. If correct, the response of the software system is printed on the simulated screen and some reinforcement message is given by our instructional system. If not correct, the error message generated by the software system is printed on the simulated screen and more detailed feedback about the error is available from our instructional system.

A simulated software system serves the purpose of instruction better than an actual software system, because the simulated system is under the control of the instructional system. For example, when the student makes an error, the instructional system knows what kind of error the student makes. This kind of information could be used to guide the future instruction.

Teaching in A Subtopic

A module is organized as a hierarchy of subtopics. A subtopic may consists of a number of subtopics or a number of commands. When the student enters a subtopic, he/she is displayed a summary of the content to be taught and a few problems which he/she can solve after completing the study of the subtopic. He/she are also displayed buttons for a list of next level subtopics or commands and buttons for practice and test.

Practice in a subtopic is different from practice in teaching a command and provides the student with the access to an actual software system. Although a simulated software system is available most of the time for demonstration, practice, and test in teaching commands, it is still worthwhile to let the student use the

actual software system. One advantage with the use of an actual software system is that it may stimulate the student's enthusiasm for learning. One problem with the use of an actual software system is that the instructional system must relinquish control to the student whenever it is connected to an actual software system. The student must have full control and hence can stay on the actual software system as long as he/she wants and can do whatever he/she wants. Therefore, the student is not allowed to use the actual software system unless he/she passes the test of the subtopic.

Practice on an actual software system is guided by the instructional system. The system provides a list of problems for the student to choose. After a problem is chosen, the steps for solving the problem is shown to the student in one window and the actual software system is open in another window. The student is free to switch back and forth between the instructional system and the actual software system. Thus, he/she can get help from the instructional system at any time he/she needs.

Student Records

A student record stores information about a student. The information stored includes: personal data such as name and ID, rank and major; computer background such as the number of years of experience in using computers; a learning history such as pretest results, the subtopics completed, learning performance; learning attitude such as the time spent on each course unit; instructional setting such as learner control mode; and learning topics.

Each time a new student enters the system, a student record is created and the personal data and computer background are entered by the student. The rest of the information is recorded by the system. For example, each time a student completes the study of a command or a subtopic, corresponding information is placed in the student record.

A student record is used for three purposes. First, an instructor can use the student record to keep track of the learning history and performance of the student. Second, the system uses the student record to inform the student of his/her learning history and performance each time the student enters the system. Third, a student record serves as a weak student model (as opposed to strong student models used in Intelligent Tutoring Systems [VanLehn, 1988]) and is used to guide the instruction to adapt to the student throughout the course.

Software Design and Implementation

In our instructional system, content knowledge is separated from the instructional delivery tool and stored in the content knowledge base. Currently, we are building seven knowledge bases each for one module. All these seven modules share the same delivery tool. Authoring a new module is to create a new knowledge base. In addition to the knowledge base and the instructional delivery tool, the instructional system includes a student database and a test question databases. The student database stores student records The test questions database consists of all test questions that are used to assess how well the student learns.

The first version of the instructional delivery tool has been completed. The implementation of the three modules: Public Access computer Labs & Electronic Mail, Document Processing, and Data Visualization and Spreadsheets, are nearing completion. All seven modules will be completed by the Fall 1996. The tools used for developing the instructional delivery tool and the knowledge base, student base, and test questions are ToolBook and Paradox.

Conclusions

In this paper, we introduced the instructional system developed for teaching computer and information literacy. The computer and information literacy courseware consists of seven modules, each having a separate knowledge base. All seven modules share the same instructional delivery tool. This system can easily be generalized to a generic tool for teaching any software applications. Authoring a course for a software application is to build a knowledge base. We are currently building an knolwedge base editor. We have also implemented a software simulation tool which can be used to create simulations of software applications.

Acknowledgments

The work reported in this paper is supported in part by a grant from the National Science Foundation, CAD-9414210, and a grant from the Department of Education, FIPSE P116B40729.

References

Fox, R. (1993). Priority Management Report. Vol.36, No. 12, p11.

Merrill, M. D., Li, Z., & Jones, M. K. (1991). Instructional transaction theory: An introduction. *Educational Technology* 31(6), p7-12.

VanLehn, K. (1988). Student Modeling. Chapter 3 in *Intelligent Tutoring Systems*. (Eds) Polson & Richardson, LEA.