

A NEW COURSE OF ACTION FOR COMPUTER ASSISTED INSTRUCTION: THE MACINTOSH-IBM CONNECTION

Nancy Newins*

ABSTRACT

Librarians interested in computer-assisted instruction (CAI) can currently select from more than 100 authoring systems--computer software which facilitates non-programmer development of CAI programs. Selection of an authoring system also necessitates making a second choice--instructional development and program presentation on either Macintosh or IBM PC-compatible hardware. This article focuses on one system, Authorware's *Course of Action*, which, in contrast to others on the market, allows for CAI design and development on Macintosh hardware, with transport to the PC for presentation. The author has found that ease of development on the Macintosh, combined with program presentation on either PC or Macintosh hardware, provides a good solution to the transportability dilemma posed by the hardware limitations of other authoring systems.

Introduction

With the proliferation of articles and professional conference programs on hypertext in the late 1980s, many librarians began to associate hypertext and Claris Corporation's *Hypercard* as the major means of authoring computer assisted instruction. Librarians in an IBM or PC-compatible environment regretted the fact that comparable software for the PC had been slower to develop and seemed to lack the ease-of-use of the

* Reference Librarian and Library Instruction Coordinator, McGraw-Page Library, Randolph-Macon College, Ashland, Virginia

Macintosh products until the recent availability of Toolbook, a hypertext software for the PC users.

Librarians interested in designing a library tour or a more advanced computer assisted instruction program should know that hypertext software is not the only authoring tool available. More than 100 authoring systems are now on the market, enabling non-programmers to develop quite sophisticated computer-assisted instruction tutorials and simulations. Unlike hypertext software, the newer authoring systems also include many pre-programmed instructional features, such as a variety of feedback and branching options in addition to student record-keeping and other course management features.¹

How does one choose from the baffling array of authoring systems? Like hypertext software, is selection invariably limited by available hardware? Do any systems allow CAI programs developed on Apple/Macintosh computers to run on PCs, bridging the Macintosh-IBM-PC platforms?

In 1988, through a faculty development grant from Randolph-Macon College, this author investigated these and other questions preparatory to the selection of authoring software for development of a basic library skills tutorial for freshmen.

Background

A change in the Randolph-Macon College Library's physical facility contributed to the decision to investigate the potential of computer assisted instruction. A building addition/renovation project doubled the library's size and added a computer room with eighteen PCs. The new computer room made the use of computer assisted instruction feasible and logistically practical. A CAI tutorial, culminating in an in-library exercise involving hands-on use of library tools, presented an excellent method for stimulating active learning.

Other pedagogical reasons made use of computer-assisted instruction an attractive adjunct to the current bibliographic instruction program. As co-authors Alessi and Trollip note in the 1991 revision of their classic text on CAI, "... the first rule for correctly using or developing CBI [computer-based instruction] is to do so in situations where the computer is likely to be beneficial," and they also list several such situations, including instances where "... student motivation is typically lacking."²

In a 50-minute session for freshman English classes, the reference librarians typically cover basic library skills; for example, rudiments of catalog

use, in addition to teaching search strategy and more advanced reference tools. Students' lack of interest in the basic skills, perceived by many as high school material, makes it difficult for the librarian to recapture interest later in the session.

A CAI tutorial to teach basic skills, independent of and preliminary to the library class sessions, afforded several potential benefits. CAI would engage student interest and motivation by providing for individualized, self-paced instruction. This individualized approach would be especially useful for freshmen with different backgrounds and levels of preparation. Frequent questions and feedback would involve students with the program in a highly interactive way. The variety and visual interest possible with CAI through the use of color, graphics and animation would also provide a more engaging presentation.³ A CAI tutorial could serve as a useful addition to the current freshman library instruction programs.

The next step was an investigation of authoring alternatives for the design and development of the project.

Authoring System Defined

What is an authoring system, and how does it differ from programming or authoring languages, which can also be used to develop computer-assisted instruction?

Programming languages, such as BASIC or PASCAL, can provide a great deal of power and flexibility in CAI lesson development. The non-programmer designer, however, generally needs to add a programmer to the development team to handle the programming of the lesson. Upon releasing the lesson to the programmer, the designer also relinquishes much of the flexibility to make changes as the lesson is transformed from printed script to computer screen. Authoring languages, such as PILOT or TUTOR, are easier to learn than programming languages, but also involve more extensive programming than authoring systems.

Although less powerful than programming or authoring languages, authoring systems are generally easiest to learn and can significantly reduce development time.⁴ Underlying programs prompt designers to enter instructional information on a screen-by-screen basis and control the presentation of the instructional material.

Authoring System Selection Criteria

A preliminary literature search revealed several articles which discussed the criteria for authoring system selection.⁵ The following four factors were determined to be most important in selecting a system for the Randolph-Macon College Library's specific needs:

Cost--Systems range in price from \$100 to more than \$20,000. Given budgetary restrictions, only systems under \$500 were seriously considered for purchase.

Hardware Limitations/Transportability--At the time of selection, R-MC had fewer than five Macintosh computers on campus. As an IBM/PC campus, only PC-compatible systems could be considered unless a system could transport the CAI tutorial to a PC for presentation.

Instructional Features--Basic authoring features would be satisfactory for the library's initial experimentation with tutorial development. Power was judged less important than ease of use and maximization of training and development time.

Developer's Experience--As a novice to CAI development, this author looked specifically for a system with good documentation that was relatively easy to learn.

Field Research

In an article on authoring system selection, Craig Locatis advises that, after first identifying systems that seem appropriate to local needs, "viewing demonstrations of these systems by experts," is a second and equally important step.⁶ This type of review and consultation proved especially beneficial, not only in identifying the most commonly used systems, but also in determining which systems would be appropriate for the library.

Through a local computer-based training users' group consisting primarily of corporate trainers and instructional designers, several authoring systems were made available for demonstration. The advice and recommendations of these experienced instructional designers also proved most helpful.

More importantly, a faculty development grant enabled this author to spend six weeks during the summer of 1988 at the University of Minnesota--Minneapolis Campus studying a variety of authoring systems and programs and consulting with system users. The University of Minnesota-St.

Paul Campus Libraries had just completed an interactive video program to teach library skills to freshmen, and consultation with the educational technology professor and the instructional designer on that project was especially beneficial.⁷

Course of Action

In the process of reviewing authoring systems at both University of Minnesota campuses, the transportability problem soon became apparent, as systems available in 1988 for the PC were compared with the new, icon-oriented authoring systems recently developed for Macintosh computers.

Icon-based systems consist of several specific icons, or pictographs, representing types of author input--a small screen representing a display screen, or a question mark representing a question screen, for example. The CAI developer, by positioning icons in the appropriate instructional sequence, forms the building blocks of the instructional program. By clicking and opening each icon, the author can simply add the desired information--the specific screen design or the specific question. As Alessi concludes: "Icon-oriented systems generally are the easiest type of system to use, yet come close to code-oriented systems in power . . . we believe that they are the best kind of authoring tool for the novice CBI developer."⁸

An icon-oriented authoring system for the Macintosh would meet the library's need for an easy-to-use system that would additionally provide a wide range of instructional features. Comparably lower in cost in 1988 than PC authoring systems, these systems could also meet the library's cost criterion. The hardware/transportability issue remained, however. How could a CAI program developed with an icon-oriented authoring system for the Macintosh be transported to the PC environment for program presentation in the library computer room?

Fortunately software which addressed this connectivity issue was being beta-tested by the Minneapolis-based company Authorware, Inc., in 1988. With the *Course of Action* authoring system, now upgraded and marketed as *Authorware Professional*, CAI programs designed and developed on the Macintosh could be transported to the PC for presentation through special transport software.

In addition to solving the transportability problem, *Course of Action* not only met but far exceeded the library's other selection criteria. Several features made this authoring software an excellent choice.

The authoring system assures an organized course structure and provides a flexible and easy-to-learn authoring approach for the novice CAI developer by the framework of a course flow line, or flow chart structure, along which icons are positioned. Additionally, by adding start and stop flags along the course flow line, small segments of the course can be run and tested, allowing for interactive debugging and ease in making changes to the course.

The following example shows the basic course flow line for the library's tutorial, "Solving the Library Puzzle," with flags positioned to run the introductory animation sequence. Each individual icon, positioned along the left side of the screen, provides a specific design function: display, animation, erase, pause, specified sequence of displays, question sequence, calculation, or map (to group a series of icons), respectively.

By selecting "Try It" from the menu at the top of the screen either the whole course or a specific segment of the course can be run, with changes made as needed (See Figure 1).

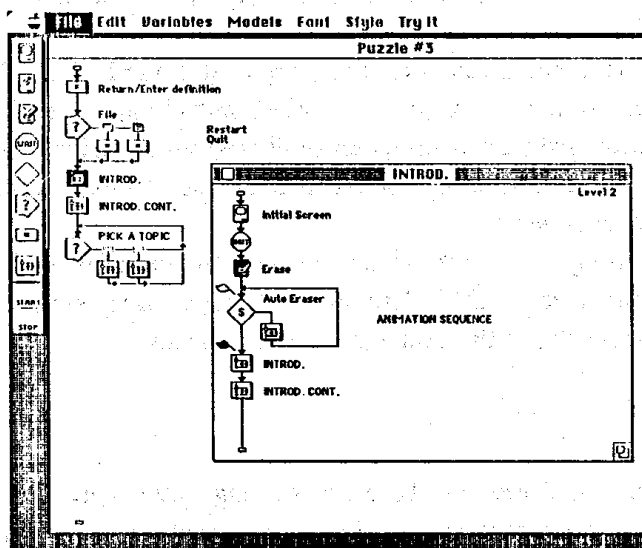


Figure 1: Course flow line for "Library Puzzle" showing icons, menus and multi-level structure (with icons for segments of the course grouped within "map" icons).

The authoring system also provides a full range of instructional features, including various branching and student response options, student record-keeping capabilities and the provision for incorporating other external programs. Graphics capabilities include animation, a video interface, sound,

and color. Finally, the simplicity and clarity of the documentation aids immeasurably in maximizing learning and development time.

Results of Transport

The transport of the library's tutorial "Library Puzzle" from the Macintosh SE on which it was developed to the PC was accomplished in a matter of minutes through Authorware transport software and a connecting cable. The transport can also be accomplished by disk. Overall transport results were quite good, with no distortion of text or graphics on a PC with VGA monitor.

A few problems were noted. Due to the memory limitations of the *Course of Action* software at the time, courses developed had to be limited to 640K of memory for presentation on the PC. Since "Library Puzzle" as developed on the Macintosh was initially much larger than 640K, the course had to be subdivided into two sub-files for transport and presentation of the PC. The final version of the course was ultimately cut to fall within the 640K memory restriction.

It is recommended that software be run on a hard drive. The PCs in the library's computer room are networked and do not have individual hard drives, thus the program had to run on floppy disks, which slowed both run and response time. However, the original *Course of Action* could not be made available on a network.

Also, although a minor inconvenience, the tutorial could not be edited after transport to the PC. Any corrections or changes needed to be made on the Macintosh and the course transported again.

Conclusion

Authorware's *Course of Action* authoring system provided a good solution to the transportability problem imposed by local hardware restrictions at Randolph-Macon College. Ease of authoring on the Macintosh, combined with presentation on the PC enabled the library's tutorial developer to combine the best of both hardware platforms. The library skills tutorial developed with *Course of Action* was used in the library's computer room this fall in the orientation of 340 freshmen to the library.

As one author of a current text on authoring instructional software notes: "Transportability of software across different systems is a major issue in the instructional domain. After spending a lot of time and money

developing a program, it is desirable to have it run on as many different computers as possible."⁹ Authorware, Inc. has upgraded *Course of Action* and is now marketing both a Macintosh and a PC version as *Authorware Professional for the Macintosh* and *Authorware Professional for Windows* respectively. One can only hope that more software developers will follow suit in producing authoring systems that address the transportability issue.

NOTES

1. *The 1987 Guide to Computer-Based Training* (1987): 5.
2. Stephen Alessi and Stanley Trollip, *Computer-Based Instruction*, 2nd ed. (1991): 6.
3. Robert Price, *Computer-Aided Instruction: A Guide for Authors* (1991): 3-5.
4. *Ibid.*, 241-242.
5. Alessi and Trollip, 346.
6. Craig Locatis, "Selecting Authoring Systems," *Journal of Computer-Based Instruction* 12, no. 2 (Spring 1985): 28-33.
7. Barbara Kautz, et al., "The Evolution of a New Library Instruction Concept: Interactive Video," *Research Strategies* 6, no. 3 (Summer 1988): 109-117.
8. Alessi and Trollip, 344-345.
9. Greg Kearsley, *Authoring: A Guide to the Design of Instructional Software* (1986): 26.