Microware Review

Authorware Professional: Part 1

Len Proctor

Best Course of Action, now known as Authorware Professional, is a design system and a courseware authoring system that has been developed for the educator-author who wishes to branch out into the design and development of computer-based instruction. Contrary to other similar systems, it is not necessary to start building course structures in any particular sequence. Instead, with the help of design icons, authors can begin where they feel comfortable and move backward or forward in the course in whatever manner they feel is appropriate. Authorware Professional automatically maintains the hierarchical structure necessary for easy course documentation, maintenance and modification. The ability of this courseware authoring system to manage "human variables" is a significant step forward in facilitating the local development of computer-based instruction.

Authorware Professional runs on a Macintosh Plus, SE, or any of the II series computers that have been equipped with a minimum of 1 Mbyte of RAM memory for monochrome systems or 2 Mbytes of memory for color systems. Two 800K drives are sufficient for smaller courseware development projects but a hard disk drive is definitely recommended for modest to large projects because of the ease and increased speed with which files may be accessed. Special hardware has been included in the package for sound digitization, but it is not required for sound playback. In addition, an advanced animation module, videodisc, CDI, and CD-ROM interfaces, and cables necessary to port courseware over to an MS-DOS platform have also been provided in the system package. One major advantage of Authorware Professional now has over its previous versions is that special runtime student course disks are no longer required for course packaging. Courseware authors can now prepare their own student disks. While the purchase price for professional courseware developers has been adjusted accordingly, educational institutions can still obtain the program a substantial discount. Author-ware Professional is available from Author-ware Inc., 8500 Normandale Lake Boulevard, Ninth Floor, Minneapolis, Minnesota, 55437.

CJEC, VOL. 19, NO. 2, PAGES 157 · 161, ISSN 0710 · 4340

One key question to be asked when reviewing any courseware authoring system is: "How well does the authoring environment facilitate or constrain the educator-author in the overall design and development process?" The collaborative evaluation model, developed by Collis & Gore (1987) to help answer this question, rests on the assumption that if an ". . .integrated set of software engineering, instructional design and collaboration principles can be taken as an appropriate basis for educational software development, we claim it is also reasonable to use the same set of principles as the theoretical foundation of an evaluation model for authoring environment..." (Collis & Gore, 1987, p. 14)

Part one of this review uses the principles identified by Collis and Gore (1987) as the basis for evaluating Authorware Professional. Part two of this review, to be published in a subsequent issue of this journal, will present the unique features of this authoring system which help to make it a flexible and versatile, stand-alone courseware development tool.

Principles Reflecting the Needs of the Software Engineer

The first question asked by Collis and Gore (1987), 'Does the system stimulate or encourage adequate attention to design before using the authoring package for technical implementation?', is answered the program's Reference Manual. Authors are encouraged to begin Using Authorware Professional in the design phase of courseware development in order to explore design ideas and experiment with a variety of options that are available. It cautions the author against attempting to complete a pencil-and-paper storyboard or flowchart before experimenting with course structure. It also cautions against trying to visualize all possible student interactions before creating a few of them. Authorware Professional has been developed as a design medium as well as an authoring system. Designs can be developed and responded to, just as a student would, in less time than it would take the author to write out the specifications necessary for a programmer to implement just one small teaching scenario. In building Authorware Professional, one of the guiding principles was 'Do not penalize (an author's) lack of planning." Good ideas should be able to be taken advantage of as they arise, even if it is in the middle of courseware implementation.

Does the system stimulate or require the developer to produce the necessary and sufficient amount of documentation for each phase of the development process? Authorware Professional has addressed this most perplexing of courseware development tasks by automatically generating all of the print documentation necessary to support the courseware design. In addition, the course documentation is instantly updated whenever a design change is made. Print options include a complete course map, a listing of all cross references by sequence, title and page number, all of the text and graphics used in the screen displays, and a cross-referenced table of any variables that the author may have developed as part of the courseware design.

For a single-developer, does the system emphasize a distinct time for design and evaluation as opposed to programming and compiling? No. One of the very attractive features of Authorware Professional is that the execution of the lesson mode can be interrupted at any time to correct text, adjust a graphic, or add/delete a frame, without losing the context of the lesson. This feature avoids the usual division between author mode and student mode and encourages author revision or refinement of the screen displays as they are encountered. Authoring computer-based instruction is, at best, a time-consuming process. This feature alone significantly helps to reduce an author's courseware development time.

Can the various components of the courseware be developed and tested independently, and linked together when convenient? Can components (modules) of the program be reused as parts of other programs? Yes. Authoring productivity is seen by Authorware Inc. as beingrelated to the ability to use and reuse basic models. A typical model consists of the logic for implementing a courseware component such as a menu or a paradigm for a particular question type. Having a library of these examples readily available can help new users learn how to use the system more quickly and save the experienced authors the time because they do not have to reinvent the model from the beginning. By using the special start/stop flags provided in run mode, models or components of the courseware can be independently developed, tested, or modified before being incorporated into the courseware.

Does the system produce software that can be conveniently altered or extended in response to specific student needs? Does the system facilitate the development of a prototype of the extended program, for evaluation and refinement of design decisions? Yes. Any screen display developed using Authorware Professional can be added to the courseware, deleted from the courseware or altered at any time in order to respond to specific student needs. Trial prototypes can be developed, tested, refined and expanded without losing any of the original effort expended in the development of the prototype.

Instructional Design Principles

Does the system, especially through its documentation, emphasize to the developer that decisions involving basic educational needs, objective and user characteristics must be adequately addressed before other decisions are confronted? No, it assumed that the author-educator is thoroughly versed in the principles of instructional design. No reference is made to any ID model in the tutorial lessons that have been developed to help the new author learn how to use the authoring system. Asubsidiary publication, now called Authorware Magazine, does publish some articles that offer guidance in the area of instructional design.

Does the system allow data to be treated independently from the main problem? Yes, for example, question banks can be developed and called at will, in any order that they are required.

Does the system unduly influence the designer, or constrain him from implementing his methodological decisions? No. Text and graphics can remain on the screen while input from the student is being generated, or the screens maybe cleared. If the author wishes, students may review one or more previous screens or skip one or more screens forward. If authors find limitations in the program that infringe on the chosen methodology, they may choose to append their own XCMD's and XFCN's or suspend courseware execution altogether and exit to another program. After the activity has been completed, students may then jump back into the courseware at the point where they left off.

Does the system support the variety and quality of feedback appropriate for the intended user? Does the system support the capture of information on student performance that the teacher defines as useful for subsequent analysis? Yes. Random, sequential, and data-driven branchingoptions areavailable. Over 100 system variables and functions may be used to track, monitor and record student activities and responses to quizzes. Data collection is automatic and may be used to generate reports. Performance data from multiple users may also be collected and used to generate a summary analysis report.

Collaboration Principles

Does the system facilitate iterative and on-going compromises between various participants in a collaborative environment? Which of the participants in a collaborative team is the intended user of the system -the educator or the computer scientist? Is the system appropriate for the intended user? How much training is required for the user of the system before it can be used as a facilitating tool rather than a focus of attention and effort? Authorware Professional is an authoring system, design tool and programming environment. Educators, instructional designers and computer scientists, who are involved in courseware development, would each have an opportunity to put their specialties to good use. While the system is designed to attract a novice author, it is sufficiently powerful to offer an expert author an excellent courseware development environment.

For anyone familiar with using a Macintosh computer, the tutorial supplied to assist authors in learning how to use this courseware development system is easy to follow. The new computer user however will first have to learn to use the Mac, or they may find the tutorial difficult. On the other hand, the power user may choose to skip the tutorial and go directly to the Reference Manual. While the independent study option may be the only option available for learning how to use the program, a two or three day workshop setting, under the guidance of a knowledgeable leader, should be sufficient time for almost any entry level author to become familiar with the basic operation of Authorware Professional. The assumption here is that authors are successful teachers and/or instructional designers who wish to transfer their skills to a new learning environment. The focus of Authorware is to provide a powerful courseware development environment, not to teach the author basic pedagogy,

Conclusion

In summary, Authorware Professional has met the software engineering principles and instructional design principles set out in the collaborative evaluation model outlined by Collis and Gore (1987). It does not teach instructional design, but supports the needs of instructional designers. Similarly, it does not teach computer programming, but supports computer scientists who wish to use their programming skills. Finally, the educator-author who wishes to work alone may do so, but the program is very amenable to a collaborative or collective approach to courseware development. Anyone who is about to engage in the process of courseware development would be well advised to give this product very careful consideration. In Part Two of this review, we will take a closer look at the special authoring features provided by this powerful tool.

REFERENCES

Collis, B.A., & Gore, M. (1987) Evaluating educational software authoring environments using a model based on software engineering and instructional design principles. ***Canadian Journal of Educational Communication,*** 16(l), 11-21

COLUMN EDITOR

Len Proctor is an Associate Professor in the Department of Communications, Continuing and Vocational Education at the University of Saskatchewan in Saskatoon, Saskatchewan, S7N OWO.