UNIVERSAL AUTHORING SYSTEM: ROUND TWO - THE WEDGE

Andrew E. Andrews, Lois Spangenberg, and Mary S. Trainor
Cognitive Systems Engineering Group, A-6
Los Alamos National Laboratory
Los Alamos, New Mexico 87545
(505) 667-0990

ABSTRACT

Recent papers in the literature have proposed a universal authoring system. While these papers are valuable because they stimulated debate and provided a baseline, pursuing a universal authoring system at this time is a search for a solution before the requirements and problems have been clearly identified. This paper explores several issues related to the concept of a universal authoring system and concludes with an alternative prescription both for users (customers) and vendors. The user prescription includes a clear definition of requirements and establishment of internal standards. The vendor prescription includes working with the users more closely to aid in system comparison. This task is very difficult now because of nonstandard criteria used by the scores of vendors involved. A model for a "universal" authoring system is presented to illustrate that the options are endless. Technical issues regarding the computer medium are discussed to illustrate the inherent difficulties of achieving universality of authoring without restricting progress in hardware. The authors agree that the plethora of authoring systems on the market today inhibits courseware portability, but we feel that our free enterprise system as well as more informed consumers will help reduce the number of surviving authoring systems.

INTRODUCTION

One of the simplest tools devised by man is the wedge or inclined plane. It permits the lifting and moving of objects by changing the force vectors being applied to the object. The wedge enables man to exert less effort over a period of time or over a given distance to achieve an otherwise unattainable goal. Today the wedge is still in use -- overtly in things such as ramps and covertly hidden in many of the complex tools used in everyday life. This simple tool demonstrates first principles of mechanics; and in its simplicity, it contributes to the engineer's tool kit. In this paper, the wedge concept will be applied to the current state of authoring systems and to the idea of developing an authoring system that can be treated as universal. **In short, the field must still develop its first principles before the universal authoring system (UAS) can be built.**

In the papers of Jensen (1987) and later Jensen and Stonge (1987), the value of a universal authoring system is argued, criteria are established, and PMS, the forerunner of the Electronic Information Delivery System (EIDS) Assist authoring system, is compared against the criteria. While these papers are valuable because they have stimulated debate (as evidenced by this paper), pursuing a universal authoring system at this time is a search for a solution before the subject area has been sufficiently understood. At this point in the early evolution of this field, a universal authoring system might impede advances.

To support the premise that a UAS is premature, this paper will explore several issues. First, authoring systems and their purpose will be described in the context of computer-based training. Second, another model for a "universal" authoring system will be presented to illustrate that the options are almost endless. Next, the domain of instructional design and learning strategies will be covered, focusing on the opportunities available for using the computer in the learning process. Fourth, some technical issues regarding the computer medium will be discussed to illustrate the inherent difficulties of achieving universality of authoring without restricting progress in hardware. Finally, some philosophical and practical issues for business will be proposed, providing an alternative prescription.

ABOUT AUTHORING SYSTEMS

The authoring process is defined as the creation, programming, debugging, and testing of courseware. This creative process is complex enough in itself, but adding the dimension of interactive videodisc and audio makes it even more complex. Combined with the fact that the field is still in its infancy, the process cannot yet be standardized. Although many schematics have been drawn to describe the steps in the process (for example, see Floyd, 1982; Held, Aggen, and Reeves, 1986), none has been established as a standard. In turn, no one authoring system has emerged as a standard. More than one hundred authoring systems are on the market today (Stein, 1987). The demand is not great enough to support this many systems, but the number is indicative of the intense interest in and differences of opinion on the subject of authoring.

The purpose of an authoring system is to support the authoring process. The author needs the creative freedom to produce courseware that reflects his/her ability to exploit the computer medium in useful ways. Of course, not all authors are employed to be

creative. Some are chartered with the task of emulating previous designs to produce a cost-effective product in a highly competitive environment. As a result, many courseware companies have developed their own authoring systems and marketed them. After all, what is good for them clearly has commercial value for others. Consequently, commercially available authoring systems reflect the individual vendor's perceptions of the requirements of an authoring system, the nature of CBT, and the role of the CBT developer. The great diversity of authoring systems currently available attests to the fact that a wide range of approaches to authoring and instructional strategy exists.

In the April 18, 1988, issue of *Time* magazine, writer Gene Roddenberry is quoted as saying, "We must learn not just to accept differences between ourselves and our ideas, but to enthusiastically welcome and enjoy them. Diversity contains as many treasures as those waiting for us on other worlds. We will find it impossible to fear diversity and to enter the future at the same time." The same statement can apply to authoring systems. Many creative, intelligent people are building tools useful for CBT development. It is a challenging problem, one that is not solvable through a single iteration and not likely to offer a single, universal solution. Instead, the solution will evolve slowly and laboriously.

Lessons can be learned from the emergence of the IBM PC as a standard. After Apple demonstrated that the microcomputer was a valuable reality, microcomputer technology emerged and many companies joined in the race, producing different versions. As a result, potential customers became confused about what they should buy, and indeed, about whether they should buy. However that stage was short lived. The IBM PC became a *de facto* standard, and many companies assumed the task of emulation. Concurrent with this hardware standardization, software also began to standardize. Today one can purchase a multitude of software packages that run on Intel 8088/80286 chips under MS DOS.

So far, no standardization of hardware exists for CBT. CBT is presented on, *inter alia*, PCs, the Macintosh, VAX minicomputers, and IBM mainframes. Nor are there standards for videodisc players. There is even more variety in the graphics boards used for CBT. Some CBT includes CD ROM and digital audio; most do not. The Army has taken a step toward hardware standardization with EIDS. It is too early to draw conclusions about the viability of this recently introduced system; at any rate, much of the CBT world is outside of the Army arena. It is premature to force a universal authoring system on developers while the hardware environment is still searching for its standards.

At this point, a distinction needs to be made between a standard authoring system and a universal authoring system. The word "universal" means present everywhere, whereas the word "standard" is much weaker, meaning something established for use as a rule. Many companies solve the courseware portability problem by establishing one authoring system as an internal standard. EIDS Assist is a standard authoring system. The Army has made a decision that it will use EIDS Assist and, as a result, accrue cost-benefits. It is being adopted to meet specific needs and is not intended to be universal. If established, a UAS must meet everyone's needs - both now and in the future. Accordingly, when one asks if a given authoring system meets the criteria of a UAS, one is asking if the authoring system developer has, in fact, finally been able to define CBT in totality.

Merrill (1985) describes drawbacks of existing authoring systems: lack of flexibility, slow execution speed, costly royalties, limited instructional design options, and frame-oriented presentation and interaction. Several of these topics are obliquely discussed in this paper, but two, speed and royalties, deserve specific attention here.

- Slow execution speed. Authoring systems are often so large that they fill much of the computer's memory, requiring the courseware itself to reside on the disc and be interpreted in small segments. The more features an authoring system has, the more memory it occupies; the amount of lesson material that can be placed in memory becomes progressively smaller. Disc accesses become more frequent, and execution slows down. By definition, a UAS could be expected to incorporate many features. One would, therefore, expect its performance to be poorer than that of smaller authoring systems and much poorer than that of authoring languages. While the speed problem could be solved through an authoring system that produces executable or compiled code, this solution represents another step that may preclude nonprogrammers from easily using the system. Of course, the system would then be less than universal.
- Costly royalties. One assumes that a UAS would not involve any royalties, which is an issue that does need to be raised as distribution of CBT courseware is a requirement for almost all development installations. Most existing authoring systems do require royalties for distribution. To highlight the complexity of the issue, the March 1988 issue of the *Videodisc Monitor* states: "EIDS Assist was developed under contract by CSC (Computer Sciences Corporation). However, the Army owns complete distribution rights to EIDS Assist." CSC markets a comparable system to the commercial market. If the government were to establish a UAS (let's say it was EIDS Assist), it would be in the public domain. If the private sector wanted to market the UAS, perhaps with improvements, would it remain universal? And, as improvements are suggested (for example, templates for a new instructional strategy), to whom would they belong, who approves them, and how is the UAS revised and redistributed? Does it have to be done at "world headquarters"?

These two topics highlight the central problem of a UAS. If a UAS is truly universal, it must do everything for everyone. In the ultimate, suppose Mr. Spock says to the computer, "Create a course on the value of the United Federation of Planets' political foundation that will convince the Klingons of the

value of peaceful coexistence." Of course, the Star Trek computers can handle the task because they have the capabilities provided by a UAS, as well as a relevant database system to draw upon. However, in today's world, computers that can be used for authoring and/or presenting CBT are limited in capacity. Decisions are required about what is to be included in the UAS. Who makes such decisions? An American National Standards Institute (ANSI) committee could be initiated to make such decisions, but such action seems premature. ANSI standards have never been directly involved in CBT before and perhaps with good reason.

THE ENDLESS POSSIBLE OPTIONS FOR AUTHORING: AN ILLUSTRATION

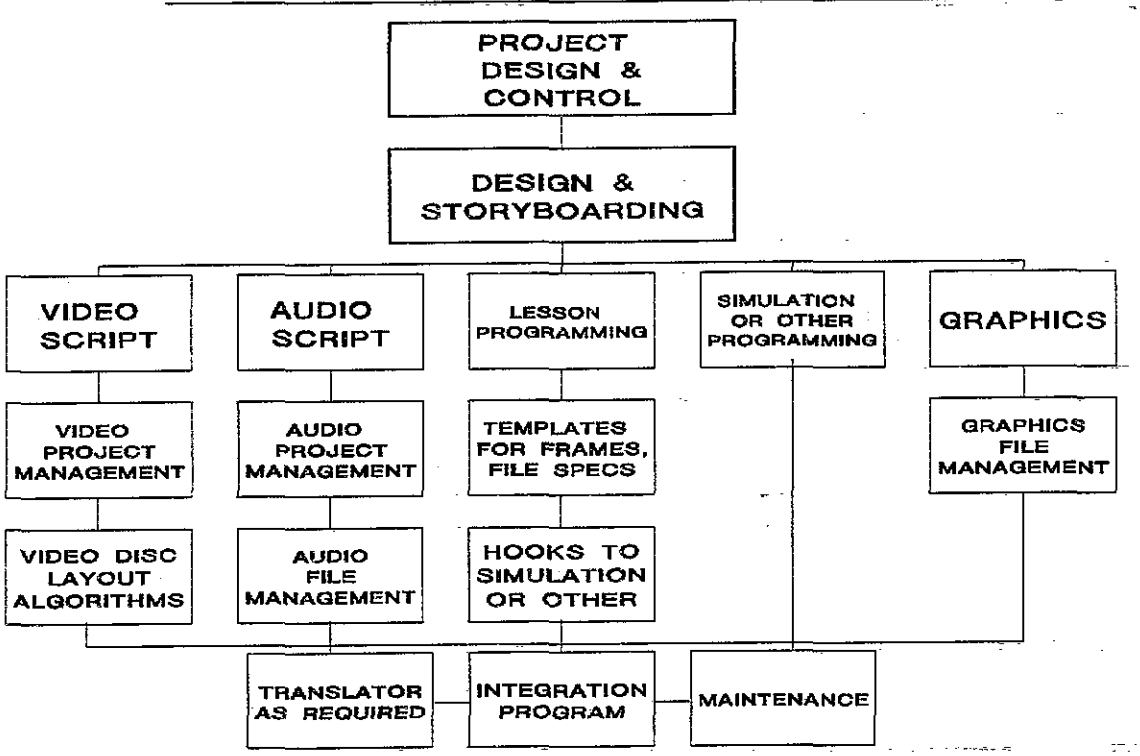
Fig. 1 shows a representation of some functional requirements for an authoring system which could be considered "universal." If one asked experts in the field to evaluate it, it would be judged as flawed and incomplete. As many opinions about functionality are available as there are creative people in the world.

In the figure, everything is networked so that as video frames are adjusted in the video software, lesson programming is automatically updated. In addition, as simulation evolves, hooks and commands to the lesson programmer are automatically created, as well as mini-CBT programs, to teach the lesson programmer how to access and use the simulation. In the long run, expert systems could possibly be automatically generated from the knowledge database which is extracted from experts automatically by the computer.

This model for an authoring system does allow for expandability. In the existing climate, many institutions are using an authoring system because there are no programmers on the staff. After a successful first CBT effort, it often becomes clear that the technology is here to stay and that the boss wants bigger and better CBT. Programmers are added to the staff. These programmers could use the supported authoring system if it had the capability of jumping out to externally generated programming language code and then jumping back into the system after the code has been executed. This feature enables the developer to take advantage of the tools of the authoring system at the same time as retaining the power of a programming language. Without this capability, one would get locked into an inflexible, unexpandable system.

This last feature, links to a programming environment, ensures that CBT developers can maintain their own style. In fact, one can envision an add-on market for additional authoring products in much the same way that many add-ons and templates are available for use within the Lotus 1-2-3 environment. Nevertheless, the UAS is now being changed in a decentralized process, consistent with free enterprise. The authoring system is no longer universal, and world headquarters must approve which add-ons will be incorporated to return to a UAS. The question again arises: Who is in charge?

Figure 1.
AN EXAMPLE OF THE ICW DEVELOPMENT PROCESS



INSTRUCTIONAL DESIGN AND STRATEGIES AS RELATED TO AUTHORING SYSTEMS

Instructional design, including scripting, storyboarding, and screen design, is the most creative aspect of the development processes. Therefore, complete automation cannot be a goal, and yet certain parts of the design phase can be facilitated through automation. Most developers generate the script on a word processor or with software tools, such as Hypercard. Graphics are described in words and then hand drawn as necessary. The script is reviewed by the project team several times prior to actual programming or use of an authoring system. The word processing system is often incompatible with the actual delivery system so that everything the designer generates must be reentered by the programmer. An authoring system should be capable of eliminating this step, allowing for direct use of the script and storyboards. Trying to automate the actual instructional strategy used and the screen designs would be like automating the creation of artwork, obviously an unreasonable authoring system requirement. However, tools to assist the human author in this process are a reasonable, indeed desirable, feature.

The instructional strategies used also greatly affect the nature of authoring. Many currently available authoring systems provide limited strategy options, mainly tutorial and drill. Tutorial and drill are strategies that are only appropriate for a particular type of training. The cognitive movement in instruction, along with greater demands for transfer of training, has resulted in an intense interest in simulation and gaming strategies. These strategies, however, might be very difficult to accomplish with the UAS.

The only type of computer program that might eventually be capable of assisting the author in creating a design appropriate for a specific application is an expert system. Such a system does not, however, exist as yet. One reason for this is that expert systems require a well-defined knowledge base. No such knowledge base can be delineated because of the current state of rapid evolution of the CBT field. M. David Merrill of the University of Utah (1987) is building an Expert System for Instructional Design. This system focuses on straightforward issues such as recommendation of the most appropriate media implementation and the number and type of modules required. It does not attempt to address the more difficult area of screen design and specific instructional strategies used in the frames and strands.

Authoring systems focus on the needs of the teacher rather than the learner. Their main intent is to facilitate the development process so that CBT is more cost effective. Yet, CBT should only be used to improve the quality of instruction, with the ultimate goal of improving human productivity. Each year educational and cognitive science researchers are producing studies that demonstrate new pedagogical approaches to improve instruction. CBT designers are obliged to put this research into practice. Intrinsic to authoring system design is a decision process regarding the instructional strategies to be implemented by the system. Therefore, the strategies authoring systems are able to accommodate become limiting and outdated. They must be augmented with the best approaches developed by the research community. However, who decides when a new approach is ready? Who is in charge?

UNIVERSALITY AND ITS POTENTIAL IMPACT ON HARDWARE: AN EXAMPLE

Universality, as defined by Jensen, implies the capability of transporting a lesson developed for one computer delivery system directly to another. This concept implies that the authoring systems should have the ability to produce object code for any microprocessor (or even a Cray, if that is the delivery system) and that component- specific graphic files should be readily translatable from one system to another. The difficulty of changing a program from an Intel 8086 environment to a Motorola 68000 series environment is evidenced by the many applications commercially available for each but the very few available for both.

A more intuitive understanding comes from the graphics issues. Consider two systems in which the screen resolution is the same (for example, 640 X 400 pixels) but the color selection available (palette) differs. System 1 supports the use of 16 colors from a palette of 64, while System 2 permits the display of any 32,768 colors. In System 1, the color of each pixel requires four bits of information to uniquely define each pixel's color. System 2 requires 16 bits or 2 bytes of information to store the color of each pixel. Translating to the size of full screen graphic files, System 1 needs 128 Kbytes per file while System 2 needs 512 Kbytes.

The implication of these numbers is that a tradeoff occurs between color fidelity and system constraints, underscoring that CBT is an art based on science. More importantly, a universal authoring system must support both options and facilitate the movement of courseware (including object code) between the systems. Going from the 16-color palette to the 32,768 color palette is easy; going the other way is not. A good example of an application here is a multicolor map.

Fig. 2 represents a 16-color palette in an RGB space while Fig. 3 shows a 32,762 color palette in the same space. Decimal as opposed to binary numbers are used to facilitate understanding. Assume that Point A in each figure represents the same color. In this case, going from one system to another is simple: multiply or divide the coordinates by the number 3 as required. On the other hand, consider Point B in Fig. 3. Using this simple algorithm, Point B translates to (2,2,2) in Fig. 2's palette. Further, assume that in Fig. 2, only 2 of the possible 16 color choices have been defined: A at (3,1,1) and C at (1,3,3). To which color do you map color B: A or C? Color B has equal components of red, green, and blue (RGB), while color A has a red dominance and color C is strongly green and blue. Of course, Fig. 2's palette has more choices (14 more), but the problem of mapping still remains: numeric algorithms are generally not satisfactory.

A universal authoring system (or world headquarters) could solve this problem by decreeing the number of colors and the required palette size. However, if the number is too small, the medium will be unnecessarily encumbered. And if the number is

FIGURE 2.

A COLOR MAP FOR A 16-COLOR PALETTE

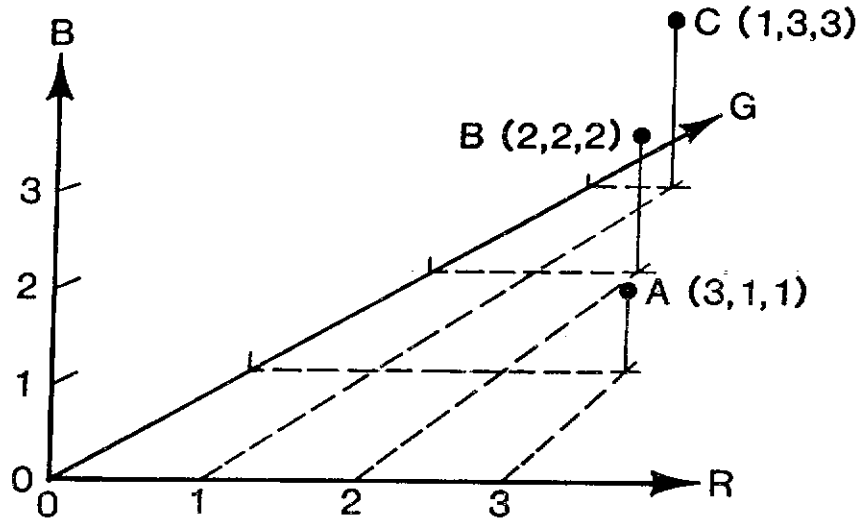
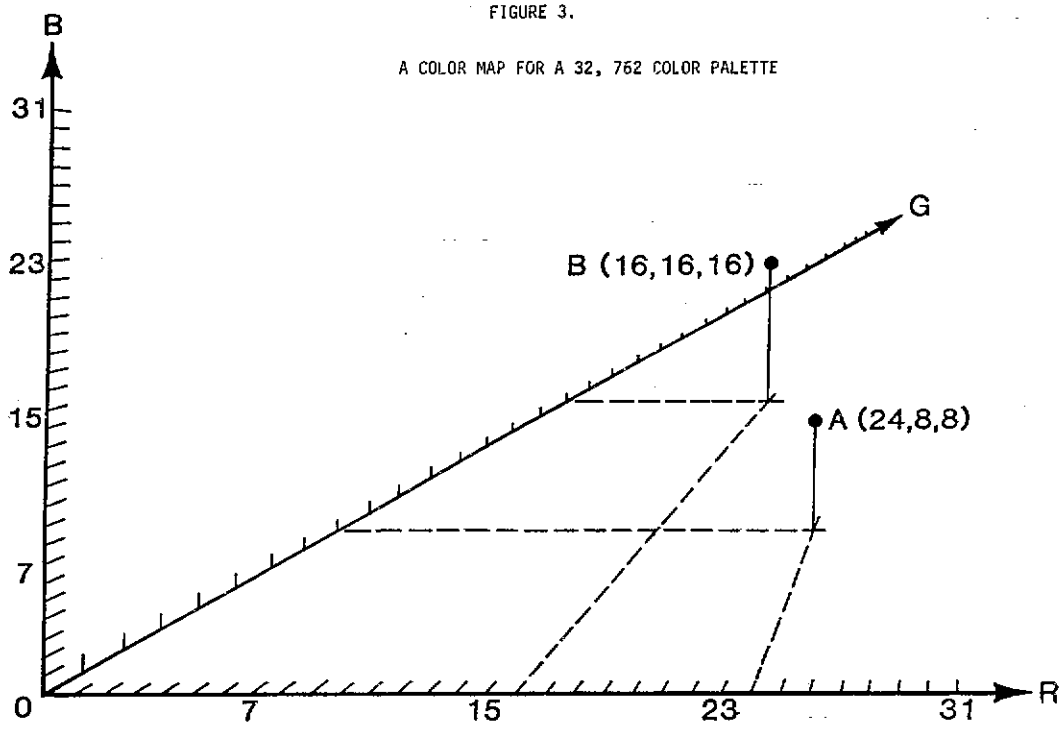


FIGURE 3.

A COLOR MAP FOR A 32, 762 COLOR PALETTE



too large, free enterprise will be less than free. Yet, if one believes in free enterprise, the situation will be self-resolving based on a compromise between cost and benefit and a UAS will not be needed.

AN ALTERNATIVE PRESCRIPTION

Jensen's UAS was a very good snapshot of the CBT community at the time it was written and will serve as a good baseline for several years. Nevertheless, most CBT developers cannot, do not, or should not want to use everything on the list. Success in a CBT program, as success in most fields, is found in the product's meeting the need effectively. For CBT, the need is teaching. By implication, CBT should be straightforward, using the simplest tools available (wedges) to do the job. Complex design and bells and whistles without purpose detract from the main objective--learning. Hence, why should a CBT developer carry extra baggage in an authoring system?

The alternative that we propose to a universal authoring system is multifaceted.

(1) Standardize the authoring method used. Within a particular institution, attempt to standardize the authoring method used. This method may be an authoring system, an authoring language, or a conventional programming language, but it should be responsive to institutional requirements. If it is an authoring system, it needs to be capable of switching to programming language code and back. For example, such a system was described at the 9th I/ITS Conference (Parks, 1987). This capability will allow for expandability beyond the authoring system capabilities if the need arises. Internal standardization will facilitate communication and portability, but exploration of new, more powerful authoring environments should also occur.

(2) Use a team. Use a team of professionals to design and develop courseware. This team needs to include the following areas of expertise: subject matter, instructional and graphics design, human factors, video/script writing, video and audio production, programming/computer science, and management. Do not expect an authoring system to replace these essential areas of expertise.

(3) Use software tools. Use software tools to aid in design and development as appropriate (for example, *Hypercard, ThinkTank, etc.*).

The limitations of authoring systems as proposed by Merrill (1985) are minimized with this solution. In addition, no danger exists for the author to outgrow his/her tools, a common occurrence with authoring systems and a frequent source of unanticipated costs in courseware development.

This alternative prescription does not disagree with Jensen's premise that portability of courseware is essential and that authoring systems vendors are currently not aiding portability. It does, however, interject some realism into our expectations for the field. Fields emerge in patterns, and we should expect our field to follow the proven Darwinian pattern.

In a free enterprise society, businesses profit when their product is perceived as having value relative to its price. An authoring system vendor must provide the tools required by the consumer. The criteria used to present an authoring system's capabilities are highly variable (for example, Becker, 1987; see also Data Training's CBT Guide, an annual review of authoring systems, April 1987). When the customers try to distinguish between features of systems using vendor information, confusion results. This situation, in turn, motivates them to look for something else, such as a UAS. Vendors need to provide accurate information about their products that permit consumers to judge them against consumer requirements.

An abbreviated listing of some of the alternatives one should use when selecting an authoring system is provided in Table I. This table emerged from a study performed by one of the authors (see Schultejann and Trainor, 1988). Such a table, if standardized among vendors, could reduce confusion among authoring system customers.

CONCLUSION

Two principal messages are being related in this paper, both opinions of the authors. The first is to users: Users (consumers) do not need a universal authoring system, but they do need to define their criteria and select an authoring system that can serve as an internal standard for them. One must pose this question: Does system "XYZ" meet the current needs of my organization? The second message is to vendors: To facilitate selection by users, authoring system vendors need to work with consumers in defining and standardizing criteria. Variations among criteria will always occur because needs will differ, but vendors need to promote a good match. As authoring systems are eliminated, just as word processors or spreadsheets were, only the fittest will survive and the issue of portability will become an historical footnote.

The goal of a universal authoring system is appealing. Identifying the requirements would provide a framework for the fundamental tools of CBT. However, it is unrealistic to expect CBT wedges to emerge quickly because the task is so complex. Instead, users must clearly state (and by implication, understand) their requirements so that vendors can respond with quality and value. The authoring system evaluation criteria provided by us and by Jensen serve as a beginning.

The essence of this article is the following:

The problem of nonuniversal criteria for selection of authoring systems is a different problem from creating a universal authoring system. The solutions are not necessarily the same. One can solve the nonuniversal criteria for authoring systems without cramping the creative initiatives of designers in this relatively new field.

The underlying issue of a UAS is the tradeoff between creativity and universality. Creativity is vital to progress, while universality implies that the domain has reached the state of being mundane, which must never happen to learning.

Table 1. Questions to Ask When Choosing an Authoring System

1. What is the cost per instructor station? per student station? What kind of agreement do I need? If number of student stations is large and undefined at this point, then an agreement for unlimited usage and no special presentation system is probably desired.

2. What is the royalty agreement? Am I willing to pay the vendor for each external usage of my courseware? If unlimited distribution is desired, ask about cost of this up front.

3. How easy-to-learn and easy-to-use is the system? Many vendors say their system is easy, and yet they require extensive training delivered by them. Call current system users and ask them and/or try the system out for yourself to answer this question.

4. What are the hardware requirements? Does the system require some specialized hardware you do not have access to or cannot afford? Some vendors developed their system to run on one set of hardware and then retrofit for others. If your hardware is retrofitted, it may be awkward to use.

5. What support does the vendor supply?: Does the vendor supply telephone consulting (you will need it as you get started)? Are software updates provided? Are the updates designed to be compatible with earlier versions?

6. What graphics, video, input devices and audio are supported?: The power of the technology comes through in the use of high quality graphics (including overlay on video), digital audio (not just audio on videodisc), the ability to use touch or mouse as well as keyboard, and the videodisc. You may not want these now, but you probably will later. Don't bind yourself to an authoring system which is constrained and not keeping up with new technologies.

7. Which instructional strategies are supported?: Many authoring system constrain you to tutorial and drill and practice formats, yet the simulation and gaming strategies may be needed for a particular application. How does the vendor define a particular strategy? The vendor may define it differently than you do as well.

8. How does the computer-managed instruction (CMI) work?: Most systems have CMI, but it may not match your needs. What record keeping and reporting do you require? Will the system require a central file server? Do you have to pay extra for CMI?

9. What response and questioning strategies can be used?: You will want to be able to ask some open-ended questions, as well as multiple choice questions. Will the system allow for this? Does it have a built in spelling-corrector? Can several specific hints be given for different wrong answers, to aid the student in reasoning to the correct answer?

10. How flexible is the interface for different screen designs?: Can you have different types of menus: located at different screen locations? of different sizes? With icons/graphics?

11. Does the system support team development?: Can one person be working on graphics while another working on audio and text for the same lesson at the same time?

12. How easy is it to change and update a lesson ?: Maintenance costs can be high and maintenance is always necessary. What features does the system have to accommodate maintenance?

13. Can it talk to programming languages?: If an authoring system has an efficient way to jump out to code written in a programming language then it provides much greater flexibility than a system which does not have this capability. This is desirable whether you need this flexibility now or not.

14. Can it import courseware written using other systems?: You may need to author or write scripts using other software systems and if you can import those files into the authoring system directly you can save a great deal of time.

REFERENCES

Becker, R. "How to Evaluate CBT Authoring Systems." *Data Training*, July, 1987.

Floyd, S. and B. Floyd. *Handbook of Interactive Video.* Knowledge Industry Publications, Inc., White Plains, New York, 1982.

Held, T., W. Aggen, and T. Reeves. "Introduction to Interactive Videodisc Development," a workshop held at the Association for the Development of Computer-Based Instructional Systems Conference, 1986.

Jensen, L. "In Search of a Universal Authoring System for Lesson Development." paper presented at the *Technology in Training and Education Conference,* Colorado Springs, March, 1987.

Merrill, M.D. "An Instructional Design Expert System," President's Forum presentation at the ADCIS Conference, November, 1987.

Merrill, M.D. "Where is the Authoring in Authoring Systems?," *Journal of Computer-Based Instruction* 12 (4), pp. 90-96, 1985.

Parks, K. J. "Efficient, Production-Oriented CBT Authoring," *Proceedings of the 9th Interservice/Industry Training Systems Conference,* November 30 - December 2, 1987.

Rodenberry, G. "A Letter to the Next Generation." *Time,* April 18, 1988.

Schultejann, A. and M. Trainor. "Selecting an Authoring System 1988: How the Vendors Make it More Difficult," *Association for the Development of Computer-based Instructional Systems (ADCIS). Proceedings,* November, 1988.

Stein, J. "Data Training's Review of Authoring Systems." *Data Training,* Warren-Weingarten Publications, 1988.

Stonge, J. and L. Jensen. "Will PMS Meet the Needs of a Universal Authoring System." *Proceedings of the 9th Interservice/Industry Training Systems Conference,* Washington, D.C., 1987.

ABOUT THE AUTHORS

Andrew E. Andrews is Group Leader, Cognitive Systems Engineering Group at Los Alamos National Laboratory, which designs and develops experimental simulation-based training. He holds an M.S. in Engineering from the University of California at Berkeley and a M.B.A. from Long Island University, and a Master of Military Arts and Sciences. Prior to his work at Los Alamos, he held varied positions in the U.S. Army that included responsibilities for exercise and training development. He culminated his Army career as the senior air defense staff officer at the Supreme Headquarters Allied Powers Europe. Andrews has many publications, principally regarding military doctrine and computer-based training.

Lois Spangenberg is a software engineer in the Cognitive Systems Engineering Group. She is a former math teacher with extensive computing experience, who is integrating her knowledge of education with her knowledge of computers in designing and coding simulation-based training. She has recently developed a sophisticated graphics-based interactive simulation, which can be used for comprehension and application training, as well as for testing.

Mary Trainor is an instructional psychologist for the Cognitive Systems Engineering Group. She holds a Ph.D. from the Group in Science and Mathematics Education at the University of California, Berkeley. Dr. Trainor provides expertise to the group in the areas of instructional design, computer-based training development, knowledge engineering, and human factors engineering. Previous positions were in the computer documentation, computing education, and information management systems areas. She has more than 30 publications, with the most recent being in the area of interactive videodisc training and computer tutors.