

## WILL PMS MEET THE NEEDS OF A UNIVERSAL AUTHORIZING SYSTEM?

James R. Stonge  
Senior Instructional Systems Engineer  
Integrated Logistics Systems  
Aircraft Systems Division of Grumman  
Linda G. Jensen  
Instructional Design Engineer  
Display/Trainer Products  
Electronics Systems Division of Grumman  
Bethpage, N.Y. 11714-3585

### ABSTRACT

At the recent Air Force Technology in Training and Education (TITE) Conference in March, a call was made for a universal authoring system. This call was based on the recent proliferation in hardware and applications software technologies during major acquisitions for training by all of the military services. The TITE presentation by Linda Jensen identified the inconsistencies between various hardware components, applications software packages, and actual training applications as a major drawback to effective and efficient government training programs. Part of the problem has been the cornucopia of "authoring systems" that have been created, marketed, and sold. These authoring systems have had three major drawbacks: they are usually incompatible with other authoring systems, they usually have limited applicability, working only with certain specific hardware and software systems, and, more importantly, they are not truly authoring systems. They are programming systems that assume the materials have already been authored and created. The paper presented at TITE addressed the need for an authoring system that works at all stages of a project, identified the general characteristics required of such an authoring package, and specified the capabilities and mechanisms that must be included. Further, it called for a standard that allows compatibility from one delivery system to another, regardless of hardware and applications software. The U.S. Army has identified a standard that approaches this ideal authoring system, the Production Management System (PMS). This paper compares PMS to the ideal authoring system described in the TITE paper and presentation. All of the requisite characteristics comprising a complete instructional system development tool are summarized, including electronic storyboarding, data management, media production, and lesson programming. The paper then looks at each of the characteristics and identifies the ability of PMS to meet the need. It also identifies any PMS capabilities not addressed in the characteristics of the universal authoring system, and evaluates those characteristics for inclusion in the proposed standard. The paper concludes with an analysis of the overall capability of PMS to serve as a universal authoring system, and specifies what capabilities need to be added to make it fully functional.

### INTRODUCTION AND OVERVIEW

Users of interactive media continue to face problems of incompatible hardware, software and authoring systems. While the problem is shared by all users, the greatest impact is on the Department of Defense due to the proliferation of delivery systems and software developed over time for a variety of projects. Materials developed under one contract cannot be used with equipment and software acquired via a second contract. Of all media, only interactive media lack universal capability for distribution and transfer across users. This incompatibility is an outgrowth of the lack of comprehensive industry standards. Hardware manufacturers and software developers attempt to create new and unique packages, often with the deliberate intent of preventing users from recycling materials developed for one system for use on another. This strategy has been used not only between rival companies, but by major industry leaders when introducing a new product line that is completely different from previously marketed systems.

The only way to alleviate this situation is to create industry standards such as were instituted for such diverse products as videotapes, tires and light bulbs. When standards are in effect, competition still thrives as producers identify new and better products within the framework of the standards. Thus, the consumer does not need to buy new lamps every time a manufacturer introduces a new light bulb. The industry needs a set of standards that can evolve, but that will allow the consumer (e.g. the Department of defense) to continue to use and update existing lesson materials, regardless of changes to hardware and operating software. In addition, new lessons could be run on 'old' equipment and software regardless of the particular

authoring package used to create them.

Two recent events indicate a desire to establish those standards: The US Army identified a standard hardware and software system - the hardware is EIDS, the software is PMS. And, at the recent Air Force Technology in Training and Education (TITE) conference, Linda Jensen identified and proposed a candidate universal authoring capability. In the case of EIDS and PMS, the specification and deliveries of hardware and software are real. The universal system identified is an ideal that has not yet been realized.

This paper addresses a comparison of PMS to this published ideal system. It specifically avoids a comparison of PMS to any other existing authoring system as no other authoring system has been officially identified as a standard. It also avoids making a value judgement, as it is recognized that PMS was not developed in accordance with the ideal system's specific characteristics and capabilities. Rather, this paper is limited to an evaluation of the relationship of PMS to the ideal, and to the steps required to enhance PMS to meet the needs of the users of the STANDARD, or ideal, system.

### THE UNIVERSAL AUTHORIZING SYSTEM

The ideal system must be a universal authoring system. Three definitions are basic to understanding the characteristics of this system. First, interactive authoring must be defined as a system that allows stimulus and response for both the student/user and the interactive media. Secondly, interactive authoring must provide for both the management of instruction and the record keeping of student performance. Finally,

interactive authoring should be defined as the process of creating the total environment of the delivery system. These three definitions address the practical and philosophical differences evident in the viewpoints of student, instructor, and producer.

#### PRODUCTION MANAGEMENT SYSTEM

The US Army, in conjunction with Computer Science Corporation (CSC), developed the Production Management System (PMS) software package to use with interim EIDS (Electronic Information Delivery System). PMS is the only authorized software system for interim EIDS. Further, it is being converted for use with the production EIDS, and will be renamed "EIDS-ASSIST." For the purposes of this discussion, PMS will refer to both PMS and EIDS-ASSIST. The version of PMS used for the analysis is version 3.11, because it was the latest version available, and Army sources state that it is identical in capability to the first version of EIDS-ASSIST.

#### HARDWARE

It has become a popular attribute for successful commercial software applications to be usable with a variety of computers and peripherals. Sometimes this is done by "installing" the software in which the particular equipment is identified during initiation. In other cases the software is sold for particular configurations. In many cases the product of these packages is transferable from one system to another, via modem, disk transfer, or other media. This same concept should be applicable to authoring.

#### IDEAL

The ideal authoring system must be able to operate regardless of the hardware configuration of either the development station or the delivery station. This is critical if the DoD is to obtain maximum benefit for existing and future resources. Therefore, the ideal will be able to meet the following requirements:

- o Use any brand of videodisc player
- o Use any and all videodisc related peripherals
  - Compressed audio encoders/decoders
  - Compact disks (CD ROM or CD/I)
- o Accept any student input devices:
  - Light pen
  - Touch sensitive screen
  - Keyboard
  - Special function keypad
  - Joystick
  - Mouse
  - Trackball
  - Voice recognition
  - 3d simulator
- o Mount on any computer
- o Accept and use any computer related peripherals:
  - Local Area Network (LAN) interfaces
  - Computer graphics and text generation
  - Mass memory storage
  - Printer interfaces

#### PMS

PMS is specifically designed to work with the US Army's Electronic Information Deliver System (EIDS), and is not intended to be used with any other hardware. EIDS is basically a PC DOS based

computer with limited graphics capability and interface with a Pioneer videodisc play. It is intended to ultimately include the capability for a variety of peripherals, but the anticipated first production capability will have only a light pen interface for students. Production EIDS was not available at the time this paper was written, but is anticipated to be available by late fall/early winter of 1987. The interim-EIDS is a SONY based machine that is generally unavailable for purchase.

#### AUTOMATED STORYBOARDING

Authoring in any other field of endeavor encompasses all stages of a project. To "author" means to create the total package, whether it be a book, software program, or slide/tape show. Therefore, authoring will include the initial stages of lesson development. For the purposes of this discussion, it will be assumed that all analyses and planning have previously been accomplished. While it is acknowledged that the ideal system will ultimately incorporate the analysis stages, and probably include some intelligence to assist the developers, it is premature to include it in this discussion.

#### IDEAL

In the ideal authoring system, flowcharting can be accomplished prior to storyboarding, in conjunction with storyboarding, or be an output of the storyboarding process. The ideal system will support any of these choices. In addition, the ideal system will have the following capabilities:

- o Generation of storyboard content:
  - Store all relevant information for later retrieval and manipulation
  - Allow flexibility for author to identify as little or as much information and documentation as is required
  - Use English language to the maximum extent; minimize abbreviations and coding
  - Allow creation of crude (stick figure) drawings as part of the documentation, both on-screen and hard-copy printout
- o Cull, sort and print:
  - The narrative script for narration production
  - All artwork requirements
  - All shot sheets, motion sequences and still frames
  - All video text requirements
  - All special effects
- o Support capabilities:
  - Immediate on-screen viewing of computer generated art or text, during development stage
  - On-screen help available at the developer's request
  - Copy, delete, add or modify format, content and sequence of instruction at any time without paging through menus
- o Lesson programming, to be discussed later, should be fully supported during storyboarding

#### PMS

PMS uses the utility called PMS I for electronic storyboarding. Much, but not all, of the information entered during this process will

be carried forward during other stages of the total PMS process. This utility allows identification of specific elements of the lesson. Not all information can be identified and recorded at this time -- for example, amplification of programming requirements and full pages of test. Specifically, the developer can identify:

- o The RECORD ID, or event number, which is the unique identifier for that information throughout the total process
- o A description of the general scene content (SG), which is intended for video production
- o A description of the specific scene content (SS), also used in video production, and used as sorting criteria for printing of the production schedule when preliminary numerical codes are inserted prior to the description and specific content abbreviations are used during the description
- o The SOURCE Reference, usually technical manual references, coded to allow review of technical source material
- o Programming functions, (PF1 thru PF4), which are described in detail in that paragraph, and at this point are limited to coded abbreviations of the type of function, without details
- o RESPONSES, which are the branching directions
- o PROCEDURE NAME and PROCEDURE TYPE, identifiers used by the developer, to code groupings of instruction (NOTE: All tests must include the word TEST in the PROC NAME and use the code T in PROC CODE)
- o Three fields of comments (C1 thru C3), which can be used for almost anything, but include the following specific characteristics
  - Only the C3 field is carried through to the programming phase (PMS III), so programming comments are restricted to this field
  - Sorting (and printing) can be accomplished inserting one of several specific codes as the first entry in the comment field
- o Four narration fields (N1 thru N4), which have uses other than narration, have the following characteristics:
  - Screen text and audio are both identified in the narration fields
  - Space is limited, so comments may have to be carried onto additional records
  - Specific codes are used to identify, for later sorting and implementation, the exact kind of "narration" intended
  - Only motion based audio is acceptable
  - Only two lines of computer generated text can be overlaid on video, each thirty six characters in length, at this stage of the process

In addition, PMS I is used to print the production schedule, as will be discussed in the paragraph on production and post-production.

#### VIDEO PRODUCTION AND POST-PRODUCTION

Most "authoring" systems do not have any application during the video production and post-production stages of a project, because they require the videodisc to be complete prior to "authoring." As with storyboarding, the video applications are an integral part of the interactive product, and must be considered as part of the authoring process.

#### IDEAL

As identified in the capabilities for automated storyboarding, the ideal system will sort and print all production requirements, in the format required by the producers. Further, the ideal system will provide the capability to update the data base during the production and post-production phases. This task could be simplified if the ideal system also created a production schedule, including production sequences. It must also be capable of working when planned sequences or schedules have been destroyed by circumstances. In addition, the ideal system will:

- o Permit documentation of source locations for completed production, regardless of media:
  - Electronic, automatic updates of the data base by interface with production equipment whenever possible
  - Manual update of the database regardless of electronic interfaces
  - Documentation of multiple sources and/or multiple locations
- o Automatic creation, storage and documentation of all video text without additional human intervention
- o Permit retrieval of all source material for rough editing:
  - Independent viewing of all source materials
  - Identification of selected source materials for final editing (post-production)
  - Call up of multiple sources, and manipulation of those sources, with automatic updating to database
- o Create an edit decision list (EDL) based on selections made during rough editing:
  - All decisions will be retained, including special effects
  - Review (or preview) of edit possible at any time
  - Manual additions, deletions, or modifications to the EDL
- o Provide complete control of the final editing (post-production) process:
  - Computer control of the total editing process
  - Single keystroke implementation of the EDL
  - Automatic update to the database via electronic interface, include SMPTE numbers and videodisc frame numbers

#### PMS

Video production and post-production are accomplished primarily using the software routines embodied by VDB I, PMS II, and VDB II. When the original storyboard laying out the video requirements is complete, a conversion process is used to create the first stage video data base (VDB I). This feature is arguably the friendliest and most powerful tool in the PMS repertoire. It is intended to be used in conjunction with the PMS I paper Production Schedule as an electronic shot sheet in situations where the computer can be set up at the production location and data entered during the shoot. The VDB I features include:

- o Reordering VDB I data files
- o Display/Edit files, when files are ordered alphabetically by subject (SG-SS)
  - Beginning with first record

- Beginning with a specific Record ID
- Skip over Records not yet shot
- Skip over records added to the data base
- New records may be added for reference, but will not be automatically carried into later processes
- o Record Shot numbers
  - Beginning with first unrecorded scene
  - Begin at specified Record ID
  - Continue from previous session, if any
- o Post SMPTE codes
- o Report status at any time
  - Find all skipped records
  - Print the file in one of two formats
    - a. Alphabetical, by subject (SG-SS)
    - b. Numerical, by shot number

With the VDB I inputs complete, another conversion process posts the recorded SMPTE codes to the PMS II data base. PMS II is then used to update the storyboard and produce the edit decision list (EDL). Updating the storyboard includes the following:

- o Verify that the narration and captions from PMS I are still appropriate to the actual video material obtained
  - Correct and/or add SMPTE codes
  - Enter any desired character generator storage identification codes
- o Manually incorporate any additional records identified in VDB I that did not come from PMS I

Once all additions and deletions have been completed, the Edit Decision List is assembled and printed. This report is a detailed listing of the VDB II data to be used to manage post-production, much as the Production Schedule is a blue-print for production. It can be constructed in different ways to support various project requirements:

- o Print all records in record ID order
- o Print only Motion Sequences in Record ID order
- o Print of still frames in SMPTE Code Order

The EDL(S) are then used in conjunction with VDB II to manage post-production. VDB II is an electronic EDL with two modes of operations, one for relatively small, linear or less complex projects under 1000 records or events, and one for large, complex projects. Mode I uses these capabilities:

- o Record Premaster SMPTE codes in one of two formats
  - Record ID order
  - Composed SMPTE order
- o Display/Edit VDB II files in one of four numerical orders
  - Production SMPTE
  - Composed Identifier
  - Premaster SMPTE
  - Record ID
- o Remove deleted VDB II records
- o Post premaster SMPTE and frame number to PMS III data file

Mode II adds these additional features:

- o Record composed edits
- o Print the file

#### LESSON PROGRAMMING

Experience has shown that if the developer is not involved in the lesson programming, deviations from the developer's intentions are possible, if not likely. Further, errors in transcription are common. In addition, as there is usually some period of time between writing of the lesson and the programming, some of the original intent may be forgotten even by the developer unless detailed and

time-consuming documentation is prepared. Experience has also shown that if programming requirements are not considered at the time of lesson development, the lesson may be incomplete or, even worse, impossible to implement as designed. This can all be solved if programming is accomplished simultaneously with storyboarding. Therefore, the ability of the developer, who may not have computer programming skills, to program while storyboarding is essential.

#### IDEAL

The following characteristics are associated with the ideal system's programming capabilities:

- o All programming will be via English language commands:
  - Shortened forms are acceptable
  - Minimal amplification will be required
- o Complete flexibility will be provided:
  - All concurrent and/or sequential system performance prior to a student input can be accomplished in a single event or record
  - Sequencing of system performance can be in any order, at the discretion of the developer
  - No artificial limitations such as the number of branches, number of computer generated text or graphic overlays, and other system capabilities, will be imposed on the developer
  - Format, structure, sequence, and/or content can be copied, added, deleted, and/or modified at any time
  - New system capabilities can be added at any time, with immediate inclusion by the developer, even if the hardware and/or software for the features has not been finalized
- o The developer will have lesson control of:
  - All video commands (assuming player has the capability), including forward, reverse, single frame, variable slow motion, and animation (rapid toggle between two adjacent frames)
  - All audio commands, including disc track 1, track 2, both tracks, no tracks, retrieval and playback of digitally stored audio (on videodisc or computer disk), and audio stored on secondary media such as tape or compact disk
  - computer generated text or graphics, including the ability to generate in background mode for instant display when needed
  - Interface with student input media, including simulator(s) when appropriate, for both stimulus and response
  - All CMI functions
- o Computer Management of Instruction (CMI) capabilities:
  - Identification of response as right or wrong, including, as a minimum, three levels of wrong responses; critical or fatal error, major error, and minor error
  - Branching within a lesson based on student performance
  - Branching between lessons based on

- student performance
- Random selection of a small number of problems from a large pool, and random ordering of a fixed set of test questions

### PMS

While elements of lesson programming are completed in earlier phases, the complete and final lesson programming is accomplished using PMS III, after the videodisc has been manufactured. The following paragraphs identify the various programming functions, according to order of use:

PMS I. During PMS I, as described in the paragraph on Automated Storyboarding, the following programming functions are accomplished:

- o Identification of RECORD ID, or event number
- o Identification of branching based on RESPONSE
- o Specification of computer text, up to two lines of 36 characters each
- o Preliminary identification of programming functions
- o Beginning and end of tests (BEG END)
- o Comments related to programming, in comment field C3

VDB II. PMS II provides no additional programming capabilities, but is the basis for VDB II. In this video data base file, the primary functions are related to post-production, but the entry of SMPTE numbers allows the system to automatically create the videodisc frame numbers required as part of the programming activity. This conversion of EDL SMPTE numbers to videodisc frame numbers, inserted into the RECORD ID's, is a programming feature that enhances PMS.

PMS III. In this last of the PMS stages, the majority of the PMS programming is accomplished. While programming related functions continue, such as testing and debugging, all final additions, deletions, and modifications are accomplished using PMS III. All of the programming is accomplished by amplifying previously entered data. Specific programming functions accomplished in this stage include:

- o Identification of audio tracks for motion sequences (MO)
- o Specification of the length of time, in seconds, a time still (TS) will appear on the screen
- o Specification of screen location from computer graphics
- o Specification of color and location of computer generated text (CT), as well as color and position of box to go behind text
- o Create pages (more than two lines) of computer generated text (PT)
- o Create frame branch files, to identify the GOTO when the sequence is interrupted
  - Interrupt motion (IM) files
  - Step thru (ST) files
  - Scan and branch (SB) files
- o Create random access (RA) file, to identify which of several branches (up to 20) the lesson will go to
- o Specification of touch areas for light pen activation
  - Standard touch areas
  - User designed touch areas
- o Specification of which responses are correct

- o Identification of start and ending frames for a series of stills (KB), plus the first frame to be displayed
- o Establish program link (PL) functions, which can be turned on and off in accordance with the lesson design
  - BACKUP - Review previous frame
  - EXIT - Allows user to exit
  - SUSPEND - Allows user to exit the program, then return to the same place at a later time
  - DEBUG - Superimposes RECORD ID, videodisc frame numbers, and graphic boxes representing touch coordinates on top of video pictures, as well as permitting access to any PMS record (event)
  - LAST DECISION - Returns user to previous decision point
  - PLACEMARK - Permits user to "mark" a record for easy access at some other point in time
  - TRACE - Outputs to hardcopy printer specific RECORD IDs accessed during lesson
  - DEFICIENCY REPORTS - Allows user to identify and annotate lesson or programming deficiencies to be printed at termination of lesson (NOTE: works only with interim-EIDS)
  - MENU - Allows user to return to the last menu accessed
  - MARGINAL NOTES - Allows user to add information, a note, which can be reviewed at a later time (NOTE: works only with interim-EIDS)
  - SOUND CUES - Provides aural cues to student, such as acceptable or unacceptable inputs (touch or keyboard)
- o Establish a loop counter (LC), used to inhibit student access to certain records, by branching to other locations
- o Specify requirement for multiple correct student responses (MC) prior to branching
- o Identify lesson locations to access, or hook (HK), non-PMS software
- o Permit linking of multiple subroutines (LS)
- o Identify a menu (MU), which has implications for program link functions
- o Past performance (PP) branches based on student performance on a given test, based on:
  - Time to respond
  - Weighted score based on time to respond
  - Point score
  - Combinations of these three
- o Record keeping functions
  - Test (TE) identifies a series of records that in total make up a test
  - The capability to define which records in a test require scoring (RR), i.e. which are test questions
  - Show the results (SR) of student performance to the student
  - Weighted time (WT) identifies a total time for a given sequence of records or events to specify a pass/fail criterion based on time
  - Record Time (RT) monitors the timing for a sequence of events, and keeps the data for later use

- o Access fill-in-the-blank (FB) capability (NOTE: works only with interim-EIDS) to turn off light pen and turn on keyboard to permit student to answer questions by entering text
- o Obtain personnel data (PD), name and/or rank and/or serial number (NOTE: works only with interim-EIDS)
- o User personnel data (UP), name and/or rank and/or serial number, obtained with PD, to customize a text screen display (NOTE: works only with interim-EIDS)
- o Deletion of production and post-production only records not needed for lesson execution

#### TESTING AND DEBUGGING

The process of testing and debugging is one of the most critical stages of interactive courseware authoring. Regardless of how well the rest of the process has been accomplished, failure at this stage can ruin the whole program.

#### IDEAL

The ideal system facilitates testing and debugging by including these characteristics:

- o Identification of syntax and/or format errors prior to lesson execution:
  - Automatic marking of specific location and type of error
  - Easy access to error location and annotation to facilitate correction
- o Run-time modification of computer generated test or graphics:
  - Automatic update of source code when modifications are made
  - Easy change of content, size, font, location, or color of text
  - Easy change of size, location, colors, or name of graphics
- o Run-time creation or modification of touch (actual touch, light pen, or mouse) responses:
  - Easy creation or modification of response location
  - Easy creation or modification of response size
- o Start at any place in lesson
- o Mark location of specific problem spots for later modification
- o Option to immediately terminate lesson execution and go to same location in source code
- o Run-time ability to view video, still or motion, anywhere on disc:
  - All normal audio and video commands available
  - Identify modifications required to video and audio commands, with automatic updates to source code
- o On-screen aides to assist instructor:
  - Full identification of record or event during run-time
  - Help function for error corrections, both run-time and off-line

#### PMS

To aid the programmer/developer in troubleshooting the lesson execution, there are programming functions and program utilities available. The programming functions include:

- o SUSPEND - allows user to exit the program, then return to the same place in the lesson at a later time

- o DEBUG - Superimposes RECORD ID, videodisc frame numbers, and graphic boxes representing touch coordinates on top of video pictures, as well as permitting access to any PMS record (event)
- o PLACEMARK - Permits user to "mark" a record for easy access at some other point in time
- o TRACE - Outputs to hardcopy printer specific RECORD IDs accessed during lesson
- o DEFICIENCY REPORTS - Allows user to identify and annotate lesson or programming deficiencies to be printed at termination of lesson (NOTE: works only with inter-EIDS)
- o MARGINAL NOTES - Allows user to add information, a note, which can be reviewed at a later time (NOTE: works only with interim-EIDS)

Specific utilities include:

- o Error checking during data entry, as incompatible programming functions are determined during the PMS stages when they are entered
- o Error Reporting Utility - This is a subset of the conversion utility, and can be used to spot check some errors prior to completion of the entire lesson, and thus reduce the amount of testing and debugging after the total lesson is complete
- o Conversion Utility - This is a five phase conversion process that continuously looks for errors to report, return to GPM (or DOS) when errors are found
  - Not every possible error will be found
  - Content errors, such as wrong video frame, text typos, location of graphics or touches, cannot be found by this or any other utility
  - Documentation does not tell what kind of errors will be found, or during which stage they will be found, although experience will quickly allow the user to identify them

#### RECORD KEEPING

As the concept behind interactive media is to provide the instructor with more time for other duties, automatic record keeping is essential. If the system provides useful, relevant data to the instructor, then instructor will accept and use the lessons. Failure of the system to provide these data makes the system unusable.

#### IDEAL

At a minimum, the record keeping functions of the system must include:

- o Student name, rank and ID number
- o Time spent on the lesson
- o Hardcopy printout of the records

In addition, there are other record keeping functions that should be available to the instructor. It may be desirable to make them instructor variables that can be turned on and off at the instructor's (or school or command) discretion. They include:

- o Student path through the lesson, assuming multiple paths are available
- o Specific errors made

- o Kind of error made, e.g. critical, major, minor
- o Correct responses by student
- o Anticipated correct response by student
- o Test results
- o Number of times a student accessed a lesson, or any part thereof
- o Complete history of all student actions
- o Performance indices; weighted scoring, comparison to pre-established norms, relative rankings, etc., both for a single lesson and for the entire curriculum

Further, data may be desired about the curriculum itself. Large groups of students may be having difficulties with the same part of a lesson or curriculum. The ideal system will assist instructors in identifying those trouble spots in order to allow modification and improvement of the instruction. Therefore, the system should be capable of generating cumulative data to identify:

- o Actual numbers and percentages of students answering specific test questions right or wrong, including breakdown of incorrect responses
- o Actual numbers and percentages of student selecting incorrect responses to non-test stimuli
- o Average and standard deviation performances by student population for each lesson or identified sub-set of a given lesson, for both errors and score

#### PMS

PMS incorporates a large amount of record keeping in the development of the lesson material. Almost any type of information about the project is available in multiple formats at any time during the development process. These capabilities are covered in detail in the paragraphs on the stages of development. The exhaustive organization and sorting possibilities available are highly commendable.

Record keeping within the realm of instruction is less comprehensive. There are basically two types of record keeping available within the lesson environment:

- o Test results which are recorded automatically according to the designer's specification of the available formats
- o Marginal notes and deficiency reports which are generated by exiting from the lesson via the program linked function capability

Also available is the initial input of personal data for identification.

#### USER FRIENDLINESS

Historically, user friendliness has been strictly defined as simplistic, menu-driven systems that require no intelligence to use. Unfortunately, that lack of intelligence is often carried over into the finished product. For the ideal authoring system, user friendliness should mean being easy to use, but not at the expense of being easy to learn. Another very important aspect of user friendliness is speed. Videodisc players have reached the ability to search to any frame in less than one second, but some authoring systems require several seconds, or longer, to accomplish even the simplest tasks.

#### IDEAL

English language structure will enhance the

friendliness of the system, as well as eliminating strange abbreviations and codes. But the system must be able to accommodate the user who is capable of intelligent improvisation and who is usually frustrated by systems that will not provide flexibility. To meet these user requirements while retaining the better aspects of the easy to learn systems, several current methodologies are promising. One of these is the use of pull-down windows, so that the user can use a simple fill-in-the-blank routine or obtain whatever level of help is needed to complete the job. Simultaneously, the actual English language source code should appear on the screen so that the user can learn, by observation, easier ways to accomplish the same tasks. If no help is needed, the developer simply enters the necessary English language commands. The ideal authoring system must be fast so that the developer can be more efficient, and also so that the developer is not frustrated by having to continuously wait for the system to "do its thing."

#### PMS

There are three types of user friendliness that need to be addressed; those of the novice developer, the experienced developer, and, ultimately, the student. These do not necessarily overlap. In the first category, the following attributes are friendly:

- o PMS is a highly structured, clearly defined environment in which it is difficult to 'get lost' as long as you remember which phase you are in
  - o There are relatively few choices available at any given juncture, making it easy to remember what is expected
  - o Once memorized, the codes are simple to recall and use
  - o VDB I is highly friendly in its entirety
- In the second category, the experienced developer will find these features especially attractive:
- o Program Linked options, especially Suspend and Debug
  - o Within the capability limitations, the ingenious developer can devise a number of nice effects without too much difficulty. The features that lend themselves to manipulation are:
    - Knob Turn
    - Loop Counter/Loop Reset
    - Scan and Branch
    - Step Through
    - Special Function

The student will find the following functions to be friendly:

- o Program linked functions including:
  - Suspend
  - Placemark

#### DISCUSSION

Again, the need for a standard is paramount. Whether that standard is PMS or some other system is less important than the identification of the standard. The truth is that no standard has been identified, and only PMS has been proposed as a universal system. In the comparison of PMS to an ideal system, PMS comes out well in some areas, but needs improvements in others. On the whole it is positive, but falls short of being a truly universal system. Its strengths are in the automated storyboarding and production areas, and its weaknesses are in hardware compatibility and programming flexibility. The following paragraphs identify those strengths and weaknesses.

## HARDWARE

PMS is designed to be used with a single source of hardware, either Interim-EIDS or Production EIDS, depending on the version of PMS available. In this respect it is no better than some of the commercially available software packages designed to help sell a specific piece of hardware. In addition to the computer restrictions, the videodisc player, student input device (light pen), computer graphics and text capability, and audio storage medium are limited. PMS has no utility for those who have, or will acquire, different hardware. It does not maximize use of existing DoD hardware systems. While there is a linking function that permits unique software to be utilized in specific applications, PMS software is not capable of integrating or supporting any lesson material invoked by the "HOOK" function. However, attempts have been made to make EIDS more compatible, with an IBM compatible system, so that PMS will in effect have more universal utility. To be truly universal, however, PMS needs to expand its hardware horizon.

## AUTOMATED STORYBOARDING

Probably more than any other available software package, PMS meets the need for automated storyboarding. As presently implemented, however, PMS needs to be upgraded, improved, in the following areas:

- o Documentation space within a single record or event needs to be increased
- o The extensive use of abbreviations and/or codes needs to be reduced
- o Art (stick figures) during the storyboarding process would be a major enhancement
- o Computer generated art and text should be viewable during the storyboard stage
- o More on-screen help should be available during this stage
- o Extensive pagination required to use system capabilities must be reduced

## VIDEO PRODUCTION AND POST-PRODUCTION

PMS again is superior to most other available products in the area of production and post-production. The strengths are the ability to generate the paper products for production control and the ability to use the data base in support of production. Greater things are planned in this area, but at present there are some problem areas that could be addressed in future versions:

- o New video requirements added to the video data base (VDBI) during video production are not automatically integrated into the PMS data base. New data entered via VDB I can only be located and viewed in VDB I. The writer or other individual must go back into the storyboarding stage and manually add the relevant information.
- o No electronic interfaces are present. There are no automatic updates to the database, no way to communicate for automatic video text generation, and no automatic generation or control of the Edit Decision List.

## LESSON PROGRAMMING

The ability of PMS to accomplish programming functions is somewhat limited. There are some good features, not common to other authoring programs. Especially notable are:

- o TIMED STILLS

- o KNOB TURN
- o INTERRUPT MOTION
- o SCAN AND BRANCH
- o STEP THRU
- o WEIGHTED TIME

On the other hand, the following programming limitations inhibit the ability of PMS to provide good instruction:

- o More than eight branches from a single record are not possible without artificial contrivances
- o Only contiguous videodisc frames or sequences (either still or motion, not both), can be used in a single record or event
- o A maximum of four graphics, less any other desired special features, can be used in a single record
- o One graphic cannot be superimposed on top of another, e.g. a circle on top of a square, unless the special HOOK function is used to call in a graphic created outside of PMS
- o Multiple events or records are often required to accomplish a single task or related group of tasks
- o Extensive menu pagination is required to change from "add" to "modify" mode, or vice versa
- o The system limits you to pre-existing capabilities, and changes are virtually impossible to obtain within the constraints of time associated with a lesson development project
- o Videodisc player capabilities are not fully utilized:
  - No reverse motion
  - Slow motion is a single speed, not variable
- o No capability exists for audio other than linear videodisc audio, tracks 1 and 2
- o "Seamless" delivery of lesson material is impossible - video or computer generated images cannot be automatically retained from one event or record to the next:
  - The video frame or computer text specified in each event or record must be re-displayed
  - Computer generated text or graphics must be re-generated for each record or event
  - Videodisc and computer generated image combination specification is restricted to which appears first; you cannot create computer text or graphics while viewing a video image, and you cannot search for a video frame while looking at a computer generated page
- o The only student interfaces possible with interim EIDS are the light pen and keyboard, but only the light pen will be available with production EIDS
- o Random selection works only in selecting the first event or record of a sequence, severely restricting testing options
- o Determination of correct and incorrect responses available only during testing
- o No discrimination between various kinds of wrong answers is possible
- o Only one right answer is acceptable per question when using weighted scoring
- o There is no branching between lessons based on performance

## TESTING AND DEBUGGING

Some of the PMS utilities are designed to facilitate testing and debugging. When the execution code is assembled, certain errors are printed on the screen, which makes it easy to go back and fix the problem. Two major problems with this scheme were immediately obvious. There are five utilities or phases during assembly, and an error may not be found until the fifth phase. A lot of time will have transpired by then (between 20 and 30 minutes for a 50 record lesson). After fixing the problem, you must go through all five phases again, you cannot just rerun the failed phase. Another facet of this is that the assembly process is halted at the end of any phase in which an error is detected, so it isn't possible to go through the complete assembly process and then resolve all detected errors at once. The other problem is that it didn't catch every error, not even all format errors. These errors are not discovered until you attempt to run the lesson. One of the good ideas was to have the software identify any records or events that are never accessed within the lesson structure.

No run-time testing and debugging aids are available, except that some movement around the lesson is possible if you are clever. The documentation for such manipulations is sketchy, but it can be done. If the location of text, graphics, or touch is not correct, you must exit execution, go back into the PMS III, make your fix, reassemble the lesson, and try it out again. That can take a very long time. It takes so long that the developer is likely to feel that the lesson is good enough as it is and not bother to fix it. This attitude is also engendered by the process of verifying video frame numbers for start and stop of such good video commands as INTERRUPT MOTION, SCAN AND BRANCH, and STEP THROUGH. Close enough. The result is that the best features become just too much trouble to use properly.

## RECORD KEEPING

In PMS, record keeping is minimal. The "TRACE" function allows an intelligent instructor to follow a student's path through the instruction, but it isn't easy. Some branching within a lesson based on student performance on a test is claimed, although it did not work as published when tested, but no capability is available outside of the test. Further, there is no ability to branch between lessons, or to carry performance across lessons. The lesson validation capabilities are non-existent. In addition, any data generated must be retrieved immediately. It cannot be stored in memory for later recall. If "TRACE" is not turned on prior to the lesson start, there is no way to activate it during the lesson.

## USER FRIENDLINESS

An asset for PMS is that it is relatively easy to learn. Inexperienced users are trained in the total capabilities of the system in a one week workshop, and then can create lessons using the software. However, in that it does not provide flexibility, it is not easy to use. Lack of capabilities is one major cause, slowness is another.

## CONCLUSION

PMS is a good start toward the ideal of a universal authoring system. However, while PMS may

meet current and projected needs for Army Extension Courses, as presently configured it does not embody all of the characteristics of a universal authoring system. A very substantial amount of work is necessary to achieve that. PMS can be improved, and has the potential to meet the needs of the ideal system, and the hope of all concerned that it will meet those needs. It certainly is superior to most competing systems in certain aspects such as storyboarding and production management.

One suggestion that has been forwarded is to perform similar analyses on other authoring systems to ascertain which is closest to meeting the needs of the universal system. There are two problems with that concept: there are too many systems that are currently in use, and all of the other systems are commercial products, and not under government control - no assurance that the changes necessary will be implemented is available. To selectively sample from the existing systems opens the possibility of missing the one best system, and, as has been the case in previous such studies, would undoubtedly elicit all kinds of protests from those not sampled.

A second suggestion is to do nothing and let the market place decide which is best. This approach hasn't worked yet and is not likely to do so. The "best" system may be forced out of the market because of factors totally unrelated to overall quality and utility.

Until a standard is approved, controversy will continue as to whether or not this or that system is best. Compared to an ideal standard, PMS holds up as well. However, without a true standard, neither PMS or any other system will meet everyone's needs.

## ABOUT THE AUTHORS

James R. Stonge is a Senior Instructional Systems Engineer currently supervising the courseware development activities of the Display/Trainer Products, Electronics Systems Division of Grumman. He is responsible for all technology associated with instructional systems development in that organization, as well as supervision of ISD personnel. Prior to assuming this responsibility he was ISD manager for AMTESS/MITS technology at Grumman. Prior to joining Grumman he was on the faculty of Indiana University. In addition to an AB, MBA, and MS in Instructional Technology, Mr. Stonge is completing his doctoral dissertation. He is a frequent contributor and presenter at I/ITSC and other national symposia.

Linda G. Jensen is an Instructional Systems Design Engineer in the Display/Trainer Products Group of Grumman's Electronics Systems Division. Her primary responsibility is E-2C training applications. She was also the ISD manager for Grumman's A-6E SWIP videodisc project. Her background includes nine years in development, modification, maintenance, and technical support of Air Force flight simulators. A graduate of the Army's PMS workshop at Fort Eustis, she wrote and presented a paper at the US Air Force Technology in Training and Education (TITE) conference on the need for a universal authoring system. She also attended the Training Technology workshop on videodisc applications at the invitation of the Department of Defense and the National Security Industrial Association.