R. Craig Richards, M.A.
Ellen M. Le Vita, M.A.

CAE-Link Corporation, Link Training Services Division
Oak Harbor, WA

DESIGNING CONCURRENCY INTO A TRAINING CURRICULUM USING COMPUTER-BASED TRAINING

ABSTRACT

How do you design a program that will be able to deliver quality training for a weapon system that is technically complex, requires a high level of operational mastery, and will change in six months? To add to the challenge, the objectives and procedures used to create the training system will change before the training can be delivered.

At first glance, training for the Navy's EA-6B aircrew appears to have the same basic requirements as for other multicrew tactical Navy aircraft. However, unlike other types of aircraft, the EA-6B's weapon system is driven by three general-purpose computers and multiple special-application microprocessors. The extensive software required to drive these computers provides optimal flexibility for the EA-6B's evolving mission. Because of the rapidly changing nature of electronic warfare, the EA-6B's hardware and software are constantly being updated to meet the challenges of newly developed detection technology. Major system changes and updates occur every six to eight months. The expanding system capabilities require continuous training restructuring to maintain operator proficiency.

The secret to conducting successful training under these circumstances is to rely heavily on Computer-Based Training (CBT), ensuring that management and programming techniques efficiently handle changes and updates. This paper will discuss CBT development techniques used in the EA-6B program and raise some training concurrency issues associated with weapon systems designed to be updated on a frequent basis.

INTRODUCTION

In June 1986, the Navy negotiated with contractors to develop a multimedia training syllabus for aircrew flying the EA-6B. When completed, the syllabus will be used by the EA-6B Fleet Replacement Squadron (FRS), VAQ-129, at NAS Whidbey Island, Washington. The project began with an analysis of the training requirements including a task analysis, development of an objectives hierarchy, and a media plan. The media plan recommended a mix of five media types for academic lessons: mediated lectures, slide/tapes, videotapes, workbooks, and Computer-Based Training (CBT). The Navy stipulated that it wanted to incorporate CBT in the syllabus because of successes at other sites using CBT with aircrew training. Based on the experience of CBT developers at other training sites, it only seemed natural that CBT would be relatively easy to incorporate in the EA-6B syllabus. Because of the evolving nature of the EA-6B's mission, however, problems arose as lesson development began.

The EA-6B's major weapon system is the ALQ-99 Tactical Jamming System (TJS) which is comprised of a number of state-of-the-art computers. The computers control receivers used to gather electronic emissions intelligence and jamming pods used in support of tactical air operations to jam enemy radars. These computers are programmed by means of a cassette tape which the Electronic Countermeasures Officer (ECMO) carries out to the airplane. The operating program used in the computers is based on microprocessor technology. This makes changing the software for the ALQ-99 relatively easy; consequently, updates are frequent.

The dynamic nature of the weapon system in the EA-6B posed some problems to the development of a multimedia training syllabus, but initially the training analysts felt that concurrency would not require any special considerations beyond those normally encountered in other aviation training programs. The frequent updates of the ALQ-99 system that were required after the start of the program did not appear to be a major factor because they were well documented, but as is the case with most software products, the final version differed from the original specifications. The differences occurred for a number of reasons including hardware/software interface problems, changing specifications, and new technology.

COMPUTER-BASED TRAINING

Computer-based training has existed for more than a decade with niches in industry and the military. As a medium for student learning, it has demonstrated its effectiveness in training a diverse population to a wide variety of tasks.[1, 2] There have been hundreds of studies measuring the effectiveness of CBT under different conditions, so there is considerable data available to determine the best ways of developing a CBT lesson.[3] One method that has become the most widely accepted is the systems approach. The systems approach relies on the task analysis, and the lesson structure is based on task organization. In a systems approach, the student is introduced to the procedure, practices the procedure, and then is tested on it.

One of the major components of the ALQ-99 is the digital display group. The digital display group is comprised of the Digital Display Indicator (DDI) and its input controller. The DDI is a computer monitor with a 12-inch diagonal screen. The Digital Display Indicator Controller (DDIC) is an operator keyboard that provides most of the input to the ALQ-99 TJS. The DDI and DDIC are ideal candidates for CBT training because they lend themselves well to modeled simulation and the systems approach. It is possible to create graphics that look identical to the actual DDI display. By displaying the DDIC and DDI on the CBT monitor, the student simulates performing procedures and observing the results.

Figure 1 illustrates a sample of the DDIC and DDI setup as displayed on the CBT monitor. Although in the actual aircraft there are slight differences in procedural techniques--for example, cursor movement is controlled by a trackball located on the DDIC in the aircraft while on the CBT it is controlled by selecting a position on the DDI with the mouse--the transfer of training has proven effective.

CBT lessons become very structured using a systems development model. Feedback is consistent throughout the lesson, and student progression is controlled almost entirely by the computer program. The student does retain control over the pacing of the lesson, but the computer assures the student completes the task-derived objectives. This approach is ideal in military training where the objectives are specific, and there is a minimum performance standard. A systems paradigm can also prescribe the types of graphics and the degree of interactivity required. This makes it a little easier for the lesson developers because the feedback, graphics, text, objectives, and even the test questions can be standardized. A systems approach also helps to make changes in the CBT lesson relatively easy.

Many experts warn that if the training requirements are going to change significantly, do not use CBT![4] CBT is more expensive to produce than most other media; the perception follows that updates are also more expensive. The experience gained by the EA-6B development team found, to the contrary, that a strict adherence to the systems approach helps make changing a CBT lesson more efficient and straightforward. One of the strengths of the systems approach to developing CBT is its rigidity and consistency. In the EA-6B program, this approach to developing CBT lessons allowed changes in the material without degrading the impact of the lesson on the students. When a CBT lesson conformed to systems programming principles, changes in lesson content were completed with minimal effect to other portions of the lesson. One of the problems facing the CBT developers was the extensive changes to the ALQ-99 TJS. Almost every graphic frame in 16

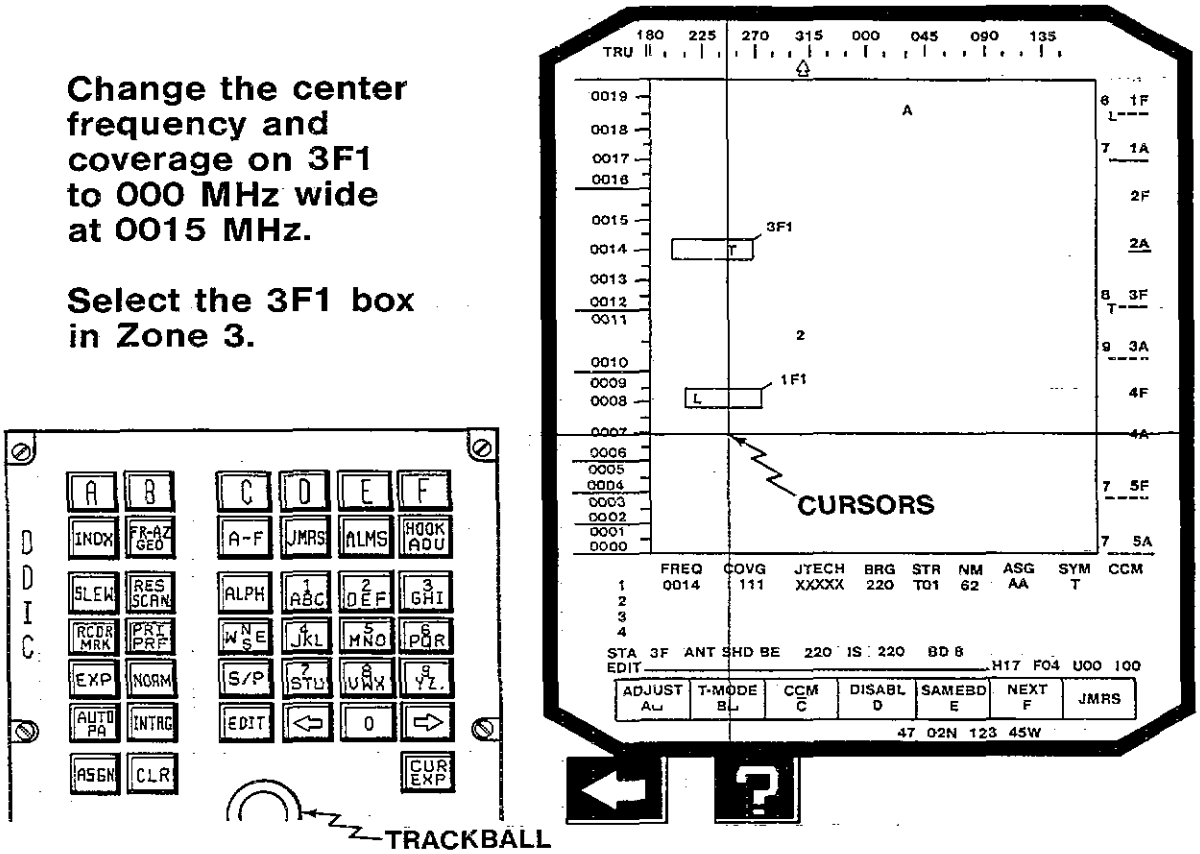

Figure 1. Digital Display Group

CBT lessons would require changes within a few months of implementation. This problem was compounded by constant amendments to the proposed weapon system update that were taking place during the writing of the lessons. It was decided early in the project that the lessons would be written for a weapon system update that was to take place a year later. It was thought that this would help preclude any changes in the syllabus for at least a year after its implementation. But almost immediately after starting the project, it became apparent to the courseware developers that changes to be included in the weapon system update were still in flux. Decisions were still being made three months before release on what was going to be included in the update. Even when decisions had been made to include an item in the weapon systems update, effects of the changes on the DDI's display were still unknown. This meant that the CBT lessons' graphics required revisions before they could be accepted by the Navy. This is analogous to building a house with the architect changing the plans after completing the foundation and framing. The CBT developers decided to complete the CBT lessons with what was known of the weapon system update and make changes to the lesson as they became known.

CBT wasn't the only medium having problems with constant changes. Other media being developed for the syllabus would also require extensive changes before and after Navy acceptance. This included lectures with slides, videotapes, slide/tapes, and workbooks. As the other media were updated to conform with weapon system changes, a surprising fact about CBT became evident: It was easier and less expensive to make changes in CBT lessons than in most of the other media.

To make changes to slide/tape lessons, it was necessary to make new slides and rerecord the narrative. This required the efforts of a SME, an illustrator, typesetter, and other graphic artists; a word processor; a narrator; and audio production technicians. To change one to 20 word slides requires approximately two man-hours. To change the same number of complex slides ranges from four to 20 man-hours. To update the accompanying narration takes approximately eight to 10 man-hours regardless of whether one narrative sequence needs changing or the entire tape. This time does not include the delays in scheduling the narrator and audio production facilities. In addition, numerous copies of the slide/tape sets also had to be made.

Videotapes are even more difficult and expensive to update than slide/tapes. Photographers, artists, sound specialists, training specialists, video specialists, Navy squadron personnel, and aircraft were all required in making changes to videotapes. Again, it did not matter if the changes were many or few in number; the editing process was expensive, and the time to make the changes was excessive. Setting up and shooting a video sequence takes a minimum of seven man-hours but usually runs around 25 man-hours. Subsequently editing the videotape ranges from six to 40 man-hours. These hours do not take into account scheduling the production which, in one case, took over six months. Because of the difficulties, the decision was made to replace videotapes with other media if the updates were going to affect the content of the lesson.

Lectures were relatively easy to change except that the slides accompanying the lecture required the same amount of time as those in a slide/tape lesson.

Interactive videodisc was not considered as a medium for the EA-6B syllabus for a number of reasons: It was expensive to produce, and changes are very difficult to make once the discs are mastered. All of the people necessary to make changes in videotape lessons are required as well as computer programmers and packagers. Due to the extent of EA-6B weapon system changes, with each new release, a new videodisc would have to be remastered.

In contrast to these media, making changes on CBT lessons went surprisingly fast and were relatively inexpensive. In many cases, all that was required to make the changes was for the programmer or artist to call up the frames involved and make the changes on-line within a few minutes. In very few cases, entire lessons could be updated in eight to 16 man-hours. Because the CBT software was mainframe-based, the changes had to be made only once. No copies were required as is the case with all the other media types.

Another advantage available with many CBT systems is the ability to reference graphics contained in a library. Essentially, a graphic is created and placed in a graphics library. When that graphic is required by the lesson, instead of copying the graphic into the lesson, the artist references the graphic. When a change to the graphic is required, the artist makes the change to the graphic in the library and subsequently, all references to the graphic will reflect the change. Library referencing was used in placing the DDIC in all the lessons. Later, when a typographical error was found on the DDIC, the correction was made to the graphic library and was reflected in all the lessons. This change took less than five minutes to make. However, had the error been copied into each lesson, it would have taken approximately one man-hour per lesson to make the same change. In this case, using library referencing saved approximately seven man-hours. In the future, should the panel be replaced or modified, library referencing will save approximately 25 man-hours.

BASIC CONSIDERATIONS FOR CBT

The paragraphs that follow contain a discussion of some basic CBT considerations that will help keep lessons concurrent with a frequently changing aircraft system. These basic principles should be implemented for any CBT program regardless of whether or not one anticipates frequent changes in the lessons at which time they become essential.

Choosing CBT Hardware/Software

One of the first considerations in developing CBT for any curriculum, static or dynamic, is choosing the hardware and software to develop and run the courseware. Almost any CBT system can be used to develop quality lessons; the design and execution of the lessons primarily determine the quality. However, some CBT systems make the initial development and the later updates easier to accomplish. CBT features that must be considered include whether the system will be stand-alone or networked, the number of student terminals required, the student interface (how the student communicates with the computer), system upgrade capabilities, memory capacity, graphic capabilities, branching flexibility, programming language capabilities (including the ability to import external languages such as Pascal or C), computer-managed instruction capabilities, and the list goes on.

The reasons for picking one system over another must take into account more things than can be discussed in this paper. However, when considering CBT hardware and software for a dynamic weapon system such as the ALQ-99 TJS, certain considerations should be addressed. Primary among these are the ease with which graphics, text, branching, and programming can be modified; whether the system is stand-alone or networked; whether to use interactive videodisc or digitized audio capabilities; and whether to use the menus provided by the system--if provided--or programming languages.

Bit Mapping Graphics

A general rule in developing computer graphics for use in CBT courseware is to use bit maps. Depending on the hardware, bit maps draw on the screen faster and distract less from the lesson than do vector graphics. Bit maps are essential when showing switch movement by using the technique of overlaying switches in new positions. However, there is one exception to using bit-map graphics especially when an aircraft system is expected to change. That exception is when drawing a graphic that requires changing alphanumerics such as on a Light Emitting Diode (LED) display.
Most graphics programs include a text feature for writing words and numbers. Modifying such text is almost as easy as modifying text on a word processor. However, if the text is bit-mapped, making modifications to the text now becomes the laborious task of modifying letters and numbers on a pixel-by-pixel basis. It is much easier just to retype the letters and numbers. The EA-6B weapon system uses the DDI which combines graphic displays with an abundance of alphanumeric information. It is this information that changes with each update of the weapon system. Therefore, although the rest of the graphics in the EA-6B program have been bit mapped, the developers of the current courseware chose to leave the DDI alphanumeric information in a text format to facilitate modifications.

Centralized Source Code

For courseware with extensive modifications anticipated, creating and delivering the courseware on a CBT system where the computers are networked into a main system is a definite advantage over slide/tapes, videotapes, and other similar media. When modifications are required, all versions of the lesson are automatically changed once the source code is changed. Of course, CBT that is stand-alone where the lessons are contained on separate disks does not have this advantage. Whenever possible for courseware that is expected to change often, it is advantageous to use a networked computer system.

Programming Template Frames

Most distributors of CBT systems market their systems based on the ease with which one can create courseware. This ease usually derives from a menu-driven authoring language. Such languages are useful for developing simple, linear-type lessons. When developing courseware that requires the flexibility of multiple correct answers on a page, multiple branching options, and/or interfacing an interactive feature such as cursors for fidelity purposes, using a programming language becomes the preferred method of authoring. However, anytime a programming language is used, it limits the numbers of people who can package, or put together, the lesson because these people must be knowledgeable of programming techniques. There is a way around the problem: create templates.

Templates are miniprograms that can be used locally in an individual lesson or globally throughout all the lessons. Templates allow the nonprogrammer to program a lesson. A properly formatted template works like a "black box," allowing the nonprogrammer to fill in the essential ingredients, or variables, that allow the program to do the rest of the work.

Figure 2 shows a partial example of a template used in the EA-6B CBT courseware. A well-designed template begins with a general description of its function and any special requirements in its graphics frame such as the required answer fields. Next, there are the variables that must be filled in by the packager. For example, the variable i[6] equals the next frame to which the program will branch. Below the area where the variables are declared are the actual programming statements. The packager need not worry about how the program works; as long as he fills in the variables properly, the program will run.

As with any good program, templates must be well-documented. This is especially true of the variables that the nonprogrammer must fill in. However, small variations must be made periodically to accommodate unusual lesson circumstances; if the program is well-documented, this becomes a relatively easy task for another programmer to accomplish. Depending on the quality of the documentation and the complexity of the modification, the nonprogrammer can make these modifications.

```
GENERAL DESCRIPTION

{*Performs Multiple Switches*}

{*Fields*}
{*1-8 switches, 9 wrong, 12 try again, 13 back*}
{*10 white prompt, 11 yellow prompt, 15 new text*}
{*16 2nd white prompt, 17 2nd yellow prompt*}
(10, 11, 15, 16, 17 only if using new text*}
{*18 bcheck, only if using bcheck*}
{*14 help, only if using help*}
=======================================
VARIABLES

i[8]:=3;                {*#of switches*}
i[6]:=30;               {*next frame*}
i[15]:=15;              {*previous frame*}
i[7]:=0;                {*help page; set to 0 if not used*}
b[11]:=F;               {*set to t if no new text*}
b[13]:=F;               {*set to t if no bcheck*}
i[21]:=34;              {*1st graphic frame*}
i[22]:=68;              {*2nd graphic frame*}
i[23]:=0:               {*3rd graphic frame*}
==============================
s[1]:="The panel is set up correctly. Now continue.";{*bcheck prompt*}
s[1000]:="Continue setting up the panel.";     {*new text*}
s[100]= "Set the Power switch to ON.";         {*1st prompt*}
s[200]:="Set the POD switch to SAFE.";         {*2nd prompt*}
2[300]:="";                                    {*3rd prompt*}
==============================
i[15]:=42;              {*frmid*}
i[10]:=20;              {*back*}
i[13]:=13;              {*bcheck*}
i[14]:=24;              {*check*}
i[12]:=17;              {*help*}
=======================================
PROGRAM

If pfrmid <> i[15] then BLOCK                {*frmid*}
    i[98]:=i[98] + i[8]; i[30]:=0;           {*amnt of q's which prompt*}
    i[1]:=0; i[2]:=0; i[5]:=i[8];            {*input, count, score*}
    If b[50] then BLOCK
       showgfrm (i[10]);                     {*back icon*}
       If i[11] > 0  then showgfrm(i[12]);   {*help icon*}
    BLOCKEND
        else
    If not b[13] then showgfrm(i[13]);       {*bcheck icon*}
    showgfrm(cfrmid);                        {*fields*}
    LOOP
        exitwhen i[2]=i[8]; i[2]:=i[2] + 1;b[i[2]]:=f
    LOOPEND;                            {*set flag for each switch*}
BLOCKEND:

etc....
```

Figure 2. Sample Template

Another advantage to using the template system is that variables are consistent between lessons and programmers. If the variable i[6] is the next frame in one lesson, it will be the next frame in all the lessons. This is essential in courseware that is expected to change often because it allows any programmer to modify a lesson with little difficulty. The original programmer does not have to make the modifications nor does it take time for a different programmer to determine the idiosyncrasies of the original programmer's thoughts. In fact, everybody should be working on everybody else's lessons to ensure consistency and standardization between lessons and packagers. Any individual knowledgeable of the courseware standards should be able to pick up any lesson for modification purposes.

Hardcopy Printouts

CBT courseware is developed through a series of steps including an author writing a storyboard, a paper and pencil script of the lesson. Storyboards contain a number of important information items such as the branching options, the graphics display, the instructional text, the directions to continue with the lesson, and the correct and wrong answer feedbacks. The initial CBT lesson is programmed using these storyboards. However, during the course of putting the lesson on-line, conducting in-house technical reviews, and obtaining acceptance of the lesson by the customer, the lesson goes through a series of modifications. Unless the graphic artists and packagers are diligent in updating the storyboards, the resulting lesson often bears little resemblance to the original storyboard. For this reason, it is essential to make hardcopy printouts of the finished lessons.

Hardcopy printouts serve three purposes: First, they serve as documentation of the lesson as it exists. Second, they can be used to identify subtle mistakes made in the courseware that may not be obvious when viewing the lessons on-line. Third, they can serve as the starting point for developing storyboards to update rapidly changing courseware. Using the hardcopies for this purpose, one should make appropriate changes on xeroxes of the hardcopy printout and develop new storyboards only when necessary. That enables the graphic artists and packagers to copy the existing lesson and readily identify the aspects of the lessons that require the changes rather than having to start from scratch and compare the old lesson to the new storyboard in order to determine what has changed.

CONCLUSION

Initial costs of CBT development is more expensive than in other media. In the EA-6B program, revisions to the CBT turned out to be less expensive and easier to make. The CBT lessons developed for the EA-6B FRS have been accepted by the community. Instructors and students are commenting on how well the CBT is teaching the basic ALQ-99 TJS procedures and how much more valuable the time spent in the simulator is because the students are better prepared. Based on its success, the authors have concluded that CBT is a viable medium for training procedural skills even in a dynamic environment. However, a coherent development plan and programming system must be determined early in the project.

REFERENCES

(1) Le Vita, E. M. & Williams, A. M. The strengths, successes and lessons learned in the use of computer-based training by the S-3A, F/A-18, and F-14A naval aviation training programs. Proceedings of the Interservice/Industry Training Equipment Conference. Washington, D.C., November 1986, 545-550.

(2) Kulik, J. A., Bangert, R. L., & Williams, G. W. Effects of computer-based teaching. In D. F. Walker & R. D. Hess (Eds), Instructional Software. New York: Wadsworth Publishing, 1984.

(3) Cohen, P. A., Kulik, J. A., & Kulik, C. L. C. Educational outcomes of tutoring: A metanalysis of findings. American Educational Research Journal, 1982, 19, 237-248.

(4) Beech, G. Computer-based Learning. Chesire, England: Sigma Technical Press, 1983.

ACKNOWLEDGMENTS

The authors would like to acknowledge and thank Diane G. Zibel for her editorial review of the manuscript.

ABOUT THE AUTHORS

R. CRAIG RICHARDS is a Subject Matter Expert with CAE-Link Corporation, Link Training Services Division. For the past four years, he has been developing multimedia training materials for EA-6B pilots and electronic countermeasures officers (ECMOs) at NAS Whidbey Island. Mr. Richards was in the United States Marine Corps for six years and has logged over 1200 hours in jet aircraft including EA-6Bs. He continues to fly EA-6As with the United States Marine Corps Reserve. Mr. Richards holds an M.A. in Computer Education from Western Washington University.

ELLEN M. Le VITA is a Senior Instructional Psychologist with CAE-Link Corporation, Link Training Services Division. For the past three years, she has been managing the development of an EA-6B Aircrew Training program which includes computer-based training. Ms. Le Vita also spent two and a half years managing and developing CBT for F-14A aircrew. Ms. Le Vita has extensive experience using both WICAT and MicroTICCIT authoring systems. Prior to developing CBT programs, she worked on various aviation and ASW projects, primarily developing training materials using Instructional Systems Development principles. Ms. Le Vita has published papers in the areas of computer-based training, memory, and developmental psychophysiology. Ms. Le Vita holds an M.A. in Experimental Psychology from San Diego State University.