SURVIVING THE PITFALLS AND TROLL BRIDGES

IN STATE-OF-THE-ART CBI DEVELOPMENT

Dr. Robin Halley, Member of the Technical Staff, Logicon, Inc., San Diego, California

Ms. Mary Hicklin, Member of the Technical Staff, Logicon, Inc., San Diego, California

ABSTRACT

Most computer-based instructional (CBI) system development is done using proven technologies in proven ways. Sophisticated development models are available for determining software (e.g., top-down design) and courseware (ISD) decisions and structures in these systems. For state-of-the-art (SOA) CBI system development, however, there are additional probelms peculiar to that form which require specific considerations and solutions. This paper is designed to define the specific characteristics of this type of development and to discuss some of the problems, considerations, and solutions involved in SOA training system research and development. These include personnel concerns, such as work environment characteristics and the type of people best suited to it; management concerns, such as the style of management required; communications concerns, such as the required interaction between different staff disciplines; and the different technical approaches required for this specific form of development.

INTRODUCTION

Logicon specializes in the integrated applications of advanced technologies and advanced concepts to system engineering problems. Within this context, the Tactical and Training Systems Division has undertaken several state-of-the-art CBI (computer-based instruction) system development projects. (1,2,3) The purpose of this paper is to present some of the problems associated with doing work in this area and some possible methods for avoiding those problems.

The problems covered in this paper are of basically two types: pitfalls and troll bridges. Pitfalls are those problems which occur where you don't expect them. You think you're on solid ground and then suddenly you're at the bottom of a hole with a lot of angry natives looking down at you. Troll bridges are generally problems which you know about. The trolls have a unique way of conducting business: they peacefully accept whatever payment you believe is reasonable for crossing the bridge. Then, when you're at the middle, they demand you pay more or they toss you off.

It is important, at this point, to define some terms. SOA development means attempting to implement the most recent technologies available within a given field. This is, by definition, a high risk endeavor with unknown commodities. It makes the developer vulnerable to many levels of non-success. A CBI system is based upon computers and associated peripheral devices such as CRTs, printers, audiovisual delivery systems, and computerized speech generation and voice recognition hardware. The techniques used include computer assisted instruction (CAI), computer-based performance measurement (PM), and computer managed instruction (CMI).

In an area as diverse as state-of-the-art (SOA) development of CBI, there are lots of pitfalls and troll bridges. They come in large (general) and small (specific) sizes and can appear anywhere. This paper attempts to present some of the ones of both sizes that Logicon has encountered in several different development areas.

This paper has been structured to move, more or less, from general to specific. The first three sections deal with more general concerns. These sections are titled "State-of-the-Art," "Computer-Based Instruction," and "People." The final two sections delve into more of the specifics of CBI development. These sections are titled "New Hardware to USE" and "New Things We're Doing."

STATE-OF-THE-ART

There are generally two sets of attitudes regarding SOA technologies. These can be categorized as "bandwagon" and "tradition-based." A good example of the bandwagon approach is the present "love affair" going on with the videodisc. It seems that lately just about every other training system RFP has requested the use of videodisc. In many cases, however, traditional technologies could provide better performance with less risk.

The tradition-based approach is characterized by people not using new technologies solely because the technologies are new or using them as an expensive version of the more traditional media. An example is the use of videotape in the classroom. Initially, there was a great deal of resistance from teachers because they thought they were going to be replaced. Then, when television finally was accepted into the classroom, if was often used to present lectures...just as though there was a teacher standing there.

Because there is no body of experience upon which to base design and development decisions, it is important to realize that state-of-the-art development involves taking risks and makes you vulnerable to failure. Thus, there must be a commitment to this vulnerability. RAdm. Albert J. Baciocco provides this perspective:

> "When I state a willingness to take risks, I am simultaneously stating a willingness to accept some failures. In my view, failure is also a measure of success in basic research...as long as it's not 100% failure." (4)

If you get involved in producing a "successful product" the first time out, you'll most likely limit your use of the technologies to the more familiar realms rather than exploring the further limits of their capacities. If the decision has been made to do SOA development, it is important to apply the knowledge that has been accrued from work in similar areas. This knowledge should be used as a starting point for identifying means for using the SOA technologies in way most amenable to their inherent characteristics. If there is no commitment to vulnerability in the use of the new technology, you would probably *be better off adapting an older, more familiar* and secure approach.

Early in the project a **needs assessment** should be done. In this case, you are assessing your need to actually utilize new technologies: how are they appropriate, what do they add, and are they cost effective. In some cases, you may be responsible for providing a technology test bed. In that case, you should carefully examine the possible applications for ones that will stretch the limits of the technology. When you're not responsible for testing the technologies, you should carefully analyze your application and choose the technologies (be they new or be they old) that most closely fit your identified needs, resources, and allowable risks.

COMPUTER-BASED INSTRUCTION

In CBI, as with most instructional approaches, it is important to define a task that can be accomplished with your given resources (people, time, money). The task itself should be driven by considerations of the instructional system outcomes: what is the system supposed to do, how quickly, for how many people, how well.

In most instructional approaches the instructional designer knows the medium sufficiently well to make many intelligent decisions concerning equipment, budget, and approach. CBI, because it involves complex and expensive components, requires three sets of specialists who **must** be involved in most of the initial system decisions and who must communicate continuously during the design and development process to insure that everyone understands what, precisely, is going on.

The three sets of specialists are hardware, software, and courseware designers and developers. Since they represent entirely different parts of the system development process, communication to ensure understanding becomes imperative. These people must establish a mutually held concept of the system and its capabilities before the limiting, defining, and restricting decisions (e.g., what hardware, what software language, what instructional outcomes) have been made. Then they must continue the cross discipline communication to ensure that the picture stays congruent for all three staff disciplines.

PEOPLE

There are several sets of people involved in the development process of SOA CBI. In addition to staff, there are managers and users. The users can be further categorized into students and instructors. This section addresses pitfalls and troll bridges associated with each of these sets of people.

Managers

SOA CBI development often lacks much of the extrinsic structure common to training systems which use more traditional methods and media. Additionally, the stress levels will typically be higher as a reflection of the risk (chance for error) associated with exploring new territories.

Many of the problems associated with this area may be avoided by maintaining a more active, more psychological project management approach than usual. It is important to reduce staff stress. This may be done by supplying strong leadership to provide structure where it is absent. Managers will want to encourage innovation and analysis at the outset to insure that the technology is being utilized in the best possible way. They will want to ensure communication between *the groups (hardware, software,* courseware).

Perhaps most importantly, managers should identify "inch pebbles" (as opposed to milestones) for identifying project progress and should be very aware of places where the inch pebbles are not being reached. This will provide a methodology for forestalling the large crises to which SOA development is more vulnerable.

Staff

In attempting SOA CBI development, your choice of staff members becomes a little more important. Of course you will want to choose people with as good skills and experience as are available to you. However, **staff members *for SOA development should, if possible, be flexible* persons with low "final product" attachments.** SOA work does not lend itself to obvious first time success and often requires multiple iterations or complete redesign in order to derive a useful product. Indeed, sometimes you may find that success with a particular technology configuration is impossible. Therefore, a person working on SOA CBI would be better served to get satisfaction from getting a job done as well as they can, rather than depending on traditionally successful end results.

Students

In terms of the student-system interface, CBI presents the design/development team with a new set of considerations. Major amongst these considerations are the learners' attitude toward automated instruction, how to supervise information presentation, and how to teach the learner to use the system.

Learner Attitudes. There are two main attitude problems. Up until the recent past the average person has stood in awe of computers. Learners have been initially reticent to use computer systems for fear of "breaking" the very expensive equipment. With the rapid inclusion of computers into many aspects of day-to-day life this concern is diminishing, but still must be considered. Fears have also been voiced, on a more or less continuous basis, about computers dehumanizing the learning experience.

The designer must take both these factors into consideration. The CBI system must be user oriented. It is wise to go to the learner population and poll their attitudes about computers. By concentrating on identifying any fears the students have about CBI, you can know where to start in providing a stress reducing introduction to the system and the experience. Both the computer fears and the possibility that the lack of human intervention will degrade the instructional effectiveness can be at least partially solved through (1) careful inclusion of humans into the system design and (2) giving the computer a "personality." Building a human-system rapport can serve to reduce both the potential effects of dehumanization and learner fears about system use.

Presenting Information. A computer-based instructional system can provide the learner with information in a variety of ways. These include CRT, various audiovisual media, printers, and computer generated speech. Guidelines about how to present information using various media already exist. Instructional Message Design (5) is one good example.

However, when you have more than one information source available in your CBI system, your problem becomes one of directing the learners' attention from one source to the next and ensuring that the learners know precisely what behavior is expected of them at all times. Not knowing how to initiate the next step can cause the learner to experience a sort of stress-induced paralysis or may result in a random pushing of keys and buttons until **something** happens.

Learning How to Use the System. The use of computer-based systems provides you with another new problem, that of familiarity. The student will not usually be familiar with computerized instruction in general and your system in particular. You, therefore are left with the task of teaching the student how to use the teaching system.

This is a very important problem inasmuch as, if the student cannot use the system, the system cannot train. It is critical that you identify what the student will need to do to use the system and then carefully teach those behaviors. It is a good idea to separate this instruction from the general instruction so the student may review aspects of system use whenever he is unsure of what he is supposed to be doing to make things go.

Instructors

As with the students, the instructors are users of this system. Their responsibilities will usually extend to managing system use and providing a human control over system use and misuse by the students. Their attitudes and knowledges must be considered in your system design and development.

Instructors will generally resist the implementation of a CBI system within their teaching realm. This is a result of fears that they are going to be replaced by the system or that they will become simple technician adjuncts while the system does the important tasks. In our experience with CBI, the instructors generally become comfortable only when they see they have some control over the system (6) and that they are still involved in important instructional system tasks.

It becomes very important, therefore, to make the system as easy to supervise as possible. It also becomes very important to provide the instructors with training on the roles that they are expected to fulfill and to accentuate how important these roles are to system success. In one system presently under development, we are developing an instructor tutorial using the system to teach the instructors as well as the students.

In any case, the current SOA CBI system tends to put the instructor into three roles: system manager (controlling and logging system operation), instructional facilitator (overseeing and smoothing system-student interactions), and tutor. Since most instructors are only familiar with being in the role of primary instructional resource, the three new roles must also be taught.

NEW HARDWARE TO USE

The new hardware which is mentioned below is but a small sample of the wondrous things which are becoming available for use in instructional systems. In this paper the two types of hardware discussed are automated speech hardware and the videodisc.

Automated Speech Hardware

Automated speech includes speech generation, speech recognition, and speech understanding. Speech generation and recognition are mostly hardware based and will be discussed in this section. Speech understanding is mostly software based and will be discussed in the NEW THINGS WE'RE DOING section.

Automated Speech Generation. There are a variety of devices available which can produce speech under computer control. Most of these devices fall into two general categories: speech synthesizers and speech encoder/decoders.

Speech Synthesizers. The synthesizers produce artificial sounding but intelligible speech by combining electronically generated sounds which correspond roughly to English phonemes. Some synthesizers are capable of performing the translation from English text to speech directly; others require human intervention to specify the phoneme strings and inflections. The devices also vary in cost and in the intelligibility of the speech output. The synthetic quality of the speech has been identified as a disadvantage, but the small data storage and I/O processing overhead requirements make synthesizers an attractive solution to many speech generation problems.

The one outstanding drawback to the synthesizer encountered in certain applications is the fact that the simulation of many different voices is impossible. Another difficulty is that it is usually impossible to vary speech rate under program control. Even at the highest available speech rate, the synthesizers tend to speak relatively slowly. This caused difficulties when they were used to simulate air traffic controllers because the synthesizer was sometimes unable to provide enough control information in the time allotted.

Speech Encoder/Decoders. These devices utilize a technique which digitally encodes and compresses of human speech in such a way that it can be subsequently decoded to produce speech. These devices provide a range of fidelity from superb (and expensive) audiophile quality, to lower cost units which provide correspondingly lower fidelity. Even the low fidelity units produce speech which is unmistakably human and sufficiently intelligible for many applications. These devices can easily be used to simulate different voices simply by recording voices of different talkers. Depending upon the encoding technique used, these devices can offer the advantage that speech is simply recorded, and there is no need for painstaking encoding of phoneme strings. Others require off-line processing of the encoded information to reduce the data requirements to practical levels. The disadvantage of the encoder/decoders lies in their relatively much larger data storage and I/O processing requirements.

Speech Recognition. Commercially available hardware ranges from inexpensive units which recognize a few words spoken in isolation to more expensive devices which recognize a large vocabulary of isolated words or phrases with high accuracy, to even more expensive devices which recognize continuous speech. Most units are still "speaker dependent" which means they must be configured to recognize an individual talker's voice by having the person repeat each vocabulary item one to ten times.

Our experience has shown that recognition accuracy is affected by several factors, including the vocabulary itself. In some applications, the vocabulary is defined a priori. In many others, it can be defined by the system designer. In the latter case, great care should be devoted to the development of a vocabulary tailored both for user acceptance and for machine recognition. The optimization for machine recognition almost has to be done by experimentation because the machine ear has different characteristics from the human ear. The inability of many systems to distinguish reliably between "five" and "nine" is often cited. In our applications, we have also found many other pairs which cause trouble, including "above" and "below," "left" and "right," and "port" and "four."

Videodisc

The videodisc is a marvelous tool for providing color, still, or moving pictures and high quality audio under computer control. It could, of course, be used to provide lecture material, but its great advantage lies in the fact that demonstrations by experts, mini reviews of specific topics, etc., can be randomly accessed by the computer or at trainee request. Theoretically, the technology could provide the basis for a talking, moving book, with a sophisticated indexing and retrieval capability.

The disadvantages to the technology include that, unlike videotape, the production aspect is akin to audio record mastering, hence the audiovisual materials cannot readily be changed. The unreliability of early devices also has proved problematical.

It is very important to carefully identify precisely what kinds of graphics and sound you wish to present in your CBI system and then make sure that videodisc is the proper presentation medium for your needs. Once that's determined, your next concern is choosing the videodisc player appropriate for your application. There are several varieties of this technology presently available with others undergoing prototype development.

Another important hurdle is production. Since this is a relatively infant medium, many of the production bugs still must be worked out. Premastering, mastering, and quality assurance are all areas to which you should pay attention. (7) It is also important to note that, as of the beginning of this year, none of the videodisc production companies had the ability to do classified premastering or mastering. If you have security related materials to show you should check carefully to see if this problem has been overcome.

NEW THINGS WE'RE DOING

This section describes some of the parts of the CBI system such as courseware, automated instructor model, and automated speech understanding. Herein are described some of the problems associated with working in these SOA areas.

Courseware

In a CBI system the courseware materials are mostly embedded in the computer memory. Therefore, the courseware language you use becomes very important. Because of the iterative nature of SOA CBI development, it becomes practically a necessity to be able to a quick, easy review of the materials you have developed. As you add hardware devices (e.g., videodisc, simulated job station) or capabilities to your system, it becomes increasingly important to have a supporting courseware language which offers simplicity in your specification of devices, what they do, when they do it, and how long they do it. The language, if possible, should also provide for easy revision of materials.

A typical CBI system development will require parallel development of software and courseware. Unless you have a very long timeline and can develop and test the whole system before you have to write and enter any of your materials into computer memory, it is a good idea to devise your system so that you can see and revise pieces (i.e., text pages, scenarios) before the entire system is finished.

There are several different courseware languages available on different systems (e.g., Pilot, Decal) and others under development. If none of those you can find offer what you need, you may want to do as we have done and develop your own courseware language to precisely fit your application. If you choose to develop a language, define very carefully amongst the three work specialties (courseware, software, and hardware) what the instructional system will be designed to do, given your particular resource parameters.

Automated Instructor

The simulation of any complex human behavior is certainly not a trivial task. In the course of our work in automating the instructor's task of evaluating a trainee, we have come to appreciate just how non-trivial the assessment of human performance can be! The following discussion highlights some of the pitfalls and troll bridges we have encountered.

Components of an Automated Instructor Capability. An automated instructor capability should have several components. It should:

- Provide an objective measure of trainee performance on a particular problem relative to objective criteria;
- Collect data suitable for statistical analysis and norm development;
- Provide the human instructor with a meaningful assessment of trainee performance at a particular stage of instruction, as well as an overall assessment;
- Provide positive and negative feedback and annotated replay;
- Provide adaptively tailored learning materials.

Distinguishing all these capabilities makes it clear that the specification of the raw performance data which the system is to collect requires detailed task analysis, and further that these raw data must be analyzed in various ways to serve meaningful ends.

Task Analysis. The CBI system designer is at a great disadvantage with respect to other training program developers. When one has a human instructor to deal with, one can simply say, "teach this." The computer is obviously unable to do that. It can be a profound shock to the subject matter expert (SME) to find that the task analysis has to be specified to an incredibly detailed level to satisfy the needs of the CBI system. In fact, SMEs are asked to quantify those qualitative judgements which they make almost automatically. This can prove to be very difficult. The system designer must be especially sensitive to the fact that the SME is being asked to think in a new way and that revisions to the the task analysis outcomes will be inevitable.

Measures of Performance. On the one hand, statisticians have demonstrated that the overall performance evaluation an expert instructor provides can be emulated by considering only a very small number of measures of performance. Statistical techniques like these are very useful for the specific purpose of providing an overall grade. However, it would be a mistake to conclude that other measures were of no use. In fact, a rich set of measures based upon the task analysis data is needed to enable the training system to identify the trainee mistakes which led up to an undesirable outcome and provide indications of correct procedures.

Like any test items, these performance measures must be tested, validated, refined, and tested again. When this has been done, it is necessary to combine the measures to provide various overall measures of trainee performance. Again, these combined measures must be validated and refined. It is foolhardy to ignore this basic principle of test instrument development just because the test items do not look like standard test items.

The validation procedure may reveal problems not encountered in the validation of standard items. For example, in evaluating ongoing performance, there can be a "domino effect" in which failure on one mini-test can cascade and cause other errors to be reported. For example, a procedure may involve several actions which have to be performed in sequence and within a certain time limit. If the learners take too long on the first step, they'll also fail the second and subsequent steps.

Another difficulty which can arise is that, on occasion, the trainee will be unable to perform a required action because another, more important action is required. This can lead to double bind situations in which the system reports an error no matter what the trainee does.

Feedback. Care must be devoted to the design of the feedback given to the trainee. An automated performance measurement system can detect every minute violation of every obscure rule. If the system merely reports errors, several problems arise:

- The feedback can become overwhelmingly negative.
- It becomes difficult for the trainee to appreciate the distinction between serious errors and nit-picking errors.
- If feedback is provided during a performance task, it can be distracting; yet if provided after the problem, it can be difficult to correlate with behavior.
- A special problem which arises in speech recognition based systems is that performance errors can be accumulated due to the fact that the system misrecognizes the trainee's speech.
- The sheer number of error reports encourages the trainee to stop paying any attention to them at all.

Adaptive Learning Material Selection. We have been striving to develop an automated instructional capability which can diagnose problem areas, prescribe remedial work, recommend further practice at a given level, and advance the trainee to the next learning task. A successful implementation depends first of all upon well validated performance measures. Secondly, it requires a thorough understanding of what learning problem a particular constellation of errors represents. This is at best difficult to achieve for a complex performance task.

Computer Based Simulation. Learning a performance task has its purely cognitive aspects, but achievement of proficiency requires practice. Often, supplying such practice can be costly and can even involve safety hazards. Air traffic controller training is an example: the novice controller cannot be entrusted with responsibility for "live" aircraft until he or she has attained to a certain minimum level of proficiency, and yet practice is needed to attain that proficiency. Simulation provides the obvious fulfillment of this training need.

Simulations of operational environments vary in the degree of fidelity which they provide, and the question the training system designer continually faces is: what degree of fidelity in simulation is required to assure transfer of training to the operational environment? To be safe, the designer can specify that a perfect simulation of the environment is necessary. This can not only prove to be expensive, but has been shown to be unnecessary in many cases. Some SOA issues we consider in developing simulations for training systems are described in the paragraphs which follow.

Environment Simulation. Environmental simulations need not be extremely complex to provide sufficient fidelity to assure transfer of training. For example, just because the AIC must learn to control a variety of aircraft which differ greatly in response characteristics, it does not mean a training system needs complex simulations of these various aircraft. The sweep rate on the radar indicator is only one sweep every twelve seconds, and a simple model of aircraft speed and altitude and perhaps radar cross section provides all the information the AIC can normally distinguish on the operational gear. Thus a relatively simple environmental simulation can support training in a cost-effective way. Furthermore, aspects of the environmental simulation can easily be manipulated to vary the difficulty of the training problem in a way that could not be done at all in the operational environment. For example, in the precision approach radar controller training system, wind speed, direction, and gusting characteristics were varied adaptively because this controller's job becomes more difficult in windy conditions. In the AIC training system, more bogeys (bad guys) could be added to increase problem difficulty without the expense of sending up more F-4s.

Team Member Simulations. The simulation of the other humans who interact with the learner is one of the more interesting training system design problems. The traditional approach of using human pseudo pilots is not necessarily the most effective because these pseudo pilots may not have been trained to respond as a pilot would. Often the trainees themselves take turns serving as pseudo pilots. This is good insofar as the trainee gets some sense of what it is like to be on the receiving end of control information, but it can be less helpful when one trainee acting as a pseudo pilot decides to make life either easy or miserable for his buddy. When the human pseudo pilot is replaced by a computer simulation, control can be exercised over this facet of the environment. In the precision approach radar controller training system for example, pilot ability was an adaptive variable. When the trainee was just beginning to learn the task, a very good pilot was provided. As the controller's proficiency increased, a few poor pilots were introduced to give the trainee experience in dealing with a variety of control situations.

The modelling of these other persons in the environment requires that mini task analyses of their respective jobs be conducted as well. The cost of this analysis is easily offset by eliminating the need for pseudo pilots, and a significant training advantage is gained in being able to control the responses of these simulated individuals.

Automated Speech Understanding

Speech understanding is a far more difficult technical problem than speech generation. There are two main aspects to it: speech **recognition** (performed more or less accurately by hardware devices), and speech **understanding** (the process of making an appropriate verbal or other response based upon the speech input).

The advent of the *speech recognition technology* has led us into new realms of adventure in training automation. It made it possible in theory to automate the training of the primarily verbal tasks of the air traffic controller. If the computer can understand the controller's advisories, it can effectively simulate pilot responses and obviate the need for human pseudo pilots in training. In our attempts to do this, we have learned a bit about how to evaluate a speech understanding application on the one hand and how to assess hardware speech recognition systems on the other. The speech understanding concerns are discussed immediately below. Speech recognition device concerns were presented in the NEW HARDWARE TO USE section under Speech Hardware.

Speech understanding, as defined above, refers to the entire process of responding intelligently to speech inputs. This involves determining what was actually spoken, and then generating appropriate verbal replies and queries, and modifying the simulated environment. The problem of determining what was actually spoken is necessitated by the fact that recognition errors will occur. Manufacturer's claims of 99+% speech recognition accuracy will probably not be realized, especially in the training environment where the trainee is just learning a verbal task. Thus, it is crucial for the speech understanding algorithm to expect misrecognitions and to check on the reasonableness of any input before taking action on it.

In our experience in the automation of this kind of training, where the trainee is engaged in a performance task and the computer is providing the simulated environment, there is a great deal of information in the system which can be used to check the reasonableness of an input and either automatically correct the input or request that the trainee "say again." This latter technique has to be used with caution, for there is nothing more frustrating than hearing the computer repeatedly demand a repeat.

When the system designer is faced with the need to incorporate a continuous speech recognition device, the speech understanding problem becomes vastly more complex. The training system developer must be aware of this fact. The decision to use such a device must be based not only upon user acceptance considerations, but also upon ability to employ talented software engineers to design the speech understanding software and the ability to commit significant computational resources to the speech understanding process.

CONCLUSION

The above discussion of pitfalls and troll bridges in SOA CBI development is not nearly comprehensive, as it reflects only our particular experience with this field. It does show some general principles to keep in mind: use a systematic approach, define your need for and uses of the technologies at the outset; use flexible, creative persons to staff the project; insist on cross discipline communication throughout the project; and leave plenty of resources for iteration and reiteration.

It's also a good idea, when you encounter problems, to try to find someone who's been there before so you can avoid reinventing the wheel. There will be plenty of new mistakes left for you to make.

State-of-the-art computer-based instruction systems represent a lot of risks and a lot of work. But, as we have found out, if you're diligent and careful, they're very satisfying to build and to use.

ABOUT THE AUTHORS

Dr. Robin Halley and Ms. Mary Hicklin are members of the technical staff in the Advanced Systems Department of Logicon, Inc.

REFERENCES

1. M. Hicklin, et al. Ground Controlled Approach Controller Training (GCA-CTS). System Documentation. Final Report. NAVTRAEQUIPCEN 77-C-0162-3. 23 February 1980.
2. L. K. Bosworth, J. T. Kryway, and S. S. Seidensticker. F-14 Instructional Support System (ISS) Weapon System Training (WST). Preliminary of Final Report. NAVTRAEQUIPCEN 80-C-0056-1. 12 March 1981.
3. E. Regelson, et al. Functional Design for Air Intercept Controller Prototype. Final Report. NAVTRAEQUIPCEN 78-C-0182-8. 28 June 1981.
4. RAdm A. J. Baciocco. "View From the Top," Military Electronics/Countermeasures. March, 1981. P. 13.
5. M. Fleming and W. H. Levie. Instructional Message Design. Englewood Cliffs, New Jersey: Educational Technology Publications, 1978.
6. Laura Joplin. Interactive Logs for Automated Speech Technology. Final Report. NAVTRAEQUIPCEN 79-C-0066-1. 9 August 1980.
7. Robin Halley. "Some Pragmatic Considerations of Videodisc Production," Videodisc News. March, 1981. Pp. 3-5.