EXPERT SYSTEM AND INTERACTIVE VIDEODISC: A POWERFUL COMBINATION

Dennis C. Hribar, Ed.D. and John A. DiDuro
Training Program Developer Logistic Engineer

NEWPORT NEWS SHIPBUILDING
Newport News, VA 23607

ABSTRACT

Newport News Shipbuilding (NNS) has integrated videodisc media with expert system (ES) technology to create an innovative method for the delivery of submarine maintenance information. NNS has developed an ES entitled Scrubber which supports Navy efforts to modernize the distribution and use of technical information. Scrubber delivers maintenance information about the Carbon Dioxide Scrubber via menus or interactive dialogue. This videodisc-based ES can be easily adapted to facilitate maintenance and diagnostics of any complex piece of equipment or machinery.

INTRODUCTION

Newport News Shipbuilding (NNS) has entered the field of expert system (ES) development because industry and the military are seeking expert maintenance systems to reduce equipment training and maintenance costs. Computerized information delivery systems offer many advantages over present printed technical materials. The use of digital systems for storage, delivery, and updating of maintenance information provides a resource for many new applications.

The Scrubber project of the Integrated Logistic Support Department of NNS represents a commitment to the following developments:

- ES design and programming
- Videodisc and computer-based training
- Automated logistic support delivery systems

Scrubber integrates the technologies of ES, videodiscs, and microcomputers to demonstrate an intelligent interactive videodisc (IIVD) application. Scrubber illustrates how maintenance information can be delivered through an ES to speed retrieval of shipboard maintenance information. First, the user enters data regarding the operation of the equipment. Then, the system, using the embedded diagnostic rules of experts to determine the problem, directly presents the required information. In the future, technicians who use such a system will need less training time to become proficient in diagnostic and maintenance skills.

Scrubber provides detailed information on three commonly maintained components of the Carbon Dioxide Removal System--the Carbon Dioxide Flowmeter, the Back Pressure Regulator Valve, and the Carbon Dioxide Compressor. System information consists of textual descriptions and audio/visual information (stills, motion, drawings, and schematics) for preventive, troubleshooting, and corrective maintenance.

The user touches the screen to enter data and to control the program. The program continues to prompt the user for additional data until it reaches a conclusion. Then, the program lists the logic of the conclusion for user verification. If the user agrees with the logic, the program presents step-by-step audio/visual information of the required maintenance steps. If the user disagrees with the logic, the program returns to the fact in question and asks for new input.

PRODUCT DESCRIPTION

Scrubber operates on an IBM XT enhanced with the IMSATT system hardware and software. Table 1 lists all items necessary for the Scrubber program.

TABLE 1. System Configuration.

HARDWARE:

- IBM XT with 640K RAM, one 360K disk drive, one serial port, one parallel port, 20mb or larger hard disk, and clock/calendar
- Sony PVM1271Q Monitor with Touch Screen
- Sony LDP 2000 Videodisc player
- VPC2100 Voice Board with speaker
- VOTAN Software
- Microkey System Model 1125 for IBM PC

- TECHMAR Graphic Master Card attached to Microkey System
- Sigma Expansion Chassis for IBM PC

SYSTEM SOFTWARE:

- IMSATT Authoring Software

COURSEWARE:

- Scrubber Videodisc
- Assorted Program, Graphics, Touch, and Voice Files

All information for the Scrubber program resides on either the videodisc or the computer hard disk. The Scrubber videodisc contains video, stills, diagrams, and voice messages. The hard disk of the computer stores the knowledge base, expert shell, graphics, and additional voice messages.

The Scrubber computer program consists of two subprograms--courseware and ES. The courseware program functions to control the system, display text and video, generate graphics, playback audio messages, operate touchscreens, present program options, accept user input, and select feedback.

The expert shell functions to evaluate the user input to seek further input, determine facts, reach conclusions, and present the requested guidance. The expert shell uses information stored in its knowledge base, rules, and inferences to assess the input and present a solution.

METHOD

The Training Technology Section of NNS applies two methods to the development of an IIVD application:

- Team Concept
- Instructional Systems Development (ISD)

The Team Concept allows various experts to blend their abilities in the production of the multi-faceted IIVD. ISD defines the five-phase process to analyze, design, develop, implement, and evaluate training materials. Table 2 outlines the interactions of the team roles during a project.

The two methods, Team Concept and ISD, supply complementary answers to the question of how to develop an IIVD. Team Concept defines the roles of the team members, and ISD outlines the development process. Together, these methods provide guidance for project management by specifying the responsibilities of each team member. Although team size varies with the project, four individuals form the core of any project team: the Curriculum Designer, Programmer, Media Specialist, and Subject Matter Expert.

<u>Team Concept</u>

Team Concept involves merging individuals from diverse fields of specialization into a group of equals that can share information and talents. IIVD development requires expertise in management, curriculum development, educational psychology, computer systems, programming, media production, writing, art, and subject matter.

TABLE 2. *Team Concept.*

IIVD TEAM	ISD PROCESS				
	ANALYZE	DESIGN	DEVELOP	IMPLEMENT	EVALUATE
Team Leader	O	O	*	*	*
Curriculum Designer	O	O	O	O	O
Interface Designer		O	*	*	*
Programmer		O	O	O	*
System Analyst	*	*	*	*	*
Video Specialist		O	O	*	
Broadcast Talent			O	*	
Writer		*	O	O	O
Illustrator		*	O	*	
Subject Matter Expert	O	O	O	O	O

O Major Role
* Minor Role

Team members assume one or more of the following ten roles:

- Team Leader
- Curriculum Designer
- Interface Designer
- Programmer
- System Analyst
- Media Specialist
- Broadcast Talent
- Writer
- Illustrator
- Subject Matter Expert (SME)

In projects of modest size, each individual performs several roles. Large IIVD projects involve more individuals who work primarily within their specialty.

ISD Process

The five phases of ISD consist of Analysis, Design, Development, Implementation, and Evaluation.

Analysis. Phase I of the ISD process establishes the scope and purpose of the program by considering the characteristics, skill level, and needs of the target population. Phase I includes collecting the necessary technical materials, interviewing SMEs, and testing initial ideas through prototyping. For an IIVD application, this phase would include selecting software for ES development.

Design. Phase II produces the design documents which detail the user-to-machine interaction, the audio/visual segments, and the program flowchart. For an IIVD application, this phase would involve extensive work with the SME to develop the knowledge base, rules, and inferences of the ES. The team approves the design before beginning the next phase.

Development. Phase III involves work in three areas: computer programming, media production, and handbook construction.

The Programmer writes and tests the computer code which executes the branching and logic of the program design. This person, translates the design documents into a functional program. Team members continually evaluate the emerging program to verify the success of the design. For an IIVD application, the Programmer integrates the ES knowledge base, rules, and inferences into the program.

The Media Specialist produces the audio/visual materials in three stages: pre-production, production, and post-production. During pre-production, this person develops the script, shot list, and storyboard which the team reviews and approves prior to the next stage. In the production stage, the Media Specialist follows the approved pre-production materials to guide the taping of the audio/visual segments required for the IIVD. During post-production, this team member edits the production materials, creates a master tape and mails the tape to a videodisc pressing facility. The facility, upon request, returns a checkdisc which the Programmer uses to finalize the program.

The team writes and illustrates a handbook for the user of the program. The handbook contains instructions for system installation, startup, troubleshooting, and program use. The handbook may include resource materials (readings, exercises, and references) to reinforce and supplement the program.

Implementation. Phase IV consists of on-site product installation and testing. The team installs the complete IIVD product at the client's site and teaches users how to operate it. Some team members remain at the site for several days to ensure smooth program operation.

These team members conduct an alpha test of the IIVD. They monitor its use, maintain its operation, answer users' questions, and revise the product based on user reactions. After the alpha test, the team publishes the final product (videodisc, floppy disk, and handbook).

All information (audio, video, program, and text) may be recorded on the videodisc to produce a stand-alone IIVD. However, only systems capable of retrieving program code from a videodisc will recognize the added code; all other systems will require the use of floppy disks.

Evaluation. Phase V involves a beta test, which is the ongoing use of the program without the presence of team members. Periodically, team members request user feedback via courseware management files, written surveys, and interview forms. With this information, the team determines additional product improvements and writes the final project report.

RESULTS

The Scrubber project demonstrates the feasibility of integrating an ES with a videodisc to deliver maintenance instructions. Periodic evaluations of the product enhanced the accuracy and effectiveness of the final product. Formative evaluations during project development supplied corrective feedback, and a summative evaluation of the finished product elicited responses from a wider variety of users. Evaluative feedback stimulated product improvements.

Formative Evaluation

The team utilized three sources for formative evaluations of Scrubber:

- Former Navy auxiliarymen employed by NNS
- Written technical materials
- Designers and testers of the equipment from its manufacturer, Vitok of Louisville, Kentucky

Ongoing evaluations provided a valuable check on the progress and direction of the project. These evaluations continually assessed the clarity and flow of the program. NNS experts reviewed the final product prior to the start of the summative evaluation.

Summative Evaluation

The team scheduled various experts and novices to evaluate the acceptability and usefulness of the program. First, the evaluators received a briefing about the program and the evaluation. Then, they operated the program for up to 50 minutes. Finally, they completed an evaluation form and discussed their appraisal with a team member. This evaluation process identified the strengths and weaknesses of the program, information that was necessary for product improvement.

The evaluators brought varied perspectives to the review of the program. Each selected evaluator approached the product from one of five backgrounds-- Carbon Dioxide Scrubber maintenance, submarine veteran, educator, computer programmer, and novice. Table 3 presents selected responses of the evaluators by background.

Feedback highlighted the strengths and weaknesses of the program. The team separated the information by background to differentiate between broadly-based and group-specific criticism. Group-unique responses highlighted concerns particular to one group and identified areas for further inquiry. Evaluation findings influenced product revision.

PROGRAM DEVELOPMENT

ES differs from standard programs in many ways. Whereas standard programs automate calculation, ES automates judgment. Standard programs follow a linear decision path; ES considers multiple factors to reach a conclusion. Standard programs automate clearly defined processes; ES focuses on skills developed through experience. The differences between standard programs and ES necessitate differences in development procedures.

TABLE 3. Evaluation Results.

SAMPLE OF RESPONSES

TOPICS:

1. Clarity of the Program Structure
2. Ease of Program Structure
3. Clarity of Explanations
4. Helpfulness at Illustrating Maintenance Steps
5. Completeness of the Maintenance Directions
6. Stimulation of Interest in Scrubber Maintenance

RESPONSE BY GROUP (1-Low, 5-High):

TOPIC	SMEs	SUB VETERANS	EDUCATORS	COMPUTER PROGRMRS	NOVICES	TOPIC AVERAGE
1.	3.50	2.25	4.33	3.50	2.00	3.12
2.	4.00	2.25	4.33	4.50	4.00	3.82
3.	2.50	4.25	3.33	4.50	4.00	3.72
4.	3.50	4.50	4.33	4.50	5.00	4.37
5.	2.50	3.50	3.66	3.50	2.00	3.03
6.	4.50	4.25	3.33	3.50	4.00	3.92
GROUP AVERAGE	3.42	3.50	3.90	4.00	3.50	3.66

Knowledge Engineering

Knowledge Engineering involves the specification of an SME's thought processes and actions relating to a limited problem-solving situation. The Knowledge Engineer (KE), using techniques from education and psychology, clarifies and organizes the SME's knowledge to form the basis for the ES.

Three factors contribute to the difficulty of Knowledge Engineering. One, the subject matter often involves information learned primarily through experience. Two, users expect the ES to solve problems, give advice, and make predictions with accuracy equal to that of a human expert. Three, the skills and opinions of one SME rarely match those of another. The selection of the SME greatly influences the usefulness and acceptability of the subsequent ES.

The effort and time necessary to create an ES pays off when it saves users' time and money by quickly determining the preferred solution rather than presenting non-productive alternatives. The integration of interactive videodisc (IVD) technology with ES makes the payoff even greater.

ES and IVD make an excellent combination because an ES can unleash the visual storage and selection capabilities of the IVD by determining the appropriate information required. The ES assesses the current situation, specifies the need for additional information, and presents the preferred solution based on the preprogrammed rule base. The IVD effectively illustrates the solution in an immediate and precise manner.

Iterative Prototyping

Iterative Prototyping exemplifies the evolutionary nature of product development. Iterative Prototyping modifies the straight-line approach of top-down, bottom-up design to recognize the evolutionary nature of IIVD development. Iterative Prototyping consists of successive stages of refinement and integration of ideas from initial concept to tested code. This spiralling development process encourages the inclusion of an increasing number of sophisticated features into the program to ensure user acceptance.

Development Tools

Software development tools for ES offer facilities that track the formation of the knowledge base, identify missing data, and produce the rules and inferences. Without a development tool, handwritten and database-processed information may suffice for the construction of a prototype ES such as Scrubber. However, the development of a full-scale ES requires a tool to organize the web of information relationships present in the subject matter.

Few ES tools exist that offer the ability to control an IVD player. However, the IMSATT system, used by NNS, combines a complete authoring environment for computer and IVD-based instruction with a rule-based language for ES programming. Research continues at NNS on the selection of a tool to interface with the IMSATT authoring system.

NNS uses database techniques in lieu of a development tool to facilitate the organization, design, and production of the IIVD. Use of a database program simplifies the ordering, cataloguing, and retrieval of information necessary to identify interdependencies and inconsistencies in the data. The database facilitates the development of the knowledge base, flowcharts, and storyboards.

Time Considerations

NNS' approach to IIVD development emphasizes the importance of the analysis and design phases of development. The complexity of ES development, IVD production, and expert selection lengthens these phases of IIVD development. Table 4 contrasts the IVD and IIVD development process.

TABLE 4. IVD versus IIVD Development.

PHASES OF IVD DEVELOPMENT	ADDITIONAL STEPS IN EACH PHASE FOR IIVD DEVELOPMENT
ANALYSIS	Selection of Development Tool and Experts
DESIGN	Development and Testing of Rules
DEVELOPMENT	Integration of Expert Logic
IMPLEMENTATION	SME Review
EVALUATION	SME Review

The tasks of Knowledge Engineering and ES development occur prior to the development of the IVD. Development of the IVD commences after the specification of the nature of the presentation and the details of the information. The complex relationship and interplay of information in an ES requires that the design of the

IIVD presentation be clearly specified. Choices made during the design phase concerning the structure of the program, information presentation, and man-machine interface have a lasting impact on the nature of the product.

The critical process of selecting the SME occurs during the analysis phase of the project. Since the SME's skills, knowledge, and opinions become the contents of the IIVD, the SME must represent the considered opinion of experts in the field. Although some controversy will result no matter which SME is selected, proper selection will minimize adverse reactions to the product. The selection process chooses from a panel of SMEs the one SME who provides the most consistent, reliable, and complete information.

PROGRAMMING TECHNIQUES

The microcomputer environment seems limited compared to the expansive requirements of ES and IVD. Scrubber requires memory space for active logic, storage space for program sections, and processing speed for near real-time response. The program code for a microcomputer-based IIVD must be compact, powerful, and flexible yet readable, testable, and expandable. The IMSATT authoring language offers two keywords, "tag" and "make," that allow the reduction of code to a fraction of its original size. Scrubber demonstrates effective use of these keywords to accommodate an IIVD application within the capacity of an IBM XT microcomputer.

Imsatt Language

The IMSATT language, a procedural language based on FORTH, consists of approximately seventy keywords. The programmer uses these keywords to define the procedures of the program. Each procedure receives a unique name. In the concluding, "main program" section of the program, the programmer arranges the names of the procedures to accomplish operation of the program.

The keyword "tag" defines a program section that can be called into another section to replace procedures and variables. The section that calls the "tagged" code forms a framework or "meta-procedure" for a large portion of the program. "Tag" offers flexibility and power because a common routine, such as menu screens, can be supplied with particular instructions to match a specific need. Menu routines vary because of the number of choices and the corresponding results of each choice. The use of "tag" reduces the code for input screens to one "meta-procedure" and the various "tag" routines.

The keyword "make" calls a "tag" routine into a "meta-procedure" to meet the particular requirements of the program at that time. The input "meta-procedure" generates a four-choice menu by using the keyword "make" to call the "tag" containing the routines associated with a four-choice menu. The size of the Scrubber program, which consists of nine "meta-procedures," would be several times larger if written without "tag" and "make." Procedures would contain a large amount of duplicate code, which would make the program too large to store and operate on an IBM XT. Table 5 defines the hierarchy of program organization.

TABLE 5. Top-Down Program Organization.

PROGRAM	- A systematic method of providing specific information and accessing the required "meta-procedures."
"META-PROCEDURE"	- A block of generic computer code which creates a frame-work for multi-use execution and calls procedures to fulfill specific requirements.
PROCEDURE	- A block of computer code that accomplishes a specific task.
KEYWORD	- Words that provide the functions of the language upon which the program is built.

Approach to Programming

The "meta-procedure" technique offers four advantages over programming unique code for each branch of the program:

One, "meta-procedures" facilitate program revision by reducing the amount of unique code, simplifying access to vital information, and facilitating program revision without affecting program control.

Two, "meta-procedures" simplify the understanding of program structure and flow. The looping patterns employed in the "meta-procedures" constitute the bulk of the program code.

Three, "meta-procedures" allow a program structure to be used for other applications. Information on a new piece of equipment can replace equipment information in the program to produce a new program. Approximately 40% of the computer code remains useful. Revision of the program requires three items: a technical manual on the equipment, a SME, and an IVD illustrating the necessary maintenance steps performed on the equipment.

Four, "meta-procedures" replace unique code with generic, multi-functional macros that reduce the memory requirements for large applications. The lean program architecture frees valuable internal memory and allows for more efficient program performance.

Scrubber consists of nine "meta-procedures" to accommodate the various looping strategies of the user-machine interface. One "meta-procedure" should not contain an entire program unless the program is very simple in structure.

PROGRAM STRUCTURE

The Scrubber program primarily consists of descriptive and diagnostic information. The user directly accesses descriptive information about system components and maintenance. But, the program controls the presentation of diagnostic information and prompts the user for necessary information to identify a maintenance solution.

Describe Function

The Describe facility contains the following technical information:

- Textual description of components
- Visuals of components
- Maintenance procedure demonstrations

The textual and visual presentation represents information from the General and Functional sections of the Air Monitoring System technical manuals. The user controls the display of information through a series of menus. The following options provide user control of the video sequences:

- Proceed to Next Maintenance Action
- Repeat the Current Maintenance Action
- Review the Previous Maintenance Action

At the conclusion of a video maintenance demonstration, the program prompts the user if any additional maintenance procedures need review. If not, the program returns to the Describe facility.

Diagnose Function

The diagnostic facility contains the ES. The ES organizes the knowledge of the human expert in the form of facts and rules within the knowledge base of the program. These rules control the dialogue between the user and the knowledge base to troubleshoot the equipment problem.

First, the program obtains the information it needs to best reduce the number of potential problem areas. The gage readings on the front panel of the instrumentation represent this vital information. Next, the program decides which problems remain possible and asks questions to either support or negate the existence of a particular class of problems. To further reduce the scope of the search, the program asks for additional information to either support or negate a given claim. The ES continues to consider problems and compile information until it identifies the problem source. Then Scrubber presents the appropriate audio/visual repair instructions through user control.

PROJECT EXTENSIONS

1987 Extension

The 1987 extension of the Scrubber project consists of the creation a self-loading videodisc. This remastered disc contains all digital program, graphics, audio, and IMSATT language information, which is accessed by the microcomputer when necessary. The current configuration requires no knowledge of computer commands for operation. After the videodisc is inserted into the player, the program prompts the user to begin.

The operation of the disc requires the addition of three circuit boards into the Sony videodisc player. These boards decode and transfer the digital information from the videodisc to the microcomputer. The videodisc contains examples of compressed audio as a test of this emerging technology.

Formal evaluation of the product will occur in the near future to assess user acceptability and ease of operation. Initial evaluation reveals a high noise level in the compressed audio examples. The 1987 project extension increases the transparency of the system to the user.

1988 Extension

The 1988 extension moves the Scrubber project into the realm of optical disc technology, which includes write-once-read-many (WORM) and compact disc-read only memory (CD-ROM). This effort builds on the digital storage accomplished in the 1987 extension. The 1988 extension consists of the digitization of selected video frames from the videodisc and the production of a WORM version of the Scrubber program.

The proposed WORM disc will contain digital program, graphics, audio, and IMSATT language information with the addition of digitized video images and index files. This project will explore the storage and retrieval capabilities of optical disc technology as a medium for the future delivery of technical information.

CONCLUSION

The Scrubber project addresses four NNS concerns:

- Evaluate the benefits of ES for Integrated Logistic Support
- Evaluate the IMSATT ES capabilities
- Develop capability for ES development
- Evaluate the IMSATT ES applicability to training, diagnostic, and maintenance

This project demonstrates that ES technology can be applied to a small, focused subject to provide interaction in the manner of an interactive tutor. Because of the extensive development required for even a limited subject, ES technology should be applied to a focused area that would benefit from the technology, such as information retrieval or problem resolution. ES proves to be ideal for diagnostics because it can reason through possible causes to find solutions. A program that combines conventional and artificial intelligence programming can assist a maintenance technician and provide an integrated record of maintenance history and diagnostic activities.

IMSATT contains a powerful, easy-to-use facility for ES programming. The language resembles the programming structure of the artificial intelligence languages of PROLOG and LISP with its procedures for FACTS, RULES, and INFERENCE. The power of the IMSATT language greatly simplifies the programming of an ES.

The team members (designer, programmer, and video producer) gained valuable experience in developing Scrubber. The team has experienced the many ways in which ES differs from typical computer-based presentations. The designer has applied the techniques of Knowledge Engineering and ES development. The programmer has developed code using the unique formats of artificial intelligence languages. The video producer has videotaped and edited the many detailed, short segments required to support the depth of information required for the ES. The project has prepared the team to pursue future work in ES.

Throughout the project, many themes of the Computer-Aided Logistic Support (CALS) Initiative have related to Scrubber. The program addresses many CALS issues, such as disc-based logistic information, expert maintenance systems, automated information access, and computerized shipboard training. Scrubber represents an initial step toward the realization of CALS.

ABOUT THE AUTHORS

DR. DENNIS HRIBAR is a Training Program Developer at Newport News Shipbuilding. He is responsible for the production of training products to include curriculum for operation and maintenance courses, videodiscs, and expert systems. His experience includes fifteen years developing and presenting instruction and six years designing microcomputer programs. He was formerly a Curriculum Developer and Instructor at the Armed Forces School of Music. He holds a Doctorate from Vanderbilt University in Educational Administration.

MR. JOHN DIDURO is a Logistic Engineer at Newport News Shipbuilding. He is responsible for computer program development such as the incorporation of videodisc and expert system technology into training and maintenance applications. He has extensive experience using the IMSATT Authoring System and teaching IMSATT Workshops. Mr. DiDuro holds a double Bachelor of Science degree in Mathematics and Physics-Engineering from Washington and Lee University.