# Broadening the Interface Bandwidth in Simulation Based Training

Larry E. Somers
MICROEXPERT Systems, Inc.
24007 Ventura Blvd. Suite 210, Calabasas CA 91302

## ABSTRACT

Currently most computer based simulations rely exclusively on computer generated graphics to create the simulation. When training is involved, the method almost exclusively used to display information to the learner is text displayed on the CRT. MICROEXPERT Systems is concentrating on broadening the communications bandwidth between the computer and user by employing a novel approach to video image storage combined with sound and voice output. An expert system is used to combine and control the presentation of analog video, sound, and voice output with computer based graphics and text.

We are currently involved in the development of several graphics based user interfaces for NASA, the U.S. Army, and the U.S. Navy. This paper will focus on the human factors considerations, software modules, and hardware components being used to develop these interfaces.

## INTRODUCTION

Advances in military and aerospace technology continue to result in increasingly complex systems requiring quick, accurate decisions under increased cognitive loads. The amounts, variety, and rate of information flow is, many times, so overwhelming that anticipated performance benefits are not realized (Rouse 1987).

Recent advances in both video and audio storage technology are providing additional resources for communications channels between computer and user. These tools may well contribute to potential solutions of the problem. This article outlines an approach we have taken in combining these tools for the development of user interfaces, including intelligent human-machine interfaces for simulation based intelligent tutoring systems (ITS).

## HUMAN FACTORS

User capacities and needs have been described as a major consideration in designing user interfaces (Shneiderman 1987). The use of several media devices can help to better meet the needs and match the capacities of the user. Described below are several of the more important factor we have considered in developing a multimedia interface.

**Cognitive Load.** A measure of the complexity, or difficulty of a task is the number of resources it requires (Moray 1977). As described by Baecker (Baecker 1987) the cognitive load of a task correlates with such factors as:

- learning time
- fatigue
- stress
- proneness to error.

It is important that the interface help minimize the cognitive load on the user. Thus, for example, the design should consider the different loads imposed in making menu selections with a one, two, or three button mouse, respectively. It may turn out that the one-button mouse has the lowest load, since there is no overhead in determining which button to select. However, in the larger context, it may turn out there is a greater penalty in, for example, an increased number of menus or menu selections that must be provided.

**Interference.** Degradation in the performance of one task can occur due to competition for cognitive resources by another task during the same time period. Problem solving requires attentive behaviors that usually involve large numbers of cognitive resources. As a result, problem solving during an ongoing simulation is highly susceptible to interference. For example a tutor that provides text for coaching during a simulation could easily interfere with the simulation reducing, instead of improving, the user's performance. In such situations an alternate communications channel using voice output or auditory cues may provide a better approach to prompting the learner without interfering with their performance.

In working with a simulation based intelligent tutoring system, there are two classes of problems that confront the user: operational and functional. Operational problems have to do with the means of operating the ITS itself. Functional problems deal with learning to perform the tasks the tutor was designed to teach. Operational problem solving often interferes with functional problem solving.

One objective of the user interface is to minimize operational problem solving. All resources expended at this level are diverted from the functional problem for which the computer was adopted in the first place. Design features such as consistency, compatibility, icon and menu design must be considered. For example operators of certain types of radar learn to access radar target information by using a joystick to position a cursor on the target and then pressing the joystick button. We have designed a simulation to train radar operators that not only simulates this operation but also provides additional information about radar symbols and controls using a very similar procedure. If, for example, the learner desires information about a symbol he does not recognize on the simulated radar display, he need only position the cursor on the symbol, using the joystick, and press the help button on the keyboard. This type of learning requires only slight stimulus generalization and is therefor easily learned by the student.

The overhead of functional problem solving can also be reduced by careful design. Information should be presented using symbols, jargon, and metaphors that are, as much a part of the users repertoire and experience as possible. In training radar operators we have employed two expert systems, a scenario expert and an interface expert. The interface expert compares the actions of the scenario expert with the actions of the user. When a discrepancy occurs the interface expert provides visual or audio coaching, during the scenario, without the learner having to request help in any specific way. Transcripts and recording made of radar instructors as they trained operators were used to design the voice output which includes training and operation related jargon already familiar to the trainees. The result is very similar to the classroom training the operators receive in which an instructor stands behind a student and provides coaching as the student operates the radar console.

**Skill Acquisition.** Simulation based training generally focuses on skill development. Training procedures, including help systems, are a part of the user interface. Their design should encourage development of skills in an isolated, non-threatening way. It is important that voice and sound output, for example, not be punishing to the learner, especially by drawing attention to the learner from his peers. The result is often an avoidance or aggression response by the learner which will decrease skill acquisition.

There is some evidence that skill is acquired more rapidly in an isolated learning situation (Schneider 1985). This may not hold for specific cases and requires testing for final validation. High-fidelity simulations are ultimately important in order for the advanced student to learn fine discriminations. However, for the novice it is often important to reduce the complexity of the simulation so that the student can more easily learn to make important preliminary discriminations. In training radar operators, the complexity of the simulation scenario is controlled by the interaction expert. As the student becomes more successful at solving the scenario correctly, the complexity is increased by adding additional targets and target types and by changing target vectors. If a student has difficulty with a specific scenario, the scenario is simplified so that important stimuli are isolated and the student can more easily focus on appropriate discriminations to be learned.

**Mental Models, Analogy, and Metaphor.** The underlying conceptual model of the software is considered to be a more important factor in user-friendliness then what is generally called "look and feel" of the system (Liddle 1989). The mental model which the user applies in trying to understand and predict systems behavior is an important consideration in the design of the interface. Users make use of analogy between systems components and previously learned stimulus-response paradigms, when operating a system. To the extent that the user interface can be designed using one or more carefully chosen metaphors familiar to the user, the interface will be perceived as user-friendly. In designing the user interface to multimedia database, in which the user can access analog video images, graphics, voice, sound, and text, we have employed the metaphor of a Library. A metaphor of a card catalog is used to specify the information used for a database search. Following the search a graphical representation of library books on a shelf, representing the results of the search, is displayed on the screen. By pointing the cursor at a book and clicking, with a mouse, the information, be it text, sound, voice, or image, is displayed to the user. Though still in the prototype stage, preliminary user acceptance has been very positive so far.

**S-R Compatibility.** When a systems cause-and-effect behavior matches the user's expectations and previous experiences, it has good stimulus-response (S-R) compatibility. Two main factors to be considered are spatial congruence and custom. Having good spatial congruence between items in a menu and the layout of function keys provides good S-R compatibility. The use of the color red to indicate danger or a stop action is an example of how custom can be used to provide good S-R compatibility. In a similar fashion the user interface should be designed to make use of customs specific to the individuals that will utilize the system. Through careful knowledge engineering it is sometimes possible to uncover customs peculiar to the target group of users. To the extent that these customs can be incorporated into the interface it will be perceived as user friendly.

## INTERFACE COMPONENTS

The diagram shown in figure 1, below, illustrates the functional modules we have used in developing intelligent human-machine interfaces. Each module is a unit of replaceable code with specified inputs, outputs, and functions to perform. Furthermore the interface, itself, can be seen as a module in the development of a larger intelligent tutoring system. In this way other groups are able to work separately on different modules of the ITS.

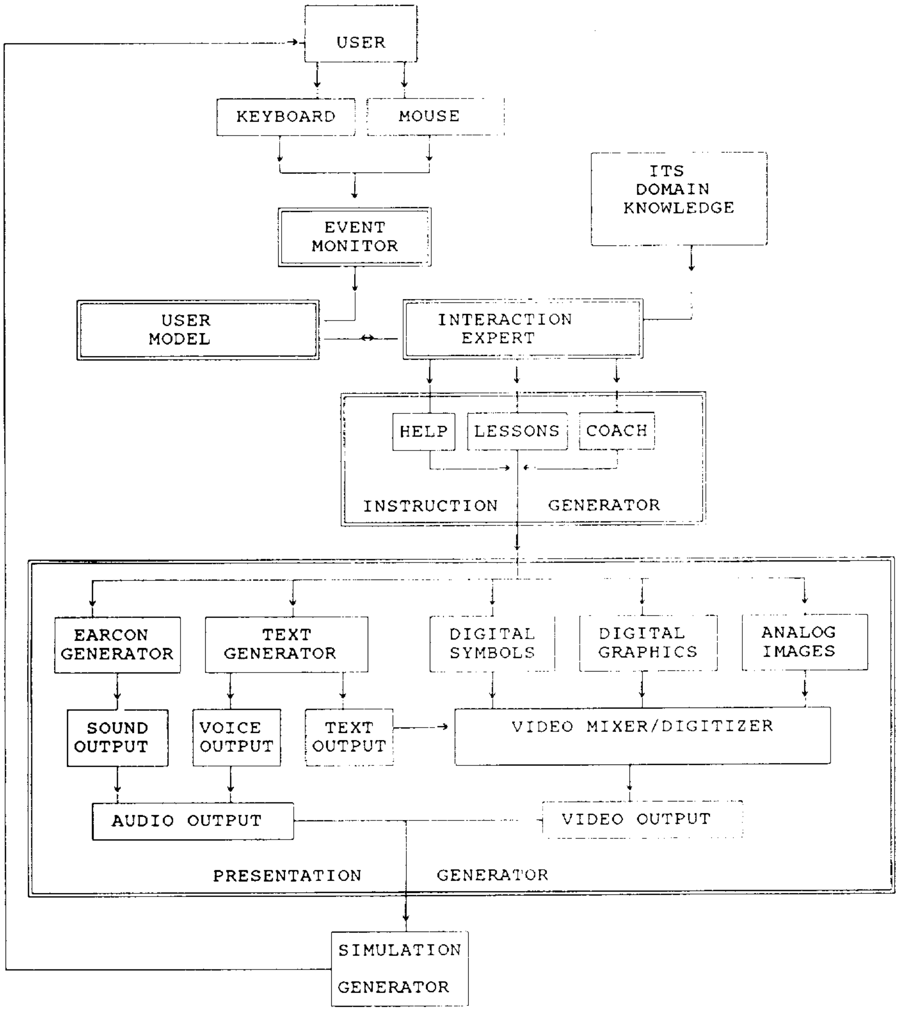

FIGURE 1
Interface Components

ORIGINAL PAGE IS
OF POOR QUALITY

**Task Analysis.**

While not represented as a separate interface component, a careful task analysis is essential to the development of the other components in the system. Intelligent Tutoring Systems attempt to capture and explicitly represent the knowledge that constitutes the expertise being taught. Our knowledge engineering efforts have focused on a task analysis that not only identifies the knowledge components to be represented, but creates a curriculum structure that associates knowledge components with each other and with the goals of the instruction.

During the knowledge engineering phase of development complex, high-level tasks are identified and decomposed into mid-level and then low-level unit tasks. For each unit task it is important to identify a measurable behavior associated with the task, the stimulus conditions upon which that behavioral response should be made, and the heuristics that describe the relationships between stimuli and responses. The process is an adaption of the goal-lattice structure described by Lesgold (Lesgold 1988). Each high-level task serves as the root node of a tree. Simple lessons are designed to teach the unit tasks of each tree. Many of the mid-level and unit tasks identified in one task tree are also common to other, separate, task trees. Figure 2, below, shows this architecture symbolically.

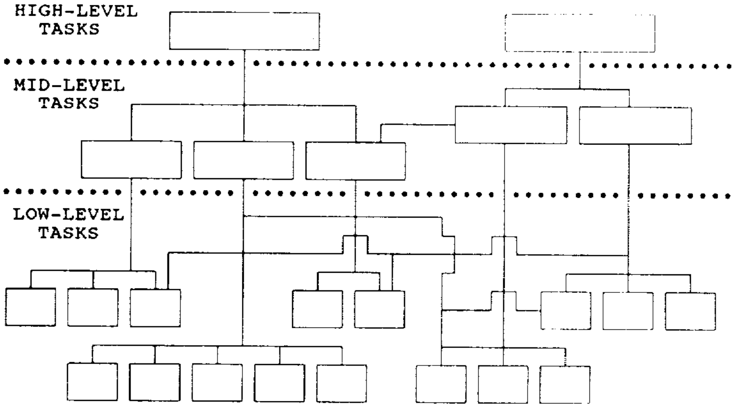

Figure 2.

Task Analysis

The resulting task trees and interconnections make up a curricular structure for the ITS which is accessed in the interaction expert. Tasks can be taught using a depth-first search, a breadth-first search, or both. Research is being carried out to determine, among other things, under what condition a specific search should be carried out.

**User Input.**

Prototype development has been carried out on a Symbolics LISP machine, DEC MicroVax. User input has been limited to a mouse pointing device and keyboard. We are currently developing a new type of wireless pointing device to be implemented when porting the interface to a PC. We are also considering voice input devices for entering commands on the PC.

**Event Monitor.**

The event monitor measures user and simulation event actions over time. Multiple timing functions are available to measure the elapse time between a task stimulus event and a specific user response (task time), between the start of sequential tasks (intratask time), to measure input from the keyboard and mouse, to determine the current task to be performed, the current position of simulation related objects on the display, and which object the cursor is pointing to at any given moment. Information measured by the event monitor is then stored in the user model.

**User Model.**

The user model is used to store task performance related data about the user. For each task performed, the time required by the user to complete the task is stored. The sequence of user performed tasks is also stored and used to calculate a task efficiency and task similarity (compared to an expert) rating. The time period between presentation of successive task stimulus conditions is also measured and provides an indication of the cognitive load on the user. This provides a user-specific fact base that is used by the interaction expert to adapt to individual user requirements and needs.

Also stored in the user model is data related to the users presentation preferences. As is described below, information can be presented to the student in a variety of modes, textual, graphical, voice, and sound. The user model is designed to measure the users preference for a specific mode of presentation as defined by his performance following the presentation.

The users teaching history is also tracked in the user model. Thus the tasks that have been taught, the presentation modes that have been used, the students task performance, and his presentation preferences are stored here and available to the interaction expert.

**Interaction Expert.**

The interaction expert is the interface rule base. Rules are designed to compare the users task performance with that of an expert. An expert system, designed as a separate component of the ITS (not shown), generates expert solutions that are available to the interface expert. The expert's solution is compared with the users solution to determine the tasks to be taught. By traversing the curriculum lattice the interaction expert determines related tasks that should be taught as well as different paths (viewpoints) from which to teach. The user model is then consulted to determine what paths have not been previously attempted for that user and what presentation mode should be tried.

**Instructional Generator.**

The instructional generator is primarily a database of instructional components designed to teach specific tasks. Instructional modules are designed to provide several instructional strategies; discovery learning, coaching, and Socratic dialog. Thus, several instructional modules are available for each task. Modules are also designed to differ in their emphasis of a specific presentation media. For example, coaching is available for a given task by presenting text on the video display or through voice output using a text-to-speech converter.

**Presentation Generator.**

The presentation generator consists of the media devices used to present information visually or auditorily along with software used to control these devices and integrate components.

**Visual Channel.** Both analog and digital, bit-mapped video images are available for display to the user. Currently different video display terminals are used for each. We are experimenting with both video digitizing boards and video mixers to combine both types of images onto one display.

**Video.** A unique video storage device, the VIEWBOX 2000, is being used to capture and display analog, RS-170, video images. The device uses a standard 20-Mbyte hard disk with a modified controller to store over 2400 RS-170 video images. Random access times are approximately 200 msec and sequential access times are under 100 msec. making a "pseudo-animation" possible. A standard video camera is used to capture images. Software drivers in the presentation generator are used to control the device over the computers RS 232 port.

**Graphics.** Graphic displays are highly machine dependant. Interfaces are currently being designed on both Symbolics and DEC MicroVax computers, using monochrome graphics, and on PC's using EGA color graphics. Currently simulation are graphics based and the VIEWBOX is used to display visual information that does not lend itself well to graphical display due to processing requirements and capabilities. We are experimenting with using the VIEWBOX to provide background scenery overlayed with graphics in the hopes of combining both in the future.

**Symbology.** Icons and symbols are separate graphical components the interface uses to help the learner make important discriminations during the simulation. Simulations are designed with varying complexities. Novices are provides simulations of very low complexity with ample use of symbols, such as pointers. While it is generally agreed that high-fidelity simulations are needed, it is possible too provide to much fidelity early in the learning process.

**Text.** Under the control of the instructional generator text can be displayed in a window on the video display or sent to a text-to-speech converter and presented as speech. In the later case the presentation generator formats the text string to control pitch, rate, and other parameters.

**Audio Channel**

**Producing Speech Electronically.** Generation of speech and sound (earcon) output from a computer requires special hardware components. Three major techniques for production of speech have evolved over the years: formant (resonant frequency) synthesis; linear predictive coding; and waveform sampling. Most commercial text-to-speech devices use one of the first two because they require smaller storage and slower data rates. However with as computer memory continues to decrease in cost, computer systems such as the Atari and Apple's MacIntosh are imbedding the hardware and software needed to sample and reproduce waveforms.

**Synthetic Speech.** The automatic conversion of text to synthetic speech has advanced remarkably in the last several years. A number of commercial devices are now available, ranging in cost from approximately $100 up to $35000. Progress in this area has resulted from advances in linguistic theory, acoustic-phonetic characterization of English sound patterns, perceptual psychology, mathematical modeling of speech production, and computer hardware design (Klatt 1987). Never-the-less a number of scientific problems remain that prevent current systems from achieving the goal of completely human-sounding speech.

The quality of voice output improves greatly in devices costing over $3000 (Kaplan et al 1987). In the $3000 - $4000 price range two text-to-speech devices stand out. Originated by Dennis H. Klatt, speech synthesis expert at MIT, DECtalk by Digital Equipment Co. has a broad range of voices including a child's voice and a female voice. In evaluations by Nusbaum et al (1984) listeners understood synthetic speech produced by DECtalk 97.7% of the time as compared to 99.4% for human speech. A rival system also originated by Klatt, the Prose 2000 by Speech Plus Inc. has similar quality but offers only a male voice and is slightly less expensive. Studies by Logan et al. (1986) indicate listeners have an error rate of 6% listening to the Prose 2000 - 3.0 compared to 1% error in understanding natural speech. Both devices can be controlled thorough the computers RS-232 Serial Port and require a data rate of approximately 100 bits, based on a typical rate of 12 phonemes per second.

We are currently using the Prose 2000 for text-to-speech conversion in several of our interface. A major advantage to these type of devices is the ability to use variables to store speech output. The major drawback of these devices is that they are limited in their ability to produce other complex sounds that would be useful for generating auditory cues.

**Voice Sampling.** A second method of producing digitized voice output is by sampling the waveform of human speech. Waveform sampling uses a common analog-to-digital conversion and requires about 64000 bits per second for uncompressed speech (8000 samples per second to capture up to 4000 Hz, multiplied by 8 bits per sample). Thus storage requirements would be 8K/second. Using a dedicated microcomputer containing a 20 megabyte fixed disk approximately 2500 seconds of speech could be digitally recorded using this method. Using data compression techniques, this number could be doubled. The results are a digital recording of the speech that is almost indistinguishable from the original source.

We are currently using an Antex Model VP 620E, PC compatible digital audio processor (Antex Electronics, Gardena, CA) to provide digital audio in some interfaces. While this type of device eliminates the ability to easily store speech components as variables, the high quality sound makes the device ideal in many teaching situation and where sophisticated auditory cues are desired.

**Earcons**. Sound is increasingly being used to convey information in computer interfaces. The term Earcon (Sumikawa 1985) has been used to define sounds that serve as the auditory equivalent of Icons. Similar to voice generation, earcons can be produced by sampling specific sounds or synthesizing sounds with a tone generator. Gaver (1986) has classified auditory icons into three groups: 1) symbolic, such as telephone bells and sirens, 2) nomic, in which the sound is a physically caused by the source such as the sound arriving mail makes in a mailbox, and 3) metaphorical such as a change of pitch used to represent falling or a hissing sound to represent a snake. Symbolic sounds are, perhaps, easiest to produce on most computers since they do not require the ability to sample sounds. However they generally require the greatest amount of learning on the part of the user. For this reason they should be used judiciously. Symbolic sounds have been shown to be effective when used as an alerting cue prior to emergency messages produced by synthesized voice (Hakkinen 1984). Anecdotal evidence from our current research supports these finding but also suggests that overuse of sound stimuli results in confusion of the user. We are now beginning to experiment with sampled sounds to produce nomic and metaphorical earcons which should require less learning by the user.

## CONCLUSION

A generic intelligent multimedia interface has been described. While research is still ongoing in many cases we have reach so interesting preliminary conclusions. We originally believed that selecting different presentation modes, e.g. voice or text, would be useful for adapting to specific types of learners. However results so far suggest that user performance improves much quicker when several modes, e.g. voice and text, are combined. This makes sense in light of the fact that the learner then comes under multiple stimulus controls.

A second factor, eluded to above, that became immediately noticeable was that earcons and auditory cues can easily be over used and become distracting to the user. However, when designed carefully, and used fastidiously, they can be of significant value in gaining the learners attention and improving his performance.

## REFERENCES

Baecker, R.M., Buxton, W.A.S., "Cognition and Human Information Processing," READINGS IN HUMAN-COMPUTER INTERACTION, Morgan Kaufmann Publishers, Los Altos, CA, 1987, pp.209.

Hakkinen, Markku T., Williges, Beverly H., "Synthesized Warning Messages: Effects of an Alerting Cue in Single- and Multiple-Function Voice Synthesis Systems," HUMAN FACTORS, Santa Monica, CA, 26, 2, April, 1984, 185-195.

Kaplan G., Lerner, E., "Realism in Synthetic Speech," in HUMAN-COMPUTER INTERACTION, R.M. Baecker & W.A.S. Buxton Eds. Morgan Kaufman, Los Altos, 1987, p. 414-419.

Klatt, Dennis H., "Review of text-to-speech conversion for English," J. ACOUST. SOC. AM. 83(3), 1987, pp.737-793.

Lesgold, Alan, "Toward a Theory of Curriculum for Use in Designing Intelligent Instructional Systems," in LEARNING ISSUES FOR INTELLIGENT TUTORING SYSTEMS, 1st Ed., Springer-Verlag, New York, NY, 1988, pp. 117-118.

Logan, J.S., Pisoni, D.B., "Preference Judgements Comparing Different Synthetic Voices," J. ACOUST. SOC. AM., Suppl. 1, 79, 1986, p. S24.

Liddle, D., in LaPlante, A., "Ease of Use Involve More Than Graphical Interface," INFO WORLD, Menlo Park, CA, 11(7), Feb. 13, 1989, p.1.

Moray, N., WORKLOAD MEASUREMENT, New York, Plenum, 1977.

Nusbaum, H.C., Pisoni, D.B., and Schwab, E.C. "Subjective Evaluation of Synthetic Speech: Measuring Preference, Naturalness, and Intelligibility," Speech Research Progress Report 10, Indiana Univ., Bloomington, IN, 1984, pp. 391-408.

Rouse, William B. Norman, Geddes, Norman D., & Curry, Renwick E., "An Architecture for Intelligent Interfaces: Outline of an Approach to Supporting Operators of Complex Systems," Human-Computer Interaction, Vol. 3, 1988, pp. 87-122

Schneider, W., "Training High-Performance Skills: Fallacies and Guidelines," HUMAN FACTORS 27(3), 1985, pp. 285-300.

Shneiderman, Ben, "Human Factors OF Interactive Software," DESIGNING THE USER INTERFACE, Addison-Wesely, Menlo Park, CA, 1987, pp. 4-11.

Sumikawa, D.A., Blattner, M.M., Joy, D.I., Greenberg, R.M., "Guidelines for the Syntactic Design of Audio Cues in Computer Interfaces," Lawrence Livermore National Laboratory, 19th Hawaii International Conference on System Sciences, Oct. 3, 1985.