Proceedings of the 1985 Winter Simulation Conference
D. Gantz, G. Blais, S. Solomon (eds.)

A MAN-MACHINE INTERFACE SIMULATOR

S. L. Morgan, J. J. Kindon and A. Nauda
HRB-Singer, Inc.
Box 60, Science Park
State College, PA 16804

ABSTRACT

An improved software development tool for the simulation and development of man/machine interfaces is described. The Man/Machine Interface Simulation Tool (MMIST) is a versatile, interactive software system which resides in a VAX 11/780 with a device independent graphics package. With this advanced capability, designers can develop and demonstrate detailed interactive displays/menus and processing sequences early in the software design phase. MMIST capabilities include a robust display/menu construction tool set, a friendly interface for inexperienced users, and a strong simulation capability to allow end users to examine and critique the menu hierarchies and display formats before any production software is generated.

The use of MMIST is demonstrated through sample menus and graphic output. Simulation capabilities are illustrated with examples of selected display sequence simulations.

BACKGROUND

A major concern of the software industry is the high cost and relatively low productivity rate for software development. The increased size and complexity of developed systems has resulted in software costs exceeding hardware costs by an increasing ratio [1,2]. Due to this trend, the Department of Defense has begun an initiative [3] to improve software productivity, establish goals for productivity improvement, and encourage and guide research in areas such as software engineering methodology and development tools.

The specific problem addressed is the lack of software development tools to expedite the development of realistic man/machine interfaces that can be critiqued and examined during system definition and preliminary/detailed design. This deficiency is particularly marked in the area of interfaces employing high resolution, color computer graphics displays, where traditional pen and ink design aids cannot communicate either the significance of the use of color or the flow of the interaction. Therefore, a reviewer at system definition and design time cannot experience or appreciate the MMI interaction and operational consequences of its implementation. This often causes design change requests subsequent to the implementation, which impact cost and/or schedule if implemented, or lead to operational rejection when the system is fielded if the changes are not implemented. For example, Figure 1 presents an interface presently designed using traditional tools.

The commercial marketplace offers some display construction tools. These range in diversity from "paint" programs for microcomputers to very complex image processing capabilities in support of the video production industry. These tools address graphic construction but do not provide the simulation capabilities to construct a frame connectivity data base to permit subsequent interactive simulation. They also do not provide capabilities for inserting timing delays into simulations. Moreover, these tools are primarily aimed at producing free form art rather than statistical plots and information displays.

REQUIREMENTS

The objective of this MMIST is to improve the development productivity on systems that include interactive graphic displays in their man/machine interface. The productivity gain is focused on doing a better job of detailing and demonstrating interactive displays/menus and processing sequences early in the design phase [4, 5]. Three capabilities are considered to be instrumental in achieving this objective:

- A robust display/menu construction tool set to permit sufficient detail
- A friendly user interface to allow inexperienced users to comfortably utilize the capability
- Strong interface simulation capabilities to allow end users to examine and critique the menu hierarchies and display formats before any production software is generated

It is anticipated that this MMI tool will improve productivity by identifying unsatisfactory elements in the man/machine interface design early in the design process before any production code is generated, rather than later in the process when rework and reimplementation will force costs to be higher and will delay development schedules. Rather than to advance the state-of-the-art in graphics display technology, the aim of MMIST is to advance the state of engineering practice in MMI design and to produce higher quality designs.

MMIST provides software developers the ability to construct sample displays and interactive menus and to construct prototypical interaction sequences. The capabilities include:

- Display/Menu Construction Tools
    - Placement of text, lines, polygons, and symbols
    - Control over text size, justification and font
    - Color assignments
    - Color fill of user specified areas
    - User symbol generation

- Man/Machine Interface Simulation
  - Interactively define connectivity between displays/menus
  - Sequences of displays to appear in a user defined order
  - Timing delay capability to better simulate time required to generate displays
  - Data base tools to manage scenario development

IMPLEMENTATION

MMIST runs on a VAX 11/780 computer. Figure 2 presents the implementation environment. The software includes the Precision Visuals DI-3000 computer graphics software library. The DI-3000 package is written to the ACM-SIGGRAPH CORE graphics standard and is device-independent. Device-independency permits MMIST to run on a wide variety of graphics devices including:

- RAMTEK 9465 color display device
- DEC VT241 color terminal
- TEKTRONIX 4014 monochromatic terminal
- VERSATEC printer/plotter (monochromatic hardcopy)

The RAMTEK 9465 color display device is the normal workstation for the MMIST operator. The RAMTEK provides 1280 x 1024 resolution of up to 256 simultaneous colors and offers both trackball and joystick interaction. The software is developed in FORTRAN 77.

The approach employs three units or sybsystems: the Display/Menu Construction Subsystem, the Display Hierarchy Definition Subsystem, and the Storage and Retrieval Subsystem. Figure 3 presents the software hierarchy.

Display/Menu Construction Tool Subsystem.

This subsystem consists of software tools developed to construct the individual displays and menus that form the basic CRT elements of the interactive system under design. The major elements of this subsystem are symbol generation, symbol placement, symbol scaling, symbol color selection, and symbol rotation. In this usage, the term "symbol" stands for text, line segments, regular polygons, circles, ellipses, user-defined symbols, grid lines, etc.

The symbol generation capability allows the user to sepcify the current display attributes for line style, line width, character font, current polygon, grid line density, and the design of user-defined symbols. Symbol generation provides menu areas for the selection of display attributes. The user is limited to the displayed attribute alternatives and uses graphics cursor interaction to select from the displayed alternatives.

The symbol placement capability permits the user to locate the current symbol type in the display and to manipulate the currently displayed symbols. These manipulations include the initial placement of a symbol, the coping of a symbol to another location, the moving of a symbol to a different location and the removal of a symbol. In addition, symbol placement supports "rubber-banding" functions for rectangles and circles, thereby allowing the user to create rectangles and ellipses of any proportion. Symbol placement also permits the user to construct closed polygons of any shape by chaining up successive line segment placements. Polygon closure is performed by symbol placement to ensure that polygons are properly filled. Symbol placement also provides for the entry of text strings and the positioning of the text on the display.

Symbol scaling performs the function of changing the displayed size of a symbol. This function is limited to the scaling of user-defined symbols and some regular polygons; e.g., triangles, hexagons, etc. The size of text is controlled in symbol generation and the "rubber-banding" capability for rectangles and circles present in symbol placement can generate these symbol types at any size and proportion.

Symbol color selection allows the user to manipulate the color set used for the display background and symbols. The current fill color is specified by the user from the symbol color selection menu. The user is able to select the colors to be utilized from a palette of 78 colors.

Symbol rotations allow users to alter the justification of text and the orientation of regular polygons and user defined symbols. The user can rotate the symbols about the centroid of each symbol in intervals as small as one degree. Cursor interaction, rather than keyboard input is used to indicate the degree of rotation desired.

Display Hierarchy Definition Subsystem.

This subsystem provides the tools for the connection and sequencing of individual displays, and for permitting interactive selection to be made in going from one CRT graphic to several possible other CRT graphics. Individual displays may be used in many simulations. The connectivity between displays may be one-to-one (a linear flow), one-to-many (a hierarchy) or many-to-many (a network) as the designer requires. The designer defines either area on the screen (e.g., command boxes) or individual symbols on the display to be the indicator to retrieve the next display. Display Hierarchy Definition software builds a data base containing the display connectivity, the connectivity indicators, and an indicator as to whether or not the connectivity between pairs of frames is replacement or overlay in nature. Replacement connectivity means that when the user performs the activity on Display A that connects A to B, Display A is erased and Display B appears on the screen; i.e., B replaces A. Overlay connectivity means that when the user performs the activity on Display A that connects A to B, Display B is written to screen over Display A; i.e., B overlays A. This subsystem unit is made up of four subunits; Display Connectivity Data Base Maintenance, Display Hierarchy Definition Menu, Areal Extent Connectivity Specification and Symbol Connectivity Specification.

Storage and Retrieval Subsystem.

This consists of tools that provide the capability to store, retrieve and run a simulated interface. This unit performs both interactive and non-interactive execution. There is a capability for the storage and retrieval of individual frames to support the editing of frames with the construction tools and to allow the user to specify overhead timing delays for individual frames. The subsystem consists of four subunits - store/retrieve a frame, frame data base maintenance, interactive simulation execution, and automatic simulation execution.

Two modes of simulation execution are supported. The interactive mode permits a user to step through the menus and displays of the man/machine interface being designed, and thereby discover any weakness in the interface flow and redesign the interface accordingly. The automatic mode is particularly useful in demonstrating the interface to large audiences at design reviews. In automatic mode, the user is also allowed to hold a frame for detailed inspection or to "fast forward/reverse" through the display sequence. The complexity of the simulation available in the interactive mode is limited only by the number of frames constructed, and the display connectivity definitions.

An overhead timing delay capability adds an element of realism to a created simulation. System displays vary in the construction time required under real world conditions. The display of an outline map, for example, requires a significant amount of retrieval of spherical coordinates and conversion into display screen coordinates. Analysis of past company work indicates that as much as 85% of displaying an outline map is consumed by other than graphic computations. To better simulate the relative number of real world computations required to display different types of frames, the user is allowed to specify an overhead timing delay factor to be used to slow down the production of individual frames. A second overhead timing delay may also be specified to better simulate the differences between varying graphics hardware and software configurations; i.e., the same display requires different amounts of time to be constructed in differing hardware and software environments. This second timing delay is applied to all frames in a simulation to slow down the display of all frames by a finite amount. A frame may be subjected to either or both types of timing delay as the user requires. The frame data base maintenance unit supports the storage and retrieval of the simulation specific timing delay data associated with each frame as well as the list of graphics instructions which build the display.

MMI SIMULATION DEVELOPMENT AND USE

The process of developing a MMI simulation occurs in three steps. The first step is to use the display menu construction tools to interactively build each frame to be used in the simulation. In the example shown in Figure 4, there are three frames to be designed - two high-resolution color graphics frames with cursor interaction, and one "fill-in" alphanumeric menu. Using the display/menu construction tools, a user can draw lines, place text and assign colors interactively. The display/menu construction tools improve frame development in a way equivalent to the improvement that word processing offers over typing.

The second step in developing the simulation is to establish the connectivity between frames in the situation. Connectivity between frames may be defined in a variety of ways, including:

- Placement of the cursor over an area of a frame and performing a button push.
- Placement of the cursor over a symbol within a frame and performing a button push
- Entry of a character string via a keyboard and striking the return key

In this context, connectivity means any action that an operator can perform which results in the MMI simulation advancing to a next frame. Each frame may be connected to more than one frame, with most frames being connected to two other frames being connected to many other frames; e.g., an executive level function selection menu. The developer of an MMI simulation generates connectivity matrices for the simulation using tools provided in the display hierarchy definition. The first step in this process is to establish basic connectivity; e.g., that frame A is connected to frames B, D, and H. When the basic connectivity is defined for a frame, the operator then defines the activity which will cause the transition to occur. The operator also specifies whether the transition mode is replacement or overlay.

The third step in setting up the simulation is the specification of "frame specific" and "entire simulation" timing delays. To better simulate actual display construction times, the operator may specify a frame specific timing delay to be applied to each frame in the simulation. Furthermore, since not all graphic devices and host computer have the same throughput rates, the operator may specify a timing delay to be applied to all frames of a simulation to compensate for differences in hardware configuration.

Once these three steps have been completed, the simulation is ready to be run in the interactive mode. The operator initiates the interactive simulation execution software and proceeds to step through the simulation by issuing the commands defined in the connectivity data base and advancing through the simulation's frame set. The fidelity of the simulation is limited only by the complexity of the connectivity data base and the effort put into the construction of the individual frames. The developer of a simulation may initially elect to "rough out" the individual frames and define a simple connectivity structure, and later refine existing frames and insert new frames into the simulation, as well as increase the complexity of the connectivity structure. In this way, the simulation may be developed and refined throughout the design phase.

To construct an automatic mode simulation, the operator runs the interactive simulation software in a "record and save" command mode. This approach allows the automatic mode simulations to be very easily constructed. To run an automatic mode simulation, the operator brings up the automatic simulation execution software, specifies the name of the simulation, and indicates the control mode for the simulation. The control modes are:

- Fully automated using timing delays specified in the simulation's data base
- Fully automated but using a user provided time delay between frames
- Manual step through of simulation sequence; i.e., user hits return key to advance
- Manual override; i.e., user can step through and change direction of simulation sequencing

The variety of control modes permits the operator great flexibility in demonstrating the simulation. In the two fully automatic modes, the simulation runs itself, and in the two manual modes, the operator controls the pace of the simulation and can even elect to "back" the simulation up to a previous point and step through that portion of the simulation again.

CONCLUSION

The Man/Machine Interface Simulation Tool offers a new approach which allows systems to be modeled very early in the software development cycle. This allows early insight into the specification of performance requirements and can identify performance problems. This will ultimately reduce costs, reduce needed system maintenance, and increase system reliability.

Although MMIST is a new capability, it is already in use on three projects. Evaluations of the productivity and quality gains resulting from MMIST's use are still incomplete and will be reported upon at a later date. Initial results are most encouraging. The response from users has been very favorable and the customers are pleased with the additional visibility into the man-machine interfaces currently being designed with MMIST.

REFERENCES

[1] Dickinson, B., Developing Structured Systems - A Methodology Using Structural Techniques, Yourdon Press, 1981.

[2] Glass, R. L., "Recommended: A Minimum Standard Software Toolset," Software Engineering Notes, Vol. 7, No. 4, Oct. 1982.

[3] Strategy for a DoD Software Initiative, Dept. of Defense, Oct. 1982.

[4] Smith, S., "User-System Interface Design for Computer-Based Information Systems," MITRE Corporation, Report Number ESD-TR-082-132, April 1982.

[5] Foley, J., et al., "The Human Factors of Computer Graphics Interaction Techniques," IEEE Computer Graphics and Applications, Vol. 4, No. 11, pp. 13-48, Nov. 1984.

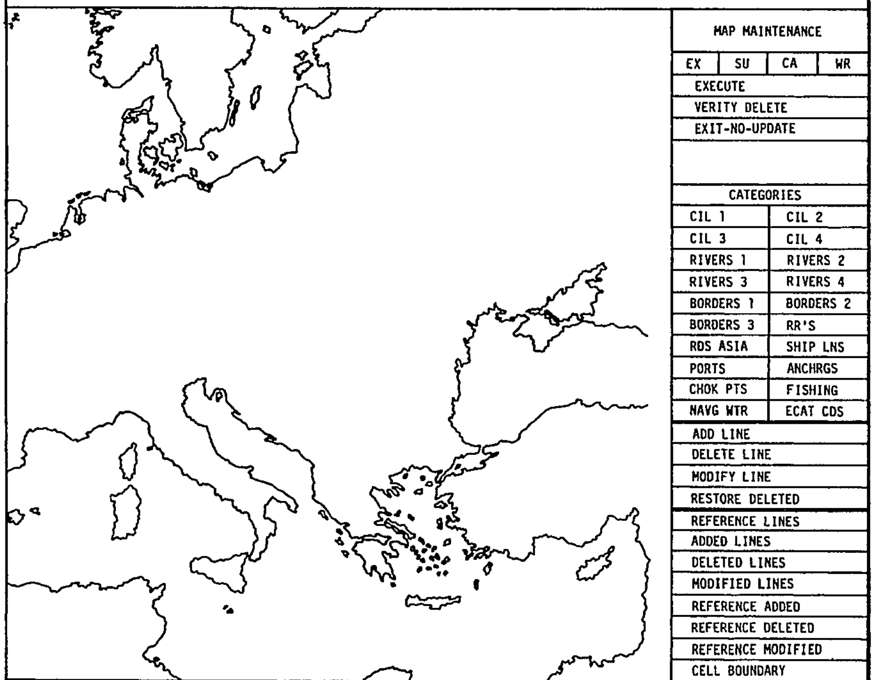

FIGURE 1. SAMPLE OF MAN-MACHINE INTERFACE DESIGN PRODUCT USING PEN AND INK TECHNOLOGY

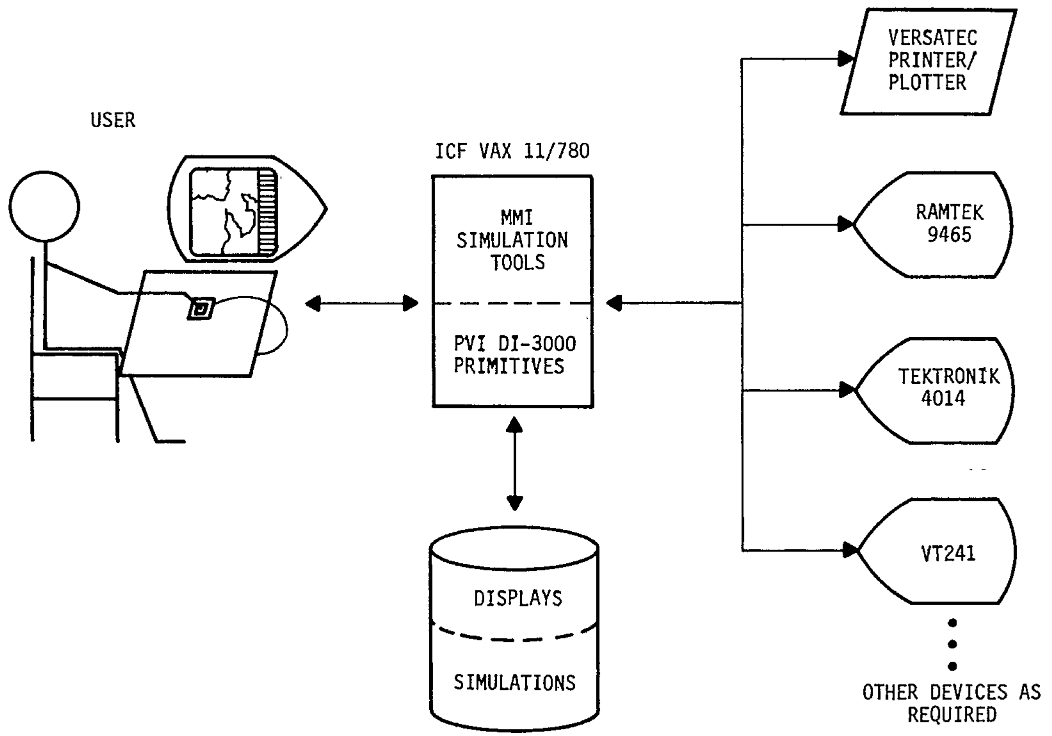

FIGURE 2. IMPLEMENTATION ENVIRONMENT

T85-13

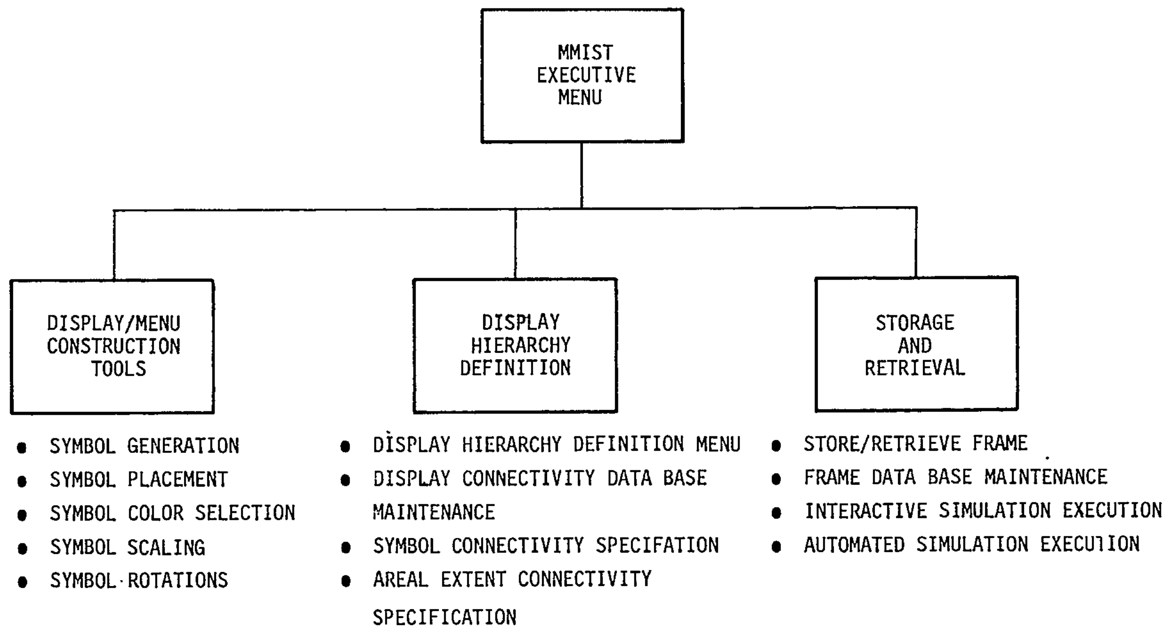

FIGURE 3. MMI SIMULATION TOOL SOFTWARE HIERARCHY

T85-13

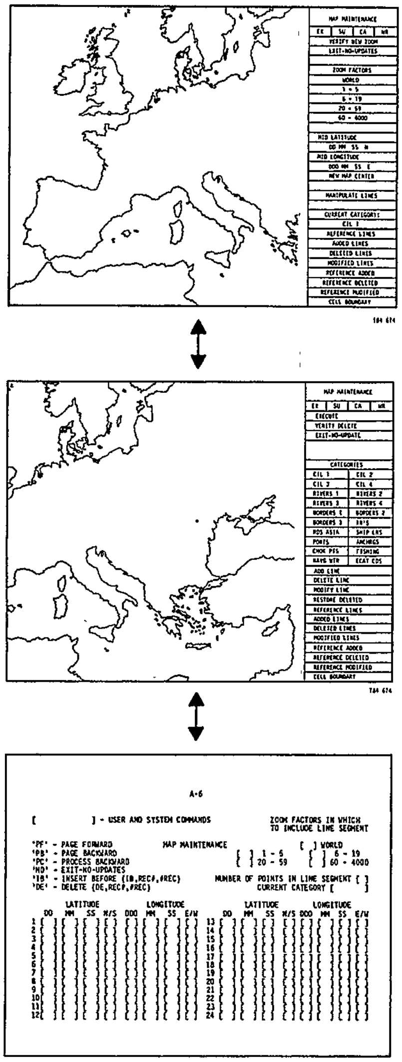

FIGURE 4. SAMPLE MMI DESIGN EFFORT PRODUCT

STEPHEN L. MORGAN has 13 years experience in the design and development of software systems for the processing and computer-assisted analysis of remotely sensed data, with emphasis on the development of man-machine interfaces using advanced computer graphics technologies and geographic information systems. Mr. Morgan is currently a Senior Engineer in the Processing Software Design and Development Department at HRB-Singer. An active member of several societies, including ACM, IEEE Computer Society, and the National Computer Graphics Association, Mr. Morgan holds degrees from Johns Hopkins and The Pennsylvania State University.

Mail Stop 123
HRB-Singer
Box 60, Science Park
State College, PA 16804
(814) 238-4311

JOHN J. KINDON is Manager of the Tactical Processing and Reporting Section at HRB-Singer, where he is responsible for software engineering related to the processing and analysis of data from a national collection system. Formerly on the technical staff of General Electric's Space Division and Boeing Computer Services, Mr. Kindon holds a degree in Aerospace Engineering from The Pennsylvania State University.

Mail Stop 123
HRB-Singer
Box 60, Science Park
State College, PA 16804
(814) 238-4311

ALEX NAUDA is a Princiapl Engineer in the Signal Analysis Center and Staff Assistant to the Director of Research and Development at HRB-Singer. A registered professional engineer, Dr. Nauda has served as a consultant to HRB and NASA in the analysis, modeling, and simulation of signal processing systems. Formerly a Professor of Electrical Engineering at Bucknell University and The Pennsylvania State University, Nauda received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from the University of Pittsburgh. He is a senior member of IEEE, the Society for Computer Simulation, and the American Society of Engineering Education.

Mail Stop 159
HRB-Singer
Box 60, Science Park
State College, PA 16804
(814) 238-4311