μuSE (Multidimensional, User-Oriented Synthetic Environment)[2] A FUNCTIONALITY-BASED, HUMAN-COMPUTER INTERFACE[3]

Creve Maples and Craig Peterson
Synthetic Environment Laboratory
Sandia National Laboratories
Albuquerque, N.M. 87185

ABSTRACT

The Multidimensional User-oriented Synthetic Environment, *μuSE*, is an open-ended software shell that provides a new approach to interacting with computer-based information. By using a real-time, device-independent software design and incorporating both cognitive and experiential models of human perception, *μuSE* greatly enhances a person's ability to examine, interact with, and understand relationships in complex information space. A *μuSE* shell may be wrapped around data, models, simulations or even complete programs. Using a new design approach, based on human functionality, it provides tools for the *presentation*, *exploration*, *navigation*, *manipulation*, and *examination of information*. Users experience a highly interactive environment, capable of dynamically mapping information into visual, auditory or kinesthetic representations. By eliminating the necessity for application programs to directly interact with devices, and by providing standard functional channels for user communication, the software environment significantly reduces and simplifies application code development. *μuSE* also permits both multiprocessor operations and distributed network-based, heterogeneously shared environments. The system currently supports flat screen, stereo, and VR operation (with full head-tracking), voice recognition, sound synthesis, data sonification, and a variety of commercial interactive devices.

1. Introduction

Richard Hamming said "The purpose of computing is insight, not numbers." Today, as the Information Age moves forward, computers touch all aspects of our lives. Far from achieving "insight", however, there is often a feeling that we are slowly sinking under the enormous volumes of data. Information processing has long been considered the province of the computer. Information processing is not, however, problem solving, path selection, or decision making. The time required for people to analyze and understand computational information is a critical factor in evaluating information and drawing conclusions. Helping people to explore, question, and understand complex information is an important criteria for future computational environments. Highly interactive human-computer environments can allow practical solutions to some problems far more rapidly than either human or computer operating independently. Synthetic environments may well provide the most effective approach for accelerating human perception of computational information.

Today people using 3-D graphics face an environment not unlike the software development environment in the 70's -- the hardware is changing very rapidly, software standards are still trying to stabilize, and the construction of graphic representations is very time consuming. For those attempting to develop the virtual, or synthetic environments, the situation is even more difficult. The demand for real-time operation and the need to support a more sophisticated level of user interaction make this one of the most difficult and challenging areas in the computing arena.

In the 1950's, John von Neumann observed that in any computer system consisting of a processor and memory, the critical factor was the link connecting them. Today, ever-increasing amounts of information exist in computer-based memories, but the final, and most important, processing must occur in the human mind. The time required for a person to analyze and understand complex information is often the limiting factor in solving problems, formulating strategies, or making decisions. A new generation of highly interactive, immersive systems offers the potential of greatly enhancing the communication bandwidth between the computer and the human mind. By developing techniques that permit people to more easily analyze and understand computer-based information (e.g., designs, data, or simulation results), we will greatly leverage investments in both computer systems and human talent.

Synthetic environments using these techniques can provide more rapid and comprehensible access to information for evaluation, planning, and training, and have application in wide ranging areas including science, engineering, economics, medicine, fabrication, design, education, and the real-time control of complex systems. Extending such systems to support network-based, simultaneously shared work places offers even greater potential for learning, training and the dynamic exchange of information. The objective of this work is to provide a uniform, device-independent, software foundation to support such highly interactive, information-rich, shared environments.

2. Background Issues

As our understanding of nature becomes more sophisticated, scientists (and non-scientists) have to deal with more and more information in order to accurately understand and represent real world systems. Complex, multi-parameter information may arise experimentally, from a variety of sensors or measuring devices, or computationally, from the

---

[2] *μuSE* is pronounced "muse". The special spelling is for trademark purposes. Patents pending.

[3] Work performed at Sandia National Laboratories is supported by the U.S. Department of Energy under contract DE-AC04-94AL85000.

results of computer calculations or models. Investigating and understanding the content of such information is currently a major hurdle between basic research and actual application of the results. The scientist still remains separated from the data, with the computer acting as a recalcitrant intermediary. Ideally, the scientist should be able to view, manipulate, and move through the data as though 'in' that multidimensional world.

There is considerable on-going effort to develop computer software and architectures that can, at least to some degree, mimic the logical operations of the mind (e.g., artificial intelligence and neural networks). We view the problem from a different perspective and are examining the feasibility and effectiveness of a more symbiotic approach to the traditional method of human-computer interaction. We are developing a human-computer interface that capitalizes on the strengths of each and allows information to flow in both directions in a maximally effective manner. This permits both to operate efficiently and in a manner for which they are best suited.

While a computer excels at performing routine types of computations, it is relatively inefficient at subjective problems such as pattern recognition and complex decision making. The mind, on the other-hand, is the most advanced device ever developed for information correlation, pattern recognition, and anomaly detection. All of our senses are superbly designed to collect, correlate, and analyze continuous streams of information, to discard irrelevant information at high speeds, and to make intuitive connections. "Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights. In many fields it is already revolutionizing the way scientists do science." [4]

Although visualization is now an accepted and important tool of scientific investigation, most current approaches depend strongly on past methodology (graph paper representations and application specific tools). Such approaches present substantial difficulties to the researcher who simply wants to 'see' the information:

(a) Applications are difficult and time-consuming to program;
(b) The operation of each program is unique and usually highly specific;
(c) Standard techniques utilize only a few interactive features (e.g. flat screen, keyboard, and mouse);
(d) The information presentation is almost exclusively visual and easily available displays are limited;
(e) Dynamic interactions with data that require new database computations (e.g., sorting, transformations, correlations) are often not supported;
(f) Real-time capabilities are severely limited;
(g) The addition of any new capabilities is usually difficult and requires detailed knowledge of the system.

Advances in microprocessor performance, specialized graphics-rendering VLSI components, sound synthesis, speech recognition, synthetic voice production, stereo-optics, spatial position-trackers, and display technologies have opened up new possibilities for interactive presentation of complex scientific and engineering data. Virtual systems can permit the scientists and engineers to enter into, and interact with, an artificial environment that consists of their data. Such systems permit our natural senses to be more fully utilized in exploring and understanding the information.

The goal of this project is to develop an open, multipurpose software interface between general classes of scientific information and a highly interactive, multidimensional visualization system (including the incorporation of immersive systems often referred to as virtual reality systems). In this manner we hope to demonstrate the feasibility of decoupling the often difficult task of developing programs (which map abstract data and information into a multidimensional representation) from the details of dealing with the complexities of a user interface and the idiosyncrasies of hardware devices. Ideally, the interface should permit the user to interactively describe the structure of the data, together with certain visualization characteristics, and then create an interactive, multidimensional display environment. From within the environment, users could then interactively view, manipulate and transform basic data to facilitate their analysis and enhance their understanding of the information it represents. Multidimensional (i.e., 3-D or greater) environments, that may be interactively entered by a user, represent the next generation of visualization systems.

3. Design Objectives

The goal of the *μuSE* Project is to provide an open software system that simplifies the development of interactive software, provides a device independent framework for presenting computer-based information in a variety of modalities (e.g., visual, auditory, kinesthetic), and creates a user-controlled environment that can enhance the speed of human perception of computer-based information by more than one order of magnitude. Design criteria for the system are:

A) *Device Independence*: A *μuSE* environment permits the easy incorporation of new interactive device technology to enhance human-computer communication. The hardware interface environment is based on patterns of data transmission, not specific device functions. Examples of interactive devices include flat-screen display technology, stereo viewing devices, VR displays, sound, speech recognition, voice synthesis, and various mechanical devices (e.g., mouse, joystick, steering wheel, 6D pointing devices). A

---

[4] McCormick, DeFanti, and Brown, "Visualization in Scientific Computing", Report to the National Science Foundation by the panel on Graphics, Image Processing, and Workstations, July, 1987.

minimal *μuSE* system involves a processor, flat-screen display, keyboard and mouse. This approach permits systems to grow and adapt to new technological advances with minimal impact on existing hardware and software environments and to still maintain continuity and consistency of operation.

B) *Software Simplification*: Functional aspects of the human-computer interface can be separated from both the details of application development and the specifics of hardware devices. This allows software development of models, data, or information spaces to proceed independently of the rendering platform, display devices, or any specifics of available interactive devices. This approach enormously simplifies the task of model construction or data-base preparation and provides users with dynamic flexibility to use different visual, auditory or interactive techniques as desired.

C) *Real-Time Operation*: In order to achieve optimal real-time performance, the system design uses a 3-layer, asynchronous shared-memory model for communication and control operations. This separates polling and external device control from the application and rendering operations. It also permits the direct acquisition of data (e.g., external sensors) or information (e.g., other computers) in a manner that is both general and highly efficient.

D) *Shared Heterogeneous Environments*: The software structure of the environment is designed so that it may be shared and/or networked to permit multiple users to examine information interactively and simultaneously. Device independence permits users with different hardware platforms and interactive devices to simultaneously share environments (e.g., flat screen systems can operate with head-mounted displays or wall projection systems).

E) *Mobile Environments*: *μuSE* provides a mobile environment (e.g., extended desk top) which travels with users as they explore and interact with information. The environment consists of an artificial craft with a wide variety of display, motion, and interactive capabilities. The transparent walls of craft can support various auxiliary information displays, which may be dynamically altered by the user.

F) *Parallel Processing:* Since real-time interaction and visualization are computationally demanding processes, the system is designed to support and exploit multiprocessing capability, if available.

G) *Foreground-Background Operation:* In some situations, users will need to be able to interact directly with applications running on other computers (e.g., massively parallel systems) and to dynamically alter simulation parameters from within the synthetic environment. The design of *μuSE* supports the generic transmission of control information to remote computers and the reception (perhaps in stages) of updated information. (This capability clearly also requires that the remote computational system have the generic network capability to receive remote user instructions and to pass them to the appropriate application program.)

4. Overview of Software Design

In 1991, Sandia National Laboratories began a project to develop a new approach for the interactive multidimensional representation of complex information. The project, named *μuSE* (Multidimensional User-oriented Synthetic Environment), involved designing a modular, device-independent system that would take users' data or models and provide a highly interactive environment to examine, explore and manipulate the information. As a result of this effort, an information-rich, immersive environment has been created which permits users greater flexibility to explore complex multidimensional information. The system was implemented in an open and modular fashion, which permits new software capabilities and hardware devices to be added without impacting existing software models.

An important aspect of the design is to separate the application software, user interaction, and hardware devices. This permits additions and modifications in one area to be carried out independently of potential changes to the other areas. Further, the system is then extensible and customizable for specific applications, while still maintaining model portability and providing a consistent 'look and feel' mode of interaction. Figure 1 provides a conceptual overview illustrating the information flow in a *μuSE* environment.

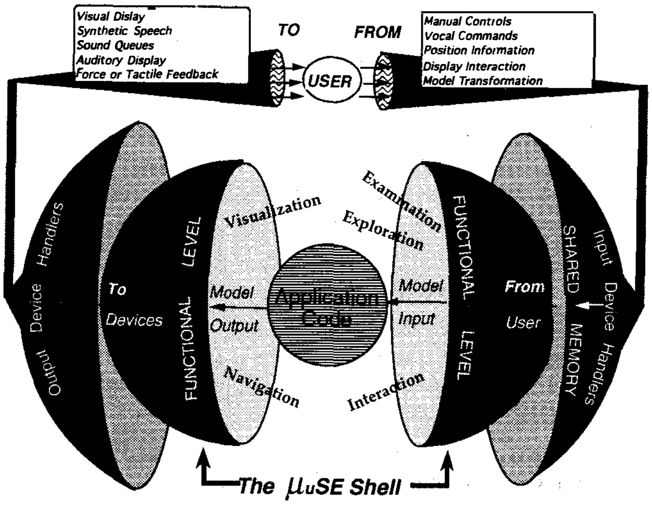

Figure 1. Conceptual flow of information in a *μuSE* environment.

4.1 HARDWARE INTERFACE

Hardware idiosyncrasies are separated by utilizing a three-level, shared-memory approach:

1) The first level that receives raw, memory-mapped information from physical devices;

2) A device level in which pointers provide the location of the latest information on each active physical device; and
3) A functional level that translates external user actions into generic functional operations which are either acted on by the system or sent to the application program.

The interactive control system was developed to support real-time operation. It can run on a separate processor and interfaces with user programs through physically shared memory. Access to this memory is provided by library calls.

All physical devices in a system communicate with a shared memory region -- not user code. Thus the programmer's model does not deal with the real-time interactive environment nor concern itself with the specific nature of physical devices. Polling of devices is carried out by specific driver tasks (or physically separate processors). A shared memory daemon manages the memory arbitration. The memory is structured to permit the incorporation of new devices. This structure decouples the handling of real-time human interaction from details of the model or data representations, and permits high-performance (low-overhead) device response. While program-device interaction can occur through this level, it typically is channeled through the two higher levels. The middle level is the device level. Here library routines may access pointers to shared memory regions to obtain current information on specific devices. Such routines may scale or normalize specific state information to provide it in a consistent framework to the user.

4.2 FUNCTIONAL OPERATION

Perhaps one of the most important aspects of the hardware interface is the third, or functional layer, which separates the dynamic user interaction from the hardware (e.g., how the command is sent) and the model (what do you do with the request). Here interaction with the user and devices is through functional calls. As an example, after drawing a scene, an application may want to know whether the user has changed the viewing position or direction. Rather than polling specific devices to determine this, the application simply calls a position function, which provides the information on the next point of view. The method of obtaining this information (e.g., keyboard input, mouse position, joystick, user head movement, etc.) is now immaterial to the application. The user, on the other hand, is completely free to use whatever capability is available on the system and to reassign functionality at will.

5. The Human Interface Model

User interaction with synthetic, immersive environments presents an exciting and relatively uncharted area. In concert with the idea of a humanistically organized software environment, discussed previously, five areas of human-computer interaction have been defined and investigated. They are *Exploration*, *Navigation*, *Presentation*, *Interaction*, and *Examination*. These five areas serve to define a functional arena for interactive environments and provide the foundation for device and model independent tools. This section examines the implementation and operation of specific capabilities in each of these generic areas.

5.1 EXPLORATION

This domain involves techniques for moving within and between various models and representations of information. To facilitate this modality, *μuSE* creates a virtual craft. This craft can be characterized as a private, extended office that travels with the viewer through the environment. The movement and position of the craft are maintained independently and separately from that of the user. The visual representation of the craft provides the user with a personal frame-of-reference within the information space. The craft functionality provides the means of transportation to explore spatial, temporal, and other dimensions within the information space. To accomplish this functionality, the system must constantly maintain and track three separate spatial coordinate systems and their changing relationship to each other: 1) world coordinates (for the model and spatial representations); 2) craft coordinates; and 3) user coordinates.

Within the spatial domain of the model, the craft provides the viewer with a variety of transportation capabilities. The default mode of movement is the *hovercraft*, which is always oriented parallel to the horizontal plane of the model. It turns left and right, moves forward and reverse, and travels up and down like an elevator. **{{movie 1}}** Another mode of spatial transport, which is not restricted to simple horizontal or vertical motion, is a *plane* **{{movie 2}}**. The *tethercraft.* provides a model-base mode of exploration. The user may attach or tether to any object in the system that has been defined by the application code. Mathematically the tether operation attaches the craft's coordinate system to the (0,0,0) coordinate of the selected object. The craft's motion now becomes centered on the object. Moving forward or back are translated as motion toward or away from the object. Left and right motion becomes equatorial rotation around the object, and up and down motion translate into polar rotation. **{{movie 3}}**. Further, if the zero coordinate of the object is in motion, the craft will be in motion with it. All virtual crafts operate within any model and are independent of the application code. The mode of movement may be dynamically selected by the user, and the specific physical devices controlling the motion may be chosen according to user preference.

Humans operate in a temporal as well as a spatial universe. *μuSE* also recognizes time as a potentially independent dimension of exploration. In data where time is a parameter, the system permits the user to control both the speed **{{movie 4}}** and direction **{{movie 5}}** of time flow. In some instances *μuSE* may permit pseudo-time to be added to an otherwise static model (e.g., the time-dependent motion of a *μuSE* tool to a static data set). It should also be noted that the system deals with three independent temporal coordinates: 1) the timebase associated with the application or simulation;

2) the flow of time inside the craft or related to craft interactions; and 3) real-time from the user's perspective.

Another means of exploration is by *teleportation.* Instantaneous teleportation between marked spatial, temporal and N-dimensional locations is provided. Locations may be statically defined by the application program (e.g. for reference) **{{movie 6}}** or dynamically defined by the user (see the following section for additional information on teleportation). The starting time and location of the user in the model are stored automatically as a teleportation point. Users may, therefore, always request to be transported to the initial spatial or temporal locations.

5.2 NAVIGATION

These are methods of assisting a user in maintaining both location and reference in N-dimensional space (e.g., to keep from becoming lost). One such tool is the navigational map that is defaulted to the side wall of the craft. This 'you-are-here' map automatically shows the position of every defined object in the space as well as the location and orientation of the craft. Since this map is maintained in real-time, it captures the location and motion of both object and craft positions. **{{movie 7}}**

Another navigation technique involves the use of *N*-dimensional markers. The system permits the user to dynamically 'mark' the craft's location at any time (e.g., by saying "mark location 3"). These markers are stored as *N*-dimensional reference points. Users may use any portion of a marker's descriptor as a reference information for teleportation. For example, a user could request to be teleported to 'time 4'. The system would then set the application time to correspond to the temporal dimension of marker 4, but would not alter the craft's spatial location. Similarly, teleporting to 'space 4' would position the craft at the 3-space location defined by the marker, but at the current time. (One can also teleport to groups simultaneously, e.g., time and space.) **{{movie 8}}** Markers may be may also be displayed and attached to as system objects.

Portals function as doorways and provide a means of moving between different information spaces. Active portals are typically displayed as rotating cubes in the very top area of each connected information space. For identification purposes, cubes typically display a representational picture of the information within (if it is available) on each face of the cube. In addition, the craft computer will identify the portal at which the craft is pointing. **{{movie 9}}** Tethering to a portal has the effect to moving the user into that information space.

5.3 PRESENTATION

This refers to different ways of representing complex, multidimensional information. It includes, for example, such techniques as multiple 3-D or 4-D projections, traveling 'sticky' displays, N-dimensional radar, N-dimensional strip charts, digitized pictorial information, multimedia-VR synthesis, auditory representations, data fusion, etc. Standard 3-D graphical techniques often utilize color overlays to add to the dimensionality of the representation. Since synthetic environments may be as large as desired, users can easily increase the information density in virtual space by populating it with additional displays situated at different locations. **{{movie 10}}**

Within the environment of a virtual craft, the user may also use cockpit displays. These allow the virtual transparent bubble around the craft to be used for opaque and translucent displays of information. Typically the left-hand or right-hand walls of the craft are used to display this information so that forward viewing is not obscured or made unnecessarily 'busy' (in a visual sense). These displays can be dynamically selected by the user and travel with the craft. Cockpit displays permit the user to expand the dimensionality of the space by requesting the display of additional or ancillary information. Such displays have several interesting properties. Since they are part of the craft, they travel with the user while exploring the space. Further, since they are always located in close physical proximity to the user, they can be viewed in various perspectives and/or detail by simply moving the head or body (e.g., leaning closer). Examples of such applications include the display of 2-D graphical data (e.g., amplitude vs. time) **{{movie 11}}** or cross-sectional projections from solid models **{{movie 12}}.** Cockpit displays can also consist of schematics, figures, photographs or even video clips.

Visual representations of data are only one way to map information on to the human senses. *μuSE* also has the capability of mapping information into the audio spectrum. The simplest example of this is the auditory representation of the movement of the craft itself. Here velocity and acceleration parameters associated with the craft motion are given a 'voice' (e.g., sound signature). These model parameters are then mapped to auditory variables associated with the selected voice. As the model parameters vary, the associated auditory parameters are similarly scaled in real-time by the system. The resulting auditory representation varies with the data, and the user 'hears' the virtual engine of the craft. Such a mapping delivers valuable sub-cognitive information to brain without cognitive distractions (i.e., the user is always aware, in a relative manner, of the speed of motion, the rate of acceleration, and where in the range of these parameters one is operating). In an analogous fashion, the relative speed of time in a model is provided a voice (either ticking, for forward time, or tocking for backward time), which is then rate-scaled in proportional to the current speed of time.

Applications can also utilize the audio mapping capability of the system. If the application code provides the *μuSE* shell with both pointers to and descriptions of data formats (e.g., variables, arrays), the user can elect to have selected information translated to an auditory representation. **{{movie 13}}** Developing algorithms for the translation of quantitative information into sound (in a humanistically meaningful matter) is a relatively new area of investigation. Some of the audio techniques used by *μuSE* have proven very successful. New approaches are being investigated (e.g., the differential and integral audio representations of 2-D information), and the research in this area is continuing.

5.4 INTERACTION

This refers to command and control structures which permit users to communicate, and directly interact with, both the system and application program (e.g., to select options and interact with the environment). Manipulative techniques permit users to interact with the information or model base (e.g., to change and move objects or displays, or to select particular objects or information). The effectiveness of interaction techniques, from a human perspective, depends on the devices available, the functions to be performed, and what other types of information are competing for attention. Traditional interactive devices usually consist of a keyboard and a mouse. The keyboard requires considerable training to use efficiently and was developed to function in a 2-dimensional world. The mouse was a considerable improvement -- it requires much less training, and is easier to use. Unfortunately it is still a 2-D device.

Popular acceptance of the mouse led, in turn, to the use of pop-down windows as a command mechanism. This technique is versatile, easy to use, and doesn't require much screen space. From the perspective of efficient human interaction, however, this approach presents difficulties in a 2-D domain and raises serious problems in the 3-D arena. The operation of locating and reading pop-down buttons physiologically competes with our concentration on, and perception of, spatial images. To illustrate the point, imagine driving a computerized car where all console controls are contained in a virtual pop-down window that scrolls down from the rearview mirror (at the touch of a button on the steering wheel).

To permit parallel control operations and minimize distracting elements, a humanistically efficient interactive environment should permit the mapping of operations to different areas of the human senses. This permits sub-cognitive input to be maximized and sensory conflict to be minimized. For example, control operations involving precision settings and movement are very effectively controlled by the hand (via steering wheels, joysticks, sliders, rotating pots, etc.). Such devices can be operated without visual involvement and, in addition, can convey valuable sub-cognitive information through tactile feel and response, as game designers and auto manufacturers have long realized.

Reading operations, as indicated previously, can create conflicts with visual concentration. Alternatively, synthetic speech generation can be used to transform most, if not all, written or textual information into conversational speech. The auditory and visual channels operate quite harmoniously in parallel, and listening to status or information doesn't normally create visual conflicts. Analogously, the act of requesting status or controlling state functions (e.g., turning things on and off) is often more efficiently performed by voice command rather than by locating buttons (either physically or virtually). Our relatively intuitive conclusions are that two-way dialogue with the computer provides an extremely powerful and normally non-conflicting mechanism for issuing commands and obtaining information.

In a typical flat-screen mode of operation, it is frequently necessary to select which window or screen you currently wish to view. Such 'picking' operations consume time, are usually a result of limited visual space, and are often distracting. As discussed in the sub-section on Presentation, the craft is surrounded by a transparent bubble onto which the user may map information. In this environment there is far more 'space' available to spread out information. Because the head position and orientation are continuously tracked, users normally select what information or display to look at (and in what detail) by simply turning their head and/or moving their body, as they do in the real world.

There are many software features in *μuSE* to facilitate user interaction with the environment. A craft-based variable intensity light permits users to obtain effective lighting anywhere they are in the information space. This lighting capability is very important in that it alters shadows. Shadows are extremely important to the mind's ability to resolve edges and in constructing accurate mental representations of the spatial environment. Magnification provides the ability to peer through a virtual telescope at distant information. Movable cutting planes permit 3-D representations to be sectioned and the interior examined. **{{movie 14}}**

Tethering, as discussed earlier, allows viewers to dynamically attach themselves to any object in the model. If the object is in motion, the user is in motion with it. This is an example of an interactive mechanism involving the craft and an object within the model. An extension of this capability provides an even more direct mechanism for interactive manipulation. While tethered, the command "control object" permits the user to alter both the spatial position and orientation of the object. (Note that a defined object may be anything from the component in a CAD assembly to a time-dependent data set.) **{{movie 15}}** Displaced objects may be reassembled by command and individual components selectively displayed or removed. **{{movie 16}}**

New interactive tools and devices are being tested and evaluated on a continuing basis.

5.5 EXAMINATION

This domain includes capabilities that permit the transformation, correlation, and overlay of information. It typically is the most cognitive of the five functional areas and plays an important role in the complex area of human perception and understanding. Examination frequently involves asking questions relating to an information base. One way of posing such questions is by means of the virtual, on-board computer. A simple example of this is when the user asks what the craft is pointing at. In response to this query, the system projects vectors from the front of the craft, evaluates all objects in the vicinity of the vector, selects the most probable object of reference, and supplies the requested information to the user. **{{movie 17}}**

The results of a user inquiry may require the sorting or correlation of information in accordance with specified criteria (e.g., show the distribution of parameter x where parameter z is within specified boundaries). This may effectively require a data-base type of search operation. Our research includes imbedding types of data-base operations within the *μuSE* shell for selected data formats (e.g., defined matrices, arrays or standard structures).

The ability to fuse and correlate information from disparate sources is another example of examination. Data fusion is usually a difficult and complex task. We feel that it can best be carried out in an environment that permits close cooperation between the user, the application code, and the available hardware devices. One example, the discriminator chip, involved fusing approximately 60 dimensions of information, obtained from 6 different programs, running on 3 different computer systems (from the Cray to a workstation). The outputs of each of these simulations were completely different (ranging from CAD models and finite element meshes, to signal variations in circuit simulations). The outcome of this example effectively demonstrated the importance of the being able to access and correlate different types of information. A problem in the chip operation was quickly located, the source of the problem easily determined, and corrections tried and verified. Assuming the availability of prior simulation test results, the identification, diagnosis, and correction of this problem can usually be carried out in the *μuSE* model in under 10 minutes. Further, the diagnostic process becomes quite intuitive and relatively easy to follow and understand. **{{movie 18}}**

Mathematically transforming data is another way that humans use to examine information. In a simplistic example, the *μuSE* command "display objects" causes a selective scaling to be applied to all defined objects. Objects whose perceptual size is very small, from the user's current point of view, are thus made visible (note that this does not solve the problems of one object occluding another). Similarly model size control permits the size of the entire model, relative to the size of the user and craft, to be varied. This means the viewer can become either very large or small with respect to the environment, thus permitting either overviews or minutely detailed examinations.

Scientists and engineers have long utilized mathematical transformations to compress and display data, particularly when it varies over large dynamic ranges (e.g., semi-log, log or polar plots). This capability also exists in the *μuSE* environment, but has been conceptually and mathematically extended to encompass more types of information. For example, while logarithmic transforms are defined for numerical data, they have no clear definition in an object based space. Applying a logarithmic transformation to an object would result in severe distortion in the shape of the object. In a dynamic, multi-object environment, particularly when objects are moving with respect to different centers, the perceptual distortion is enormous. To overcome this problem and still take advantage of the scaling properties of various mathematical transformations, a cascading approach has been developed which operates only on the size, not the shape, of objects and reinitializes the origin of the transformation each time a new coordinate system is invoked. The resulting transformation alters sizes and distances, but does not effect shapes, angular information, or time. **{{movie 19}}** Note that a user can explore and interact with a transformed space, using *μuSE* capabilities, just one would in a normal representation. **{{movie 20}}**

6. Shared Environments

Permitting multiple people to both independently and simultaneously explore the same synthetic environment is a unique aspect of immersive technology. The shared memory architecture of *μuSE* is designed to enable and extend this capability. The architecture is based on a distributed, asynchronous communication model. The design attempts to minimize the network traffic required to maintain synchronization and supports heterogeneous operation at each site. The system not only supports multiple users, but permits each to maintain a personalized or private 'cockpit' within the shared space. The cockpits' information displays are individually configured and not available for access by others in the environment. From within the cockpit, users may access and display both shared general data as well as information that is only available locally. Although the existence of this proprietary information is not known to other users, it may, if desired by the owner, be dynamically shared (broadcast) with others. The system also permits 'teleportation' between users, actually permitting one user to 'see' the shared environment through another user's eyes.

Work on shared *μuSE* environments is currently in progress and operational systems are expected later this year.

7. Conclusions

Advances in microprocessor technology and multiprocessor systems have greatly increased our computational capability to simulate multivariate systems and analyze complex N-dimensional information. In many cases, however, understanding the results of the computation or analysis may require significantly more human time than computer time. Immersive environments, such as virtual reality, coupled with a flexible and highly interactive user interface, have the potential for greatly enhancing the information transfer between computers and ourselves, thereby greatly enhancing our ability to understand its meaning.

Future applications of computing will critically depend on the development of more effective human-computer interfaces which are designed to significantly improve a person's ability to explore, interact with, perceive, and understand computer-based information. The purpose of this work is therefore to: 1) develop a highly interactive, device-independent immersive environment as a stable test bed for applying the technology to information perception; 2) develop an open hardware and software environment that is extensible and will simplify the exploration of complex information space; and 3) implement new techniques to allow users to visualize and interact with information in a more rapid and intuitive manner.

A *μuSE* system has been developed and tested over the past three years. It has achieved and demonstrated a very high level of hardware device independence which has led to the easy incorporation of new technology as it became available. The humanistic basis of the software design has proven enormously effective, both in enabling people to interact with computer-based information and in speeding up the analysis and decision making process. The functionality-based system architecture has simplified application development and permitted complex data, models, and simulations to be rapidly incorporated within a *μuSE* shell (typically within a few days). The system has been very successfully applied to a wide variety of applications (e.g., CAD models, explosive welding simulations, medical data, seismic information, finite element analysis, circuit simulation, kinematic evaluation of mechanical operation, geographical data), and has been wrapped successfully around complete simulation programs. It has accepted data input in forms ranging from N-dimensional matrices, to CAD models, to surface or terrain information, to photographs, to video, to sound. In conclusion, one of the greatest surprises and delights in applying *μuSE* to different applications has been the frequency with which it has allowed people to discover important aspects of their own applications of which they were previously unaware.

ACKNOWLEDGMENTS

The authors wish to specifically thank Ken Gant, Patricia Crossno, David Gardener, Randall Smith, Rob Bugos, Johnny Biffle, Carl Digert and the many others at Sandia that have worked on the system and models, and have generally helped and supported this effort.

BIOGRAPHY

CREVE MAPLES **{{photo 1}}** obtained his B.S. in Chemistry from MIT and his Ph.D. in Nuclear Science from the University of California, Berkeley, in 1970. He is the author of over 100 articles in the field of computing and is recipient of awards in parallel computing, real-time systems, and computer architecture and scientific visualization.

He was previously Chief Scientist and director of software development at VITESSE Electronics Corp.; president of CARD Associates (Computer Application Research and Development); and head of the Advanced Computer Architecture Laboratory at the University of California's Lawrence Berkeley Laboratory.

Currently he is a Senior Scientist at Sandia National Laboratories in Albuquerque, where he heads the Synthetic Environment Laboratory. He is the principal investigator of the MERLIN (Memory Routed, Logical Interconnection Network) Project, an ultra high-speed, minimum latency, heterogeneous communication system, and the developer of the *μuSE* (Multidimensional User-oriented Synthetic Environment) Project.

Contact Information:
(505) 844-2047
maples@cs.sandia.gov
FAX (505) 844-6482