THE SOFTWARE PROJECTS

William Bricken
July 1993

Here's a brief description of the suite of software projects that I put together and managed at UW. These projects utilized about 40 graduate students over three years. My role was as Principal Scientist and Architect. Projects for which I served as the lead designer and for which I contributed substantively to code development are marked by an asterisk in the Contents and by "(william)" in the body.

CONTENTS

WORLDS AND APPLICATIONS

Education Project *
Pacific Science Center Project
Architecture: Spatial Perceptions in VR
Manufacturing in VR
Metro Project
Embedded Narrative *
Spatial Awareness *
Virtual Holographic Workstation
Virtual Lunar Lander
Boundary Reality *
Experiential Mathematics and Block Logic *
Mathematical Foundations of Cyberspace *
Parallel Computation of Numerical and Symbolic Systems *
Magic as a VR metaphor *

SYSTEM ARCHITECTURE

FERN/VEOS (william)

VEOS is an extendible environment for prototyping distributed applications for UNIX. The VEOS application programmer's interface is provided by XLISP 2.1 (by David Betz). XLISP provides familiar program control; VEOS provides inter-process message passing and content addressable database access.

VEOS (The Virtual Environment Operating Shell) was developed for distributed Virtual Reality applications and has been in use for two years. We have already built VEOS tools for real-time graphics, sound, position sensing and voice synthesis and input. However, VEOS is by no means limited to these types of applications.

VEOS is ideal for applications where hardware resources are not physically proximal or where machine-dependent resources (e.g. software packages, interface devices, etc..) are isolated because of their platform. VEOS is also ideal for prototyping programs that employ coarse grain parallelism. That is, VEOS uses heavyweight sequential processes, corresponding roughly to UNIX processes. In this way, VEOS can be used to utilize a network of workstations as a virtual multiprocessor.

C programmers can build custom VEOS tools that are accessible from XLISP and thus are immediately compatible with other VEOS tools. Lisp programmers can quickly design and run distributed programs that utilize diverse hardware and software resources through these custom tools.

VEOS is not an operating system. VEOS is a user-level framework for prototyping distributed applications. Its primary focus is flexibility and ease of use. This design comes somewhat at the expense of real-time performance. This is not to say, however, that VEOS cannot achieve good performance with proper application structuring and tuning.

Relying on only the most common UNIX interface, VEOS is platform independent. VEOS 2.0 has been extensively tested on platforms such as DEC 5000, Sun 4, and Silicon Graphics VGX and Indigo.

Mercury, the Virtual Body

Creating a virtual world requires three major components: the database containing information about the objects (or entities) in the world, the system to support the database, and an interface for the user to interact with the database. Currently, VEOS is the system, FERN II is the database, and Mercury is the interface. Mercury monitors the sensors that track the user's physical state, sends this information to the database, and displays the data in the database appropriately.

Previous virtual reality systems have assigned all or most of these tasks to a single computer. In doing so, the interface objects such as displays and sensors are reduced to running at the speed of the database, and the database is itself limited by the interface processes running on the same CPU. Mercury lets each system do what it does best, enabling the interface to run as fast as possible while communicating with the database as fast as it can keep up.

This approach provides several major advantages. Mercury can be compiled and run to work with any display system on any processor with the proper communications capabilities. Since the protocol for communicating with Mercury is standard, a flavor of Mercury that takes advantage of its specific hardware can be used with any database that speaks the standard Mercury protocol. Mercury takes the hardware and software tasks that are common to all virtual worlds and packages them neatly for use in any virtual world running on any system.

Currently, Mercury only takes advantage of two position trackers and a set of buttons as input devices, but future versions could be created to take advantage of not only additional position trackers, but other inputs like the Spaceball or the BIOMUSE. In addition, the way each sensor is mapped onto the virtual body could be chosen from world to world, so that with the same version of Mercury one could choose not only which of the available input devices to use, but what each of those inputs will mean in the virtual world.

AnteChamber

This project creates a continually-running multi-participant virtual world to serve as a gateway to other worlds. The Antechamber world also serves as a central meeting place, hopefully providing a social center and a venue for exploring social issues of multi-participant virtual interaction. A design objective is to allow all VR worlds to be loaded and entered from within the antechamber, by an arbitrary number of participants, from a number of different hardware platforms.

Imager

The Imager is a fast graphical rendering system which provides consistent, hardware-independent imaging for the Mercury Participant System and for other applications. Its task is to take graphics-related attributes from a virtual environment and present them to the participant system's graphics hardware, taking advantage of any capabilities present in the hardware for speed or realism. The Imager handles the details of stereo rendering for both head-mounted, fully inclusive and liquid-crystal shutter displays, and can also render into an on-screen window when stereo is either not desired or not available. As an entirely in-house system, the Imager also serves as a

research tool for high speed rendering, through which we can explore issues, optimizations, and improvements in real-time and near-real-time rendering.

The Imager works with our SGI (VGX) and Sun (GT) graphics systems, and also runs under X Windows using a GL-compatibility library called VOGL. Current work is expanding this list to include Kubota Pacific's Denali graphics engine and the SGI RealityEngine. For stereo output, the Imager works with StereoGraphics' CrystalEyes shutterglasses and most dual NTSC head mounted displays such as the Virtual Research Flight Helmet.

Sound Renderer

The Sound Renderer is the auditory counterpart to the Imager. It provides spatially localized (3D) sound to virtual environments. Currently, the Sound Renderer employs two Crystal River Convolvotrons, yielding eight individually spatialized sound sources, each with 256 point head-related transfer functions (HRTFs) and controllable attributes such as intensity, pitch, duration, roll-off, and Doppler effects. Ongoing development includes an option which would simulate simple room acoustics (rectangular prism).

Spatial Sound Server

The goal of this project is to implement high quality 3D sound localization using conventional workstations in a networked environment. The computations may be done locally or on a remote server. One application of this technology is teleconferencing, where each participant's voice is localized to a unique "seat". A tool for easy access to conference participation will also be developed.

MIDI

The MIDI interface to VEOS allows virtual environments to behave as MIDI devices. By both sending and accepting MIDI data, the VEOS MIDI interface permits use of MIDI instruments such as keyboards to control virtual entities, and to manifest virtual actions in the real world in the form of music.

OverLays

The real world is interesting, useful, and in need of our attention. We are integrating the real with the virtual, and importing it into computational environments. We have assembled a system which augments stereoscopic live video with interactive stereoscopic computer generated imagery. The combined imagery, with real world video as background, and CGI as foreground, has advantages over environments made of purely synthetic imagery.

TOOLS

Wand Tools (william)

The Wand is an interface tool that uses a simple physical device for a wide range of functions. The physical device is a rod with a 6 degree-of-freedom sensor on one end which supplies position and orientation information to the model. The sensor information inhabits a virtual rod held by a virtual hand. We assign functionality to the Wand by attaching a voice sensor to it and inserting rules into its set of dispositions. The general form of the response rule is "if recognize-input then generate-associated-output". Some functions of the Wand include:

Ray on/off:
A ray emanates from the end of the virtual rod, collinear with it.

Identify:
The first object which the ray penetrates returns its name.

Connect:
Construct a communications port between the rod and the identified object.

Jack:
Teleport the viewpoint (the ray vector) of the rod to the identified point on the object.

Grasp:
Attach the end of the ray to the identified object. When the Wand is moved, the object stays attached. When the Wand is rotated, the object rotates.

Normal:
Rotate the identified object so that the intersecting ray is normal to the object's surface.

Sight:
Jack into the Wand; the viewpoint of the patron issuing the command is linked to the ray vector.

Move faster/slower:
Move the viewpoint of the patron along the ray vector.

Space Entity (william)

The purpose of the Space Entity Project is to provide first class modeling and dynamics tools for environments. Environments consist of a space and

objects within that space. Characteristics of every object within a space can be abstracted to be characteristics of the space itself. For example, gravity can be attributed to space when it acts on every object within it. Other properties of space include metric structure (grids, orderings, sets, reals), gradients (gravity, wind, electromagnetic forces), continuity, grain-size and coordinate systems.

Deductive Entities (william)

A major design goal of VR systems is to integrate spatial and symbolic computation. Entities can be reactive, just responding to the circumstances of the environment, or they can be responsive, incorporating some of their own computational reflection to a behavior decision. We are incorporating AI-based tools for rulebases and for inference within entities in a virtual environment. The deductive engine is small, very efficient, and local.

Universal Motivator (william)

The universal motivator is a graphical VE development tool that allows programming of virtual entity attributes by drawing Entity Relativity Graphs (ERGs) that specify the relationships between attributes. For example, one can build a simple color control by relating the color of one object to the coordinates of another.

Entity Editor (william)

Building virtual worlds has focused to date on static models. We are extending classical CAD techniques to dynamics. The Entity Editor permits a designer to specify both form and behavior. This project integrates several other tool-based projects, providing an interface for the designer.
Generalized Sweep Construction

God's Eye View

Also known as exocentric viewpoints, this project is a small version of a virtual environment that shows all the action and entities in that VE. It can be used to determine a participant's location, get an overview of the space, and change entities - including manipulating things about the participant, such as location. A collection of viewpoints would serve as a gateway world to different VE's.

Dynamics Toolkit (william)

The goal of this project is to develop a software library of dynamical functions that one can easily incorporate into the software for the objects present in a virtual world. These dynamical functions will enable the simulation of the possible motions of a particular object (or system of objects), with the type of motion determined interactively by the user at run-time. The implementation of extremely efficient numerical algorithms permits the simulation of complex systems (systems with large numbers of degrees of freedom) in real time.

Recursive Construction (william)

Complex virtual environments require powerful construction techniques. The RC project is developing a specification and construction capability based on generalized Lindermeyer systems (recursive term-rewriting over graphics specifications). RC permits top-down design of environments, providing control of the level of detail and refinement in a graphical object. Models well suited for this technique have many repetitions of small groups of basic elements, such as trees in a forest, buildings in a city, or clouds in the sky. RC provides fine grained control over the coherence and juxtaposition of elements and the probabilistic ranges of their form using several techniques. Both the geometry and the topology of forms can be specified by equations in an algebra of form, by production rules, by form abstraction, or by direct model modifications. RC provides rapid modeling capabilities which bridges the gap between photographic images and solid modeling with polygons. Project objectives include L-systems and fractal modeling, an algebra of spatial form, form abstraction and decomposition, and an interactive graphics composition system.

Voice Recognition/Synthesis

This method of isolated word speech recognition combines Hidden Markov models (HMM), Multi-Section Vector Quantization (MSVQ) Code Book techniques, and Natural Language Understanding.

The use of the first two techniques together enables the system to correct half of the wrong decisions made when only one technique is used.

Natural Language Understanding is used in order to have a dialog between the user and the computer. The grammar of this system is restricted to the description of a small environment. The user interacts with the system through phrases that indicate an action or specify the system status.

This voice recognition system is implemented on a digital signal processor (DSP). The Natural language understanding was developed using XLISP.

Audio Browser

The Audio Browser is a hierarchical sound file navigation and audition tool. The intent is to speed up the laborious process of selecting appropriate audio segments from vast archives of sound files. Sound is naturally linear; we cannot scan a "page" of sound as we would a page of text or images. Textual descriptions of sound hardly describe the content. Conversely, we can process many audio streams simultaneously, while we cannot interpret many images at once.

The Audio Browser takes advantage of the fact that we can naturally monitor many audio streams and selectively focus our attention on any one if they are all spatially separate. Also, using spatial audio sequences, audible transitions from node to node in the database are used to give the listener a feeling that they are "moving" through the tree of nodes, giving location.

In the current implementation, the sound files are arranged in a hierarchy. Representative sounds from each node below the current node are displayed around the listener. The listener navigates through the sound file hierarchies by choosing the sound representing the node they wish to follow.

Auditory Shape Perception in VR

An exploratory project using spatialized sounds produced by the Convolvotron to convey physical shapes through auditory information. The objective is to achieve a sense of shape through direct perception, rather than via audio-driven visualization or rasterization.

Inconsistency Maintenance (william)

Virtual worlds with multiple participants must present different perspectives on the same environment. But the virtual environment can also maintain inconsistent views for different participants, to intermix personal realities. In VR, communality can be negotiated rather than assumed. In VR, the color of my shirt can appear to be green to me, but blue to you. So long as we do not talk about or interact with the color of the shirt, how it is rendered to each of us is irrelevant. Using a mathematical technique called the imaginary Boolean value, the Inconsistency Maintenance Project provides tools to maintain inconsistent views and interpretations, and tools to negotiate differences when consistency is desired. Differences can be resolved by presenting each participant with personal views, by providing a meso-space where communalities can be openly negotiated, or by maintain inconsistency through mathematical techniques.

Object Library

A large collection of .dog ("description of graphic") files is being assembled in a central location so that world designers will not have to redesign common objects. Grids, cubes and polyhedron collections, etc., will be available, along with representative natural world objects. A method of browsing and selecting from the library is also under development.

WORLDS AND APPLICATIONS

Education Project (william)

Several education-related projects that address pedagogical issues and strategies for transferring VR techniques into the classroom. These projects demonstrated that it is possible to build worlds in which students can learn, and that it is possible for students to construct worlds for themselves. This work entails:

-- Learning about and drawing from educational applications of VR.

-- Developing a theoretical framework within which to conduct educational activities. Educational and psychological research is beginning to shed light on how students actively construct knowledge from the information they encounter in their environment and how this knowledge guides the actions they perform. This *constructivist* concept of learning stands in contrast to the traditional notion that knowledge simply arises from information that teachers transmit to students, and converges with the experiential and participant-centered nature of VR.

-- Identifying the characteristics of VR that are most likely to contribute to student education at all ages.

-- Establishing criteria for the development and selection of projects to maximize their effectiveness.

Pacific Science Center Project

Local children were given a chance to design a virtual world over the course of a week. The worlds developed were interesting, intensely creative and showed promise for the capacity of VR as an educational, creative and instructive tool.

Architecture: Spatial Perceptions in VR

One of the most often cited potential applications of VR is its use of simulated architectural spaces. This enthusiasm is based on the assumption that our perception of virtual spaces is identical to our perception of real spaces. This project explores the relationship between these two forms of spatial perception.

Simulated spaces are perceived to be considerable smaller than the real spaces, and the "feel" of a space is most closely predicted by the fully inclusive, tracked simulation conditions. Partial results indicate that participants' sense of orientation does not differ significantly between conditions.

Manufacturing in VR

This project demonstrates that VEOS is flexible enough to be integrated with commercial software (AUTOMOD) for virtual interface applications. Integrating commercial software can facilitate the application of virtual interface technology as a realizable and useful tool.

Most participants tested did agree that VR may be a better tool than traditional industrial simulation for the engineers in a physical plant, especially for quality circle activities. The VR platform would help people find ways to solve problems, get ideas across, and visually test these ideas before actually enacting them.

Metro Project

The Metro project was put together for the Seattle area transit authority, to demonstrate some basic capabilities of our VR system. Metro was interested in the possibility of using VR to facilitate planning and community awareness.

The demonstration environment consisted of a simple landscape with a transit train moving around a track. The user could fly around the space or "jack into" the train and feel as though they were inside and riding the train. The user could also control the train's speed and sound an air horn. This demonstration featured three dimensional movement, spatially localized sounds, jacking (changing one's motion reference frame), and object interaction.

Embedded Narrative (william)

Humans use stories to structure their interaction with knowledge. The Embedded Narrative Project has the goal of providing story telling tools (characterization, dramatic tension, plot, voice) within a dynamic, interactive virtual environment. It relies on behavioral entities.

Spatial Awareness (william)

Exploring the human factors and software issues of spatial awareness in virtual environments.

Virtual Holographic Workstation

Little is known about the human factors guidelines for virtual interfaces. To date, most designs are based on hardware restrictions and personal

speculations. Guidelines derived from experiments with human subjects are needed. Using the Sun head-tracked stereographic workstation, a high resolution screen based virtual interface, this research examined performance on a manual tracking task under different interaction and display conditions.

Virtual Lunar Lander

This is a VR incarnation of the classic Lunar Lander arcade game. It uses FERN, VEOS, Mercury, and the virtual body, and includes general implementation of force fields, mathematically defined surfaces, and simple collision detection.

Boundary Reality (william)

This project describes for virtual world design based on boundaries and relativity, called SPAM: Spatial, Perspective And Movement.

Experiential Mathematics and Block Logic (william)

We have been able to demonstrate that mathematics itself (in particular logic, integers, sets and equations) can be expressed concretely, using 3D arrangements of physical things, such as blocks on a table, doors open or shut, rock walls that respond to gravity, the things of everyday life. String-based symbolic representations of mathematical concepts are typographically convenient, but tokens are not at all essential to mathematical expression. VR makes it convenient to express abstract ideas using spatial configurations of familiar objects. One benefit of this approach is that we can build visual programs, set them on a virtual table, and watch them work. We can experience programs as other entities rather than as dumps of text. Bugs manifest as structural anomalies, as visual irregularities. Architectural design has a sensual, experiential semantics. It is but a quirk of typography that we have ignored the experiential semantics of computational languages. More fundamentally, experiential computing unites our spatial and our symbolic cognitive skills, permitting mathematical visualization, analytic gestalt, and whole brain processing.

Mathematical Foundations of Cyberspace (william)

Development of formal axiomatic systems for the description of cyberspace. Includes void-based models, theory of inclusion, the participant as an operator, and boundary mathematical techniques.

Parallel Computation of Numerical and Symbolic Systems (william)

Development of techniques for extremely fast computation of logical, functional, and numerical algorithms based on VR concepts such as spatial mathematical objects, void-based computation, and direct participation.

Magic as a VR metaphor (william)

The task of designing and building virtual environments can be accomplished from within a virtual environment. The Magic Project is an exploration of comprehensible metaphors for the super-natural task of manipulating realities.

The Magic Project employs principles of magic casting, familiar from folklore and common fantasy games, as a manipulation language. For example, we map the law of Similarity (like objects interact) onto pattern matching, the law of Contagion (objects once in contact remain in contact) onto search ordering heuristics, and manipulations such as enhancements and mergings onto parameter changes, property list transferals and inheritance in a type hierarchy. The net result is a powerful metaphor for building and modifying virtual worlds, available to participants as opposed to expert programmers.