Projects in VR

Editors: Michael Macedonia and Lawrence Rosenblum

Cyberspace and Mock Apple Pie

A Vision of the Future of Graphics and Virtual Environments

Michael Capps, Kent Watsen, and Michael Zyda

Naval Postgraduate School

In the processed foods industry, manufacturers delight in releasing recipes that require their product. The maker of Ritz crackers, for example, has a "Mock Apple Pie" recipe (see http://www.nabisco.com) that adorned its boxes for years—a recipe that replaces apples with lemon-juice-soaked Ritz crackers. It uses an awful lot of crackers. Very interesting when you consider that apples are easily available in the targeted markets, regardless of season; that the crackers cost as much or more than the required apples; and that few people would prefer the taste of damp, lemon-soaked crackers to real apples.

The same behavior occurs in the computer industry. The biggest proponents of computer graphics research these days aren't the graphics companies, but companies who have invested heavily in the development of general-purpose hardware, software, and operating systems and need a way to sell their "crackers." The computer technology of five years ago performed quite adequately for most household tasks (word processing and tax preparation), so manufacturers need a way to sell in a saturated market.

Collaborative virtual reality research is driven similarly. The companies making the greatest investment, both through in-house development and funding of external research programs, are those with similar depth of investment in network infrastructure. They need a way to sell their bandwidth.

Are next-generation computer graphics and networked VEs just modern computing's version of mock apple pie?

Diminishing returns

Special-purpose graphics hardware technology has improved steadily, and graphics software has leveraged off of spectacular developments in general-purpose hardware. General frameworks and interfaces for graphics production have undergone the scrutiny of the entire community, resulting in a suite of effective standards.

Striving towards solving the rendering problem has served computer graphics researchers well for years. Likely another 20 years or more of research remains. The speed at which we're drawing pictures (or warping depth images, or computing radiosity solutions) will double, and double again, and then again. Drawing today is a million times faster than it was for Sketchpad in 1963, with moderately improved realism.

But at some point, when the rendering quality and speed have doubled yet again, no one will notice. The CPU industry is rapidly approaching that point: the processing speed of desktop computers has doubled eight times in the last 15 years. Most users who purchased first-generation Pentium II's don't need an upgrade now, even though available speed has doubled since. The latest attraction is the speed of the bus that conveys information to the processor, rather than the processor itself.

> **Theorem 1 (Corollary to Moore's Law):** There exists a point in time after which Moore's Law is still applicable but not relevant.

The logical extrapolation of the current research focus in computer graphics is smooth, realistic, ubiquitous rendering capability. Clearly, there exists a point of diminishing returns for this line of inquiry. Now is precisely the time to think about the next step.

The medium isn't the message

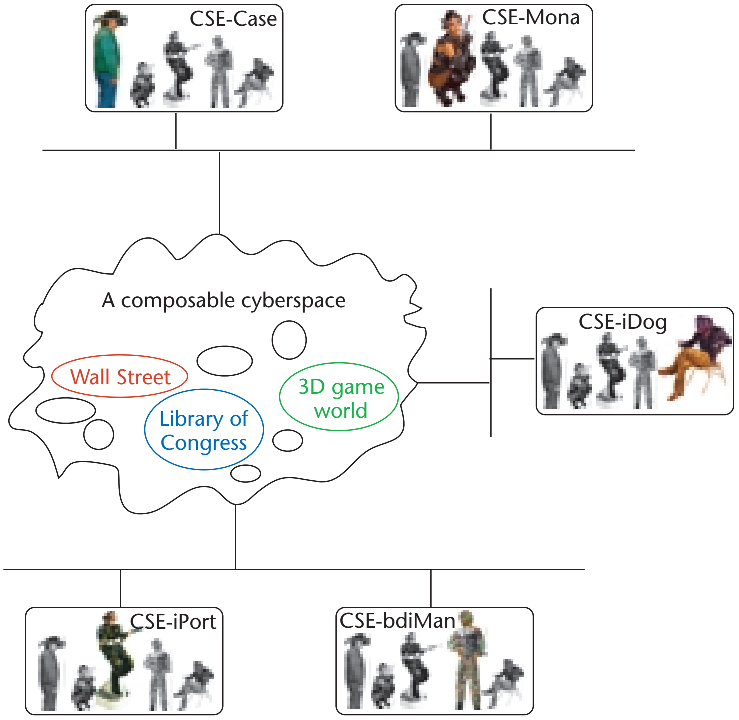
We assume, then, that in the future any user's display platform can render fantastically complex scenes. Having finally shed the concerns related to the computer graphics medium, developers will concentrate on the message. Content will be key—no longer will users accept nonsensical, artistically vacant environments simply because they're presented in a head-mounted display.

> **Theorem 2 (Technical Darwinism):** With cutting-edge technology, someone is bound to get cut.

This will also mean that static worlds, no matter how aesthetically pleasing, will come second to environments offering interactive content. The development and provision of dynamic content lie at the heart of the problem we face. For an environment to attract signficant and regular participation, it must react in an intelligent and unpredictable fashion. Today, that intelligence can come from only two sources: live human collaboration and computer-generated autonomy.

Collaborative VE research combines graphics, networking, human perception, and distributed computing issues. However, these facets betray a disappointing lack of coordination. Most such projects from graphics researchers consider the network a magical transport mechanism, while network scientists treat graphics content as no different than any other. You don't distribute a VE experience by opening sockets and slapping them together, any more than attaching a telephone wire to a microphone makes a conference out of a speech. Myriad issues relate to floor control, session management, awareness, and persistence, and many more besides.[1]

Perhaps most telling to the lack of consensus in shared VR research is in the name. We have CVE, net-CVE, shared VR, and shared VE referring to the general topic of multiuser, distributed virtual experiences. We use the acronym CVE here, representing collaborative VEs.



1 **Cyberspace combines many virtual environments, authored separately but composable for interoperability. Each client-side entity (CSE) can export its geometry, behaviors, and interactions. The development of each CSE must be as simple as writing a Web page.**

> **Theorem 3 (Law of Names):** When a scientific research area has a single, generally accepted name, the research in that area is either essentially unexplored, proprietary, or essentially complete.

Computer-generated autonomy (CGA) will certainly become inextricably melded with computer graphics. The video-game industry already spends as much as two-thirds of product development effort on intelligence for characters—most of it not reused. Graphics researchers should hardly begin rediscovering the achievements of the artificial intelligence community—the situation calls for cooperation between the producers and consumers of CGA technology. While this article focuses on other aspects of CVEs, the National Research Council's report on Modeling and Simulation[2] provides excellent recommendations for future avenues of research in CGA, such as behavior adaptability and human representation. Many of the infrastructure requirements for CGA-enhanced systems with a large number of synthetic actors are the same as those needed for large-scale CVEs.

The space of spaces

Having demonstrated a logical consequence of the current research phase (rendering) and postulated that the next phase is CVE research, it remains for us only to conjecture on the future of CVEs. All indications are that every CVE, and networked electronic data of all kinds, will converge to a single environment inhabited simultaneously and persistently by millions. This space of spaces is the real "cyberspace."

The pertinent issue, then, is exactly what fundamental innovations will be required as steps along the way toward making that cyberspace a reality. Cyberspace can never shut down for maintenance, so it must support dynamic extensibility to its control software. Current CVEs comfortably support only a few users, and the most populated CVE applications provided limited support for a few thousand, so scalability is crucial. Cyberspace will combine many environments and information sources, authored in a piecemeal fashion, so composability is equally necessary. Figure 1 illustrates a cyberspace consisting of many subspaces, where vastly different participants can interact.

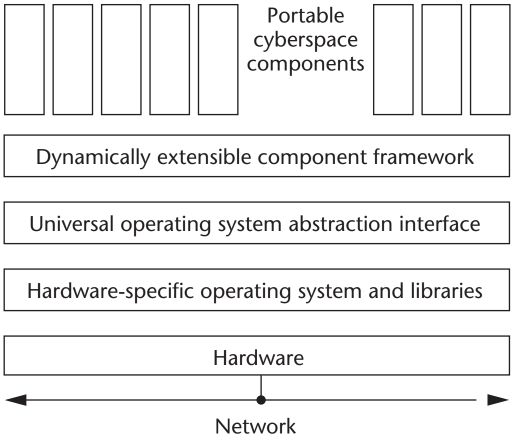
Construction on the globally accessible network infrastructure has essentially just begun. Cyberspace awaits the ubiquity of realistic rendering and high-speed networking; CVE systems developed before that time must fail gracefully as we test the limitations of current resources.

Now we examine each of these requirements in turn, with special attention given to frustrating hindrances and promising efforts.

Dynamic extensibility

The first worlds to constitute the core of cyberspace won't yet be technologically ready for immortality, but they must be ever-present. The loss of key cyberspace environments could prove just as devastating as the loss of a major World Wide Web gateway site. The failure might have no far-reaching effects upon other sites, but would certainly affect user experience. If the beta generation is critical to the long-term success of the technology, then finishing touches must take place transparently.

For successful interaction in cyberspace, its effects must persist. Just like a 24-hour eatery, the doors can never close and the vacuuming must occur with customers present. And like a shopping mall, renovations occur during business hours, behind a privacy screen and a "Pardon Our Dust, but Work We Must" sign.



2 **Client-side entity for cyberspace. Dynamic extensiblity requirements include an architecture-independent plug-in system.**

Supporting such dynamic modification of an executing program, although essential, currently eludes our reach. Most of today's extensible software simply includes a facility for module integration, allowing static linking of plug-ins upon initial execution. Figure 2 presents a framework for a dynamically extensible client-side entity (CSE) for cyberspace. A key to this development is the abstraction from the hardware and operating system, permitting true cross-platform authoring.

This multiple platform support is the single most frustrating issue in engineering dynamically extensible systems. Support for needed code-linking functionality varies widely across operating systems, as do the methods for performing such feats. Cross-platform efforts such as the Java Virtual Machine, which abstract away the underlying platform, have proven moderately successful. So far, any such abstraction that aids us with these architectural issues concomitantly frustrates the performance tuning required for scalability and graceful degradation.

Today's large-scale Internet games provide excellent experimental data for our extensible CVE architecture. The fantasy world of Ultima Online (http://www.owo.com), for instance, supports thousands of users whose actions have a persistent effect on the world and the storyline. The client-side software is extensible, with updates fetched over the network and patched in during initial connection to the world. However, server database maintenance requires daily system-wide downtime, and server software updates cause frustrating (albeit less frequent) open-ended down periods. Ultima Online not only demonstrates the possibilities for remote software updates and persistence, it also exposes the need for true dynamic extensibility.

Scalability

Scalability issues arise when we wish to build CVEs that support compelling interaction between large numbers of widely dispersed players. A number of successful entertainment-based CVEs support tens of players, and special-purpose systems (usually military) that can support up to a thousand have had limited success. But a thousand users won't suffice for cyberspace, or even for the needs of current urban or military exercise simulations.

Network characteristics

The most significant hindrances to scalable delivery of content and events in CVEs stem from characteristics of available network technology, namely bandwidth, latency/delay, and reliability. Network infrastructure upgrades are constant but slow. While the typical corporate desktop machine has evolved many times in the past 10 years, sadly many of us have had the same 10-Mbit Ethernet network connection to that desktop during that time. Though we finally see next-generation networks deployed, at no time in the foreseeable future will the network allow pair-wise updates among thousands of users. It's therefore necessary, for both the short and long term, to conserve bandwidth wherever possible in CVE applications.

Similarly, while we can reduce network delay, we can't eliminate it, thanks to the poor signal propagation properties of light. Network delay causes a host of concerns for consistent, distributed delivery of contents and actions in a shared virtual world. For instance, an action already experienced by some participants may not yet have been delivered to others; accordingly, the ordering of events may vary among clients. Causally related to this is the near impossibility of coordinating a discrete action between delayed clients. While some promising efforts attempt to ameliorate the effects of delay on perceived fidelity (such as time warping and event prediction), scaling a CVE geographically has an unavoidable cost. Note that a major factor in delay is the level of network saturation, so bandwidth conservation should contribute in a major way to the solution here as well.

In the face of unreliable network delivery, CVE systems must deal appropriately with the resulting data inconsistency—similar to the case of excessive delay. Further effort is required in researching dynamically reconfigurable systems to allow continued operation in the face of permanent network or node failure.

Promising efforts

Awareness management techniques can significantly reduce communications volume in CVEs and are vital for scalability. Given some indication of what information participants want, we must reduce communications to the minimum required for realism. Choice of technique depends on system architecture, so often the architecture is selected to suit. For instance, message culling at a centralized server can prevent pair-wise communication between mutually uninterested participants. However, the centralized bottleneck topology isn't particularly scalable, and it increases the latency of each communication. Performing filtering at the recipient eliminates that latency, with concomitant cost in bandwidth and client processing load. Participants can certainly filter messages locally to eliminate that delay, but recipient-side techniques have no effect on bandwidth consumption.

Probably the simplest method for interest management involves regional interest. In such a system, a participant is only aware of (and therefore receiving update messages concerning) other participants within a certain distance or within a predefined grid area of the world. In fact, some systems completely encapsulate these regions as isolated communities and don't allow direct synchronous interaction between separated participants. Other awareness managers use visual occlusion (or attenuation in other media), world topology, or the more promising method of task-dependent explicit functions.

Computing interest as a continuous value gives the system more flexibility. For example, rather than sending full positional updates on unoccluded objects 95 percent obscured by fog, systems could instead transmit reduced fidelity (and reduced size) information. In fact, combined with a specification of a client machine's network and processing capacity, this approach can offer different levels of realism to participants as appropriate. Various network transport protocols should be selected on a per-update, per-client basis, letting the system trade between communications overhead and reliability.

Composability

Composability is the holy grail of the networked VE community. With it we have the ability to dynamically import models and behaviors from one VE into another. We wish to bring an object in over the network and have it instantly work properly in the context of the new virtual world. With such capability, we can imagine the entire user population participating in the authoring of cyberspace, similar to the participation we now see in the World Wide Web.

Supporting that global participation requires either a single standard for the authoring of all digital information or established avenues of data interchange. Authors must have the flexibility to create in the mode most compatible with their creativity, and data should be stored in its most appropriate form.

The key to supporting composability lies in the proper specification of VE components. Given a rich enough understanding of an object, its original environment, and the new environment, it's perfectly possible to extrapolate the object's behavior in the new environment. So far it appears possible to provide sufficient description in only the most limited of domains.

A popular example involves exchanging a basketball across virtual worlds. In an environment whose behavior mimics reality, the ball falls when dropped and bounces when it hits a hard surface. But in a new environment, the possibilities for interaction become endless—a world so warm that the ball bursts into flames, a world so cold that a dropped ball shatters, or a world so small that the ball brings widespread devastation with every bounce. It may not be possible to specify the semantics of object behavior in a universal manner, but object specification methods look promising for composability and interoperability.

Representations of object behavior semantics fall into three categories: informal, formal, and analyzable. Informal semantics are text attachments that the participant is meant to understand, telling the behavior, meaning, or purpose of an item in a VE space. In the previous example, "This is a basketball" would serve nicely. Until major problems are solved in natural language understanding, however, no program or automated agent can use such semantics to assist in navigating or using a VE.

Formal semantics are expressed in some notation manipulable by algorithm. Thus, any computer language can represent formal semantics, and in fact this happens in practice. The meaning or behavior of an object is stored as a code block, then the semantics are realized by execution. However, nearly the same problem of automated understanding exists for this extreme as for semantics in plain text. The portability of code in a programming language is just as questionable as informal semantics in a particular human language.

Analyzable semantics are formal semantics expressed in a notation less powerful than arbitrary code. The goal is to use a notation that some inference engine can operate on, thereby allowing automated agents to "understand" the behavior of objects and make decisions based on the encoded information. If this problem were solved, then other problems in CVE composition could be attacked on a deeper level. We see no broad, effective solutions to the analyzable semantics problem, though components of a future solution are being researched.

Effective definition and use of semantics depends on a common knowledge base and domain of discourse for the participants in a CVE. The term "ontology" denotes this shared semantic domain. An ontology specifies conceptualizations used to help programs and humans share knowledge. In practice it consists of definitions of representational vocabulary, including axiomatic theories. Such procedures, along with semantics processing tools, must be integrated into upcoming CVE projects for composability of VEs to become a reality. Equally important, behavior and interaction protocols must be presented as abstract data to a VE, rather than buried throughout the implementation, if those protocols are to be extended through dynamic composition.

End game

We hope these suggestions give a hearty push toward building the cyberspace of which we can now only dream. We're barely on the edge of the developments we expect in the next century, and nowhere close to what science fiction has told us to expect. Now we need to build and discard prototypes, and occasionally field our successes.

Experience is our apple pie. With the advent of cyberspace we'll have yet another recipe with a substitution for the world around us—essentially, lemon-soaked artificial experience. Still, cyberspace will carry us to realms we cannot even conceive today, just as television has. We humbly suggest that no one allows their children to log on for more than two hours a night. ■

References

1. S. Singhal and M. Zyda, *Networked Virtual Environments—Design and Implementation*, Siggraph Series, ACM Press Books, New York, 1999.
2. National Research Council, *Modeling and Simulation: Linking Entertainment and Defense*, M. Zyda and J. Sheehan, eds., National Academy Press, Washington, D.C., 1997.

Contact the authors by e-mail: capps@computer.org, kent@watsen.net, and zyda@acm.org.