

Web3D in Ocean Science Learning Environments: Virtual Big Beef Creek

Bruce Campbell, Paul Collins, Hunter Hadaway, Nick Hedley, Mark Stoermer

University of Washington

Mail Stop 352142

Seattle, WA 98195

(206) 685-5356

brucedc, pdavidc, hunterh, nix, mstorm @u.washington.edu

ABSTRACT

The Virtual Reality Modeling Language (VRML), Java 3D software development packages, and World Wide Web (the Web) offer great potential for delivering three-dimensional, collaborative virtual environments to broad, on-line audiences. These capabilities have significant potential in ocean sciences, so a visualization environment was developed to explore these possibilities. The University of Washington's Virtual Big Beef Creek (VBBC) project has been continuously refined since its initial implementation in April 1999. VBBC affords users the ability to navigate through a data-rich representation of a physical world estuary on Washington State's Olympic Peninsula. One important project goal is to give users a better sense of the overall watershed before they venture out to experience it in person. A second significant goal is to provide a single on-line repository for geo-referenced data obtained through fieldwork (both quantitative and qualitative). The research team has gained insight into application improvements through the participation of Ocean Sciences graduate students, video game enthusiasts, and the general public. In this paper, research challenges, project successes, and project shortcomings are discussed that may inform the larger Web3D community.

Categories and Subject Descriptors

I.6.5 [Computer Graphics]: Simulation – Model Development;

I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Virtual Reality.

General Terms: Design, Human Factors, Languages.

Keywords: Virtual Worlds, Virtual Environments, Virtual Geography, Interface Paradigms, VRML.

1. INTRODUCTION

The process by which students learn about Ocean Sciences and other related Geo-Sciences is complex. The underlying processes studied by students are complex. Geographical Information Systems (GIS) technology is more effective when providing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Web3D '02, February 24-28, 2002, Tempe, Arizona, USA.

Copyright 2002 ACM 1-58113-468-1/02/0002...\$5.00.

learners with tools they can use to effectively build conceptual models. Given a National Science Foundation (NSF) grant to determine the effect virtual environments have on students' ability to understand complex scientific processes, the VBBC project team set out to build an on-line tool in which to carry out research. We believe that, in desktop interface settings, the third dimension adds value to the learning process and, as a result, began building a multi-participant, three-dimensional (3D) environment to support students in April 1999. Soon after conception, the project scope confronted issues of representing environmental process and change over time (requiring 4D interfaces).

Within the project team, environmental content is developed by the University of Washington's Center for Environmental Visualization (CEV). Geo-spatial visualizations developed at the CEV focus on the marine and coastal environments of the Northeast Pacific and the Pacific Northwest. Spatially registered representations of ecosystems and environmental processes are generated using scientific datasets from the University of Washington Earth Science programs at the Big Beef Creek field station [1] and multidisciplinary PRISM research project [2]. VBBC development picked up substantially upon agreement to meet the needs and aims of NSF REPP #9873620. This grant studies the effectiveness of virtual environments for helping students understand complex processes and has provided the team the ability to bring VBBC interfaces into the classroom. The primary University undergraduate users consisted of 24 third and fourth-year students participating in an introductory GIS/GPS class. Their input, as well as the input of the instructor, guided us to design and develop the tools necessary to teach remote sensing concepts within a virtual environment. They provided our best feedback loop.

Long-term, expected VBBC users include researchers, instructors, and both formal and informal lifelong students. Web connectivity provides access for younger students and those not physically on campus. Instructors can connect with one or more students for formal or informal tutoring sessions. The wide range of intended participants drives the need for interface flexibility that we build into our software architecture through object-oriented design principles. The range also requires us to focus on low cost delivery to classroom and living room, a focus today in all our exploration aids.

Looking for an appropriate development platform that could incorporate all project goals, members of the team began work with Java and Java 3D in October 1997, building a Virtual Playground (VP) application framework on which new virtual

environments could be built and shared [3]. The VP uses the vrm1-java3d working group's vrm197.jar archive to load any VRML97 compliant 3D content and adds utilities to convert .wrl files into native Java 3D classes. Currently, all geo-referenced models are registered in Universal Transverse Mercator (UTM) Region 10 (northern hemisphere) coordinates, produced and exported in most cases by NAG's IRIS Explorer software [4]. Perhaps more important than content loading, the VP includes a layered networking infrastructure for communicating across the Web. The VP supports four modes: broadcast, multicast, client-server, and stand-alone. Client-server mode gets the most use and relies on the Open Source VNet server package to communicate both scene-graph transformation and text messaging packets [5].

Participants in a VBBC session download and run the Java 3D-based application on a Linux, Windows, NT, or Solaris operating system. Upon launching the system, users configure their mode of interaction with the 3D data-rich environment. They choose an avatar (representation of themselves in the 3D environment) from a collection designed with a Pacific Northwest/research theme in mind, but more importantly provide metaphors by which users move around the 3D environment, and the perspective from which they view phenomena and content in the virtual world. By default, users enter with a bird avatar (to get a god's eye view) and identify themselves by name. Once the world loads, each participant is free to fly around the world to investigate the default content set, watch other participants fly around the world, or change their avatar to experience VBBC from another being's perspective (human, scientist, whale, fish, etc.). Their avatar changes to inform others of their new navigation constraints (based on abilities of the physical world animal). Participants in human or scientist mode have the ability to perform inquiries such as getting weather data, remote sensing data, or performing distance measurements. Users select appropriate tools from menus or toolbars to perform each task. Traditional communication tools such as chat and asynchronous messaging are among those available.

VBBC development challenges fall into four categories: navigation, data access, data registration, and user interface. Navigation issues arise as users struggle to find a natural locomotion using only the mouse and keyboard. Data access issues arise as the repositories of available geo-referenced data grow logarithmically and a security layer guards sensitive data. Data registration issues arise as different resolutions and units exist within available sets. User interface issues arise as added features and the third dimension intensify VBBC complexity. While trying out solutions to confront usability difficulties, the VBBC team tries not to reinvent solutions that have demonstrated success in other applications or have a wide support base. Application deployment issues loom on the horizon as a have-to-face-someday concern since many in our interested audience request the application to run on their Macintosh computers.

In hindsight, we admit we could have done further research beyond being satisfied with the vrm197.jar loader loading all necessary VBBC content as standard VRML97 files. As the team turned to focus more on the interface and usability issues that were the primary feedback of student users, we missed the opportunity to contribute to emerging standards efforts.

The GeoVRML community had already begun providing a written vision for standardizing geographical-based data set formats as

early as the VRML'98 Symposium [6]. By early 2000, Reddy, Iverson, and Leclerc published their landmark GeoVRML paper, *Under the Hood of GeoVRML 1.0* [7]. Specifically related to ocean sciences, McCann had begun using GeoVRML to visualize remotely operated vehicle dives [8] and had joined Brutzman and others in investigating interactive 3D data visualization in Monterey bay, California. [9].

As the primary grant for the VBBC effort expires, the team has proven the reusability of the software design by using the same basic Java packages to let users investigate underwater environments deep undersea in the Northeast Pacific Ocean. Only with that success in hand have project members fully realized the promise of using the VBBC interface to explore GeoVRML-based content worldwide. As Web3D 2002 approaches, the team has agreed to implement a more general ocean sciences System for Environmental Exploration (SEE) [10]. We look to form partnerships with conference attendees of similar interest.

2. DESIGN DECISIONS

2.1 Navigation

Earlier applications based on the VP architecture tried to emulate successful VRML viewer plug-in navigation modes from that era. Users chose from four modes (walk, fly, transport, and examine) and the keyboard and mouse responded differently based on the chosen mode. First time 3D cyberspace users often exhibited their discomfort level through clunky behaviors where they attempted one thing at a time even though they expressed wanting to perform multiple transformations simultaneously. Over time, the VBBC team began to consider the navigation approaches of popular 3D action video games (where timely navigation is critical to successful play).

Currently, VBBC participants can choose first-person mode to move with keyboard arrow keys and rotate with the mouse. Unlike VRML viewer interfaces, first-person mode navigators don't need to press mouse buttons at all for locomotion. Using the Java Native Interface (JNI), the VBBC connects to machine-dependent mouse events to provide mouse movement navigation outside of the application window (C source code available at : www.cev.washington.edu/lc/TMOPENSRC/). Users can continue navigating even when the Java mouse cursor runs up against the edge of the display. This navigation mode immediately paid dividends in terms of a reduced learning curve (as first-person mode experienced users picked up VBBC navigation faster and vice versa). The VBBC team continues to look at successful 3D action video game navigation modes and follows a flexible, object-oriented, code architecture to rapidly implement and test those modes.

The VBBC constrains avatar movement, simulating the physical world abilities of each avatar. For example, only a bird avatar can leave the ground plane to navigate from high above. All other avatars must leave their avatar on the ground and use the camera tools to gain a high altitude perspective. Whale and salmon avatars are restricted to appropriate waterways. Scientists can use simple, flat cursor avatars to emulate popular GIS tool appearance. Since each user can freely change from one avatar to another, the constraints become an opportunity for experiential learning and can evolve to become part of scripted game play.



Figure 1 – Menu Bar and Toolbar



Figure 2 –Navigation Feedback Panel

Users can choose between two different modes of navigation, fly and walk. Both allow the user to move and rotate on the basis of first-person navigation with the possibility of additional restraints. Avatars are constrained to appear perpendicular to the ground whenever navigating in walk mode, whereas fly mode has no such constraints. Users can toggle between navigation modes and viewpoints via the menu bar or standard toolbar (see Figure 1).

Users receive navigation feedback via a control panel along the bottom of the application window. The control panel provides (from left to right) system messages, compass, and current position readout in UTM coordinates (see Figure 2).

2.2 Data Access

The VBBC builds a data access strategy on the Structured Query Language (SQL) by storing and accessing data in an SQL-compliant Relational Database Management System. Though we currently use PostgreSQL on the Red Hat 7.1 operating system, we can rapidly export and import our data into other RDMS products. Every content record stored contains a northing, easting and elevation field that provides the VBBC a UTM coordinate location when loading content. For example, a panoramic image at the mouth of Big Beef Creek has UTM coordinate (516408.1, 5277556, 11.2) in Zone 10 (northern hemisphere). Users add new content via tools within the VBBC or HTML forms accessible through a Web browser. Both methods call Java servlets that connect to the RDMS via JDBC and execute INSERT SQL commands. Users load and remove catalog content using standard SELECT and DELETE commands, taking full advantage of the WHERE clause to filter based on date, author, location, and other metadata fields. Icon-textured cubes identify media types that require a helper Web browser application to view in separate windows. Unique 3D objects identify content to be seen in world. Figure 3 shows the link to a surround panoramic accessed from a red cylinder icon at the location where the photography was taken. Alternatively, participants can choose to load surround images as textures on surround geometry in world.

Some database tables within the content library take advantage of the PostgreSQL large object blob type to store binary streams of remote sensing data. Reader classes read blob fields and create procedural texture maps based on the data. Much of our data uses the Band-Interleaved Line (BIL) format. So far, primitive tools exist for swapping and blending textures in real-time. GIS tools such as Arc View and ERDAS provide file format standards and interaction tools we should consider in future VBBC releases.

2.3 Data Registration

All our VBBC data is stored in UTM Zone 10 (northern hemisphere) coordinates, given in meters. In order to load datasets of varying resolution, we built a range of terrain and procedural texture map classes that facilitate model loading. Users can load terrain for any sub-region of our current coverage geography. They can map content images onto loaded terrain or request that only the terrain covered by a content image be loaded.

As we move to cover larger geographic areas (certainly beyond UTM Zone 10) and open the application to include datasets maintained outside of campus, we encounter datasets registered to latitude-longitude coordinates. As we wrote this paper, we were investigating available coordinate conversion tools that translate back and forth between UTM and lat/long. Upon reviewing the GeoVRML 1.0 specification of the Web3D GeoVRML Working Group, we became interested in using the extended GeoVRML nodes to represent UTM and lat/long coordinates directly. Mapping our data to an earth ellipsoid allows us to participate in larger shared ocean science and geo-science efforts. We began a conversion to the GeoVRML standard in October 2001 and, when complete, will be included in the VBBC software at www.cev.washington.edu/lc/TMOPENSRC. We are especially interested in using successful streaming, level-of-detail approaches that continue to emerge in geographical visualization interfaces and continue to pursue ideas that incorporate time scale registration.

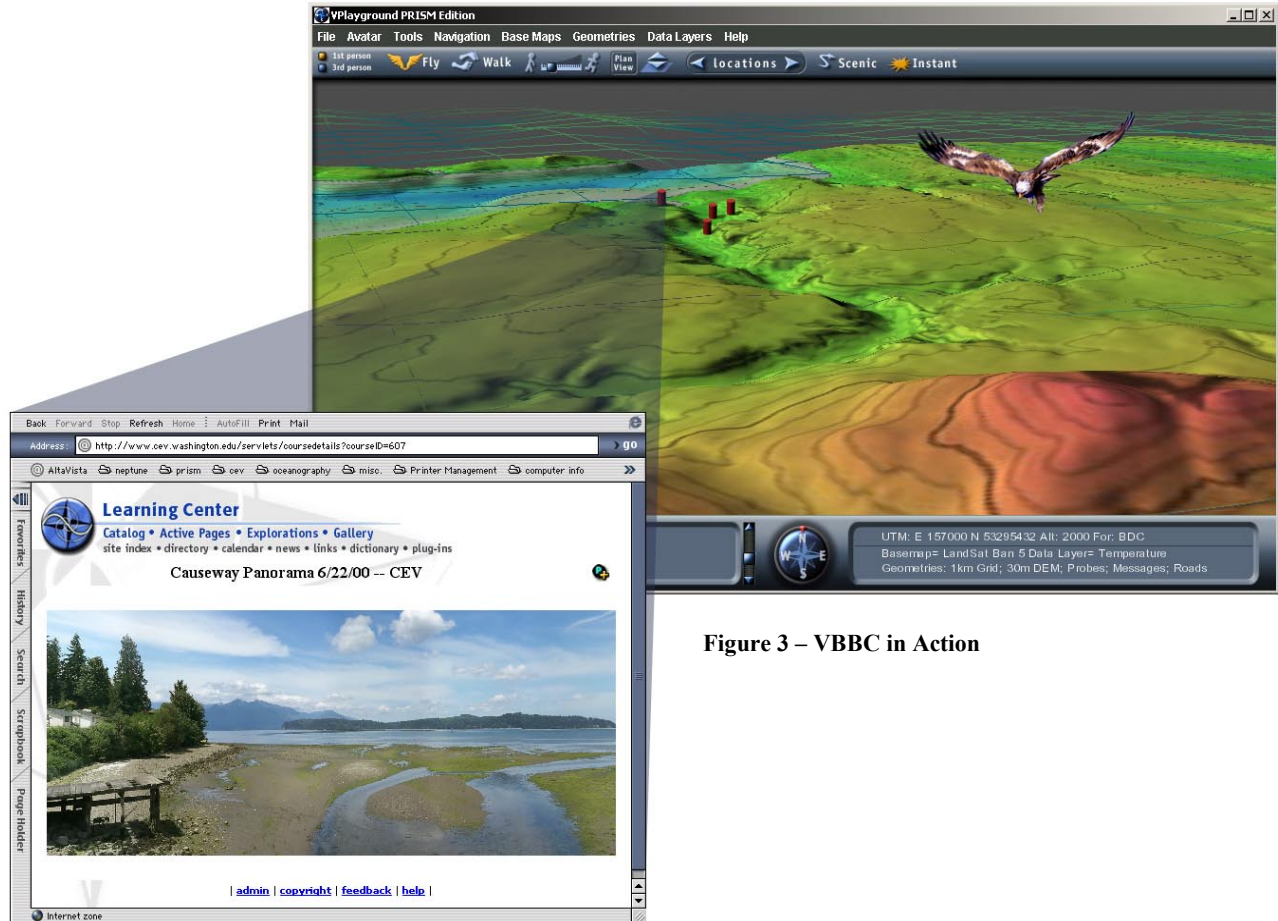


Figure 3 – VBBC in Action

2.4 User Interface

We designed the VBBC GUI for ease of use. Most users immediately understand how to use the menu bar and toolbars in light of their experience with office productivity applications. The Java 2D package made toolbar visuals easier to implement as we maintain multiple layers in our graphics. The layers allow for visual highlighting and rollover user feedback effects. We have undergone several iterations of our menu hierarchy as we get usability feedback from first time users. Currently our menu bar

provides access to eight drop-down, cascading menus (see Figure 4).

The File menu allows participants to load and unload content files that aren't already provided as options within the other menus. Users can save the current state of the world and reload a previously saved state. The Avatar menu lets users change their avatar representation when appropriate. The Tools menu provides an array of collaboration tools for use within the VBBC world. A Chat tool lets participants send synchronous text messages to each

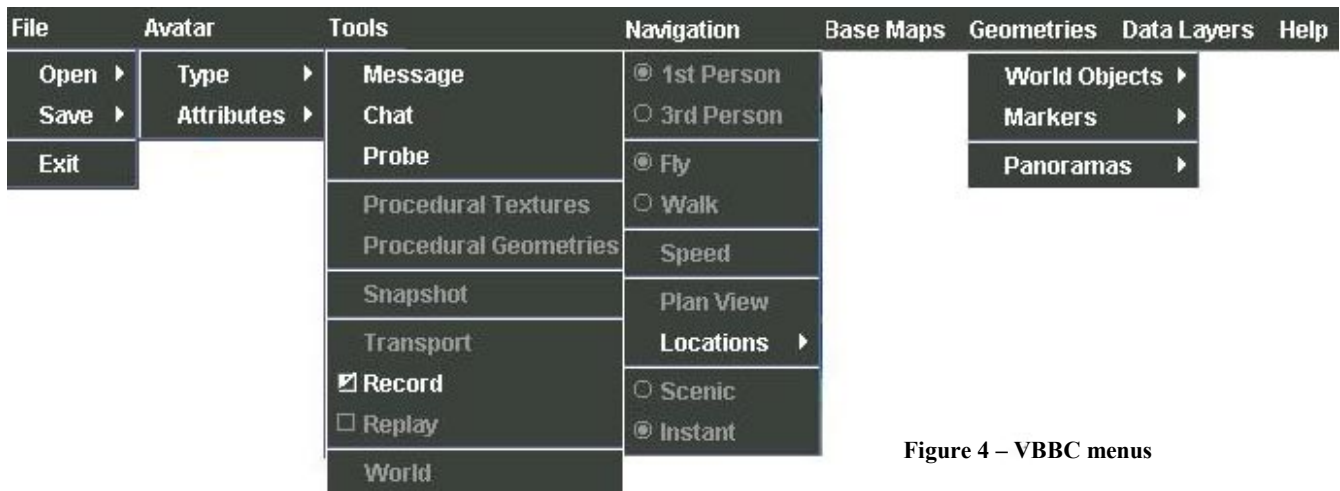


Figure 4 – VBBC menus

other while connected. The Message tool lets users create or reply to asynchronous messages in the world. The Probe tool lets users probe geo-referenced datasets to see location-based scientific data which then is made available in a table for others to see (used to simulate scientific discovery for learners). Users can also take snapshots of the current VBBC canvas and save them in the catalog with annotations. Eventually, students will be able to save their current viewpoint as a panoramic QuickTime VR file, allowing them to look around in a QTVR helper application without loading the VBBC.

The Navigation menu allows participants to fine-tune their mode of locomotion or change from one mode to another. Users can access a list of pre-set locations (like VRML viewpoints) from the Navigation menu as well. The Base Maps menu allows users to choose from cascading lists of geographical maps. When chosen, the base map replaces the current base map and attaches to the terrain according to its geographic coverage. The Geometries menu gives users access to terrain and other 3D geometries considered part of the current loaded world. The Data Layers menu lists scientific data texture maps that users select to drape over terrain. The Help menu provides a link to a Web-based help reference set of documents.

2.4 User Interface

We designed the VBBC GUI for ease of use. Most users immediately understand how to use the menu bar and toolbars in light of their experience with office productivity applications. The Java 2D package made toolbar visuals easier to implement as we maintain multiple layers in our graphics. The layers allow for visual highlighting and rollover user feedback effects. We have undergone several iterations of our menu hierarchy as we get usability feedback from first time users. Currently our menu bar provides access to eight drop-down, cascading menus (see Figure 4).

The File menu allows participants to load and unload content files that aren't already provided as options within the other menus. Users can save the current state of the world and reload a previously saved state. The Avatar menu lets users change their avatar representation when appropriate. The Tools menu provides an array of collaboration tools for use within the VBBC world. A Chat tool lets participants send synchronous text messages to each other while connected. The Message tool lets users create or reply to asynchronous messages in the world. The Probe tool lets users probe geo-referenced datasets to see location-based scientific data which then is made available in a table for others to see (used to simulate scientific discovery for learners). Users can also take snapshots of the current VBBC canvas and save them in the catalog with annotations. Eventually, students will be able to save their current viewpoint as a panoramic QuickTime VR file, allowing them to look around in a QTVR helper application without loading the VBBC.

The Navigation menu allows participants to fine-tune their mode of locomotion or change from one mode to another. Users can access a list of pre-set locations (like VRML viewpoints) from the Navigation menu as well. The Base Maps menu allows users to choose from cascading lists of geographical maps. When chosen, the base map replaces the current base map and attaches to the terrain according to its geographic coverage. The Geometries

menu gives users access to terrain and other 3D geometries considered part of the current loaded world. The Data Layers menu lists scientific data texture maps that users select to drape over terrain. The Help menu provides a link to a Web-based help reference set of documents.

The object-oriented design of toolbar-related classes allows for quick and easy customization or context-sensitive swapping. An actions class hierarchy allows GUI designers to quickly re-map menu, toolbar, or keystroke events to handling routines.

2.5 Application Deployment

Currently we distribute the VBBC application as a single archive file, downloadable from the Web. In order to participate, users must first install a Java 2 SDK or run-time environment and install the Java 3D packages. Then, participants can extract our archive into a recommended subdirectory and run a batch file that sets their PATH and CLASSPATH environment variables. Users access scientific data and VRML models via the Java java.net package, relying heavily on the URL and URLConnection classes.

The VBBC connects users via a client-server networking model using any one of four different server packages – one developed in house initially and three from other Open Source projects. The underlying VP architecture allows for connection via broadcast (for easy demonstration on a portable hub) or multicast as well.

Ideally, willing participants would open their favorite Web browser and click on a link to run the VBBC in an applet panel. In a naïve user's case, the VBBC would load as smoothly as a respective HTML document of the same size. However, even when we have the Java 2 plug-in working with test Java 3D applets, the VBBC, when run as an applet, sits in loading mode without feedback or exceptions and without completion for minutes on end. Knowing that we have difficulties getting the VBBC to work in a Web browser, we've abandoned our attempts to distribute it today to a general Web audience via a Web page at this time. But, we continue to look to the broader Web3D community for deployment process improvements over time.

2.6 Multi-Modal Collaboration

The Human Interface Technology Laboratory (HIT Lab), the incubator for VP development, continues to partner with the CEV to investigate advanced interfaces for 3D computing environments. Since the work done on immersive Virtual Reality (VR) in the early 1990's [11], the HIT Lab has expanded by working with Augmented Reality (AR) via the ARToolkit software libraries and natural gesture interaction via the HI-SPACE project. The ARToolkit enables the overlay of three-dimensional virtual images on a video stream from the real world so that both the real world and virtual imagery can be seen at the same time (see Figure 5) [12]. The HI-SPACE libraries track all hands and objects placed over the interaction space. These serve as input for applications and replace or augment other input devices such as keyboards, mice, or speech (see Figure 6) [13].

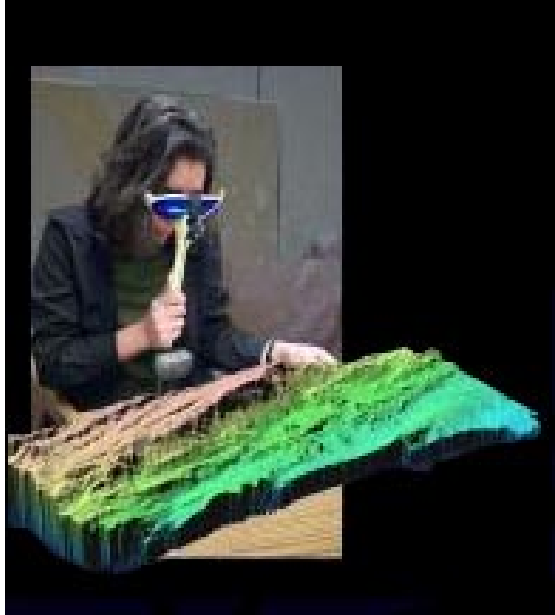


Figure 5 –ARToolkit in action.

VBBC content, since stored in VRML97-compliant format, loads into both the ARToolkit and HI-SPACE environments. Current plans include expanding the collaboration mode beyond desktop clients to VR, AR and other presentation clients. Implementing the GeoVRML extensions in the ARToolkit VRML parser would facilitate shared use of VBBC datasets.

To emphasize the collaborative nature of future development, the CEV is building a collaborative environment learning lab across campus to network with current HIT Lab facilities and will perform the research behind the SEE development at both institutions.



Figure 6 – HI-SPACE in action.

3. DISCUSSION

After four years of working on a 3D, multi-participant virtual environment platform distanced from the VRML community, VBBC team members are again supportive of any efforts to integrate the platform with Web3D standards. The underlying technology we build upon often seems a surrogate Web browser and we encounter frustrations of reinventing the wheel. Iteration after iteration of re-design has taught us a great deal and we've gained a strong belief in the process of trying out new or alternative approaches. As a result of always changing, our code becomes more modular over time. After four years, it's easier to change part of the system without affecting the rest. In other words, it is a perfect time to incorporate suggested Web3D standards, even if through a piecemeal approach.

If we could spend less time on the base 3D technology, we would spend more time specializing on 4D GIS for the Web. We consider time to be a critical element when presenting geo-science information and we have access to real-time streams of data. Weather interfaces, for example, provide time-sensitive visualizations. We would like to spend time investigating controls for time slicing and speed. Dynamic engineering of real-time datasets into visualization output based on time requests would allow the flexibility a Web audience desires. Representing multiple collaborating entities using the same dataset at different time points presents a significant challenge. We have not seen much research into the fourth dimension discussed on the Web and consider it a rich area to pursue.

We find ourselves required to investigate flat Web page front-end delivery vehicles as we strive to deliver 3D solutions. Many of our campus partners are not ready to implement 3D on the Web. So, in the meantime, we find ourselves looking at storing information in flexible formats that lead to both 2D and 3D presentation. The VBBC connects to the same back-end data system as its sister Web page draws dynamic data from, including real-time data streams. Through flat Web forms, the Web audience can geo-reference BBC-related data, enter it into a catalog, and then provide visual links in the VBBC.

We consider informal ocean science education our niche and look to larger efforts with which to collaborate on the subject. For each new sensor a scientist places in the ocean, we realize there is an opportunity for that device to play in a Web-based informal science education program. Connecting all the worldwide efforts into a common technology framework will make all of our work more accessible and marketable.

4. ACKNOWLEDGMENTS

As mentioned above, we would like to thank the following groups for their support of VBBC:

NSF REPP. VBBC is funded in part by NSF REPP #9873620. This grant studies the effectiveness of virtual environments for helping students understand complex processes.

BBC (depts.washington.edu/bbc/). The University Washington's Big Beef Creek Field Station on Hood Canal offers teaching and research activities that include natural history, artificial rearing studies, and whole-life-history studies of organisms that alternate between fresh and salt water.

PRISM (www.prism.washington.edu). The Puget Sound Regional Synthesis Model is a major University of Washington program looking at the movement of water through atmosphere, across the land, into rivers and streams and throughout waterways of Puget Sound.

5. REFERENCES

- [1] Big Beef Creek Field Station, <http://depts.washington.edu/bbc/> (accessed September 18, 2001)
- [2] PRISM: Puget Sound Regional Synthesis Model, <http://www.prism.washington.edu> (accessed September 18, 2001)
- [3] Schwartz, P et al, (1998). Virtual Playground: Architectures for a Shared Virtual World. In Proceedings of ACM Symposium on Virtual Reality Software and Technology 1998, pp. 43-50.
- [4] NAG, IRIS Explorer™ Home Page, http://www.nag.co.uk/Welcome_IEC.html (accessed September 18, 2001)
- [5] What is VNet? <http://vnet.sourceforge.net/> (accessed September 18, 2001)
- [6] M. Abernathy and S. Shaw (1998). Integrating Geographic Information in VRML Models. In Proceedings of the Third Symposium on the Virtual Reality Modeling Language, Monterey, CA. February 16-19, pp. 107-114.
- [7] M. Reddy, L. Iverson, and Y. G. Leclerc (2000). Under the Hood of GeoVRML 1.0. In Proceedings of The Fifth Web3D/VRML Symposium. Monterey, California. February 21-24, 2000.
- [8] M. McCann, Remotely Operated Vehicle (ROV) Dive Visualization, <http://www.siggraph.org/publications/newsletter/v34n2/contributions/mccann.html> (accessed December 17, 2001)
- [9] M. McCann, J. Paduan, D. Brutzman (1999), Interactive 3d Data Visualization In Monterey Bay, Poster session and demonstrations, Monterey Bay National Marine Sanctuary Symposium, Seaside, CA, March 20.
- [10] The Center for Environmental Visualization, System for Environmental Explorations, http://www.cev.washington.edu/see/exp_sys.html (accessed December 17, 2001).
- [11] Bricken, M (1991). Virtual Reality Learning Environments: Potentials and Challenges. Computer Graphics: SIGGRAPH 91, 25(3), 178-184.
- [12] Billingham, M. and Kato, H. (1999). Collaborative Mixed Reality. In Proceedings of International Symposium on Mixed Reality (ISMR '99). Mixed Reality--Merging Real and Virtual Worlds, pp. 261-284.
- [13] May, R.A. (1999). HI-SPACE: A Next Generation Workspace Environment. Masters Thesis, Washington State University, Pullman, WA