4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage
VAST (2003)
D. Arnold, A. Chalmers, F. Niccolucci (Editors)

# Portals: Aiding Navigation in Virtual Museums

Ioannis Kotziampasis, Nathan Sidwell and Alan Chalmers

Department of Computer Science, University of Bristol, Bristol, UK

**Abstract**

*Distributed virtual environments offer an efficient way for a number of users to create complex 3D virtual worlds. However, navigation within these virtual environments can be significantly hampered by the lack of visibility between parts of the world, particularly when these parts are on different machines. This paper describes the use of portals to connect distributed virtual archaeological environments in such a way that the interconnected virtual environment will be visible from different machines in an efficient manner. A case study shows how a virtual museum can benefit from such a system.*

**Keywords:**
*Portals, Distributed Virtual Environments, User Navigation, VRML, Virtual Museums.*

## 1. Introduction

The World Wide Web is one of most successful and rapidly evolving projects in the history of computers. An important factor for this success is the ability for users to publish data and generate links to other web sites independently of each other. Virtual environments (VEs) are increasingly being used to present archaeological information, including site reconstructions, virtual museums, site finds etc. Although the web offers some significant advantages for information sharing and publishing, navigation within this large distributed environment can be difficult and frustrating.

Distributed publication of archaeological information is attractive as the data may well be the property of several organizations and such a distributed virtual environment enables the individual data to remain securely hosted by the owner while the whole collection is accessible by a wide audience. The only standard currently existing for embedding Virtual Environments into the web is VRML 2.0 [1]. This is a description language that can be used to describe non-distributed (single site) VEs with simple animation. VRML can also define links to other VEs, however, it does not allow any visual connectivity between them. The user thus does not know where a link leads to. In addition, when a new link is activated the old virtual world is deleted and the new one is loaded. Furthermore, VRML does not support multi-user interaction.

In a multi-user virtual environment, users run an interactive application which allows them to communicate and interact with other users inside the same VE. The user sees, by means of a client application, a 3D rendered world and any other users inside it. The other users usually take the form of some 3D models, known as avatars, for example virtual humans or something simple such as a box with their name on it. The VE, as well as the users that are inside it, is typically hosted by a single server. Since the capabilities of one computer system are limited, the number of users that can be supported and the size of the world is limited as well. A further difficulty with the web is its enormous size and rapidly changing nature. This results in any virtual environment often having a significant number of misleading, broken and unavailable links at any one time.

One of the concepts of cyberspace is that in the near future people from all around the world will meet, chat, shop, educate etc in shared virtual environments. Events will take place in such virtual places and the users that populate them will participate and interact with each other. The challenge thus for future distributed VEs is to have a highly scalable and expandable environment that is able to host a significant number of users as well as to provide an efficient, user-friendly connection mechanism between its different parts. This VE should be maintained and minimise user frustration from misleading and broken links.

### 1.2 CurlSpace

CurlSpace is a 3D web browser, which has been implemented using C++ and OpenGL for 3D graphics, using VRML for basic geometry description. Furthermore, CurlSpace is a system that supports separate virtual environments with visual connectivity between them, by using a wormhole-like object we call a portal [2, 3]. These portals, which usually take the form of a door or a window, define the connectivity between environments and the visibility of one environment from another. This allows the user to see through the portal the destination before entering into it. The destination is defined by another portal inside the target world which we call target portal. Now when navigating back and forth, the user can see both the source and destination, before traversing the portal. The linked worlds do not need to be globally consistent, something that would be impossible with a decentralized design [4].

Consistency only need be maintained at the portals. When the user looks at a portal, the environment that is linked is rendered only within the aperture of the portal. Because the rendering is limited, a user sees nothing amiss with a static scene. It is only by exploring that global inconsistencies might become apparent. Figure 1 contrasts a standard VRML links with a CurlSpace portal.

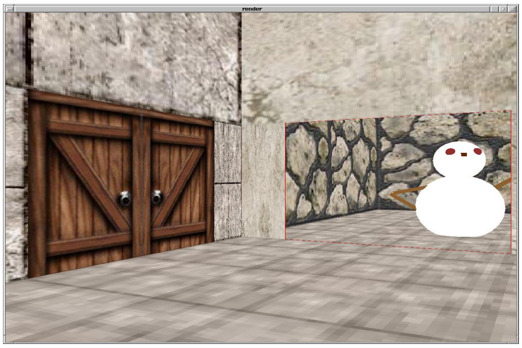

**Figure 1:** VRML anchor link (left) and CurlSpace portal (right).

Portal links allow each virtual world to be hosted on a different server making the VE highly scalable. Furthermore CurlSpace has multi-user support by using a client-server architecture that supports a significant number of users inside the virtual environments.

### 1.3 Related Work

There have been a number of other approaches for scalable region management in large-scale distributed virtual environments (DVE) systems. In general, these systems represent a virtual environment as a set of independent entities each of which has a geometric description and behavior. Entities can be static (e.g. buildings, terrain, etc.), autonomous (e.g. agents), or user-controlled (e.g. user avatars). A DVE can consist of many entities but each one is usually managed by one workstation.

Multi-user environments can be characterized by their approach for message distribution. Reality Built for Two[5] is based on unicast peer-to-peer architectures. SIMNET [6] uses broadcast peer-to-peer architectures to send updates to all of the other participating workstations simultaneously. NPSNET [7] uses multicast peer-to-peer architectures to send updates to a part of the other participating workstations. RING [8] uses a client-server architecture. Communication between clients is managed by servers. That means that a client sends a message to a server and the server forwards it to one or more clients. In this way the server can alter the message in any way if necessary.

Most current virtual museum systems do not use advanced technologies such as CurlSpace and the above systems. For instance, the Virtual Museum of Computer Science in Italy [9] is using a 3D world organized in pods floating in space. Each pod hosts an area of the museum and there are teleport points so that the user can travel back and forth to these floating pods. These teleport points are used to teleport the user locally, to different areas of the same world, but do not provide any visual connectivity of the target areas to where a teleport point can actually lead. This may be adequate for a virtual world of this size, but in larger interconnected virtual environments, not having virtual connectivity can significantly hamper navigation [10]. The technology that the Virtual Museum of Computer Science system uses is WebTalk-I [11]. This uses standard VRML for geometry building and a Java-based TCP/IP communication layer using centralized server control that is capable of hosting limited number of users. WebTalk-II [11] technology has some improvements over standard network architecture by adding support for unicast and multicast peer-to-peer support when required. User management though is still centralized.

## 2. SYSTEM DESIGN

### 2.1 Extending VRML

CurlSpace is based on the VRML 2.0 standard for geometry building. It extends VRML with a special EXTERN PROTO node, whose implementation is internal to CurlSpace. This Portal node has many of the features of a Shape node, but also describes a mapping to a target world, and optional visual effects such as magnification. The portal may have any geometry. When rendering the Portal node, the target environment is rendered through the aperture described by the portal's geometry.

For portal unaware browsers, the EXTERN PROTO Portal node can have an implementation consisting of an Anchor containing a Shape node, which simply renders the geometry using the fallback material specified in the Portal node. Typically the material would be a flat rendering of the target environment in the form of a texture map. CurlSpace uses a client-server architecture in which virtual environments are hosted in several servers. This does not mean that the server that hosts a VRML world need necessarily be the host for the participants of the world. VRML has been extended so that each world specifies the server in which the articipants of this world will be hosted by extending the WorldInfo Node into a CurlSpaceInfo Node.

### 2.2 Rendering Portals

Overall, the are two techniques for rendering worlds that lie beyond portals. The first one is by rendering the target world and applying it on the portal plane as a texture map [12]. The later is by rendering each object inside the target world by calculating the transformation between the source and the target worlds. This way all inter-connected worlds are rendered in the same scene. The later technique has a significant advantage over the first technique which is related with user transitioning from one world to another. More specifically, when an avatar's geometry intersects a portal, the avatar, belongs to both worlds. By using the first technique, this object should be rendered two times [12]. Using the later technique though, the object would be rendered without any problems, since all worlds and avatars that are rendered through portals are rendered in the same scene.

### 2.3 System Architecture

The CurlSpace client, which uses a distributed client-server architecture, is the front-end application that a user uses in order to navigate in VE. Computer systems that are going to accommodate the users of the VE must run the CurlSpace server. Each computer running the CurlSpace server can accommodate one or many worlds, but the world designers should bare in mind that if the number of users/traffic in these worlds hosted by one server grows really large, some of these worlds might be required to be hosted elsewhere in order to maintain reasonable performance.

When the user first runs the CurlSpace client application to visit a world, he/she registers to the CurlSpace server on which this world is hosted as a USER. That means that when the user moves around, the server is updated by the new position and orientation of the user. While the user navigates other portals that are linked with other worlds become visible. In that case, the client becomes a VIEWER of these worlds, meaning that the server that these worlds are hosted is going to update its VIEWERS according to the new position and orientation information of his USERS.

More details on the system architecture of the system will be available on future publications. This paper focuses on the applications of the system as it relates to virtual museums and site reconstructions.

## 3. APPLICATIONS

Virtual environments on the Internet are used by many people mostly for entertainment purposes. Although the ability for a user to create his/her own 3D homepage is not as easy as creating a simple HTML homepage, the number of users who are attracted by the idea of a 3D web grows every day. This section describes some of the applications that a shared virtual environment system could be used and some advantages the CurlSpace's technology would add to these applications.

### 3.1 CurlSpace Features

The use of portal technology allows CurlSpace to have some significant advantages compared to other existing systems in 3D space. These advantages are related with the ability of CurlSpace to support continuous environments. These features include:

**Security:** Virtual environments can be hosted in any server, similar to HTML web pages. This allows organizations to host their 3D worlds on their own servers and still let other users add links to these worlds. Furthermore organizations can give restricted access to only some of their 3D information. Restricted access portal links can be shown to users by a pre-defined texture used on the portal plane. Once the user gives the authentication information required to access this world, the texture will vanish and the world that lies beyond the portal will be rendered, figure 2.

**Prevention of misleading or broken links:** Portals allow the user to see where a link leads to before actually activating it. When navigating using VRML Anchor links, the only information that the user has of where a link could lead is from the shape and/or the texture of the object that the link is placed on. There are cases though that this information is inaccurate, misleading or expired leading to the frustration of the user when trying to access it. By using portals, the user gets an idea of the actual state of a link before traversing through it as well as to what kind of an environment it leads to. Broken links are represented by using pre-defined textures so that the user is notified for the current state of a portal link, figure 2.

**Multi-user Priority Mechanism:** With low bandwidth connections it is important to ensure users that are connected in this way, nevertheless have a similar "response'' from the system as users with high bandwidth connections. Priority mechanisms enable the user to get updates from the users that are 'closer' to them more quickly, where here 'closer' means less 'portals' away.

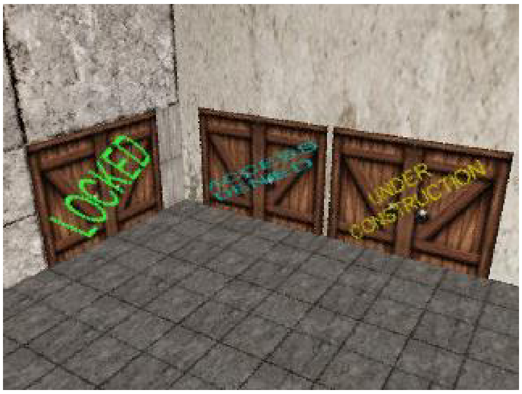


**Figure 2:** VEs using portals.

Linking a virtual world with another one does not require any special programming. The only information required is the URL and the name of a target plane or portal. Although knowing a target plane on the target world might not be complicated, this issue introduces a the need for a global naming concept. HTML sites accessed by a URL usually have a main HTML page typically called welcome.html or index.html which is the standard entry page on a web site. Thus, a standard name for an entry portal or plane might be useful. When a portal is linked to a world specified by a URL only without any portal or plane names, CurlSpace can assume that there is an entry portal name EntryPortal or WelcomePlane.

### 3.2 Sharing geometry in multi-levels

There are cases where virtual environments on the Internet can be heavily populated. This can be disadvantageous for users that simply just want to navigate in the environment and do not want to be disturbed by other users nor interact with them. There is also the case that a certain group of people want to navigate a virtual environment in a way that only users from that same group can communicate between them, for example in game playing.

If, for instance, ten users want to play singles tennis in a virtual tennis court. Of course, only two users can play singles at a time, and having geometry for five tennis courts is not efficient. The solution for this situation is to use different levels of communication. This idea is very similar to the way channels in IRC (Internet Relay Chat) works (Reid 1991[13], Bechar-Israeli 1995, [14], Danet, Ruedenberg and Rosenbaum-Tamari 1997 [15]). In IRC there is a server which accommodates hundreds of users. These users though create different channels according to the topics they want to talk about and users appear on one or more channels according to their interest. When a user is talking inside a channel, only the users that appear in that specific channel are listening. Similarly in CurlSpace, users entering a virtual environment have a level. This level, in a way represents a different universe. So, for the tennis playing problem, 2 users can enter the virtual tennis field in level 1, two other in level 2 and so on. This way, the server manages the communication of users of the same level only. This has a two significant advantages:

**Geometry Sharing:** The same geometry is shared by many different groups of users in different universes.

**Privacy:** Users can communicate and navigate on their own, without being 'watched' by everybody.

## 4. THE VIRTUAL MUSEUM PROTOTYPE

To demonstrate CurlSpace's abilities for virtual heritage applications, a prototype virtual museum has been designed. The purpose of the prototype is to demonstrate that a virtual museum can be constructed by many people, perhaps even independently of each other. Each person models a specific area and then adds links to the other areas related to it. The goal of this is to enable archeologists from all around the world to participate in constructing different types of exhibits relating to one theme and then linking them all together. Figure 3 shows a screenshot of the prototype.

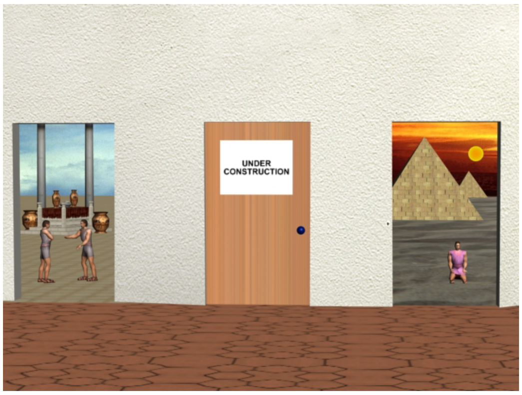


**Figure 3:** Portals within distributed virtual museum.

The screenshot shows three of the virtual worlds which have been put together by users, one of which is still under construciton. While navigating within this distributed virtual museum, a user is able to see what each exhibit has to offer, including the presence of perhaps a virtual guide, before actually entering the environment.

## 5. CONCLUSIONS AND FUTURE WORK

CurlSpace provides an efficient system that supports real-time visual interaction between many users. Another significant advantage is the support of continuous environments by using portals which improve navigation in both consistent and inconsistent distributed virtual environments.

Future work will continue to investigate psychological issues that may arise while navigating in virtual worlds, comparing portals with the traditional VRML approach [10]. This is of particular interest for worlds which may be Euclidian inconsistent.

Finally, we will consider new ways of history keeping inside virtual environments. Since portals can be unidirectional, a new way of history keeping is required since a user may enter a portal and then may not be able to return back from whence he came. This will include a study on how such a system could affect applications such as Virtual Communities, and how to address some other technological issues such as path-finding in VE that are linked through portals.

## 6. ACKNOWLEDGMENTS

This work has been funded by EPSRC grand GR/M27814.

## 7. REFERENCES

1. VRML. Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997, http://www.vrml.org/Specifications/VRML97/
2. D Luebke & C Georges, 1995. ‘Portals and Mirrors: Simple, Fast Evaluation of Potentially Visible Data Sets’, ACM Proceedings of 1995 Symposium on Interactive 3D Graphics Conference, Monterey, CA.
3. David Stuttard & Nathan Sidwell, December 1997, ‘Hulls and Portals’, British Patent Application BPO/ADP 1255003
4. John Barrus, Richard Waters & David Anderson, August 1996. ‘Locales and Beacons: Efficient & Precise Support for Large Multi-User Virtual Environments’, Mitsubishi Electric Research Laboratory, TR95-16a
5. Blanchard, C., S. Gurgess, Y. Harvill, J. Lanier, A. Lasko, M. Oberman, and M. Teitel, 1990. ‘Reality Built for Two: A Virtual Reality Tool’, ACM SIGGRAPH Special Issue on 1990 Symposium on Interactive 3D Graphics, (Snowbird, Utah), 35-36.
6. Calvin, James, Alan Dickens, Bob Gaines, Paul Metzger, Dale Miller and Dan Owen, 1993. ‘The SIMNET Virtual World Architecture’, Proceedings of the IEEE Virtual Reality Annual International Symposium, 450-455.
7. Macedonia et al, Fall 1994. ‘NPSNET: A Network software architecture for large scale virtual environments’, Presence Vol. 3, No 4, MIT Press.
8. Funkhouser, Thomas A. 1995, ‘RING: A Client-Server System for Multi-User Virtual Environments’ ACM SIGGRAPH Special Issue on 1995 Symposium on Interactive 3D Graphics, Monterey.
9. Barbieri T., Garzotto F., et al, September 2001, ’From Dust to StardDust: a Collaborative Virtual Computer Science Museum’, in proceedings Short Papers Ichim 01, Milan, Italy.
10. Ioannis Kotziampasis, Nathan Sidwell, Alan Chalmers, 2003, ‘Portals: Increasing Visibility in Virtual Worlds’, Proceedings of the Spring Conference in Computer Graphics, Budmerice.
11. Barbieri T., July 2000, ‘Networked Virtual Environments for the Web: The WebTalk-I and WebTalk-II Architectures’, in proceedings IEEE for Computer Multimedia & Expo 2000 (ICME) , New York, USA.
12. D. Schmalstieg, G. Schaufler, 1999, ‘Sewing Virtual Worlds Together With SEAMS: A Mechanism to Construct Complex Virtual Environments’, Presence – Teleoperators and Virtual Environments, Vol. 8, No. 4, pp. 449-461, MIT Press.
13. Reid Elizabeth M., 1991, ‘Electropolis: Communication and Community on Internet Relay Chat’ Honors Thesis, University of Melbourne.
14. Bechar-Israeli, Haya, 1995, ‘From <Bonehead> to <cLoNehEad>: Nicknames, Play and Identify on Internet Relay Chat’, Journal of Computer-Mediated Communication 1.
15. Danet, Brenda, Lucia, Ruedenberg and Yehudit Rosenbaum-Tamari, 1997, ‘Hmmm… Where’s All That Smoke Coming From? Writing, Play and Performance on Internet Relay Chat’, In press in Network and Netplay: Virtual Groups on the Internet, edited by Sheizaf Rafaeli, Fay Sudweeks and Margaret McLaughlin, Cambridge, MA,: AAAI/MIT Press.