

October 20, 1998  
RT0282  
Telecommunications 5 pages

# Research Report

## RVE System: Case Study of the Living Worlds System

Ryo Yoshida, T. Muraio, and T. Miyazawa

IBM Research, Tokyo Research Laboratory  
IBM Japan, Ltd.  
1623-14 Shimotsuruma, Yamato  
Kanagawa 242-8502, Japan



**Research Division**  
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

### **Limited Distribution Notice**

This report has been submitted for publication outside of IBM and will be probably copyrighted if accepted. It has been issued as a Research Report for early dissemination of its contents. In view of the expected transfer of copyright to an outside publisher, its distribution outside IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or copies of the article legally obtained (for example, by payment of royalties).

# Reactive Virtual Environment System: Case Study of the LivingWorlds System

Ryo Yoshida \*, Takaaki Muraio, and Tatsuo Miyazawa

Tokyo Research Laboratory, IBM Japan Ltd.

## Abstract

We describe a case study of a multi-user VRML system that supports the LivingWorlds specification. A VRML browser and server that we have been developing were applied to a scenario in the electronic commerce area. We discuss the concepts underlying the system design, and the problems encountered and solutions reached in actually implementing the system.

**CR Categories and Subject Descriptors:** C.2. [Computer Communication Networks]: Distributed Systems; I.3.2. [Computer Graphics]: Graphics Systems; I.3.7. [Three-Dimensional Graphics and Realism]: Virtual Reality

**Additional Keywords:** Virtual Reality Modeling Language (VRML), multi-user environments.

## 1 Introduction

Several research groups are currently developing contents featuring the LivingWorlds multi-user environment. LivingWorlds [4] is a specification related to the file syntax representing the shared-state virtual space in VRML [6]. It was proposed on the basis of technical discussion by the LivingWorlds working group approved by the VRML consortium. In almost all of the con-

tents, an avatar appears in the virtual space, and by controlling the avatar a user can communicate with other users by means of text or voice chat. Of course, applications in this entertainment area are important for developing multi-user environments; however, we are investigating applications in the manufacturing and distribution business areas. Since electronic commerce is one of our target applications, we built a prototype system using our VRML browser and server that enables users to share some contents according to the LivingWorlds specification. We have been developing an experimental VRML browser and server called the Reactive Virtual Environment (RVE) system [5] [7], in which we have tried to integrate our results with those of other IBM research groups, such as a geometrical compression module and a panoramic image generation module. Therefore, we used our browser and server to build the prototype system.

In section 2 of this paper, we give an overview of our system and related work. In section 3, we introduce one example of an application using the system and discuss the direction of our activity. Section 4 is a summary of the paper.

## 2 Prototyped system

### 2.1 System architecture

The RVE system follows a client-server model [7]. The browser supports the VRML Java platform API for Script nodes, which is known as Java in Script Nodes Authoring Interface (JSAI), and uses it as a program to communicate with the server. The communication program is our LivingWorlds module; this was implemented by using the above API, and therefore the module can also run with other browsers supporting the API, such as the WorldView

---

\*1623-14 Shimotsuruma, Yamato-shi, Kanagawa 242-8502, Japan, e-mail:yoshidar@trl.ibm.co.jp

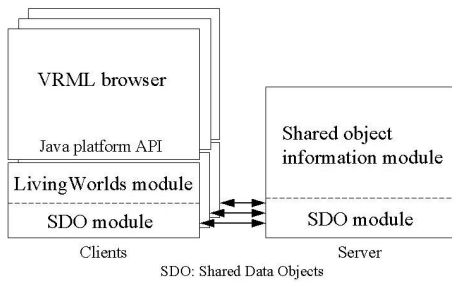


Figure 1: Overview of our system

VRML browser of Intervista Software Inc. The functions supported by the LivingWorlds module are based on the “Core LivingWorlds” document issued in March 1998. Figure 1 shows an overview of our system.

The previous version of the LivingWorlds module was developed on the basis of our infrastructure called “InVerse”; however, the current version was ported onto a new infrastructure called Shared Data Objects (SDO) developed by the IBM Software Solution team, which can support functions similar to those of the Java Shared Data Toolkit by Sun Microsystems. As in the previous InVerse version of the module, the infrastructure basically manages the shared object’s information, such as addition, removal, and positioning of objects in a shared zone. This information changes every moment. Some of the functions specified in the LivingWorlds documentation have not yet been implemented. The definition files by the PROTO and Java class files are located on the local hard disk of each client.

This new LivingWorlds module enables several persistent shared objects to be handled in the zone of a LivingWorlds virtual world. A world author can create shared objects as children of the zone, and the shared objects will be able to exist in the zone from the beginning of the rendering, when the user first enters the zone. The user can control the locations of the persistent shared objects in addition to the location of the avatar representing the user himself/herself in the world. Figure 2 shows the main part of the example file of the VRML vir-

```
#VRML V2.0 utf8
EXTERNPROTO IBMSharedObject [
    ....
] "IBMSharedObject.wrl"
EXTERNPROTO IBMZone [
    ....
] "IBMZone.wrl"
# EXTERNPROTOs for LivingWorlds node
.....

DEF Z Zone {
  private DEF PZ PrivateZone {
    muTech DEF MZ IBMZone {
      # MuTech node for IBM system
      ....
    }
    zone USE Z
    children [
      # persistent S0 #1
      DEF S1 SharedObject {
        private DEF P1 PrivateSharedObject {
          ....
        }
      }
      # persistent S0 #2
      DEF S2 SharedObject {
        private DEF P2 PrivateSharedObject {
          ....
        }
      }
    ]
  }
}
```

Figure 2: VRML file incorporating LivingWorlds’ persistent shared objects.

tual world loaded by the browser, which includes some omissions.

We are additionally implementing an annotating function in our LivingWorlds module. By using this function, the user can attach to the 3D model several types of annotation, such as text and arrows indicating points of issue. In the same way as other shared objects, it will be possible to share these annotations as 3D virtual pointers among the clients for online discussion, like the pen of a virtual white-board.

## 2.2 Related Work

Several studies of multi-user shared-state environments have been reported, while publicly available contents follow VRML rather than the LivingWorlds specification.

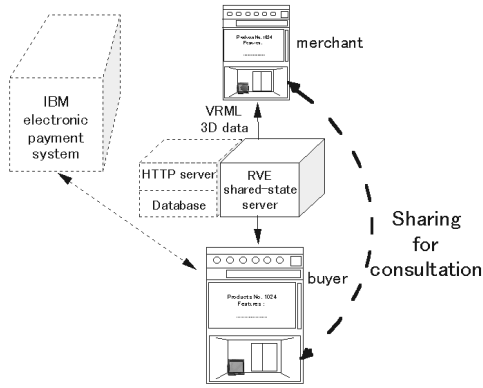


Figure 3: Online shopping system

Sony's Community Place [2] is one system that supports multi-user environments using VRML. Users are offered a chat space called Personal Agent World (PAW). blaxxun's Community Platform [1] is another system that supports multi-user environments. Promotion sites and shopping sites with text chat capability are provided to users of the system.

Most contents, including those of the above two systems, are basically chat applications. The user walks around a VRML world as his/her own avatar, meets other user's avatars, and experiences various kinds of events in the world virtually. The emphasis is thus on entertainment.

### 3 Application example

We are investigating applications in the manufacturing and distribution industries, but applications in entertainment areas, such as chat spaces, are also important for multi-user environments. In line with our direction, we created a sample application of online shopping with 3D simulation and consultation. Figure 3 shows a conceptual system architecture for online shopping.

Our browser and server have been combined with another prototyped system for the electronic commerce area, which was developed by the IBM Taiwan team for the National Palace Museum in Taipei. The purpose of the prototyped system is to provide a Web page for an online shop selling replicated items whose origi-

nals they own. These items include vases, boxes, and hanging scrolls that belonged to the old Imperial Household, and thus even replicas of them would be expensive and bulky for display in an ordinary room. This is the most important point of our application in terms of added value: a potential customer can try putting a desired item in his/her room virtually and visually before actually purchasing it. If the customer is looking for a book at an online shop, he/she can read some information about it and then decide whether or not to buy it. If the item that the customer is looking for is a music CD, he/she can listen to a sample of the music before deciding. In a similar way, a customer can simulate several situations visually by using our combined system before making a purchase decision.

The system for the National Place Museum's online shop consists of Netscape Navigator, the contents in HTML, a database program, and an HTTP server program. The customer can search for items on the basis of historical periods, and get the search results in the form of text explanations and still JPEG images. For several items, 3D VRML models are ready for trial, and the customer can also see an animated image explaining the behavior of the item (e.g., opening a lid) by using the VRML browser.

The system we developed for this application example enables a customer to overlay a textured 3D model of a desired replicated item over a still 2D image captured with a digital camera, and to see the simulated scene as if he/she were looking at the photograph of the item in his/her room. The system can be effectively used to provide a service for consultation between a customer and merchant during a shopping session. Used with an Internet phone, the system will enable the customer to consult the merchant about where the item can be placed or which color coordination is good for the room, enjoying both the system's visual presentation and the Internet phone conversation. The customer's interests before purchasing an item of interior goods, for example, might be: Where it can be placed-on the floor, or on top of the desk? Does it look good as part of the interior? Does its color match that of the wall behind? We therefore developed this system to enable the customer to get answers to such questions. Figure 4 shows a captured screen image of the system. As one of the interior goods, a cylindrical 3D model of the emperor's toy box is dis-



Figure 4: Screen image

played over the background image of the room.

The image of the room must be calibrated in relation to the depth of the room, since its background image is a plain 2D image without any depth. In our scenario, we are assuming that the calibration would be done by the merchant before the simulation. When the customer's image of the room in which the item is to be placed is sent to the merchant, it is possible to manually or automatically calibrate the image size and the virtual room space for the simulation. Consequently, the customer can download a set consisting of a 3D model of the room and the item for trial, with a still image of the room as a background for the display.

For this example, we emphasize the handling of persistent shared objects, instead of the event processing generated by many shared objects, which we can see in chat applications using avatars. Applications that manage many avatars for clients run up against a performance problem related to the processing of events generated whenever the avatar walks around in the virtual world. On the other hand, in our scenario, not many clients are connected simultaneously to the server - just two or three users - and therefore the performance of the event processing is not critical. The functions required for the scenario are:

- To keep the values of shared fields up to date
- To modify the scene graph of shared objects

- To get the same viewpoints

The first requirement concerns the values of fields shared among clients. The document issued in March 1998 did not mention the reliability of the current value of the arbitrary shared field that is a part of the shared object, but only the position and orientation of the object within a zone when the client joins a zone in which several shared objects already exist. Supporting the first requirement necessitates ensuring that the value of the NetworkState node is updated to the current value when the client joins, and that the value is different from the initial value. Merely updating each field of the NetworkState node is not a perfect solution to the problem of sharing the latest scene graph among clients, since some events in the field cause unexpected changes to be made in the scene graph by a program running with a Script node. This first requirement still depends on the implementation of the system, and we are developing the system to make it possible to obtain and share the latest value of the NetworkState node among clients.

The second requirement concerns the editing of a scene graph shared among clients. Supporting the second requirement for the annotating function that we discussed requires at least the capability to add some nodes, such as a Text node or a simple Geometry node, to the scene graph of a shared object. Implementing this capability requires the browser to provide a function similar to a VRML authoring tool, in order to specify the position at which a virtual pointer and attached text annotation are located. It also requires the LivingWorlds module to save the modified scene graph as an output file and to load it for sharing. The latter is necessary because most VRML browsers do not support a function for converting a scene graph into strings representing the scene graph, like the toString() method of JSAL. Since several functions are related to the satisfaction of the requirement - namely, saving the scene graph as a VRML file, uploading it to the server, and distributing it to other clients - we are working on implementing them in our system.

The third requirement concerns the control of the user's viewpoints. This requirement can be satisfied by authored content consisting of standard VRML nodes. If the user chooses the same viewpoint as another party, he/she can get the same view. However, if the user compels the other party on the remote client to change

his/her viewpoint, an additional function is required for the browser to switch viewpoints.

The main theme of this development project was: What is the advantage of this 3D simulation? If we develop the same application by using a panoramic image generation technology such as IBM's PanoramIX [3] or Apple's QuickTimeVR, it would enable the user to manipulate the overlaying item as a 2D image instead of a 3D model. In this case, the system is required to calculate the image size depending on the perspective of the room, and cannot support a function for rotating the item in any direction. If we do the object's sharing by using a window-sharing program such as IBM's ContactFusion or Microsoft's NetMeeting, it would enable the user to share the entire content of a window among clients. Since this function is basically independent of the content displayed within the window, conceptually two users may also share the 3D model provided both of them do not change each other's viewpoint. Since the system we developed in accordance with our scenario uses a 2D still image as a background, we cannot find a clear distinction between the window-sharing approach and our 3D sharing approach. However, if the room is a 3D model instead of a 2D image, users will be able to change their viewpoints as they like. In this case, a simple additional interference check will be also effective; for example, the aperture between the item and the surrounding walls can be checked by means of a function for detecting collisions between the two 3D models.

## 4 Summary

We have described an application of LivingWorlds in the electronic commerce area with an overview of our browser and server. We also described the requirements we had to meet in developing the system. We have completed the development of the second stage of the RVE system, which consists in the implementation of a LivingWorlds client-server system. We hope that multi-user VRML environments will give all Internet users a chance to enjoy using an intuitively intelligible cyberspace.

## Acknowledgments

The authors would like to thank all the reviewers of this paper. We are grateful to Masaaki Taniguchi and Jacob Chen for useful discussions and comments. Many thanks also to the RVE development team, the IBM Taiwan team, and the IBM Software Solution team for their invaluable contributions.

## References

- [1] blaxxun interactive, "blaxxun Reference Sites," <http://www.blaxxun.com/solutions/referencesites/index.html>.
- [2] R. Lea et al., "Community Place: Architecture and Performance," *Proc. Second Symposium on the Virtual Reality Modeling Language*, pp. 41-50, 1997.
- [3] J. Lipscomb et al., "PanoramIX", <http://www.software.ibm.com/net.media/panoramix/index.html>, 1998.
- [4] Mitra et al., "Living Worlds," <http://www.vrml.org/WorkingGroups/living-worlds/>, 1998.
- [5] M. Taniguchi, "Event Processing for Complicated Routes In VRML2.0," *Proc. Third Symposium on the Virtual Reality Modeling Language*, pp. 83-88, 1998.
- [6] VRML Consortium Inc., "The Virtual Reality Modeling Language ISO/IEC 14772-1:1997," <http://www.vrml.org/Specifications/VRML97/>, 1997.
- [7] R. Yoshida and C. F. Greco, "Reactive Virtual Environment System: LivingWorlds multi-user world," *Proc. Virtual Environments 98*, pp.55-1-55-12, 1998.

October 20, 1998

RT0282

Telecommunications 5 pages

# Research Report

RVE System: Case Study of the LivingWorlds System

Ryo Yoshida, T. Murao, and T. Miyazawa

IBM Research, Tokyo Research Laboratory  
IBM Japan, Ltd.  
1623-14 Shimotsuruma, Yamato  
Kanagawa 242-8502, Japan

## Limited Distribution Notice

This report has been submitted for publication outside of IBM and will be probably copyrighted if accepted. It has been issued as a Research Report for early dissemination of its contents. In view of the expected transfer of copyright to an outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or copies of the article legally obtained (for example, by payment of royalties).

