From Proceedings of Role of Universities in the Future Information Society (RUFIS), Arizona, USA, 1999

Distant Learning and Cooperation Using VRML

Jiri Zara
Department of Computer Science & Engineering, Czech Technical University,
Prague, Czech Republic

Abstract
Future information society will widely use Internet-based technologies not only for gathering information but also for educational purposes and for the cooperative work. The paper gives an overview of techniques suitable for these tasks using standardized 3-dimensional virtual environment (VRML) as a framework and a common platform. Theoretical considerations are followed by practical experiments.

Keywords
Distant learning, distant cooperative work, VRML, virtual environment, Java, EAI.

1 Introduction

Currently used methods for education and presentation on Internet are mostly based on two-dimensional information scheme. Users of the web can easily find textual, pictorial, video, and audio information in HTML documents. A relatively new way of presenting information is to incorporate and to utilize three-dimensional data. This allows the VRML (Virtual Reality Modeling Language), an ISO standard [1] for presentation of and interaction with spatial objects in virtual environments.

Virtual worlds described by VRML can be explored using a web browser with VRML plug-in. A user is able to navigate through virtual environment, to manipulate and interact with objects, to change the virtual world. The natural way of work and thinking within the three-dimensional virtual environment is a good condition for extending a single-user experience to multi-user cooperative work. Since VRML browsers do not support multiple users interacting with the same world, an extension has to be developed.

The paper deals with several issues of distant cooperation based on VRML environment. Section 2 contains basic concepts used in distributed systems. Section 3 introduces VRML and its interface EAI [2] and discusses different implementation problems. A sample application is described in Section 4. Results are presented in Section 5, where possible future extensions are also discussed. Section 6 concludes the work.

2 Virtual reality and Internet based cooperation

Virtual Reality Modeling Language (VRML) allows to present three-dimensional objects and to simulate their behavior. Lot of applications can benefit from the VRML features in terms of presenting new models, optional designs, and various objects with different functionality. Since VRML has been also developed as a standard for Internet and WWW, it currently represents the most important way how to deliver the 3rd dimension to a screen of arbitrary user.

Not all users are familiar with virtual walking through spatial scenes or exploring features of virtual objects. While well-known 2D user interfaces based on WIMP (Windows, Icons, Menus, and Pointers) represent a good analogy for a desk and documents on it, a 3D environment requires more experience and certain level of abstraction. This is important especially when using general-purpose computers without any special virtual reality (VR) devices. Most of VRML users have to work in VR just with a mouse and /or keyboard as the only input/interactive device, and a standard flat screen used as a view into the 3D space.

Distant learning systems help to get experience and right habits for working in virtual environment. Distributed educational programs distinguish between a tutor (supervisor) and other users. Typically, a tutor presents different activities in virtual space and users either watch the same on their computers or try to follow individually. Advanced distributed systems allow a cooperation in virtual space – object positioning, setting geometrical and material properties, creating and editing new objects, etc.

Implementations of such systems differ in computing platforms, richness of functions, and usage of standard/special approaches. Highly specialized systems have usually high functionality and allow efficient cooperation and communications among many users. Their use is often limited to specific hardware and software, special network and/or particular VR devices. Military training and simulating systems are examples of such approaches.

Distributed systems for common purposes like education, training, and scientific cooperation do not have too many advanced functions, but emphasize the need of availability, platform and network independence, simplicity and understandability. They use standards for protocols, implementation, and exchange of data. This paper deals with this kind of systems.

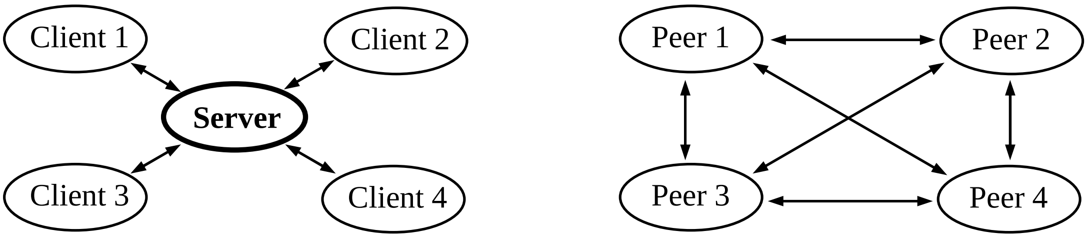


Figure 1: Client/server model (left) and Peer-to-peer model (right)

When speaking on distributed systems, two main approaches can be found: centralized (client/server model) and decentralized (peer-to-peer model), see also Fig. 1. The client/server model is simple to implement. Access to data can be easily managed, since all data are in the central database. Peer-to-peer system is generally more flexible, but the maintenance of consistency is difficult - all participants work on their copy of data and changes have to be properly distributed to other users. Hybrid models are also possible but still not too common. Most of currently available systems use the hierarchical, client/server approach. This is also the case we suggest for simple cooperative work with number of participant limited to about 20.

3 Using standards - VRML and Java

The usage of standards is highly important when running applications on various systems. Widely available Internet connection, platform independent and network oriented Java programming language, standardized VRML file format for description of 3D scenes – all these things form a powerful tool for the implementation of flexible distributed systems. While Java language still does not achieve the level of ISO standard, the VRML is already standardized. Moreover, the specification of programmer's interface between VRML and Java is going to be soon published as the Part 2 of VRML standard. This interface is known as EAI - External Authoring Interface [2]. The EAI is a layer between VRML browser and external programs and utilities. Supported languages are Java and JavaScript (officially ECMAScript). Applications can be designed either as stand-alone programs or as embedded components placed into web browsers. The second approach is more common (see Fig. 2).

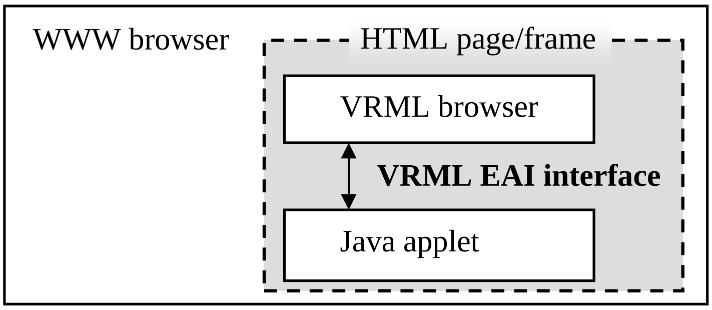


Figure 2: Standard components of collaborative applications based on Internet and VRML

Currently, most systems for distributed work in virtual environment are implemented in Java and VRML EAI. Based on either client/server or peer-to-peer model, they further differ in solving the following typical issues:

- ***Data access***
  The scene has to be consistent, i.e. just one user can change properties of given object at given time. The simple solution is to lock the whole scene for a privileged user (a tutor, an owner of a token), the better one is to lock objects or their parts only. The latter case requires a smart synchronization, update and distribution of changes among participants.
- ***Managing additional information***
  Practically, it is almost impossible to take arbitrary VRML file and put it directly into a collaborative environment. Certain analysis and data preparation is always necessary. While simple application data can be written directly into VRML file, complex information has to be stored in a database [3]. Some systems use their own, non-standardized extensions to VRML and special preprocessing is required before the VRML data are transferred to a viewer [4].
- ***State coding***
  One of the most difficult tasks is to manage the state of the virtual world. The late coming users have to find the world in up-to-date state, thus the history of all changes should be encoded and stored [5]. The interactive parts of virtual worlds cause a lot of problems, since accidental activation of a sensor in a local copy of the world can completely change the future behavior and possible reaction to events coming from the server.
- ***User management, security***
  Some systems are open to any visitor, some require to log in with password. The direct communication among participants can be similar to popular Internet chat, including speech, video and other media [6]. Systems have to manage the situation when a participant

disconnects (either intentionally or by network error), especially when doing changes in virtual worlds. The method, known as 'dead-reckoning' is used for detecting non-active users.

Several other differences can be found when comparing systems for distant cooperation in virtual reality. Last but not least is the use of widely used VRML and WWW browsers. This is the case of the following sample implementation.

4 A simple system for distant VR presentation

In this section, a simple application based on client/server model is described. The aim is to show that the design of such a system is quite straightforward and a good functionality can be achieved when using purely standard tools. The application has been developed as a one-semester student work and it is currently used for educational and experimental purposes.

The following principles have been stressed: usage of standards (common VRML and WWW browsers), minimal preprocessing for arbitrary VRML data (no special VRML extensions), simplicity, and understandability. The final application allows sharing a virtual world by number of participants. One of them has a privilege to be a 'Master' - he or she walks through the world, visits arbitrary places, examines virtual object. The others (from technical reasons called 'Slaves') can automatically follow the Master. The Master can pass on the privilege to another participant and watch his/her work in virtual environment.

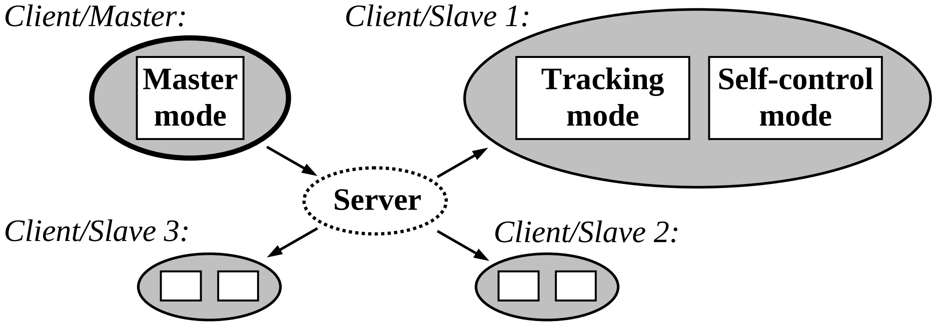
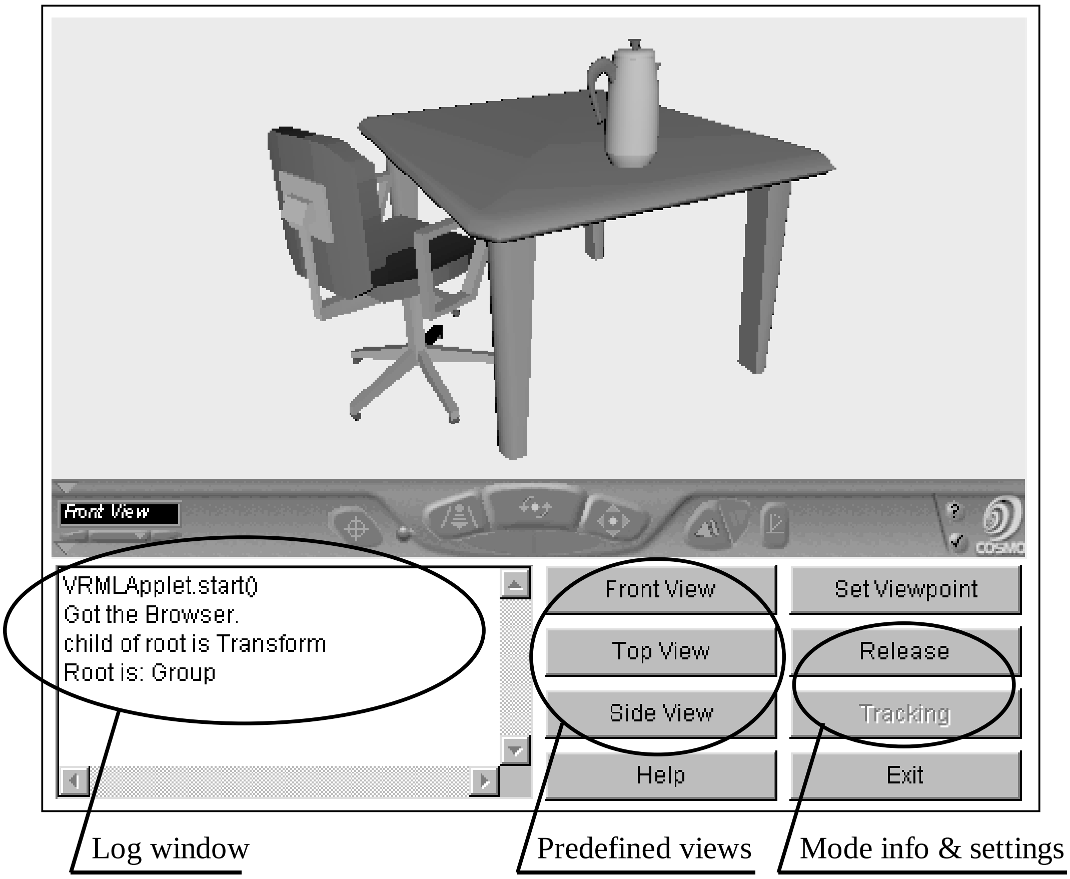


Figure 3: Modes for manipulation with the virtual world

The scheme of the system is shown on Fig. 3. The system consists of two separate Java programs - Server and Client. Server is designed as a stand-alone Java application. Its task is to add/remove users, to transfer data from Master to all Slaves, and to assign the privilege when released by current Master. The Server listens on a predefined socket, known to all Clients in advance.

The Client program is Java applet placed into an HTML page and performed in WWW browser environment. The program works in three modes: Master mode, Tracking mode and Self-control mode. In Master mode, information about Master's activity is immediately sent to the server and then distributed to other Clients. Since Master just examines the world, distributed data contain only Master's position and orientation in virtual space. Due to the simplicity reasons, interactions with the world and modifications of objects are currently not supported.

If the Client works as a Slave, it allows either to watch Master's activities (Tracking Mode) or to examine the virtual world independently (Self-control mode). When switching back to Tracking Mode, the position and orientation in virtual world is immediately updated from the Server.



```
VRMLApplet.start()
Got the Browser.
child of root is Transform
Root is: Group
```

Log window Predefined views Mode info & settings

Figure 4: Simple 3D world presentation (a tutor's screen)

Figure 4 shows the arrangement of a Client window. The upper part contains 3D view into a VRML world with a control panel for manipulation with virtual worlds and objects. Common CosmoPlayer VRML browser has been used in this case. The Java applet with Client implementation is placed below. It communicates with VRML browser via VRML EAI interface as also shown on Fig. 2. The implementation details can be found in [7].

5 Results and possible improvements

We have chosen VRML with EAI interface and Java applets for experiments with cooperative work in a virtual environment. The task was to allow limited interaction of several users with the same 3-dimensional world in the same time. We have designed a scheme "1 plus N" with one supervisor (a tutor) and many co-workers (visitors, students). The role of supervisor can be passed on to arbitrary user in the network (chosen by cyclic shift). The supervisor is able to explore any virtual world, while others observe the results immediately. The application has been successfully tested on various platforms (MS Windows, IRIX), WWW browsers (MS IE, Netscape) and VRML browsers (CosmoPlayer, WorldView) with about 10 participants.

Any VRML file can be put into the system. The requirements for such a file are simple - it has to contain a few VRML nodes with predefined names. This is due to the limitation of VRML EAI interface, where only named nodes are accessible. Although files are currently prepared "by hand" in textual editor, we are going to do that automatically on the server side, when the file first comes to the system.

Further improvements are under development. The main drawback of the current implementation is the absence of sharing interactive events. This will be overcome by searching VRML file for interactive nodes (called 'sensors' in VRML specification). A simple parser can find sources for interaction in a file and to inform both Master and Client about the need of proper events distribution. Another possible improvement is based on filtering data distributed among participants. Since the frequency of sending data depends on the speed of Master's computer, slower Slaves can be overloaded by number of messages. Smart filtering helps to distribute data smoothly and to increase number of Clients. After those extensions, the usage of application will change from ”joint observation” to real ”cooperative editing” of arbitrary virtual environment.

6 Conclusion

The distribution of three-dimensional information will play important role in the future information society. We have shown that valuable program for distant learning and cooperation based on VRML can be implemented using standard tools and environments. Such kind of applications will increase the efficiency of working with spatial data, either for individual or for cooperative purposes.

References

[1] ISO/IEC 14772-1:1997, Information technology - Computer graphics and image processing - The Virtual Reality Modeling Language (VRML) - Part 1: Functional specification and UTF-8 encoding. http://www.vrml.org/Specifications/VRML97/

[2] Marin, Ch.: Proposal for a VRML 2.0 informative annex, External Authoring Interface. Silicon Graphics Inc., 1997. http:/www.cosmosoftware.com/products/player/developer/eai/

[3] Saar, K.: VIRTUS: A Collaborative Multi-User Platform. In Proceedings of VRML 99 – Fourth Symposium on the VRML, Paderborn, Germany, 1999, ACM Press, pp. 141-152.

[4] Carson, J., A., Clark, A., F.: Multicast Shared Virtual Worlds Using VRML97. In Proceedings of VRML 99 – Fourth Symposium on the VRML, Paderborn, Germany, 1999, ACM Press, pp. 133-140.

[5] Mauve, M.: Transparent Access To And Encoding Of VRML State Information. In Proceedings of VRML 99 – Fourth Symposium on the VRML, Paderborn, Germany, 1999, ACM Press, pp.29-38.

[6] Blaxxun Interactive. http://www.blaxxun.com/

[7] Kubec, M., Zara, J.: Remote VRML browser control using EAI. In Proceedings of CESCG 99, April 26-27, Budmerice, Slovakia, 1999, pp. 69-78. http://www.cg.tuwien.ac.at/studentwork/CESCG99/MKubec/

About the author:

Jiri Zara (e-mail: zara@fel.cvut.cz, http://www.cgg.cvut.cz/staff/zara.html) is the associated professor at the Czech Technical University in Prague. His lectures cover Computer Graphics, Virtual Reality, and Multimedia. He also works as the expert for ISO, subcommittees SC24 (Comp. graphics & Image processing) and SC29 (Coding for hypermedia and multimedia). He published several books and lecture notes.