

# Architectural Framework for Immersive Web Based Virtual Environments

Nuha El-Khalili

Faculty of Information Technology, University of Petra, Jordan

**Abstract:** *The development of 3D web based applications has been possible with the establishment of a number of 3D web technologies. Such applications have been utilized in different disciplines such as surgery planning, architectural design and games. However, most of these applications are trapped to the desktop setting provided by default in 3D technologies. Very few attempts have been made to present 3D immersive web based applications and to evaluate the usability of the technologies used and the interaction techniques. In this paper, we will attempt to address these questions by examining different ways of providing an immersive virtual environment on the web.*

**Keywords:** *3D Web applications, VRML, Java 3D, immersive virtual environments.*

*Received September 6, 2006; accepted April 20, 2008*

## 1. Introduction

Research in 3D graphics has achieved great advancements in the virtual reality field; making it possible to create 3D virtual environments that vary from totally immersive to desktop. The benefits of immersive VR applications affected a wide range of disciplines such as Geographical Information System (GIS) [22], medical applications [20], digital art [17] and city and architectural planning applications [20]. These applications utilized full range of specialized input/output devices (e.g., CAVE, projection tables, head and hand tracking, etc.) that gives the user the feeling of being immersed inside the application. However, the term “web based virtual environment” is rather new and is used to refer to 3D graphical applications that use the World Wide Web as an environment to run over. Such applications have the advantage of being ubiquitous (as in [5] which uses PDA devices to access the virtual environment), in addition to the potential of being collaborative or multi-user (e.g., [13], [16]).

3D web based applications have been utilized in different fields such as surgery planning [4, 9], architectural and manufacturing design [2, 11], mathematical visualization [23], environmental behavior simulation [26] and e-learning [25, 13, 14]. However, most of these applications have been limited to standard desktop 2D input/output devices. One reason for this limitation is to ensure the wide accessibility of the application to different users of the web. Another reason is the variations in technologies available to construct such applications. These technologies, e.g., Virtual Reality Modelling Language (VRML), Java 3D and X3D (Extensible 3D), only

support by default the standard mouse, keyboard and monitor interface.

Nevertheless, there are situations where more advanced input/output interfaces are needed, whether to simplify the interaction process- as in visualization applications, or to make it more interesting- as in games, or simply because it is a requirement of the intended application, e.g., in surgery planning applications where you need to use devices more similar to real life surgical instruments.

This paper investigates two technologies used to create 3D web based applications VRML and Java 3D, in order to provide support for other immersive input/output devices in an attempt to achieve an immersive virtual environment on the web.

The next section reviews previous attempts to present web based virtual environments with input/output devices other than the standard ones. Section three presents the two proposed architectures utilising VRML and Java 3D technologies and evaluates them. Section four describes the system implementation that was done to verify the recommended architecture. Section five introduces the procedure designed to evaluate the usability of the interaction technique used in the system and reports the results of this evaluation. Finally, we conclude the paper and look ahead at future enhancements.

## 2. Related Work

Few researches have investigated the addition of other virtual reality input/output devices to 3D web based applications. Most of these attempts presented proprietary solutions that require special software. For instance, in [7] the authors present an immersive

Virtual Reality Aided Design system (VRAD) using VRML. The authors implemented an immersive VRML browser to be able to use special input/output devices like the Head Mounted Display (HMD), magnetic trackers and stereo projection screens. The system was implemented to work under Windows and UNIX.

Another attempt was presented in [21], where the VRML standard was implemented on Vista system, which is a virtual environment display server that supports devices like the HMD, data gloves and 3D tracking devices.

[1] Presented a full featured VR toolkit called Avalon software to create 3D immersive web applications using VRML/X3D standards. Their work extended the definition of sensors provided in the W3C standard to include abstract generic sensors more suitable for 3D immersive input/output devices. The mapping between hardware devices and the abstract application sensors was achieved with a one-time configuration process that stores the mapping details on a file on the client side.

In addition, work done by [19] developed a tool called JINX to develop distributed immersive virtual environments that utilize commodity computer clusters. The input/output devices can be connected to any computer outside the system. They utilize X3D for the development of 3D worlds and XML for the exchange of data between different computers.

Xrooms is another system that provide web based immersive visualization environment. This system utilizes a cluster of PC's to provide stereo projection on several walls. It also supports voice commands and 3D input devices. VRML was used to create the 3D environment. The VRML browser was modified to allow synchronization between clusters of PC's. An Application Program Interface (API) was developed to provide support for 3D devices and to modify the VRML world from Java applets. They also added more fields on some VRML nodes to make them more suitable for stereo viewing [10].

Although all the previous solutions used VRML/X3D to construct their virtual environments, which is the standard file format for representing 3D data on the web, they require special systems or software to use the input/output devices with these environments. Furthermore, extending the VRML/X3D node specification to include new fields (as done in [10]) violates the standard. Thus these solutions can not be generalized to become standard solutions for creating immersive virtual environments on the web.

In [8] the concept of first class browser was presented to extend the VRML capability to support special input/output devices. The idea of a "first class" browser is to implement only core functions of input/output devices inside the VRML browsers, while keeping the appearance and behaviour of devices as part of the scene graph as script nodes. This solution is more

flexible for the virtual environment creators, though, it makes VRML browsers dependent on the evolution of input/output devices utilised in virtual environments.

The it3d toolkit presented in [15] is a library aiming for facilitating the development of 3D virtual environments. One part of this toolkit provides a unified interface for distributed input/output devices. The library implements a unified programming interface for the device drivers using OMG IDL. Because input/output devices can be distributed over the network, the library uses multi-cast peer-to-peer communication between the devices and the application. The application specifies a listener interface in order to receive the device events. The whole library is implemented in Java and Java 3D, which makes it portable. Although the library was not developed specifically for web based virtual environments, it can be very helpful in creating such applications.

This paper looks for ways of providing immersive web based virtual environments using standard 3D web technologies. The first technology is the VRML, which is the ISO standard for creating and exchanging 3D worlds over networks. The second technology is Java 3D, which is a set of classes extending the Java language functionality to handle 3D worlds. Both VRML and Java 3D do not automatically support immersive input/output devices such as head mounted displays, 3D trackers, Data gloves, or other virtual reality devices. Instead the standard mouse/keyboard and monitor are the input/output devices supported by default by these technologies.

In this paper two different architectures will be presented to extend the capabilities of VRML and Java 3D to support immersive virtual environments. The scenarios were implemented and tested using a 5DT- 5 sensor data glove (from 5<sup>th</sup> dimension technologies) and a VFX 3D head mounted display with a 480x480 resolution and a head tracking system (from Interactive Imaging Systems) with a simple virtual environment. Both devices come with a C/C++ APIs to read and control the device. The scenarios proposed in the next section rely mainly or partially on Java, since it provides portable, device independent solutions. This required creating a Java- C/C++ interface for the devices, which was achieved by developing a Java Native Interface (JNI) for each device driver.

In this work interactions with the virtual environment were based on gestures. We reviewed some of the research in the virtual reality literature that used gestures to interact with virtual environments ([12] has a good review of gesture-based interfaces for VR applications). For example work done by [3] used two handed gesture system to control interface widgets. They used video tracking to track the position of the hand, thus they limited the gestures to simplify the video tracking. In this work they allowed one hand

to grab the widget and the other to apply the required gesture on top of the virtual object. These gestures are interpreted differently based on the application.

In the next section we will present the architectures of the two proposed solutions.

### 3. Proposed Architectures

#### 3.1. VRML Solution

The VRML standard has capabilities to represent 3D static models and to add some dynamic behavior to these models. In many simple applications the capabilities of VRML are sufficient to build desktop virtual environments. However, in many cases, these standard capabilities are no longer sufficient. In these cases, VRML has two methods of extending its capabilities with other languages: Scripting and the External Authoring Interface (EAI). Currently both of these methods allow VRML to be extended with Java language. In case of scripting, the Java Script Application Interface (JSAI) can be used to write a Java Script code that is included either directly within the VRML file or a class name is specified in the VRML file, which refers to the compiled Java code [24]. This capability allows more sophisticated web applications. It was used in developing a surgical planning web tool [9] and a web based virtual environment for searching and viewing antiquarian collection [6]. On the other hand, the EAI is an interface between the VRML world and the Java language that allows access to VRML nodes from a Java code that should exist with the VRML world in the same web page. This technology has been utilised in a number of applications such as [18, 9, 7] to achieve multi-user collaboration or native language extensions.

Our first proposed solution is to utilise the EAI to provide access to other input/output devices. In this scenario VRML will be used to develop the virtual world. A Java applet that exists in the same web page will be used to connect the virtual world with the JNI device interface. Communication between the Java applet and the VRML world is done using the EAI. The security restrictions imposed on Java applets prohibit access to native code from applets. To overcome this restriction a Java application is responsible for calling the JNI device interface and sending device events to the Java applet via a socket connection open on the same local machine. Figure 1 shows the architecture of the first scenario. The Java applet pulls device events from the Java application based on a Java timer, which gives the flexibility to the virtual environment to determine the frequency of getting input and sending output depending on the critical needs of the application.

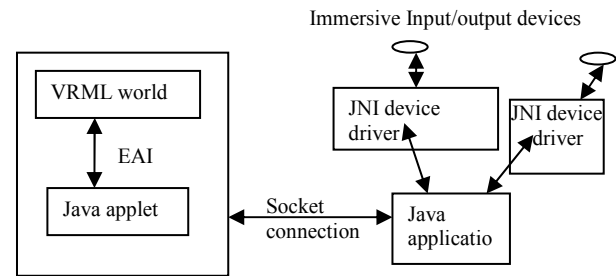


Figure 1. First scenario VRML- EAI extension.

#### 3.2. Java 3D Solution

Java 3D is another technology that has been used to develop web based virtual environments. The Java 3D classes extend the Java language functionality to handle geometry construction and manipulation, appearance definition and animation. The fact that it extends the Java language makes it more flexible and scalable than VRML, since features like network connections, user interface development, portability and other Java features are directly available in Java 3D. However, VRML is more abstract and simpler to develop- being a text file format- than Java 3D. Java 3D has been used to develop web based applications for architectural design and civil engineering project management [11].

Java 3D supports the standard keyboard, mouse and monitor input/output devices. It also provides a number of classes and interfaces to support other input devices (e.g., the input device interface, sensor and sensor read classes). In addition, Java 3D provides a number of utility classes to interact with the virtual universe using standard keyboard and mouse (e.g., key navigation behavior, mouse translate, and mouse rotate). Nevertheless the abstract class (behavior) can be used to implement any customized interaction technique.

In this scenario, the virtual world is created using Java 3D classes. Two classes implementing the input devices interface were implemented for the data glove and the HMD tracking system (data glove interface and HMD interface). In addition, special behavior classes were implemented to track navigation and detect object picking using the data glove (my object behavior, my VP behavior and my grab behavior). Each object in the 3D scene that can be grabbed and moved will be surrounded with a "bounding sphere" of the same size as the object. This sphere will detect that the viewpoint is very close to the object which allows it to be picked. The design of classes in this solution achieved reusability of code for different applications by separating the content of the VR world and the desired behavior of the objects of this world from the input/output devices interfaces. Figure 2 shows the class diagram representing the classes developed in this solution and their relationships.

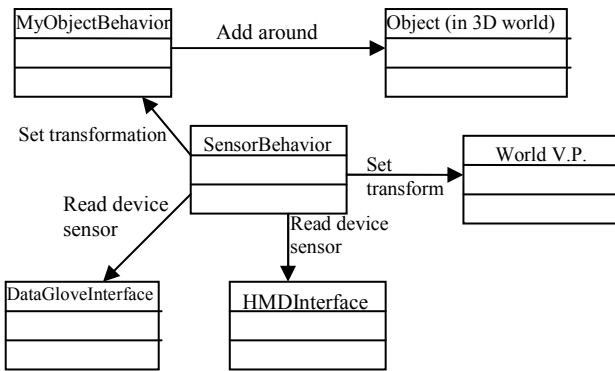


Figure 2. Class diagram for Java 3D solution.

### 3.3. Architecture Evaluation

The previous two scenarios were implemented on PC Pentium III 550 MHz, 128 MB with a special graphics card. Both scenarios achieved the desired objective of interacting with the web based virtual environment using special input/output devices. However, the VRML solution is limited to work on web browsers that support the external author interface APIs (which are Netscape version 3 and 4 and internet explorer version 4). The new Sun Java virtual machines do not support the EAI APIs. Furthermore, a delay in the VRML rendering was noticed between the user interaction and seeing the effect of this interaction on the scene, especially for large scenes. It is anticipated that the delay is due to limited VRML capabilities in updating large worlds in high frequencies as desired by immersive virtual environments input devices. Finally, the fact that the web 3D consortium is currently adopting X3D technology as the new 3D web standard indicates that VRML will soon disappear. These reasons were enough to make us abandon the first proposed architecture.

### 4. System Implementation

The two architectural solutions were implemented on a prototype virtual world, which contains simple geometric objects (e.g., sphere, cone, cylinder and box), in addition to lights and camera (viewer).

Interactions with the virtual world have been divided into two types: navigation and manipulation. Navigation interactions affect the viewer or camera position and orientation. Meanwhile, manipulation interactions allow the user to grab objects, rotate them in two directions (pitch and roll) and release objects. Gestures from the data glove were interpreted as interactions as shown in Table 1. Our gesture recognition algorithm was based on work done by [12], where a threshold value of 0.5 was used to decide whether a figure is flexed or not. Furthermore, pitch and roll values read from the data glove were applied to change the grabbed object orientation.

Finally, three values tracked using the head tracker (pitch, roll and yaw) were interpreted as changes in the camera (viewer) orientation. Because the VRML world can contain more than one viewpoint node, the transformation matrix combined from the three values was applied to the currently active viewpoint node.

Grab-able objects in the virtual world are surrounded with sensors to detect if the viewer is close enough to the object to apply a grab gesture. The colour of the object changes when a grab gesture is successfully applied on the object. This object will then be moved along with the camera (viewer) until it is released. Figure 3 shows some of the gestures used in the system.



Figure 3. Three of the gestures used in the system.

Table 1. Gestures interpretation.

Gesture	Thumb	Index	Middle	Ring	Little
Move forward	Flex	Not	Flex	Flex	Flex
Move backward	Flex	Not	Not	not	Flex
Move right	Not	Not	Not	Flex	Flex
Move left	Not	Not	Flex	Flex	Flex
Move up	Not	Not	Not	not	Flex
Move down	Flex	Not	Not	not	not
Object grab	Flex	Flex	Flex	Flex	Flex
Object release	Not	Flex	Flex	Flex	Flex

### 5. System Evaluation

In order to evaluate the interaction techniques implemented in the system- described in the previous section- and its usability, we performed user hands-on evaluation sessions. Following is a description of the procedure followed during these sessions:

- The purpose of the system and the special input/output devices used are explained to the candidates.
- The candidates are allowed to use the devices freely for 5 minutes to get used to the head mounted display and the data glove.
- The candidates are given another 5 minutes to freely navigate in the virtual world.
- Then they are given a set of tasks to perform in the world (such as finding and moving objects in the world) and they will be timed.
- An observer watches the candidates while performing the tasks and takes notes of any difficulties.

- At the end of the evaluation session, the candidates are asked to answer a questionnaire that evaluates the following aspects of the system:
- Usability of the head tracking feature.
- Usability of the gesture interface using the data glove.
- Usability of manipulating objects in the world using the data glove.
- Interactivity of the system.

Ten candidates from the computer science department participated in the system evaluation. Following are the results of this evaluation:

- As mentioned before, the Java 3D solution was reported to be more interactive and usable than the VRML solution due to the delay in the lateral.
- Some of the candidates suffered fatigue from the HMD as well as being lost in the VR world, which is consistent with other research results in the virtual reality literature.
- All candidates achieved the desired tasks, however they reported difficulty in navigation through the world using gestures (they thought using the mouse is easier).
- Candidates thought that changing the colour of the grabbed objects helped them during their task, because of the difficulty of distance estimation in the VR world.

## 6. Conclusion

In this paper we presented two generic architectural solutions to provide immersive web based virtual reality applications. Both architectures were implemented, tested and achieved the desire goal. However, the Java 3D architecture was found to be more portable than the VRML solution (Java 3D solution is not only platform independent, but also browser independent). We also evaluated the usability of the proposed interaction technique described in section 4. The evaluation procedure and results were described in section 5.

In the future, we would like to investigate the utilisation of X3D technology to achieve the same objective of immersive web based virtual environments.

## References

- [1] Behr J., Dähne P., and Roth M., "Utilizing X3D for Immersive Environments," in *Proceedings of the 9<sup>th</sup> International Conference on 3D Web Technology*, California, pp. 71-78, 2004.
- [2] Beier K., "Web Based Virtual Reality in Design and Manufacturing Applications," in *Proceedings of 1<sup>st</sup> International Euro Conference on Computer Applications and Information Technology in the Maritime Industries*, Germany, pp. 191-194, 2000.
- [3] Boussemart Y. and Rioux F. "A Framework for 3D Visualisation and Manipulation in an Immersive Space Using an Untethered Bimanual Gestural Interface," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, Hong Kong, pp. 162-165, 2004.
- [4] Brodlie K., El-Khalili N., and Li Y., "Using Web Based Computer Graphics to Teach Surgery," *Computer Journal of Computers and Graphics*, vol. 24, no. 4, pp. 157-162, 2000.
- [5] Burigat S. and Chittaro L., "Location Aware Visualization of VRML Models in GPS-Based Mobile Guides," in *Proceedings of the 10<sup>th</sup> International Conference on 3D Web Technology*, Bangor, pp. 57-64, 2005.
- [6] Cubaud P. and Topol A., "A VRML Based User Interface for an Online Digitalized Antiquarian Collection," in *Proceedings of The 6<sup>th</sup> International Conference on 3D Web Technology*, Germany, pp. 51-59, 2001.
- [7] Donath D. and Regenbrecht H., "Using Immersive Virtual Reality Systems for Spatial Design in Architecture," [http://infar.architektur.uni-weimar.de/infar/deu/forschung/public/downloads/avoocad\\_1999\\_using\\_final.pdf](http://infar.architektur.uni-weimar.de/infar/deu/forschung/public/downloads/avoocad_1999_using_final.pdf), 2001.
- [8] Edwards J. and Chris H., "MaPS: Movement and Planning Support for Navigation in an Immersive VRML Browser," [citeseer.ist.psu.edu/92324.html](http://citeseer.ist.psu.edu/92324.html), 1997.
- [9] Holten H., Hvidtfeldt M., Madsen J., and Pedersen S., "VRML Visualization in a Surgery Planning and Diagnostics Application," in *Proceedings of the 5<sup>th</sup> Symposium on Virtual Reality Modelling Language (Web3D-VRML)*, Brazil, pp. 111-118, 2000.
- [10] Isakovic K., Dudziak T., and Köchy K., "X-Rooms: A PC Based Immersive Visualization Environment," in *Proceedings of the International Conference on 3D Web Technology (Web3D'02)*, Arizona, pp. 173-177, 2002.
- [11] Jung T., Gross E., and Yi-Luen D., "Annotating and Sketching On 3D Web Models," in *Proceedings of the 7<sup>th</sup> International Conference on Intelligent User Interfaces*, pp. 95-102, 2002.
- [12] Kavakli M., Taylor M., and Trapeznikov A., "Designing in Virtual Reality (DesIRe): A Gesture Based Interface," in *Proceedings of the 2<sup>nd</sup> International Conference on Digital Interactive Media in Entertainment and Arts Perth (DIMEA '07)*, Western Australia, pp. 131-136, 2007.
- [13] Kirner T. and Kirner C., "Development of a Collaborative Virtual Environment for Educational Applications," in *Proceedings of the*

- 6<sup>th</sup> International Conference on 3D Web Technology, Germany, pp. 61-68, 2001.
- [14] Le Blanc A. and Bunt J., "The Virtual Learning Space: an Interactive 3D Environment," in *Proceedings of the 10<sup>th</sup> International Conference on 3D Web Technology*, Bangor, pp. 93-102, 2005.
- [15] Osawa N., Asai K., and Saito F., "An Interactive Toolkit Library for 3D Applications: it3d," in *Proceedings of the Workshop on Virtual Environments*, USA, pp. 149-157, 2002.
- [16] Presser C., "A Java Web Application for Allowing Multiuser Collaboration and Exploration of Existing VRML Worlds," in *Proceedings of the 10<sup>th</sup> International Conference On 3D Web Technology*, Bangor, pp. 85-92, 2005.
- [17] Ryu C., Cho Y., and Chai Y., "Development of an Immersive VR Display System for 3D Digital Art," *International Conference on Computer Graphics Imaging and Visualization (CGIV'04)*, UK, pp.93-98, 2004.
- [18] Salmela M. and Kyllönen H., "Smart Virtual Prototypes: Distributed 3D Product Simulations for Web Based Environments," in *Proceedings of the 5<sup>th</sup> Symposium on Virtual Reality Modeling Language (Web3D-VRML)*, Canada, pp. 87-94, 2000.
- [19] Soares L. and Zuffo M., "JINX: an X3D Browser for VR Immersive Simulation Based on Clusters of Commodity Computers," in *Proceedings of the 9<sup>th</sup> International Conference On 3D Web Technology*, California, pp. 79-86, 2004.
- [20] Steinicke F., Ropinski T., and Hinrichs K., "A Generic Virtual Reality Software System's Architecture and Application," in *Proceedings of The International Conference on Augmented Tele Existence*, New Zealand, pp. 220-227, 2005.
- [21] Stiles R., Tewari S., and Mehta M., "Adapting VRML 2.0 for Immersive Use," in *Proceedings of the Second Symposium on Virtual Reality Modeling Language*, UK, pp. 75-81, 1997.
- [22] Su S. and Sherman W., and "TAVERNS: Visualization and Manipulation of GIS Data in 3D Large Screen Immersive Environments," in *Proceedings of 16<sup>th</sup> International Conference on Artificial Reality and Telexistence Workshops (ICAT'06)*, pp. 656-661, 2006.
- [23] Wang Q. and Saunders B., "Web Based 3D Visualization in a Digital Library of Mathematical Functions," in *Proceedings of the 10<sup>th</sup> International Conference on 3D Web Technology*, Bangor, pp. 151-157, 2005.
- [24] Web3d Consortium Virtual Reality Modeling Language Specifications, <http://www.web3d.org/x3d/specifications/vrml/>, 2004.
- [25] Wu Y. and Teyi C., "A Web Based Virtual Reality Physics Laboratory," in *Proceedings of the 3<sup>rd</sup> IEEE International Conference on Advanced Learning Technologies*, Canada, pp. 455-455, 2003.
- [26] Yan W. "Integrating Web 2D and 3D Technologies for Architectural Visualization: Applications of SVG and X3D/VRML in Environmental Behavior Simulation," in *Proceedings of the 11<sup>th</sup> International Conference on 3D Web Technology*, Maryland, pp. 37-45, 2006.



**Nuha El-Khalili** is an assistant professor at the Computer Science Department, Faculty of Information Technology at the University of Petra. She teaches software engineering, system analysis and design, and internet programming courses. She received her Bachelor's from American University in Cairo in 1993. She received her Master and PhD from University of Leeds in 1995 and 1999, respectively.

