

A simple event model in Java3D-based VRML browser

S. Kim, D. Kwon, T. Park, B. Choi

Virtual Reality Research Center, Electronics & Telecommunications Research Institute
161, Gajeong-dong, Yuseong-gu, Daejeon, Korea

Abstract

With the growth of the World Wide Web (WWW), the needs of representing 3D world on Internet have increased rapidly. To this VRML97 has played an important role in the growth. The present time, there are a many VRML browser based on OpenGL and Direct3D rendering libraries, and Sun Microsystems has already prototyped a 3D browser. We describe a certain event model is used to implement our Java3D-based VRML browser as network client in our Networked VR project.

Keywords: VRML browser, event model, Java3D, X3D.

1. INTRODUCTION

The growth of the World Wide Web (WWW) has caused an explosion of the needs of representing 3D virtual world in the potential of the Internet. With this bases, VRML was designed to create a more "friendly" environment for the World Wide Web. VRML incorporates 3-D shapes, colors, textures, and sounds to produce a "virtual world" that a user could walk and fly through. VRML is an interpreted language. That is, commands written in text are interpreted by the browser and displayed on the user's monitor. Many of these worlds can be found on the web today. The current specification, VRML97, supports JAVA, JavaScript, sound, animation, the linking of external application using EAI and of files located in other networked server, and an interaction with BIFS (Binary Format for Scenes) node of MPEG-4. It allows the world to be dynamic, always changing. Nowadays, the Web3D consortium makes efforts to promote the VRML 97. To do this, the consortium maintains mailing lists for various issues and runs 4 focused teams for better coordination between the consortium and its members. And under the jurisdiction of one or more of the consortium's teams, there are specific working groups to develop standards and technologies, and so forth. The teams review and vote on proposals provided by working groups, following a well-documented, tried-and-true process. The consortium also seeks to build the web3d market by cooperative development and ISO standardization of relevant technologies for real-time, networked 3D graphics. The consortium is actively engaged in the development of standards technologies with other leading organizations such as the World Wide Web Consortium (W3C) for XML and the Motion Pictures Experts Group (MPEG) for MPEG-4. Therefore, the next generation of VRML standard requires a lot of things including geographical data representation, humanoid animation, multimedia content description, and so on. Using the similarity between Java3D and VRML, a few VRML browsers based on Java3D and VRML authoring tools are developing[1]. In this essay, we describe some data structures and processes of events. The former is used to implement an event model in VRML browser based on Java3D, the latter is needed to manage animations and interactions.

2. RELATED WORKS

2.1 VRML browser

The VRML is not techniques for 3D processing but only a sort of language specification for making mention of shapes of a virtual environment, actions of an object in it and various interactions. In other words, the VRML specification doesn't include the techniques, 3D processing and action control, to display an environment. Therefore all developers can develop VRML browser using creative techniques of their own. For instance, if a certain application can represent accurately the environment expressed with VRML specification of the Web3D consortium using Java3D, we can consider it as VRML browser. Figure 1 shows general information of platforms and rendering libraries of VRML browsers[2]. As you can see in the Figure 1, most of VRML browsers use OpenGL or Direct3D as a rendering library. SUN has already prototyped a VRML 1.0 and 2.0 3D-only browser. This browser has limited browsing functionality but includes the ability to load worlds, navigate, and pick. Figure 2 shows a common architecture of VRML browser[3].

Figure 1: General information of some VRML browsers

Browser	Platforms	Rendering Libraries
CosmoPlayer	Mac, SGI, Win32	OpenGL
blaxxun Contact	Win32	OpenGL
blaxxun CC3D	Win32	Direct3D
OpenWorlds	SGI, Win32, Solaris	Performer, Optimizer, OpenGL, IrisGL
WorldView	Win32, Mac	Direct3D, QuickDraw3D
DpIV	Win32	Direct3D
Cortona	Win32	OpenGL, Direct3D
CASUS Presenter	SGI, Solaris, Win32	Inventor (on OpenGL)
FreeWrl	Linux, Solaris, Digital	OpenGL

2.2 Java3D

The Java3D API was developed by Sun Microsystems and used for writing 3D graphics applications. It gives developers high level constructs for creating and manipulating 3D geometry and

for constructing the structures used in rendering that geometry[4]. Java3D's scene graph-based programming model that is similar to that of VRML provides a simple and flexible mechanism for representing and rendering scenes. And, behavior nodes of the

Java3D are provide the means for animating objects, processing user inputs, reacting to movement, and enabling and processing other events.

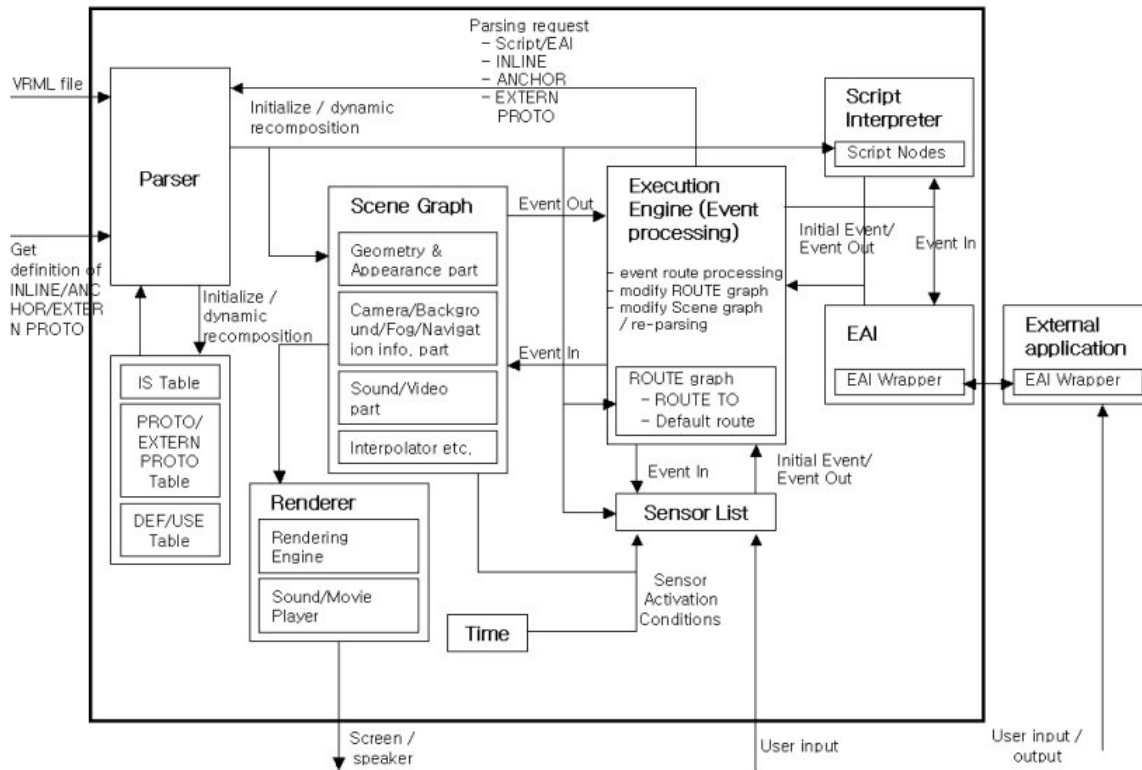


Figure 2: Architecture of VRML browser

3. EVENT PROCESSING

In VRML world, an event means information that should be transferred from one node to the other node in scene graph. For instance, user inputs generated in the outside of scene will be able to offer a source of an animation of some nodes in scene graph. Once events are generated, those are sent to their routed destinations in time order and processed by the receiving node. This processing can change the state of the node, generate additional events, or change the structure of the scene graph. This serial process is called an event routing. Like that, the change of a virtual environment or scene graph is caused by these interactions and animations. To process interactions and animations, VRML provides various interpolator nodes and sensor nodes. Moreover the script nodes of VRML allow developer to define an arbitrary event processing. The ideal event model processes all events instantaneously in the order that they are generated. In practical, the VRML97 specification states the frozen time for an event routing[5]. Figure 3 shows the conceptual execution model of VRML browser[6]. As you can see in the Figure 3, scene graph is composed of geometry node for representing a virtual environment, interpolator nodes, sensor nodes and Script nodes. Among them, only sensor nodes and script nodes can generate an initial event. And then the initial event, other event in-outs are propagated a destination node through the reference of route graph included in an execution engine. When a developer want to

construct this VRML execution model satisfactorily, many functions that are offered by Java3D API can be used readily.

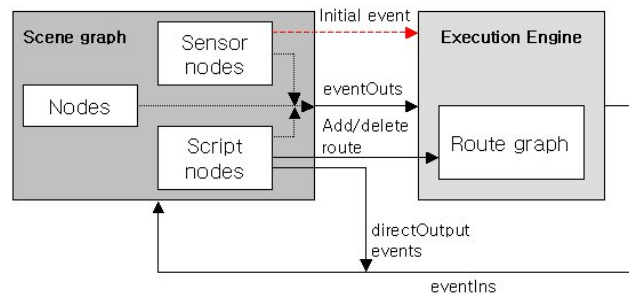


Figure 3: Conceptual Execution model (EM) of VRML browser

Figure 4 shows that classification animation nodes and interaction nodes specified in each VRML97 and Java3D API specification with similar functions. We can access easily to implement various nodes of VRML97 by using this base for Java3D-based VRML browser. In the cases of animations and interactions, many parts of Java3D abilities have a one-to-one correspondence with those of VRML. In practice, all interpolator nodes of VRML can be implemented with interpolator node and pathInterpolator objects of Java3D. In Figure 4, the Sensor object of Java3D provides an

abstract concept of a hardware input device is different from sensor node of VRML. For example, it includes Joystick and HMD.

Figure 4: Nodes related to animation & interaction

	VRML97	Java3D API
Sensor	CylinderSensor	Alpha
	PlaneSensor	Behavior
	ProximitySensor	Pick Utility
	SphereSensor	
	TimeSensor	
	TouchSensor	
	VisibilitySensor	
Interpolator	ColorInterpolator	ColorInterpolator
	OrientationInterpolator	RotationInterpolator
	PositionInterpolator	PositionInterpolator
	ScalarInterpolator	ScaleInterpolator
	CoordinateInterpolator	TransparencyInterpolator
	NormalInterpolator	SwitchValueInterpolator
Etc.	Script	KeyNavigatorBehavior
		Sensor

In java3D-base VRML browser, events can be classified with two types. The first is the event generated by user inputs from external environment. In other words, it is an initial event. We don't need to consider the propagation and management of this event because it is process at Java3D level. The second is a non-initial event that is processed through reference of a route graph. We can make the process to be managed under the event manager. Figure 5 shows a block diagram of event model as stated above. We didn't consider Script node yet.

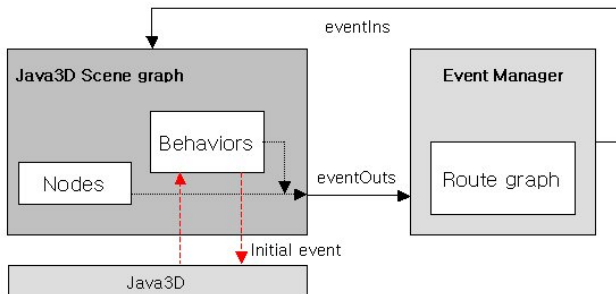


Figure 5: Conceptual EM of our Java3D-base VRML browser

4. CONCLUSION

Figure 6 shows three steps in the animation during one event routing. In this example, PositionInterpolator, ColorInterpolator, ScalarInterpolator and OrientationInterpolator nodes of VRML are processed, and the source of this event routing is a TouchSensor. We used three primitives for simple test. A ScalarInterpolator and an OrientationInterpolator are applied to a cylinder, and a cone

moves along x and y axes as changing color simultaneously. When user touches any primitive among three primitives, the cone starts the movement. A box is increased and decreased in a size automatically by a TimeSensor node of VRML.

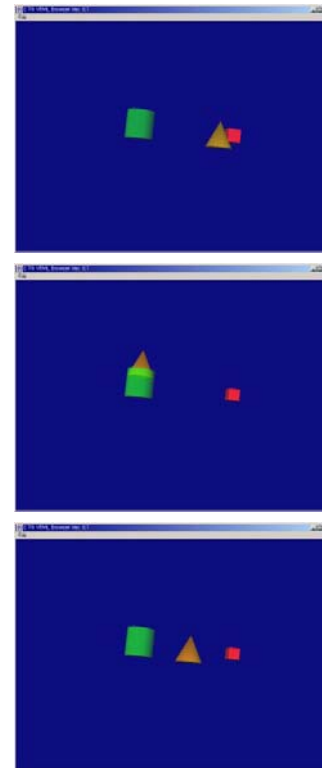


Figure 6: Simple example of animation & interaction

Figure 7 shows a screenshot of our simple java3D-based VRML browser. This VRML model have two types of interpolator node. Like the preceding, we can also get interactive responses for an animation by interpolator nodes and an interaction by user's mouse manipulation.

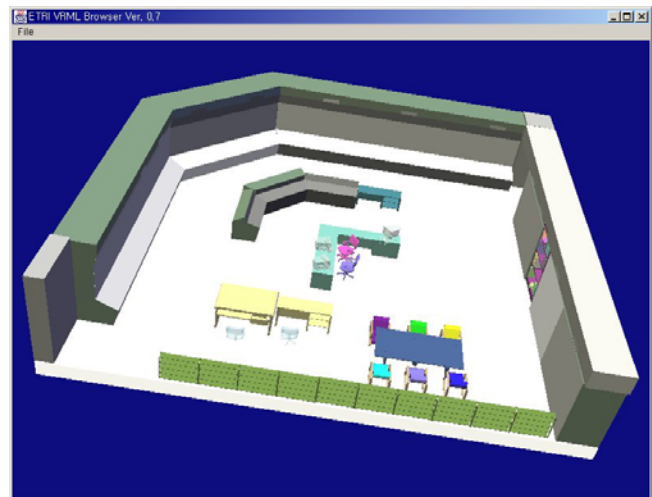


Figure 7: Complex model (GoRi nuclear power plant)

5. REFERENCES

- [1] <http://www.cybergarage.org/top/index.html>
- [2] <http://web3d.about.com/compute/3dgraphics/library/weekly/a070698.htm> (Greg Seidman's VRML97 Browser Datasheet)
- [3] T. J. Park, block diagram of X3D browser, ETRI TM200000815, May, 2000.
- [4] Sun Microsystems, The java3D API Specification, Version1.2, April, 2000
- [5] Rikk Carey, Gavin Bell, The Annotated VRML97 Reference Manual
- [6] Bernie roehl, Justin Couch, Cindy Reed-ballreich, Tim rohaly, and Geoff brown, Late Night VRML2.0 with Java, ZD Press, 1997

About the author

Sungye Kim is a researcher in Virtual Reality Research Center, Electronics & Telecommunications Research Institute in Daejeon, South Korea. She received a M.S in computer science from Chung-Ang University in 2000. Her research interests include global illumination, radiosity, non-photorealistic rendering, Image based rendering and virtual reality.

E-mail: inside@etri.re.kr

Telephone: +82-42-860-5828

Homepage: <http://insideblue.hihome.com>

Postal address: Virtual Reality Research Center, Electronics & Telecommunications Research Institute (ETRI), 161 Gajeong-dong, Yuseong-gu, Daejeon, Korea, 305-350