The implementation of interactive virtual campus based on VRML

Wang Ming
Department of New Media
Zhejiang University of Communication and Media
Hangzhou, China
wangming@zjicm.edu.cn

Sun Zhi Gang
Information Office
Zhejiang University of Communication and Media
Hangzhou, China
sunzhig@zjicm.edu.cn

Huang Chang Fei
Department of Information
Zhejiang University of Communication and Media
Hangzhou, China
huangcf@zjicm.edu.cn

Abstract - **Virtual reality technology is the multimedia technology by computer through people's sense of smell, sight, touch, hearing and other aspects to simulate environment. Virtual reality technology engenders an immersive feeling, through human-computer interaction, to obtain the appropriate information or experience. This paper analyses the virtual campus system under web2.0, and design the virtual campus based on VRML. Through the introduction of virtual reality technology, the paper points out a method to build an interactive web 2D and 3D virtual campus system by VRML language and Java programs. The paper describes the key development technologies, and focuses on the communication of java program and the virtual scene in this virtual campus system.**

Index terms – **VRML, External Authoring Interface, Node, Java**

I. INTRODUCTION

In digital information society, there are three representative techniques which are multimedia, Internet and virtual reality, and those three new technologies have the same intersection that is virtual reality modeling language (Referred to as VRML). VRML is the results that people pursue the development of Internet as a graphic interactive to 3D virtual living space. It is able to provide a more natural experience, including telepresence, interactive, dynamic effects, and the exploration of participants. The use of VRML technology combined with JAVA technology to build a virtual campus system, it can truly realize object-oriented, cross-platform and distributed network browsing. Thus, the interactive of virtual reality scene is an important feature and is a major advantage of the virtual campus by virtual reality technology. The essence of interactive acts is communication between scenes and programs, so the design and implementation of communication in the virtual campus plays a decisive role, and it will be directly related to the success of the virtual campus system.

II. COMMUNICATION OF JAVA AND VRML SCENES

The interaction between VRML scene and user require two basic factors: behavior and execution mode. Behavior describes the events which would take place, and execution mode describes the methods which send back and forth scene entities, behavior changes the state of object in scene through execution mode. In the VRML scene, there are two actions which determine the event's generation; these two actions are static behavior and dynamic behavior. Static behavior is no need to use any program and this kind of behavior is combine movement patterns in scene with new defined node. Because only two nodes through a statement to combine, so it determine that only one motion is possible, this prefabricated motion-mode can not satisfy user's animation and interactivity needs. On the other hand, dynamic behavior is used to determine the event's trigger by a logic program. VRML can not display the logic of scene behavior's change, because it has not any node to support this kind of logic, the scene will be limited to the pre-defined route, passing along the intended route from the previous event. Therefore, the procedures controlled by event can only be outside of the scene.

Interactivity is the biggest improvement on VRML2.0, as support for Java and thus VRML2.0 can produce a new interactive applications, there are two ways for using java in VRML, through internal Script node and external application interface (Referred to as EAI). It not only establishes a standard EAI, and also allows browser development companies to develop EAI, which means VRML can continue to have expansion. Most of VRML browsers support Java.

A. *Communication by script node*

Script node is the bridge connected to VRML and Java program, and its URL field is the key to contact with external programs. URL is an object's complete address on web, it pointed out the location of external program, whether local code or remote procedure in the network can both be used legally. Even the field's value can be directly assigned to a legitimate source program.

Java programs have been able to access the Script node, because vrml.node package, in java, contains a Script class, it is an abstract class. Through inherit Script class; Java programs will be able to establish contact with the Script node, so as to achieve a purpose of controlling other VRML nodes.

VRML and JAVA can communicate which depend on the Script node and Script class. First, define the Script node's field values, then set Route to establish contacts with script nodes and other nodes in VRML. When a VRML node which linked with the script node triggers an event, the event is passed to the Script node's eventIn field by Route, the Script

978-1-4673-0242-5/12/$31.00 ©2012 IEEE

node can find java class by URL field which is specified the corresponding class's address. Be noted that the class must be inherited from Script class. At present time, Java classes start to work, the first executed method is initialize ( ). The initialize ( ) method must obtain eventIn field value of the script node, or return JAVA program's value to eventOut field of the script node. Script class provides three methods which are getField, getEventIn, getEventOut.

getField（”fieldName”） ;

getEventIn（”eventInName”） ;

getEventOut（”eventOutName”） ;

As parameters, fieldName, eventInName, eventOutName must be consistent with the name which defined in script node, corresponding to the subclass inherited from the Field class should be defined in the same type with the script node. This kind of subclass is exists in vrml.filed package, which provides getValue () and setValue () method.

After initialization, processEvent(Eventp0) or processEvents(int p0, Event [ ] p1) method would capture the event passed by the script node. ProcessEvent and processEvents method is a common way to input for all external events. Java programs handle the corresponding event by reload these two methods. In VRML package, it has getName( ) method of Event class, the method returns the captured event name (the event name must be same with eventInName which occurred in the Script node), the events is determined to be captured, then JAVA program would execute appropriate code to handle it . In event class, getValue( ) method returns a instance of ConstField class. ConstField class is a subclass of Field class, because it does not have setValue( ) method, ConstField can only take out eventIn field value of the Script node, and it can not modify the eventIn field value. After processing all events, JAVA program calls eventsProcessed( ) method of Script class, this method is the key to adding animation, interactivity and other effects for users.

Through the Script node, JAVA program can return the processed data to the VRML world. By initialization, getEventOut( ) method get variable value, then call setValue( ) method to pass the value to the Script node's eventOut fields, at last Route would pass the fields value to various nodes in VRML, so the communication process has been completed. The communication of java and the script node has been shown in Figure 1.

1) *Create VRML scene directly by script node and Browser class*

JAVA program can modify VRML nodes through the Script node and Node class, saving computing time and improving generate speed of the VRML scene, but this kind of scheme is only generally modify the value of nodes of each domain, although JAVA program can add and delete specific nodes by setting addChindren and removeChindren value in Group node, but very inconvenient to use, and can not operate Route. If there are too many different and interrelated nodes would be created by JAVA program in VRML, the workload is unimaginable. In order to meet such needs, developer must make some improvements.

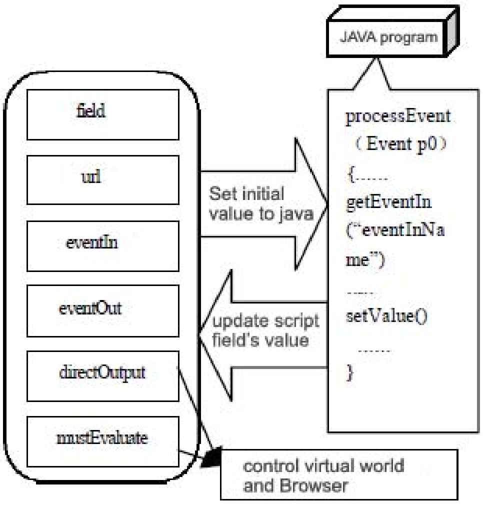
JAVA program
field
url
eventIn
eventOut
directOutput
mustEvaluate
Set initial value to java
update script field's value
processEvent （Event p0）
{......
getEventIn (“eventInNa me”)
......
setValue()
......
}
control virtual world and Browser

Figure 1 script nodes interact with java program

Script class inherits from BaseNode class, and BaseNode class defines two methods: getBrowser( ) and getType( ). getBrowser( ) method will return which browser is BaseNode class in. Browser class exists in vrml package, which provides a serious of methods to control the entire VRML world.

By addRoute (SFNode fromNode, SFString fromEventOut, SFNode toNode, SFString toEventIn) method and deleteRoute (SFNodefromNode, SFString fromEventOut, SFNode toNode, SFStringtoEventIn) method can add and delete Route.

The method named createVrmlFromString(SFString vrmlSyntax) create all the virtual world directly, the node syntax to be created will be written by string type as parameter, Using this method in VRML world to establish the corresponding node; it can achieve the same effect that createVrmlFromURL (MFString url, SFNode node, SFString event) method , there are some difference with the two method, createVrmlFromURL( ) provides parameter of syntax node which is indicated on VRML file, this file is specified by URL field. Browser locates the joining node by URL address; add it to eventIn field of the event parameter which specified by the node's parameter. Moreover, replaceWorld(MFNode nodes) method can be used to replace all VRML world by specified node; but loadURL(MFString url, MFString parameter) method can be used to replace the current virtual world by another VRML world that url specified.

As script class call initialize ( ) method, the returned result is an instance of browser, and processEvent (Event e) method will be executed using the above method to establish, modify VRML scene.

Through the improvement, this scheme can not only reduce the length of VRML language code, so as to java code length. The scheme further increases the speed of generated VRML scene.

B. *Control VRML world by EAI*

Since java does not support multiple inheritance, java program class inherited from Script class can no longer inherit other classes. If developer write a small application (Applet) to control the VRML scene, the used java classes have to inherit from Applet class, and then it is quite complex that by using the above Script class method. In particular, it's not only hard to integrate Applet and VRML into the same HTML page by using Script class, but also code's scalability is poor. The External Authoring Interface (Referred to as EAI) provided from VRML achieve a java applet class to communicate with VRML scene. EAI has defined how external environment through the existing VRML event-driven mode to access the VRML nodes.

EAI is mostly described by Browser class and Node class. In addition, class processing eventIn and eventOut field and class handling all exception also describe EAI. Here's Browser class is different from above mentioned Browser, here's Browser class is encapsulated in vrml.external package, which contains many methods; these methods can implement various functions of EAI.

Not only that, here's Browser class also provides the following three methods: getBrowser(Applet pApplet), getBrowser(Applet pApplet, String frameName, int index) and getNode(String name) method. The first two methods are static methods, JAVA directly call these two methods to get browser instance where Applet class and VRML existed. getNode() method can get node (Node class) instance of VRML world. Here's Node class has some difference from above mentioned Node class, it uses its own getEventIn () and getEventOut () method to get each fields in VRML nodes (Note: There is no getexposedField( ) method because of VRML node's exposedField has implicit event such as set_fieldname and fieldname_changed to control the specific field value, and set_fieldname is eventIn event, and fieldname_changed is eventOut event), then uses the returned instance of event* , the instance has methods such as getValue( ) and setValue( ) to control specific value. EventIn* class only provides setValue() method for setting value; EventOut* class is relatively complex, it not only provides getValue( ) method to obtain the current value, and also provides an interface for monitoring EventOut field which interface name is EventOutObserver, the interface is listening new EventOut event occurred in VRML node at any time, if there is a new event occurs, callback (EventOut value, doubletimeStamp, Object data) methods would respond to events.

Just because vrml.external package provides Browser class and Node class, Script node and script class even route would be useless. Browser class and Node class can directly control all nodes in VRML worlds. However, this method also has its limitations, because parameter used by getBrowser( ) method must be an instance of Applet, and therefore the method can control VRML world only through java applet.

III. SYSTEM CHARACTERISTIC

EAI implement view three-dimensional virtual world in the same HTML page (by VRML language) and implement the display of two-dimensional layout (achieved by the Applet), and the three-dimensional view can display viewer's location of two-dimensional world in real-time. When VRML virtual scene using java to control, the deficiencies existed in VRML can be made a huge improvement, it will not be difficult to achieve interactive three-dimensional scene. The interactive use between node class and EAI achieve roaming campus, ultimately shown in Figure 2, which has the following features:

1) Let user find them location in a virtual campus. In the tour process, three-dimensional structures around the scene, in small map indicates the user's plane location.

2) In the small map, user can intuitively switch viewing. The small map mark the name of location around school, user just click on the small map, virtual world can instantly be shown in the site.

3) On the small map, user can use signs wizards. It means that when user walk around in the virtual campus, just touch certain scene, there would be a small text prompts on the map: either a description of all types of buildings, or an introduction to the school, or recommend a viable road tour, is also possible that prevent user from somewhere not developed or the area under construction.

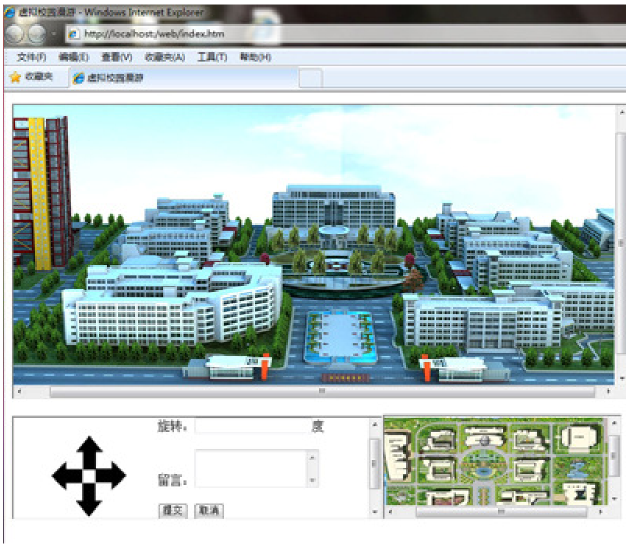

Figure2. Campus roaming renderings

4) Small map has zoom function. In the small map viewer can zoom in and out for the selected local area.

5) User can switch between in the virtual reality world and the traditional two-dimensional webpage. In order to meet user's need, developer use browser / server mode, the browser provides user with an easy to operate and control interface, allow user to browse the campus to express their views; allow user to select their favorite images to complete school building structure and the development of the virtual campus, allow user submit their designed school construction to server. Server-side program has two parts: network part will handle requests from clients, including user's login name and password, implementation, and client communications, upload and download files, modify and save the database. Database part will query user's information, and connect to user databases through JDBC, user authentication. The relationship between each system call has been shows in Figure 3:

JAI (Java Authentic Interface) is a external access interface EAI based on language, JavaScript and Java class access to VRML scene through JAI. When user browses campus, they would feel immersive feeling. As the rapid development of network technology and hardware devices, people's pursuit about image in network is no longer a simply rigid painting; users want to see a perfect independent virtual world on network. The virtual campus system is such a perfect system that has 3D effects, interactive and multimedia.

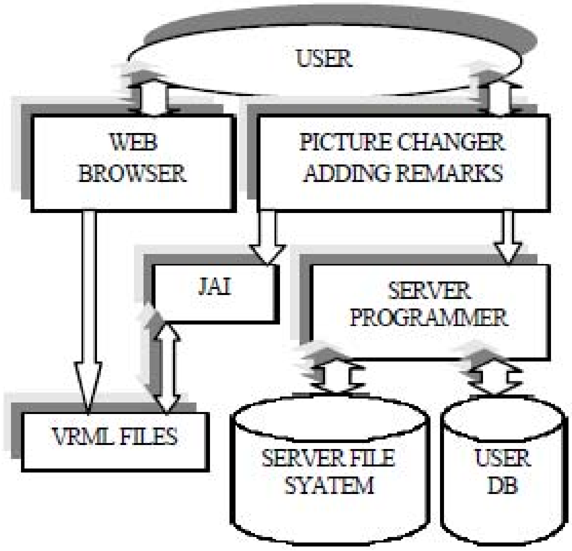

Figure3. Relationship between each system call

IV. CONCLUSIONS

This paper discusses the VRML world and java language; greatly improve VRML virtual world authenticity and interactivity. With the involvement of java program, the system can generate complex VRML worlds in animation, a variety of logic (branching and looping, etc.) scene changes, also make virtual world to respond to keyboard, mouse, windows and other types of events which java language supported. So to build a better virtual world, the communication between java programs and VRML virtual environment is essential.

V. REFERENCES

[1] The Virtual Reality Modeling Language (VRML) [s], ISO/IEC14772-1:1997

[2] Waller John C, Foster Natalie. Training via the web:a virtual instrument. Computers and Education, Vol.35.(2000)161-167

[3] Yeung Kin, Huang jie. Development of a remote-access laboratory:a dc motor control experiment. Computers in Industry,Vol.52(2003)305-311