

# 基于Java的VRML网络通信功能的实现

庞辽军<sup>1</sup>, 李慧贤<sup>2</sup>, 王力<sup>3</sup>, 王育民<sup>1</sup>

(1. 西安电子科技大学综合业务网国家重点实验室, 西安 710071; 2. 西北工业大学计算机学院, 西安 710072;  
3. 西安电子科技大学软件工程研究所, 西安 710071)

**摘要:** VRML 是应用于 Internet 上的虚拟现实建模语言, 本身不具有通信功能。该文扩展了其 Script 节点的功能, 能够支持网络通信, 使得 VRML 节点能够通过网络接收数据并能实时进行场景的渲染。由于 VRML 和 Java 都不依赖于平台, 因此利用该方法搭建的应用系统也具有跨平台性。在不考虑速度的情况下, 它可以将整个 Internet 改造成一个大的多用户应用系统。应用实例说明了该方案的可行性。

**关键词:** VRML; 虚拟现实; 网络通信; 分布式控制

## Implementation of VRML's Network Communication Function Based on Java

PANG Liaojun<sup>1</sup>, LI Huixian<sup>2</sup>, WANG Li<sup>3</sup>, WANG Yumin<sup>1</sup>

(1. State Key Lab. of Integrated Service Networks, Xidian Univ., Xi'an 710071; 2. School of Comput. Sci., Northwestern Polytechnical Univ., Xi'an 710072; 3. Software Engineering Institute, Xidian Univ., Xi'an 710071)

**【Abstract】** VRML is a kind of virtual reality modeling language used on Internet, but it does not support network communication. The function of Script node is extended to support network communication which makes VRML nodes receive data through network and render the scene in real time. VRML and Java are independent of operating system platform, so are the application systems built with the proposed method. Without considering the speed, the method can be used to rebuild the whole Internet network into a large-scale multi-user application system. An application example is implemented to illuminate the feasibility of the scheme.

**【Key words】** VRML; Virtual reality; Network communication; Distributed control

虚拟现实建模语言 (Virtual Reality Modeling Language, VRML) 是一种描述交互式三维媒体的标准文件格式, 它被设计成为与互联网上的三维环境可集成的开放系统, 可以用在 Internet、Intranet 以及本地机系统中, 并且正成为通用的三维图形和多媒体交换格式。VRML 定义了当今三维应用中绝大多数常见概念, 如变换层级、光源、动画、材质属性、雾和纹理映射等。VRML2.0 提供了 54 个功能强大的节点<sup>[1]</sup>, 从而使创建一个三维动态交互环境变得轻而易举。VRML 最吸引人的地方是其互动的场景, 但随着分布式虚拟现实 (DVR)<sup>[2]</sup> 又称网络虚拟现实 (NVR) 的发展和日益广泛的应用, 仅有完美的交互能力是不能满足要求的。DVR 技术是 VR 技术和网络技术相结合的产物, 它建立在先进的网络平台之上。为了使 VRML 满足发展的需要, 必须解决 VRML 的网络通信问题。本文通过扩展 VRML 中 Script 节点的功能, 使其引用的 Java 语言可以通过 TCP/IP 协议接收网络数据, 并用网络数据控制虚拟环境中对象的属性, 从而实现分布交互。

### 1 VRML 的交互和分布式控制

三维虚拟世界的一个重要的内容就是其与浏览者的交互能力, 即用户干预能力。VRML 的交互性涉及到事件体系、传感器和脚本等方面。事件体系使 VRML 的节点可以发送和接收事件; 传感器使用一个定点设备 (如鼠标) 来感知用户的动作并产生相应的事件; 脚本节点可以通过编程使节点产生复杂的动作。这种交互可以增加新节点, 删除已有节点; 控制对象的位移、旋转; 控制对象的视点; 控制对象的颜色;

控制地形地貌; 控制对象的纹理等。VRML 通过设置路由, 设计触发器及编写脚本程序实现交互能力。事件的产生是通过设计触发器来实现的。而产生的事件可以通过 ROUTE 到目的节点而引发新的事件或得到新的位置值、颜色值、视点值, 并把新的数据传递到场景中, 从而引起位置或颜色甚至视点的变化。使用这些技术就可以实现基本的交互功能, 如点击鼠标而引起视点变化、颜色变化、位置变化等。

当需要动态产生一些较复杂的数据时, 可以利用 Script 节点实现。动态行为具有决定逻辑, 可以询问场景的状态, 再根据不同的状态做出不同的决定。VRML2.0 引入 Script 节点, 可以将任意程序逻辑置入场景以及关键点的结点, 产生程序语言以完成这项工作。它将传入的事件自动转交和 Script 结点相关的程序处理, 而结点与程序间的联系是由 URL 字段提供的。URL 字段指向包含了 Script 结点运用的程序文件, 这个文件可以在本地或 Internet 上的任意位置, 可以是 VRMLScript 或 Java 类。通过这种方式, VRML 可以将任意一个代码段捆绑在 Script 结点内, 从而实现动态行为的决定逻辑, 扩展了 VRML 的功能。

**基金项目:** 国家“973”计划基金资助项目 (G1999035805); “十五”军事通信技术预研基金资助项目

**作者简介:** 庞辽军 (1978 -), 男, 博士, 主研方向: 虚拟现实技术, 电子商务安全技术; 李慧贤, 博士后; 王力, 教授; 王育民, 教授、博导

**收稿日期:** 2006-06-25      **E-mail:** lj pang@mail.xidian.edu.cn

虚拟现实是采用以计算机技术为核心的现代高技术生成逼真的视、听、触觉一体化的特定范围的虚拟环境，用户借助必要的装备，以自然的方式与虚拟环境中的对象进行交互、相互影响，从而产生亲临等同真实环境的感受和体验。分布式虚拟现实带来了人机界面、人机交互的新概念和新方法，使得人机交互的内容更加丰富、形象，方式更加自然、和谐。分布式虚拟现实技术推动了虚拟现实技术的变革。然而，若要对VRML场景进行分布式控制，还需要Script节点具有网络通信功能<sup>[3]</sup>。但这是一般脚本语言（如JavaScript、VRMLScript）所不具有的功能。

## 2 VRML 和 Java 的结合

VRML 通过事件通路使相应的节点发生变化，达到简单的动态和交互的效果，为了扩展 VRML 的交互能力和处理能力，提高其分布式控制能力需要程序语言的介入。在网络环境下，Java 是最具魅力的语言，与 VRML 结合得相当自然，当前的主流 VRML 浏览器也都对 Java 编程提供了支持。在 VRML 中使用 Java 有 2 种方式，即通过内部 Script 节点( Script Authoring Interface ,SAI)和外部编程接口( External Authoring Interface , EAI)。

(1)内部 Script 节点。在 VRML 中有一个 Script 节点，它拥有域、eventIn 事件、eventOut 事件。其本身没有任何动作，但可以通过程序脚本赋予脚本节点的动作。Script 节点和利用 Java 或 JavaScript 语言写成的程序脚本，提供了对 VRML 的交互性的强有力的扩充。通过 Script 节点的 URL 域可以包含一段 JavaScript 程序或者是链接的一个外部 Java 的类文件。在 Script 节点中引用外部的 Java 的类文件，通常只是实现一些特殊的、描述性语言不能实现的功能，如通信功能，或者是为了源程序保密。

(2)外部编程接口 EAI。Script 节点从 VRML 场景内部提供了与 Java 的连接，而 EAI 定义了与外部 HTML 页面中的 Applet 通信的接口。它的基本思想是将 JavaApplet 与 VRML 置于同一 Web 页面，在 JavaApplet 中建立 Browser 对象以标识一个唯一的 VRML 场景，并由此获得对 VRML 场景中已定义节点的引用，它通过直接指定引用节点的域值达到动态效果。

这 2 种方式各有所长<sup>[4]</sup>。使用 EAI 可以一目了然地在 JavaApplet 中输入各种参数以改变场景的内容，控制灵活；SAI 方式不需要有 JavaApplet 的存在，只要为场景中的物体写相应的脚本代码即可，可以实现许多 VRML 规范不能提供的复杂逻辑显示。由于 EAI 方式不是真正意义上的分布式控制，因此本文主要介绍第 1 种方式。

## 3 VRML 与外界网络通信的 Java 实现

利用 Script 节点的特点，用 Java 编写成脚本语言，并将 Java 的网络功能和文件访问功能引入到脚本语言中，从而能够实现 VRML 的分布式交互仿真。图 1 给出了一个有网络功能的 VRML 的实现模型。后面将介绍如何利用 Java 实现 VRML 的通信功能。

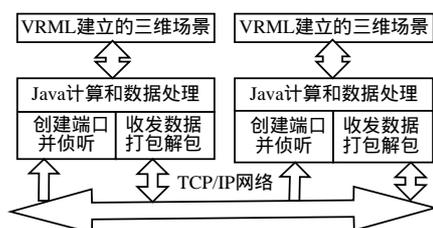


图 2 具有通信功能的 VRML 实现模型

### 3.1 Java 程序的定义

VRML 中使用的 Java 文件，必须满足 3 个条件<sup>[5,6]</sup>：(1) 包含一个名字和主文件名完全相同的类定义；(2) 属于 Script 类 (vrml.node 包) 的子类；(3) 将其说明为“public”类。其定义如下：

```

public class className extends Script {
    ...
    initialize();
    processEvents();
    processEvent();
    eventsProcessed();
    shutdown();
    ...
}
  
```

初始化函数 initialize() 在 VRML 文件加载时运行，将 Script 节点中的事件入口 (eventIn) 和事件出口 (eventOut) 转变成 Java 语言中相应的变量类型。事件入口或事件出口的处理由此变成了对这些变量的处理。

用户收到一系列事件时，调用 processEvents() 函数进行处理。processEvent() 用于处理某一事件，事件处理完毕调用 eventsProcessed()。在函数 processEvent() 中处理事件入口 socketPort (须在 VRML 文件中定义)，该事件入口表示 Socket 通信的端口号。获得这个事件后，函数便生成一个新线程，新线程提供了端口号为 socketPort 的 Socket 服务。

shutdown() 函数在删除 Script 节点时调用，用以关闭 Socket 连接。

### 3.2 Script 节点的域和事件的获取

在 Java 程序中，通过调用 getField() 函数获得 Script 节点中的域对象；调用 getEventOut() 函数获得事件出口 (出事件)；调用 getEventIn() 函数获得事件入口 (入事件)。

### 3.3 读域和事件

在 Java 程序中通过调用 getValue() 函数将 VRML 的域类型转化成相应的 Java 平台的数据类型并返回其值。

### 3.4 写域和事件

在 Java 程序中，通过 setValue() 函数将 Java 平台的变量 value 转化成 VRML 类型的数据并拷贝到目标对象，可用于设置事件出口；通过 addValue() 函数将 Java 平台的变量 value 转化成 VRML 类型的数据并添加到目标对象中；通过 insertValue() 函数将 Java 平台的变量 value 转化成 VRML 类型的数据并在指定位置插入；通过 clear() 函数删除目标对象中的所有元素，通过 delete() 函数删除目标对象中索引值为 index 的元素。

### 3.5 VRML 网络通信的 Java 实现

(1) 在 Script 的事件入口处理函数 processEvent() 中，增加了启动一条新线程的语句：

```

processEvent (Event event) {
    ...
    Thread thread;
    ...
    thread = new Thread(this);
    thread.start();
    ...
}
  
```

(2) 在线程执行函数 run() 中，加入了提供 socket 服务器服务的语句：

```

public void run(){
    Socket socket;
  
```

```

...
ServerSocket server = null;
...
while(true){
    server = new ServerSocket(port);
    socket = server.accept();
    ... }
}

```

当 VRML 场景接收到事件入口 socketPort 时, Script 节点通过事件处理函数起动了新的线程。在该线程中,程序运行了 socket 服务器。socket 服务器循环感应网络连接请求,当有请求到来时,socket 服务器进行连接,由此在 VRML 场景与外界间就形成一条通路,使 VRML 场景可以通过这条路径与外界进行通信。外界来的数据通过这条通路送到 VRML 场景中,在 Script 节点处产生新的事件,经过路由又送到了 VRML 场景中的其它节点中。

值得注意的是,这时 VRML 场景的通信功能还不完善,不具有主动请求连接功能。然而,只须在 Script 的事件入口处理函数 processEvent()中加入提供 socket 客户服务的语句“Socket client=new Socket (Host, Port);”,就可以和外界服务器建立连接。这里的 Host 和 Port 是服务器的 IP 地址和侦听端口。至此,VRML 场景的通信功能就完善了,其体系结构图 2 所示。

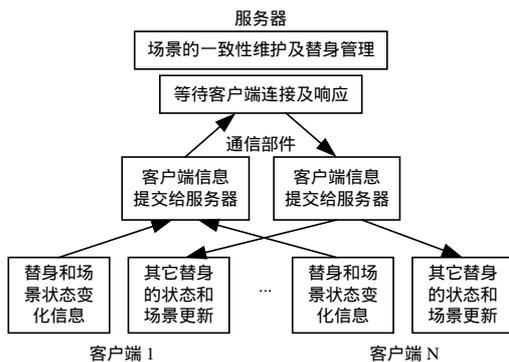


图 2 VRML 场景体系结构

#### 4 应用实例

图 3 是一个虚拟酒吧演示实例<sup>[4]</sup>。



图 3 虚拟酒吧实例

在系统中,Web 客户端是一个 VRML 场景,场景中的桌子、椅子、柜子、杯子和电灯等以及窗外的风景、楼房、树木对象都是由 VRML 语言编写制作成的,下方是一个控制面板,可以执行用户的命令。VRML 场景的通信功能就是利用上述扩展的 Script 节点实现的。

#### 5 结论

使用 Java 语言对 VRML 中 Script 节点功能进行扩展,实现了 VRML 的网络通信功能,使得对 VRML 场景中对象的分布式控制成为可能。由于具有了网络通信功能和分布式控制功能,因此 VRML 场景不再局限于单机环境,拓宽了 VRML 在分布式交互仿真环境下的应用。

#### 参考文献

- Ames A L. VRML 资源手册[M]. 宗志方,译. 北京: 电子工业出版社, 1998.
- Schmalstieg D. Demand-driven Geometry Transmission for Distributed Virtual Environments[C]//Proc. of Virtual Environment'97. 1997.
- Raposo A B. Working with Remote VRML Scenes through Low-bandwidth Connections[C]//Proc. of VRML Workshop, Brazil. 1997.
- 庞辽军. 多用户共享虚拟环境应用平台[D]. 西安: 西安电子科技大学, 2003-03.
- 黄铁军, 柳 健. VRML 国际标准与应用指南[M]. 北京: 电子工业出版社, 1999.
- 冯 武. 仿真可视化通用平台的研究[D]. 西安: 西安电子科技大学, 2000-03.

(上接第 43 页)

#### 3 结语

利用本体对上下文感知系统进行构建有着其独特的优势: 因为基于本体, 不同的开发者提供的上下文感知组件可以达到共享和互用。另外, 利用本体建立的各种智能模块能够高效地进行上下文推理, 从而动态地跟踪实体变化, 适应不同的环境, 提供不同的服务。

但是, 由于本体的概念源于哲学, 随着发展, 它还涉及到计算机科学、语言学、逻辑学等多门学科。因此, 在运用本体进行上下文建模时, 不可避免地遇到上述各门学科研究领域的一些问题, 有的问题本身还处于探讨之中, 为我们的研究增加了相当的困难。

#### 参考文献

- Nixon P, Dobson S, Terzis S. Architectural Implications for Context-adaptive Smart Spaces[C]//Proceedings of the International Workshop on Networked Appliances. IEEE Press, 2003: 156-161.
- Chen Harry, Finin T, Joshi A. Intelligent Agents Meet the Semantic Web in Smart Spaces[J]. IEEE Internet Computing, 2004, 8(6): 122.
- Helal S. Programming Pervasive Spaces[J]. IEEE Pervasive Computing Magazine, 2005, 4(1).
- McGuinness D L, Fikes R, Hendler J, et al. DAML+OIL: An Ontology Language for the Semantic Web[J]. IEEE Intelligent Systems, 2002, 17(4).
- 何海芸, 袁春风. 基于 Ontology 的领域知识构建技术综述[J]. 计算机应用研究, 2005, 22(3): 14-18.