

# VRML与外界数据交互的技术实现

王 硕 (中国人民解放军 37216 部队 82 分队 266102)

周昕宇 (中国人民解放军第四〇九医院 266100)

**摘要:**本文详细讨论了 VRML 技术与其他数据访问技术相结合,实现对数据库实时交互的技术实现方法,并简要阐述了相关技术规范的语法结构和技术要求。所用技术手段安全可靠,具有良好的实际应用表现,便于系统移植。

**关键词:**VRML 数据库 交互 javascript ASP

VRML(Virtual Reality Modeling Language)技术规范是由 SGI 公司提出并经国际标准化组织 ISO 管理的 ISO/IEC 14772 标准进行全面定义的一种网络环境下描述虚拟三维模型和提供交互操作的跨平台开放式解释性文本语言。它提供了完善的对象结构模型和良好的人机交互界面,可以在网络环境中实现三维信息的即时传输、显示和远程访问许可等远程实时控制功能,因此在大跨度低带宽异构网络环境下的分布式三维动态演示和状态分析系统中得到广泛的应用。

由于 VRML 数据规范要求用户在用户终端对源代码副本进行基于本地资源的逐行解释和三维数据的即时写屏,导致客户端工作负担重,显示效率低,对控制命令的响应速度慢。同时 VRML 虚拟空间中的各节点(实质上是带有各自属性和事件接口的对象模型)都通过软接口连接来实现状态交互,数据规格要求苛刻,接口扩展性差,不利于数据的大规模采集和运算处理。

而在结构分析和虚拟演示系统中要求模型能高效的响应用户操作并且可以与各种信息源进行数据交互。针对 VRML 技术规范逐行解释和即时写屏的特点,在 VRML 虚拟空间中数据传输量必须尽可能小,对信息的处理运算工作量要尽可能少,而把这些工作放在其他事务处理过程中。本文就 VRML 空间通过和其他网络数据访问技术的混合编程实现与外界信息源的数据交互提出具体技术方法,仅供参考。

## 一、VRML 空间中各节点间数据交互的方法

VRML 技术规范将物理模型和环境因素都定义成节点,实际上是将其赋予各自不同的属性和独立的触发事件并建立统一的结构架构的对象集合。这些对象有严

格的数据规格要求和统一的数据接口,在 VRML 空间中都是独立存在的和可感受用户操作的。可见的物理模型还可以与其他类型的可见节点相互嵌套,但不可见节点不能包含可见节点。

VRML 技术规范中定义了两大类节点:一类是专门感应用户三维操作的传感器节点(sensor node),这类节点在虚拟空间中是不可见的;另一类是可以显示在三维虚拟空间中的物理模型节点(shape node)。此外还定义了多个特殊的功能节点,如下面将要提到的 script 节点和 inline 节点等。

传感器节点在 VRML 虚拟空间中是不可见的,但可以感应到用户指点设备的二维操作(位移、点击、拖拉等)并自动加入用户操作所涉及虚拟模型在 VRML 空间中相对于用户的透视关系,从而判断出用户操作的三维坐标变换。如果用户使用三维手套、立体鼠标等三维指点工具,传感器节点就可以直接感应用户的三维操作如拉近、拉远、三维滚转等。传感器节点还可以根据用户的操作改变自身的属性或设置变量,这些属性和变量可以在 VRML 空内部各节点的接口域之间传递并能与外界数据源交互。

物理模型节点在虚拟空间中的三维位置关系是基于空间的世界坐标来确定的,并且是可以显示的。它有统一的数据传输接口和完善的属性集合和事件集合,可以接收传感器节点发送过来的数值并根据所对应的接口属性域设置提取数据、改变自身的属性值、触发事件进而改变自己在空间中的位置、外形和显示效果,还可以将处理后的数据送到输出域传递给其他模型节点。

各节点通过接口域的相互路由(route)实现数据传输,路由的映射关系自由,但接口域的数据规格必须严格一致。

## 二、VRML 与外界数据源交互的方法

VRML 技术规范中大部分节点的数据规格特殊,难以扩展,但 script 节点拥有独立的 url 域,可以与其他脚本语言实现混合编程,例如可以利用 javascript 语言规范中的对象模型提取外界数据并传输到虚拟空间中,由此可用多种方法实现 VRML 空间与外界信息源的交互。

### 1. script 节点的语法要求

script 节点是一个特殊的可编程节点,它不可直接感应用户的操作,在三维空间中也不可见,但可以嵌套为模型节点的下级子节点并可与任意节点进行接口连接。其语法及数据规格如下。

```
Script {
    url [] # 字符变量
    eventIn eventInType eventInname # 输入接口设置
    eventOut eventOutType eventOutname # 输出接口设置
    mustEvaluate boolean # 布尔变量
    directOutput boolean # 布尔变量 }
    接口允许的数据规范及缺省接口名如表 1 所示。
```

表 1

输出 (eventout)	输入(eventin)
SFBool Active	SFBool set enabled
Mfcolor eyeColor changed	Mfcolor set skyColor
Sffloat stockIndexes changed	Sffloat set height
Sfimage panorama changed	Sfimage set texture
Sfint32 age changed	Sfint32 set population
Mfnode shape changed	Mfnode addchildren
SFRotation Orientation changed	SFRotation set orbitAngle
Sfstring foatFamily changed	Sfstring set candyFlavor
SFTime creationTime	SFTime set stopTime
Mfvec2f point changed	Mfvec2f set point
Mfvec3f coordinate changed	Mfvec3f set position

在 url 域中允许用户编制对虚拟模型进行本地操作的脚本。VRML 技术规范支持 java 和 javascript 编程语言,但必须在脚本命令行的最前面向 VRML 解释器声明脚本所使用的语言。EventIn 域定义了触发器的输入接口和初始状态,一般以“set-”开头。节点可以通过该域感受到用户的空间操作并将信息送入封装在 url 域中的脚本中进行处理。

EventOut 域定义了触发器的输出接口和初始状态,一般以“-changed”结尾。节点可以通过该域将本地脚本的运行结果和操作信息记录在自定义的变量中并向其他节点发送。

在编制用户自定义脚本和设定虚拟模型的传感事件时,script 节点的输入、输出域和其他感受器节点的输入、

输出域的数据类型必须严格匹配。另外 script 节点还包含 mustEvaluate 域和 directOutput 域,分别用来设置 VRML 空间在接收到外界输入的数据时是否立即调用程序脚本进行本地化处理以及是否允许脚本直接控制其他节点的空间状态。

通过与其他传感器进行接口连接,Script 节点建立起了 VRML 空间与外界信息交互的有效通道并丰富了虚拟模型对用户操作的处理能力,使虚拟空间具有更好的自主判断能力和更多的感知能力。

### 2. VRML 通过内嵌 javascript 脚本语言实现对数据库的信息交互

VRML 技术规范本身没有定义数据库节点,因此必须利用封装在 script 节点中 url 域内的 javascript 语言脚本与外界交互信息,实际上是通过 javascript 规范中的数据库对象与数据库相连。

(1) javascript 规范中的数据库对象事件简介。javascript 语言规范中定义了专门的数据库对象用于对数据库进行远程操作,数据库对象具有完善的事件结构并支持现时被广泛采用的 SQL 数据库架构,其内置方法和相应语法解释如表 2。

表 2

数据库对象的内置方法	语法解释
DataBase. connect (DataBaseType, servername, username, Password, DataBaseName)	连接数据库并指定连接设备和用户信息
DataBase. execute(SQLstatement)	SQL 数据库语言操作
DataBase. cursor(SQLstatement)	对检索数据格式化控制和检索多行数据
DataBase. SQLtable	检索数据库
DataBase. beginTransaction()	打开对数据库的事务处理
DataBase. commitTransaction()	永久修改数据并结束事务
DataBase. rollbackTransaction()	取消当前事务所做的数据修改
DataBase. majorErrorCode()	从数据库服务器返回严重错误代码
DataBase. majorErrorMessage()	从数据库服务器返回严重错误信息字符串
DataBase. minorErrorCode()	从数据库服务器返回小错误代码
DataBase. minorErrorMessage()	从数据库服务器返回小错误信息字符串

JavaScript 语言编制的脚本可以在通过 DataBase. SQLtable 和 DataBase. consor 事件对数据库操作后自动得到的数据改变为自身支持的类型,从而可以在本次的系统调用周期中自由使用这些数据,当写数据库时又会自动改写数据为数据库支持的格式。数据类型对照如表 3。

表 3

数据库中的数据类型	相应的 Javascript 规范所支持的数据类型
String	String object
Data	Data
Time	Data object
null	null

(2)VRML 与 javascript 混合编程实现数据库连接的方法。在 script 节点的 url 域中声明并插入 javascript 脚本。在脚本中利用数据库对象与数据库连接并操作数据库,将得到的数据封装为 VRML 规范支持的数据格式,再由 script 节点传输到其他节点的输入接口域上。程序片段如下:

```
DEF demo script {
  # 声明所使用脚本语言的种类
  url"javascript:
  # 操作数据库
  this.database.connect ( databasetype, servername, user-
  name, password, databasename )
  # 检索数据并记录
  temp = this.database.execute ( search * from
  database while ..... )
  # 封装数据并送数据输出接口
  stockindexes-changed = temp"
}
```

由此其他节点就可以直接调用接口域的变量。

### 3. VRML 通过与 ASP 混合编程实现对数据库的信息交互

当数据库本体不允许直接访问时,VRML 就需要通过中间数据通道与数据库连接。这个中间数据通道应该能够将数据记录在内存变量中并且可以将用户输入的更新数据存储在内变量中送到 VRML 空间中即时显示,还可将数据库中相应的数据更新。比较成熟的创建数据通道的方法就是利用 ASP 技术。

(1)ASP 技术简介。ASP(ActiveX Server Page)技术实质上是服务器端的 ActiveX 技术,是 MicroSoft 公司近年推出的面向对象的网络数据服务技术。ASP 命令先在服务端解释执行,再将执行结果下载到在客户端运行的网络浏览器上或内嵌网络浏览器的本地应用系统中。它运行在服务端,用户看不到 ASP 源代码,无法干预程序的正常运行,从而保证了服务端程序的安全,同时它在网

络中传输的只是执行结果,对网络的数据传输带宽没有过高要求。封装有 ASP 指令的页面必须以 HTTP 方式向服务端程序提交访问申请,否则服务器将因无法确定正在进行申请的客户端的具体逻辑位置而拒绝提供客户端副本和操作结果。

作为面向对象的网络应用技术,ASP 拥有六种内建对象,以此来完成对远程数据的全部操作。在这里只介绍可记录所提取数据的 Session 对象和 Request 对象。

Session 对象,用于存储一个特定的用户将要使用的对象或各种标识数据结构的中间变量。它不允许存储 ASP 内建对象。

Response 对象,代表当前的 ASP 所得到的服务端响应。它可以存储服务端程序对具体某个用户发出的访问申请所作出的响应信息。

(2)VRML 与 ASP 混合编程实现数据库连接的方法。ASP 首先与数据库连接并提取数据,将数据转换为 javascript 语言规范支持的数据规格并存储。VRML 再利用内嵌在 script 节点 url 域中的 javascript 语句将数据封装为接口的标准格式并送到输出接口域传递到其他接口中。程序片段如下:

ASP 程序片段:

```
<% Set Connectionname = Server.Createobject
("adodb.connection"); 建立数据库对象
Connectionname.open "DataBasename"; 连接数据库
sql="select * from DataBasename where.....";
设定对数据库的操作
Set temp = Connectionname.execute(sql); 提取并
记录数据
%>
```

在 script 节点中的程序片段:  
url "javascript:

stockindexes-changed = temp"; 封装数据为 VRML 规范中数据接口支持的数据规格

### 4. VRML 通过内嵌于 javascript 语言脚本实现与数据库的交互

如果要同时提供与用户交互的 HTML 界面和显示三维虚拟空间的 VRML 界面同时要求应用系统优先满足数据远程传输时的安全、准确并对远程控制指令能够及时响应,就必须将应用的界面设计为多帧模式。由于 VRML 解释器不能与 HTML 解释器同时有效写屏效率又远比 HTML 解释器来得低,因此应将虚拟空间界面帧

中的 VRML 解释器内嵌在 HTML 解释器中。

(1)有关语句规则解释。writeIn()语句把正文字符串或 HTML 表达式写到指定的窗口中(缺省为当前窗口),后面那跟一个换行符。它必须和 document.open 方法和 document.close 方法一起使用。

document.open(mimetype)方法支持各种在现行网络环境中得到广泛支持的 mime 类型的可选变元,如:文本(text、html、wrl),图象(gif、jpeg、x-bitmap),插件(plugin、applet)等,缺省类型为 html 格式。在被调用时在相应的语法解释器中打开流(stream),各种解释器就可以接收 write()/writeIn()语句的输出数据并显示结果。

document.close(mimetype)方法则用于关闭文档,如果流用于创建 HTML 布局,close 方法将迫使流的内容显示并终止语法解释器。

(2)javascript 脚本语言内嵌 VRML 实现数据库连接的方法。首先将 VRML 语句用 writeIn 语法封装为标准 javascript 语句,并将其写入页面脚本。这样在终端运行的应用系统会先调用 HTML 解释器对脚本进行高效的文本解释,再根据 writeIn 语句中声明的 mime 类型调用 VRML 解释器,将封装的 VRML 语句提取出来还原为标准 VRML 语言并写屏显示。这样在提取 VRML 语句时就可以将数据库中的数据存储在局部变量中直接传到 VRML 解释器中与 VRML 语句一同解释、写屏。具体与数据库的连接方法则可以根据实际情况灵活使用前面提到的两种方法。程序片段如下:

设把要改变的 VRML 模型代码标记为变量 model.

VRML 文件内容

Javascript 语句内容

```
# vrm1 v2.0 utf8
```

```
shape{
```

```
appearance Appearance{
```

```
material Material{ } }
```

```
geometry box{ } }
```

```
WriteIn"# vrm1 v2.0 utf8"
```

```
WriteIn"shape{"
```

```
WriteIn"appearance Appearance{"
```

```
WriteIn"material Material{ } }"
```

```
WriteIn"geometry"+model+" } }"
```

这样,VRML 空间中的模型就可以根据 model 变量内容的不同而改变形状、颜色甚至属性。

### 三、小结

上述三种方法在技术上都是可行的,但在网络环境下的实际应用中三者的执行效率是不相同的。

VRML 利用内嵌 javascript 语言连接数据库时,命令行需要逐行解释和处理,系统的整体运行速度慢,执行效率低,同时因为是直接操作数据库本体,外界数据源的安全性难以保证。但所有的数据操作都集成在系统的本地页面中,便于维护和修改。

VRML 通过与 ASP 混合编程连接数据库时,VRML 程序、ASP 程序和数据库经 Web 服务器隔离,相互间可以独立配置、运行和管理。系统的灵活性和可扩展性好,适合于分布式系统应用。缺点是数据格式多,格式间经常需要相互转换,因此数据接口设计困难。

VRML 通过内嵌于 javascript 语句连接数据库时,用户可以按照操作习惯同时从多个界面发出指令,人机交互性好,但因为经过封装的语句需要进行两次解释,因此系统的显示效率低,对用户操作的响应速度慢。

随着网络通信技术的高速发展,数据访问技术规范化的扩充和标准化,以及三维图形加速卡写屏效率的进一步提高,VRML 空间将更加丰富多彩,显示效果将更好。立足于分布式数据访问架构的 VRML 技术与其他数据访问技术相结合建立的三维信息访问和处理系统,将越来越多的应用于宇航研究、机械设计分析、军事模拟等各种虚拟环境的演示和技术探讨中。

### 参考文献

- [1] 《VRML 资源手册》(美) Andrea L. AMES 等著 电子工业出版社 1998 年版
- [2] 《Visual Interdev 6.0 开发指南》(中)任伟等著 北京航空航天大学出版社 1999 年版
- [3] 《MicroSoft SQL 6.5 管理员手册》(美) MicroSoft 公司著 科学出版社 & 龙门书局 1997 年版

(来稿时间:1999年9月)