多用户VRML虚拟环境设计

华中理工大学计算机学院　周英飚　冯玉才

本文介绍基于Web服务器—浏览器结构的多用户虚拟环境系统。该系统具有投资低、可移植性高的优点。

VRML简介

VRML1.0草案[1]于1995年正式发布。VRML与超文本置标语言HTML十分相似，都是一由普通文本构成的类编程语言。VRML 1.0定义了若干类型的节点以描述虚拟环境中的实体造型。每个节点包含了若干属性用以描述实体的大小、颜色、材质纹理等信息。VRML 1.0还提供了LOD(Level of Detail)节点，用来建立细节层次随观察视点与实体之间相互距离变化而变化的实体，以提高真实感和优化VRML浏览器的性能。通过WWW inline节点可以将一个VRML文件描述的场景引入到另一个场景之中。而WWW Anchor节点可以在场景之间建立超链接关系，这也是VRML 1.0所支持的唯一交互手段。

VRML1.0虽然能够描述三维虚拟场景，但这种场景是静态的，用户同三维环境之间唯一的交互手段是通过超链接。1995年发布的VRML2.0的草案以SGI的Moving Worlds方案为基础[2]，它扩展了静态的场景描述，提供了描述可交互的动态三维场景的能力。此外，VRML2.0还改善了VRML 1.0的视觉效果。

VRML2.0的缺陷

虽然VRML 2.0可以进行动态的三维虚拟环境的场景描述，但是因为目前VRML2.0标准尚不支持多用户环境所必需的共享行为和环境状态的持续性。其实现的只是一种单用户的环境。

虚拟环境研究的目的是提供用户之间信息交换和共享，因此必须使用户认识到各自与环境的交互作用，并处理好感受的一致性问题，虚拟环境的每个用户对于环境中各个实体的行为和用户交互都应有着一致的感觉，这样才能对彼此的意图和工作有明确的了解，以相互配合进行工作。一般地，用户交互触发消息，消息传送引起相联系实体的行为，所以感知环境的一致性的实质是消息处理的一致性，目前VRML并未提供这一功能。

在目前VRML的实现中，每当用户进入一个虚拟环境时，都会发现所有的实体均位于其设计的初始状态。例如：当某个人在一个区域内推动桌子后，离开区域，再次进入这个区域，发现桌子还在原来的位置。这种持续型的缺乏是因为目前客户与VRML服务器的信息交换是单向的，不能主动修改服务器上的内容。

虽然VRML目前尚缺乏建立多用户虚拟环境的基本支持，但它提供强大的Java脚本语言能力，利用Java内建的Internet网络支持。下面我们探讨一下在Internet上使用VRML建立多用户虚拟环境的基本系统结构。

多用户VRML环境

VRML环境与HTML一样，是依赖于Web服务器－浏览器体系的。用户浏览环境的客户端工具是使用支持VRML的Web浏览器，如CosmoPlayer。下面我们以此结构为基础加入消息管理的功能。与Web服务器－浏览器体系相对应，基于客户－服务器模型的虚拟环境消息管理系统分为两个模块：消息服务器模块和消息客户模块。所有环境的用户的客户程序都与一个消息服务器模块相联系。服务器进程不仅负责用户的加入，退出的管理，更重

要的是保证用户环境感受的一致性。当用户与环境进行交互时，所引起的环境实体的行为不是立即表现出来，而是将消息发送给服务器，消息服务器模块按照某种策略（如先来先处理或按时间戳）决定每一条交互消息的全局处理顺序，分发给所有用户，同时根据交互消息修改Web服务器上的VRML数据以满足持续性的要求。消息客户模块的任务是接收服务器分发的消息，收到后依次进行处理，改变实体的属性以产生相应的行为。同时监控用户的交互，向服务器发送消息[3]。

在VRML环境中，用户与环境的交互都是通过各种传感器节点来实现的。传感器节点是事件发生的源泉，它检测用户与虚拟环境的交互，产生相应的事件，然后通过定义的路径进行传播。改变节点的属性，产生行为，实现用户交互的目标。因此，消息客户模块必须监测传感器节点的事件输出，故在每个区域的VRML描述中，设计一个Java脚本程序，通过路径，将传感器的事件输出连接到其事件输入，并在其事件处理程序中解析出各种传感器得到的交互，然后向消息服务器模块发送消息。并监测服务器模块分发来的消息，在通过路径转发给其他节点，从而保证消息处理的一致性。

VRML的Java脚本程序由Script节点控制，当事件输入到Script节点时，Script类中的事件响应函数processEven得到触发，对于每个事件输入可以使用Event类的成员函数来获取它的名称，以决定相应的处理过程。消息客户模块的结构如下：

```
DEF Client Script
{  EventIn In1
   ……
   EventOut Out1
   ……
   url "client class"
}
# 联结本区域中所有传感器的eventOut
ROUTE  Sensor1 To Client.In1
…
…
# 将一致性排序后的事件传给其他节点
ROUTE  Client.Out1 To Node
…
…
```

Client.java程序为：

```
Import vrml.*;
Import vrml field.*;
Import vrml node.*;
Class Client extends Script
{  public void initialize()
   {  建立与消息服务器模块的Socket连接;
      创建连接上的输入流和输出流;
      使用输出流向消息服务器发JoinRequest消息;
      从输入流上读取为自己分配的ID号，包括区域代
      号和子ID号;
      进行客户端初始化;
      创建一个监听线程ClientReceive;
   }
   public void shutdown ( )
   {  执行客户端清理工作;
      建立与消息服务器模块的Socket连接;
      创建连接上的输出流;
      使用输出流向消息服务器发ExitRequest消息;
      删除监听进程;
   }
   public void processEvent (Event e)
   {  String eventName=e.getName( );   // 获取
事件输入的名称
      if (eventName.euuals("Node1")      // 处理事件1
      {  eventInstance=e.getValue( );   // 获取事件
实例
         value=eventInstance.getValue( ); // 获取
事件实例的值
         使用ClientSend向服务器发送一条消息;
      }
      else if (eventName.euuals("Node2") // 处理事件2
      {  ……
      }
   }
```

整个系统的工作流程图如图1所示：当用户下载了一个区域的VRML数据后，其所附带的脚本Client中的Initialize过程首先得到执行，完成客户端初始化任务。当退出区域时，脚本中的Shutdown过程被调用，完成退出区域的处理。当区域内的一个传感器节点检测到用户的交互时，通过路径将事件发送给processEvent过程，由其调用ClientSend向服务器发送消息。

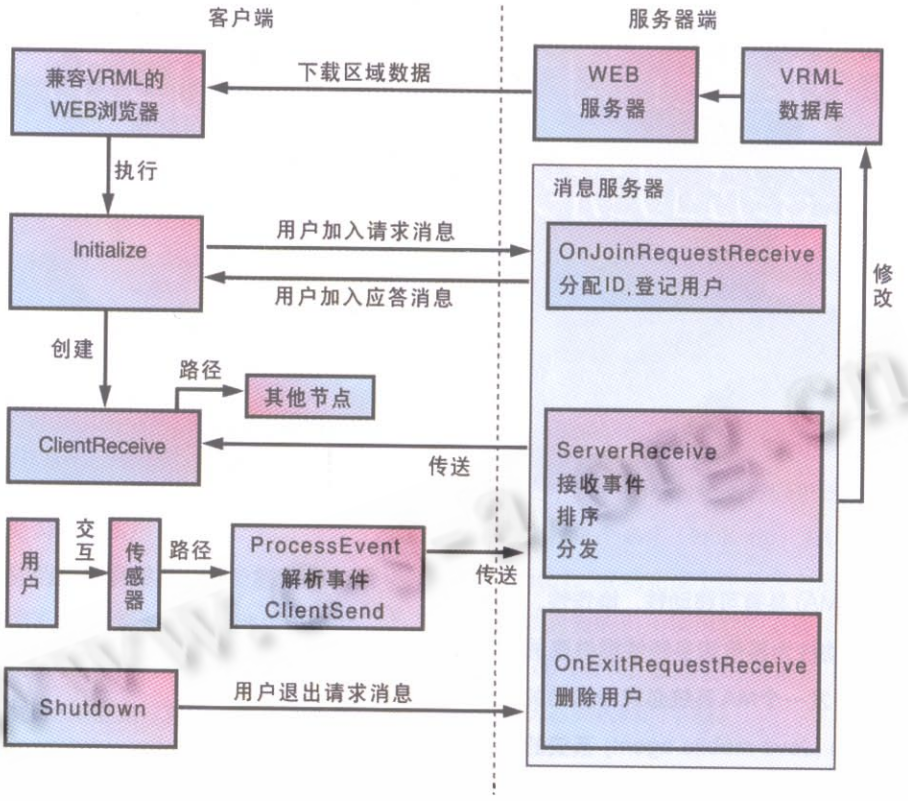

图 1 基于Internet的虚拟环境系统

客户端监听线程ClientReceive的作用是：接收服务器转发来的交互消息，按服务器决定的全局处理顺序进行处理，通过路径进行传播，改变实体的属性以产生相应的行为。

在服务器端，消息服务器模块包括ServerReceive，OnJoinRequestReceive，OnExitRequestReceive等子模块。其中，ServerReceive子模块的任务是：接收客户传送来的交互消息，决定交互消息的全局处理顺序，分发给环境的所有用户，根据交互消息修改Web服务器上的VRML数据。OnJoinRequestReceive子模块的任务是：处理用户申请加入环境的请求，登记用户。OnExitRequestReceive子模块的任务是：处理用户申请退出环境的请求，删除用户。

小结

在本文中，我们分析了基于Internet的Web体系服务器—浏览器上的多用户VRML虚拟环境系统，它具有的优点是：客户端使用通用的浏览器界面，用户易接受，培训费用低。采用的VRML和Java都是具有平台无关性的语言，因此系统的可移植性很好，特别是客户端可以运行在各种软／硬件平台上。■

参考文献

(1) Bell,G and Parisi,A, The virtual Reality Modeling Language Version 1.0 Specification, Online Document, http://vrml.wired.com/vrml.tech/

(2) The Virtual Reality Modeling Language, Introduction, Version 2.0, Online Document, http://siisg1.epfl.ch/VRML2.0/spec/part1/introduction.html

(3) 周英飚 冯玉才，基于客户—服务器的共享白板一致性算法，计算机研究与发展，1998(12)，1112-1117