

多用户共享虚拟环境中 VRML 系统模型^①

许爱军¹ 张文金² 黄正午² (1.广州铁路职业技术学院 信息工程系 广东 广州 510430;

2.广州铁路职业技术学院 教育技术中心 广东 广州 510430)

摘要: 提出一个多用户共享虚拟环境中,基于VRML的虚拟现实协同系统模型。采用B/S结构,设计了模型的工作流程,重点对虚拟场景与用户替身、VRML场景通讯、模型的协同并行控制、模型中数据交互等关键技术进行了描述和实现。该模型能够实现多用户共享虚拟环境空间的一致性、网络通讯的及时性和多用户替身行为的真实性。原型系统的实现验证了模型的可行性。

关键词: 分布式虚拟环境 虚拟现实建模语言 系统框架 关键技术 协同系统

A Model for VRML-Based Cooperative System in Multi-User Shared Virtual Environment

XU Ai-Jun¹, ZHANG Wen-Jin², HUANG Zheng-Wu² (1. Department of Information Engineering of Guangzhou Institute of Railway Technology, GuangZhou 510430, China; 2. Education Center of Guangzhou Institute of Railway Technology, GuangZhou 510430, China)

Abstract: This paper puts forward a model for VRML-based cooperative system in multi-user shared virtual environment. The model adopts B/S structure, designs the workflow of model, and lays emphases on the key technology for virtual scene and avatar, VRML scene communication, the model's cooperative and parallel control and the database interconnection. It can realize the consistency of scene, the timeliness of network communication and the authenticity of avatar in multi-user shared virtual environment. Its feasibility is validated through the implementation of a prototype system.

Keywords: distributed virtual environment; VRML; system framework; key technology; cooperative system

分布式虚拟环境(Distributed Virtual Environment, DVE)是虚拟现实技术与计算机网络技术相结合的产物,它建立在先进的网络平台上,具有分布性、交互性、逼真性、场景一致性和可伸缩性等特点。虚拟现实建模语言(Virtual Reality Modeling Language, VRML)作为一种在网上描述交互三维实体和虚拟世界的标准文件格式,本身不支持多用户环境所必须的共享行为和状态连续性^[1],但它提供了与Java结合的接口,使其引用的Java语言通过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,从而为多用户共享虚拟环境中协同系统的实现提供了可行的

解决方案。本文采用B/S模式,提出一个系统模型,设计了工作流程,并对主要关键技术进行了描述与实现,最后给出了原型系统的实现效果。

1 多用户共享虚拟环境中VRML系统模型

1.1 系统模型

模型结构采用主流的B/S三层架构。其中,服务器负责管理和维护场景的一致性、用户的注册和用户替身的管理,同时监听客户端的连接请求和响应连接,必要时服务器可能需要与数据库进行交互。客户端负责场景的显示和更新,同时感知其它用户的状态。客户端和服务器的连接由通讯软件层实现,主要任务

① 基金项目:广东省教育厅“十一五”规划课题(08JT063)

收稿时间:2009-04-16

是将客户端场景和替身状态的变化信息及时送到服务器端，同时将服务器管理的场景数据和所有替身的状态信息通知到其他客户端。系统模型如图 1 所示。

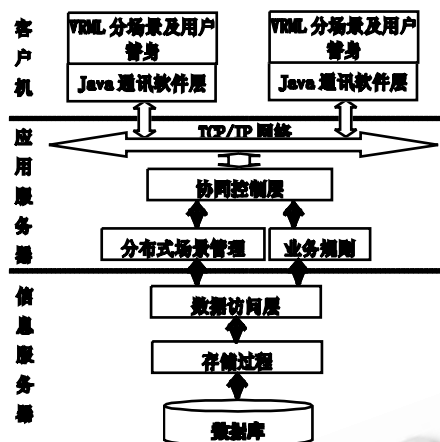


图 1 多用户共享虚拟环境中 VRML 系统模型

1.2 工作流程

客户端的主要工作是监听与接收服务器端的数据，维护客户端的操作，并及时将客户端的数据提交到服务器进行处理。当客户端登录服务器时，执行 Script 类的 initialize 方法，建立与服务器端的连接，创建监听服务器以及客户操作线程，并加载 VRML 分场景到客户端。此后，客户端的场景和用户替身状态的变化信息通过通讯软件层传送到服务器。同时，客户端也不断监听并接收服务器端传来的消息，通过路由将消息发送到本地场景节点，保证客户端场景空间的一致性。当客户端注销时，调用 shutdown 方法删除监听服务器以及客户操作的线程，撤销与服务器端的连接。

服务器端的主要工作是根据每个客户端的请求类型，对 VRML 场景中的物体或物体的属性进行查询、修改，并提供维护共享空间场景一致性的运行机制。实现方法是当客户端与场景进行交互以后，如果改变了场景内容、属性，甚至需要刷新其他客户端的场景，此时并不急于将交互的结果表现出来，而是交由服务器进行处理。服务器首先比较客户端的操作权限、实体的属性、处理的优先级等策略，并将这个事件插入到相应的处理优先队列中。服务器不断处理队列中的事件，只有当事件成功处理完毕后才修改 VRML 场景数据库，同时将更新信息分发给所有客户端。如果事件由于某种原因执行失败，服务器将向提交这个事件的客户端发送出错提

示信息。模型的工作流程如图 2 所示。

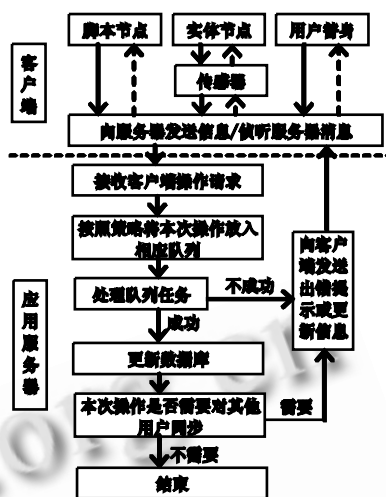


图 2 多用户共享虚拟环境中 VRML 系统工作流程

2 关键技术及其实现

2.1 虚拟场景与用户替身

建立虚拟场景时主要考虑 2 个因素，即如何模拟场景的真实感、如何生成数据量小且适于网络传输的模型。前者涉及 VRML 的建模方式，后者涉及 VRML 文件的优化。

VRML 有 2 种建模方式。一种是利用其自身的几何造型语句，用基本的几何实体及其布尔运算进行几何造型，对于一些复杂曲面可能要用坐标点方式拟合。这种造型方式的优点是文件小，但方法较为烦琐，一般只适合建立简单规则的模型；另一种是借助第三方软件。流行的三维造型开发工具，如 3DS MAX、CAD、Maya、LightScape 等都支持 VRML 格式的输出，可以很方便地导出 VRML 模型数据。这种造型方式的突出优点是便于创建比较复杂且不规则的模型，但可能会造成 VRML 文件过于庞大，不利于网络传输。

对虚拟场景文件进行优化是必要的^[2]。VRML 支持多种优化技术，从减少模型面片数方面考虑，常用的方法有预测计算、场景分块、可见消隐、细节层次模型(LOD)等。从优化描述模型矩阵、减少矩阵元的数目方面考虑，可采用相关算法实现。近几年提出的多边形网格简化算法^[3]，如 Schroeder 的顶点删除法、Hinker 的合并共面多边形法、顶点簇法、Hammann 的三角形删除法等，都能有效降低场景的复杂度，加速分布式虚拟场景的传输。

用户在分布式场景中的存在感通过标识用户身份

的 3D 替身(Avatar)来实现,替身是用户形象和行为在虚拟环境中的表现,也是用户体验虚拟环境沉浸感的依据。替身的三维模型可用 3D 软件创建。本文提出的系统模型中,服务器对用户替身进行统一认证与管理,用户替身的登录、退出、动作行为等信息由通讯软件层送到服务器,在行为逻辑所定义的业务规则下,对该用户替身状态进行更新,并决定是否以广播的方式通知其他客户端。这种方式有效解决了用户替身之间的可视化与协同感知。

2.2 VRML 场景通讯

VRML 场景之间的通讯由通讯软件层实现。VRML 不是一种编程语言,不支持多用户环境所必须的共享行为和状态环境的连续性,其网络通讯能力和分布式控制能力可以借助与 Java 语言的结合来实现。结合的方式有 2 种,即通过内部 Script 节点(Script Authoring Interface, SAI)和外部编程接口(External Authoring Interface, EAI)。这 2 种方式各有所长^[4]。使用 EAI 可以在 JavaApplet 中输入各种参数以改变场景的内容,控制灵活,但它不是真正意义上的分布式控制;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。利用 Script 节点的这一特点,可用 Java 编写成脚本语言,将 Java 的网络功能和文件访问功能引入到虚拟场景中,以实现 VRML 的交互式仿真。Java 通讯软件层的实现模型如图 3 所示。

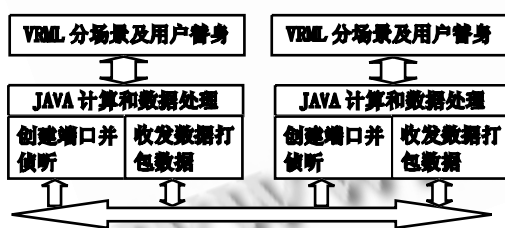


图 3 通讯软件层的实现模型

在此模型下,Java 通讯软件层的实现首先进行 2 个预处理,即在 Script 的事件入口处理函数 processEvent()中,增加起一条新线程的语句;在线程执行函数 run()中,加入提供 socket 服务器服务的语句。这样,当 VRML 分场景接收到事件入口 socketPort 时,Script 节点通过事件处理函数起一个新的线程。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,这样在 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讯。同样,外界来的数据也通过这条通路送到 VRML 场景中,在 Script 节点处产生新的事件,经过路由又送回 VRML 场景中的其它节点中。为了实现客户端的主动请求连接功能^[5],可在 Script 的事件入口处理函数 processEvent()中加入提供 socket 客户端服务的语句“Socket client=new Socket(Host, Port);”,就可以主动与外界服务器建立连接。至此,VRML 场景之间的底层通讯就畅通了。

2.3 模型的协同并行控制

协同控制层要解决的一个主要问题是避免多个用户同时对某个对象操作而引发冲突^[6]。可采取 2 种方法进行处理。一种是优先级控制方法,将客户端用户按照不同的优先级存入数据库中,可以允许多个用户进入同一场景,当两个以上用户操作场景中同一个物体时,根据优先级规则所定义的触发顺序,决定哪个用户对该物体进行有效操作。另一种是对象锁方法,当用户要对某个物体进行操作时,先向服务器发送对该物体操作的请求,若该物体未被其它用户操作,服务器则向用户发出可操作响应,同时屏蔽其它用户的操作请求。一个用户只有对该物体解锁后,才可以提出另一个锁定请求。其方法是服务器对 VRML 场景节点的(ExposedField SFBool)touchsensoreabled 域值的控制来实现的。这两种方法都能有效进行冲突检测,具体采取哪种控制策略,可依据分布式环境的复杂程度和协同工作的任务要求来确定。

2.4 与数据库交互

在生成客户端 VRML 分场景时,分布式场景管理层需要读取场景物体的几何参数数据、属性数据、替身的位置信息、用户登录信息;用户在虚拟场景中进行操作时,需要通知应用服务器更新有关信息,这些操作都需要与数据库进行交互。在 B/S 模式下,客户端并不直接与数据库交互,客户端场景的变化和用户替身的状态更新信息都由 Java 软件通讯层发送到服务器端,由服务器统一控制数据的存取。

服务器与数据库的通讯技术有很多,基于 Java 语言的有 JDBC(Java Database Connectivity)技术。JDBC 实现了 Java 程序与数据库系统的无缝连接,是一个支持基本 SQL 功能的通用底层 API,在不同的数据库功能模块层次上提供了一个统一的用户接口,为

Java 程序员提供了独立于特定的数据库系统而实现与网络数据库互连的方式。

在服务器端, VRML 还可采用 EAI 与 SAI 方式与数据库进行交互^[7]。采用 EAI 方式时, Java Applet 可根据需要随时利用 JDBC 与数据库建立连接, 获取物体的有关数据, 然后利用 EAI 的 `creatVrmlFromString()` 方法将物体以 `IndexedFaceSet` 节点形式写入 VRML 场景。采用 SAI 方式时, 可以在 `Script` 节点中用 `JavaScript` 直接编写连接数据库的程序, 也可以用 `JAVA` 编写连接数据库的程序, 经过编译后形成 `*.class` 文件, 再在 `URL` 中指定 `JAVA` 程序的地址。这样, 传感器节通过 `Route` 命令启动 `Script` 节点就可实现与数据库的交互。

3 应用实例

我们在校园网分布式虚拟环境下, 实现了一个多用户共同组装电脑的原型系统。用户登录虚拟装机实验系统后, 搬动电脑配件安装到主板的相应位置。实验场景与电脑配件用 3DS MAX 创建, 用户替身采用 Poser 绘制, 导出的 VRML 文件在 CosmoWorlds 中修改并添加交互。考虑到实例的简单性, 协同并行控制采用对象锁方法实现, 后台数据库采用 SQL Server 2000, 主要存储电脑配件设备的参数信息和用户信息, 与数据库的连接采用 JDBC 实现。图 4 是在分场景中安装 CPU 的界面。图 5 是在分场景中安装键盘接口的界面。

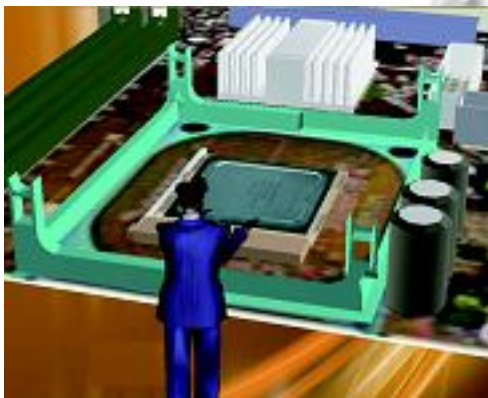


图 4 分场景中安装 CPU 的界面



图 5 分场景中安装键盘接口的界面

4 结语

本文提出一个多用户共享虚拟环境中 VRML 系统模型, 并在此基础上初步实现了原型系统。为了实现真正高效、大规模的分布式虚拟现实, 还需要在模型的实时性方面作进一步研究, 解决诸如分布式场景生成与变更的响应速度、网络带宽造成的时延和抖动等问题。另外, 服务器端业务规则的协同策略和多用户之间的相互协商, 协同工作过程的触发策略、突发事件的处理等, 都是模型需要仔细研究并加以改进的地方, 也是我们今后研究工作的重点。

参考文献

- 1 许爱军, 张文金, 易丹. 基于虚拟现实技术的远程教育平台研究与实现. 计算机系统应用, 2007, 16(8): 23-26.
- 2 李华, 孙智勇, 吴中福. 虚拟计算机网络组网平台的设计与实现. 计算机应用研究, 2006, 23(8): 258-260.
- 3 孙范荣, 王鹏, 黄有群. 基于 VRML 的交互式三维地形生成技术的研究. 系统仿真学报, 2007, 19(16): 3731-3733.
- 4 McGuinness DL, Fikcs R, Hendler J, et al. DAML+OIL: An ontology language for the semantic Web. IEEE Intelligent Systems, 2002, 17(4).
- 5 庞辽军, 李慧贤, 王力, 王育民. 基于 Java 的 VRML 网络通信功能的实现. 计算机工程, 2007, 33(11): 47-49.
- 6 行开新, 田凌. 支持异地协同设计的异构 CAD 虚拟装配系统. 清华大学学报(自然科学版), 2009, 49(2): 226-231.
- 7 许爱军, 张文金, 易丹. 基于 VRML 的虚拟现实技术及应用. 科学技术与工程, 2009, 37(4): 186-188.