

分布式虚拟现实系统中基本结构对象的分析*

郑 援 李思昆

(国防科技大学计算机系 长沙 410073)

摘 要 通过对现有几个系统实现的分析研究,提出实现分布式虚拟现实系统的最基本的九类结构对象及系统基本软件框架,主要对这些基本结构对象的功能及对象之间的关系进行详细的阐述,同时还对某些对象的具体实现方案作一些探讨,以方便分布式虚拟现实系统的构造。

关键词 虚拟现实(VR), 对象, 分布式, 多向广播, 锁定推算

分类号 TP311.5

The Analysis on Fundamental Objects of Distributed Virtual Reality Systems

Zheng Yuan Li Sikun

(Department of Computer Science, NUDT, Changsha, 410073)

Abstract Having studied a few systems, we present a frame of the essential software architecture for distributed virtual reality systems. In this paper, we investigate the fundamental objects with object-oriented principles and discuss several concrete implementation problems of some objects.

Key words virtual reality, object, distributed, multicast, dead-reckoning

多用户、多世界、多应用并发的存在构成了未来虚拟现实系统发展趋势,为适应大规模应用的要求,由单机到多机,由单用户到多用户,已成为虚拟现实发展方向。而分布式系统丰富的资源及强大的计算能力使之成为建立大规模虚拟现实系统的一种较好选择。

目前,国外已经成功地研制了 SIMNET^[1]、MR^[2]、DIVE^[2]、AVIARY^[3]、NPSNET IV^[5]等大规模分布式虚拟现实系统。在对这些系统进行分析研究的基础上,采用面向对象的方法,我们分析出实现分布式虚拟现实系统的最基本的九类结构对象,并提出一种实现

* 1996年5月23日收稿

分布式虚拟现实系统的基本软件框架。

1 系统的最基本的结构对象及其基本软件结构框架

目前出现的分布式虚拟现实系统各具特色,互有不同。尽管如此,从功能角度来看,这些系统均包含两个必不可少的功能子系统,即支持分布式应用的和支持虚拟现实应用的功能子系统。如果要支持多世界并发,系统还需增加用于支持多世界并发的功能子系统。由于多世界并发是未来虚拟现实系统的发展趋势,所以这三个功能子系统就是构成分布式虚拟现实系统的最基本的功能子系统。支持分布式应用的功能子系统是实现另外两个功能子系统的有力支持。支持多世界并发的功能子系统为支持虚拟现实应用的功能子系统提供信息和定位服务,并通过后者体现它自身的特点。支持虚拟现实应用的功能子系统则是实现虚拟现实应用的核心部分。

支持虚拟现实应用的功能子系统中最基本的对象是用户服务对象类、参与者对象类、碰撞检测对象类和建模对象类。支持分布式的功能子系统由名字服务对象类、对象管理对象类、系统管理对象类组成。世界管理对象和世界对象类则主要用于支持多世界并发。这九类对象就是构成分布式虚拟现实系统的最基本的对象。这些对象分布在系统的每一个点站上,它们之间通过消息进行信息交换,以完成系统的整体功能。

根据这三个子系统和九类对象的功能,我们提出实现分布式虚拟现实系统的基本软件结构框架,见图1。

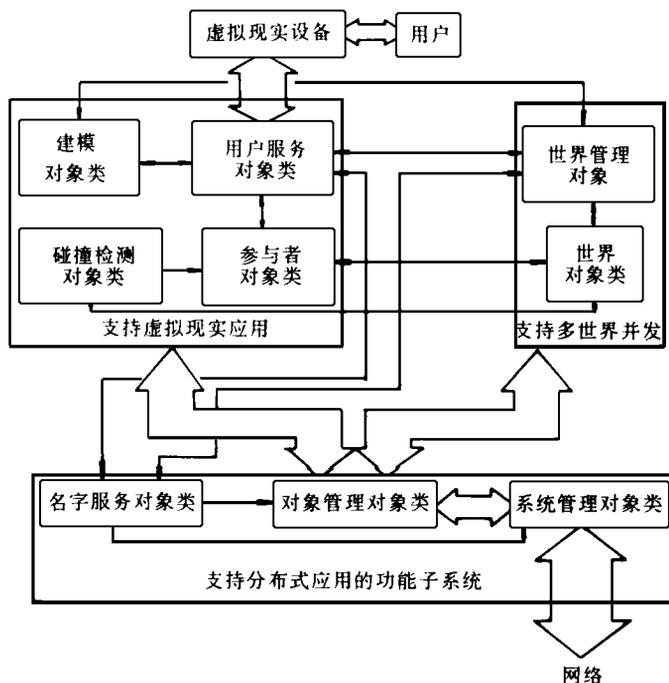


图1 分布式虚拟现实系统的基本软件结构框架

图1中箭头线表示对象或子系统之间的消息传递,箭头的粗细近似地表明消息量大

小。下面,我们用面向对象的方法分析各功能子系统中对象的功能及它们之间的关系。

2 支持虚拟现实应用的功能子系统

支持虚拟现实应用的功能子系统的基本对象存在于分布式虚拟现实系统的各个本地环境中,担负着直接同用户进行交互,表现虚拟现实环境的重要任务,是体现虚拟现实的三个特征即沉浸(Immersion)、交互(Interaction)和想象(Imagination)的主要部分。

(1) 用户服务对象类

用户服务对象类是系统与用户之间的界面,它们管理各种虚拟现实设备,负责将用户的输入转换成系统可以接收的数据,以便和系统内其他对象进行交互。同时,将系统输出反映到各种虚拟现实设备。这类对象的主要功能就是实现虚拟环境的沉浸、交互效果。

一般说来,每个用户拥有一个用户服务对象为其服务。当用户期望建立符合自己要求的应用或世界时,用户服务对象向建模对象发出请求,建模对象响应后,用户服务对象将用户的输入转换后发给建模对象,建模对象根据用户的输入建立虚拟世界的模型。用户要进入虚拟世界时,用户服务对象向世界管理对象发消息,获取虚拟世界的信息,同时还需要向名字服务对象发消息,为即将生成的参与者对象申请一个ID值,然后为用户生成参与虚拟现实应用的工具参与者对象,并告知参与者对象其要参与的虚拟世界的ID值。与此同时,该对象仍属于对象管理对象的管理范围,它的定位信息由对象管理对象掌握。

(2) 参与者对象类

这类对象是系统中同用户关系最密切的对象之一,它们代表用户在虚拟现实环境中的角色,同环境中的其他事物发生作用,是进行模拟的重要部分。

这类对象接收用户服务对象发来的有关用户输入输出的消息,根据这些消息计算用户的状态改变,在适当的时候将这个改变多向广播给其他有关对象。每个参与者对象保留一个世界数据库的部分拷贝,保存有当前参与的虚拟世界的最新数据,包括静态的从世界对象拷贝来的环境数据和模拟中动态生成的其他参与者的数据。当参与者对象接收到碰撞检测对象发来的有关碰撞消息时,将对世界数据库进行修改,如果碰撞改变了环境,还将向世界对象发消息。参与者对象还负责根据数据库中的信息,计算最新的虚拟现实场景,向用户服务对象发消息,后者将使用相应设备表现。

为了减少信息量,参与者对象采用锁定推算^[4](Dead reckoning)算法来绘制场景。状态改变消息并不是简单地汇报自己的位置坐标或者再加一点自己的形态信息,而是含有可用于计算自己某一时刻的状态的信息,例如,初始坐标、线速度、角速度和方向等。这样,在参与者对象的上述状态没有发生改变或改变未超过某个阈值的情况下,参与者对象不必发消息,而其他参与对象仍旧能够根据它原来的状态值推算它的当前状态,并绘制它的影像。某些情况下,其他参与对象绘制的参与者图像可能同参与者自己绘制的稍有不同。这种方法可使系统中的消息总量降低66%,从而可以较大地提高系统效率和实时性。

参与者对象的另一件任务是发心跳(Heart-beating)^[4]消息,即每隔一段时间多向广播一则有关自己状态的消息。这基于两个方面的考虑:一方面允许新的参与者随时进入已存在的虚拟世界,新参与者只要监听一段时间就能掌握该虚拟世界的情况;另一方面基于容错的考虑,当某个参与者非正常地退出时,其他有关对象可以做出正确的判断。

参与者对象需要维护两个时间: 一个是虚拟世界的实时时间, 该时间必须同其他有关对象同步; 另一个是自己的可伸缩的时间, 可以根据用户的需要加快或放慢。

(3) 碰撞检测对象类

这类对象担任一个重要角色, 即检测虚拟世界中众多事物的相对位置是否达到发生碰撞的程度, 若碰撞发生就通知相关对象。从 Kay & Kaijya 的包围盒策略^[6]借鉴思路, 将空间完全划分成不重叠的子空间, 按照二叉树的形式编码, 每个子空间由一个碰撞检测对象管理。进入某区域的事物增多时, 该区域内碰撞检测对象的效率将降低, 该对象将向世界对象发消息申请分裂成两个子空间, 并增加一个碰撞检测对象。这样, 多碰撞检测对象的并发操作大大提高检测速度, 子空间的划分又避免了无关物体之间的检测操作。

(4) 建模对象类

这类对象用于建立新的应用, 为应用建立虚拟世界的背景和规律, 是体现虚拟现实想象特征的天地。建模的范围包括静态景物建模、物理规律建模、运动规律建模、通讯建模和几何外形建模。这些模型规定了虚拟世界中的方方面面。

3 支持分布式应用的功能子系统

这个子系统是实现多用户的虚拟现实应用的基础。主要任务是高效可靠地实现各结点之间的信息交换和信息共享, 尽量降低时间开销, 消除用户之间地域差异的影响。

(1) 名字服务对象类

名字服务对象类掌握系统内所有对象的 ID 值, 它们接收来自各对象有关名字的请求, 为新产生对象取 ID 值。各个名字服务对象保存的系统 ID 值数据库必须保持一致。

(2) 系统管理对象类

通讯和负载均衡是分布式虚拟现实系统低层的功能。系统管理对象类负责这两项工作, 它们的目标是实现对象之间的端-端通讯、世界内对象之间的多广播^[3](Multicast)及系统内的广播通讯, 在对象管理对象协助下完成对象在处理机之间的迁移。

分布式虚拟现实系统中, 各种通讯通过消息传递实现, 对象的大量存在必然导致消息量的增大, 处理这些消息的时间开销增大, 系统的通讯效率降低。引入多向广播就是为了解决这个问题。多向广播发的消息只有同组成员才可以接收到, 此时同组成员只要知道在哪个频道发送和接收消息就可以了。利用这种方法, 可使消息量从端-端通讯的 $O(n^2)$ 降到了 $O(n)$, 并且还可使与本消息无关的对象避免接收一些无用的消息。

负载均衡在各处理机之间调度各进程, 是分布式应用中需要考虑的重要问题之一。

(3) 对象管理者对象类

这类对象的设置是为了管理存在于同一台机器上的对象, 向系统管理者对象类报告本机上的对象的情况, 以便建立本地对象同远程对象之间的通讯和系统的负载均衡。

4 支持多世界并发的功能子系统

这个子系统用于支持多世界并发, 主要功能是: 管理系统中出现的多个世界及每个世界中出现的多个参与者、多种因素; 向虚拟现实应用子系统提供有关多世界的信息服务; 支持用户在多个世界之间的漫游。这类对象包括世界管理者对象和世界对象类。

(1) 世界管理者对象

这个对象是针对多世界并发的特点而设置的。它提供对系统内多个世界的管理,协助系统内的用户在多个世界内的漫游,支持多应用多世界的并发。世界管理者对象掌握整个系统内世界的情况及本系统内正在运行的世界的 ID 值,负责回答来自其他对象的询问,生成和终止世界对象,接受来自建模对象对世界最后的修改结果。

但是如何实现这个对象需要深入考虑,这与世界数据库的存储位置有关。世界数据库的存储位置是影响整个分布式系统效率的重要因素,它的存储方法目前有许多种。实现世界管理对象的原则是,能够使参与该系统的其它对象,无论他们何时加入模拟,都能够很方便地找到这个对象,而且尽量减少对世界管理对象的访问次数。

我们提出一种动态 Client/Server 结构。将有关所有世界的数据库存储在一个中心 Server 上,将世界管理者对象也建立在上面,以便减少延迟。当第一个使用某个世界的对象要求进入时,则由世界管理者对象协助该对象拷贝全部的有关该世界的信息,使该对象成为一个有关该世界的局部 Server,后来者访问世界信息时只需访问这个 Server 就可以了。从而同时保证了数据库的可访问性,又减少了中心 Server 的负担。这种结构对局域网比较适合,而对广域网就不太适合了,因为中心 Server 的设置变得很困难。

(2) 世界对象类

这类对象是一个虚拟世界的维护者,一个世界对象保存完整的该世界的静态环境数据库,是一个局部 Server,负责向参与者对象提供相应的环境数据,它掌握着虚拟世界的实时时间和特殊规律。它还负责维护世界内所有参与者的名单以支持参与者的介入退出。它掌握本世界内空间的划分,掌握碰撞检测对象的分布,向参与者对象提供有关信息。

5 结 论

大规模虚拟现实系统是虚拟现实发展的必然趋势,而采用分布式实现又是很有前途的发展方向。本文采用面向对象的方法,对我们提出的分布式虚拟现实系统的基本软件框架中结构对象的功能、对象之间的关系进行详细的阐述,同时还对某些对象的具体实现方案作一些探讨。实际上,面向对象方法的使用并不限于分析实现的软件结构,在具体做系统的过程中也可以按照这种方法做。

参 考 文 献

- 1 Jack A. Thorpe, netted engagement simulation: 1 ~ 10
- 2 Grigore Burdea & Philippe Coiffet, Virtual Reality Technology: 217 ~ 219
- 3 David N S, Adrian J W. AVLARY: Design Issues for Future Large-Scale Virtual Environments. PRESENCE 1994, 3(4): 288 ~ 308
- 4 Rich Gossweiler, Robert J L, Michael L K, Rardy P. An Introductory Tutorial for Developing Multiuser Virtual Environments PRESENCE 1994, 3(4): 255 ~ 264
- 5 Michael R M, etc. NPSNET: A Network Software architecture for Large-Scale Virtual Environments. PRESENCE 1994, 3(4): 265 ~ 287
- 6 Kay T L, etc. Ray tracing complex scenes. ACM Computer Graphics, 20(4): 269 ~ 278

(责任编辑 张 静)