

虚拟环境中行为建模技术研究*

苏绍勇 陈继明 潘金贵

(南京大学计算机软件新技术国家重点实验室 南京大学计算机科学与技术系 南京 210093)

摘要 本文从软件方法学的角度将虚拟环境中行为建模技术分为基于过程的行为建模、基于对象的行为建模和基于 Agent 的行为建模,介绍这些技术的同时并分析了其各自的优缺点及适用范围,最后展望了行为建模技术的发展方向。

关键词 虚拟环境,行为建模,基于对象,基于 Agent

Research on Behavior Modeling Technology of Virtual Environment

SU Shao-Yong CHEN Ji-Ming PAN Jin-Gui

(State Key Lab for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract According to software methodology, this paper categorizes behavior modeling technologies into three classes which are procedure-based behavior modeling, object-based behavior modeling and Agent-based behavior modeling, including their pros and cons and applicable area. Finally, we conclude with the directions for future work.

Keywords Virtual environment, Behavior modeling, Object-based, Agent-base

虚拟环境^[1] (Virtual Environment, VE) 是一种计算机图形交互技术,其基本特征是沉浸性、交互性、构想性。根据研究的侧重点不同,可以将虚拟环境分为两类,一类是分布式虚拟环境^[2,3] (Distributed Virtual Environment, DVE),侧重在网络环境下与其中的内容进行交互和通信;另一类是协同式虚拟环境^[4] (Collaborative Virtual Environment, CVE),侧重对虚拟环境中协作和交互的支持。在所有这些类型的虚拟环境中,物体之间的交互是很重要的一个方面,因此如何对物体之间的交互行为进行合适的建模是虚拟环境中的一个研究热点。

行为建模是探索一种能够尽可能贴近真实实体对象行为的模型,使构造实体对象的人能够按照这种模型方便地构造出一个行为上真实的虚拟实体对象^[5]。它起源于人工智能领域的基于知识系统(模拟外在智能的抽象形式,如问题求解、博弈等)、人工生命(动物行为学中的适应性行为,适应周围环境的变化)和基于行为的系统(Behavior-Based System, BBS),基于行为的系统中自主 Agent 行为表示与计算机动画结合还产生了行为动画机制^[6~8]。行为建模的主要目标是对真实物体的行为(包括反应行为和智能行为)进行准确的建模,使得能在计算机上对其进行模拟。

行为建模的目标与虚拟环境中建模的需求是一致的,因此行为建模被认为是虚拟环境中建模技术研究的一个重点。虚拟环境中的行为建模包括设计事件的类型、触发方式和时机,事件和对象任务的结构以及每个事件处理函数和任务的代码实现等。

虚拟环境中的行为建模技术有很多种,本文从软件方法学的角度将这些技术方法分为基于过程的行为建模、基于对象的行为建模和基于 Agent 的行为建模,下面对这些方法逐

一进行介绍。

1 基于过程的行为建模

基于过程的行为建模是指对虚拟环境中物体建模时,将物体的属性数据与其行为动作分离,利用过程式的方法和编程语言来实现整个建模过程。其典型的代表是 DIVE 系统中采用的建模方法^[9]。

DIVE 是由瑞典计算机学院开发的分布式虚拟环境系统,该系统中的物体是通过 DIVE 实体和 Tcl 脚本构成。其中 DIVE 实体中仅包含物体的属性数据,物体的行为动作都是通过触发器触发 Tcl 脚本来执行。其过程是这样的:通过 DIVE/Tcl 接口,系统触发器 trigger 检测系统内发生的事件,并引发对应的 Tcl 脚本执行相应的动作。例如,一个物体有以下脚本:

```
proc move-up {id type actor srcid} {
    for { set i 0 } { $i < 100 } { incr i } {
        dive-sleep 100
        dive-move $id 0 0.5 0 LOCAL-C
    }
}
dive-register select move-up
```

则当有用户对该对象进行 select 操作时,进程 move-up 会被触发执行。

该系统的物体间冲突检测机制使用冲突管理者 (collision manager)。进程注册感兴趣的物体对象和用户化身,当有感兴趣事件发生时冲突管理者发出冲突信号,系统调用已写好的 DIVE/Tcl 脚本处理冲突事件。如一盏灯检测到有物体靠近它时自动打开:

```
proc turn-on { id1 id2 type } {
    set id {dive-find-subobj $id1 "bulb"}
    dive-light $id on
}
```

* 本课题得到国家自然科学基金项目(60473113)和重点项目(60533080)的资助。苏绍勇 硕士研究生,主要研究领域为虚拟环境、Agent 技术;陈继明 博士研究生,主要研究领域为虚拟环境、Agent 技术;潘金贵 教授,博士生导师,主要研究领域为多媒体信息处理、多媒体远程教育系统。

dive-register collision-on turn-on

DIVE 系统中这种属性与行为分离的方法,采用的是过程式的思想。用该方法建模的虚拟环境系统其触发执行过程如图 1 所示。

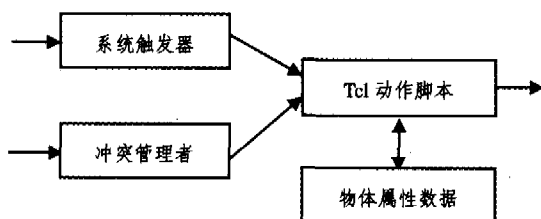


图 1 面向过程式行为建模触发执行过程

这种行为建模方法需要预定义物体的行为,运行时在应用层进行触发,使得物体执行该行为动作。预定义物体行为动作限制了该建模方法的描述能力,应用层触发也使得该建模方法缺乏灵活性。随着技术的发展,这种行为建模方法使用得越少了。

2 基于对象的行为建模

基于对象的行为建模方法将物体的属性数据与行为描述结合在一起,这是现在比较流行的一种建模方法,该方法的典型代表是 Smart Object 和角色分类法,下面对这两种方法介绍。

2.1 Smart Object

最早使用对象思想进行建模的是 Marcelo Kallmann 和 Daniel Thalmann,他们在文[10,11]中提出了 Smart Object, Smart Object 通过特征建模(feature modeling)把物体的行为动作包含到了物体的描述中来。Smart Object 包含图形表示、内在属性、如何交互、自身行为等内容,这样 Smart Object 的属性、行为都包含在自身中,相当于面向对象概念中的类或对象。Smart Object 通过特征建模实现,所有的特征都存在于基于文本的脚本文件中,其行为特征主要有三个方面:命令(command)、变量状态(variable state)和行为(behavior)。命令是交互的接口,变量状态指示物体当前状态,行为描述物体在当前状态下接受特定命令时应执行的动作。其事件触发机制与执行方式是基于文本的命令驱动(文[11]中有提到用虚拟现实输入设备驱动),通过匹配文本命令来执行相应动作。Smart Object 行为特征结构示意图如图 2 所示。

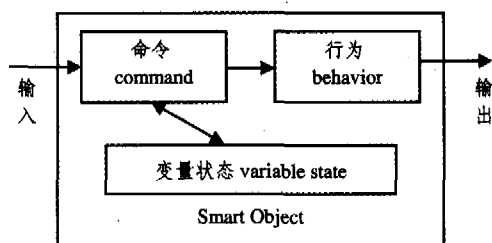


图 2 Smart Object 行为特征结构示意图

使用 Smart Object 对虚拟环境中物体建模,使得物体间的交互行为由原来的应用层预定义动作转为对象层的选择性执行。在此之前的虚拟环境中物体间交互触发机制都是当某一特定事件发生时执行某一特定的已定义好的动作,缺乏灵活性。使用 Smart Object 建模的虚拟环境,其物体间的交互行为可由物体当前状态和所接受到的命令来共同决定执行何种动作,使得交互行为少了一些机械化多了一些真实感。

Smart Object 作为第一个运用面向对象思想对物体建模的方法,也有其局限性。Smart Object 只对虚拟环境中狭义的对象建模,而没有包括虚拟人(当时虚拟环境中虚拟人仅指用户化身,因为环境中还不存在计算机控制的自主人,也称计算机生成人员),这使得该建模方法不够通用。另外,它的事件触发机制(基于文本的命令驱动)也限制了虚拟环境的交互方式,仅通过文本命令方式驱动的虚拟环境也会使用户失去与虚拟环境交互的沉浸感。

Pieter Jorissen 和 Wim Lamotte 在文[12]中对 Smart Object 进行了改进。首先他们将 Smart Object 扩展,使得虚拟环境中广义上的物体(包括虚拟人)都可以通过 Smart Object 进行建模。建模后所有物体组成一个交互层(interaction layer),交互层还会创建一些必需的链(如接触链 contact links, 隶属链 attach links, 控制链 controller links, 父子链 parent/child links)。新的触发机制就与交互层以及这些链有关。事件由触发器(包括冲突触发器,命令触发器,时间触发器)触发,这些触发器都属于物体对象级的,交互层负责检测所有对象的触发器,捕获到事件发生后再通过检查各个链来决定该事件对应的处理过程。

2.2 角色分类法

Weihua Wang 和 Qingping Lin 在文[13]中提出一种角色分类的物体建模方法。该方法是对协同式虚拟环境中交互物体建模,它将交互物体抽象成各种角色,每种角色都有对应的类实现(如经理类 Manager、店员类 ShopAssistant、播放机类 CDPlayer 等等),这些类之间还存在继承等关系。它使用监听器来触发事件,监听器包括位置监听器 PositionListener、方向监听器 OrientationListener、私聊监听器 PrivateChatListener、公聊监听器 PublicChatListener、命令监听器 CommandListener、计时监听器 ClickListener。这些监听器负责监听用户通过图形用户接口发来的操作与命令。不同于 Smart Object 的是,该行为建模方法通过现在流行的面向对象的编程手段实现,并未使用特殊的方法(特征建模法),这使得该方法更易于实现。在文[14]中,这种建模方法还被上升为了一种面向对象式的建模描述语言 CDBL (Collaborative Behavior Description Language)。

当然,除了这两种方法外,还有其他应用面向对象思想对虚拟环境中物体进行行为建模的方法,不过 Smart Object 和角色分类两种方法更为典型。基于对象的行为建模将物体的属性与行为结合在一起,使得虚拟环境中物体自身的行为、物体之间的交互更加灵活自然,用户与虚拟环境的交互也更为自然。加之现在有大量的面向对象语言(如 C++、Java)的支持,基于对象的行为建模方法成为现在虚拟环境中主流的建模方法。

3 基于 Agent 的行为建模

基于对象的行为建模方法有其优势,但也有其局限性。如虚拟环境中计算机控制的虚拟人应具有一定的智能性,能根据环境变化自主做出反应等,基于对象的行为建模方法在这些方面能力有限,而基于 Agent 的行为建模技术则可以在这些方面有所作为。基于 Agent 的行为建模技术是虚拟环境中行为建模的发展方向。

目前,采用基于 Agent 的行为建模技术进行建模的主要有军事领域的计算机生成兵力(Computer Generated Force, CGF)和民用领域的计算机生成人员(Computer Generated

Actor, CGA)。CGF 和 CGA 都属于虚拟环境中计算机控制的虚拟人, 基于对象的行为建模很难表达这些虚拟人的信念、愿望、意图等精神状态。而 Agent 从模拟人的行为出发, 对一个实体的信念、承诺、意图、愿望等精神状态进行描述, 加之 Agent 具有自主性、自治性和智能化等特征, 因此基于 Agent 的建模技术更能准确地描述和表达虚拟环境中具有智能的虚拟人。

基于 Agent 的行为建模主要是对虚拟中具有智能性的物体建模, 所以实现较为复杂。实现过程中有两个问题需要注意: 反应实时性和处理不确定性。反应实时性是指对外界变化要及时做出反应, 不能容忍过长时间的推理; 处理不确定性是指要有处理各类情况的能力, 对外界环境的各种变化都应能给出相应的行为反馈。

根据侧重点不同, 基于 Agent 的行为建模可以分为, 侧重虚拟环境中单个物体建模的单 Agent 行为建模和侧重虚拟环境中多个物体之间交互的多 Agent 行为建模。

3.1 单 Agent 行为建模

Agent 可以分为反应型和认知型, 反应型不对内部状态进行推理, 仅对环境变化和其他 Agent 的消息做出反应; 认知型 Agent 基于符号表示和推理来采取动作和完成任务, 具有较高的智能, 但不能对环境变化做出快速反应。与此对应, 基于 Agent 的建模可以分为两层: 反应层和规划层。反应层采用神经网络加以建模, 基本不做推理, 直接由感知信息映射到行动, 模拟了反应性行为的处理过程; 规划层则进行推理工作。当 Agent 接受到外界信息后, 若该信息属于响应性的, 直接做出响应, 否则由规划层进行规划并处理。

文[15]基于人认知的信息流及行为框架, 提出一种 Agent 的模型。该框架将人认知的信息流及反应分为三个层次: 基于知识的层次, 基于规则的层次和基于技巧的层次。基于知识的层次采用规划的方法加以实现, 基于规则的层次采用规则建立状态与任务的关系, 模拟反应性行为; 基于技巧的层次类似于机械的动作, 它直接根据外部的输入进行动作, 不作推理及判断工作。文[15]实现了基于规则和基于技巧的两层。

这种分层的 Agent 建模方法具有多种类行为的处理能力, 但其局限性也很明显: 反应层的行为是预先静态定义的, 所以对很多超出其范围的事件无法实时响应; 而规划层经过非实时规划产生的控制信息只可用一次, 它并不具备记忆功能, 使得下次遇到同样情况时还是需要进行非实时的规划。

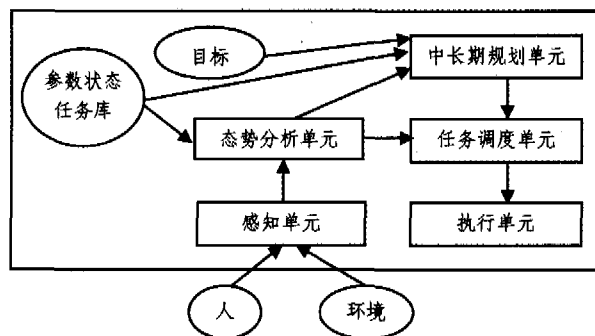


图3 基于 Agent 的框架

针对这些不足, 文[16]提出了一种基于 Agent 的 CGA 体系结构(其结构如图3所示, 图3来源于文[16]), 它的基本认识思想是: 行为虽然具有不同的类型, 体现不同的认知能

力, 但高层次的行为最终要通过低层次的行为表现出来。该方法抽取环境中的基本行为构建基本行为集, 基本行为采用参数化的方法构建, 复杂行为则被规划为基本行为的序列。Agent 内采用态势分析单元进行实时的规划, 对任务序列进行修正, 必要时在进行中长期的非实时规划, 这样系统可以兼顾反应实时性和处理不确定性两方面问题。这种方法对行为的处理不再是静态的划分, 而是通过参数化的方法提高其适应性。

3.2 多 Agent 行为建模

与单 Agent 行为建模不同的是, 多 Agent 行为建模将物体行为分为自主行为和外部互操作行为^[17]。自主行为是指 Agent 本身所特有的活动行为, 外部互操作行为是指该 Agent 与周围环境及其它 Agent 有关的交互行为。如图4所示, 多 Agent 虚拟环境中的行为模型包括属性、自主行为模型、互操作行为模型等几个部分。自主行为模型由消息和规则组成, 它与 Agent 局部行为约束有关。互操作行为模型由消息、规则和约束组成, 它与全局行为约束有关。

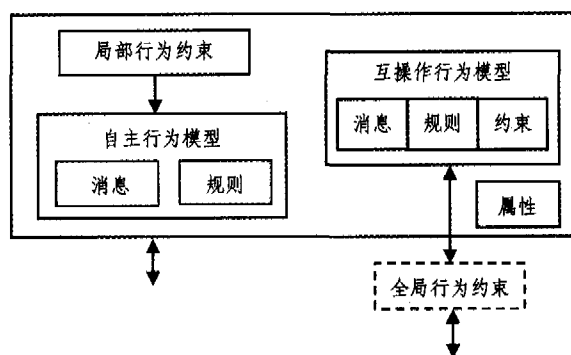


图4 多 Agent 建模行为模型

多 Agent 行为建模将与其他智能体交互的行为单独列出, 并单独建模, 该方法对存在多智能体的协同式虚拟环境尤其适用。除此之外, 目前在其他方面多 Agent 行为建模与单 Agent 行为建模使用的技术比较相似。基于 Agent 的行为建模尤其适用于对虚拟环境中具有智能性的物体进行建模。

4 行为建模技术分析

基于过程的行为建模实现简单, 对建模物体可以方便地添加自定义行为, 适用于侧重仿真研究的虚拟环境, 如动物行为的仿真、虚拟人在特定情况下的动作仿真(如手抓取物体)等。但该方法对环境中物体间的交互支持较差, 其预定义行为描述脚本特性也限制了环境的灵活性和可扩展性。

基于对象的行为建模运用面向对象思想将物体属性数据和行为描述结合在一起, 较好地支持了环境中物体间的交互, 并且其行为动作的触发也在对象层选择执行, 使得所建模的虚拟环境灵活且易于扩展, 主要适用于涉及物体种类且其间交互较多的虚拟环境。但基于对象的行为建模对虚拟环境中具有智能性的物体(如计算机控制的虚拟人等)支持不够, 在表示智能性物体的信念、愿望、意图和推理等方面存在一定局限性。

基于 Agent 的行为建模通过引入 Agent 概念实现了对智能物体的信念、愿望等精神状态以及判断、推理等思维过程的建模, 使得环境中智能物体的行为反应更加真实、自然, 适用于智能型的虚拟环境。不过, 目前基于 Agent 的行为建模技术正处于发展之中, 尚未实现完全意义上的智能行为建模(如

人的思考方式、自主行为等)。基于 Agent 的行为建模仍存在一些困难^[18]:如人脑思维模型的复杂性导致无法保证系统建模中抽象、表达、推理及学习等过程的正确性等。

小结 随着虚拟环境系统越来越复杂、要求越来越高,对环境中物体进行合适的行为建模也越来越成为人们关注的焦点。行为建模经历了基于过程的行为建模、基于对象的行为建模到基于 Agent 的行为建模,由于 Agent 具有描述人的信念、愿望、意图等精神状态的能力,所以基于 Agent 的行为建模会成为未来智能化、真实化虚拟环境的主要建模方法。

目前,基于 Agent 的行为建模技术虽然还处于研究与探讨阶段,但仍已被认为具有较为广阔的研究前景。主要在以下两个方面需要进一步的研究:一、系统应用方面,即如何综合运用现有的基于 Agent 的行为建模技术构建更加智能、快捷的虚拟环境;二、模型构建方面,即如何构建更加智能、更能反应人各种真实行为的 Agent 模型。

参考文献

- 1 Stuart R. The Design of Virtual Environments. McGraw-Hill, New York, NY, 1996
- 2 Greenhalgh C. Large Scale Collaborative Virtual Environments; [PhD Thesis]. The University of Nottingham, 1997
- 3 潘志庚,姜晓红,张明敏,石教英. 分布式虚拟环境综述. 软件学报, 2000, 11(4): 461~467
- 4 Benford S, et al. Collaborative Virtual Environments. Communications of The ACM, 2001, 44(7): 79~85
- 5 郑援,李思昆. 虚拟实体对象的行为建模方法研究. 国防科技大学学报, 1998, 20(1): 78~82
- 6 Cermer J, Kearney J, Ko H. Simulation and scenario support for virtual environments. Computer & Graphics, 1996, 20(2): 199~206

- 7 Reynolds, Flocks C W, Herds, Schools. A Distributed Behavioral Model. Computer Graphics, 1987, 21(4): 25~34
- 8 Tu Xiaoyuan, Terzopoulos D. Artificial fishes: physics, locomotion, perception, behavior. In: Proceedings of SIGGRAPH 94, 1994
- 9 Hagsand O. Interactive Multiuser VEs in the DIVE System, IEEE Multimedia Magazine, 1996, 3(1): 30~39
- 10 Kallmann M, Thalmann D. Modeling Objects for Interaction Tasks. In: Proc. Eurographics Workshop on Animation and Simulation, 1998, 73~86
- 11 Kallmann M, Thalmann D. A Behavioral Interface to Simulate Agent-Object Interactions in Real Time. In: Proc. IEEE International Conference on Computer Animation, 1999. 138~146
- 12 Jorissen P, Lamotte W. A Framework Supporting General Object Interactions for Dynamic Virtual Worlds. In: Proc. International Symposium on Smart Graphics, 2004. 154~158
- 13 Wang Weihua, Lin Qingping. Behavior Based Interaction Management Mechanism for an Internet CVE System. In: Proc. IEEE International Conference on Multimedia and Expo, 2001. 820~823
- 14 Lin Qingping, Wang Weihua, et al. A Novel Method for Supporting Collaborative Interaction Management in Web-based CVE. In: Proc. Conference on Virtual Reality Software and Technology, 2003
- 15 Mulgund SS, Harper KA, Zacharias GL. SAMPLE: Situation Awareness Model For Pilot-In-The-Loop Evaluation. In: Proc. 9th Computer Generated Forces and Behavior Representation, 2000
- 16 梁晓辉,胡晓雁. 一种基于 Agent 的构造性虚拟实体的体系结构. 系统仿真学报, 2001, 13(11): 138~141
- 17 党岗,刘华峰,等. 多 Agent 虚拟环境中行为建模及其动画表现. 系统仿真学报, 2001, 13(11): 142~145
- 18 张航义. 基于 Agent 的 CGF 行为建模技术研究. 计算机仿真, 2003, 20(8): 79~81

(上接第 266 页)

在上面的例子中, $libz.so \propto libz.so.1.2.2$, 所以在用备份 $libz.so.1$ 替换 $libz.so.1.2.2$ 后, 也应该同时用备份替换 $libz.so$. 可信恢复算法不仅保证了系统内部单个客体的完整性, 而且还保证了系统内部客体的一致性。

结束语 可信恢复主动保护安全操作系统的安全性。文章在对安全操作系统安全模型研究的基础上, 首先描述了一个通用的安全模型, 然后扩展该模型描述了安全操作系统中安全性损害。随后, 描述了如何根据安全策略借助审计日志的安全监视功能来检测安全操作系统中发生的种种安全性损害。最后, 讨论了可信恢复如何修复恶性机密性损害和恶性完整性损害, 保证安全操作系统的安全性。可信恢复主动保证安全操作系统的安全性, 增强了安全操作系统的可靠性和可恢复性。本文的工作为今后开发《访问验证保护级》安全操作系统和对可信恢复的进一步研究提供了理论模型和方法指导。

参考文献

- 1 石文昌, 孙玉芳. 安全操作系统研究的发展(上)[J]. 计算机科学, 2002, 29(6)
- 2 石文昌, 孙玉芳. 安全操作系统研究的发展(下)[J]. 计算机科学, 2002, 29(7)
- 3 Sullivan M, Chillarege R. Software Defects and their Impact on System Availability - A Study of Field Failures in Operating System [J]. 21th Int. Symp. On Fault-Tolerant Computing, June

1991. 2~9
- 4 Randell B. Operating systems: The problems of performance and reliability [J]. In: Proc. IFIP Congress, North-Holland, 1971. 281~290
- 5 National Computer Security Center. A Guide to Understanding Trusted Recovery in Trusted Systems [R]. NCSC-TG-022, Library No. 5-236, 061, Version 1, December 1991
- 6 中华人民共和国国家标准: 计算机信息系统安全保护等级划分准则[S], GB17859-1999
- 7 Dept. of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria [S]. DOD 5200. 28-STD, GPO 1986-623-963, 643 0, Dec. 1985
- 8 National Computer Security Center. Final Evaluation report Wang Government Services, Incorporated XTS-300 STOP 5. 2. E [R]: [Report NO. CSC-EPL-92/003. E]. August 2000
- 9 Gasser M. Building a Secure Computer System [M]. Van Nostrand Reinhold. ISBN 0-442-23022-2
- 10 Bell DE, LaPadula LJ. Secure Computer Systems; Mathematical Foundations [R]: [Technical Report M74-224]. The MITRE Corp., Bedford, Mass., May 1973
- 11 Biba K J. Integrity considerations for secure computer systems [R]. USAF Electronic System Division, Hanscom Air Force Base; [Tech. Rep., ESD-TR-76-372]. 1977
- 12 Naval Research Laboratory. Handbook for the Computer Security Certification of Trusted Systems. NRL Technical Memorandum 5540; 062A, 12 Feb. 1996. 14