

# 虚拟实体对象的行为建模方法研究\*

郑援 李思昆 胡成军 邴来斌

(国防科技大学计算机学院 长沙 410073)

**摘要** 虚拟实体对象的行为建模方法已成为虚拟现实技术的研究热点之一。本文总结了前人在行为建模方面的成就,分析了其中存在的问题,并在此基础上提出了一种新的适于虚拟现实开发平台的虚拟实体对象行为模型 SCP 模型,并对这种行为模型的基本理论和应用界面进行了详细阐述。SCP 模型以控制论为基础,适用领域广,表达能力强,适合并发实现,且易于重用已有的行为建模成果。

**关键词** 行为模型, 虚拟世界, 实体对象

**分类号** TP391

## Research on Behavior Models of Virtual Artifacts

Zheng Yuan Li Sikun Hu Chengjun Yan Laibin

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

**Abstract** This paper addresses the behavior models of virtual artifacts. First we provide the results in our research on related work. Then we present a novel behavior model SCP, which is based on principles of cybernetics, for creating virtual artifacts that move and behave in simulated virtual environments. Finally, we also provide its user interface and characteristics.

**Key words** behavior model, virtual reality, virtual artifact

虚拟世界中存在大量的实体对象,例如动物、人类、机器、汽车和建筑物等,这些实体对象不但具有几何形体,而且均有自己的行为方式。仅当这些实体对象都以令人信服的方式进行行为运动时,虚拟世界的真实感才能得以体现,因此,对虚拟世界中的实体对象进行行为建模就成为虚拟现实的一项关键性技术。

行为建模的任务就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。

## 1 发展现状

鉴于行为建模技术的重要性和复杂性,前人已经做了不少探索。目前已有的行为建模方法大致可以划分为两个研究方向。(1)针对某种具体行为,研究特定行为模型。例如使用关键帧方法描述简单重复的对象行为<sup>[1]</sup>;使用反向动力学模型描述物理对象由初始状态到终止状态的变化过程<sup>[2]</sup>;使用约束分析模型描述对象之间的运动关系<sup>[3]</sup>;使用位置参数模型描述人体及动物的行走行为<sup>[4]</sup>等。这些方法能够逼真和高效地表现对象的特定行为,但却不适用于其他领域的行为问题。(2)面向虚拟现实应用开发平台,研究适用领域较广的行为模型,著名平台 VRML、MR Toolkit 和 BrickNet 的行为建模方法代表了这一方向的成果。VRML<sup>[5]</sup>是一种网上语言,2.0 版引入了基于事件驱动模式的行为实现。事件的传递和处理全部由系统完成,系统根据事件内容直接修改对象的某些域(如位置、转角等),使对象表现出行为。在表现连续行为时,由时钟定时激活显示关键帧。MR Toolkit<sup>[6]</sup>允许为实体对象指定多种行为(实际是动作),应用程序在每个模拟时钟周期中根据事件的发生或条件的出现,激活对象的某些行为,并在每个模拟周期末执行这些行为(动作),表现连续行为时,在每个模拟周期,由时钟滴答事件

\* 1997 年 6 月 27 日收稿  
第一作者:郑援,女,1973 年生,博士生

激活一次相应行为。BrickNet<sup>[7]</sup>不同于 MR Toolkit，它将行为描述嵌入对象的“动画方法”，由于动画方法每个模拟周期均被调用，实体对象的行为也因此得以连续进行。为了确定某一时刻的准确动作，在每个模拟周期，动画方法都必须进行大量的条件测试。

上述工作作为行为建模打下良好的基础，但仍存在不足之处。主要问题有：事件作为对象行为过程中重要的一环，其传递乃至处理由应用系统完成，造成对象行为被动实现，削弱了虚拟实体对象的自主性，也降低了虚拟实体对象之间的并发性；特定行为的建模成果难以直接移植到虚拟现实开发平台，以便在平台上建立具备这种特定行为的虚拟实体对象，已在平台上开发好的虚拟实体对象的行为又难以被其它虚拟实体对象重用。

为克服上述不足，采用控制论的思想，面向虚拟现实开发平台提出了一种新的基于控制论<sup>[8]</sup>行为模型 SCP。这种模型表达能力强，适合并发实现，且易于重用已有的行为建模成果。

## 2 SCP 的概念模型

控制论的行为观<sup>[8,9]</sup>认为：行为是实体对象随时间推移而产生的自身状态的变迁，它是实体对象内在规律和能力在外部干扰下的表现。

基于控制论的行为观，构造出基本原理如图 1 所示的行为模型，即 SCP (Sensor, Controller Performer)。它是由三个相对独立而又密切相关的行为构件（感受器、控制器和效应器）组成的控制系统。其中，感受器主动从虚拟实体对象内部和外部收集涉及该实体对象的所有事件，并将相关事件过滤出来，传递给控制器。控制器的主要功能是：对感受器传来的事件进行处理并测试某些相应条件，在此基础上，实行对虚拟实体对象行为状态的控制，为虚拟实体对象指定新的行为状态。它是虚拟实体对象行为的控制中心，使虚拟实体对象表现出行为的智能性和对外部环境的适应性。控制器指定行为状态后，由效应器负责按新的行为状态执行，使虚拟实体对象表现出特定行为。效应器能够发送事件给其它对象，实现该虚拟实体对象的影响。控制器没有指定新的行为状态之前，效应器按原行为状态执行。

反馈是控制论最重要的概念之一，也是虚拟实体对象行为过程中必不可少的部分。内部反馈事件由效应器直接发送到感受器，利用这种反馈，控制器能够对效应器的执行过程进行监督、调整，实现对虚拟实体对象行为的正确控制。外部反馈事件以外部事件的形式反馈给感受器，它是虚拟实体对象对其它对象施加影响（输出事件）后，其它对象做出反应时发出的事件。利用外部反馈事件，虚拟实体对象能够对自己行为的作用有所认识，从而相应地调整行为。

反馈是控制论最重要的概念之一，也是虚拟实体对象行为过程中必不可少的部分。内部反馈事件由效应器直接发送到感受器，利用这种反馈，控制器能够对效应器的执行过程进行监督、调整，实现对虚拟实体对象行为的正确控制。外部反馈事件以外部事件的形式反馈给感受器，它是虚拟实体对象对其它对象施加影响（输出事件）后，其它对象做出反应时发出的事件。利用外部反馈事件，虚拟实体对象能够对自己行为的作用有所认识，从而相应地调整行为。

## 3 SCP 的形式化模型

为描述状态随时间推移的变化，在 SCP 模型中，总假定存在一个统一的全局时钟，记时钟的值域为  $T$ 。还假定任何虚拟实体对象的状态，都可以在一定精确度上，用决定它的一组值来描述。考虑任意一个虚拟实体对象  $o$ ，记  $o$  的所有属性变元（如几何外形、地理位置、转角、速度和角速度等）组成的集合为  $V$ 。任一时刻，对变元集  $V$  的一个类型一致的赋值，称为  $o$  的一个状态，并记  $o$  的所有状态的集合为  $S$ 。所有能够影响  $o$  行为的事件，包括外部事件（含外部反馈事件）和内部反馈事件组成的集合称为  $o$  的相关事件集，记为  $E$ ；而  $o$  可能产生的所有事件（包括内部反馈事件和向外部环境输出的事件）构成的集合称为  $o$  的产生事件集，记为  $Y$ 。

**定义 1** 虚拟实体对象的行为状态是形如  $\langle \delta, \lambda \rangle$  的二元偶，其中  $\delta: S \times T \rightarrow S$  是下一状态函数， $\lambda: S \times T \rightarrow 2Y$  是输出函数。

记  $\Omega = \{ \langle \delta, \lambda \rangle \mid \delta: S \times T \rightarrow S, \lambda: S \times T \rightarrow 2Y \}$  是行为状态的全集。

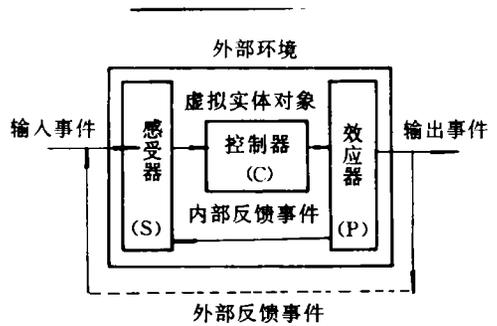


图 1 SCP: 基于控制论的虚拟实体对象行为模型

**定义 2** 虚拟实体对象的行为模板是形如  $V_0 \times V_1 \times V_2 \times \dots \times V_{n-1} \rightarrow \Omega$  的单值映射, 其中  $V_0, V_1, V_2, \dots, V_{n-1}$  是参数定义域,  $n \in$  自然数集。

记关于  $o$  的一个有穷行为模板集为  $G$ 。对于  $\forall g \in G$ , 记  $Q_g = \{q | q \in \Omega, \exists v_i \in V_i (i=0, 1, \dots, n-1), q = g(v_0, v_1, \dots, v_{n-1})\}$ , 而记  $Q = \bigcup_{g \in G} Q_g$ , 称  $Q$  是  $o$  的行为状态集。

直观地说, 行为模板是构造虚拟实体对象具体行为的框架, 反映了虚拟实体对象的某种行为规律, 表示着该对象的某种行为方式。行为状态则是某段时间上虚拟实体对象具体行为的表征, 它直接产生实体对象的状态迁移。使用同一模板的不同参数设置就可以构造出多个行为状态, 即虚拟实体对象的具体行为。通过重用行为模板, 就能够在不同的虚拟实体对象及 VR 应用之间实现行为的重用。行为建模过程的一项重要工作就是构造虚拟实体对象的有穷行为模板集  $G$ 。

**定义 3** 虚拟实体对象是如下的一个七元组  $\langle E, Y, S, Q, A, s_0, q_0 \rangle$ , 其中,  $E$  和  $Y$  分别是该对象的相关事件集和产生事件集;  $S$  是该对象的非空状态集,  $s_0 \in S$  是初始状态;  $Q$  是该对象的非空行为状态集,  $q_0 \in Q$  是初始行为状态;  $A = \langle \text{Sensor}, \text{Controller}, \text{Performer} \rangle$ , 是该虚拟实体对象的行为机构。

通过  $A$  中三个构件的执行与交互, 虚拟对象将表现出行为, 即: 对各种事件作出相应反应, 产生自身状态的迁移, 向虚拟世界发出事件实现自己的影响。

感受器的主要功能可由抽象的功能函数  $\lambda: T \rightarrow 2^E$  来描述。若对象  $o$  有下式成立:

$$\text{Eset} = \lambda(t), t \in T \quad (1)$$

则  $\text{Eset} \subseteq E$  中含有  $t$  时刻发生  $o$  的所有相关事件, 若  $\text{Eset}$  不空, 它将被转发给 Controller。

Controller 和 Performer 组成一个双层自动机, 这个自动机的执行产生虚拟实体对象状态的迁移, 其输入来自 Sensor 的输出, 而输出就是虚拟实体对象的产生事件。

Controller 处于双层自动机的上层, 可用状态自动机  $\langle 2^E, Q, \delta_c \rangle$  表示, 它的输入来自 Sensor, 状态字母表和输出字母表均为  $Q$ ,  $\delta_c: 2^E \times Q \rightarrow Q$  是下一行为状态函数。则有

$$q' = \delta_c(\text{Eset}, q), \text{Eset} \subseteq E, q, q' \in Q \quad (2)$$

其中,  $q$  是虚拟实体对象  $o$  的当前行为状态,  $q'$  既是它的下一个行为状态, 同时还将作为输出字母输入 Performer。

Performer 处于双层自动机的下层, 可表示为  $\langle Q, Y, S, \delta_p, \lambda_p \rangle$ , 它直接产生  $o$  的状态迁移和双层自动机的输出。它的下一状态函数为  $\delta_p: Q \times S \times T \rightarrow S$ , 输出函数为  $\lambda_p: Q \times S \times T \rightarrow 2Y$ , 则有

$$s' = \delta_p(q, s, t), y = \lambda_p(q, s, t) \quad s, s' \in S, q \in Q, y \subseteq Y, t \in T \quad (3)$$

其中,  $s$  是  $o$  的当前状态,  $s'$  是其下一状态,  $y$  是  $q$  在当前状态  $s$  的输出, 即  $o$  在当前时刻  $t$  产生的事件, 既可能是内部反馈事件, 也可能是向外部输出的事件 (含外部反馈事件)。

Performer 接收 Controller 的输出。当 Controller 没有输出时, Performer 维持原行为状态的执行; 否则, 将根据这个输出改变自己的映射函数, 从而保证执行的总是  $o$  的当前行为状态。Performer 的状态迁移和输出由当前行为状态  $q = \langle \delta, \lambda \rangle$  决定,  $Q$  中所有行为状态的所有输出构成集合  $Y$ 。即任一时刻  $t$  ( $t_0 \in T$  是  $q$  的起始时间), 有下式成立:

$$\begin{aligned} s' &= \delta_p(q, s, t) = \delta(s, t - t_0) \\ y &= \lambda_p(q, s, t) = \lambda(s, t - t_0) \end{aligned} \quad (4)$$

## 4 应用界面

为便于平台开发虚拟实体对象, 基于 SCP 模型开发了一种面向对象的虚拟实体对象建模语言——SCPL。其中, 用于定义虚拟实体对象类及声明虚拟实体对象的程序段的 BNF 描述如下:

$\langle \text{artifactclassdefinition} \rangle ::= \text{ARTIFACTCLASS} \langle \text{artifactclassname} \rangle$

```

{ ATTRIBUTE {<attributes>}
  GEOMETRY {<appearance>}
  [SENSOR {<eventnames>}]
  [PERFORMER {<behaviortemplates>}]
  [CONTROLLER {<eventprocesses>}]
  STARTUP {<startupstate>}}
<artifactdeclaration> ::= ARTIFACT<artifactname>IS-A<artifactclassname> [STARTUP {<
startupstate>}];

```

上述语法规则中大写的是保留字。ATTRIBUTE 部分中除了程序员设定的各种属性外，隐含有属性：位置、转角、速度、角速度向量和包围盒结构。<appearance>描述该类虚拟实体对象在坐标原点处、转角为零时的三维几何形体。<behaviortemplates>是虚拟实体对象的有穷行为模板集。<eventprocesses>给出控制器对各种相关事件的处理过程，并为虚拟实体对象指定新的行为状态。<startupstate>给出该类虚拟实体对象进入虚拟环境时的省缺初始状态和初始行为状态。SCPL 为 SENSOR、CONTROLLER 和 PERFORMER 部分提供省缺实现，以方便程序员描述行为简单的虚拟实体对象。<artifactdeclaration>中可选的<startupstate>使说明的实体对象可具有与类定义的省缺值不同的<startupstate>。下面给出运动的球对象类的定义和一个球的声明。

```

ARTIFACTCLASS ball
{ ATTRIBUTE { float r; }
  GEOMETRY { scale (r, r, r); callobj (SPHERE); }
  SENSOR { collision }
  PERFORMER { move; pause }
  CONTROLLER { collision: {vx=-vx; vy=-vy; vz=-vz; } NEXTBEHAVIOR move (); }
  STARTUP { { x=50 * rand48 (); y=50 * rand48 (); z=50 * rand48 (); r=l;
    vx=rand48 (); vy=rand48 (); vz=rand48 (); getboundingbox ("SPHERE", r);}
  NEXTBEHAVIOR move () } }
ARTIFACT redball IS-A ball;

```

SCPL 程序经过 SCPC 预编译和 CC 编译后，生成可执行代码。运行可执行代码，该程序将分配虚拟世界事件表，生成碰撞检测对象和场景绘制对象，统一全局时钟，初始化各个虚拟实体对象，生成虚拟世界对象表。完成上述初始化工作后，各个虚拟实体对象将在共同的虚拟环境中自主地行为运动。

## 5 结语

本文总结了前人对虚拟实体对象建模方法研究的成果，提出并详细阐述了一种新的虚拟实体对象的行为模型 SCP。SCP 模型从控制系统的角度抽象出虚拟实体对象的行为机构，适用领域广，适合虚拟现实开发平台使用。感受器、控制器和效应器使虚拟实体对象具备很强的自主性、独立性及实现连续和更复杂行为的能力，即使没有事件发生，虚拟实体对象仍能维持某种行为运动。整个控制过程放在对象内部实现，执行效率高，不需要额外系统开销。除了行为模板的可重用性以及面向对象技术固有的虚拟实体对象的可重用性外，SCP 模型还能够很方便地在行为构件中重用前人的成果。按照 SCP 模型构造的虚拟实体对象，活动相对独立，是自备的单元，有统一的通讯机制“事件”，不需要串行执行的系统控制开销，使虚拟实体对象适于并发实现，有利于提高系统的速度和效率。另外，在虚拟实体对象内部，相对独立的三个行为构件，增强了对对象内部的并发性。

## 参考文献

- 1 Gerard Hegron. Motion Control in Animation Simulation Visualization. Computer Graphics Forum, 1989, 8 (5): 347~352
- 2 Philip Lee. Strength Guided Motion. Computer Graphics, 1990, 24 (4): 253~262
- 3 Sang Mah. A Constraint-Based Reasoning Framework for Behavioural Animation. Computer Graphics Forum 1994, 13 (5): 315~324
- 4 James Cremer. Simulation and Scenario Support for Virtual Envrionments. Computer & Graphics 1996, 20 (2): 199~206
- 5 The Virtual Reality Modeling Language. Version 2. 0, ISO/IEC CD 14772, 1996
- 6 Mark Green. Object Modelting Language (OML) Version 1. 1-Programmer's Manual, 1994
- 7 Gurminder Singh. BrickNet: Sharing Object Behaviors on the Net. Virtual Reality Annual Intl. Sympoism, 1995
- 8 Nobert Wiener. Cybernetics or Control and Communication in the Animal and the Machine, 1961
- 9 Lerner A Y. Fundamentals of Cybernetics. Chapman and hall, 1972