

基于下推自动机的仿真模型形式化描述^{*})

方 敏 王宝树 邱素蓉

(西安电子科技大学计算机学院 西安710071)

A Formal Description of the Simulation Model Based on Pushdown Automator

FANG Min WANG Bao-Shu QIU Su-Rong

(Institute of Computer Science, Xidian University, Xi'an 710071, China)

Abstract A particular discussion on the application of Pushdown Automator(PA)in the description of the simulation entity is made in this paper. It first introduces the mathematical model of PA, then it analyzes the conformation method of the simulation nesting entity by graphical interfaces and gives algorithms of using PA to create the model of the simulation entity. In the end, it introduces a formal description of the graphical simulation entities model by using of this algorithms.

Keywords Simulation models, Pushdown automator, Graphic simulation interface

1. 前言

随着计算机仿真的不断发展,人们对仿真软件的要求不断提高,主要表现在两个方面。一方面,用户对软件的功能,仿真对象的规模和复杂度的要求不断提高;另一方面,用户希望对仿真软件的使用尽可能简单、直观,希望能方便地使用计算机并充分发挥计算机系统的功能进行仿真对象的设计。一般情况下,这两方面的要求是相互矛盾的,解决这一矛盾的主要途径就是提供一个描述能力强的可视的交互仿真环境。

交互仿真是将各仿真应用自主地加入或离开整体环境并在此环境中能彼此交互以实现特定的目的。由于在仿真过程中要进行多种模型的仿真,而在模型仿真时需要各种基本数据,如情报侦察系统仿真中雷达系统性能数据、飞机或军舰的平台数据等;同时仿真过程又产生大量的实时数据,如目标分配方案、目标威胁数据等。因此情报侦察仿真系统具有一定的复杂性,设计模型中存在大量的经验系数^[4]。在复杂的情报侦察仿真系统的研究中,如何描述仿真模型、表示平台仿真数据的嵌套关系、准确地抽取设计经验系数已成为研究焦点^[1-5]。本文从仿真对象描述出发,采用自动机形式化描述情报侦察仿真系统的结构模型,将其转化为状态转换矩阵^[2],以元组形式存储于仿真数据库中,并用可视化的方式显示仿真结果。

2. 系统总体描述

2.1 系统体系结构

本文结合面向对象思想提出了一种基于自动机建模的仿真方法。系统在仿真模型库、仿真实体嵌套层次库、仿真经验数据库的基础上建立可视化的仿真界面,其中图形仿真平台提供基本仿真图元,用户拖动仿真图元设计想定的仿真实体对象,系统基于仿真模型库中的规则产生仿真描述,根据仿真描述和仿真对象的参数设定从经验数据库中提取仿真数据产生仿真结果数据,仿真模型描述和结果数据存储在仿真设定库中。系统结构如图1所示。

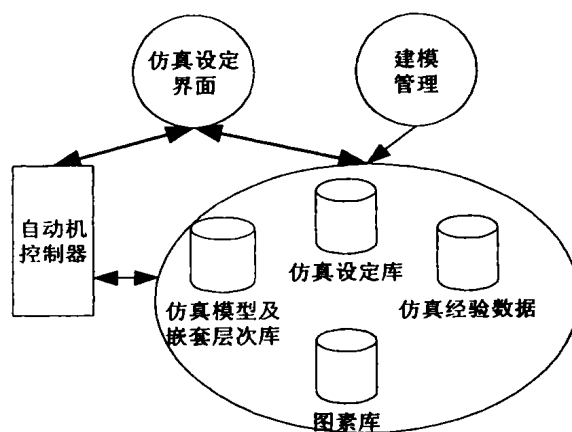


图1 仿真系统体系结构

模型库主要描述了仿真界面提供的基本仿真图素的模型参数;嵌套层次库描述实体间嵌套层次规则,以实体类的继承关系表示仿真实体的嵌套层次,它们是构造自动机状态转换的依据,是用户描述仿真实体模型的基础。仿真实体设计及生成流程如图2所示。

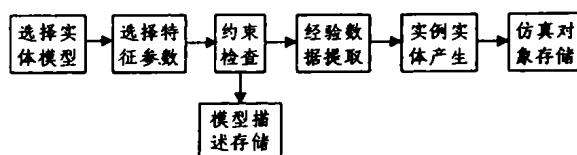


图2 仿真实体设计及生成流程

·每一个仿真场景,可划分成可相互联系并可独立描述的仿真实体。

·每一个仿真实体模型都由一个自动机来描述;

系统程序结构采用客户/服务器模式。数据库服务器中存储仿真模型、仿真实体嵌套层次、基本仿真经验数据、图素库等数据。这些数据可通过建模管理平台加以修改,仿真结果存放到仿真设定库中。

^{*})本文获图像处理与智能控制教育部重点实验室基金项目资助、国防科技预研基金(00J6. 6. 1. DZ0103)资助。

2.2 仿真实体定义

一个实体类型用于表示一类给定的对象类型,它定义为:

(1)实体:实体是仿真系统的基本组成部分,它拥有表示其特征的属性和特有的实体行为。实体还可以由其他一些实体组合而成。

(2)实体属性:实体属性是实体所拥有的能够表征其特性的、并可以被其他实体观察到的特征。它是实体的外在表现之一,如:飞机的速度、航向、位置等信息。属性可以用一组有序变元来表示。实体属性取某一特定值时,称为实体的一个状态。实体状态的集合构成实体的状态空间。

(3)实体行为:行为是实体以自身的活动为中介改变外部环境某状态的过程,是以发出方为中心的概念。行为又可分为主动行为和被动行为,主动行为是指没有直接外部输入而发出的行为,具有目的性;被动行为是指在外输入刺激下发出的行为。实体可分为:

主动实体:具有检测能力,它接收来自其他实体的消息,对消息进行必要的加工和处理后,可能改变自身的状态,并向外发出行为。

被动实体:其本身具有监测其他实体的能力,它接收来自其他实体的消息后,可以改变其自身的状态,但不能向外发出行为。也就是说被动实体只接收消息,但不向外发送消息。

(4)实体引擎:实体引擎是实体外在表现的内在机理的计算机化的表示,是仿真建模术语中的模型,通过对实体本身状态和外部环境映象的加工改变自身的状态,并对其他实体产生影响。

3. 下推自动机

定义1 下推自动机由6元组构成, $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$, 其中:

(1) $Q = \{q_0, q_1, q_2, \dots, q_n\}$ 是非空的有穷状态集合;

(2) $\Sigma = \{a_0, a_1, a_2, \dots, a_n\}$ 是输入字符集合;

(3) $q_0 \in Q$ 是初始状态;

(4) $\Gamma = \{Z_0, Z_1, \dots, Z_n\}$ 是下推栈符号的有限集合,称下推栈字母表。

(5) δ 是 $Q \times (\Sigma \times \Gamma) \rightarrow 2^{Q \times \Gamma^*}$ 映射。

(6) $Z_0 \in \Gamma$, 称 Z_0 为栈底符号;

该自动机采用空栈下推自动机的形式,以空栈作为结束状态,即:

$$L(A) = \{\omega | (q_0, \omega, Z_0) \xrightarrow{*} (p, \epsilon, \epsilon)\}.$$

下推自动机是上下文无关语言的识别器。初始时,自动机处于 q_0 状态,控制器指针指向输入带最左端符号,下推栈的栈底为 Z_0 。在运行过程中,从左到右逐次输入各个字符,该模型在动作的每一步,下推自动机的控制器根据当前状态、输入符号以及下推栈的栈顶符号确定自动机下一个转向状态和栈顶压入字符串,直到控制器的指针指向输入字符串的最右端并且下推栈弹空时,自动机接受该输入字符串。

下面以飞机航迹规划为例,说明利用下推自动机进行仿真建模的过程。其仿真结构如下: $(\{s_1, s_2, s_3, s_4\}, \Sigma, \{\text{初始速率、方位角、俯仰角、起始时间、终止时间、初始位置}\}, \delta, s_1, Z_0)$, 其中 δ 见图3。弧上标记分别为:当前栈顶符号/下推栈弹栈后压入符号/输入符号。图3只给出了一个航迹简单模型的示例,对于复杂的实体模型,表示方式类似。

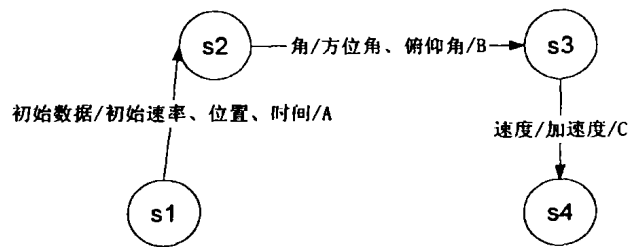


图3 简单航迹模型的下推自动机转换图

4. 图形仿真界面

4.1 仿真实体的嵌套关系

在图形仿真界面环境中,每一个图形对象为一个图元,用户通过对图元的直接操作实现对仿真实体的设置操作,计算机通过图元状态的转换将产生仿真结果。在仿真环境中执行某个操作的过程为:用户点击图元,激活该图元对应的仿真对象,这时,仿真设置界面转去执行该仿真实体所规定的某些参数设置或处理。实体对象设置完毕后,决定其后继的转向状态,并根据各仿真实体间的层次关系自动或手动击活其子实体的图形对象,进行子实体参数设置。因此,每个图元有两个状态:静止状态、运行状态(即激活后的状态)。图形仿真界面交互环境是一个运行系统,其在任何时刻的动作取决于3个因素:当前所处的状态、激活的图元对象及仿真实体配置后的状态转移。图形仿真环境中的一个图元往往呈现为一个树型的继承关系,如图4所示。

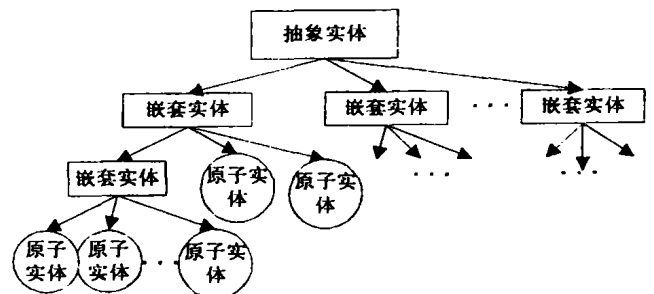


图4 嵌套的仿真实体

4.2 仿真实体的形式化描述

在进行仿真实体设置时,激活的实体图元将产生一个或多个子图元,不同的子图元以及不同层次的图元对应着不同的仿真模型,因此需要根据其运行轨迹来确定它的下一个运行状态^[7]。处在叶结点的仿真实体图元代表一个原子仿真实体,非叶结点图元表示一个嵌套的仿真实体。下面利用文法给出图形仿真实体的形式化定义。

仿真超类描述 $\rightarrow$ Entityinit 状态描述表

状态描述表 $\rightarrow$ 状态描述 | 状态描述 状态描述表

状态描述 $\rightarrow$ attribute_value | 图元描述表

图元描述表 $\rightarrow$ 图元 | 图元 图元描述表

图元 $\rightarrow$ 图元输出函数 | ϵ

图元输出函数 $\rightarrow$ func_name 图元输出函数 | func_name

其中: Entityinit: 仿真界面环境运行的第一个界面状态,对应着抽象实体及仿真环境。attribute_value: 仿真对象原子属性值; inco_name: 基础图元名称,原子仿真对象的图元名称; func_name: 图元的仿真数据输出函数名。

不难找出其中仿真实体的文法描述同下推自动机的对应关系^[3,8]。

仿真实体实现引擎是由下推自动机实现,将图形仿真界面的形式化描述为下推自动机 $A=(Q, \Sigma, \Gamma, \delta, q_0, Z_0)$, 其中:

(1) 在仿真界面上的各个图元的静止状态构成仿真引擎的状态集合 Q ;

(2) 仿真实体特征参数构成实体引擎的输入符号集 Σ ;

(3) 仿真实体模型属性值构成下推符号表 Γ ;

(4) 初始状态 q_0 是仿真环境运行时进入的第一个界面状态;

(5) 不同的图元具有不同的仿真实体的模型参数设置以及不同的图元输出函数。根据自动机的当前状态 s , 栈顶符号 Z , 输入符号 a , 构造状态转换函数 δ 以及图元输出函数 μ 见算法1。

算法1 状态转换函数 δ ;

① 若 $s=q_0, Z=Z_0$ 时, 图元转入执行状态, 自动机将产生一个新的图元标识, 控制器进行抽象实体图元处理; 该图元是仿真实体的超类对象。

② 当 $s=q_i$ 时, 若 $a \in \Sigma$ and $a \in \{q_i \text{ 的特征参数}\}$, 则接受该输入符号 a , 并有 $(q_i, a, Z) \vdash (p, Z_1 Z_2 \cdots Z_m), m \geq 0$ 。表示 q_i 为嵌套实体, p 为 q_i 的一个子实体, Z_i 是 p 的属性标识; 即当 q_i 是嵌套属性状态, 输入字符为 a 时, 下一个状态变为 p , 下推栈弹出栈顶符号 Z , 压入 $Z_1 Z_2 \cdots Z_m$, 输入指针右移一位。

③ 当 $s=q_i$ 时, 若 $a \in \Sigma$ and $a \in \{q_i \text{ 的特征参数}\}$, 则接受该输入符号, 有 $(q_i, a, Z) \vdash (p, \epsilon)$, 其中 q_i 为原子仿真实体, Z 是 q_i 的属性描述。表示完成 Z_i 的设置, 弹出栈顶属性符号 Z_i , 转入下一个状态变为 p , 控制器指针右移一位。

④ 若 $s=q_i, Z=Z$, 输入为 a , 时 δ 无定义, 则停机并拒绝接受输入符号串。

⑤ 若 $s=q_i$, 堆栈为空, 输入符号串为空, 则停机, 接受输入符号串。

输出映射函数 $\mu: Q \times \Gamma \times \Sigma \rightarrow Y$; Y 为仿真实体输出数据集, $y = \mu(s, Z, a)$ 。对于具体的图元, μ 取图元的输出函数, 即 $\mu = \text{func_name}$ 。

4.3 仿真控制器

算法2给出了下推自动机控制器根据特征参数字符串 $\omega = \omega_1 \omega_2 \cdots \omega_n$ 以及下推自动机模型, 实现仿真实体设置过程。

算法2

1) 根据仿真输入获得仿真实体的标识号, 由该标识查找相应的下推栈的栈底符号 Z_0 , 并用 Z_0 初始化下推栈。自动机的起始状态 $q = \text{Entityinit}$ 为仿真环境起始界面状态。

2) 置 $j=1$ 。

3) 根据状态转换函数 $\delta(q_i, \omega_j, z_i \alpha) = \{(q_{i+1}, \gamma \alpha)\}$, 其中 $\gamma, \alpha \in \Gamma^*, z_i \in \Gamma$ 进行状态转换: $(q_i, \omega_j, \omega_{j+1}, \omega_{j+2} \cdots \omega_n, z_i \alpha) \vdash (q_{i+1}, \omega_{j+1}, \omega_{j+2} \cdots \omega_n, \gamma \alpha)$, $j=j+1$ 。

4) 重复3)直到 $(q_i, \epsilon, \epsilon)$ 为止, 则 $\omega = \omega_1 \omega_2 \cdots \omega_n$ 构成了仿真实体 Z_0 的属性描述。 $\{\mu(\omega_1), \dots, \mu(\omega_n)\}$ 为仿真输出集合。

结束语 从以上讨论可知, 用下推自动机可以实现对图形仿真实体的形式化描述。在仿真描述阶段, 可以基于模型设计方法, 利用仿真实体的设计流程、设计模型进行仿真平台的设计。由于仿真系统的复杂性, 设计模型中存在大量的经验系数, 在数据库中存储了以往的设计场景结构数据。自动机的状态转换矩阵能准确地表示平台仿真数据的嵌套关系, 还有助于抽取设计经验系数的选取规律。由于该仿真方法是基于规范的体系描述结构进行的, 有利于模型扩充和变更。

参考文献

- Calvin J O. Data Subscription. In: 12th Workshop on Standards for the Interoperability of Distributed Simulation, March 1995. 12
- Senay H, Ignatius E. A Knowledge-Base System for Visualization Design, 1994, 14(6): 36~47
- 陈崇昕. 形式语言与自动机. 北京邮电出版社, 1988
- 康耀红. 数据融合理论与应用. 西安电子科技大学出版社, 1997
- 王锦, 卿杜政, 欧阳伶俐, 等. 分布式交互仿真技术综述. 系统仿真学报, 1996, 8(3): 1~5
- DIS Steering Committee, The Dis Vision: A Map to the Future of DIS. <http://ftp.sc.ist.ucf.edu/SISO/dis/library/vision.doc>, 1995
- Johnson K, Puckett J. Using Finite State Machines for Behavior Representation with Composable Object Models in NASM[A]. 1999 Spring Simulation Interoperability Workshop
- Hopcroft J E, Ullman J D. Introduction to Automata Theory Languages and Computation. Addison Wesley Publishing Company, 1979
- Some S, Dssouli R. From Scenarios to Timed Automata: building specifications from users requirements
- Whittle J, Schumann J. Generating Statechart Designs From Scenarios
- Arnold A. Finite Transition Systems. Prentice Hall, 1992. 5~45
- Glinz M. An integrated formal model of scenarios based on statecharts. In: 5th European Software Engineering Conf. (ESEC), Sitges, Spain, 1995. 254~271
- Harel D. Statecharts: A visual formalism for complex systems. Science of Computer Programming, 1987, 8: 231~274
- Gehrke T, Firley T. Generative sequence diagrams with textual annotations. In Formal Beschreibungstechniken für verteilte Systeme (FBT99), 1999. 65~72

(上接第158页)

成, 如何在自动合成的过程中发现和消除不一致性, 如何利用 UML 对实时系统自动建模, 以及实现一个基于 UML 的计算机辅助程序设计工具。

参考文献

- Rational Software, Microsoft, Hewlett-Packard, Oracle, Sterling Software, MCI Systemhouse, Unisys, ICON Computing IntelliCorp, i-Logix, IBM, ObjecTime, Platinum Technology, Ptech Tskon, Reich Technologies, Softeam, "UML Summary, Version 1.1", 1997. 7
- 王云, 周伯生. 标准建模语言 UML 简介. 计算机应用研究, 1999, (12): 44~49
- 张幸儿. 计算机编译理论. 科学出版社