国 防 科 技 大 学 学 报
第 21卷 第 1期 JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY Vol. 21 No. 1 1999

面向对象的仿真方法综述*

王维平 赵 雯 朱一凡 华雪倩

(国防科技大学系统工程与数学系 长沙 410073)

摘 要 面向对象仿真方法是当前仿真技术发展的一个重要方面。它是一种框架化、层次化和模块化的建模方法，有利于提高模型的封装性、灵活性、重用性、可扩展性和可维护性，在各个领域得到了广泛的应用。本文介绍基于 OODB的 OOS、基于 DEVS的 OOS和基于 Actor的 OOS等三个主要流派的主要思想和建模仿真方法

关键词 面向对象，仿真建模方法

分类号 TP391.9

Survey on the Object-oriented Simulation Method

Wang Weiping Zhao Wen Zhu Yifan Hua Xueqian

(Department of Systems Engineering and Mathematics, NUDT, Changsha, 410073)

Abstract Object-oriented Simulation (OOS) is an important developing trend in the current simulation community. Owing to the macro-structural hierarchical and modular feature of its modeling method, OOS can improve the encapsulation, flexibility, reusability, extensibility and maintainability of simulation models. It has been put into use in many domains. The authors divide current OOS schools into three categories: OODB based, DEVS based and Actor based, introduce their main thoughts and relative simulation modeling method respectively.

Keywords Object-orientation, simulation modeling method

80年代兴起的面向对象分析和设计方法对仿真技术的发展影响很大，产生了面向对象的仿真方法 (Object-oriented Simulation, OOS)。OOS强调模型的框架化、层次化和模块化，有利于提高模型的封装性、灵活性、重用性、可扩展性和可维护性，在国防、工业、交通运输等各个领域得到了广泛的应用。国外的典型系统有用于队列网络仿真的 PRISM，用于神经网络建模的 SESAME，用于超大规模集成电路系统仿真的 OPERAS等。国内的典型系统有面向对象的多媒体仿真环境 SimStudio，面向对象的连续系统建模仿真支撑环境 ICSL2++ 等。

OOS的产生和发展与面向对象程序设计语言的发展息息相关，但是不能把 OOS等同于面向对象的仿真软件设计方法。具有抽象、封装、继承、多态等特性的仿真语言 SIMULA，是公认的第一个面向对象的程序设计语言，对面向对象程序设计方法的形成产生过很大影响。面向对象仿真的程序设计语言分为两类：一类是比较通用的语言，如 C++，Objective-C，SimPOL，SELF等；另一类是人工智能类型的语言，如 SCOOPS，Scheme，DEVS-Scheme等。后者大多是基于 LISP开发的。近年来，OOS不断从面向对象数据库、人工智能和系统集成化设计方法中吸取精华，朝着集成化、一体化、通用化的方向发展。OOS与多媒体技术的结合产生了面向对象的多媒体仿真方法。面向对象的多媒体仿真环境是 OOS发展中的一个重要里程碑，它强调建模、仿真、表现及交互过程的集成化和一体化，并支持模型的框架技术和模型重用技术，典型系统有 TRAXX，SimStudio等。

面向对象仿真发展到现在，从理论体系上讲主要形成了基于 OODB的 OOS、基于 DEVS的 OOS、基于 Actor的 OOS等三个主要流派。本文介绍这三个流派的主要思想和建模仿真方法。

* 重大试验技术研究项目资助
1998年 10月 5日收稿
第一作者：王维平，1962年生，教授

1 基于 OODB的面向对象建模仿真方法

近年来, 许多学者将面向对象的数据库 (OODB)技术与 OOS结合, 以 OODB为核心设计实现 OOS环境。基于 OODB的仿真环境将对象模型按照一定的结构在 OODB中存储并管理起来, 有利于仿真模型的重用和仿真结果的查询、分析、有利于提供良好、一致的人机交互方式及交互界面, 可以提高仿真建模的效率和模型管理的水平。

1.1 查询驱动仿真 QDS

仿真建模是一项复杂的工作。从信息技术的角度来看, 仿真系统是一种数据、知识和模型的集成化信息系统, 仿真系统的功能就是给用户提供确切、有用的信息, 以及对模型信息的操作和查询功能。查询驱动的仿真建模方法 QDS (ouery Driven Simulation)的基本思想是: 仿真过程是对数据进行输入、输出和处理的过程。仿真得到的数据可以存储在 OODB中管理起来。如果数据库中存有用户所要的数据, 则可通过 QDS系统直接从数据库中检索获得: 如果用户所需的数据无法从数据库中直接获得, QDS系统就会对用户的查询请求进行分析, 找到一个相匹配的、可以实例化的模型, 运行后得到所要的数据。典型 QDS系统有 SimPack, AWESME等。

QDS支持对模型的形式化描述和对模型的组装, 模型存储在 OODB中。QDS系统的软件结构是一个多层次的框架, 框架的底层部分包括编程语言、编译器和数据库管理系统等, 顶层部分包括对象生成器、仓库管理器、模型管理系统、CASE工具等。QDS系统一般有三个数据库, 即应用数据库、字典数据库和字典定义数据库。QDS的数据库结构如图 1所示。

1.2 对象流模型 OFM

对象流模型 (Object Flow Model, OFM) 是一种基于 OODB的仿真模型。OFM 使用面向对象的数据库建立描述系统结构化信息和操作方法的对象模型, 即系统的静态结构: 用对象流图 OFD (Object Flow Diagram) 描述系统的动态行为。OFD是由类节点和进程节点两种节点组成的有向图, 通常采用预测转移网来描述。OFD可以描述对象到达后, 激活、初始化某些进程, 并在进程节点间流动的过程。其中, 类节点代表一类对象或事件, 它们参与 OFD进程中的活动, 并按层次关系放置在 OODB中。每个进程节点都有一个可选的时间延迟属性和一个可选的守卫条件, 后者用来进行条件匹配和条件判断, 选择允许进入进程节点的对象。进程节点和点火条件由输入类节点中的对象数量决定。OFD中定义了进程守卫语言 PGL (ProcessGuard Language) 和进程流动语言 PEL (Process Flow Language), 它们以文本方式给出 OFD进程节点的定义, 描述对象的状态在进入在该节点后所发生的变化。PGL用于判断条件的描述, 它采用标准化的 Select-Join语句来完成以面向对象数据库为中心的对象查询和选择功能, 查询方式既可以是关系型, 也可以是关系-对象型或对象型。PEL是一种概念上比较自然的嵌入式语言, 它使用了消息传递和方法等面向对象的概念, 用来描述一个进程中的各个对象的状态变化过程和动力学机制: 它可以描述输入对象的流动过程, 也可以描述输出对象的产生过程。

附表 QDS系统的数据库结构

Table. The database structure of QDS system.

	元元模型	定义模型
字典模型	元模型	定义数据
字典数据	元数据	应用模型
	数据	应用数据

OFM 建模分为行为、事务和流三个层次。行为层建立对象模型, 并且通过与对象相联的局部函数来描述对象的操作方法。事务层建立对象的事务处理模型, 通过对与某事务相联的进程节点的活动描述给出事务处理的内容。流层通过 OFD描述各个对象模型之间的相互关系及其对对象状态的影响。

2 基于 DEVS的面向对象建模仿真方法

DEVS是美国学者 B. P. Zeigler提出的一种离散事件系统形式化描述体系。它把每个子系统都看作是一个具有独立内部结构和明确 I/O接口的模块, 若干个模块可以通过一定的连接关系组成组合模型, 组合模型可以作为更大的组合模型的元素使用, 从而形成对模型的层次模块化描述。在 DEVS中,

模型的执行是通过抽象仿真器实现的。抽象仿真器是一种算法描述，用以说明怎样将执行指令隐含地传送给模型，从而产生模型的行为。抽象仿真器与模型之间存在一一对应的关系。每个模块或组合模型都有一个与之对应的抽象仿真器，它负责收发消息，调用模块的转移函数，并修改本地的仿真时钟。

DEVS建模方法论作为一种层次、模块化的形式体系，为面向对象仿真的发展提供了强大的推动力。DEVSim++、DEVS-Scheme等仿真环境是基于DEVS理论建立起来的。另外，还有一些研究人员在DEVS的基础上提出了新的面向对象建模方法，并实现了相应的仿真环境。例如层次化的对象网理论和Euler网理论等。

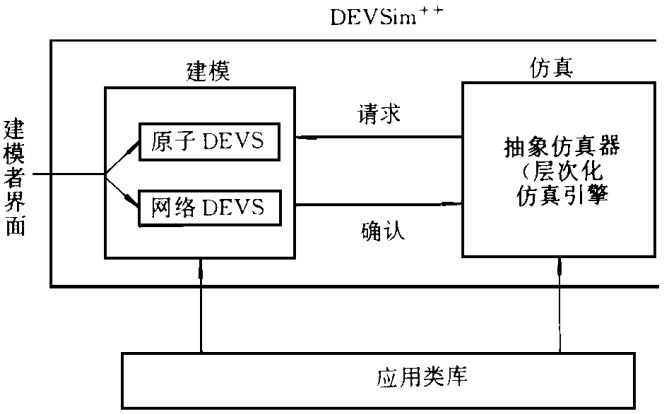

图1 DEVSim++的体系结构

Fig 1 The architecture of DEVSim++

2.1 基于DEVS的仿真环境DEVSim++

DEVSim++是一个采用C++语言实现的基于DEVS的面向对象仿真环境，为建模者提供了各种基于DEVS语义的对象类，实现了抽象仿真器的功能和层次模块化的调度算法。DEVSim++的体系结构如图1所示。

DEVSim++的建模过程完全遵守DEVS规范，并且支持面向对象的建模方法论，具有良好的模型重用性，如图2所示。

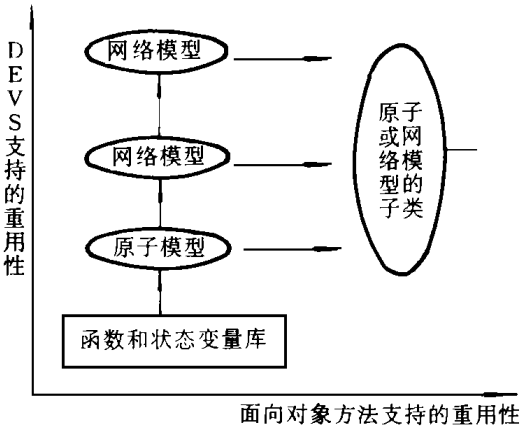

图2 DEVSim++建模方法对模型重用性的支持

Fig 2 DEVSim++ modeling method

2.2 基于DEVS的多层次建模框架CMF

CMF (C++ Modeling Framework)是Douglas A. Porken等人提出的多层次建模框架的C++版本。它采用了DEVS形式体系，但对DEVS做了修改和扩充。CMF与DEVS形式体系的不同之处主要体现在协调器的使用上。

DEVS和CMF都采用协调器来进行层次化模型的仿真时间管理。但是，CMF协调器具有不同于其子类的状态的行为，而且CMF协调器与其子类的行为可以相互发生影响。此外，DEVS和CMF都通过模块化来实现封装特性，通过消息传递来实现通信和同步的功能。但是，DEVS要求模型与仿真器相分离，而CMF中模型与仿真器是不分离的。相比之下DEVS的模块化程度更高一些。

CMF采用分类结构和组装结构两种方法描述模型。分类结构描述系统的类层次结构，组装结构描述模型元素之间的部分-整体关系。典型的分类结构和组装结构如图3所示。

模型类(ModelObject)除了包含主协调器(MasterCoordinator)之外，还至少要包含一个其它子类，它们既可以是原子(Atom)也可以是协调器(Coordinator)。协调器起抽象和聚集各个子类的作用，原子类则处于类结构的最底层。原子可以与协调器通信，协调器可以与父协调器、顶层协调器或原子通信。CMF对协调器进行改造的目的，是使其更适用于不同模型层次上的抽象。

3 基于ACTOR的面向对象建模仿真方法

基于Actor的建模方法是一种生态系统建模方法。生态系统可以看作是由多个并发代理(Agent)进程组成的系统。代理具有适应环境的能力，它可以改变环境或者自身。代理之间相互联系、相互作用，为竞争资源进行各种活动。

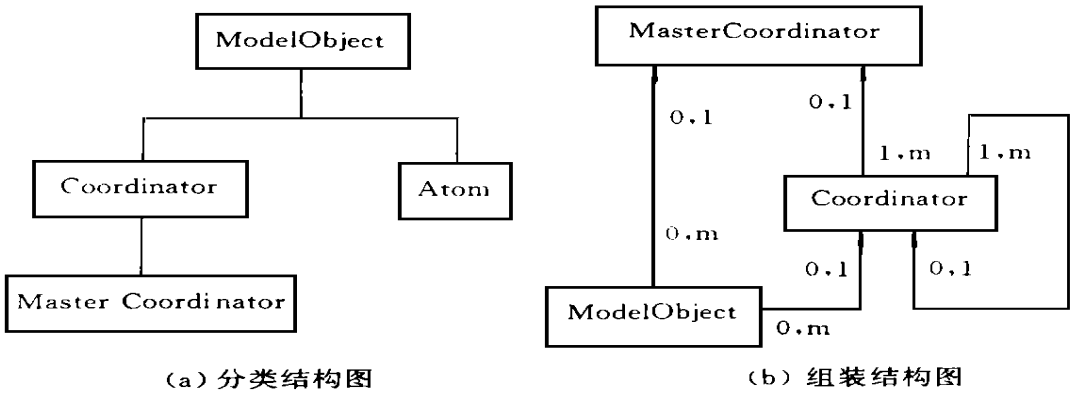

图 3 分类结构图和组装结构图

Fig. 3 Classification structure diagram and composition structure diagram

生态系统思想为面向对象的并发系统仿真建模提供了思路, Actor 反射的 Actor和代理构成了基于 Actor的面向对象建模方法的基本概念体系。Actor是生态系统建模的基本元素, 表示自包含的并发交互的实体。Actor之间通过消息进行交互。Actor之间可以建立反射关系, 反射的 Actor具有与其它 Actor之间因果关系的描述, 可以通过推理完成推断并作出相应的动作。代理是由许多并发的 Actor组成的共同体, 是一个与其他代理有关联的 Actor系统。它可以引发其它代理的活动, 并可以根据对其它代理的认识作出自己的推断。多个代理构成一个生态系。

生态系统模型可以用有向图< A, R> 表示。其中, A 表示代理, 有向边的集合 R 定义了生态系统中各个代理间的关系。两个代理之间的通信路径用三元组< S, D, P> 表示。S是源代理, D 是目的代理, P是负责通信的代理。关系< a1, a2, a3> 表示代理 a1可以通过由代理 a3定义的通信协义与 a2进行交互。

基于 Actor的建模仿真方法可以较好地描述并发系统的各种特性, 如知识表达、并发性、反射性等, 为并发系统的仿真提供了新的思路。目前, 虽然这方面的研究取得了一定的进展, 但是应用的例子都还比较简单。

参考文献

1 Object-oriented Simulation: reusability, adaptability, and maintainability. Edited by G. W. Zobrist, James V. Leonard, 1993

2 G. Birtwisle. Discrete-Event Modeling in SIMULA. New York: Macmillan, 1979

3 D L Eldredge et al. Applying the OO paradigm to discrete event simulations using the C++ language. Simulation, 1990

4 C Landauer. Incorporating simulation in a design environment. Proc. 1991 Winter Simulation Conf

5 J Miller et al. Query driven simulation using active KDL: A functional object-oriented database system. Intl. Journal of Computer Simulation, 1991, 1 (1)

6 T I Oren and B P Zeigler. Concept for advanced simulation methodologies. Simulation, Mar, 1979

7 D P Sanderson. OO modeling using C++. Proc. Annu. Simulation Symp., 1988

8 P Vaughan et al. PRISM: An OO System Modeling Environment in C++. Proc. OOS91, 1991

9 B P Zeigler. Object-Oriented Simulation with Hierachical Modular Models. Orlando, FL: Academic, 1990

10 A Comelio and S B Navathe. Applying active databasemodels for simulation. Proc. 1993 Winter Simulation Conf. Los Angeles, CA, 1993

11 P A Fishwick. SimPack: Getting started with simulation programming in C and C++, Proc. 1992 Winter Simulation Conf, Arlington, VA, Dec. 1992

12 M Hitz, H Werthner, and T I Oren. Employing database for large-scale reuse of simulation models. Proc. 1993 Winter Simulation Conf, Los Angeles, CA, 1993

13 M Lenard. A prototype implementation of amodelmanagement system for discrete-eventmodels. Proc. 1993 Winter Simulation Conf, Los Angeles, CA, 1993 (下转第 36页)

但并非越多越好，必须在性能与成本之间选择一个折衷方案，通过仿真发现。当 N > 6时，“爱国者”雷达网检测性能的改善并不显著；④另外，使用 N-P 判决准则时通过增加组网雷达部数，可以做到既改善系统虚警概率，又提高系统检测性能。

参考文献

1 E Conte, E. D'Addio, A Farina and M Longo. Multistatic radar detection: Synthesis and comparison of optimum and sub-optimum receiver. IEE Proc. Pt F, 1983, 130 (6): 484~494

2 M Barkat and P K Varshney. Decentralized CFAR Signal Detection. IEEE Trans. 1989, AES-25 (2): 141~149

3 胡卫东. 多传感器数据融和的理论与方法研究. [博士学位论文]. 国防科技大学, 1997

(上接第 40页)

14 L F Pollacia and L M L Delcambre. The object flow for discrete event simulation. Intl. Journal of Computer Simulation (Special Issue on Object-Oriented Simulation), G. W. Zobrist, ed., Norwood, N.J. Ablex Publ. Corp., 1995

15 J Rumbaugh et al. Object-Oriented Modeling and Design. Englewood Cliffs, N.J. Prentice-Hall. 1991

16 S Shlaer and S Mellor. Object Life Cycles: Modeling the World in States. N.J. Yourdon Press. 1992

17 A A Tron. Simulation nets, A simulation modeling and validation tool. Simulation, Aug. 1985, 45: 71-75

18 D K Baik. Performance evaluation and optimal decomposition of hierarchical distributed simulation. Ph. D. dissertation, Dep. Comput Sci., Wayne State Univ., Detroit, M.I. 1985

19 A I Conception and B P Zeigler. DEVS formalism: A framework for hierarchical model development. IEEE Tran. Software Eng., vlol. 14, Feb. 1988: 228~241

20 T G Kim and S B Park. The DEVS formalism: Hierarchical modular systems specification in C++. in proc. 1992 European Simulation Multiconf., York, England, 1992: 152~156

21 K E Gorlen, S M Orlow, and P S Plexcin. Data Abstraction and Object-Oriented Programming in C++. Reading, MA: Addison-Wesley, 1990

22 Y Wang. Discrete-event simulation in a massively parallel computer. Ph D dissertation. Dep Elec Eng., Univ. Arizona. Tucson. 1992

23 J R Nichol, C T Wilkes, and F A Manola. Object Orientation in heterogeneous distributed computing systems. IEEE Computer, June 1993: 57~67

24 C J Coomber and R E Childs. A graphical tool for the prototyping of real-time systems. ACM Software Eng. otes Apr. 1990, 15: 70~82

25 G Agha. Actors: A Model of Concurrent Compution in Distributed Systems. Cambridge, MA: M. I. T. Press, 1986

26 D Jefferson. Virtual time, ACM Trans. Programming style and Languages and Syst, 7: 404~425, 1985

27 A Senteni, P Salle, and G Lampalne. Simulation with actors using time-referenced message-passing. in Proc. SCS European Simulation Multiconf, 1989: 109~114