

# 基于关系数据库的仿真模型管理\*

孙郭经 李群 华雪倩 高霞 王维平

(国防科技大学系统工程与数学系 长沙 410073)

**摘要** 本文将讨论仿真模型库管理技术。从仿真模型库管理的实际需求出发,提出以关系代数系统实体结构(RASES)框架为理论依据、利用关系数据库进行模型库管理的方法,并给出模型库管理算法的形式化描述及系统设计。

**关键词** 模型库, 模型库管理系统, 关系数据库, 仿真模型

**分类号** TP391.9

## Research on Model Base Management

Sun Guojing Hua Xueqian Gao Xia Wang Weiping

(Department of Systems Engineering and Mathematics, NU DT, Changsha, 410073)

**Abstract** In this paper, we discuss the method for simulation model base management. To meet the practical demand for simulation model base management, we propose a way of model base management, which is based on the relation algebraic system entity structure (RASES) formalism and relational database. Furthermore, system design and the formal description of model base management algorithm are presented.

**Key words** Model base, Model base management, Relational database, Simulation Model

## 1 概述

### 1.1 模型库管理背景

随着仿真对象日趋复杂及多媒体技术介入使得仿真模型的结构日趋复杂,模型库日趋庞大,管理日亦趋繁琐。因此,需要采用一种新的系统模型管理方法管理庞大复杂的模型库,才能有效地利用仿真模型,简化建模过程。本文将具体讨论这一问题,并提出解决方法和思路。

### 1.2 概念基础

仿真模型包含了模型属性信息、模型组合信息、模型算法信息、模型关联信息等。为了控制建模过程的复杂性,必须对模型信息进行层次化管理。模型层次化过程中,一个复杂的模型可描述成一系列标准组件模型的组合,组件模型具有标准输入/输出端口并通过它进行信息传递,其中的复杂模型本身也可作为组件模型构造更高层次的复杂模型。因此,模型库管理方法中必然涉及到如何利用模型库,如何管理模型,如何组织仿真模型的结构化信息,如何构造仿真模型,如何整合仿真模型。

为解决这些复杂问题,必须建立一种在计算机框架基础之上的高度结构化形式描述体系。众所周知,关系代数系统实体结构(RASES)是建立于系统实体结构形式化描述(SES)和关系代数形式化描述(RA)基础之上的一种高度结构化形式描述体系。系统实体结构的形式化描述体系(RASES)为系统模型的层次化提供了概念基础。层次结构表示模型的结构形式。模型结构本身表述为一系列基于关系代数(RA)的表。这些关系代数表能够在关系数据库中进行存储。更进一步说,这些操作可用关系代数定义,RASES为我们提供了一种系统构建仿真模型的集成方法。这样就大大简化建模过程,使得建模者从繁重的建模细节工作中解脱出来,并且为建模者提供组件模型的特征参数。这些特点有助于建模人员将模型提交仿真器之前校验所建仿真模型的正确性。因此,RASES为模型库管理方法提供

\* 重大试验技术研究项目资助

1998年10月5日收稿

第一作者:孙郭经,1972年生,硕士生

了坚实的概念基础。

### 1.3 模型库管理系统平台

RASES 充分发挥了关系数据库的功能, 克服了传统管理方法的不足, 关系代数系统实体结构框架不仅为大量数据提供了有效的管理方法, 而且为多用户提供了数据共享和快速查询功能。模型库存储了大量共享模型及模型数据, 模型库的管理之中就存在对大量数据的管理问题。关系数据库所提供强大的大量数据管理功能可以大大减轻大量共享模型及大量共享数据所带来管理难度。另外, 模型库管理系统中理应具有的创建、删除、更新功能与数据库管理系统所提供的功能十分相似。因此, 关系数据库是实现“关系代数系统实体结构”管理框架的理想平台。模型库的管理系统在关系数据库基础上, 为用户提供了良好的图形管理接口。在 RASES 框架之上建立的模型库管理系统将所有系统模型实体结构通过“模型结构化操作”转化为实体结构表存储于关系数据库之中。组建系统模型时, 从关系数据库之中提取相应的实体结构表, 并通过“模型修整操作”从模型库之中取出相应模型整合出系统仿真模型。系统仿真模型整合完成后, 建模人员可利用设计好的实验框架在仿真器上对整合后的模型进行实验和表现。

## 2 RASES 形式化描述体系

### 2.1 RASES 形式化定义

$RASES = \langle ASP, SPEC, EVAR, ACOUP, SSEL, GSEL \rangle$

系统实体结构被表示为六元组的形式, 其中各元的含义如下:

**ASP** [ent, asp, subent]: 包含了实体间的分解关系。其中 ent 表示实体, asp 表示一个分解, subent 表示实体 ent 的子实体。ASP 表示了子实体 subent 是实体 ent 在分解 asp 下的分解结果。

**SPEC** [ent, spec, specent]: 包含了实体间的约束关系。其中 ent 表示实体, spec 表示一个约束, specent 表示实体 ent 的约束实体。spec 表示了确定实体 specent 是父实体 ent 在实体约束条件 spec 下的结果。

**EVAR** [ent, variable, value]: 包含了实体结果相关变量 variable。其中 ent 表示变量 variable 相关的实体, value 表示变量 variable 的值。

**ACOUP** [asp, ent1, port1, ent2, port2]: 包含了分解相关耦合。asp 为一分解, ent1 为实体, port1 为实体 ent1 的端口, ent2 为实体, port2 为实体 ent2 的端口。ACOUP 表示了实体 ent1 的端口 port1 与实体 ent2 的端口 port2 在分解 asp 下相连接。

**SSEL** [spec, cond, specent]: 包含了约束的相关选择条件。spec 表示约束, cond 表示条件, specent 表示确定实体。SSEL 表示条件 cond 满足的情况下根据约束 spec 选择了确定实体 specent。

**GSEL** [spec1, specent1, spec2, specent2]: 包含全局选择约束。spec1 和 spec2 为约束, specent1 为约束 spec1 下的确定实体, specent2 为约束 spec2 下的确定实体。GSEL 表示, 如果 specent1 在约束 spec1 下被选定, 那么 specent2 必须在约束 spec2 下被选定。

所有这些关系不依赖于所表示的特殊的实体结构, 因此, 创建实体结构是将数据添入预先定义的关系表之中, 而不用定义新关系表。由于数据库的结构事先知道, 因此关系代数实体结构 RASES 框架的实现相对容易一些。另外, 由于所有实体结构在框架中建立方法相似, 因此很容易理解, 也很容易组合和互换。

### 2.2 RASES 修整操作算法

实体结构图中确定了系统层次结构, 在系统层次结构中每个实体与一组分解边或约束边相关联。修整过程即从初始实体结构中提取纯实体结构。纯实体结构满足以下约束条件: 任意非叶实体与单一的分解边相关联; 任意叶节点实体无分解边相关联; 任意实体不与约束边相关联。纯实体结构中与非叶节点实体相关联的单一分解边表示该非叶节点实体的唯一分解。修整过程主要包含三个步骤: (1) 从实体结构图中按宽度优先法选出与非叶节点实体相关联的可选边(分解边或约束边); (2) 在选定的边(分解边或约束边)上选择与其相关实体; (3) 将选定实体上移并继承其父实体的所有属性(包括: 子

结构、变量、耦合等)。以下在 RASES 框架基础之上利用关系代数描述修整算法来实现修整操作算法。

算法: Prune

输入: ES< ASP, SPEC, EVAR, ACOUP>

输出: PES< ASPpruned, EVARpruned, ACOUPpruned>

步骤:

begin

Step1

Prune\_Select (ES);

Step2

ASPpruned (ASPsel  $\Theta_{sub=it=ent}^{sub=it=ent, spec=ent}$  SPECsel)  $\Theta_{ent=ent}^{ent=ent, spec=ent}$  SPECsel ;

Step3

ACOUPsel  $\pi_{ACOUP} (ACOUP \dot{\cup}_{asp=asp} ASPsel)$ ;

ACOUPpruned (ACOUPsel  $\Theta_{ent=ent}^{ent=ent, spec=ent}$  SPECsel)  $\Theta_{ent=ent}^{ent=ent, spec=ent}$  SPEsel;

Step4

EVARsel  $\pi_{EVAR} (EVAR \dot{\cup}_{ent=ent, ent=subent} ASPsel)$   $\pi_{EVAR} (EVAR \dot{\cup}_{ent=ent, ent=subent} SPECsel)$ ;

EVARpruned  $EVARsel \Theta_{ent=ent}^{ent=ent, spec=ent}$  SPECsel;

end

算法: Prune\_Select (略)

### 2.3 RASES 组合模型过程

在合成过程中，将一个系统模型组成树与模型库中的模型相结合构造系统仿真模型。其步骤如下：(1) 针对系统模型组成树（已修整的系统层次结构图），如果已修整实体结构中的实体在模型库中存在相应的组合模型，则用组合模型的层次结构替代此实体。进行此转化过程直到所有叶节点实体在模型库中具有相应的原子模型。转化后的系统模型组成树成为合成树模。(2) 将合成树合成为仿真模型。对于合成树的每一叶节点实体，从模型库中检取出其相应的原子模型。对于高层实体根据组合信息组织子实体将其合成为高层组合模型。其它相关信息（如变量等）也将转化为恰当的描述，最终完成组合仿真模型。

## 3 基于 RASES 框架的模型库管理系统设计方案

### 3.1 模型库管理系统功能设计

- (1) 增加、修改：允许用户向模型库中增加/删除新的仿真模型。
- (2) 检索、查询：模型库将每一个模型的结构信息、动态特征等分类存储，使得用户可以对模型库中的模型部件进行查询/检索。
- (3) 模型结构化：提供一个基于模型库的结构化算法，将建模环境所建模型转化为树型表示形式（结构化），并将结构化关系信息表存于关系数据库中。
- (4) 替换/裁剪：模型库管理系统支持用户利用模型库中的模型替换模型组合树上的模型，并允许用户对系统仿真模型进行裁剪。

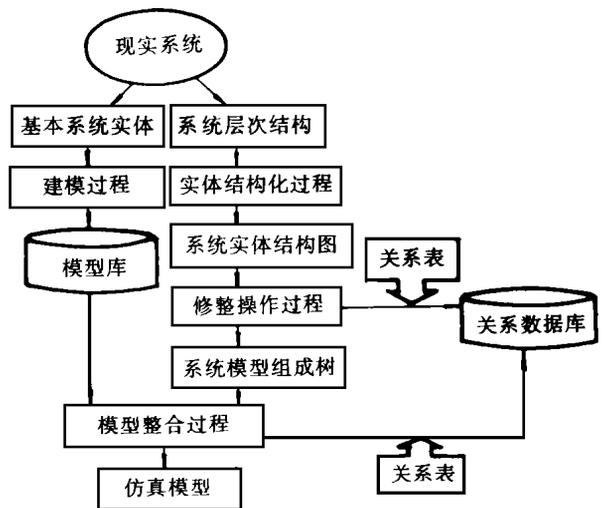


图 1 模型库构建及管理流程

Fig.1 Model Base construction and the flow of management

### 3.2 模型库构建及管理过程

模型库管理方法依赖于系统实体结构如图 1 所示。现实系统的原始子模型作为原子模型存储于仿真模型库之中，原子模型包含了模型算法信息、模型属性信息（声音、图像、动画等）和模型（I/O）接口信息。模型库管理系统通过“结构化过程”将系统层次化，使其转化为系统实体结构图。这种将仿真原子模型构建与系统实体结构构建相分离的方法，可大大减轻建立现实系统仿真模型的复杂程度，并且可依据现有原子模型构建不同结构的系统仿真模型。“修整操作过程”将经过结构化过程的系统实体结构简化，将其转化为模型组成树。“整合操作过程”将模型组成树与模型库相结合，完成仿真模型的组合并送至仿真器进行仿真。

### 3.3 模型库管理框架

模型库管理框架是依据模型库管理的特殊性及相关数据库的管理方法特性提出的。管理框架从系统的角度分为三个层次：接口层、操作层和数据层（如图 2 所示）。

接口层为用户提供了友好的交互界面，用户可轻松地根据系统提示完成系统模型建立与管理。操作层包含了结构化操作、修整操作和整合操作，其中，“结构化操作”完成将系统模型转化为系统结构图；“修整操作”主要完成系统结构图的修整；“整合操作”主要是将已修整的实体结构与模型库中的模型组合，合成仿真模型。

### 3.4 模型库管理系统的设计

模型库管理系统的基础是数据库，它的设计好坏直接影响到管理系统性能与功能。因此，根据系统框架数据层要求，分两部分来设计数据库元件，包括关系数据库设计和模型库设计。

模型库设计主要是根据仿真环境及表现的需要，将仿真模型的媒体素材（声音、动画和图像等）和仿真模型（模型属性、模型算法和模型接口参数等）统一为文件形式进行存储。这样可最大限度支持模型重用，而不会因模型重用带来数据存储量的增加。

关系数据库设计主要针对系统实

体结构与修整后的实体结构，它们以关系数据表的形式存储于关系数据库之中。利用数据库的强大管理功能及强大的数据查询功能，将大大减轻模型管理的难度。以每一个仿真工程对应于一个实体结构库，即实体结构库中包含了仿真工程的初始实体结构表。修整结构库包含了满足纯实体约束条件（如 2.2 节所述）的纯实体结构表。根据 RASES 框架和修整算法给出数据库表（各字段解释如 2.1 节所示）的设计如下：

实体结构库：

ASP [ent, asp, subent], SPEC [ent, spec, specent], EVAR [ent, variable], ACOUP [asp, ent1, port1, ent2, port2], SSEL [spec, cond, specent], GSEL [spec1, specent1, spec2, specent2]

修整结构库：

ASPpruned [ent, asp, subent], ACOUPpruned [asp, ent1, port1, ent2, port2]  
 EVARpruned [ent, variable]

## 4 展望

本文描述了将数据库应用于模型库管理的一种方法。虽然利用关系数据库管理模型库具有建模快

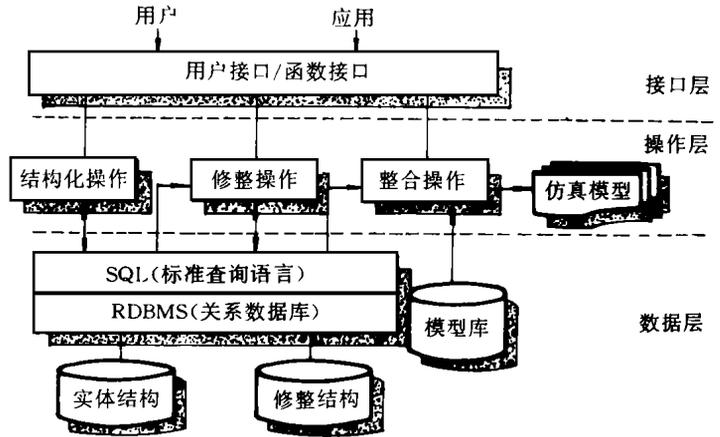


图2 模型库管理框架

Fig. 2 Model Base Management Framework

捷、查询快速等许多优点，但在数据类型、对象间关系和对象间互操作等问题上尚存在不足。随着近年来数据库技术的发展，面向对象数据库具有一些先进特性，如任意数据对象、继承特性和封装特性。利用面向对象数据库开发模型库管理系统将是未来的发展方向。

### 参考文献

- 1 王维平. 离散事件系统建模与仿真. 国防科技大学出版社, 1997
- 2 熊光楞等. 连续系统仿真与离散事件系统仿真. 北京: 清华大学出版社, 1991
- 3 E F Codd. Extending the database relation model to capture more meaning. ACMTans. Database Systems, 1979, 4 (4)
- 4 T T Lee and M Y Lai. A relational algebraic framework for models management, in: proceedings of the 1994 Winter Simulation Conference. Lake Buena Vista, FL 1994: 649 ~ 656

(上接第 61 页)

### 参考文献

- 1 North American Technology And Industrial Base Organization. Collaborative Virtual Prototyping Sector Study. Report May, 1997
- 2 ED Haug. Simulation Based Design for Military System Supportability and Human Factors. Report. October. 1995
- 3 DM SO. Virtual Prototyping-Concept to Production.
- 4 U. Jasnoch, H. Kress. Towards a Virtual Prototyping Environment. IFIP Workshop on VP, 1994
- 5 Stefan Haas. Cooperative Working on Virtual Prototypes. IFIP Workshop on VP, 1994