

# 分布式仿真系统层次设计商空间粒计算模型

陈杰<sup>1,2</sup> 吴狄<sup>1,2,3</sup> 张娟<sup>1,2</sup>

**摘要** 大规模分布式交互仿真系统的开发与应用对仿真系统设计提出了更高的要求. 本文针对分布式仿真系统中存在的设计层次不明确的问题, 提出了基于商空间粒计算理论的系统分层模型, 该模型包括系统总体设计层、系统体系结构设计层和系统执行结构设计层, 解决了系统设计的层次问题. 在此基础上, 对各个层次的纵向和横向信息粒之间的关系进行了深入的分析, 并综合考虑了仿真任务、仿真资源和仿真服务的影响. 最后根据一个实际的仿真系统设计实例对模型进行了验证, 结果表明, 利用分布式仿真系统层次模型进行设计可以极大地提高系统的灵活性和可重用性.

**关键词** 系统设计层次模型, 商空间, 粒计算, 分布式仿真系统

**DOI** 10.3724/SP.J.1004.2010.00923

## Distributed Simulation System Hierarchical Design Model Based on Quotient Space Granular Computation

CHEN Jie<sup>1,2</sup> WU Di<sup>1,2,3</sup> ZHANG Juan<sup>1,2</sup>

**Abstract** To answer the problem of ambiguous design levels for large-scale distributed simulation systems, this paper proposes a hierarchical system model based on the quotient space theory. This model consists of a system global design level, an architecture level, and a execution structure level to solve the problem of diverse levels on system design. Furthermore, we deeply analyze the relationship between the granules of different levels in both horizontal and vertical directions, and integratedly consider simulation tasks, resources and services. Finally, the model is verified based on a real simulation system design example, and the results illustrate that the flexibility and reusability of the system can be improved with the application of the hierarchical design model.

**Key words** Hierarchical system design model, quotient space, granular computation, distributed simulation system

大规模分布式交互仿真系统的开发与应用是应对复杂系统研究的一种重要手段, 并已经在武器系统开发、模拟训练以及混杂系统复杂性研究与实践等方面取得了许多重要的成果. 分布式仿真系统具有分布性、交互性、异构性、时空一致性、开放性等特点. 在系统设计中, 需要在各个层次同时考虑仿真任务、仿真资源、仿真系统运行支持服务之间的协调分配关系, 以保证系统高效运行. 由于系统设计的各个层次不明确, 且各层次之间的关系也很难确定, 使得系统设计存在很多难点, 包括层次关系难以描

述、信息难以聚合和分解等.

目前, 分布式仿真系统的构建主要参照 DIS (Distributed interactive simulation)<sup>[1-2]</sup> 和 HLA (High level architecture)<sup>[3-5]</sup> 标准. 这两个标准主要针对定义一个公共的接口和服务平台, 从而使得大规模的分布式仿真系统可以协调一致地工作, 完成资源共享和仿真任务的实现. 这些标准的建立对于分布式仿真系统的设计和开发具有重要意义, 对于系统构建形成了强约束, 影响着从系统顶级任务划分到系统的具体执行过程的各个层次. 然而这些标准本身却并未在相应的层次中给出具体而完整的形式化设计模型, 因而难于完成系统的可组合和可重用问题.

本文针对以上问题, 将系统设计层次化和分布化的特点与粒计算理论相结合, 给出一个基于商空间理论的分布式仿真系统开发层次模型. 粒计算研究<sup>[6-9]</sup> 对于分布式仿真系统的设计问题具有很强的针对性和适应性. 在这个分布式仿真系统设计粒模型中, 我们将系统的任务、资源和服务都抽象为不同层次的信息粒, 并着重对这些信息粒在横向和纵向上的相互关系进行分析, 为系统的优化设计提供一种层次化的解决方案.

收稿日期 2009-09-11 录用日期 2010-01-27  
Manuscript received September 11, 2009; accepted January 27, 2010

国家重点基础研究发展计划 (973 计划) (97361361), 国家杰出青年科学基金 (60925011) 资助

Supported by National Basic Research Program of China (973 Program) (97361361) and National Science Fund For Distinguished Young Scholars (60925011)

1. 北京理工大学自动化学院 北京 100081 2. 北京理工大学复杂系统智能控制与决策教育部重点实验室 北京 100081 3. 南布列塔尼大学计算机科学与技术系 VALORIA 研究室 瓦纳 法国 56017

1. Department of Automation, Beijing Institute of Technology, 100081, Beijing, P. R. China 2. Education Ministry Key Laboratory of Complex System Intelligent Control and Decision, Beijing Institute of Technology, 100081, Beijing, P. R. China. 3. VALORIA Laboratory, Department of Computer Science and Technology, University of South Brittany, Vannes Cedex 56017, France

## 1 问题分析

分布式仿真系统是一组仿真设备在分布式环境中通过局域网或广域网进行交互的系统。这些设备拥有一致的结构、标准、协议和数据库。分布式仿真系统的结构主要是星形结构，即每个卫星节点负责部分的仿真系统计算，然后通过接收和分发相关的对象进行信息共享。所有的交互信息必须遵守标准中对接口的定义来调用中心节点的服务进行数据交互。随着分布式仿真系统的广泛应用，越来越大的系统规模已经给系统设计带来了许多新的困境，包括：

1) 必须采用逐层深入的设计原则。分布式仿真系统涉及许多分散的子系统，而设计过程不能一次性地考虑实现的全部细节，只能从较抽象的层次开始，然后，一层层地深入到其中的细节，因此设计过程要遵循这种逐层深入的设计原则。

2) 缺乏明确的系统层次设计框架模型。目前，关于分布式仿真系统设计并没有明确的层次设计理论，系统设计往往只针对具体的设计实例，使得系统的可重用性难以得到根本上的提高，因此需要一个分布式仿真系统设计的形式化层次模型，来描述和分析设计的层次性问题。

3) 难以处理不同层次的多种信息。在不同的系统设计层次中，需要将系统信息进行不同程度的聚合与分解，组成不同的信息组合，同时还需要考虑这些信息组合内部和彼此之间的关系。这使得系统设计信息的聚合更加复杂多样，需要考虑新的智能方法来解决。

粒计算理论正是结合人类在解决问题时的信息分层、聚合和分解能力提出的一种信息粒度分析方法，也给解决分布式仿真系统设计问题提供了一个新的思路和理论支持。在此将粒计算理论应用到分布式仿真系统设计中，以达到系统优化设计的目的。

所谓的信息粒度，就是人类在解决处理和存储信息的有限能力上的一种反映，即人类把大量复杂信息按其各自的特征和性能划分成几个较简单的信息块，以方便处理，每个如此划分的信息块就被认为是一个粒<sup>[6]</sup>。

商空间粒计算理论是基于抽象代数中的商集概念，认为概念可以用子集来表示，不同粒度的概念体现为不同粒度的子集，一簇概念构成空间的一个分划—商空间（知识基），不同的概念簇构成不同的商空间。在此，我们采用商空间理论<sup>[9]</sup>作为分布式仿真系统设计的基础理论，主要基于以下原因：

1) 商空间理论侧重信息的分层和描述层次之间的关系，因此符合分布式仿真系统逐层深入的设计原则。

2) 商空间是一整套基于抽象代数的理论，有着

完备的形式化模型，将这些模型应用到分布式仿真系统设计中，可以形成系统设计的形式化模型，从而对系统的优化设计进行求解，得到高效的系统模型。

3) 商空间理论基于人工智能，将信息高度聚合为不同层次的信息粒，并将这些粒内部和粒之间的关系进行讨论，也符合系统设计信息复杂性的要求。

## 2 分布式仿真系统设计分层模型

分布式仿真系统的设计问题是如何利用有限的资源高效地完成系统需求所规定的仿真任务。另外，由于分布式仿真系统自身的特点，同时需要考虑合理地利用标准的交互服务来实现信息共享与交互<sup>[10]</sup>。

而在解决整个系统设计问题时，通常不能一次性地考虑问题的全部细节，如仿真任务、资源和各种服务就很难直接给出所有的具体内容定义和解决方案，这时需要先把问题分解或简化，忽略其中细节。然后，从较抽象的层次开始，一层层深入到其中的细节，并建立起各层次之间的关联关系。这样做不仅符合认识问题的普遍规律，同时也有利于错误发生时间问题的定位和解决，促进了不同层次之间的无缝设计连接<sup>[11]</sup>。

### 2.1 层次化模型

仅仅依靠数学模型，是难以描述大型分布式仿真系统设计问题的，这里我们采用大系统控制论中提出的广义模型论<sup>[12]</sup>建立仿真系统设计的层次化“变粒度”模型。在系统分析、设计与控制过程中，大系统、中系统、小系统所需要的“粒度”不同。一般来说，大系统的需求是宏观的、粗粒度的；中系统是中观的，中粒度的；小系统的需求是微观的、细粒度的。

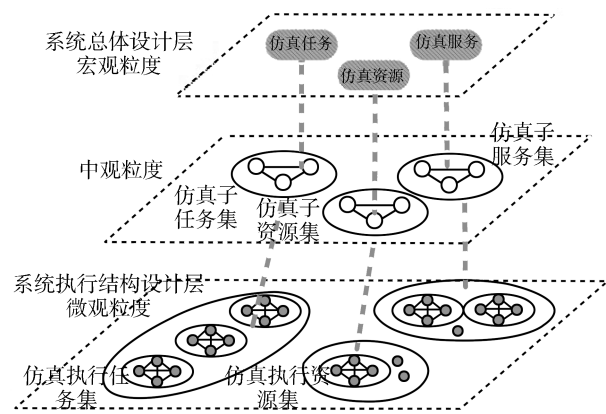


图1 分布式仿真系统粒度层次模型

Fig. 1 Granular hierarchical model of distributed simulation system

遵照分布式仿真系统软件设计的一般过程, 可以将系统设计问题分为如图 1 所示的三个层次, 分别对应系统总体设计层、系统体系结构设计层和系统执行结构设计层. 在这三个层次中, 分别包含仿真任务、仿真资源和仿真服务信息粒, 这些信息粒相互之间存在横向和纵向关系.

### 2.1.1 系统设计的层次

系统总体设计层包含了对于系统的最基本和总体的认识. 系统的总体设计来源于系统需求分析. 这里将需求分为三个主要的类别: 仿真任务主要指系统设计的目标、仿真的功能等; 仿真资源主要是对已有的可用资源及其能力做一个基本的估计; 仿真服务是应用的协议服务标准的一个确定. 将这三个类别抽象为不同的信息粒, 分别用  $[A]$ 、 $[R]$ 、 $[S]$  表示.

进而, 在系统体系结构设计层, 将系统总体设计的各个信息粒度进行细化, 得到体系结构的信息粒度. 体系结构的仿真任务信息粒需要明确仿真子任务, 原则上可以根据不同问题进行不同形式的划分, 如根据任务的不同功能属性 (信息获取、信息传输、信息处理) 等. 仿真资源信息粒在系统体系结构层需要明确可用资源的大体情况, 可以按照其不同的规模或智能水平 (计算机组、智能体等) 对仿真资源进行划分和评价, 组成资源信息粒. 仿真服务信息粒应该与系统体系结构确定的 DIS 或 HLA 标准相一致, 因此可以按照其标准将仿真服务划分为功能组 (仿真管理、数据交互等), 从而组成仿真服务信息粒. 这些信息粒可以用  $\{A_1, A_2, \dots, A_m\}$ ,  $\{R_1, R_2, \dots, R_n\}$ ,  $\{S_1, S_2, \dots, S_k\}$  表示.

在系统执行结构设计层, 所有的仿真任务、资源和服务都应该给出具体的可执行的定义, 并且可以直接应用到系统的编程实现中, 因此这一层的各信息粒更加贴近实际应用, 需要将系统体系结构的设计进一步细分为可执行的仿真任务、资源和服务, 可以分别用  $\{a_1, a_2, a_3, \dots\}$ ,  $\{r_1, r_2, r_3, \dots\}$ ,  $\{s_1, s_2, s_3, \dots\}$  表示.

### 2.1.2 粒度表示模型

商空间信息粒是依据抽象代数中商的概念对信息进行抽象划分的一种方法<sup>[9]</sup>, 是用一个三元组  $(X, f, \mathcal{S})$  来描述一个问题. 其中,  $X$  表示问题的论域,  $f(\cdot)$  表示论域的属性,  $\mathcal{S}$  是论域的结构, 指论域  $X$  中各元素的相互关系. 分析或求解问题  $(X, f, \mathcal{S})$ , 是指对论域  $X$  及其有关的结构、属性进行分析、研究. 设  $X$  代表论域中最细的粒度, 比如仿真运行中具体的某个交互参数. 若对  $X$  进行简化, 产生一个

较大粒度的论域  $[X]$ , 那么就把原问题  $(X, f, \mathcal{S})$ , 变成新层次上的问题  $([X], [f], [\mathcal{S}])$ , 于是得到不同粒度世界的数学模型. 在分布式仿真系统设计中, 我们将不同的粒度层次建立成不同的商空间模型, 模型中包含仿真任务、资源和服务信息粒. 同时考虑到时间和约束的影响, 各层次的分布式仿真系统设计问题可以抽象为一个五元组,

$$P = \langle A, R, S, T, C \rangle \quad (1)$$

其中,  $A$  为仿真任务信息粒;  $R$  为仿真资源信息粒;  $S$  为仿真服务信息粒;  $T$  为时间参数;  $C$  为约束集, 包括:

$C_a$ : 任务约束, 主要表示各仿真任务之间的时间逻辑关系;

$C_r$ : 资源约束, 主要表示仿真资源的能力限制, 以及与任务之间的兼容关系;

$C_s$ : 服务约束, 主要表示仿真服务在系统运行时对服务调用的特定前提;

$C_t$ : 时间约束, 主要表示任务、资源和服务对于时间的特殊要求.

具体来说, 考虑一个简单的海战仿真系统设计问题, 系统设计问题就可以抽象为一个五元组, 如式 (1) 所示. 其中, 仿真任务可以从总体任务的海战仿真分解为潜艇、舰船作战模型、白方观察模型, 在系统执行层, 仿真任务可以分为潜艇射击与毁伤、舰船射击与毁伤、态势观察, 这样, 根据信息粒的抽象原理, 仿真任务的设计关系可以描述为:

$$[A] = \{\text{海战仿真}\} = \{\text{潜艇、舰船、白方}\} = \{\{\text{潜艇射击、潜艇毁伤}\}\{\text{舰船射击、舰船毁伤}\}\{\text{态势观察}\}\}$$

同理, 仿真资源和仿真服务也可以表示为:

$$[R] = \{\text{仿真总体资源}\} = \{\text{计算机组 1、计算机组 2}\} = \{\{\text{计算机 1、计算机 2} \dots\}\{\text{计算机 1}\}\}$$

$$[S] = \{\text{RTI 仿真服务}\} = \{\text{仿真管理、时间管理、交互管理}\} = \{\{\text{加入退出}\}\{\text{推进等待}\}\{\text{发送接收}\}\}$$

时间即为系统时间运行参数;

约束方面, 我们假设: 仿真管理必须被仿真资源调用来加入系统运行.

从这个例子中不难发现, 在系统设计的层次得到了明确的划分, 但是层次之间和层次内部的关系都不清晰, 也使得约束难以在设计中得到充分考虑, 在下面一节中, 将讨论系统设计层次的各种关系.

## 2.2 分层模型层次关系分析

广义模型是知识模型、数学模型、关系模型相互结合的集成模型. 广义模型体系空间中包含的模

型关系主要可以分为纵向关系和横向关系. 在本节中, 主要分析分布式仿真系统层次设计模型的纵向和横向关系.

### 2.2.1 纵向关系

分布式仿真系统不同层次之间的任务信息粒的纵向关系如下式所示. 该式包含了三个主要的设计层次之间的关系, 上层的粒度是由下层粒度聚合而来的, 聚合关系可以根据实际需要选择, 例如仿真任务可以按照其不同功能属性进行聚合 (信息获取、信息传输、信息处理等). 这些信息粒可以根据不同的聚合关系组成半序空间或拓扑空间<sup>[9]</sup>.

$$\underbrace{[A]}_{\text{总体设计宏观粒度}} = \underbrace{A_1, A_2, \dots, A_m}_{\text{体系结构中观粒度}} = \underbrace{\{\{a_{11}, \dots, a_{1i}\}, \{a_{21}, \dots, a_{2j}\}, \dots, \{a_{m1}, \dots, a_{ml}\}\}}_{\text{执行结构微观粒度}} = \underbrace{a_1, a_2, a_3, \dots}_{\text{执行结构微观粒度}} \quad (2)$$

同理, 可得各层次资源信息粒和服务信息粒之间的关系分别如下所示:

$$\underbrace{[R]}_{\text{总体设计宏观粒度}} = \underbrace{R_1, R_2, \dots, R_m}_{\text{体系结构中观粒度}} = \underbrace{\{\{r_{11}, \dots, r_{1i}\}, \{r_{21}, \dots, r_{2j}\}, \dots, \{r_{n1}, \dots, r_{nl}\}\}}_{\text{执行结构微观粒度}} = \underbrace{r_1, r_2, r_3, \dots}_{\text{执行结构微观粒度}} \quad (3)$$

$$\underbrace{[S]}_{\text{总体设计宏观粒度}} = \underbrace{S_1, S_2, \dots, S_m}_{\text{体系结构中观粒度}} = \underbrace{\{\{s_{11}, \dots, s_{1i}\}, \{s_{21}, \dots, s_{2j}\}, \dots, \{s_{k1}, \dots, s_{kl}\}\}}_{\text{执行结构微观粒度}} = \underbrace{s_1, s_2, s_3, \dots}_{\text{执行结构微观粒度}} \quad (4)$$

从信息粒的纵向关系的定性关系来讲, 不同层级的对应信息粒存在定性合取关系, 即上层空间的信息粒变量是下层空间信息粒的“合取”. 其中,  $X$  为  $\{A, R, S\}$ .

$$[X] = X_1 \wedge X_2 \wedge \dots = x_1 \wedge x_2 \wedge \dots \quad (5)$$

而不同层级的对应信息粒并不存在明确的纵向定量关系.

### 2.2.2 横向关系

分布式仿真系统设计模型在不同层次中的横向关系包含两部分内容, 即仿真任务、资源、服务信息粒的内部关系和外部关系.

#### 1) 内部关系

各仿真任务信息粒之间关系根据具体的系统需求而不尽相同, 主要类型包括条件、蕴含、时序和因果关系.

条件关系表示任务之间存在逻辑关系, 例如, 如果任务  $A_i$  必须完成, 则任务  $A_j$  也必须完成, 这个条件关系可以表示为如式 (6) 所示.

$$P(A_i) \rightarrow Q(A_j) \quad (6)$$

其中,  $A_i$  和  $A_j$  为同层仿真任务粒. 如果条件关系中的条件为永真, 则存在蕴含关系, 任务  $A_i$  和  $A_j$  的蕴含关系可以表示为

$$P(A_i) \implies Q(A_j) \quad (7)$$

在横向的任务信息粒中, 时序关系是表示任务的执行存在某种次序关系, 例如由于工艺流程而存在的任务信息粒时序关系可表示为

$$R_t(A) = [A_i(t_1), A_j(t_2), A_k(t_3), \dots] \quad (8)$$

因果关系是表达同层仿真任务信息粒之间存在一定的因果关系, 如下式表示

$$A_i = R_m[A_j] \quad (9)$$

各仿真资源粒之间的关系类型主要包括条件、蕴含和因果关系. 各仿真服务粒之间的关系类型主要包括条件、蕴含、因果、时序等.

#### 2) 外部关系

这里的外部关系主要指同一层次上仿真任务、资源和服务信息粒之间的关系. 分布式仿真系统设计的目的是将不同的仿真任务分配给有限的仿真资源, 同时仿真资源调用标准的仿真服务来完成相应的任务. 因此外部关系主要是处理不同信息粒之间的分配关系. 这种分配关系较纵向关系显得更为复杂, 因此可以用函数关系来表达.

仿真任务-资源的分配关系可以表达为一个分配矩阵, 如下式所示,

$$AR(t) = \begin{bmatrix} A_1 R_1(t) & A_1 R_2(t) & \dots & A_1 R_n(t) \\ A_2 R_1(t) & A_2 R_2(t) & \dots & A_2 R_n(t) \\ \vdots & \vdots & \ddots & \vdots \\ A_m R_1(t) & A_m R_2(t) & \dots & A_m R_n(t) \end{bmatrix}$$

其中,  $t$  是时间参数, 代表系统运行的阶段.  $A_i R_j(t)$  是任务  $i$  对资源  $j$  的分配值, 且  $A_i R_j(t) \in \{0, 1\}$ .

仿真资源-服务的分配关系也可以表达为一个空间叉乘分配矩阵,如下式所示,

$$RS(t) = \begin{bmatrix} R_1S_1(t) & R_1S_2(t) & \cdots & R_1S_k(t) \\ R_2S_1(t) & R_2S_2(t) & \cdots & R_2S_k(t) \\ \vdots & \vdots & \ddots & \vdots \\ R_nS_1(t) & R_nS_2(t) & \cdots & R_nS_k(t) \end{bmatrix}$$

其中,  $R_iS_j(t)$  表示资源  $R_i$  对服务  $S_j$  在时刻  $t$  的调用量值, 其中,  $1 \leq i \leq n, 1 \leq j \leq k$ . 这个决策矩阵的元素之间的关系可以归纳为条件、时序、因果, 表达式参照第 2.1 节.

另外, 为了更直接地指导系统设计开发过程, 这里还需要对外部关系进行适当的约束, 约束条件主要包括任务约束、资源约束、服务约束和时间约束等. 例如, 任务之间可能存在某些逻辑关系, 任务  $A_2$  需要  $A_1$  的数据进行仿真, 因此任务  $A_1$  必须在任务  $A_2$  之前完成, 则可得约束:

$$T_{1\text{end}} \leq T_{2\text{end}} \quad (10)$$

其中

$$\begin{aligned} T_{1\text{end}} &= \max(t), \quad \forall A_1R_i(t) = 1 \\ T_{2\text{start}} &= \min(t), \quad \forall A_2R_i(t) = 1 \end{aligned}$$

由于资源能力的限制, 资源  $R_1$  不能执行任务  $A_1$ , 则有:

$$\forall t > 0, \quad A_1R_1(t) \neq 1 \quad (11)$$

资源调用服务  $S_p$  的前提是其应用过  $S_q$  服务进行声明, 则这种关系表示为

$$\text{if } R_iS_p(t_0) > 0, \quad \text{then } \sum_{t=1}^{t_0} R_iS_q(t) > 0 \quad (12)$$

全系统的时间约束、任务的最早开始时间和最晚结束时间等参数.

这些约束关系与系统设计的实际要求紧密相连, 可以在系统设计过程中进行考虑, 作为系统优化设计的依据.

### 3 建模实例分析

大型分布式仿真系统的设计目的在于提高系统运行效率, 然而在设计过程中, 很难一次性全面考虑问题的所有细节, 因此依据软件系统设计的一般过程, 我们提出了这个分布式仿真系统设计的三层模型. 目前仿真系统设计与开发一般遵照 IEEE 的 DIS 和 HLA 标准. DIS 标准<sup>[2]</sup>指出了 DIS 仿真的基本功能要素, 即仿真实体信息、交互、管理和环境,

但并未给出具体的系统设计和开发实现方法. 在本文提出的分层模型中, 充分考虑了这些要素, 其中, 这些要素之间的关系可以分别作为系统体系结构层建模的约束条件; HLA 标准确定了分布式仿真系统设计中需要考虑的底层交互标准, 而这些标准需要在层次模型中的执行结构层进行考虑, 并结合具体的标准定义为执行结构层选择相应的约束条件, 这些约束条件是系统设计中需要考虑的强约束. 在本节中, 通过对实例的分析介绍具体的设计方法.

#### 3.1 系统总体设计层建模

这里考虑一个实际应用的系统. 这个系统设计要完成一个测试电子信息系统效能的仿真任务, 具有少量可用仿真资源和基于 HLA 标准的仿真服务器.

在系统总体设计层, 明确的仿真任务、资源和信息服务信息粒分别为

$$\begin{aligned} [A] &= \{\text{电子信息系统仿真}\}; \\ [R] &= \{\text{计算机若干组}\}; \\ [S] &= \{\text{HLA 标准服务器}\}. \end{aligned}$$

#### 3.2 系统体系结构层建模

进一步, 在系统体系结构层, 随着设计的深入, 将电子信息系统仿真任务分为信息获取、信息传输和信息处理; 仿真资源存在两个大的计算机组, 各有若干台计算机; 仿真服务可以具体分为仿真管理、时间管理和数据交互三种. 由层次模型中的纵向关系符合合取关系, 因此系统体系结构层的信息粒可以表示为:

$$\begin{aligned} [A] &= \{A_1(\text{信息获取}), A_2(\text{信息传输}), A_3(\text{信息处理})\}; \\ [R] &= \{R_1(\text{计算机组 1}), R_2(\text{计算机组 2})\}; \\ [S] &= \{S_1(\text{仿真管理服务}), S_2(\text{时间管理服务}), S_3(\text{数据交互服务})\}. \end{aligned}$$

在系统体系结构层次上, 信息粒之间的横向关系主要包括:

- 1) 信息获取任务必须在信息传输任务之前结束, 即时序关系:  $R_t(A) = [A1, A2]$ ;
- 2) 两个仿真资源组都要得到充分运用, 即蕴含关系:  $P(R_1) \implies Q(R_2), P(R_2) \implies Q(R_1)$ ;
- 3) 仿真管理服务的使用会导致时间管理服务被调用, 即条件关系:  $P(S_1) \rightarrow Q(S_2)$ ;
- 4) 如果资源承担某项任务, 则必须调用仿真管理服务加入系统运行, 即条件关系  $P(A_iR_j) \rightarrow Q(R_jS_k)$ .

根据仿真系统分配模型和各横向关系, 用随机

取样法, 可求得两个分配矩阵的某可行解为

$$\begin{bmatrix} (-1, -1) & (0, 5) \\ (-1, -1) & (6, 14) \\ (0, 11) & (-1, -1) \end{bmatrix} \quad (13)$$

$$\begin{bmatrix} 0.29 & 0.45 & 0.26 \\ 0.30 & 0.44 & 0.26 \end{bmatrix} \quad (14)$$

其中, 式 (13) 为任务-资源分配矩阵, -1 代表非任务选择资源, 数字代表任务的执行时间段; 式 (14) 为资源-服务分配矩阵, 数值为资源对于服务的占用率.

这个结果给出了满足各种关系限制的一组解, 下面根据这组解继续对系统的执行结构层进行设计.

### 3.3 系统执行结构层建模

在系统执行结构层, 根据软件体系结构设计的结果和软件开发的设计标准, 我们将体系结构和执行结构关联起来. 这样, 3 个仿真任务可以继续细分为 10 个可执行仿真任务, 而仿真资源也由 2 个计算机组细化为 7 台计算机, 仿真服务根据 HLA 平台的规定划分为 7 个服务组. 这个层次的仿真任务、资源和服务信息粒的描述、横向关系及说明如表 1~表 3 所示.

表 1 执行结构层仿真任务描述  
Table 1 Description of the execution level simulation tasks

体系结构层任务粒	执行结构层任务粒	横向关系	说明
A <sub>1</sub>	a <sub>1</sub>	R <sub>1</sub> (A) = [a <sub>1</sub> , a <sub>6</sub> ]	a <sub>1</sub> 为空间信息获取任务, 空间信息通过异构网传输, 且在传输任务之前执行.
	a <sub>2</sub>	R <sub>2</sub> (A) = [a <sub>2</sub> , a <sub>4</sub> ]	a <sub>2</sub> 为对空传感器信息获取任务, 对空信息通过有线网传输, 且在传输任务之前执行.
	a <sub>3</sub>	R <sub>3</sub> (A) = [a <sub>3</sub> , a <sub>4</sub> ∧ a <sub>5</sub> ]	a <sub>3</sub> 为对地传感器信息获取任务, 对地信息通过无线网或有线网传输, 且在传输任务之前执行.
A <sub>2</sub>	a <sub>4</sub>		a <sub>4</sub> 为有线网传输任务.
	a <sub>5</sub>		a <sub>5</sub> 为无线网传输任务.
	a <sub>6</sub>		a <sub>6</sub> 为异构网传输任务.
A <sub>3</sub>	a <sub>7</sub>		a <sub>7</sub> 为空间信息处理任务.
	a <sub>8</sub>		a <sub>8</sub> 为对空信息处理任务.
	a <sub>9</sub>		a <sub>9</sub> 为对地信息处理任务.
	a <sub>10</sub>	R <sub>3</sub> (A) = [a <sub>7</sub> ∧ a <sub>8</sub> ∧ a <sub>9</sub> , a <sub>10</sub> ]	a <sub>10</sub> 为信息融合任务, 其必须在其他类信息获得之后执行.

表 2 执行结构层仿真资源描述  
Table 2 Description of the execution level simulation resources

体系结构层资源粒	执行结构层资源粒	横向关系	说明
R <sub>1</sub>	r <sub>1</sub>		
	r <sub>2</sub>	P(r <sub>1</sub> ∧ r <sub>2</sub> ∧ r <sub>3</sub> ) ⇒ Q(r <sub>4</sub> ∧ r <sub>5</sub> ∧ r <sub>6</sub> ∧ r <sub>7</sub> )	
	r <sub>3</sub>		这里假设所有资源性能相同, 两个仿真资源组都要得到充分使用, 第一组的使
R <sub>2</sub>	r <sub>4</sub>		用使得第二组资源也要使用, 反之亦然.
	r <sub>5</sub>	P(r <sub>4</sub> ∧ r <sub>5</sub> ∧ r <sub>6</sub> ∧ r <sub>7</sub> ) ⇒ Q(r <sub>1</sub> ∧ r <sub>2</sub> ∧ r <sub>3</sub> )	
	r <sub>6</sub>		
	r <sub>7</sub>		

表 3 执行结构层仿真服务描述

Table 3 Description of the execution level simulation services

体系结构层服务粒	执行结构层服务粒	横向关系	说明
S <sub>1</sub>	s <sub>1</sub>	$P(s_1) \rightarrow Q(s_4)$	s <sub>1</sub> 为联邦管理服务, 其使用必须同时申请时间管理服务的使用.
	s <sub>2</sub>	$P(s_2) \rightarrow Q(s_4)$	s <sub>2</sub> 为声明管理服务, 必须同时申请时间管理服务的使用.
	s <sub>3</sub>	$P(s_3) \rightarrow Q(s_4)$	s <sub>3</sub> 为所有权管理服务, 必须同时申请时间管理服务的使用.
S <sub>2</sub>	s <sub>4</sub>		s <sub>4</sub> 为时间管理服务.
S <sub>3</sub>	s <sub>5</sub>	$P(s_5) \rightarrow Q(s_1)$	s <sub>5</sub> 为对象管理服务, 其使用前必须调用联邦管理服务加入仿真运行.
	s <sub>6</sub>	$P(s_6) \rightarrow Q(s_1)$	s <sub>6</sub> 为数据分发管理服务, 其使用前必须调用联邦管理服务加入仿真运行.
	s <sub>7</sub>	$P(s_6) \rightarrow Q(s_1)$	s <sub>7</sub> 为其他管理模式服务, 其使用前必须调用联邦管理服务加入仿真运行.

这样就得到了第三层——系统执行结构层的仿真系统设计模型以及其中信息粒的相互关系, 根据体系结构设计层的分配结果对这个模型进行分配求解得到如下式的结果, 其中式 (15) 为任务-资源分配矩阵, -1 代表非任务选择资源, 数值代表任务执行的时间段; 式 (16) 为资源-服务分配矩阵, 数值为资源对于服务的占用率.

$$\left[ \begin{array}{ccc|cccc} (-1, -1) & (-1, -1) & (-1, -1) & (0, 4) & (-1, -1) & (-1, -1) & (-1, -1) \\ (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (1, 3) \\ (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (2, 5) & (-1, -1) & (-1, -1) \\ (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (6, 9) & (-1, -1) & (-1, -1) \\ (-1, -1) & (-1, -1) & (-1, -1) & (10, 12) & (-1, -1) & (-1, -1) & (-1, -1) \\ (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (7, 14) & (-1, -1) \\ (0, 2) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) \\ (-1, -1) & (5, 8) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) \\ (3, 11) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) \\ (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) & (-1, -1) \end{array} \right] \quad (15)$$

$$\left[ \begin{array}{ccc|cccc} 0.02 & 0.06 & 0.01 & 0.20 & 0.09 & 0.02 & 0.01 \\ 0.02 & 0.04 & 0.03 & 0.15 & 0.03 & 0.04 & 0.01 \\ 0.02 & 0.05 & 0.04 & 0.10 & 0.03 & 0.02 & 0.01 \\ 0.02 & 0.04 & 0.02 & 0.13 & 0.04 & 0.01 & 0.01 \\ 0.02 & 0.03 & 0.01 & 0.16 & 0.05 & 0.02 & 0.01 \\ 0.02 & 0.05 & 0.02 & 0.11 & 0.03 & 0.03 & 0.01 \\ 0.02 & 0.04 & 0.01 & 0.14 & 0.03 & 0.01 & 0.01 \end{array} \right] \quad (16)$$

在系统开发中, 利用最终结果对仿真资源和中心节点的服务进行合理的分配, 从而提高系统运行效率.

## 4 结论

随着分布式仿真系统研究的深入和实践中的广泛应用, 对其系统设计提出了更高的要求. 作为目前分布式仿真系统构建通用准则的 DIS 和 HLA 标准只考虑了系统执行层所提供的服务交互接口, 为目前的系统开发带来了一定的限制, 延长了开发周期. 本文针对分布式仿真系统开发中层次不明确的问题, 提出了基于商空间粒计算理论的系统优化设计分层

模型, 这个模型包括系统总体设计层、体系结构设计层和执行结构设计层, 并深入分析了各个层次之间的信息粒关联关系. 这个层次模型为分布式仿真系统提供了一种可行的形式化设计方法, 使得分布式仿真系统设计层次更加清晰, 提高了系统的灵活性和可重用性, 为实现系统优化设计提供了基础.

## 致谢

非常感谢欧盟 Aisa-Link 项目, 南布列塔尼大学计算机科学与技术系 Flavio Oquendo 教授和北京理工大学模式识别与智能系统的辛斌博士在论文完成过程中提供的帮助.

## References

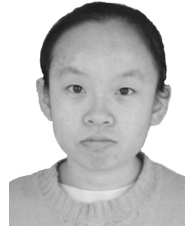
- Jacob S W. DMSO website [Online], available: <http://www.dmsol.com>, December 12, 2009
- IEEE Standard for Distributed Interactive Simulation — Application Protocols, IEEE Standard 1278, 1998
- IEEE Standard for Modeling and Simulation (M and S) High Level Architecture (HLA) — Framework and Rules, IEEE Standard 1516, 2000
- IEEE Standard for Modeling and Simulation (M and S) High Level Architecture (HLA) — Federate Interface Specification, IEEE Standard 1516.1, 2000
- IEEE Standard for Modeling and Simulation (M and S) High Level Architecture (HLA) — Object Model Template (OMT) Specification, IEEE Standard 1516.2, 2000
- Miao Duo-Qian, Wang Guo-Yin, Liu Qing, Lin Zao-Yang, Yao Yi-Yu. *Granular Computing: Past, Present and Future*. Beijing: Science Press, 2007  
(苗夺谦, 王国胤, 刘清, 林早阳, 姚一豫. 粒计算: 过去、现在与展望. 北京: 科学出版社, 2007)
- Zadeh L A. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 1997, **90**(2): 111–127
- Pawlak Z. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Berlin: Springer, 1991

- 9 Zhang Ling, Zhang Bo. *Problem Solution Theory and Application — Quotient Space Granular Computing Theory and Application*. Beijing: Tsinghua University Press, 2007 (张铃, 张钺. 问题求解理论及应用 — 商空间粒度计算理论及应用. 北京: 清华大学出版社, 2007)
- 10 Wu Di, Chen Jie, Flavio O. A formal model-driven engineering approach for composable simulations. *Journal of System Simulation*, 2009, **21**(18): 5608–5613 (吴狄, 陈杰, Flavio O. 一种基于 MDE 的形式化组合建模仿真方法. 系统仿真学报, 2009, **21**(18): 5608–5613)
- 11 Wu Di, Chen Jie, Flavio O, Zhang Juan. The distributed simulation system optimal design model based on quotient space. *Journal of Central South University (Science and Technology)*, 2009, **40**(1): 268–272 (吴狄, 陈杰, Flavio O, 张娟. 基于商空间的分布式仿真系统优化设计模型. 中南大学学报自然科学版, 2009, **40**(1): 268–272)
- 12 Tu Xu-Yan. *Control Theory for Large Scale Systems*. Beijing: National Defense Industry Press, 1994 (涂序彦. 大系统控制论. 北京: 国防工业出版社, 1994)



陈 杰 北京理工大学教授, 博士. 主要研究方向为复杂系统多指标优化与多目标决策、智能控制、约束非线性控制.  
E-mail: chenjie@bit.edu.cn  
(**CHEN Jie** Ph. D., professor at Beijing Institute of Technology. His research interest covers complicated system multi-object optimization and de-

cision, intelligent control, and constrained nonlinear control.)



吴 狄 北京理工大学自动化学院博士研究生. 2004 年获得北京理工大学自动控制系学士学位. 主要研究方向为大型分布式仿真系统、系统优化设计、形式化系统开发理论. 本文通信作者.

E-mail: wudi312858@gmail.com

(**WU Di** Ph. D. candidate at the Institute of Automation, Beijing Institute of Technology. She received her bachelor degree from the Department of Automation, Beijing Institute of Technology in 2004. Her research interest covers large scale distributed simulation system, optimal system design, and formal system development theory. Corresponding author of this paper.)



张 娟 北京理工大学自动化学院副教授, 博士. 主要研究方向为约束非线性控制、分布式仿真.

E-mail: zhjuan@bit.edu.cn

(**ZHANG Juan** Ph. D., associate professor at Beijing Institute of Technology. Her research interest covers constrained nonlinear control and distributed simulation.)