

# 基于 HLA 的超联邦管理的研究与实现<sup>\*</sup>

唐 剑<sup>1,2</sup> 吴 跃<sup>1</sup> 余春东<sup>2</sup> 毛万标<sup>3</sup> 张利强<sup>2</sup>

(电子科技大学计算机科学与工程学院 成都 610054)<sup>1</sup>

(中国科学院软件研究所 北京 100080)<sup>2</sup> (西昌卫星发射中心技术部 四川西昌 615000)<sup>3</sup>

**摘 要** 本文简述了 HLA 架构在分布式系统设计中的重要作用,分析了 HLA 系统的以 FOM 为中心的编程模型的特点及带来的开发、维护和重用问题。本文提出的 HLA 中的超联邦管理方法是在联邦执行的层面上对其进行控制,实现了灵活机动的仿真方式,并成功地解决了上述问题。本文还对 HLA 和流行的场景渲染技术的结合做了有益的尝试,对大型可视化仿真具有一定的参考价值。

**关键词** 仿真,分布式系统,HLA,超联邦,FOM

## Research and Implementation of Super Federation Management Based on HLA

TANG Jian<sup>1,2</sup> WU Yue<sup>1</sup> SHE Chun-Dong<sup>2</sup> MAO Wan-Biao<sup>3</sup> ZHANG Li-Qiang<sup>2</sup>

(College of Computer Science and Engineering,UESTC,Chengdu 610054)<sup>1</sup>

(Institute of Software,Chinese Academy of Sciences,Beijing 100080)<sup>2</sup> (Department of Technology,XSLC,Xichang,Sichuan 615000)<sup>3</sup>

**Abstract** The High Level Architecture (HLA) has a significant effect on the design of the distributed system, as introduced in this paper. The paper analyses the character of the programming model centered as FOM and problems on developing, maintaining and reusing arising with it. Super federation management proposed in this paper is designed to control the federation on the level of execution, realizes the flexibility of simulation and solves the problems mentioned above successfully. In addition, the paper make beneficial attempted to combine the HLA with the popular scene romancing technology, which is valuable to large-scale visualization simulation.

**Keywords** Simulation, Distributed system, HLA, Super federation, FOM

## 1 引言

在系统分析和研究的过程中,系统仿真方法有着极为突出的地位和十分重要的作用。而分布交互式仿真(distributed interactive simulation, DIS)<sup>[1]</sup>的迅速发展,不但在军事部门而且在政府部门和商业领域也得到了广泛应用,显示出巨大的发展潜力与良好的应用前景。但是,随着计算机技术、网络技术和仿真技术的发展,在新的需求的推动下,建立在 IEEE1278-2. x 系列标准上的 DIS 在其体系结构方面显示出一定的局限和不足,难以将更广泛的仿真系统集成到一个综合环境。因此,就需要建立一个新的仿真体系结构,建立一系列的相应标准,便于实现各种类型的仿真系统间的互操作、仿真系统及其部件的重用。从而真正实现将构造仿真、虚拟仿真和实况仿真集成到一个综合环境中,以满足各种类型仿真的需要。于是,用于仿真和建模的高级体系结构(high level architecture, HLA)<sup>[2]</sup>应运而生。

HLA 标准是由美军建模与仿真办公室(DMSO)制定并已在美军中强制执行,目的是促进分布仿真系统的互操作和重用。随着其成为 IEEE 标准,已逐渐为世界各国所认可。在我国,HLA 也已成为仿真领域发展的新趋势。它由 3 个相互关联的部分组成,即 HLA 框架和规则<sup>[3]</sup>、运行支撑结构 RTI<sup>[4]</sup>、对象模型模板 OMT<sup>[5]</sup>。其中,HLA 框架和规则给出了仿真纳入 HLA 体系所必须遵循的规则,共 10 条,目的是

保证仿真应用间可以互操作;运行支撑环境 RTI 为多种类型的仿真之间的交互提供通用服务,是仿真运行的接口规范;对象模型模板是对仿真中的 HLA 对象属性及 HLA 对象间信息交换的格式和内容进行描述、定义的规范,是联邦成员间互操作的“交互协议”。

## 2 HLA 系统的编程模型

在实际开发中,遵循 HLA 标准开发的联邦成员,其重用性和一致性往往难以令人满意。其实这是由 HLA 基本原理决定的。HLA 规定了一组强大的、由 RTI 实现、通过 RTI API 接口访问的仿真服务。在特定的联邦内,联邦成员调用这些服务,可以创建和维持一个复杂的联邦仿真的实现,其中关键是各联邦成员的邦元对象模型(SOM)必须与预先在联邦对象模型(FOM)中规定的对象类和交互类集合相一致。否则,即使各联邦成员都符合 HLA 标准,也难以实现互操作。这是因为 FOM 将以 FED 文件形式为各联邦成员共享,用于初始化 RTI 服务,SOM 中需要互操作的对象类、交互类都需在 FOM 中以公布/订购方式预先定义,并且在各联邦成员中通过编程实现。而联邦成员的开发,特别是与 RTI 接口的编程,是一项极为繁重的工作,也最容易产生错误。首先,掌握、熟悉 RTI 的各种服务并不是一件很轻松的工作,开发、调试分布式仿真的工作量和难度也可想而知。

HLA 的原理决定了系统开发要以 FOM 为中心,美军提

<sup>\*</sup> 中国科学院知识创新工程方向性研究项目基金(KGCX2-JG-09)和总装备部试验技术项目基金(批准号:[2003]装司字第 1164 号)资助项目。  
唐 剑 硕士生,主要研究方向为系统仿真与分布式计算技术;吴 跃 教授,博士生导师,主要研究方向为系统仿真与数据库技术;余春东 博士,副研究员,主要研究方向为分布式计算与仿真技术;毛万标 高级工程师,主要研究方向为数据处理技术;张利强 工程师,主要研究方向为分布式计算技术。

出的 FEDEP(Federation Development and Execution Process) 模型明显反映了这一思想,但这也带来另一方面的问题。当开发系统时,如果对联邦功能即 FOM 进行修改(这往往是难以避免的),即使是细微改变,也可能造成联邦成员程序的大量修改。由于成员的仿真功能与 RTI 接口紧密耦合,也为重用这些功能带来了困难。诚然,HLA 可以支持直接重用,但应明确的是,这是建立在 RTI 与 FOM 一致的基础上的联邦成员级的重用。因此,这种与 RTI 和 FOM 密切相关的开发方式使系统开发复杂度增大,而且大大限制了联邦成员的重用和可移植性。但瑕不掩瑜,HLA 提出的仿真应用模型、仿真支撑平台、公用平台分层设计思想,面向对象开发设计,对象交互与数据通信服务分离的思想,以及实现了不同时间管理算法的混合应用及数据分发管理等思想对建模与仿真的应用与发展都起到了巨大的推动作用,加速了建模与仿真向集成的、协同的全局开发方式的转变。之所以出现上述问题,是联邦成员的仿真实现与 RTI 接口紧密耦合,联邦成员开发缺乏统一的编程模型所致。

### 3 如何解决 HLA 系统的重用性和一致性问题

#### 3.1 相关工作

当前对此问题的解决一般是在 RTI 服务和具体仿真应用之间建立一个类似于软件开发导向的可灵活配置的软件框架,这个框架封装了 RTI 编程的复杂性,通过把 FOM 映射为相应的成员源代码,实现各联邦成员对 RTI 服务调用的重用,提供较为统一的源代码级的编程框架,从而使仿真开发人员更专注于具体的方针问题。基于构件的软件开发技术(Component-Based Development, CBD)的发展<sup>[6~9]</sup>是另一种解决此问题的有效途径。文[10]则在 HLA 仿真开发的模型体系基础上分析各层次模型应遵守的模型规范,采用自动化工具来支持模型的开发可以在一定程度上提高各层次模型的重用性、可靠性和易维护性。但这只降低了成员开发的难度,加快仿真系统的开发速度,并未根本解决开发维护和重用问题。

#### 3.2 超联邦管理技术

HLA 中的超联邦管理技术就是在联邦执行的层面上对其进行控制,实现灵活机动的仿真方式。在 HLA 中,联邦成员可通过订购 MOM 中的交互类来获得联邦执行的信息。但是 MOM 只是支持对本联邦的信息的订阅和反射,对于其它的联邦执行的信息无法获取。这点就限定了一个联邦成员必须加入到一个程序设定的联邦执行中,不能根据情况选择一个联邦执行加入,更不用说在联邦仿真执行间互相切换。而在实际的仿真中存在着这样的情况:即在网络中存在着多个仿真组,加入到哪个组中仿真,需要用户来选择。对此,本文提出“超联邦管理”来解决这个问题,满足网络仿真的要求。

### 4 超联邦管理的分析、设计与实现

HLA 中 MOM(Manager Object Model)包含一些联邦执行内一些预定义的交互类和对象类,比如联邦执行的名字、联邦内联邦成员。这些信息由 RTI 在运行时注册和维护,邦员通过订购这些类,可以获得本次联邦执行的信息。但是,假如在 RTI 上存在多个联邦执行时,对本联邦执行以外的联邦执行的信息是无法获取的。而在仿真中,为了实现灵活操作,存在着多种情况的仿真:我们所用的 RTI 服务器是唯一的,可以进行单人仿真或者多人协作仿真,也可以进行分组仿真。即对于每一个终端来说,可以自己创建联邦执行并加入,也可以探测到 RTI 上运行着的所有联邦执行,并选择一个加入。

这样,就需要我们必须利用 RTI 现有的 API 对 HLA 结构进行扩展,以实现对所有联邦执行信息的管理。

#### 4.1 体系结构

本文采取的技术方案是在 RTI 上创建一个“超”联邦执行,它的功能是专门管理其它联邦执行的信息。每个应用程序中嵌入一个联邦执行维护类,获得自己的所要加入的联邦执行的信息。当程序仿真时,需要到“超联邦执行”上去注册自己所在仿真联邦的信息。并在以后随时保持更新。包含“超”联邦的 HLA 平台结构如图 1 所示。

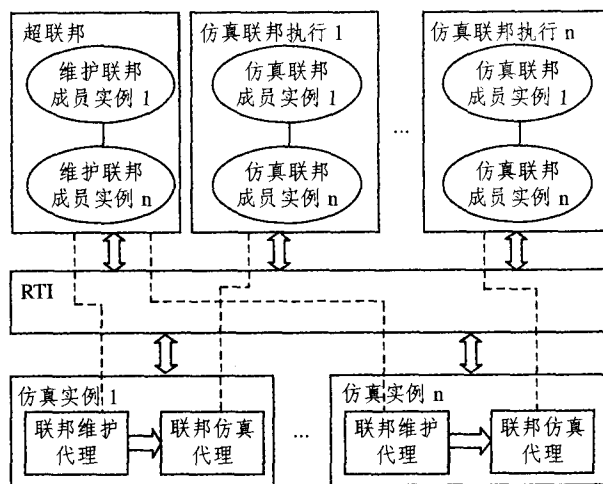


图 1 HLA 平台结构图

在 RTI 上如果存在仿真联邦执行,则必存在唯一的一个超联邦。在每一个仿真实例中,有两种联邦成员对象:唯一的联邦维护代理对象和若干联邦仿真对象。其中,联邦仿真代理负责具体的仿真事务,联邦维护代理负责加入超联邦执行,从联邦仿真代理中获得本仿真实例所在的仿真联邦执行的信息,以更新超联邦执行所维护的所有仿真联邦的信息。

#### 4.2 超联邦管理的实现

在系统中,由于在仿真时可能存在大量的对象。使用大量的对象,会造成很大的存储开销。而在大型仿真系统中,常常使用 3D 渲染引擎,所以开销会更大。被维护代理类所需的仿真代理类的信息可以外部化。基于以上几点考虑,本文利用设计模式中的享元模式 Flyweight 的思想来设计联邦维护代理类和联邦仿真代理类。类设计如图 2 所示。

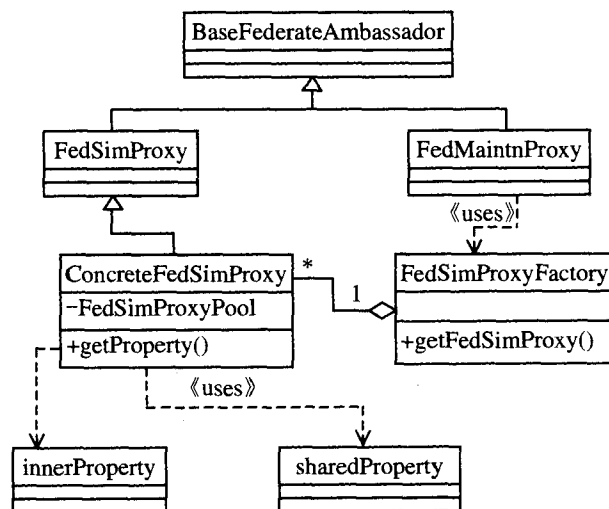


图 2 联邦维护代理类和联邦仿真代理类设计

在此设计模式中, FedMaintnProxy 不能直接将具体的 ConcreteFedSimProxy 实例化, 它必须通过 FedSimProxyFactory 来获得。FedSimProxyFactory 负责维护 FedSimProxy 的内部状态, 当联邦仿真代理请求一个共享的 FedSimProxy 时, FedSimProxyFactory 搜索存在池里面的 FedSimProxy。

程序在初始化时, 创建具体的仿真联邦执行前, FedMaintnProxy 负责探测 RTI 上所运行的联邦执行的信息, 并提供给用户选择: ①创建新的联邦执行; ②选择已经存在的联邦执行, 并加入它和它的联邦成员协同仿真。由用户确定仿真方式后, 运行仿真时, FedMaintnProxy 通过“拉”的方式获得“超联邦执行”上联邦信息对象的所有权, 通过 FedSimProxyFactory 获得具体仿真联邦成员所在的联邦执行的信息, 更新“超联邦执行”上联邦信息对象的属性。

在运行中, 因为各个仿真终端在网络的不同节点, 所以应避免各种因通信问题引起程序的崩溃。例如, 拥有“超联邦执行”上联邦信息对象所有权的仿真终端与网络断开连接, 会引起联邦信息对象无联邦成员占有, 成为“野对象”。此时, 如果有新的程序运行, 它的 FedMaintnProxy 必然会试图获得联邦信息对象的句柄, 虽然此对象是在 RTI 上注册过并存在的, 但是仍会因得不到对象句柄而引起程序错误。为了避免这个错误, 本文在仿真中采用轮询的方式, 即在它的每一次仿真循环中 FedMaintnProxy 调用 attributeOwnershipAcquisitionIfAvailable 查询一下对象的所有权是否可用。如果可用, 则拥有它。这样就保证在任何时刻, 对象总是被某一个 FedMaintnProxy 拥有。

退出时, FedMaintnProxy 再次获得“超联邦执行”上对象的所有权, 更新对象属性中本联邦执行的, 并释放所有权。

## 5 超联邦管理的具体应用

我们将超联邦管理应用到某大型火箭的仿真系统中, 解决了多人的协同训练问题。

实验的硬件环境是: 联想计算机, 内置 3D 显卡, CPU P4 2.6G, 内存 512M。

软件环境是: WindowsNT 平台, MarkRTI, 虚拟仿真平台, DirectX9.0。

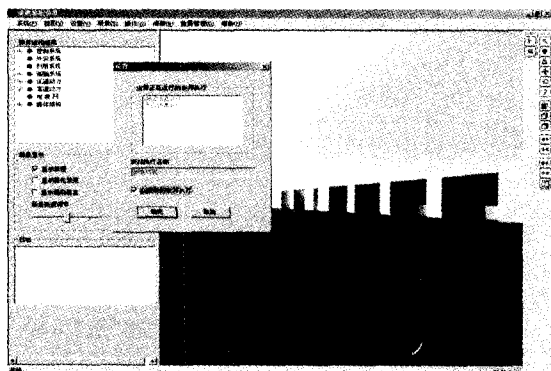


图3 仿真平台界面(客户端3)

初始时, MarkRTI 上存在两个联邦执行: 发动机测试仿真和二级箭体拆装仿真。其中, 测试仿真联邦执行由客户端 1 创建并加入, 拆装仿真联邦执行由客户端 2 创建并加入。当客户端 3 启动程序时, 就会探测到 RTI 上存在的两组联邦仿真执行: 测试仿真和拆装仿真, 两组训练中分别只有一个邦员。此时, 客户端 3 从列表中选择名称为拆装仿真的联邦执

行加入。类似地, 客户端 4 也选择拆装仿真并加入。这样便完成了任务的分组和分配, 并且组之间的信息不会发生干扰, 实现了灵活的分组训练模式。仿真平台的界面如图 3 和 4 所示。

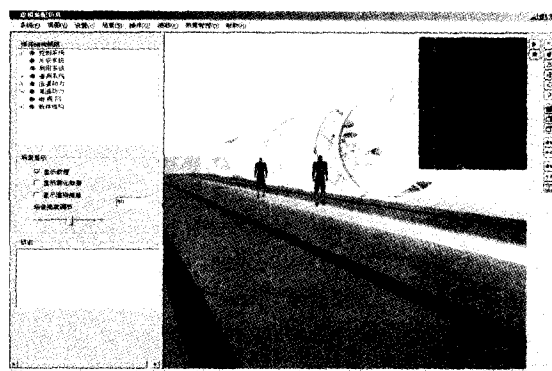


图4 仿真平台界面(客户端2)

图 3 为客户端 3 加入时的界面, 系统会自动探测 RTI 上存在的联邦执行, 并将交互界面提供给用户选择。图 4 为在客户端 2 的仿真平台界面。在本平台上采用的是第一视点, 所以在客户端 2 上看到的是客户端 3, 4 加入后的共同协作的场景。本实验的 3D 渲染场景采用 OGRE 渲染引擎。在网络协作中, 为了保证场景的一致性, 采用比较复杂的数据结构来实现场景同步。但是由于采用了分组训练, 在仿真中大大减少了网络中的数据传输量, 进而减少了网络拥塞, 提高了仿真的效率。实验中各个客户端的 3D 场景是一致的, 画面是比较流畅的, 实验的效果令人比较满意。

**小结** 本文在现有的 HLA 的体系结构上提出了一种用联邦来管理联邦的思想——“超联邦”技术, 并结合 3D 渲染引擎实现了一个初步的仿真实验平台。这是先进仿真技术和流行的场景渲染技术相结合的有益尝试, 对于实现基于 HLA 大型可视化仿真系统具有一定的参考价值。

## 参考文献

- 1 Miller C. The DOD High Level Architecture and The Next Generation of DIS[E]. DIS Workshop Paper 1996-14-115
- 2 DMSO. High Level Architecture Rules [E], Version 1.0, August 15, 1996
- 3 DMSO. Federation Development and Execution Process Model Version 1.5[M]. 1999-12
- 4 DOD. RTI 1.3-Next Generation Programmer's Guide [M]. 1999-4
- 5 DMSO. High level Architecture Object Model Template Specification Version 1.3 [M], 1998-4
- 6 Burg W D, et al. Exploring a Comprehensive CBD Method; Use of CBD/e in Practice [R]. The Third International Workshop on Component-Based Software Engineering, February, 2000
- 7 Kruchten P. Modeling Component System with the Unified Modeling Language[R]. ICSE'98 Workshop, 1998
- 8 Bachmann F. Technical Concepts of Component-Based Software Engineering[R]. CMU/SEI-2000-TR-008, 2000
- 9 Wang Nambor, et al. Overview of the CORBA Component Model-Component-Based Engineering: Putting the Pieces Together. Addison-Wesley, 2001
- 10 王志勇, 李志猛, 凌云翔, 等. HLA 仿真的模型体系和支持工具研究. 计算机仿真, 2005, 22(5): 115~118