

6-14-2018

A Lightweight Modeling and Simulation Technical Framework for Complex Systems

Ji Hang

1.State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing 100854, China;;2. Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing 100854, China;;3. Science and Technology on Space System Simulation Laboratory, Beijing Simulation Center, Beijing 100854, China;

Junhua Zhou

1.State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing 100854, China;;2. Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing 100854, China;;3. Science and Technology on Space System Simulation Laboratory, Beijing Simulation Center, Beijing 100854, China;

Guoqiang Shi

1.State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing 100854, China;;2. Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing 100854, China;;3. Science and Technology on Space System Simulation Laboratory, Beijing Simulation Center, Beijing 100854, China;

Tingyu Lin

1.State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing 100854, China;;2. Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing 100854, China;;3. Science and Technology on Space System Simulation Laboratory, Beijing Simulation Center, Beijing 100854, China;

See next page for additional authors

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Original Article is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

A Lightweight Modeling and Simulation Technical Framework for Complex Systems

Abstract

Abstract: Aiming at the modeling and simulation verification of complex systems, the paper established a lightweight simulation technical framework and a system building method. The simulation technical framework is built based on distributed network communication middleware and has *graphic user interface, simulation management and run control modules*, it *packages effective and lightweight network middleware, object management, event management and time management* and it is capable of interoperability and its members have abilities of reusable and combinable. The results show that the framework can supports flexible construction and effective simulation on complex systems.

Keywords

complex system, simulation framework, complex product, simulation framework

Authors

Ji Hang, Junhua Zhou, Guoqiang Shi, Tingyu Lin, and Junjie Xue

Recommended Citation

Ji Hang, Zhou Junhua, Shi Guoqiang, Lin Tingyu, Xue Junjie. A Lightweight Modeling and Simulation Technical Framework for Complex Systems[J]. Journal of System Simulation, 2018, 30(6): 2013-2019.

一种复杂系统轻量化建模与仿真技术框架

姬杭^{1,2,3}, 周军华^{1,2,3}, 施国强^{1,2,3}, 林廷宇^{1,2,3}, 薛俊杰^{1,2,3}(1. 复杂产品智能制造系统技术国家重点实验室, 北京电子工程总体研究所, 北京 100854; 2. 北京市复杂产品先进制造系统工程技术研究
中心, 北京仿真中心, 北京 100854; 3. 航天系统仿真重点实验室, 北京仿真中心, 北京 100854)

摘要: 针对复杂系统建模与仿真验证, 提出一种轻量级技术框架和系统构建方法, 仿真框架基于分布式网络通信中间件搭建并具有逻辑完善的图形化建模、仿真管理和仿真运行模块。框架封装了高效简洁的网络通信中间件、对象管理、事件管理和时间管理引擎, 具有较强互操作性, 仿真成员开发具有可重用性和可组合性, 能够完成复杂系统中分布式仿真节点的灵活搭建与高效仿真。

关键词: 复杂系统; 建模仿真; 复杂产品; 仿真框架

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2018) 06-2013-07

DOI: 10.16182/j.issn1004731x.joss.201806002

A Lightweight Modeling and Simulation Technical Framework for Complex Systems

Ji Hang^{1,2,3}, Zhou Junhua^{1,2,3}, Shi Guoqiang^{1,2,3}, Lin Tingyu^{1,2,3}, Xue Junjie^{1,2,3}

(1. State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering, Beijing 100854, China; 2. Beijing Complex Product Advanced Manufacturing Engineering Research Center, Beijing Simulation Center, Beijing 100854, China; 3. Science and Technology on Space System Simulation Laboratory, Beijing Simulation Center, Beijing 100854, China)

Abstract: Aiming at the modeling and simulation verification of complex systems, the paper established a lightweight simulation technical framework and a system building method. The simulation technical framework is built based on distributed network communication middleware and has *graphic user interface, simulation management and run control modules*, it packages *effective and lightweight network middleware, object management, event management and time management* and it is capable of interoperability and its members have abilities of reusable and combinable. The results show that the framework can supports flexible construction and effective simulation on complex systems.

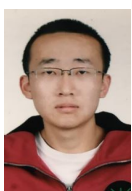
Keywords: complex system; simulation framework; complex product; simulation framework

引言

复杂产品通常是指由机械、电子、控制、液压、气动和软件等系统组成的有机整体, 一般体现出客户需求复杂、产品组成复杂、产品技术复杂、制造过程复杂和项目管理复杂等特点。典型复杂产品系

统包括大型通讯系统、航空航天系统、大型船只、电力网络控制系统、高速列车等。目前以各类CAx(CAD、CAM、CAE等)和DFx(DFA、DFM等)技术为代表的产品建模与仿真技术已在各个领域取得了众多研究成果和成功应用, 但想依靠某一种或一类工具完成这一涉及多领域的复杂产品建模仿真问题是不现实的^[1]。

采用建模与仿真技术在虚拟制造环境中对复杂产品进行设计、加工、装配和操作, 能够使得产品开发走出主要依赖于经验的狭小天地, 发展到全方位预报的新阶段。



收稿日期: 2018-05-05 修回日期: 2018-05-09;

基金项目: 国家重点研发计划(2018YFB1004000);

作者简介: 姬杭(1991-), 男, 陕西靖边, 硕士, 助理, 研究方向为虚拟样机结构设计。

<http://www.china-simulation.com>

• 2013 •

1 相关工作

典型的建模与仿真框架包括分布交互仿真架构(DIS)、高层体系架构(HLA)和 TENA 架构(The Test and Training Enabling Architecture)等。

DIS 通过异构节点间的协议数据单元实现实体状态信息的交互^[2],使得仿真过程能够在时空一致的人机交互环境中并行进行。然而随着需求的发展,DIS 越来越不适应仿真要求:DIS 虽然具有一定的计算机生成兵力(CGF)能力,但未能解决 CGF 的聚合解聚问题;DIS 仅支持无连接的广播方式,一方面可能会造成网络通信拥挤、堵塞,而且接收方必须将所有数据接收,然后进行过滤,制约系统规模,广播方式也不能支持由事件驱动和时间驱动的仿真;DIS 拓展能力有限,包括数据描述方式多样性、数据传输多样性等^[3]。

HLA 通过运行支撑环境、联邦成员大使、对象模型模板和成员接口规范实现联邦仿真和管理^[4],是实现各种仿真应用之间互操作性的通用仿真技术框架。但 HLA 体系结构在使用不同仿真资源时必须针对不同接口设计不同访问过程,为仿真应用开发带来了额外工作和开销,不利于仿真资源重用;HLA 内仿真应用采用串行程序处理方式,导致应用内部计算速度缓慢;同时,HLA 还存在编程规范复杂、上手慢,实现与特定编程语言、系统平台相关,互操作性在仿真应用规模非常大时有限,与商业应用不兼容等劣势。

TENA 具有与 HLA 相似的对等消息传递架构,支持多种标准协议,系统开发力图保持原模拟系统协议和标准,使用中间件技术实现数据交换和时间管理,但其应用领域还主要停留在靶场训练上,对更加复杂的仿真需求还需加入网关等技术来解决。

2 仿真框架概述

本文提出了一种复杂系统轻量化建模与仿真技术框架,能够通过统一的一体化高层建模和仿真验证把复杂系统中不同学科的数学模型、物理模型

和系统模型统一描述,以增强不同学科虚拟样机、实物样机的可重用、互操作和可组合^[5],是解决当前信息化条件下联合仿真诸多问题的有效手段。

复杂系统建模与仿真验证技术框架基于离散事件系统实现,使用图形化建模对仿真对象的种类和接口进行描述,自动生成具有完整运行逻辑、实体管理模块、事件管理模块和时间管理模块的仿真原型系统,框架组成如图 1 所示。

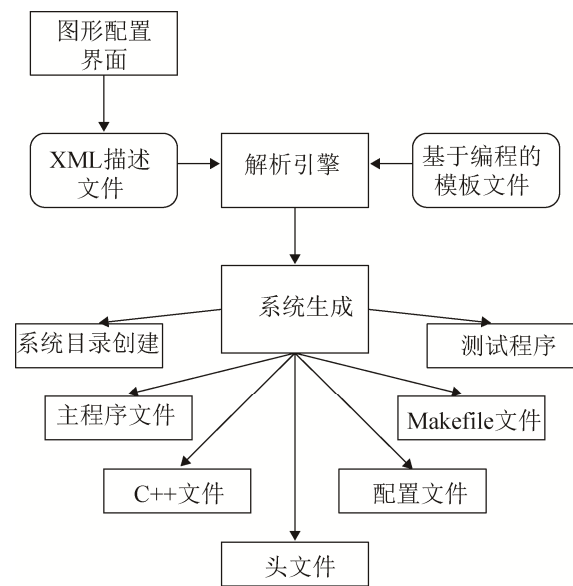


图 1 仿真框架组成

Fig. 1 Simulation Framework Composition

以解析引擎为中心,以可扩展标记语言(XML)描述文件和模板文件为输入文件,结合已经定制的模板,逐一解析生成目标系统文件,XML 文档结构如图 2 所示。XML 作为自动代码生成的描述文件包含直接描述和间接描述两部分内容,直接描述是对用户输入信息的归档,是 XML 文件的主体,用于替换模板中的内容;间接描述用来描述系统的整体架构信息,通过直接描述足够多的信息归档,也可以使系统的整体架构呈现出来。若自动生成的系统需要更改,只需将修改后的 XML 用户文件重新输入到解析引擎中即可得到目标系统,而无需修改源代码。

仿真系统使用高效分布式中间件作为通信核心,分布式中间件技术能够将处于应用层的各应用

元素协同并具有标准统一、代码开源、跨语言、跨平台、服务透明、安全、防火墙穿透、负载均衡、通信屏蔽、分布式和面向对象等特点^[6], 底层基础为远程过程调用(RPC), RPC 通过接口定义语言(IDL)定义目标对象支持的服务, 将界面定义与对象实现相分离; 通过“存根-骨架”模型实现对象信息交互, 存根(Stub)代表客户发出的请求, 框架(Skeleton)将请求交给代理对象实现, 通过存根和框架的调用为静态调用^[7], 对象请求代理规定了分布式对象的定义和接口并实现了这些对象之间的互操作, 在对象请求代理之上定义了公共对象服务, 用来提供对象管理、事件管理和时间管理服务。

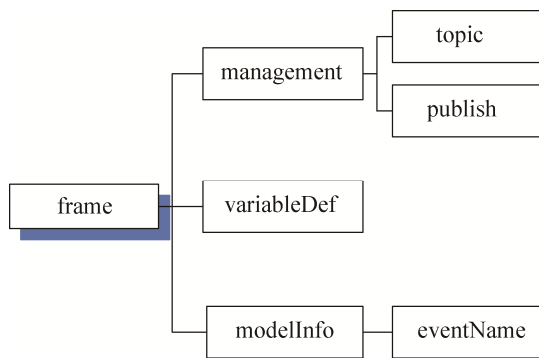


图 2 XML 描述文件结构图
Fig. 2 Structure of XML Description File

3 仿真系统建模

仿真框架的输入是由用户进行图形化建模, 图形化建模步骤如图 3 所示。首先需要声明进行符合接口定义语言规范的通信变量结构, 接口定义语言的优点为跨语言、跨平台, 结果文件可以被编译为任意编程语言或系统平台的通信 API, 实现由独立语言向特定语言的转化, 且这些通信 API 可以实现实时通信, 有利于解决当服务器与客户端使用不同语言开发时遇到的通信协议差异问题; 其次需要声明接口信息, 即事件结构设计; 最后为模型信息, 表示仿真框架中的对象。这种建模方法能够省去对仿真支撑层、通信层和模型实现层的开发, 只需建立包括模型信息和接口信息的描述性框架, 即可自动生成包含联邦描述、模型描述和仿真通信中间件的原型系统框架, 能够便捷快速地搭建复杂系统仿真验证底层支撑环境。

另外, 图形化建模完成了对仿真系统的组成视图描述、接口视图描述、运行视图描述和部署视图描述, 解决了针对不同仿真资源联合仿真信息的一致性与全面描述问题, 各部分描述内容如表 1 所示。



图 3 图形化建模界面
Fig. 3 Graphic User Interface

表1 仿真框架建模的描述项和描述内容
Tab. 1 Description Items and Contents of Framework Modeling

描述项	描述内容
组成视图	系统构成, 系统的层次组合;
接口视图	描述各仿真对象对外提供的实体函数服务、数据对象服务, 以及服务之间的继承、重用等层次关系;
运行视图	定义仿真过程中的周期性事件、非周期性事件、初始化事件、条件中断, 以及仿真对象针对事件的响应;
部署视图	定义各仿真成员服务调用的时间容忍度, 以及超过时间容忍度的处理策略(保持、外推、等待);
部署视图	网络地址设置、通信端口设置。

4 仿真运行管理

4.1 对象管理

对象管理基于 C++ 对象类实现。一个标准的对象类模型由头文件区、通信函数区、非通信函数区、变量定义区和用户填充区组成。

头文件区定义了该模型访问和依赖的标准 C++ 头文件, 包括底层通信支持、接口定义语言、时间管理、离散事件管理必须的头文件, 使得对象类能够在仿真中调用相关依赖库中的方法。

通信函数区定义了对象接收初始化信息、生成远程过程调用句柄、生成本地通信代理和接收仿真推进信号等方法。

非通信函数区包含了与通信无关但在仿真中需要的函数, 即非通信函数, 主要完成本地事件处理、本地仿真推进、未来事件队列生成和内部循环事件处理等功能。

变量定义区即对各对象类添加仿真中必须的变量, 并对这些变量进行初始化和值域规定。

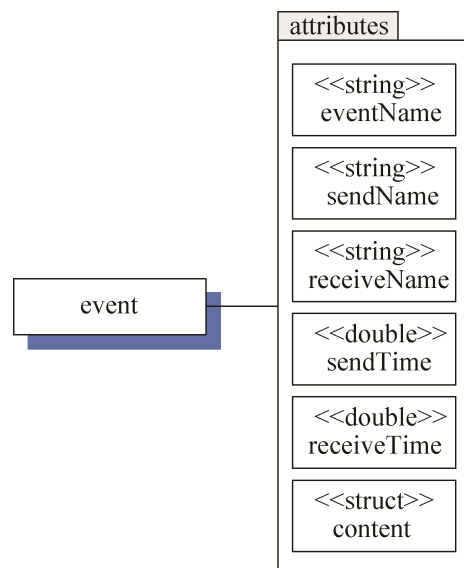
用户填充区的目的在于让用户填充对象类的业务逻辑, 使对象类能够完整的在仿真中运行。其中第一项为实体属性组织, 实体为仿真中的最小单位, 也是实际参与到仿真中的个体, 实体属性的变化与更新是仿真的重要目的和分析仿真结果重要依据。第二项为用户函数定义, 用户可以在这里对仿真中需要的函数进行定义及实现。第三项为对象类初始化区域, 在这里用户需要对该对象类在仿真开始时的状态、实体数量、实体属性初始值进行制定。第四项为事件处理区域, 包括处理该对象类接收到订阅事件时的操作和收到各种内部循环事件

时的操作。

4.2 事件管理

仿真框架基于离散事件的订阅发布运行, 所以事件管理模块主要指事件结构的定义和事件的路由管理。

事件结构如图 4 所示。“sendName”和“receiveName”代表此事件的发送方和接收方;“eventName”为此事件的事件名, 唯一标识此事件且作为订阅发布的关键词;“sendTime”和“receiveTime”为此事件产生的时间和此事件被执行的时间;最后“content”为用户在此事件中包含的事件内容, 在一次仿真中, 内容的结构是唯一的。

图4 离散事件结构
Fig. 4 Structure of Discrete Events

事件的订阅发布路由管理实现了对象类间的通信解耦合, 即任意一个对象类不需要知道哪些对象类会订阅自己发布的事件, 也不需要知道哪些对

对象类会发布自己感兴趣的事件;对象类只需向仿真框架提供自己需要订阅的事件名和自己可以发布的事件名即可;仿真框架会对事件名进行管理,对事件进行路由。所以,仿真框架中任意两个对象类之间都没有对方的网络节点位置信息,做到了通信解耦合,任意一个对象类的加入和离开都不会对系统产生影响,使系统具有可组合性和可重用性。另外,事件的订阅发布引擎仅负责事件名的路由,实际事件交互通过对象类之间的远程过程调用实现,在减轻发布订阅引擎网路负载的同时简化了事件路由拓扑和开发、使用过程。

4.3 时间管理

时间管理算法是仿真框架设计开发的关键,算法需要保证对象类间仿真时间的协调推进和数据的准确运行,提高系统真实性和可靠性。

本文时间管理算法各对象类中均具有当前时间变量。仿真开始时当前时间为 0 s,随后各对象类根据本地物理时间均匀推进,可以选择超实时、实时和欠实时的推进速度;对象类内部有一个发送事件列表,列表中存储着该对象类需要发送的事件,且按照执行时间由近到远排序;每当前时间向前推进一次,对象类会从发送事件列表表头开始寻找执行时间不大于当前时间的事件,发送完毕后等待其他对象类的发布;对象类内部针对每一个订阅事件存储着一个历史事件列表;当接收到某事件时,若此事件执行时间与当前时间差值在容忍范围内,则执行此事件且将事件写入相应历史事件列表中;若此事件执行时间与当前时间差值大于容忍度范围,则将此事件写入相应历史事件列表中,并在列表中通过插值的方法得出此刻数据的估计值,继续运行仿真,触发新的事件存入发送事件列表中,算法流程图如图 5 所示。

时间管理算法中各成员按照本地时间乐观推进,具有简便的开发、运行过程,省去了前瞻、回滚等混合资源仿真系统不需要的环节,同时能够保证复杂系统仿真验证成员时间的准确性。

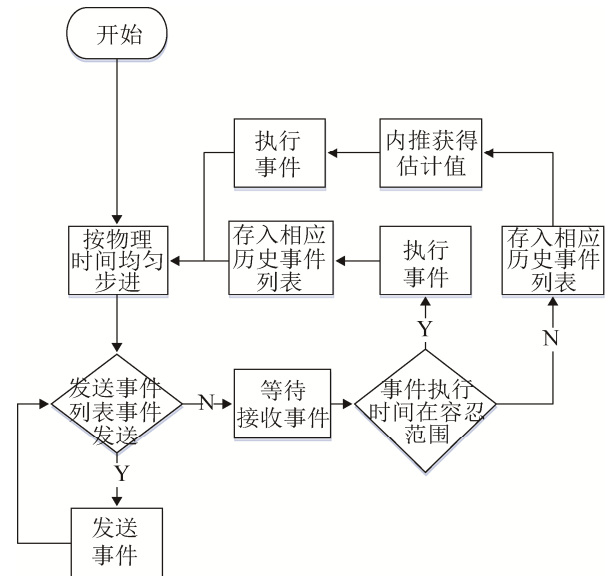


图 5 对象类时间管理流程图

Fig. 5 Flow Chart of Time Management

5 仿真框架效率研究

基于本文仿真框架实现一个简单仿真系统并对其运行效率进行研究。仿真系统建模生成的 XML 结构图如图 6(a)所示,系统运行流程如图 6(b)所示。解析 XML 生成能够分布式处理数据和运行仿真的实体类适配器,每个实体类适配器包含一个实体类和通信 API,实体类中包括了该对象类在仿真中的基本信息,通信 API 中包括了仿真全局网络拓扑,作为远程调用支撑,将每个实体类部署在相应节点,实体类适配器与各节点内模型通过解决方案内部的函数调用实现。

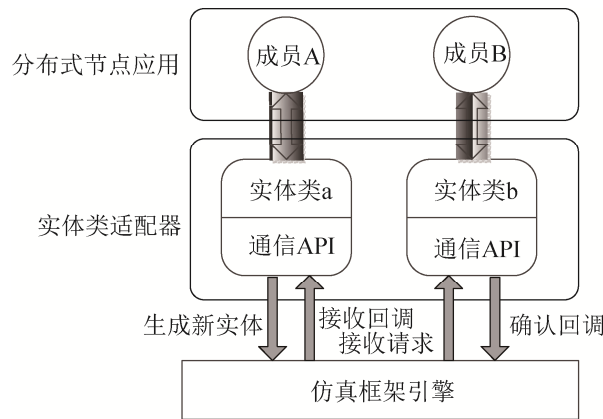
针对以上仿真系统和另一款高效分布式互操作体系结构 HLA,本文设计实验对两种体系架构进行了对比,其中 HLA 实现使用的是由国防科技大学开发的 KD-RTI。实验过程为在两种实现下分别通过发送事件触发生成仿真实体;联邦内包括两个联邦成员,成员 A 向 B 发送事件触发 B 生成一个新实体;生成后新实体随即向 A 返回一个回调事件;每步仿真生成一个新实体;实验目的为通过 500 步仿真,生成 500 个实体,记录仿真时间。两个联邦所在节点的处理器分别为 Intel Core i5-3380M 和 Intel Core2 Duo E8400。

```

<?xml version="1.0" encoding="gb2312" ?>
- <frame>
  <!-- 主题管理器 -->
  - <manager location="-h 127.0.0.1 -p 12337">
    <topic>-h 127.0.0.1 -p 12338</topic>
    <publish>-h 127.0.0.1 -p 12339</publish>
  </manager>
  <!-- ICE变量声明 -->
  <variableDef>struct position { double px; double py; double pz; };</variableDef>
  <variableDef>sequence<position> seqPoint;</variableDef>
  <variableDef>sequence<int> seqInteger;</variableDef>
  <variableDef>sequence<string> seqString;</variableDef>
  <variableDef>struct event { string name; double time; seqString seqStr; seqInteger seqInt; seqPoint seqPt; };</variableDef>
  <!-- 各模型 -->
  - <model name="EnemyMissile" location="-h 127.0.0.1 -p 12340">
    <event name="evTargetInf">HANDLEevTargetInf</event>
    <event name="evScanResult">HANDLEevScanResult</event>
    <event name="evBlockSuc">HANDLEevBlockSuc</event>
  </model>
  - <model name="GuardedTarget" location="-h 127.0.0.1 -p 12341">
    <event name="evAttackSuc">HANDLEevAttackSuc</event>
    <event name="evChangeInf">HANDLEevChangeInf</event>
  </model>
  - <model name="RadarUnit" location="-h 127.0.0.1 -p 12342">
    <event name="evEnemyInf">HANDLEevEnemyInf</event>
  </model>
  - <model name="CommandCenter" location="-h 127.0.0.1 -p 12343">
    <event name="evTargetInf">HANDLEevTargetInf</event>
    <event name="evFireInf">HANDLEevFireInf</event>
    <event name="evFindInf">HANDLEevFindInf</event>
  </model>
  - <model name="FireUnit" location="-h 127.0.0.1 -p 12344">
    <event name="evReload">HANDLEevReload</event>
    <event name="evBlockAllo">HANDLEevBlockAllo</event>
  </model>
  - <model name="BlockMissile" location="-h 127.0.0.1 -p 12345">
    <event name="evLaunchOne">HANDLEevLaunchOne</event>
    <event name="evFindInf">HANDLEevFindInf</event>
    <event name="evAttackSuc">HANDLEevAttackSuc</event>
  </model>
</frame>

```

(a) 模型文件



(b) 系统运行图

图6 基于仿真框架搭建仿真系统的模型文件和系统运行图

Fig. 6 Modeling File and Operation Diagram of the System Built by Simulation Framework

通过图7结果看出,使用本文仿真框架的实体生成速度明显快于KD-RTI,并且在500个实体范围内速度基本保持不变,分析原因为本文框架所设

计的轻量级运行引擎具有灵活的通信机制、强大的服务支持和高效的运行管理逻辑。另外,本文仿真框架相比HLA具有更加简洁的仿真系统开发流

程, 仿真成员开发简单、上手快, 并支持广域网通信, 而目前市场上实现的 HLA/RTI 产品大多只能面向局域网通信。

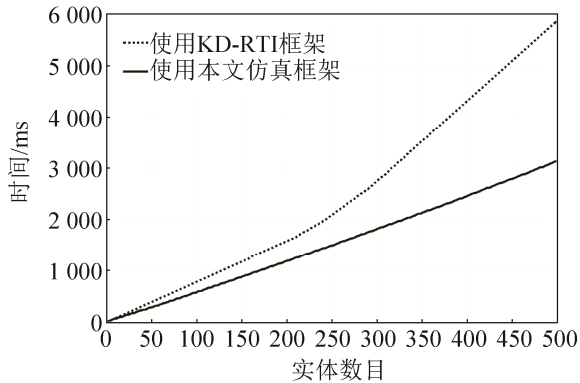


图7 两种框架生成实体速度效率对比

Fig. 7 Entity Generation Efficiency Comparison between KD-RTI and Our Simulation Framework

6 结论

本文提出了一种复杂系统轻量化建模与仿真验证技术框架。框架基于分布式网络通信中间件并具有逻辑完善的图形化建模、对象管理、事件管理和时间管理模块, 具有底层通信、系统接口和系统实现相分离的特点, 具有互操作性强、可重用性强和可组合性强的特点, 能够完成复杂系统搭建任务, 框架运行速度快于同等通信量级下的 HLA 仿真系统, 并且具有更简洁的建模运行流程和支持广域网仿真。

参考文献:

- [1] 李伯虎, 柴旭东. 复杂产品虚拟样机工程[J]. 计算机集成制造系统, 2002, 8(9): 678-683.
Li Bohu, Chai Xudong. Virtual Prototyping Engineering for Complex Product[J]. Computer Integrated Manufacturing Systems 2002, 8(9): 678-683.
- [2] IEEE Computer Society. IEEE Standard for Distributed Interactive Simulation Communication Service and Profiles[S].
- [3] 肖田元. 系统仿真导论[M]. 北京: 清华大学出版社, 2009.
- [4] 郝建国. 高层体系结构中的多联邦互联技术研究与应用[D]. 长沙: 国防科学技术大学, 2003.
Hao Jianguo. Research and Implementation of Multi federated Interconnection Technology in High Level Architecture [D]. Changsha: National University of Defense Science and Technology, 2003.
- [5] 周玉芳. LVC 仿真技术综述[J]. 指挥控制与仿真, 2010, 32(4): 1-7.
Zhou Yufang. A Summary of LVC Simulation Technology[J]. Command Control and Simulation, 2010, 32(4): 1-7.
- [6] 张云勇. 中间件技术原理与应用[M]. 北京: 清华大学出版社, 2016.
Zhang Yunyong. The Principle and Application of Middleware Technology[M]. Beijing: Tsinghua University Press, 2016.
- [7] Andreas Koerner, Stefanie Winkler, Felix Breitenecker. Benchmarking Simulation Models for Dynamic Hybrid Systems[C]// Computer Modelling & Simulation, 2017 UKSim-AMSS 19th International Conference on. 2017: 91-96.
- [8] Yuanli Wu. Using CORBA as a distributed extension solution for OSGi[J]. IEEE Workshop on Electronics, Computer and Applications. 2014: 178-181.
- [9] Zhang Haibo. Research and development of distributed data integration query system based on CORBA[J]. Innovative Smart Grid Technologies-asia, 2012: 1-4.
- [10] Stefanie Winkler, Andreas Körner. A Comparison of Different Modelling and Simulation Approaches for Hybrid Dynamical Systems[C]// Computer Modelling & Simulation, 2017 UKSim-AMSS 19th International Conference on, 2017: 97-102.
- [11] Song Xiao, Zhang Lin. A DEVS Based Modelling and Methodology-COSIM[J]. APPLIED MATHEMATICS & INFORMATION SCIENCES (S2325-0399), 2012, 6(2): 417-423.
- [12] Richard M Fujimoto, Christopher Carothers, Alois Ferscha. Computational challenges in modeling & simulation of complex systems[J]. Simulation Conference (WSC), 2017 Winter (S1558-4305), 2017: 431-445.