

High Level Architecture

HLA

周彦 戴剑伟

主编

蒋晓原

主审

飞思科技产品研发中心

监制

仿真程序设计



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

7P391.9

278

HLA 仿真程序设计

周 彦 戴剑伟 主编

蒋晓原 主审

飞思科技产品研发中心 监制

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书围绕分布交互仿真的高层体系结构 (HLA), 系统、全面地介绍了 HLA 的基本原理和基于 HLA 的仿真程序设计方法。全书共计 12 章。前 5 章结合 DMSO HLA 1.3, 系统介绍了 HLA 中六大管理功能域和管理对象模型 (MOM) 的基本原理、对象模型模板 (OMT)、联邦开发和执行过程模型 (FEDEP) 以及联邦运行支撑环境 (RTI) 等主要内容。后 7 章结合一个具体的仿真实例, 全面分析了 HLA 中联邦管理、声明管理、对象管理、时间管理、所有权管理和数据分发管理的编程方法以及主要 RTI 接口函数的编程应用。全书注重理论与实践的结合, 可操作性强。

本书体系完整, 内容实用、覆盖面广, 可作为相关专业研究生和高年级本科生的教材, 也可以作为广大工程技术人员的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 翻版必究。

图书在版编目 (CIP) 数据

HLA 仿真程序设计 / 周彦等编著. —北京: 电子工业出版社, 2002.6

ISBN 7-5053-6708-0

I .H... II.周... III.计算机仿真—程序设计 IV.TP391.9

中国版本图书馆 CIP 数据核字 (2002) 第 036909 号

责任编辑: 郭 晶 刘韦韦

印 刷: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 30 字数: 768 千字

版 次: 2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

印 数: 5 000 册 定价: 45.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077

前言

近年来,随着计算机技术、信息技术和系统技术的飞速发展,计算机仿真的应用领域不断拓宽,国民经济发展特别是军事需求的强大推动,促使计算机仿真在理论和实践两方面都走上了快速发展的道路。当前,计算机仿真已广泛应用于航空、航天、通信、船舶、交通运输、军事、化工、生物、医学、社会经济系统等自然科学与社会科学的各个领域,其重要性已广为人知。随着计算机仿真应用领域的拓宽,计算机仿真需要解决的问题也越来越复杂,许多问题靠单个仿真系统已无法解决,必须依靠多个仿真系统进行联合协同仿真,因此,分布仿真已成为计算机仿真领域的前沿和热点之一。

与传统的单个系统仿真相比,分布仿真的关键问题是多个仿真系统间的互操作问题,为此,美国国防领域的相关组织通过大量的研究,提出了建模与仿真的高层体系结构(High Level Architecture,简称HLA),HLA的核心思想是互操作和重用,其显著特点是通过运行支撑环境RTI(Run-Time Infrastructure),提供通用的、相对独立的支撑服务程序,将仿真应用同底层的支撑环境分开,即将具体的仿真功能实现、仿真运行管理和底层通信传输三者分离,隐蔽了各自的实现细节,从而使各部分可以相对独立地进行开发,并能充分利用各自领域的先进技术。相对于早期的DIS标准,HLA解决了仿真系统的灵活性和可扩充性问题,减少了网络冗余数据,并且可以将真实仿真、虚拟仿真和构造仿真集成到一个综合的仿真环境中,满足复杂大系统的仿真需要。目前,HLA已正式成为IEEE建模与仿真标准(IEEE 1516.X系列),因此,基于HLA的分布仿真将是今后仿真发展的主要方向。

要推广基于HLA的仿真应用需要研究的问题很多,本书在作者大量学习和研究HLA的基础上,根据理论与实践相结合的原则,系统地介绍了HLA的基本理论和技术,并结合具体的应用实例,详细讨论了基于HLA的仿真程序设计方法,书中很多是作者学习的心得。

全书共计12章。前5章主要介绍了HLA的基本理论和相关技术。第1章介绍了计算机仿真的发展历史以及HLA的发展过程;第2章和第3章结合HLA标准,系统地介绍了HLA的基本理论和技术;第4章重点介绍了HLA的运行支撑环境RTI;第5章详细讨论了联邦开发和执行过程(FEDEP)模型。后7章结合具体实例,详细分析了基于HLA的仿真程序设计以及DMSO RTI 1.3-NG服务接口的编程应用。附录中还列出了大量与HLA相关的网站,读者在学习本书的同时可以上网查阅相关资料。建议将本书与DMSO HLA 1.3标准结合使用。本书体系完整,内容实用、覆盖面广,可作为相关专业研究生和高年级本科生的教材,也可以作为广大工程技术人员的参考书。

本书由蒋晓原研究员担任主审,周彦、戴剑伟担任主编;全书的纲目由周彦、戴剑伟讨论确定。蒋晓原研究员审阅了全部书稿并对章节内容给予了具体的指导。本书编写的任务分工为:第1章由王春江博士编写,第3章第6、7、8、9、10节以及第4章、第5章

由戴剑伟博士编写，第 7 章由唐宏高工编写，第 12 章由陈少卿博士编写，其余各章由周彦编写。附录 A 和附录 B 分别由陈少卿博士、戴剑伟博士翻译，周彦审校。张金明博士、潘习林博士以及李华、朱琳杰等同志参与了大量的资料收集、整理和绘图工作。本书的实例由周彦设计改编。全书的统稿工作由周彦、戴剑伟完成。

感谢李伯虎院士在百忙中为本书作序。感谢电子工业出版社的张毅编辑和范同祥编辑为本书的出版所做的工作。在我们学习和研究 HLA 的过程中，得到了国防科技大学机电工程与自动化学院的黄柯棣教授、邱晓刚主任、郝建国博士以及总装备部炮兵装备技术研究所的王月平、赵志强等同志的大力帮助与指导，在此向他们表示衷心的感谢。另外，本书在编写过程中参考了大量的中外文资料，在此向相关的作者和单位一并表示感谢！

HLA 是一个全新的内容，其许多方面还有待进一步深入研究，本书的出版主要起一个抛砖引玉的作用。由于时间仓促，再加上我们对 HLA 的理解也很有限，书中难免有不当甚至错、漏之处，敬请读者批评指正^[1]。

本书由飞思科技产品研发中心组织策划。联系方式如下：

电 话：（010）68134545 68134811

E - m a i l: support@fecit.com.cn

网 址: <http://www.fecit.com.cn> <http://www.fecit.net>

通用网址: 计算机图书 FECIT 飞思教育 飞思科技 飞思在线

作 者

^[1] 为便于和读者交流，我们特设了一个读者信箱，地址为：HLAProgrammer@sohu.com，欢迎读者来信。

目 录

第 1 章 概述.....	1
1.1 计算机仿真的发展历史.....	1
1.1.1 仿真计算机.....	1
1.1.2 仿真软件.....	2
1.2 分布交互仿真.....	5
1.2.1 概念及特点.....	5
1.2.2 发展历史.....	6
1.2.3 主要目标.....	7
1.2.4 应用领域.....	9
本章小结.....	10
第 2 章 高层体系结构.....	11
2.1 引言.....	11
2.1.1 HLA 的组成.....	12
2.1.2 HLA 的基本思想.....	12
2.1.3 HLA 的主要特点.....	13
2.2 HLA 规则.....	13
2.2.1 联邦规则.....	14
2.2.2 成员规则.....	15
2.3 联邦管理.....	16
2.3.1 联邦管理的主要内容.....	16
2.3.2 联邦管理的基本过程.....	16
2.3.3 联邦管理服务.....	17
2.4 声明管理.....	19
2.4.1 基本概念.....	19
2.4.2 声明管理的作用.....	21
2.4.3 声明管理服务.....	22
2.5 对象管理.....	22
2.5.1 基本概念.....	23
2.5.2 对象管理服务.....	24
2.6 所有权管理.....	25
2.6.1 所有权管理的基本内容.....	25
2.6.2 所有权管理服务.....	27
2.7 时间管理.....	28
2.7.1 基本概念.....	28
2.7.2 HLA 时间管理的任务.....	29

2.7.3	HLA 的时间管理原则.....	30
2.7.4	HLA 的时间管理机制.....	31
2.7.5	HLA 中的时间推进.....	36
2.7.6	HLA 的时间管理服务.....	40
2.8	数据分发管理.....	41
2.8.1	基本概念.....	41
2.8.2	数据分发管理的基本原理.....	44
2.8.3	数据分发管理服务.....	46
2.9	管理对象模型.....	48
2.9.1	管理对象模型中的对象类.....	48
2.9.2	管理对象模型中的交互类.....	51
	本章小结.....	54
第 3 章	HLA 对象模型模板.....	55
3.1	概述.....	55
3.1.1	联邦对象模型 (FOM)	55
3.1.2	成员对象模型 (SOM)	55
3.1.3	HLA 和 OOAD 相关概念的区别	56
3.1.4	HLA OMT 的组成.....	56
3.2	对象模型鉴别表.....	57
3.2.1	对象模型鉴别表的作用.....	57
3.2.2	对象模型鉴别表的格式.....	58
3.2.3	例子.....	58
3.3	对象类结构表.....	59
3.3.1	对象类结构表的基本作用和原理.....	59
3.3.2	对象类结构表的格式.....	60
3.3.3	对象类结构的设计原则.....	62
3.3.4	例子.....	62
3.4	交互类结构表.....	64
3.4.1	基本原理.....	64
3.4.2	交互类结构表的格式.....	64
3.4.3	说明.....	65
3.4.4	例子.....	67
3.5	属性表.....	67
3.5.1	属性表的作用.....	67
3.5.2	属性表的格式.....	68
3.5.3	说明.....	72
3.5.4	例子.....	72
3.6	参数表.....	74
3.6.1	参数表的作用.....	74

3.6.2	参数表的格式.....	74
3.6.3	说明.....	77
3.6.4	例子.....	77
3.7	枚举数据类型表.....	77
3.8	复杂数据类型表.....	78
3.9	路径空间表.....	81
3.9.1	基本原理.....	81
3.9.2	路径空间表的格式.....	81
3.9.3	例子.....	83
3.10	FOM / SOM 词典.....	84
3.10.1	FOM/SOM 词典的目的	84
3.10.2	相关表格的格式.....	84
	本章小结.....	86
第 4 章	联邦运行支撑环境.....	87
4.1	概述.....	87
4.1.1	RTI 的主要作用.....	87
4.1.2	RTI 的发展历史.....	88
4.1.3	RTI 的体系结构模型.....	89
4.1.4	RTI 的通信方式.....	89
4.1.5	RTI 的配置文件.....	90
4.2	RTI 1.3-NG	90
4.2.1	RTI 1.3-NG 的主要组成.....	90
4.2.2	RTI 1.3-NG 的接口服务.....	93
4.3	联邦执行数据文件.....	93
4.3.1	FED 文件的结构	94
4.3.2	FED 文件的语法	95
4.4	RTI 初始化文件.....	98
4.4.1	ProcessSection.....	99
4.4.2	FederationSection.....	104
4.4.3	FederateSection	114
4.4.4	RID 参数一致性问题	115
	本章小结.....	116
第 5 章	联邦开发和执行过程模型.....	117
5.1	引言.....	117
5.2	FEDEP 模型概述.....	118
5.3	FEDEP 模型的详细内容.....	119
5.3.1	定义联邦目标.....	121
5.3.2	开发联邦概念模型.....	122
5.3.3	设计联邦.....	123

5.3.4	开发联邦.....	124
5.3.5	集成和测试联邦.....	127
5.3.6	运行联邦并准备结果.....	129
5.4	联邦开发和执行过程的自动化.....	130
本章小结.....		132
第 6 章	HLA 仿真程序设计基础.....	133
6.1	联邦执行的生命周期.....	133
6.1.1	创建联邦执行.....	133
6.1.2	联邦执行存在.....	134
6.1.3	撤销联邦执行.....	136
6.2	联邦成员的程序框架.....	136
6.2.1	基于命令行方式的联邦成员.....	137
6.2.2	基于 Windows 方式的联邦成员.....	139
6.2.3	RTIambassador 类和 FederateAmbassdor 类.....	140
6.3	HLA 仿真程序实例.....	142
6.3.1	FightGame 简介.....	143
6.3.2	FightGame 的执行过程.....	150
本章小结.....		153
第 7 章	联邦管理.....	155
7.1	引言.....	155
7.2	联邦执行的管理.....	157
7.2.1	基本过程.....	157
7.2.2	主要服务详解.....	158
7.2.3	编程应用.....	164
7.2.4	tick()函数.....	175
7.3	联邦成员的同步.....	177
7.3.1	基本过程.....	178
7.3.2	主要服务详解.....	179
7.4	联邦状态的保存和恢复.....	185
7.4.1	联邦状态保存的基本过程.....	186
7.4.2	联邦状态保存的主要服务.....	187
7.4.3	联邦状态恢复的基本过程.....	196
7.4.4	联邦状态恢复的主要服务.....	196
本章小结.....		205
第 8 章	时间管理.....	207
8.1	时间管理策略的设置.....	207
8.1.1	时间管理策略的设置过程.....	207
8.1.2	主要服务.....	208
8.1.3	编程应用.....	215

8.2	逻辑时间的推进.....	219
8.2.1	基于时间步长的时间推进.....	219
8.2.2	基于事件的时间推进.....	223
8.2.3	乐观成员的时间推进.....	225
8.2.4	FightGame 中的时间推进.....	228
8.3	辅助服务.....	229
8.3.1	queryFederateTime().....	230
8.3.2	queryLookahead().....	231
8.3.3	modifyLookahead().....	232
8.3.4	queryLBTS().....	233
8.3.5	queryMinNextEventTime.....	234
	本章小结.....	235
第 9 章	声明管理.....	237
9.1	引言.....	237
9.1.1	对象类/交互类的层次结构.....	237
9.1.2	公布/订购的特点.....	238
9.1.3	声明管理中的控制机制.....	240
9.2	公布和订购对象类.....	241
9.2.1	对象类公布/订购的基本过程.....	241
9.2.2	主要服务详解.....	243
9.2.3	公布对象类.....	254
9.2.4	订购对象类.....	256
9.3	公布和订购交互类.....	259
9.3.1	交互类公布/订购的基本过程.....	259
9.3.2	主要服务详解.....	260
9.3.3	编程应用.....	269
	本章小结.....	273
第 10 章	对象管理.....	275
10.1	对象实例的注册、发现和删除.....	275
10.1.1	基本过程.....	275
10.1.2	主要服务详解.....	276
10.1.3	注册对象实例.....	285
10.1.4	发现对象实例.....	287
10.1.5	删除对象实例.....	289
10.2	对象实例的更新和反射.....	292
10.2.1	基本过程.....	292
10.2.2	主要服务详解.....	293
10.2.3	对象实例的更新.....	301
10.2.4	反射实例属性值.....	304

10.3	交互实例的发送和接收.....	306
10.3.1	基本过程.....	306
10.3.2	主要服务详解.....	307
10.3.3	交互实例的发送.....	311
10.3.4	交互实例的接收.....	313
10.4	辅助对象管理.....	316
10.4.1	传输方式控制.....	316
10.4.2	对象更新控制.....	317
10.4.3	属性范围控制.....	317
	本章小结.....	318
第 11 章	所有权管理.....	319
11.1	引言.....	319
11.2	所有权的“拉”模式.....	320
11.2.1	“拉”模式的基本过程.....	320
11.2.2	“拉”模式中的主要服务.....	322
11.3	所有权的“推”模式.....	336
11.3.1	“推”模式的基本过程.....	336
11.3.2	“推”模式中的主要服务.....	337
11.3.3	FightGame 中的所有权管理.....	346
11.4	相关服务.....	358
	本章小结.....	358
第 12 章	数据分发管理.....	361
12.1	RTI 1.3-NG 数据分发管理简介.....	361
12.1.1	基本概念.....	361
12.1.2	数据分发管理策略.....	362
12.2	创建区域.....	366
12.2.1	创建区域的基本过程.....	366
12.2.2	主要服务详解.....	367
12.3	属性和区域的绑定.....	370
12.3.1	实例属性更新和区域的绑定.....	370
12.3.2	类属性的订购和区域.....	371
12.3.3	带区域的请求更新.....	371
12.3.4	主要服务详解.....	373
12.4	交互实例和区域的绑定.....	384
12.4.1	基本交互过程.....	384
12.4.2	主要服务详解.....	385
12.5	编程应用.....	391
	本章小结.....	395
	参考文献.....	397

附录 A	RTI 1.3-NG 的管理对象模型.....	399
A.1	交互类.....	400
A.2	对象类.....	422
附录 B	RTI 1.3-NG 的类和支持数据类型.....	427
B.1	类.....	427
B.2	支持类型.....	452
附录 C	相关网站.....	457
附录 D	缩略词和术语.....	459
D.1	缩略词.....	459
D.2	术语.....	459

第 1 章 概 述

计算机仿真技术是以计算机为工具，以相似原理、信息技术以及各种相关应用领域的基本理论与技术为基础，根据系统试验的目的，建立（实际或设想）系统的模型，并在不同的条件下，对模型进行动态运行（试验）的一门综合性技术。随着计算机技术的提高，带动了计算机仿真技术的快速发展。当前，计算机仿真已经成为系统特别是复杂的大系统分析、研究、测试、评估、研制和技能训练的重要手段，并已广泛应用于国防、制造、能源、交通、农业、医疗、教育、经济、天气预报等各个行业的重要领域。

1.1 计算机仿真的发展历史

广义上的“仿真”历史久远，可以追溯到 2000 多年以前人类在建筑、造船等行业中对系统比例模型的应用。但严格意义下的“仿真”则是从 20 世纪 40 年代末，几乎与计算机同步诞生和发展，至今已经历了 50 多年的发展历程。从仿真硬件的角度讲，计算机仿真大致经历了三个主要发展阶段，即：模拟计算机仿真，模 / 数混合计算机仿真和数字计算机仿真（即现代计算机仿真或全数字仿真）。从仿真软件的角度讲，计算机仿真可以分为五个相互交叉的发展阶段：仿真程序包和仿真语言、一体化仿真环境、智能化仿真环境、面向对象的仿真、分布式交互仿真。

1.1.1 仿真计算机

1. 模拟计算机仿真

模拟计算机仿真是指将系统模型编排在模拟电子计算机上并运行，以期达到对系统进行分析和研究的目的。模拟计算机仿真的基本原理是不同物理系统间的特性比例相似，即两个物理本质完全不同的系统，其物理运动所遵循的微分方程相似，且参数一一对应。例如，一个由质量系统（参数为 M ）、线性弹簧系统（参数为 K ）和阻尼系统（参数为 D ）构成的简单机械系统在外力作用下的机械运动的数学描述，与一个由电感（参数为 L ）、电容（参数为 C ）和电阻（参数为 R ）组成的简单电路系统在电源作用下的电子运动的数学描述，两者是相似的。它们都可以被描述为二阶线性微分方程的形式，而且它们的参数之间有相互对应的关系。因此，我们可以利用其中任何一个系统研究另一个系统，这种研究方法称为直接相似法。这就是模拟计算机仿真的基础。

由于使用模拟电子计算机求解微分方程的方法方便有效，而且所有运算都是并行进行的，因此模拟计算机仿真适合于对连续动力学系统进行仿真，其执行效率也比较高。但采用模拟计算机进行仿真时，模型校核复杂，计算精度差，结构复杂，因而不适合对大规模系统进行仿真。这种仿真技术在 20 世纪五六十年代十分流行，但现在已经基本上被数字

计算机仿真所取代。

2. 混合计算机仿真

所谓混合计算机仿真，是指利用混合模拟机或混合计算机系统来实现的仿真。混合模拟机是指在模拟计算机中加入大量的数字逻辑部件、模拟开关以及模拟/数字混合式部件（如数控系数器、数控积分器、数控函数发生器等）的一种将模拟技术和数字技术紧密结合在一起的计算机。它不仅能完成原有模拟计算机的所有功能，而且还具有迭代运算和描述混合系统模型的能力。混合计算机系统是指由模拟计算机、数字计算机以及相关接口设备组成的计算机系统。混合仿真系统主要应用在以下几个方面：

- 系统仿真速度和仿真精度要求在模拟仿真或数字仿真单独执行有困难的场合，比如空间轨道飞行器的仿真任务。
- 若系统本身是一个既有数字又有模拟的混合系统，采用混合计算机仿真是十分方便的。
- 混合仿真中模拟子系统的高速求解能力，使它在参数存优、最优控制、随机过程统计计算等方面的研究中发挥作用。

混合仿真系统用途广泛，但它的造价昂贵，其用户主要是政府或大型企业。混合计算机仿真在 20 世纪六七十年代曾广泛应用于航空、航天、兵器 and 核电等大型系统的研究，但现在许多应用领域已逐渐被数字计算机仿真所取代。

3. 全数字仿真

全数字仿真，即基于全数字计算机的仿真。它从 20 世纪 50 年代开始发展，但由于当时的数字计算机计算能力较差而且价格昂贵，所以整个 50 年代，全数字仿真在应用范围 and 市场份额方面都非常有限，仅限于一些数学仿真。上世纪 60 年代至 70 年代中叶是全数字仿真与混合计算机仿真展开激烈竞争的年代，但由于当时的市场需求主要是实时仿真，而当时的数字计算机技术还不能很好地满足这种需要，因此混合计算机仿真还是当时的主流。随着数字计算机从第三代进化到第四代，一些软硬件配备齐全、价格低廉的小型通用机很快进入市场，加之数字仿真算法和高级仿真语言获得了突破性进展，到 20 世纪 70 年代中期，中型混合计算机的性能价格比已不能与配上完备仿真软件的小型数字计算机相抗衡。近年来随着超大规模集成电路（Very Large Scale Integration, VLSI）工艺、RISC 技术、分布/并行处理技术、仿真算法及软件的迅速发展，高档微机及工作站等通用计算机在计算速度、精度及存储容量等方面为通用计算机做实时仿真打下了基础，而且在价格方面比专用仿真计算机有明显的优势。因此，自 20 世纪 90 年代以来，国内一些高校和科研院所通过对通用计算机的软件系统、时间调度系统和 I/O 系统等的适应性改造，将通用计算机进一步开发为不同档次的、满足不同需要的实时仿真机。

1.1.2 仿真软件

仿真软件是支持各类仿真活动的计算机软件的总称。它随着仿真方法学和计算机技术（包括计算机软/硬件技术、计算机网络技术、数据库技术、计算机操作系统、软件工程、计算机图形/图像技术、多媒体技术、人工智能技术等）的发展而发展，同时通过不断吸收系统工程和自动控制等技术的新成果，使其自身不断充实和完善。仿真软件的发展大致

经历了五个阶段，除分布交互仿真外，另外四个发展阶段列举如下：

1. 仿真程序包和仿真语言

1961年 G. W. Morgenthler 首次对“仿真”一词做了技术性的解释，认为“仿真”是在实际系统尚不存在的情况下，对系统或活动本身的复现。从该定义来看，早期人们对“仿真”一词的理解主要侧重于“仿真实验”本身，即着重于如何获得系统中有关变量的时间响应——轨迹。因此，仿真软件发展的早期，主要是开发和研制各类仿真程序包和仿真语言。同时，计算机高级语言和编译技术的成熟也推动了仿真语言的发展。

仿真语言是一种面向问题的高级语言，通常由模型与实验描述语言、翻译程序、实用程序、算法库以及运行控制程序组成。其中翻译程序是将模型与实验描述语言书写的源程序翻译成宿主语言（如 FORTRAN、C、C++、Java 等）的程序。仿真语言可以分为三类：连续系统仿真语言（如 CSSL67、CSSL81、ICSL1、ICSL2、ACSL 等），离散事件系统仿真语言（如 ECSL、EXPRESS、GASP、GPSS、SIMULA 等），混合系统仿真语言（如 SIMAN、SLAN、IHSL 等）。

仿真语言比较侧重“仿真实验”，其建模功能和分析功能都比较弱，它忽视了仿真过程的建模和结果分析，因此不能完全满足现代仿真应用的需要。但从目前看，仿真语言仍是仿真软件的主体。

2. 一体化建模与仿真环境

1979年，Oren 及 Zeigler 在《先进的仿真方法学概念》一书中，首次提出了建模与实验分离，模型基于系统理论，非过程建模，模型库等新概念。1984年，Oren 提出“仿真是一种基于模型的活动”的概念，并给出了仿真的基本概念框架，即“建模—实验—分析”三段式。基于上述认识，仿真软件必须能够支持仿真的全生命过程，应该设计多种功能的软件，如建模软件、实验设计软件、仿真执行软件、结果分析软件等，而这些软件之间存在密切的信息交换。为了提高仿真效率，必须将它们集成在一起，形成一体化的仿真环境。同时，在 20 世纪 70 年代末，由于微机、工作站和高分辨率终端的出现，使 CASE（计算机辅助软件工程）工具趋于实用，数据库技术、计算机网络技术和操作系统技术也有了很大的发展，这些技术为建立一体化建模与仿真环境奠定了基础。因此，从 20 世纪 80 年代中期以来，开发一体化的仿真环境成为仿真软件的一个发展趋势。根据一体化的程度，这些仿真支持环境可以分为如下四个层次：

- 不同的功能软件通过一个软件，利用数据建模接口实现一体化，这是最低一级的一体化。
- 重新划分功能模块，建立模型库、参数库、实验框架库，然后通过数据库实现一体化。如美国 Pritsker and Associates INC. 公司于 1985 年研制成功的 TESS 就是通过数据库实现一体化的。20 世纪 90 年代初，为增强仿真环境的灵活性、易于新工具的集成、提高面向不同用户的能力，Standridge 提出的模块化仿真环境也是以数据库为核心的一体化仿真环境。
- 在仿真操作系统的支持下，实现对仿真关联资源的有效管理，并执行这些资源的匹配与运行，实现整个仿真环境的高度一体化。如清华大学自动化系研制的 IPSOS。
- 基于软总线的即插即用集成方式，通过仿真框架实现仿真全生命周期各类活动的



集成优化。

3. 智能化建模与仿真环境

20 世纪 80 年代末, Oren 在“仿真是一种基于模型的活动”的基础上进一步提出: 仿真是基于三类知识(即描述性知识、目的性知识和处理知识)的知识处理过程。Oren 认为, 仿真是基于模型的经验知识的生成过程, 而人工智能则是面向目标的, 它具有适应性的知识处理能力。因此, 仿真和人工智能(Artificial Intelligence, AI)是可以用来处理相似问题的两类不同的工具, 两者可以紧密地结合在一起, 互相推动, 共同发展。AI 在仿真过程中的主要作用有: 模型的开发, 知识的提炼和表示, 仿真中的决策制定, 模型的快速原型, 仿真输出数据的处理和解释, 模型的修改和维护等。上述功能可以通过建模专家系统、仿真专家系统、输出结果分析专家系统以及定性仿真和基于智能代理的仿真来实现。将上述各类系统集成成为一个一体化的、智能化的建模与仿真环境, 应该是今后仿真软件的重要发展方向之一。

4. 面向对象的仿真软件

面向对象的仿真软件(Oriented Object Simulation System, OOSS)是基于 OO 方法学的一种仿真软件。它采用面向对象的建模方法, 认为客观系统是由各种相互作用的对象组成, 这符合人们的思维方式。同时它又具备 OO 软件的数据抽象、信息隐藏、继承、动态链接等特点, 因而使得仿真更为直观、自然, 也使得这类仿真软件具有更优良的模块性、可重用性、可维护性和灵活性, 并支持知识工程的实现和快速原型法。面向对象仿真还可以在不同的处理器上运行, 这是由于实体体现了逻辑上相对确定的执行过程, 所以各实体可在不同的处理器上运行, 支持并行仿真。此外, 面向对象仿真技术已经成为开发结构复杂系统的大型分布式仿真的重要手段。

OO 仿真软件的核心问题是建模、软件构成框架和实现。OO 仿真软件与传统仿真软件相比, 具有以下异同点:

- 建模方法不同 OO 仿真软件的模型开发基于面向对象的模式, 通过把问题域分解成一系列相互作用的对象来构造出模型。传统的仿真软件属于过程模式, 它采用面向任务的观点, 当提出一种解决目标问题的方法时, 它将解决方法分解成一系列任务来完成, 这些任务就构成了基于过程的仿真语言的基本结构。在大多数情况下, OO 仿真软件的分解是一种更自然的方式。
- 软件框架和实现方式不同 OO 仿真软件是由一个个对象的模块构成, 相互间通过消息进行通信, 一个对象根据提交给它的消息, 通过定义的操作来采取相应行动, 完成仿真运行、输出分析等工作, 可被视为以数据(即对象)为中心, 利用消息传递进行仿真的实现机制。而传统仿真语言中, 程序是由传递参数的过程和函数的集合组成的, 即每个过程处理它的参数, 因此是以执行单元(即过程)为中心, 通过功能函数的调用来完成仿真。
- 支撑环境不同 OO 仿真软件一般基于面向对象的程序设计语言开发, 而传统仿真软件则大多利用基于过程的过程语言开发。

另外, 从方法学的角度看, 使用 OO 仿真软件并不意味着必须使用某种特别的仿真建模方法。例如, 离散时间仿真建模有三种方法: 事件调度法, 活动扫描法, 进程交互法。OO 仿真软件同样可以基于这三种方法来完成建模和仿真运行, 只是在利用这些方法建模

仿真时，对各实体的定义描述与传统仿真不同。

当前，仿真应用在广度和深度上都有了很大的发展，仿真的计算问题已经覆盖了数值计算、事件处理和逻辑推理等方面，而且仿真所用的计算机的类型很多。从混合计算机到全数字仿真机，从个人计算机、工作站到小型、大型和巨型计算机，从专用仿真计算机到通用计算机等都可以用在不同的仿真场合。特别是开放的复杂巨系统仿真，必须将异构的多台计算机通过具有统一的通信标准、规范和协议的高速网络互连起来，以构成一个可互操作的一体化仿真环境。总的来说，建模与仿真技术的发展趋势是分布式、交互式、开放式以及标准化、组件化和集成化的，其立足点是解决各类仿真模型、仿真系统之间的互操作和可重用等问题。

1.2 分布交互仿真

1.2.1 概念及特点

分布交互仿真（Distributed Interactive Simulation, DIS）是指采用协调一致的结构、标准、协议和数据库，通过局域网或广域网，将分散在各地的仿真设备互联，形成可参与的综合性仿真环境。计算机仿真通常可分为三种类型，即虚拟仿真、构造仿真和真实仿真。虚拟仿真（Virtual Simulation）是指由作战人员操作仿真的武器系统进行的作战仿真，即人在回路中的仿真。构造仿真（Constructive Simulation）是指仿真中的所有实体均由计算机模型产生，即由仿真的人操作仿真的武器系统进行的仿真。真实仿真（Live Simulation）是由真实的战斗人员使用真实的武器系统和保障系统，在尽可能真实的作战环境中进行的作战演习。分布交互仿真既可以是某种单一类型的仿真，也可以是几种类型的综合。与以往的仿真技术相比，它们的不同之处体现在以下几个方面：

- 在体系结构上，由过去的集中式、封闭式发展到分布式、开放式和交互式，构成可互操作、可移植、可伸缩及强交互的协同仿真体系结构。
- 在功能上，由原来的单个武器平台的性能仿真，发展到复杂环境下，以多武器平台为基础的体系与体系对抗仿真。
- 在手段上，从单一的构造仿真、真实仿真和虚拟仿真，发展成集上述多种仿真为一体的综合仿真系统。
- 在效果上，由只能从系统外部观察仿真的结果或直接参与实际物理系统的测试，发展到能参与到系统中，与系统进行交互作用，并可得到身临其境的感受。

分布交互仿真是计算机技术的进步与仿真需求不断发展的结果，其特点主要表现在五个方面，即：分布性、交互性、异构性、时空一致性和开放性。在分布交互仿真系统中，各个仿真节点在地理位置上是分布的，在功能和计算能力上同样也是分布的，既可联网交互运行，也可独立运行各自的仿真功能，而且不同硬件和操作系统的节点可以并存于同一个环境中。在仿真运行过程中，一方面人可以与虚拟实体、计算机生成的构造实体和实际存在的现实实体进行交互；另一方面各个实体之间具有交互能力。另外，分布交互仿真系