

产品全生命周期管理的适应性建模与情景驱动在线仿真

白勇军 陈阳 赵勇

(华中科技大学系统工程研究所 武汉 430074)

摘要 建模与仿真应用于产品全生命周期管理中,提高了产品开发、使用及维护的效率和质量。但随着产品复杂程度的提高,其建模与仿真的框架越来越复杂,对环境变化的适应性要求也越来越高。以武器装备采办为背景,将仿真系统置于与真实系统平等、并行的地位,建立仿真情景模型,并采用多 Agent 技术探讨基于递阶结构的适应性仿真框架和技术。其中仿真情景的在线生成和逐层映射反映了系统的进展和环境的变化,并以此为动力驱动仿真的重组,体现了仿真系统的并行性、动态性、集成性等特点,可促进建模与仿真技术在产品全生命周期管理中的实施和应用。

关键词 产品全生命周期管理,适应性建模,情景驱动,在线仿真

中图分类号 N945 文献标识码 A

Research on Adaptive Modeling and Scenario-driven Online Simulation for Product Lifecycle Management

BAI Yong-jun CHEN Yang ZHAO Yong

(Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract Modeling and Simulation(M&S), which are applied to product lifecycle management(PLM), have the ability to improve the efficiency and quality of product development, usage and maintenance. But with products' complexity, the M&S need high adaptability of environment change. Based on the background of weapon acquisition, an adaptive simulation framework, which adopts hierarchical multi-agent technology, was presented. Under the framework, scenario models, which drive reorganization of the simulation systems, were generated and mapped level by level according to the evolution of the real systems and their environment. The framework reflects the parallelism, dynamics and integrativity of the simulation systems, and can enhance the adaptability and application ability of M&S throughout PLM.

Keywords Product lifecycle management, Adaptive modeling, Scenario-driven, Online simulation

1 引言

随着企业信息化建设的深入,一个新的概念 PLM(Product Lifecycle Management,产品全生命周期管理),在继 ERP(企业资源规划)、CRM(客户关系管理)、SCM(供应链管理)之后尤其是在制造业正在逐渐成为新的热点。PLM 是一种应用于在单一地点的企业内部、分散在多个地点的企业内部以及在产品研发领域具有协作关系的企业之间的,支持产品全生命周期的信息的创建、管理、分发和应用的一系列应用解决方案,它能够集成与产品相关的人力资源、流程、应用系统和信息^[1]。

进入 21 世纪以来,PLM 技术得到了迅速发展,尤其是建模仿真技术的应用,在提高了产品开发的效率和品质的同时,也使得 PLM 的结构和运作机理变得更丰富。PLM 主要包括两方面内容:过程管理和产品管理。其中,过程管理是对产品开发过程的规划和调度,如生产调度、装配序列等,产品管理则包括产品的需求分析、方案设计、工艺设计等。这些内容中,一部分完全区别于真实系统,通过建模仿真提供一种预测和决策支持,如生产调度仿真;而另一部分在宏观上与真实系

统是没有区别的,如需求分析。因此 PLM 实际上可视为仿真系统与真实系统的集成与协同,即真实系统的运行直接影响到仿真系统的执行,而仿真结果则用于指导真实系统的运行和优化。此时的 PLM 已经成为一种在线仿真系统,其要点之一在于仿真系统与真实系统的集成以及仿真系统的动态重组,即仿真系统的适应性^[2-4]。

目前,对仿真适应性的研究多集中在多尺度(多分辨率)建模和仿真动态组合方面。Sabine Timpf 提出将不同尺度的数据保存在一个层次数据结构中,用树形结构来表示实体之间的联系,但描述层次数据模型的树是一种有向树(层次顺序),限制了数据存取的路径,且不能直接表示多对多的联系^[5];Zeigler 等基于 SES/MB(System Entity Structure/Model Base)框架来描述多分辨率模型^[6];针对多分辨率模型的一致性度量, Brent D. Kornman 和 Peter B. Marion 提出了模型间的一致性度量(Measure of Consistency, MOC)的概念,但并未给出 MOC 的确切定义以及具体方法^[7]。另外,用于仿真组合的形式化模型可以分为基于 Petri 网的模型^[8-11]、基于有限状态自动机的模型^[12]和基于进程代数的模型^[13]等,其中 Petri 网能够显式描述系统的状态和事件,尤其在描述离

到稿日期:2010-06-01 返修日期:2010-09-07 本文受国家重点基金和 863 计划课题(2009AA04Z107)资助。

白勇军(1976-),男,博士生,主要研究方向为复杂系统建模与仿真, E-mail: yongjunbai@yahoo.com.cn; 陈阳(1977-),男,博士,讲师,主要研究方向为复杂系统建模仿真与决策; 赵勇(1967-),男,教授,博士生导师,主要研究方向为决策分析及复杂系统建模与优化等。

散事件动态系统方面具有较强的模型分析能力;有限状态自动机可通过状态的变迁来描述组合模型间的交互;进程代数可以简单、抽象和形式化地表示定义动态交互。上述研究着重于如何支持和提高仿真的适应性,使得仿真系统具备一定适应能力,但在仿真环境和需求的变化识别以及如何用这些变化来驱动仿真系统的动态重组等领域尚有不足。

本文以武器装备采办为背景,根据影响仿真的情景因素将系统环境和真实系统状态等的改变映射为仿真情景的演化,研究产品全生命周期管理的适应性建模与情景驱动在线仿真技术,并探讨相应的仿真执行机制,为产品全生命周期决策提供支持。

2 适应性建模

真实系统对仿真系统的影响主要体现在仿真范围和仿真尺度。对于真实系统已经完成的任务,仿真系统不需要重复论证,即反映为仿真范围的变化;真实系统中产品概念方案形成后对制造、维护、销毁等的仿真是一个大尺度仿真。而当真实系统中产品结构方案形成后,系统需要根据结构方案对制造、维护、销毁等作进一步的小尺度仿真,所以随着真实系统的进展,系统的仿真尺度将经历一个由粗变细的过程,即仿真尺度的变化。对一给定任务,由于仿真系统对应着多个不同尺度、视角描述的模型,需要在应用需求、粒度、时间与资源之间进行折中,并动态选择合理的模型及交互机制,进行仿真的动态组合与重构。

为了从整体上理解产品全生命周期管理中的仿真需求,需要用规范的表示方法描述仿真模型的元信息,形成产品全生命周期仿真元模型,用于保证产品信息在循环演化过程中的一致性,以支持后续的仿真执行及决策过程。

2.1 主要元模型

产品全生命周期仿真模型按过程可分解为多个任务模型,按结构可分解为产品模型,而按功能可分解为维任务模型。任务模型可认为是由多个任务(Task)按一定逻辑关系组成的序列。单个任务则是完成序列内一组逻辑步骤的一项工作,是一个目标明确、语义完整、相对独立、长时间的群体协作行为。任务的执行需要一定资源的支持,这些资源包括人力、软件/工具、信息/数据等。产品模型则是对产品的需求、功能、结构、工艺等信息的组织。任务、产品及资源等都由相应的指标体系描述其特征。因此元模型应包括指标基本信息描述、指标体系描述、产品信息描述、任务信息描述、流程信息描述、资源信息描述等。

1) 指标基本信息

指标基本信息主要是对指标类型的分类描述, $IND_INF=(IND_DIM, IND_ID, IND_NAME, IND_TYPE, IND_UNIT)$ 。其中 IND_DIM 表示指标维,如进度、费用、性能、风险等; IND_ID 是该指标的唯一标识; IND_NAME 表示指标名; IND_TYPE 表示指标类型,包括定性和定量两种; IND_UNIT 表示指标计量单位,如米、公里、千克等。

2) 指标体系信息

指标体系是多个指标的有机组成体,即除了描述各个指标的状态,指标之间还存在一定映射关系。因此 $IND_SYS=\{INDS, F_{ind}\}$, 其中 $INDS=\{IND_i, i=1, 2, \dots, n\}$ 是指标的集合, $IND_i=(IND_ID, IND_BDGT_MAX, IND_BDGT_$

$MIN, IND_VALUE)$ 表示指标体系中的各个指标, IND_ID 是指标标识, IND_BDGT_MAX 和 IND_BDGT_MIN 分别表示该指标预期最大值和最小值, IND_VALUE 表示该指标实际(或仿真)结果。

F_{ind} 表示指标之间的映射关系,用矩阵表示 $F_{ind}=[f_{ij}]$, 其中 $i=1, 2, \dots, n, j=1, 2, \dots, n, f_{ij}$ 表示映射函数。

3) 资源

资源 $Resource=(RID, RType, RName, RVersion, RInput, ROutput, RAddr, IND_SYS_{Res})$ 。其中, RID 是资源实体的唯一标识; $RType$ 表示资源类型; $RName$ 是资源名称; $RVersion$ 表示版本; $Rinput$ 和 $ROutput$ 分别表示该资源的输入和输出集合; $RAddr$ 表示资源访问地址; IND_SYS_{Res} 表示描述该资源的指标体系。

4) 产品

产品描述为产品特征信息的集合。特征信息根据研究角度的不同可划分为不同属性,主要包括需求、功能、结构等,即 $Product=\{RS, FS, SS, DM\}$ 。其中,需求属性 $RS=\{REQ_1, REQ_2, \dots, REQ_n\}$, 是产品开发的目标和要求的集合。

功能属性是由需求分析得到的功能结构和要求。功能属性按功能结构形成层次描述,即 $FS=\{F_1, F_2, \dots, F_n\}$, 功能节点 $F_i=(FID, FName, IND_SYS_{Func}, ParentF), i=1, 2, \dots, n$ 。其中, FID 表示功能/子功能的标识, $FName$ 表示功能/子功能的描述, IND_SYS_{Func} 表示功能/子功能的指标体系, $ParentF$ 表示该功能/子功能的父层功能。

结构属性是对产品组成结构的层次描述 $SS=\{S_1, S_2, \dots, S_n\}$, 结构节点 $S_i=(SID, SName, IND_SYS_{Struct}, ParentS), i=1, 2, \dots, n$ 。其中, SID 表示产品组成部分的标识, $SName$ 是对该组成部分的描述, IND_SYS_{Struct} 表示该组成部分的指标体系, $ParentS$ 表示该组成部分的父层。

$DM=[a_{ij}], (i=1, 2, \dots, m, j=1, 2, \dots, n)$ 为设计矩阵, 表示结构对功能的满足关系。 $a_{ij} \neq 0$ 表示第 i 项功能由第 j 项结构实现。

5) 任务

产品全生命周期管理在纵向上涵盖了需求分析、方案设计、生产制造、试验评估、使用维护、退役报废等不同工作域,称之为域任务。针对特定工作域,按照工作的性质和安排将该工作域逐层分解形成任务。不同工作域由于目标和资源配置等的差异,需要采用不同的任务分解标准。例如,方案设计可按学科分解,其中学科是指按知识体系或知识性质进行的分类,导弹设计可分解为弹药学设计、动力学设计、结构学设计等;详细设计则按产品物理结构分解,如某型导弹作为一个整体可细分为战斗系统、制导系统、动力系统和弹体结构系统等,其中战斗系统再按结构细分为战斗装药、引爆装置、弹头壳体、突防装置等;退役报废则按工作性质细分为拆解、运输、销毁等。任务/子任务进一步细分为维任务,表示在某一维度上执行的任务。比如进度维任务是指与进度相关的仿真、估计任务;费用维是与费用相关的计算任务;方案维指具体的工作内容、制造任务等等。维任务是按功能进行的划分,成为任务模型的最底层。

任务定义为 $Task=(TID, TType, TInput, TOutput, IND_SYS_{Task}, TStatus, Executor)$ 。其中, TID 是任务的唯一标识; $TType$ 表示任务类型,包括域任务、任务和维任务; $TInput$

和 $TOutput$ 表示任务的输入和输出集合; IND_SYS_{Task} 表示该任务的指标体系, 包括进度、费用、资源和风险等; $TStatus$ 表示状态, 分为就绪 (Ready)、进行中 (Executing)、挂起 (Suspend)、完成 (Finish)、取消 (Cancel) 等; $Executor$ 表示任务的执行者, 即系统中的资源。

进度 (T)、费用 (C)、资源 (R)、风险 (H) 等维是对任务执行不同维的论证, 其指标不尽相同, $IND_SYS_i = \{IND_SYS_i, i \in \{T, C, H\}\}$ 。具体地, 进度指标集 $INDS_T = (P_T, T, ST, FT, S)$, 其中 P_T 表示工期的概率分布函数, T 表示持续时间, ST 表示开始时间, FT 表示完成时间, S 表示总时差; 费用指标集 $INDS_C = (C, P_C)$, 其中 C 表示费用, P_C 表示费用的概率分布函数; 资源指标集为资源使用集合 $IND_SYS_R = \{(r_1, q_1), (r_2, q_2), \dots, (r_n, q_n)\}$, r 表示某类资源, q 表示资源使用数量; 风险属性 $IND_SYS_H = \{HT, P_H, H\}$, HT 表示风险类型, P_H 表示风险概率, H 表示风险后果。

6) 流程

任务按照一定逻辑关系执行形成流程, 而流程定义为 $Flow = (FID, Nodes, Arcs)$ 。其中, FID 是流程唯一标识; $Nodes = \{Task_1, Task_2, \dots\}$ 是节点的集合, 节点代表某个任务; $Arcs$ 是节点间的关系弧, 代表任务间的依赖关系。其中, 节点分为原子节点和复合节点两种类型。如果节点对应的任务已经足够明确, 不需要再细分, 则称之为原子节, 用 \square 表示; 如果需要进一步分解成更细的单元, 则称之为复合节点, 用 \square 表示。通过复合节点可将流程定义为多层结构。每个流程实体包含且只包含一个开始节点和一个结束节点, 分别用 \blacklozenge 和 \bullet 表示。

关系弧用带箭头的弧线表示, 代表任务间的衔接关系。关系弧所连接的两个节点沿箭头方向分别称为起节点和止节点。所有以某节点为终点的关系弧称为该节点的输入弧, 以该节点为起点的关系弧称为输出弧。

关系弧的状态分为等待、就绪、激活和取消。任务关系模型初始化时, 所有关系弧处于等待状态; 某弧的起节点完成后, 该弧处于就绪状态; 处于就绪状态的弧在条件满足时被激活, 否则被取消。根据激活条件的性质, 关系弧分为确定型、概率型和条件型 3 种, 其中确定型表示该弧就绪后百分之百被激活; 概率型表示该弧就绪后是否被激活服从一定概率分布, 用关系弧上的概率分布描述; 条件型表示该弧就绪后需满足一定条件才被激活, 用关系弧上的条件表达式表示。

节点的输入弧之间的关系构成该节点的输入逻辑, 决定了节点的启动条件, 其类型可归纳为顺序、与合并、或合并、异或合并 4 种, 并用不同符号表示。

\square —顺序, 表示有且仅有一条输入弧, 当输入弧被激活时该节点被启动;

\square —与合并, 表示所有输入弧被激活时该节点被启动;

\square —或合并, 表示输入弧中至少有一条被激活时该节点被启动;

\square —异或合并, 表示输入弧中有且只有一条被激活时该节点被启动。

节点的输出弧之间的关系构成该节点的输出逻辑, 决定该节点完成后输出弧的就绪数量, 其类型可归纳为顺序、与分支、或分支和异或分支 4 种, 分别表示为:

\square —顺序, 表示有且仅有一条输出弧, 该节点完成后其输

出弧就绪;

\square —与分支, 表示该节点完成后其所有输出弧就绪;

\square —或分支, 表示该节点完成后其输出弧中至少有一条就绪;

\square —异或分支, 表示该节点完成后其输出弧中有且只有一条就绪。

2.2 情景模型

情景 (Scenario) 是指对仿真执行的环境、范围、态势、过程和方法等的设想和假定, 是对仿真系统的目标和范围的描述, 并作为联结真实系统与仿真系统的纽带。针对产品全生命周期管理的要素和特点, 提出一个由产品、工作域、约束等要素组成的仿真情景模型 $Scenario = (SID, Version, SInput, SOutput, Domains, Product, Constraints, Status)$ 。其中, SID 是情景标识; $Version$ 表示版本, 用于区分在同一情景基础上继承和发展的子情景; $SInput$ 和 $SOutput$ 表示情景中的输入和输出集合; $Domains = \{domain_1, domain_2, \dots, domain_n\}$ 是工作域的集合; $Product$ 表示产品特征信息的集合; $Constraints$ 是仿真约束的集合; $Status$ 表示状态, 包括初始化 (Ready)、执行中 (Executing)、完成 (Finish) 等。

1) 工作域

工作域定义为 $domain = (DID, DInput, DOutput, Schedule, Cost, Risk, Status)$ 。 DID 是工作域的唯一标识; $DInput$ 表示工作域的输入集合; $DOutput$ 表示输出集合; $Schedule$ 表示进度维属性集合; $Cost$ 表示费用维属性集合; $Risk$ 表示风险维属性集合; $Status$ 表示工作域进展状态, 包括就绪 (Ready)、进行中 (Executing)、挂起 (Suspend)、完成 (Finish)、取消 (Cancel) 等。如需求分析域以产品需求为输入, 输出为根据需求形成的产品功能结构和要求; 方案设计域则以产品功能结构和要求为输入, 设计产生相应的产品结构方案、工艺方案、维护方案、销毁方案等。

在真实系统中, 各工作域是完全串行的关系, 在仿真系统中则按工作域的输入输出依赖形成并行与串行并存的结构。需求分析域是仿真的起点, 由于方案设计需要根据需求分析域产生的功能结构和要求作为输入, 因此两者是串行关系; 而生产制造、使用维护、退役销毁等域都以产品结构为输入, 彼此之间不存在输入输出的依赖关系, 所以这些域之间是并行关系。

2) 仿真约束

产品需求除了产品功能性需求外, 还包括对产品全生命周期过程的要求, 如周期、费用、风险等。因此在仿真情景中将原始输入的产品需求进行分解, 一部分形成产品特征信息的需求属性, 即产品功能性需求; 另一部分则作为仿真过程的约束, 如进度 (T)、费用 (C)、风险 (H) 等。仿真约束定义为维约束的集合 $Constraints = \{IND_SYS_i, i \in \{T, C, H\}\}$ 。其中 IND_SYS_i 分别是各维约束指标体系。仿真约束是对产品全生命周期管理全局的约束, 但在仿真过程中可根据仿真进程分解为多域约束。

2.3 情景生成机制

影响仿真情景的因素主要包括产品需求和真实系统的状态。仿真情景生成机制如图 1 所示。

产品需求是仿真系统和真实系统共同的输入, 需求的变动对仿真系统和真实系统的影响是全局的。因此当产品需求

发生变化时,需要生成新的仿真情景,并对其进行初始化。需求中除了对产品本身功能和性能的要求外,还包括对进度、费用、风险等过程管理的要求。因此当产品需求变化时,应先将产品需求分解为产品功能需求和过程约束。产品功能需求映射至仿真情景中产品模型的需求属性;过程约束则按研究角度细化为进度、费用、风险等维,映射于情景中仿真约束部分。对于新生成的情景,工作域将包含产品全生命周期管理中所有的工作域,并将状态初始化为 Ready。

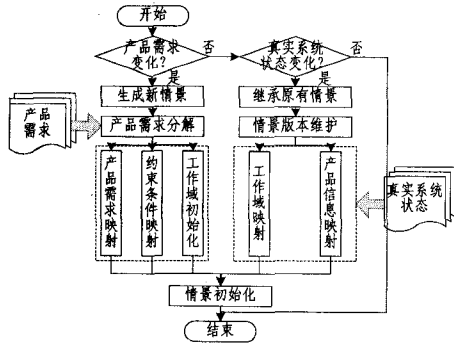


图1 仿真情景生成机制

为了与真实系统并行,仿真系统必须能根据真实系统状态调整仿真结构。针对过程管理和产品管理两项基本内容,这里把真实系统状态细分为真实系统中工作域进展情况和产品信息状态。由于仿真系统与真实系统在工作域的划分完全对应,因此可直接将真实系统中已经完成的工作域状态和输入输出等信息直接映射至仿真情景中。同时产品信息按功能、结构等属性进行划分并映射至仿真情景的产品模型中。由于真实系统发展对仿真系统的影响并不是全局的,因此仿真情景是在原有情景上的继承和发展,这里体现为情景版本的变化。

3 情景驱动的仿真执行

鉴于多 Agent 系统具有分散自治性、网络合作性、结构开放性、智能性以及对已有软件系统的重用等重要特征,采用多 Agent 技术以递阶控制结构建立情景驱动的仿真框架,如图 2 所示。

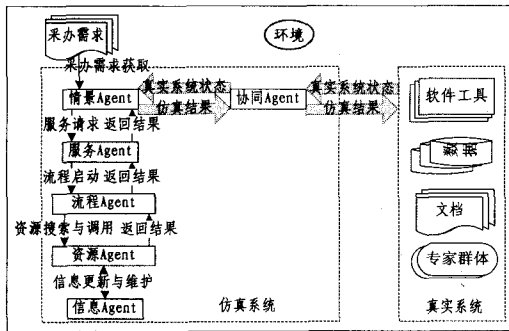


图2 情景驱动的仿真框架

图2中,仿真系统与真实系统共享运行环境,处于并行地位,两者通过协同 Agent 进行交互。仿真系统由情景 Agent、服务 Agent、流程 Agent、资源 Agent 和信息 Agent 等形成多层控制结构。

1)协同 Agent:是仿真系统与真实系统实时交互的接口,负责监控真实系统的进展情况,且能识别真实系统状态的变化并发送相关信息至仿真系统。协同 Agent 还负责接收仿真

系统的结果信息并发送至真实系统。

2)情景 Agent:是仿真系统运行的始端,监控和接收来自环境的产品需求和来自协同 Agent 的真实系统状态,并相应地生成仿真情景。情景 Agent 的功能主要包括情景版本维护,即根据情景影响因素的不同分别为新版本生成、版本继承以及不同版本情景之间的比对分析;情景状态维护,包括情景初始化、仿真启动和根据仿真进展更新情景状态;产品需求提取和映射则是对产品需求进行分类整理,划分为进度、费用、风险和性能等不同维的需求,并相应地映射至仿真情景中产品模型的需求维,同时对情景的仿真约束进行维护;工作域信息提取和映射,即对真实系统的工作域进展情况提取各工作域的状态和输入输出并相应地映射至仿真情景的工作域模型中;产品信息提取和映射,是对真实系统提供的产品信息按仿真情景定义的产品模型划分功能、结构属性并更新至情景模型中;情景输出的收集,是指根据情景进行仿真结果的收集和分析;与协同 Agent 的交互,负责从协同 Agent 接收真实系统的状态信息,并将仿真结果发送给协同 Agent。

3)服务 Agent:仿真情景生成后,启动该情景对应的仿真,按照该情景的工作域模型,执行其中的工作域仿真。在此,仿真服务可认为是工作域仿真的执行序列。因此,服务 Agent 的主要功能即为工作域仿真的启动以及工作域状态和输入输出的维护。服务 Agent 另一重要功能是定义工作域仿真的流程。产品全生命周期管理的最终交付成果为产品,而其仿真是对真实系统提出的需求进行的模拟和论证,这种模拟和论证主要以产品方案为基准,并对产品方案进行性能、制造、维护、销毁等仿真。随着真实系统的进展,产品方案从概念方案至详细设计逐步细化,系统的仿真尺度也相应地经历了一个由粗变细的过程。仿真尺度的变化可通过仿真流程的逐步深入得以体现,因此在仿真执行过程中应能动态定义不同尺度的仿真流程。根据产品模型中的结构属性生成仿真流程,且仿真流程具有多层结构,实际上体现了仿真模型多尺度的适应性。

4)流程 Agent:主要负责完成仿真流程的启动和监控,并根据组成流程的任务的需要发送仿真资源请求。

5)资源 Agent:负责系统中资源的全生命周期管理,包括资源注册、资源请求、资源搜索、资源分配、资源使用及资源注销等,并搜索资源请求和匹配合适的仿真资源,负责该资源的执行和监控。这里的资源包括专家资源、仿真资源、信息资源和真实系统资源。专家资源的使用体现了人机交互的动态过程,真实资源的使用体现了仿真系统与真实系统的结合能力,信息资源的使用则实现了真实系统信息和历史数据的搜索与获取。资源 Agent 根据资源请求搜索可用资源并进行组合重用。

6)信息 Agent:仿真资源的执行涉及仿真信息的更新和维护,信息 Agent 则是对特征信息操作模式的分类和规范化,并根据特征信息操作模式进行特征信息操作。特征信息的选择与重组分为 3 种:映射、聚集和转化。

a)映射:特征信息映射出现在同一尺度不同任务之间。同尺度的任务间通常存在依赖关系,即一个任务完成后才能开始其后继任务。从信息角度,则体现为一个任务的输出信息映射为后继任务的输入信息。这里,特征信息映射是指任

务的输出信息转化为其同尺度后继任务的输入信息的过程。

b) 聚集: 任务模型和产品模型都具有层次性的特点, 不同层次具有不同的尺度。不同尺度实体的特征信息间存在聚集关系, 即上层特征信息由下层特征信息组成。特征信息的聚集因维而异, 且聚集操作只发生在同一维上。对于产品模型的需求、功能、结构等属性, 聚集关系体现为层次包含关系; 而对于任务模型中的进度、费用、风险等维, 这种聚集关系通常不能简单加和。例如, 设父层任务 a 有 n 个相邻子层任务 $\{b_1, b_2, \dots, b_n\}$, 则有由任务 $\{b_1, b_2, \dots, b_n\}$ 组成的流程模型中从始点到终点、沿箭头方向总时差为零的关键任务的持续时间总和即为 a 的工期, 而 a 的费用为各子层任务费用的总和。

c) 转化: 特征信息转化针对某一特定任务, 是指某维特征信息向另一维特征信息演变的过程。例如, 费用被定义为任务执行过程中使用各种资源所产生的费用, 根据进度仿真中各任务的持续时间以及资源使用情况, 可计算该任务产生的费用。

按上述过程完成仿真情景所要求的仿真后, 对仿真结果进行收集、分析和更新至仿真情景的输出中; 并可由协同 Agent 将仿真结果反馈给真实系统以对真实系统的运行提供支持。

4 仿真资源组合与重用

PLM 使用的模型和工具不论是种类还是数量都会较多, 如果对每一个建模和仿真需求单独开发工具, 将费时费力且效果不会理想。所以通过已有仿真模型和工具的重用来实现仿真的组合, 具有现实价值和意义。仿真组合机制如图 3 所示。

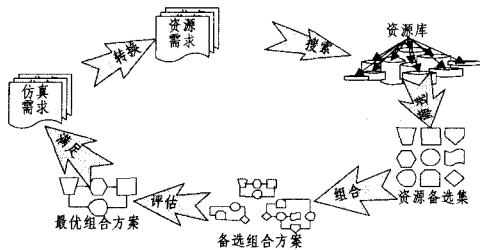


图 3 仿真组合机制

图 3 中, 首先提出仿真需求, 分析需求约束, 并对需求约束进行优先级排序; 然后将仿真需求转换成仿真资源功能的标识, 并根据这些标识从资源库中选取符合的仿真资源, 形成资源备选集; 再对选取的仿真资源按一定组合策略进行组合, 并评估不同的组合方案; 最后从备选组合方案选取最佳组合进行仿真。以上过程是一个多次循环、螺旋上升的过程, 以最大可能地满足仿真需求。

对应每一个仿真需求, 定义仿真耦合模型 $N = (X, Y, \{Resource_i\}, \{I_i\}, \{Z_{ij}\}, F_s)$ 。其中, X 和 Y 分别表示耦合模型的输入和输出集合, $\{Resource_i\}$ 为资源备选集, I_i 表示资源备选集中受第 i 个资源影响的资源的下标集, Z_{ij} 为第 i 个资源与第 j 个资源的耦合关系, F_s 为选择函数。耦合关系包括: 外部输入耦合, 将耦合模型的输入映射为其资源的输入; 外部输出耦合, 将资源的输出映射为耦合模型的输出; 内部耦合, 将资源的输出映射为其他资源的输入。

基于耦合模型的仿真组合策略如图 4 所示。

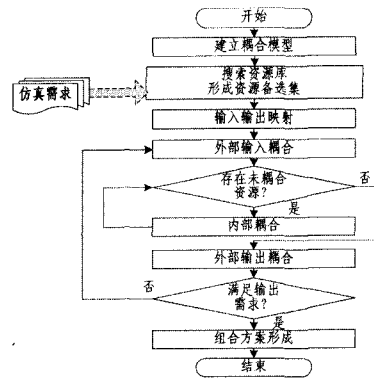


图 4 仿真组合策略

1) 根据资源需求的约束条件搜索资源库, 匹配并形成耦合模型的资源备选集, 同时将资源需求中的输入和输出映射为耦合模型的输入和输出集合。

2) 从耦合模型向资源进行外部输入耦合操作, 接受外部输入耦合的资源须满足其输入集为耦合模型输入集的子集。

3) 对已经完成耦合的资源, 合并最末端资源的输出集, 接着完成该合并输出集向其他资源的内部耦合; 接受内部耦合的资源, 其输入集应是合并输出集的子集。重复本步骤直至所有资源被耦合。

4) 完成以上耦合操作后, 合并最末端资源的输出集, 并完成该输出集的外部输出耦合。如果耦合模型的输出集为合并输出集的子集, 则外部输出耦合满足所有输出要求, 选择存在于耦合模型的输入集到输出集的可达路径上的资源, 形成组合方案; 否则须返回步骤 2) 重新进行。

5 系统设计与实现

为了具有良好的可扩展性、可维护性和支持异构网络的操作, 基于面向服务架构 (Service-Oriented Architecture, SOA), 提出一个实现框架, 如图 5 所示。系统主要包括以下 5 类组成部分:

1) 协同控制管理支撑服务平台, 依托 Web 技术构建系统所需要的核心支撑管理和协同服务, 综合运用计算机网络技术、数据库技术、虚拟现实技术、分布式人工智能技术和群决策支持技术等先进技术成果, 实现产品全生命周期内各类资源服务的动态重组、无缝集成和按需取用。

2) 核心服务, 包含与解决问题相关的数据、模型、知识、专家、工具等, 主要包括产品管理、产品仿真、研讨管理和结论管理等。其中, “产品管理”对从需求提出到结论形成的产品开发全过程进行宏观控制; “产品仿真”通过建模仿真提供一种预测和决策支持; “研讨管理”则实现一种螺旋式上升的、从定性到定量的逐次逼近方式, 快速、柔性、低风险、低成本地求解问题新模式。

3) 系统服务, 使用信息搜索技术和 Agent 技术提高系统分布式信息管理与自主智能的能力。信息搜索服务主要对 PLM 的本地资源和 Web 资源进行有效整合, 使不同形式的信息相互补充, 有机地融合在 PLM 过程中。Agent 服务通过提供多种类型 Agent 提高系统的智能、分布式、反应性、移动性、人机交互与协作的能力。

(下转第 257 页)

变[J]. 经济学家, 2007, 18(2): 21-26

[21] 柴盈, 何自力. 当议现代经济学的理性假设[J]. 安徽大学学报: 哲学社会科学版, 2007, 31(2): 131-137

[22] 张荣楠. 从有限理性到适应性理性[J]. 经济社会体制比较, 2004, 116(2): 79-84

[23] Kahneman D. Maps of Bounded Rationality; a Perspective on Intuitive Judgment and Choice [R]. Prize Lecture, December 2002

[24] Tversky A. Contrasting Rational and Psychological Principles of Choice [M] // Richard J. Zeckhauser, Ralph L. Keeney, and James K. Sebenius, eds. Wise Choices: Decisions, Games, and Negotiations. Harvard: Harvard Business School Press, 1996: 5-21

[25] 杨小凯. 不完全信息与有限理性的差别[J]. 开放时代, 2002, 21(03): 76-81

[26] Simon H A. Bounded Rationality in Social Science: Today and Tomorrow [J]. Mind & Society, 2000, 1(1): 25-39

[27] 赫伯特·西蒙. 管理行为-管理组织决策过程的研究[M]. 北京: 北京经济学院出版社, 1988

[28] Lupia, Arthur, McCubbins M D, et al. Beyond Rationality: Rea-

son and the Study of Politics [M] // Arthur Lupia, Mathew D. McCubbins, and Samuel L. Popkin, eds. Elements of Reason: Cognition, Choice and the Bounds of Rationality. New York: Cambridge University Press, 2000: 1-20

[29] Stigler S M. The history of statistics; The measurement of uncertainty before 1900 [M]. Cambridge, MA: Belknap Press, 1986

[30] 仪伟林. 中国证券市场价格波动: 基于投资者情结与决策理论的研究 [D]. 南京: 河海大学, 2006

[31] 张学军. 决策者有限理性的心理根源探析[J]. 电子科技大学学报: 社科版, 2008, 10(3): 64-67

[32] 吴力专, 张国锋. 悖论问题情绪机制消解[J]. 安徽大学学报: 哲学社会科学版, 2008, 32(5): 46-50

[33] 张国锋, 李祖枢. 基于情绪的人工生命行为选择机制研究[J]. 系统仿真学报, 2009, 21(6): 1701-1705, 1709

[34] 张国锋, 李祖枢. 基于情绪的人工生命合作型行为选择机制研究[J]. 信息与控制, 2009, 38(6): 718-723

[35] 张国锋, 李祖枢. 战斗行为情绪机制[J]. 计算机工程与应用, 2009, 45(18): 28-31

(上接第 235 页)

4) 系统资源, 描述 PLM 中所需的各种资源, 包括仿真资源、真实资源、专家资源、信息资源以及其他支持工具集等。

5) 人机交互系统, 采用 B/S 结构进行设计部署, 将 Web 浏览器作为用户与系统交互的界面, 少量逻辑处理在前端实现, 主要逻辑处理在服务器端实现, 并且允许用户对异构平台服务器中的信息进行访问。

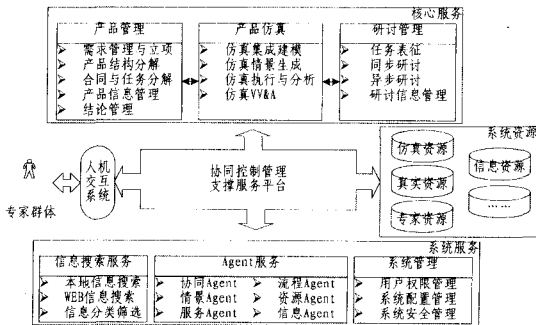


图 5 系统实现框架

本框架采用 Web 服务方式和 XML 统一表达方式, 按 B/S 体系结构提供服务, 能够适应环境和主题变化, 具有自学习能力, 能及时更新模型工具, 具有良好的可扩展性。在操作方式、运行环境需做某些变更时, 本框架具有一定的适应能力, 并且由于采用 Web 服务方式, 可以在不影响系统正常运行的前提下新增、修改、删除服务和资源, 从而降低系统功能耦合性的要求, 提高操作的灵活性。

结束语 随着产品的复杂化, 服务于 PLM 的建模仿真对环境变化和突发事件适应能力的要求越来越高。本文将与仿真系统平等和并行发展的真实系统纳入建模范围, 真实系统的进展情况和原始产品的需求变化作为仿真系统变化的诱因, 利用仿真情景驱动仿真的执行。在此思路指导下, 建立以情景为驱动的分层模型结构, 将真实系统对仿真系统的影响细化为工作域和仿真尺度的变化, 通过仿真情景描述这种变化并驱动仿真执行, 动态地生成不同尺度的仿真流程, 并根据仿真需求动态选择和组合仿真模型, 从而提高仿真系统对环境变化的适应性, 以便为产品全生命周期管理的综合决策提供有效支持。

参 考 文 献

[1] Stark J. Product Lifecycle Management; 21st Century Paradigm for Product Realisation [M]. Berlin: Springer Verlag, 2005

[2] 韩守鹏, 邱晓刚, 黄柯棣. 动态数据驱动的适应性建模与仿真 [J]. 系统仿真学报, 2006, 18(2): 147-151

[3] 周云, 黄柯棣, 胡德文. 动态数据驱动应用系统的概念研究 [J]. 系统仿真学报, 2009, 21(8): 2138-2141

[4] Kasputis S, Ng H C. Composable Simulations [C] // Proceedings of the 2000 Winter Simulation Conference. New York: ACM Press, 2000: 1577-1584

[5] Timpf S. Map Cube Model-a Model for Multi-scale Data [C] // Proceedings of 8th International Symposium on Spatial Data Handling. Vancouver: International Geographical Union, 1998: 190-201

[6] Zeigler B P, Luh C J, Kim T G. Model Base Management for Multifaceted Systems [J]. ACM Transactions on Modeling and Computer Simulation (TOMACS), 1991, 1(3): 195-218

[7] Kornman B D, Marion P B. Cross-Model Consistency in JSIMS [R]. Norfolk: Lockheed Martin Information Systems Advanced Simulation Laboratory, 1998

[8] Van Der Aalst W, Ter Hofstede A. Verification of Workflow Task Structures: A Petri-net-based Approach [J]. Information Systems, 2000, 25(1): 43-69

[9] Van der Aalst W, Dumas M, Ter Hofstede A. Web Service Composition Languages: Old Wine in New Bottles? [C] // Proceedings of 29th Euromicro Conference. IEEE Computer Society Press, 2003: 298-307

[10] Held M, Blochinger W. Structured Collaborative Workflow Design [J]. Future Generation Computer Systems, 2009, 25(6): 638-653

[11] Seungchul H, Hyo-Won S. A Timed Colored Petri Nets Modeling For Dynamic Workflow In Product Development Process [J]. Computers in Industry, 2008, 59(2/3): 193-209

[12] Berardi D, Calvanese D, De Giacomo G, et al. Automatic Service Composition Based on Behavioral Descriptions [J]. International Journal of Cooperative Information Systems, 2005, 14(4): 333-376

[13] Salaun G, Bordeaux L, Schaerf M. Describing and Reasoning on Web Services Using Process Algebra [J]. International Journal of Business Process Integration and Management, 2006, 1(2): 116-128