

基于 XYZ/ADL 的网络中心化仿真运行支撑 平台体系结构形式化描述

孙黎阳^{1,2} 毛少杰² 林剑柠² 刘 中¹

(南京理工大学电子与光技术学院 南京 210094)¹

(中国电子科技集团 28 所信息系统工程重点实验室 南京 210007)²

摘要 网络中心化仿真运行支撑平台体系结构是为了满足动态构建仿真任务共同体而提出的相应支撑环境。采用基于 XYZ/ADL 的双重软件体系结构描述框架,分别从图形语言和形式语言对仿真运行支撑平台体系结构进行了描述并对仿真任务共同体构建中仿真核心服务层实施了分解与逐步求精。该体系结构描述方法不但从可视化角度表示了仿真运行支撑平台体系结构框图及行为抽象,还从形式化角度验证了体系结构设计的正确性和完整性。该研究是形式化描述方法在军事领域的全新尝试,为仿真运行支撑平台服务的合成与重用机制提供了形式化的理论指导。

关键词 网络中心化仿真,任务共同体,运行支撑平台体系结构,体系结构描述语言,XYZ/ADL

中图分类号 TP391 **文献标识码** A

Formal Description of Simulation Runtime Support Platform Architecture with XYZ/ADL

SUN Li-yang^{1,2} MAO Shao-jie² LIN Jian-ning² LIU Zhong¹

(School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)¹

(Science and Technology on Information System Engineering Laboratory, The 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, China)²

Abstract Net-centric simulation runtime support platform(NCS-RSP) provides an environment supporting the construction of community of simulation task(CoST). This paper adopted dual software architecture description framework XYZ/ADL to describe the architecture of NCS-RSP by graphics language and formal language respectively. Then we decomposed and refined the core service layer during the construction of CoST. This description method not only expresses the architecture graphics and behavioral abstraction of NCS-RSP from visual viewpoint, but also validates the correctness and completeness of architecture design from formal view. The research is a new attempt of formal description in military simulation domain and it provides a guide for the composition and reuse of NCS-RSP service.

Keywords Net-centric simulation, CoST, Runtime support platform architecture, Architecture description language, XYZ/ADL

1 引言

网络中心化仿真概念和方法是针对军事领域网络中心战时代仿真应用需求提出的,是一种带有领域特征的网络化仿真方法,其主要内涵是以面向服务思想为指导,采用统一的仿真资源描述、接入和共享标准,依托军事信息栅格基础设施,建立仿真资源互联共享的环境,以动态创建和运行仿真任务共同体的方式完成各种仿真应用任务^[1]。网络中心化仿真目标是通过仿真与军事信息栅格技术的结合,建立仿真与 C⁴I 系统兼容的体系结构,真正实现仿真与 C⁴I 系统的无缝集成和互操作,使建模与仿真应用成为 C⁴I 系统能力包的组成部分。网络中心化仿真运行支撑平台(Net Centric Simulation- Runtime Support Platform, NCS-RSP)是实现网络中心化仿真方法的核心技术,其为动态按需构建仿真任务共同体提供了必需的支撑能力。仿真任务共同体是指为了完成共同的仿真任务,由分布在网络上的相关仿真资源和设施组成的“虚拟组织”,其通过 NCS-RSP 相关服务以及仿真资源动态组合,运行在网络中心化环境下,是一种服务组合技术。

体系结构描述语言(architecture description language, ADL)是软件体系结构开发的基础。它具有坚实的数学基础,通过形式规约给出,提供了诸如一致性、完备性、规约、实现、正确性等概念的精确定义,以便人们以系统的方法来设计、开发和验证软件。

在软件设计阶段,如何对模型进行有效的分析和验证是最终实现高可靠性软件的重要前提,也是当前软件设计阶段急需解决的问题。该问题的一种有效解决途径是将软件工程中常用的描述方法(图形建模)与形式化方法(严格的语义及验证)相结合。

针对形式化语言对软件体系结构设计的诸多好处,本文对众多形式化语言进行了分析和比较,介绍了多种形式化语言的特点;接着选用 XYZ/ADL 语言,对其进行了介绍并用其描述 NCS-RSP 体系结构。通过 XYZ/ADL 描述体系结构,不仅提供了直观的框图描述及行为抽象,更为动态构建仿真任务共同体提供语义层分析和验证。采用 XYZ/ADL 体系结构描述框架思路,可以发现,形式化语言一方面可以精确、完整、无歧义地描述体系结构,避免了文字描述的烦冗以及可能造

成的非准确性；另一方面又可对从可视化角度提出的体系结构模型进行分析并验证其完整性与设计正确性。这是服务组合技术与形式化语言在军事仿真领域的发展与延伸。

2 相关工作

2.1 形式化语言

从软件体系结构研究和应用的现状来看,当前对软件体系结构的描述,在很大程度上来说还停留在非形式化的基础上,其很大程度上依赖于软件设计师个人的经验和技巧。这种描述方法难以被开发人员理解,难以适应进行形式化分析和模拟,缺乏相应的支持工具帮助设计师完成设计工作,更不能用来分析其一致性和完整性等特性。相对于非形式化的表示模式,形式化的描述方法相对而言较为精准、完备且是无歧义的。体系结构描述语言 ADL 就是以形式化描述方法为目标的研究成果。体系结构描述语言 ADL 吸收了传统程序设计中语义严格、精确的特点,并针对软件体系结构的整体性和抽象性,定义和确定了适合于软件体系结构表达与描述的有关抽象元素,从而能精确、无歧义地描述软件体系结构,更好地支持软件体系结构的求精、验证、演化和分析。

现已提出了若干适用于特定领域的 ADL,典型的有: C2^[3] 是一种基于构件和消息的 ADL,适用于大型频繁交互的层次型图形用户界面的软件体系结构描述;Rapid^[4] 是一种基于事件的可视化体系结构描述语言,适用于对分布式体系的体系结构的建模;Darwing^[5] 和 Wright^[6] 分别以 π 演算和 CSP 为数学基础,适用于分布、并发类型的体系结构描述;其他比较有影响的 ADL 有 Aesop, ArTek, SADL, UniCon, Weaves 等。国内一些学者也提出了几种比较有特色的 ADL,例如多智能体体系结构描述语言 A-ADL^[7] 等。文献[8]描述了一个 ADL 分类框架,该文献详细分析了多种典型 ADL 的优点与不足。该观点指出了体系结构描绘的多层次、多视图和代码生成的重要性。

2.2 XYZ/ADL

XYZ/E^[9] 是中国科学院软件研究所唐稚松先生提出的世界上第一个可执行时序逻辑语言,它基于 Manna-Pnueli 命题线性时序逻辑系统,将时序逻辑算子融进程序设计语言之中,使其既是一个逻辑系统又是一个程序设计语言,支持实时、混成、通信及可视化等程序设计方法。其表达方式与普通高级语言相似,因而易于为工程师和程序员所接受。同时作为一种形式化语言,它既可表达以状态转换为特征的动态语义,又可表达以前后断言及递归函数为特征的静态语义。XYZ/E 的最大特点是能在统一的语义框架中表示从形式规范到可执行程序的不同抽象层次的系统描述,即既能表示程序的动态语义,又能表示规范的静态语义。这个特点使其很适合用于描述软件体系结构。

XYZ/ADL 是一种基于可执行时序逻辑语言 XYZ/E 的软件体系结构描述语言,它能够在统一的时序逻辑框架下表示从形式规范到可执行程序的不同抽象层次的系统描述,既可以表示软件体系结构的动态语义,又可以表示其静态语义。XYZ/ADL 的这一特色弥补了现有的 ADL 不足之处——难以同时描述和分析体系结构的静态与动态语义,如 CSP 较适合于动态行为的描述,而 Z 适合描述静态性质;而且它们不具有可执行性,一般都把对体系结构的规范描述和结构实现截

然分开,难以对规范进行逐步求精,不能有效地支持基于体系结构的软件开发全过程。XYZ/ADL 不仅可用于对软件体系结构进行形式化描述,也可用于对不同抽象层次的体系结构进行求精,而且还可以应用 XYZ 系统所提供的验证工具检验求精过程中的语义一致性。XYZ/ADL 形式化语言提出之后,受到了很多学者的关注及深入研究,并将其运用到实际项目中,使其得到进一步发展。文献[10-15]列举了 XYZ/ADL 语言部分的研究成果及应用。

文献[16]对 XYZ/ADL 进行了更深一步研究,提出了双重软件体系结构描述框架。该框架支持工程中软件体系结构的基本概念,前端用一般的体系结构框图作为结构描述,用 UML 活动图、状态图作为抽象行为表示;后端用既可表示系统动态语义又可表示系统静态语义的时序逻辑语言 XYZ/E 作为一致的语义基础。前端的图形语言便于软件工程师的交流和使用,后端的形式语言是进一步的形式化分析验证的基础。本文采用文献[16]思路,先从前端角度给出了体系结构框图,以此对 NCS-RSP 进行结构描述;除此之外针对 NCS-RSP 中仿真核心服务层,我们采用 UML 顺序图对 NCS-RSP 行为进行抽象表示,先说明为何采用顺序图,再对仿真核心服务层抽象行为进行文字描述;最后在后端用形式化语言 XYZ/ADL 对仿真核心服务层服务交互进行进一步形式化描述并对其服务交互进行逐步求精,从语义层面对其进行分析。

3 NCS-RSP 体系结构图形语言描述

3.1 框图描述

为了满足网络中心化仿真各种能力要求,实现随时随地按需建立各类仿真服务,我们构建了相应的支撑环境,提出了一种 NCS-RSP 体系结构,该体系结构是一种面向服务的层次化体系结构,为按需动态构建仿真任务共同体提供支撑能力。NCS-RSP 体系结构如图 1 所示。文献[1,2]对该体系结构各层内部具体服务功能以及各层之间的相互支撑关系进行了详细描述。

3.2 UML 顺序图描述

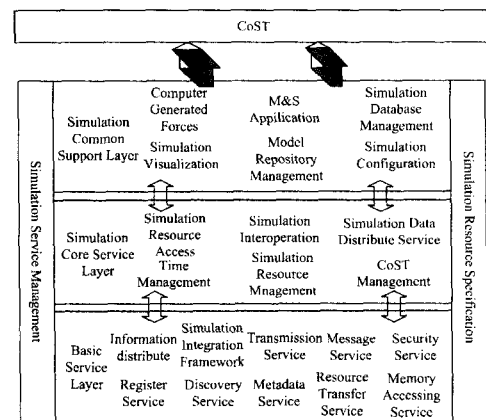


图 1 NCS-RSP 体系结构

上述提出的体系结构框图从可视化角度说明了 NCS-RSP 如何对仿真任务共同体的构建与运行提供支撑。以仿真核心服务层对任务共同体的支撑为例,当动态构建仿真任务共同体后,将通过网络中心化仿真环境运行该任务共同体。一般可以通过仿真任务共同体管理服务发出仿真开始命令,通过仿真数据分发服务将其发送给仿真任务共同体内所有的

成员,所有成员收到仿真开始命令后开始执行仿真任务。当仿真任务完成时,仿真任务共同体管理服务向所有成员发出仿真结束命令,仿真任务共同体停止运行。在整个仿真任务共同体运行过程中,都受到仿真任务共同体管理服务以及仿真资源管理服务的支撑。在仿真任务共同体运行过程中,网络中心化仿真环境中的其它成员可以向仿真管理申请随时加入仿真任务共同体,仿真任务共同体内的成员可以申请随时退出。网络中心化仿真环境上可以同时运行任意多个仿真任务共同体,不同的仿真任务共同体可以使用不同的仿真时钟管理策略、仿真步长等。此时需要时间管理,仿真交互等服务支撑。除了上述对体系结构进行框图描述,在 NCS-RSP 前端分析中,

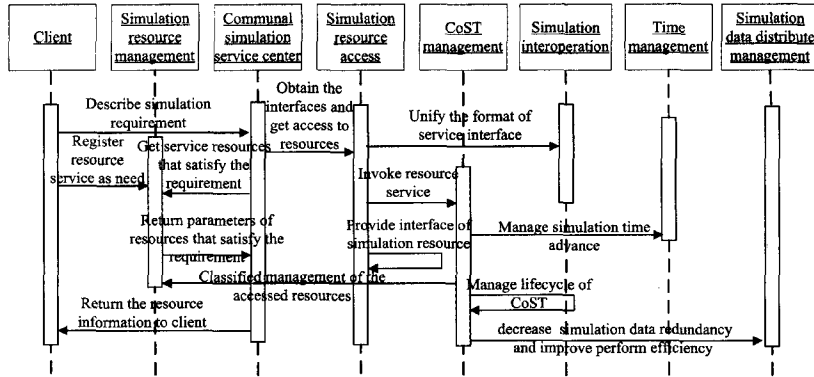


图 2 NCS-RSP 核心服务层 UML 顺序图

4 NCS-RSP 体系结构形式语言描述

4.1 仿真核心服务层形式化描述

在前面一节,通过 UML 顺序图对 NCS-RSP 体系结构中仿真核心服务层进行了行为抽象,并且通过文字对该层服务交互进行了描述。这样的描述复杂冗余,并不能被用户轻易理解及接受。因此本节采用 XYZ/ADL 语言,对仿真核心服务层 UML 顺序图进行形式化描述。在形式化描述之后,我们还将对仿真核心服务层的时间管理服务与仿真任务共同体管理服务交互过程进行逐步求精。通过本节介绍,不仅可以形式化语言清晰准确地理解仿真核心服务层交互关系,而且可以通过形式化描述来验证软件体系结构设计的正确性及完整性。

仿真核心服务层有多种服务,每种服务均作为一个服务组件存在并发挥作用,这使得该层服务交互较为复杂。而在 NCS-RSP 中,仿真核心服务层只是其中一部分,其也作为组件,与仿真基础服务层以及仿真共性支撑层有交互关系。因此,本节将仿真核心服务层作为由多种组件根据一定要求连接而成的复合组件,并对其结构模型进行描述,即对仿真核心服务层组件的第一步求精,它反映了其内部主要组件之间的信号传输关系。

复合组件是由一些组件根据一定的要求连接而成的,复合组件则在横向上体现了组件之间的体系结构配置关系,在纵向上体现了组件之间的层次关系。复合组件的接口描述与简单组件相同,而它的行为是通过几个子组件的连接来表示的。所以,在描述中我们必须说明一个复合组件包含哪些组件和连接件实例,它们是如何连接在一起的。针对仿真核心服务层中各项服务内容,我们定义如下子组件: SimResMan、SimAccess、CoSTMan、SimInter、TimeMan、DataDistri。

我们还将采用 UML 来对 NCS-RSP 行为进行抽象描述。

NCS-RSP 由众多服务构成,为了显示多个服务对象之间的交互情况,用状态图或活动图都不适用于描述 NCS-RSP 行为;状态图是为了显示一个对象在其生命周期内的行为;活动图是为了分析用例,理解多个用例的工作流程或处理多线程应用等,但如果要显示多个对象之间的交互情况,用状态图或活动图都不适合,此时可以用顺序图来描述。因此我们采用 UML 顺序图来对平台行为进行抽象。受篇幅所限,我们仅在此给出 NCS-RSP 核心服务层 UML 顺序图,以此来说明仿真核心服务层内各项服务之间的交互关系,如图 2 所示,其余层 UML 图也可类似给出。

仿真核心服务层复合组件的组合声明指示其包含哪些组件和连接件实例,语法如图 3 所示。

```

Description of Core Service Layer
%COMPONENT CoreService == [
%PORT Receive1 == MESSAGE;
  □ [LB = START_receive1=>Receive1 ? ComService ∧ $O LB = e1]
  //receive messages from common support layer
%PORT Receive2 == MESSAGE;
  □ [LB = START_receive2=>Receive2 ? BasicService ∧ $O LB = e1]
  //receive messages from basic service layer
%PORT Send1 == MESSAGE;
  □ [LB = e1=> Send1| state1 ∧ $O LB = e1]
  //send messages to common support layer
%PORT Send2 == MESSAGE;
  □ [LB = e2=> Send2| State2 ∧ $O LB = e2]
  //end messages to basic service layer
%COMPOSITION == [SimResMan; SimAccess; CoSTMan; SimInter; TimeMan; DataDistri;
  c1:Conne1; c2:Conne2; c3:Conne3; c4:Conne4; c5:Conne5]
%ATTACHMENT == [ [ CoSTMan.Send1 # c1.Source; SimResMan.Receive1 # c1.Sink;
  SimAccess.Send1 # c2.Source; SimInter.Receive1 # c2.Sink;
  SimAccess.Send2 # c3.Source; CoSTMan.Receive1 # c3.Sink;
  CoSTMan.Send2 # c4.Source; TimeMan.Receive1 # c4.Sink;
  CoSTMan.Send3 # c5.Source; DataDistri.Receive1 # c5.Sink;
  CoSTMan.Send1##Send1; SimResMan.Send2##Send2;
  CoSTMan.Receive1##Receive1; SimResMan.Receive2##Receive2 ]
%COMPUTATION == [
LB = START_receive1=> $O LB = e1;
LB = e1=> $O Receive1 = CoSTMan.Receive1 ∧ $O Receive2 = SimResMan.Receive2 ∧ $O
Send1 = CoSTMan.Send1 ∧ $O Send2 = SimResMan.Send2 ∧ $O LB =
e2;
LB = e2=> || [SimResMan.COMPUTATION; c1.GLUE; SimAccess.COMPUTATION;
c2.GLUE; CoSTMan.COMPUTATION; c3.GLUE; SimInter.COMPUTATION;
c4.GLUE; TimeMan.COMPUTATION; c5.GLUE; DataDistri.COMPUTATION ] ]

```

图 3 NCS-RSP 核心服务层形式化描述

图 3 是对仿真核心服务层复合组件的第一步求精过程,也是初步的形式化验证过程。首先定义了复合组件输入输出端口;接着按仿真核心服务层内部服务组合规则,对复合组件中子组件端口与连接件角色进行了粘贴与绑定操作,其中“#”定义了粘接操作,表示左边的组件端口与右边的连接件角色相粘接,从中可以看出哪个组件参与了哪一种交互。有些子组件的端口没有和某个角色相粘接,而是被指定为复合组件的端口,这种绑定操作由“##”定义,右式是该复合组件的端口,因此将组件名省略;最后在%COMPUTATION部分对复合组件语义进行了解释,即初步求精。

4.2 逐步求精

文献[17]介绍了软件体系结构的3种求精方法:基于行为替代的体系结构求精、基于风格的体系结构求精和基于构件的体系结构求精。基于构件的体系结构求精过程可以演化为两个过程:从粗粒度构件向细粒度构件的求精;从非形式化到形式化的求精。其基本思路是:确定了风格的体系结构级粗粒度构件按照规范说明,划分成多个构件,构件又可以按照需求确定自身的子风格,逐步形成细粒度构件。

通过上节的初步求精,我们知道了核心服务层各服务之间的交互关系。但是这种描述只是对抽象的软件体系结构的基本描述。此节中,我们将采用基于构件的体系结构求精方法来建立核心服务层体系结构的模型。这种求精方法将在实现细节上比先前初次求精更加精确及复杂。求精到该层,不仅可以基本达到可执行程序的程度,更能全面验证体系结构设计正确性与完整性。由于篇幅限制,这里仅给出时间管理服务组件与仿真任务共同体管理组件的交互求精过程。

4.2.1 服务组件描述

以核心服务层时间管理服务为例,在图4中定义Time_Management组件,该组件有一个端口TimeServiceport,类似还可以定义任务共同体管理服务组件CoST_Management及其端口CoSTServiceport等。

按照组件定义方法,可以定义其他相关服务,由于篇幅限制,这里不再赘述。

```
Description of Time Management Component
%COMPONENT Time_Management == [
%PORT TimeServiceport = timeserviceprovide (% Recive1 int identifyID;
% Recive2 DWORD timestep;
% Send1 BOOL block;
% Send2 BOOL AdvState)
//define messages in TimeServiceport port
//define advance node ID, time step, deadlock identifier, time advance state
//similarly define messages timeservicereceive
];
//define operations in TimeServiceport port
[timedatareceive [Required; Type: Input; Message: timeservicereceive];
timedataprovide [Provided; Type: Output; Message: timeserviceprovide]];
//define behavior in TimeServiceport port
[//provide messages to CoST management and receive call back information]
%SERVICEBEHAVIOR == [//component behavior is the same as port behavior]
%PROPERTY == Resource Specification
//service components should meet the specification requirement
```

图4 时间管理服务组件形式化描述

4.2.2 服务连接件描述

```
Description of Time Management Connector
%CONNECTOR Time_ManagementConn == [
//define CoSTManagementRole messages: costmanagementresponse.
costmanagementrequest;
TimeServiceRole messages: timeserviceprovide, timeservicereceive
//define operations of CoSTManagementRole: Serviceinvoke, ReturnServiceinvokeMessage;
Operations of TimeServiceRole: timedatareceive, timedataprovide
%ROLE CoSTManagementRole == [costmanagementresponse; costmanagementrequest];
[Provided, Operation; Serviceinvoke;
Required_Operation; ReturnServiceinvokeMessage];
[//receive service calls information and return results]
//define behavior of connector role
//similarly define role CoSTManagementRole
%GLUE == [//describe interoperation of two roles: provide time management information by
timedataprovide and execute time management service]
```

图5 时间管理服务连接件形式化描述

在本节中,我们描述了时间管理服务与任务共同体服务是如何交互的。它们通过XYZ/ADL定义的连接件来进行连接。图5定义描述了时间管理连接件Time_ManagementConn。

4.2.3 逐步求精

从上面NCS-RSP体系结构的研究可以看出,仅用文字描述很难将问题说清,而用图形描述不仅很直观,而且也比文字说明易于理解。例如图1、图2可以使仿真NCS-RSP体系结构以及其仿真核心服务层整体布局变得清晰易懂。但是仅有图形的描述也难以分析,例如对于核心服务层体系结构,图形描述只能给出所含各项服务内容,不但无法确切知道其中交互关系,更无法对该层体系结构设计正确与否进行验证,因此,这些答案必须从严格的形式化语言描述中寻找。对于我们设计的NCS-RSP体系结构,使用形式化语言对其进行描述,成功从最高层抽象的体系结构逐步求精到足够具体的体系结构,形成了一个体系结构层次,每步求精对应一个求精模式,这样保证了求精的正确性,从而也验证了我们体系结构设计的正确。以时间管理服务为例,其形式化求精如图6所示。

```
Refinement of Time Management Service
%COMPONENT Time_Management == [
%PORT TimeServiceport.Receive1 == MESSAGE;
[] [LB = START Receive1=>Receive1 ? ID ^ $O LB = re1]
%PORT TimeServiceport.Receive2 == MESSAGE;
[] [LB = START Receive2=>Receive2 ? TimeStep ^ $O LB = re2]
%PORT TimeServiceport.Send1 == MESSAGE;
[] [LB = START Send1=>Send1 ! ID ^ $O LB = se1]
%PORT TimeServiceport.Send2 == MESSAGE;
[] [LB = START Send2=>Send2 ! ID ^ $O LB = se2]
%COMPUTATION == [
LB = START ^ Receive1 ? ID ^ Receive2 ? TimeStep -> $O LB = zs;
//receive advance node ID and set simulation advance time step
LB = zs ^ ~ (ID = EOF) => $O (LB = z0 | LB = START);
//judge whether ID node matching
LB => initiate AdvTime ^ $O LB = z1;
//initialization: calculate advance time
LB = z1 ^ GALT => $O LB = z2; //calculate GALT
LB = z2 ^ (Isblock = ST) => (LB = z3 {z10});
//judge whether deadlock
LB = z10 => Send1 ! block ^ $O LB = z11;
LB = z3 => TimeAdvance ^ AdvState = Available ^ $O LB = z4;
//advance time and set advance state as advance available
LB = z4 => Send2 ! AdvState ^ $O LB = END]]
```

图6 时间管理服务求精

结束语 本文针对NCS-RSP,对其体系结构采用了一种基于XYZ/ADL的双重体系结构描述框架。首先给出了体系结构框图以及UML顺序图,从可视化角度描述了NCS-RSP体系结构;接着从形式化语义角度对体系结构中相关服务进行描述,并进行了逐层求精。通过对NCS-RSP体系结构的形式化研究,为NCS-RSP服务的合成与重用机制提供了形式化的理论指导,这是服务组合形式化描述在军事仿真领域一次全新的尝试。

我们认为,要使其在软件开发过程中真正发挥作用,一个ADL不仅应有严格的形式语义,还应兼顾工程使用习惯,并能够方便地表示不同抽象层次的体系结构。本文通过XYZ/ADL语言对仿真任务共同体系统组件进行逐步求精,实现了从可视化图形到形式化方法的平滑衔接,并通过连接件角色与组件端口的交互机制以及端口绑定,最终把系统形成一个有序组合的模板。实现了在军事仿真领域用形式化语言对NCS-RSP体系结构进行描述,从而支持对服务组合的语义分析,这是一次初步的尝试。

值得一提的是用UML作为软件体系结构描述的研究工具,由于UML在实际开发中的广泛应用,以及它的可扩充机制,用其作为体系结构行为描述还是很合适的。本文也采用了软件体系结构描述与UML相结合,采用UML的顺序图作为体系结构的行为描述。

本文在对 NCS-RSP 体系结构的形式化描述中,由于篇幅限制,只列举了其中具有代表性的相关服务进行描述,对于整个平台体系结构描述也可类似给出。时间管理是 NCS-RSP 的关键技术之一,文中对其进行求精只是给出了较为初步的时间管理算法,我们将在相关论文中对该服务进行详细完善的深入研究。

参考文献

[1] 毛少杰,李玉萍,林剑钊,等.网络中心化仿真概念与方法研究[J].系统仿真学报,2010,22(7):1660-1663

[2] Sun L Y, Liu Z, Mao S J, et al. Research on the Runtime Support Platform for the Net-Centric Simulation [J]. IEEE ICACTE Conference, 2010(1):253-257

[3] Taylor R N, Medvidovic N, et al. A component-and-message-based architecture style for GUI software[J]. IEEE Trans. Software Engineering, 1996, 22(6):390-406

[4] Luckham D C, et al. The Rapide language. Stanford University [EB/OL]. <http://pavg.stanford.edu/rapide/language.html>, 1997-07-29

[5] Magee J, Kramer J. Dynamic structure in software architectures[C]//Proc. the 4th ACM SIGSOFT Symposium on Foundations of Software Eng. New York: ACM Press, 1996:3-14

[6] Allen R, Garlan D. A formal basis for architectural connection [J]. ACM Trans. Software Eng, and Methodology, 1997, 6(3):

213-249

[7] 马俊涛,傅韶勇,刘积仁. A-Adl:一种多智能体系结构描述语言[J].软件学报,2000,11(10):1382-1389

[8] Medvidovic N, Taylor R N. A classification and comparison framework for software architecture description languages [J]. IEEE Trans. Software Engineering, 2000, 26(1):70-93

[9] 唐稚松,等.时序逻辑程序设计与软件工程[M].北京:科学出版社,2002

[10] 朱雪阳.软件体系结构形式描述研究[D].北京:中国科学院软件所,2004

[11] 陈琳琳,张广泉. XYZ/ADL 到 UML 的一种映射方法[J].苏州大学学报:自然科学版,2006,22(1):49-54

[12] 刘艳丽.基于 XYZ/ADL 描述的选修课管理系统体系结构[J].云南师范大学学报,2006,26(4):7-10

[13] 饶元,李尊朝.基于 XYZ/ADL 的 Web 服务体系结构描述[J].系统工程理论与实践,2006,3:56-60

[14] 杨敬中,戎玫,张广泉.面向方面的软件体系结构描述语言 AO-ADL[J].计算机工程,2008,34(10):80-82

[15] 戎玫,张广泉.形式化与可视化的相结合的软件体系结构描述方法研究[J].计算机科学,2005,32(4):205-208

[16] 朱雪阳.双重软件体系结构描述框架 XYZ/ADL[J].计算机研究与发展,2007,44(9):1485-1494

[17] 戎玫,张广泉.软件体系结构求精方法研究[J].计算机科学,2003,30(4):108-110

(上接第 334 页)

结束语 R/M 集成供应链风险预测是风险管理的一个重要内容,供应链风险事件发生的敏感性体现在每条供应链风险发生的规律都不尽相同,因此,在该供应链上已经发生的风险事件可以提供该供应链风险发生规律最直接的、可靠的信息来源,通过 Apriori 算法思想分析提取该 R/M 集成供应链上风险数据库中的强风险关联规则,分析出的风险发生规律可以提供较客观、可信度比较大的风险预测结果。通过参考这些风险预测的结果,管理者可以做出合理的决策方案以减小风险造成或将要造成的损失。

风险发生的不确定性导致任何风险的预测都不可能绝对准确,Apriori 算法的预测在一定程度上可以提高预测的准确性并作为辅助决策信息,依靠的是风险数据库中记录的内容,能够预测的是风险数据库中有记载的风险,对于没有出现在风险数据库中的新的风险事件是不可能预测出来的,缺乏灵活性及主观能动性,是很多预测算法的一个局限性,也包括 Apriori 算法,因此,在实际的风险预测中,对任何预测算法过分依赖都是不正确的。预测算法的计算结果可以作为辅助决策的信息考虑,预测算法的好坏与预测结果的可信度成正比,Apriori 算法在预测时采用的数据在很大程度上提高了预测结果的可信度,这是 Apriori 预测算法相较于其它预测算法的一种优势。

参考文献

[1] 顾巧论,陈秋双.再制造/制造系统集成物流网络及信息网络研究[J].计算机集成制造系统,2004,10(7):721-726,731

[2] Agrawal R, Imielinski T, Swami A. Mining Association Rules between Sets of Items in Large Databases [C] // SIGMOD. Washington, DC, USA, 1993:207-216

[3] Dunham M H. 数据挖掘教程[M].郭崇慧,田凤占,靳晓明,等译.北京:清华大学出版社,2005:144

[4] 张存禄,黄培清.数据挖掘在供应链风险控制中的应用研究[J].科学与科学技术管理,2004,1:12-14

[5] 赵春玲,宁红云. Apriori 算法的改进及其在物流信息挖掘中的应用[J].天津理工大学学报,2007,23(1):30-33

[6] 王瑜,刘连臣,吴澄.面向 web 关联规则的快速 Apriori 算法[J].微计算机信息,2008,24(5-3):109-111

[7] 朱添福,舒红平,朱冬梅.改进 Apriori 算法在制造业中的应用[J].计算机科学,2010,37(7A):142-143,147

[8] 魏茂林. Apriori 算法的改进及其在教育决策系统中的应用[D].长春:吉林大学,2010

[9] Kim W, Soh W. Applying Association Rule of the Data Mining Method for the Network Event Analysis [C] // Computation in Modern Science and Engineering, Proceedings of International Conference on Computational Methods in Science and Engineering. American; American Institute of Physics, 2007, CP963, 2A:567-570

[10] Prakash S, Parvathi R M S. An Enhanced Scaling Apriori for Association Rule Mining Efficiency [J]. European Journal of Scientific Research, 2010, 39(2):257-264

[11] Cavinato J L. Perspective Supply Chain Logistics Risks From The Back Room to The Board Room [J]. Physical Distribution & Logistics Management, 2009, 34(5):383-387

[12] Norrman A, Jansson U, Ericsson's Proactive Supply Chain Risk Management Approach After a Serious Sub-Supplier Accident [J]. Physical Distribution & Logistics Management, 2009, 34(5):434-456

[13] 丁伟东,刘凯,贺国先.供应链风险研究[J].中国安全科学学报,2003,13(4):64-66

[14] 陈长斌,缪立新.供应链风险类别、脆弱性因素及管理方法解析[J].商业经济,2009(5):98-101