

基于原语-模式的体系结构作战活动模型建模方法

谢文才¹ 沈伟业² 罗雪山¹

(国防科技大学信息系统工程重点实验室 长沙 410073)¹ (61062 部队 北京 100091)²

摘 要 为了理解、比较和集成由不同框架、不同工具及建模方法开发的体系结构,更好地支持以数据为中心的体系结构开发策略,提出了基于原语-模式的军事信息系统体系结构建模方法;依据 XML 的模型转换实现,重点研究了基于建模原语-模式的体系结构元模型与建模方法之间的语义映射规则;基于国际国防企业体系结构规范构建了作战活动模型(OV-5)的元模型,分别研究了 OV-5 的 IDEF0 建模和 UML 活动图建模的具体实现,构建了 IDEF0 及 UML 活动图与 OV-5 元模型数据元素严格的语义映射规则,规范了体系结构建模方法并保证了体系结构语义的精确性和一致性。

关键词 体系结构,建模原语-模式,作战活动模型(OV-5),可扩展标记语言(XML)

中图法分类号 TP303 **文献标识码** A

Primitives-patterns Based Architecture Modeling Method of Operational Activity Model

XIE Wen-cai¹ SHEN Wei-ye² LUO Xue-shan²

(Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China)¹

(The PLA 61062 Unit, Beijing 100091, China)²

Abstract In order to realize the understanding, comparison and integration of architecture developed under different architecture frameworks and tools with various modeling methods, and support the data-centric methodology of architecture development better, a new method named primitives-patterns based modeling method of architecture development was proposed. According to the transformation approach from architecture meta-model to modeling languages with the extensible markup language(XML) format, the primitives-patterns based mapping specification was designed. Based on the international defense enterprise architecture specification, the meta-model of operational activity model was built. By analyzing the modeling principles of IDEF0 and UML activity diagram, this method constructed the semantic mapping relationship between OV-5 meta-model and modeling languages, such as IDEF0 and UML activity diagram, ensuring the precision and coherence of architecture data and semantics.

Keywords Architecture, Modeling primitive-pattern, Operational activity model(OV-5), Extensible markup language(XML)

随着体系结构技术在军事信息系统建设中应用越来越广泛,体系结构方法论的发展也不断向前推进,各主要军事强国、同盟体(美、英、北约等)都先后发布了各自的体系结构框架,并伴随信息技术进步和需求的改变而不断演化。其主要体系结构框架的演化过程如图 1 所示^[1,2]。框架的每一次改进都会带来体系结构数据模型的变化。由于支持各国框架的体系结构数据模型之间的差异,导致了基于不同框架所开发的体系结构之间难以实现相互理解和交互,影响了体系结构的开发效率,阻碍了分析决策活动的有效执行。为此,由美、英、加、澳 4 国联合组成的技术工作组开发了国际国防企业体系结构规范(IDEAS),为构建多国体系结构领域建模的顶层本体来支持多国体系结构数据的交互,更好地实现多国联合作战系统体系结构数据的理解和沟通提供了一定的基础。然

而,不同的组织和团体依然根据各自的习惯选择不同的框架和工具来开发体系结构,而体系结构工具和建模方法并没有集成 IDEAS 的数据规范,以致不同体系结构框架或工具下开发的体系结构难以进行有效的交互^[1-3]。

此外,在同一框架方法论的指导下,不同的组织和团体可能会依据不同的建模方法、建模语言来开发体系结构^[4,5](见图 1)。如作战活动模型(OV-5)可以选择 IDEF0 或 UML 活动图建模,作战事件跟踪描述(OV-6c)产品可以选择 BPMN 或 UML 时序图建模等;并且不同工具所集成的建模方法之间没有遵循一致的数据规范,这同样会导致体系结构描述的不一致,使得对体系结构产品的理解、交互和集成变得困难。即便是使用同样的建模方法,如果所采用的建模风格和技术不同,也会带来类似的问题。

到稿日期:2011-06-01 返修日期:2011-11-02 本文受国家自然科学基金(71171197),国家部委项目资助。

谢文才(1984-),男,博士生,主要研究方向为军事信息系统体系结构, E-mail: xiewencai@sohu.com; 沈伟业(1982-),男,硕士生,主要研究方向为军事信息系统需求分析及验证; 罗雪山(1965-),男,博士,教授,主要研究方向为军事信息系统基础理论、系统建模、分析与仿真、体系结构技术。

针对以上问题,本文以作战活动模型(OV-5)为例,基于IDEAS的上层本体结构建立了OV-5的元模型,研究体系结构元模型与不同建模语言的模型转换方法。基于建模原语模式的映射规则重点研究OV-5元模型分别与IDEF0建模方法和UML活动图建模方法之间的语义映射关系,从而规范体系结构建模工具和建模方法的使用,提供不同建模方法所收集的体系结构数据的语义一致性基础,为实现不同框架、不同工具及建模方法下所开发体系结构之间的理解、交互和数据重用提供了一种解决方案。

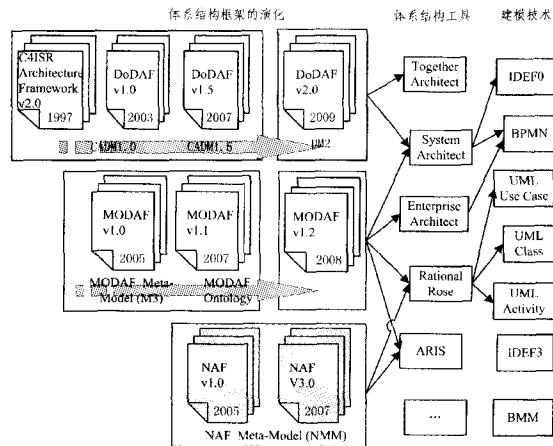


图1 体系结构框架、工具及建模方法之间的关系

1 模型转换

随着可扩展标记语言(XML)技术的广泛应用,XML已成为基于不同体系结构框架方法论及数据模型之间数据交换的基础。基于MDA的思想,本文依据MOF元建模规范,提出了基于XML的体系结构元模型到建模语言的模型转换方法。基于本方法,能够应用多种建模方法对体系结构建模,并能保证数据的一致性,实现体系结构数据的重用和交换。同时,可以基于不同的决策需要,将体系结构数据在多种建模方法所支持的环境中灵活地展现。基于XML的体系结构元模型到建模语言的模型转换方法如图2所示。

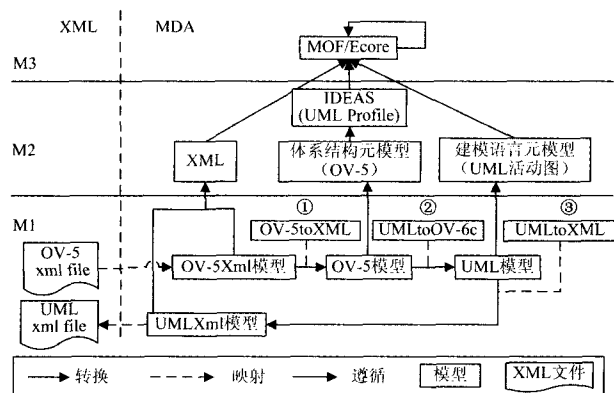


图2 基于XML的体系结构元模型到建模语言的转换方法

如图2所示,OV-5的XML文件经过转换、映射为UML活动图的XML文件是一个可逆的过程,也可通过UML活动图建模OV-5来生成OV-5的XML文件。如果构建OV-5的所有建模方法(如IDEF0、UML活动图等)都是基于以上方法实现的,并能提供元模型数据规范的导入、导出接口,那么体系结构数据就可以通过不同的建模工具和方法进行有效的组

织、收集、存储,并能在不同的体系结构工具和建模方法中展现而不会引起语义丢失和歧义。以下分别对图2所示方法中涉及到的相关转换做简单的分析说明。

转换规则1 OV-5 to XML Schema. 本文所建立的OV-5元模型是以IDEAS高层本体为规范的。依据UML建模规范,IDEAS工作组建立了UML的配置文件,来表达IDEAS的高层本体。基于IDEAS高层本体,为本体中的每个数据元素定义一个类别模版;再基于XMI^[6](XML metadata interchange)技术来创建XML Schema;最后依据IDEAS的XML Schema规范,可生成OV-5的XML Schema。这样,不同的建模方法和开发工具只要依据OV-5的XML Schema规范,提供体系结构数据的XML导入、导出接口,那么它们之间的体系结构数据是可以被发现和重用的。

转换规则2 UML活动图 to OV-5. 研究UML活动图到OV-5的映射关系,是实现OV-5的UML活动图建模方法的首要前提,是基于UML活动图来精确收集、存储及展现OV-5模型数据的基础,其有效保证了与其它OV-5建模方法之间体系结构数据的共享和交换。通过构建UML活动图到OV-5严格的语义映射关系,能够生成OV-5模型产品的XML文件。当面向不同的体系结构分析决策需求时,依据图2所示的体系结构建模方法,可以在其它建模语言及其建模环境中有效地展现OV-5模型数据(XML格式文件)。关于UML活动图到OV-5语义映射关系实现方法的研究将在下节重点展开。

转换规则3 UML to XML Schema. 研究将UML模型转换为XML格式文件,是实现OV-5在UML活动图建模方法下导入、导出相应的XML Schema文件的基础。当然,在UML活动图建模环境下要生成OV-5的XML文件,还需要依据规则2中提出的映射规则。如果其它建模语言和工具不是基于以上方法设计的,那么通过获取相应建模方法的XML文件,对图2所示转换规则进行逆向执行,同样可以得到规范体系结构数据,并能依据规则2在其它建模语言及工具中展现数据,实现体系结构数据的共享和重用。OMG颁布了适用于UML以及XML Schema的XMI技术,其提供了模型之间的转换标准,为程序员和用户提供了元数据信息交换的标准方法^[7]。因此,利用XMI标准进行UML和XML Schema转换更加合理,并且可以保证不同程序从同一模型产生的模式相同。

2 建模原语-模式的映射规则

支持体系结构开发的建模方法多种多样,如IDEF0、UML、BPMN等。这些建模方法本身都不是针对面向体系结构的开发而设计的,它们都包含了大量的建模原理和模式。要精确地掌握其所有,是一个非常复杂的过程,这就给体系结构产品的开发和数据收集带来了很大的困难。因此,在构建上节中的转换规则2时,不能简单地将建模语言的所有概念及模式关联到体系结构元模型中的概念元素。在构建体系结构产品时,可能只需要部分建模原理和模式就可以满足建模的要求。如果没有对相应建模方法进行有效的裁减,将会导致学习和理解上的困难,并且无法实现数据语义的一致性。

建模原语-模式(Modeling Primitives and Patterns)的映射方法是通过有效构建所采用建模语言的原语及模式与体系

结构元模型元素的对应关系,并对建模语言的模式进行最大化的裁剪定制,来实现所建模型的使用目的。建模原语-模式语义映射方法的本体描述如图3所示(以OV-5的UML活动图建模为例)。通过构建OV-5元模型到UML活动图的语义映射关系,基于图2所示的转换规则,能精确有效地收集OV-5产品的数据,并形成基于OV-5的XML Schema的XML文件,以支持体系结构数据的重用和交换,提供了体系结构数据语义一致性的基础。如果同样构建其它OV-5建模方法(IDEF0)与OV-5元模型的映射规则,那么分别采用IDEF0和UML活动图所构建的OV-5是可以集成和共享的。

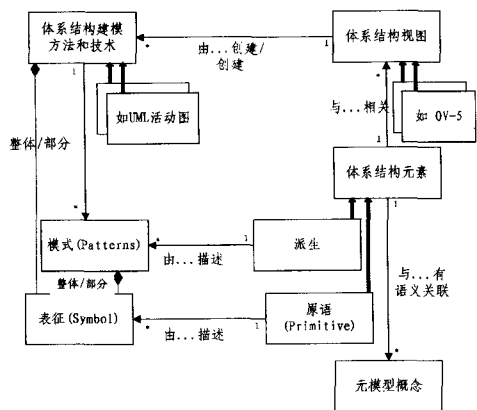


图3 建模原语-模式语义映射方法的本体描述

原语建立了体系结构核心概念与它们相关联的体系结构模型之间、体系结构模型与不同的建模方法技术之间的关系,以及建模方法技术与体系结构概念标准化描述之间的关系。体系结构建模方法、工具与体系结构元模型的关系如图4所示。原语构建了建模方法和体系结构元模型之间的桥梁,有利于精确收集体系结构数据,促进了对体系结构形成一致的理解。原语构建了不同建模方法之间的桥梁(基于规范的体系结构元模型),使得体系结构产品可应用不同的工具和建模方法进行建模,并且能依据图2所示的转换规则保证体系结构数据的一致性。由一种建模方法构建的体系结构产品和数据,同样可以在其它建模方法和环境下展现。这样就实现了不同体系结构工具、建模方法下所开发的体系结构产品之间能有效地进行集成和交互,提高了体系结构开发的效率,真正体现了以数据为中心的体系结构开发思想。基于此方法,以下将分别采用IDEF0和UML活动图对OV-5建模进行详细研究。

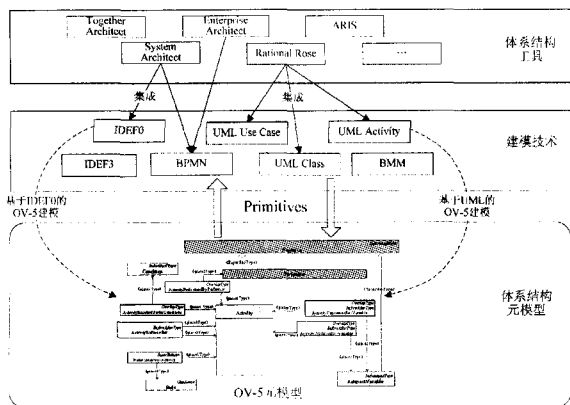


图4 体系结构建模方法、工具与元模型之间的关系

3 作战活动模型(OV-5)建模方法

3.1 OV-5元模型

为实现与不同框架数据模型之间体系结构数据的共享和交互,本文构建的OV-5元模型是基于IDEAS上层本体(Upper Ontology)进行建模的。IDEAS是一个形式化、高序、4D、有着良好扩展机制的本体,其提供了一个精确、基于严格数学基础的方法来构建数据模型。模型中的任何元素都经过严格分析,为构建领域的元模型提供了一个高层的标准,并为不同元模型(基于IDEAS上层本体构建的)之间的互操作性提供了重要支持。

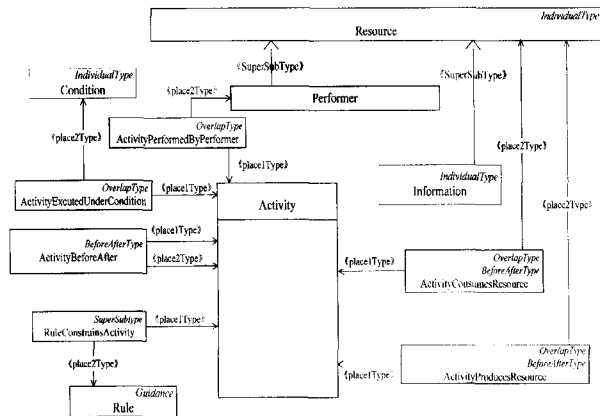


图5 OV-5元模型

表1 OV-5元模型中数据元素的语义说明

数据类型	定义	类别模版 (Stereotype)
概念类		
Activity	执行者为完成作战任务而履行的行为、动作	IndividualType
Performer	在系统支持的军事行动中具备相对独立执行任务的实体	IndividualType
Resource	作战资源,是系统、装备、执行主体及信息类型的统称	PowerType
Condition	作战活动在执行过程中所依附的条件	IndividualType
Rule	作战活动在执行过程中所遵循的规则	PowerType
Information	与作战相关的、能感知的一切事物状态的直接或间接表述	IndividualType
关联类		
activityProducesResource	作战活动的执行将产生新的资源	BeforeAfterType; OverlapType
activityConsumesResource	作战活动的执行需要消耗资源	BeforeAfterType; OverlapType
activityPerformsByPerformer	作战活动由执行者来执行	OverlapType
activityPerformableUnderCondition	作战活动的执行依赖于特定的条件	OverlapType
ruleConstrainsActivity	作战活动的执行受到相关规则的约束	OverlapType

OV-5主要描述为完成使命和作战任务而进行的作战活动,它描述能力、作战活动(或任务)、作战活动之间的信息流关系,其基本设计原则是^[5]:

(1)作战活动描述是对完成使命或任务所需的作战活动以及作战活动之间关系的描述。其中,作战活动之间的关系

包括作战活动的层次或分解关系,以及为完成作战任务,作战活动之间的输入/输出关系。

(2)作战活动描述的粒度由需求的应用决定,不同应用的需求描述,其作战活动描述的层次或粒度有所不同。

(3)作战活动必须由执行者来完成。

(4)作战活动的输入/输出关系反映作战活动之间的信息交换关系。依据以上作战活动模型的设计内容和原则构建的OV-5元模型如图5所示。

关于OV-5元模型中概念类及关联类的语义说明如表1所列。表1中“类别模版”列指定了IDEAS上层本体对应的类别模版,便于数据检索、追踪以及OV-5的XML Schema的生成(图2所示转换规则1)。

3.2 OV-5的IDEF0建模

IDEF0建模方法^[8]结构化地描述了所研究系统的活动和处理进程,通过一系列的图形符号来表示模型,图形元素主要有盒子和箭头。IDEF0中的基本模型是活动,在图中用一个方框表示。一个活动代表系统所执行的功能,它对一个输入集进行转化。这个输入可分为3类:输入(input)、控制(control)和机制(mechanism)。输入是一个功能所要转化或是要消耗掉的信息或资源(resource);控制是转化的条件、环境或约束(condition, rule),一个活动至少要有一个控制箭头;机制是活动执行功能所需要的资源(performer/resource)。

当前多数体系结构工具都集成了IDEF0建模方法来构建OV-5,然而并没有依据统一元模型的数据规范来收集、存储体系结构数据,导致不同工具下开发的体系结构难以理解、比较和集成。基于本文提出的映射规则构建OV-5元模型与IDEF0原语-模式之间的语义映射关系,如表2所列,通过规范的数据收集实现不同框架和工具下所开发体系结构之间的共享和交互。以上对于IDEF0建模原语与OV-5元模型的映射关系进行说明,表2只对IDEF0建模的模式进行了分析研究。

表2 基于原语-模式 IDEF0 建模方法与 OV-5 元模型语义映射关系

IDEF0 建模原语-模式	语义	对应 OV-5 元模型的概念元素 模式(Patterns)
	输入是一个功能所要转化或是要消耗掉的信息或资源	Information/Resource, Activity, activityConsumesResource
	输出是一个功能将输入转化为信息或资源	Information/Resource, Activity, activityProducesResource
	控制是活动执行需要领带的条件、环境以及受到规则约束	Condition; Rule; Activity; RuleConstrainsActivity; ActivityPerformableUnderCondition
	机制是活动执行功能所需要的资源,表示为活动的执行者	Performer; Activity; activityPerformedByPerformer
	活动的先后执行模式	Activity; RuleConstrainsActivity; Rule; ActivityBeforeAfter

依据以上所构建的映射规则,结合图2提出的模型转换方法,设计基于IDEF0构建OV-5模型数据的XML导入、导出接口的实现算法,见算法1。

算法1 基于IDEF0建模需求数据的XML导入导出接口设计

```
//遍历 IDEF0 中所有的建模对象,包括活动、输入输出、控制和机制
CArray objects[n] = {obj01, obj02, .....objn, };
for(int i = 0; i < n; i++)
{
    //判断建模对象的类型
    switch(object[i]. type)
    {
        case activity: //对象类型为活动
            str_ID = input. ID; str_name = input. name; //获取对象 ID, 名称
            str_stereotype = activity; //指定对象对应的元模型中的类别模版
            pXMLDocument->createElement (bstrTagName, &pElementActivityArray);
            //向 XML Schema 中添加 activity 节点的属性
            InsertActivityIntoXML (pXMLDocument, pElementActivityArray, pNode);
        case input: //对象类型为输入
            str_ID = input. ID; str_name = input. name; //获取对象 ID, 名称
            object = input. activity; //获取输入指向的活动对象
            Str_stereotype = resource; //指定对象对应元模型中的类别模版
            //分别创建 Resource、activity、activityConsumesResource 节点标签
            ...
            //分别添加 Resource、activity、activityConsumesResource 节点的属性
            ...
        case output: //对象类型为输出并依据映射规则关联的类型添加 XML 标签
            ...
        case control: //对象类型为控制并依据映射规则关联的类型添加 XML 标签
            ...
        case mechanism: //对象类型为机制并依据映射规则关联的类型添加 XML 标签
            ...
    }
}
```

基于以上设计的映射规则及体系结构数据导入、导出接口的实现算法,运用 IDEF0 建模方法来构建 OV-5,能依据 OV-5 的元模型数据规范并通过 XML 数据导入、导出接口来组织、收集和存储体系结构数据,为数据交换和集成提供有效支持。

3.3 OV-5 的 UML 活动图建模

活动图是 UML 用于对系统的动态行为建模的一种常用工具^[9]。一个简单的活动图示例如图6所示,图中基本包含了活动图建模的建模原语及模式。在UML中,活动表示成圆角矩形,活动的图标包含对活动的描述(如活动名),两个活动之间用带箭头的直线连接,表示活动间的信息流。活动图中还包括分支与合并、分叉与汇合等建模模式,每种模式的执行都关联约束规则及依赖条件。关于活动图各建模元素及建模模式所表达的语义,本文不做详细介绍,相关内容可参考相关UML建模方法的文献。

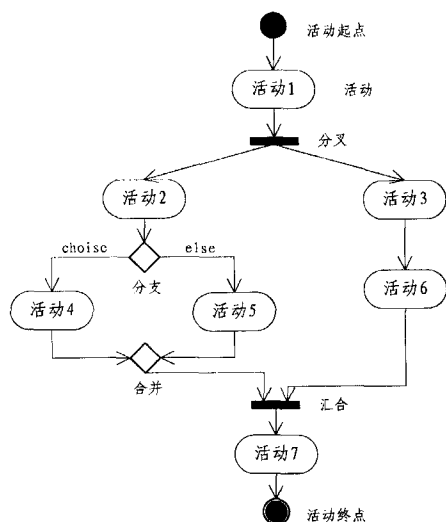


图6 UML活动图示例

依据本文所提出的原语-模式的映射规则,构建 UML 活动图建模原语-模式与 OV-5 元模型概念元素之间的语义映射关系,如表 3 所列。

表3 基于原语-模式 UML 活动图与 OV-5 元模型映射规则

UML 活动图建模原语-模式	语义	对应 OV-5 元模型的概念元素
模式 (Patterns)		
	顺序模式: 表示活动的顺序执行	Information/Resource; Activity; activityConsumesResource; activityProducesResource
	分叉模式: 用来描述活动的并发执行,每个分叉有一个输入,两个或多个输出转换	Information/Resource; Activity; activityConsumesResource; activityProducesResource; activityBeforeAfter
	汇合模式: 用来描述多个并发控制流同步发生,当所有活动都达到汇合点后,才能继续执行以下的活动	Activity; activityConsumesResource; activityProducesResource; activityBeforeAfter
	分支模式: 描述一个有条件的活动执行过程,即用一个布尔表达式的真假来判定需要执行的活动	Condition; Rule; Activity; RuleConstrainsActivity; ActivityPerformableUnderCondition; activityBeforeAfter
	合并模式: 表示从对应的分支开始的条件行为的结束	Activity; RuleConstrainsActivity; Rule; ActivityBeforeAfter
	泳道-活动模式: 描述活动由执行者来执行	Activity; Performer; activityPerformedByPerformer

表 3 中裁剪了 UML 活动图的建模模式,6 种模式可以满足 OV-5 的建模需求(见 3.1 节),并扩展了 OV-5 描述的内

容,为实现体系结构模型的多重使用提供了基础,从而规范了基于 UML 活动图的 OV-5 建模,提供了精确一致的体系结构描述。限于篇幅,本文不再对基于 UML 活动图建模的 OV-5 模型数据 XML 导入、导出接口进行设计,具体实现可参照算法 1。

结束语 当前国内外主流的体系结构建模工具有 System Architecture(SA)、Enterprise Architecture(EA)、Rational Rose 以及国内相关研究单位(包括国防科技大学、中电 28 所、解放军理工大学等)开发的体系结构建模工具,它们所支持的框架方法论及采用的建模方法和原理都不尽相同。为更好地支持体系结构的理解、比较和集成,关于实现不同框架、不同工具及建模方法下所开发体系结构之间共享和交互的需求显得越来越迫切。本文基于 XML 的模型转换方法,分析了方法中涉及到的 3 种转换规则,重点研究了建模语言原语-模式与元模型数据元素严格的语义映射关系,有效地维护了底层数据的规范性、一致性,降解了不同体系结构工具、建模方法由于选用框架及设计原理上的不一致而引起的语义冲突。并且,本方法不强制用特定的建模方法进行体系结构建模,如 OV-5 建模可以选择 IDEF0 建模方法或 UML 活动图建模方法。基于本方法进行体系结构建模,能有效支持体系结构数据的重用,提高体系结构开发的效率,为支持多种框架、多种建模方法的体系结构建模方法提供了良好的理论基础。

参考文献

- [1] DoD Architecture Framework Working Group, DoD Architecture Framework Version 2.0[R]. U S: Department of Defense, 2009
- [2] UK Ministry of Defense. UK Ministry of Defense Architectural Framework(MODAF) v1.2.004[R]. UK Ministry of Defense, 2010
- [3] Bailey I, Partridge C. Working with Extensional Ontology for Defence Applications[C]//Ontology in Intelligence Conference. 2009
- [4] 罗雪山,罗爱民,张耀鸿. 军事信息系统体系结构技术[M]. 北京:国防工业出版社,2010
- [5] 罗爱民. 基于框架的 C4ISR 体系结构语法、语义设计与分析方法研究[D]. 长沙:国防科学技术大学,2006
- [6] OMG. OMG XML metadata interchange specification version 2.1 [EB/OL]. <http://www.omg.org/cgi-bin/doc?formal/2005-09-01.pdf>, 2005
- [7] 李清,李伟明,徐大丰. 基于元模型的企业模型表达[J]. 清华大学学报,2008,48(7):1209-1212
- [8] 王杏林,曹晓东. 概念建模[M]. 北京:国防工业出版社,2007
- [9] 严悍,刘冬梅,等. UML 软件建模:概念、规范与方法[M]. 北京:国防工业出版社,2009

(上接第 132 页)

- [11] 李珏峰. 基于量化方法的大型遗留系统迭代再造研究[D]. 杭州:浙江大学,2008
- [12] 许鹏. SOA 架构下 Web Services 实现的企业遗留系统重用研究[D]. 合肥:合肥工业大学,2008
- [13] 王少锋. 基于多 agent 的程序理解方法研究[J]. 计算机科学,2002(5):131-133

- [14] 汪小林,邓浩,王海波,等. Fortran 地理模型的拆分与服务化封装[J]. 计算机科学与探索,2011(3):221-228
- [15] 钱佳. 基于 MAS 技术遗留系统包装器体系结构风格的研究[D]. 太原:太原理工大学,2006
- [16] 詹剑锋,程虎. 基于 Mobile Agent 技术的遗留系统再工程方法[J]. 软件学报,2002(12):2343-2348