

基于 DM2 的体系结构可执行模型构建方法*

张晓雪, 罗爱民, 黄力, 罗雪山

(国防科技大学 信息系统工程重点实验室, 湖南 长沙 410073)

摘要:基于元模型的体系结构设计思想和基于可执行模型的评估方法是体系结构领域研究的两个热点,但当前的研究没有将两者结合起来。文章基于 DoDAF2.0 提出的元模型数据(Meta-Model Data, DM2),将元模型的思想与基于可执行模型的评估方法相结合,分析了基于 DM2 的逻辑数据与可执行模型各构建要素的对应关系,构建了基于 DM2 的逻辑数据模型直接转可执行模型的过程框架,重点研究了如何直接从体系结构底层数据转可执行模型的方法,从而为进行基于元模型的体系结构可执行评估提供模型基础,也为进一步实现体系结构自动化验证评估提供技术支持。实例验证了文章提出的方法。

关键词:体系结构;DM2;可执行模型;对象 Petri 网

中图分类号:N94 **文献标志码:**A **文章编号:**1001-2486(2013)02-0027-07

Method of creating architecture executable model based on DM2

ZHANG Xiaoxue, LUO Aimin, HUANG Li, LUO Xueshan

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

Abstract: Architecture design and development based on Meta-Model and architecture evaluation method based on executable model are two hot topics in architecture domain study, but there is no study combining these two topics. Based on the conception of Meta-Model Data (DM2) which puts forward DoDAF 2.0, the research combined the idea of DM2 and the evaluation method based on the executable model, analyzed the relationship between the logical data based on DM2 and the executable model, designed the progress of supporting data using DM2 Foundations change directly into executable model, especially focused on the method of changing the architecture data into executing model. The study provides the basis for architecture executable evaluation, and helps to take further study on architecture automatic validation and evaluation. Finally, the case study demonstrates the method proposed in this paper.

Key words: architecture; meta-model data (DM2); executable model; object Petri net

体系结构验证评估的主要任务是检查体系结构设计正确性,确定体系结构描述的体系结构方案是否满足系统的功能需求和非功能需求及其满足程度^[1]。现有的体系结构评估方法主要有两类,一类是基于想定的评估方法,如基于场景的体系结构分析方法(Scenario-based Architecture Analysis Method, SAAM)、体系结构权衡分析方法(Architecture Tradeoff Analysis Method, ATAM)等,该类方法主要采用定性的评估方法,主观性较强,难以评价体系结构的逻辑或者行为方面的特征属性,而逻辑和行为方面的特征属性正是分析评价体系结构方案中的作战活动能否按照预期完成的重要因素。第二类是基于可执行模型的评估方法,这类研究方法最早由乔治梅森大学的 Levis A H 等人提出^[2],通过将描述的体系结构产品转为着色 Petri 网(Color Petri Net, CPN)仿真模型来评

估体系结构的逻辑和行为方面的属性。由于其能够客观定量分析体系结构设计,基于可执行模型的体系结构评估方法是体系结构评估领域研究的一个热点。现有的可执行评估方法主要是根据产品的表现形式,来研究不同类型的产品如何转各类可动态执行的仿真模型。如 CPN^[3-4]、对象 Petri 网(Object Petri Net, OPN)^[5-6]、离散事件仿真系统(Discrete Event System Specification, DEVS)^[7-8]、ExtendSim 模型^[9]等。

美军国防部体系结构框架的 1.5 版本(Department of Defense Architecture Framework version 1.5, DoDAF 1.5)采用基于核心体系结构数据模型(Core Architecture Data Model, CADM)的思想,体系结构设计的结果是视图产品,在转换为可执行模型时往往需要规则模型(如 OV6a、SV10a)、逻辑模型(如 OV5、SV10b)和数据模型

* 收稿日期:2012-06-26

基金项目:国家自然科学基金资助项目(71171197);国家部委资助项目

作者简介:张晓雪(1986—),女,河北邯郸人,博士研究生,E-mail:snowing1124@126.com;

罗雪山(通信作者),男,教授,博士,博士生导师,E-mail:luoxueshan@gmail.com

(如 OV3、SV7)三类模型产品的信息。DoDAF2.0 版本中提出了一致性实现策略,使用元模型数据(Metal-Model Data, DM2)的思想替代了核心体系结构数据模型(CADM),体系结构描述中的数据都是通过 DM2 中的概念、关系、属性来定义的,设计成果不是体系结构产品,而主要是体系结构数据。体系结构数据能依照 DM2 中的物理交换规范(Physical Exchange Specification, PES)进行交换,从而支持不同体系结构之间信息和数据共享与集成。基于 DM2 的体系结构设计方案是否正确有效,需要对其进行验证评估,但是以往针对具体产品的体系结构验证评估方法不再适用。本文研究的前提是体系结构设计和描述数据按照 DM2 的 PES 规范存储,探讨了基于 DM2 的体系结构数据转可执行模型的方法。

1 基于 DM2 的体系结构可执行评估过程

1.1 DM2 结构介绍

美国防部在 2009 年提出 DoDAF2.0 中使用了 DM2 取代了核心体系结构数据模型(CADM)。DM2 的建模基础是体系结构国际数据交换规范(International Data Exchange Architecture Specifications, IDEAS)上层本体中的一个子集。这个子集具有严格的数学逻辑基础,在此基础上 DM2 定义了体系结构的数据元素,能更好地实现精确的体系结构描述,规范体系结构建模方法,支持体系结构信息的交换与重用。DM2 包括概念数据模型(Conceptual Data Model, CDM)、逻辑数据模型(Logical Data Model, LDM)和物理交换规范(PES)三个层次:CDM 给出了 IDEAS 的高级模式以及基于 IDEAS 规范的 DM2 通用模式(DM2 Common Patterns),表示了体系结构描述时的高层的数据结构,它是 DoDAF 建立元模型的底层逻辑关系的参考基础。DM2 的 LDM 基于 CDM,提供了 12 类簇群,给出了与不同的数据组相关联的本体。PES 规范了体系结构中数据的存储模式,体系结构数据能依照 PES 规范进行存储、共享和交换等,PES 遵循在 CDM 和 LDM 定义的底层数据逻辑关系规范。DoDAF2.0 在第三卷使用了 xsd 文件表示了体系结构设计及描述的 PES^[10]。在美国国防部发布的相关文件中,基于 DM2 的体系结构数据存储规范的 xsd 文档包括 IDEASData 和 IDEASView 两个部分。其中,IDEASData 存储的是体系结构公共数据部分,数据项包括了 DM2 中的各类本体。IDEASView 采用对 IDEASData 部分引用的方式给出了各个产品

的数据。

1.2 基于 DM2 的可执行评估过程

以往的可执行模型研究往往受限于产品或者描述语言的具体表现形式,难以支持基于 DoDAF 2.0 的体系结构建模环境下的体系结构动态评估分析。本文所研究的遵循 DoDAF2.0 的 PES 规范的体系结构数据转可执行模型的方法,具有两个特点:其一是它能够不受限于体系结构产品的表现形式,可以从底层数据出发直接转所需要的可执行模型,能够保证与体系结构设计数据的一致性。其二是强调可执行性和定量评估。一般的体系结构评估往往是从定性的角度,评估结果具有主观性,而通过对可执行模型进行执行,研究其执行表现出来的数据,能够从定量的角度评价作战活动、系统、服务等性能或者效能表现,通过评估结果的分析,并与定性的评估方法结合,能够得出可靠性较高的评估结论。

基于 DM2 可执行模型的体系结构验证评估方法的本质是根据 DoDAF2.0 体系结构设计元模型的物理交换规范,建立转换体系结构可执行模型的规则,通过建立可执行模型并运行,全面反映体系结构的动态行为、特性以及效用。在此基础上,分析体系结构的逻辑关系、行为关系、接口关系的合理性,验证系统功能是否满足需求,作战活动、系统状态的变化是否正确等。通过收集、分析执行数据,评估系统的性能和效能。具体的评估过程如图 1,它是一个反复的过程。

第一步,确定体系结构评估的内容和准则。根据利益相关者的关注点和可执行模型的特点,分析体系结构的设计内容以及内容之间的相互关系,并根据体系结构验证评估的目的和要求来确定体系结构验证评估的内容,同时分析评估内容的合理性,使各级评估人员对评估内容达成一致理解。然后确定对应评估内容的评估准则,如作战活动和作战任务的需求。

第二步,确定采用可执行模型的类型。由于不同类型的可执行模型需要的体系结构数据类型有所不同,在配置可执行模型时就应该针对不同类型的模型做不同处理。

第三步,根据确定的验证评估内容和采用的可执行模型类型,研究将基于 DM2 的体系结构数据转换为可执行模型。这一步骤是核心。不同的评估内容,对应选择的体系结构设计的数据项内容也不同,根据所选择的评估内容从体系结构设计数据中提取出关联的数据项。表 1 中列出了对应不同评估内容所需要的视图产

品或者数据项。在第 2 部分详细地介绍可执行模型构建过程。

第四步,可执行模型的运行。根据生成的可执行模型和验证评估的内容,在配置可执行模型及增加相关运行参数之后,运行所得到的可执行模型,获取可执行模型的执行数据。

第五步,分析可执行模型运行后得出的数据结果进行合适的处理,综合得出评估结果。分析体系结构设计结果与用户需求之间的差异,若满足需求,则完成体系结构的验证评估,否则修改在评估过程中发现的体系结构设计短板,重新进行验证评估。

表 1 对应于体系结构评估内容的体系结构产品和主要数据项

Tab.1 The main architecture products and data items corresponding to the architecture evaluation contents

评估内容	转可执行模型需要的视图产品(DoDAF1.5)	转可执行模型需要的 XML 文件中的关键数据项(DoDAF2.0)
作战活动的逻辑关系、行为关系是否正确、规则合理性	OV5、OV6a、OV6b/OV6c	Activity、Performer、activityResourceOverlap、BeforeAfterType、WholePartType、SuperSubType、activityProduceResource、activityConsumeResource 等
系统节点性能,系统节点繁忙程度、系统结构、系统节点时间性能	OV5、SV1、SV2、SV7、SV6、SV10a、SV10b、SV10c	Activity、Performer、System、BeforeAfterType、WholePartType、SuperSubType、activityPerformedByPerformer、ruleConstraintActivity、activityPerformedUnderCondition、activityProduceResource、activityConsumeResource 等
功能正确性	SV4、OV5、SV1	Activity、Performer、System、BeforeAfterType、WholePartType、SuperSubType、ruleConstraintActivity 等
服务节点性能,服务节点繁忙程度、服务结构、服务节点时间性能、服务规则正确性	OV5、SV1、SV2、SV7、SV6、SV10a、SV10b、SV10c	Activity、Performer、Service、ActivityResourceOverlap、BeforeAfterType、WholePartType、SuperSubType、activityPerformedByPerformer、activityPerformedUnderCondition、activityProduceResource、activityConsumeResource、ruleConstraintActivity 等

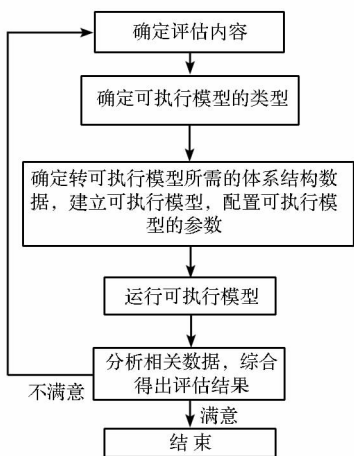


图 1 基于 DM2 体系结构可执行评估过程

Fig.1 The process of architecture evaluation based on executable model using DM2

2 基于对象 Petri 网的可执行模型构建

本文使用 OPN 构建可执行模型,旨在通过对体系结构数据转换对象 Petri 网模型的转换说明基于 DM2 的体系结构数据转可执行模型的构建方法。DM2 中的通用模式(Common Patterns)

和 LDM 描述了基于 IDEAS 本体中一个具有严格数学逻辑关系的子集,包括 Individual、IndividualType、Tuple 以及在此基础上建立的本体。关于 DM2 中的通用模式和 CDM、LDM 的介绍详见文献[10]。

分析转换可执行模型元素需要的 Tuple 类型包括 SuperSubType、BeforeAfterType、WholePartType、OverlapType、TemporalWholePartType 等。其表示的含义和对应转 OPN 元素的关系见表 2 和表 3。仅有这些 Tuple 类型数据并不能实现转换,需要结合各类实体类型 IndividualType 类型的实体(如 Activity、Performer、Resource、System、Service 等)描述和 Type 类型的实体(如 Measure、Rule、Condition、Information 等)描述。

在构建可执行模型结构和配置可执行模型的参数时,应综合考虑各个 Tuple 类型及它们之间的关系。根据 DM2 中的与可执行模型生成相关联的数据项,使用 DM2 的 PES 规范构建可执行对象 Petri 网模型过程及各步骤的元模型如图 2,具体过程如下:

表 2 DM2 的 CDM 和 LDM 中主要的 Tuple 含义和对应可执行模型元素

Tab. 2 The meaning of Tuples in DM2's CDM and LDM and corresponding executable model items

Tuple 类型	在 DM2 中表示的含义	对应转 OPN 的元素
SuperSubtype	表示一类集合是另一类集合的子集。可表示活动之间的子集关系, 类似于服务之间、系统节点之间或者能力之间。	确定模型框架结构和确定模型对象类
BeforeAfterType	表示概念类之间, 如活动之间、系统之间或者服务之间执行的先后顺序。是构建可执行模型的重要部分。	确定各个模型的内部结构
WholePartType	概念类之间部分与整体的关系。通常表示数据之间、活动之间、系统之间、服务之间以及度量组成元素之间的关系。	转模型框架; 转移的规则和参数等
TemporalWholePartType	在某个阶段时具有的部分与整体关系。通常表示执行者的状态或者阶段, 能力的或者项目的不同成长阶段, 以及活动的顺序。	构建具体模型
OverlapType	表示个体在时空范围的交叠关系或者状态。如活动与系统之间的执行关系、活动在何种状态下执行的关系等。	配置可执行模型的参数, 如动作函数、延时函数等
TypeInstance	个体和个体类型对应关系如 Measure 和 MeasureType 的对应关系, Information 和 InformationType 的对应关系	配合其他数据项使用来配置动作函数、延时函数

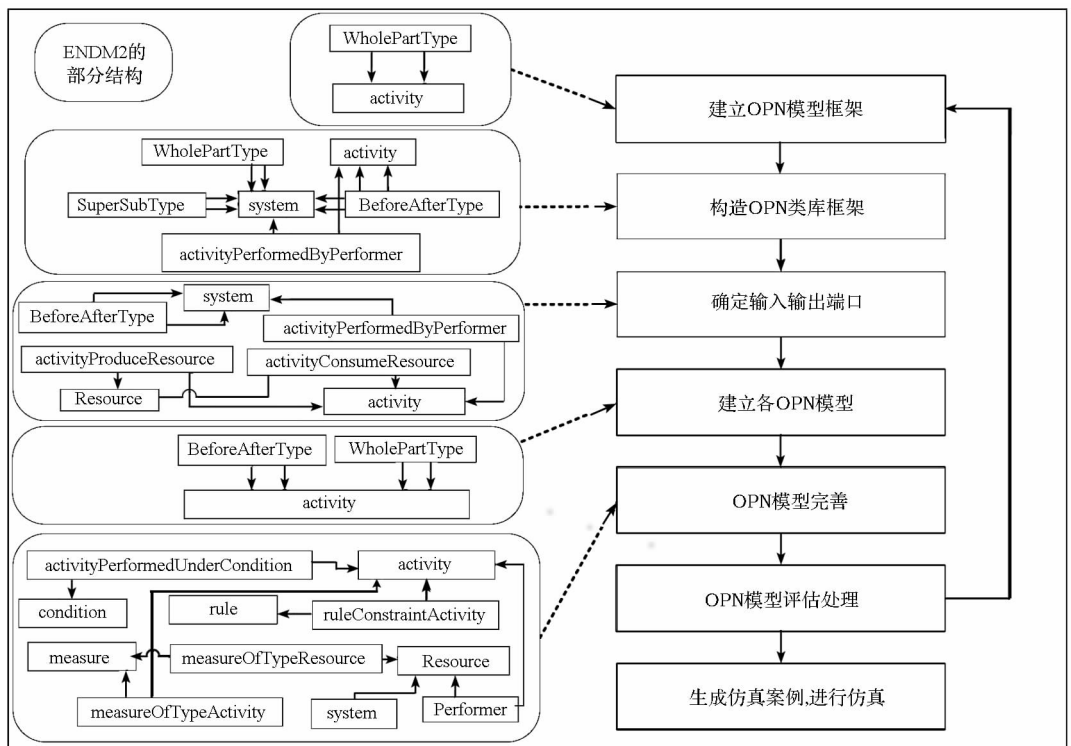


图 2 基于 DM2 的对象 Petri 网可执行模型过程及元模型

Fig. 2 The creation process and Meta-models of OPN executable models based on DM2

Step 1 建立系统模型框架: 分析 Activity 数据项及对应的 WholePartType, 将只有子活动而没有父活动的 Activity 集合转为整个模型框架;

Step 2 构造 OPN 类库框架: 根据 activityPerformedByPerformer 对 Activity 和 System

数据项中的描述, 对相关的 WholePartType 和 SuperSubType 中的 System 中同一个或同一类对象进行合并、抽象, 提取出相应的 OPN 模型类, 根据 WholePartType 和 SuperSubType 抽象出 OPN 模型类之间的关系, 构造 OPN 类库框架;

表 3 DM2 的 LDM 中的主要关系类型及对应的可执行模型元素

Tab.3 Main technical relationship in DM2's LDM and corresponding executable model items

DM2 的 LDM 中主要的关系类型	属于的 Tuple 类型	DM2 中的含义	对应的 OPN 元素
activityProducesResource	BeforeAfterType	活动执行产生资源的关系	配置可执行模型的动作函数
activityConsumesResource	BeforeAfterType	活动执行消耗资源的关系	配置可执行模型的作战函数
rulePartofMeasureType	WholePartType	表示规则的度量类型	配置可执行模型参数
activityPerformedByPerformer	OverlapType	表示执行者执行活动的关系	确定对象类的框架结构
activityPerformedUnderCondition	OverlapType	表示活动在何种条件下执行	配置可执行模型的参数
measureOfTypeActivity	SuperSubType	活动的度量	配置可执行模型的参数
measureOfTypeEffect	SuperSubType	效果的度量	配置可执行模型的参数
measureOfTypeResource	SuperSubType	资源的度量,如系统、服务等	配置可执行模型的参数
ruleConstraintActivity	SuperSubType	规则约束活动的关系	配置可执行模型的参数

Step 3 确定输入输出端口:根据 BeforeAfterType, 确定不同类之间的交互关系, 根据 activityProducesResource 和 activityConsumesResource 明确交互信息, 根据 Information 和 Resource 数据项明确交互信息的特征, 确定 OPN 模型框架的输入输出端口;

Step 4 建立各 OPN 框架模型: 根据 BeforeAfterType 中有关对 Activity 运行过程的描述, 建立各 OPN 框架模型, 如果 Activity 存在子活动(根据 WholePartType 判断), 将其转换为对象类, 如果不存在子活动, 将其转换为开关。并将模型中输入输出端口与相应的转移连接起来;

Step 5 OPN 模型完善: 由 ruleConstraintActivity、measureOfTypeResource、activityPerformedUnderCondition、activityProducesResource 和 activityConsumesResource 对系统性能参数和信息交互的描述, 结合 BeforeAfterType 和 Rule 对 Activity 运行和转移规则的描述, 添加 OPN 框架模型中的脚本函数, 包括动作函数、延时函数、事件处理函数, 实现各模型的相关功能;

Step 6 OPN 模型评估处理: 根据建立的评估指标体系, 在模型中添加相关的指标数据采集脚本, 完成 OPN 模型;

Step 7 生成仿真案例, 进行仿真: 添加初始化函数、实例化后函数, 实例化各顶层节点模型, 生成总体仿真案例。根据评估需求, 设计仿真实验, 配置模型进行仿真。处理采集到的仿真数据, 与评估指标对应起来, 按照事先确定的评估模型实施对系统体系结构的评估分析。对于脚本不易实现的复杂功能, 可以用 VC 编制动态链接库算法, 并在 OPN 模型中加以调用。

3 案例研究

设计案例背景为: 当一个游艇(yacht)在海上遇难时, 向监控中心(monitoring unit)发出求救信号, 指挥中心(controlling center)在收到监控中心的消息后派一架直升机(halo)和搜救船(lifeboat)及舰艇(navy ship)对遇难船只实行目标搜索定位和营救活动, 如图 3。仿照网站 <http://www.w3.org/2001/XMLSchema-instance> 上发布的一个海上

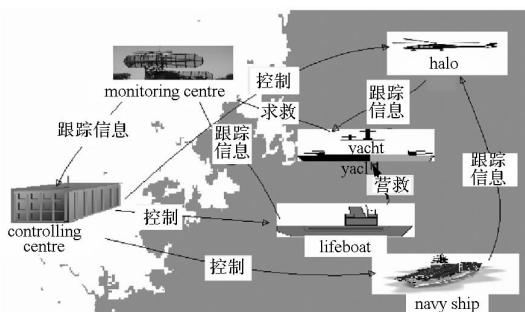


图 3 海上救援高级作战概念图

Fig.3 The high level operational concept of maritime rescue 救援的例子底层数据的 XML 文档(该文件采用了 DoDAF2.0 的 PES 规范), 对该作战任务进行基于 DM2 的体系结构描述后生成的部分体系结构数据如图 4 所示。下面按照第 1 和第 2 节所介绍的步骤, 假定评估内容为作战活动行为关系和逻辑关系, 转换可执行模型的过程为:

(1) 找到确定模型框架类的信息。根据图 4 表示 Activity 和 WholePartType 数据项, 确定对象 Petri 网框架类为 Search 和 Rescue, 建立对象 Petri 网模型的分层结构为两层。

(2) 找到确定各个对象类组成部分的特性描述信息和组成部分之间关系的 BeforeAfterType 描述的 Activity 之间的关系信息, 建立底层 Petri 网结构。

(3) 根据 Activity 运行机制信息,如活动消耗资源 activityConsumeResource、活动产生资源 activityProduceResource、规则约束活动 ruleConstarintActivity 等。实例中的运行机制信息为 activityResourcesOverlap,替代了 activityProduce-Resources 和 activityConsumesResources 表示活动之间的信息传递情况,由此设置底层 Petri 网的初始标识和转移事件函数,根据 Resource、Rule、Information 的信息确定不同层次对象 Petri 网中各个转移的执行延时。

(4) 确定转移事件函数中 Token 的结构。根据资源 Resource、rule、Measure 与 Activity 相关数据项,分析出 Token 的资源种类,将资源随时间的变化增加到转移函数中。

(5) 初始化对象 Petri 网模型并根据仿真需要添加脚本,对脚本不易实现的功能可以通过编写

动态链接库实现。图 5 表示了对应图 4 中海上救援实例的体系结构设计数据而生成的活动可执行模型结构。其中,图 4 中 BeforeAfterType 数据项对应了图 5 中活动之间的顺序关系,WholePartType 对应了 OPN 对象的层次关系,activityResourceOverlap 对应了 OPN 各子对象变迁之间传递的令牌信息。在生成可执行模型,再根据 Rule、Measure 等确定 OPN 模型的运行规则后,可通过仿真运行得出关于活动之间逻辑或者行为的属性。另外,可以结合 System (或者 Service)数据项及与 System(或者 Service)相关联的数据项建立对应的可执行模型,也可来评价系统执行活动时的逻辑行为关系等动态属性。在根据可执行模型获得体系结构的动态评估结果后,结合其他的体系结构评估方法,如基于场景或者想定的评估方法,综合得出评估结果。

```

<!-- IdeasData XML: TagsBoundToNamingScheme="DM2Names" ontologyVersion="2.01" ontology="DM2" -->
<NamingScheme ideas:FoundationCategory="NamingScheme" id="ns1">
<ideas:Name namingScheme="n0" id="n0" exemplarText="DM2Names"/>
</NamingScheme>
<?START?>

<?Activity?>
Activity信息
<Activity ideas:FoundationCategory="IndividualType" id="a1"><ideas:Name exemplarText="Search" namingScheme="ns1" id="n1"/></Activity>
<Activity ideas:FoundationCategory="IndividualType" id="a2"><ideas:Name exemplarText="Rescue" namingScheme="ns1" id="n9"/></Activity>
<Activity ideas:FoundationCategory="IndividualType" id="a3"><ideas:Name exemplarText="Find Victim" namingScheme="ns1" id="n10"/></Activity>
<Activity ideas:FoundationCategory="IndividualType" id="a33"><ideas:Name exemplarText="Track Victim" namingScheme="ns1" id="n161"/></Activity>
<Activity ideas:FoundationCategory="IndividualType" id="a41"><ideas:Name exemplarText="Monitor for distress signal" namingScheme="ns1" id="n202"/></Activity>
<Activity ideas:FoundationCategory="IndividualType" id="a42"><ideas:Name exemplarText="Determine if assistance is required" namingScheme="ns1" id="n203"/></Activity>
<Activity ideas:FoundationCategory="IndividualType" id="a6"><ideas:Name exemplarText="Monitor Health" namingScheme="ns1" id="n13"/></Activity>
<Activity ideas:FoundationCategory="IndividualType" id="a13"><ideas:Name exemplarText="Be Distressed" namingScheme="ns1" id="n5"/></Activity>
<Activity ideas:FoundationCategory="IndividualType" id="a9"><ideas:Name exemplarText="Transit to SAR Operation" namingScheme="ns1" id="n16"/></Activity>

<?WholePartType?>
整体-部分关系
<WholePartType ideas:FoundationCategory="WholePartType" id="wp11" place1Type="a1" place2Type="a3"/>
<WholePartType ideas:FoundationCategory="WholePartType" id="wp12" place1Type="a1" place2Type="a33"/>
<WholePartType ideas:FoundationCategory="WholePartType" id="wp13" place1Type="a1" place2Type="a41"/>
<WholePartType ideas:FoundationCategory="WholePartType" id="wp14" place1Type="a1" place2Type="a13"/>
<WholePartType ideas:FoundationCategory="WholePartType" id="wp15" place1Type="a2" place2Type="a42"/>
<WholePartType ideas:FoundationCategory="WholePartType" id="wp16" place1Type="a2" place2Type="a6"/>
<WholePartType ideas:FoundationCategory="WholePartType" id="wp17" place1Type="a2" place2Type="a9"/>

<?ActivityResourceOverlap?>
Activity 使用资源情况,这里的资源指的是信息 (DomainInformation)
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro35" place1Type="a13" place3Type="a41" place2Type="di1"/>
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro36" place1Type="a41" place3Type="a33" place2Type="di2"/>
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro37" place1Type="a33" place3Type="a3" place2Type="di23"/>
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro38" place1Type="a3" place3Type="a42" place2Type="di24"/>
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro39" place1Type="a42" place3Type="a6" place2Type="di25"/>
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro40" place1Type="a6" place3Type="a9" place2Type="di27"/>
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro41" place1Type="a9" place3Type="a41" place2Type="di26"/>
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro42" place1Type="a3" place3Type="a41" place2Type="di22"/>
<activityResourceOverlap ideas:FoundationCategory="TripleType" id="aro43" place1Type="a42" place3Type="a41" place2Type="di21"/>

<?BeforeAfterType?>
Activity 的执行顺序
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat23" place1Type="a13" place2Type="a41"/>
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat24" place1Type="a41" place2Type="a33"/>
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat25" place1Type="a33" place2Type="a3"/>
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat26" place1Type="a3" place2Type="a42"/>
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat27" place1Type="a42" place2Type="a6"/>
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat28" place1Type="a6" place2Type="a9"/>
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat29" place1Type="a9" place2Type="a41"/>
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat30" place1Type="a3" place2Type="a41"/>
<BeforeAfterType ideas:FoundationCategory="CoupleType" id="bat31" place1Type="a42" place2Type="a41"/>

<?DomainInformation?>
资源信息
<DomainInformation ideas:FoundationCategory="IndividualType" id="di1"><ideas:Name exemplarText="distressSignal (DS)" namingScheme="ns1" id="n17"/></DomainInforma
<DomainInformation ideas:FoundationCategory="IndividualType" id="di2"><ideas:Name exemplarText="warningOrder (WO)" namingScheme="ns1" id="n22"/></DomainInform
<DomainInformation ideas:FoundationCategory="IndividualType" id="di21"><ideas:Name exemplarText="No assistance required" namingScheme="ns1" id="n204"/></DomainInfor
<DomainInformation ideas:FoundationCategory="IndividualType" id="di22"><ideas:Name exemplarText="Search cancelled" namingScheme="ns1" id="n205"/></DomainInforma
<DomainInformation ideas:FoundationCategory="IndividualType" id="di23"><ideas:Name exemplarText="Estimated victim location" namingScheme="ns1" id="n206"/></Domain
<DomainInformation ideas:FoundationCategory="IndividualType" id="di24"><ideas:Name exemplarText="Victim found" namingScheme="ns1" id="n207"/></DomainInforma
<DomainInformation ideas:FoundationCategory="IndividualType" id="di25"><ideas:Name exemplarText="Assistance required" namingScheme="ns1" id="n208"/></DomainInfor
<DomainInformation ideas:FoundationCategory="IndividualType" id="di26"><ideas:Name exemplarText="Victim secure" namingScheme="ns1" id="n209"/></DomainInforma
<DomainInformation ideas:FoundationCategory="IndividualType" id="di27"><ideas:Name exemplarText="Victim stable" namingScheme="ns1" id="n210"/></DomainInforma

```

图 4 海上救援例子的部分 xml 数据文件

Fig. 4 The segment of xml document of maritime rescue

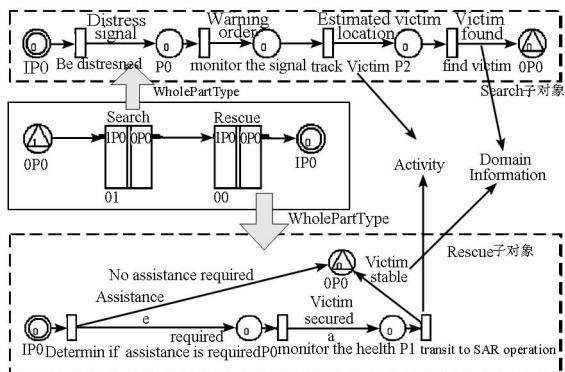


图 5 海上救援活动的可执行模型

Fig. 5 The executable model of maritime rescue

4 小结

基于元模型的体系结构设计和建模是体系结构研究的一大趋势,而基于元模型的可执行模型是评价体系结构设计质量的重要手段之一。文章探索了基于元模型研究如何转换可执行模型:分析了基于 DM2 的逻辑数据与可执行模型各构建要素的对应关系,研究了如何直接从体系结构底层数据转可执行模型的方法。在本文研究的基础上,可以进行体系结构相关质量属性的分析,另外,考虑在体系结构建模时候对于转可执行模型的数据集进行单独存放,以支持可执行模型的自动化构建和评估结果分析,进而实现体系结构的自动化评估,或设计与评估的一体化。

参考文献 (References)

[1] 罗爱民. 基于可执行模型的体系结构验证评估方法[J]. 计算机科学, 2010, 37(8): 294 - 297.
 LUO Aimin. Research on information system architecture description method based on multi-views [J]. Computer Science, 2010, 37(8):294 - 297. (in Chinese)

[2] Wagenhals L W, Levis A H. Service oriented architectures, the DoD architecture framework 1.5, and executable architectures[J]. System Engineering, 2009,4(12):56 - 72.
 [3] Wagenhals L W, Haider S, Levis A H. Synthesizing executable models of object oriented architectures[J]. System Engineering, 2003, 6(4):266 - 300.
 [4] Wang R Z, Dagli C H. Executable system architecting using sysML in conjunction with colored petri nets[C]//Proceedings of Conference on Systems Engineering Research, Los Angeles: IEEE, 2008.
 [5] 修胜龙,罗雪山,罗爱民. C4ISR 体系结构描述的逻辑和行为验证[J]. 系统工程与电子技术, 2005, 27(2):275 - 279.
 XIU Shenglong, LUO Xueshan, LUO Aimin, et al. Logic and behavior validation of C4ISR architecture description [J]. Systems Engineering and Electronics, 2005,27(2): 275 - 279. (in Chinese)
 [6] 姜军,罗雪山,罗爱民,等. 可执行体系结构研究[J]. 国防科技大学学报, 2008,30(3):76 - 80.
 JIANG Jun,LUO Xueshan,LUO Aimin. A study of executable architecture [J]. Journal of National University of Defense Technology, 2008,30(3):76 - 80. (in Chinese)
 [7] Zeigler B P, Mittal S. Enhancing DoDAF with DEVS-Based system lifecycle development process [C]//IEEE International Conference on Systems, Man and Cybernetics, Hawaii: IEEE,2005.
 [8] Mittal S. Extending DoDAF to allow Integrated DEVS-based modeling and simulation[J]. Journal of Defense Modeling and Simulation, 2006,3(2):95 - 123.
 [9] Huynh T V, Osmundson J S. A systems engineering methodology for analyzing systems of systems using the systems modeling language (SysML) [C]//2nd Annual System of Systems Engineering Conference, 2006.
 [10] DoD architecture framework working group. DoD architecture framework version 2. 0 volume I-III: architect's guide-architectural data and models [R]. U. S. : Department of Defense, 2009.