

文章编号: 1001-2486(2008)03-0076-05

## 可执行体系结构研究\*

姜军, 罗雪山, 罗爱民, 杨春辉

(国防科技大学 C<sup>4</sup>ISR 重点实验室, 湖南长沙 410073)

**摘要:** 体系结构是系统设计与实施的蓝图, 它能否正确、合理地指导实践在很大程度上依赖于体系结构的验证与评估。可执行体系结构研究为体系结构验证与评估开辟了新的途径。为了使各体系结构产品在整体上形成一个执行体, 提出了可执行体系结构执行规则描述这一个概念。执行规则描述体系结构产品的执行、调用与数据流关系, 为体系结构的执行提供了特定的背景信息。它通过标准的动态可执行模型接口分别调用该模型的执行引擎, 从而达到体系结构在整体上执行的目的。体系结构的正确执行为体系结构验证提供依据, 同时其运行结果为体系结构的评估提供依据。

**关键词:** 体系结构; 可执行体系结构; 执行规则; 体系结构验证与评估

**中图分类号:** N94; TP303 **文献标识码:** A

## A Study of Executable Architecture

JIANG Jun, LUO Xue-shan, LUO Ai-min, YANG Chun-hui

(Key Laboratory of C<sup>4</sup>ISR Technology, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract:** Architecture is a blueprint of system design and implementation, and whether it can guide practice correctly largely depends on the architecture verification and evaluation. Executable architecture provides a new way to validate and evaluate the architecture. In order to make all architecture products into a whole executive entity, this paper puts forward a new concept—execution rules description. Execution rules describe the execution, call and data flow relationship between architecture products, and thus provide special background information for the architecture execution. Through standard dynamic executive model interfaces, execution rules can call the models' executable engines, and then attain the aim to run the architecture on the whole. That architecture can run successfully, providing justification for architecture verification, and its executive result also provides justification for architecture evaluation.

**Key words:** architecture; executable architecture; execution rules; architecture verification and evaluation

体系结构是系统各组成部分的结构、它们之间的关系以及制约其设计与随时间演进的原则和指南<sup>[1]</sup>。它是系统使用人员需求的体现, 是系统设计和实现人员对系统进行设计和实现的依据, 是维护人员熟悉系统的切入点。同时, 它也代表了系统最早的一组决策, 这种早期的约束对系统开发、设计、运行、维护等阶段有很强的指导作用。当前, 描述体系结构的产品都是采用静态的表现形式, 如图形、文本、表格等。这种静态的表现形式使体系结构的验证与评估很难进行, 主要表现在缺乏有效的方式来对系统在特定作战环境中的行为、性能和效能做出有效的动态分析<sup>[2-3]</sup>。这样, 就无法获知体系结构的优劣, 也就无法获知体系结构能否正确、合理地指导实践。

可执行体系结构主要解决体系结构的执行问题。由于体系结构产品是从特定角度来反映系统, 某些产品反映了系统的静态结构关系, 而另一些产品反映了系统的可执行的动态特性, 因此要让体系结构执行, 必须让体系结构产品中可执行的动态模型能执行起来, 同时要提供这些产品之间的逻辑调用关系和数据流关系, 让体系结构产品在逻辑上形成一个执行整体, 从而达到对体系结构执行的目的, 为体系结构验证与评估提供可信依据。

\* 收稿日期: 2007-12-19

基金项目: 国家部委基金资助项目

作者简介: 姜军(1979-), 男, 博士生。

## 1 可执行体系结构概念

体系结构描述以可视化的方式描述了体系结构的组成, 组成部件之间的关系、规则和约束, 如何让体系结构按照体系结构描述所描述的组成、关系、规则和约束进行执行, 这是可执行体系结构所要解决的问题。可执行体系结构是对体系结构的扩展, 它不仅描述了体系结构的组成, 组成部件之间的关系、规则和约束, 而且还描述了体系结构动态模型产品的可执行性, 动态模型产品之间的执行逻辑。可执行体系结构可以定义如下: 可执行体系结构= 静态模型产品+ 动态可执行模型产品+ 执行规则描述。

(1) 体系结构。可执行体系结构也是体系结构, 因此它也符合体系结构的定义, 也包括部件, 部件之间的关系, 以及制约设计与随时间演进的原则与指南。

(2) 具有可执行的动态模型产品。可执行的动态模型产品是可执行体系结构执行的主要逻辑单元。如果一个体系结构没有可执行的动态模型产品, 而只有静态模型产品, 那么它就失去了可执行的最基本条件, 体系结构的执行也就无从谈起。


(3) 体系结构产品执行规则描述。体系结构从整体上描述一个系统, 并且可能具有多个可执行的动态模型产品。这些产品从不同角度反映了系统的执行特性, 彼此之间必定存在一定的交互, 同时也可能从静态模型产品中获取系统信息。执行规则描述可执行体系结构中动态模型产品之间的执行、调用关系以及与静态模型产品之间的数据流关系。体系结构的执行将从执行规则描述开始, 并根据执行规则描述, 对相应的体系结构产品进行调用、执行。

体系结构产品执行规则描述将详细说明产品之间执行、调用关系的接口, 即从这个产品需要传递哪些参数到另外一个产品, 并期望从另一个产品获取哪些信息。这里需要对动态可执行模型产品与静态模型产品作说明, 如在 DoDAF 中, 当前都是以静态模型产品来进行描述的, 事实上像 SV1、OV5 等这样的产品, 描述的是系统、活动之间的信息交换。系统、活动是在获取一定信息之后进行处理, 并将结果分发出去, 这几个产品看起来是描述静态关系, 事实上蕴含着动态可执行特性。它们交换的信息, 是由 OV7 逻辑数据模型中的实体来表示的, 进一步通过 SV11 和具体的数据库关联。有些体系结构产品描述的信息是从不同角度来描述系统的, 它们可能存在信息重叠, 如 DoDAF 中的 OV6b 和 OV6c, 虽然是两个不同的体系结构产品, 一个描述状态的变化, 一个描述时序, 但这两个产品描述的信息是可以切换的, 是描述内容的两个不同表现。在可执行体系结构中可以这样来理解这些产品: 体系结构产品执行规则描述将所有这些产品串起来, 以完成体系结构使命为前提, 根据体系结构执行规则进行执行, 如果某些产品没有得到执行或者某些产品之间的执行发生矛盾, 则说明这些地方设计得不是很合理, 需要重新进行设计与开发。可执行体系结构所描述的体系结构产品以及它们之间的逻辑执行关系, 为体系结构验证与评估工具提供了详细内容。一个优秀的体系结构验证与评估工具应当是一个框架式的、具有执行能力的工具, 这样, 当把体系结构产品以及逻辑执行规则装载进工具时, 工具就可以根据体系结构产品中的主要过程进行执行, 从而达到验证与评估体系结构的目的。

和静态体系结构相比, 可执行体系结构将带来巨大的优势, 主要体现在: 通过体系结构内部自身的一致性来验证体系结构产品; 仿真、动态观察、验证并精炼作战概念; 检查、评估作战计划; 评估系统间的折衷; 评估体系结构; 支持费效分析; 支持定量的需求决策。

## 2 执行规则描述

执行规则描述是可执行体系结构的重要组成部分, 它描述体系结构中动态可执行模型产品之间的调用、执行关系以及动态模型产品与静态模型产品之间的信息交换关系, 是体系结构开始执行的地方, 其主要描述元素有:

(1) 动态可执行模型产品图元 (Dynamic) 

代表动态可执行模型产品。动态可执行模型产品图元可以和动态可执行模型产品图元、静态模型产品图元连接, 表示动态模型产品对另一动态模型产品的调用或者根据静态模型产品对相关信息进行

存取。动态模型产品中的元素有可能会调用它的子图,这种关系表示为调用,调用关系需要清楚表示是由哪个元素来调用相应子图,如图1所示。Order图和Ensure Stock图都是用IDEF3<sup>[4]</sup>描述的体系结构产品,且Ensure Stock图为Order图中UOB(Unit of Behavior,行为单元)Ensure Stock的子图,Ensure Stock详细描述了UOB Ensure Stock是如何被处理的。图1中的右边描述了动态可执行模型产品Order的图元及其对UOB Ensure Stock的子图的调用关系。调用将把UOB Ensure Stock此时的状态信息传递给子图Ensure Stock,并触发它的运行,其运行结果将被返回给UOB Ensure Stock,以继续执行父图Order。

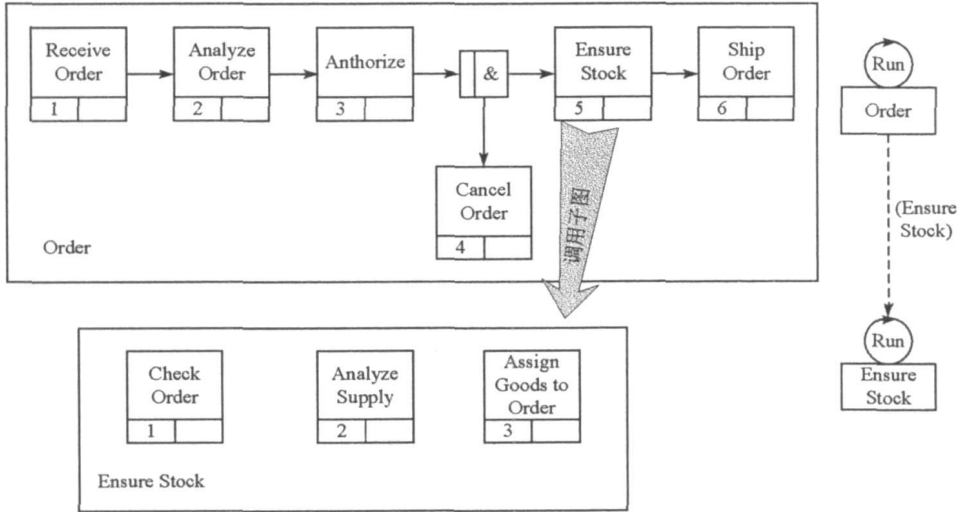




图1 动态可执行产品图元到动态可执行产品图元的调用

Fig. 1 The call from dynamic executable product model to dynamic executable product model

动态可执行模型产品必须详细说明模型执行的必需条件,包括它需要从外部获取哪些信息,这些信息是同步到达还是异步到达等。

(2) 静态模型产品图元(Static) 

静态模型产品图元只能和动态可执行模型产品图元建立连接,表示动态可执行模型产品从静态模型产品所描述的实体中获取系统信息或存储信息,如图2所示。由于静态模型产品信息描述系统的结构以及实体关系,动态可执行模型产品获取的不是这些实体,而是由这些实体描述的具体对象。如图2中的Order实体关系图中的实体Order,动态可执行模型产品Order就是获取该实体描述的具体订单。这种执行不仅是可行的,而且是有意义的,因为Order实体关系图会和物理数据模型关联,而物理数据模型会直接和具体的数据库关联,那么动态可执行模型产品Order事实上就是间接和具体的数据库进行相关操作。

(3) 调用关系图元(Call) 

表示动态可执行模型产品图元之间的调用关系。调用指动态可执行模型产品中某个对象调用其子图,并将其运行结果返回给父图。调用关系说明该节点的详细执行过程是由该子动态可执行模型产品图元来完成的。由于动态可执行模型产品详细定义了触发模型执行的外部条件,即需要从外部获取哪些信息,这些信息是如何达到的,模型执行后会产生哪些结果信息,这样调用关系将详细说明调用者是如何将自身的状态信息传递给被调用者,并说明调用的结果是什么,这个结果将由调用者如何使用。在图1中调用者UOB Ensure Stock将把Stock及其状态信息传递给Ensure Stock子图,并触发其运行。Ensure Stock子图将按照检查订单、分析供货、配置货物给订单的顺序执行,并将最终结果——配置了货物的订单返回给调用者UOB Ensure Stock。

(4) 连接关系图元(Link) 

表示动态可执行模型图元必须在箭头所指的动态可执行模型图元之前执行,强调的是数据流关系

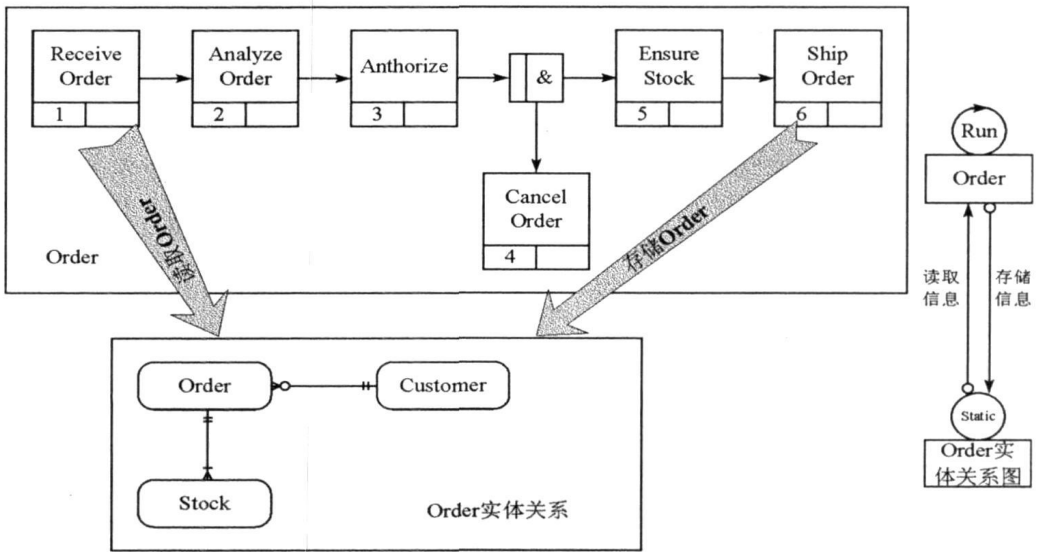


图 2 动态可执行模型产品图元对静态模型产品图元的数据操作

Fig. 2 Operations between dynamic executable product model and static product model

和执行的先后。连接关系有数据传递时, 必须要清楚说明前一个动态可执行模型产品的输出是如何和后一个动态可执行模型产品的输入对应的, 如图 3 所示。

(5) 提取/ 存储图元(Extract) ○ →

提取/ 存储图元表示动态可执行模型产品图元从静态模型产品图元中提取/ 存储信息。静态模型产品可能会描述多个实体或对象, 而动态可执行模型产品有可能只是部分地获取该静态模型产品描述的实体, 这时提取/ 存储图元必须要详细说明动态可执行模型产品是对静态模型产品中哪些实体对象进行操作的, 以使提取/ 存储操作有明确的对象。

(6) 连接点图元(Junction) ◆

当连接点图元为分叉时, 表示动态可执行模型产品图元的输出可以经过若干路径流向不同的终端, 该终端可以是动态可执行模型产品图元, 也可以是静态模型产品图元, 路径的选择可以由不同的筛选条件确定; 当连接点图元为合并时, 表示若干动态可执行模型产品的输出合并为一个动态可执行模型产品或静态模型产品的输入, 合并可以满足不同的条件, 如图 4 所示。图 4 中描述的分叉与合并, 动态可执行模型产品的输出到动态可执行模型产品的输入可以由图 3 描述, 连接点图元仅仅描述输出到输入的条件, 这些条件包括同步、异步、与、或、异或等。

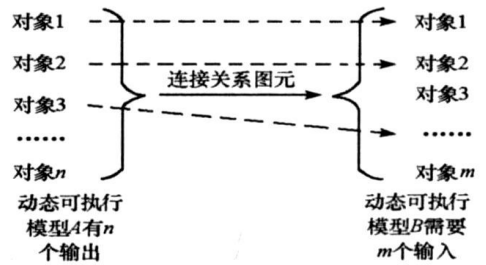


图 3 连接关系图元描述的动态可执行模型产品输出与输入的对应

Fig. 3 Linking Relationship depicting dynamic executable product model's output and input correspondence

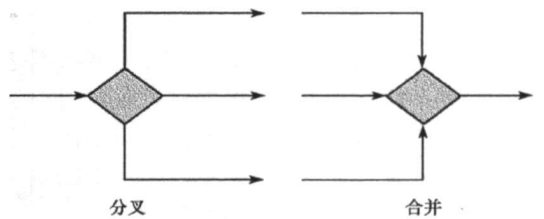


图 4 分叉和合并连接点

Fig. 4 Divergence and mergence junction

### 3 体系结构执行过程

体系结构的执行将按照执行规则描述中的动态可执行模型产品以及静态模型产品之间的关系进行, 如图 5 所示。

假设体系结构由五个体系结构产品模型组成, 分别是: (1) Order 实体关系图, 描述系统所涉及的相

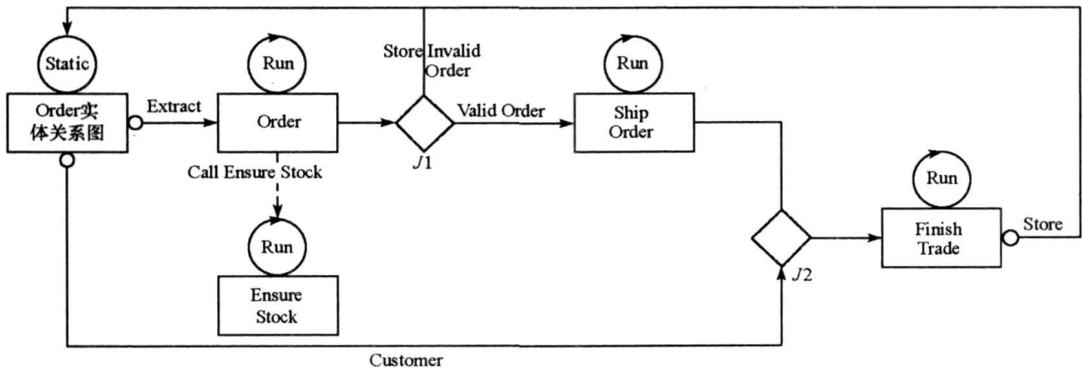


图5 体系结构的执行

Fig. 5 Architecture Execution

关实体及其属性; (2) Order 动态可执行模型, 描述系统是如何分析 Order 的, 同时它还调用了元素 Ensure Stock 的子图 Ensure Stock; (3) Ensure Stock 动态可执行模型, 描述系统如何确保 Stock; (4) Ship Order 动态可执行模型, 描述系统如何发送 Order; (5) Finish Trade 动态可执行模型, 描述系统如何结束交易。

执行启动时, 将先启动第一个动态可执行模型 Order, Order 从实体关系图中读取需要处理的 Order 信息, 然后根据模型中所描述的过程来分析 Order, 在分析 Order 的过程中调用 Ensure Stock 子图, 执行确保 Stock 的过程, 子图 Ensure Stock 执行完毕后, 将结果返回给 Order, 然后到达连接点 J1。连接点 J1 根据预先设置的逻辑条件, 将非法的 Order 存储到实体数据库中, 将合法的 Order 发送到下一个动态可执行模型产品 Ship Order。Ship Order 按照特定的配送方案将 Order 发送到各个目的地, 最后在连接点 J2 和客户信息合并, 最终到达 Finish Trade 动态可执行模型产品, 完成最后的 Trade 相关信息, 并存储到实体关系图中。

可执行体系结构的执行必须要兼容各种动态可执行模型的执行引擎, 这样执行规则描述才具有更广泛意义, 从而达到体系结构在整体上的可执行性, 为体系结构的验证与评估提供依据。

## 4 结论

构造可执行体系结构的执行规则是一个需要详加定义与设计的过程, 主要表现在:

(1) 定义动态可执行模型产品标准调用接口: 动态可执行模型产品的执行将触发其他的动态可执行模型产品的执行, 因此需要将当前的执行信息传递给被触发的动态可执行模型产品, 定义基于不同执行引擎的动态可执行模型产品的标准调用接口是一个复杂而庞大的任务, 这将是后续工作的主要内容。

(2) 动态可执行模型产品执行引擎: 当前各大主流建模方法都有相关的执行引擎供该模型执行, 如 BPMN、IDEF3、有色 Petri 网模型等。这些引擎执行所需的环境和条件都不太一样, 必须要提供一个统一描述各执行引擎启动执行的方法, 这样不同动态可执行模型之间的调用才能成为现实。

(3) 可执行体系结构执行流程的设计: 体系结构的执行将按照一定的想定或场景来进行设计, 让体系结构各模型产品实例化, 具有实际意义, 这样才能体现可执行体系结构具有仿真、动态观察作战概念、检察、评估作战计划的重要意义。

## 参考文献:

- [1] DOD Architecture Framework Working Group. DoD Architecture Framework Version 1.0 [R]. Volume I, Volume II, Volume III. 2003.
- [2] Pawlowski T. Executable Architecture Methodology for Analysis [R]. Army-contract MOIE, the MITRE Corporation, 2004.
- [3] Pawlowski T, Hoffman K. Multiagency Executable Architecture Case Study [R]. MITRE Technology Program, 2005.
- [4] Mayer R J, Ph. D. Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report [M]. Human Resources Directorate Logistics Research Division, September 1995.