

文章编号: 1001-2486(2006)05-00133-04

C⁴ISR 体系结构产品设计研究*

罗爱民, 黄力, 罗雪山

(国防科技大学 信息系统与管理学院, 湖南 长沙 410073)

摘要: 体系结构产品是 C⁴ISR 体系结构框架中定义的体系结构描述形式, 因此体系结构产品设计是体系结构设计的关键环节。基于 C⁴ISR 体系结构框架, 对作战节点连接关系、作战活动模型、系统接口描述等核心体系结构产品进行规范化描述, 分析了这些核心体系结构产品的数据相关性, 提出了体系结构产品设计的检验方法。

关键词: 体系结构框架; 体系结构产品; 检验

中图分类号: TP302.1; E072 **文献标识码:** A

Design on Architecture Products of C⁴ISR

LUO Ai-min, HUANG Li, LUO Xue-shan

(College of Information System and Management, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: C⁴ISR architecture framework defines the products which are used to describe system architecture. It follows that the design of products is an important process of building architecture. Based on C⁴ISR architecture framework, the core products which include Operational Node Connectivity Description, Activity Model, System Interface Description and the like are described normatively. Furthermore, the relationships of products are analyzed, and the methods to validate the design of product are presented.

Key words: architecture framework; architecture product; validate

体系结构设计是复杂大系统设计中不可缺少的内容。为此, 美军先后颁布了 C⁴ISR 体系结构框架^[1]和 DOD 体系结构框架 1.0^[2]以指导体系结构设计。框架提出了利用作战视图、系统视图和技术标准视图描述体系结构, 并定义了体系结构产品。框架对产品的定义主要采用文字描述, 内容和形式都不规范, 可操作性差, 在具体使用中容易引起歧义。此外, 体系结构框架采用多视图的体系结构描述方法, 通过多视图的数据共同构成对体系结构的整体描述, 因此, 多视图、多产品数据的一致性也是体系结构设计面临的关键问题。针对体系结构产品设计, 有学者研究采用结构化方法^[3]和面向对象方法^[4]设计产品。按照结构化方法设计, 主要利用 IDEF 系列建模语言设计部分产品。面向对象方法主要采用 UML 语言, 虽然 UML 语言具有形式化的特征, 但是也不支持全部产品设计。本文针对以数据为中心体系结构, 提出独立于设计方法的产品描述方法, 规范了体系结构产品含义, 并能用于检查和保证体系结构数据的一致性。由于篇幅有限, 这里对其中的核心产品进行规范描述。

1 体系结构产品规范化描述

1.1 作战节点连接关系描述

作战节点连接关系(简称 OV-2)主要描述完成任务或活动的作战节点以及它们之间的关系, 其中主要包含作战节点实体。作战节点(ON)是完成一定作战任务或活动的作战元素的集合。

设 $OV2_ONs$ 是作战节点的集合。作战节点包括以下属性:

$ON: = ON_Name, ON_Activities, Father_ON, ON_Organization[, other_performance]$

其中 $ON_Activities$ 表示节点完成作战活动的集合, $Father_ON$ 表示作战节点的父节点, $ON_$

* 收稿日期: 2006-04-03

基金项目: 国家部委资助项目

作者简介: 罗爱民(1970—), 女, 副教授, 在职博士生。

Organization 表示节点所属的组织。

在 OV-2 中,需求线(NL)表示节点间的信息交换关系,为支持活动完成,节点间需要什么信息,而不是信息传输怎样实现。设 NL 是需求线, $NL \subset ON_i \times ON_j$, $ON_i, ON_j \in ONs, i \neq j$, 需求线的属性:

$$NL := NL_Name, From_ON, To_ON, IERs[, other_performance]$$

其中, *From_ON*, *To_ON* 表示需求线的起始和终止节点, $From_ON, To_ON \in ONs$, *IERs* 表示需求线完成的信息交换集。

1.2 作战信息交换矩阵

作战信息交换矩阵(简称 OV-3)主要描述为完成作战节点承担的作战活动,作战节点所需的信息交换。其中关键内容即信息交换关系(IER)。信息交换关系的属性:

$$IER := IER_Name, NL, From_ON, Send_Activities, To_ON, Receive_Activities \\ Exchange_Performance[, other_performance]$$

其中 *From_ON*, *To_ON* 表示信息交换相关的源节点和汇节点。*Send_Activities*, *Receive_Activities* 表示信息交换中信息产生和使用的作战活动集。*NL* 表示信息交换属于的需求线, *Exchange_Performance* 描述信息交换的性能。

1.3 作战活动描述

作战活动模型主要描述完成使命和作战任务中进行的作战活动,以及作战活动之间信息流关系。作战活动描述中定义主要实体为作战活动。

作战活动属性:

$$Activity := Activity_Name, Father_OA, Perform_ONs[, other_performance]$$

其中 *Perform_ONs* 表示完成该作战活动的作战节点集。

作战活动描述中包含的关系包括作战活动的组成关系 $\delta(Activity)$ 和作战活动之间的信息交换关系 IER。 $\sigma: OA \rightarrow 2^{OA}$ 表示作战活动之间的层次分解关系,即作战活动之间的父子(孙)关系。

作战活动之间的信息交换关系的属性:

$$IER := IER_Name, Send_Activity, Receive_Activity[, other_performance]$$

其中 *Send_Activity*, *Receive_Activity* 是发送信息和接收信息的作战活动。

1.4 系统接口描述

系统接口描述(简称 SV-1)用来描述对作战节点表示的组织和人员实施支持的系统节点和驻留在这些节点上的系统。SV-1 主要包括系统节点、系统等实体。

系统节点实体的属性:

$$System_Node := SN_Name, SN_Systems[, other_performance]$$

SN_Systems 表示系统节点内包含的系统集合。系统实体的属性:

$$System := Sys_Name, Sys_Functions, Sys_Father[, other_performance]$$

其中 *Sys_Functions* 表示系统提供的系统功能集, *Sys_Father* 是系统所属的节点或系统。

SV-1 中的实体关系包括实体之间的组成(结构)关系和实体之间的接口关系。实体之间的组成关系是系统节点包含系统,系统可以包含子系统,不可再分的系统又称为组件。

实体之间的接口关系包括系统节点与系统节点,系统与系统之间的接口关系(*Interface*),即 $Interface \subset Entity1 \times Entity2$, *Entity* 可以是系统节点或系统。*Interface* 的属性:

$$Interface := In_Name, From_Entity, To_Entity, DERs[, other_performance]$$

其中, *From_Entity*, *To_Entity*, *DERs* 分别表示开始实体、结束实体和接口完成的数据交换集。

1.5 系统功能描述

系统功能描述(简称 SV-4)主要描述系统功能的层次性、系统功能以及它们之间的系统数据流。系统功能可以自顶向下分解。系统功能描述中的实体为系统功能。

系统功能的属性:

$$\text{System_Function} := \text{Function_Name}, \text{Entities}[, \text{other_performance}]$$

其中, *Entities* 表示完成系统功能的实体(系统、组件)的集合。

系统功能描述中定义的关系包括功能分解关系 $\delta(\text{Function})$ 和系统功能之间的数据交换关系。

数据交换关系(DER)的属性:

$$\text{DER} := \text{DER_Name}, \text{From_Funcs}, \text{To_Funcs}, \text{Exchange_Pr o}[, \text{other_performance}]$$

其中 *From_Funcs*, *To_Funcs* 分别表示产生交换数据的功能和接收的功能。

1.6 作战活动与系统功能跟踪矩阵

作战活动与系统功能跟踪矩阵(简称 SV-5)主要描述作战活动与系统功能之间的对应关系,即定义哪些系统功能支持某项作战活动。作战活动与系统功能跟踪关系的属性:

$$\text{Activity_Function} := \text{Function}, \text{Activity}, \text{A_F_Relationship}[, \text{other_performance}]$$

其中 *A_F_Relationship* 表示作战活动与系统功能的映射类型。

2 体系结构产品相关性分析

根据 C⁴ISR 体系结构框架,体系结构必须按照作战视图、系统视图和技术标准视图规定的产品进行设计。由于体系结构产品是对同一系统不同侧面的内容的描述,体系结构产品之间在数据上是相互关联的。体系结构产品的关联特性不仅存在于同一视图产品之间,不同视图产品之间也存在相关性。在体系结构产品中,就包含一些建立不同视图或不同产品之间联系的产品,如作战活动到系统功能追踪性矩阵等。分析体系结构产品之间的相关性,是为了保证体系结构设计数据一致性。

根据体系结构产品数据相关程度的不同,可将数据之间相关性分为两类,第一类关联关系,两个产品中的一个或多个数据内容是一致的,称这两个产品是关联的。在产品设计中,其中一个产品的这部分数据可以从其他产品设计数据继承,如 OV-5 中作战活动和 OV-2 节点完成的活动;第二类为对应关系,一个产品的设计内容和另一个产品的设计内容存在一定的对应关系,产品相互影响、相互制约,但产品数据之间没有严格的继承关系。如作战节点与系统节点之间存在多对多的对应关系,需求线和接口关系也是多对多的对应关系,但数据不是相互继承的。

在前面讨论的核心体系结构产品中,产品之间的相互关系如下^[3]:

OV-2 中的需求线定义的信息交换关系与 OV-3 中的信息交换之间存在关联关系,信息交换矩阵详细描述了这些作战节点之间的信息交换。

OV-5 中的作战活动映射到 OV-2 中的作战节点,活动的输入/输出信息交换关系映射到 OV-2 的需求线。

OV-5 中的信息交换关系映射到 OV-3 中的信息交换,这种映射并不是一对一的关系。

SV-1 中的系统节点与 SV-4 中的系统功能相互关联,由系统实现这些功能,SV-1 中的接口与 SV-4 中的信息流之间存在映射关系。

SV-4 中定义的系统功能与 SV-5 中的系统功能存在关联关系。

OV-5 中的作战活动与 SV-5 中的作战活动存在关联关系。

OV-2 中的作战节点与 SV-1 中的系统存在映射关系,系统通过提供自动化的支持功能来支持作战节点;作战节点和系统节点、作战需求线和系统接口之间都是多对多的关系。

3 体系结构产品设计检验

体系结构产品设计检验包括产品数据的一致性检验、完备性检验和逻辑检验。产品数据的一致性检验主要是检查相互关联的分布在不同产品中的相同数据是否一致。一致性检验主要是针对具有关联关系的产品数据。产品数据的完备性检验主要检查体系结构产品数据是否完整,特别是检验不同体系结构产品中相关数据的完备性。产品数据的逻辑性检验则是检查存在逻辑关系产品的数据逻辑设计是否合理。

3.1 一致性检验

针对前面描述的体系结构核心产品,产品数据一致性检验的基本原则包括:

- 作战活动中完成作战活动的节点集与作战节点连接关系中定义的作战节点集一致;
- 作战活动中定义的作战活动集合与作战节点连接关系中定义的作战活动一致;
- 作战活动与系统功能跟踪矩阵描述中的作战活动集与作战活动描述中的作战活动集一致,系统功能集与系统功能描述中的系统功能集一致;
- 作战节点的信息交换集与作战信息交换矩阵中的信息交换集一致;
- 系统接口描述的系统功能集和系统功能描述的系统功能集一致;
- 信息交换矩阵中的信息交换关系的作战节点、作战活动与作战节点连接关系中需求线对应的作战节点以及作战节点完成的作战活动一致。

3.2 逻辑合理性检验

1. 作战节点完成活动逻辑关系检验算法

已知 $\forall x, y \in ON$, 若 x 是 y 的子节点, 则

$$x.ON_Activities \subseteq \bigcup_j \{A_j\} + y.ON_Activities$$

其中, $\{A_j\}$ 表示作战活动 A_j 包含的子活动的集合, $A_j \in y.ON_Activities$, $0 < j \leq |y.ON_Activities|$, 其中 $|y.ON_Activities|$ 表示集合 $y.ON_Activities$ 元素的个数。

2. 系统功能逻辑关系检验算法

已知 $\forall x, y \in System$, 若 x 是 y 的子系统, 则

$$x.Sys_Functions \subseteq \bigcup_j \{F_j\} + y.Sys_Functions$$

其中, $\{F_j\}$ 表示系统功能 F_j 包含所有子功能的集合, $F_j \in y.Sys_Functions$, $0 < j \leq |y.Sys_Functions|$ 。

3.3 完备性检验原则

完备性检验的基本原则包括:

- 每个作战节点至少完成一个以上的活动, 即 $ON.ON_Activities \neq \Phi$;
- 每个作战活动至少由一个作战节点完成, 即 $Activity.Perform_ON \neq \Phi$;
- 每个组织至少与一个作战节点对应, 即 $Organization.Organization_Nodes \neq \Phi$;
- 每条需求线至少包含一个信息交换, 即 $NL.IERs \neq \Phi$;
- 每个信息交换至少包含在一条需求线中, 即 $IER.NL \neq \Phi$;
- 系统至少具有一个以上的系统功能, 即 $System.Sys_Functions \neq \Phi$;
- 每个系统功能至少由一个以上的系统完成, 即 $System_Function.Entities \neq \Phi$ 。

4 结束语

体系结构产品设计是体系结构框架规范的体系结构设计内容, 因此保证体系结构产品设计质量是提高体系结构设计质量的重要环节。本文通过对作战节点连接关系、作战活动模型、系统接口描述等核心产品的规范化描述, 在分析这些产品数据之间关系的基础上, 特别强调了以数据为中心的体系结构产品设计, 提出上述产品数据一致性、逻辑性和完备性检验的原则和方法。这些研究成果对研制体系结构产品开发工具、检验体系结构的设计质量有指导作用。

参考文献:

- [1] C⁴ISR Architecture Working Group. C⁴ISR Architecture Framework Version 2.0[R]. The United States: Department of Defense, 1997.
- [2] DoD Architecture Framework Working Group. DoD Architecture Framework, Version 1.0[R]. The United States: Department of Defense, 2003.
- [3] 修胜龙. C⁴ISR 体系结构产品一致性开发及验证方法研究[D]. 国防科技大学, 2004, 12.
- [4] Wagenhals L W, Shin I, Kim D, Lewis A H. C⁴ISR Architectures II: Structured Analysis Approach for Architecture Design [J]. Systems Engineering, 2000, 3(4):248-287.
- [5] Bienvu M P, Kim D, Lewis A H. C⁴ISR Architectures III: An Object-oriented Approach to Architecture Design [J]. Systems Engineering, 2000, 3(4):288-312.

