

文章编号: 1001- 2486(2010) 05- 0118- 05

信息系统体系结构设计中系统内聚度分析方法*

罗爱民

(国防科技大学 信息系统工程重点实验室, 湖南 长沙 410073)

摘要: 系统的组成及其结构是体系结构中的一个核心元素, 系统内聚度是评价信息系统体系结构的一个重要指标。以体系结构框架为基础, 给出系统接口和系统功能的规范化描述。从功能和数据关系的角度, 分别定义了系统功能内聚度和数据内聚度的计算方法, 提出一种体系结构中系统内聚度的分析方法, 并通过一个实例说明了方法的应用。该方法有助于规划系统和分配功能, 提高体系结构设计质量。

关键词: 系统内聚度; 系统功能; 体系结构

中图分类号: TP391 文献标识码: A

System Cohesion Analysis Method on Information Systems Architecture

LUO Ai-min

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

Abstract: System cohesion is an important evaluation factor in information system architecture. Based on the architecture framework, the formal description of system interfaces and system functions were provided. In terms of functions and data relationships, the function cohesion and data cohesion were defined respectively, and the system cohesion evaluation method was proposed. Furthermore, an application example was presented. The method is helpful in system design, function assignment, and improving the quality of architecture development.

Key words: system cohesion; system functions; architecture

体系结构是信息系统的建设蓝图。对于复杂的信息系统, 科学合理的体系结构, 可为系统设计与研制、系统集成提供指导。因此, 体系结构设计已经成为复杂信息系统设计的一个重要阶段。

系统作为体系结构设计的最基本单元, 其设计的好坏直接关系到体系结构的设计质量。一般讲, 在满足系统功能需求的前提下, 系统的内聚度越高, 耦合性越低, 其复杂性就越低, 也越容易理解、实现、修改和维护。因此, 在体系结构设计中, 希望系统具有较高的内聚度和较低的耦合性。

目前, 国内外在内聚度分析方法方面开展了不少研究, 其中大多是针对程序设计和软件体系结构的内聚度分析^[1-9]。这些方法主要以类、属性、方法等作为分析对象。这些研究是针对系统设计层面的, 研究成果难以在体系结构中应用。此外, 针对业务流程的复杂性, 文献[10-11]从角色任务分配的角度, 研究角色的活动内聚度、数据内聚度和耦合性。对于信息系统体系结构, 文献[12]提出一种系统内聚度分析方法, 主要考虑系统接口描述中系统之间数据交换关系。而在体系结构设计中, 系统的提取和设计的基本依据是功能以及功能之间的关系。因此本文从系统完成功能的角度, 定义了系统的功能内聚度和数据内聚度的概念, 并提出一种针对体系结构的系统内聚度分析方法。

1 体系结构中的系统设计

目前体系结构设计多采用通用的体系结构框架, 如美国国防部体系结构框架(DoD Architecture Framework, DoD AF)、Zachman 框架、开发组织体系结构框架(The Open Group Architecture Framework)等。

* 收稿日期: 2010- 07- 14

基金项目: 国家部委资助项目

作者简介: 罗爱民(1970-), 女, 副教授, 博士。

这些框架规划与设计系统的方法基本类似。因此,本文选用通用体系结构框架——DoD 体系结构框架作为研究对象。

按照 DoD 体系结构框架,系统规划和设计主要在系统视图中完成。系统视图产品的设计首先要根据业务视图得到的业务流程和活动,分析体系结构必须具有的功能。在得到功能及其关系的基础上,再规划、设计系统组成和结构。在设计系统时,应合理划分功能,并将它们分配给相应的系统。系统的规划和设计要充分考虑到功能组成以及功能之间数据交换关系,只有将功能合理地分配给相应的系统,系统才会具有较高的内聚度。体系结构中的系统设计是一个人工参与分析的过程,其规划与设计人员的设计水平、经验知识密切相关。在不同的设计方案中,系统内聚度会存在差别。

与系统设计密切相关的两个系统视图产品是系统功能描述和系统接口关系。其中系统功能描述(SV-4)主要反映体系结构完成的系统功能以及相互关系。系统接口关系(SV-1)主要描述系统的组成、结构以及它们之间的数据交换关系。为了描述方便,根据体系结构框架中产品的描述规范^[13-14],对这两个产品进行规范化描述。

定义1 SV-4 是一个五元组,

$$SV4 = (F, FF_R, D, D_Inputs, D_Outputs)$$

其中,(1) F 表示功能集合, $F = \{function_i \mid i = 1, 2, \dots, M\}$ 。

(2) $FF_R: F \rightarrow F$ 是功能之间有向弧,表示功能与功能之间的数据流向关系。若功能 f_i 到 f_j 有数据流向关系,则功能 f_i 和 f_j 之间存在一条有向弧, $f_i \rightarrow f_j$ 。一般情况下,设计中不允许自环。

(3) D 表示完成功能所需要的数据集, $D = \{data_i \mid i = 1, 2, \dots, K\}, K \geq 1$ 。

(4) $D_Input: D \rightarrow F$ 是数据集到功能之间的映射函数, $D_Input(f_i)$ 表示功能 f_i 的输入数据集。

(5) $D_Output: FS \rightarrow D$ 是功能到数据集之间的映射函数, $D_Output(f_i)$ 表示功能 f_i 的输出数据集。

在体系结构设计中,SV-4 通常可以采用数据流图来描述。

定义2 系统接口描述 SV-1 是一个七元组,

$$SV1 = (SN, SS, SF, NN_DR, SS_DR, SN_R, SS_R, S_F)$$

其中,(1) SN 表示系统节点集合,设 $SN = \{node_i \mid i = 1, 2, \dots, L\}$ 。

(2) SS 表示系统集,设 $SS = \{system_i \mid i = 1, 2, \dots, N\}$ 。

(3) F 表示功能集, $F = \{function_i \mid i = 1, 2, \dots, M\}$ 。

(4) $NN_DR: SN \rightarrow SN$ 表示系统节点与系统节点的数据映射关系。若系统节点 $node_i$ 与 $node_j$ 之间存在数据交换关系,则 $node_i \rightarrow node_j$ 。

(5) $SS_DR: SS \rightarrow SS$ 表示系统与系统之间的数据映射关系。若系统 $system_i$ 与 $system_j$ 之间存在数据交换关系,则 $system_i \rightarrow system_j$ 。

(6) $SN_R: SS \rightarrow SN$ 描述系统集与系统节点集之间的映射关系,反映系统在系统节点上的部署关系。

(7) $SF_R: SS \rightarrow F$ 是系统与功能之间的函数关系。 $SF_R(S)$ 表示系统 S 完成的功能集。 $|SF_R(S)|$ 表示功能集的元素个数。

2 系统内聚度分析

在体系结构设计中,功能及其关系是系统规划和设计的基础。系统是完成功能的对象或实体。系统完成功能的执行关系越紧密,数据关联关系越强,那么系统的内聚度就越高。因此,系统内聚度与功能复杂性以及功能之间数据流关系密切相关。本文提出利用两个指标来分析系统内聚度,即系统的功能内聚度和数据内聚度。

功能内聚性主要从功能执行的紧密程度来分析系统的内聚度,即主要通过分析系统完成多个功能执行顺序和相关程度来评估系统的内聚度。系统完成多个功能之间的执行间隔越短,则系统的功能内聚性越高。功能间的执行间隔可用功能之间的距离来描述。

定义3 设 $F = \{f_1, f_2, \dots, f_N\}$ 是系统功能流图中 f_1, \dots, f_N 的执行过程。 q 是功能 f_i 与 f_j 之间相隔的功能数, 则功能 f_i 与 f_j 之间的距离为 $q+1$ 。

如果任意功能 f_i 与 f_j 之间不存在执行顺序关系, 则设功能 f_i 与 f_j 之间的距离为 0。

若系统 S 完成的功能 f_i 与 f_j 之间存在多条路径, 则功能 f_i 与 f_j 之间的距离取多条路径的距离平均。一般情况下, 在体系结构设计中, 功能描述不允许自环。

设 $d(f_i, f_j)$ 表示功能 f_i 与 f_j 之间的一条距离。如在系统 S 中, 功能 f_i 与 f_j 之间存在 L 条路径, 则在系统 S 中, 功能 f_i 与 f_j 之间的平均距离 $D(f_i, f_j)$

$$D(f_i, f_j) = \frac{\sum_{k=1}^L d_k(f_i, f_j)}{L} \quad (1)$$

假设系统 S , 其完成的功能集为 $SF_R(S)$, 这些功能之间的关系由 SV-4 中相关数据决定, 那么系统 S 的功能内聚度定义如下。

定义4 系统 S 的功能内聚度 C_F

$$C_F(S) = \begin{cases} \frac{1 + \sum_{f_i, f_j \in SF_R(S), i \neq j} (Max_D - D(f_i, f_j))}{P^2_{|SF_R(S)|} \times Max_D} & Max_D \neq 0 \\ \frac{1}{P^2_{|SF_R(S)|}} & Max_D = 0 \end{cases} \quad (2)$$

其中, Max_D 表示体系结构设计中, 系统执行功能流图中功能之间的最大距离, $P^2_{|SF_R(S)|}$ 表示系统 S 完成的功能中任意两个的排列数。

式(2)反映功能之间的关系对系统内聚度的影响。系统完成的功能数量以及功能之间的关系决定功能内聚度的大小。功能之间的距离越长, 则功能内聚度越低。功能的数量越多, 功能内聚度越低。功能内聚度取值在 $(0, 1]$ 区间, 当系统完成单一功能时, 功能内聚度最大, 取值为 1。当功能内聚度趋近 0 时, 功能内聚度最差。

通常 SV-4 采用数据流图来描述。数据流图可以认为是一个有向图 $G = (F, V, W)$ 。其中 F 表示功能集, 是有向图中的顶点集, V 表示功能之间的数据流关系, 是图中的有向弧集, W 是权重集, 表示有向弧的权重。

根据 SV-4 数据可建立关联矩阵 R , 利用图论中相关算法可以得到可达矩阵 A 。若功能 f_i, f_j 存在执行顺序关系, 则可达矩阵 A 的元素 $a_{ij} = 1$ 。通过遍历算法可以得到 f_i, f_j 之间的路径, 进而可以计算 f_i, f_j 之间的距离 $d(f_i, f_j)$ 。

除系统完成功能之间的距离影响系统内聚度之外, 系统中数据内聚性也是影响系统内聚度的重要因素。数据内聚性从数据关系的角度衡量系统的内聚情况, 它关注不同数据的使用次数及功能中的共用数据情况。若不同功能之间的数据交集较大, 系统所负责的多个功能之间的共性更多。

根据定义 1, 对于功能 f_i 来说, 与它相关的数据集

$$D(f_i) = D_Input(f_i) \cup D_Output(f_i) \quad (3)$$

定义5 系统 S 的数据内聚度

$$C_D(S) = \begin{cases} \frac{\sum_{f_i, f_j \in SF_R(S), i \neq j} |D(f_i) \cap D(f_j)|}{C^2_{|SF_R(S)|} \times |D_s|} & |SF_R(S)| > 1 \\ 1 & |SF_R(S)| = 1 \end{cases} \quad (4)$$

其中, $|x|$ 表示集合 x 的元素个数, D_s 表示系统 S 完成功能涉及的数据集, $D_s = \bigcup_{f_i \in SF_R(S)} D(f_i)$ 。

$C^2_{|SF_R(S)|}$ 表示系统 S 完成的功能中任意两个的组合数。

按照定义 5, 当系统 S 只完成一个功能时, 其数据内聚性最大, 数据内聚度为 1。当系统完成多个功

能时, 功能之间的数据交集元素越多时, 系统数据内聚度越大。

可以证明式(4)定义的数据内聚度满足 Briand 提出的内聚度度量准则的基本性质^[8-9]。

定义 6 系统 S 的内聚度

$$C_o(S) = C_{-F}(S) \times C_{-D}(S) \tag{5}$$

系统内聚性为系统完成功能的相关性程度, 高内聚性既表示系统完成的功能之间具有较短的执行间隔, 也表示不同功能之间的数据交集较大, 这也意味着系统所负责的多个功能之间的共性更多。

定义 7 信息系统体系结构 X 的系统内聚度为

$$C_o(X) = \sum_{S_i \in SS} \frac{|SF_{-R}(S_i)|}{|F|} C_o(S_i) \tag{6}$$

其中, SS 是系统接口描述中系统集, F 是功能集合。

3 实例分析

设某信息系统体系结构设计中, 系统功能流图如图 1 所示。假设上述功能由四个系统来完成, 设系统集 $S = \{S1, S2, S3, S4\}$ 。

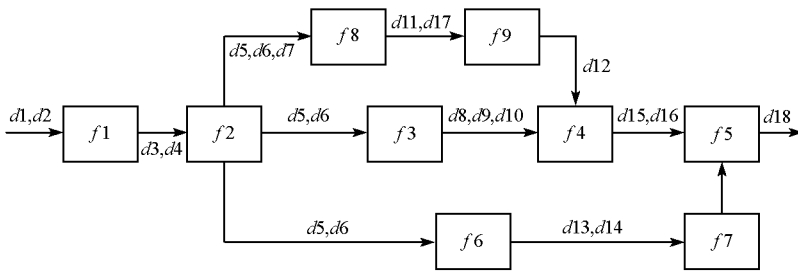


图 1 某系统的系统功能流图

Fig. 1 The function flow diagram of a system

根据图 1, 可以得到数据流和功能组成关系如下。

$$D = \{d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12, d13, d14, d15, d16, d17, d18\}$$

$$F = \{f1, f2, f3, f4, f5, f6, f7, f8, f9\}$$

假设存在两种设计方案。在方案 1 中, 各系统完成的功能分配如下:

$$SF_{-R}(S1) = \{f1, f2\}, SF_{-R}(S2) = \{f8, f9\}, SF_{-R}(S3) = \{f3, f4\}, SF_{-R}(S4) = \{f6, f7, f5\}。$$

在方案 2 中, 各系统完成的功能分配如下:

$$SF_{-R}(S1) = \{f1, f5\}, SF_{-R}(S2) = \{f2, f8, f6, f3\}, SF_{-R}(S3) = \{f9, f4\}, SF_{-R}(S4) = \{f7\}。$$

按照方案 1, 计算各系统的功能内聚度分别为

$$C_{-F1}(S1) = \frac{1+(4-1)}{2 \times 4} = \frac{1}{2}, C_{-F1}(S2) = \frac{1}{2}, C_{-F1}(S3) = \frac{1}{2}, C_{-F1}(S4) = \frac{1}{4}。$$

在方案 1 中, 各系统的数据内聚度分别为

$$C_{-D1}(S1) = \frac{2}{7}, C_{-D1}(S2) = \frac{2}{6} = \frac{1}{3}, C_{-D1}(S3) = \frac{3}{8}, C_{-D1}(S4) = \frac{3}{16}。$$

按照方案 2, 各系统的功能内聚度分别为

$$C_{-F2}(S1) = \frac{1}{8}, C_{-F2}(S2) = \frac{5}{24}, C_{-F2}(S3) = \frac{1}{2}, C_{-F2}(S4) = 1。$$

计算得到方案 2 中各系统的数据内聚度分别为

$$C_{-D2}(S1) = 0, C_{-D2}(S2) = \frac{7}{72}, C_{-D2}(S3) = \frac{1}{8}, C_{-D2}(S4) = 1。$$

在方案 1 中, 系统 $S1$ 分配了功能 $f1, f2$ 。在方案 2 中, 系统 $S1$ 分配了功能 $f1, f5$ 。但是 $C_{-F1}(S1) > C_{-F2}(S1)$ 。因为 $f1, f5$ 之间的距离较大, 所以方案 2 中的内聚度较小。在方案 1 和方案 2 中, $S3$ 的功能内聚度相同, $C_{-F1}(S3) = C_{-F2}(S3)$, 但是 $S3$ 的数据内聚度 $C_{-D1}(S3) > C_{-D2}(S3)$, 因为 $f3$ 与

f_4 之间的数据关联性比 f_9 与 f_4 之间的数据关联性要紧密。方案 1 的系统内聚度为 0.128, 方案 2 的系统内聚度为 0.134。方案 2 的系统内聚度略好于方案 1, 因为方案 2 中系统 S_2 和 S_4 的设计比较合理。

4 结束语

功能及其关系是决定系统内聚度的关键因素。本文从系统成功能的角度, 定义了功能内聚度和数据内聚度指标, 并以此为基础提出一种系统内聚度分析方法。该方法对体系结构设计中系统的提取和规划有指导意义。目前本文主要分析系统的内聚度评估问题, 系统耦合度也是影响系统规划的一个重要因素。因此, 在后续工作中, 要进一步研究系统耦合度分析方法。结合内聚度和耦合度两项指标, 进一步完善体系结构设计中系统规划的分析方法。

参考文献:

- [1] 刘宗田. 面向对象软件度量中的内聚缺乏度分析[J]. 计算机研究与发展, 2001, 38(2): 138-139.
- [2] Briand L, Daly H W. A Unified Framework for Coupling Measurement in Object Oriented Systems[J]. IEEE Transactions on Software Engineering, 1999, 25(1): 91-121.
- [3] Allen E B, Khoshgoftaar T M. Measuring Coupling and Cohesion: An Information-theory Approach[C]//Proceedings of the 6th International Software Metrics Symposium. IEEE Computer Society, 1999: 119-127.
- [4] 陈振强, 徐宝文. 一种基于依赖性分析的类内聚度度量方法[J]. 软件学报, 2003, 14(11): 1850-1856.
- [5] Chae H S, Kwon Y R. A Cohesion Measure for Classes in Object-oriented Systems[C]//Proceedings of the 5th International Software Metrics Symposium. IEEE Computer Society Press, 1998: 158-166.
- [6] Chen Z Q, Zhou Y M, Xu B W, et al. A Novel Approach to Measuring Class Cohesion Based on Dependence Analysis[C]. IEEE International Conference on Software Maintenance (ICSM 2002). 2002: 377-383.
- [7] 唐家华, 李文宏, 李彤. 一种基于结构熵的类内聚度度量方法[J]. 系统工程与电子技术, 2009, 31(3): 702-704.
- [8] Briand L C, Morasca S, Basili V R. Property-based Software Engineering Measurement[J]. IEEE Transactions on Software Engineering, 1996, 22(1): 68-85.
- [9] Briand L C, Daly J W, Wust J K. A Unified Framework for Coupling Measurement in Object-oriented System[J]. Empirical Software Engineering, 1998, 3(1): 65-117.
- [10] Vanderfeesten I, Reiers H A, Aalst W M P. Evaluating Workflow Process Designs Using Cohesion and Coupling metrics [EB/OL]. (2007-07-02). <http://is.tn.tue.nl/staff/wvdaals/BPMcenter/reports/2007/BPM-07-02.pdf>
- [11] 颜志军, 王天梅. 业务流程的角色复杂性[J]. 北京理工大学学报, 2008, 28(3): 278-232.
- [12] 姜志平. 基于 CADM 的 C⁴ISR 系统体系结构验证方法及关键技术研究[D]. 国防科技大学, 2007.
- [13] DoD Architecture Framework Working Group. DoD Architecture Framework Version 1.0 [R]. U.S.: Department of Defense, 2003.
- [14] DoD Architecture Framework Working Group. DoD Architecture Framework Version 2.0 [R]. U.S.: Department of Defense, 2009.