

信息系统体系结构的复杂性分析方法

罗爱民

(国防科技大学信息系统工程重点实验室 长沙 410073)

摘要 体系结构设计是信息系统设计中的一个重要环节。体系结构的复杂性是影响系统质量的重要因素。针对信息系统体系结构设计的特点,定义了系统耦合度和内聚度计算模型;基于耦合度和内聚度评估结果,提出一种体系结构复杂度的分析方法。实例证明,该方法是一种有效的复杂性评估方法。

关键词 复杂度,内聚度,耦合度,体系结构

中图分类号 TP311.5 **文献标识码** A

Complexity Analysis Method on Information Systems Architecture

LUO Ai-min

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

Abstract The architecture design is an important phase in information system development. The complexity of architecture is an key influence factor for information systems. According to the characteristics of information systems architecture development, the measure model of system Cohesion and that of system couple were defined in the paper. Based on these measure models, a complexity analysis method on information systems architecture was provided. Furthermore, an application example was given to illustrate the method is effective.

Keywords Complexity, Cohesion, Couple, Architecture

1 引言

信息系统体系结构复杂度决定了它所对应系统的复杂度。一般来说,若完成的功能相同,体系结构越复杂,开发系统所需的成本就越高,带来的开发风险也越大。因此,体系结构设计中必须考虑复杂性的问题。

系统作为体系结构设计的最基本单元,其结构的复杂性直接反映体系结构的复杂性。通常最简单的系统结构复杂性分析方法就是分析体系结构总的节点数、链路数、层次结构等,以这些指标来衡量结构的复杂程度。虽然这些指标是决定结构复杂度的关键指标,但实际上,相同的节点数和层次关系,其结构复杂性常常也存在较大差别。因此,单纯从节点、链路、层次等的数量分析复杂性是不全面的。

目前,就复杂性问题已经开展了很多研究^[1-12],但这些研究主要集中在研究软件、程序和业务等复杂性问题。目前软件或程序结构复杂性主要通过评估软件设计中构件以及连接关系的复杂性来分析,其中构件的耦合度和内聚度是评价复杂性的两个重要指标。由于这些方法是以软件设计中的基本单位——构件,或者对象,或者类以及它们之间的关系为基本分析对象,因此难以将这些成果直接用于信息系统体系结构分析。针对信息系统的复杂性,可采用熵理论,通过计算各组成部分的结构熵来分析、研究业务流程和体系结构的复杂性^[13,14]。但是这些主要是针对结构化的设计过程,而且复杂度的计算有较大重复,不能很好地反映体系结构的复杂程度。

一般来讲,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,系统的复杂性就越低,系统也越容易理解、开发和维护。因此,在体系结构设计中,希望其中的系统具有较高的内聚度和较低的耦合性,以降低体系结构的复杂性。本文根据信息系统体系结构设计的特点,提出一种综合考虑系统耦合度和内聚度的复杂性分析方法。

2 体系结构设计中的系统设计

为规范和科学地设计信息系统体系结构,目前信息系统体系结构设计多采用多视图的设计方法和通用体系结构框架,如美国国防部体系结构框架(DoD Architecture Framework 1.0)、Zachman 框架、开发组织体系结构框架(The Open Group Architecture Framework, TOGAF)等。在这些框架中,系统组成与结构的描述内容和描述方法基本类似。因此,本文选用 DoD 体系结构框架 1.0^[15]作为研究对象,其他框架可以参照此方法分析。

按照 DoD 体系结构框架,系统设计要按照从业务需求到系统实现的过程完成。在设计体系结构时,首先设计业务视图。在得到业务视图中业务流程和活动的基础上,分析体系结构必须完成的系统功能,再根据功能及其关系,规划和设计系统组成以及结构,并将功能分配给相应的系统。功能的分配以及由此产生的数据交换关系决定了系统组成、结构以及数据交换关系,这些因素是影响系统内聚度和耦合度的主要因素。

系统视图中与系统设计密切相关的产品主要有系统功能描述(SV-4)、系统接口关系(SV-1)和系统数据交换矩阵(SV-

6)。其中系统功能描述主要反映信息系统完成的系统功能及其相互关系,系统接口关系主要描述系统的组成、结构以及系统之间的数据交换关系,系统数据交换关系则描述系统之间进行数据交换的具体内容、交换要求。这些产品的详细介绍可参看有关文献[16]。

为了描述方便,根据体系结构框架中产品的描述规范^[12],对上述3个产品进行规范化描述。

定义1 系统接口描述是一个六元组 $SV1 = (SN, SS, SF, INFs, SN_R, S_F)$ 。

其中,

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

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

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

(4) $INFs, SS \rightarrow SS$ 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 $inf_k: system_i \rightarrow system_j$ 。

(5) $SN_R, SS \rightarrow SN$ 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。

(6) $SF_R, SS \rightarrow F$ 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。

定义2 系统数据交换矩阵是一个三元组 $SDEM = \{DERs, INFs, ID_R\}$ 。

其中,

(1) $DERs$ 表示数据交换关系的集合, $0 < |DERs| < +\infty$, $der \in DERs$, der 描述每条数据交换的属性,如数据量、类型、交换频度等。

(2) $INFs$ 表示接口关系集合, $inf \in INFs$ 。

(3) $ID_R, INFs \rightarrow DERs$ 表示接口与数据交换的映射关系, $ID_R(inf_i)$ 表示 inf_i 完成数据交换的集合。

定义3 $SV-4$ 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, $F = \{function_i | i=1, 2, \dots, M\}$ 。

(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 到 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, $D = \{data_i | i=1, 2, \dots, K\}$, $K \geq 1$ 。

(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。

3 系统耦合性分析

系统耦合性描述体系结构中不同系统之间的关联程度。如果两个系统之间存在信息交换关系,则这两个系统是相互关联的,存在耦合关系。在信息系统体系结构设计中,系统耦合性可以根据 $SV-1$ 和 $SV-6$ 来计算。其中, $SV-1$ 中的 $INFs$ 体现系统之间的数据交换关系, $SV-6$ 则定义了每个 $INFs$ 的数据交换数量和属性。

$SV-1$ 采用结构图的方式描述系统组成与结构、系统之间

的接口关系,因此根据 $SV-1$ 可以构建无向图。其中,系统相当于无向图中的顶点,接口 inf_k 表示系统 S_i 与 S_j 之间的数据交换关系,对应顶点 S_i 与 S_j 之间的弧。 inf_k 的权重相当于对应弧的权重。利用 $SV-1$ 构建无向图,计算每条弧的权重,从而计算系统的耦合度。

定义4 接口 $inf_k = (S_i, S_j)$ 的权重

$$W_{(S_i, S_j)} = \frac{|ID_R(inf_k)|}{|DERs|} \quad (1)$$

式中, $|DERs|$ 表示 $SV-6$ 中定义数据的交换总量, $ID_R(inf_i)$ 表示接口 inf_i 中数据交换元素的集合。

定义5 系统 S_i 耦合度为

$$Cou(S_i) = \frac{\sum_j W_{(S_i, S_j)} + \sum_j W_{(S_j, S_i)}}{C_{|ss|}^i} \quad (2)$$

式中, $C_{|ss|}^i$ 表示 $SV-1$ 对应的无向图中弧最大数量, $0 \leq Cou(S_i) < 1$ 。当系统与其他系统都没有发生数据交换关系时,即在 $SV-1$ 中,系统与其他系统没有接口关系,该系统的耦合度为0。通常在体系结构设计中,不允许出现独立系统单元。

定义6 信息系统体系结构中系统耦合度为

$$Cou(S) = \frac{\sum_i Cou(S_i)}{2} \quad (3)$$

4 系统内聚度分析

一般来讲,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,系统的内聚度可以通过功能内聚性和数据内聚性来描述。其中,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。数据内聚性从数据关系的角度来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,则系统所负责的多个功能之间的共性较多。

根据上面的分析,系统内聚度可利用 $SV-4$ 的设计数据进行分析。

定义7 系统 S_i 的内聚度为

$$Coh(S_i) = C_F(S_i) \times C_D(S_i) \quad (4)$$

式中, $C_F(S_i)$ 表示系统 S 的功能内聚度, $C_D(S_i)$ 表示系统 S_i 的数据内聚度。系统 S_i 的功能内聚度和数据内聚度的详细计算过程和方法可参看文献[17]。

定义8 信息系统体系结构的内聚度为

$$Coh(S) = \sum_{S_i \in SS} \frac{|SF_R(S_i)|}{|F|} Coh(S_i) \quad (5)$$

式中, SS 是系统接口描述中系统集, F 是功能集合, $SF_R(S)$ 表示系统 S 完成的功能集。

5 系统复杂性分析

一般来讲,系统的复杂性是与系统内聚度和耦合度密切相关的,系统内聚性越高、耦合性越低,则系统的复杂性越低。因此,可采用耦合性与内聚度的比值的大小来描述系统的复杂性。

定义9 系统 S_i 的复杂性为

$$Com(S_i) = \frac{Cou(S_i)}{Coh(S_i)} \quad (6)$$

定义 10 信息系统体系结构 S 的复杂度为

$$Com(S) = \frac{Cou(S)}{Coh(S)} \quad (7)$$

6 实例分析

设某信息系统体系结构设计中,系统功能流程图如图 1 所示。

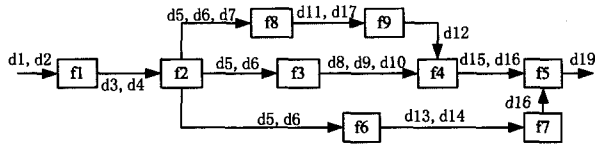


图 1 某系统的系统功能流程图

假设上述系统功能由 4 个系统来完成,设系统集 $S = \{S1, S2, S3, S4\}$ 。根据图 1,可以得到如下数据:

$$D = \{d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, d12, d13, d14, d15, d16, d17, d18, d19\}$$

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

现假设存在两种体系结构设计方案。

(1) 方案 1 中, $SF_R(S1) = \{f1, f2\}$, $SF_R(S2) = \{f8, f9\}$, $SF_R(S3) = \{f3, f4\}$, $SF_R(S4) = \{f6, f7, f5\}$ 。SV-1 对应无向图如图 2 所示。

(2) 方案 2 中, $SF_R(S1) = \{f1, f5\}$, $SF_R(S2) = \{f2, f8, f6, f3\}$, $SF_R(S3) = \{f9, f4\}$, $SF_R(S4) = \{f7\}$ 。SV-1 对应的无向图如图 3 所示。

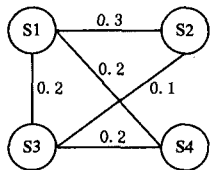


图 2 方案 1 中 SV-1 对应的无向图

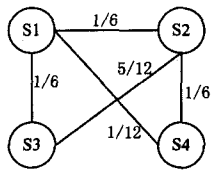


图 3 方案 2 中 SV-1 对应的无向图

按照方案 1,计算各系统的内聚度、耦合度以及复杂度分别为

$$Coh_1(S1) = 0.0375, Coh_1(S2) = 0.1667$$

$$Coh_1(S3) = 0.1875, Coh_1(S4) = 0.075$$

$$Coh(A_1) = 0.1112, Cou_1(S_1) = 0.1167$$

$$Cou_1(S_2) = 0.0667, Cou_1(S_3) = 0.0833$$

$$Cou_1(S_4) = 0.0667$$

$$Cou(A_1) = \frac{\sum_i Cou(S_i)}{2} = 0.1667$$

按照方案 2,各系统的内聚度、耦合度以及复杂度分别为

$$Coh_2(S1) = 0, Coh_2(S2) = 0.02$$

$$Coh_2(S3) = 0.0625, Coh_2(S4) = 1$$

$$Coh(A_2) = 0.1338, Cou_2(S_1) = 0.0694$$

$$Cou_2(S_2) = 0.125, Cou_2(S_3) = 0.083$$

$$Cou_2(S_4) = 0.04167$$

$$Cou(A_2) = \frac{\sum_i Cou(S_i)}{2} = 0.1595$$

根据前面的计算结果,得到两种体系结构设计方案的复杂度分别为

$$A_1 = 1.499, A_2 = 0.3114$$

第二种设计方案优于第一种方案。

结束语 体系结构的复杂性是影响系统复杂性的关键因素之一。本文结合体系结构设计的特点,利用体系结构的设计数据,定义了系统耦合度和内聚度的计算方法,并提出一种基于耦合度和内聚度的评估结果的体系结构复杂度的分析方法。该方法对体系结构设计中系统单元的定义和体系结构方案的决策具有指导意义。目前复杂性度量方法主要从数据交换的数量来分析。在系统实现中,数据交换的数据量以及实现方式也是影响复杂性的因素,在后续研究中应该考虑这方面的因素。

参考文献

- [1] Zhou Yu-ming, Xu Bao-wen. Measuring structural complexity for Class Diagrams: An Information Theory Approach[C] // SAC'05. 2005;1679-1683
- [2] Darcy D P, Kemerer C F, Slaughter S A, et al. The Structural Complexity of Software: An Experimental Test [J]. IEEE Transactions on Software Engineering, 2005, 31(11): 982-995
- [3] Hung H Q, Marechal Y, Coulomb J L. Managing Structure Complexity in a Multi-physics Simulation Software[J]. IEEE Transactions on Magnetics, 2006, 42(4): 1239-1292
- [4] Abran A, Ormandjieva O, Abu Talib M. A Functional Size and Information Theory-based Functional Complexity Measures: Exploratory study of related concepts using COSMIC-FFP measurement method as a case study[C] // 14th International Workshop of Software Measurement (IWSM-MetriKon 2004). Königs Wusterhausen, Germany; Shaker-Verlag, 2004; 457-471
- [5] Claussen J C. Characterization of networks by the off diagonal complexity[J]. Physica A, 2007; 321-354
- [6] 黄颢, 高建华. 面向对象设计复杂性度量计算与分析. 计算机工程与设计[J]. 2006, 27(8): 1346-1348, 1356
- [7] 李兵, 王浩, 李增扬, 等. 基于复杂网络的软件复杂性度量研究[J]. 电子学报, 2006, 34(12A): 2371-2375
- [8] 黄万良, 陈松乔. 基于构件运算的软件体系结构及其复杂性度量[J]. 计算机工程与应用, 2007, 43(14): 66-70
- [9] 焦锋, 王丽平, 侯建民. 一种软件体系结构复杂性的评价方法[J]. 计算机应用研究, 25(8): 2377-2379, 2449
- [10] Bandi R K, Vaishnavi V K, Turk D E. Predicting maintenance performance using object-oriented design complexity metrics [J]. IEEE Transactions on Software Engineering, 2003, 29 (1): 77-87
- [11] Harrison W. An entropy-based measure of software complexity [J]. IEEE Transactions on Software Engineering, 1992, 18: 1025-1029
- [12] Sinder G. Measuring the Entropy of Large Software Systems [R]. HPL_2001_221. Palo Alto; HP Laboratories, 2001
- [13] 王忠杰, 徐晓飞, 战德臣. 基于熵的信息系统业务模型复杂性度量[J]. 计算机科学, 2006, 33(1): 104-107
- [14] Luo Ai-min, Huang Li. Entropy-based complexity analysis for architecture of information system[C] // 2008 IEEE International Conference on Networking Sensing and Control. Sensing and Control. Volume II, 2008; 1734-1738
- [15] DoD Architecture Framework Working Group. DoD Architecture Framework[S]. Version 1. 0, Volume I: Definitions and Guidelines. U. S; Department of Defense, 2003
- [16] DoD Architecture Framework Working Group. DoD Architecture Framework[S]. Version 1. 0, Volume II: Definitions and Guidelines. U. S; Department of Defense, 2003
- [17] 罗爱民. 信息系统体系结构内聚度分析方法[J]. 国防科技大学学报, 2010, 32(11): 77-81