

引用格式:秦怀斌,郑瑶,梁斌.一种控制软件体系结构复杂性的方法[J].微电子学与计算机,2019,36(1):32-36. [QIN H B, ZHENG Y, LIANG B. A Control Method of Complexity for Software Architecture[J]. Microelectronics & Computer, 2019, 36 (1): 32-36.]

一种控制软件体系结构复杂性的方法

秦怀斌,郑瑶,梁斌

(石河子大学 信息科学与技术学院,新疆 石河子 832003)

摘 要: SA 作为软件系统的高层抽象及后续开发、测试、维护等阶段工作的依据,其复杂性对系统影响深远.依据 CN 及 SA 理论特点,提出了利用 CN 进行 SA 复杂性的度量及控制方法.论文首先介绍了复杂网络及无向无权网络方法,在此基础上,提出了通过构件的度、构件的度分布、SA 簇系数、SA 平均路径长度、SA 模块度等度量属性进行 SA 复杂性的度量,并给出了 SA 复杂性控制方法及具体度量属性值的参考范围.最后,对给出的 SA 复杂性的度量方法及控制方法进行了实例验证.结果表明,该方法在严格的形式化基础上,可以较好地对所开发系统的 SA 复杂性的属性进行度量并进行控制,为系统后续开发及维护等阶段的工作提供支持和参考.

关键词: 软件体系结构;复杂网络;复杂性;控制

中图分类号: TP311

文献标识码: A

文章编号: 1000—7180(2019)01—0032—05

A Control Method of Complexity for Software Architecture

QIN Huai-bin, ZHENG Yao, LIANG Bin

(College of Information Science and Technology, Shihezi University, Shihezi 832003, China)

Abstract: As the basis of the high-level abstraction of the software system and subsequent development, testing and maintenance, the complexity of SA has a far-reaching impact on the system. According to the characteristics of CN and SA, the measurement and control method of SA complexity by using CN is proposed. This paper first introduces the method of complex networks and undirected and unweighted network, on this basis, the measurement of SA complexity is proposed through the degree of component, degree distribution of component, clustering efficient of SA, average path length of SA, modularity of SA, etc, then the complexity control method of SA and the reference range of the specific measurement attribute value are also given. Finally, the measurement method and control method of the SA complexity are verified by an example. The results shows that this method can metric and control the complexity of software system well based on strict formalization, and provides support and reference for subsequent development and maintenance of SA.

Key words: software architecture; complex networks; complexity; control

1 引言

软件体系结构(Software Architecture, SA)自提出以来,就日益受到研究者及实践者的关注,并发展成为软件工程一个重要的研究领域^[1].作为系统设计的高层抽象,SA体现了系统设计的基本思想

和系统开发中的早期决策,对系统具有深远影响.作为系统实现的蓝本,SA的复杂性直接影响到系统后续设计、编码、集成、测试、维护等阶段的工作.对所开发系统的SA的复杂性进行分析及控制,显得尤为重要^[2-4].

当一个系统由大量相互作用的构件构成,其整

体的复杂性超过了对其构成构件行为的简单叠加,那么,这个系统就是复杂系统. 复杂网络(Complex Networks, CN)作为复杂性科学的一部分, CN 致力于复杂系统非线性和非均衡方面的研究,为系统复杂性研究提供了一个全新的视角. CN 不仅研究系统的各个组成部分的复杂性问题,同时, CN 还研究系统的组成部分之间及系统整体复杂性问题.

鉴于此,结合软件工程中软件体系结构设计的相关思想^[5],本文提出了一种利用 CN 进行 SA 复杂性控制的方法,分别从节点(构件)的度、节点(构件)的度分布、SA 簇系数、SA 平均路径长度、SA 模块度等方面对 SA 的复杂性进行控制,实现从较高的抽象层对软件系统的复杂性进行控制,为后续软件系统的开发提供帮助.

2 复杂网络及无向无权网络

2.1 复杂网络

复杂网络通过图论和统计物理方法对系统的结构、演化机制及规律等进行描述和捕捉. 在复杂网络中,将系统的组成元素抽象为节点,元素之间的关系抽象为边. 通过二元组 $CN = (V, E)$ 来表示一个复杂网络,其中, $V = \{V_1, V_2, \dots, V_n\}$ 为复杂网络中的节点集合, $E = \{E_1, E_2, \dots, E_m\}$ 为复杂网络中的边集合. $N = |V|$ 表示节点总数, $L = |E|$ 表示边总数. 在复杂网络中,如果 (V_i, V_j) 与 (V_j, V_i) 为同一条边,则网络为无向网络,否则为有向网络;如果复杂网络中每条边都赋予了相同的固定的权值,则网络为无权网络,否则为加权网络^[6-7].

2.2 无向无权网络

为了简化研究难度,本文采用无向无权网络对 SA 复杂性进行研究. 下面给出无向无权网络中的相关定义.

定义 1 节点的度 k_i

节点 i 的度 k_i 定义为与该节点 i 直接连接的其他节点的数目. 在 CN 中,节点的平均度 $\langle k \rangle$ 定义为 CN 中所有节点的度的平均值.

定义 2 节点的度分布 $P(k)$

节点的度分布 $P(k)$ 定义为在 CN 中随机选定的节点的度刚好等于 k 的概率,或在 CN 中度为 k 的节点的个数所占节点总数的比例.

定义 3 CN 簇系数 C

CN 中的一个节点 i 有 k_i 条边将它与其他节点相连,这 k_i 条边连接的 k_i 个节点之间最多有 $k_i(k_i - 1)/2$ 条边. 将这 k_i 个节点之间实际存在的边数 E_i 和

总的可能的边数 $k_i(k_i - 1)/2$ 之比定义为节点 i 的簇系数 C_i ,即 $C_i = 2E_i / (k_i(k_i - 1))$. CN 的簇系数 C 则就是网络中所有节点的簇系数的平均值^[7].

$$C_i = \frac{E_i}{\frac{k_i(k_i - 1)}{2}} = \frac{2E_i}{k_i(k_i - 1)} \quad (1)$$

$$C = \frac{1}{N} \sum_{i=1}^N C_i \quad (2)$$

式中, C_i 为节点 i 的簇系数; C 为 CN 簇系数; N 为 CN 中的节点总数. 根据以上分析可知, $C \in [0, 1]$, $C = 0$ 表示网络中所有节点均是孤立的,即不存在任何连接边; $C = 1$ 表示网络是全连通的,任意两个节点之间都直接相连接.

定义 4 CN 平均路径长度 L

CN 中两个节点 i 和 j 之间的路径长度 d_{ij} 定义为连接这两个节点的最短路径上的边数. CN 平均路径长度 L 是指网络中所有节点对之间最短路径的平均值.

$$L = \frac{1}{\frac{1}{2}N(N-1)} \sum_{i \neq j} d_{ij} \quad (3)$$

式中, L 为 CN 平均路径长度; d_{ij} 为节点 i 和 j 之间的最短路径; N 为 CN 中的节点总数.

定义 5 CN 模块度 Q

CN 的模块度用于度量 CN 网络的划分合理性. 根据复杂网络思想,一个 CN 网络的模块度 Q 值越大,则说明该网络划分越合理. 在复杂网络中,这种结构叫社团结构. 即, Q 值越大,社团划分越合理,越是符合“社团内部连接紧密,社团之间连接稀疏”的理论^[8-9].

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j) \quad (4)$$

式中, k_i 和 k_j 为节点 i 和 j 的度值; C_i, C_j 为节点 i 和 j 所属社团; m 为网络总边数, A_{ij} 为节点 i 和 j 之间边的数目. 当 $C_i = C_j$ 时, $\delta(C_i, C_j) = 1$, 否则为 0. Q 值在 0 和 1 之间.

3 软件体系结构复杂性及控制方法

3.1 软件体系结构复杂性

根据 Garlan & Shaw 的定义,软件体系结构是由构件、连接件和约束构成,即, $SA = \{\text{构件(component), 连接件(connector), 约束(constrain)}\}$. 其中,构件是构成软件系统的单元或子程序(如,函数、类、模块等),连接件是构件之间的关联,约束是构件与连接件之间连接的规则.

根据复杂网络 CN 的定义,软件体系结构 SA

也可以通过一个二元组来定义,即, $SA = (C, L)$, 其中, $C = \{C_1, C_2, \dots, C_n\}$ 是构件的集合, $L = \{L_1, L_2, \dots, L_m\}$ 是连接件的集合, $N = |C|$ 为构件总数, $M = |L|$ 为连接件总数。

作为软件系统的高层抽象和系统实现的蓝本, SA 的复杂性问题对系统影响深远。同时,越来越多的研究表明, SA 结构不是规则和随机的,而是具有诸如“无标度”、“小世界”等复杂网络的许多特征,即, SA 结构一般具有较大的簇系数和较短的平均路径,同时, SA 结构中构件之间的连接呈指数分布或幂律分布。

依据无向无权网络中的定义,下面给出软件体系结构复杂性度量属性的相关定义。

定义 6 构件的度 k_i

构件 i 的度 k_i 定义为与该构件 i 直接连接的其他构件的数目。在 SA 中,构件的平均度 $\langle k \rangle$ 定义为 SA 中所有构件的度的平均值。

定义 7 构件的度分布 $P(k)$

构件的度分布 $P(k)$ 定义为在 SA 中随机选定的构件的度刚好等于 k 的概率,或在 SA 中度为 k 的构件的个数所占构件总数的比例。

定义 8 SA 簇系数 C

由定义 3 中 SA 簇系数 C 的定义如下:

$$C_i = \frac{E_i}{\frac{k_i(k_i-1)}{2}} = \frac{2E_i}{k_i(k_i-1)} \quad (5)$$

$$C = \frac{1}{N} \sum_{i=1}^N C_i \quad (6)$$

式中, C_i 为构件 i 的簇系数; C 为 SA 簇系数; N 为 SA 中的构件总数。同理, $C \in [0, 1]$, $C = 0$ 表示 SA 中所有构件均是孤立的; $C = 1$ 表示 SA 是全连通的。

定义 9 SA 平均路径长度 L

由定义 4 中 SA 平均路径长度 L 的定义如下:

$$L = \frac{1}{\frac{1}{2}N(N-1)} \sum_{i \neq j} d_{ij} \quad (7)$$

式中, L 为 SA 平均路径长度; d_{ij} 为构件 i 和 j 之间的最短路径; N 为 SA 中的构件总数。

定义 10 SA 模块度 Q

SA 的模块度用于度量 SA 结构的划分合理性。

由定义 5, SA 模块度 Q 的定义如下:

$$Q = \frac{1}{2m} \sum_{i \neq j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j) \quad (8)$$

式中, k_i 和 k_j 为构件 i 和 j 的度值; C_i 、 C_j 为构件 i 和 j 所属社团; m 为 SA 总边数; A_{ij} 为构件 i 和 j 之间连接

件的数目。当 $C_i = C_j$ 时, $\delta(C_i, C_j) = 1$, 否则为 0。

3.2 SA 复杂性控制方法

在 SA 复杂性度量属性中,构件的度 k_i 数值的大小直接反映了该构件的复杂性,同时也反映了该构件相对于相邻构件的重要性;构件的度分布 $P(k)$ 反映了 SA 结构中每个构件以及全部构件连接的结构特征,展示了 SA 结构的宏观拓扑特性,同时也反映了 SA 结构的局部中心性在全局结构中的分布情况,可以帮助相关人员理解软件系统的结构信息^[10]。

SA 簇系数 C 反映了 SA 结构中构件之间的聚集状况,即 SA 结构的“紧密性”,也即 SA 结构中连接在一起的“集团化”构件各自的近邻构件中有多少共同的近邻构件。SA 簇系数 C 对于揭示软件系统的模块化及模块之间集成特性具有重要意义。

SA 平均路径长度 L 直接反映了软件系统的复杂性和可维护性,如果 SA 平均路径长度 L 过大,则表明该软件系统结构较为松散,构件复用度不高;相反,如果 SA 平均路径长度 L 过小,则表明该软件系统结构过于紧密,构件之间耦合度太高,构件划分不合理,不利于软件系统后续的维护。

SA 模块度 Q 用于反映 SA 结构划分的合理性,如果 SA 模块度 Q 值越大,则划分出结构越好,越符合“社团内部连接紧密,社团之间连接稀疏”的思想。

在软件工程设计中,也提出了许多设计思想,如模块及模块化(Modularization)、内聚(Cohesion)、耦合(Coupling)、包、扇出(Fan Out)、扇入(Fan In)、结构深度、结构宽度等。其中,模块化是把软件系统按照规定原则,划分为一个个较小的、相对独立的但又相互关联的构件;内聚是一个模块内部各个元素彼此之间结合的紧密程度;耦合是模块之间的相对独立性,或模块之间互相连接的紧密程度;包是用来对软件系统组成元素进行分组,并通过包图来描述软件系统、子系统的宏观组成和结构;扇出是一个模块直接调用的模块数目;扇入是能直接调用该模块的模块数目;结构深度是指结构图中控制的层数;结构宽度是指结构图中同一层次上模块总数的最大值。

根据软件工程理论,模块化体现了“分而治之”的设计思想,把复杂问题分解成许多容易解决的小问题,则复杂问题也就容易解决了;模块内部各元素之间的联系越紧密,则该模块的内聚性就越高,相对地,该模块与其他模块之间的耦合就会降低,功能独立性就越强,所以,软件工程强调“高内聚,低耦合”的设计思想。同时,软件工程强调,通过包来进行软件系统的层次的划分,及系统组成元素的分组,有利

于降低软件系统开发的复杂性;深度和宽度反映了软件系统的规模和复杂性;模块的扇出过大,表明该模块比较复杂;模块的扇入表明有多少模块可以直接调用该模块,反映了该模块的复用程度;模块的扇出和扇入会直接影响软件结构的宽度和深度,比如,减小模块的扇出,可能导致结构宽度变小,而深度增加。

结合软件工程思想及 SA 复杂性度量属性,我们提出,在进行 SA 复杂性控制时,应该从如下几个方面进行控制:(1)构件的度控制;(2)构件的度分布控制;(3)SA 簇系数 C 的控制;(4)SA 平均路径长度 L 的控制;(5)SA 模块度 Q 的控制。

3.2.1 构件的度控制

根据 George A. Miller 提出的 7 ± 2 原则,我们认为,SA 结构中每个构件的度值最好不应该超过 9,如果某个构件的度值大于 9,则该构件很复杂,需要对该构件进行更多的主体内容的表达,势必会影响到整个结构的复杂性。所以,对于度值大于 9 的构件,应该对该构件进行拆分,重新划分构件。

3.2.2 构件的度分布控制

研究者们根据大量的研究得出结论,在实际系统的 SA 结构中,度分布服从幂律分布 $P(k) \sim k^{-\gamma}$,即无标度特性,并且 γ 的取值在 $(1.0, 3.0)$ 之间,即,SA 结构的度分布中,绝大多数构件的度值较小,而只有少数构件的度值很高。同时,研究结果表明,SA 结构的度分布 $P(k)$ 幂律分布特征越明显,表明该 SA 结构设计越合理、SA 结构越成熟、SA 结构越有序、SA 结构复杂性越低。我们认为,如果设计的 SA 结构的度分布没有服从幂律分布特征,则该 SA 结构复杂性较高,需要对该 SA 结构进行重新设计,直到该 SA 结构的度分布服从幂律分布为止,并且, γ 的取值一般应该在 $(1.0, 3.0)$ 之间。

3.2.3 SA 簇系数 C 的控制

SA 簇系数 C 反映的是软件系统中构件与构件之间的聚集情况。研究表明,对于构件数量规模为 N 的软件系统,簇系数远远大于 $O(N^{-1})$ 。对于具有“无标度”、“小世界”等复杂网络特征的非规则软件系统和非随机软件系统而言,表明该软件系统具有较高的聚集特性。结合国内外研究者的研究结果,本课题组成员对部分软件系统进行了分析,发现这些软件系统的 SA 簇系数基本为 0.1 或 0.2 左右,最大的为 0.4。鉴于此,我们认为,如果设计的软件系统的 SA 簇系数 C 不为 0.1 或 0.2 左右,则认为所设计的 SA 结构不够“紧密”,SA 结构的“集团化”特

性不够明显,需要进行修改该软件系统的 SA 结构。

3.2.4 SA 平均路径长度 L 的控制

国内外研究者们通过对大量开源软件系统进行了研究,这些软件包括系统软件、开发工具、网络互联软件、多媒体软件等类型,开发语言包括 C++、C 语言、Java 等,结果表明,这些软件系统的平均路径长度大概为 5 左右。本课题组成员对部分应用软件系统进行了类似的分析,发现这些应用软件系统的 SA 结构的平均路径长度大概为 6 左右。鉴于此,结合复杂网络理论,我们认为,所设计软件系统的 SA 平均路径长度 L 不能太小(比如 1 或 2),太小意味着 SA 结构设计不合理,SA 结构的复杂性较高,系统各个构件之间耦合度过高,不利于后续的软件维护;SA 平均路径长度 L 也不能太大,太大意味着 SA 结构组织松散,构件复用程度不高。所以,一般情况下,一个好的软件系统的 SA 结构的平均路径长度 L 最好为 5 或 6 左右(特殊软件系统除外)。

3.2.5 SA 模块度 Q 的控制

SA 模块度是对 SA 结构划分的优劣进行评价的指标,SA 模块度 Q 值越大,表明软件系统的 SA 结构具有较好的网络结构划分,相反,如果 SA 模块度 Q 值越小,表明软件系统的 SA 结构划分越模糊。 Q 值是介于 $0 \sim 1$ 之间的值,如果 Q 值大于 0,则表明 SA 结构的社团特征开始出现,如果 Q 值大于 0.3,表明 SA 结构具有明显的社团特征。所以,国内外研究人员一般把 $Q=0.3$ 作为网络具有明显社团结构的下界。研究表明, Q 值通常介于 $(0.3, 0.7)$ 之间^[5]。我们认为,在较好的控制了 SA 构件的度、构件的度分布、SA 簇系数 C 、SA 平均路径长度 L 的前提下,应该尽量使设计的软件系统的 SA 结构的模块度 Q 值大一些,这样,就能较好的满足软件工程设计思想,并尽可能的降低 SA 复杂性。

4 实例分析

本文以新疆生产建设兵团空间信息服务系统中的基于遥感 RS 和地理信息系统 GIS 洪灾检测子系统为例进行系统的复杂性验证。该子系统是在进行洪灾区域分析的基础上,搭建洪灾快速识别模型,达到对兵团不同地块淹没情况实施监控和分析、为各职能部门提供参考的目的。根据前期的分析及设计,该子系统的体系结构图如图 1 所示。

为简化研究难度,将图 1 转换为如图 2 所示的简化的系统 SA 图(无向无权网络图)。在图 2 中,构件集合 $C = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10},$

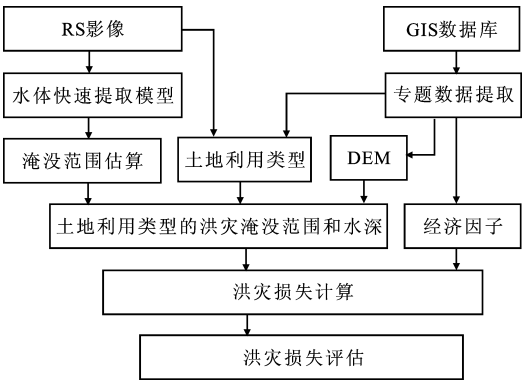


图1 系统的体系结构

$C_{11}\}$, 边集 $W = \{W_{12}, W_{16}, W_{23}, W_{38}, W_{45}, W_{56}, W_{57}, W_{59}, W_{68}, W_{78}, W_{8-10}, W_{9-10}, W_{10-11}\}$.

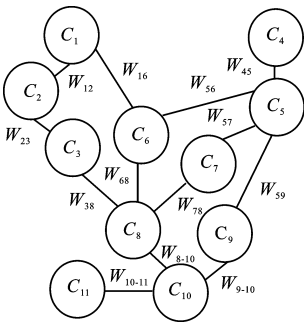


图2 简化的系统软件体系结构图

根据图2 计算构件 $C_1 \sim C_{11}$ 的度值如表1 所示.

表1 构件的度值

构件	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
总强度	2	2	2	1	4	3	2	4	2	3	1

根据定义7,结合表1 构件的度值,本系统的构件的度分布 $P(k)$ 服从幂律分布. 根据定义9,本系统的SA 结构的平均路径长度 L 为1.288. 同理,根据定义8 及定义10,可计算本系统的SA 簇系数 C 和SA 模块度 Q .

通过利用构件的度、构件的度分布、SA 簇系数 C 、SA 平均路径长度 L 、SA 模块度 Q 的计算数值,达到度量及控制所设计系统的SA 复杂性的目的,如果计算数值跟期望的值相差过大,则需要重新修改和完善软件系统的SA 结构,然后重新计算SA 复杂性度量的属性值,直到达到系统设计要求为止. 当然,有些系统在设计其SA 结构时,不一定每个SA 复杂性度量属性值都可以满足要求,这时,需要重新衡量所开发系统的特点,采取比如折衷的方案,满足某些SA 复杂性度量属性值即可.

5 结束语

本文基于复杂网络理论中无向无权网络的复杂性度量及控制方法,进行软件体系结构SA 复杂性的度量及控制. 结果表明,该度量及控制方法在复杂网络严格的形式化基础之上,可以较好地对所开发系统的SA 复杂性的属性进行度量并进行控制,度量及控制过程较简便,度量值可信,控制方法可行. 度量结果及具体的控制方法为相关人员从高层把握系统复杂性及控制系统SA 复杂性提供了参考,同时度量结果及控制方法也为系统后续的设计、编码、集成、测试、维护等阶段的工作提供了支持和参考.

参考文献:

[1] 梅宏,黄翌,张路,等. ABC:一种全生命周期软件体系结构建模方法[J]. 中国科学:信息科学,2014,44(5): 564-587.

[2] 秦怀斌,李道亮,郭理,等. 基于复杂网络的软件体系结构复杂性度量方法[J]. 微电子学与计算机,2013,30(2):5-8.

[3] 汪北阳,吕金虎. 基于软件系统的软件网络结点影响分析[J]. 软件学报,2013,24(12):2814-2829.

[4] 张浩华. 复杂网络视域下的大规模软件静态结构度量及进化分析[D]. 沈阳:东北大学,2013.

[5] 钱乐秋,赵文耘,牛军钰. 软件工程[M]. 2 版. 北京:清华大学出版社,2015.

[6] 张英杰,龚中汉,陈乾坤. 基于免疫离散差分进化算法的复杂网络社区发现[J]. 自动化学报,2015,41(4): 749-757.

[7] Song Y, Liu J, Yu Z, et al. Multifractal analysis of weighted networks by a modified sandbox algorithm [J]. Scientific Reports, 2015(5):17628.

[8] New Man M E J. Communities, modules and large-scale structure in networks[R]. Canada :Nature Physics, 2012,8(1):25-31.

[9] 贾琨,胡晓峰,贺筱媛. 基于节点动态连接度的网络社团划分算法[J]. 复杂系统与复杂性科学,2016,13(4): 56-61.

[10] 高洋,彭勇,谢丰,等. 基于复杂网络的软件结构度量[J]. 清华大学学报:自然科学版,2012,52(10): 1474-1481.

作者简介:

秦怀斌 男,(1980-),硕士,副教授. 研究方向为软件工程、复杂网络. E-mail:qhb_inf@shzu.edu.cn.

郑瑶 女,(1976-),硕士,副教授. 研究方向为软件工程.

梁斌 女,(1959-),硕士,教授. 研究方向为软件工程.