

基于构件使用依赖关系的构件复用可信度计算方法

王燕玲^{1,2*}, 曾国荪¹

(1. 同济大学 计算机科学与技术系, 上海 201804; 2. 高效能服务器和存储技术国家重点实验室, 济南 250101)

(* 通信作者电子邮箱 wyl030703@126.com)

摘要: 在网络构件库中构件数量持续增长的情况下, 为解决用户无法从质量参差不齐的海量构件中选取优质构件的问题, 提出了一种基于构件使用依赖关系的复用可信度计算方法。该方法将构件库作为证据库, 首先, 对证据库中的构件依赖信息进行采集整理; 接着, 定义每一个构件的基本信任函数, 在此基础上根据构件依赖信息不同来源为每一条证据设置不同的可信权重值; 然后, 通过特定的转换算法将由此获得的结果生成构件最终的可信度。实例分析中, 利用该方法评估构件取得的结果值与预想中的一致, 且与参考构件的内部与外部质量模型所得出的结论相符, 但该方法大大降低了构件可信评估的工作量, 提高了评估效率。结果表明该方法能够客观反映构件的可信性, 可作为构件库中构件检索的一种可信度量机制, 帮助实现构件的优质检索和复用。

关键词: 软件构件; 依赖关系; 构件描述; 可信证据; 可信度

中图分类号: TP311.5; TP391 **文献标志码:** A

Trust evaluation method for component reuse based on component use dependency relation

WANG Yanling^{1,2*}, ZENG Guosun¹

(1. Department of Computer Science and Technology, Tongji University, Shanghai 201804, China;

2. State Key Laboratory of High-end Server and Storage Technology, Jinan Shandong 250101, China)

Abstract: The number of components is continuously growing in the network component library, it is hard for users to select high-quality components from the massive uneven-quality components. In order to solve the problem, a reuse trust evaluation method based on component use dependency relations was proposed, in which component base was used as an evidence base. Firstly, component dependency relations were collected from evidence base. Secondly, the basic trust function was defined for each component, and the different believable weight value was set up for each evidence according to the different sources of component dependency relations on the above basis. Finally, the final trust value of component was generated by a specific conversion algorithm with the obtained results. In instance analysis, the component evaluation result of the proposed method was consistent with the expectation and the conclusion which was gotten by the internal and external quality model of the reference components. However, the proposed method greatly reduced the workload of component's credible evaluation and improved the evaluation efficiency. The results of analysis show that the proposed method can objectively reflect the credibility of components, and can be used as a trusted measurement mechanism of component retrieval in the component library, which helps to realize the high quality retrieval and reuse of components.

Key words: software component; dependency relation; component description; credible evidence; trust value

0 引言

随着基于构件的软件工程(Component-Based Software Engineering, CBSE)的迅速发展, 构件开发技术已经相对成熟, 网络中已出现了众多可复用构件。这些构件可以提高软件的开发效率, 节约开发的时间, 为软件工程师带来诸多便利, 但同时也带来了新的挑战。在多个软件构件库中, 构件数量繁多, 良莠不分, 构件复用者往往需要花费大量的时间多次查找与测试, 才能获得自己所需的、满意的、高质量的构件。因此, 进行软件构件质量可信评估研究, 为软件构件提供一种柔性的、科学的构件度量机制, 实现构件的优质检索, 十分必要。

目前, 已有一些学者从多个方面对软件构件的可信性评估进行了研究。比如, ISO9126 的软件质量模型^[1]的提出, 其贡献在于将软件质量特性分为外部特性和内部特性, 考虑了软件产品不同生命周期阶段的不同形态问题。毛国蓓等^[2]在 ISO9126 的内外部属性模型基础上, 给出了软件构件的质量模型, 同时也说明了构件质量思路与质量公式, 但并未给出度量的量化思路。Meyer^[3]提出的可信构件的 ABCDE 模型, 从 5 个角度正交划分可信构件应具备的属性, 但也没有明确给出现有模型与实际软件质量特性、子特性之间的相互关系, 缺乏在此基础上进行有效定性定量评估的方法。郭树行等^[4]给出了可信构件的度量模型, 为可信构件的研究提供了定性的参考。唐莹等^[5]、汪海涛等^[6]基于 ISO/IEC25010 软

收稿日期: 2015-06-23; 修回日期: 2015-08-19。

基金项目: 国家 863 计划项目(2009AA012201); 国家自然科学基金资助项目(61272107, 61202173, 61103068); 上海市优秀学科带头人计划项目(10XD1404400); 华为创新研究计划项目(IRP-2013-12-03); 高效能服务器和存储技术国家重点实验室开放基金资助项目(2014HSSA10)。

作者简介: 王燕玲(1990-), 女, 福建泉州人, 硕士研究生, 主要研究方向: 软件工程、软件构件; 曾国荪(1964-), 男, 江西新余人, 教授, 博士生导师, 博士, 主要研究方向: 并行计算、可信软件设计、信息安全。

件质量模型,分别提出了基于某个特定领域的可信构件度量模型、基于模糊物元评价法的构件质量度量模型,为构件研究提供了可度量的属性指标。此外,还有一些学者,如汪永好等^[7]考虑不同用户对构件有不同的可信需求,提出了一种可扩展的构件信任评估模型。再如徐锋等^[8]、殷脂等^[9]通过设计合理的协同推荐机制来评估并选取可信构件,这些方法主要是通过收集用户使用评价来权衡构件可信度。然而,由于网络环境中推荐信息来源的可靠性无法保证,因此这类方法仍然存在问题。Ishihara 等^[10]认为被复用过的构件说明曾经被需要并被使用,而这样的构件也将满足将来的构件需求。构件被使用过说明该构件是可用的,同时一个构件被使用越多次,说明它越值得被相信。而构件依赖关系也是一种使用关系,这种使用关系除客观存在外,其相关使用记录还可以在构件库中获得。

综上所述,构件可信性的度量还缺乏统一的度量模型和量化方法。本文对构件可信性进行研究,通过研究构件库中构件的依赖关系挖掘复用过的构件,给予这些构件更高的复用可信度,由此提出一种基于构件使用依赖关系的复用可信度计算方法,以此来确保所搜构件的质量是合理可行的。

1 软件构件的定义和描述

1.1 构件及构件可信度的定义

目前对软件构件并没有一个公认的、统一的定义。广义上讲,构件泛指可以被明确标识的软件产品,它可以是需求分析、设计、代码、测试用例、文档或软件开发过程中的其他产品。狭义上讲,软件构件是指可以复用的、提供明确接口完成特定功能的程序代码模块,如源代码、二进制代码、可执行文件等。本文将软件构件作为研究对象,其定义参考计算机百科全书的解释说明,具体定义如下:

定义 1 软件构件是指具有相对独立功能、可以明确辨识、接口由契约指定、可独立部署、多由第三方提供的可组装的、不透明的软件实体。

简单来说,构件是一个可发布、可执行、有良好接口的软件模块,可在不同项目的不同上下文环境中被复用。复用是构件开发和推广的根本目的。

早在 1968 年的北大西洋公约组织(North Atlantic Treaty Organization, NATO)软件工程会议上,复用这一概念就由 McIlroy 在其论文“大量生产的软件构件”中提出。作为软件复用的一种有效手段,构件技术成功地将系统开发的重点从程序设计转换为对已有构件的选取、组装和部署^[11]。在构件模型的支持下,通过对已有构件的复用,软件开发者可以“即插即用”地快速构造应用系统。这样做不仅可以缩短开发周期,降低软件开发成本,提高工作效率,而且可以产生更加规范、更加可靠的应用软件。但是,实现这一切的前提是,复用的构件必须是有用的、高质量的、可信的,最好是完全正确的。

每一个构件都具有一定程度的信任度。所谓构件复用可信度,一般来说是用户根据经验,对构件的相信程度,是一个相对主观的概念,难以进行客观的度量。此外,构件可分为原子构件和复合构件。原子构件是不可再分的构件,而复合构件是由其他复合构件或原子构件组合而成,这种组合关系即称为构件依赖关系。如果一个构件被多个其他构件依赖,即该构件被多次使用,那么,这个构件的复用可信度是相对较高

的。从以上两方面考虑,本文给出如下构件信任度的定义。

定义 2 构件的可信度也称为构件的复用可信度,是指构件多次被使用和依赖的程度,即用户所能获得的构件是否真实可信(Trust)的综合期望值。在一个包含 n 个构件的构件库中,对于构件 i 的可信度用 $T(Component_i)$ 表示, $1 \leq i \leq n$ ($n \in \mathbf{N}_+$), $T \in [0, 1]$, $T = 0$ 为不可信, $T = 1$ 为完全可信。

在构件的可信度计算方面,本文将在构件可信依赖关系分析的基础上,通过考虑每一个构件对其他构件的依赖关系,以及自身可信度对其他依赖构件的影响,综合得出构件的可信度。

1.2 构件描述

构件作为封装的、规范的、可重用的软件模块,其编码实现对用户是透明的。为了使构件能够更加规范化、标准化、满足构件可搜索、可复用、方便管理等要求,必须对构件进行描述。以构件的具体描述规约告诉用户构件提供的功能、性能、部署和使用信息等,用户才能通过构件描述规约了解构件,并通过构件接口获得服务。换言之,构件描述是对构件进行组织、存储、检索、浏览、调用的基础。

现有的构件描述一般提供四部分信息,分别为基本信息、应用环境、开发环境和详细描述。图 1 表示了构件描述所对应的树形结构:其基本信息描述包含了构件编号、构件名称、构件作者、构件版本等;应用环境描述包括构件功能、应用领域、应用操作系统、构件形态;开发环境描述则由构件的开发语言、开发日期、开发平台等定义;详细描述信息包含了接口信息和关联信息。其中接口信息由接口名称、参数名、参数类型、返回值以及接口应表现出的行为规定。关联信息刻画了构件之间的依赖关系,如继承关系、调用关系、朋友关系等。构件的详细描述信息帮助用户在获取构件后对构件的组装等操作进行了解并使用构件。

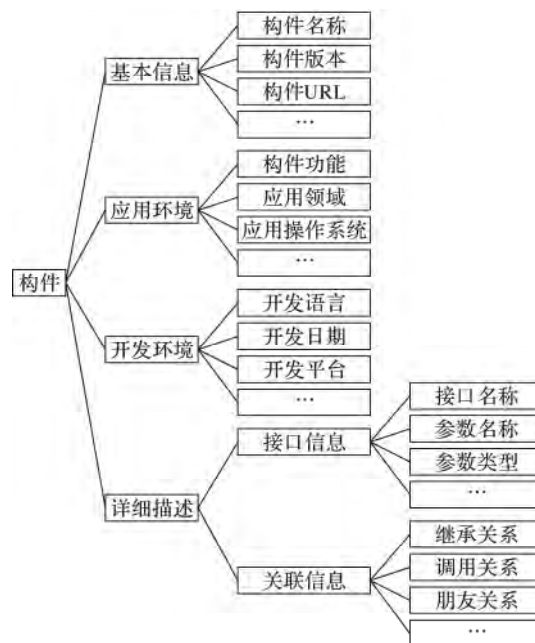


图 1 构件描述对应的树形结构

为了更好地组织管理构件,网络构件库通常用标记语言(eXtensible Markup Language, XML)来描述构件。这样每个构件的描述就成为一个 XML 文档。因此,构件的搜索、访问等操作就可以转换为对 XML 文档的读写和检索处理。以

下为构件描述 XML 文档的具体文档类型定义片段:

```
<? xml version = "1.0" encoding = "GB2312"? >
<! ELEMENT component ( basicFacets , appEnvironment ,
    devEnvironment , detailFacets ) >
<! ELEMENT basicFacets( ID , Name , Version , URL ) >
<! ELEMENT appEnvironment ( AppDomain , Appfunction ,
    AppOS ) >
<! ELEMENT devEnvironment ( Language , PlatForm , Date ) >
<! ELEMENT detailFacets ( Interfaces , Related ) >
<! ELEMENT interfaces( interfaceName , retvalue , pramas ) >
<! ELEMENT Related ( inherit , reference , friends ) >
<! ELEMENT inherit ( i_name , i_number ) >
<! ELEMENT reference ( r_name , r_number ) >
<! ELEMENT i_name ( #PCDATA ) >
...
```

由于本文利用构件依赖关系来计算构件的可信度,因此对关联信息作进一步说明。在关联信息中,继承关系指类体系中的继承性。调用关系指一个构件通过调用另一个构件的接口函数从而完成自身功能的关联性。朋友关系指构件间输入输出可匹配的关系,例如,一个构件的输出和另一个的输入类型和个数完全匹配,从形式上两者可以进行合成。因此,这部分信息能够帮助构件在检索后进行组装合成。

在本文中,依赖关系实际上是指一种已完成的使用关系,它表明一个事物规格说明的变化可能影响到使用它的另一个事物。因此,本文所述的构件依赖关系主要是指关联信息描述中的继承关系和引用关系,并且不对这两部分信息加以区分。为了进一步说明构件的依赖关系,以构件依赖结构图进行说明,如图2所示。

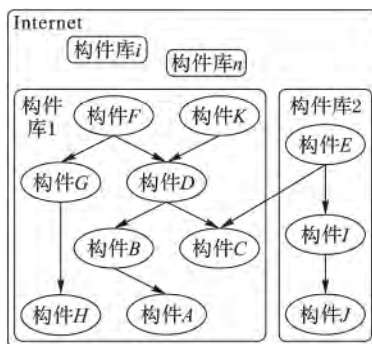


图2 软件构件依赖结构

在图2中,对于构件库1和构件库2而言,构件D依赖于构件B和C,即构件D使用了构件B和C;反过来说,构件B和C被构件D所依赖。同理,构件E也依赖于构件C和I;而构件F依赖于构件G和D。构件之间的依赖关系对于构件库而言是非常重要的信息和可信证据^[12],在下文中将进一步阐述。

2 网络构件库及构件可信依赖关系的获取

2.1 网络构件库

散落在多处的单个构件不利于软件工程项目查找和复用,因此构件常常被存放于构件库中,以便管理和利用。随着构件技术的发展,Internet上已出现了许多专门的构件库组织,例如著名的ComponentSource构件库(www.componentsource.com)、IBM Alphaworks构件库(alphaworks.com),以及国内的青鸟构件库、上海构件库等。这些构件库作为一种支持软件复用的基础设施,它提供对软件构件进行

描述、存储和检索等功能,具体的系统组成结构如图3所示。

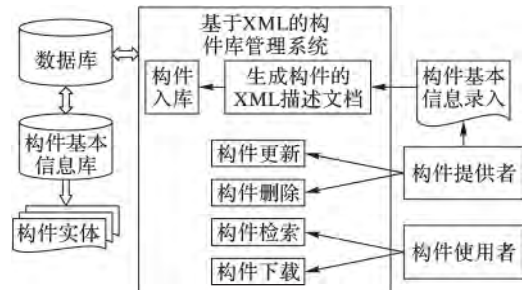


图3 构件库管理系统框架

值得说明的是,在构件库系统中,构件被分为构件实体和构件描述信息两部分。构件实体可能比较大,存储在文件系统中,通过指明文件路径对其进行访问。构件描述信息即2.2节中介绍的构件描述规约(XML文档),系统将采用数据库(SQL Server2005)来存储这些XML数据,可以使用XQuery对其进行查询,可以使用XML数据修改语言(Data Manipulation Language, DML)在元素级修改XML文档。因此系统可以很容易获得与特定构件匹配的构件描述信息,为基于构件的软件工程提供了支持。在本文中,构件库将作为构件的证据库来开展构件的可信质量分析研究。

2.2 证据库与可信证据

2.2.1 证据库思想

构件的可信质量分析需要依据所获得的可信证据。证据是认定事实的材料,人工智能领域相关理论认为,证据是对命题信任情况的一种反映,通过对证据的分析可以得出对一个命题的看法^[13]。本文中,构件可信证据定义如下。

定义3 可信证据是指能够经过查证确定属实,并用来证明构件可信度情况的一切事实。其存在形式可以是文档信息、关系信息、命题、事实、度量值,以及其他信息等。

在软件构件库中,每个构件都有自己的功能、地位,且它们之间存在着依赖关系。当构件A使用了构件B,即构件A依赖于构件B,而构件B被构件A所依赖。以单个基本的依赖为单元,当构件依赖和被依赖数量较多时,会形成错综复杂的构件依赖关系,而这种关系都以两个主体间的使用为基本单元,且客观存在。因此,对构件的依赖关系进行采集和分析,可作为构件可信证据依据,继而通过特定的转换算法生成可直接用于可信性评估的评估证据,最后再基于评估证据得到构件最终的可信度。

可见,在构件库中存在丰富的可信证据。由此,一方面库功能可以获取可信证据计算构件可信度;另一方面,可以为构件库中的构件检索提供一种可信度量机制,实现构件的优质检索和复用。

2.2.2 证据基本信任分配函数

在确定了证据库,即证据来源后,可以运用证据理论来评估构件复用可信度。根据证据理论,需要为证据建立一个信任程度的初始分配,确定证据对构件是可信的这个命题本身的支持程度。在证据理论中,一般用集合来表示命题,本文也采用这种方法,对一个构件的支持框架以及基本信任分配函数进行如下定义。

定义4 构件支持框架。假设一个构件A依赖于n个其他构件,对于该构件所支持的所有构件的完备集合用 Θ 来表

示, 称该完备集合 Θ 为构件 A 的支持框架, 可表示为 $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ 。其中: $\theta_i \in \Theta$ 表示构件所依赖的其中一个构件, 亦为支持框架 Θ 中的一个元素。

定义 5 基本信任分配函数。设 Θ 为构件 A 的支持框架, 基本信任分配函数 m 是一个从集合 Θ 到 $[0, 1]$ 的映射, X 表示支持框架 Θ 的任一元素, 记作 $X \in \Theta$, 且满足 $\sum_{X \in \Theta} m(X) = 1$ 。式中: $m(X)$ 称为构件 X 的基本信任分配函数, 它表示构件 A 对构件 X 的信任程度。

本文不妨认为一个构件对它所依赖的所有构件的支持程度应该是等概率的, 即 $m(X) = 1/N(A)$, $N(A)$ 表示构件 A 所依赖的构件个数。

2.2.3 多源证据分析

构件可信证据是构件可信性评估的前提和基础。构件库作为证据库为构件可信度计算提供了有效的数据。然而, 一个构件可能被多个构件库中的构件所调用。事实上, 当构件开发者决定是否让所开发的构件依赖于另一个构件时, 他仅关心被依赖的构件是否满足自己的功能需求, 而不在乎该构件来源于哪个构件库, 而最终构件发布于哪个构件库也仅取决于构件开发者自己的选择。换言之, 一个构件的依赖关系并不局限于一个构件库中。因此, 一个构件的可信证据可能来源于多个相互独立的构件库, 本文将这类证据称为多源证据。显然, 多源证据包含两类证据: 同源 (Domestic) 证据和异源 (External) 证据。对其进行形式化描述如下:

EvidenceModel = { DomesticEvidence, ExternalEvidence }

DomesticEvidence = { dData1, dData2, dData3, ... }

ExternalEvidence = { eData1, eData2, eData3, ... }

同源证据指来源于要计算可信度的构件所在的构件库的证据, 例如 dData_{*i*} 即表示该构件库中第 *i* 条证据对应的证据数据。异源证据指来源于除要计算可信度的构件所在的构件库之外的其他构件库的证据, 例如 eData_{*i*} 表示第 *i* 条异源证据对应的证据数据。

一个构件的可信度取决于这两部分的证据数据。然而证据本身也存在着可信问题, 需要为不同的证据赋不同的可信权重值 $\varepsilon_1, \varepsilon_2$ 。类比于人类社会网络, 一个人相信另一个人的程度往往取决于这两个人的亲疏关系。即当陌生人与亲人同时为某件事情提供说辞时, 更愿意相信自己的亲人。因此, 同源证据的可信权值应该更大一些, 即 $\varepsilon_1 > \varepsilon_2$ 。

由此可见, 一个构件对另一个构件的真正支持程度还取决于两个构件是否存在于同一个构件库中, 记一个构件对另一个构件的最终支持程度为 h , 结合 2.2.2 节中证据基本信任分配函数的定义:

$$h = \begin{cases} \varepsilon_1 m(X), & \text{Domestic} \\ \varepsilon_2 m(X), & \text{External} \end{cases}$$

2.3 构件库中可信使用依赖关系的获取

针对所有的构件库, 提取和分析所有构件的描述信息, 形成目标构件集合 $L_c = (C_1, C_2, \dots, C_n)$, 以 L_c 作为输入集, 分析所有构件的依赖关系, 表示成构件可信依赖矩阵的形式, 步骤如下:

1) 通过 XML 解析技术, 找出构件的依赖描述信息, 继而构建构件依赖图, 如图 2 所示。

2) 根据构件依赖图以及 2.2.2 节中证据基本信任分配函

数的定义, 构建构件依赖矩阵 A 。当构件 i 与构件 j 存在依赖关系时 $A(i, j) = m(j) = 1/N(i)$; 否则 $A(i, j) = 0$ 。

3) 根据构件依赖矩阵 A 以及 2.2.3 节中多源证据的不同权重值分配, 进一步构建构件可信依赖矩阵 B 。当 $A(i, j) \neq 0$, $B(i, j) = h$; 否则 $B(i, j) = 0$ 。

3 构件的可信度计算

3.1 构件可信判断的直观思想

在构件库中, 构件之间的使用依赖关系构成了一个生态系统, 每个构件都有自己的职能、地位等。那么如何将这些依赖关系, 即 2.2 节介绍的可信度量证据进行整合呢? 构件的使用依赖关系可被理解成一种推荐关系, 即当一个构件 A 依赖于另一个构件 B 时, 就表示构件 A 给构件 B 投了一票。一个构件被其他构件依赖得越多, 它得到的票数就越多, 相应地, 可信级别就越高。同时, 这种推荐方式应具有传递作用, 一个构件被级别越高的构件依赖, 它的可信性也应有所增加。该思想类似于搜索引擎的 PageRank 思想, 但在构件依赖系统中, 一个构件 A 依赖于另一个构件 B , 说明构件 B 在构件 A 之前就存在, 也就是不可能存在构件 B 依赖于构件 A 的情况, 这样就使得整个计算过程控制在不大于构件总数 n 的步骤数内。

此外, 构件的依赖关系作为构件可信度量的证据, 每一个证据的可信度并不相同, 取决于依赖构件与被依赖构件是否在同个构件库中。当构件 A 与构件 B 存在于同个构件库时, 它给 B 投的票的重要性将高于当构件 A 与构件 B 不在同个构件库时投的票的重要性。

根据以上现象和事实, 本文对构件可信度的判断提出以下规则。

规则 1 构件库中的一个构件被越多的构件所依赖, 它的可信度越高。

规则 2 构件库中的一个构件被可信度越高的构件所依赖, 它的可信度越高。

规则 3 构件库中的一个构件被与自己同个构件库中的构件所依赖而得到的可信度, 比被与自己不在同个构件库而可信度与前一个依赖构件相同的构件所依赖而得到的可信度高。

由上述规则可知, 一个构件的可信级别由依赖它的构件数、所依赖构件的可信级别, 以及依赖它的构件来源三者共同决定。

3.2 构件可信度计算

记构件库中构件 C_i 的可信度为 $T(C_i)$, 那么根据 3.1 节中构件可信度的判断规则, 其相应的计算公式如下:

$$T(\text{Component}_i) = s + \varepsilon_1 \sum_{\text{Component}_j \in \Omega} \frac{T(\text{Component}_j)}{N(\text{Component}_j)} + \varepsilon_2 \sum_{\text{Component}_k \in \Psi} \frac{T(\text{Component}_k)}{N(\text{Component}_k)} \quad (1)$$

其中: Ω 表示依赖于构件 Component_i 且与 Component_i 在同个构件库的构件集; Ψ 表示依赖于构件 Component_i 且与 Component_i 不在同个构件库的构件集; $T(\text{Component}_j)$ 表示第 j 个依赖于 Component_i 的构件的可信度。式 (1) 是个递归的定义, 因而需要迭代计算。特别地, 构件库中存在一些构件, 这类构件并不被任何构件所依赖, 这样它的可信度为 0, 它能传递给它所依赖的构件的可信度也将为 0。为了处理这种情况,

式(1)中增加了一个阻尼系数 s , 这里可将 s 理解为用户相信一个未知构件可信的概率, 且 $s + \varepsilon_1 + \varepsilon_2 = 1$ 。本文将所有构件可信度初始值设置为 1, 然后通过迭代计算可信度:

$$T_{\omega}(Component_i) = s + \varepsilon_1 \sum_{Component_j \in \Omega} \frac{T_{\omega-1}(Component_j)}{N(Component_j)} + \varepsilon_2 \sum_{Component_k \in \Psi} \frac{T_{\omega-1}(Component_k)}{N(Component_k)}; \omega = 1, 2, \dots \quad (2)$$

假设记 $D_{\omega}^T = [T_{\omega}(Component_1), T_{\omega}(Component_2), \dots, T_{\omega}(Component_n)]$ 则式(2)可以改写成:

$$D_{\omega}^T = [s \ s \ \dots \ s] + D_{\omega-1}^T B \quad (3)$$

其中: B 是 2.3 节中得到的构件可信依赖矩阵。如果一个构件库中有 n 个构件, 那么通过 $k(k \leq n)$ 次循环之后, 所有构件的可信度将达到一个稳定状态, 而不再变化。之后, 对所有的构件可信度进行归一化处理, 归一化公式如下:

$$T(Component_i) = \frac{T_k(Component_i)}{\sum_{Component_j \in Base} T_k(Component_j)} \quad (4)$$

其中 $Base$ 代表所有的构件库的所有构件集合。

3.3 构件可信度的计算算法

根据 3.1 节中的经验规则, 以及 3.2 节中的可信度计算方法, 可以知道 $T(Component_i)$ 的计算过程是一个不断的迭代计算过程, 直到构件库中构件的 $T(Component_i)$ 值达到一个稳定值。构件可信度 $T(Component_i)$ 计算算法伪代码描述如下。

```

算法 1 构件复用可信度计算 ComponentTrustDegree()。
输入 所有构件库 Base 中的所有构件  $C_i$ ;
输出 所有构件的可信度  $T(C_i)$ 。
for all component  $C_i \in$  in Base do
    // 得到构件  $C_i$  的所有依赖构件
     $R_i \leftarrow \text{find\_components\_rel}(C_i)$ ;
    // 利用 2.2.2 节中的方法构造构件依赖矩阵
    for all  $C_j \in R_i$ 
         $A_{(i,j)} \leftarrow \text{build\_matrix}(C_i)$ ;
        // 有依赖关系的两个构件在同一个构件库
        if  $(C_i, C_j) \in \text{DomesticEvidence}$ 
            // 利用 2.2.3 节中的方法构造可信依赖矩阵
             $B(i,j) \leftarrow \varepsilon_1 A_{(i,j)}$ ;
            // 有依赖关系的两个构件不在同一个构件库中
        else
             $B(i,j) \leftarrow \varepsilon_2 A_{(i,j)}$ ;
    // 设定构件  $T(C_i)$  和  $s$  的初始值
     $T_0(C_i) \leftarrow \text{initialValue}_t$ ;  $s \leftarrow \text{initialValue}_s$ ;
     $D_0^T \leftarrow (T_0(C_i))$ ;
    // 根据式(3) 循环计算构件的  $T(C)$ 
loop:
     $D_j^T \leftarrow \text{component\_trust\_degree}(D_{j-1}^T)$ ;
    // 计算两次迭代之间的每个构件  $T(C)$  值之差的和
     $d \leftarrow \|D_j^T - D_{j-1}^T\|$ ;
    while( $d = 0$ )
        // 将稳定值归一化得到构件的可信度
         $T(C_i) \leftarrow \text{normalize}(D_{j-1}^T(i))$ ;
end

```

4 实例分析

为了验证本文提出方法的有效性, 本章通过一系列实验对本文提出的构件可信度计算方法进行测试。首先以图 2 构

件依赖结构图中的构件为例, 对其进行可信度计算并分析。在此基础上, 将该构件可信度计算方法运用于 ComponentSource 构件库中的特定构件质量评估。

4.1 构件依赖结构图中构件可信度分析

根据式(1)对构件复用可信度评估方法的定义, 本节对图 2 构件依赖结构图中的构件进行了可信评估。分析计算期间 $s, \varepsilon_1, \varepsilon_2$ 分别取值 0.2 0.5 0.3, 计算结果如表 1 所示。

表 1 构件库中不同构件的可信度评估结果

序号	构件名	构件库 1 中被依赖数	构件库 2 中被依赖数	构件可信稳定值	可信度
1	A	1	0	0.343 75	0.113
2	B	1	0	0.287 50	0.094
3	C	1	1	0.317 50	0.104
4	D	2	0	0.350 00	0.115
5	E	0	0	0.200 00	0.066
6	F	0	0	0.200 00	0.066
7	G	1	0	0.250 00	0.082
8	H	1	0	0.325 00	0.107
9	I	0	1	0.250 00	0.082
10	J	0	1	0.325 00	0.107
11	K	0	0	0.200 00	0.066

在开始计算之前, 由图 2 可以推出构件 E、F、K 的可信度将低于其他所有构件, G、I 次之, 而构件 D、H、J 的可信度将相对较高。分析表 1 中的数据可看出, 虽然构件在构件库中被依赖数相差不大, 但计算出来的结果却相差较大, 且完全符合了由图 2 得出的结论, 因此该计算方法可以体现不同构件调用质量上的差异。以构件 A 与构件 I 为例, 它们都只被依赖一次, 但构件 A 的可信度高于构件 I, 原因是构件 I 的依赖关系为异源证据, 其可信权重低于同源证据。

4.2 构件库中构件可信度实例分析

综合考察了多个现有的网络构件库之后, 本文选取了 ComponentSource 构件库中的构件进行实验, 从中随机抽取 43 个构件, 分别对其可信度进行评估计算。实验中 $s, \varepsilon_1, \varepsilon_2$ 分别取值 0.2 0.5 0.3, 且令被依赖的构件可信度初始值皆为 1。实验结果如表 2 所示。

表 2 ComponentSource 构件库中不同构件的可信度评估结果

序号	构件名	构件库中被依赖数		可信度
		同源	异源	
1	vsSPELL	2	0	0.036
2	fully-managed-code component	2	1	0.045
3	EasyMail. NET Verify	0	2	0.024
4	Microsoft Dynamics CRM Source Component	3	1	0.061
5	E-Commerce components	2	0	0.036
6	AFD BankFinder	1	0	0.021
7	Extractor	0	2	0.024
8	Universal Data Access Components (UniDAC)	1	2	0.039
9	dBarcode. NET Dual Universal	1	1	0.030
10	Aspose. Cells for JasperReports	0	1	0.015
...

在本实验中, 所有的依赖构件可信度初值都设置为 1, 但在实际应用中, 每一个构件的可信度都不同, 能给予被依赖构

件的可信支持度也不同,因此最终算出来的结果将与本实验中的结果有所差别。但从表2中仍然可以看出,来自于同一个构件库(ComponentSource)的构件由于其依赖关系不同,可信度也不同,被越多可信构件依赖的构件复用可信度也越高。因此,这种对不同构件复用可信度的评估方法能够在一定程度上体现不同构件质量上的差异,符合人工选择的常理和规则。

此外,为了进一步说明该方法的合理性,本文参考构件的内部与外部质量模型,利用贺久松^[14]对构件可信性的评估方法对所选构件进行了再次评估,并将两次的评估结果进行对比,对比结果如图4所示。

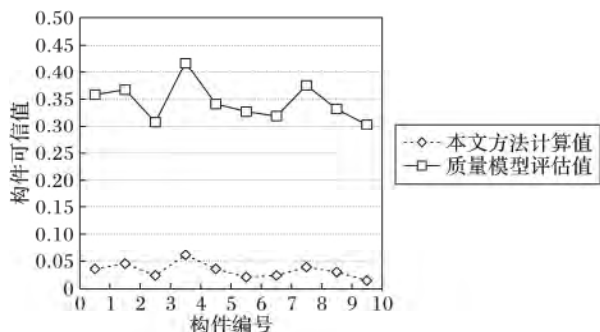


图4 构件可信评估实验对比

整体上,由质量模型评估出的可信值要高于本文方法得出的可信值,但曲线的变化趋势基本一致,说明基于构件使用依赖关系得到的构件复用可信度能够基本反映构件的实际质量情况。而与质量模型评估方法相比,基于构件使用依赖关系的构件复用可信度计算方法考虑的因素较少,收集证据相对容易,因此可以大幅度降低构件可信评估的工作量,提高评估效率。

5 结语

当前,网络上存在大量软件构件,如何找到和复用高质量的可信构件是软件工程领域中的重要问题。本文认为被多次使用过的构件拥有更高的复用可信性,而构件使用依赖是一种切实存在且有记录可查的证据信息,因此提出了基于构件使用依赖关系的构件复用可信度计算方法。该方法将构件库作为证据库,对证据库中的构件依赖信息进行采集整理,其获得的结果作为构件可信证据为构件可信性度量提供支持,通过特定的转换算法生成可直接用于可信性评估的评估证据,最后再基于评估证据得到构件最终的可信度。在此基础上,本文还充分考虑了不同证据的不同可信度,为来源于同一构件库和不同构件库的证据分配不同的可信权重。之后,将提出的可信度计算方法运用到构件库实例中,测试了该方法是可行和实用的。但是尽管如此,还是存在缺点与不完善处,本文的方法考虑因素较少,构件的可信性度量指标较少。接下来的研究将仍然对构件的可信性度量进行研究,试图对构件发布时间、构件的版本信息、使用反馈等因素进行考虑,在完善此方法基础上,将该方法运用于构件库的管理中。

参考文献:

- [1] ISO. ISO 9126, standard for software engineering-product quality [S]. Geneva: ISO, 2001.
- [2] MAO G, LI J, GE X, et al. The metrics and application of software-based component quality model [J]. Computer Applications and Software, 2005, 22(5): 1-4. (毛国蓓,李雪静,葛孝堃,等.

基于软件构件质量模型的度量及应用[J]. 计算机应用与软件, 2005, 22(5): 1-4.)

- [3] MEYER R. The grand challenge of trusted components [C]// Proceedings of the 25th International Conference on Software Engineering. Washington, DC: IEEE Computer Society, 2003: 660-667.
- [4] GUO S, LAN Y, JIN M. Some issues about trusted components research [J]. Computer Science, 2007, 34(5): 243-246. (郭树行,兰雨晴,金茂忠. 软件构件的可信保证研究. 计算机科学[J]. 2007, 34(5): 243-246.)
- [5] TANG Y, ZHANG Y, CHEN H. Trusted component measurement model based on specific areas [J]. Computer and Modernization, 2014(10): 12-15. (唐莹,张育平,陈海燕. 一种基于特定领域的可信构件度量模型[J]. 计算机与现代化, 2014(10): 12-15.)
- [6] WANG H, LIU S, JIANG Y, et al. A component quality measurement model based on fuzzy matter element evaluation method [J]. Journal of Yunnan University: Natural Science Edition, 2015, 37(1): 31-42. (汪海涛,刘帅,姜瑛,等. 一种基于模糊物元评价法的构件质量度量模型研究[J]. 云南大学学报: 自然科学版, 2015, 37(1): 31-42.)
- [7] WANG Y, ZENG G. Research on trust evaluation model for component assets based on fuzzy sets [J]. Application Research of Computers, 2014, 31(5): 1467-1469, 1474. (汪永好,曾广平. 基于模糊集合的构件资源信任评估模型研究[J]. 计算机应用研究, 2014, 31(5): 1467-1469, 1474.)
- [8] XU F, LYU J, ZHENG W, et al. Design of a trust valuation model in software service coordination [J]. Journal of Software, 2003, 14(6): 1043-1051. (徐锋,吕建,郑玮,等. 一个软件服务协同中信任评估模型的设计[J]. 软件学报, 2003, 14(6): 1043-1051.)
- [9] YIN Z, YIN W. Game theory analysis of conflict in trust of collaboration for internetware [J]. World Sci-Tech R & D, 2008, 30(4): 449-451. (殷脂,殷卫红. 开放协同环境下网构软件的信任管理研究[J]. 世界科技研究与发展, 2008, 30(4): 449-451.)
- [10] ISHIHARA T, HOTTA K, HIGO Y, et al. Reusing reused code [C]// Proceedings of the 20th Working Conference on Reverse Engineering. Washington, DC: IEEE Computer Society, 2013: 457-461.
- [11] XIA Y, YU L, LUO W. Redundant autonomous software component based trusted computing mechanism [J]. Computer Engineering and Design, 2008, 29(10): 2467-2469. (夏榆滨,于玲,罗万华. 基于多余度自治软件构件的可信计算机制[J]. 计算机工程与设计, 2008, 29(10): 2467-2469.)
- [12] XUE D, PAN Y, XIE B, et al. A relational description model for supporting component relationship management [C]// Proceedings of the 2004 National Software Application Conference. Beijing: China Machine Press, 2004: 50-55. (薛大宇,潘颖,谢冰,等. 一种支持构件关系管理的描述模型[C]// 2004 全国软件和应用学术会议论文集. 北京: 机械工业出版社, 2004: 50-55.)
- [13] DUAN X. Evidence theory and decision making, artificial intelligence [M]. Beijing: Renmin University of China press, 1993: 20-40. (段新生. 证据理论与决策、人工智能[M]. 北京: 中国人民大学出版社, 1993: 20-40.)
- [14] HE J. Design and implement a trust evaluation system of BPM domain component [D]. Xi'an: Northwest University, 2010: 22-46. (贺久松. BPM 领域构件可信评估系统的研究与实现[D]. 西安: 西北大学, 2010: 22-46.)