ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.2, February 2010, pp.359−372
doi: 10.3724/SP.J.1001.2010.03786

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种支持软件资源可信评估的框架*

蔡斯博[1,2], 邹艳珍[1,2+], 邵凌霜[1,2], 谢 冰[1,2], 邵维忠[1,2]

[1](北京大学 信息科学技术学院 软件研究所,北京 100871)

[2](高可信软件技术教育部重点实验室(北京大学),北京 100871)

Framework Supporting Software Assets Evaluation on Trustworthiness

CAI Si-Bo[1,2], ZOU Yan-Zhen[1,2+], SHAO Ling-Shuang[1,2], XIE Bing[1,2], SHAO Wei-Zhong[1,2]

[1](Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

[2](Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871, China)

+ Corresponding author: E-mail: zouyz@sei.pku.edu.cn, http://www.pku.edu.cn

Cai SB, Zou YZ, Shao LS, Xie B, Shao WZ. Framework supporting software assets evaluation on trustworthiness. *Journal of Software*, 2010,21(2):359−372. http://www.jos.org.cn/1000-9825/3786.htm

Abstract: This paper proposes a framework supporting software assets evaluation on trustworthiness, analyzes the technologies included in this framework, such as evidence collection, trust management for evidence, trustworthiness evaluation and so on. Furthermore, this paper presents the design decisions and solutions of this framework in software asset library of Peking University. A detailed case study is also given

Key words: software asset library; software trustworthiness; trustworthiness evaluation

摘 要: 提出了一种支持软件资源可信评估的框架,并分析了该框架涉及到的技术,如证据收集、证据信任管理和可信评估等.阐述了该框架在北京大学软件资源库中的设计决策和实现方案,并给出一个详尽的实例分析.

关键词: 软件资源库;软件可信;可信评估

中图法分类号: TP311 **文献标识码**: A

软件资源库作为有效管理软件资源及其相关信息的基础设施,为软件开发者提供可复用的软件资源[1,2].软件资源库存有多种形态的软件资源,例如源代码构件、Web服务、软件工具等.随着Internet技术的兴起和发展,软件资源库的服务形式从早期面向封闭的、熟知用户群体和相对静态的形式,逐渐转变为开放的、公共可访问的、高度动态的形式.此时,复用者能够更加高效、充分地利用各种软件资源,但同时也带来了软件资源的质量难以预测和控制等问题.下载和集成不符合复用者需求的软件资源势必会损害到复用者的利益,这直接影响了复用者使用软件资源的信心.针对这一问题,本文提出了一套在软件资源库中对资源进行可信评估的解决方案.

软件可信是近年来被广泛研究的热点.早在 1995 年,Avizienis等人就提出了可信计算(dependable computing)的概念,并将软件系统的可信赖性(dependability)定义为“系统提供可信赖的计算服务的能力,这种可

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2007AA010301-01 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2005CB321805 (国家重点基础研究发展计划(973)); the Fund for Creative Research Groups of China under Grant No.60821003 (国家创新研究群体科学基金)

Received 2009-06-15; Revised 2009-09-11; Accepted 2009-12-07

信赖性是可以验证的”,在其后续的工作中[3,4]给出了软件可信性所包含的属性.另外,微软公司于 2002 年提出可信计算(trustworthy computing)的概念,同时发布了可信计算白皮书[5],重点关注保密安全性(security)、私密性(privacy)、可靠性(reliability)和业务完整性(business integrity)这 4 个属性.美国的信息与通信委员会(the committee on information and communications,简称CIC)则提出了高可信(high confidence)软件系统的概念.Wang等人[6]提出了一个Internet环境下软件的可信概念模型(concept model for trustworthiness)及可信保障框架(trustworthiness assurance framework).该框架综合考虑了软件在身份、能力和行为等方面的可信属性.陈火旺等人[7]提出了软件系统的高可信性质,并描述其为“系统需要满足的关键性质,当软件一旦违背这些关键性质会造成不可容忍的损失”.吕建等人[8]针对开放环境下的软件系统这一新的软件形态,提出了基于信任管理的软件可信保障方法的新思路.

上述相关工作集中关注了软件可信的定义以及软件可信所包含的属性.然而,对于实际中如何去评估一个软件是否可信,如何获得评判资源可信的依据,并没有给出明确的解决方案.另外,Wang等人[6]和吕建等人[8]的工作主要针对于特定软件形态的可信保障,其工作并不完全适用于解决软件资源的可信评估问题.为此,本文提出一个在软件资源库中评估软件资源可信的框架,以帮助复用者在软件资源库中了解、复用可信赖的软件资源.该框架共由 3 个部分组成:

1) 证据收集.采集、存储和展示支持软件资源可信评估的各类证据.研究软件资源可信证据的内涵和外延,分析如何获取这些证据,如何提供一套灵活的机制对其进行描述.
2) 证据信任管理.分析证据的可靠性和真实性,建立资源复用者与证据提供者之间的信任关系,保障软件资源可信评估过程中所使用证据的有效性.
3) 可信评估.建立适用于软件资源的可信评估方法,通过评估结果帮助复用者使用可信的软件资源.

基于该框架,本文在北京大学软件资源库中设计并实现了动态的软件资源可信评估机制,并结合实例进行了相关模型、方法和技术的检验.

本文第 1 节介绍与软件可信相关的一些概念.第 2 节重点描述软件资源库中资源的可信评估框架.第 3 节介绍该框架在北京大学软件资源库中的实现,并通过一个具体案例描述该框架在实际系统中的应用.第 4 节总结全文并展望下一步的工作.

1 软件资源的可信

一般认为,软件可信指的是软件的行为和结果符合用户的预期[6,9].由于软件的行为及其产生的结果通常可以通过定义一组适当的属性来表达,软件可信性可通过可信属性以及用户在可信属性上的预期共同表达[9].若一个软件在可信属性上的度量值均满足某用户预期,则称该软件对该用户来说是可信的;同理,若一个软件对用户来说是可信的,那么该软件在可信属性上的度量值均满足用户的预期.

可信属性是用于描述和评价软件可信的一组属性[9].它更强调软件非功能的质量属性.在实际应用中,可信属性往往根据不同的环境可能具有不同的含义.因此,在进行软件可信评估的过程中,往往根据自身应用的背景和技术特点,选择并支持一些特定可信属性.例如,Avizienis等人[3]从软件系统避免产生不可接受的服务故障的角度看,认为可信属性包括可访问性(availability)、可靠性(reliability)、安全性(safety)、完整性(integrity)和可维护性(maintainability).而微软公司[5]从软件用户的关注点出发,认为软件可信应包含保密安全性(security)、私密性(privacy)、可靠性(reliability)和商务完整性(business integrity)这 4 个属性.

基于上述定义,软件资源的可信是指软件资源的行为和结果符合用户的预期,资源的可信性体现为用户对软件资源在满足一定可信属性要求的前提下完成某个操作的信心.在资源库中,资源的用户为资源复用者.必须考虑到在资源检索过程中,由于应用领域、任务需求的不同,复用者之间必然存在着不同的需求.例如,当需要一个记录日志的资源时,从事实时软件开发的复用者更关心资源的响应时间等性能方面的指标;而在完成一个电子商务软件的开发过程中,复用者更关心的是该资源能够具有很好的可维护性,以适应多变的业务流程.也就是说,在不同的复用场景下,资源所应对的可信需求是不同的.为此,本文在评估框架中引入了可信需求的概念.可

信需求描述了主体对客体进行评估的属性约束条件,包括属性名称、属性权重、属性值约束以及属性之间的关系,支持用户依据软件资源的应用需求和应用领域选择使用一定的可信属性,保证了可信属性定义的灵活性.

为了评判软件资源是否符合用户的预期,需要对软件资源的可信性进行评估,即可信评估.可信评估是基于相关证据的,称为可信证据.在软件资源库中,可信证据描述了与软件资源相关的,能够反映资源某些可信属性的度量值数据、文档或其他信息.可信证据可以从多个阶段采集,例如软件资源的开发阶段、使用阶段等.由于提交到资源库中的软件资源是已经开发完毕以供复用的,本文更多地是从软件资源的实体本身和被复用情况来获取可信证据.首先,通过在软件资源入库之初对提交的资源进行自动化或手工审查获取可信证据,例如,对源代码资源进行静态代码分析检测其是否存在代码缺陷等;其次,软件资源的使用情况,例如被复用次数、使用者的反馈等,则是软件资源使用阶段可获取的可信证据.特别地,对于 Web 服务这类特定的软件资源,其运行时的 QoS 数据也是一类有力的可信证据.

可信证据为软件资源评估提供了有力的依据.但是,还必须考虑到证据的可靠性和真实性问题.在资源库中,每个用户关注的技术领域不同,不同用户对某一个技术领域的熟悉情况不同.因此,在资源可信评估过程中,不同用户往往对不同来源的证据存在不同的信赖程度.为此,本文借鉴信任管理的基本思想,建立并维护了资源复用者与证据信息之间的信任关系,以保障资源可信评估所使用证据的有效性.

2 软件资源可信评估框架

图 1 给出了软件资源库中软件资源的可信评估框架.该框架是一个自底向上的层次结构,底层的证据模型定制和证据实例采集用于支撑上层软件资源的可信评估及应用.评估框架应用(application layer)是可信评估框架与软件资源库的主要接口,它利用软件资源可信评估的结果支持基于可信的相关应用,如基于可信的软件资源获取.软件资源可信评估框架作为软件资源库中资源可信的评估支撑机制,用于保障软件资源库向复用者提供可信的软件资源.软件资源的可信评估框架包括证据收集、证据信任管理和软件资源的可信评估这 3 个部分.

证据收集(evidence collection).证据收集是整个可信评估框架的基础,为上层软件资源的可信评估及应用提供支持.它包括了证据模型定制和证据实例采集与存储.其中,证据模型定制用于定义证据模型,并以此组织获取到的证据实例.证据实例有多种方式的获取渠道,如复用者提交的使用反馈(reuse data)、Web 服务的 QoS 数据采集等.不同获取来源的证据实例由于侧重点不同,其内容和组织结构通常是各异的,这就要求证据模型定制方法具有较好的灵活性以支持这一点.

证据信任管理(trust management for evidence).证据信任管理是软件资源可信评估的关键一环,主要用于处理采集到的证据中存在不实或不准确证据信息的情况,从而提高软件资源可信评估的有效性.本文通过建立复用者对证据提供者的信任关系来做到这一点.

可信评估(trustworthiness evaluation).可信评估是软件资源可信评估框架的核心部分,它利用获得到的证据实例以及用户可信需求,对软件资源可信进行评估.其中,可信需求模型的定制用于表达用户对软件资源的期望.由于软件资源复用者受到实际系统中应用需求和应用领域的限制,其对软件资源的可信需求往往不同,可信评估框架需要支持用户表达其可信需求.将资源复用者对软件资源的可信需求表示为一个需求模板,通过信任属性、属性权重和属性约束的定制,实现了可信评估指标体系的可定制性和可扩展性;建立了需求模板和证据信息之间的属性关联,使得评估方法能够对不同来源的证据信息进行分析,具有较好的可操作性.

本节后续部分将具体阐述软件资源可信评估框架的 3 个主要组成部分.

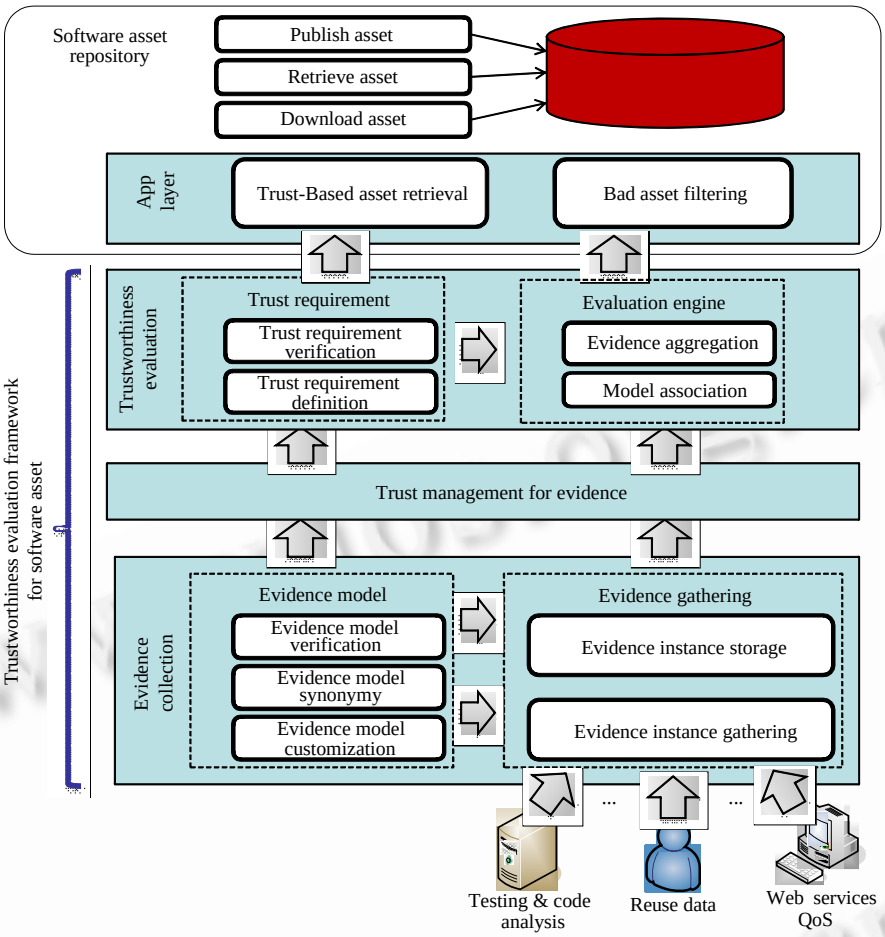


Fig.1 Trustworthiness evaluation framework for software asset

图 1 软件资源可信评估框架

2.1 证据收集

证据收集位于可信评估框架的最下层,为软件资源可信性评估的依据获取提供支持.证据收集包含证据模型定制和证据实例采集两个部分.证据模型定制用于定义和组织获取到的证据信息,为证据信息的采集和存储提供数据结构描述.证据采集利用已定义的证据模型向外提供采集接口获取证据信息,其中,按照证据模型进行描述的证据信息本文称为证据实例.

证据实例可以从多个获取来源获得,例如,软件资源经过测试、分析的结果以及复用者的使用评价信息等.不难发现,从不同获取来源获得的证据实例的内容往往是不一样的,如源代码的静态分析就无法获得易用性,使用评价得不到内存泄漏个数等.

由于不同获取来源的证据实例的内容和组织结构通常是各异的,需要为这些各异的证据实例提供一套统一的管理机制.一种做法[10]是建立一个足以描述所有实例的数据模型,然而,使用这样的数据模型往往会带来模型过度膨胀等问题.因为一旦有新的异构实例出现,就需要扩充数据模型.相反地,本文采用一种基于来源定制证据模型的方式支持多种来源证据信息的描述,系统允许同时存在多个证据模型.采用该证据建模方法具有如下优点:1) 与实际情况相符.证据模型根据实际采集的证据信息来源定制,更符合实际情况.2) 证据模型易于扩展.对不同获取来源的证据信息分别建模,当捕获到一种新的证据信息来源,直接对其建模即可,不影响之前已定义的证据模型.3) 证据模型易理解.同一证据信息来源的提供者们对相应证据模型的理解类似,例如,Web服

务QoS数据的提交者很容易就能理解QoS模型中的QoS属性.4) 证据模型易于学习.证据信息的提交者只需根据自身情况学习需要的证据模型,无须学习所有证据模型.另外,根据单一证据来源定制的证据模型一般会较为简单,易于理解.5) 便于管理.利用多个证据模型代替一个庞大的模型,避免单个证据模型过度膨胀带来的 问题.

2.1.1 证据模型定制

对于单一证据模型,本文采用一种多层次树状结构的证据模型对证据实例进行表达,如图 2 所示.证据模型可由一个三元组描述,即⟨*EM_INFO*,*EM_NODE*,*EM_VALTYPE*⟩.其中:

- *EM_INFO* 指代一个证据模型,由一个三元组来描述,即⟨*EM_ID*,*EM_NAME*,*EM_SRC*⟩,分别是该证据模型的唯一标识符、证据模型的名称以及证据模型的证据获取来源.*EM_SRC* 的取值是一个可扩展的枚举型列表,目前包含有资源测试、代码分析、使用评价、QoS 数据这 4 个证据来源.
- *EM_NODE* 用于指代证据模型中除根节点外的所有节点,它包含了两种类型的节点、叶子节点和非叶节点.非叶节点本文称为证据特性,证据特性可以嵌套,子特性是对父特性更详细的描述.叶子节点本文称为证据属性,证据属性具有原子性,它不可再分.*EM_NODE* 可以用一个三元组来描述,即⟨*EM_NODE_ID*,*EM_NODE_NAME*,*EM_SUBNODESET*⟩,分别是 *EM_NODE* 的唯一标识符、节点名称以及子节点集.*EM_SUBNODESET* 是一个以 *EM_NODE* 为元素的集合.当 *EM_SUBNODESET* 不为空时,该*EM_NODE* 是证据特性;反之,当*EM_SUBNODESET* 为空时,该*EM_NODE* 是证据属性.在证据模型中,证据特性主要用于辅助描述证据模型,它最终由证据属性组成.证据特性可以认为是其包含证据属性的综合,本身并不具备计算意义,只有证据属性是资源可信评估中有意义的计算单元.
- *EM_VALTYPE* 定义了证据属性的取值类型,证据模型中的每个证据属性都会对应到一种取值类型.证据属性的可选取值类型见表 1.取值类型可由一个三元组来描述,即⟨*EM_VALSET_ID*, *EM_VALSET_TYPE*,*EM_VALSET_TYPE*⟩,分别表示证据属性类型的唯一标识符、类型以及取值范围.为了支持软件资源的自动评估,证据属性暂不考虑字串型的取值类型.

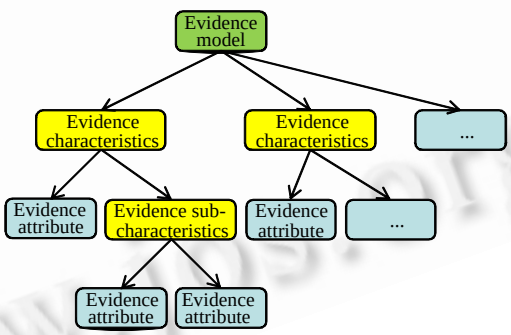


Fig.2 Evidence model

图 2 证据模型

Table 1 Value type for evidence attribute

表 1 证据属性取值类型

ID	Type	Value
0	General	Float, 0~1
1	Boolean	0 or 1
2	Level	Integer, 0~5
3	Ratio	Float, 0~100
4	Any	Float

在可信评估框架中引入多证据模型,考虑到不同证据模型可能存在表达相同含义的证据属性,需要为这样的证据属性建立同义关联.建立模型属性同义关联的优点是明显的,在基于证据模型属性的可信评估中,利用证

据属性间的同义关联,可以在证据模型间获取到更多对某一证据属性的度量值.本文使用公式(1)所示的二元关系表示证据属性间的同义关联.

$$syn=\langle EM\_NODE1,EM\_NODE1\rangle \quad (1)$$

其中,*EM_NODE*1 和 *EM_NODE*2 表示两个隶属于不同证据模型的证据属性,且具有相同的含义.通过定义证据属性的同义关联,证据模型的关联如图 3 所示.不难发现,模型属性的同义关联具有自反性、对称性和传递性等特点;另外,同一个证据模型中不存在两个证据属性具有同义关联.

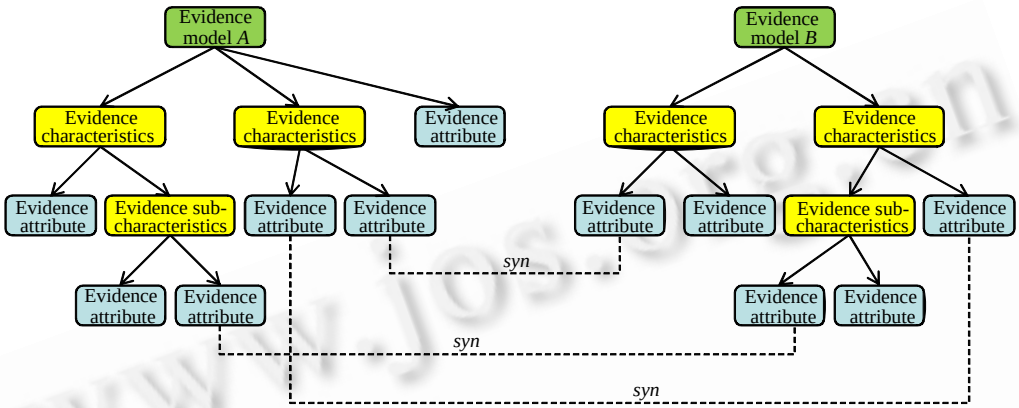


Fig.3 Synonymous association between evidence model attributes

图 3 证据模型属性同义关联

2.1.2 证据实例采集

基于已定义的证据模型,评估框架对外提供接口采集证据信息.证据信息具有多个获取来源,根据不同的来源,如源代码评测、QoS 监控等定制不同的证据模型,并将采集的数据提交并保存起来.此外,证据实例采集模块还支持对证据信息进行预处理.

首先,证据模型包含两个方面的内容:模型内容以及模型节点的数据类型.在基于来源的证据模型定制中,模型内容根据可采集的证据信息而定;另一方面,采集到的证据信息是原始数据,不一定完全符合模型节点的数据类型,如第 3 节的证据信息实例 1,因此有必要作进一步的数据类型转化,转化的方式因具体采集的证据信息和证据实例的差异而定.

其次,用户的使用评价是一类比较特殊的证据信息,来源于用户的主观反馈.由于用户的主观评价可能存在评价差异较大的问题,本文利用我们先前的工作[11]对用户使用评价进行预处理,进一步保证所采集证据信息的合理性.

2.2 证据信任管理

证据信任管理是软件资源可信评估框架中关键的一环,它涉及到评判获取到的证据实例是否真实和准确的问题.关于证据信任管理的引入,本文考虑了以下 4 个方面的因素:

- 证据实例来源的权威性,如权威第三方机构提供的评测结果.
- 复用者和证据提供者应用软件资源环境的差异,如网络环境的差异[12].
- 恶意证据实例提交者为了吹嘘或贬低某一特定软件资源,提供不实证据信息[13].
- 复用者的信任偏好[11],如复用者可能更愿意信任从客观渠道获取的证据信息.

无论是从何种证据获取渠道获得,证据实例最终都是通过具体用户来提交的.细究上面 4 个因素不难发现,证据实例信任的根源在于其提交者.为此,本文通过建立复用者对证据实例提交者的信任关系解决证据信任的问题.证据信任管理描述了复用者对证据发布者提供准确证据信息的信赖程度.本文定义的信任关系是一个二元的有向关系,由一个三元组来表示,如公式(2)所示.其中,*Truster* 表示信任者;*Trustee* 表示被信任者;*deg* 表示信

任程度,值为 0~1 之间的浮点数.

$$Tr=\langle Truster,Trustee,deg\rangle \tag{2}$$

将一个用户建立的所有信任关系集合起来,本文用一个 N 维的向量 T 表示该用户建立的所有信任关系,其中,N 为用户总数,向量中的元素均为 0 到 1 之间的浮点数.向量 T 在第 t 维的值表示该用户对第 t 个用户提供的证据信息的信任程度.若该用户未建立对某一特定用户的信任值,则用一个默认值替代.

2.3 可信评估

软件资源的可信评估是整个可信评估框架的核心部分.它利用获取的证据实例以及用户定义的可信需求模型对软件资源进行可信评估.

从本文第 1 节可知,仅当软件资源的行为及结果符合用户的预期时,才称软件资源对用户而言是可信的.本文使用可信需求模型对用户预期进行建模.注意到不同复用者受到其实际项目中应用需求和应用领域的限制,其对软件资源的需求往往不同.为此,复用者需要根据其实际需求定制可信需求模型.需要指出的是,可信需求模型不仅可以用于反映某一特定用户对软件资源的需求,也可用于表达应用于特定领域,如航天、金融等领域的软件资源的需求.

在软件资源可信评估过程中,可信需求模型是评估的目标.它需要证据模型和证据实例的支撑.然而,可信需求模型与证据模型存在着鸿沟,因为它们来源于不同的定制者.为此,在可信评估过程中还需要建立可信需求模型到证据模型的关联以获得评估依据.最后,利用可信需求模型及其与证据模型的关联聚集证据实例,以确认软件资源是否符合用户需求.

2.3.1 可信需求模型

类似于证据模型定制,本文采用多层次树状结构对可信需求模型进行描述.可信需求模型可用一个三元组进行描述,即⟨*EXPM_INFO*,*EXPM_NODE*,*EXPM_VALUE*⟩,其中:

- *EXPM_INFO* 指代一个可信需求模型,通过一个三元组来描述,即⟨*EXPM_INFO_ID*,*EXPM_INFO_NAME*,*EXPM_INFO_PBER*⟩,分别是可信需求模型的唯一标识符、名称和发布者.
- *EXPM_NODE* 指代可信需求模型中除根节点外的所有节点.类似于证据模型的节点,*EXPM_NODE* 可以用一个三元组来描述,即⟨*EXPM_NODE_ID*,*EXPM_NODE_NAME*,*EXPM_SUBNODESET*⟩,分别是该节点的唯一标识符、节点名称以及子节点集.这里,我们称 *EXPM_SUBNODESET* 不为空的节点为需求特性,称 *EXPM_SUBNODESET* 为空的节点为需求属性.需求特性主要用于辅助描述可信需求模型,可以认为是其包含需求属性的综合,只有需求属性是具有评估意义的单元.
- *EXPM_VALUE* 指代用户在需求属性上的期望值,每个需求属性均包含有一个期望值.

2.3.2 可信需求模型的定制

可信需求模型需要复用者根据自身实际需求自定义需求属性,并以此为基础建立可信需求模型到证据模型的关联.

如前所述,可信需求模型与证据模型存在着鸿沟.可信需求模型是从复用者角度建立起来的,反映了复用者对软件资源的期望,而软件资源的评估依据来自于证据模型,这造成了软件资源复用者和证据信息发布者的数据模型鸿沟.为了填补这一鸿沟,需要建立从用户可信需求模型到证据模型的关联.

一个可信需求模型可能会同时关联到多个证据模型,图 4 给出了一个可信需求模型与证据模型关联的示例.另外,一个可信需求属性可能需要由多个证据属性才能描述.例如,安全性(security)被认为是可用性(availability)、完整性(integrity)和保密性(confidentiality)的总和[3].因此,本文将每个需求属性构造成如公式(3)所示的集合.其中,$ExpAttr$为需求属性,$EviAttr_k$为证据属性,deg_k为关联度.

$$AssSet_{ExpAttr}=\{\langle EviAttr_1,deg_1\rangle,\langle EviAttr_2,deg_2\rangle,\ldots,\langle EviAttr_k,deg_k\rangle\},\text{满足}\sum_{i=0}^{k}deg_i=1 \tag{3}$$

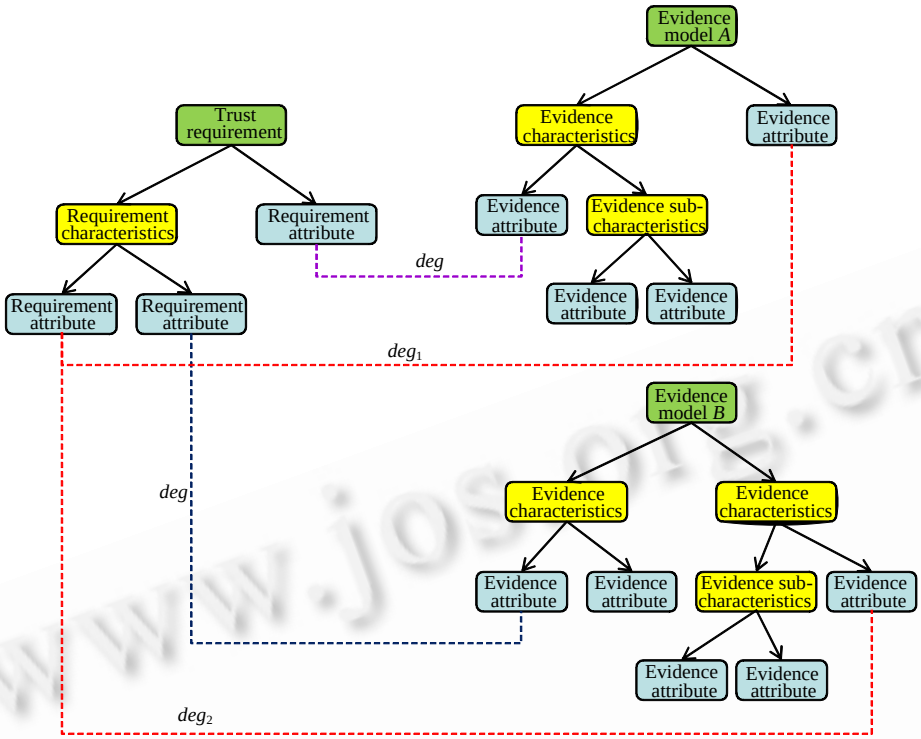


Fig.4 Association from trustworthiness requirement model to evidence model

图 4 可信需求模型与证据模型的关联

2.3.3 证据聚集

基于定制的可信需求模型,评估引擎聚集所有可用证据实例对软件资源进行可信评估.证据聚集的过程涉及到用户的可信需求模型、信任向量 T 以及证据模型和证据实例.

算法 1 给出了证据实例的聚集算法.算法的输入包括用户可信需求模型 *ExpModel*、信任向量 T、证据模型集 *EviModelSet*、证据实例集 *EviModelInsSet*.算法的输出是根据用户可信需求模型聚集证据实例的结果.

算法 1. 证据聚集算法.

输入:可信需求模型 *ExpModel*、信任向量 T、证据模型集 *EviModelSet*、证据实例集 *EviModelInsSet*.

输出:证据聚集结果.

1: 初始化 *ExpModel* 关联的证据模型集 *AssEviModelSet*

2: 初始化临时结果集 $R=\varnothing$

3: For each *AssEviModel* in *AssEviModelSet* //分别对每个证据模型聚集证据实例

Begin

4: 获取按*AssEviModel*构造的证据实例集$AssEviModelInsSet=\{Ins_1,Ins_2,\ldots,Ins_m\}$

5: 获取$\{Ins_1,Ins_2,\ldots,Ins_m\}$的提交者$\{pu_1,pu_2,\ldots,pu_m\}$

6: 获取$\{pu_1,pu_2,\ldots,pu_m\}$在信任向量$T$中的值$\{t_{pu_1},t_{pu_2},\ldots,t_{pu_m}\}$

7: 构造 *AssEviModel* 的一个实例 *AggIns*,*AggIns* 的被信任程度 t

8: $AggIns=\sum_{i=1}^{m}t_{pu_i}\times Ins_i,\ t=\sum_{i=1}^{m}t_{pu_i}$

9: $R=R\cup\{\langle AggIns,t\rangle\}$

End

10: 构造 *ExpModel* 的实例 *ExpIns*

11: for each 需求属性 *ExpAttr* in *ExpIns* //处理需求属性

Begin

12: $AssSet_{ExpAttr}=\{\langle EviAttr_1,deg_1\rangle,\langle EviAttr_2,deg_2\rangle,\ldots,\langle EviAttr_k,deg_k\rangle\}$

13: For each $EviAttr_i$ in $AssSet_{ExpAttr}$ //处理同义属性

14: Begin

15: $SynSet_{EviAttr_i}=\{SynAttr_1,SynAttr_2,...,SynAttr_n\}$ //$EviAttr_i$的同义属性集

16: $$EviAttr_i=\frac{\sum_{j=1}^{n}FetchInsTrFromR(SynAttr_j)\times FetchInsValFromR(SynAttr_j)}{\sum_{j=1}^{n}FetchInsTrFromR(SynAttr_j)}$$

End

17: $$ExpAttr=\frac{\sum_{i=1}^{k}deg_i\times EviAttr_i}{\sum_{i=1}^{k}deg_i}$$

End

18: return *ExpIns*

算法的具体流程是:

- 首先,根据用户可信需求模型 *ExpModel* 获得关联的证据模型集 *AssEviModelSet*.*AssEviModelSet* 包括直接与 *ExpModel* 建立关联的证据模型和利用证据模型间的同义属性关联得到的证据模型.
- 其次(从第 3 行开始),对 *AssEviModelSet* 中的各个证据模型聚集证据实例,聚集的同时考虑证据实例发布者的被信任程度.其中,第 7 行、第 8 行基于每个证据模型构造一个聚集后的证据实例,为对证据实例的属性分别作相应运算,记录该实例的被信任程度.根据证据模型聚集后的证据实例及其被信任程度存放在临时结果集 *R* 中.
- 再次(从第 10 行开始),计算可信需求模型*ExpModel*中每个需求属性的度量值.在第 15 行、第 16 行,算法分别计算需求属性关联到的各证据属性$EviAttr_i$的值,这里需要考虑$EviAttr_i$的同义属性.其中,*FetchInsTrFromR*表示从*R*中获取指定证据属性所在证据实例的信任值,*FetchInsValFromR*表示从*R*中获取指定证据属性的度量值.在第17行,考虑到一个需求属性可能关联到多个证据属性,需要根据关联程度对多个证据属性的度量值作加权平均.
- 最后,返回值 *ExpIns* 是按照用户可信需求聚集证据实例的结果.当聚集结果 *ExpIns* 满足用户的可信需求时,称软件资源对用户而言是可信的.

2.4 讨 论

本文所述的软件资源可信评估框架采用一种较为灵活的证据信息提交方式,证据模型的定制基于证据信息的来源.对证据提交者来说,他只需花费较少的时间和精力就能把证据信息提交到软件资源库中.但这同时也为后续评估工作带来了一定的困难,例如,如果存在着多个证据模型,我们就需要解决同义属性关联的问题.

本文所谈的软件资源可信评估是从一个特定用户的角度出发去评估一个软件资源.换言之,同一个软件资源对不同的用户来说其可信评估的结果可能是不一样的,这与用户的实际应用需求和应用领域相关.基于这样的理解,可信评估框架把用户的不同应用需求建模成用户对软件资源的可信需求,同时给出一套定制方法,更加贴合用户的实际需要.这一做法带来的困难在于,可信评估时我们需要解决用户可信需求模型与证据模型之间存在鸿沟的问题.

在软件资源可信评估过程中不难发现,评估的结果是二值的,即软件资源可信或不可信.本文工作无法给出

一个软件资源对用户而言的可信度,这也是本文工作存在的一个局限.尽管如此,我们认为在给定某一特定软件资源证据实例聚集结果的前提下,软件资源对用户而言的可信度是与特定用户相关的,而这种相关性在实际中是难以精确建模的.一种解决方案是用户在给出可信需求模型的同时,给定几组不同层次的需求属性期望值,当证据实例聚集结果达到某一个层次之后,称软件资源对用户来说达到了相应的可信度.

3 应用实例

基于上节所述软件资源可信评估框架,我们在北京大学软件研究所研发的软件资源库中建立了软件资源可信评估的原型子系统.在本节中,我们通过对软件资源库中若干软件资源进行可信评估,从而说明本文的软件资源可信评估框架在实际系统中的应用.

3.1 系统简介

北京大学软件资源库是北京大学软件研究所研发的软件资源管理系统,软件资源库中的资源类型包括了构件(包括源代码构件和二进制构件)、Web 服务、软件工具、软件构架等.系统使用 EJB 接口规范和 J2EE 技术实现业务逻辑,并且对外提供 Web 形式的访问界面和 Web Service 形式的 API 接口,实现对软件资源的描述、分类、存储和检索等功能.目前,该系统已在北京、长沙、西安等软件园区得到了正式的使用,并在向全国其他十几个软件园推广应用.

为了保障软件资源库中软件资源的质量和复用的成功率,软件资源库系统需要对库中软件资源进行可信评估.我们将本文所述的软件资源可信评估框架应用到软件资源库中.可信评估依据的获取采用了较为灵活的多证据模型,基于来源定制证据模型;考虑到证据来源权威性不尽相同以及用户间应用软件资源环境的差异等实际情况,我们引入了证据信任管理;最后,根据用户对软件资源的可信需求模型对资源进行可信评估.

3.2 证据信息实例

北京大学软件资源库中包含多种类型的软件资源,如源代码构件、Web 服务等.针对不同类型的软件资源,可信证据获取的方式不同,形式也是多种多样的.例如,对于源代码构件,利用源代码分析工具可以获取软件资源某些质量属性的度量值;又如 Web 服务,采集的 QoS 数据可作为一项有力的证据信息;此外,复用者的使用评价也是证据信息的来源之一.下面,我们通过具体的证据信息实例说明本文框架如何对多种证据信息的采集提供支持.

例 1:PKUAS Container 是发布到北京大学软件资源库中的一个源代码构件.表 2 给出了 PKUAS Container 的一个源代码分析结果.该分析结果是利用 Telelogic 公司的 LOGISCOPE(http://www.telelogic.com/products/Logiscope/)静态度量工具分析得到的.LOGISCOPE 可以从系统、类和方法等 3 个层次对源代码的质量进行分析.表 2 给出了 PKUAS Container 在类层次上的分析结果,其中的百分数表示相应列所示级别的类占所有类的百分比.基于 LOGISCOPE 的分析结果就可以获得 PKUAS Container 在可维护性、可分析性、可修改性、稳定性和可测试性等属性上的度量值.

Table 2 Analysis result for PKUAS Container source code (%)

表 2 PKUAS Container 源代码的分析结果(%)

	EXCELLENT	GOOD	FAIR	POOR
Maintainability	6	88	6	—
Analyzability	77	23	—	—
Modifiability	78	12	9	—
Stability	88	11	1	—
Testability	17	79	4	—

例 2:天气预报服务 CDYNE Weather 是北京大学软件资源库从 Internet 上收集得到的一个 Web 服务.表 3 给出了该 Web 服务近一个月来 QoS 采集数据的统计结果.该数据来自部署于北京大学软件所的采集客户端.客户端收集了该 Web 服务的可用性、响应时间及可靠性这 3 个属性的信息.

Table 3 QoS data for CDYNE weather

表 3 CDYNE Weather 的 QoS 采集数据

QoS attribute		Value
Performance	Availability (%)	99.6
	Response time (ms)	3 174
Robustness	Reliability (%)	100

从上面的实例中不难看出,不同来源的证据信息的内容和组织结构是存在差异的.针对不同的证据信息来源,本文的可信评估框架允许证据信息发布者定制证据模型,同时提供一定的后台支持(如证据模型验证等),从而灵活、有效地管理各种证据信息,并为上层的可信评估提供支持.例如,基于 LOGISCOPE 分析的结果,系统可以为其建立包含有可维护性、可分析性、可修改性、稳定性和可测试性这 5 个证据属性的证据模型,并收集证据实例.当使用其他源代码分析工具,如 FindBugs(http://findbugs.sourceforge.net/),Jtest(http://www.parasoft.com/jsp/products/home.jsp?product=Jtest)时,系统可直接建立基于 FindBugs 和 JTest 分析结果的证据模型,完全不影响已有的证据模型.另外,由于不同的源代码分析工具可能会对相同的质量属性进行分析,系统通过建立模型间的同义属性关联,从而更加有效地支持后期的可信评估.

3.3 可信评估实例

如今,Web 服务技术是目前研究界和产业界普遍关注的一项技术,Web 服务是软件资源库中一类重要的软件资源.下面,本文以评估软件资源库中 5 个提供相似功能(天气预报)的 Web 服务为例,说明本文所述软件资源可信评估框架的应用.这里,本文假设一个软件资源复用者登录到软件资源库中,他根据自己实际的情况和应用需求定制了信任关系和可信需求模型,希望能够找到符合其需求的天气预报 Web 服务.天气预报服务的基本信息见表 4.

Table 4 Web service instances

表 4 Web 服务实例

ID	Name	Provider	WSDL address
WS1	CDYNE weather	CDYNE	http://ws.cdyne.com/WeatherWS/Weather.asmx?wsdl
WS2	City weather forecast Web service	WebXml	http://www.webxml.com.cn/WebServices/WeatherWebService.asmx?WSDL
WS3	Weather Web service	ws.79777.cn	http://ws.79777.cn/Weather.asmx?wsdl
WS4	National weather service	NOAA	http://www.weather.gov/forecasts/xml/DWMLgen/wsdl/ndfdXML.wsdl?wsdl
WS5	Weather	Deep training	http://www.deeptraining.com/webservices/weather.asmx?wsdl

证据模型.目前,软件资源库中与 Web 服务相关的证据模型有两个,分别是 QoS 模型和使用评价(evaluation of use)模型.图 5 展示了 QoS 模型、使用评价模型以及它们的同义关联属性.

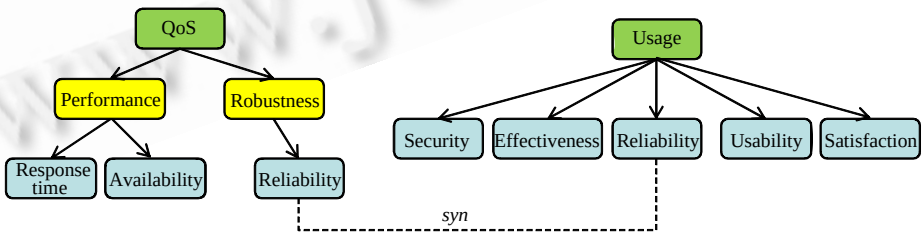


Fig.5 QoS model and evaluation of use model

图 5 QoS 模型和使用评价模型

QoS模型是基于我们以往的工作[14]定制并发布的,所不同的是,它更侧重于客户端QoS数据的采集.如图 5 所示,QoS模型包含了 2 个证据特性以及 3 个证据属性.由于证据特性主要用于辅助理解证据模型.这里重点介绍QoS模型的 3 个证据属性:

- 响应时间.Web 服务完成请求的任务平均花费的时间.
- 可用性.Web 服务是否已就绪可提供服务,通常指非超时调用次数占总调用次数的百分比.
- 可靠性.维持 Web 服务质量的程度,通常指非失效次数占总调用次数的百分比.

使用评价模型是从使用软件资源的角度出发,同时结合ISO/IEC 9126 质量模型[15]定制并发布的.它用于用户在使用完软件资源后提供反馈,是一种用户主观评价,共包含 5 个证据属性:

- 安全性.软件产品在指定使用条件下,获得可接受的损害人类、事务、软件、财产或环境风险级别的能力.
- 有效性.软件产品在指定的使用条件下,使用户能够准确和完整地获得规定目标的能力.
- 可靠性.在指定条件下使用时,软件产品维持规定的性能级别的能力.
- 易用性.在指定条件下使用时,软件产品被理解、学习、使用和吸引用户的能力.
- 满意度.软件产品在指定的使用条件下,使用户满意的能力.

另外,注意到 QoS 模型中定义的可靠性和使用评价模型中定义的可靠性具有相同的含义,为两者建立了同义属性关联.

证据实例获取.根据 QoS 模型采集到的证据实例来源有两个,它们分别是来自公网(网通(Public))的客户端和教育网(EDNET)的客户端,采集到的 QoS 数据见表 5.此外,软件资源库也收集到了来自用户的使用评价信息,由于使用评价的证据实例较多,这里不再一一列举.

Table 5 QoS data (Public/EDNET)

表 5 QoS 采集数据(Public/EDNET)

ID	Reliability (%)	Response time (ms)	Availability (%)
WS1	100/100	654/3 174	100/99.6
WS2	100/100	1 166/225	99.9/100
WS3	100/100	178/249	100/99.7
WS4	100/100	1 238/4 089	95/100
WS5	100/100	590/3 700	100/99.5

证据信任管理.该用户通过软件资源库提供的 Web 访问接口建立信任关系.假设该用户处于教育网环境,其对教育网客户端采集的 QoS 数据具有较高的信任度,指定为 0.9;反之,则对公网的客户端的信任度较低,指定为 0.1.由于我们对其他提供使用评价信息的用户并不了解,所以则并没有指定对他们的信任度.因此,我们对其他用户的信任度均使用了系统的默认值 0.4.

可信评估.该用户同时定制了对天气预报服务的可信需求模型,如图 6 所示.可信需求模型共包含 3 个需求属性,每个需求属性都包含相应的期望值:可靠性(*reliability*>90%)、响应时间(*response time*<1s)和满意度(*satisfaction*>90%),同时建立了可信需求模型到证据模型的关联关系.

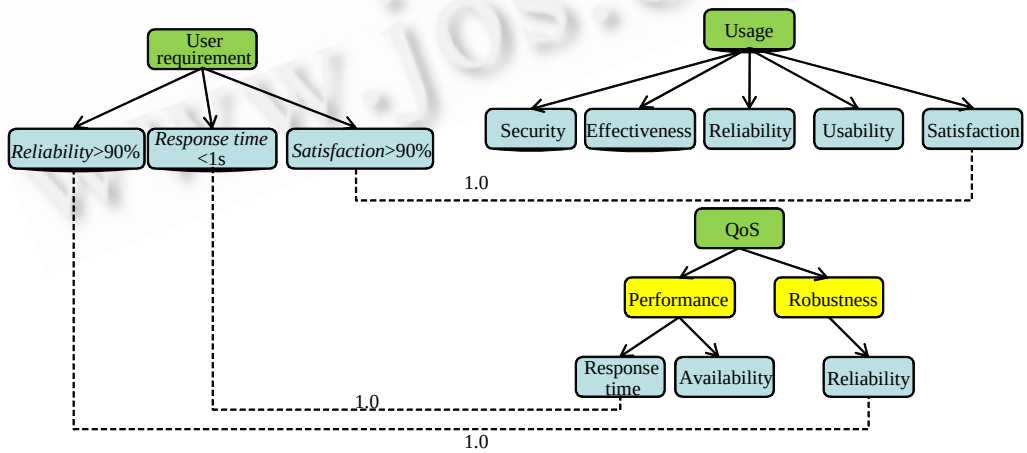


Fig.6 Trustworthiness requirement model and its association with evidence model

图 6 可信需求模型及其与证据模型的关联

利用定义的信任关系和可信需求模型,根据证据聚集算法聚集证据实例,5 个天气预报服务的聚集结果见表 6.由证据聚集的结果可知,只有 Web 服务 2 和 Web 服务 3 满足该用户需要.因此,Web 服务 2 和 Web 服务 3 对该用户而言是可信的.当用户在软件资源库中检索天气预报 Web 服务时,Web 服务 2 和 Web 服务 3 就会作为可信的软件资源推荐给该用户.

Table 6 Evidence aggregation result

表 6 证据聚集结果

ID	Reliability (%)	Response time (ms)	Satisfaction (%)
WS1	100	2 915	95
WS2	100	307	93
WS3	100	242	95
WS4	100	3 804	85
WS5	100	3 389	95

4 结束语

目前,软件资源库以开放、公共可访问、高度动态的形式对外提供服务,为软件开发者提供可复用的软件资源.然而,由于缺乏一套有效的软件资源评估机制,软件资源复用者在使用资源库的过程中可能下载到不符合其需求的软件资源,集成此类资源势必会损害复用者的利益.

基于这样一个事实,本文提出了一个在软件资源库中支持软件资源可信评估的框架,并详细分析了该框架涉及到的重要方法与核心内容,如证据的采集、证据信任管理、软件资源的可信评估等.此外,本文还介绍了该可信评估框架在北京大学软件资源库系统中的设计及实现方案,同时给出详尽的实例分析.

本文的工作虽然有效地支持了软件资源的可信评估,但原型系统在支持软件资源可信评估上仍需要一定的人工参与,例如,同义属性关联、可信需求模型与证据模型的关联等.在下一步工作中,我们将寻找一种自动或半自动的方法,在不降低评估有效性的同时尽可能减少人工参与的工作量,进一步提高可信评估框架的可用性.另外,提供更加有效的手段支持用户表达其可信需求也是我们未来的工作之一.

致谢 在此,我们向对本文的工作给予支持和建议的同行表示感谢.另外,特别感谢房路同学和马秀娟同学提供的数据支持.

References:

[1] Yang FQ, Mei H, Li KQ. Software reuse and software component technology. Acta Electronica Sinica, 1999,27(2):68−75 (in Chinese with English abstract).

[2] Pan Y, Liu Y, Xie B, Yang FQ. The basic component description model supporting management of the on-line components. Acta Electronica Sinica, 2003,31(12A):2110−2114 (in Chinese with English abstract).

[3] Avizienis A, Laprie JC, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. on Dependable and Secure Computing, 2004,1(1):11−33.

[4] Avizienis A, Laprie JC, Randell B. Dependability and Its Threats: A Taxonomy. Boston: Springer-Verlag, 2004. 91−120.

[5] Mundie C, de Vries P, Haynes P, Corwine M. Trustworthy computing. Microsoft White Paper, 2002. http://download.microsoft.com/download/a/f/2/af22fd56-7f19-47aa-8167-4b1d73cd3c57/twc_mundie.doc

[6] Wang HM, Tang YB, Yin G, Li L. Trustworthiness of Internet-based software. Science in China—Series F: Information Sciences, 2006,49(6):759−773.

[7] Chen HW, Wang J, Dong W. High confidence software engineering technologies. Acta Electronica Sinica, 2003,31(12A): 1394−1397 (in Chinese with English abstract).

[8] Lü J, Xu F, Wang Y. Trust management based software confidence assurance in open environment. Communications of the CCF, 2007,3(11):26−34 (in Chinese).

[9] Liu XD, Lang B, Xie B, Mao XG, Wang HM. Software trustworthiness classification specification. Version 2.0, the Key Program of the National High-Tech Research and Development Plan “Trustworthy Software Tools and Integration Environment” Technical Documentation, TRUSTIE-STC V2.0, 2009 (in Chinese).

[10] Zhao JF. Research on feedback management and run-time application supporting techniques for software component library [Ph.D. Thesis]. Beijing: Peking University, 2005 (in Chinese with English abstract).

[11] Zou YZ, Gu L, Li G, Xie B, Mei H. Rectifying prejudicial feedback ratings in reputation based trust management. In: van der Aalst W, Zhang LJ, eds. Proc. of the IEEE Int’l Conf. on Services Computing. Los Alamitos: IEEE Computer Society Press, 2007. 530−535.

[12] Shao LS, Zhang J, Wei Y, Zhao JF, Xie B, Mei H. Personalized QoS prediction for Web services via collaborative filtering. In: Zhang LJ, Birman KP, eds. Proc. of the IEEE Int’l Conf. on Web Services. Los Alamitos: IEEE Computer Society Press, 2007. 439−446.

[13] Jurca R, Faltings B, Binder W. Reliable QoS monitoring based on client feedback. In: Patel-Schneider PF, Shenoy P, eds. Proc. of the 16th Int’l Conf. on World Wide Web. New York: ACM Press, 2007. 1003−1012.

[14] Shao LS, Li T, Zhao JF, Wang YS, Xie B, Mei H. An extensible management framework for Web service QoS. Chinese Journal of Computers, 2008,31(8):1458−1470 (in Chinese with English abstract).

[15] National Standards of the People’s Republic of China——Software Engineering Product Quality. 2001 (in Chinese).

附中文参考文献:

[1] 杨芙清,梅宏,李克勤.软件复用和软件构件技术.电子学报,1999,27(2):68−75.

[2] 潘颖,刘洋,谢冰,杨芙清.支持管理在线构件的基本构件描述模型.电子学报,2003,31(12A):2110−2114.

[7] 陈火旺,王戟,董威.高可信软件工程技术.电子学报,2003,31(12A):1394−1397.

[8] 吕建,徐峰,王远.开放环境下基于信任管理的软件可信性保障.中国计算机学会通讯,2007,3(11):26−34.

[9] 刘旭东,郎波,谢冰,毛晓光,王怀民.软件可信分级规范.版本 2.0,国家高技术研究发展计划(863)重点项目“高可信软件生产工具与集成环境”技术文档, TRUSTIE-STC V2.0, 2009.

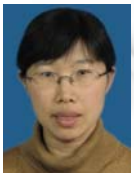
[10] 赵俊峰.构件库反馈管理及运行时应用支持技术的研究[博士学位论文].北京:北京大学,2005.

[14] 邵凌霜,李田,赵俊峰,王亚沙,谢冰,梅宏.一种可扩展的 Web Service QoS 管理框架.计算机学报,2008,31(8):1458−1470.

[15] 中华人民共和国国家标准——软件工程产品质量.2001.

蔡斯博(1984－),男,福建泉州人,博士生,主要研究领域为软件工程,软件构件管理技术.

谢冰(1970－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程,形式化方法.



邹艳珍(1976－),女,博士生,讲师,CCF 会员,主要研究领域为软件复用,软件构件管理技术.

邵维忠(1946－),男,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程环境,面向对象方法,软件复用,软件构件技术.

邵凌霜(1979－),男,博士,讲师,主要研究领域为软件工程,软件构件技术.