

# 基于相似度的软件可信性评估方法

仲晓敏

(淮海工学院计算机工程学院, 江苏 连云港 222005)

**摘要:**为了提高软件可信评估的精确性,提出利用关联可信属性之间相似度的计算结果评估软件的可信性。首先建立可信证据模型、可信属性的数据结构和可信期望模型,便于数据的统一定制和下一步处理。然后对分定性可信属性、定量可信属性进行相似度的计算;根据各自的特点,定性可信属性利用 Jaccard 系数求解,定量属性则利用了相似度理论进行计算。最后,将该方法应用于实例进行验证,结果验证了它的有效性和稳定性。

**关键词:**软件可信性评估;可信证据;可信需求模型;相似度;Jaccard 系数

**中图分类号:**TP311      **文献标志码:** A

## Method for software trustworthiness evaluation based on similarity

ZHONG Xiao-min

(School of Computer Engineering, Huaihai Institute of Technology, Lianyungang 222005, China)

**Abstract:** In order to improve the accuracy of software trustworthiness evaluation, a credibility evaluation model based on similarity calculation between associated trusted attribute was demonstrated. The semantic similarity of corresponding qualitative trusted attribute values in the two different semantic tables were calculated by Jaccard coefficient. In the quantitative trusted attribute's similarity, Similar threshold values were set according to the standard value on the respective time points to avoid singular values, when trusted attribute values were dynamic. Finally, the main cause of the similarity between the credible evidence model and credible expectations model and its dynamics were analyzed. The method was implemented to evaluate the software, and the results showed the effectiveness and stability of the method for the software trustworthiness evaluation.

**Key words:** software trustworthiness evaluation; trust evidence; trust expected model; similarity; Jaccard coefficient

## 0 引言

根据 ISO 对软件质量的定义“软件质量是软件产品满足规定和隐含需求能力有关的所有特征和所有特性的总和”<sup>[1]</sup>,软件质量的关注重心从系统质量转移到使用质量。软件可信性对软件运行时的使用质量属性进行综合,决定了软件在实际环境执行时,满足明确和隐含要求的程度。目前,国内外都非常重视软件可信性,成为软件行业的研究热点。国

内的“国家中长期科技规划纲要”中多次出现关于可信性的论述;同时国家高技术发展(863)计划、国家重点基础研究发展(973)计划、国家自然科学基金委都多次设立或者批准立项软件可信性研究课题<sup>[2-4]</sup>。软件可信性研究需要针对软件运行环境的复杂性和不确定性来建立软件可信性度量及评估体系<sup>[5-6]</sup>。譬如文献[7]提出基于 SOA 软件系统可信性评价方法,从软件体系结构角度研究可信性评价问题;文献[8]提出一种支持软件资源可信评估的框架;文献[9]提出一种需求驱动的软件可信性评

估及演化模型。考虑到软件可信性偏重于使用质量,本研究首先站在用户角度建立起可信期望模型,将其作为评估的标准。实际采集得到的可信证据模型与可信期望模型在建模方式上有很强的对应关系,并体现在两种模型的关联可信属性上。然后,利用相似度理论来计算可信证据模型和可信期望模型的相似度,据此对软件可信性进行评估和度量。

## 1 软件可信评估模型

### 1.1 可信证据

对软件可信性进行评估需要基于足够的可信证据。所谓可信证据,是指所有与软件相关的,具有能够反映该软件某种可信属性的数据、文档或其他信息<sup>[10]</sup>。它们必须具备的特征有如下3点<sup>[11-12]</sup>:客观性、关联性、可获得性。

对软件可信性的评估基于软件生命周期中各个阶段的表现和特征,因此在软件开发阶段、交付阶段、运行阶段都可采集到相应的可信证据。软件开发过程中的各类相关文档资料(代码、配置文件和动态链接库)、以及对程序源代码进行静态分析检测所获得信息;在软件交付时进行的自动化或手工审查结果数据;软件在运行时所表现出的性能、可能影响程序运行状态的相关对象;以上均可作为可信证据的采集来源。软件可信性在本质上是传统软件质量概念的延伸,它更关注使用层面的综合化的质量属性及其保障<sup>[13]</sup>。因此,主要的可信证据需要在软件运行时采集得到。利用软件运行时对象的状态、操作、时间等信息可以分析与对象状态、操作相关的安全性、可靠性、可用性、程序执行正确性等可信属性;而通过分析软件的操作序列来确定软件的运行过程是否被攻击过或者被篡改过可以分析出安全性。具体来说,譬如使用者的反馈、Web 软件的 QoS 数据等。

因为采集的来源和对象不同,可信证据之间无论是内容还是组织结构都必然存在差异。为了便于统一管理和更好地支持软件可信性评估,需要设计具有较高灵活度的可信证据模型。可信证据模型通过对证据信息数据结构的定义:一是为相关信息的采集提供接口;二是为信息的存储提供统一的形式。考虑到可信证据的多源性,本研究设计的开放性可信证据模型见图1,这种多层次树状结构的模型便于当捕获到新的异构证据时,通过生成树算法进行扩展,且在最大程度上满足了所有可信证据的结构一致性。

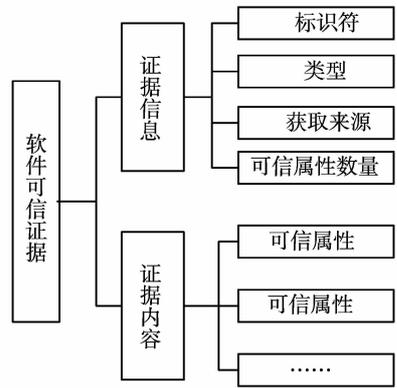


Fig. 1 Trust evidence model

开放性证据模型包括了证据信息和证据内容两部分。证据信息用四元组描述为  $(Evi\_ID, Evi\_Type, Evi\_Src, Evi\_Num)$ , 对应着证据模型的标识符、类型、获取来源, 以及具有的可信属性数目。Evi\_Src 的值域为枚举型集合, 包括了所有可能的证据源。Evi\_Type 取值为布尔型, 若为 True 表明证据内容嵌套有“子证据”, 若 False 则表明证据内容只包括若干个可信属性。证据内容分为可信属性名和对应的可信属性值。

### 1.2 可信属性及其数据结构的定义

上文构建的软件可信证据模型中, 叶子节点为“可信属性”。可信属性通过定义一组适当的软件属性达到描述软件行为及其产生结果的目的。德国达姆施塔特大学计算机科学和工程系的数据库和分布式系统组认为可信性是比任何其它特征都具有综合性的一个性质, 并总结出以下的可信属性: 端到端安全、可用性、可靠性、实效性、一致性、可预测性、可扩展性<sup>[14]</sup>。

AVIZIENIS A 等人定义的可信属性有可访问性、可靠性、安全性、完整性和可维护性; 微软公司定义的可信属性有保密安全性、私密性、可靠性和商务完整性<sup>[15]</sup>; 可以看出, 前者更注重软件系统避免产生不可接受服务故障的能力, 而后者更注重软件用户的需求。迄今为止, 软件可信属性还未有统一的标准, 但是大部分研究者较为认同的有: 可用性, 可靠性, 防危性, 安全性可维护性等属性。从以上所列的不同可信属性定义可看出, 软件可信性关注的是系统非功能特性的指标体系, 而且在不同环境下可信属性的内涵也会存在差异。因此, 在进行软件可信评估的过程中, 应该根据特定的应用背景和技术特点, 选择特定的软件可信属性集合来确定一个面向特定应用的软件可信性评估模型。本研究基于已有的研究, 综合考虑并定义了部分定量可信属性:

(1)实时性 待测软件从调用开始到运行结束,所开销的时间;以多次运行的平均时间进行度量。

$$R_t = \frac{\sum_{i=1}^n T_i}{n} \quad (1)$$

$R_t$  以秒为单位,其中  $n$  为测量次数,  $T_i$  表示每次测得的操作执行时间。

(2)可靠性 软件能够在规定的条件和时间间隔内正常运行,并输出与预期相符效果的概率。

$$S_r = N_s / K \quad (2)$$

$S_r$  取值区间为  $[0, 1]$ , 其中,  $N_s$  表示软件在可允许的时间限度内成功运行的次数,  $K$  则为运行的总次数。

(3)可用性 软件能够被调用时,成功的概率。

$$U_r = N_u * R_t / t \quad (3)$$

$U_r$  取值区间为  $[0, 1]$ , 其中,  $R_t$  表示软件在时间  $t$  内被成功的调用的次数。

(4)防危性 软件在规定的条件和时间内,避免发生灾难事故的概率。

$$R_s(t) = p(t \geq t_0) \quad (4)$$

$R_s(t)$  取值区间为  $[0, 1]$ , 规定时间  $t_0 > 0$ 。

(5)可移植性 软件对目标环境的适应能力。

$$D_p = 1 - \frac{COST_{port}}{COST_{redevelop}} \quad (5)$$

$COST_{port}$  表转移成本,  $COST_{redevelop}$  表示开发成本。容易看出,在转移成本较低时,软件的可移植性较高。

为了便于将以上可信属性进行统一的数字化存储,定义了属性的数据结构:

```

typedef struct TrustedAttr {
/* 可信属性的数据结构 */
    int ID;
    String name;
    int weight;
    String[] in_dimension; /* 维数 */
    String metricType; /* 属性值 */
}
    
```

所定义可信属性的数据结构包括属性名称、属性 ID、属性权重、属性值等。该数据结构虽然有一定的冗余度,但是有利于扩展,保证了可信属性定义的灵活性。

### 1.3 可信期望

可信期望模型是对用户的可信需求进行建模,是主体对客体进行评估的属性约束条件描述,也是

软件可信评估的目标对象。

为了与前文定义的证据模型具有一致性,可信期望模型采用了与其相类似的结构。可信期望模型也包括了信息和内容两部分。信息部分用四元组描述为  $(EviEx\_ID, EviEx\_Type, EviEx\_Post, EviEx\_Num)$ , 对应着模型的标识符、类型、发布者,以及期望的可信属性项数目。Evi\_Type 取值为布尔型,若为 True 表明该期望模型包括多个期望属性组合,此时 EviEx\_Num 取值大于 1;若 False 则表明该期望模型没有期望属性组合,此时 EviEx\_Num 取值可以等于 1。EviEx\_Post 为字符串型,其值域为可扩展的枚举型集合。内容部分分为期望属性,以及对应的期望值,每个期望属性的期望值为非空。

可信期望模型是软件用户根据自身实际需求进行构建的,是对软件可信性的需求;而可信证据模型是软件实际所能达到的可信效果。二者的审视角度不同,造成了可信期望模型和可信证据模型之间的差异,因此二者之间关联的建立很有必要。通常情况下,某个可信期望可能对应着多个可信证据,见图 2 所示。

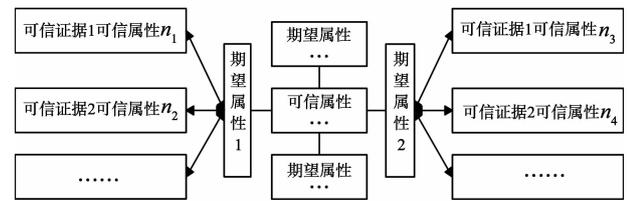


图2 可信期望模型与可信证据模型的关联  
Fig. 2 Relationship of expectation model and evidence model

## 2 可信性评估方法

可信证据分为定量与定性两种,本研究对于软件可信性的评估,就是通过对可信证据模型和可信期望模型相关联属性之间进行相似度评估<sup>[16-17]</sup>,从而达到可信性评估的目的。

### 2.1 定性可信属性的相似度计算

由图 2 可知可信期望模型上某些可信属性,对应着可信证据模型上多个可信属性。如果将所有与某个属性相关联的其他可信属性作为一个集合,则对该可信属性的评估,可以通过计算集合之间的相似度来完成。相关联的可信属性集合表示成两栏的语义表,分别由可信属性名和属性值构成。因此可用 Jaccard 系数计算两个不同语义表中相对应的可信属性值的语义相似度:

$$Vsim(v_1, v_2) = \left( \sum_{i=1}^m J(S_1, bug_i, S_2, Bag_i) \right) \setminus m \quad (6)$$

其中,  $v_1, v_2$  是同一可信属性在可信期望模型和可信证据模型上的不同取值。  $S_1, S_2$  分别对应  $v_1, v_2$  的语义表;  $bug_i$  是语义表中属性  $A_i$  对应的值;  $J()$  是 Jaccard 系数:

$$J(A, B) = \frac{A \cap B}{A \cup B} \quad (7)$$

利用上述方法,可计算出同一属性所关联的不同可信属性之间的一个相似度矩阵。由于  $Vsim(v_1, v_2) = Vsim(v_2, v_1)$ , 因此该矩阵是对称的, 只需计算上半矩阵, 相似度矩阵的创建算法如下:

输入 定性的可信属性数目为  $n$ , 相关联的属性数目为  $m$ , 属性的值为 bags

输出 相似度矩阵 **Matrix**

**Matrix**  $x \leftarrow \phi$

for  $i = 1 \cdots n - 1$  do

$i$ bags = getAttributeBags( $i$ );

    for  $j = i + 1 \cdots n$  do

$j$ bags = getAttributebags( $j$ );

        for  $k = 1 \cdots m$  do

$Sim(k) = Sim(ibags[k], jbags[k]);$

        End for

$SimDegree = (\sum_{k=1}^m Sim[k]) \setminus m$

**Matrix**[ $i$ ][ $j$ ] =  $SimDegree$ ;

**Matrix**[ $j$ ][ $i$ ] = **Matrix**[ $i$ ][ $j$ ];

    End for

End for

return **Matrix**

## 2.2 定量可信属性的相似度计算

对于定量型可信属性之间的评估,使用相似理论的基本思想。相似理论主要思想是将软件的可信用该系统的相似度来描述<sup>[18-19]</sup>,通过计算可信期望模型与可信证据模型的相似度,最终得到软件的可信度。

可信证据模型的相似度是由可信证据模型与可信期望模型的关联可信属性数量和可信属性对系统相似度影响权系数等因素决定的。设证据可信模型为  $A$ , 可信期望模型为  $B$ ,  $A, B$  之间存在  $n$  个关联可信属性,不妨设每个相似元值为  $q_i$ , 每一相似元对相似程度的影响权重为  $w_i$ , 则系统  $A$  与  $B$  的相似度可以定义为

$$Q(A, B) = \sum_{i=1}^n w_i q_i, \quad (8)$$

其中相似元  $q_i$  的计算分两种情况。当可信属性获得的是静态值,则可以设可信证据属性值为  $y_s$ , 对应的可信期望属性值为  $y_r$ , 则:

$$q_i = \frac{y_r - |y_r - y_s|}{y_r} \quad (9)$$

当可信属性获得的是动态值,则可以设可信证据属性值为  $Y_r(t) = (y_{r1}, y_{r2}, \dots, y_{rm})$ , 对应的可信期望属性值为  $Y_s(t) = (y_{s1}, y_{s2}, \dots, y_{sn})$ 。则  $Y_r(t)$  和  $Y_s(t)$  可以视作时间序列,选取某个关键时间段( $t_1, t_2$ ), 计算期间各个时间点上的相似元值:

$$q_t = \frac{y_{rt} - |y_{rt} - y_{st}|}{y_{rt}}, t \in (t_1, t_2), \quad (10)$$

其中  $Y_r(t)$  和  $Y_s(t)$  可以看做子系统,因此可以利用公式(8)计算出该时间区间段上的相似元值:

$$q_i = \sum_{t=t_1}^{t_2} w_t q_t, \quad (11)$$

其中  $w_t$  为时间点  $t$  上的权重。在上述计算过程中,在选定的时间段中可能出现奇异值,并将影响最终计算结果。因此,进行如下改进处理:以可信期望属性作为标准值,根据各个时间点上的标准值设置相似阈值  $\beta$ , 则  $Y_r(t) \notin (Y_s(t) - \beta, Y_s(t) + \beta)$  作为奇异值。根据  $Y_r(t)$  所在的不同区间,选择对应的百分比值作为权重,并记该权重为  $w_\beta$ 。若设  $Y_r(t) \in (Y_s(t) - \beta, Y_s(t) + \beta)$  的权重为  $\alpha$ , 且规定当  $Y_r(t) \gg Y_s(t)$  时,  $w_\beta = 0$ , 则有:

$$w_\beta = \begin{cases} 0, & Y_r(t) \gg Y_s(t); \\ \alpha, & Y_r(t) \in (Y_s(t) - \beta, Y_s(t) + \beta); \\ 1 - \alpha, & \text{else.} \end{cases} \quad (12)$$

如此一来,在处理动态可信属性值时,式(11)变换为

$$q_i = \sum_{t=t_1}^{t_2} w_t w_\beta q_t \quad (13)$$

一般情况下,可以假设各时间点上相似元的权重相同,即:

$$w_t = \frac{\Delta t}{t_2 - t_1} \quad (14)$$

上式中  $\Delta t$  为采样点的时间间隔,代入式(13),化简可得到:

$$q_i = \frac{\Delta t}{t_2 - t_1} \sum_{t=t_1}^{t_2} w_\beta q_t \quad (15)$$

## 3 应用实例及评估效果分析

因为“校园一卡通”是数字化校园建设的关键,与财务、教务等系统集成,可信性要求很高,所以选择“校园一卡通”软件作为可信性评估对象。进行可信性评估的分层操作如图3所示。

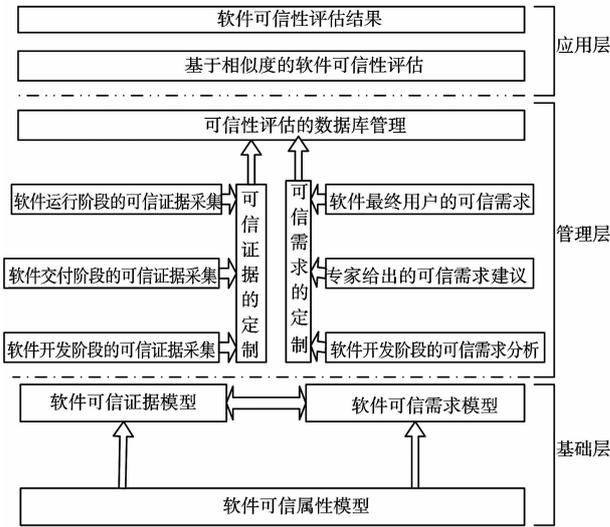


图3 软件可信分层操作图

Fig. 3 Stratified operation of software trustworthiness

在上文所创建的软件可信属性模型、可信证据模型和可信期望模型的基础上,与可信性评估的分层操作图相对应,评估步骤如下:

(1)可信性需求的分析和可信需求的定制。软件可信需求的原始数据可通过专家座谈、客户访谈和调查收集得到。抽取可信需求对应的设计质量特性,并逐一分析设计质量特性之间的相互影响关系。根据软件可信期望模型对所得原始数据进行定制,生成可信期望实例。

(2)可信证据的采集。在软件生命周期的各个阶段采集可信证据,对定性可信属性和定量可信属性的内涵和外延进行分析,并根据可信证据模型进行定制,生成可信证据实例。

(3)软件可信性评估。调用本研究提出的算法进行评估,根据改进相似度理论计算得到的结果给出最终评估结果。

在可信证据的采集步骤中,考虑软件必须保证系统长时间无故障运行,即便遭遇事故造成中断后也能确保数据的准确性、完整性和一致性,并可迅速恢复正常,故可靠性是最为重要的可信属性之一。此外,实时性、可用性、防危性、可移植性等都不可忽略。利用效用函数实时性进行统一量化:

$$U_{RT}(t) = \begin{cases} U_{RT0}, & 0 < t < t'_0; \\ (t - t_0)^2, & t'_0 < t < t_0; \\ 0, & t > t_0, \end{cases} \quad (16)$$

其中  $U_{RT0}$  为最优效用值,用户可接受的最大响应时间为  $t_0$ ,用户认为最优效用时的最大响应时间为  $t'_0$ 。可信评估即考虑了定量可信属性,又选用了定性可信属性;因为篇幅限制,表1 仅仅展示了部分定量可信属性的值。

表1 可信证据实例

Table 1 Trust evidence instances

序号	$U_i$	$U_{RT}$	$S_r$	$U_r$	$R_s$	$D_p$
实例1	0.21	0.913	0.564	0.971	0.536	0.699
实例2	0.64	0.870	0.978	0.349	0.987	0.979
实例3	0.72	0.953	0.782	0.535	0.898	0.614
实例4	0.69	0.764	0.659	0.845	0.896	0.684
实例5	0.53	0.168	0.469	0.546	0.656	0.699

对定性和定量可信属性分别采用相应的相似度计算方法进行处理;然后,将得到的各个属性相似度值乘以专家给定的可信属性权重值,得到该软件的最终可信属性评估值;评估值越接近1,则可信属性越高。使用本研究方法对同一软件进行了3次评估,得到的评估值分别为0.9219,0.9196,0.9251,平均值为0.9292;从3次评估的相近值可得知本研究提出的方法具有较强的稳定性。利用 JOANNE S Kaczmarek 所提出的方法<sup>[20]</sup>对该软件进行3次评估,得到的评估值分别为0.8904,0.9432,0.9525,平均值为0.9287。两种方法3次评估结果的均值相近;使用式(17)平均绝对偏差(MAD)计算方法对这两种方法的评估结果进行分析,得到表2。

$$\beta(j) = \frac{\sum |e_j|}{\sum j}, \quad (17)$$

其中平均绝对偏差  $\beta(j)$  值作为评估稳定性衡量指标,其值越接近0,则评估稳定性越高。 $e_j = u_j - u_a$ ,  $u_j$  为第  $j$  次评估值,  $u_a$  是多次评估的平均值。 $\sum j$  为评估总次数。

表2 评估结果稳定性分析

Table 2 Analysis for evaluation stbleness

算法	评估次数	$u_j$	$u_a$	$\beta(j)$
本研究方法	$j=1$	0.9219	0.9292	0.0070
	$j=2$	0.9196		
	$j=3$	0.9251		
文献[20]方法	$j=1$	0.8904	0.9287	0.0255
	$j=2$	0.9432		
	$j=3$	0.9525		

由表2可知,本研究方法比文献[20]方法具有更强的稳定性。

## 4 结论

在软件可信性研究中,可信性评估结果在软件正式交付使用之前能够帮助发现问题,以及驱动软件的进一步完善,所以越来越受到重视。在吸取已有研究成果经验的基础上,本研究构建了软件可信属性模型、可信证据模型和可信期望模型。其中,可

信期望模型表达了最终用户的需求,并吸纳了专家的建议和历史数据的分析结果,可以作为软件可信性的衡量标准。而可信证据模型的合理性创建,便于将采集到的各类可信证据进行统一定制。在评估过程中,注重了可信证据模型和可信期望模型的可信属性之间关联,避免了现有研究中大多只关注当个可信属性的弊端。软件可信性评估考虑到可信属性分为定量型和定性型,对于定性可信属性用 Jaccard 系数求解关联属性集合的相似度,而对于定量可信属性则使用了改进的相似度理论来计算。

通过本研究所提出方法在“校园一卡通”软件的可信性评估中的应用,多次评估结果相近,而且与现有评估方法所得结果相同。说明本研究提出评价方法的有效性,并在综合多可信属性方面有所改进,方法具有较强的稳定性。但是,本研究中各个可信属性的权重值由专家给出,有很强的主观性,在下一步工作中需要对各个可信属性的权重赋值进行研究。

#### 参考文献:

- [1] 中华人民共和国信息产业部. GB/T 16260.1-200X 软件工程产品质量[S]. 北京:中国标准出版社,2006.
- [2] 科技部 863 计划信息技术领域办公室. 国家高技术研究发展计划(863 计划)信息技术领域“高可信软件生产工具及集成环境”重点项目 2009 年度课题申请指南[EB/OL]. (2009-02-16) [2012-02-18] [http://www.most.gov.cn/tztg/200904/t20090409\\_68548.htm](http://www.most.gov.cn/tztg/200904/t20090409_68548.htm).
- [3] 孙家广. 现代设计大型应用软件的可信性研究[EB/OL]. (2009-07-23) [2012-02-18]. <http://www.973.gov.cn/Arealtem.aspx?fid=03>.
- [4] 国家自然科学基金委员会. 重大研究计划“可信软件基础研究”2010 年度项目指南[EB/OL]. (2009-12-23) [2012-02-18] [http://www.nsf.gov.cn/nsfc/cen/yjih-new/2010/20100419\\_02.htm](http://www.nsf.gov.cn/nsfc/cen/yjih-new/2010/20100419_02.htm).
- [5] 王远,吕建,徐锋,等. 一个适用于网构软件的信任度量及演化模型[J]. 软件学报,2006,17(4):682-690.  
WANG Yuan, LÜ Jian, XU Feng, et al. A trust measurement and evolution model for Internetware [J]. Journal of Software, 2006, 17(4):682-690.
- [6] 黄茗云. 贝叶斯网络在软件可信性评估指标体系中的应用[D]. 济南:山东轻工业学院信息科学与技术学院,2009.  
HUANG Mingyun. The application of Bayesian network in evaluation index system of software dependability[D]. Jinan: School of Information, Shangdong Institute of Light Industry, 2009.
- [7] 赵会群,孙晶. 一种 SOA 软件系统可信性评价方法研究[J]. 计算机学报,2010,33(11):2202-2210.  
ZHAO Huiqun, SUN Jing. A methodological study of evaluating the dependability of SOA software system [J]. Chinese Journal of Computers, 2010, 33(11):2202-2210.
- [8] 蔡斯博,邹艳珍,邵凌霄,等. 一种支持软件资源可信评估的框架[J]. 软件学报,2010,21(2):359-372.  
CAI Sibo, ZOU Yanzhen, SHAO Lingshuang, et al. Framework supporting software assets evaluation on trustworthiness [J]. Journal of Software, 2010, 21(2):359-372.
- [9] 丁帅,鲁付俊,杨善林,等. 一种需求驱动的软件可信性评估及演化模型[J]. 计算机研究与发展,2011,48(4):647-655.  
DING Shuai, LU Fujun, YANG Shanlin, et al. A requirement-driven software trustworthiness evaluation and evolution model [J]. Journal of Computer Research and Development, 2011, 48(4):647-655.
- [10] Trusted Computing Group. TCG specification architecture overview [EB/OL]. (2006-11-10) [2012-02-18]. [https://www.trustedcomputinggroup.org/groups/TCG\\_1\\_4\\_Architecture\\_Overview.pdf](https://www.trustedcomputinggroup.org/groups/TCG_1_4_Architecture_Overview.pdf).
- [11] 郭树行,兰雨晴,金茂忠. 软件构件的可信保证研究[J]. 计算机科学,2007,34(5):243-246.  
GUO Shuxing, LAN Yuqing, JIN Maozhong. Some issues about trusted components research [J]. Computer Science, 2007, 34(5):243-246.
- [12] Trustie 课题组. Trustie 系列技术规范(V2.0) [EB/OL]. (2009-05-30) [2012-03-09]. <http://www.trustie.net/download/STEF2-0.pdf>
- [13] BABAMIR S M, JALILI S. Synthesizing a specification-based monitor for safety requirements [J]. Iranian Journal of Science and Technology, Transaction B: Engineering, 2010, 34(3):235-256.
- [14] 梅宏,曹东刚. 软件可行性:互联网带来的挑战[J]. 中国计算机学会通讯,2010,6(2):20-27.  
MEI Hong, CAO Donggang. Reliability of software: the challenges of the internet [J]. Communication of the CCF, 2010, 6(2):20-27.
- [15] 陈火旺,王戟,董威. 高可信软件工程技术[J]. 电子学报,2003,31(12A):1933-1938.  
CHEN Huowang, WANG Ji, DONG Wei. High confidence software engineering technologies [J]. Acta Electronica Sinica, 2003, 31(12A):1933-1938.
- [16] 汤永新,刘增良. 软件可信性度量模型研究进展[J]. 计算机工程与应用,2010,46(27):12-16,42.  
TANG Yongxin, LIU Zengliang. Progress in software trustworthiness metrics models [J]. Computer Engineering and Applications, 2010, 46(27):12-16,42.
- [17] TEKLI J, CHBEIR R, YETONGNON K. An overview on XML similarity: background, current trends and future directions [J]. Computer Science Review, 2009, 3(3):151-173

- ization using test information [J]. *Journal of Computer Science and Technology*, 2009, 24(5):962-974.
- [7] WONG W E, WEI T, QI Y, et al. A crosstab-based statistical method for effective fault localization [C]//Proceedings of the 1st International Conference on Software Testing, Verification and Validation. Lillehammer: IEEE Computer Society, 2008:42-51.
- [8] XIE X, WONG W E, CHEN T Y, et al. Spectrum-based fault localization without test oracles [C]//Proceedings of the 11st International Conference on Quality Software (QSIC). Xi'an: IEEE Computer Society, 2011:1-10.
- [9] ZHANG X, GUPTA N, GUPTA R. Locating faults through automated predicate switching [C]//Proceedings of the 28th International Conference on Software Engineering. ShangHan: IEEE Computer Society, 2006:272-281.
- [10] TUCEK J, LU S, HUANG C, et al. Triage-diagnosing production run failures at the user's site [C]//Proceedings of the 21st ACM Symposium on Operating Systems Principles. Stevenson: Association for Computing Machinery, 2007:131-134.
- [11] CHEN M, KICIMAN E, FRATKIN E, et al. Pinpoint: Problem determination in large, dynamic internet services [C]//Proceedings of the 32nd IEEE/IFIP International Conference on Dependable Systems and Networks. Los Alamitos: IEEE Computer Society, 2002:595-604.
- [12] JONES J, HARROLD M, STASKO J. Visualization of test information to assist fault localization [C]//Proceedings of the 24th International Conference on Software Engineering. Orlando: IEEE Computer Society, 2002: 467-477.
- [13] ABREU R, ZOETEWIJ P, VAN GEMUND A. On the accuracy of spectrum-based fault localization [C]//Proceedings of the Testing: Academic and Industrial Conference Practice and Research Techniques — MUTATION. Windsor: IEEE Computer Society, 2007:89-98.
- [14] WONG W E, DEBROY V. A survey of software fault localization [R]. Dallas, UTD: Department of Computer Science, 2009.
- [15] MACQUEEN J B. Some methods for classification and analysis of multivariate observations [C]//Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley: University of California Press, 1967:281-297.
- [16] HARTIGAN J A, WONG M A. A *k*-means clustering algorithm [J]. *Applied Statistics*, 1979, 28(1):100-108.
- [17] REPS T, BALL T, DAS M, et al. The use of program profiling for software maintenance with applications to the year 2000 problem [C]//Proceedings of the 6th European Software Engineering Conference Held Jointly with the 5th ACM SIGFOFT International Symposium on Foundations of Software Engineering. Zurich: Springer, 1997:432-449.
- [18] HARROLD M, ROTHERMEL G, SAYRE K, et al. An empirical investigation of the relationship between spectra differences and regression faults [J]. *Software Testing Verification and Reliability*, 2000, 10(3):171-194.
- [19] GAMMA E, BECK K, Junit [CP/OL]. Eugene, USA: University of Oregon, 2008.

(编辑:陈燕)

(上接第 18 页)

- [18] LI Dengfeng. Some measures of dissimilarity in intuitionistic fuzzy structures [J]. *Journal of Computer and System Sciences*, 2004, 68:115-122.
- [19] HANEL M, BUIHAND T A. On the value of hourly precipitation extremes in regional climate model simulations [J]. *Journal of Hydrology*, 2010, 393(3-4):265-273.
- [20] IMMONEN A, PALVIAINEN M. Trustworthiness evaluation and testing of open source components [C]//Proceedings of the Seventh International Conference on Quality Software. Los Alamitos: IEEE Computer Society Press, 2007:316-321.

(编辑:陈燕)