IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 25, 2018, accepted December 22, 2018, date of publication January 11, 2019, date of current version May 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2892518

Updating Model of Software Component Trustworthiness Based on Users Feedback

BAOHUA WANG[1,2], YIXIANG CHEN[1], SHUN ZHANG[3], AND HENGYANG WU[1]

[1]MoE Engineering Center for Software/Hardware Co-Design Technology and Application, East China Normal University, Shanghai 200062, China
[2]School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China
[3]Office of Foreign Cooperation and Exchange, Guizhou Academy of Sciences, Guiyang 55001, China

Corresponding author: Yixiang Chen (yxchen@sei.ecnu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China through the Research and Application Demonstration of Information Technology in Food Safety Society under Grant No. 2017YFC1601800, in part by the Natural Science Foundation of Anhui Province through Research on Quantitative Model of Software Approximate Accuracy Based on Environment Interaction under Grant 1708085MF159, and in part by the Provincial Projects of Natural Science for Anhui Universities through Research on 2/3 Approximate Accuracy of Software under Grant KJ2017A375.

ABSTRACT The software trustworthiness measurement is one of the hot topics. Software component technology is the mainstream technology of software development. How to get the trustworthy degree of software component efficiently and accurately is a challenging issue for the component-based software development. Getting the trustworthy degree of software component needs many users' success cases. In this paper, we propose an updating model of software component trustworthiness. First, the trustworthy degree of the software component is computed based on users' feedback. Then, the weight of updating is determined by the number of users. Finally, the method of cluster different companies is based on the Euler distance. A case study shows that the method is reasonable and effective.

INDEX TERMS Software trustworthiness, software component, weight, updating model, user feedback.

I. INTRODUCTION

Software in some safety-critical areas, such as aerospace control, finance, transportation and communication, need to achieve higher trustworthy degree [1], [2]. The trustworthiness of software means that the behavior and the running results of the software should be in line with expectations [3], [4]. The measurement of software trustworthiness can provide quantitative evaluation for the trustworthiness of software. The quantification of software trustworthiness has attracted more researchers' attentions and has become a hot topic [5]. Usually, this quantification, i.e., trustworthy degree of software, is determined by the trustworthy degree of attributes(for example, reliability, correctness and security) and the weights of attributes that reflect the important degree of attributes. Many models for trustworthy degree are given. German Oldenburg University studies the structure of the trustworthy software, the evaluation of the software trustworthiness and verification techniques [6]. Voas [7] divides the trustworthiness of software into many attributes according to the functional requirements. Lang *et al.* [8] introduce a hierarchy model for software trustworthiness classification based on the analysis of software trustworthiness connotation. Shanlin *et al.* [9] give a software trustworthiness model based on software utility and evidence theory. Tao and Chen [3] propose a model based on multi-dimensional attributes [4]. Later Ma and Chen [5] simplified it as follows:

$$\begin{cases} T = \prod_{i=1}^{n} y_i^{w_i}, \\ \sum_{i=1}^{n} w_i = 1, \end{cases}$$

where T is the trustworthy degree of the software, y_i is the trustworthy degree of the i-th attribute and w_i is the weight of the i-th attribute. This model has properties like monotonicity, acceleration and sensitivity etc. See [3] for details.

Component-Based Software Development (CBSD) [10], [11] is the mainstream technology of developing software. CBSD avoids duplication of effort, reduces the development cost and improves productivity. By reusing software component, CBSD avoids repeated emergence of errors and improves the trustworthiness of software. The software component has several definitions. CMU/SEI defines software component as an opaque implementation of functionality, subject to third-party's composition and conformation with a component model [13]. Szyperski's definition: software component is a unit of composition with contractually specified interfaces and explicit context dependencies [14].

Reference [15] defines a software component as units which can be deployed independently subject to composition by third parties. References [16] and [17] adopt the Szyperski's definition. In this paper, we also adopt Szyperski's definition.

The trustworthy degree of software component should be measured accurately. Getting the trustworthy degree of software component needs many users' success cases. In this paper, we propose the updating model of software component's trustworthy degree. First, the trustworthy degree of software component is computed based on users' feedback. Then, the weight of updating is determined by the number of users. The method of cluster different companies is based on Euler distance. Finally, a case study is presented, which reflects that the proposed method is effective.

The remainder of this paper is organized as follows. In section 2, we propose the updating model of software component trustworthiness. Section 3 gives a case study. Section 4 is the conclusion section.

II. UPDATING MODEL OF SOFTWARE COMPONENT TRUSTWORTHINESS BASED ON USERS FEEDBACK

The "users" of users feedback refer to the software companies that they use software component to develop software. On one hand, because the software companies are not the developer of the software component, they can evaluate fairly the trustworthy degree of software component. On the other hand, the software companies have a very good professional ability to evaluate it. The updating model is

$$\begin{cases} T_n = w_o \times T_o + w_u(n) \times T_u \\ w_o + w_u(n) = 1 \end{cases}$$

where T_n is the updating trustworthy degree of software component, T_u is the trustworthy degree of users' feedback, T_o is the origin trustworthy degree, w_o is the weight of origin trustworthy degree, n is the number of the users and $w_u(n)$ is the weight of users' feedback.

A. CONSTRUCTION OF THE WEIGHT FUNCTION $w_u(n)$

The weight $w_u(n)$ is related to the number of users. When the software companies use the software component, we use $\frac{w_u(n+x) - w_u(n)}{x}$ to represent the average weight of the x users after n users. λ is the feedback influence ratio of the users, and different software components can have different λ values. The λ is positive correlation with $w_u(n)$,

$$\lambda = \frac{w_u(n+x) - w_u(n)}{x \times (1 - w_u(n))}.$$

In order to solve the function $w_u(n)$, we suppose the $w_u(n)$ as a continuous function and solve the limit of λ

$$\lambda = \lim_{x \to 0} \frac{w_u(n+x) - w_u(n)}{x \times (1 - w_u(n))}.$$

Further, we get the formula

$$\frac{dw_u(n)}{dn} = \lambda \times (1 - w_u(n)). \tag{1}$$

The formula (1) can be converted into the formula (2)

$$\frac{dw_u(n)}{dn} + \lambda \times w_u(n) = \lambda. \tag{2}$$

We firstly solve the general solution of the corresponding homogeneous equation to the formula (2). We get that

$$\frac{dw_u(n)}{dn} + \lambda \times w_u(n) = 0. \tag{3}$$

We solve the differential equation (3) by variable separation method.

$$w_u(n) = C \times e^{-\lambda \times n}. \tag{4}$$

Using the method of variation of parameters, we replace the C of the equation (4) with the unknown function $u(x)$, and the formula (4) is converted into the equation (5)

$$w_u(n) = u \times e^{-\lambda \times n}. \tag{5}$$

Form the equation (5), we have that

$$\frac{dw_u(n)}{dn} = u' e^{-\lambda \times n} - \lambda \times u \times e^{-\lambda \times n}. \tag{6}$$

We put the equation (5) and (6) into the equation (2), and obtain the equation (7)

$$u' e^{-\lambda \times n} - u \times \lambda \times e^{-\lambda \times n} + \lambda \times u \times e^{-\lambda \times n} = \lambda. \tag{7}$$

Solving the equation (7), we get

$$u = e^{\lambda \times n} + c_1. \tag{8}$$

We put the equation (8) into the equation (5), and obtain the equation (9)

$$w_u(n) = 1 + c1 \times e^{-\lambda \times n}. \tag{9}$$

We combine the initial conditions of $w_u(0) = 0$, and obtain the function of the wight

$$w_u(n) = 1 - e^{-\lambda \times n}. \tag{10}$$

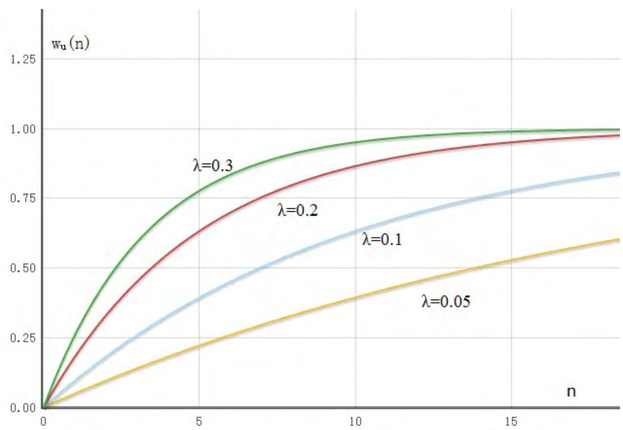


FIGURE 1. The relation of the wight function and λ.

The relation of the weight function and λ is shown in figure 1.

The weight function has the following propositions.

Proposition 1 (Monotonicity): With the increasing of the users' number n, the value of the function is monotone increasing.

Proof: From the equation (10), we have that

$$\frac{dw_u(n)}{dn} = (1 - e^{-\lambda \times n})' = \lambda e^{-\lambda \times n} > 0.$$

So, the function is monotone increasing.

Proposition 2 (Acceleration): With the increasing of the users' number n, the increasing rate of the function value is becoming small.

Proof: From the equation (10), we get that

$$\frac{dw_u^2(n)}{dn^2} = \frac{d(\frac{dw_u(n)}{dn})}{dn} = \frac{d(\lambda e^{-\lambda \times n})}{dn} = -\lambda^2\, e^{-\lambda \times n} < 0.$$

So, the increasing rate of the function value is becoming small.

Because we have that

$$w_u(0) = 1 - e^{-\lambda \times 0} = 0.$$

The number of the user feedback is 0, the value of $w_u(n)$ is 0.

Because we have that

$$\lim_{n \to +\infty} (1 - e^{-\lambda \times n}) = 1.$$

The number of user feedback is $+\infty$, the value of $w_u(n)$ is closed to 1.

B. THE TRUSTWORTHY ATTRIBUTES OF SOFTWARE COMPONENT

Meyer [18] introduces the "ABCDE" model of trusted component. The characters of component trustworthiness are called attributes in this paper, shown as table 1.

Acceptance is a non-technical dimension: a provider may claim a component is reusable, but it helps to have evidence of usage.

Behavior addresses a key property of a good component: that it should be equipped with a precise list and individual specification of the functionalities it offers.

Constraints, covers performance considerations. Here we no longer have a scale, just a set of partly independent criteria, such as response time, security , bandwidth requirements, ease of use.

Design considers the internal perspective. What you want as a component consumer is a guarantee of the component's external properties-properties characterizing what it makes available to you-but any information on the principles and techniques that the component authors have used during development can reinforce your trust.

Extension addresses an important feature of software reuse: it's not only about reusing components as they are, but also adapting a component to your specific needs.

This classification provides a first grid of criteria against which to assess components.

TABLE 1. The trustworthy attributes of software component.

Trustworthiness	Trustworthy attribute	Sub-attribute
The trustworthy degree of the software component	Acceptance	Some reuse attested
		Producer reputation
		Published evaluations
	Behavior	Examples
		Usage documentation
		Preconditioned
		Some postconditions
		Full postconditions
		Observable invariants
	Constraints	Platform spec
		Ease of use
		Response time
		Memory occupation
		Bandwidth
		Availability
		Security
	Design	Precise dependency documentation
		Consistent API rules
		Strict design rules
		Extensive test cases
		Some proved properties
		pre and post conditions, invariants
	Extension	Portable across platforms
		Mechanisms for addition
		Mechanisms for redefinition
		User action pluggability

C. THE CLASSIFICATION OF TRUSTWORTHY DEGREES

The classification for trustworthy degrees is the quantitative scale. The length of trustworthiness classification are not fixed. The higher the trustworthy degree is, more difficult to improve it. The paper proposes a method of classification where the length of the $i+1$-th grade is m times the length of the i-th grade, where $0 < m < 1$.

If the length of the first grade is x, the sequence of the lengths is $x, mx, m^2x, \cdots, m^{n-1}x$. It is true that $x + mx + m^2x + \cdots + m^{n-1}x = (1 + m + m^2 + \cdots + m^{n-1})x = 1$. For example, the length of the $i+1$-th is $(\sqrt{5}-1)/2$ times the length of the i-th level. The number of the grade is five. We get that

$$x + (\sqrt{5}-1)/2 \times x + \cdots + ((\sqrt{5}-1)/2)^4 \times x = 1. \quad (11)$$

From the equation (11), we get that x is 0.45. The lengths of the grades are computed by the following equation.

$$\begin{aligned} 0.45 \times (\sqrt{5}-1)/2 &= 0.25, \\ 0.45 \times ((\sqrt{5}-1)/2)^2 &= 0.15, \\ 0.45 \times ((\sqrt{5}-1)/2)^3 &= 0.10, \\ 0.45 \times ((\sqrt{5}-1)/2)^4 &= 0.05. \end{aligned}$$

D. MEASUREMENT METHOD OF SUB-ATTRIBUTE

The trustworthy measurement of software component is carried out from the bottom to the top. The trustworthy degrees of the sub-attributes are measured by the users' feedback of software component. The users give the trustworthy degrees

TABLE 2. The grade assess based on trusted evidence.

sub-attribute	Evaluation evidence	grades
some reuse attested	The reuse of the software component has verified formally.	L5
	The reuse of the software component has passed related reviews.	L4
	The reuse of the software component has cases.	L3
	The reuse of the software component meets relevant standards.	L2
	There is no the evidence for components reusable.	L1

TABLE 3. The symbolization of five grades.

grade	symbolization
L5	p
L4	$\neg p \bigwedge q$
L3	$\neg q \bigwedge r$
L2	$\neg r \bigwedge s$
L1	$\neg s$

TABLE 4. The logical relations of the propositions.

The logical relations
$p \rightarrow q \bigwedge r \bigwedge s$
$q \rightarrow r \bigwedge s$
$r \rightarrow s.$

to sub-attributes. In this paper, we propose a measurement method for sub-attributes, which is based on trusted.

The measurement based on trusted evidence is started with the second grade. The sub-attribute will get higher grade when we find more trusted evidences. If the sub-attribute does not meet the second grade, it is the first grade. If the sub-attribute meets some grade, it will be judged to meet the higher grade by the method. The sub-attribute will get a grade when it does not meet some grade or has got the highest grade.

The sub-attribute of "some reuse attested" is taken as an example of the measurement method based on trusted evidence, which is shown as table 2.

The propositions are performed the symbolization in the measurement based on trusted evidence.

p represents that the reuse of the software component has been verified formally.

q represents that the reuse of the software component has passed related reviews.

r represents that the reuse of the software component has cases.

s represents that the reuse of the software component meets relevant standards.

The grades are represented with the symbolization, and are shown as following table 3. The logical relations of the propositions are shown as table 4.

The method of measurement based on trusted evidence has two propositions.

Proposition 3 (Completeness): The union of the all grades is true.

Proof: We have the logical relationship of table 4. So, we get that

$$
\begin{aligned}
& L5 \bigvee L4 \bigvee L3 \bigvee L2 \bigvee L1 \\
& \quad = p \bigvee (\neg p \bigwedge q) \bigvee (\neg q \bigwedge r) \bigvee (\neg r \bigwedge s) \bigvee \neg s \\
& \quad = q \bigvee (\neg q \bigwedge r) \bigvee (\neg r \bigwedge s) \bigvee \neg s \\
& \quad = r \bigvee (\neg r \bigwedge s) \bigvee \neg s \\
& \quad = s \bigvee \neg s \\
& \quad = T.
\end{aligned}
$$

The union of the all grades is true.

Proposition 4 (Mutually Exclusive): The intersection of any two grades is false.

Proof: Because $p \rightarrow q \bigwedge r \bigwedge s$, we have that

$$
\begin{aligned}
& L5 \bigwedge L3 \\
& \quad = p \bigwedge (\neg q \bigwedge r) \\
& \quad = q \bigwedge r \bigwedge s \bigwedge (\neg q \bigwedge r) \\
& \quad = F.
\end{aligned}
$$

So, the intersection is false.

The other intersections can be prove by the same method.

TABLE 5. The trustworthy degree of sub-attributes given by five companies.

User	The attributes of components trustworthiness								
	Acceptance			Behavior					
zxy	L5	L4	L3	L4	L3	L4	L5	L3	L4
hxj	L5	L4	L3	L4	L4	L4	L4	L4	L5
ych	L5	L5	L3	L5	L4	L5	L5	L3	L5
wh	L3	L5	L3	L5	L3	L5	L3	L4	L4
lm	L5	L4	L5	L3	L5	L4	L5	L5	L3

User	The attributes of components trustworthiness						
	Constraints						
zxy	L5	L5	L4	L4	L3	L4	L5
hxj	L5	L5	L4	L3	L3	L5	L4
ych	L5	L3	L5	L4	L5	L4	L5
wh	L4	L4	L5	L5	L4	L5	L4
lm	L4	L5	L4	L5	L3	L5	L4

User	The attributes of components trustworthiness									
	Design						Extension			
zxy	L5	L5	L4	L4	L5	L4	L4	L4	L5	L5
hxj	L5	L4	L5	L5	L4	L3	L4	L4	L5	L3
ych	L3	L5	L5	L3	L5	L4	L5	L5	L4	L5
wh	L5	L3	L4	L5	L5	L4	L4	L5	L5	L4
lm	L4	L5	L4	L4	L5	L4	L5	L4	L3	L5

III. CASE STUDY

We developed a software component C-PAY of payment function. It can be measured by the method based on trusted. After the software component is used by the five companies for a period of time, the trustworthy degrees of the sub-attribute are shown in table 5.

The trustworthy degrees of five grades are show in table 6.

TABLE 6. The trustworthy degrees of five grades.

grade	score
L5	1
L4	0.95
L3	0.85
L2	0.70
L1	0.45

A. CLUSTERING THE TRUSTWORTHY DEGREES OF THE SUB-ATTRIBUTES GIVEN BY DIFFERENT COMPANIES

In order to synthesize different companies opinions and obtain an overall trustworthiness degree of software component, we need to cluster the trustworthy degrees of the sub-attributes. We proposal a cluster approach based on the Euler distance, which is called ED approach. The approach is $s^*_{ij} = (s^{(1)}_{ij})^{ws_1} \times \ldots \times (s^{(r)}_{ij})^{ws_r} \times \ldots \times (s^{(m)}_{ij})^{ws_m}$. where ws_r is the accepted degree of the r-th company, s^*_{ij} is the synthesize trustworthy degree of the i-th attribute's j-th sub-attribute, $s^{(r)}_{ij}$ is the trustworthy degree of the i-th attribute's j-th sub-attribute given by the r-th company.

We will introduce an approach to computing ws_r, which is divided into four steps.

Step 1: Computing the geometric mean of the trustworthy degree of sub-attribute in use $\overline{s_{ij}} = (s^{(1)}_{ij} \times \ldots \times s^{(r)}_{ij} \times \ldots \times s^{(m)}_{ij})^{\frac{1}{m}}$.

Step 2: Calculating the Euler distance between the trustworthy degrees given by the r-th company $s^{(r)}_{ij}$ and the geometric mean $\bar{s_{ij}}$ as follows:

$$d_r = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(a^{(r)}_{ij} - \overline{a_{ij}})^2}.$$

Step 3: Obtaining the sum of $\frac{1}{d_r}$,

$$sum = \sum_{r=1}^{m}\frac{1}{d_r}(d_r > 0).$$

Step 4: Computing $ws_r = \frac{\frac{1}{d_r}}{sum}$.

According to algorithm 2.1, ws_1, ws_2, ws_3, ws_4 and ws_5 are 0.2715,0.2062,0.1748,0.1815,0.1660 respectively. Adopting the ED method, we get $s^*_{ij} = (s^{(1)}_{ij})^{0.2715} \times (s^{(2)}_{ij})^{0.2062} \times (s^{(3)}_{ij})^{0.1748} \times (s^{(4)}_{ij})^{0.1815} \times (s^{(5)}_{ij})^{0.1660}$.

The common cluster methods are the geometric mean and the arithmetic mean. The geometric mean(GM) method gets $\overline{s_{ij}} = (s^{(1)}_{ij} \times \ldots \times s^{(r)}_{ij} \times \ldots \times s^{(m)}_{ij})^{\frac{1}{m}}$.

The arithmetic mean(AM) method obtains $s^a_{ij} = \frac{1}{m} \times (s^{(1)}_{ij} + \ldots + s^{(r)}_{ij} + \ldots + s^{(m)}_{ij})$.

The trustworthy degrees of sub-attributes of tree methods are shown in table 7.

Figure 2 and 3 are the comparisons of the some sub-attributes's trustworthy degrees given by five companies and the tree methods.

From the figure 2, we get that the trustworthy degree of the 'some reuse attested' sub-attribute in our method is bigger than the trustworthy degrees given by the GM and AM method. Because the trustworthy degrees of the sub-attribute given by four companies are big, Our method is more reasonable.

Algorithm 1 Calculation of ws_r

for($i = 1; i \leq n; i++$)
for($j = 1; j \leq n; j++$)
$\bar{s}_{ij} = ((s^{(1)}_{ij}) \times (s^{(2)}_{ij}) \times \cdots \times (s^{(m)}_{ij})^{\frac{1}{m}}$;
for($r = 1; r \leq m; r++$)
{for($i = 1; i \leq n; i++$)
$d_r = d_r + (s^{(i)}_{ij} - \bar{s}_{ij})^2$;
$d_r = \sqrt{(d_r)}$; }
for($r = 1; r \leq m; r++$)
if ($d_r \neq 0$)
$sum = sum + \frac{1}{d_r}$;
for($r = 1; r \leq m; r++$)
if ($d_r \neq 0$)
$ws_r = \frac{\frac{1}{d_r}}{sum}$;

TABLE 7. The trustworthy degrees of sub-attributes of tree methods.

Methods	The attributes Acceptance						
ED	0.971	0.968	0.873				
GM	0.968	0.97	0.878				
AM	0.97	0.97	0.88				
Methods	The attributes Behavior						
ED	0.95	0.911	0.968	0.961	0.912	0.951	
GM	0.948	0.918	0.97	0.958	0.918	0.948	
AM	0.95	0.92	0.97	0.96	0.92	0.95	
Methods	The attribute Constraints						
ED	0.982	0.963	0.968	0.945	0.892	0.977	0.972
GM	0.98	0.958	0.97	0.948	0.898	0.98	0.97
AM	0.98	0.96	0.97	0.95	0.98	0.97	0.97
Methods	The attribute Design						
ED	0.964	0.961	0.969	0.950	0.990	0.929	
GM	0.958	0.958	0.97	0.948	0.990	0.929	
AM	0.96	0.96	0.97	0.95	0.99	0.93	
Methods	The attribute Extension						
ED	0.967	0.968	0.965	0.958			
GM	0.97	0.97	0.950	0.950			
AM	0.91	0.97	0.96	0.96			

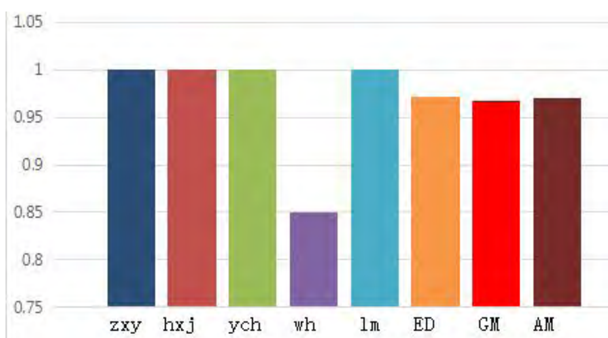


FIGURE 2. The trustworthy degree of the 'some reuse attested'.

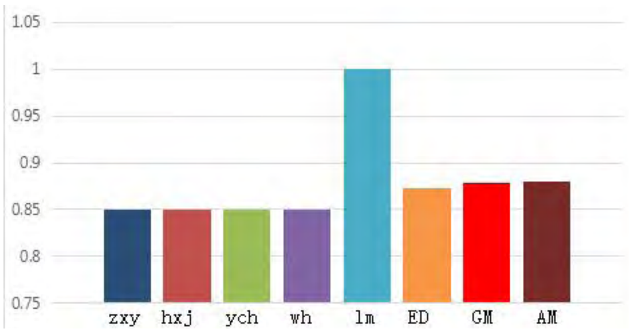


FIGURE 3. The trustworthy degree of the 'published evaluations'.

TABLE 8. Scaling.

scaling	definition
1	the i-th trustworthy attribute is as important as the j-th trustworthy attribute
3	The i-th trustworthy attribute is slightly more important than the j-th trustworthy attribute
5	The i-th trustworthy attribute is more important than the j-th trustworthy attribute
7	The i-th trustworthy attribute is much more important than the j-th trustworthy attribute
9	The i-th trustworthy attribute is absolutely more important than the j-th trustworthy attribute
2,4,6,8	The comparison of the i-th trustworthy attribute to the j-th trustworthy attribute is between the two adjacent grades.
$a_{ji} = \frac{1}{a_{ij}}$	The comparison of the j-th trustworthy attribute to the i-th trustworthy attribute is the reciprocal of a_{ij}

From the figure 3, we get that the trustworthy degree of the 'published evaluations' sub-attribute in our method is smaller than the trustworthy degrees given by the GM and AM method. Because the trustworthy degrees of the sub-attribute given by four companies are small, Our method is more reasonable.

Our method can better unify the opinions of the different companies. The trustworthy degrees of some sub-attributes given by different companies are similar or distributed dispersion, the cluster results of the three methods are also similar.

B. DETERMINING WEIGHTS OF ATTRIBUTES

We calculate the trustworthy degree of software component as the following equation. The weights of the attributes are determined by the approach based on positive reciprocal matrix [19].

$$\begin{cases} T = \prod_{i=1}^{n} y_i^{w_i} \\ \sum_{i=1}^{n} w_i = 1 \end{cases}$$

Usually, the area experts are used to show the relations between two attributes by using fuzzy language [20]. For example, attribute a is *slightly more important than* attribute b, attribute a is *more important than* attribute b. For convenience of weight comparisons, we use the relative scale to represent them, which is shown in table 8.We determine $a_{ij} = 3$ if the i-th attribute is slightly more important than the j-th attribute.Conversely, the importance of the j-th attribute is $\frac{1}{3}$ of the i-th attribute, i.e., $a_{ji} = \frac{1}{3}$. When attribute a is two times of the importance of attribute b and attribute b is slightly more important than attribute c, we say the weight relation between a, b, c is $6 : 3 : 1$. After normalization, we get the weights of a, b, c are respectively 0.6, 0.3, 0.1.

We introduce the notion of positive reciprocal matrix as follows. The following matrix $A^{(*)}$ is the aggregation matrix given by four experts.

$$A^* = \begin{bmatrix} 1 & 0.5903 & 0.4475 & 0.6004 & 0.7522 \\ 1.6941 & 1 & 0.9091 & 0.9987 & 1.5006 \\ 2.2346 & 1.1 & 1 & 1.1667 & 1.7503 \\ 1.6656 & 1.0013 & 0.8571 & 1 & 1.4996 \\ 1.3294 & 0.664 & 0.5713 & 0.6668 & 1 \end{bmatrix}$$

Calculating the weights is divided into four steps.

Step 1: Normalizing each column of a matrix A^*,

$$w_{ij} = \frac{a^*_{ij}}{\sum_{i=1}^{n} a^*_{ij}}.$$

Step 2: Calculating the sum of each line w_{ij},

$$w_i = \sum_{i=1}^{n} w_{ij}.$$

Step 3: Obtaining the sum of w_i,

$$sum = \sum_{i=1}^{n} w_i.$$

Step 4: Computing the weights

$$w_i = \frac{w_i}{sum}.$$

Calculating the matrix $A^{(*)}$, the wights w_1, w_2, w_3, w_4,w_5 are 0.1262, 0.2279, 0.2662, 0.2246, 0.1551 respectively.

C. UPDATING THE TRUSTWORTHY DEGREE OF THE SOFTWARE COMPONENT

The trustworthy degree of software component is measured by attributes, and the attributes are measured by the sub-attributes. The weights of sub-attributes of a trustworthy attribute are almost the same important, we give the same weights. The trustworthy degree of the i-th attribute is computed by

$$y_i = (s_{i1} \times s_{i2} \times \cdots \times s_{ij} \times \cdots \times s_{it})^{\frac{1}{t}},$$

where, y_i is the trustworthy degree of the i-th attribute, s_{ij} is the trustworthy degree of the i-th attribute's j-th sub-attribute and t is the number of the sub-attribute. For example, $y_1 = (s_{11} \times s_{12} \times s_{13})^{\frac{1}{3}} = (0.971 \times 0.968 \times 0.873)^{\frac{1}{3}} = 0.9362$. Computing the five attributes, y_1, y_2, y_3, y_4 and y_5 are 0.9362, 0.9419, 0.9566, 0.9603, 0.9645 respectively. The trustworthy degree of users' feedback is $T_u = y_1^{w1} \times y_2^{w2} \times y_3^{w3} \times y_4^{w4} \times y_5^{w5} = y_1^{0.1262} \times y_2^{0.2279} \times y_3^{0.2662} \times y_4^{0.2246} \times y_5^{0.1551} = 0.9527$.

The origin trustworthy degree of the software component C-PAY is 0.85 and λ is 0.2. The weight of users' feedback is

$$w_u(5) = 1 - e^{-0.2\times 5} = 1 - e^{-1} = 0.6321.$$

The trustworthiness degree of updating is

$$\begin{aligned} T_n &= (1 - w_u(n)) \times T_o + w_u(n) \times T_u \\ &= 0.3679 \times 0.85 + 0.6321 \times 0.9527 \\ &= 0.3217 + 0.6022 = 0.9239. \end{aligned}$$

The updating model of component trustworthiness can well reflect the trustworthy degree of using component and the number of users. The model is reasonable.

IV. CONCLUSION

The trustworthy degree of software component should be measured accurately, which needs to some time even after a long period of time to use the software. In this paper, based on the number of "users", the trustworthy degree of the software is updated. To construct the weight function by the number of the users, we compute the trustworthy degree of the component by the updating model. Finally, a case study is presented, which shows that the proposed method is reasonable.

REFERENCES

[1] J. Wang *et al.*, "An approach to measureing and grading software trust for spacecraft software," *Sci. Sinica Technol.*, vol. 45, no. 2, pp. 221–228, 2015.

[2] K. Liu, Z. Shan, J. Wang, J. He, Z. Zhang, and Y.-W. Qin, "The trusted software basic research summary of major research plan," *Nat. Natural Sci. Found. China*, vol. 3, pp. 145–151, 2008.

[3] H. Tao and Y. Chen, "A new metric model for trustworthiness of softwares," *Telecommun. Syst.*, vol. 51, nos. 2–3, pp. 95–105, Nov. 2012.

[4] L. Zhang, Y. Zhou, Y. Chen, M. Zhang, and J. Zhang, "Stability of software trustworthiness measurements models," in *Proc. IEEE 7th Int. Conf. Softw. Secur. Rel.-Companion*, Jun. 2013, pp. 219–224.

[5] Y. Ma, Y. Chen, and B. Gu, "An attributes-based allocation approach of software trustworthy degrees," in *Proc. IEEE 9th Int. Conf. Softw. Secur. Rel.-Companion*, Aug. 2015, pp. 89–94.

[6] W. Hasseibring, "Research on trustworthy software systems," *Computer*, vol. 39, no. 4, pp. 91–92, 2006.

[7] J. Voas, "Trusted software's holy grail," *Softw. Qual. J.*, vol. 11, no. 1, pp. 9–17, 2003.

[8] B. Lang, X. D. Liu, H. M. Wang, B. Xie, and X. Mao, "A classification model for software trustworthiness," *J. Frontiers Comput. Sci. Technol.*, vol. 4, no. 3, pp. 231–239, 2010.

[9] Y. Shanlin, D. Shuai, and C. Wei, "Trustworthy software evaluation using utility based evidence theory," *J. Comput. Res. Develop.*, vol. 46, no. 7, pp. 1152–1159, 2009.

[10] M. D. McIlroy, J. Buxton, P. Naur, and B. Randell, "Mass-produced software components," in *Proc. Softw. Eng. Concepts Techn.*, 1968, pp. 88–98.

[11] C. Szyperski, "Component technology: What, where, and how?" in *Proc. IEEE Int. Conf. Softw. Eng.* Washington, DC, USA: IEEE Computer Society Press, May 2003, pp. 684–693.

[12] J. Sametinger, *Software Engineering With Reusable Components*. Berlin, Germany: Springer-Verlag, 1997, pp. 1–63.

[13] F. Bachman *et al.*, "Technical concepts of component-based software engineering," Tech. Rep. CMU/SEI-2000-TR-008, 2000, pp. 1–36.

[14] C. Szyperski, *Component Software: Beyond Object-Oriented Programming*, 2nd ed. Reading, MA, USA: Addison-Wesley, 2003, pp. 1–22.

[15] M. Abdellatief, A. B. M. Sulta, A. A. A. Ghani, and M. A. Jabar, "Component-based software system dependency metrics based on component information flow measurements," in *Proc. 6th Int. Conf. Softw. Eng. Adv. (ICSEA)*, Barcelona, Spain, 2011, pp. 76–83.

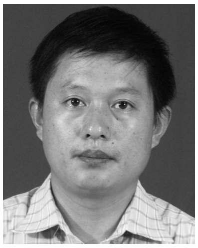
[16] M. von Detten, M. C. Platenius, and S. Becker, "Reengineering component-based software systems with archimetrix," *Softw. Syst. Model.*, vol. 13, no. 4, pp. 1239–1268, 2014.

[17] T. Vale *et al.*, "Twenty-eight years of component-based software engineering," *J. Syst. Softw.*, vol. 111, pp. 128–148, Jan. 2016.

[18] B. Meyer, "The grand challenge of trusted components," in *Proc. 25th Int. Conf. Softw. Eng. (ICSE)*, Portland, OR, USA, May 2003, pp. 660–667.

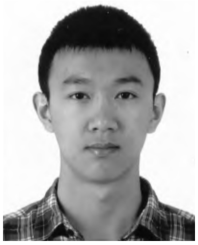
[19] B. Wang and S. Zhang, "A subjective and objective integration approach of determining weights for trustworthy measurement," *IEEE Access*, vol. 6, pp. 25829–25835, 2018.

[20] T. L. Saaty, "Analytic hierarchy process," in *Mathematical Models for Decision Support*. 2001, pp. 109–121.



BAOHUA WANG received the M.Sc. degree in computer science from Guizhou University, Guiyang, China, in 2007. He is currently pursuing the Ph.D. degree with the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China. His research interests include database security, software trustworthiness, reliability, and weights of attributes.

YIXIANG CHEN received the Ph.D. degree in mathematics from Sichuan University, in 1995. He is currently a Full Professor of computer science and software engineering with East China Normal University and also the Director of the Research Engineering Center of Software and Hardware Co-Design Technology and its application of the Ministry of Education of China. He has authored or co-authored more than 100 academic papers in the *Theoretical Computer Science*, *Information Sciences*, and related international conferences. His research interests include the Internet of Things, human cyber-physical systems, optimizational design method of intelligent systems, formal semantics, and trustworthiness of software, knowledge science, and engineering.



SHUN ZHANG received the M.Sc. degree in computational intelligence from The University of Sheffield, U.K., in 2014. He is currently a Researcher with the Guizhou Academy of Sciences, Guiyang, China. His research interests include the Internet of Things, robotics, data mining, artificial intelligence, and cyber security.

HENGYANG WU received the B.S. degree in mathematics from Xuzhou Normal University, Xuzhou, China, in 1996, and the M.Sc. and Ph.D. degrees in mathematics from Shanghai Normal University, Shanghai, China, in 2004 and 2007, respectively. From 2008 to 2011, he was a Postdoctoral Researcher with East China Normal University, Shanghai. From 2011 to 2016, he was an Associate Professor of computer science with Hangzhou Dianzi University. He is currently an Associate Research Fellow with East China Normal University. His current research interests include formal methods and domain theory.