Attributes Oriented Software Trustworthiness Measure Based on Axiomatic Approaches

Hongwei Tao[1], Yixiang Chen[2*], Han Liu[2], Hengyang Wu[3], Yinghui Hu[1]

[1] College of Computer and Communication Engineering, Zhengzhou University of Light Industry, China
[2] MoE Engineering Center for Software/Hardware Co-Design Technology and Application, East China Normal University, China
[3] School of Computer and Information Engineering, Shanghai Polytechnic University, China
tthhww_811@163.com, yxchen@sei.ecnu.edu.cn, 52194501008@stu.ecnu.edu.cn, Wuhengy_1974@aliyun.com, hyingh6@163.com

Abstract

Software trustworthiness measurement can provide evidence for increasing the trustworthiness of the design and implementation of software. We once studied the application of axiomatic approaches in software trustworthiness measurement, presented seven desirable properties of software trustworthiness measure based on attributes, and established three measures. However, none of the three measures can fully satisfy these seven properties. In this paper, we propose an improved software trustworthiness measure based on a partition which satisfies all of the seven properties. Meanwhile, we conduct an empirical validation of this measure by a real case. Comparative study shows that this measure is better than the three measures we once built.

Keywords: Trustworthy software, Trustworthy attribute, Trustworthiness measurement, Axiomatic approaches

1 Introduction

Software trustworthiness can be characterized by a set of attributes [1-3], which are called trustworthy attributes. Regarding that different users have different trustworthy requirements, we divided trustworthy attributes into critical attributes and non-critical attributes [4]. The trustworthy attributes that a trustworthy software must have are referred as critical attributes and the other trustworthy attributes are termed non-critical attributes [4]. It is necessary to state that for different software the critical and non-critical attributes may be different, while even for the same software they may also change according to user's trustworthy requirements. The quantification of the software trustworthiness through the software trustworthiness measurement is one of the ways to control software trustworthiness. The National Natural Science Foundation of China's Fundamental Research for Trustworthy Software major research plan lists software trustworthiness measurement and evaluation as the top of four important core issues [1].

In order to use more rigorous methods to measure software trustworthiness and validate the measures from theory, we once used axiomatic approaches to evaluate software trustworthiness, presented four properties which can be expected of software trustworthiness measure based on attributes [4], including monotonicity, acceleration, sensitivity and substitutivity. We improved the above property set later and added three properties: non-negativity, proportionality and expectability [5]. We also established three software trustworthiness measures: PBSTM1 [4], PBSTM2 [6], PBSTM3 [7]. PBSTM1 satisfies non-negativity and the four properties proposed in [4], but it does not comply with proportionality and expectability, moreover both the substitutivity between the critical attributes and that between the non-critical attributes in this measure are more difficult than the substitutivity between the critical and the non-critical attributes, however, in reality the substitutivity between the critical and the non-critical attributes should be more difficult than that between critical attributes or that between non-critical attributes. PBSTM2 partially resolves the problem related to substitutivity that is described above, but it does not satisfy sensitivity, proportionality and expextability. On the other hand, we can not change the substitutivity between attributes according to user requirements in PBSTM1 and PBSTM2. The substitutivity between attributes can be changed in PBSTM3, moreover, it satisfies non-negativity and the properties given in [4], however, it does not comply with proportionality and expextability, and has the same problem about substitutivity between attributes as that exists in PBSTM1. In this paper, we develop an improved software trustworthiness measure by partitioning critical attributes into several groups, which is referred as PBSTM4. PBSTM4 complies with all the seven properties, can change the substitutivity between trustworthy attributes as needed, and solves the problem about substitutivity between attributes in PBSTM1 and PBSTM3.

The rest parts of the paper are organized as follows. In Section 2 we review some related works. In Section 3 we describe software trustworthiness measure properties presented in [4-5]. We introduce PBSTM4 and carry out its theoretical validation in Section 4. The measurement procedure based on PBSTM4 is presented in Section 5. We describe the empirical validation of PBSTM4 by a case study in Section 6. Section 7 contains the comparative study. The conclusion and future work come in the last section.

*Corresponding Author: Yixiang Chen; E-mail: yxchen@sei.ecnu.edu.cn
DOI: 10.53106/160792642022052303016

2 Related Works

There are three types of software trustworthiness measurements, which are attribute-based software trustworthiness measurement, behavior-based software trustworthiness measurement and process-based software trustworthiness measurement [8-10].

The first category converts the software trustworthiness measurements into the selection, measures, and synthesis of trustworthy attributes. Representative ones include axiomatic approaches [4-7, 11], dynamic system [12], evidence theory [13], data mining [14], uncertain theory [15-16], Bayesian Networks [17], questionnaire survey and statistical analysis [18-19], and so on.

The second category measures the software trustworthiness based on the analysis results of software behaviors. The software normal behavior and the software actual behavior are built under the trustworthy operating environment and the actual operating environment separately. Then, these two behaviors are compared to evaluate the software trustworthiness. Classic comparison methods include program slicing [20-21], concurrency theory [22-24], Behavior Trajectory Matrix [25], Software Behaviour Entropy [26], etc.

Process-based software trustworthiness measurement analyzes whether the software development process follows a series of guarantees related to the software trustworthiness or data collected in the software process to evaluate the software trustworthiness. Typical methods contain Trusted Software Methodology [27-28], Trustworthy Process Management Framework [29-30], and so on.

3 Properties for Software Trustworthiness Measures Based on Attributes

With the same symbols as that presented in [4], we set $y_1,\ldots,y_m$ to be the degrees of critical attributes and $y_{m+1},\ldots,y_{m+s}$ be the degrees of non-critical attributes. Let T be a software trustworthiness measure function regarding $y_1,\ldots,y_{m+s}$. The desirable software trustworthiness measure properties based on attributes presented in [4-5] are depicted below.

1) Non-negativity

$$T \geq 0.$$

Non-negativity requires that software trustworthiness is non-negative.

2) Proportionality

$$\left(\exists c_1, c_2 \in R^+\right) c_1 \leq y_i / y_j \leq c_2, 1 \leq i, j \leq m+s.$$

Proportionality means that there should be an appropriate proportion assumption between the trustworthy attributes. For example, assuming that critical attributes include reliability and maintainability, trustworthy software requires both high reliability and good maintainability. Very high reliability and poor maintainability or very good maintainability and low reliability are not suitable.

3) Monotonicity

$$\frac{\partial T}{\partial y_i} \geq 0, 1 \leq i \leq m+s.$$

It implies that the increment of a trustworthy attribute leads to an increase in software trustworthiness.

4) Acceleration

$$\frac{\partial^2 T}{\partial y_i^2} \leq 0, 1 \leq i \leq m+s.$$

It means that the increase of a trustworthy attribute does not lead to its utilization efficiency to increase.

5) Sensitivity

$$0 \leq \frac{\partial T}{\partial y_i}\frac{y_i}{T} = f(y_i, w_i), 1 \leq i \leq m+s.$$

where w_i is the weight of y_i and f is a function regarding y_i and w_i. Sensitivity is used to describe the percentage change of software trustworthiness caused by the percentage change of trustworthy attributes. It should be non-negative and related to the corresponding attributes and their weights. The software trustworthiness is more sensitive to the minimal critical attribute compared with its weight.

6) Substitutivity

$$\left(\exists c_3, c_4 \in R^+\right) c_3 \leq \sigma_{ij} \leq c_4, 1 \leq i, j \leq m+s, i \neq j.$$

where

$$\sigma_{ij} = \frac{d\left(y_i / y_j\right)}{d(h_{ij})} \times \frac{h_{ij}}{y_i / y_j}, 1 \leq i, j \leq m+s, i \neq j \quad \mathbf{(1)}$$

is used to give expression of the difficulty of the substitution between the attributes, of which

$$h_{ij} = -\frac{\partial T / \partial y_j}{\partial T / \partial y_i} = \frac{dy_i}{dy_j}, 1 \leq i, j \leq m+s, i \neq j.$$

σ_{ij} satisfies $0 \leq \sigma_{ij} \leq 1$. The bigger σ_{ij} is, the easier substitution between the y_i and y_j is. The attributes y_i and y_j are completely substitutable at $\sigma_{ij}=1$ and they are not substitutable at $\sigma_{ij} = 0$. This property states that the trustworthy attributes can substitute each other to some extent.

7) Expectability

$(y_0 \leq \min\{y_1, \ldots, y_{m+s}\}) \Rightarrow (y_0 \leq T \leq \max\{y_1, \ldots, y_{m+s}\})$.

where y_0 is the user's minimum expectation of all trustworthy attributes, which is the threshold decided by the users. This asserts that if all the trustworthy attributes achieve the user expectations, then the software trustworthiness should meet the users' expectations too and be less than the maximum of all the trustworthy attributes.

4 An Improved Software Trustworthiness Measure Based on Partition and its Theoretical Validation

In this section we develop an improved software trustworthiness measure called PBSTM4 and carry out its theoretical validation by showing that it satisfies the properties described in Section 3

Definition 4.1 (An Improved Property Based Software Trustworthiness Measure PBSTM4): The improved software trustworthiness measure based on partition is defined as Equation (2)

$$\begin{cases} T=\left\{ \begin{array}{l} \alpha\left[\left(\dfrac{y^{(\min)}}{10}\right)^{\varepsilon}\left(y^{(1)}\right)^{\alpha^{(1)}}\left(y^{(2)}\right)^{\alpha^{(2)}}\cdots\left(y^{(S)}\right)^{\alpha^{(S)}} \right]^{-\rho} \\ +\beta\left[y_{m+1}^{\beta_{m+1}} y_{m+2}^{\beta_{m+2}} \cdots y_{m+s}^{\beta_{m+s}} \right]^{-\rho} \end{array} \right\}^{-\frac{1}{\rho}} \\ y^{(t)}=\left(\sum_{N_t}\alpha_i^{(t)} y_i^{-\rho_t}\right)^{-\frac{1}{\rho_t}},\ 1\le t\le S \end{cases} \quad \textbf{(2)}$$

where

1) m is the number of critical attributes, and s is the number of non-critical attributes.

2) $y_1,\dots,y_m$ are the trustworthy degrees of critical attributes, $y_{m+1},\dots,y_{m+s}$ are the trustworthy degrees of non-critical attributes, and all of them satisfy $1\le y_0\le y_i\,(1\le i\le m+s)$, of which y_0 is a specified degree that all of the trustworthy attributes must reach. $\{1,2,\dots,m\}$ is separated into S subsets $\{N_1,N_2,\dots,N_S\}$, and critical attributes are partitioned into S groups $\{Y^{(1)},Y^{(2)},\dots,Y^{(S)}\}$ with $y_i\in Y^{(t)}$ if $i\in N_t$. The principle of partition is that if there exists a positive correlation between two critical attributes, then they will be assigned to the same group. The quantitative relations between attributes can be decided according to the model presented by us in [31]. In fact, we can partition non-critical attributes too, however, for the sake of simplicity we will not discuss these situations in detail here.

3) T is the software trustworthiness measure function regarding $y_1,\dots,y_{m+s}$.

4) α and β are used to distinguish the contributions of critical attributes and non-critical attributes to the software trustworthiness which satisfy that $\alpha+\beta=1$ and $0\le\beta<0.5<\alpha\le 1$.

5) $\alpha_i\,(1\le i\le m)$ are the weight values of critical attributes with $\sum_{i=1}^{m}\alpha_i=1$ and $0\le\alpha_i\le 1$; $\alpha^{(t)}(1\le t\le S)$ are the notion of proportion among the critical attribute groups, then $\alpha^{(t)}=\sum_{i\in N_t}\alpha_i$ which satisfies that $\sum_{t=1}^{S}\alpha^{(t)}=1$ and $0\le\alpha^{(t)}\le 1$; $\alpha_i^{(t)}(i\in N_t,1\le t\le S)$ are the weights of all critical attributes within tth group, then $\alpha_i^{(t)}=\alpha_i/\alpha^{(t)}$ which follows $\sum_{i\in N_t}\alpha_i^{(t)}=1$, $0\le\alpha_i^{(t)}\le 1$.

6) $\beta_j\,(m+1\le j\le m+s)$ express the relative importance of the non-critical attributes such that $\sum_{j=m+1}^{m+s}\beta_j=1$ and $0\le\beta_j\le 1$.

7) ε denotes the effect of the critical attribute group with the smallest attribute value (called the minimum critical attribute group) on the software trustworthiness, which satisfies $0\le\varepsilon\le\min\left\{1-\alpha^{(\min)},\dfrac{\ln y_0-\ln y^{(\min)}}{\ln y^{(\min)}-\ln 10}\right\}$, $\min$ is the t with $\min_{1\le t\le S}\{y^{(t)}\}$ and $y^{(\min)}=\min_{1\le t\le S}\{y^{(t)}\}$.

8) $0<\rho$ is a parameter related to the substitutivity between critical and non-critical attributes.

9) $-1<\rho_t<0\,(1\le t\le S)$ are parameters that are associated with substitutivity among critical attributes.

For convenience, in the following of this paper we denote the i with $\max_{1\le i\le m}\{y_i\}$ by $\max'$, the i with $\min_{m+1\le i\le m+s}\{y_i\}$ by $\min''$, the i with $\max_{m+1\le i\le m+s}\{y_i\}$ by $\max''$, the t with $\max_{1\le t\le S}\{y^{(t)}\}$ by $\max$, and let

$$y_{\max'}=\max_{1\le i\le m}\{y_i\},\qquad y_{\min''}=\min_{m+1\le i\le m+s}\{y_i\},$$

$$y_{\max''}\ \max_{m+1\le i\le m+s}\{y_i\},\quad y^{(\max)}=\max_{1\le t\le S}\{y^{(t)}\},$$

$$a_1=\alpha\left[\left(\frac{y^{(\min)}}{10}\right)^{\varepsilon}(y^{(1)})^{\alpha^{(1)}}(y^{(2)})^{\alpha^{(2)}}\cdots(y^{(S)})^{\alpha^{(S)}}\right]^{-\rho},$$

$$b_1=\left[y_{m+1}^{\beta_{m+1}}y_{m+2}^{\beta_{m+2}}\cdots y_{m+s}^{\beta_{m+s}}\right]^{-\rho}.$$

It is easy to prove that Proposition 4.1 holds by the Definition 4.1.

Proposition 4.1: T satisfies non-negativity.

Proposition 4.2: T complies with proportionality.

Proof: Because

$$0<y_0\le y_i\le 10,\ 1\le i\le m+s,$$

it follows that

$$\frac{y_i}{10}\le\frac{y_i}{y_j}\le\frac{y_i}{y_0}\Rightarrow\frac{y_0}{10}\le\frac{y_i}{y_j}\le\frac{10}{y_0},1\le i,j\le m+s.$$

Therefore, proportionality is satisfied by T.

Proposition 4.3: T satisfies monotonicity.

Proof: Since

$$\frac{\partial T}{\partial y^{(t)}}=\begin{cases}\alpha\alpha^{(t)}T^{1+\rho}a_1\left(y^{(t)}\right)^{-1} & t\ne\min\\ \alpha\left(\alpha^{(t)}+\varepsilon\right)T^{1+\rho}a_1\left(y^{(t)}\right)^{-1} & t=\min,\end{cases}$$

$$\frac{\partial y^{(t)}}{\partial y_i}=\begin{cases}\alpha_i^{(t)}\left(\sum_{N_t}\alpha_i^{(t)}y_i^{-\rho_t}\right)^{-\frac{1}{\rho_t}-1}y_i^{-\frac{1}{\rho_t}-1} & i\in N_t\\ 0 & i\notin N_t,\end{cases}$$

$$\frac{\partial T}{\partial y_i}=\beta\beta_i T^{1+\rho}\left[y_{m+1}^{\beta_{m+1}}\cdots y_{m+s}^{\beta_{m+s}}\right]^{-\rho}y_i^{-1},\quad m+1\le i\le m+s$$

and for $1\le i\le m$, $m+1\le j\le m+s$, $1\le t\le S$

$$\begin{cases}0\le\alpha,\alpha^{(t)},\alpha_i^{(t)},\alpha^{(\min)}+\varepsilon\le 1\\ 0\le\beta,\beta_j\le 1\end{cases},$$

then we can get

$$\begin{cases}\dfrac{\partial T}{\partial y_i}=\dfrac{\partial T}{\partial y^{(t)}}\dfrac{\partial y^{(t)}}{\partial y_i}\ge 0 & 1\le i\le m\\ \dfrac{\partial T}{\partial y_i}\ge 0 & m+1\le i\le m+s.\end{cases}$$

Therefore, monotonicity is satisfied by T.

Proposition 4.4: Expectability holds for T, i.e.,

$$y_0\le T\le\max\{y_1,\dots,y_{m+s}\}.$$

Proof: From the proof of Proposition 4.3, we know that for $1\le i\le m$, $1\le t\le S$, $\dfrac{\partial y^{(t)}}{\partial y_i}\ge 0$, and because for $1\le i\le m$, $0<y_0\le y_i\le 10$, then we have

$$y_0\le y^{(t)}=\left(\sum_{N_t}\alpha_i^{(t)}y_i^{-\rho_t}\right)^{-\frac{1}{\rho_t}}\le y_{\max'}\le 10.$$

Note that $\sum_{t=1}^{S}\alpha^{(t)}=1$, it follows that

$$\frac{\left(y^{(\min)}\right)^{1+\varepsilon}}{10^{\varepsilon}} \le (\frac{y^{(\min)}}{10})^{\varepsilon}\left(y^{(1)}\right)^{\alpha^{(1)}}...\left(y^{(S)}\right)^{\alpha^{(S)}} \le y^{(\max)}.$$

Similarly, since $\sum_{j=m+1}^{m+s}\beta_j = 1$. Therefore,

$$y_{\min''} \le y_{m+1}^{\beta_{m+1}} y_{m+2}^{\beta_{m+2}} \dots y_{m+s}^{\beta_{m+s}} \le y_{\max''}.$$

Because $\rho > 0$, $0 \le \alpha, \beta \le 1$, and $\alpha + \beta = 1$, thus

$$\alpha\left(y^{(\max)}\right)^{-\rho} + \beta\left(y_{\max''}\right)^{-\rho} \le \alpha a_1 + \beta b_1$$

$$\le \alpha\left(\frac{\left(y^{(\min)}\right)^{1+\varepsilon}}{10^{\varepsilon}}\right)^{-\rho} + \beta\left(y_{\min''}\right)^{-\rho},$$

then we can deserve that

$$\min\left\{\frac{\left(y^{(\min)}\right)^{1+\varepsilon}}{10^{\varepsilon}}, y_{\min''}\right\} \le T \le \max\left\{y^{(\max)}, y_{\max''}\right\}. \quad \textbf{(3)}$$

Because of $0<y_0 \le y_i \le 10$ $(1 \le i \le m + s)$ and

$$0 \le \varepsilon < \min\left\{1-\alpha^{(\min)}, \frac{\ln y_0 - \ln y^{(\min)}}{\ln y^{(\min)} - \ln 10}\right\},$$

it follows that

$$y_0 \le \min\left\{\frac{\left(y^{(\min)}\right)^{1+\varepsilon}}{10^{\varepsilon}}, y_{\min''}\right\}. \quad \textbf{(4)}$$

Observe that

$$\begin{cases} y^{(\max)} \le & y_{\max'} \le \max\{y_1,...,y_{m+s}\} \\ & y_{\max''} \le \max\{y_1,...,y_{m+s}\} \end{cases} \quad \textbf{(5)}$$

then from the inequations (3), (4) and (5) we have that

$$y_0 \le T \le \max\{y_1, \dots, y_{m+s}\}.$$

i.e., T complies with expectability.

Proposition 4.5: Acceleration holds for T.

Proof: Notice that for $1 \le i \le m$

$$\frac{\partial^2 T}{\partial y_i^2} = \frac{\partial^2 T}{\partial y^{(t)2}}\left(\frac{\partial y^{(t)}}{\partial y_i}\right)^2 + \frac{\partial T}{\partial y^{(t)}}\frac{\partial^2 y^{(t)}}{\partial y_i^2}.$$

By solving the derivatives, we can obtain

$$\frac{\partial^2 T}{\partial y^{(t)2}} = \begin{cases} \alpha\alpha^{(t)}T^{1+\rho}a_1(y^{(t)})^{-2} & \\ [\alpha\alpha^{(t)}(1+\rho)T^{\rho}a_1 - \rho\alpha^{(t)} - 1] & t \ne \min \\ \alpha(\alpha^{(t)}+\varepsilon)T^{1+\rho}a_1(y^{(t)})^{-2} & \\ [\alpha(\alpha^{(t)}+\varepsilon)(1+\rho)T^{\rho}a_1 - \rho(\alpha^{(t)}+\varepsilon) - 1] & t = \min, \end{cases}$$

$$\frac{\partial^2 y^{(t)}}{\partial y_i^2} = \begin{cases} -\alpha_i^{(t)}(1+\rho_t)(y^{(t)})^{1+2\rho_t} y_i^{-\rho_t-2}(\sum_{N_t, j\ne i}\alpha_j^{(t)} y_j^{-\rho_t}) & i \in N_t \\ 0 & i \notin N_t. \end{cases}$$

Due to $0< \frac{\alpha a_1}{T^{\rho}} \le 1$, and for $1 \le i \le m$, $1 \le t \le S$,

$$\begin{cases} 0 \le \alpha, \alpha^{(t)}, \alpha_i^{(t)}, \alpha^{(\min)} + \varepsilon \le 1 \\ 0 \le 1 + \rho_t < 1, \end{cases}$$

it follows that

$$\frac{\partial^2 T}{\partial y^{(t)2}} \le 0, \frac{\partial^2 y^{(t)}}{\partial y_i^2} \le 0.$$

From the proof of Proposition 4.4 and the definition of T, we know that for $1 \le t \le S$, $\frac{\partial T}{\partial y^{(t)}} \ge 0$. So

$$\frac{\partial^2 T}{\partial y_i^2} \le 0, 1 \le i \le m.$$

For $m + 1 \le i \le m + s$,

$$\frac{\partial^2 T}{\partial y_i^2} = \beta\beta_i T^{1+\rho} b_1 y_i^{-2}\left[(1+\rho)\beta\beta_i T^{\rho} b_1 - \rho\beta_i - 1\right].$$

Since $0< \frac{\beta b_1}{T^{\rho}} \le 1$ and

$$\begin{cases} 0 \le \beta_i \le 1, m+1 \le i \le m+s \\ 0 < 1+\rho \end{cases}.$$

Then

$$\frac{\partial^2 T}{\partial y_i^2} \le 0, m+1 \le i \le m+s.$$

In summary, for $1 \le i \le m + s$, we have

$$\frac{\partial^2 T}{\partial y_i^2} \le 0.$$

Proposition 4.6: T complies with sensitivity.

Proof: By calculating, it easy to get that for $1 \le i \le m$,

$$\frac{\partial T}{\partial y_i}\frac{y_i}{T} = \frac{\partial T}{\partial y^{(t)}}\frac{\partial y^{(t)}}{\partial y_i}\frac{y_i}{T} =$$

$$\begin{cases} \alpha\alpha^{(t)}\alpha_i^{(t)}T^{\rho}a_1\left(y^{(t)}\right)^{\rho_t} y_i^{-\rho_t} & i \in N_t, t \ne \min \\ \alpha\left(\alpha^{(t)}+\varepsilon\right)\alpha_i^{(t)}T^{\rho}a_1\left(y^{(t)}\right)^{\rho_t} y_i^{-\rho_t} & i \in N_t, t = \min, \end{cases}$$

and for $m + 1 \le i \le m + s$,

$$\frac{\partial T}{\partial y_i}\frac{y_i}{T} = \beta\beta_i T^{\rho} b_1.$$

Since for $1 \le i \le m$, $m + 1 \le j \le m + s$, $1 \le t \le S$,

$$\begin{cases} 0 \le \alpha, \alpha^{(t)}, \alpha_i^{(t)}, \alpha^{(\min)} + \varepsilon \le 1 \\ 0 \le \beta, \beta_j \le 1 \end{cases}$$

and for $1 \le i \le m + s$, $0<y_0 \le y_i \le 10$. Therefore,

$$\frac{\partial T}{\partial y_i}\frac{y_i}{T} \ge 0, 1 \le i \le m+s.$$

Then we obtain the conclusion that T is sensitive to all trustworthy attributes and the minimal critical attribute group has more effects on software trustworthiness through adding ϵ.

Proposition 4.7: Substitutivity is satisfied by T.

Proof: First, we consider the substitutivity between two attributes that belong to the same group.

According to the equation (1), the substitutivity between attributes which belong to the same critical attribute group can be derived as

$$\sigma_{ij} = 1/\rho_t, i, j \in N_t, 1 \le t \le S. \quad \textbf{(6)}$$

For the substitutivity between non-critical attributes, the following result can be deserved

$$\sigma_{ij} = 1, m+1 \le i, j \le m+s, i \ne j. \quad \textbf{(7)}$$

Now consider the substitutivity between two attributes belonging to different groups and discuss them in the following cases.

1) The substitutivity between attributes in different non-minimal critical attribute groups can be determined as follows

$$\sigma_{ij} = \frac{c_2 + d_2}{a_2 c_2 + b_2 d_2}, i \in N_t, j \in N_r, r \neq t, r \neq \min, t \neq \min, \quad \textbf{(8)}$$

where

$$\begin{cases} a_2 = 1 + \rho_t - \rho_t \alpha_i^{(t)} \left(\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t} \right)^{-1} y_i^{-\rho_t} \\ b_2 = 1 + \rho_r - \rho_r \alpha_j^{(r)} \left(\sum_{N_r} \alpha_j^{(r)} y_j^{-\rho_r} \right)^{-1} y_j^{-\rho_r} \\ c_2 = \alpha^{(r)} \alpha_j^{(r)} \left(\sum_{N_r} \alpha_j^{(r)} y_j^{-\rho_r} \right)^{-1} y_j^{-\rho_r} \\ d_2 = \alpha^{(t)} \alpha_i^{(t)} \left(\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t} \right)^{-1} y_i^{-\rho_t}. \end{cases}$$

Since $-1 \le \rho_t < 0$ $(1 \le t \le S)$ and

$$0 \le \frac{\alpha_i^{(t)} y_i^{-\rho_t}}{\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t}} \le 1, i \in N_t,$$

we get

$$\begin{cases} 1 + \rho_t \le a_2 \le 1 \\ 1 + \rho_r \le b_2 \le 1. \end{cases}$$

Because of $0 \le c_2, d_2$, $0 \le 1 + \rho_t, 1 + \rho_r$, it follows that $\min\{1+\rho_t, 1+\rho_r\}(c_2 + d_2) \le a_2 c_2 + b_2 d_2 \le c_2 + d_2$. Dividing $c_2 + d_2$ by the above inequations, we can deserve that

$$1 \le \sigma_{ij} \le \max\left\{ \frac{1}{1+\rho_t}, \frac{1}{1+\rho_r} \right\}, \quad i \in N_t, j \in N_r, r \neq t, r \neq \min, t \neq \min. \quad \textbf{(9)}$$

2) Likewise, for the substitutivity between attributes in non-minimal critical attribute group and attributes in minimal critical attribute group, we have

$$\sigma_{ij} = \frac{c_2 + d_2}{a_2 c_2 + b_2 d_2}, i \in N_t, j \in N_r, r \neq \min, t = \min, \quad \textbf{(10)}$$

where

$$\begin{cases} a_3 = 1 + \rho_t - \rho_t \alpha_i^{(t)} \left(\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t} \right)^{-1} y_i^{-\rho_t} \\ b_3 = 1 + \rho_r - \rho_r \alpha_j^{(r)} \left(\sum_{N_r} \alpha_j^{(r)} y_j^{-\rho_r} \right)^{-1} y_j^{-\rho_r} \\ c_3 = \alpha^{(r)} \alpha_j^{(r)} \left(\sum_{N_r} \alpha_j^{(r)} y_j^{-\rho_r} \right)^{-1} y_j^{-\rho_r} \\ d_3 = (\alpha^{(t)} + \varepsilon) \alpha_i^{(t)} \left(\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t} \right)^{-1} y_i^{-\rho_t}. \end{cases}$$

By the similar proof, we can obtain

$$1 \le \sigma_{ij} \le \max\left\{ \frac{1}{1+\rho_t}, \frac{1}{1+\rho_r} \right\}, \quad i \in N_t, j \in N_r, r \neq \min, t = \min. \quad \textbf{(11)}$$

3) Similarly, for the substitutivity between attributes in non-minimal critical attribute group and non-critical attributes we have

$$\sigma_{ij} = \frac{c_4 + d_4}{a_4 c_4 + b_4 d_4}, i \in N_t, m+1 \le j \le m+s, t \neq \min, \quad \textbf{(12)}$$

where

$$\begin{cases} a_4 = \left(\rho \alpha^{(t)} - \rho_t\right) \alpha_i^{(t)} y_i^{-\rho_t} \left(\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t} \right)^{-1} + 1 + \rho_t \\ b_4 = 1 + \rho \beta_j \\ c_4 = \beta \beta_j b_1 \\ d_4 = \alpha \alpha^{(t)} \alpha_i^{(t)} a_1 \left(\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t} \right)^{-1} y_i^{-\rho_t}, \end{cases}$$

Since

$$\begin{cases} -1 \le \rho_t < 0 & 1 \le t \le S \\ 0 < \rho & \\ 0 \le \beta_j & m+1 \le j \le m+s \end{cases}$$

and

$$0 \le \frac{\alpha_i^{(t)} y_i^{-\rho_t}}{\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t}} \le 1, i \in N_t, 1 \le t \le S.$$

Then it follows that

$$\begin{cases} 1 + \rho_t \le a_4 \le 1 + \rho \alpha^{(t)} & 1 \le t \le S, t \neq \min \\ 1 \le b_4 \le 1 + \rho \beta_j & m+1 \le j \le m+s. \end{cases}$$

Therefore, we get

$$\min\left\{ \frac{1}{1+\rho\alpha^{(t)}}, \frac{1}{1+\rho\beta_j} \right\} \le \sigma_{ij} \le \frac{1}{1+\rho_t}, \quad i \in N_t, 1 \le t \le S, t \neq \min, m+1 \le j \le m+s, \quad \textbf{(13)}$$

4) For the substitutivity between attributes in minimal critical attribute group and non-critical attributes we have

$$\sigma_{ij} = \frac{c_5 + d_5}{a_5 c_5 + b_5 d_5}, i \in N_t, m+1 \le j \le m+s, t = \min, \quad \textbf{(14)}$$

where

$$\begin{cases} a_5 = \left[\rho \left(\alpha^{(t)} + \varepsilon \right) - \rho_t \right] \alpha_i^{(t)} y_i^{-\rho_t} \left(\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t} \right)^{-1} + 1 + \rho_t \\ b_5 = 1 + \rho \beta_j \\ c_5 = \beta \beta_j b_1 \\ d_5 = \alpha \left(\alpha^{(t)} + \varepsilon \right) \alpha_i^{(t)} a_1 \left(\sum_{N_t} \alpha_i^{(t)} y_i^{-\rho_t} \right)^{-1} y_i^{-\rho_t}. \end{cases}$$

By the similar proof, we can obtain

$$\min\left\{\frac{1}{1+\rho\left(\alpha^{(t)}+\varepsilon\right)},\frac{1}{1+\rho\beta_j}\right\}\le\sigma_{ij}\le\frac{1}{1+\rho_t}, \qquad \textbf{(15)}$$
$$i\in N_t, t=\min, m+1\le j\le m+s.$$

To sum up the above arguments, from the equations (6), (7), (9), (11), (13) and (15), we know that T satisfies the substitutivity.

From equations (6), (8), (10), (12) and (14), we can obtain that the substitutivity between critical attributes and the substitutivity between the critical attributes and non-critical attributes can be changed according to user expectations by changing the parameters $\rho_t(1 \le t \le S)$ or ρ. In fact, if we separate the non-critical attributes by the similar methods, then the substitutivity between the non-critical attributes can be changed too.

We can treat all non-critical attributes as a non-critical attribute group. From equations (6) and (7), we know that in this measure T the substitutivity between attributes within the same group are identical. From equations (9), (11), (13), and (15), we can obtain that in this measure T the substitutivity between two attributes belonging to different groups is more difficult than the substitutivity between attributes within each group to which the two attributes belong. Therefore, the substitutivity between critical attributes and non-critical attributes are more difficult than the bigger of the substitutivity between attributes within each group.

From the conclusions of Propositions 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, we can get the following result.

Theorem 4.1: T satisfies all of the properties described in Section 3.

5 Measurement Procedure Based on PBSTM4

The measurement procedure based on PBSTM4 are shown in Figure 1. For a given software, the degrees of critical attributes $y_1,...,y_m$ and the degrees of non-critical attributes $y_{m+1},...,y_{m+s}$ are first determined. Then, the weights of critical attributes $\alpha_1,...,\alpha_m$ and the weights of non-critical attributes $\beta_{m+1},...,\beta_{m+s}$ are computed. For example, the method presented in [6] can be used to calculate these weights. In the Step 3, the critical attributes are partitioned into S groups $\{Y^{(1)},Y^{(2)},...,Y^{(S)}\}$ based on the positive correlations between two critical attributes and the subscript set of the attribute in the tth group is recorded as N_t. In the Step 4, $\alpha^{(t)}=\sum_{i\in N_t}\alpha_i(i\in N_t,1\le t\le S)$, $\alpha_i^{(t)}=\alpha_i/\alpha^{(t)},(i\in N_t,1\le t\le S)$ are calculated and the parameters ρ, $\rho_t(1 \le t \le S)$, and ϵ are determined. In the Step 5, the software trustworthiness is obtained by utilizing PBSTM4.

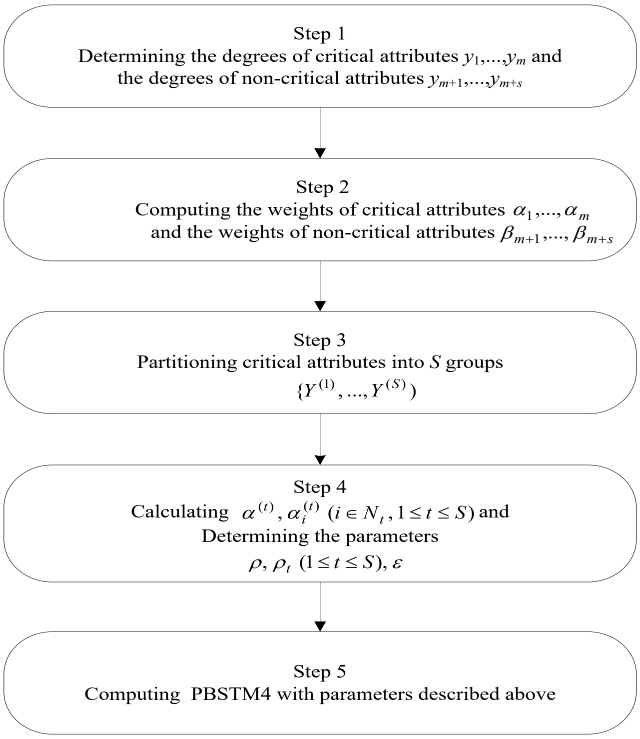


Figure 1. Measurement procedure based on PBSTM4

6 Case Study

To demonstrate the effectiveness of PBSTM4, an empirical validation is carried out by using it to evaluate 23 spacecraft software. The trustworthy attributes of spacecraft software and their weight values are presented in Table 1 [32]. The weight values are computed by the method given in [6]. These 9 attributes consist of 28 sub-attributes. 10 experts are invited to grade the 28 sub-attributes of the 23 spacecraft software to obtain the trustworthy attribute values, which are respectively denoted by y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, and y_9. The trustworthy attribute values of 11 representative spacecraft software are shown in Table 2 [32].

Table 1 Trustworthy attributes and their weight values

Trustworthy attribute	Weight value
1. Test verification	0.20
2. Analysis and design	0.17
3. Reliability and safety	0.15
4. Configuration management	0.11
5. Software technology status change	0.09
6. Quality problem close loop	0.09
7. Third party evaluation situation	0.09
8. Overall planning and implementation	0.05
9. Software development environment	0.05

On the basis of the weight values of 9 trustworthy attributes, the first four are taken as critical attributes and the last five are chosen as non-critical attributes. Then from Table 1, we can obtain the following parameter values of PBSTM4: $m = 4$; $s = 5$, $\alpha = 0.20 + 0.17 + 0.15 + 0.11 = 0.63$, $\beta = 0.09 + 0.09 + 0.09 + 0.05 + 0.05 = 0.37$, and

$$(\alpha_1,\alpha_2,\alpha_3,\alpha_4) = (\frac{0.20}{0.63},\frac{0.17}{0.63},\frac{0.15}{0.63},\frac{0.11}{0.63}) = (0.32, 0.27, 0.24, 0.17),$$

$$(\beta_1,\beta_2,\beta_3,\beta_4,\beta_5) = (\frac{0.09}{0.37},\frac{0.09}{0.37},\frac{0.09}{0.37},\frac{0.05}{0.37},\frac{0.05}{0.37}) = (0.24, 0.24, 0.24, 0.14, 0.14).$$

Considering that test verification is positively related to reliability and safety, and analysis and design is positively related to configuration management, then y_1, y_3 are assigned to the same group recorded as the first group, and y_2 and y_4 are assigned to the same group recorded as the second group. We can obtain that

$$\alpha^{(1)} = \alpha_1 + \alpha_3 = 0.56,\ \alpha^{(2)} = \alpha_2 + \alpha_4 = 0.44,$$

$$\alpha_1^{(1)} = \frac{\alpha_1}{\alpha^{(1)}} = 0.57,\ \alpha_2^{(2)} = \frac{\alpha_2}{\alpha^{(2)}} = 0.61,$$

$$\alpha_3^{(1)} = \frac{\alpha_3}{\alpha^{(1)}} = 0.43,\ \alpha_4^{(2)} = \frac{\alpha_4}{\alpha^{(2)}} = 0.39.$$

Let $\rho = 0.5, \rho_1 = -0.5, \rho_2 = -0.2.$ The trustworthy degrees of these 11 software calculated based on PBSTM1, PBSTM2, PBSTM3, and PBSTM4 are given in Table 2, and their distributions are shown in Figure 2. From the measurement result of No. 6 given in this table, we can observe that if the threshold y_0 decided by user is 5.90, then PBSTM1, PBSTM2, and PBSTM3 do not satisfy expectability. From Table 2 we can also find that we want to increase the rank of trustworthiness of a software. Firstly, we would better improve the minimal critical attribute not only because it is easier to be improved, but also because it is more sensitive to the software trustworthiness compared with its relative importance. Secondly, we can make the most important attribute better.

Table 2 Trustworthy attribute values and trustworthy degrees of 11 representative software based on PBSTM1, PBSTM2, PBSTM3, PBSTM4

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	ε	PBSTM1	PBSTM2	PBSTM3	PBSTM4
No. 2	6.13	7.61	8.26	9.00	8.28	8.37	9.65	8.28	9.00	0.1 0.05	5.09 5.17	4.08 4.16	5.14 5.20	7.72 7.81
No. 4	4.76	8.28	7.61	9.00	8.26	8.37	9.33	7.66	4.24	0.1 0.05	4.78 4.89	3.83 3.94	4.86 4.94	6.97 7.09
No. 6	5.90	9.32	7.00	7.00	8.26	8.37	9.65	7.66	7.00	0.1 0.05	4.97 5.05	3.99 4.07	5.00 5.07	7.35 7.46
No. 7	6.15	9.32	4.63	7.62	8.26	8.37	9.65	7.66	7.00	0.1 0.05	4.77 4.88	3.79 3.90	4.82 4.91	6.94 7.08
No. 9	7.94	8.56	7.00	8.28	8.26	7.94	9.65	7.66	7.94	0.1 0.05	5.22 5.28	4.24 4.30	5.25 5.30	7.93 8.01
No.18	8.41	9.00	9.00	10.00	8.28	9.49	8.26	8.33	9.00	0.1 0.05	5.58 5.61	4.57 4.60	5.59 5.62	8.77 8.81
No.19	9.16	9.00	8.28	9.66	8.26	8.37	9.00	8.33	9.00	0.1 0.05	5.57 5.60	4.57 4.60	5.59 5.61	8.76 8.79
No.20	8.28	5.48	7.61	8.59	7.61	8.37	9.00	7.00	9.49	0.1 0.05	4.98 5.07	4.00 4.09	5.04 5.11	7.45 7.55
No.21	8.20	9.32	7.00	8.59	9.00	8.37	8.86	8.33	9.00	0.1 0.05	5.34 5.40	4.33 4.39	5.37 5.41	8.27 8.34
No.22	5.99	9.32	7.00	8.59	9.00	8.37	10.00	8.33	9.00	0.1 0.05	5.11 5.19	4.08 4.15	5.13 5.20	7.741 7.85
No.23	8.40	9.00	9.00	10.00	8.58	8.37	8.26	9.00	9.00	0.1 0.05	5.57 5.60	4.58 4.61	5.58 5.61	8.74 8.78

It can be found from Figure 2 that under the same parameter values, the measurement results of the 11 software trustworthiness obtained based on PBSTM4 are greater than those obtained based on PBSTM3, the measurement results calculated based on PBSTM3 are greater than those calculated based on PBSTM1, and the measurement results deserved based on PBSTM1 are greater than those deserved based on PBSTM2. The reason is that PBSTM1 considers the influence of both critical attributes and non-critical attributes on software trustworthiness, while PBSTM2 only considers the impact of critical attributes and the minimal non-critical attributes on software trustworthiness. PBSTM3 is obtained by partitioning the critical attributes in PBSTM2. After partition, the substitution between the critical attributes in PBSTM3 is easier than the substitution between the critical attributes in PBSTM1, which is conducive to improving the software trustworthiness measurement results by compensating the lower trustworthy attribute values with the higher trustworthy attribute values. PBSTM4 further optimizes the substitutivity between attributes under the condition of satisfying the ex- pectability.

It can also be seen from the Figure 2 that PBSTM4 will not greatly increase the trustworthy degree of the software because of the high trustworthy degrees of individual attributes, but the attributes with lower values will greatly reduce the trustworthy degree of the software.

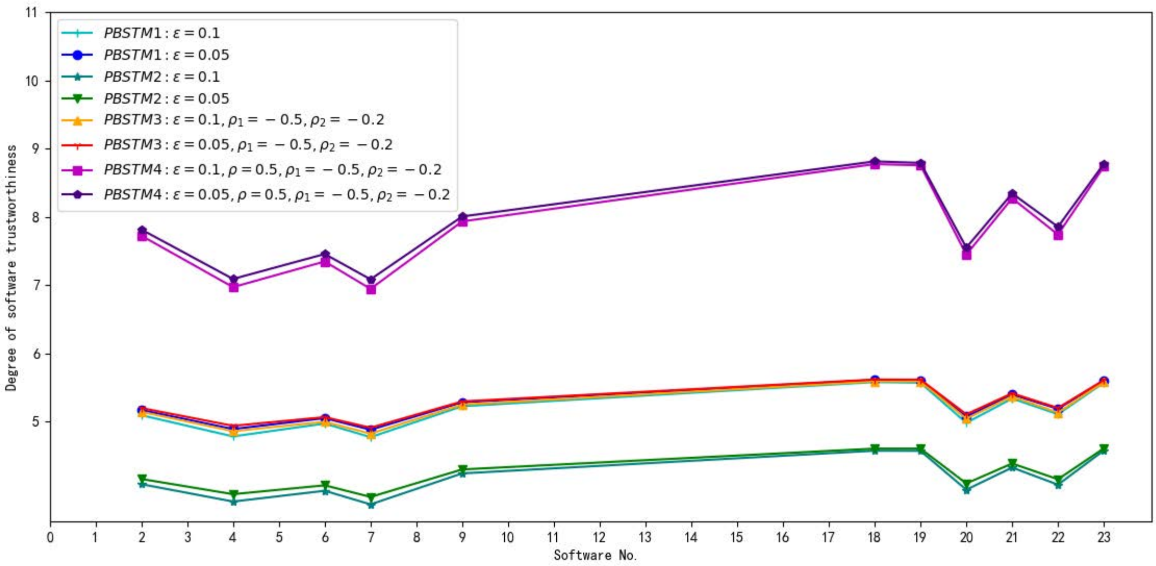


Figure 2. Distributions of trustworthy degrees of 11 representative software based on PBSTM1, PBSTM2, PBSTM3 and PBSTM4

7 Comparative Study in Terms of the Properties Described in Section 3

In this section we compare PBSTM4 with PBSTM1 [4], PBSTM2 [6], PBSTM3 [7] and two other popular software trustworthiness measures: evidence theory based software trustworthiness measure (ERBSTM) [13] and fuzzy theory based software trustworthiness measure (FTBSTM) [15] through the properties given in Section 3. The comparative results are summarized in Table 3, of which × represents the measure does not satisfy the corresponding property and √ expresses the measure complies with the corresponding property. PBSTM1, PBSTM2, PBSTM3, ERBSTM, and FTBSTM do not consider the problem that relates to the proportionality of trustworthy attributes, therefore all of them do not comply with the second property. In the last section we have proved that PBSTM1, PBSTM2, and PBSTM3 do not satisfy expectability by a counterexample. Reference [6] proved that sensitivity does not hold for PBSTM2. ERBSTM and FTBSTM do not satisfy acceleration and substitutivity. The reason is that both of them do not consider the efficiency of using attributes and the quantitative relations between trustworthy attributes, while correlations between trustworthy attributes are the cause that trustworthy attributes can substitute each other to some extent.

From Table 3, we can obtain that PBSTM4 is better than all the other five measures through the properties presented in Section 3.

Table 3 Comparison of PBSTM4 with 5 established measures in terms of properties presented in Section 3

Property	PBSTM4	PBSTM1	PBSTM2	PBSTM3	FTBSTM	ERBSTM
Non-negativity	√	√	√	√	√	√
Proportionality	√	×	×	×	×	×
Monotonicity	√	√	√	√	√	√
Acceleration	√	√	√	√	×	×
Sensitivity	√	√	×	√	√	√
Substitutivity	√	√	√	√	×	×
Expectability	√	×	×	×	√	√

8 Conclusion and Future Work

In this study we develop an improved software trustworthiness measure by partitioning critical attributes into several groups. It not only satisfies all the seven properties described in Section 3, but also can solve the problem about substitutivity between trustworthy attributes in PBSTM1 and PBSTM3. Moreover, in the PBSTM4 the substitutivity between trustworthy attributes can be changed according to user requirements. Comparative study shows that PBSTM4 is better than the other five software trustworthiness measures in term of the properties given in Section 3.

There are several problems that are worth further study. First, we consider the software trustworthiness measure properties that are necessary but not sufficient. It is possible that some important properties are omitted because of oversight. We will extend and refine the set of properties in the future. Secondly, the software trustworthiness measures based on trustworthy attributes that do not satisfy the properties presented in this paper cannot be taken as legitimate measures. However, the measures that do satisfy the set of properties should only be taken as candidate measures, as they still need to be better examined. Third, we do not give methods for computing the parameters ϵ, ρ and ρ_t. How to determine the values of these parameters is an important future work. Lastly, we partition critical attributes into groups based on the positive correlation between two critical attributes in this paper. We will experiment this partition method using the real cases in the future.

Acknowledgements

This work was financially supported by the National Key Research and Development Projects of China (2018YFB2101300); the Doctoral Research Fund of Zhengzhou University of Light Industry (2016BSJJ037); and the Science and Technology Project of Henan Province (202102210351, 212102210076).

References

[1] J. F. He, Z. G. Shan, J. Wang, G. G. Pu, Y. F. Fang, K. Liu, R. Z. Zhao, Z. T. Zhang, Review of the Achievements of Major Research Plan on "Trustworthy Software", *Bulletin of National Natural Science Foundation of China*, Vol. 32, No. 3, pp. 291-296, June, 2018.

[2] D. D. Wang, D. X. Wang, Q. J. Wei, D. S. Wu, J. He, L. Z. Wang, Q. Wang, Z. Z. Liu, Y. L. Lai, Y. Hu, Z. C. Feng, J. F. Cui, Y. Y. Zhang, *GB/T 37970-2019 Trustworthiness Assessment for Software Process and Artifact*, China National Standardization Management Committee, August, 2019.

[3] J.-H. Cho, S. H. Xu, P. M. Hurley, M. Mackay, T. Benjamin, M. Beaumont, STRAM: Measuring the Trustworthiness of Computer-based Systems, *ACM Computing Surveys*, Vol. 51, No. 6, pp. 1-47, February, 2019.

[4] H. W. Tao, Y. X. Chen, A Metric Model for Trustworthiness of Softwares, *2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops*, Milan, Italy, 2009, pp. 69-72.

[5] H. W. Tao, J. Zhao, An Improved Attributes-Based Software Trustworthiness Metric Model, *Journal of Wuhan University (Natural Science Edition)*, Vol. 63, No. 2, pp. 151-157, September, 2017.

[6] H. W. Tao, Y. X. Chen, A New Metric Model for Trustworthiness of Softwares, *Telecommunication Systems*, Vol. 51, No. 2-3, pp. 95-105, November, 2012.

[7] H. W. Tao, Y. X. Chen, Another Metric Model for Trustworthiness of Softwares based on Partition, *Quantitative Logic and Soft Computing 2010*, Xiamen, China, 2010, pp. 695-705.

[8] V. del Bianco, L. Lavazza, S. Morasca, D. Taibi, A Survey on Open Source Software Trustworthiness, *IEEE Software*, Vol. 28, No. 5, pp. 67-75, September-October, 2011.

[9] S. Maza, O. Megouas, Framework for Trustworthiness in Software Development, *International Journal of Performability Engineering*, Vol. 17, No. 2, pp. 241-252, February, 2021.

[10] T. J. Yang, Comparative Study on the Performance Attributes of NHPP Software Reliability Model based on Weibull Family Distribution, *International Journal of Performability Engineering*, Vol. 17, No. 4, pp. 343-353, April, 2021.

[11] J. Wang, Y. X. Chen, B. Gu, X. Y. Guo, B. H. Wang, S. Y. Jin, J. Xu, J. Y. Zhang, An Approach to Measuring and Grading Software Trust for Spacecraft Software, *Scientia Sinica Techologica*, Vol. 45, No. 2, pp. 221-228, January, 2015.

[12] X. Zhang, W. Li, Z. M. Zheng, B. H. Guo, Optimized Statistical Analysis of Software Trustworthiness Attributes, *Science China (Information Sciences)*, Vol. 55, No. 11, pp. 2508-2520, November, 2012.

[13] S. Ding, S. L. Yang, C. Fu, A Novel Evidential Reasoning based Method for Software Trustworthiness Evaluation Under the Uncertain and Unreliable Environment, *Expert Systems with Applications*, Vol. 39, No. 3, pp. 2700-2709, February, 2012.

[14] Y. Yuan, Q. Han, A Software Behavior Trustworthiness Measurement Method based on Data Mining, *International Journal of Computational Intelligence Systems*, Vol. 4, No. 5, pp. 817-825, January, 2011.

[15] H. L. Shi, J. Ma, F. Y. Zou, A Fuzzy Comprehensive Evaluation Model for Software Dependability Based on Entropy Weight, *2008 International Conference on Computer Science and Software Engineering*, Wuhan, China, 2008, pp. 683-685.

[16] B. Li, Y. Cao, An Improved Comprehensive Evaluation Model of Software Dependability based on Rough Set Theory, *Journal of Software*, Vol. 4, No. 10, pp. 1152-1159, December, 2009.

[17] N. E. Fenton, B. Littlewood, M. Neil, L. Strigini, A. Sutclife, D. Wright, Assessing Dependability of Safety Critical Systems Using Diverse Evidence, *IEE Proceedings Software*, Vol. 145, No. 1, pp. 35-39, February, 1998.

[18] D. Taibi, V. del Bianco, D. D. Carbonare, L. Lavazza, S. Morasca, Towards the Evaluation of OSS Trustworthiness: Lessons Learned from the Observation of Relevant OSS Projects, *OSS: IFIP International Conference on Open Source Systems*, Milano, Italy, 2008, pp. 389-395.

[19] L. Lavazza, S. Morasca, D. Taibi, D. Tosi, Predicting OSS Trustworthiness on the Basis of Elementary Code Assessment, *2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, Bolzano/Bozen, Italy, 2010, pp. 1-4.

[20] D. Wang, Y. Lu, W. B. Zhao, L. H. Fu, Trust-measuring Model for Software Using Dependent Relation Between Variables, *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, Vol. 41, No. 1, pp. 41-45, January, 2013.

[21] D. Wang, H. Y. Sun, J. Wang, L. H. Fu, Trusted Analysis Model for Interactive Behavior of a Software System based on Slicing Technology, *Journal of Beijing*

University of Technology, Vol. 39, No. 5, pp. 713-721, May, 2013.

[22] F. Zhang, M. D. Xu, L. You, A Behavior Modeling Method based on System Call and Algebra Process CCS, *Journal of Wuhan University*, Vol. 56, No. 2, pp. 133-137, February, 2010.

[23] Q. Zhao, H. Q. Wang, G. S. Feng, J. Zhao, Measuring Method of Software Dependability based on Pi Calculus, *Journal of Jilin University (Engineering and Technology Edition)*, Vol. 41, No. 6, pp. 1684-1689, November, 2011.

[24] N. Fu, X. S. Zhou, T. Zhan, QtPi: a Calculus to Enforce Trustworthiness Requirements, *Journal of Computer Research and Development*, Vol. 48, No. 11, pp. 2120-2130, November, 2011.

[25] J. F. Tian, Y. H. Guo, Software Trustworthiness Evaluation Model based on a Behavior Trajectory Matrix, *Information and Software Technology*, Vol. 119, No. 1, Article No. 106233, March, 2020.

[26] Q. Han, Trustworthiness Measurement Algorithm for TWfMS based on Software Behaviour Entropy, *Entropy*, Vol. 20, No. 3, pp. 1-13, March, 2018.

[27] E. Amoroso, T. Nguyen, J. Weiss, J. Watson, P. Lapiska, T. Starr, Toward An Approach To Measuring Software Trust, *1991 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA, USA, 1991, pp. 198-218.

[28] E. Amoroso, C. Taylor, J. Watson, J. Weiss, A Process-oriented Methodology for Assessing and Improving Software Trustworthiness, *the 2nd ACM Conference on Computer and Communications Security*, Fairfax, Virginia, USA, 1994, pp. 39-50.

[29] H. P. Zhang, F. D. Shu, Y. Yang, X. Wang, Q. Wang, A Fuzzy-based Method for Evaluating the Trustworthiness of Software Processes, *ICSP 2010: New Modeling Concepts for Today's Software Processes*, Paderborn, Germany, 2010, pp. 297-308.

[30] J. Du, Y. Yang, Q. Wang, M. S. Li, Evidence-based Trustworthy Software Process Assessment Method, *Journal of Frontiers of Computer Science and Technology*, Vol. 5, No. 6, pp. 501-512, June, 2011.

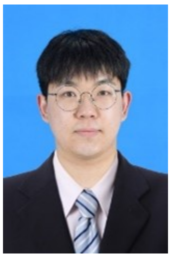
[31] H. W. Tao, Y. X. Chen, A Quantitative Relation Model Between Trustworthy Attributes, *the 3rd International Conference on Quantitative Logic and Software Computing*, Xi'an, China, 2012, pp. 197-204.

[32] H. W. Tao, H. Y. Wu, Y. X. Chen, An Approach of Trustworthy Measurement Allocation based on Sub-Attributes of Software, *Mathematics*, Vol. 2019, Vol. 7, No. 3, pp. 1-15, March, 2019.

Biographies

Hongwei Tao received the Ph.D. degree in computer applications technology from East China Normal University in 2011. Currently, he is an Associate Professor with Zhengzhou University of Light Industry, China. He has published more than 30 papers in various journals and conferences. His current research interests include software trustworthiness measurement, big data analysis and formal methods.

Yixiang Chen received the Ph.D. degree in mathematics from Sichuan University in 1995. He is currently a Full Professor with East China Normal University, China. He has published more than 90 papers in various journals and conferences. His current research interests include trustworthy software, the Internet of things and human-cyber-physical systems.



Han Liu received the B.S. degree from Jiangsu University of Science and Technology, China, in 2019. He is currently pursuing the Ph.D degree in East China Normal University, China. His current research interests include software and systems trustworthiness measurement.

Hengyang Wu received the Ph.D. degree in mathematics from Shanghai Normal University in 2007. He has published more than 40 papers in various journals and conferences. He is currently an Associate Professor with Shanghai Polytechnic University, China. His current research interests include formal methods and domain theory.

Yinghui Hu received the B.S. degree from Zhengzhou University of Light Industry, China, in 2018. He is currently pursuing the M.S. degree in Zhengzhou University of Light Industry, China. His current research interests include blockchain technology and artificial intelligence.