

## Trustworthiness Measure Platform of Intelligent Subway Information System

Zhenjiang-Zhang, Wang Bo, XinLei-Jin, Sumeng-Diao and YunLiu

*Department of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing, 100044, China*

*{zhjzhang1, 11125032, 11120100, 11111009, liuyun}@bjtu.edu.cn*

### **Abstract**

*With the development of urban rail transit, the wisdom subway information system based on the technology of wireless sensor network and RFID emerged at the right moment. This paper analyzes the framework of the wisdom subway information system and studied the system trustworthiness in detail. We develop a measurement model that provides a new method for measuring the software trustworthiness and a platform used to measure the software trustworthiness of the information system.*

**Keywords:** *the wisdom subway information system, software trustworthiness, set of software trustworthiness indices, SAF*

### **1. Introduction**

As the promoting of urbanization process, the operation of rail transport achieves rapid development. However, the informatization and integration do not reach a high level. In order to maintain a good rail transportation environment and avoid major security incidents, the intelligent subway information system which revolves about the Internet of things technology, combining the Wireless sensor network technology and RFID technology arises at the historic moment. The information system collects all kinds of information along the rail transit and then sent them to the information platform for processing. The intelligent subway information system runs in the Internet environment, it is not safe for hostile attack which may be occurred at any time. In addition, there may be software vulnerability and defects which will bring instability during the development of the system. This decreases the extent to which users can trust their software, so there is an urgent need for reliable ways to measure the trustworthiness of intelligent subway information system.

In this paper, we studied the trustworthiness measure methods in subway information system and propose a new trustworthiness measurement model. We also develop a trustworthiness measure platform of intelligent subway information system and we will introduce it in detail.

## **2. Related work**

### **2.1. Analysis of intelligent subway information system**

As the points of focus of trustworthiness are different with the difference of the types of systems and software, we should perform detail analysis for a specific software system. In this paper, we aim at measuring the trustworthiness of intelligent subway information system. First of all, we make some analysis of the system. The function of our intelligent subway information system is collecting environment information parameters along the rail transit and performing real-time monitoring for the safety status of tracks and bridges. The system includes the perception layer, base station transport layer, data process layer and application layer, and capable of sensor data acquisition, data analyzing and storage, data transmission and data processing. Therefore our intelligent subway information system is a kind of information management software.

### **2.2. Software trustworthiness research**

Currently there's no unified definition and standard for software trustworthiness. Different experts and organizations have different understanding to "Trust". Trusted Computing Group (TCG) defined the "Trust" as follows: An entity is trusted if its behavior is always achieving the desired goals in the expected way [1]. Huaimin-Wang defined the "Trust" as follows: We say software is trusted when its system's behavior is always the expected behavior and results consistent with the users [2].

According to the research result of the definition of trust previously, we give a new definition of trust in this paper:

Definition 1: Software trustworthiness refers to the software system can make sure the operating results consistent with user's expectations and win the trust of users in the objective in a certain environment.

## **3. Key Technology**

### **3.1. The establishment of the software trustworthiness indices set**

Based on existing researches [3-5], in this paper, we establish a new set of software trustworthiness indices. We provided the definition of software trustworthiness, and divided it into five levels indices: availability, reliability, security, maintainability and survivability, which ranging from high to low. Then, we refined the index at each level in order to establish a new set of software trustworthiness indices comprised of 43 secondary indices.

**Availability:** When used under specified conditions, the software has the capacity of meeting the demand of both explicit and implicit functional, as well as the capacity of being understood, learned, used, and transplanted.

**Reliability:** In the specified conditions and within the stipulated time, the capacity of operating without failure.

**Security:** The ability of ensuring the confidentiality, integrity, availability and authenticity of data and information for a software system.

**Maintainability:** The ability for software product of being modified, being improved, or adapting to the environment when operating environment or the functional requirements change.

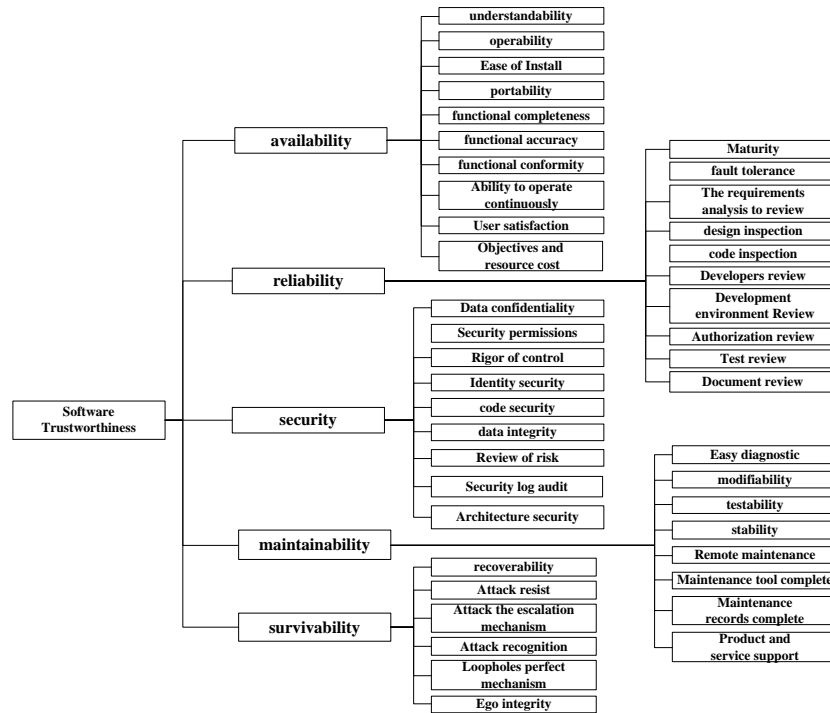
**Survivability:** The ability of against attacks to guarantee important functions running and restore the service within the required time when the software is under attack or unexplained failure.

Specific system of software trustworthiness indices is shown in Figure 1.

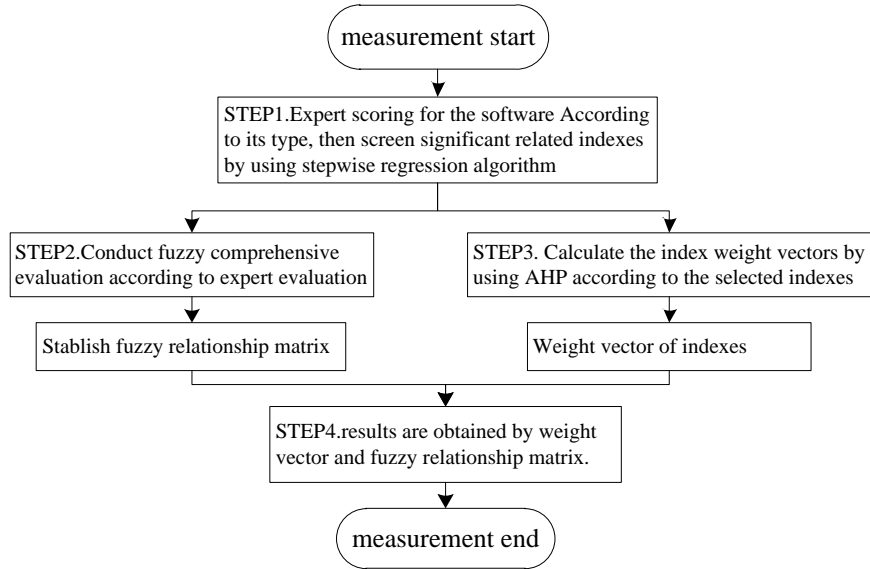
### 3.2. Stepwise-AHP-Fuzzy

The processing of measuring software trustworthiness is a relatively subjective method that is restricted by the existing research and technology, so the process still unable to be totally objective. As the software trustworthiness has social nature at the same time, so we should conduct a comprehensive analysis of the software to be measured and using the mathematical theories and models to minimize the effect of subjective factor. This paper presents a new software trustworthiness measurement algorithm: Stepwise-AHP-Fuzzy (SAF).

The SAF is divided into four steps: Step 1, A number of expert scoring for the software according to the established software trustworthiness index set and type of software, then screen significant related indices by using stepwise regression algorithm; Step 2, Calculate the index weight vectors by using AHP according to the selected indices and software; Step 3, Establish fuzzy relationship matrix by using fuzzy comprehensive evaluation method; Step 4, Software trustworthiness measurement results are obtained by weight vector and fuzzy relationship matrix. The process of measuring software trustworthiness is shown in Figure 2:



**Figure 1. System of software trustworthiness indices**



**Figure 2. Flow chart of measuring software trustworthiness**

**3.2.1. Screening index of stepwise regression algorithm:** According to the software trustworthiness measurement model proposed by Voas [6],

$$Q = aR + bP + cF + dSa + eSe + fT + gA + hM \quad (1)$$

Where Q expresses the trustworthiness of software, and R, P, F, Sa, Se, A, T, M represents the reliability, property, faulttolerance, security, confidentiality, availability, testability and maintainability respectively; a, b, c, d, e, f, g, h indicate the weight of trustworthiness index. It follows that the software trustworthiness can be linearly represented by the trustworthiness indices.

The new trustworthiness measurement model proposed in this paper is as follows:

$$TW = aA + bR + cSa + dM + eL + \gamma \quad (2)$$

Where TW expresses the trustworthiness of software, A, R, Sa, M, L represents availability, reliability, security, maintainability and survivability respectively, a, b, c, d, e is the weight of corresponding trustworthiness index, which are unknown attributes.

The level indicators of trustworthiness can also linearly represented by the secondary indicators:

$$A = \sum_{i=1}^n \alpha_i AX_i \quad (i = 1, 2, \dots, 10) \quad (3)$$

Among them  $AX_i$  as secondary level indices of availability, for example: comprehensibility, easy operation etc.  $\alpha_i$  as the corresponding weight coefficient of indices. Similarly available:

$$R = \sum_{i=1}^n \beta_i RX_i \quad (i = 1, 2, \dots, 10); \quad Sa = \sum_{i=1}^n \chi_i SaX_i \quad (i = 1, 2, \dots, 10) \quad (4)$$

$$M = \sum_{i=1}^n \delta_i MX_i \quad (i = 1, 2, \dots, 8); \quad L = \sum_{i=1}^n \varepsilon_i LX_i \quad (i = 1, 2, \dots, 6)$$

Here,  $RX_i$ ,  $SaX_i$ ,  $MX_i$ ,  $LX_i$  as secondary level indices of reliability, security, maintainability and survivability respectively.  $\beta_i$ ,  $\chi_i$ ,  $\delta_i$ ,  $\varepsilon_i$  as the corresponding weight coefficient of indices respectively. The type (4) into type (2) we can get:

$$TW = a \sum_{i=1}^{10} \alpha_i AX_i + b \sum_{i=1}^{10} \beta_i RX_i + c \sum_{i=1}^{10} \chi_i SaX_i + d \sum_{i=1}^8 \delta_i MX_i + e \sum_{i=1}^6 \varepsilon_i LX_i + \gamma$$

$$= a\alpha_1 AX_1 + \dots + a\alpha_{10} AX_{10} + b\beta_1 RX_1 + \dots + b\beta_{10} RX_{10} + c\chi_1 SaX_1 + \dots + c\chi_{10} SaX_{10} \quad (5)$$

$$+ d\delta_1 MX_1 + \dots + d\delta_8 MX_8 + e\varepsilon_1 LX_1 + \dots + e\varepsilon_6 LX_6 + \gamma$$

In order to ensure the correctness and rigor of software trustworthiness indices measurement model, before the stepwise regression algorithm screening indices, put forward the following 4 conditions:

Condition 1: Software trustworthiness can be expressed by the indicator of centralized indicator linearly.

Condition 2: Software trustworthiness indices set includes not significant indicators which Impacts on the measurement model.

Condition 3: The number of experts who do credibility indices selection is greater than the size of the software trustworthiness indices set.

Condition 4: The correlation of software reliability focus indicator is not bigger than it in software trustworthiness itself.

Hypothesis1: Existing software trustworthiness indices set contains all known software trustworthiness indicators, if there are unknown indices, representation used in the measurement model, and hands it as interference.

In expert scoring step, the number of scoring experts  $k$  is greater than the indices set number  $n$ . That is,  $n+1$  time fight each expert score, a total of  $k$  experts. The software trustworthiness  $TW$  as the dependent variable, is  $k \times 1$  matrix. The indices set as the independent variable, is  $k \times n$  matrix.

The use of the stepwise regression algorithm is a relatively mature statistical algorithm [7]; it has already been integrated in many mathematical statistics software. We use stepwise regression algorithm function `stepwisefit()` in Matlab for computing. The function format of `stepwisefit()` is:

$$[\sim, \sim, \text{inmodel}, \sim, \sim] = \text{stepwisefit}(x, y, 'penter', 0.1, 'premove', 0.2)$$

Where input parameter  $Y$  is the dependent variable, namely  $TW$ , is  $k \times 1$  matrix;  $X$  is the independent variables, the set of indicators scoring,  $k \times n$  matrix; `penter`: significance probability limit of regression equation significant parity, the independent variables are introduced into the equation's biggest significance probability; `premove`: regression equation significant calibration's significant probability limit. The output data, the corresponding item value is 1 in vector in model, said the indices is significantly linearly, the indices were

screened. The corresponding item value of 0, Said the indicators no significant linear, has been dropped out of the equation.

Assume that there is a total of  $n$  software trustworthiness indices, after stepwise regression algorithm selected  $p$  of  $n$ ,  $p \leq n$ , remember to FS (Filtered Set). FS is a  $1 \times p$  matrix. That is  $FS = [FX_1 \ FX_2 \ \dots \ FX_p]$ , among them  $FX_i$  is an item in  $n$  indices, a total of  $p$ .

**3.2.2. Fuzzy comprehensive evaluation:**

1. According to the first step in the hypothesis selected the FS, which is divided into five remark factor set [8], respectively for the collection of secondary indicators of the availability, reliability, security, maintainability, and survivability. Assume that software reliability remark factor set is  $U = \{u_1, u_2, u_3, u_4, u_5\}$ , among them  $u_1, u_2, u_3, u_4, u_5$  said availability, reliability, security, maintainability, and survivability remark factor set.  $U \triangleq FS$ , among them,  $u_1 \cup u_2 \cup u_3 \cup u_4 \cup u_5 = FS$  and  $u_1 \cap u_2 \cap u_3 \cap u_4 \cap u_5 = \Phi$ .

2. Determine the comment set

In the process of fuzzy comprehensive evaluation, the software indices grade is classified into five grades: very low, low, normal, high, and very high. Namely, to establish evaluation set  $V = \{v_1, v_2, v_3, v_4, v_5\} = \{\text{very low, low, normal, high, very high}\}$ . For example, expert 1 may think “very low “the score is expressed as 20, but expert 2 may think it is 40, that is comment set membership degree classification for each expert is not consistent. So we should score comment set membership, using the triangular fuzzy number to show membership, to do comment set  $V$  expert scoring. Assume that  $\tilde{A} = (l, m, u)$  is the triangular fuzzy number, represents membership [9], as function (9) shows

$$A(x) = \begin{cases} (x - l) / (m - l), & l \leq x \leq m \\ (u - x) / (u - m), & m \leq x \leq u \\ 0, & \text{other} \end{cases} \quad (6)$$

Among them  $l$  represents lower,  $m$  represents most likely values,  $u$  represents upper. According to comments set membership, set up Table 1,

**Table 1. Scoring Tables of Remark Set Membership**

Comment set	Expert 1	...	Expert m
VL	(0, 20,30)	...	(0, 10,20)
L	(30,40,50)	...	(20,30,40)
N	(50,60,70)	...	(40,50,60)
H	(70,80,90)	...	(60,70,80)
VH	(90,100,100)	...	(80,90,100)

Put Table 1 into comment set membership degree of normalized processing, using the weighted average algorithm, as follows:

$$\bar{A}_{VL} = \sum_{i=1}^m (l_{vLi}, m_{vLi}, u_{vLi}) / m = (\bar{l}_{VL}, \bar{m}_{VL}, \bar{u}_{VL}) \quad (7)$$

We can get the average membership function of evaluation set is  $\bar{A}_L, \bar{A}_N, \bar{A}_H, \bar{A}_{VH}$ , then according to the blur function (7).

$$\tilde{a}_i = \frac{1}{2} [a_m + \lambda a_l + (1 - \lambda) a_u] \quad (8)$$

Assume that decision makers risk aversion degree scale  $\lambda = 0.5$ , said policymakers to risk a neutral attitude. We can get defuzzification function:

$$\bar{A}_{0.5} = m / 2 + (l + u) / 4 \quad (9)$$

Put the average membership function into formula (9), we can get the precise value set corresponding to comments set V. which is

$$V = \{v_1, v_2, v_3, v_4, v_5\} \mapsto \bar{A} = \{\bar{A}_1, \bar{A}_2, \bar{A}_3, \bar{A}_4, \bar{A}_5\}$$

At this point, the use of triangular fuzzy numbers to establish evaluation set is complete; we get a set of accurate numerical  $\bar{A}$  corresponding to the comment set V.

### 3. Fuzzy Comprehensive Evaluation

Experts judge the individual elements in the remark factors set, first, based on the remark factor set U, establishing expert fuzzy remark form; there are m experts, to evaluate p indicators.

Availability, for example, assumes that remark factors set  $u_1 = \{\text{Comprehensibility, Operational ease, Easy Installation}\}$ . According remark sets, to constructs the fuzzy relation matrix  $R_1$ . First, construct single factor remark vector of comprehensibility. Expert I's comment on comprehensibility is  $v_j$ ,  $v_j$  is j-thin comment set ( $i = 1, 2, \dots, m; j = 1, 2, \dots, 5$ ). Assume that  $num_{1j}$  is the number which n experts' remark of comprehensibility, thereby establishing single factor remark vector of comprehensibility.

$$R_{11} = (num_{11} / m, num_{12} / m, num_{13} / m, num_{14} / m, num_{15} / m)$$

Similarly we can derive:

Single factor remark vector of operational ease:

$$R_{12} = (num_{21} / m, num_{22} / m, num_{23} / m, num_{24} / m, num_{25} / m)$$

Single factor remark vector of easy installation:

$$R_{13} = (num_{31} / m, num_{32} / m, num_{33} / m, num_{34} / m, num_{35} / m)$$

We can get availability fuzzy relation 3\*5 matrix  $R_1$  by  $R_{11}, R_{12}, R_{13}$ ,

$$R_1 = \frac{1}{m} \begin{bmatrix} num_{11} & num_{12} & num_{13} & num_{14} & num_{15} \\ num_{21} & num_{22} & num_{23} & num_{24} & num_{25} \\ num_{31} & num_{32} & num_{33} & num_{34} & num_{35} \end{bmatrix}$$

Repeat the above steps; you can construct fuzzy relation matrix of reliability  $R_2$ , fuzzy relation matrix of security  $R_3$ , fuzzy relation matrix of maintainability  $R_4$ , fuzzy relation matrix of survivability  $R_5$ .

#### 4. AHP to Determine the Weight

This step is the use of analytic hierarchy process (AHP) to make sure the weight first level indicators among availability, reliability, security, maintainability, survivability. Then make sure the weight between every first level indicator and its secondary level indicators.

Grading way using the Delphi method with the method of anonymous comments, through multiple rounds of survey that experts view the establishment of judgment matrix, after much consultation, summarize, modify, and finally aggregated into expert consensus, as final result of the two-two judgment matrix.

Assume that after using the Delphi method, experts unified views on final result of the two-two judgment matrix, established a first level indicators judgment matrix of availability, reliability, security, maintainability, survivability.

Assume that first level indicators judgment matrix was a matrix of  $P_{5 \times 5}$  compare the importance of each indicator for software reliability, using proportional scale method 1-9 for scoring. Suppose the results as follows.

$$P = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1/2 & 1 & 2 & 3 & 4 \\ 1/3 & 1/2 & 1 & 2 & 3 \\ 1/4 & 1/3 & 1/2 & 1 & 2 \\ 1/5 & 1/4 & 1/3 & 1/2 & 1 \end{bmatrix}$$

Then calculate judgment matrix  $P$ 's largest eigenvalue  $\lambda_{MAX}$  and the weight vector  $W$  by using the Specifications listed average method.  $W$  is the weight coefficient of each index allocation results.

Specific steps of Specifications listed average method are as follows:

Each column vector of  $P$  normalized  $\tilde{w}_{ij} = a_{ij} / \sum_{i=1}^n a_{ij}$ , and then sums each row  $\tilde{w}_i = \sum_{j=1}^n \tilde{w}_{ij}$ ,

normalization of the sum  $w_i = \tilde{w}_i / \sum_{i=1}^n \tilde{w}_i$ , and the vector  $w = (w_1, w_2, \dots, w_n)^T$  is the weight vector. Calculate the largest eigenvalue as follows:

$$\lambda_{MAX} = \frac{1}{n} \sum_{i=1}^n \frac{(Pw)_i}{w_i} \quad (10)$$



Then according the AHP consistency check method, use C.R to verify whether the consistency of judgment matrix is passed.  $C.R = C.I / R.I$ , where  $C.I = (\lambda_{MAX} - n) / (n - 1)$ ,  $R.I$  is the average consistency checking value, which can be gained by checking Table 2. If  $C.R < 0.1$ , we consider P has passed the consistency check, otherwise, it needs to be adjusted.

**Table 2. R.I Reference Table**

N	1	2	3	4	5	6	7	8	9
R.I	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

Similarly, we can get the weight vectors and perform consistency check of the judgment matrix of the secondary indices.  $W_1$  is the availability weight vector,  $W_2$  is the reliability weight vector,  $W_3$  is the security weight vector,  $W_4$  is the maintainability weight vector, and  $W_5$  is the survivability of weight vector.

### 5. Get the Judgment Results through the Fuzzy Matrix and Weight Vector

After the step 3 we can get the fuzzy relation matrix of availability, reliability, security, maintainability and survivability, represented by  $R_1, R_2, R_3, R_4, R_5$  respectively, we have also gained the weight vectors  $W_1, W_2, W_3, W_4, W_5$  of each matrix by step 4. Then we will combine the weight vector  $W$  with the fuzzy relationship matrix  $R$ . The fuzzy operator is indicated by the weighted average operator  $M(\bullet, \oplus)$ , if  $A \circ R = B = (b_1, b_2, \dots, b_m)$ , then

$$b_j = \sum_{i=1}^n a_i r_{ij} = \min(1, \sum_{i=1}^n a_i r_{ij}), j = 1, 2, \dots, m, b_j \in [0, 1].$$

We can get the comprehensive evaluation of various factors. Namely

$$B_i = W_i \circ R_i (i = 1, 2, 3, 4, 5) \tag{11}$$

Combine judgment results of each Levels index:

$$R_l = (B_1, B_2, B_3, B_4, B_5) \tag{12}$$

$R_l$  is the relationship fuzzy matrix of levels indices. By combining  $R_l$  and  $W_l$ , we can get the fuzzy comprehensive evaluation results  $G$  of software trustworthiness.

$$G = W_l \circ R_l = (g_1, g_2, g_3, g_4, g_5) \tag{13}$$

Then quantified the comment set  $V$ 's precise set of values  $\bar{A}$  as follows

$$S = \sum_{i=1}^5 g_i \bar{A}_i \tag{14}$$

Finally we can get the result of metrics for software's trustworthiness indicated by  $S$ .

The  $S$  we get before is a hundred mark numerical result, which can get the level of fuzzy comprehensive evaluation according to maximum membership principle. Namely that evaluation result  $G$  is an item of the comment set  $V$ .

## 4. The Overall Design Scheme of the Platform

### 4.1. System architecture design

The main function of the platform is to screen out the software trustworthiness indices which have a significant impact on the software trustworthiness. Screening is based on the intelligent subway information system software category that is information management. Then the experts perform fuzzy comprehensive evaluation of the selected indicators, obtaining the results of software's trustworthiness indices measurement and measurement report to measure the intelligent subway information system. According to the platform requirements, we choose MyEclipse and Matlab as the development tools, the programming language are Java, Matlab, HTML, JSP and JavaScript, the underlying database use MySQL. Based on B/S architecture, we use the MVC pattern of development, Struts2 + Hibernate3 technologies.

### 4.2. System function design

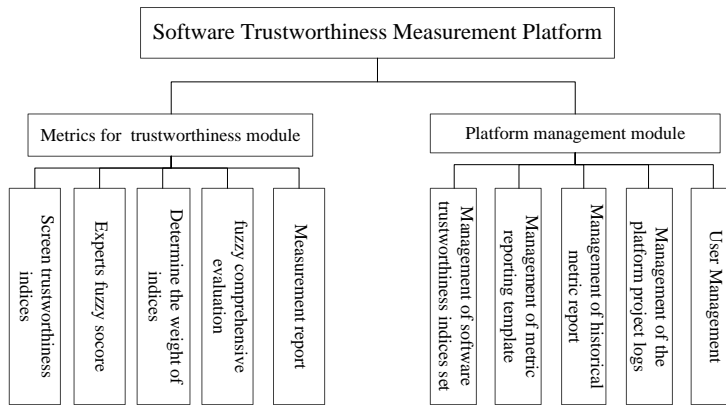


Figure 3. System function module

#### 1. Metrics for software trustworthiness module

Mainly complete the function of measuring the software trustworthiness indices. Firstly, screen the software category indicators according to the existing set of software trustworthiness indices, then the experts give fuzzy scores and determine the index weight according to screened software trustworthiness indices and intelligent subway information system software. Finally, we get results of the fuzzy comprehensive evaluation and give the measurement report.

#### 2. Management of software trustworthiness indices set

This function module is to stores all the software trustworthiness indices in the management system. We can perform adding, deleting, changing or checking to the indices. But we should have a panel discussion before the corresponding operation.

#### 3. Management of metric reporting template

Metric report templates pre-populated with some of the metric report basic information, such as the metric process description, measurement level description, and experts

suggestions to the metric results and other information. The module can change the information according to the actual situation.

#### 4. Management of historical metric report

This module achieves functions such as downloading metrics reports, viewing, uploading and deleting historical metric reports. The report is downloaded in PDF format or Word format.

#### 5. Management of the platform project logs

Implement management of the metric projects and view historical metric project logs.

#### 6. User Management

Manage user information including user name, login name, password, user units, user permissions and other information. According to the user to add, delete, change, check etc., assign user roles and permissions.

### 5. Conclusions

This paper describes the methods of metrics for software trustworthiness of the intelligent subway information system. We propose a new algorithm for measuring the software trustworthiness: fuzzy comprehensive evaluation based on stepwise regression and the analytic hierarchy process' (SAF), we also conducted the system design and implementation of the measurement platform. We carried out measurements and tests of the Intelligent Subway Information System and getting out the measurement report by using the metric platform. The result will provide references and opinions to intelligent subway information system's improvement.

### References

- [1] TCG Web Site, <http://www.trustedcomputinggroup.org>.
- [2] H. M. Wang, X. D. Liu and B. Lang, "Software Trustworthiness Classification Specifications V2.0. National "863" key Project", High confidence software tools and integrated environment technical documentation, (2009).
- [3] K. Liu, Z. G. Shan and J. Wang, "The basic research of trusted software", Summary of Major Research Plan, NSFC, vol. 22, no. 3, (2008), pp. 145-151.
- [4] H. W. Chen, J. Wang and W. Dong, "High confidence software engineering technology", ACTA Electronica Sinica, vol. 31, no. z1, (2003), pp. 1933-1938.
- [5] NSTC, "Research Challenges in High Confidence Systems", Proceeding of the Committee on Computing Information and Connection Workshop, (1997).
- [6] J. Voas, "Trusted software's holy grail", Software Quality Journal, vol. 11, no. 1, (2003), pp. 9-17.
- [7] X. Q. He and W. Q. Liu, "Applied Regression Analysis", Beijing China Renmin University Press, (2011), pp. 141-149.
- [8] H. G. Zhang and B. Zhao, "Trusted Computing", WuHan, Wuhan University Press, (2011) August, pp. 42-51.
- [9] S. B. Su and R. H. Huang, "Attribute hierarchical model based on triangular fuzzy number", System Engineering Theory and Practice, vol. 26, no. 12, (2006), pp. 115-119.
- [10] F. Kong, "Fuzzy Multiple Attribute Decision Making Theory Methods and Applications", Beijing Chinese Agricultural Science Press, (2008), pp. 79-85.
- [11] E. Bulut, O. Duru, T. Kececi, *et al.*, "Use of consistency index, expert prioritization and direct numerical inputs for generic fuzzy-AHP modeling: A process model for shipping asset management", Expert Systems with Application, vol. 39, no. 2, (2012), pp. 1911-1923.
- [12] J. Tian and P. Z. Zhang, "Based on expert opinion Delphi Method Integration Model", System Engineering Theory and Practice, vol. 24, no. 1, (2004), pp. 57-62, 69.

