The Multiple Attribute Group Decision Making model for Web Software Trustworthiness

Yujie Zhao
[1,] Business School of Central South University, Changsha 410083, China
*E-mail: yuanyuan860831@163.com

Xinxing Luo
[2] Business School of Central South University, Changsha 410083, China
E-mail: star@mail.csu.edu.cn

***Abstract*—The evaluation of web software trustworthiness is an urgent issue to tackle in the background of internet and information society. In this paper ,the Multiple Attribute Group Decision Making model for Web Software Trustworthiness (WST-MAGDM) is established: first, the ISO/IEC 0126 and web software attributes diagram is chose as the basic models and the simple web software trustworthy model as the contrast model, then ,the model is improved by using Principal Component Analysis (PCA) to questionnaire survey; second, the model is the foundation for building the web software comprehensive evaluation index system and evaluation grading criterion; finally, combing subjective and objective weighting methods in group decision-making, this paper can evaluate web software trustworthiness. Selecting planning-deployed internet banking from a bank in China can provide reference for enterprise decision-maker according to the decision rule set.**

***Index Terms*—Web Software Trustworthiness (WST); Multiple Attribute Group Decision Making (MAGD); Trustworthiness decision rule set; Internet banking.**

I. INTRODUCTION

As software is the soul of computer system, the dependence on computer system in modern society is mainly reflected in the dependence on software. Over the past decade, the development of internet technology has spawned a revolution of information technology, an extreme extension of software's applied field, a boost of social demand for software as well as constant changes of the requirement and exploiting environment for software. In terms of social demand, software shall be more complicatedly functioned and characterized by security, reliability, timeliness and so on. As to the operation and developing environment, it has been transferred from the traditionally static software to the dynamically changeable software under internet. Meanwhile, there exist both uncontrollability and uncertainty in the behavior of software entities, which proposes severe challenges to the traditional developing methods and techniques as well as the trustworthiness guarantee of operating software.

In the open-up internet environment, constructing trustworthy software system and providing a trustworthy operation guarantee to ensure that the software can provide more trustworthy service, which is the focus of international software industry and academia, and more is the key technical foundation for the future development of software technology and the development of information industry. The United States <the national software development strategy (2006-2015) > plays developing high trusted software on first, and puts forward the idea of the next generation of software engineering; National natural science foundation of China also launches the major research project - "trustworthy software basic research" which studies the theory and methods, measuring and modeling on trustworthy software with emphasis.

Software trustworthiness is impacting aerospace, nuclear reactor, military science, national defense construction, finance, securities, health care, automobile, electric power and other various high-tech sector of the national economy, it has been directly involved in the quality of people's lives, even to the human life and property security problems [1].In the internet development history, there are many complicated, expensive, and "successful" (i.e., payment has been made and delivered to the user) software systems failed because of not paying enough attention to trustworthiness. According to the United States authority Standish Group's CHAO Reports [2] each year which specializes in tracking IT project success or failure , the relevant data shows that more than 60% of all software projects finally ended in failure, which bring nearly 40% direct loss of the total American IT investment, and more indirect loss is incalculable. There are two well-known cases included the European Ariane 5 launch vehicle inertial guidance system anomalies in data conversion which led to the first launch failure [3] and "the London Ambulance System "which soon paralyzed because of they could not meet the trustworthiness requirements [4].

Web software is the birth of a new software system [5] which software paradigm is changing to networked and service-oriented under the rapid development of internet technology. The whole web software life cycle is during network environment, its development and deployment, its evolution and growth, its service and running are also on the network. Web software as a service carrier

doi:10.4304/jsw.9.3.763-776

provides on-demand online service for users through network resources polymerization. Web software with its openness plays an excellent example of the perfect polymerization and embodying of internet resources, technology and methods, which is one of most representative of trustworthy software. But at the same time, due to the uncertainty and complexity of its running environment, the dynamics and evolvability of its running mode, web software trustworthiness is a critical factor for web software success. Therefore, evaluating web software trustworthiness needs to be solved urgently.

This paper endeavors to do research on web software trustworthiness with fully consideration of the web software domain features,then studies multidimensional trustworthy attributes and the establishment, acquisition, measurement and evaluation for multi-scale quantitative and qualitative web software trustworthy attributes.

This section has given the general background to the study. Section Ⅱ outlines the entirely process of Multiple Attribute Group Decision Making model for Web Software Trustworthiness (WST-MAGDM). Section Ⅲ elaborates how to build the web software trustworthiness model. Section Ⅳ is the specific steps and algorithms of WST-MAGDM model. Planning-deployed internet banking from a Chinese bank is chose as a living sample in Section Ⅴ. Finally, a conclusion and future works are given in Section Ⅵ.

II THE WST-MAGDM MODEL

Software Trustworthiness(ST) refers to software behavior and results that can meet user expectations, and the ability to provide continuous service even when subject to interference [6]. Software Trustworthiness which portrays the overall software behavior is comprehensive evaluating a series of trustworthy attributes such as reliability, security, accuracy, timeliness, etc. The essence of software trustworthiness is choosing and evaluating attributes to users' expectations, and quantifying whether the measurements can meet users' expectations or not [7].

Measuring software trustworthiness involves a series of qualitative and quantitative trustworthy attributes. Early researches mostly pay attention to one-dimensional attribute's analysis, test and verification, but with more and more complex software application environment, when evaluating software trustworthiness, we should consider trustworthiness comprehensively from multidimensional attributes including reliability, security, availability, etc. Therefore, evaluating software trustworthiness can be seen as a kind of multiple attribute decision-making(MADM), and is a comprehensive measurement and evaluation process for software system features.

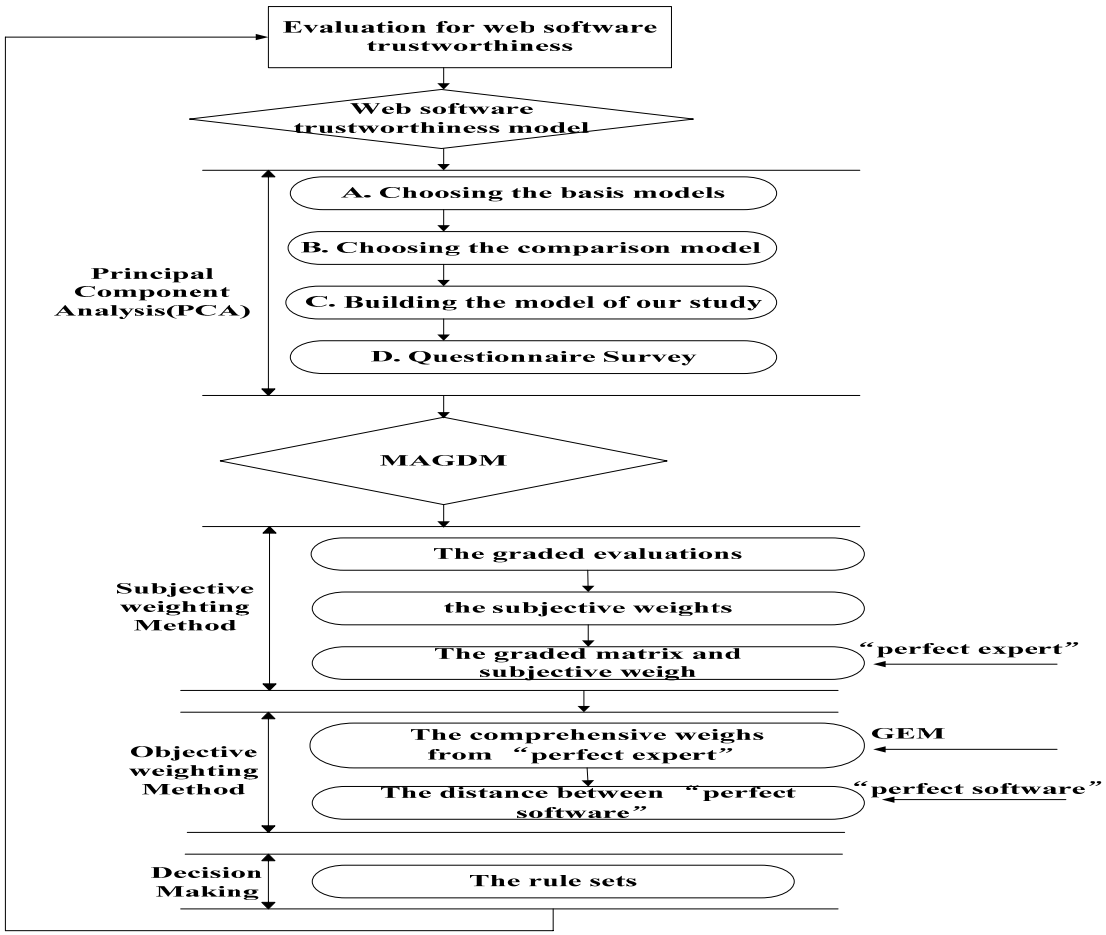

Fig.1 The entirely specific process of WST-MAGDM model

The state of the art in software trustworthiness has proposed many models and methods focusing on one-dimensional or multidimensional trustworthiness attributes. The typical researches include: unified trustworthiness model (UMD) [8], the Bayesian network model [9], the evaluation method based on ontology integration [10], the model with multi-dimensional attributes for evaluating web software [11], Markov usage models [12], and fuzzy method for evaluation multidimensional attributes [13], trustworthiness estimation model [14], self-adaptive trustworthiness model [15] and open source component method [16].

Summarizing the above-mentioned models, we can not hard to find that there are three shortcomings as follows: shortcoming one, the pro-existing models are not in combination with the practical application background and professional field of software, so they can not meet the specific software trustworthiness customizing; shortcoming two, the pro-existing models almost consider a one-dimensional trustworthy attributes in isolation and do not evaluate software trustworthiness as a whole; shortcoming three, even the pro-existing models consider multidimensional trustworthy attributes on the whole, but the evaluating criteria are not systematic and the measuring models are not structural.

According to these three shortcomings, this paper establishes the Multiple Attribute Group Decision Making model for Web Software Trustworthiness (WST-MAGDM model). After fully considering the unique web software trustworthy attributes, it builds an all-sided web software trustworthy model to make up shortcoming one. Then, according to the established web software trustworthy model, it establishes a comprehensive evaluation index system for web software trustworthiness including multi-dimensional trustworthy attributes and their evaluation criteria, which can make up shortcoming two. Finally, it applies group decision method with the consideration of the multi-dimensional web software trustworthy attributes to evaluate web software trustworthiness more scientifically and effectively, thereby to make up shortcoming three. Figure 1 is the specific process of this model; each step will be detailed in the following paragraphs.

III. The Model Of Web Software Trustworthiness

A The Basic Models for Our Study

For evaluating web software trustworthiness, domestic and overseas scholars already do some studies. Evangelos Grigoroudis etc. [17] use benchmark test to evaluate web software trustworthiness from user's angle, although there have been criticisms that the evaluating criteria is relatively dispersed and the model is non-structural. TIAN Xia etc. [18] study the network video software trustworthiness which is only a special case study. XIAO XIAO–Cong's model [19] uses grey correlation analysis to study web software, however it suffers from the lack of only focusing on quality not trustworthiness attributes. Doaa Nabil [20] proposes a basically conceptual model of web software trustworthiness on the basis of ISO/IEC9126 model, also with the disadvantage of not validating the validity. Behshid Behkamal [21] puts forward "web software attributes diagram" born out of studying B2B applications, but he also neglects the trustworthiness attributes.

The disadvantages of the above models can be generally divided into two types: first, only considering the web software quality attributes, not too much considering the trustworthy attributes; second, lacking the validation process for the established models Aiming at two disadvantages in previous study, this paper intends to build a more comprehensive and complete model for evaluating web software trustworthiness, and furthermore verifies the validation of the model. Here are two basic models of our research--ISO/IEC 9126 model and the web software attributes diagram.

Previous researches almost base on ISO/IEC9126 model which is the most accepted internationally and relatively mature model because of its versatility and authority. ISO/IEC9126 model has three layers including quality attributes, quality sub-attributes and quality factors, specifically consisting of 6 first-level attributes-functionality, reliability, usability, efficiency, maintainability and portability, and further divided into 21 sub-attributes.

The outstanding characteristics of ISO model are hierarchical structure, having evaluating criteria, comprehensive expression and terms, simple and accurate definition, and one-to-many relationship between different layers of the model. Therefore, ISO/IEC9126 model is chose as the base model of our study for evaluating web software trustworthiness. ISO/IEC9126 model only summarized the common attributes for general software, because web software is not-yet-emergent before, so ISO/IEC9126 model is not sufficient and abundance for customizing web software trustworthiness.

Web software attributes diagram proposed by Behshid Behkamal [21] is the most representative model for web software. It includes security, scalability, efficiency, and accessibility, etc. The attributes in this diagram are also verified in later study. For example, traceability, customizability, recoverability which never mentioned in ISO/IEC9126 model emerged in Doaa Nabil's [20] web software conceptual model, so web software attributes diagram have certain reference value and provides a good material for our study. But the shortcoming of web software attributes diagram is also obvious: first, without considering the trustworthy attributes of web software; second, simply dividing the web software quality attributes without enough carefulness.

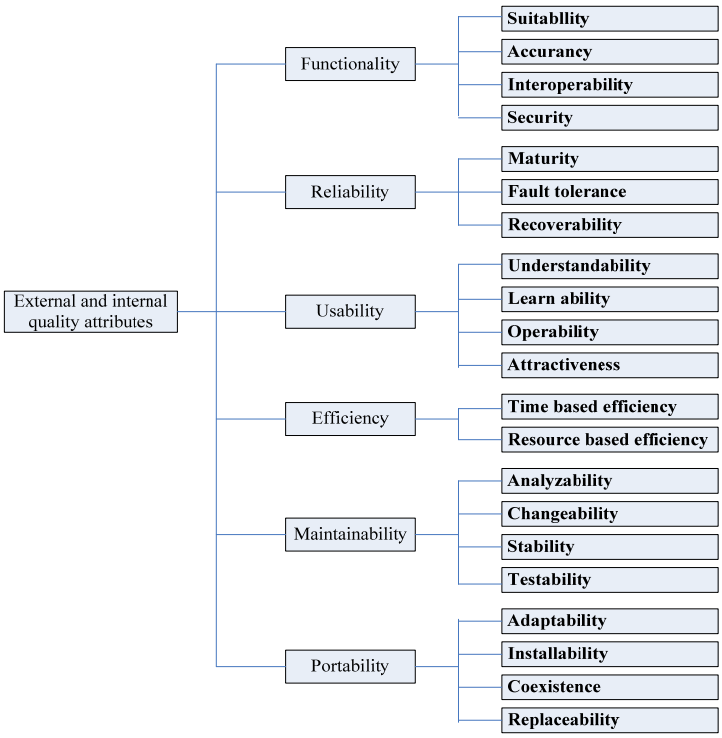

Fig.2 ISO/IEC9126 model

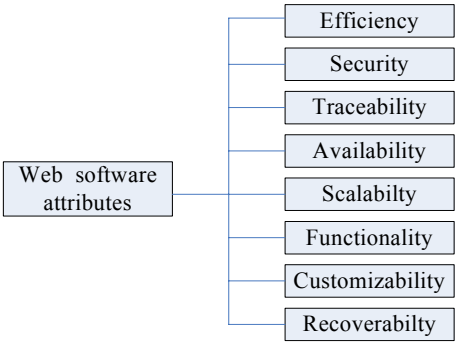

Fig.3 Web software attributes diagram

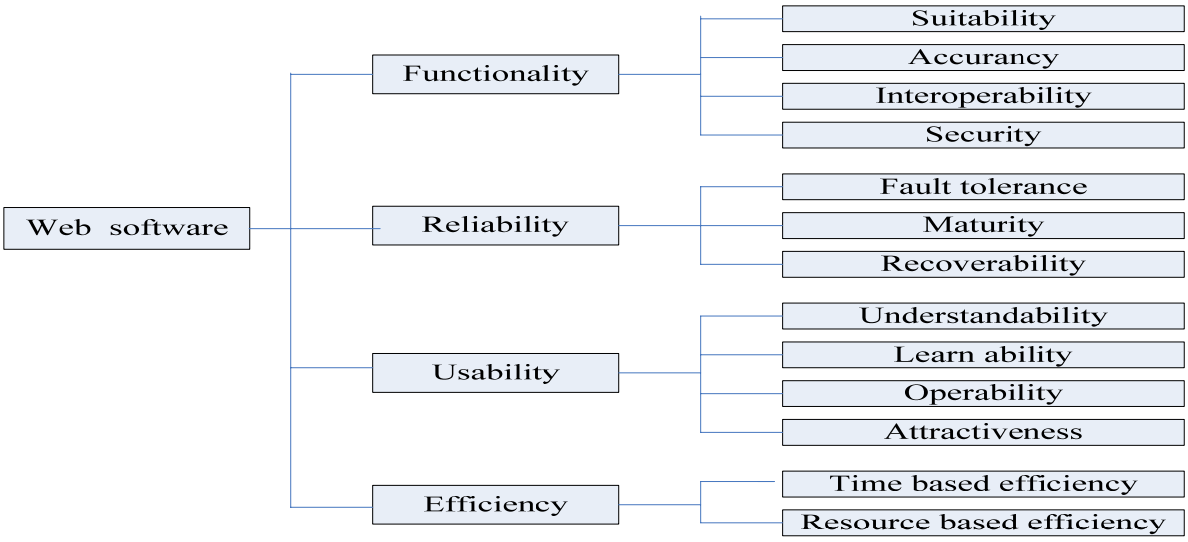

Fig.4 The simple web software trustworthy model

B The Contrast Model for Our Study

On the basis of ISO/IEC9126 model,Xin-Xing LUO etc. [22] proposes a simple web software trustworthy model for considering that web software mainly focus on external user satisfaction, so the model only chooses external attributes rather than internal attributes. It has certain reference significance for our study, but it is only a roughly preliminary model because of one-sided considering web software trustworthiness, so it can be used as the contrast model to verify the validity of our study.

C Building the Model for Our Study

ISO/IEC9126 model and web software attributes diagram provide good reference and guidance for our study. Combining carefully analysis of web software, this paper intends to build a model which has better structure and relatively complete for web software particularity trustworthiness .Here we will introduce the specific process of choosing trustworthy attributes, sub-attributes and improving the previous model.

Choosing of attributes and sub-attributes for our study follows three principles: (1) Web software trustworthiness focuses on external user satisfaction ,so we choose external attributes rather than internal attributes; (2) Considering the network environment of web software running and user's particularly attention on confidentiality of personal information, so we upgrade security to the secondary attribute; (3)According to trustworthy attribute synonymity, we only choose one of

the same attributes between “web software attribute diagram” and ISO/IEC9126 model.

According to the above three principles, 5 first-level trustworthy attributes for web software are preliminary chose: functionality, reliability, usability, efficiency, security, and 18 sub attributes are chose: maturity, fault tolerance, recoverability, understandability, learn ability, operability, attractiveness, accuracy, interoperability, adaptability, time based efficiency, resourced based efficiency, RSA Encipher, DES Encipher, traceability, accessibility, customizability and navigability. Among them, 5 first-level trustworthy attributes and the former 12 sub-attributes are already included in ISO/IEC9126 model, so we are not going to talk about them here, and only explaining in detail the new added 6 attributes into the model: RSA Encipher, DES Encipher, traceability, accessibility, customizability and navigability.

RSA Encipher, DES Encipher: they are mainly used for measuring software system confidentiality and information security [23]. As the reason for web software running on a network environment, users particularly concern about software information confidential.

Traceability: software's ability to handle information on different platforms correctly [22]. This feature is only appropriate for web software which can be track the operation of the system.

Accessibility: software’s ability to provide a specified service in required time [24]. It is a critical factor when web software runs on a network system specially.

Customizability: it refers to the web software can be added various services on user’s request to improve satisfaction degree [25]. Furthermore, customizability which can satisfy users’ customized services is one of the main reasons why users choose web software

Navigability: it refers to software trait for user easily accessing efficient information, and usually getting information through the following channels: the alphabet, subdirectory, sitemap (horizontal, vertical and mixed) [26]. Navigability can make web software to meet the user's convenience.

Portability, maintainability and the corresponding sub-attributes in ISO/IEC9126 model doesn't meet the three choosing principles, so they are not appeared in our study. In addition, the attributes reflected trustworthiness from "web software attributes diagram" should be chose. In the next step, we will solve the ownership between newly added sub-attributes and the first-level attributes.

In our study the model is composed of the attributes above-mentioned; we need to classify all the attributes which are in the same horizontal level to make the model hierarchy. Choosing trustworthy attributes and sub-attributes whether can apply or not in web software also need empirical validation. This paper takes Principal Component Analysis (PCA) to validate the model and divide the dimensions.

Principal Component Analysis (PCA) is a statistical method which can be used in the study of the dimension of a potential concept. The same dimension is composed of highly correlated factors, and the correlation coefficients of factors in the same dimension vary from 1 (perfect negative correlation) to 1 (perfect positive correlation), the greater the absolute value of correlation coefficient is, the stronger the correlation is, and the correlation coefficient of factors in the same dimension have the same signs. The general model of PCA can be expressed as:

$$\left.\begin{aligned} Z_1 &= P_{11}X_1 + P_{12}X_2 + P_{13}X_3 + \ldots + P_{1m}X_m \\ Z_2 &= P_{21}X_1 + P_{22}X_2 + P_{23}X_3 + \ldots + P_{2m}X_m \\ &\cdots\cdots \\ Z_m &= P_{m1}X_1 + P_{m2}X_2 + P_{m3}X_3 + \ldots + P_{mm}X_m \\ Y &= \frac{\lambda_1}{\sum_{i=1}^{m}\lambda_i}z_1 + \frac{\lambda_2}{\sum_{i=1}^{m}\lambda_i}z_2 + \frac{\lambda_3}{\sum_{i=1}^{m}\lambda_i}z_3 + \ldots \frac{\lambda_k}{\sum_{i=1}^{m}\lambda_i}z_k \end{aligned}\right\}$$

$X_1, X_2 \ldots X_m$ are the measured variables; $Z_1, Z_2 \ldots Z_m$ are principal component; k is the choose number of principal components.

$P_{ij}(i = 1,2,...,m; j = 1,2,...,m)$ is factor loading of measured variable j on the principal component i. The bigger of the factor loading means the closer correlationship between measured variable *j* and principal component *i*, and vice verse. *Y* is the predicting score standing for contribution rate for principal component *i*.

Applying PCA to validate the model and divide the dimensions, we can get factor loading values for each sub attribute on each principal component, and using the factor loading values to divide the dimensions. The cumulative for common factors can explain whether the new model can effectively express web software trustworthiness or not, so PCA here is feasible.

D Questionnaire Survey

This paper designs a questionnaire for the preliminary chose 18 sub trustworthy attributes. The measurement scale uses Likert five-point scale [14], from "very important", "important" and "general", "not important" to "very unimportant" , respectively giving 5, 4, 3, 2, 1 score. When we choose the samples, two cautions: first, the questionnaire has to be answered by the users who have rich experience and comprehensive understanding of web software; second, the samples can be easy to access. Based on these two cautions, we have a bifurcated questionnaire: one part is the ordinary users for web software: choosing 100 postgraduate students from software school, 92 valid paper questionnaires are collected by personal field interview; the other part is the professional users including research personnel, technical person and scholars for web software, 78 valid questionnaires are collected by sending 100 emails. The data have been preprocessing such as linearization and the dimensionless normalization before performing PCA. The preprocess uses very common methods so that won't be covered again here.

Doing PCA to the sample data by SPSS17.0 for the preliminary selected 12 sub-attributes in the contrast model- The simple web software trustworthy model. Then we choose the sub-attributes whose correlation coefficient are greater than 0.6. From Table 1, the cumulative contribution of four attributes in the contrast model only reaches 57.535% that can not obviously explain web software trustworthiness.

Table 2 shows the factor load matrix for the preliminary chose 18 sub-attributes in our study. We also choose the sub-attributes whose correlation coefficient are greater than 0.6, the load matrix shows that: the correlation coefficients of attractively and compatibility are less than 0.6,so these two sub-attributes should not be considered. The cumulative contribution of the five principal components' achieves 82.219% that is 24.684% higher than the contrast model's 57.535%, so our model is observably valid. The factors' score of newly added sub-attributes are all above 0.6, so the six sub attributes' influences are significant for web software trustworthiness.

According to the definitions for each attributes in ISO/IEC 9126 model, we named them and arrange their ownerships:4 sub-attributes included in common factor 1 all reflect the respondents' understanding of system reliability, so factor 1 is named reliability factor.

Similarly, factor 2 named usability, factor 3 named functionality, factor 4 named efficiency, factor 5 named security. Dimension classifications of newly added six sub-attributes are: accessibility belongs to reliability; navigability and customizability belong to usability; traceability belongs to usability; RSA Encipher and DES Encipher belong to security.

TABLE Ⅰ
FACTOR LODA MATRIX 1

Sub-attributes	The factor load			
	Reliability	Functionality	Usability	Efficiency
Maturity	0.707			
Fault tolerance	0.673			
Recoverability				
Understandability		0.723		
Learn ability		0.642		
Operability		0.638		
Attractiveness				
Accuracy			0.697	
Interoperability			0.604	
Adaptability				
Security			0.643	
Time based efficiency				0.683
Resourced based efficiency				0.605
Eigenvalue	1.864	1.784	1.463	1.329
Contribution (%)	18.694	17.973	11.043	9.825
Cumulative Contribution (%)	18.694	36.667	47.710	57.535

Note: The above attributes are significant under 5% confidence level

According the dimension classifications, this paper builds an improved web software trustworthiness model (Fig.4) from the verified common factors as the child nodes and the sub-attributes as the leaf nodes.

From Fig 4,the model of web software trustworthiness includes 5 first-level trustworthy attributes and 16 sub-attributes. Compared with the original model, the model has both hierarchies of the ISO model and web software trustworthy attributes. It upgrades security to secondary attribute, which is one of the goals in Trustworthy Computing Microsoft White Paper [27], and adds six web software trustworthy attributes- traceability, accessibility, customizability, navigability, RSA Encipher and DES Encipher. Although having the same name, the sub-attributes belonging to the five improved attributes are different from which are mentioned in ISO/IEC model, so they can better express web software trustworthiness.

TABLE II
FACTOR LODA MATRIX 2

Sub-attribute	The factor load				
	Common factor 1	Common factor 2	Common factor3	Common factor 4	Common factor 5
Maturity	0.877				
Fault tolerance	0.773				
Recoverability	0.629				
Accessibility^	0.637				
Understandability		0.853			
Learn ability		0.665			
Operability		0.687			
Navigability^		0.697			
Customizability^		0.614			
Attractiveness					
Accuracy			0.787		
Interoperability			0.638		
Adaptability					
Traceability^			0.764		
Time based efficiency				0.767	
Resourced based efficiency				0.638	
RSA Encipher^					0.738
DES Encipher^					0.684
Eigenvalue	2.597	2.516	1.683	1.425	1.472
Contribution (%)	22.615	21.942	13.998	11.710	11.954
Cumulative Contribution (%)	22.615	44.557	58.555	70.265	82.219

Note: ^ stands for the newly added sub-attributes in the model
The above attributes are significant under 5% confidence level

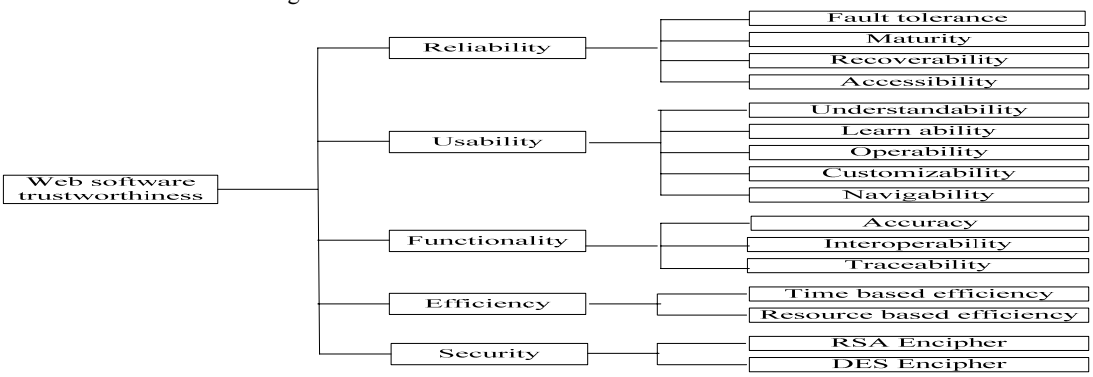

Fig.4 The web software trustworthiness model

TABLE III
THE TRUSTWORTHY ATTRIBUTES SPECIFICATION FOR WEB SOFTWARE

Trustworthy attributes	Sub-attributes	Attribute specification
Reliability	Fault tolerance	The web software ability that can avoid system failure caused by propagation of the external interface errors (mainly the isolation of external error).
	Maturity	The web software ability that can avoid system failure caused by propagation of the internal interface errors (mainly the isolation of internal error).
	Recoverability	The web software ability that can recovery original function and performance if an error occurred.
	Accessibility	The web software ability that can provide specified services in requires time.
Usability	Understandability	The web software ability that can provide clear, accurate and simple information which users can quickly understand.
	Learn ability	The web software ability that can provide applications which users can learn.
	Operability	The web software ability that can be easily operated and controlled by users.
	Customizability	The web software ability that can add required services to improve users' satisfaction.
	Navigability	The web software ability that can provide valuable information to meet users' convenience.
Functionality	Accuracy	The web software ability that provide required-accurate functions to conform the target.
	Interoperability	The web software ability that can use information when interacting with other systems.
	Traceability	The web software ability that can process information correctly and track system operation effectively on different platforms
Efficiency	Time based efficiency	The response time that web software handle specific operation request.
	Resource based efficiency	The system resource consumption that web software handle specific operation request.
Security	RSA Encipher	The information security when web software running on the network (mainly permission problems).
	DES Encipher	The data security when web software running on the network (mainly password problems).

IV THE WST-MAGDM MODEL

A The Comprehensive Evaluation Index System and the Measurement

This section will specify each trustworthy attribute (Table.3) in web software trustworthiness model established in section 3, and then builds web software trustworthiness comprehensive evaluation index system which contains trustworthy attributes, and sub-attributes with their measurement criterion. The original measurements are determined by the trustworthy attributes' essence, and the grading measurements from level 5 to level 1 for each trustworthy attribute is on the basis of the original measurement (level 5 for the best, level 1 for the worst). According to original measurement formula, we can determine the level for each trustworthy attribute.

Table 4 shows 16 trustworthy attributes, the former 12 attributes are numerical whose original values can be got through testing data when software running in a real environment, the corresponding grades are: level 5: $X=1$; level 4: $0.85 \le X < 1$; level 3: $0.7 \le X < 0.85$; level 2: $0.5 \le X < 0.7$; level 1: $0 \le X < 0.5$.The latter 4 attributes are non-numerical whose grades can be got through expert's experience and practical situation according to the original measurement definition.

B The Concrete steps for the WST-MAGDM Model

On the basis of established comprehensive evaluation index system for web software trustworthiness, this paper uses the Multiple Attribute Group Decision-Making (MAGDM) method to measure web software trustworthiness. Using group decision making method to comprehensively consider the multidimensional trustworthy attributes of web software that can measure the web software trustworthiness more scientifically and effectively.

First of all, the grading measurement and subjective weight for each trustworthy attribute by experts in group decision making can be got with using subjective weighting method. Considering the limitation of experts' knowledge in group decision making, this paper assumes existing a "perfect expert", then uses Group Digenvalue Method (GEM) to get evaluation matrix and the subjective weight for each trustworthy attribute. Then, this paper uses the entropy method which is one of objective weighting method to modify the group decision results. Finally, it calculates the distance between the evaluated software and the "perfect software" to measure software trustworthiness.

TABLE Ⅳ
THE COMPREHENSIVE EVALUATION INDEX SYSTEM AND THE GRADING MEASUREMENT

Trustworthy attributes	Sub-attributes	The original measurement definitions	The grading (level 5 to 1)
Reliability	Fault tolerance	X=1- external error occurred times/ software failure times	{best, better, medium, worse, worst}
	Maturity	X=1- internal error occurred times/ software failure times	{maturest,maturer,medium, immaturer,immaturest}
	Recoverability	X=software attack times/successful recovery times	{best, better, medium, worse, worst }
	Accessibility	X=1-the proportion of functions that can't provide service in time	{easiest, easier, medium, harder, hardest}
Usability	Understandability	X=1-the proportion of functions that can't be understood by users	{best, better, medium, worse, worst}
	Learn ability	X=1-the proportion of functions that can't be studied by users	{best, better, medium, worse, worst}
	Operability	X=1-the proportion of functions that isn't easy to operate by users	{easiest, easier, operable, harder, hardest}
	Customizability	X=1-the proportion of functions that can't be added the users' required service	{best, better, medium, worse, worst}
	Navigability	X=1-the proportion of functions that can't provide service according to users' convenience	{best, better, medium, worse, worst }
Functionality	Accuracy	X=1-the proportion of functions that can't meet the users required accuracy	{best, better, medium, worse, worst }
	Interoperability	X=1-the proportion of functions that can't use information when interacting with other systems.	{easiest, easier, medium, harder, hardest}
	Traceability	X=1-the proportion of functions that can't track system operation on multiple platforms.	{best, better, medium, worse, worst}
Efficiency^	Time based efficiency^	Time difference between getting the users' required results and completing command.	{fastest, faster, ordinary, slower, slowest}
	Resource based efficiency^	The size of memory usage, CPU utilization rate etc.	{best, better, medium, worse, worst}
Security^	RSA Encipher^		{Verified Protection, mandatory protection, Controlled access Protection, Discretionary Security Protection, non-protection}
	DES Encipher^		

Note: ^ stands for non-numerical trustworthy attributes, others are numerical trustworthy attributes

The advantage of the model is combining subjective and objective weighting methods, and both reflecting the subject opinion from group decision making experts and considering the inner objective laws between trustworthy attributes. Judging by the subject opinion from group decision making experts in the initial phase of evaluating software trustworthiness, with the steady accumulation of the sample data, modifying the weight from the inner objective laws between trustworthy attributes, forming learning samples based on multiple decisions and transforming experience to knowledge. As the decision-making process carrying out, the measurement of software trustworthiness is tending towards more scientifical and reasonable [28] [29] [30].

C Synthesizing Expert Opinion in Group Decision-making

Given l experts grading evaluate n trustworthy attributes in m software (level 5 to 1 and lever 5 is the best), and getting the evaluation matrix $X^k = \left(x_{ij}^k\right)_{m\times n} (i=1,2,...,m; j=1,2,...,n; k=1,2,...,l)$, then unifying trustworthy attributes' dimension by normalization equation formula (1) and transforming $X^k = \left(x_{ij}^k\right)_{m\times n}$ to normalized matrix $R^k = \left(r_{ij}^k\right)_{m\times n}$.

$$r_{ij}^k = \frac{x_{ij}^k - \min_i x_{ij}^k}{\max_i x_{ij}^k - \min_i x_{ij}^k} \quad (1)$$

Taking the limitation of experts' knowledge in group decision making into consideration,there couldn't possibly exist a expert who can make perfect decision in the real world, so this paper assumes a "perfect expert" whose decision is most "perfect", namely, reliability at 1 and uncertainty at 0. The "perfect expert" grading vector angle between other experts is the minimum, therefore he (she) is most consistent with all experts in group

decision-making. The "perfect expert" grading is recorded as $X^* = \left(x_1^*, x_2^*, ..., x_n^*\right)^T$, and X^* can be got by solving formula (2).

$$\max_{\|b\|=1} \sum_{i=1}^{m} \left(b^T X_i\right)^2 = \sum_{i=1}^{m} \left(X^T X_i\right)^2 = \rho_{\max} \qquad (2)$$

$\rho_{\max}$ is the largest eigenvalue of matrix $X^T X$, X^* is the corresponding positive eigenvector $\|X^*\| = 1 \quad \forall b = \left(b_1, b_2, ..., b_n\right)^T \in E^n, \|b\| = 1$. Getting the experts' normalized weights vector after normalizing X^* .Then, getting the "perfect expert" normalized grading matrix $R^* = \left(r_{ij}^*\right)_{m\times n}$ on the basis of the m software grading $Z^j = \left(z_{ki}^j\right)_{l\times m}$ matrix $\left(k = 1, 2, ..., l; j = 1, 2, ...n; i = 1, 2, ..., m;\right)$ from l experts. z_{ki}^j means the grading result from the *kth* expert of the the *jth* trustworthy attributes in the *ith* software, and $b^j \cdot Z^j$ means the grading vector from the "perfect expert" of the *jth* trustworthy attributes in all software. Finally, group decision-making can be transformed into single-person decision

D The Comprehensive Weighs for Trustworthy Attributes

Because of the different relative importance of each trustworthy attribute, the experts in group decision making are required to give subjective weight $\lambda_j^{k'}$ for n trustworthy attributes. Integrating all experts' opinion by Group Digenvalue Method (GEM), then solving the "perfect expert" subjective weight $\lambda_j^{*'}$ for each trustworthy attributes, at last, modifying the weight by objectively entropy weighting method.

When evaluating n trustworthy attributes from m software, defined the entropy of the *jth* $\left(j = 1,\ 2 \ldots\ n\right)$ attribute as:

$$H_j^* = -k \sum_{i=1}^{m} f_{ij}^* \cdot \ln f_{ij}^* \qquad (3)$$

Among, $f_{ij}^* = \dfrac{r_{ij}^*}{\sum_{i=1}^{m} r_{ij}^*}$; $k = \dfrac{1}{\ln m}$; $H_j \geq 0; k \geq 0$.

Defined the entropy weight w_j^* for the jth trustworthy attribute as

$$w_j^* = \frac{1 - H_j^*}{n - \sum_{j=1}^{n} H_j^*} \qquad (4)$$

Entropy [31] which is used to measure information's out- of- order statement, and it is the concept on information theory [32]: the bigger the degree of variation for trustworthy attribute is, the smaller the entropy is, and the greater the amount of information provide by the trustworthy attribute is, so the smaller the weight of the trustworthy attribute ; conversely, the smaller the degree of variation for trustworthy attribute is, the bigger the entropy is, and the lesser the amount of information provide by the trustworthy attribute is; so the bigger the weight of the trustworthy attribute is. Measuring objective weight for each attribute by entropy weight and then combining the subjective opinion from expert in group decision making, the comprehensive weight for each trustworthy attribute can be got from formula:

$$\lambda_j^* = \frac{\lambda_j^{*'} w_j^*}{\sum_{j=1}^{n} \lambda_j^{*'} w_j^*} \qquad (5)$$

E Evaluating software trustworthiness

Through above-mentioned steps, we can get the grading matrix and comprehensive weight for each trustworthy attribute from the "perfect expert". For the software to be measured, the trustworthiness can be judged by the distance Y^* from the "perfect software". Calculating respectively the distance d_p between n trustworthy attributes for each software and the "perfect software" Y^*, and $Y^* = \left(1, 1, ..., 1\right)_{1\times n}$

$$d_p = \left[\sum_{j=1}^{n} \lambda_j^{*p} (1 - r_{ij}^*)^p\right]^{\frac{1}{p}} \qquad (6)$$

Evaluating the software trustworthiness according to the d value, the smallest d value means the smallest distance from the "perfect software, so the software trustworthiness is the best; calculating d' when d values are same, the smaller the d' value is, the better the software trustworthiness is.

$$d = 1 - \sum_{j=1}^{n} \lambda_j^* r_{ij}^* \qquad (7)$$

$$d' = \left[\sum_{j=1}^{n} \lambda_j^{*2} \left(1 - r_{ij}^*\right)^2\right]^{\frac{1}{2}} \qquad (8)$$

Finding d corresponding evaluation interval from Table 5 and the relevant decision rule can aid the decision-maker

TABLE Ⅵ
THE DECISION RULE SET FOR TRUSTWORTHINESS

$L(ST)$	$G(ST)$	$d(ST)$	$DR(ST)$
T1	Completely untrustworthy	[0.75,1]	Refused to apply
T2	Positive untrustworthy	(0.55,0.75]	Re-evolution after substantial modification
T3	Slightly untrustworthy	(0.35,0.55]	Re-evolution after minor modification
T4	Slightly trustworthy	(0.25,0.35]	Apply after global modification
T5	Positive trustworthy	(0.15,0.25]	Apply after local optimization
T6	Significantly positive trustworthy	(0.05,0.15]	Risk-free of application
T7	Completely trustworthy	(0,0.05]	Apply with no changes

Note: $L(ST)$ stands for the software trustworthiness level, and the higher the level is, the better the software trustworthiness is.

$G(ST)$ stands for the correspondingly specific grading for software trustworthiness.

$d(ST)$ stands for the evaluation interval for software trustworthiness.

$DR(ST)$ stands for the decision rule for software trustworthiness.

VI SPECIFIC EXAMPLE AND APPLICATION

Our study chooses internet banking as the living example. The reason is chiefly as follows: Internet banking software is an outstanding kind of web software. The limit of space and time gradually cancels with the realization of electronization, informatization and networking; the scope of bank business has been expanded from traditional bar business to interne business and moving forward comprehensiveness and almightiness. Internet banking web software is called online banking, network banking, and is one kind of extremely typical web software. Internet banking is also called 3A banking because that it can provide the business online such as the opening, the closing, the inquiring and the transferring client account, furthermore the credit and financial service business etc., so clients can deal with these business without leaving home. 3A means that internet banking can provide business at any time, any place, and any way. With the advantages of electronization, networking, systematization and paperless, the trustworthiness of internet banking cannot be ignored.

Our study choose a planning-deployed internet banking program from one Chinese bank, and get 4 short listed software solutions- S1, S2, S3 and S4 after primary election. According to the established web software comprehensive trustworthiness evaluation index system and using MAGDM to sort the 4 short listed software, and then providing reference for enterprise decision maker.

Our study choose each 2 from users, developers and scholars for internet banking , these 6 experts establish a decision-making group, the group is assigned to grading 16 trustworthy attributes from level 5 to 1 in the 4 solutions and the basic of group decision-making is the original and grading measurement mention in Table 1. Due to the limit of paper space, here we only list the first expert corresponding grading and subjective weight $\lambda_j^{k'}$, other 5 experts' opinion can also list as Table 6.

TABLE Ⅶ
THE FIRST EXPERT'S CORRESPONDING GRADING AND SUBJECTIVE WEIGHT $\lambda_j^{k'}$

A_i / S_i	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
S_1	5	3	4	2	3	4	4	2	2	4	3	2	5	3	2	1
S_2	4	4	2	5	3	3	5	1	2	3	4	3	4	2	5	3
S_3	5	4	3	4	4	5	3	4	4	3	3	5	4	3	4	2
S_4	4	5	2	3	2	2	3	4	5	2	1	4	3	2	3	4
$\lambda_j^{k'}$	0.05	0.05	0.1	0.05	0.1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.1	0.1

From formula (1), normalizing the 6 experts' evaluation matrix for 16 trustworthy attributes and getting matrix $R_j\left(j=1,2,...,16\right)$. Then calculating the "perfect expert" evaluation for 16 trustworthy attributes in the 4 solutions, Z^1 is the normalized evaluation matrix for trustworthy attribute A1-Fault tolerance from 6 experts.

$$Z^1=\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0.500 & 0 & 0.500 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0.500 & 0.500 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Getting the the largest eigenvalue of matrix Z^1Z^{1T} is 6.13327, the corresponding positive eigenvector is $(0.4550, 0.4250, 0.2772, 0.3117, 0.4232, 0.5092)$,

$b_1=(0.1895, 0.1770, 0.1154, 0.1298, 0.1762, 0.2121)$

is the normalized eigenvector .

$Z^{1*}=b^1\cdot Z^1=(0.3661, 0.2035, 0.7961, 0.5045)$ is the attribute A_1-Fault tolerance evaluation vector from the "perfect expert "in four short listed software. Similarly, the "perfect expert" evaluation matrix for other 15 attributes in four short listed software can be got by Minitab16.

$$Z^*=\begin{bmatrix} 0.3661 & 0.4426 & 0.4683 & .... & 0.5986 \\ 0.2035 & 0.2098 & 0.3928 & .... & 0.6784 \\ 0.7961 & 0.6210 & 0.2750 & .... & 0.8674 \\ 0.5045 & 0.8361 & 1 & .... & 0.2310 \end{bmatrix}_{4\times16}$$

$Z=\left(Z^{1*},Z^{2*},....,Z^{16*}\right)^T$ is the evolution matrix from the "perfect expert" for all 16 trustworthy attributes in the index system.

Similarly, the subjective evaluation weight $\lambda_j^{*'}$ from the "perfect expert" is:

$$\lambda_j^{*'}=(0.023, 0.038, 0.089, 0.022, 0.070, 0.055, 0.035, 0.070, 0.032, 0.076, 0.069, 0.057, 0.016, 0.059, 0.129, 0.161)$$

Defined R^* as the normalized Z^*, the entropy H_j^* and the entropy weight w_j^* can be got by formula (3) and (4), and then calculating the compressive entropy weight λ_j^* by formula (5). Table 7 shows the calculated results.

TABLE Ⅷ

THE ENTROPY H_j^* AND THE COMPREHENSIVE ENTROPY WEIGHT λ_j^*

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
H	0.778	0.625	0.564	0.789	0.658	0.459	0.658	0.316
w_j^*	0.029	0.050	0.058	0.028	0.045	0.071	0.045	0.090
λ_j^*	0.009	0.026	0.069	0.008	0.042	0.053	0.021	0.085
	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
H	0.689	0.2585	0.331	0.445	0.845	0.421	0.372	0.216
w_j^*	0.041	0.098	0.088	0.073	0.020	0.076	0.083	0.103
λ_j^*	0.018	0.100	0.081	0.056	0.004	0.061	0.143	0.223

Finally, the four distances between the short listed software solutions and the "perfect software" can be got from formula (7) and (8).

TABLE IX
DISTANCES FROM THE "PERFECT SOFTWARE"

	S1	*S2*	*S3*	*S4*
d	0.389	0.321	0.664	0.229
d'	0.232	0.198	0.226	0.065
L(ST	T3	T4	T2	T5

From Table 8, the trustworthiness sorting of the four short listed software is $S_4 \succ S_2 \succ S_1 \succ S_3$, so the decision-maker should give priority to the fourth internet banking software solution under the same other conditions.

VII CONCLUSION AND FUTURE WORKS

Although software trustworthiness research has become one of the hot research fields, previous studies mostly focus on isolated one-dimensional trustworthy attributes, rarely considering the multidimensional trustworthy attribute and the comprehensive measurement method for software trustworthiness. Web software has become one of the most popular software used in people's daily life, so this paper establishes the WST-MAGDM model which can evaluate the trustworthiness of web software reasonably and effectively.

In general, the contributions of this paper can be summarized into three areas: applying PCA to improve the web software trustworthiness model from questionnaire survey; building web software comprehensive evaluation index system, original measurement formula and grading measurement criterion; combing group decision-making, entropy weighting method and decision rule set for web software trustworthiness to provide reference for enterprise decision maker. Finally applying the WST - MAGDM model to select a planning-deployed internet banking program from one Chinese bank can specify the model steps and algorithms, and further verify the rationality and validity of the model. The method for establishing model can be also used in other software fields similarly.

Software trustworthiness evaluation is a complex process, and in the next step of study we still need to strengthen the modern methods such as meta-model, data mining, formal semantic analysis, the quantitative method and other key areas' technology research, and further improve the model with the consideration of the relationship between multidimensional attributes.[33][34][35]

ACKNOWLEDGMENT

The research work presented in this paper was supported by National Natural Science Foundation of China (Grant Nos.71221061).

REFERENCES

[1] Liu K, Shan Z G, Wang J, et al. Overview on major research plan of trustworthy software[J]. Bulletin of national natural science foundation of China, 2008, 22(3): 145-151.
[2] Molokken K, Jorgensen M. A review of software surveys on software effort estimation[C]//Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on. IEEE, 2003: 223-230.
[3] Paech, Barbara, Allen H. Dutoit, Daniel Kerkow, and Antje Von Knethen. "Functional requirements, non-functional requirements, and architecture should not be separated œA position paper." (2002).
[4] Finkelstein A, Dowell J. A comedy of errors: the London Ambulance Service case study[C]//Proceedings of the 8th International Workshop on Software Specification and Design. IEEE Computer Society, 1996: 2.
[5] Casado R, Younas M, Tuya J. Multi-dimensional criteria for testing web services transactions[J]. Journal of Computer and System Sciences, 2013.
[6] Zhu M X, Luo X X, Chen X H, et al. A non-functional requirements tradeoff model in Trustworthy Software[J]. Information Sciences, 2012, 191: 61-75.
[7] Şen C G, Baraçlı H, Şen S, et al. An integrated decision support system dealing with qualitative and quantitative objectives for enterprise software selection[J]. Expert Systems with Applications, 2009, 36(3): 5272-5283.
[8] Basili V, Donzelli P, Asgari S. A unified model of dependability: Capturing dependability in context[J]. Software, IEEE, 2004, 21(6): 19-25.
[9] Uzunov K, Nguyen T. Dependability of software in airborne mission systems[J]. 2008.
[10] Paul R, Yen I L, Bastani F, et al. An ontology-based integrated assessment framework for high-assurance systems[C]//Semantic Computing, 2008 IEEE International Conference on. IEEE, 2008: 386-393.
[11] Whitworth B, Fjermestad J, Mahinda E. The web of system performance[J]. Communications of the ACM, 2006, 49(5): 92-99.
[12] Gutjahr W J. Software dependability evaluation based on Markov usage models[J]. Performance evaluation, 2000, 40(4): 199-222.
[13] Huiling S, Jun M, Fengyi Z. Software dependability evaluation model based on fuzzy theory[C]//Computer Science and Information Technology, 2008. ICCSIT'08. International Conference on. IEEE, 2008: 102-106.
[14] Ma Y, Zhang M. A computation model of trustworthy degree[C]//Intelligent Information Technology Application Workshops, 2008. IITAW'08. International Symposium on. IEEE, 2008: 523-526.
[15] Lin M, Yongsen X. An adaptive dependability model of component-based software[J]. ACM SIGSOFT Software Engineering Notes, 2003, 28(2): 10.
[16] Zhang Y, Fang B, Xu C. Trustworthy metrics models for internetware[J]. Wuhan University Journal of Natural Sciences, 2008, 13(5): 547-552.
[17] Grigoroudis E, Litos C, Moustakis V A, et al. The assessment of user-perceived web quality: Application of a satisfaction benchmarking approach[J]. European Journal of Operational Research, 2008, 187(3): 1346-1357.
[18] Xia T, Mei T, Hua G, et al. Visual quality assessment for web videos[J]. Journal of Visual Communication and

Image Representation, 2010, 21(8): 826-837.
[19] Xiao X C, Wang X Q, Fu K Y, et al. Grey Relational Analysis on Factors of the Quality of Web Service[J]. Physics Procedia, 2012, 33: 1992-1998.
[20] Nabil D, Mosad A, Hefny H A. Web-Based Applications quality factors: A survey and a proposed conceptual model[J]. Egyptian Informatics Journal, 2011, 12(3): 211-217.
[21] Behkamal B, Kahani M, Akbari M K. Customizing ISO 9126 quality model for evaluation of B2B applications[J]. Information and software technology, 2009, 51(3): 599-609.
[22] LUO X, ZHU M, CHEN X. Non-Functional Requirement Conflict Tradeoff FO-QSIG Model in T rustworthy Software [J][J]. Systems Engineering, 2010, 2: 101-105.
[23] Luo X X, Zhu M X, Tang Z J. Literature review on trustworthy software acquisition and analysis[J]. Application Research of Computers, 2010, 10: 006.
[24] Davis A, Overmyer S, Jordan K, et al. Identifying and measuring quality in a software requirements specification[C]//Software Metrics Symposium, 1993. Proceedings., First International. IEEE, 1993: 141-152.
[25] Tang Z, Liu S. The Constructing Method of Meta-requirement Analysis Model[C]//Business Intelligence and Financial Engineering (BIFE), 2010 Third International Conference on. IEEE, 2010: 118-121.
[26] Luo X X, Wang W F. Research on meta-requirement model and meta-requirement set based on ontology[J]. Application Research of Computers, 2012, 4: 055.
[27] Mundie C. Trustworthy computing. Microsoft white paper. October 2002[J].
[28] LIU Y, ZHOU H, YAN K. A Method for Urban Parking Facility Planning Based on Entropy Group Decision-making[J]. Systems Engineering, 2007, 2: 006.
[29] Wanhua Q.An eigenvalue method on group decision[J]. Applied Mathematics and Mechanics, 1997, 18(11): 1099-1104.
[30] Tianlei Z, Zhengyou H, Qingquan Q. Distribution Network Service Restoration Multiple Attribute Group Decision-making Using Entropy Weight and Group Eigenvalue[C]//Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on. IEEE, 2013: 602-606.
[31] Kapur J N, Kesavan H K. Entropy optimization principles with applications[M]. Academic Pr, 1992.
[32] Maxwell's Demon 2 Entropy, Classical and Quantum Information, Computing[M]. CRC Press, 2010.
[33] Hu W, Hu Z, Cheng Y, et al. Modeling and Simulation on Dynamic Allocation and Scheduling of Multi-resource problem[J]. Journal of Computers, 2011, 6(7): 1369-1377.
[34] Lin G, Bie Y, Lei M. Trust Based Access Control Policy in Multi-domain of Cloud Computing [J]. Journal of Computers, 2013, 8(5): 1357-1365.
[35] Zhang L, Wang R, Wang H. Trusted decision mechanism based on fuzzy logic for open network [J]. Journal of Computers, 2008, 3(12): 76-83.

Yu-jie Zhao She is currently a PhD candidate in the Business School of Center South University. Her research interests include Trustworthy Software Evaluation, Management Information System,Quantitative Analysis, and Decision Support Systems.

Xin-xing Luo He is a full Professor of Business School of Center South University. He has more than 30 years of experience in Management Information System,Decision Support System, Expert System and Database System. He got Japanese Ministry of Education Scholarship in 1987-1989, and advanced studied for “university information management” as a faculty member of trainees in Yokohama National University. In 1998-1999, he was funded by China Scholarship Council as a visiting scholar. Dr. Xin-xing LUO has published more than 40 international journals and conference articles.