Using ELECTRE TRI Outranking Method to Evaluate Trustworthy Software

Gang Lu, Huaimin Wang, and Xiaoguang Mao

Computer School, National University of Defense Technology,
410073 ChangSha, China
lugang@263.net

Abstract. Trustworthy software evaluation is taken as the multi-criteria decision aiding process in this paper. The use of ELECTRE TRI method for evaluating software trustworthiness is presented. Software under evaluation is compared with some predefined norms and is assigned to one of trust levels. The entire evaluating process is described, including definition of problem situation and formulation, determination of the model and its parameters, and the application of the model. A metric for trustworthiness and an attributes weighting method are also presented. Some practical considerations are discussed in the final part of the paper.

Keywords: trustworthy software, evaluation, models, ELECTRE TRI method, MCDA.

1 Introduction

In the past few years there has been increased in the demand for more trustworthy software and services to cope with the growing scale and complexity of computing systems. Evaluating trustworthiness is getting more important in research on trustworthy software and services. Generally, software trustworthiness is viewed as a composite attribute; it's a multidimensional concept [1][2][3]. A lot of researches on trustworthiness evaluation focus on a single or a few features, and little attention is paid to evaluating it in a comprehensive way.

During our study on trustworthy resources management on internet, we found that if the potential users lack an easy, effective, and trustworthy process to justify the software trustworthiness, they will be confronted with considerable complexity in making a decision on software selection. Rating the software trustworthiness can not only reduce the cognitive effort required from the users in evaluating phase, but also guide the developing projects to the direction of trustworthiness evolution.

Since software trustworthiness is a multidimensional concept, the Multicriteria Decision Aid (MCDA) approach is adopted as a powerful tool taking several concerns into account to achieve a compromise evaluation solution. Roy distinguished four types of MCDA problems [4]:

1. Choice problematic $P.\alpha$: help choose a best alternative;
2. Sorting problematic $P.\beta$: help sort actions according some predefined norms;

B. Xie et al. (Eds.): ATC 2010, LNCS 6407, pp. 219–227, 2010.

3. Ranking Problematic $P.\gamma$: help rank alternatives with partial order;
4. Description Problematic $P.\delta$: help describe alternatives in a formalized way.

$P.\alpha$ and $P.\gamma$ base their results on the pairwise comparisons of alternatives, but in $P.\beta$ each alternative is considered independently from others and the result only depends on the intrinsic values of the alternative and the predefined norms. Evaluating and selecting software packages is normally $P.\alpha$ or $P.\gamma$ for acquisition managers. However, from the perspective of our management on trustworthy resources, rating software trustworthiness is $P.\beta$.

Based on their aggregation procedures, MCDA methods are classified into two categories: compensatory and noncompensatory. The weighted average sum (WAS) aggregation technique used in everyday evaluations is compensatory, so are some other techniques based on multi-attribute utility theory. ELECTRE TRI outranking method [5][6] is a noncompensatory MCDA method and suitable for the features of our trustworthiness evaluation:

1. dimensions of trustworthiness are not generally considered to compensate each other;
2. the measurements of trustworthiness dimensions may be imprecise, and/or uncertain, and/or inaccurate.

In addition, software submission mode on the internet is open for every supplier in our study. The supplier often does not present sufficient evidences for proving that the software can be justified trustworthy, and providing each software package with some specific evaluators was neither practical nor necessary. Therefore, the automation or less labor during the evaluation is desired.

2 Software Trustworthiness Evaluation

For nearly 30 years, numerous models and methodologies have been developed for evaluating and selecting software [7][8]. These models and methodologies focus on different features of software and evaluation process to meet the needs of different motivations and purposes of evaluation. Morisio and Tsoukias design a methodology to evaluate software products in a formal and rigorous way, namely Iusware, which is based on the MCDA approach and encompasses the activities during a software product evaluation process [9]. Stamelos and Tsoukias provide a partial list of software evaluation problem situations based on Mosirio's work [10]. Following the terminology defined in [9][10], we describe the software trustworthiness evaluation process (STEP) as:

1. define problem situation of trustworthiness evaluation;
2. define problem formulation;
3. determine evaluation model and its parameters;
4. apply evaluation model.

The details of the STEP are presented below.

2.1 Problem Situation and Problem Formulation

Problem Situation (PS) refers to intrinsic structure and external context of problem. Our PS for trustworthiness evaluation can be defined as a triple $< A_{\mathrm{PS}}, O_{\mathrm{PS}}, RS >$ where:

A_{PS} is the set of actors involved, including:
- the software suppliers;
- the trustworthy resources manager;
- the users(evaluating users and potential users);
- the evaluator.

O_{PS} is the set of objects introduced by each actor in A_{PS}, including:
- the software itself;
- the quantity and satisfaction of the users;
- the competence and stability of the develop team;
- the evidences for software trustworthiness.

RS is the set of resources allocated by each actor in A_{PS} for each object in O_{PS}, including:
- knowledge of trustworthiness;
- time constraint, the available time was short after new evidence was arrived;
- human resource constraint, less management was expected.

A Problem Formulation (PF) is a formal presentation of the concerns expressed in the above PS. Different PF can be defined for different actor in PS. Our PF for the evaluator can be defined as a triple $\Gamma =< A_\Gamma, V_\Gamma, \Pi_\Gamma >$ where:

A_Γ is the set of software which will be evaluated;

V_Γ is the set of the dimensions of trustworthiness (According to the definition in Trustie project [11], the dimensions are availability, reliability, security, real time, maintainability, and survivability);

Π_Γ is a problem statement defining what form of the final result is expected. It is an assignment for software evaluated to some predefined trust levels in our case, in other words, it is $P.\beta$ in terms of Joy's problem classification.

The set of trustworthiness dimensions may differ from one domain to another. Dimensions we used here coincide with the dimensions listed in Trustie. However, the evaluation methodology does not depend on a specific division of trustworthiness.

2.2 Evaluation Model

The evaluation model we build uses ELECTRE TRI method to evaluate trustworthiness, consisting of a 7-tuple $< A^*, D, M, E, G, U, R >$ where:

- A^* are the alternatives selected from A_Γ, let $A^* = A_\Gamma$;
- D is a set of evaluation attributes transformed from V_Γ, including: availability(d_1), reliability(d_2), security(d_3), real time(d_4), maintainability(d_5), survivability(d_6);
- M is the set of measures associated to the attributes in D. Original measurement data can be obtained from software supplier using the methods

recommended in ISO9126 and can be provided as evidence for trustworthiness. Then users scored the sub-attributes according the user satisfaction on original data and use experience;

- E is the set of scales associated to D. we propose a scale inspired by the Analytic Hierarchy Process (AHP) [12] and associate it to each attribute in D. The scale for single user is given in Table 1.

Table 1. Score the sub-attribute of trustworthiness by single user

Scoring	Explanation
9	Very satisfy the expectation for mission-critical use
7	Basically meet the expectation for mission-critical use
5	Very satisfy the expectation for Routine use
3	Basically meet the expectation for Routine use
1	Satisfy the expectation for Occasional use
0	Not available

If the user hesitates between two adjacent scores of 1, 3, 5, 7, 9, then scores it one of 2, 4, 6, 8.

Let C be the set of users who scored the software, L_u be the level of user u (treated as user's weight, scale = 1, 2, 3), s_i be the score u given to attribute i. Then the multi-user evaluation on attribute i, e_i, can be defined as follow:

$$e_i = \frac{\sum\limits_{u \in C} (L_u \times s_i)}{\sum\limits_{u \in C} L_u} \quad ; \tag{1}$$

- G is the set of criteria. A criterion g_i is an attribute equipped with a preference relationship. Let $g_i(a) = e_i(a)$ where a is the software evaluated, g_i maps the attribute scale to real;

- R is the aggregation technique adopted in the evaluation model. Let $R =$ ELECTRE TRI.

The weight assignment plays a crucial role in the ELECTRE TRI method, as well as in other aggregation algorithms. we introduce a base weight vector and a weight expectation function to solve this problem in a way that combines the opinions from the experts and the users. The values of other ELECTRE TRI parameters are also assigned below.

Definition 1. *Base Weight Vector*
Let $D = \{d_i | i = 1, \ldots, k\}$ be a set of attributes and w_b be a vector function from D to $[0,1]^k$ representing non-informative a priori weight over D before any user scoring the attributes, satisfying:

$$w_\mathrm{b}(d_i) \geq 0 \quad \textit{and} \quad \sum_{d_i \in D} w_\mathrm{b}(d_i) = 1 \quad . \tag{2}$$

Then w_b is called a base weight vector.

Definition 2. *Weight Expectation Function*
Let $D = \{d_i | i = 1, \ldots, k\}$ *be a set of attributes and* w_b *be a base weight vector on* D*. Let* C_b *be the* w_b *priori constant. Let* N_i *be the number of the users who scored the attribute* d_i*. The vector function* w_e *from* D *to* $[0,1]^k$ *representing the posteriori weight expectation over* D *expressed as:*

$$w_\mathrm{e}(d_i) = \frac{w_\mathrm{b}(d_i) \times C_\mathrm{b} + N_i}{C_\mathrm{b} + \sum\limits_{d_j \in D} N_j} \tag{3}$$

is then called the weight expectation function.

It can be shown that w_e satisfies the additivity principle:

$$w_\mathrm{e}(d_i) \geq 0 \quad \text{and} \quad \sum_{d_i \in D} w_\mathrm{e}(d_i) = 1 \quad . \tag{4}$$

N_i reflects how many users concern about the attribute d_i, and it can be used as a factor of attribute weight. In case the number of the users is not big enough initially, the base weight vector can play a role in determining weight values over D. C_b reflects the importance of priori weight assignment.

The profiles setting also coincides with Trustie project standard, namely $b_h (h = 1, \ldots, 5)$. We set the default value of credibility index as proposed by ELECTRE TRI software($\lambda_{cutting} = 0.76$) [13]. Since the evaluation value of an attribute has been normalized, we simplify the other parameters' setting as follows and the settings for all attributes are the same:

Indifference threshold $q = 0.2$, preference threshold $p = 1$, veto threshold $v = 2$.

The values given above are priori. They vary according to the software's domain and other relevant concerns, and can be set by experts before an evaluation application being submitted.

2.3 A Case Study

A CASE tool TEvaluator to support evaluating software was evaluated using the model presented above. Let $w_\mathrm{b}(d_i) = 1/6$ where $i = 1, \ldots, 6$ and $C_\mathrm{b} = 60$. The number of users who scored the attributes and the w_e obtained by applying the formula 3 are shown in Table 2.

The parameters for ELECTRE TRI method are assigned in Table 3. The values of the attributes were obtained by applying the formula 1.

A more intuitive description of the model is shown in Fig.1.

Table 2. Weight vector for the attributes

	Availability	Reliability	Security	Real time	Maintainability	Survivability
w_b	1/6	1/6	1/6	1/6	1/6	1/6
N_i	35	30	15	30	20	10
w_e	0.225	0.2	0.125	0.2	0.15	0.1

Table 3. Parameters for ELECTRE TRI method

Criteria	Attribute	Value	Weight	q	p	v	b_1	b_2	b_3	b_4	b_5
g_1	availability	7.3	0.225	0.2	1	2	1	3	5	7	9
g_2	reliability	6.14	0.2	0.2	1	2	1	3	5	7	9
g_3	security	4.85	0.125	0.2	1	2	1	3	5	7	9
g_4	real time	7.1	0.2	0.2	1	2	1	3	5	7	9
g_5	maintainability	5.11	0.15	0.2	1	2	1	3	5	7	9
g_6	survivability	3.93	0.1	0.2	1	2	1	3	5	7	9

$\lambda_{\text{cutting}} = 0.76$

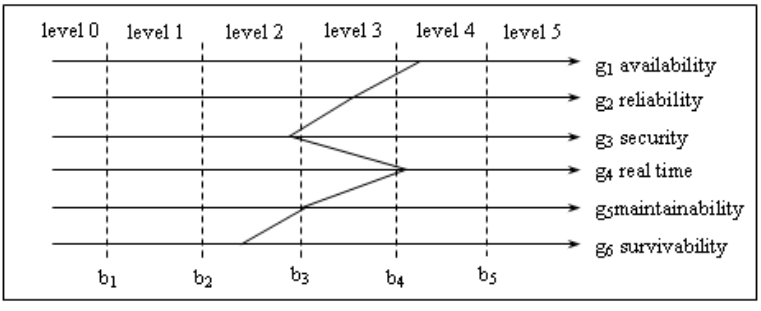

Fig. 1. Description of categories and profiles of the evaluation model

Applying the pessimistic rule, we compared TEvaluator, t for short, with the profiles. Let S be outranking (at least as good as) relationship. It is shown in Fig.1 that $t\ S\ b_5$ does not hold.

Next, we compared t with b_4, since $g_3(b_4) > g_3(t) + v$, then veto condition for $t\ S\ h_4$ is satisfied.

The outranking relationship between t and b_3 is inferred as follows:

According to the pseudo-criterion model defined by ELECTRE method:

$$a'\ S_i\ a \quad \text{iff } g_i(a') \geq g_i(a) - q_i\ , \text{ and } \quad a\ P_i\ a' \quad \text{iff } g_i(a) + p_i\ , \tag{5}$$

Where S_i is the outranking relationship, P_i is the strict preference relationship, q_i is the indifferent threshold, and p_i is the strict preference threshold restricted to the criterion g_i.

$$\Longrightarrow t\ S_1\ b_3,\ t\ S_2\ b_3,\ t\ S_3\ b_3,\ t\ S_4\ b_3,\ t\ S_5\ b_3,\ b_3\ P_6\ t\ , \tag{6}$$

$$\Longrightarrow \text{the concordance index } c(t, b_3) = w_{\text{e}}(d_1) + w_{\text{e}}(d_2) + w_{\text{e}}(d_3) + w_{\text{e}}(d_4) + w_{\text{e}}(d_5) = 0.9\ , \tag{7}$$

$$\text{the discordance index } d_i = \begin{cases} 0\ , \ i = 1,2,3,4,5 \\ \frac{g_6(b_3)-g_6(t)-p_6}{v_6-p_6} = 0.07 < c(t,b_3), i = 6\ , \end{cases} \tag{8}$$

$$\Longrightarrow \text{The credibility index } \rho(t\ S\ b_3) = c(t,b_3) = 0.9 > \lambda_{\text{cutting}} = 0.76\ , \tag{9}$$

$$\Longrightarrow t\ S\ b_3\ . \tag{10}$$

Then we assigned the software to level 3.

2.4 Discussion

To judge the trustworthiness of the software is a complex decision-making process. ELETRE methods provide a better way to solve such real-world problems. By adjusting the model parameters, the model can be adapted to different applications and domains. On the other hand, to set too many parameters for each trustworthiness dimension and aggregation technique also becomes a challenge for the model application. Expertise and skill will be needed to utilize the adaptability of the model. Our settings are based on prior experience and users' feedbacks. The prior values can be set at software category level and be taken as default settings for the software submitted to those categories. Using assignment examples and disaggregation approaches to infer parameters for ELECTRE TRI are also the practical methods [14][15].

Furthermore, since our evaluation model bases the result on a large number of users' feedbacks, in order to make the result more reliable, the evidences that users used to score the software can be taken into account. The considerations are as follows:

1. users are apt to rush to judge the software's trustworthiness regardless of whether the evidences are enough;
2. to assign software to higher level of trust should depend on more reliable cognitive evidences;
3. more requirements for evidence of software development process may guide to improve the software trustworthiness at development phase.

Our work aims at rating the software resources according their trustworthiness to meet the needs of potential users who want to choose more trustworthy software. However, different users have different trustworthiness models with different dimensions of trustworthiness, different weight assignment for the dimensions, and different aggregation methods. They always like to construct a specific model for their application and domain. This is not to say that the results produced by applying our evaluation model are useless, but to argue that they should be used as guidelines and as the basis for selecting trustworthy software. Hence we keep the dimensions at a coarse granularity (trustworthiness is only decomposed into one level) and without further sub-division. The evidences for two or more levels hierarchical model are still important for some users to get deeper understand of the software trustworthiness. But multi-level model may confuse others, and even make them refused to comment on software. It will also bring more arguments for sub-attributes weight assignment.

3 Conclusion

In this paper, we presented a method based on the MCDA approach to evaluate software trustworthiness. The method encompassed key activities performed during the STEP: define problem situation and formulation; design evaluation model based on ELECTRE TRI method; and apply the model. we especially presented a metric and a weight-inferring method for all trustworthiness dimensions. A case study was briefly reported for illustrating the model application. The process can produce evaluation result automatically after new feedback from users comes. The model parameters adjustment for different domains and the more evidence requirements for justifiable trust in software should be studied in our further work.

Acknowledgments. This work was supported by National High Technology Research and Development Program of China (863 Program) (No.2007AA 010301) and Key Program of National Natural Science Foundation of China (No.90818024).

References

1. Schneider, F.B.: Trust in cyberspace. National Academic Press, New York (1999)
2. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. on Dependable and Secure Computing 1(1), 11–33 (2004)
3. National Software Strategy Steering Group.: Software 2015: A National Software Strategy to Ensure U.S. Security and Competitiveness. Center for National Software Studies (2005)
4. Roy, B.: Multicriteria Methodology for Decision Aiding. Kluwer Academic, London (1996)
5. Mousseau, R., Slowinski, R., Zielniewicz, P.: A User-oriented Implementation of the ELECTRE TRI Method Integrating Preference Elicitation Support. Computers and Operations Research 27(7-8), 757–777 (2000)
6. Figueira, J., Greco, S., Ehrgott, M.: Multiple Criteria Decision Analysis: State of the art surveys. Springer, New York (2004)
7. Tian, J.: Quality-Evaluation Models and Measurements. IEEE Software 21(3), 84–91 (2004)
8. Sen, C.G., Baracli, H.: A Brief Literature Review of Enterprise Software Evaluation and Selection Methodologies: A Comparison in the Context of Decision-Making Methods. In: Proceedings of 5th International Symposium on Intelligent Manufacturing Systems, SaKarya, pp. 874–883 (2006)
9. Morisio, M., Tsoukias, A.: IusWare, A Methodology for the Evaluation and Selection of Software Products. IEEE Proceedings on Software Engineering 144, 162–174 (1997)
10. Stamelos, I., Tsoukias, A.: Software Evaluation Problem Situations. European Journal of Operations Research 145(2), 273–286 (2003)
11. Trustworthy software tools and integration environment, `http://www.trustie.org`
12. Saaty, T.L.: Decision Making with the Analytic Hierarchy Process. Int. J. Services Sciences 1(1), 83–98 (2008)

13. Mousseau, V., Slowinski, R., Zielniewicz, P.: ELECTRE TRI 2.0a Methodological Guide and user's Manual. In: Document du LAMSADE, vol. 111, Universite Paris-Dauphine (1999)
14. Mousseau, V., Figueira, J., Naux, J.P.: Using Assignment Examples to Infer Weights for ELECTRE TRI Method: Some Experimental Results. European Journal of Operational Research 130(2), 263–275 (2001)
15. Dias, L., Mousseau, V.: Inferring Electre's Veto-related Parameters from outranking Examples. European Journal of Operational Research 170(1), 172–191 (2006)