Spectrum Analysis for Quality Requirements by Using a Term-Characteristics Map

Haruhiko Kaiya[1,2], Masaaki Tanigawa[1], Shunichi Suzuki[1],
Tomonori Sato[1], and Kenji Kaijiri[1]

[1] Dept. of Computer Science, Shinshu University, Nagano 380-8553, Japan
kaiya@cs.shinshu-u.ac.jp
[2] GRACE Center, National Institue of Informatics (NII), Tokyo 101-8430, Japan
http://grace-center.jp/

Abstract. Quality requirements are scattered over a requirements specification, thus it is hard to measure and trace such quality requirements to validate the specification against stakeholders' needs. We have already proposed a technique called "spectrum analysis for quality requirements" which enables analysts to sort a requirements specification to measure and track quality requirements in the specification. However current spectrum analysis largely depends on expertise of each analyst, thus it takes a lot of efforts to perform the analysis and is hard to reuse experiences for such analysis. We introduce domain knowledge called term-characteristic map (TCM) to improve current spectrum analysis for quality requirements. Through several experiments, we evaluated the improved spectrum analysis.

Keywords: Requirements Analysis, Quality Requirements, Non-functional Requirements.

1 Introduction

Software quality requirements of a system are specifications for defining how well functions of the system are accomplished. Defining quality requirements has more problems than defining functional ones, and there was a special issue about quality requirements in IEEE Software. In its guest editors' introduction [3], the following three problems are mentioned: implicit understanding of quality requirements by stakeholders, trade-offs among quality requirements and difficulty of measuring and tracking quality requirements.

There are several techniques for resolving one or more problems above, and we proposed a simple and general technique called "spectrum analysis for quality requirements" [15] for measuring and tracking quality requirements. A wave such as sound or light can be decomposed into several regular (or sine) waves each of which has different cycle (or wavelength) and power (or amplitude). Spectrum analysis in optics is based on this fact. In spectrum analysis for quality requirements, a quality characteristic such as suitability, accuracy, and interoperability is regarded as wavelength, and the power of the characteristic as its importance as shown in Figure 1. By using a quality requirements spectrum of a system, stakeholders can identify relative attention to quality requirements in a software engineering artifact such as a requirements specification

P. van Eck, J. Gordijn, and R. Wieringa (Eds.): CAiSE 2009, LNCS 5565, pp. 546–560, 2009.

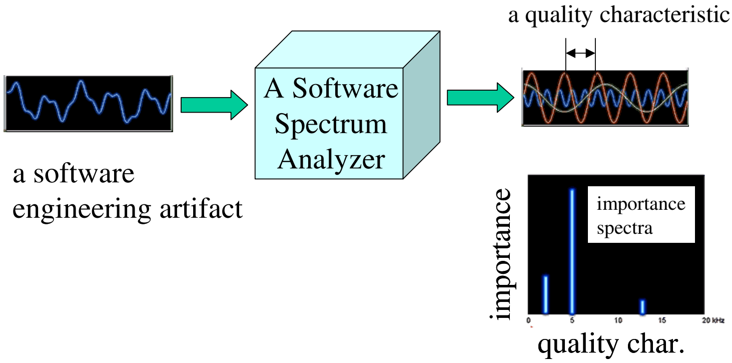


Fig. 1. Basic idea of spectrum analysis for quality requirements

or a design document. Such relative attention enables stakeholders to validate quality requirements defined in such software engineering artifact. Suppose a power of security is larger than one of usability in a quality spectrum of a requirements document for a system. If a stakeholder regards usability is more important than security, he can easily suspect one of his quality requirements could not be reflected in the document.

There are two systematic comparative analyses for quality spectrum analysis. One is comparison among spectra of similar systems to identify mandatory and optional quality characteristics. There are a lot of similar systems for each application domain, e.g., a lot of web browsers, painting tools, e-learning systems and so on. Systems in the same domain usually have similar quality spectrum, and such similarity shows mandatory quality requirements in such a domain [15]. On the other hand, differences among spectra of the systems in the same domain show optional or specific features of each system. Although different segments or different price ranges of the same domain do not always have similar spectrum, we may regard each segment or each range as a sub-domain and may compare spectra of parts in a segment or spectra systems in a range with each other. Another is comparison among spectra of a system in different development phases, e.g., requirements, design, implementation and so on. Quality requirements should be inherited along the progress of development, but it is not easy to track such inheritance during the progress of such development. Quality requirements spectrum enables developers to track such inheritance.

However, there is a serious problem in the current quality spectrum analysis [15]. As shown in Figure 2, the power of each quality characteristic is calculated based on the number of relationships between an element of an artifact, e.g., a sentence in a requirements document, and each quality characteristics. Making such relationships largely depends on the expertise and subjective decision of an analyst. Therefore, it takes a lot of efforts to perform quality spectrum analysis and is hard to perform the analysis (semi-) automatically. In this paper, we will introduce an improved version of quality spectrum analysis for resolving this problem. In addition, we show a prototype of a CASE tool that supports quality spectrum analysis.

The rest of this paper is organized as follows. In the next section, we briefly explain the original quality spectrum analysis method (called “the old method” in this paper) [15], and clarify its problem. We then introduce the improved method (called "TCM method" in this paper) by using domain knowledge called term-characteristic map (TCM). In section 3, we evaluate TCM method with respect to the following three points: results of TCM method inherit those of the old method, domain specific spectrum can be found by using TCM method and results of TCM method are objective, i.e., results of TCM method do not depend on analyst's subjective decision. In section 4, we will show a supporting tool to perform TCM method. Finally, we briefly review related works, summarize our current results and show the future issues.

2 A Method for Generating Quality Spectrum

2.1 The Old Method and Its Problems

As mentioned in introduction, we call the procedure to generate quality spectrum in our previous paper [15] as "the old method" in this paper. Figure 2 shows a typical application of the old method. Inputs of the old method are a list of requirements and a list (catalog) of quality characteristics. In Figure 2, five requirements are listed in the list and quality factors in ISO9126 [14] are used for the catalog. ISO9129 contains one of the famous catalogs of quality characteristics. Such kind of catalogs helps requirements analysts to find missing quality requirements. A quality model in ISO9126 categorizes software quality attributes into six characteristics (functionality, reliability, usability, efficiency, maintainability and portability), which are further subdivided into subcharacteristics such as resource efficiency, changeability and so on in Figure 2.

An analyst then makes relationships between a requirement and a characteristic subjectively. To make such relationships, the analyst takes into account whether a quality characteristic is mentioned in a requirement. In this figure, four characteristics, resource efficiency, changeability, interoperability and security are related to three, one, one and one requirement(s) respectively. Finally, the number of requirements related to each characteristic is counted respectively, and the numbers are normalized into 0 to 1 based on the total number of requirements. The normalized result is visualized as a bar chart at the bottom right in Figure 2, and this kind of vector value is called "quality spectrum" in our previous paper [15]. As mentioned in the first section, quality spectrum tells us which quality characteristics are more important than the others.

One of the serious problems of the old method is the step to make relationships between requirements and characteristics. The old method largely depends on the expertise of an analyst performing the method. As a result, it takes a lot of hours to perform the old method even if the analyst has enough expertise such as domain knowledge of both the application and the quality characteristics. In addition, it is hard to reuse experiences performing the old method.

2.2 TCM Method

To overcome the problem in the old method, we introduce a term-characteristic map (TCM) as domain knowledge for making relationships between documents and characteristics. Figure 3 shows an extended example of Figure 2. TCM is a simple mapping

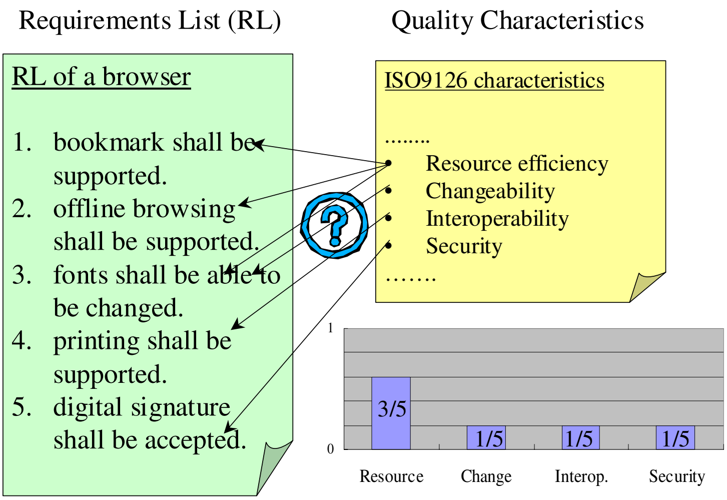
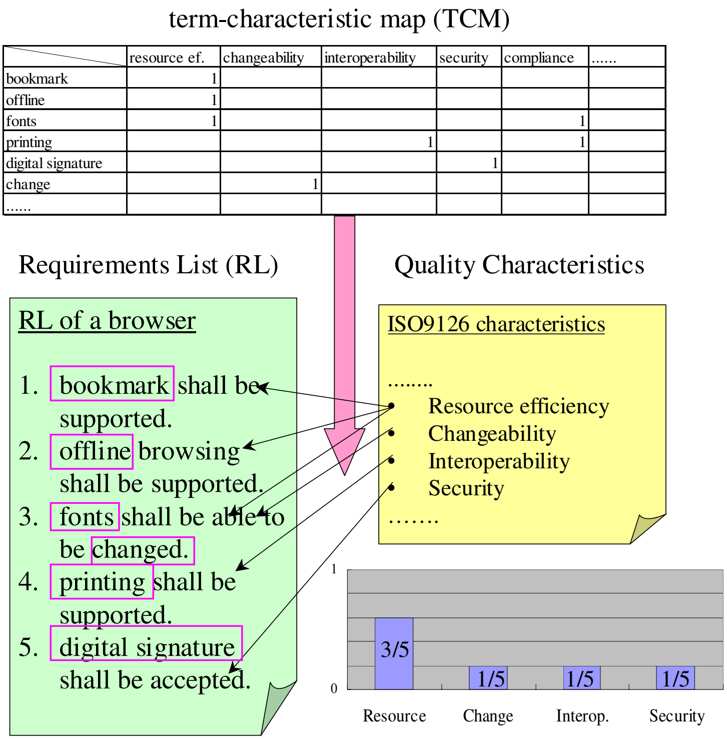


Fig. 2. An Example of the Old Method


term-characteristic map (TCM)

	resource ef.	changeability	interoperability	security	compliance	......
bookmark	1					
offline	1					
fonts	1				1	
printing			1		1	
digital signature				1		
change		1				
......						

Fig. 3. An Example of TCM Method

between terms and characteristic, that tells potential relationships between them. In Figure 3, terms in "RL of a browser" can be looked up in TCM, and an analyst can easily make relationships between requirements and characteristics. TCM merely tells potential relationships. In addition, the possibility whether a term is related to a characteristic depends on the contexts of the term usage. Therefore, relationships between requirements and characteristics cannot be made automatically and the analyst should make some subjective choice. In this example, the analyst does not use some mappings between several terms and a characteristic "compliance". Currently, we simply fill 1 or 0 value (blank if the value is 0 in Figure 3) in cells of TCM to show whether there is potential relationship or not, but we would like to introduce some ordinal or ratio values to show the degree of its potential.

Although some domain expert or an analyst himself should develop TCM beforehand, TCM will be able to be reused and be improved among similar systems in the same application domain. We would like to confirm this point in the future.

3 Evaluation

We evaluate TCM method mentioned in the last section with respect to the following three points.

- Results of TCM method inherit those of the old method.
- Domain specific spectrum can be found by using TCM method, i.e., spectra of several different systems in the same domain are similar with each other.
- Results of TCM method are objective, i.e., different analysts can generate similar spectra of a system.

3.1 Data Gathering and Evaluation Method

To evaluate TCM method with respect to three points above, we need the following kinds of spectra: a spectrum generated by using the old method, a spectrum generated by using TCM method, a spectrum generated by another analyst using TCM method with the same TCM and a spectrum by the analyst with his own TCM. Figure 4 shows the outline how to gather such spectra data for our evaluation. This figure is written in data flow diagram, where boxes and notes correspond to data and ovals correspond to processes. We had two subjects called subject A and B, and we asked them perform spectrum analysis to documents of three browsers, Internet Explorer (IE), Fire Fox (FF) and Opera (OP), respectively. Subject A well knew this application domain as a user, and subject B was an average user. At the left side of Figure 4, subject A generated a spectrum without TCM. Note that this spectrum was generated before TCM method was proposed. Subjects A and B developed TCM of web browsers' domain respectively by using documents of web browsers. Subjects A and B then performed TCM method respectively by using the same TCM developed by subject A as shown in the middle of Figure 4. Subject B also developed another spectrum by using his own TCM as shown in the right side of Figure 4. Subject A developed spectra of another types of systems mentioned in the next sub section. Both subjects used general spreadsheet to

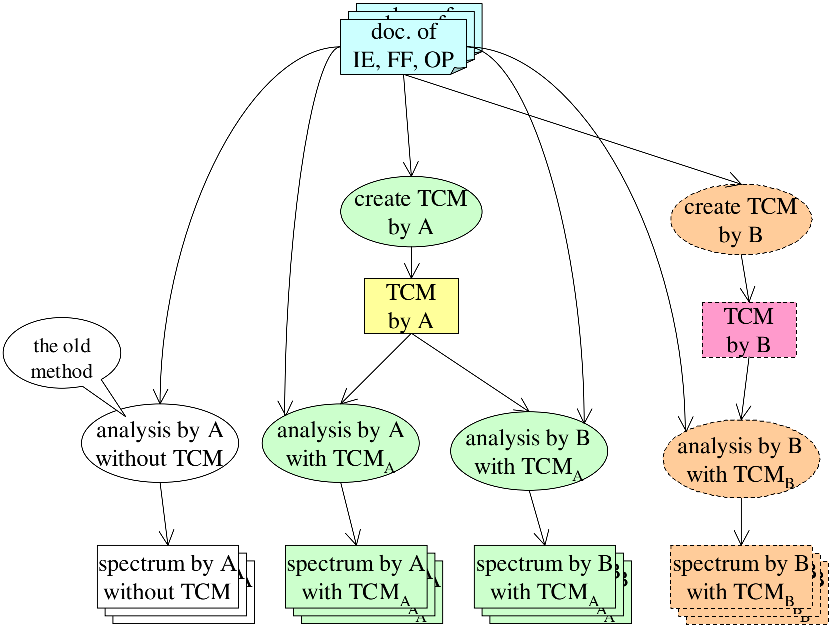


Fig. 4. Data Gathering

perform old or TCM method. Especially, subjects performing TCM method did not use a supporting tool mentioned in the next section because we decided to develop the tool based on the results of this evaluation.

Because a quality spectrum is a kind of vector, we use cosine similarity (cossim) to decide whether two spectra are similar with each other. The definition of cosine similarity between $\boldsymbol{a}$ and $\boldsymbol{b}$ is as follows.

$$cossim(\boldsymbol{a}, \boldsymbol{b}) = \frac{a_1 * b_1 + a_2 * b_2 + \cdots + a_n * b_n}{\sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} * \sqrt{b_1^2 + b_2^2 + \cdots + b_n^2}}$$

When two vectors are completely the same, the value is one. Because quality spectrum never has negative value in its vector, cosine similarity between two quality spectra varies from 0 to 1. Therefore, we may regard two quality spectra are similar if their cosine similarity is close to 1. For example, $cossim(\boldsymbol{a}, \boldsymbol{b})$ is 0.99 when $\boldsymbol{a}$ is (0.00 0.08, 0.15, 0.25, 0.00, 0.00, 0.01, 0.15, 0.20, 0.85, 0.04, 0.09, 0.01, 0.25, 0.00, 0.00, 0.01, 0.00, 0.01, 0.01) and $\boldsymbol{b}$ is (0.00, 0.05, 0.18, 0.21, 0.00, 0.00, 0.01, 0.18, 0.12, 0.86, 0.02, 0.08, 0.01, 0.18, 0.00, 0.00, 0.03, 0.00, 0.00, 0.02). Note that $\boldsymbol{a}$ corresponds to a spectrum by A with TCM A in Figure 5, and $\boldsymbol{b}$ corresponds to a spectrum by B with TCM A in the same figure.

3.2 Inheritance from the old method

Because TCM method is one of the improved version of the old method in our previous paper [15], a quality spectrum generated by the method should be similar to one by the

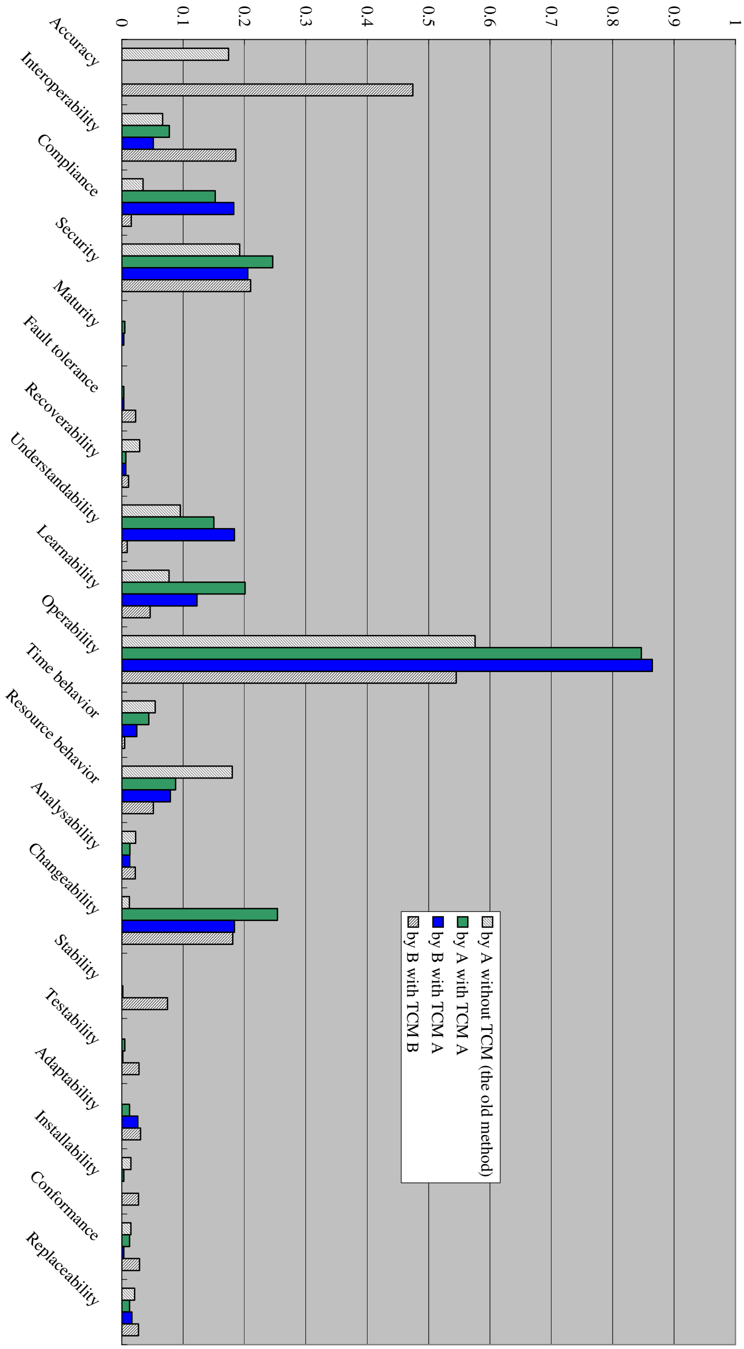


Fig. 5. Four Spectra from the old method, TCM method by subject A and B

old method. Figure 5 shows four quality spectra corresponding to the outputs in Figure 4. Note that each spectrum in Figure 5 is the average of spectra of three web browsers; IE, FF and OP. Horizontal axis of this figure shows quality characteristics used in our quality spectrum analysis. Because we have no explicit users of web browsers, we cannot identify objectives of such users. Therefore, we do not use a quality characteristic "suitability" in ISO9126 during this evaluation. Vertical axis shows the values of spectrum for each quality characteristic. Because documents of web browsers are analyzed and browsers are highly interactive system, "operability" has the highest value in a spectrum. "Security" has also higher value because of a lot of threats over the Internet. According to the definition of cosine similarity, similarity value between first and second spectra is 0.91, and the value between first and third spectra is 0.92. Therefore, we may regard TCM method inherits analytic ability from the old method.

3.3 Different Systems in The Same Domain

Quality spectrum is used to identify mandatory and optional quality requirements in an application domain, and this usage is based on the fact that different systems in the same domain have similar quality spectrum [15]. We confirm this fact by using TCM method.

Figure 6 shows three spectra for each browser by subject A ("spectrum by A with $\mathrm{TCM_A}$" in Figure 4). Figure 7 shows three spectra by subject B ("spectrum by B with $\mathrm{TCM_A}$" in Figure 4). As mentioned in last subsection, A and B used the same TCM developed by subject A. As shown in these figures, the spectra for each subject are

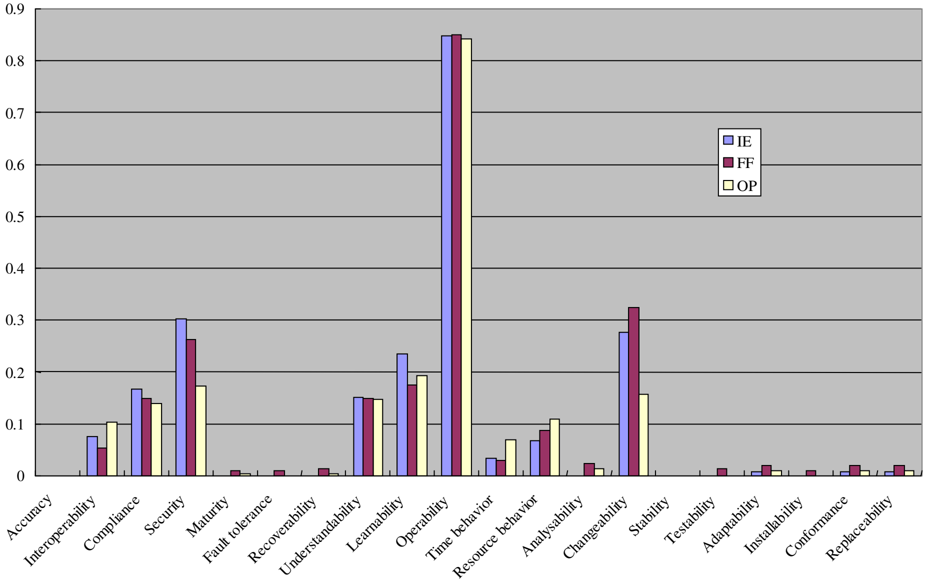


Fig. 6. Three Quality Spectra for each browser by Subject A

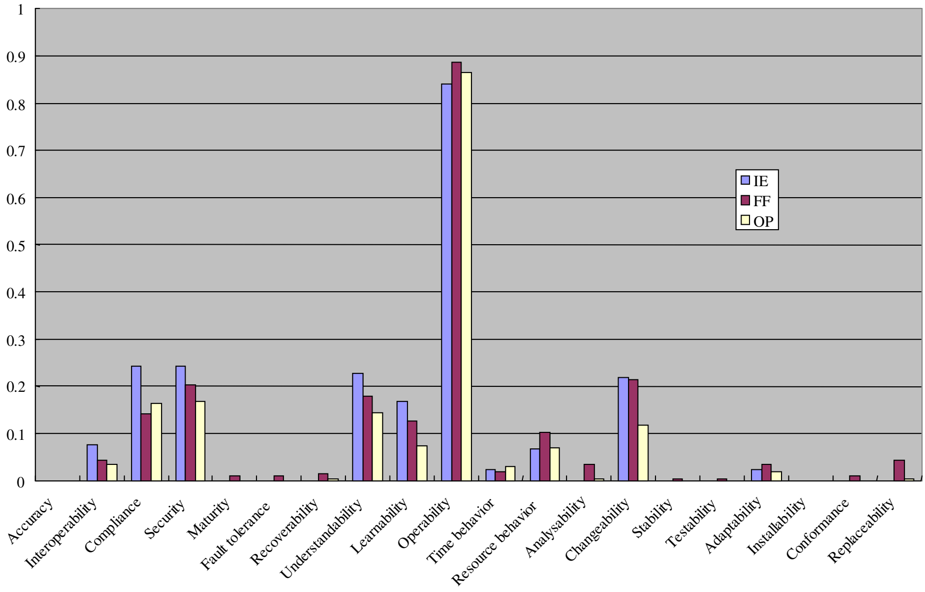


Fig. 7. Three Quality Spectra for each browser by Subject B

similar. In the case of subject A in Figure 6, cosine similarity between two out of three browsers are 0.99, 0.98 and 0.98. In the case of subject B in Figure 7, cosine similarity between two out of three browsers are 0.98, 0.97 and 0.99.

We also have quality spectra of software systems other than browsers, and we show bar charts of both browsers and other types of systems in Figure 8. A system labeled "NEWS" is a kind of a proxy system for feeding news articles to a specific intranet, and another system labeled "DB" is a kind of a document management system. Documents for both systems were published by our government [18]. Both systems are neither browsers nor interactive systems. Therefore, the spectra of NEWS and DB should be different from spectra of browsers. As shown in Figure 8, spectra of NEWS and DB are clearly different from spectra of browsers. Cosine similarity between NEWS or DB and each browser is almost 0.5. Therefore, we may regard different systems in the same domain have similar quality spectrum generated by TCM method. In addition, different types of systems have different quality spectrum.

As shown in figures 6 and 7, the power (amplitude) of some characteristics is different within three browsers. For example, powers of "interoperability" and "changeability" are different with each other. Because these three browsers are different with respect to its license (open source software or not), its platform (multi-platform including mobile devices or not) and so on, quality characteristics such as "interoperability" and "changeability" will be differently focused. On the other hand, powers of "operability" are similar because this characteristic is important for interactive systems such as browsers in general.

3.4 Different Analysts

Because one of the expected advantages of TCM method is that the result is more objective than the old method. To confirm this advantage, we compare a spectrum by subject A with TCM_A and another by B with TCM_A in Figure 5. Because cosine similarity between these two spectra is 0.99, we may regard results by using TCM method is almost the same. On the other hands, cosine similarity between a spectrum by B with TCM_A and another spectrum by B with TCM_B is 0.75. We may also regard these two spectra are slightly different with each other. As a result, sharing TCM seems to help analysts to analyze requirements documents objectively. As mentioned in 3.2, subjects performing TCM method only use general spreadsheet. Therefore, they have to achieve tedious tasks that can be performed automatically because no supporting tools existed.

4 A Supporting Tool for TCM Method

Through evaluation in the last section, we can confirm TCM method seems to work well. To improve the efficiency of the task using TCM method, we are developing a supporting tool as shown in Figure 9. In an example in this figure, NEWS system mentioned in Figure 8 is analyzed. As stated in subsection 2.2, TCM method includes both subjective and automatic tasks. Therefore, its supporting tool should be interactive one.

Before performing quality spectrum analysis, someone especially domain expert has to perform the following task.

1. To create TCM of a domain.

An analyst then analyzes requirements with the help of following automatic tasks.

2. To look up terms appearing in each requirement in TCM, and to look up characteristics related to each term.
3. To generate quality spectrum by counting the number of requirements related to each quality characteristic.

The analyst finally performs the following tasks.

4. To choose quality characteristics actually related to each requirement based on TCM, contexts of each term and his expertise.

The tool in Figure 9 supports its users to perform four tasks above in the following ways.

1. The domain analyst can manually generate TCM by using “Term Characteristic Map (TCM)” tab in this figure (the contents of the tab are not shown in this figure). He first enumerates terms usually appearing in an application domain, and fill the checkboxes corresponding to quality characteristics for each term. By using this tool, candidates of terms can be automatically extracted and enumerated in “Term Characteristic Map (TCM)” tab from text documents. Therefore, the analyst can pick terms out from such candidates.

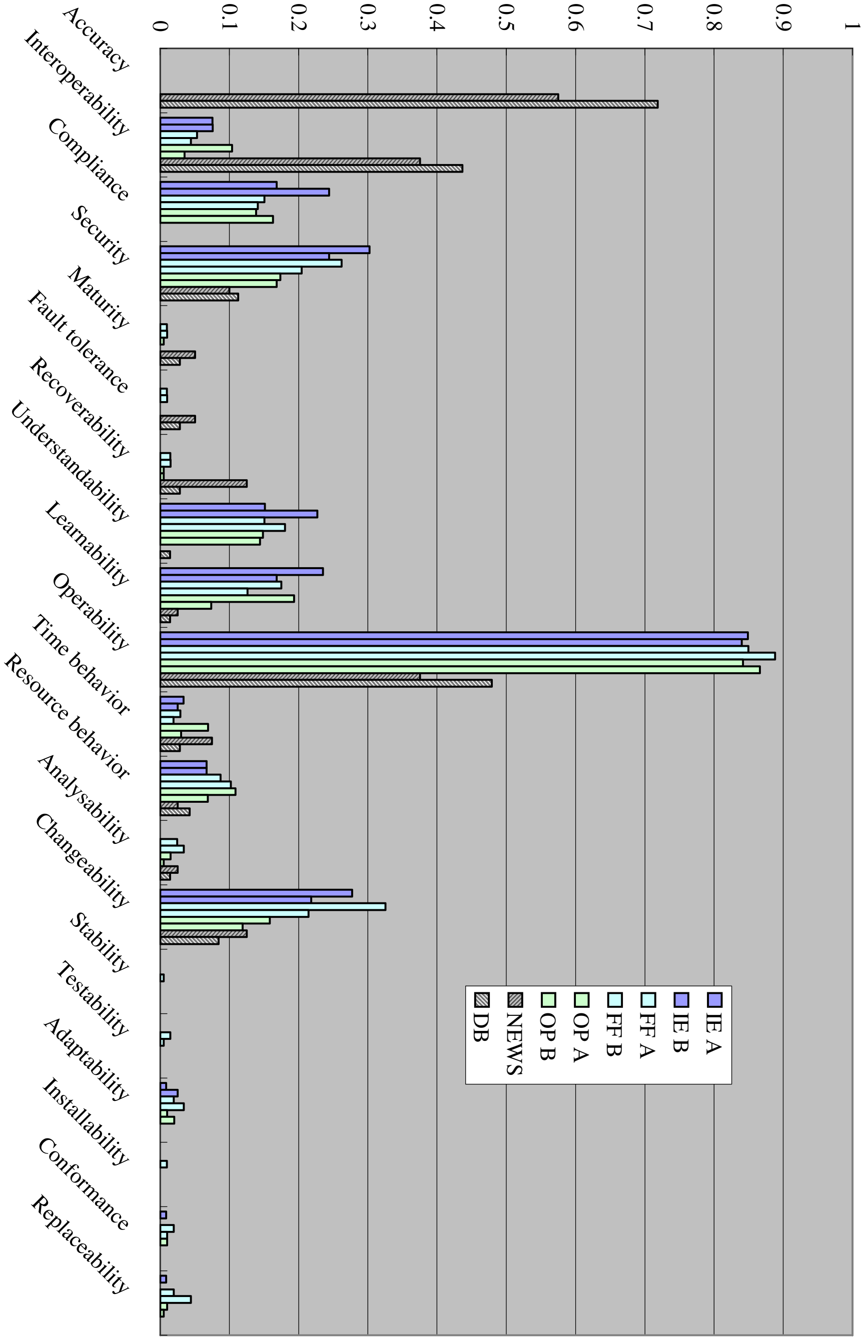
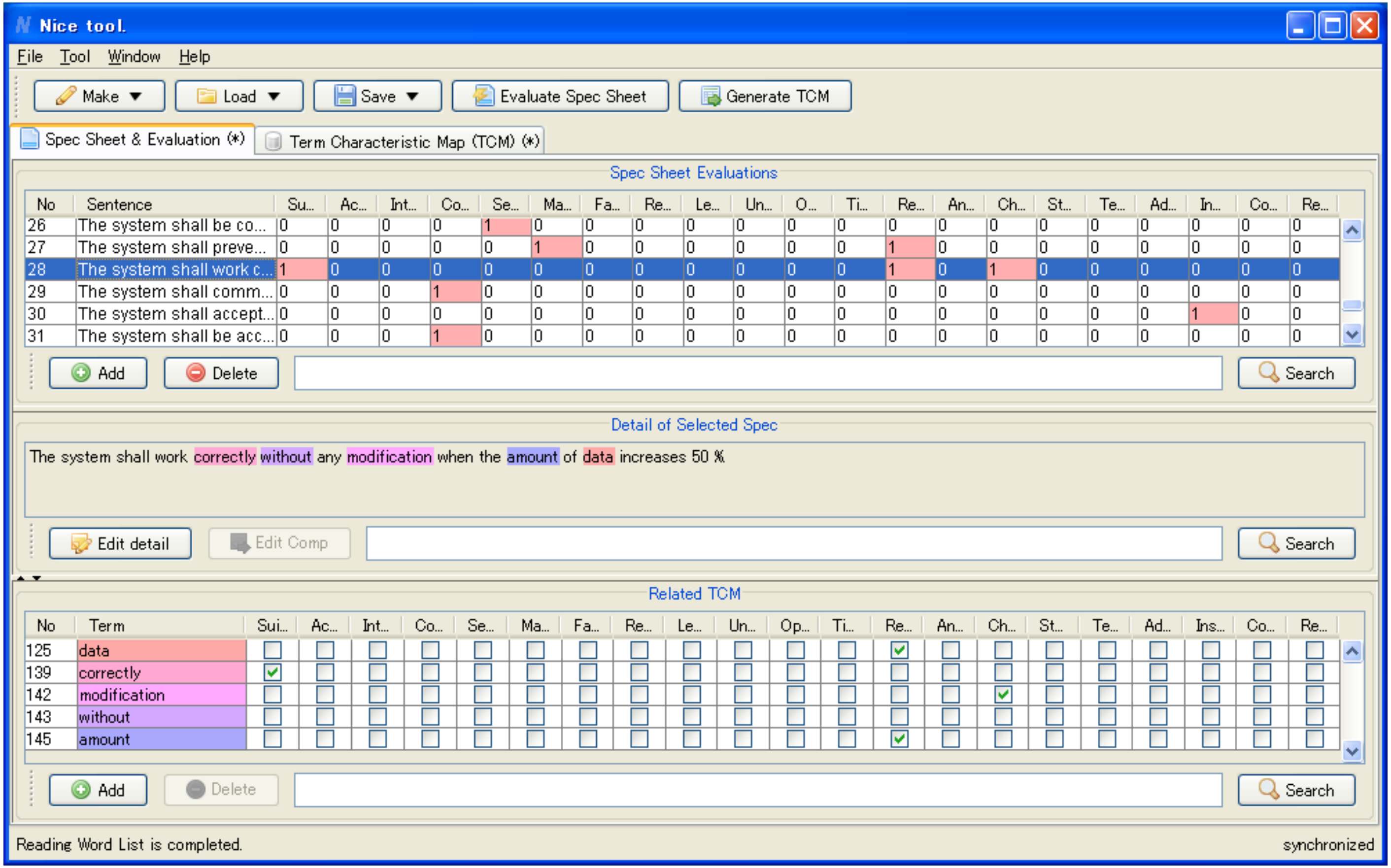


Fig. 8. Quality Spectra of browsers and other types of software



Spec Sheet Evaluations

No	Sentence	Su...	Ac...	Int...	Co...	Se...	Ma...	Fa...	Re...	Le...	Un...	O...	Ti...	Re...	An...	Ch...	St...	Te...	Ad...	In...	Co...	Re...
26	The system shall be co...	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	The system shall preve...	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
28	The system shall work c...	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
29	The system shall comm...	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	The system shall accept...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
31	The system shall be acc...	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Detail of Selected Spec

The system shall work correctly without any modification when the amount of data increases 50 %

Related TCM

No	Term	Sui...	Ac...	Int...	Co...	Se...	Ma...	Fa...	Re...	Le...	Un...	Op...	Ti...	Re...	An...	Ch...	St...	Te...	Ad...	Ins...	Co...	Re...
125	data	□	□	□	□	□	□	□	□	□	□	□	□	☑	□	□	□	□	□	□	□	□
139	correctly	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
142	modification	□	□	□	□	□	□	□	□	□	□	□	□	□	□	☑	□	□	□	□	□	□
143	without	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
145	amount	□	□	□	□	□	□	□	□	□	□	□	□	☑	□	□	□	□	□	□	□	□

Reading Word List is completed.

synchronized

Fig. 9. A Snapshot of A Supporting Tool

2. As shown in the center area “Detail of Selected Spec” of the figure, terms are automatically identified according to the terms in pre-loaded TCM. A subset of the TCM is then shown at the bottom table “Related TCM” in the figure. In this example, five terms “correctly”, “without”, “modification”, “amount” and “data” are looked up.
3. According to the checks on the table “Related TCM” in the figure, related quality characteristics are automatically accumulated at the top table of this figure “Spec Sheet Evaluations”. Based on the checks of characteristics for each requirement (wrote “sentence” in this tool), the tool will visualize or output quality spectrum.
4. An analyst may freely change the value of checkboxes on the table “Related TCM” in this figure. Because this table is a copy of original TCM (generated in the first task), the original TCM gets no effects according to the changes.

5 Related Works

There are several studies how to define each quality requirement. In ISO25021 [13], concrete examples how to measure quality requirements are shown, and these examples help analysts to make quality requirements measurable. Donald Firesmith gives some format to specify quality requirements rigorously [10]. In ATAM (Architecture Tradeoff Analysis Method) [16,2], a template for quality requirements called “quality attribute scenario” is provided to support stakeholders writing quality requirements.

Studies mentioned above focus on the *micro-view* of quality requirements because they focus on each requirement. Quality spectrum analysis [15] and its extension in this paper rather focus on the *macro-view* because it focuses on distribution of quality requirements in an artifact such as a requirements specification. We think both views are important to improve the quality requirements analysis, but there are few studies with macro-view. Studies (e.g., [4]) about the quality of requirements documents, e.g., completeness, correctness, and so on mentioned in IEEE 830 [1], also focus on the macro-view, but this kind of studies is not directly related with studies of quality requirements.

In an article by Ozkayad et al. [19], an empirical data of the most common quality attributes was shown based on the ATAM. The idea to analyze this kind of data is similar to quality spectrum analysis [15], but comparative analysis mentioned in introduction is not proposed in the article [19]. In DDP (defect detection prevention) [7,9], the relationships among requirements, risks and their mitigations are visualized. Although this visualization shows a macro-view of quality requirements, trade-offs between risks and their mitigation costs are mainly focused.

Basic idea about relationships between requirements and quality characteristics seems to be imported from QFD (Quality Function Deployment) [12]. TCM as domain knowledge is imported from a probabilistic model among terms, documents and queries in a paper by Cleland-Huang et al. [6].

6 Conclusion

In this paper, we improved quality requirements spectrum analysis proposed in our previous paper [15] by introducing term-characteristic map (TCM) as domain knowledge.

Quality requirements spectrum analysis is a technique for measuring and tracking quality requirements over a requirements document written in natural language. TCM helps analysts to derive amplitude of each characteristic in a quality spectrum because TCM plays a role of domain knowledge for finding quality characteristics related to each requirements statement. Through several experiments, we evaluate quality requirements analysis method with TCM, and confirmed TCM method inherited some features of the old method in our previous paper [15] and results by TCM method became more objective.

Even if TCM for a domain can be reused, it is still hard to develop and maintain TCM for each domain. A technique called LSA (Latent Semantic Analysis) [8] seems to be used for developing and maintaining TCM because terms with the similar meaning can be found automatically by using LSA. In addition, LSA seems to be used to look up quality characteristics in TCM because the semantic similarity of a requirement sentence and a quality characteristic with its description can be calculated based on term occurrences in such sentences and descriptions.

Requirements documents written in natural languages are only analyzed now, but we would like to apply quality spectrum analysis to other types of artifacts. In a paper by Zhang et al. [20], UML notation is extended for representing quality attributes. In a paper by Chowdhury et al. [5], detailed characteristics about security in source codes are identified. These studies can be used to develop methods for quality spectrum analysis for design and source codes.

As mentioned in 2.2, we do not specify the degree of relationships between terms and characteristics. In the same way as shown in a paper by Cleland-Huang [6], frequency of relationships among requirements documents in the same domain can be used for specifying such a degree. We do not also specify the degree of relationships between requirements and characteristics. Frequency of terms in each requirement can be used for specifying such a degree. In addition, types of requirements representation can be used. In an article by Glinz [11], several different types of representations, e.g., qualitative, by example, quantitative and so on, are proposed. By using such types, we can give higher degree to a requirement sentence if a quality requirement is represented not qualitatively but quantitatively, for instance.

Currently, we only focus on coarse-grained quality as characteristics in a system, but we may use another kind of characteristics scattered over the system, e.g., fine-grained quality or an attention for each type of stakeholders. In an article by Ozkayad et al. [19], fine-grained quality characteristics about security can be found, and they can be used for quality spectrum analysis. In a book by Macaulay [17], types of stakeholders are shown such as designer, financial person, maintainer and users, and each stakeholder is interested in different part of requirements. We can perform "stakeholder spectrum analysis" over a document based on such types. That is the reason why we regard spectrum analysis over software artifacts is general.

References

1. IEEE Recommended Practice for Software Requirements Specification (October 1998) IEEE Std 830-1998 (ISBN 0-7381-0332-2 SH94654) (Print)
2. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice, 2nd edn. Addison-Wesley, Reading (2003)

3. David Blaine, J., Cleland-Huang, J.: Software Quality Requirements: How to Balance Competing Priorities. IEEE Software 25(2), 22–24 (2008)
4. Bucchiarone, A., Gnesi, S., Pierini, P.: Quality Analysis of NL Requirements: An Industrial Case Study. In: IEEE International Requirements Engineering Conference (RE 2005), pp. 390–394 (2005)
5. Chowdhury, I., Chan, B., Zulkernine, M.: Security Metrics for Source Code Structures. In: International Workshop on Software Engineering for Secure Systems (SESS 2008), pp. 57–64 (2008)
6. Cleland-Huang, J., Settimi, R., BenKhadra, O., Berezhanskaya, E., Christina, S.: Goal-Centric Traceability for Managing Non-Functional Requirements. In: International Conference on Software Engineering (ICSE) (2005)
7. Cornford, S.L., Feather, M.S., Kelly, J.C., Larson, T.W., Sigal, B., Kiper, J.D.: Design and Development Assessment. In: Proceedings of the Tenth International Workshop on Software Specification and Design (IWSSD 2000), pp. 105–114 (2000)
8. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the Society for Information Science 41(6), 391–407 (1990)
9. Feather, M.S., Cornford, S.L., Hicks, K.A., Kiper, J.D., Menzies, T.: A Broad, Quantitative Model for Making Early Requirements Decisions. IEEE Software 25(2), 49–56 (2008)
10. Firesmith, D.: Quality Requirements Checklist. Journal of Object Technology 4(9), 31–38 (2005)
11. Glinz, M.: A Risk-Based, Value-Oriented Approach to Quality Requirements. IEEE Software 25(2), 35–41 (2008)
12. Herzwurm, G., Schockert, S., Pietsch, W.: QFD for Customer-Focused Requirements Engineering. In: Proceedings of 11th IEEE International Requirements Engineering Conference, pp. 330–338 (September 2003)
13. International Standard ISO/IEC 25021. Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Quality measure elements (October 2007)
14. International Standard ISO/IEC 9126-1. Software engineering - Product quality - Part 1: Quality model (2001)
15. Kaiya, H., Sato, T., Osada, A., Kitazawa, N., Kaijiri, K.: Toward Quality Requirements Analysis based on Domain Specific Quality Spectrum. In: Proc. of the 23rd Annual ACM Symposium on Applied Computing 2008, Fortaleza, Ceara, Brazil, vol. 1(3), pp. 596–601. ACM, New York (2008) (Track on Requirements Engineering)
16. Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., Carriere, J.: The Architecture Tradeoff Analysis Method. In: IEEE International Conference on Engineering of Complex Computer Systems (ICECCS), p. 68 (1998)
17. Macaulay, L.A.: Requirements Engineering. In: Applied Computing. Springer, Heidelberg (1996)
18. Minister of Economy, Trade and Industry, Japan (in Japanese), `http://www.meti.go.jp/feedback/data/i30728aj.html`
19. Ozkayad, I., Bass, L., Sangwan, R.S., Nord, R.L.: Making Practical Use of Quality Attribute Information. IEEE Software 25(2), 25–33 (2008)
20. Zhang, Y., Liu, Y., Zhang, L., Ma, Z., Mei, H.: Modeling and Checking for Non-Functional Attributes in Extended UML Class Diagram. In: Annual IEEE International Computer Software and Applications Conference (COMPSAC 2008), pp. 100–107 (2008)