情報処理学会論文誌 Vol.53 No.2 510–522 (Feb. 2012)

分析履歴を用いたソフトウェア品質要求のスペクトル分析法

海谷 治彦[1,a)] 鈴木 駿一[1] 小川 享[1] 谷川 正明[1] 梅村 真弘[1] 海尻 賢二[1]

受付日 2011年5月27日, 採録日 2011年11月7日

概要：品質要求の記述量に基づき品質要求の重要度を測定する手法として，ソフトウェア品質要求のためのスペクトル分析法がすでに提案されている．しかし，この手法での測定結果が重要度を示すか否かの十分な考察が従来研究では行われていなかった．また，この手法を実施するには分析者の主観的な判断や経験が必要であるという問題点もある．本稿では，既存研究のこれら 2 つの問題点を解決した結果を示す．まず，品質要求の記述量に基づき，その重要度を測定することが妥当か否かを検討した結果を示す．検討の結果，補正を加えることで，品質要求の記述量に基づき，その重要度を測定することは妥当であるという結論に至った．次に，ソフトウェア品質要求のためのスペクトル分析を自動的に実施するための改善手法とツールを提案し，評価した結果を示す．提案手法では過去の分析実績の記録を利用することで，品質要求の定義量の測定を自動化する．これによって，スペクトル分析を行う際の分析者の判断を軽減もしくは除去することが可能となる．分析実績が増加することで，提案手法による機械的な分析結果がより正確になることを実験を通して評価し確認した結果も示す．

キーワード：要求分析，分析履歴，品質要求，スペクトル分析

A Spectrum Analysis Method for Software Quality Requirements Analysis Using History of Analyses

Haruhiko Kaiya[1,a)] Shunichi Suzuki[1] Toru Ogawa[1] Masaaki Tanigawa[1] Masahiro Umemura[1] Kenji Kaijiri[1]

Received: May 27, 2011, Accepted: November 7, 2011

Abstract: A method to analyze a requirements specification for identifying the importance of quality requirements in the specification has been already proposed, and the importance is derived from the amount of quality requirements in the specification. However, whether such amount really reflects the importance was not examined enough in the method. In addition, the method required human decisions a lot. In this paper, we show the solutions for these two problems. We first examine whether the amount of each quality requirement described in a specification reflects its importance. We conclude such amount reflects the importance when we revise the amount based on a priority among quality requirements and the distribution of shared quality requirements. We next propose an improved method by using analyses records of similar systems for automated analysis. We also developed a supporting tool to enact our method. We expect such analyses records help an analyst to make or skip decision during the method, and the result becomes more correct than before. We applied our method to several requirements specifications, and found our method contribute to making the result more correct than before without a lot of human effort.

Keywords: requirements analysis, analyses records, quality requirements, spectrum analysis

1. はじめに

高品質な要求仕様書を記述するためには，correctness，unambiguity，completeness，rank of importance，stabil-

[1] 信州大学
Shinshu University, Nagano 380–8553, Japan
[a)] kaiya@shinshu-u.ac.jp

ity，verifiability そして traceability を考慮する必要がある [1]．機能要求については上記を考慮するための手法が数多く提案されている [2], [3], [4]．しかし，非機能要求や品質要求については十分な提案があるとはいえない．品質要求はある機能がどの程度よく達成されるかを述べたものである [5]．機能要求に比べ品質要求にはいまだ問題点が多くあるため，IEEE Software において特集が組まれた．その前書きとしての記事 [5] において，ステークホルダの非明示的な理解，品質要求間のトレードオフ，品質要求の測定と追跡の難しさの問題点が注目されていた．

上記の最後の問題点「品質要求の測定と追跡の難しさ」に注目し，品質要求のためのスペクトル分析という手法がすでに考案されている [6]．この手法によって，ある要求仕様書に，どのような品質要求がどの程度の重要度で記述されているかを測定することが可能となると述べられている．この手法の入力は要求項目のリスト（文のリスト）であるため，要求仕様書，設計書，コードの説明書等，自然言語で記述されたソフトウェア成果物に容易に適用可能であるとも述べられている．これによって，要求定義段階で定義された品質要求が，設計や実装に正しく継承されているかを分析することが可能となったことも報告されている [7]．しかし，この手法には 2 つの問題点がある．第 1 に，測定された数値が要求仕様書における品質要求の重要度であるか否かが明らかでない．第 2 に，この手法の遂行には分析者の主観的な判断を必要とするため，自動的に分析を行うことは困難である．本稿の目的はこの 2 つの問題点を解決することである．

第 1 の問題点解決のために，品質要求および測定の定義に立ち戻り，要求仕様書における品質要求の重要度とは何か，そして，それを測定するためには，どのような技術が必要かを明らかにした．結果として，要求仕様書全体における，ある品質要求で修飾されている機能要求の割合を，その要求仕様書における，その品質要求の重要度と考えてよいことを示した．ただし，2 点の補正が必要であることも明確にした．第 1 に，要求項目間の重要度の違いを，要求項目間の優先度付け技術を用いて補正する必要があることである．第 2 に，複数の要求項目を修飾する品質要求が，単一の要求項目に記述されている場合，その分配を行うことで補正する必要があることである．

第 2 の問題点解決のために，スペクトル分析の結果を履歴として蓄積し利用することで，この手法を自動的に遂行するための改善手法を提案した．従来の手法 [6] では，要求項目が品質要求に修飾されいるか否かを，要求項目に出現する語句の種類から判定していた．本稿で提案する手法では，この語句と品質要求とが実際に関係があったか否かの実績を蓄積することで，ある語句とある品質要求との関係の確率を求める．そして，履歴を一定以上蓄積することで，ある語句がある品質要求と関係の確率の信頼を改善し，分析者の主観的判断なしでもスペクトル分析が可能となるようにする．履歴の蓄積を繰り返すことで自動的に分析されたスペクトル値が正解に近づくか否かを実験を通して確認することで，この提案が妥当か否かを確認した．

本稿の構成は以下のとおりである．まず，次章では第 1 の問題点を解決するために，品質要求の測定と重要度とは何かについて考察をする．次に，本稿で提案する手法の基盤となるソフトウェア品質要求に関するスペクトル分析手法を概説する．4 章で第 2 の問題点を解決する提案手法の手順を詳説し，5 章でその遂行を支援するツール QRAST を概説する．6 章で提案手法の評価実験とその結果を示す．最後にまとめと今後の課題を述べる．

2. 品質要求の測定と重要度

文献 [5] において品質要求は「どの程度良く機能要求が遂行されるかを述べている」と定義されている．また，別の文献 [8] において「利用に対する適合度」が品質という用語の定義の 1 つとして紹介されており，これは文献 [5] の品質要求の定義とよく一致する．これらより品質要求は機能要求を修飾する要求であると考えてよい．文献 [9] によると，測定（measurement）は測度（measure）を決定する行為であり，測度はソフトウェア製品やプロセスの，ある属性の程度，量，次元，キャパシティ，サイズに対する定量的な指標を与えるものと定義されている．また，このような指標は分析者に対してソフトウェア製品やプロセスに内在する隠れた真実を理解する能力（洞察力 insight）を与えなければならないとも述べられている．

それぞれの機能要求において，どのような種類の遂行の良さ（品質要求）に注目するかは異なる．たとえば，ある機能では使い勝手の良さが注目され，別の機能では障害が発生し難いことが遂行の良さとして注目される．また，複数種類の遂行の良さを注目する機能もある．遂行の良さ（品質要求）の種類を分類するためのカタログ [10], [11] がいくつか存在し，このようなカタログは，どのような種類の品質要求に着目すべきかの指針となる．

あるソフトウェアの要求仕様書において，どの種類の品質要求が重視されているかが測定できれば，我々は測定対象であるソフトウェアを洞察することが可能となる．たとえば，同種類のソフトウェアは，同種類の品質要求を同程度に重視しており，異なる種類のソフトウェアは異なる種類の品質要求が重視していることがすでに報告されている [6]．もし，同種類のソフトウェアの間において，重視している品質要求の種類や度合いが異なるのであれば，どちらかの品質要求定義に誤りがあるか，もしくは，同種類といえども異なるソフトウェアであるため，差別化のために意図的に重視する品質要求の種類や度合いを変えているかのどちらかである．前者の場合，修正すべき誤りを知ることができる．後者の場合，要求定義を行った者の意図が要

表 1 品質要求の重要度，単純な割合，補正した割合のそれぞれの順序（左表は文書 A について，右表は文書 B について）

Table 1 The ordering of quality requirements based on their importance, the ordering based on simple ratio of functional requirements qualified by each quality requirement and the ordering based on revised ratio. Left table is about a document A. Right table is about a document B.

重要度	単純割合	補正割合
時間効率性	運用性	時間効率性
規格適合性	資源効率性	セキュリティ
セキュリティ	規格適合性	規格適合性
相互運用性	セキュリティ	相互運用性
資源効率性	理解性	運用性
変更性	相互運用性	資源効率性
環境適応性	環境適応性	理解性
理解性	標準適合性	環境適応性
成熟性	変更性	標準適合性
標準適合性	試験性	変更性
運用性	時間効率性	試験性
習得性	正確性	正確性
回復性	習得性	習得性
正確性	置換性	置換性
安定性	成熟性	成熟性
置換性	障害許容性	障害許容性
設置性	回復性	回復性
試験性	安定性	安定性
解析性	解析性	解析性
障害許容性	設置性	設置性

重要度	単純割合	補正割合
時間効率性	正確性	セキュリティ
セキュリティ	運用性	時間効率性
正確性	セキュリティ	規格適合性
変更性	資源効率性	相互運用性
規格適合性	変更性	正確性
標準適合性	環境適応性	運用性
回復性	規格適合性	資源効率性
置換性	試験性	変更性
成熟性	相互運用性	環境適応性
設置性	標準適合性	試験性
運用性	安定性	標準適合性
理解性	時間効率性	安定性
習得性	解析性	解析性
環境適応性	理解性	理解性
相互運用性	置換性	置換性
障害許容性	習得性	習得性
資源効率性	成熟性	成熟性
解析性	障害許容性	障害許容性
安定性	回復性	回復性
試験性	設置性	設置性

求仕様書に正しく記述されていることを確認することができる．

本章の冒頭で述べたように，品質要求は機能要求を修飾する情報と考えられる．よって，ある要求仕様書におけるすべての機能要求をある品質要求が修飾していれば，その品質要求は重要と仮定できる．逆に，まったく修飾する機能要求がない品質要求の場合，その品質要求は重要でないと仮定できる．しかし，この仮定には 2 つの問題点がある．第 1 に，すべての機能要求項目が等しく重視されているわけではなく，機能要求項目の間に優先順位があるため，品質要求が修飾している機能要求の割合が，単純にその品質要求の重要度とはならないことである．第 2 に，実際の要求仕様書においては複数の機能要求を同時に修飾する品質要求をまとめて記述する場合がある．よって，単純にそれぞれの機能要求項目の記述に基づき品質要求に修飾されているか否かを調べるだけでは，重要度を得ることができない可能性がある．

第 1 の問題点については，すでに数多く提案されている要求項目の優先度付けの技術を利用して解決可能である．既存の優先度付け技術は文献 [12] に数多く紹介されている．たとえば，これらの技術によって優先度付けされた順位に基づき，要求項目に重み付けすることで，個々の機能要求の重要度の差異を扱うことが可能である．また，既存の優先度付け技術には品質要求にも適用可能なもの [13] があり，品質要求自体の重要度の差異も識別することが可能である．

第 2 の問題点について，我々は品質要求によって修飾されている機能要求項目の単純な割合が重要度と一致するか否か，そして一致しない場合，一致するように補正することが可能かを実験を通して確認した．実験は以下のとおりである．まず，日本語で記述された要求仕様書において，品質要求の重要度の順序付けを専門家が行う．これを正解の重要度に基づく順位とする（**表 1** の「重要度」の列に相当）．この順位付けは AHP の一対比較 [14] を専門家が行った結果に基づき決定した．これとは別に，機能要求を述べている文を仕様書から抜き出し，修飾している機能要求の割合の高さに基づき，品質要求を順序付けする（表 1 の「単純割合」の列に相当）．さらに，以下に示す方針で複数の機能要求を修飾する品質要求を分配することで，品質要求が修飾している機能要求の割合を再計算し，品質要求を順序付けする（表 1 の「補正割合」の列に相当）．我々はこれらの順序の差の比較に際して，特に上位 5 位に位置する品質要求の差異に着目して分析を行った．その理由は，正解における上位 5 位の重要度がそれ以下の重要度に比べ大きい（上位 5 位は AHP の総合ウエイトが 0.1 以上，それ以下は 0.06 未満）ためである．実験は，システム開発に

かかわる外注仕様書作成マニュアル（案）[15] の中にあるサンプルから，文書 A「技術調査報告書 DB システム」と文書 B「集計システム」を利用して行い，文書 A については 3 人の専門家が合議のうえ，重要度を決定した．文書 B については 1 人の専門家が重要度を決定した．なお，対象とする品質要求の種類は，ISO-9126 [10] における品質副特性群を利用した．

複数の機能要求を修飾する品質要求を分配する方針は以下のとおりである．この方針は，実際の仕様書の内容の検討に基づき，分配すべき品質要求を含む要求項目，および分配先の機能要求項目の構文的，文書構造的な特徴を形式化したものである．

(1) 1 もしくは 2 種類の品質要求に修飾されている要求項目に着目する．
(2) 要求仕様書に出現する語句のうち，1%以下の出現頻度である語句を含む要求項目に着目する．
(3) 上記 (1), (2) で選択された要求項目は，他の機能要求項目に分配すべき項目か否かを分析者が判断する．判断に際しては，システムの目的を述べた序章や，システムの品質要求をまとめて記述した章に含まれるか否かが参考になる．分配すべきでないと判断した場合，着目しないこととする．
(4) 上記で着目した要求項目それぞれについて，修飾されている品質要求を調べる．
(5) 上記で着目した要求項目それぞれについて，機能要求項目に含まれる語句が，表題を表す文字列に含まれる章と節を選ぶ．
(6) (5) で選ばれた章や節に含まれる機能要求項目それぞれに対して，(4) で調べた品質要求を分配する．

この方針はツールによる半自動分配を可能とするため，機械的に処理できるステップを多く含んでいる．しかし，分配すべき要求項目や，分配先の決定において，人間がより多く介在すれば，より正解の重要度順序に近づけることが可能である．

結果として，品質要求によって修飾されている機能要求項目の割合に基づく品質要求の順序は，品質要求の重要度の順序とは単純には一致しないことが表 1 の「重要度」と「単純割合」の比較から確認できた．しかし，表 1 の「重要度」と「補正割合」の比較から，上記の方針に基づく補正によって，品質要求によって修飾されている機能要求項目の割合に基づく品質要求の順序を，品質要求の重要度の順序に近づけることが可能であることを確認した．

要求仕様書は主に自然言語で記述されるが，ゴールモデル等の高度に構造化された言語で記述される場合もある．しかし，ゴールモデルでさえ，モデルから文書を生成する研究 [16] や，実際に文書による仕様書を生成する機能を備えたツール [17] が存在する．よって，自然言語の要求仕様書における品質要求の重要度の手法は，ゴールモデル等の高度に構造化された言語による要求仕様書にも適用することが可能である．

以上から，あるソフトウェアの要求仕様書において，それぞれの品質要求が，どの程度重視されているかを測定することは可能であると結論付ける．測定は，それぞれの品質要求が修飾している機能要求項目の割合に基づき算出できる．機能要求項目間に優先度の違いがある場合には，その補正を既存技術で行うことが可能である．また，複数の機能要求項目を修飾する品質要求が，1 つの項目にまとめて記述されていた場合，それを分配することで補正を行うことが可能である．

3. 品質要求スペクトル分析の基本的なアイディア（旧手法）

本章では以降の章で改善を行う品質要求のスペクトル分析法 [6]（以降，旧手法と呼ぶ）の解説を行う．旧手法では要求仕様書における，ある品質要求の言及の度合いを重要度と見なしているが，その根拠は既存研究では明確に示されていなかった．しかし，本稿の 2 章の検討を通じて，その根拠を示すことができた．

ソフトウェア品質要求定義の記述は，要求仕様書内に横断的に記述される場合がある．そのため品質要求が十分に定義されているか否かを確認することは容易でない．このような背景からソフトウェアの品質要求のためのスペクトル分析手法 [6], [18] が提案され，品質要求が十分に定義されているか否かを確認する助けとなっている．手法の名称にあるように，同手法は電波や光学におけるスペクトル分析の手法から想起されている．光学等でのスペクトル分析では，複雑な波形が複数の異なる正弦波形（サイン波）に分解される．このような考えを**図 1** に示すようにソフトウェアにかかわる成果物にあてはめたのがソフトウェアの品質要求のためのスペクトル分析手法である．ソフトウェアにかかわる成果物（要求仕様書，設計書，コード等）には複数の品質要求にかかわる要素を含んでいる．そして，製品によって，どの品質要求が重視されるかは異なる．個々の品質要求を正弦波形と見なし，個々のソフトウェアにかかわる成果物を複雑な波形と見なすことで，光学のスペクトル分析と同様に，ある成果物では，どのような品質要求が言

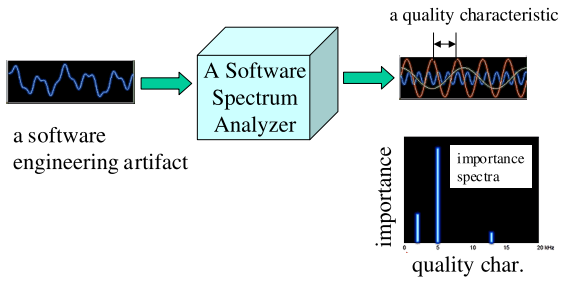


図 1 品質要求のためのスペクトル分析の基本的なアイディア
Fig. 1 Basic idea of spectrum analysis for quality requirements.

及されており，それらがどの程度重視されているかを測定できると考えた．ソフトウェア品質要求のためのスペクトル分析は「分析」という言葉が持つ本来の意味である「ある事物を分解して，それを成立させている成分・要素・側面を明らかにする」という意味を反映している．加えて，その分析結果から，以下のような洞察が得られることもすでに報告されている．

- 類似ソフトウェアとの比較から，仕様書における品質要求定義の妥当性を確認．類似したソフトウェアの品質要求スペクトルは類似しており，異なるソフトウェアのスペクトルは異なっていることに基づき [6], [18]，定義した要求仕様書の妥当性を確認する．たとえば，Web による販売システムは多数あるが，セキュリティ，パフォーマンス，ユーザビリティ等は，どれも同様に考慮するものと思われるため，品質要求のスペクトルも相互に類似していると思われる．もし，ある Web による販売システムのスペクトルが他の類似システムのスペクトルと異なる場合，品質要求定義に抜けや誤りがある可能性がある．むろん，意図的に特定の品質要求を重視（軽視）している場合もあるが，品質要求のスペクトル分析により，そのような意図も明示的に確認できる．

  この分析結果の実例は文献 [6] にすでに発表されている．この実例では，3 つの特徴が異なる Web ブラウザ，FireFox（FF），Internet Explore（IE），Opera（OP）が分析対象となっている．具体的な特徴の違いは，オープンソースか否か（公開ポリシの違い，FF はオープンソース），サポートするプラットフォームの違い（単一か多数か，OP は他に比べ多数，特にモバイルデバイスに強い）である．これらの特徴の違いから，相互運用性や変更容易性の重要度が異なってくるはずである．文献 [6] には，このような差異を数値化して示すことができたと報告されている．一方，ブラウザのような不特定多数が利用する対話型システムの場合，理解容易性，学習容易性，操作性はほぼ差異がないと思われる．この点も数値化して示すことができたと報告されている．なお，同文献では，別途，データベース系のシステムが分析されており，理解容易性，学習容易性，操作性はブラウザと比べて高くなく，直感的な重要度と一致していると報告されている．
- ソフトウェア開発過程を通して品質要求定義が継続的に考慮されているかの確認．品質要求定義は要求，設計，コーディング，保守等の開発過程を通して継続的に継承されるのが望ましい．しかし，途中の過程において品質要求への考慮が欠けていたため，品質要求を充足するために，たとえばコーディング段階に大きな負担がかかる場合もある．要求分析段階で時間効率性が要求されていたにもかかわらず，設計過程でそれが考慮されておらず，コーディング過程の技巧によって，要求された時間効率性を達成しなければならない場合が典型的な実例である．機能要求については伝統的な段階的詳細化の技術が，このような開発過程をまたがるトレーサビリティの監視に貢献する．しかし，品質要求については伝統的な手法は適用し難い．スペクトル分析を用いると，要求，設計，実装それぞれの成果物の品質スペクトルを独立して求めることができる．各過程の成果物から得た品質スペクトルを相互に比較することで，品質要求定義に関するトレーサビリティを確認でき，開発途中において品質要求を考慮していなかった兆しを発見することができる．

  この分析結果が有効であることは文献 [7] にすでに発表されている．同文献では，駐車場の自動集金システムの制御ソフトウェアを分析対象としている．同文献によると，要求にある 2 つの品質要求（パフォーマンスと変更容易性）が，実装（Java code）において欠落していることを，本分析手法によって，発見することができたと報告されている．

品質要求のスペクトル分析では，要求仕様書に横断的に記述されている品質要求に関する成分をいったん分解し，品質要求の種類ごとに累計し，他のスペクトルと比較する．よって，本来誤りがある（正解と差異がある）品質要求に関する抜けや，過剰記述が検出されない場合がある．たとえば，ある種類の品質要求の記述漏れと，同じ種類の品質要求の過剰記述が重なると，これらの誤りは相殺されなかったことになってしまう．これは品質要求のスペクトル分析の大きな制限である．ただし，比較の結果が異なる場合，品質要求の記述に差があることは保証できる．

図 2 を用いて，どのように品質要求のスペクトル分析

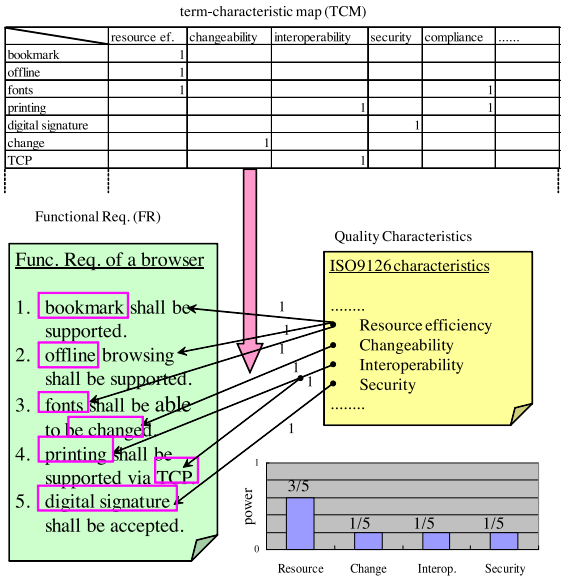


図 2 品質要求のためのスペクトル分析の基本的な考え方
Fig. 2 Basic idea of spectrum analysis for quality requirements.

器を実現するかを説明する．分析器の入力は要求文のリスト（要求項目リスト）と，品質要求項目のカタログ，そして，TCM（a term-characteristics map）と呼ばれる語句と品質要求項目との対応表である．要求項目リストは前章で述べた補正をすでに行ったものとする．TCM はある語句がある品質要求を述べるのに利用される可能性を示している．よって，その語句が含まれる要求文は，その品質要求によって修飾されている要求文の候補と見なされる．たとえば図 2 にある TCM では，語句 “digital signature” は品質要求 “security” を述べるのに利用されることを示している．図中では ISO9126 [10] の品質副特性を品質要求の分類項目として利用している．分析器の出力は要求項目リストの品質要求スペクトルであり，このスペクトルはそれぞれの品質要求項目の重要度のベクトルとして表現される．図 2 では説明の簡素化のため，4 種類の品質要求項目のみを記載しているため，四次元のベクトルであり，それぞれの次元がそれぞれの品質要求項目の重要度を示している．ISO9126 を用いた場合の品質要求スペクトルは二十次元以上のベクトルとなる．図 2 のスペクトルからは，資源効率性（resource efficiency）が，この要求項目リストにおいて，最も重視されているという結果となっている．そして，他の 3 つの品質要求は同程度に重要であるが，資源効率性よりは軽視されているという結果を示している．この出力を得るためには，それぞれの要求項目と品質要求項目との関係付けを行い，それぞれの品質要求項目と関連付いている要求項目の数を全要求項目の数で正規化することで，それぞれの品質要求項目の重要度を得ている．図中の例では要求項目 1，2，3 番は資源効率性と関係があると判断されている．この判断は，TCM に資源効率性と関係ある語句として “bookmark”，“offline”，“fonts” があげられていることを根拠としている．要求項目 4 のように 1 つの品質要求項目に関係ある語句が複数（“printing” と “TCP”）含まれていても，単に当該要求項目が当該の品質要求項目と関係あると見なす．

このように TCM は要求項目と品質要求項目との関係付けを行う分析者の判断を助ける材料となる．しかし，あくまで目安であるため，実際に関係があるか否かの判断は分析者の経験や能力に依存する．よって，旧手法ではソフトウェア品質要求のスペクトル分析を自動化することは容易ではなかった．

4. 分析履歴を用いたスペクトル分析（SAhis）

3 章で述べた従来の品質スペクトル分析手法（本稿では旧手法と呼んでいる）に基づき，改善手法である分析記録を用いた品質要求のスペクトル分析法 SAhis（Spectrum Analysis using History）を提案する．SAhis の目的は，要求分析者が品質要求のスペクトル分析を行う際に，主観的な判断を行う労力を低減させるか，もしくは主観的な判断を不要とすることである．SAhis では新しい TCM（確率付き TCM: TCMp: Term-Characteristics Map with probability）を用いる．TCMp を更新する必要がない限り，SAhis は自動的に遂行することが可能である．TCMp を更新する必要があるか否かは主観的に判断する必要がある．

4.1 TCMp の基本的な考え方

ソフトウェア工学の分野では，過去の実績値を用いて，累積移動平均を求め，見積りや予測を行うことが一般的である．たとえば，開発規模，開発時間，欠陥数の予測を行う手法が提案されている [19]．しかし，ソフトウェア成果物や開発にかかわる，どのような特性についての累積移動平均に着目すべきかは自明であるとはいえない．実際，既存手法においても，どのような測定値が，累積移動平均に基づき見積りを行うことに適したものかについての工夫がされている．たとえば，手法 [19] ではソースコード行数を新規，変更，再利用の 3 通りに分類することで，見積り精度を高めようとしている．旧手法 [6] は，要求項目中のある語句の出現をもとに，ある品質要求が要求項目を修飾しているか否かを判定している．しかし，ある要求項目に当該語句が出現しても，必ずある品質要求によって要求項目が修飾されているわけではない．そこで，過去に分析したある語句が出現する要求項目群の中で，その語句が出現していることを根拠として，ある品質要求に修飾されていると判断できる要求項目群の割合に着目する．この割合を，当該語句が出現するため，ある品質要求に関係するという事柄の確からしさ，すなわち確率と仮定する．本稿で提案する分析履歴を用いたスペクトル分析法は，このような仮定に基づき構築されている．この仮定が妥当か否かは 6 章で示す．

図 2 に示すように，従来の TCM のセルは 0 か 1 の値のみとり，語句と品質要求の関係の有無の情報しか与えない．**図 3** にあるように，SAhis における TCMp では，この値を前述のような 0 から 1 の間の確率値をとることができるように拡張した．図 3 を用いて，TCMp を用いた SAhis をどのように遂行するかを説明する．図 2 の TCM と同様に，TCMp はある要求項目とある品質要求項目が関係があるか否かの判断に利用される．図 2 にある旧手法では，この関係の有無のみに着目していたが，SAhis では，図 3 にあるように，この関係の有無について 0 から 1 の間の中間値（たとえば 0.3 や 0.6）を使うことを許し，どの程度，関係があるらしいかを示すことができるようにする．たとえば，図 3 の TCMp では，語句 “TCP” が出現する要求項目は，品質要求 “interoperability” と確率 0.9 で関係があるということを，“TCP” と “interoperability” の交点の値 0.9 で示している．一方，“bookmark” と “resource efficiency” の交点値は 0.6 であるため，“bookmark” が出現する要求

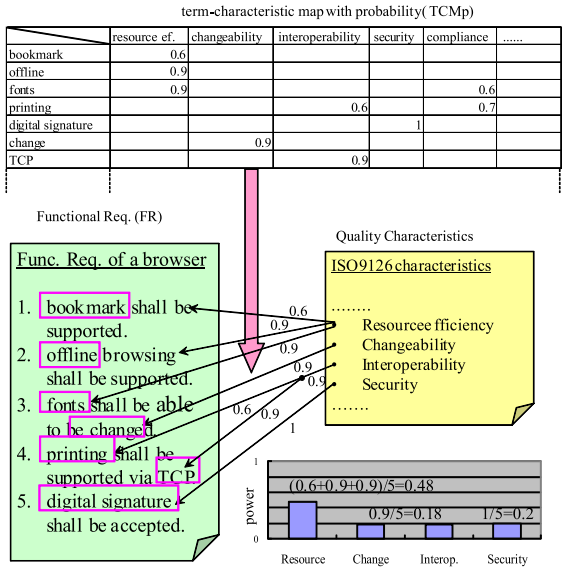


図 3 SAhis における品質要求のスペクトル分析
Fig. 3 Spectrum Analysis for quality requirements using analyses records (SAhis).

項目が “resource efficiency” について述べている確率は，前述の “TCP” が出現する要求項目が “interoperability” について述べている確率より低いことになる．なお，TCMp の空白の交点は TCM と同様にゼロを示している．

図 2 の旧手法では，ある品質要求の重要度を計測するために，単純に当該の品質要求と関係がある要求項目数を合計していた．SAhis でも同様に，ある品質要求の重要度を計測する．ただし，単純に要求項目数を合計するのではなく，前述の確率値を合計する．なお，この合計値は旧手法と同様に要求項目数で除算することで正規化する．たとえば図 3 では，品質要求 “resource efficiency” と 3 つの要求項目が関係し，それぞれ関係の度合いは TCMp により 0.6，0.9，0.9 である．よって，品質要求 “resource efficiency” の，この要求項目リストの重要度は，上記 3 つの度合いの合計を要求項目の数 5 で除算を行った値，すなわち，0.48（$= (0.6 + 0.9 + 0.9)/5$）となる．旧手法では単純に 0.6（$= 3/5$）となるが，SAhis では，品質要求が，それぞれの要求項目を修飾しているか否かの確率に基づき値が補正されている．個々の確率値は，前述のように，ある語句の出現する要求項目の中で，ある品質要求によってその要求項目が修飾されていた実績の割合である．たとえば，ある語句の出現はある品質要求を必ず示す場合，表現が断定的であると考えられるため，その語句を含む要求項目は当該の品質要求を重視していると受け取ることができる．一方，ある語句の出現はある品質要求を示す場合もあるし，そうでない場合もあったとすると，表現が断定的でないと見なされ，その語句を含む要求項目は当該の品質要求をそれほど重視していないと受け取ることができる．よって，ある語句がある品質要求を示している実績の高低を用いて，当該品質要求の重要度の高低の補正に利用することは妥当である．

ある要求項目が単一の品質要求との関係を示す複数の語句を含有していた場合，TCMp において当該語句と当該品質要求との確率値の最大値を用いる．図 3 の例では，要求項目 4 には 2 つの語句 “printing” と “TCP” が含まれており，それぞれの “interoperability” との確率は，TCMp によると 0.6 および 0.9 である．この場合には，要求項目 4 と “interoperability” との確率は最大値の 0.9 とする．

4.2 SAhis の遂行法

図 3 において，SAhis をどのように遂行するかを例を用いて説明した．**図 4** において，SAhis をデータフロー図（DFD）を用いて説明し，SAhis において何が自動的に遂行でき，何を主観的に判断しなければならないかを明確にする．なお，図 4 中の吹き出しは例もしくは補足説明を示している．この図では TCMp の交点要素を 0.6 や 0.9 等の小数ではなく，4/5 や 1/10 等の分数で表記する．4.1 節の冒頭で述べたように，TCMp における確率値は割合に基づき計算される．よって，確率値を利用する際には，分数や小数の表記の差に意味はない（計算法については次節で説明する）．また，図中のデータフローに流れるデータの種類は，データフローの流入先もしくは流入元のデータストア（図 4 では，矩形，ノート，角の丸い矩形，五角形で記述）に保存されたデータが流れるものとする．

図 4 に示すように，SAhis の入力は TCMp（図中の “oldTCMp :TCMp”）と要求項目リスト（“systemX :Reqs”）である．主な出力は “systemX :Reqs” のスペクトル（図中の “Xspectrum :Spectrum”）である．図中のプロセス “pickup terms” において，要求項目と品質要求の関係（図 3 中の矢印群に相当）が自動的に識別され，その関係がデータストアー “RCM”（Requirements Characteristics Map）に内部的に保存される．“RCM” 中のデータはプロセス “summarize” で自動的に品質要求ごとに累積し，正規化され，データ “Xspectrum :Spectrum” が出力として生成される．図 4 に示すように “Xspectrum :Spectrum” を棒グラフとして可視化してもよいし，他のスペクトラムとの類似度を計算してもよい．上記のように，スペクトラムの生成は自動的に行うことができる．

図 4 中にある吹き出し中の例を用いて，より具体的にスペクトルの生成法を説明する．“systemX :Reqs” の吹き出しにあるように 3 つの要求項目からなる要求項目リストを分析することを想定する．TCMp は “oldTCMp :TCMp” の吹き出しにある 3×3 の TCMp を利用する．要求項目 1 と 3 は語句 “term2” を含み，TCMp によるとこの語句は “YYYbility” と 4/5 の確率で関係しているため，図中の “:RCM” の吹き出しにあるように RCM において，要求項目 1 と 3 および “YYYbility” との交点の値は 4/5 に自動的

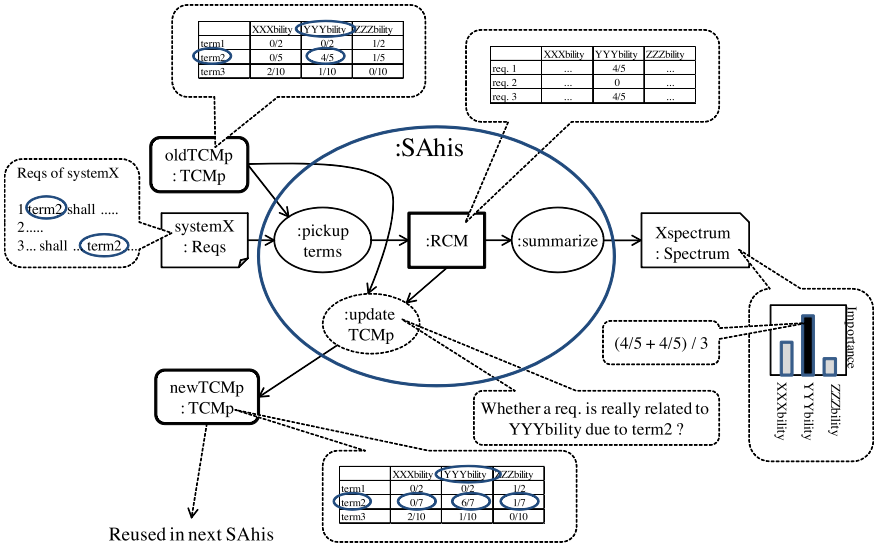


図 4 分析履歴を用いてスペクトル分析を行う手順と TCMp の更新手順（DFD により記述）
Fig. 4 How to perform Spectrum Analysis for Quality Requirements using Analyses Records (SAhis), and how to update TCMp written in DFD. A process SAhis contains an internal data store RCM (Requirements Characteristics Map).

に設定される．なお，要求項目 2 には “term2” は出現していないものとする．図 3 の例と同様に，“YYYbility” の重要度は $(4/5 + 0 + 4/5)/3$ の計算結果となる．なぜならば，“systemX :Reqs” は 3 つの要求項目から構成され，それぞれと “YYYbility” との関係の確率は，それぞれ，4/5，0，4/5 だからである．他の品質要求についても同様の計算を行い，結果として，出力 “Xspectrum: Spectrum” を生成することができる．

4.3 TCMp の更新法

SAhis の分析結果の品質は TCMp の品質に依存するため，TCMp をどのように生成し更新するかは重要な問題である．図 4 のデータフロー図は，TCMp をどのように更新するかのプロセス（図中の “update TCMp”）も含んでいる．ある分野の安定した信頼性のおける TCMp がすでに存在するのであれば，このプロセスを遂行する必要はない．TCMp が安定し信頼性がおけるか否かは，その運用から運用者が判断することになる．4.1 節の冒頭で述べたように，TCMp における確率の値は，ある語句が出現する要求項目の中で，実際に，その語句が出現するために，ある品質要求に修飾されていると判断された要求項目の数の割合を用いている．

未知の分野の要求項目リストを最初に分析する際には TCMp は存在していないため，スペクトル分析を分析者が主観的に遂行しなければならない．最初の分析結果に基づき，ある分野についての，最初の TCMp を作成しなければならない．ここで，ある要求項目リスト中の 5 項目に語句 “term2” が含まれていたと仮定し，どのように最初の TCMp を作成するか説明する．“term2” が含まれない要求項目は確率値の計算には利用しない．まず，分析者は，この 5 項目の内容を示す文章をそれぞれ読み，その内容を理解する．そして，それぞれの項目がどの品質要求と関係があるかを判断しなければならない．仮に，この 5 項目のうち 4 項目が語句 “term2” が含まれているという理由から，“YYYbility” に関係があると分析者が判断したとする．また，この 5 項目のうち 1 項目が語句 “term2” が含まれているという理由から，“ZZZbility” に関係があると分析者が判断したとする．そして，語句 “term2” が含まれているという理由から，“XXXbility” に関係がある要求項目はなかったと分析者が判断したとする．このような判断結果をもとに，図 4 の “oldTCM :TCMp” の吹き出しにあるような初期の TCMp を作成することができる．むろん，初期の TCMp は分析者が主観的に作成してもよい．

既存の TCMp を分析結果に基づき更新する場合でも，分析者の主観的な判断が必要となる．TCMp は，図 4 中のプロセス “update TCMp” で，:RCM のデータに基づき更新する．プロセス “update TCMp” の入力は既存の TCMp（図中の “oldTCMp: TCMp”）と直前の分析結果の “:RCM” であり，出力は更新された TCMp（図中の “newTCMp: TCMp”）である．TCMp 更新のためには，“:RCM” に保存された自動的な分析結果を，分析者が内容を確認し，吟味し直さなければならない．図中の例にある “:RCM” では，要求項目 1 と 3 が語句 “term2” を含んでいるため，“oldTCMp: TCMp” に基づき，それぞれの項目は “YYYbility” と 4/5 の確率で関係があると自動的に判断されている．

TCMp を更新には，この 2 つの要求項目が “YYYbility” と本当に関係があるか否かの判断結果を用いる．そのため

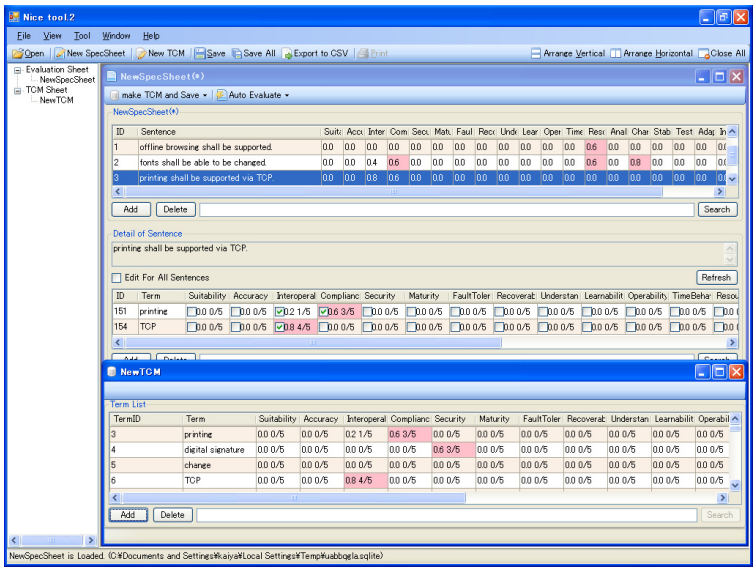


図 5 SAhis 遂行支援ツール QRAST の画面例
Fig. 5 A snapshot of QRAST, a supporting tool for SAhis.

に，要求項目 1 と 3 の内容を分析者が理解し，それぞれの項目が “YYYbility” と本当に関係があるか，あるとしたら，その理由は “term2” が出現しているためかを判断しなければならない．要求項目 2 は，“term2” が出現しないため，“term2” に関する TCMp の更新には利用されない．たとえば，要求項目 1，3 ともに “term2” が出現するため “YYYbility” に関係があることが正しいと分析者が判断した場合，TCMp における “term2” と “YYYbility” の交点の値は，4/5 から 6/7（$= (4+2)/(5+2)$）に更新される．分母に追加された 2 は要求項目 1 と 3 の 2 個の要求項目を示す．分子に追加された 2 も同様である．もし，ともに正しくないと分析者が判断した場合，TCMp における “term2” と “YYYbility” の交点の値は，4/5 から 4/7（$= (4+0)/(5+2)$）に更新される．図 4 の例では前者の判断がされたことを例示している．なお，更新段階で新規に出現した語句は，最初の TCMp を作成するのと同様に確率を決定する．この更新戦略が TCMp の品質を向上させることができるか否かは 6 章で確認する．

5. 支援ツール QRAST

SAhis の遂行を支援するために，我々は QRAST（Quality Requirements Analysis Support Tool）というツールを試作した．このツールは図 4 にあるプロセス “pickup terms” と “summarize” を自動的に行い，分析者がプロセス “update TCMp” を遂行するための支援を行う．ツールは C#で実現されており，文章中の語句を識別するために外部の形態素解析ツール [20] を利用している．

図 5 にツールの画面例を示す．確率を見やすくするために，確率が 0.5 以上のセルは明るい色（ピンク）で強調されている．図 5 の右下部には，図 4 の “oldTCMp :TCMp” に相当する TCMp が “NewTCM” とラベル付けされたウインドウに表示されている．“RCM” は “NewSpecSheet” とラベル付けされた右上部のウインドウに表示されており，そこには 3 つの要求項目が見えている．この画面例では要求項目 3 “printing shall be supported via TCP” がフォーカスされており，当該項目から TCMp に登録された 2 つの語句 “printing” と “TCP” を自動的に抜き出し，TCMp のこれらの語句に対応する行を図中央に列挙している．この例における TCMp では，“printing” は “Compliance” と 3/5 の確率で関係しているため，この値 3/5 が要求項目 3 と “Compliance” との関連の度合いとなる．一方，“printing” は “Interopera(bility)” と 1/5，“TCP” は “Interopera(bility)” と 4/5 の確率で関係しているため，大きい方の値 4/5 が要求項目 3 と “Interopera(bility)” との関連の度合いとなる．

6. 評価

6.1 目的

SAhis を繰り返し TCMp を更新すればするほど，SAhis で自動的に分析された品質要求のスペクトルは正しくなることを確認することを本評価の目的とする．この目的を達成するために，我々は実験を行い，その結果を分析することとした．

4.3 節で説明した TCMp 値の更新は一種の機械学習 [21] である．なぜなら，まず，学習データとして，ある語句が出現する要求項目について，その語句が出現するため，その要求項目はある品質要求に修飾されている事例を分析履歴から収集する．その学習データに基づき確率値を計算し，確率値に基づき未分析の要求項目が，ある品質要求に修飾されているかどうかを予測しているからである．なお，多

くの機械学習においても，学習データ自体は主観的な判断や実際に発生した事実に基づき収集せざるをえない．また，予測結果の成否を確認するために，正解が必要である．本評価の目的を，機械学習の視点から述べると，4.3 節で述べた TCMp 値の更新法に基づき学習を続けることによって，予測結果が正解に近づくか否かを確認することで，更新法が妥当であるか否かを判断することである．

6.2 実験設定

我々は FireFox（FF），Internet Explore（IE），Opera（OP）の 3 つの Web ブラウザの要求項目リストを用いて，SAhis における TCMp の更新部分を遂行することで，3 世代の TCMp を作成した．

- $TCMp_1$：FF の要求項目リストのみに基づく TCMp.
- $TCMp_2$：FF および IE の要求項目リストに基づく TCMp.
- $TCMp_3$：FF，IE，OP の要求項目リストに基づく TCMp.

なお，それぞれの要求項目リストはそれぞれのブラウザのヘルプファイルをもとに著者らが作成した．要求項目間には重要度の差はないものとした．また，作成に際しては，複数の要求項目を修飾する品質要求をまとめて記述せず，分配して記述した．これによって，2 章で言及した問題点を解消し，言及度を重要度と見なすことができる．TCMp の更新も著者のうちの 1 人が行った．

上記の 3 世代の TCMp それぞれを用いて，FF，IE，OP それぞれを SAhis に基づいて分析し，品質要求スペクトルを QRAST を用いて自動的に計算した．

前述の評価目的から，我々はより後世代の TCMp を用いた分析結果ほど，正解に近いと仮定している．この仮定を確認するためには，それぞれの要求項目リストの正解の品質スペクトルが必要である．そこで，我々は正解の品質スペクトルの作成を専門家に依頼した．この専門家はテレビ局の Web ページ作成業務にかかわっており，多数のブラウザに精通している．この専門家は本実験での TCMp を更新した人物とは別人である．

正解のスペクトルと SAhis によるスペクトルの類似性を測定することで，分析結果が正解に近いか否かを判断できる．スペクトルは多次元ベクトルの一種であるため，類似性判定の尺度としてコサイン類似度を用いる．ベクトル $\vec{a}$ と $\vec{b}$ のコサイン類似度（cossim: cosine similarity）は以下である．

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

2 つのベクトルが完全に一致する場合，その値は 1 である．品質要求のスペクトルは正の値のみで構成されるため，コサイン類似度の値は 0 から 1 の値をとる．コサイン類似度は 2 つのベクトル間の角度のコサイン値を示しているが，ベクトルの類似度をより直感的に理解するためには，ベクトル間の角度そのものが有効である．品質要求のスペクトルは正の値のみで構成されることから，角度は 0 度から 90 度の値をとる．たとえば，コサイン類似度において 0.96 と 0.99 との差は，0.03 であり，とりうる値が 0〜1 であることを考慮すると，3%の差ということになる．しかし，角度に変換すると 16 度と 8 度となり，その差の 8 度は 8.9%（= 8/90 ∗ 100）となり，コサイン値に比べおよそ 3 倍の差が認識できる．この差はコサイン関数が線形関数ではないことに起因する．そこで，コサイン類似度（*cossim*）を以下の定義で角度に変換したものを実験結果に利用する．この値が 0 度に近ければ 2 つのベクトルは，より類似しており，90 度に近ければ，相互にまったく関係のないベクトルということになる．なお，*arccos* は逆コサイン関数である．

$$angle(\vec{a},\vec{b}) = arccos(cossim(\vec{a},\vec{b})) * 180/\pi$$

表 2 各世代の TCMp を用いた場合の分析結果の正解との類似度（角度で表示）

Table 2 Similarity between the correct spectrum and a spectrum analyzed using $TCMp_1$, $TCMp_2$ or $TCMp_3$ (Degree of angle).

	FF	IE	OP
$TCMp_1$	26.0	30.7	10.0
$TCMp_2$	10.0	13.9	7.7
$TCMp_3$	8.0	7.1	3.7

6.3 結果

表 2 に各世代の TCMp を用いて FF，IE，OP それぞれの要求項目リストを SAhis に基づきスペクトル分析した結果と正解との類似度を示す．表に示すように第 1 世代 $TCMp_1$ を用いた場合に比べ，第 3 世代 $TCMp_3$ を用いた場合は小さな開きとなっている．たとえば，FF の場合，26.0 度から 8.0 度に減少している．よって，TCMp の世代が重ねられるごとに，分析結果はより正確になっているといえる．参考までに $TCMp_1$ で FF を分析した結果と正解との違い（26.0 度の差に相当）の実データを棒グラフで図示した結果を**図 6** に示す．

6.4 議論

表 2 に示すように，TCMp の世代が更新されるほど，SAhis によるスペクトル分析の結果は正解に近づいている．よって，本評価における仮説である TCMp を更新すればするほど，SAhis による分析結果は正しくなることは確認されたといえる．表 2 に示すように，それぞれの世代で分析された要求項目リストは FF，IE，OP の 3 件である．よって，統計によって，世代による類似度の差異が有意か否かを検定することは困難であると思われる．実際，第 1 世代（$TCMp_1$）と第 3 世代（$TCMp_3$）との平均の差について

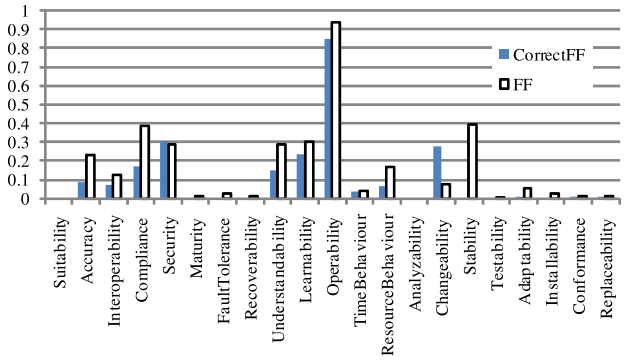


図 6 $TCMp_1$ で FF を分析した結果と正解との比較（FF が分析結果，CorrectFF が正解）

Fig. 6 Comparison between the correct spectrum and a spectrum analyzed using $TCMp_1$ about FF.

t 検定を行うと，p 値が 0.064 となり 5%（0.05）の有意水準には届かない．しかし，たった 3 件のデータにおいて，0.064 であるため，TCMp を更新すればするほど，SAhis による分析結果は正しくなる傾向はあるといえる．

以下に，この評価における妥当性への脅威をまとめたい．内部妥当性は実験における独立変数への影響が妥当であるか否かである．この実験での独立変数は SAhis によって分析されたスペクトル，そして正解のスペクトルである．正解のスペクトルは SAhis 考案前に作成されており，正解を作成した専門家は SAhis 考案および今回の分析にいっさい関与していない．よって，独立性は確保されている．外部妥当性は結果の一般性についてである．Web ブラウザのドメインの要求項目リストのみを実験で用いたため，他のドメインでも有効であるか否かという点で外部妥当性への脅威は残る．また，FF，IE，OP の順に情報を追加した TCMp の更新のみを扱っているという点でも外部妥当性への脅威は残る．構成妥当性は独立変数の測定に関する脅威である．独立変数である品質要求のスペクトルは N 次元（実験では N は 20 以上）の正数ベクトル値であるため，その類似度の測定にコサイン類似度を用いることは妥当である．結果にあるように，それぞれの世代における分析結果は 3 通りしかない．これらのデータを用いて統計的検定を行った場合，かなりの大きな差異がない限り，妥当な有意水準（5%程度）で，世代間の類似度が異なることを示すことは難しい．この点から結果妥当性への脅威が残る．

7. 関連研究

本稿で扱う TCM や TCMp は一種のドメイン知識である．このようなドメイン知識を用いて要求獲得や要求分析を行う研究は多数あるが，ドメイン知識自体の収集や改善に関係する研究は少ない．たとえば，文書から要求獲得のためのドメインオントロジを生成する研究 [22] や，Web マイニングでドメイン知識を改善する研究 [23] は見られる．しかし，これらは要求獲得がターゲットであり，記述された要求仕様書を評価・解析することを目標としていない．また，品質要求のような記述が横断的になる要求にも特化していない．

信頼性やセキュリティ等の品質要求について，そのレベルを測定するための研究 [24], [25] は数多く行われている．我々の研究はその点に注目しているのではなく，品質要求が適切な分量，定義されているか否かに注目している．品質要求定義の量を測定することは要求仕様書の完全性や正当性（correctness）に関係し [1] 我々が提案するスペクトル分析法の具体的な利点は 3 章で紹介済みである．仕様書の品質に着目したある文献 [26] では，仕様書の品質とプロジェクトの成功との関係に注目した研究を発表している．この研究での成果から，仕様書の品質とプロジェクトの成功との関係は見出せるが，仕様書の品質を向上させる方針は与えていない．一方，スペクトル分析は，類似した既存分析結果との比較に基づき，具体的に定義が不足している品質要求を指摘することができる．これによって，仕様書の品質を向上させる方針を与えることができる．機能要求の品質に関する経験的研究 [27] も報告されている．しかし，この研究では品質要求には注目していない．ある文献 [28] では ISO9126 等の分類は要求獲得の時点ではあまり使われていないという報告がある．我々の手法はすでに記述された，もしくは予稿として記述された仕様書を分析するための手法である．よって，要求獲得段階とは異なり ISO9126 等の分類を利用することは妥当であると思われる．

本稿では，ある語句が要求項目に出現することにより，その要求項目がある品質要求に修飾されているか否かの実例を過去の分析履歴から学習データとして収集し，未知の要求項目がある品質要求に修飾されているか否かを予測している．この考えを，さらに一般化した場合，たとえば，複数の語句の組合せに基づき，要求項目が品質要求に修飾されているかの予測をすることも可能である．このような予測は機械学習 [21] の世界で広く行われおり，ある語句の組合せを含む要求項目と実際に修飾されているか否かの対である学習データを必要とする．前述のように本稿における分析履歴も，この学習データの一種である．本稿の貢献の 1 つは，要求項目に含まれる複数の語句の組合せや，場合によっては順序まで考慮した複雑な機械学習を行わなくても，正確な予測ができたことを示せたことである．複雑な機械学習を用いた予測によって，どの程度，予測品質が向上するかを今後検討してゆきたい．

ソフトウェア分野では，過去の分析履歴を加味した累積移動平均により予測を行うことは広く行われている．たとえば，文献 [19] では，規模，工数，欠陥数の予測にこの考え方を適用している．このような予測を行うアイディア自体は自明であるが，何に基づき，何を予測するのかは，自明ではない．文献 [19] の例では，規模予測を行うために，ソースコードを新規，修正，再利用の 3 通りに分類する工

夫を行っている．また，前述の機械学習においても，学習データが持つ属性として何を選択するか自体が重要な要素となる [29]．本稿での手法は，前段落で述べたように，ある語句が要求項目に出現するか否かという属性のみに注目している．

8. おわりに

要求仕様書において，どの品質要求が，どの程度，重視されているかを測定することは，その仕様書に基づくソフトウェア開発に有益な知見を与える．すでに提案されている品質要求のためのスペクトル分析はこのような測定を行うための方法の 1 つである．しかし，同手法には問題点があった．第 1 の問題点は，測定されている内容が重要度を表しているか否かの検討が十分に行われていなかったことである．第 2 の問題点は，分析に主観的判断が必要であるため，その遂行が効率的でなかったり，場合によっては誤った結果になったりする恐れがあったことである．本稿では，これらの問題点を解決した結果を示した．

第 1 の問題点については，既存技術の調査と実験を通して，要求仕様書内における機能要求項目を修飾している割合を重要度と見なしてよいという結論を得た．第 2 の問題点については，分析結果の履歴に基づき分析判断の確率を蓄積することで，スペクトル分析での主観的判断を低減，もしくは削除する改善法を考案し，評価を行った．今後は関連研究で概説したように，多数の属性に着目した複雑な機械学習を行うことで，予測品質を高めることが可能か否かを検討したい．

参考文献

[1] IEEE Recommended Practice for Software Requirements Specifications, IEEE Std. 830-1998 (1998).

[2] Hamilton, D., Covington, R., Kelly, J., Kirkwood, C., Thomas, M., Flora-Holmquist, A., Staskauskas, M., Miller, S., Srivas, M., Cleland, G. and MacKenzie, D.: Experiences in applying formal methods to the analysis of software and system requirements, *Workshop on Industrial-Strength Formal Specification Techniques*, p.30 (online), DOI: http://doi.ieeecomputersociety.org/10.1109/WIFT.1995.515477 (1995).

[3] Sengupta, S., Kanjilal, A. and Bhattacharya, S.: Requirement Traceability in Software Development Process: An Empirical Approach, *IEEE International Workshop on Rapid System Prototyping*, pp.105–111 (online), DOI: http://doi.ieeecomputersociety.org/10.1109/RSP.2008.14 (2008).

[4] Ohnishi, A.: Improvement of the Correctness of Scenarios with Rules, *IEICE Trans.*, Vol.89-D, No.4, pp.1337–1346 (2006).

[5] Blaine, J.D. and Cleland-Huang, J.: Software Quality Requirements: How to Balance Competing Priorities, *IEEE Software*, Vol.25, No.2, pp.22–24 (2008).

[6] Kaiya, H., Tanigawa, M., Suzuki, S., Sato, T., Osada, A. and Kaijiri, K.: Improving Reliability of Spectrum Analysis for Software Quality Requirements Using TCM, *IEICE Trans. Information and Systems*, Vol.E93-D, No.4, pp.702–712 (2010).

[7] Kaiya, H., Amemiya, K., Shimizu, Y. and Kaijiri, K.: Towards an Integrated Support for Traceability of Quality Requirements using Software Spectrum Analysis, *Proc. 5th International Conference on Software and Data Technologies* (*ICSOFT*), Athens, Greece, INSTICC, pp.187–194 (2010).

[8] Kan, S.H.（著），古山恒夫ほか（監訳）：ソフトウェア品質工学の尺度とモデル，共立出版 (2004).

[9] Pressman, R.S.: *Software engineering: A practitioner's approach*, 5th edition, McGraw-Hill computer Science series, McGraw-Hill (2001).

[10] International Standard ISO/IEC 9126-1: Software engineering - Product quality - Part 1: Quality model (2001).

[11] Chung, L., Nixon, B., Yu, E. and Mylopoulos, J.: *Non-Functional Requirements in Software Engineering*, Academic Publishers (1999).

[12] Eds., A.A., Berander, P. and Andrews, A.: *Engineering And Managing Software Requirements*, chapter Requirements Prioritization, pp.69–94, Springer (2005).

[13] Giesen, J. and Volker, A.: Requirements Interdependencies and Stakeholders Preferences, *IEEE International Conference on Requirements Engineering*, p.206 (online), DOI: http://doi.ieeecomputersociety.org/10.1109/ICRE.2002.1048528 (2002).

[14] Saaty, T.L.: *The analytic hierarchy process: Planning, priority setting, resource allocation*, 2nd edition, RWS, Pittsburgh (1990).

[15] 経済産業省大臣官房情報システム厚生課情報システム室：システム開発に係る外注仕様書作成マニュアル（案）(2003)，入手先 ⟨http://kaiya.cs.shinshu-u.ac.jp/i30728aj/⟩（参照 2011-08）．オリジナルリンクは消失しているため複製を参照．

[16] Ncube, C., Lockerbie, J. and Maiden, N.A.M.: Automatically Generating Requirements from i^* Models: Experiences with a Complex Airport Operations System, *REFSQ*, pp.33–47 (2007).

[17] Objectiver: Modeling tool for KAOS, available from ⟨http://www.objectiver.com/⟩ (accessed 2011-08).

[18] Kaiya, H., Tanigawa, M., Suzuki, S., Sato, T. and Kaijiri, K.: Spectrum Analysis for Quality Requirements by Using a Term-Characteristics, *21th International Conference Advanced Information Systems Engineering* (*CAiSE 2009*), Amsterdam, The Netherlands, LNCS 5565, pp.546–560 (2009).

[19] Humphrey, W.S.: *A Discipline for Software Engineering*, Addison-Wesley (1995). The Complete PSP^{SM} Book.

[20] A morphological parser for the Japanese language (2007), available from ⟨http://chasen-legacy.sourceforge.jp/⟩ (accessed 2010-12).

[21] Witten, I.H.: *Data Mining, 2nd Edition: Practical Machine Learning Tools and Techniques*, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann (2005).

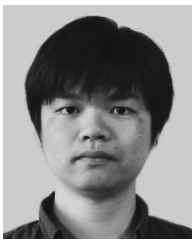
[22] Kitamura, M., Hasegawa, R., Kaiya, H. and Saeki, M.: An Integrated Tool For Supporting Ontology Driven Requirements Elicitation, *ICSOFT 2007, 2nd International Conference on Software and Data Technologies*, Barcelona, Spain, pp.73–80 (2007).

[23] Kaiya, H., Shimizu, Y., Yasui, H., Kaijiri, K. and Saeki, M.: Enhancing Domain Knowledge for Require-

ments Elicitation with Web Mining, *Proc. 17th Asia-Pacific Software Engineering Conference* (*APSEC 2010*), Sydney, Australia, pp.3–12 (2010).
[24] Kumar, K.S. and Misra, R.: An Enhanced Model for Early Software Reliability Prediction Using Software Engineering Metrics, *Secure System Integration and Reliability Improvement*, pp.177–178 (online), DOI: http://doi.ieeecomputersociety.org/10.1109/SSIRI.2008.32 (2008).
[25] Mellado, D., Fernández-Medina, E. and Piattini, M.: A comparison of software design security metrics, *Proc. 4th European Conference on Software Architecture: Companion Volume*, *ECSA'10*, pp.236–242, ACM, New York, NY, USA (online), DOI: http://doi.acm.org/10.1145/1842752.1842797 (2010).
[26] Knauss, E. and Boustani, C.E.: Assessing the Quality of Software Requirements Specifications, *RE*, pp.341–342 (2008).
[27] España, S., Condori-Fernández, N., González, A. and Pastor, O.: Evaluating the Completeness and Granularity of Functional Requirements Specifications: A Controlled Experiment, *RE*, pp.161–170 (2009).
[28] Ozkayad, I., Bass, L., Sangwan, R.S. and Nord, R.L.: Making Practical Use of Quality Attribute Information, *IEEE Softw.*, Vol.25, No.2, pp.25–33 (2008).
[29] Menzies, T., Greenwald, J. and Frank, A.: Data Mining Static Code Attributes to Learn Defect Predictors, *IEEE Trans. Softw. Eng.*, Vol.33, pp.2–13 (online), DOI: http://doi.ieeecomputersociety.org/10.1109/TSE.2007.10 (2007).

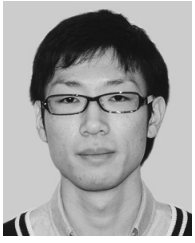
海谷 治彦 （正会員）

1994 年東京工業大学博士（工学）取得．現在，信州大学工学部准教授，国立情報学研究所客員准教授．



鈴木 駿一

2010 年信州大学修士（工学）取得．

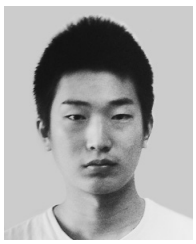


小川 享

2011 年信州大学学士（工学）取得．

谷川 正明

2011 年信州大学修士（工学）取得．



梅村 真弘

2011 年信州大学大学院修士課程在学．

海尻 賢ニ （正会員）

1977 年大阪大学工学博士取得．現在，信州大学工学部教授．