

自然言語処理を用いて設計書から試験項目と そのメタ情報を抽出する研究

寺澤 賢人[†] 丸山 広[†] 高嶋 章雄[†] 中村 太一[†]
東京工科大学大学院[†]

1. はじめに

システム開発における上流工程の各工程では設計書が成果物として作成される。設計書に記述されている要件が曖昧であることや記述されていなければならない内容の抜け漏れ等があると、テスト工程に進むまでシステムのエラーを発見することが困難である [1][2]。そのため、開発工程が進むにつれ、エラーを発見した際に戻らなければならない工程数が増える。その結果、システム開発の工数とコストが増加する。また、システム開発における手戻りは、システム開発費全体の 30~50%を占めている[3]。さらに、要件定義工程の成果物である要件定義書の誤りによる手戻りにかかるコストがその内の 70~85%を占めている[3]。これらのことから上流工程の設計書の完成度を高め、手戻りの回数を減らすことが重要である。

2. 関連研究

システム開発の各工程に対し支援手法が提案されている。外部設計書からシステムテスト工程の試験項目の抽出する手法[4]、ソースコードを解析し構成要素を効果的に可視化する支援手法[5]、労力の配分を考えシステムの信頼性と工数を削減する手法[6]、要件抜け漏れを防ぐための要求整理手法[7]などが提案されている。

要件定義書から要件の抜け漏れを防ぐ方法や要件定義書のレビューを繰り返す方法は組み込まれているが、要件定義書から運用テスト工程の試験項目を自動抽出して運用テスト工程からの手戻り回数を減らす手法は提案されていない。

3. 研究の位置づけ

要件定義書は企業により記述方法や要件の粒度が異なる。そのため、様々な記述方法からの試験項目の抽出が必要となる。

システムの規模が大きいほど要件定義書の作成に携わる人数と要件定義書を構成するドキュ

メント数が増加し、要件定義書が複雑化される。そのため要求が変更されるに伴い変更の影響を受けた試験項目を的確に探し出す必要がある。

本研究は、要件定義工程と運用テスト工程に着目し、要件定義書から、運用テスト工程で使用される試験項目の自動抽出を行う。運用テスト工程で用いる試験項目を要件定義工程でレビューすることで、手戻りの回数を減らす。さらに、抽出した試験項目間の関係をメタ情報として抽出することで、システムに対する要求の変更があった際にも要件の抜け漏れを防ぐ方法を提案する。

4. 提案手法

試験項目の抽出では、試験項目の抜け漏れを防ぐため再現率を重視する。このため、不要な試験項目を除外し、再度除外された試験項目が除外して良いのかを確認し、自動抽出する必要がある。

要求変更が生じた際に、変更の影響を受けた試験項目を特定するため、要件定義書を構成するドキュメント間の関係を示すメタ情報を抽出する必要がある。メタ情報として、参照関係とシステムを構築するサブシステム間のデータの送受信の関係を抽出する。

図 1 に本研究で提案する、試験項目とメタ情報を抽出するまでの流れを示す。

①辞書を用いた試験項目の抽出

ホワイトリスト辞書には試験項目として抽出したい文章に含まれているキーワードを登録する。ブラックリスト辞書には、試験項目として不要なキーワードを登録する。ブラックリスト辞書で不要な試験項目をはじいた後にホワイトリスト辞書を用いて不要かどうかを再確認する。構文パターン辞書には参照関係とサブシステムとデータの流れを示す構文パターンを登録する。表 1 に各辞書の具体例を示す。

②試験項目 DB について

データベースには抽出した試験項目と抽出したメタ情報と過去に抽出した試験項目を格納しておく。更に、抽出した試験項目と過去に抽出した試験項目を N グラム方式にて比較を行い、新たに追加された試験項目、変更された試験項

A Method for Extracting test case and its meta data from specification by using natural language processing
Kent Terasawa, Hiroshi Maruyama,
Akio Takashima, Taichi Nakamura
[†]Tokyo University of Technology Graduate School

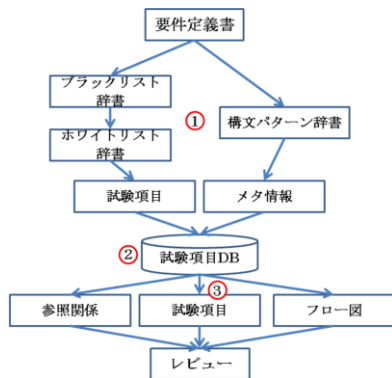


図 1 試験項目とメタ情報の抽出手順

目、削除された試験項目を判別する。

③抽出される試験項目とメタ情報について

試験項目とメタ情報を出力する際にそれぞれのドキュメントから抽出したかが分るようにし、試験項目とメタ情報を関連付けることで、要求の変更があった際に影響が出る試験項目を特定することができる。

5. 評価実験

提案手法で作成した試験項目がレビューの際に有効か否かを考察するため、人手と提案手法による試験項目の抽出をそれぞれ行い、適合率、再現率を評価した。

評価に使用する要件定義書には、東京証券取引所一次世代システム arrowhead-の要件定義書と東京工科大学・タンジブル・ソフトウェアにて Web アプリケーションとして外部公開している PROMASTER[8]の要件定義書を用いた。arrowheadは A4用紙約 1400枚、PROMASTERは A4用紙約 7枚で要件定義書が構成されている。

表 2 に arrowhead の要件定義書の一部を用いた試験項目抽出結果を示す。表 3 に PROMASTER の要件定義書を用いた試験項目抽出結果を示す。

6. 考察

表 2 と表 3 より、システムの規模が大きい程、文章中に様々な表現が使用されるためブラックリスト辞書の網羅性が低くなり、再現率及び適合率が下がる。一方、再現率と適合率を比較すると、適合率が低いことが分る。これは、ホワイトリスト辞書に登録されている単語数が少ないためであり、単語数を増やすことで適合率は上がると考えられる。

7. おわりに

要件定義書から運用テスト工程で用いる試験項目の抽出手法を提案した。評価結果から適合率より再現率が高いため、人手でレビューをす

表 1 辞書の内容

ブラックリスト	ホワイトリスト	構文パターン
表[0-9]	障害	トリガー.*?する
図[0-9]	市場	参照
以下に	例外	参照のこと

表 2 評価結果 (arrowhead)

	人手による	提案手法
抽出試験項目数	291	324
正しい試験項目数	291	276
抽出時間	58分 12秒	3分 24秒

再現率 0.948 適合率 0.834

表 3 評価結果 (PROMASTER)

	人手による	提案手法
抽出試験項目数	58	58
正しい試験項目数	58	58
抽出時間	8分 2秒	2秒

再現率 1.000 適合率 1.000

ることを考慮すればこの提案手法は有効であると考えられる。

今後の課題とし、要件の仕様が変更された際に、仕様変更により影響が出る試験項目に対し参照関係やフロー図を用いて特定した後に、試験項目ごとに優先順位をつける必要がある。これにより致命的な修正エラーを防ぐことができる。また、試験項目を抽出する際には適合率が高いことが条件であるため適合率を更に上げる必要がある。

謝辞

本研究は、文部科学省の平成 22 年度私立大学学術研究高度化推進事業オープン・リサーチ・センターのタンジブル・ソフトウェア教育の研究の助成による。

参考文献

- [1]劉功義, 他, "RFP の評価と品質向上に関する研究", プロジェクトマネジメント学会 2009 年度春季研究発表大会予稿集.
- [2]宇治浩明, "東証が新株式売買システム arrowhead で実践した「手戻り」と「不具合」を無くす開発手法", X-over Development Conference 2010. [3]石井信明, "要件定義フェーズにおけるプロジェクトマネジメントの研究課題に関する考察", 大学紀要.
- [4] Akio Takashima, Hiroshi Maruyama, YaMin Lu and Taichi Nakamura: "A Method for Extracting Test Cases from a Basic Design Document", Proceedings of 9th Joint Conference on Knowledge-based Software Engineering (JCKBSE'10) Kaunas, Lithuania, (2010, in print)
- [5]上原伸介, 他, "ソフトウェア理解支援のための多粒度ソフトウェアマップ", 情報処理学会研究報告. ソフトウェア工学研究会報告 2008.
- [6]森哲夫, 他, "ソフトウェア開発の単体・結合量テストを考慮した最適テスト努力", 日本経営工学会論文誌
- [7]劉功義, 他, "QFD を用いた要求整理手法の適用方法に関する考察", プロジェクトマネジメント学会 2007 年度春季研究発表大会予稿集.
- [8]Taichi Nakamura, Akio Takashima, Akane Mikami: "The use of agents to represent learners in role-play training", IEEE 1st Annual Engineering Education Conference (EDUCON2010), (2010)