

係り受け解析を活用した設計仕様書とテスト仕様書の紐づけ手法 Linking Japanese Design Document and Test Specification with Dependency Analysis

小川 貴史[†] 島川 博光[†]
Takafumi Ogawa Hiromitsu Simakawa

1. はじめに

ソフトウェア開発において、成果物と顧客の要望に乖離が生じる問題がある。この乖離が解消されないことで手戻りが発生し、開発が長期化する。この問題が発生する原因は、ソフトウェアの設計業務にあると考える。設計では開発者が顧客の漠然とした要求をくみ取り実装レベルに具体化する必要がある。ここで開発者が正しく組み取ることができず、顧客の要望とは異なる独自の解釈で設計することが乖離の要因となる。この乖離を解消する最も直接的な方法は、顧客が設計仕様書を修正することである。しかし、これは非現実的である。顧客はソフトウェア開発の専門家ではなく、設計仕様を十分に理解する知識を持ち合わせていないことが多いからである。現在のソフトウェア開発においては、アジャイル開発やラピッドプロトタイピングによって乖離を最小限に留めている。これらの方法では、成果物を小規模かつ短時間で試作することを繰り返す。その中でこまめに方針を修正する。つまり、要求と設計の間の乖離を、成果物を使って解消する。これに対し本研究では、テストケースを使って乖離を解消するテストケース駆動再設計を提唱する。

顧客と開発者が議論してテストケースを修正し、その修正を設計仕様書に反映することで、顧客の要望を満たした設計にすることができると考えている。しかし、このテストケース駆動の再設計における課題として、テストケースの修正を設計仕様書に反映させる際に、修正の影響範囲が分かりづらく見落としや把握漏れの危険性があることが挙げられる。

本研究はこの課題に対し、設計仕様書とテスト仕様書を紐づける手法を提案する。紐づけることで、修正されたテストケースから設計仕様書の修正すべき箇所が分かるようになると思われる。

設計仕様とテストの両方を対象とする事例として、テストケースを自動生成する研究[1-3]がある。Fischbach ら[1]は、受け入れ基準からテストケースを自動で生成する手法と SPECMATE というツールを提案している。自然言語文の中のテストの入力と出力に該当する文節の間に係り受け解析から規則性を見出し、テストケースを作る手法を提案している。Wang ら[3]は、RUCM という記法に則った自然言語で書かれたユースケース仕様からシステムレベルのテストケースを生成する手法を提案している。ユースケース仕様から動作の情報を抽出する。形態素解析を用いて文章をトークン化し、それぞれのトークンに”入力”、”出力”、”条件”などのタグ付けを行うことで実現している。

また、制限言語を扱う研究[4-7]がある。形式言語という既定のフォーマットに則ることで、要求を構造的に解釈することができ、記載のない要素の把握がしやすくなる。

[†]立命館大学 Ritsumeikan University

Gröpler ら[4]は、ドメイン固有言語である IDRL を採用し、自然言語から IDRL を介して UML ステートマシンを生成する手法を提案している。池田ら[5]は SOFL 形式仕様という記法を採用し、入力変数、出力変数、事前条件、事後条件の原子論理式からテストケースを生成する手法を提案している。制約に同じ変数があった場合に失敗する例が示されている。増田ら[6]はデジジョンテーブルテスト技法を採用し、“条件”と“動作”という意味的役割を付与する方法を提案している。青山ら[7]はセミ形式記述という記法を採用し、自然言語から形式言語への変換手法を提案し、形式言語からのテストケースの生成手順を解説している。

本論文では、紐づけのための係り受け解析の活用方法について論じる。形式言語にはそれぞれ適用しやすい場合とにくい場合があり汎用性に疑問がある。このことから、本研究は自然言語からのテストの入力と出力に該当する文節を抽出し紐づけることを目指す。本論文は、自然言語文に対して汎用性の高い係り受け解析を用いる手法を提案する。

2. 係り受け解析の活用と既存技術の検証

2.1 係り受け解析の活用方法

[1]では、係り受け解析を活用し自然言語文からテストケースを自動生成する。概要図を図 1 に示す。自然言語文を係り受け解析し、規則に基づいて「原因」と「結果」となる要素を抽出した Cause-Effect 図[2]を作成する。そして、「原因」を入力、「結果」を出力としたテストケースを作るという手法を提案している。



図 1 SPECMATE の手法概要図[1]

英語とドイツ語で記述された 300~500 弱文程度の受け入れ基準のデータセットを使用し、英語は 38 個、ドイツ語は 43 個の係り受け関係の規則を報告している。

2.2 単語の分散表現化手法を用いた紐づけの検証

既存技術を用いた試行として、Python の自然言語処理ライブラリの spaCy を用いた紐づけを検証した。設計仕様書の文章とテスト仕様書のテストケースをそれぞれベクトルにし、そのベクトルのコサイン類似度を算出する。テスト仕様書のいずれのテストケースともコサイン類似度が 0.95 以下である設計仕様書の文章を紐づけが失敗しているとして、その数を測った。その結果、設計仕様書の 775 文中 507 文と約 65% が紐づけに失敗した。失敗した要因は、文章をそのままベクトル化したことにより、設計仕様書の文中のテストとは関係のない部分がノイズとなってしまったことで、適切なベクトルにできなかったことにあると考える。

このことから、ノイズをなつた部分を省略することで、より精度の高いアルゴリズムになると考えられる。

3. 係り受け解析を用いた設計とテストの紐づけ

3.1 設計仕様書とテスト仕様書の対応性

一般的に、設計仕様とテスト仕様には同様の内容が記述される。設計を検証するためにテストをするからである。テストケースには、対応した検証する設計仕様がある。

本研究ではこの構造的な特性に着目した。要求と設計の間の乖離を解消するために、設計の代わりに設計に対応しているテストを使えるのではないかと考えた。これをテストケース駆動再設計と呼ぶ。図 2 にその概要を示す。

本研究は、設計仕様書の自然言語文からテストの入力と出力に該当する文節の対を抽出する手法と、テストケースと紐づける手法を提案する。

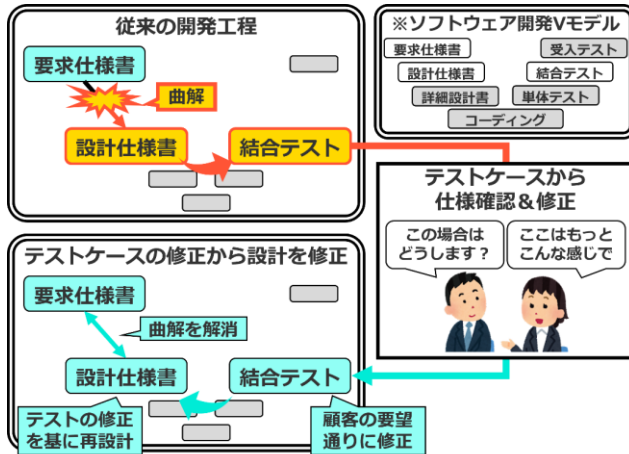


図 2 テストケース駆動の再設計

3.2 係り受け関係規則の作成

本研究では[1]の手法を参考に、設計仕様の自然言語文の係り受け関係の規則性を定式化する。本節では、係り受け関係の規則性を定式化する手法を提案する。

先の検証で用いた例を用いて説明する。設計仕様書の文章 775 文について、入力と出力に該当する部分を手作業でラベル付けする。ここで文書の作成者に確認してもらい、ラベル付けの妥当性を担保する。次に、この 775 文の文章データを用いて、文章を入力としテストの入力と出力に該当する文節を判定するニューラルネットワークを作成する。このニューラルネットワークモデルは、入力と出力の間にある規則性をカーネルとして判定していると考えられる。このカーネルからモデルが判定している仕組みを分析し、仕組みを定式化することで規則を作る。

3.3 設計仕様書とテスト仕様書の紐づけ

本節では、紐づけの手法を提案する。紐づけはテストケースと設計仕様の文から抽出した入力と出力の対の類似度を確率に回帰するモデルを作成することによって実現する。紐づけ手法の概要を図 3 に示す。

手順としては、まず、テスト仕様書から入力列と出力列から入出力の対を取得する。次に、設計仕様書の全文から

係り受け関係の規則に基づき入力と出力の対を抽出する。そして、テスト仕様書の入出力と設計仕様書の入出力から類似確率を算出する。類似確率が 0.5 よりも大きい数値であるものを紐づいているものとして扱う。

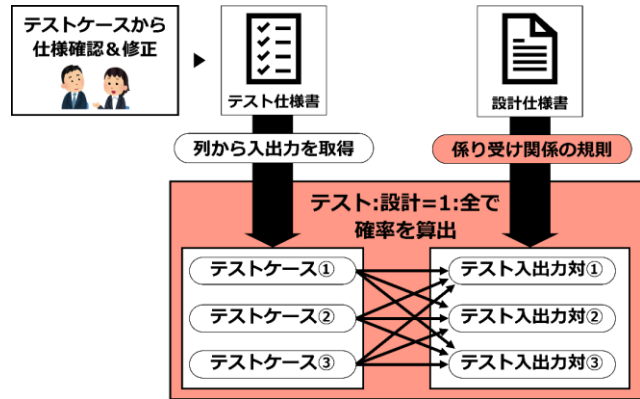


図 3 紐づけ手法

4. おわりに

本研究は、ソフトウェア開発における成果物と顧客の要望に乖離が生じる問題に対し、テストケースを使って乖離を解消するテストケース駆動の再設計を提唱した。加えて、テストケース駆動再設計の中で課題となる設計の修正業務において修正の影響範囲を明確にするために、設計仕様書とテスト仕様書を紐づける手法を提案した。設計仕様の自然言語文の中のテストの入力と出力に該当する文節の関係性を係り受け解析を活用して定式化する手法を提案した。今後は係り受け関係の規則の作成し、紐づけの精度を検証する。

参考文献

- [1] J. Fischbach, A. Vogelsang, D. Spies, A. Wehrle, M. Junker and D. Freudenstein, "SPECMATE: Automated Creation of Test Cases from Acceptance Criteria," 2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST), Porto, Portugal, 2020, pp. 321-331, doi: 10.1109/ICST46399.2020.00040.
- [2] J. Fischbach, M. Junker, A. Vogelsang and D. Freudenstein, "Automated Generation of Test Models from Semi-Structured Requirements," 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), Jeju, Korea (South), 2019, pp. 263-269, doi: 10.1109/REW.2019.00053.
- [3] C. Wang, F. Pastore, A. Goknil and L. C. Briand, "Automatic Generation of Acceptance Test Cases From Use Case Specifications: An NLP-Based Approach," in IEEE Transactions on Software Engineering, vol. 48, no. 2, pp. 585-616, 1 Feb. 2022, doi: 10.1109/TSE.2020.2998503.
- [4] Gröpler, R., Sudhi, V., García, E. J. C., & Bergmann, A., "NLP-Based Requirements Formalization for Automatic Test Case Generation", CS&P, Vol. 21, 2021, pp. 18-30.
- [5] 池田 逸人, 劉 少英, "形式仕様に基づくテストケースの自動生成とテスト結果の自動評価," 情報処理学会第 79 回全国大会, 2017, pp. 365-366.
- [6] 増田 聡, 松尾谷 徹, 津田 和彦, テストケース作成自動化のための意味役割付与方法, コンピュータ ソフトウェア, 2017, 34 巻, 2 号, p. 16-27,
- [7] 青山 裕介, 黒岩 文瑠, 久代 紀之, "テストケース生成のためのシステム仕様書の論理記述変換アルゴリズム", 情報処理学会論文誌, vol.61, no.3, 2020, pp.521-534.