6A-3

自然言語処理を利用したユースケース記述推敲支援

山梨 敦志† 松浦 佐江子‡
芝浦工業大学システム工学部電子情報システム学科†‡

1 はじめに

近年，オブジェクト指向分析設計の手法としてユースケース（use case）が使用されるようになった．ユースケースとは，システムが果たすべき機能をユーザレベルで記述したものとして，システムの外部仕様を表現するのに使用される．

ユースケースの内容を具体的に記述したものがユースケース記述となる．ユースケース記述ではアクターとシステムのやり取りを 1 ステップずつ記述する．チームによるユースケース設計における最大の障害は，整合性の欠如である．これは，アイデアや手法が，プロジェクトが進行するに従って進化しているからである．また，人々は，単に異なるアイデアを持っているだけでなく，その表現方法も異なる．[1]

ユースケース記述の整合性を欠いた場合，ユースケースを元にクラスを抽出する際に，同様の機能を持った違うメソッドが作成される等，システムの品質への影響が大きい．本稿では，ユースケース記述の整合性を「書式の統一」,「単語の統一」,「ユースケース分割」,「実装に依存しない単語の使用」とし，これらの不整合を減少させることを目的としたユースケース記述推敲支援を提案する．

2 ユースケース記述の整合性を欠く原因

2.1 書式の不統一

ユースケース記述には決まったスタイルが無いため，設計者のスキルによって表現方法が変わる．ユースケース記述は短く明確に，読みやすいように記述しなければならないため，動詞は受動態ではなく能動態を使用するべきである．[2]

2.2 単語の不統一

同じ単語が異なる概念を表現し，異なる単語で違う概念を表現する．この場合，クラスの抽出を正しく行うことが出来ない．

2.3 実装に依存した単語の使用

ユーザインターフェース設計に対するわずかな変更でも，「要求」自体を変更しなければならなくなるため，要求が不安定になる．[2]

2.4 ステップの重複

重複しているステップに誤りがあった場合には，訂正するための作業量が増加してしまう．また，冗長性が発生する．

Writing Tool to Developing Use Case Documents using Natural Language Processing
†Atsushi Yamanashi ‡Saeko Matsuura
†‡Shibaura institute of technology
Department of electronic information system

3 整合性を欠く原因に対する解決方法の提案

3.1 形態素解析を利用した書式と単語の統一

ユースケース記述は一般的に自然言語で書かれているため，分析には自然言語処理の技術である形態素解析を用いる．[3]それによって，ユースケース記述の単語（名詞，動詞）の抽出を行い，どのユースケースで使用されているか確認できる単語一覧表を作成する．単語一覧表では，単語，使用しているユースケース数，使用しているユースケース名を表す．

3.2 要求仕様書に基づく単語の選別

要求仕様書を分析し，ユースケース記述との整合性を検証する．要求仕様書では使用されていないが，ユースケース記述では使用されている単語を抽出する．これを元に，実装に依存している単語を見つけ出し訂正を行う．

3.3 ステップの比較によるユースケース分割

単語のみを抽出したユースケース記述を元にステップ毎の比較を行う．同様の記述がされているステップが多い場合には，ステップを分割することができる可能性がある．

同様の記述とは使用されている単語が完全一致した場合である．

3.4 推敲の流れ

図 1 に示すように要求仕様書，依頼者の定義した要求文からそれぞれユースケース記述，要求文を抽出する．それらから抽出した単語一覧表を元に推敲を行う．単語一覧表による推敲が終了したらステップの重複検出を行う．

4 推敲の実践

4.1 使用する要求仕様書

情報実験Ⅱにおいて，あるグループが作成した要求仕様書を用いる．情報実験Ⅱは，本学科 3 年を対象とした授業であり，グループワークによるソフトウェア開発を行っている．このグループは，9 人で構成されていて自動販売機シミュレータの開発を担当している．このソフトウェアでは，ロジック部分とシミュレータと呼ばれるユーザインターフェース部分の区切りを明確にし，ロジック，シミュレータ間をオブジェクトのやり取りで表すものである．また，要求仕様書の段階ではユースケース数 58，総ステップ数 161 である．総ステップ数とは，ユースケース全体のステップ数の和である．

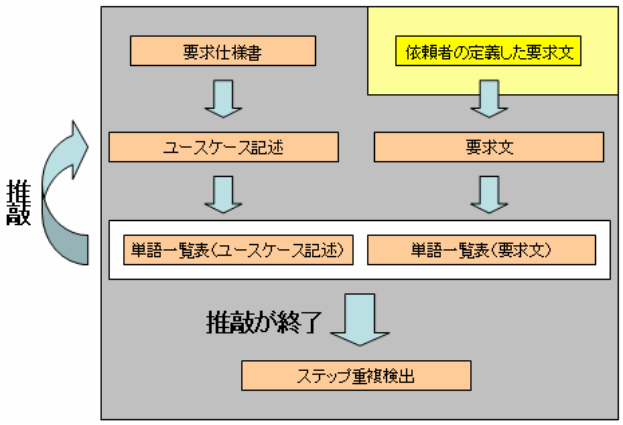


図 1　推敲の流れ

4.2　書式と単語の統一

単語一覧表を元に，使用されているユースケース数が多い単語から確認を行う．

受動態で記述されている単語を能動態に直す．この作業を行った結果,「読み出される」を含む 8 種類の受動態を確認することが出来た．

単語一つ一つをユースケース記述と照らし合わせ，その単語の使用用途をその単語が使用されている前後のステップを元に単語一つ一つの使用用途を記述する．例として,「シミュレータが内部キーボードで次のコマンドを打つ.」では,「打つ」は「入力する」という意味で使用されていることが分かる．この際に，異なる単語で同じ概念を表現していた場合には，重複している単語を記述しておき，使用用途が同じ単語の中で何に統一するかを決める．統一の基準は，その単語を使用しているユースケース数を元に行うが，最終的にはユーザ自身の判断に委ねることになる．この作業を行った結果,「情報を送る」という意味で使われていた動詞が一番多く検出された. 例として,「渡す」,「送る」,「通知する」,「送信する」等である．この場合にはユースケース数が一番多かった「渡す」に統一した．

同じ単語で異なった概念を表現していた場合には，その概念を使用している単語がある場合は，その単語に置き換える．また，使用している単語が無い場合には新たな単語を用いて表す．この作業を行った結果,「保存する」という動詞が,「物を保存する」という意味と「物を入れる」という意味で使用されていたので，後者の意味で使用されていた動詞を「入れる」に訂正した．

「～し，～する」の様な書式で文を区切っている場合には, 理解が困難になる可能性があるので, ステップを 2 つに分割する．この作業を行った結果,「アクセスし，～する」等の書式を検出することが出来たため,「アクセスする．～する.」の様にステップを分割した．

単語の変更（統一，訂正）は，全ての単語の確認が終わった時点で行う．推敲前と推敲後のユースケース記述の状態を表 1 に示す.

表 1　推敲前と推敲後の動詞の比較

	推敲前	推敲後
総単語数	87	67
重複単語数		20
訂正単語数		4
総ステップ数	161	173
分割ステップ数		12

4.3　単語の選別

単語の統一を行ったユースケース記述と要求文から抽出した名詞を比較し，要求文では使用されていない名詞を抽出する．その名詞一覧から実装に依存している名詞の有無を確認する．この作業を行った結果, 要求文で使用されている名詞は 29, 使用されていない名詞は 166 であった.

4.4　ユースケース分割

書式と単語の統一後に名詞と動詞のみを抽出したユースケース記述をステップ毎に比較を行う．重複したステップの一部を表 2 に示す．また，統一前のステップ重複は 9 種類だったが，統一後では 13 種類となった.

表 2　重複したステップ

ステップの内容	重複数
シミュレータ,内部キーボード,次のコマンド,打つ	8
キーボード,区切られた各入力,CPU,渡す	8
データベース,データ,読み出す	4
シミュレータ,データベース,アクセスする	4

5　まとめ

書式と単語の統一では, 動詞の総単語数が 20 減少しており，単語を統一する助けになったことが分かる．また,ステップを分割することにより，第三者がユースケース記述を理解する上での助けになったのではないだろうか.

単語の選別では，使用されていない名詞の数を出すことは出来たが，それを元にどのようにすれば単語の選別が行えるか明確な手段を考える必要がある.

ユースケース分割では，重複数の多いステップを検出することが出来た．これにより，このステップが使用されているユースケース記述を分割することが出来る可能性があることを示した．また，単語の統一後の方がより多くのステップ重複を検出できたため，単語の統一を行うことが重複検出の助けになることが分かった.

6　参考資料

[1]ダリル・クラク イーモン・ギニー,ユースケース導入ガイド, 株式会社ピアソン・エデュケーション
[2]アリスター・コーバーン,ユースケース実践ガイド, ウルシステムズ株式会社
[3]茶筅, http://chasen.aist-nara.ac.jp/