

## 要求仕様の誤解釈を検出する Domain Word Modeling の提案

柏原 一雄  
株式会社デンソークリエイト  
kazuo.kashiwabara.j3a@jpgr.denso.com

不破 慎之介  
株式会社デンソークリエイト  
shinnosuke.fuwa.j8h@jpgr.denso.com

石川 冬樹  
国立情報学研究所  
f-ishikawa@nii.ac.jp

長井 亘  
テックスエンジニアリング株式会社  
nagai.wataru.d1@tex-sol.com

林 香織  
株式会社デンソークリエイト  
kaori.hayashi.j4e@jpgr.denso.com

栗田 太郎  
ソニー株式会社  
taro.kurita@sony.com

### 要旨

仕様が自然言語のような曖昧さを含む記法で記述されている場合、仕様の誤解釈を誘発しやすい。実際に我々の組織では、「抽象的に表現された用語」を「不足している前提知識」で解釈することにより、誤解釈が発生していた。誤解釈の可能性のある用語を検出するために、用語の解釈を可視化する手法「Domain Word Modeling」を考案した。「Domain Word Modeling」は、導入のしやすさを重視した手法である。実験で、実開発において定義された要求仕様を入力として、考案手法を実行し、効果を確認した。

### 1. はじめに

ソフトウェア開発において、仕様は関係者間の情報ハブ<sup>[1]</sup>となり、仕様策定者は多数の関係者と仕様によりコミュニケーションをする。仕様が自然言語のような曖昧さを含む記法では他の人にわかるように表現することは難しく、用語の曖昧さや関連の曖昧さが、読み手の解釈を必要とし、異なる理解による問題を生じさせることが、文献[1]で述べられている。

実際に我々の組織では、要求仕様の誤解釈による手戻りが発生していた。誤解釈は、「抽象的に表現された用語」を「不足している前提知識」で解釈することにより発生していた。要求仕様のレビューは実施していても、このような用語の誤解釈を防止しきることはできていなかった。

本研究では、要求仕様の誤解釈による手戻りを減らすことを目的とし、誤解釈を誘発する可能性のある用語を検出するための手法を考案した。考案手法は、導入障壁をできる限り低くし、コストやスケジュールの制約が厳しいプロジェクトでも効果を得やすくした。

考案手法は、「Domain Word Modeling」と呼び、モデルの記述ルールとモデリングの手順を技術要素とする。考案手法で作成する Domain Word Model は、要求仕様と前提知識を入力として、用語の関連を木構造で表現する。また、手法の導入障壁をできる限り低くするため、先行研究の「自然言語による仕様記述の改善のアプローチ<sup>[2]</sup>」と「マインドマップモデリング<sup>[3]</sup>」を参考としている。

実験では、実開発において自然言語で記述された要求仕様を入力として、考案手法を実行し、考案手法が以下の要求を満たすかを評価した。

- ・ レビューで見逃した「誤解釈を誘発する用語」を検出できる
- ・ 要求仕様を自然言語で記述しているプロジェクトに対して導入しやすい

実験の結果、誤解釈を誘発する「抽象的に表現されている曖昧な用語」と「解釈のための前提知識が不足している用語」を検出できることを確認した。また、実開発で使用されている要求仕様の形式を変更せずに手法が実行できること、Domain Word Modeling 未経験でも手法の説明を受けただけで実行できることを確認した。

実開発において、Domain Word Modeling を活用することで、要求仕様の誤解釈による手戻りを減らす効果が期待できる。

これ以降の本稿の構成は次のとおりである。2 章で現

状分析の結果と研究の課題を示す。3章では、課題解決の参考とした先行研究を示す。4章では、考案した解決策を提案する。5章では、提案手法の評価結果と考察を示す。6章では、まとめと今後の進め方を示す。

## 2. 課題設定

### 2.1. 現状分析

#### (1) 要求仕様の誤解釈の要因

我々の組織では、要求仕様のレビューを実施しているも、要求仕様の誤解釈が防止しきれず、手戻りが発生していた。要求仕様を誤解釈した事例を分析すると、「抽象的に表現された用語」を「不足している前提知識」で解釈していることにより発生していた。

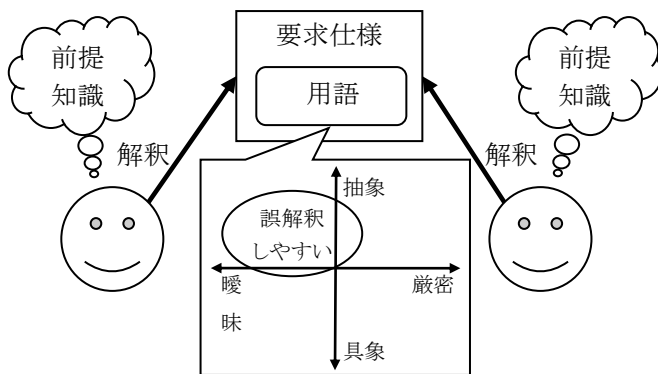


図 1 誤解釈のメカニズム

特に、派生開発<sup>[4]</sup>では、「抽象的に表現された用語」を「不足している前提知識」で解釈する状況になりやすく、要求仕様の誤解釈が発生しやすいと考えられる。

文献[4]では、「部分理解」の下で作業しなければならないことが、派生開発の特徴として示されている。派生開発においては、担当者がベースソフトを部分的にしかり理解できていない状態、つまり前提知識が不足している状態で、開発する状況になりやすい。

また、文献[5]では、派生開発において定義される変更要求仕様では、ベースソフトから変化しない既存仕様が抽象的に表現されやすく、テストケースの漏れを誘発しているという調査結果が示されている。

#### (2) 誤解釈を誘発する用語

我々の組織で、要求仕様の誤解釈を引き起こした事例を調査した。その結果、抽象的に表現された用語が誤解釈を誘発していることがわかった。また、調査により抽出された用語は、単に抽象的というだけでなく、他の用語と関連がある用語であった。つまり、単独の用語ではなく用語間の関連が把握できていなかった。

実際に誤解釈を誘発した用語の具体例を表 1に示す。表 1の用語を分類すると、次の 3 つの種類に分けられる。どれも抽象的に表現された用語であり、種類・構成要素・対象を示す用語との関連がある。

- a. 複数の種類がある概念
- b. 複数の構成要素からなる概念
- c. 対象によって具体的な行為が変わる概念

表 1 誤解釈を誘発した用語の例

用語	説明	分類
ユーザ	X ユーザ, Y ユーザなど、複数の種類がある	a
支店	X 支店, Y 支店など、複数の種類がある	a
英語	米国英語, 英国英語, 豪州英語など、複数の種類がある	a
原点	左上原点, 右下原点など、複数の種類がある	a
改行	CR+LF, LF, CR など、複数の種類がある	a
診断情報	DTC, Freeze frame data など、複数の構成要素から成る	b
記憶	対象データや対象製品によって記憶場所が変わる	c
開始	対象機能によって開始タイミングが変わる	c
中止	対象機能によって中止時に実施する処理が変わる	c

調査対象には、業務システム・情報システムの要求仕様、組込みソフトのミドルウェアの要求仕様など、様々な種類が含まれている。

#### (3) 要求仕様のレビュー

要求仕様のレビューで、「抽象的に表現された用語」を問題として指摘することは難しい。また、要求仕様の読み手にとって「不足している前提知識」を明らかにするこ

とも難しい。

レビューは、レビュー対象の成果物から、問題を検出する行為である。レビューで、人によって問題か否かの判断が分かれることを指摘することは難しい。また、成果物に表現されていないことを指摘することも難しい。

厳密な仕様<sup>[1]</sup>を記述するときには、記述する仕様の抽象度と、記述の方法を決定する必要がある。最適な抽象度は、開発の目的や事情によって異なる。したがって、レビューで、抽象的に表現された用語を、一律に問題として指摘することはできない。

レビューでは、レビュー対象の成果物である要求仕様は確認されても、要求仕様を読むうえで必要となる前提知識は可視化されておらず、確認されにくい。したがって、レビューで、「不足している前提知識」を明らかにすることは難しい。小林は、文献[6]で、レビューで確認する情報の背景・前提を可視化する必要性を述べている。

## 2.2. 課題提起

本研究では、要求仕様の誤解釈による手戻りを減らすことを目的とし、誤解釈を誘発する可能性のある用語を検出するための手法を考案することを課題とする。

考案手法は、導入障壁をできる限り低くし、コストやスケジュールの制約が厳しいプロジェクトでも効果を得やすくする。考案する手法は、以下の 2 つの要求を満たすものとする。

- ・ レビューで見逃した「誤解釈を誘発する用語」を検出できる  
レビューで防止できなかった誤解釈は、「抽象的に表現された用語」を「不足している前提知識」で解釈することにより発生していた。誤解釈を誘発する「抽象的に表現されている曖昧な用語」と「解釈のための前提知識が不足している用語」を検出できる手法とする。

- ・ 要求仕様を自然言語で記述しているプロジェクトに対して導入しやすい

プロジェクトに新たな手法を導入しやすくするためには、「プロセスを変更するコスト」「手法を習得するコスト」「必要となるツールを用意するコスト」を下げる必要がある。「プロセスを変更するコスト」を下げるため、要求仕様の形式を変更せずに実行できる手法とする。要求仕様の形式を変更しなければ、要求仕様でやりとりをする関係者のコミュニケーションの仕方を変える必要がなくなる。「手法を習得するコスト」を下げるため、考案手法を未経験の状態からでも、手法の説明を受けただけで実行でき

る手法とする。「必要となるツールを用意するコスト」を下げるため、フリーソフトウェアで実行できる手法とする。

文献[7]で、あいまい表現が類型化されている。定義されている類型の一つに「用語のあいまいさ」がある。本研究では、この「用語のあいまいさ」が誘発する誤解釈による手戻りを減らすことを目的としている。

## 3. 先行研究

### 3.1. 自然言語による仕様記述の改善のアプローチ

大森らは、文献[2]で、自然言語による仕様記述の品質向上を目的として、関係者の共通理解を得やすい自然言語記述と、厳密な検証に基づく安定したモデル化を可能とする形式モデルのそれぞれの長所を両立させるアプローチを提案している。図 2のように、自然言語記述から形式モデルへ変換し、その形式モデル上で検証を行い、検証結果のフィードバックによる自然言語記述の改善を繰り返し行うというアプローチである。

要求仕様を自然言語で記述している開発プロセスに対して、要求仕様の形式を変更する必要がなく、導入しやすいアプローチであると考えられる。

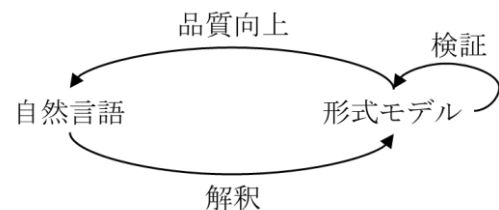


図 2 要求の洗練プロセス

### 3.2. シソーラス

用語の関連を表現するものとしてシソーラスがある。

加藤らは、文献[8]で、シソーラスは、要求分析者のドメイン知識を補完し、要求の欠落を防止するために有用であることを示している。また、文献[9]で、日本語で書かれた技術文書を入力に、形態素解析を活用し、シソーラスを構築する手法を提案している。

小川らは、文献[10]で、シソーラスを作成する行為は、関係者が相互の理解の相違を確認する手段となることを示している。

JIS X 0901:1991<sup>[11]</sup>では、シソーラスにおける概念間関

係が区別され、関係を表現するための略語が定義されている。以下に、シソーラスの記法の例を示す。

- ・ 種類関係
  - BTG 種類関係の上位語
  - NTG 種類関係の下位語
  - 例：
    - サル BTG 霊長類
    - 霊長類 NTG サル
- ・ 全体部分関係
  - BTP 全体部分関係の上位語
  - NTP 全体部分関係の下位語
  - 例：
    - 中枢神経系 BTP 神経系
    - 神経系 NTP 中枢神経系

シソーラス及びその作成行為は、前提知識の不足を補う目的と前提知識の相違を明らかにする目的のために有用であると考えられる。ただし、シソーラスを活用するためには、シソーラスの記法を学習する工数や記法からの逸脱を指摘・修正する工数が必要となる。

### 3.3. マインドマップモデリング

文献[3]で説明されているマインドマップモデリングは、マインドマップを使って、自然言語からモデルを作成する手法である。自然言語から、UML によるモデルを直接作成するのは難しいという考えから考案された手法である。マインドマップモデリングは、以下の手順で実施する。

1. 文章の中から名詞を抽出する
2. 文章の中から動詞を抽出する
3. 名詞の構造 (is-a 関係, has-a 関係) を定める

### 4. SVO(名詞-動詞-名詞)で考える

マインドマップモデリングは、木構造で用語をつなげるだけのシンプルな記法を使用している。そのため、記法を学習するコストはシソーラス等と比べると低いと考えられる。また、マインドマップを作成するためのツールがあれば利用可能な手法であり、ツールにはフリーソフトウェアも存在する。

## 4. 解決策の提案

### 4.1. 課題の解決方針

考案手法は、モデリング実施者とモデルのレビューアの間で解釈の相違を検出するために、用語を解釈するための前提知識を、モデルで可視化し、レビューする。モデルの作成とモデルのレビューを通して、「抽象的に表現されている曖昧な用語」と「解釈のための前提知識が不足している用語」を検出する。モデルは、要求仕様とそれを解釈するための前提知識を入力として、用語の関連を表現する。用語を解釈するための前提知識も、モデリングにより可視化されることで、確認が可能となる。

考案手法は、要求仕様の形式を変更せず実行できる手法にするために、先行研究の「自然言語による仕様記述の改善のアプローチ」を参考とした。要求仕様の検証をするために、要求仕様のレビューに加えて、用語の関連を表現するモデルの作成と作成したモデルのレビューをする。仕様策定者と「プロジェクトの外部」「プロジェクトの次工程」の担当者とのコミュニケーションは、手法導入前と変わらず、自然言語で記述された要求仕様を使用して行う。

考案手法は、作成するモデルをできる限りシンプルな

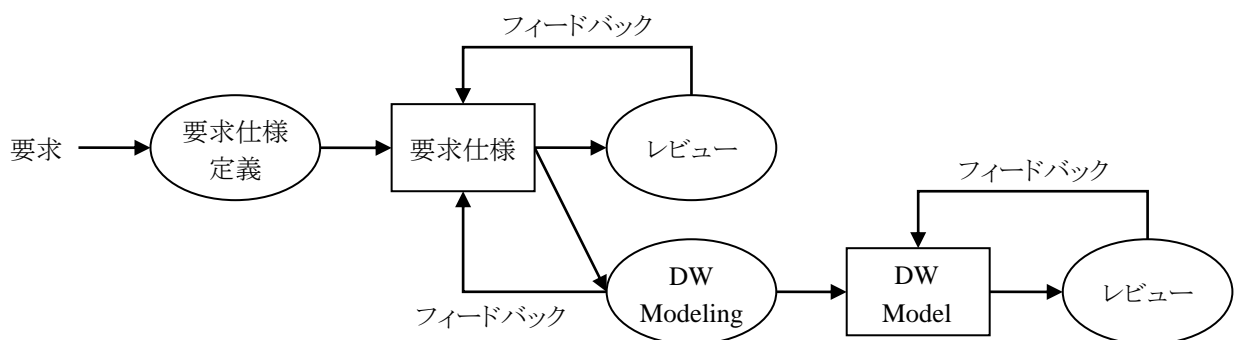


図 3 DW Modeling の概要

記法で表現するため、「マインドマップモデリング」を参考とした。マインドマップモデリングで作成されるモデルは、概念と概念間のつながりを木構造で示すだけのシンプルなものであるが、is-a 関係と has-a 関係を表現し、「抽象的に表現されている用語」を特定することが可能である。また、マインドマップはフリーソフトウェアで作成することが可能である。マインドマップをベースとした手法とすることで、手法を習得のためのコスト、手法の実行に必要なとなるツールを用意するコストの両方が小さくなる。

考案手法は、Domain Word Modeling(DW Modeling)と呼び、手法により作成された用語の関連を表現したモデルを Domain Word Model(DW Model)と呼ぶ。図 3 に、DW Modeling の概要を示す。考案手法は、DW Model の記述ルールと DW Modeling の手順を技術要素とする。

図 4 に、DW Model のイメージを示す。これは、テスト設計コンテスト'14 の ASTER 自動販売機ユースケース仕様書に登場する用語を入力に作成したモデルの一部である。

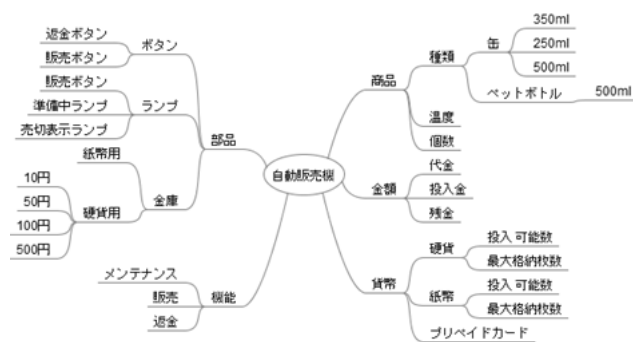


図 4 DW Model のイメージ

#### 4.2. ドメイン用語の定義

DW Modeling では、開発対象のシステムを定義するために必要となる用語を、ドメイン用語 (Domain Word) と呼ぶ。ドメイン用語は、サ変名詞を含む名詞を対象とする。対象によって具体的な行為が変わる概念もモデルに表現するため、行為を示すサ変名詞もドメイン用語とする。

#### 4.3. Domain Word Model の記述ルール

DW Model の記述ルールを以下に定義する。

- 同義語が存在しない状態にする。同義語は 1 種類

のみ登場させる。

- 各階層には、集約 (A has-a B) と汎化 (A is-a B) の関係を混在させない。
- 必要に応じて要求仕様で使用されていない用語をモデルに登場させてよい。
- 末端の枝にのみ、名詞ではなく、文・式・値等を示してもよい。

#### 4.4. Domain Word Modeling の手順

DW Modeling は、「1.形態素解析」、「2.モデリング」、「3.モデルのレビュー」の 3 ステップで、要求仕様で使用されている用語の解釈を可視化する。形態素解析では、ドメイン用語を漏れなく効率的に抽出する。モデリングでは、抽象的な用語に対して下位概念となる用語を抽出し関連付け、用語の解釈を可視化する。モデルのレビューでは、モデリング実施者とモデルのレビューアの間で解釈の相違を明らかにする。

DW Modeling の手順を以下に定義する。

##### 1. 形態素解析

要求仕様を入力に、形態素解析を行い、要求仕様で使用されている単語を抽出する。複合語として正しく抽出できていない単語や未知語と判断される単語がある場合には、ドメイン用語と判断すべき単語を形態素解析で使用する辞書に追加し、形態素解析を再実施する。

##### 2. モデリング

形態素解析で抽出した単語から、要求仕様で使用されているドメイン用語を特定する。DW Model の記述ルールに従い、特定したドメイン用語を木構造で関連付ける。抽象的な用語に対して下位概念となる用語を抽出し関連付けをするために、必要に応じて入力としている要求仕様以外の技術文書等も参照する。

##### 3. モデルのレビュー

モデリング実施者以外で、用語を解釈するための前提知識をもっている人をレビューアとする。レビューアでは、DW Model の記述ルールを満たしているかを確認し、更に木構造で表現されたモデルの各階層が MECE な状態であるかを確認する。モデルの確認によって、モデリング実施者とモデルのレビューアの間で解釈の相違があった用語を検出する。レビューアで、不足している前提知識が判明した場合、入力とした要求仕様以外の技術文書等から必要な情報を得て、モデリングを再実施する。

## 5. 解決策の評価

### 5.1. 評価方法

考案手法が以下の 4 つの条件を満たすか確認し、考案手法の有効性と導入容易性を評価する。

- A) 「抽象的に表現されている曖昧な用語」を検出できる  
抽象的に表現されていることが要因で誤解釈を誘発する可能性のある曖昧な用語を、モデリング実施者が検出できたかを確認する。
- B) 「解釈のための前提知識が不足している用語」を検出できる  
モデリング実施者の前提知識の不足が要因で誤解釈を誘発する可能性のある用語を、モデリング実施者またはモデルのレビューが検出できたかを確認する。
- C) 要求仕様の形式に依存せず実行できる  
複数の異なる形式の要求仕様を入力に DW Modeling を実行し、DW Model が作成できることを確認する。
- D) DW Modeling が未経験の状態からでも実行できる  
DW Modeling が未経験の状態から、手法の説明を受けるだけで、DW Model が作成できることを確認する。更に、モデリング中に、手法に関する質問が発生しないことを確認する。

評価のために表 2 に示すデータを計測する。

評価のための実験は、以下の条件下で実施している。

- 入力とする要求仕様は、実開発で使用されている文書で定義されているものとする。
- 入力とする要求仕様は、レビューが完了している状態とする。
- モデリング実施者は、DW Modeling 未経験者とする。
- モデリング実施者は、DW Model の記述ルールと DW Modeling の説明を受ける。
- モデリング実施者は、要求仕様に記載されている用語を知っている。
- 形態素解析は、KH Coder<sup>[12]</sup>を使用して実施する。

る。

- モデリングは、マインドマップ作成が可能なフリーソフトウェアを使用して実施する。

### 5.2. 評価結果

実験の結果を表 3 に示す。被験者:1~4 は同じ要求仕様を入力とした。被験者:1 は自分自身が書き手である要求仕様を入力とした。被験者:1 以外は他者が書き手である要求仕様を入力とした。

実験の結果から、DW Modeling が、評価方法に示した A) から D) の 4 つの条件を全て満たすことが確認できた。確認結果を以下に示す。

- 全ての被験者が、「抽象的に表現されている曖昧な用語」と「解釈のための前提知識が不足している用語」のどちらかを検出できた。「解釈のための前提知識が不足している用語」の検出数が少ない場合、つまりモデリング実施者が要求仕様の理解度が高い場合、「抽象的に表現されている曖昧な用語」を検出できる傾向があった。また、一部の誤解釈を誘発する可能性のある用語は、モデルのレビューの前に、モデリングをしながら検出された。
- 形式が異なる複数の要求仕様を入力として、DW Model を作成できた。要求仕様:A は、派生開発で作成された変更要求仕様書であり、仕様の変化点のみが記述されている。モデルを MECE な状態にするため、DW Model に、入力とした文書にない用語が多く追加された。
- DW Modeling が未経験の状態でも、一通りの説明を受けるだけで、全ての被験者が DW Model を作成できた。モデリング中に、被験者から手法に対する質問もなかった。

また、複数のモデリング実施者から、以下の所感が得られた。

- 変更要求仕様書<sup>[4]</sup>に定義されている一件の変更要求仕様に対しても実行可能である。
- 知らない用語については、モデルに用語の関連を表現することができない。
- 入力とする文書に同義語が多いと、同義語の判定をする手間がかかり、モデリングの効率が落ちる。

表 2 計測データの種類

データ		説明	
用語数	要求仕様	要求仕様に登場する名詞の数. KH Coder で抽出された名詞の数. 出現回数ではない.	
	DW Model	要求仕様にあり	DW Model に登場する用語の中で, 要求仕様に存在している用語の数.
		要求仕様になし	DW Model に登場する用語の中で, 要求仕様に存在していない, モデリング実施者が追加した用語の数.
誤解釈の可能性のある用語数		要求仕様から検出した, 誤解釈を誘発する可能性のある用語の数.	
	表現の要因	抽象的に表現されている曖昧な用語の数.	
	知識の要因	解釈のための前提知識が不足している用語の数.	
手法への質問の数		モデリング中に発生した, 手法に対する質問の数.	

表 3 実験の結果

被験者 ID	要求仕様 ID	立場	用語数			誤解釈の可能性のある用語数		手法への質問の数
			要求仕様	DW Model		表現の要因	知識の要因	
				要求仕様にあり	要求仕様になし			
1	A	書き手	30	13	38	2	1	0
2	A	読み手	30	33	11	0	4	0
3	A	読み手	30	26	15	1	2	0
4	A	読み手	30	26	10	1	3	0
5	B	読み手	61	35	3	4	1	0
6	C	読み手	23	17	5	2	1	0

### 5.3. 結果の考察

DW Modeling は, レビューでは見逃した「誤解釈を誘発する用語」を検出できる有効な手法である. モデリング実施者が厳密に解釈できていない用語はモデルに表現することも難しいため, 自分自身で誤解釈の可能性に気づくことができる. モデリングを通して, 「わかったつもり」<sup>[13]</sup>の状態に気づくことができる. また, 用語の解釈が可視化されることで, 関係者間で解釈の比較することが可能となり, 誤解釈する可能性のある用語を検出できる.

DW Modeling は, 要求仕様を自然言語で記述している開発プロセスに対して導入しやすい手軽な手法である. 要求仕様の形式を変更しなくても実行できる手法であるため, 要求仕様を情報ハブとしたコミュニケーションの仕

方を変える必要もない. また, 木構造で用語をつなげるだけのシンプルな記法であり, 覚えるルールが少ないため, DW Modeling が未経験の状態でも, 一通りの説明を受けるだけで実行できる.

DW Modeling の有効性と効率性は, 「モデリング実施者とモデルのレビューアがもっている前提知識」と「入力文書の質」に依存すると考えられる. 本稿の評価では, モデリング実施者とモデルのレビューアがもっている前提知識の違いにより, 「誤解釈を誘発する用語」の検出結果がどのように変化するかは確認できていない. また, 入力文書の質の違いにより, 考案手法の効率がどのように変化するかは確認できていない. 今後, 考案手法により効果を得るための前提条件を明確にすることが必要となる. また, DW Modeling は要求仕様のレビューに加えて追加で実行する手法であり, 導入障壁を下げるために, 更に

実行する工数を小さくする対策も必要となる。

DW Modeling を実行する工数を小さくし、更に手法を導入しやすくするためには、モデリング実施者の所感から、以下の3つの対策が有効であると考えられる。

- ・ 小さく繰り返す

DW Modeling は、要求仕様書の全体ではなく、一部の記述に対しても、実行可能である。開発期間が限られている場合でも、DW Modeling の対象とする記述を絞り込むことで、現実的に投入可能な工数で実行可能となる。更に、作成されたドメイン用語のリストと DW Model は再利用可能であり、手法を繰り返し実行する度に、必要となる工数は低減される。

- ・ 書き手が実行する

DW Modeling は、要求仕様の書き手が実行しても、自分自身で定義した要求仕様に対して「誤解釈を誘発する用語」を検出することが可能である。書き手が自分自身で、「わかったつもり」の状態に気づくことができる。要求仕様の書き手であれば、解釈が難しい用語は少ないため、短時間でモデリングができる。効率的に仕様記述の改善をするためには、DW Modeling を要求仕様の書き手が実行したほうがよい。

- ・ 入力文書の質を高めたうえで実行する

要求仕様書内に、対象とするシステムを定義するためには不要である用語が多く存在する場合に、ドメイン用語を特定する作業の効率は悪くなる。例えば、同義語、誤記が多い文書、例が示されている箇所が特定しにくい文書、主語が異なる記述(例:サーバーとクライアント、システムとサブシステム)が混在している文書、仕様と設計の位置づけの記述が混在している文書などである。DW Modeling をする前に、入力とする要求仕様の記述をモデリングがしやすいように改善する取り組みをすると、効率が向上する可能性がある。

## 6. おわりに

用語の曖昧さや用語間の関連の曖昧さは、読み手の前提知識をもとにした解釈を必要とし、要求仕様の誤解釈による手戻りにつながる可能性がある。実際に、自然言語で要求仕様を記述しているプロジェクトにおいて、抽象的に表現されている用語が、要求仕様の誤解釈を誘発していた。

本研究では、要求仕様の誤解釈による手戻りを減らすことを目的とし、DW Modeling を考案した。実験で、DW Modeling は、以下の要求を満たす手法であることが確認できた。

- ・ レビューで見逃した「誤解釈を誘発する用語」を検出できる
- ・ 要求仕様を自然言語で記述しているプロジェクトに対して導入しやすい

DW Modeling は、「わかったつもり」の状態を抜け出すための有効かつ手軽な手法である。また、「小さく繰り返す」「書き手が実行する」「入力文書の質を高めたうえで実行する」という工夫をすることで、DW Modeling の効率を上げられる可能性が高い。

DW Modeling は、実開発においてもスムーズに導入でき、要求仕様のレビューに加えて実行されることで、要求仕様の誤解釈による手戻りを減らす効果を生み出すことが期待できる。

今後は、DW Modeling の有効性と効率性を更に高めるために、以下の取り組みを行う。

- ・ 考案手法で効果を得るための前提条件を明らかにする。具体的には、「モデリング実施者とモデルのレビューアがもっている前提知識」と「入力文書の質」が考案手法の実行結果に与える影響を確認する。
- ・ 考案手法を効率的に実行するためのポイントを明らかにする。具体的には、「小さく繰り返す」「書き手が実行する」「入力文書の質を高めたうえで実行する」という工夫の有効性を確認する。
- ・ 実開発で考案手法を継続的に活用し、その結果をもとに、手法を改善する。具体的には、DW Model の記述ルールを見直す。
- ・ ドメイン用語を漏れなく誤りなく効率的に抽出するために、自然言語処理技術の活用を検討する。具体的には、複合語や未知語を適切に抽出するための技術の活用を検討する。
- ・ 開発全体を効率化するため、要求仕様を参照する次工程でも DW Model を活用することを検討する。例えば、DW Model をテスト観点ツリー<sup>[14]</sup>のベースとして活用することを検討する。
- ・ 仕様記述ルール・ガイドや文書校正ツールなどを活用し、入力とする文書の質を高め、考案手法の効率が上がることを確認する。例えば、USDM<sup>[4]</sup>、SRS 記述ガイド<sup>[15]</sup>などの活用を検討する。



## 謝辞

本研究に対して有益なご助言をいただいた ソニー株式会社 栗田太郎氏, 国立情報学研究所 石川冬樹氏, 熊本高等専門学校 荒木啓二郎氏, 南山大学 張漢明氏, 第 35 年度ソフトウェア品質管理研究会研究コース 5 のメンバに感謝の意を表す。

本研究の実験にご協力いただいた 株式会社デンソークリエイトとテックスエンジニアリング株式会社のプロジェクトメンバに感謝の意を表す。

## 参考文献

- [1] 厳密な仕様記述WG委員, “厳密な仕様記述入門”, 独立行政法人情報処理推進機構, 2013
- [2] 大森洋一, 荒木啓二郎, “自然言語による仕様記述の形式モデルへの変換を利用した品質向上に向けて”, 情報処理学会誌 プログラミング Vol.3 No.5 18-28, 2010
- [3] 浅海智晴, “マインドマップではじめるモデリング講座”, 翔泳社, 2008
- [4] 清水吉男, “「派生開発」を成功させるプロセス改善の技術と極意”, 技術評論社, 2007
- [5] 柏原一雄, 白井正人, 小嶋秀和, 都築功, 新留光治, 村上雅哉, “派生開発におけるテスト漏れを防止する Difference Statement Coverage 分析法の提案”, ソフトウェア品質シンポジウム 2019, 2019
- [6] 小林展英, “D-Case を用いたレビューを見える化する手法の導入事例”, 12th Workshop on Critical Software System, 2015
- [7] 阿部圭一, “情報伝達型の日本語文章に現れるあいまい表現の類型化とその改善例”, 情報処理学会デジタルプラクティス Vol.5 No.1 70-79, 2014
- [8] 加藤潤三, 佐伯元司, 大西淳, 海谷治彦, 山本修一郎, “シソーラスを利用した要求獲得方法 (THEOREE)”, 情報処理学会論文誌 No.50 No.12 3001-3017, 2009
- [9] 加藤潤三, 佐伯元司, 大西淳, 海谷治彦, 林晋平, 山本修一郎, “要求獲得のためのシソーラス構築支援”, 情報処理学会論文誌 No.57 No.7 1576-1589, 2016
- [10] 小川清, 斉藤直希, 吉川直邦, “リアルタイム組み込みソフトウェアの用語の木”, 電子情報通信学会技術研究報告 コンピュータシステム 102(700) 13-18, 2003
- [11] “JIS X 0901:1991 シソーラスの構成及びその作成方法”, 日本工業規格, 1991
- [12] 樋口耕一, KH Coder 3, <http://khc.sourceforge.net/>,

2019

- [13] 西林克彦, “わかったつもり 読解力がつかない本当の原因”, 光文社新書, 2005
- [14] 羽田裕, 青木教之, “テスト視点による上流工程での予防活動と検知活動の成熟度向上”, 組込みソフトウェアシンポジウム 2013, 2013
- [15] 不破慎之介, 山田ひかり, 蛸島昭之, “要求記述のスキル不足に対する SRS 記述ガイドの有効性評価”, ソフトウェア・シンポジウム 2018, 2018