

アクターに着目した要求仕様書の品質検証手法

木村 隼人[†] 北川 貴之[†] 位野木 万里[†]

東芝ソリューション株式会社 IT 技術研究所 〒183-8512 東京都府中市片町 3-22[†]

1. はじめに

要求仕様書は開発対象となるシステムの機能や非機能など実現範囲を規定する重要な成果物である。高品質な要求仕様書の作成には、ステークホルダ間での要求の検証が重要である。要求工学知識体系 REBOK では、要求検証の観点として、完全性、追跡可能性、必要性、一貫性などの要求特性が定義されている[1]。筆者らは、そのような特性のうち、完全性、追跡可能性、必要性の検証を支援する手法およびツールを開発した[2]。しかし、その他の特性の検証は、ベテラン技術者の経験に基づき実施していた。要求間に矛盾がないことを示す一貫性の検証では、要求仕様書を構成する設計要素間の用語名が一致することを確認するが、手作業による確認は手間がかかり問題である。

要求定義の初期段階で実施するユースケース分析において、特定すべき重要な要素にアクターがある[3]。要求仕様書内で定義したアクター名の間の一貫性がないと、仕様書全体の品質に影響がある。例えば、アクターである利用者と管理者を区別せずに機能を定義すると、アクセス権が曖昧になり不適切な機能になるリスクが高まる。しかし、膨大な量の仕様書から人手で用語間の一貫性を検証するにはコストを要する。そこで本稿では、要求仕様書内のアクター間の一貫性の検証作業を効率化する手法を提案する。

2. 一貫性検証手法

ここで提案する一貫性検証手法は、ベテラン技術者のノウハウを形式知化した検証ルールを用いる。一貫性の検証ルールとして、用語不一致検出ルールと定義漏れ検出ルールの2種を定義した。これらの検証ルールは、要求定義ノウハウの構成要素に対応している。要求定義ノウハウは、テンプレート、メタモデル、検証ルー

A Requirements Verification Method Focusing on Actors

[†] Advanced IT Laboratory, Toshiba Solution Corporation, 3-22

Katamachi, Fuchu-City, Tokyo, 183-8512 Japan

ルから構成し、それぞれベテラン技術者等のノウハウを形式知化したものである[4]。図1に要求定義ノウハウの構成要素とその関係を示す。一貫性の検証ルールは、要求仕様の網羅性や整合性の観点で品質を検証するプロダクト検証ルールに対応している。

実際の要求仕様書を人手で検証するには、作業コストがかかるため、文書診断技術[5]を用いて検証を自動化した。図2に、一貫性の検証ルールと文書診断技術を組み込み開発した一貫性検証システムを用いて、要求仕様書のアクター間の一貫性を検証する流れを示す。一貫性検証システムは、アクター間の一貫性がない箇所を検出するシステムである。一貫性検証システムが出力した検証レポートを人が確認し、正しい指摘か判断する。検証レポートには、一貫性の検証ルールが想定する検証結果が記述される。以下、提案手法の特徴である一貫性の検証ルール、用語不一致検出ルール、定義漏れ検出ルールについて詳細を述べる。

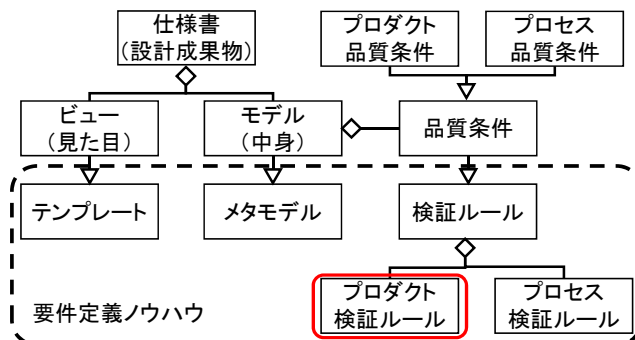


図1 要求定義ノウハウ

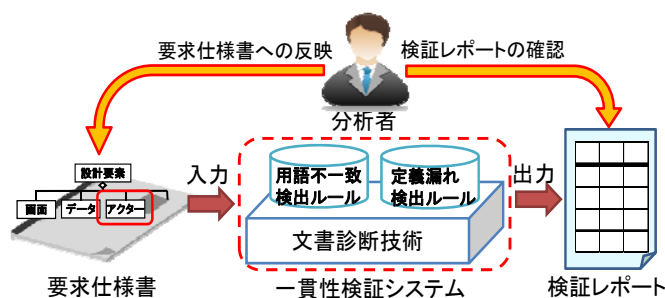


図2 一貫性検証手法の構成

2.1 用語不一致検出ルール

用語不一致検出ルールは、定義されていない設計要素が仕様書内で利用されている箇所を検出するものである。

用語不一致検出ルールは、図 3に示すような事例を想定している。アクター一覧には、「ユーザ（管理者）」、「ユーザ（担当者）」が定義されているが、仕様書内では、「ユーザ」のみで記述されている。このようにユーザに関するアクターが定義されているのに対して、別の表現でそのアクターを利用されている場合、用語不一致検出ルールでは、用語表現が不一致した箇所、つまり用語不一致箇所を検出する。

仕様書		アクター一覧	
セキュリティ要件	内容	アクター名	説明
マスタ管理	マスタ情報は権限のあるユーザのみ管理できる。用語不一致	ユーザ(管理者)	システム管理側のユーザ
		ユーザ(担当者)	システム利用側のユーザ
	

図 3 用語不一致検出ルールが想定する事例

2.2 定義漏れ検出ルール

定義漏れ検出ルールは、定義すべき設計要素が仕様書内で利用されている箇所を検出するものである。

定義漏れ検出ルールは、図 4に示すような事例を想定している。仕様書内に、「管理担当者」というアクターと思われる記述があるが、アクター一覧には定義されていない。定義漏れ検出ルールでは、このような定義漏れと思われる記述箇所を検出する。

仕様書		アクター一覧	
ユースケース名	内容	アクター名	説明
商品を管理する	システムから通知を受け、管理担当者が商品を確認する。定義漏れ	SW管理者	ソフトウェアを管理する人
		SW開発者	ソフトウェアを開発する人
	

図 4 定義漏れ検出ルールが想定する事例

3. 評価実験

実システムを対象として 2 章で提案したツールを評価した。要求仕様書は 690 ページからなり、今回は機能仕様（要件数：90 件）とアクター一覧（定義数：4 件）をとりあげて検証実験を行った。本実験の手順は①～③である。

- ① アクター一覧に定義されていないアクターの記述箇所を手手で指摘し、正解データを作成
- ② 一貫性検証システムに実験対象の仕様書を入力し、検証レポートを取得
- ③ 手順①、②で得られた指摘結果を比較し、指摘が一致した箇所を取得

上記の手順①、②、③で得られた指摘箇所の数をそれぞれ①'、②'、③'とし、下式より再現率、適合率を算出した。

$$\text{再現率} = \text{③}' \div \text{①}'$$

$$\text{適合率} = \text{③}' \div \text{②}'$$

4. 実験結果・考察

実験には、人手による指摘に 70 人 H、システムによる指摘に 10 人 H（システムの利用：2 人 H、指摘結果の妥当性確認：8 人 H）を要した。実験で計測した再現率は 87.1%（正解の指摘 93 箇所中、システムの指摘 81 箇所）、適合率は 64.8%（システムの指摘 125 箇所中、正解の指摘 81 箇所）であった。

以上のように、本手法により仕様書内のアクター間の一貫性検証作業を 60 人 H 分、工数削減できた。再現率の結果から、指摘できなかったアクター（12 箇所）の典型例、「ログイン済みのユーザ」といった「～の～」という形式のアクターについては、文書診断技術を改良し、精度を向上させる必要がある。適合率の結果については、システムは 44 箇所分、正解より多く指摘しているが、その指摘には、人手では気づきにくい、「ユーザ」といった役割が不明瞭なアクター等があった。そのような指摘結果の妥当性の確認作業を通じて、対象のドメインの理解に繋がった。

5. まとめ

本稿では、アクター間の一貫性の検証ノウハウを形式知化し、人手では困難な要求仕様書の一貫性の検証手法を提案した。文書診断技術を用いて、用語不一致・定義漏れのアクター候補を自動検出することで、要求仕様書内のアクター間の一貫性の検証作業を効率化できた。

今後は、アクター以外の設計要素（データ、画面、帳票等）についても適用・評価を実施し、提案手法の有効性を示していく。

6. 参考文献

- [1] 情報サービス産業協会 REBOK 企画 WG；要求工学知識体系（REBOK），近代科学社，2011.
- [2] 位野木万里，松尾尚典，甲田修策；メタモデルに基づき仕様書作成と仕様検証を支援するツール SpecPrince™，東芝レビューVol. 63，No. 12，2008.
- [3] Alistair Cockburn；ユースケース実践ガイド，株式会社翔泳社，2001.
- [4] 北川貴之，橋本憲幸，吉田和樹，位野木万里；要求定義における暗黙知の形式知化手法，コンピュータソフトウェア，Vol. 27，No. 3，2010.
- [5] 小林幹門；文書間整合性診断，東芝レビュー Vol. 67，No. 8，2012，pp56-57.