段階的形式化手法による要求仕様の品質向上

弓倉 陽介† 鷲見 毅† 藤本 宏† 和田 大輝† 村田 由香里†

近年，ソフトウェア要求仕様の不備を上流工程の段階で排除できる技術として形式手法が注目を集めている．しかし，開発者が形式手法を用いるには専門的な知識が求められるという問題がある．本研究では，形式仕様記述を作成する前に，半形式的な記述を用いて要求仕様の不備を排除し，形式手法の適用を容易化する段階的形式化手法を考案した．

A Stepwise Formalization Method for Improving Quality of Requirement Specifications

Yosuke Yumikura† Takeshi Sumi† Hiroshi Fujimoto† Taiki Wada† Yukari Murata†

Recently, formal methods receive a remarkable attention from industries as a means of improving quality of requirements specifications in the early stages of development. Unfortunately, the cost of trying to be proficient in formal methods is too expensive for most developers. In this paper we will present a stepwise method based on semi-formal notations, with which developers can eliminate deficiencies of requirements and make them amenable to the application of formal methods.

1. はじめに

組込み機器や社会インフラシステムの発展により，様々な場面でソフトウェアが重要な役割を担うようになっている．これに伴い，ソフトウェアに要求される機能は大規模・複雑化し，また品質も従来と比較し格段に高いものが求められている．このような背景において，上流工程で作成する要求仕様の不備が，ソフトウェアの品質に大きく影響していることが多いと考えられる．このため，上流工程でいかに必要な機能を正確に仕様として定義できるかということが重要になってきている．

これに対し，数学的手法を用いて厳密な仕様を記述し，検証するための技術として形式手法が注目を集めている．特に開発段階の早期に仕様の漏れ，曖昧さ，矛盾を取り除き，要求仕様の不備に起因する後工程における不具合数を低減することが期待されている．

しかし，形式手法の利用には形式仕様記述言語，および，それを用いて形式検証を実行するツールに習熟しなければならない．このような専門性やノウハウが要求されるため，形式手法は一般の開発者にとって活用が困難な技術となっている[1]．一方で，形式手法の適用によって見つけられる要求仕様の不備の多くは形式仕様記述を作成している過程で開発者が気づくということが知られている[2]．

これらの点に注目し，本研究では形式仕様記述を作成する前に，半形式的な記述を作成することにより，要求仕様の不備を排除し，形式手法の適用を容易化する段階的形式化手法を提案する．

†株式会社東芝 ソフトウェア技術センター
TOSHIBA Corporation, Corporate Software Engineering Center

2. 段階的形式化

2.1. 段階的形式化の概要

本研究で提案する段階的形式化では，要求仕様から形式化仕様記述を作成するために，3つの段階で仕様を形式化する(図 1).

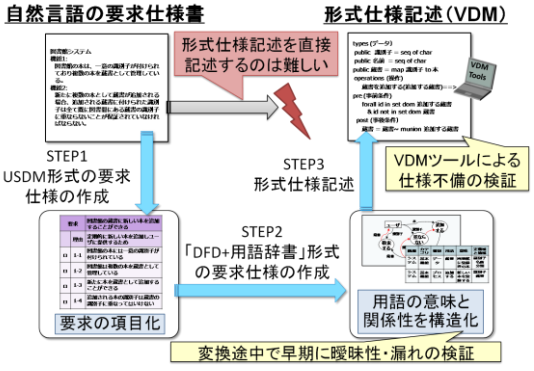

図 1:段階的形式化

2.2. STEP1:USDM 形式の要求仕様の作成

自然言語で記述された要求仕様を USDM

(Universal Specification Description Manner)形式に人手で書き下す[3]. これにより, 仕様を項目化でき, 仕様の把握が容易になる.

2.3. STEP2:「XDFD＋用語辞書」形式の要求仕様の作成

STEP1 で作成された USDM 形式の要求仕様を入力とし, 要求仕様を自然言語解析した結果から拡張 DFD(Extended Data Flow Diagram; XDFD)と用語辞書のひな形という半形式的な記述を自動生成する. また, DFD の構文チェックなどにより, 半形式化による要求仕様の洗練作業を容易にする.

XDFD は DFD に制御フローを導入したもの[4], 用語辞書は XDFD に表れる各要素のカテゴリ, 意味, 定義域, 値域を表形式で記述したものである. この形式は形式仕様記述言語よりも構文規則が少なく, 理解も容易である. また, 機能間の接続関係とその間でやり取りされるデータの定義域と値域を明確化することができるため, データ入出力の観点で要求仕様の不備を排除することができる.

表 1:XDFD の自動生成方法の具体例

パターン名	ガ格の単独パターン	ニ格、ガ格の複合パターン(受動態)
原文	蓋センサが3sec以上onとなったら、	設定値にはデフォルト値がセットされ、
係り受け解析結果	蓋センサ —ガ格→ 3sec以上onとなったら	設定値 —ニ格→ セットされ(受動態) デフォルト値 —ガ格→ セットされ(受動態)
DFDパターン	蓋センサ → 3sec以上onとなる	セットされ, デフォルト値 → 設定値

2.4. STEP3:形式仕様記述の作成

STEP2 で作成された「XDFD＋用語辞書」形式の要求仕様の情報から VDM の作成を行う. STEP2 によって, 要求仕様に含まれる機能間のデータのやり取りに関わる不備が排除できており, VDM による検証時に必要な事前／事後条件の情報が十分に得られる. このため, 自然言語で記述された要求仕様から VDM を直接記述するよりも容易に形式仕様記述を作成できる.

3. 適用事例

本稿では, STEP2 における半形式的な記述により要求仕様を洗練することの有効性を示す. 具体的な要求仕様の例として,『話題沸騰ポット GOMA-1015 型要求仕様書』第7版を用いる[5]. この仕様から自動生成される XDFD に対して, 構文チェックを実施したところ, 仕様に曖昧性が含まれていることが判明した(図 2). この曖昧性は「状態はアイドルのままである」という表現が事前条件としてアイドル状態であることを暗黙の前提としていることにある. 実際, 同じ状況下で他の状態にあるときにも状態をアイドルにする処理が必要なため, この仕様は「状態をアイドルに設定する」と記述するべきである.

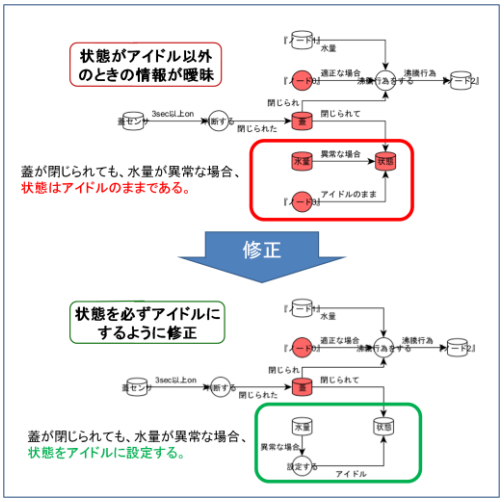

図 2:半形式的な記述を用いた仕様の洗練

4. おわりに

本研究では, 要求仕様を「USDM」→「XDFD＋用語辞書」→「VDM」という段階的形式化手法を用いることで, 専門性が必要な形式仕様記述を作成する前の段階で, 機能間でやり取りされるデータの網羅性に関する要求仕様の不備を排除できる手法を提案した. さらに,「USDM」→「XDFD＋用語辞書」作成を自動化することで作業容易化とコスト削減ができた.

今後は, STEP3 で示した XDFD+用語辞書から VDM への変換の自動化を検討する.

参考文献

[1] 中島 震, ソフトウェア工学の道具としての形式手法, 国立情報学研究所, 2007

[2] 情報処理推進機構, 情報系の実稼働システムを対象とした形式手法適用実験報告書. 2012

[3] 清水 吉男, 要求を仕様化する技術表現する技術 改訂第 2 版, 2010

[4] Paul T. Ward, The Transformation Schema: An Extension of the Data Flow Diagram to Represent Control and Timing, IEEE Tr. SE, 1986

[5] http://www.sessame.jp/workinggroup/WorkingGroup2/POT_Specification.htm