

演習コースⅡ 形式手法と仕様記述

VDM と USDM を組み合わせた仕様記述方法

- VDM による USDM 仕様記述の改善提案 -

Improving Specification Description by USDM and VDM

- A Proposal for Improving Specification Quality by VDM -

宮本 陽子 (株式会社メタテクノ)

日下部 雄三 (富士通九州ネットワークテクノロジーズ株式会社)

研究概要

本研究では、USDM 形式の仕様に対して、VDM を用いて仕様の厳密性を高める場合に、その前段階として、モデリングの観点を観点一式(ラベルセット)として対応づけることが、仕様の不具合の早期発見につながることを示す。また、実験の手順をまとめ、複数人の開発体制で VDM と USDM を組み合わせて使う場合の手順について提案する。

Abstract In this research, we study effectiveness of a specification description technique, labeling modeling point of view to USDM specifications before describing specifications with VDM. Through experiments, we show the technique effective for improving specification quality at an early stage of system development. We propose a procedure for specification description, based on the proposed technique, when using VDM and USDM in software development project.

1. はじめに

1.1 要求仕様記述とその改善における問題点

著者らの所属する開発現場では、要求仕様書(以下、仕様書)に起因する欠陥が上流工程で検出できず、このことが手戻りによる工数増加やスケジュールの遅延につながる場合もあるため、仕様書の品質向上が重要な課題となっている。仕様書に起因する欠陥を後続の工程に先送りしてしまう原因はいくつかあると考えられ、現場での議論では、次のことが挙げられた。

- 仕様書がどのような品質になれば仕様策定完了なのか、明確な基準がない。
- 仕様書のレビュー指摘数とその質はレビューアの知識や経験に依存する。
- 仕様が曖昧であることを客観的、系統的に判断する術がない。

前年度の先行研究で、USDM に則った仕様記述において、VDM を併用することが、仕様書の厳密性、無矛盾性、変更容易性を高めることに有効である[1]ことがわかった。しかし、複数人の開発体制で、実際に VDM と USDM を組み合わせて使おうとした場合、具体的な方法や手順を合意する必要がある。そのため本年度は、具体例を用いて効果的な方法を検討し、手順をまとめることにした。

1.2 仕様記述と設計方法に関するこれまでの知見

既に、仕様書の書き方に関し参考になる知見として、例えば、USDM(Universal Specification Describing Manner)がある。

USDM とは、自然言語(日本語)の要求文から、動詞と目的語に着目して仕様をとらえることや、要求を分割・階層化し、要求仕様・理由・説明をセットにした表形式で、機能仕様・非機能仕様を記述することなどを、具体例を交えて提案したものである[2]。

一方、良い仕様・文を書くための手法として、形式手法 VDM , 形式仕様記述言語 VDM++

があり、日本国内では、最近特に注目されている。形式手法とは、数理論理学に基づく科学的な裏付けを持つ言語を用いて設計対象の機能や性質を表現することにより、ある側面の仕様を厳密に記述し、開発工程で利用する手段の総称であり、VDM (Vienna Development Method) は、モデル規範型に属する手法の一種である。形式仕様記述言語 VDM++ は、ISO で標準化されている VDM-SL (VDM Specification Language) に対して、主にオブジェクト指向の拡張を行った言語である。[3]

また、仕様を機能ごとに分離し、開発作業の分業を容易にするため、Web システムで良く使われるアーキテクチャの一つとして、MVC (Model, View, Controller) がある。

MVC では、ユーザとの入力を取得しビジネスロジックを起動する Controller、ビジネスロジックを実行する Model、ビジネスロジックの結果を表示する View の 3 つの役割に分けてアプリケーションを開発する。[4]

1.3 目的

本研究の大きな目的は、VDM と USDM をどのように組み合わせることが互いの利点を活かした仕様記述につながるかを明らかにすることであるが、今回は、主に USDM 形式で表記された仕様に対して、VDM で仕様を記述する場合を取り上げ、USDM 仕様記述から VDM 仕様を記述する前段階として、モデリングの観点をラベルという形で対応付けることが効果的であることを示す。また、実験の手順をまとめ、複数人の開発体制で VDM と USDM を組み合わせて使うことを想定した場合の手順について、提案することである。

本論文の構成は、以下のとおりである。2 章で本研究の方法について、3 章で方法に対する結果、4 章で評価・考察・議論、5 章で提案について述べる。

2. 方法

本研究の方法として、対象ドキュメントと、手順について説明する。

2.1 対象ドキュメント

対象ドキュメントは、『AFFORDD 勉強会、USDM・XDDP 研修の演習資料 USDM・XDDP 演習問題解答例』[5]を対象とした。これは、派生開発推進協議会 (AFFORDD) による USDM セミナーで研修資料として用いている USDM の仕様書である。限られた時間で行う研修用の資料であるため、機能に対して必要な情報を完全にまとめたものではないが、USDM で表記された例として、参考になるものである。この USDM の仕様書は、「荷物配送システム」を題材とした Web アプリケーションで、概要説明資料 10 ページと、USDM 表記の 96 項目の仕様で記述されている。なお、全てを記述しなくても、ある程度の傾向が読み取れそうな内容であったため、今回指摘をあげる部分は「1. 集荷依頼」から「3. 荷物の照合」の 34 項目を対象とした。

2.2 研究手順

本研究は、著者 2 名が以下の手順で行った。

(1) 同一の USDM 表記の仕様書に対して、以下の 3 パターンでソフトウェア開発者の視点から、設計・実装する際に確認したい指摘点をあげる。

- ① USDM で記述された日本語のみを読んだ場合
- ② VDM++ で記述するためのモデリングを行い、ラベルを付けた場合
- ③ VDM++ で実際に記述した場合

なお、②のラベルは、複数のレビュアーの視点を考慮するため、作業者の着眼点を分け、以下の通りとした。

表 1 ②で作業者 1 と作業者 2 が付けたラベル

【作業者 1】 ②-1 : VDM ラベル	【作業者 2】 ②-2 : MVC ラベル
1. 操作の定義 (定義するクラス)	1. Model

2. 操作の名前	2. View
3. 操作の引数	3. Controller
4. 操作の本体, または事後条件	
5. 操作の戻り値	
6. 操作の事前条件, または例外処理	

②-1 VDM ラベル: USDМ では機能仕様がいくつかの処理に階層化されることに着目し, USDМ の処理が VDM の操作と対応づけられると考え, オブジェクト指向のプログラミング言語で設計を行う場合と同じように, 操作の名前・引数・戻り値などに分けることとした.

②-2 MVC ラベル: 実際の Web アプリケーション開発では MVC アーキテクチャを用いて設計・実装へと進むことが多いため, 仕様を MVC でモデリングし, Model, View, Controller の 3 種に分けることとした.

- (2) (1)で抽出した①②③の仕様問題点の傾向を見るため, 下記表 2 の種別に分類する. この種別は, 独立行政法人情報処理推進機構 (IPA) の図書館システム記述実験に適用された, 仕様不具合種別による分類^[6]を参考とした.
- (3) (1)(2)の結果から, どのような差分や傾向, 効果があるかを分析する.
- (4) 分析結果を基に USDМ で記述された仕様に対して VDM の利点を活かす仕様記述方法をまとめる.

表 2 仕様問題点の種別一覧 (分類 A)

番号	種別	内容
1	仕様の衝突	仕様の不一致. 用語のゆれ. ある設計書に記述されている仕様と他の設計書に記述されている仕様が異なること.
2	仕様の不足	仕様の漏れ. ある設計書に記述されている仕様が他の仕様書の記述されるべき箇所に存在しないこと. 例えば, 画面遷移図に存在する画面遷移が画面設計書に存在しないなど.
3	仕様の曖昧さ	仕様書に書かれている内容の定義が不明確で, 複数の意味に解釈できること.
4	仕様そのものの疑問	常識とかけ離れている仕様など. 例えば, 休館日に貸し出しができる図書館など.
5	参考確認	仕様不具合とは直接的には関係しないが, 確認したいこと.

3. 結果

研究手順 (1) (2)の結果は以下の表 3~6 にまとめた.

表 3 指摘段階ごとの指摘数

指摘段階の分類	① USDМ 読解時	②-1 VDM ラベル	②-2 MVC ラベル	③ VDM 記述時	合計
作業員 1	22 (28%)	30 (38%)	—	26 (34%)	78 (100%)
作業員 2	4 (20%)	—	5 (25%)	11 (55%)	20 (100%)

表 4 IPA の仕様不具合種別で分類した結果(左: 作業員 1, 右: 作業員 2)

作業員 1					作業員 2				
A 種別	①	②-1	③	合計	A 種別	①	②-2	③	合計
仕様の衝突	0	2	2	4	仕様の衝突	0	1	3	4
仕様の不足	9	19	17	45	仕様の不足	1	0	4	5
仕様の曖昧さ	12	8	5	25	仕様の曖昧さ	1	4	2	7

仕様そのものの疑問	0	0	1	1	仕様そのものの疑問	0	0	0	0
参考確認	1	1	1	3	参考確認	2	0	2	4

表 5 ②-1 の指摘が、どの VDM ラベルによって指摘できたか

No.	②-1 VDM ラベル	仕様の衝突	仕様の不足	仕様の曖昧さ	参考確認	合計
1	操作の定義	0	0	3	0	3
2	操作の名前	0	0	0	1	1
3	操作の引数	0	0	1	0	1
4	操作の本体, または事後条件	2	2	1	0	5
4-1	操作の本体内で利用する他の操作呼び出し	0	2	0	0	2
5	操作の戻り値	0	4	1	0	5
6	操作の事前条件, または例外処理	0	11	0	0	11
7	ラベルづけ困難	0	0	2	0	2

表 6 ②-2 の指摘が、どの MVC ラベルによって指摘できたか

No.	②-2 MVC ラベル	仕様の衝突	仕様の不足	仕様の曖昧さ	参考確認	合計
1	Model	1	0	0	0	1
2	View	0	0	4	0	4
3	Controller	0	0	0	0	0

4. 評価・考察・議論

4.1 指摘段階と指摘数からの傾向について

4.1.1 分析結果 1：段階ごとの指摘数の傾向

表 3 から、作業員 1 の場合、①での指摘数が 22 件 (28%) であるのに対し、②-1 での指摘数が 30 件 (38%)、③での指摘数が 26 件 (34%) である。②-1 の VDM ラベルを付けるという施策によって、①の指摘後に、さらに約 1.3 倍の指摘を上げることができた。なお、②-1 と③の指摘数に優位な差が見られないのは、①と②-1 の段階で、指摘した仕様不具合が解決できていないため、VDM の記述が進まなかったことが影響している。

他方、表 3 から、作業員 2 の場合、①における指摘数が 4 件 (20%)、②-2 における指摘数が 5 件 (25%)、③における指摘数が 11 件 (55%) となった。②-2 において MVC ラベルを付与する施策によって 1.25 倍の指摘が挙げられた。また、③の段階で挙げられた指摘数は 11 件と指摘全体の半数以上を占める。これは③の段階において VDM による厳密な仕様記述の効果が見られたためと考える。

4.1.2 分析結果 2：仕様不具合種別での傾向

表 4 から、作業員 1 の場合、①では、「仕様の曖昧さ」の指摘が最も多く、②-1 と③では、「仕様の不足」の指摘が最も多かった。

作業員 2 の場合、指摘種別は①の段階では「参考確認」、②-2 の段階では「仕様の曖昧さ」、③の段階では「仕様の不足」が最も多かった。この結果より、①の段階では仕様の不具合を発見できていないのに対し、②-2 の段階において仕様曖昧であることを指摘でき、さらに③の段階で VDM 仕様記述により仕様の不足や衝突、曖昧さを指摘できている。

4.1.3 分析結果 3：ラベル毎の仕様不具合種別での傾向

表 5 から、作業員 1 は、VDM ラベルのうち、「操作の事前条件、または例外処理」によ

って、「仕様の不足」を 11 件と最も多く指摘できた。しかし、それ以外には VDM ラベルと仕様の不具合種別に、特筆すべき関係は見出せなかった。

表 6 から、作業員 2 の場合、MVC ラベルにより発見できた指摘数は 5 件であり、その中の 4 件は View ラベルによる「仕様の曖昧さ」の指摘であった。他のラベルにおける指摘は「仕様の衝突」が 1 件しかなく、その他の種別については指摘が挙がらなかった。

4.2 ラベルによる指摘について

4.2.1 VDM ラベルについて

4.2.1.1 VDM ラベルによる指摘数の増加

4.1.1 分析結果 1 の作業員 1 の場合に、②-1 の VDM ラベルによって指摘が増加したのは、「システムが行う機能・処理に関する仕様を実現するには、操作の本体または事後条件だけでなく、入力となるデータの種別（引数）・戻り値・事前条件・例外処理という観点を明らかにする必要がある。各仕様項目と観点との対応関係がわからなければ、実装に差し支える。」と考えて、仕様項目のグループと、一通りのラベルがあるかを対応付けながら確認したことによる。例えば、下記、表 7 の指摘例 1~3 は、処理を表す仕様グループと、操作のラベルを照らし合わせたことによって、発見できたものである。指摘例 1 は、まず、「操作の引数」ラベルと対応づけられる仕様項目があるかないかを確認し、次に、対応づくものが無い場合は、引数になるような条件がなくても実現可能か、と思考する範囲を限定し、深く考えることによって気づいたものである。同様に、指摘例 2 と 3 は、「操作の戻り値」「例外処理」ラベルと対応づく仕様項目があるかないかを確認し、対応づかない場合は、それで良いかと考えを進めることで指摘できた。

このように、機能・処理を表す仕様項目のかたまりに対して、VDM の操作で用いるものを、揃えるべき観点一式（ラベルセット）とし、照らし合わせることは、観点によって考える対象を狭め、深く考えることにつながり、USDM 形式の仕様のレビューに効果がある。

表 7 作業員 1 の VDM ラベルによる指摘事例

番号	要求仕様	仕様グループの分類	VDM のラベル	指摘内容	指摘分類	
	<配送データの読み出し>	処理効果・ 処理結果	2.操作の名前	-	-	
30.2	荷物 DB から配送データを読み出す		4.操作の本体または事後条件	-	指摘例 1	-
	処理に関する仕様であるため、VDM の操作で用いる一通りの観点（ラベルセット）がそろっているか確認することにより、指摘できた。		3.操作の引数	配送 DB から、何を条件にして読み出すか？	仕様の曖昧さ	
			5.操作の戻り値	配送データ 1 件のみを読み出すという解釈でよいか？1 件ではなく、該当するものを全て読み出すという解釈もできるため。	指摘例 2	仕様の曖昧さ
		6.操作の事前条件、または例外処理	荷物 DB から配送データを読み出せない場合は、どのようなエラー処理を行うか？ (配送 DB には、配送データが登録されている前提条件か？)	指摘例 3	仕様の不足	

4.2.1.2 VDM ラベルと VDM 記述の指摘

分析結果 1 の作業員 1 の考察を通して、③では、①でも②-1 でも気づかなかった 26 件の仕様不具合を指摘できたことを述べた。どのような指摘ができたかは、下記表 8 の例で紹

介する。指摘例 4 は、生成した配送データの「問い合わせ用番号」に対して、システムから何らかの手段で取得した番号を設定する仕様となっている。VDM で記述してみると、他の部分を読んでも、システムから何の番号をどのように取得するかがわからず、VDM で記述できないことにより気づいたものである（“--” を使い VDM ではコメントとしている）。指摘例 6 も、同様に、VDM でどのように「今日の日時」を取得するかがわからず、記述できないことにより指摘できた。また、指摘例 5 は、「預かり日」と、「今日の日時」を、VDM で詳細に記述すると型が異なるため、VDM ツールの型チェックにより気づいたものである。このように、②-1 の手順でラベルを付与する際に挙げた指摘と、③で VDM を記述する際に挙げた指摘は、同じ種別のものが多く、③より前の段階で、指摘できた可能性もある。しかし、②まで気づけなかったことに、③では、VDM ツールによる構文チェックや型チェックなどの機械的な支援があることにより気づきやすくなり、チェックが徹底できる。

表 8 作業員 1 の VDM 記述による指摘事例

番号	要求仕様	VDM ラベル	VDM 記載内容	
205	新規の配送データを生成する この時、配送データの項目は「空」の状態 で生成する	4. 操作の本体または事後条件	<pre> operations public static 配送データの生成 : 「荷物番号」 ==> 配送データ 配送データの生成(a 荷物番号) == (def w 配送データ = new 配送データ() in(--スキャナーで読み込んだ番号 w 配送データ.荷物番号設定(a 荷物番号); --★どのようにシステムから入手する? --w 配送データ.問い合わせ用番号設定(); w 配送データ.荷物ステータス設定(<集荷済>); --★今日の日時?日付? --★どのように今日の日付を取得する? --w 配送データ.日付設定(); : return w 配送データ;)); </pre>	
:	「荷物番号」=スキャナーで読み込んだ番号			指摘例 4 仕様の不足
	「問い合わせ用番号」=システムから入手する			指摘例 5 仕様の曖昧さ
	「荷物ステータス」=集荷済			指摘例 6 仕様の不足
	「預かり日」=今日の日時			

4.2.2 MVC ラベルについて

4.2.2.1 MVC ラベルによる指摘数増加

MVC ラベルは仕様記述に対して MVC アーキテクチャレベルでの設計を意識することにより Model, View, Controller 階層に該当する仕様の問題点を事前に抽出することを目的として実施した。その結果、View ラベルで 4 件、Model ラベルで 1 件の指摘を挙げることができた。以下に最も指摘が多かった View ラベルによる指摘事例を示す。

表 9 作業員 2 の MVC ラベリングによる指摘事例

番号	要求仕様	仕様グループ	MVC ラベル	指摘内容	指摘分類
10.2.3	担当配達員が複数表示された時は、そこから 1 名を選ぶ	処理結果	View	「選ぶ」のはユーザか、あるいはシステムか。前者の場合はユーザにどのように選ばせるのか。後者の場合は選ぶ判断基準は何か。	仕様の曖昧さ

上記の指摘事例の場合、要求仕様に対して View ラベルを割り当て、画面操作の観点か

ら仕様を考えることにより，担当配達員を1名選ぶのはユーザ自身が行うのか，またはシステムが自動で選ぶのが曖昧であることを指摘できた．また，前者の場合はユーザにどのように担当配達員を選ばせるか，例えば，ラジオボタンを使用するのか，セレクトリストを使用するのが曖昧であることを指摘できた．後者の場合は何を基準に担当配達員を1名選ぶのが不明であることが指摘できた．

4.2.2.2 MVC ラベルと VDM 記述の違い

MVC ラベル付与後に VDM 記述を行った結果，新たに 11 件の指摘を抽出した．以下に VDM 記述の指摘事例を示す．

表 10 作業員 2 の VDM 記述時の指摘事例

番号	要求仕様	MVC ラベル	VDM 記載内容
30. 20	配達先の郵便番号から集配拠点と拠点からの配達時間を割り出す 【説明】 配達先の郵便番号から集配拠点番号と拠点名を割り出す処理は既に用意されている	controller	Operations public 配達先の郵便番号から集配拠点と拠点からの配達時間を割り出す：「配達先情報」*「拠点番号」=>「日時」 配達先の郵便番号から集配拠点と拠点からの配達時間を割り出す(a 配達先情報, a 拠点番号) == is not yet specified; -- ★集配拠点と拠点から配達時間を割り出す方法が不明

上記の指摘事例は集配拠点と拠点の間の配達時間を割り出すために使用できるデータが住所しかなく，配達時間の計算方式またはそのためのデータが不足していることを指摘した事例である．この事例のように MVC ラベルでは，設計や実装に近いレベルの仕様の問題点を指摘できなかった．これは MVC ラベルが，仕様を機能ごとに分離することとどまり，ラベル付けによって仕様を深く考えたり，考える対象を狭めたりすることには寄与しないためだと考えられる．

5. 提案

本章では，前章までの考察結果をもとに，開発現場に適用するための手順の一提案と適用工程を示す．また，仕様のレビューにおいて，ラベルや VDM により得られる指摘と同様の指摘を得るための方法を提案する．さらに，提案方法における USDM，ラベル付け，VDM 記述の役割を示す．

5.1 開発現場に適用する手順・開発プロセス

今回の手順を整理し，開発現場に適用する手順を以下にまとめる．なお，以下の手順は既に USDM 形式の仕様が存在し，VDM と組み合わせることによって，仕様の品質向上を目指すものであるため，適用する開発プロセスは，USDM を用いる工程と同じである．

(1) 観点一式(ラベルセット)の定義

一般的なシステムで頻繁に目にするような仕様，例えば，ユーザ入力 UI 仕様，システムの処理仕様などごとに，それらの仕様項目のまとまりに対して，「一通り付いてほしい」という観点一式(ラベルセット)を定義する．(今回は代表例のみを示した．)

(2) 仕様グループの分類

表 7, 9 のように USDM に仕様グループの分類欄を設ける．USDM の自然言語仕様を目を通して，仕様項目のまとまりである<グループ>に対して，仕様グループ欄に「システムが行う処理」などと，記載していく．

(3) 仕様グループとラベルセットとの対応づけ

(2)で設けた仕様グループの分類欄に記されたものと、ラベルセットを対応づけていく。その際、仕様グループと観点一式がそのまとまりにおいて一通り使われているか(それらの観点が一通り記述されているか)記述しながらラベルを付け、徹底するようにする(+レビュー時にラベル一式と照らし合わせてチェックする)

(4)典型的なテンプレートと指摘集の定義

仕様項目のラベルごとに、典型的なテンプレート(記述時に使う)および指摘集(レビュー時に使う)を定義する。

(5)ラベルに応じたテンプレートを用いてVDMを記述

ラベルに応じたテンプレートを用いて、VDMを記述する。記述したVDMに対しては、指摘集を用いてレビューを実施し、確認する。

5.2 USDM・ラベル付け・VDM記述の役割分担

自然言語・USDMでは機能仕様・非機能仕様だけでなく、理由や背景などの表記欄があるため、仕様を「広く」記述できる。一方、VDMは機械的な支援を得ることができ、記述の制約があることによって、仕様の不足や曖昧さの除去を徹底できる。それが仕様を「深く」理解し、議論することにつながる。ラベル付けは、両者を橋渡しする役割をもち、自然言語・USDMで記述された仕様の中から、機能と処理に関するVDMの記述に向く部分と、そうでない部分、例えば、保守性などの品質に対する要求を区別する方法として有効である。また、実際の開発プロセスにおいて、自然言語・USDMとVDMの両方で仕様を記述するのは、コストが問題となることも多い。VDMで記述するコストがかかけられない場合には、自然言語・USDMの処理ごとに、VDMの操作ラベルが一通りそろっているか、確認を行うだけでも、仕様の不具合指摘に効果がある。

なお、自然言語・USDMとVDMのどちらが先ということもないため、VDMの詳細仕様をUSDMの概要仕様で反映するなど、VDMからUSDMの順で仕様を記述することも有り得る。VDMで記述された仕様に対して、Javadocのようなコメント形式で、その仕様が属する上位の要求と理由、および説明を併記し、VDMからUSDM形式の仕様をツールで変換できるようにする方法を将来課題として検討したい。

6. おわりに

USDM表記の自然言語とVDMとを組み合わせる仕様を検討したり、記述したり、レビューしたりする際には、仕様の範囲を狭めるVDMの観点を「ラベルセット」として用いることを提案した。システムやドメインによっては、必要とする観点が異なるため、今後は、それぞれの特性に合わせた観点一式を検討していきたい。また、提案した方法を開発現場に適用し、効果を検証していきたい。

謝辞

本論文の執筆に当たり、システムクリエイツの清水吉男氏、九州大学大学院の荒木啓二郎教授、栗田太郎主査、石川冬樹副主査、日本科学技術連盟・演習コースⅡの研究員の皆さま、日本科学技術連盟・事務局の皆さまにお世話になりました。厚く御礼申し上げます。

1 ソフトウェア品質管理研究会, 第27年(2011年度)分科会成果報告, 日本科学技術連盟

2 清水吉男,【改訂第2版】[入門+実践]要求を仕様化する技術・表現する技術~仕様が書けていますか?, 技術評論社, 2010

3 栗田太郎・荒木啓二郎, モデル規範型形式手法 VDM と仕様記述言語 VDM++ -高信頼性システムの開発に向けて-, 信頼性, 2009

4 日本OSS推進フォーラム, <http://ossforum.jp/node/645>, 2013/01/24 アクセス

5 派生開発推進協議会(AFFORDD), <http://www.xddp.jp>, 2013/01/24 アクセス

6 独立行政法人 情報処理推進機構, 図書館システム記述実験報告書, 独立行政法人 情報処理推進機構, 2012