VDM を用いた機能仕様記述に関する研究
-自動販売機システムを題材として-

2011SE103 岩田陽平　2011SE101 岩瀬拓也
指導教員：張漢明

1　はじめに

ソフトウェア開発において，安全性や信頼性を確立するために，開発文書の形式手法は有効性と言われている．開発文書で形式手法を使うことで，開発上流過程でのミスを減らし，作業の手戻りを減らすことができ，要求を正しく実現することが期待できる．形式手法の中で VDM が一般的によく使われているものの一つであり，本研究では，VDM での仕様書の記述が有用であるか確認を確認することで，形式手法の有用性を示すことができると考える．開発文書の仕様書では，主に機能と振る舞いが記述されており，本研究では機能に着目する．

本研究の目的は事例を用いて機能仕様記述における VDM の有用性を確認することである．機能仕様を記述するのに VDM を用いて記述することでどういった利点や欠点があるかを評価することで，有用性を確認する．

本研究の基本的なアイデアは，実用レベルの開発文書を分析して VDM-SL の関数に基づいた機能仕様を記述することである．仕様記述の機能を抽出し，ユーザから見た機能と機器の機能に分類を行い，ユーザから見た機能と各機能ごとにモジュール化をして，VDM を記述する．VDM 記述と既存の仕様書を比較，評価を行う．

本研究ではテスト設計コンテストの自動販売機システムの機能仕様を事例として仕様書の分析し，モジュール化して VDM を記述する．開発文書自体の評価を行い，記述した VDM 記述と比較することで，ソフトウェア開発において，実用レベルの開発文書で形式手法を用いることの有用性を確認し，提示した．

2　背景技術

本章では，形式手法の概要と VDM の概要について述べる．

2.1　形式手法の概要

形式手法とは，数学的に厳密に意味付けられた手法である．情報システムの要求や設計等を記述し、情報システムがユーザの要求等を満たしているかなど論理的に推論するための仕組みを提供する手法である．抽象化，厳密化といった面において優れたモデルがあれば，正確な分析と予測が妥当なコストで可能となる [1].

2.2　VDM の概要

VDM は，歴史ある代表的な形式手法の一つであり，VDM-SL，VDM++，VDM-RT の三つの形式記述言語がある．VDM には，関数型や手続き型などの様々な記述スタイルがある．以下に特徴を示す [2].

- 実際の開発で使われている
- 識別子に日本語が使える
- 信頼性の高い無償ツールがある

3　基本的なアイデア

本研究のアイデアは，テスト設計コンテスト ASTER 自動販売機ハードウェア構成および販売者用機能仕様書を事例として，以下の手段で VDM の有用性を評価する．

- 開発文書の分析
- VDM を用いて仕様書を記述

3.1　開発文書の分析

本研究では，機能に関する記述のみに着目した．機能とは，入出力の関係で，入力に対して出力が決まるものである．VDM-SL の関数で記述するためには，機能の名前，入力，出力が必要である．また，既存の文書にはユーザの要求記述とハードウェア要求記述が混在しており，抽象度もバラバラであるから，それを分類する必要がある．さらにオブジェクトの抽出を行う．オブジェクトの抽出は仕様書に記述された各機能ごとの機器をクラス図で記述する．機能の抽出は以下のように行い分析する．

- ハードウェアの構成　クラス図
- 仕様書から機能の候補を抽出
- ユーザから見た機能，機器の機能を分類
- 各機能ごとに入力と出力を記述

3.2　VDM 記述の枠組み

本研究では，VDM-SL を使用する．機能は入出力の関係を表すので，関数スタイルで記述する．実行結果のテストを可能にするので，陽仕様で記述する．仕様書の分析によって構造化した機能を以下の方法で VDM で記述する．

- 対象のグループ化
- 対象物の構造を記述
- 機能を操作で記述

3.2.1　対象物のグループ化

機器ごとにモジュール化して記述することで，仕様書が構造化されて見やすくなる．本研究では，以下のように VDM でモジュール化して記述する

```
module A
imports
  from B
```

```
  from C
  from D
export all
```

3.2.2　対象物の構造

以下のように構造を複合型で記述する．

```
複合型名：：フィールド名１：型
        ：：フィールド名２：型
                :
                :
        ：：フィールド名ｎ：型
```

不変条件は以下のように，記述する．
inv = 条件

3.2.3　機能の記述

以下のように操作を function で記述する．

```
識別子：入力型＊Ａ（引数）　-> Ａ（返り値）
識別子（引数）　= 関数本体
```

以下のように，事前条件を記述する．

```
function
操作Ａ：入力型＊引数の型 -> 返り値
操作Ａ（引数）＝ 関数本体
pre 操作Ｂ
操作Ｂ：引数の型 -> bool
```

操作Aが行われる前の条件を操作 B で bool 型で真偽値を返す定義をする．

4　事例:自動販売機システム

本章では，自動販売機システムを事例として，機能の分析と VDM 記述を行う．

4.1　機能の分析結果

本節では，自動販売機システムのユーザの要求記述とハードウェアの要求記述を分類する．

4.1.1　ハードウェアの構造

図 1 は各機器ごとに，時間監視機器，商品選択機器，検証ルーレット機，入力機器，商品処理機器，金銭処理機器の 6 つグループにクラス分けをした．これによりクラスがどの機能に対応しているかを明確になる．

4.1.2　機能

既存の仕様書に色付けをし，機能を抜き出して機能ごとの入出力を記述した．

ユーザから見た機能
ユーザから見た機能は以下の 4 つである．

- 購入 (販売ボタン押下)　入力：どのボタンを押すか，出力：商品排出
- 紙幣投入　入力：紙幣投入する，出力：紙幣あり状態
- 硬貨投入　入力：硬貨投入する，出力：硬貨あり，金額表示される
- 返金 (返金ボタン押下)　入力：返金レバー押下，出力：貨幣返金，紙幣無し状態

機器の機能
以下は機器の機能の一部である．

- 準備中ランプ点灯　入力：当該商品のラックが適温でない，出力：準備中ランプ点灯
- 紙幣処理機器格納　入力：商品の販売が完了した，出力：紙幣格納

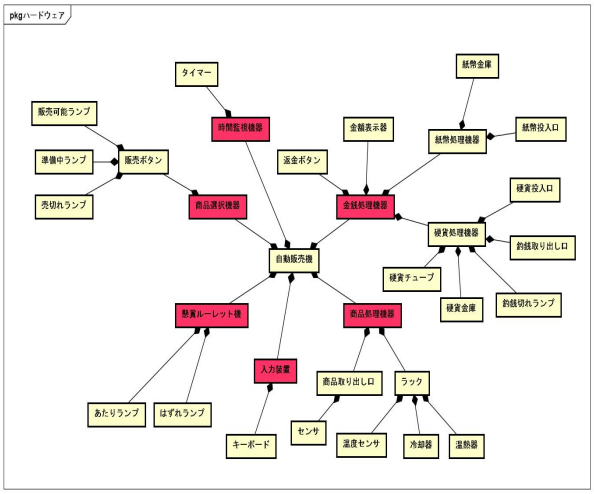

図 1　自動販売機システムのハードウェア構造

4.2　VDM 記述

本節では，VDM は 3.3 に基づいて記述した．

4.2.1　対象物のグルーピング

構造をグルーピングしてモジュールで VDM を記述する．

```
module 自動販売機
imports
    from 商品処理 all,
    from 金銭処理 all,
    from 商品選択 all,
    from 時間監視機器 all,
    from 懸賞ルーレット all,
    from 入力装置 all
exports all
definitions
```

4.2.2　構造の記述

構造は type で記述し，図 2 のようなコンポジションの関係を VDM の複合型で定義をする．図 2 は商品選択機器の構造をクラス図で表現したものである．

```
types
 販売ボタン型::
     販売可能ランプ：販売可能ランプ型
     準備中ランプ：準備中ランプ型
     売切れランプ：売切れランプ型;

 販売可能ランプ型::
```

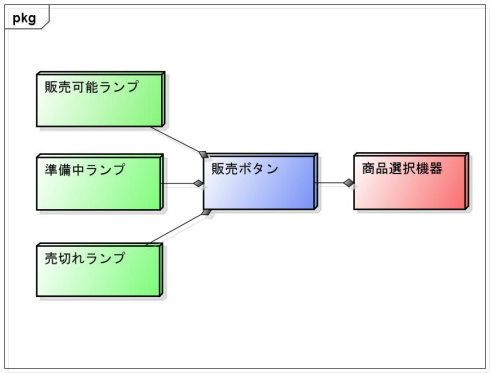

図 2　商品選択機器

```
    ランプ: ランプ型;

準備中ランプ型::
    ランプ: ランプ型

売切れランプ型::
    ランプ: ランプ型
```

4.2.3　機能の記述

機能の記述は操作で記述する．操作を VDM で記述する際は入出力をはじめに考え，function で記述する．例として商品処理機器の操作を記述した．図 3 は，商品処理機器の構造のクラス図である．

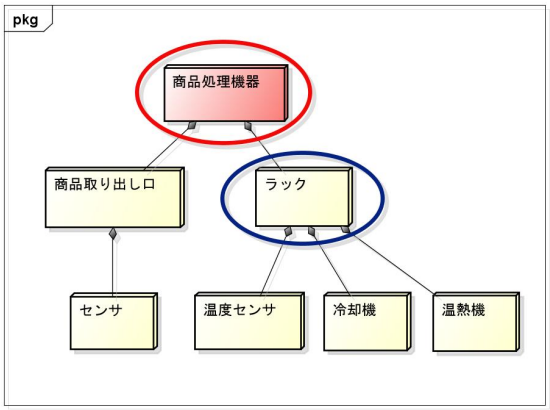

図 3　商品処理機器

以下はラックのクラスの在庫数を減らすという操作の記述である．ラックの型であるラック型を引数にとり，ラック型を返す．つまり，在庫数が減るというラックの状態遷移を表している．

```
functions
 在庫数を減らす:ラック型 -> ラック型
 在庫数を減らす(ラック) ==
   mu(ラック, 在庫数 |-> ラック.在庫数 - 1)
```

以下は商品処理機器のクラスの商品を排出するという操作である．ラック map とは，ID とラックの写像である．ID 型と商品処理機器の型である機器型を引数にとり，引数にとられた ID に対応するラックを上書きしたラック map を更新するという商品処理機器の状態遷移を表している．つまり，商品が排出されると在庫数が減ることを表している．

```
functions
 商品排出:ID 型 * 機器型 -> 機器型
 商品排出(id, 機器) ==
   mu(機器, ラック map |-> 機器.ラック map
        ++ {id |-> 在庫数を減らす(機器.ラック map(id))});
```

4.2.4　事前条件

操作によって，状態が変わること前の状態を事前条件で表す．本研究では，VDM で事前条件を記述するには function で pre と記述する．例として金銭処理機器の紙幣処理機器紙幣投入という function で事前条件を記述した．以下の図 4 は金銭処理機器の構造のクラス図である．

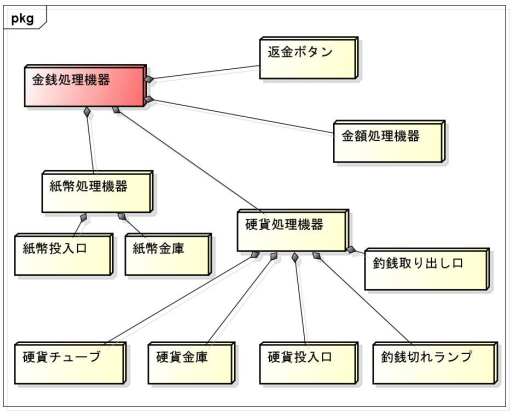

図 4　金銭処理機器

紙幣を投入するという操作が行われる前の条件として，紙幣が受付可能であることを表している．

```
functions
 紙幣処理機器紙幣投入:紙幣処理機器型 -> 紙幣処理機器型
 紙幣処理機器紙幣投入(紙幣処理機器) ==
    mu(紙幣処理機器, 紙幣投入口 |-> 紙幣投入口紙幣投入(紙幣処理機器.紙幣投入口))
 pre 紙幣受付可(紙幣処理機器.紙幣投入口);

functions
 紙幣受付可:紙幣投入口型 -> bool
 紙幣受付可(紙幣投入口) == 紙幣なし(紙幣投入口);
```

4.2.5　不変条件

構造の記述の中で不変条件を記述する．不変条件とは，絶対に変わらない条件のことである．本研究では，VDM で不変条件を記述するには，type で inv と記述する．例として図 3 である商品処理機器のラックの型で記述した不変条件が以下である．

```
types
 ラック型::
    モード: ラックモード型
```

```
  収納数上限: nat
  在庫数: nat
  温度センサ: 温度型
  温商品温度範囲: 上限下限型
  冷商品温度範囲: 上限下限型
  温度制御機器: 温度制御機器型
inv r == r.在庫数 <= r.収納数上限;
ラックモード型 = <温商品> | <冷商品>;
```

収納上限数と在庫数の型を nat で定義し，不変条件で絶対に在庫数が収納上限数を上回らないことを表している．

5 考察

本章では，開発文書自体の評価と VDM の評価を行う．

5.1 開発文書の評価

本研究で，事例として分析した「ASTER 自動販売機ハードウェア構成および販売者用機能仕様書」に対する評価は，構造化できていない，言葉が厳密に定義ができてないという点である．本研究で分析を行った手順では，機能の抜き出す際に決められたルールがないため，人によって抜き出す機能が変わってきてしまう．

5.1.1 構造化について

「ASTER 自動販売機ハードウェア構成および販売者用機能仕様書」は，機能が機器ごとに分類していないので，読み手に分かりにくくなっている．仕様書の段階で構造化することで，次の開発過程が効率よくなり，ミスも減る．

5.1.2 言葉の定義について

仕様書には以下のような厳密に定義されていない言葉が記述されていた．仕様書にあった厳密に条件が定義されていない言葉を以下に挙げる．
「販売可能時に点灯する販売ボタン」
仕様書には，販売ボタンに内蔵するランプは準備中ランプと売切れランプと記述されている．「販売可能時に点灯するランプ」が記述されていない．本研究では分析の際に，この「販売可能時に点灯するランプ」を販売可能ランプと名前をつけた．販売可能ランプの言葉自体の意味は販売可能時に点灯するランプである．仕様書では，販売可能である条件をランプの点灯で表している．しかし，販売ボタンのランプには他に準備中ランプと売切れランプがあり，販売可能ランプだけ色の定義がされていない．

5.2 VDM 記述の評価

仕様書の VDM 記述と自然言語の仕様書を比較し，評価を行う．
自然言語の仕様書には以下の問題があった．

- 同義語
- 多義語
- 厳密に定義されていない言葉
- 行間で書かれている

同義語は，複数の言葉が同じ意味を持っている言葉である．例：釣銭用硬貨，釣銭…釣銭 VDM 記述する際に同じ意味の言葉を一つの言葉にまとめて定義した．

多義語は，一つの言葉が複数の意味を持つ言葉である．例：ラック…ラック単体，ラック全体 VDM 記述では，ラック単体の意味をラックと定義し，ラック全体のことを写像型でラック map と記述した．

言葉一つ一つの定義が厳密ではなく, それを構造化して書かれていないので, 書くべき内容がまとまっていない. 次に仕様書には, 多義語や同義語などによって言葉や意味が様々な書き方をしているので読み手が内容を考えながら読まなければならない. しかし,VDM で記述することによって一つ一つの言葉が厳密に定義され，なお且つ構造されて書くことができるので，読み手に誤解を生むことがなくなる．

- VDM の良い点
  - 機能の入出力を記述するので，各機能の状態遷移が分かる．
  - 本研究では，行っていないが構文のテストができるので，ミスを見つけることができる
- VDM の悪い点
  - 一度仕様書の分析をしてから記述しないといけないので，時間的コストが大きい
  - 記述する人も，読み手も VDM の構文を覚える必要がある
  - 仕様書の分析によって，抽出した機能の入出力がない場合の VDM 記述が困ると気づいた

6 おわりに

本研究では, 機能仕様記述の VDM で記述することで, ソフトウェア開発において実用レベルの開発文書での VDM の有用性を確認することができた．今後の課題としては，本研究で書いた VDM 記述をテストすることである．テストケースを作成して，VDM で定義した関数を実行することより，VDM の定義が妥当であることを確認する．自然言語の仕様書に VDM 記述をリンク付けして，提示する．

参考文献

[1] J. fitzgerald, Peter. G. Larsen, P. Mukherjee, N. Plat, and M. Verhoef, Validated Designs for Object-oriented Systems, Springer, 2005.

[2] 荒木啓二郎，張漢明，プログラム仕様記述論，オーム社，2002.

[3] ASTER テスト設計コンテスト，自動販売機ハードウェア構成および販売者用機能仕様，*http://aster.or.jp/business/contest/doc/2015tdc-v1-1.zip*，2015.