仕様モデルに基づく VDM-SL 記述支援に関する考察

2013SE002 天野大樹　2013SE232 宇野喬雄
指導教員：張漢明

1　はじめに

形式手法は,「信頼性」や「安全性」が求められるソフトウェア開発において有効な手法である. 形式仕様言語を用いることによって作業の手戻りを減らすことが期待できる. 形式手法の有名な事例の一つとして日本では,「携帯電話組込み用モバイル FeliCa IC チップ開発」が成功例として有名である. 記述されたものが顧客のニーズと合っているかどうか妥当性確認及び設計やプログラムが仕様を満たしているか正当性検証を調べるための基準として仕様は重要である. 仕様は重要で曖昧さがないものが求められるが, 形式手法は数学の概念に基づいた厳密な記述による曖昧さの排除をすることによって有効な手段だと言われている. しかし実用的なソフトウェア開発において現実的に形式仕様言語は普及していない. ソフトウェアの仕様記述における形式仕様言語を導入する障害として「仕様記述の手順がない」,「実用的な形式仕様記述の事例が少ない」ということが挙げられる. 形式仕様言語を普及するためには仕様書に対する形式仕様記述の導入方法を提示する必要がある.

昨年度の卒業研究 [4] は 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 [3] に VDM を適用して, その有用性を確かめた. そして関数スタイルの記述法を提案して, 自動販売機の仕様書に対し適用し, 実用性を確認した. しかし昨年度の卒業研究では記述のプロセスは言及していない.

本研究の目的は, 自然言語の仕様書から VDM-SL を用いた機能仕様書を作成するための仕様モデルを提案して作成指針を考察することである. 仕様モデルを作成するために自然言語の仕様書と VDM-SL 記述を対応し分類する. 「どのようにして形式仕様記述を得たか」や「なぜそのように記述したのか」,「より良い記述とは何か」の観点から分析する.

2　背景技術

形式手法,VDM-SL の概要について述べる.

2.1　形式手法

形式手法は情報システムの要求や設計等を記述したもので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. 1960 年代の後半から 1970 年代にわたって盛んに研究されたプログラムの検証理論であり 30 年以上の長い歴史を有している. またヨーロッパを中心に基礎研究や実用化が進められている.

2.2　VDM-SL

機能仕様を記述するための言語として VDM-SL を用いる. VDM は形式手法の中で一般的によく使われるものである. VDM-SL は VDM の記述言語の一つである.

VDM-SL を用いる理由として

- 日本語の識別子が利用可能
- 実績のあるフリーのツールの存在
- 日本語を含めたマニュアルと参考文献の存在

が挙げられる.

3　研究のアプローチ

研究のアプローチは以下の 4 つである.

- VDM-SL 記述のための仕様モデルの提示
- 作成指針の検討
- VDM-SL のテンプレートを用意
- 適用事例

VDM-SL 記述のための仕様モデルの提示について, 昨年度はテンプレートを「もの」と「操作」の二つを用意した. 本年度は仕様についての構成要素を昨年度より詳細化しそれらの構成要素の関係をクラス図としてまとめた仕様モデルを提示する.

自然言語の仕様書と VDM-SL 記述の間の関係を分析する. 行うことは,「VDM-SL 記述の分析」と「自然言語の仕様書と VDM-SL 記述の対応」である.

「VDM-SL 記述の分析」は,「自然言語の仕様書と VDM-SL 記述の対応」とは逆に VDM-SL の記述を見て分析を行うことにより作成指針の検討を行うことである.

「自然言語の仕様書と VDM-SL 記述の対応」は, テンプレートをそのまま使用したものと, テンプレートを使用してもうまく記述できなかったものに分け, テンプレートを使用してもうまく記述できなかったものについてどのように記述するかについての指針を考察する. 自然言語の記述を分析する際の指針とする.

VDM-SL のテンプレートについては関数スタイルで用意する. 関数のスタイルでは代入や繰り返しを含まず宣言的で簡潔な記述を促す.

事例には昨年度と同様, 自動販売機を用いる. 仕様モデルの分類ごとのテンプレートに事例を適用させる.

4　仕様モデルと記述のテンプレート

VDM-SL で記述するための仕様モデルを提示して, そのテンプレートを説明する.

4.1　仕様モデル

仕様モデルを作ることにより, 機能仕様の分類関係をひと目で理解することが出来る. 仕様モデルを元に, 分類ごとに記述のテンプレートに書き換える. 提案する仕様モデルを図 1 に示す. 図の仕様モデルを見ると, 仕様は用語で成り立っていることがわかる. 用語は意味と名前で成り立っている. 用語には次の 5 つがある.

・もの・対応・述語・操作・事象

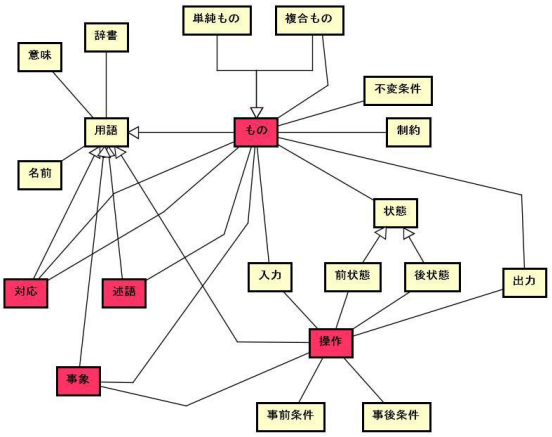

図 1　仕様モデル

（1）もの

仕様モデルのものにはコンポジットパターンで表されている. 「単純もの」と「複合もの」に分けられており,「不変条件」と「制約」が使われるときもある.

（2）対応

対応は, 対応元と対応先の 2 つのものが存在する. 対応元と対応先の関係は関数の入力と出力の関係を表している. 関数は複数の入力に対してただ 1 つの出力が定まるものである.

（3）述語

述語はもの, もしくは, ものとものの間の性質を表す.

（4）操作

操作は, 状態の変化を表す.

（5）事象

事象は入力に対して操作と対応付ける. 入力されたものを操作で定義する.

4.2　VDM-SL のテンプレート

ものの分類についてそれぞれテンプレートの観点から説明する.

（1）もの

「もの」は, 自然言語の仕様に存在する用語の名前の語尾に「型」をつけたものである. 「もの」には以下の 2 つが存在する.

・単純もの・複合もの

単純もの

「単純もの」については, 用語がどのような型であるかを説明するものや列挙を表すものがある.

```
types
  名前型 = 型
types
  名前型 = <値 1> | <値 2> | … | <値 n>
```

複合もの

「複合もの」はある用語について複数の解説を加えるときにレコード型として表される.

```
types
  名前型::
  名前 1: 名前 1 型
  …
  名前 n: 名前 n 型;
```

（2）対応

「対応」は対応元のものを与えることで対応先のものへと変化し返すことを表す. 対応元のものは複数個存在するときもあるが対応先のものはただ 1 つに定まる.

```
functions
  対応名: もの型 1 * もの型 2 * … * もの型 n -> もの型 x
  対応名（対応元 1, 対応元 2, … , 対応元 n） ==
    対応先
```

（3）述語

「述語」は VDM-SL の関数を使って bool 値を返す. ものを与えることで「true」か「folse」が返される.

```
functions
  述語名: 型 1 * 型 2 * … * 型 n -> bool
  述語名（名前 1, 名前 2, … , もの n） ==
    述語の性質
```

（4）操作

「操作」は引数に「入力」と「前状態」を取り「出力」と「後状態」を返す関数として表す. また入力するものは, 事前条件を満たすものであり, 出力されるものは事後条件を満たすものである必要がある.

```
funcitons
  操作名: 型 1 * 型 2 * … * 型 n * 型 x ->  型 x
  操作名（入力 1, 入力 2, … 入力 n, 操作前状態） ==
    操作後の状態
  pre 事前条件
  post 事後条件
```

（5）事象

「事象」は, もののパラメータを与えることで操作によって事象の意味を定義している.

```
functions
  事象名: 型 1 * 型 2 * … * 型 n * 型 x -> 型 x
  事象名（パラメータ 1, パラメータ 2, … パラメータ n,
   操作前状態） ==
    操作
```

5　事例:自動販売機

ソフトウェアテスト技術振興協会（ASTER）が開催したテスト設計コンテスト'15 [*1] で用いられた自動販売機の仕様書を対象として, 前節のテンプレートを用いて VDM-SL に書き換える.

5.1　ASTER 仕様書の概要

自動販売機の仕様書は,

[*1] http://aster.or.jp/business/contest/contest2015.html

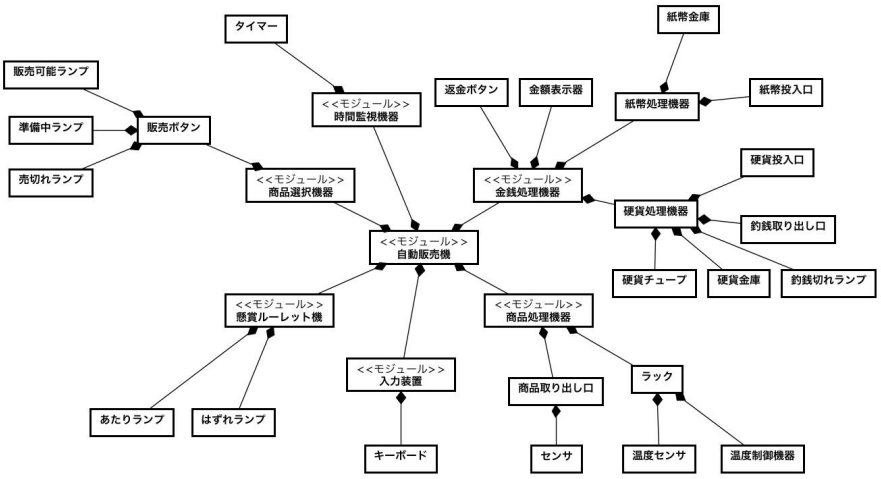

図 2 自動販売機のハードウェア構成

- 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

を対象とする．文献 [3] には，自動販売機の機器の観点から，全体の構造から具体的な数値に至るまで，その機能が詳細に記述されている．

5.2 VDM による機能記述

自動販売機の構造を機能の観点からモジュールに分割して，自動販売機の機能をモジュールで定義されている機能を用いて定義した．

5.2.1 ハードウェア構成のモジュール化

自動販売機の構成を図 2 に示すように機能の観点から 7 つのモジュールに分割した．以降では，自動販売機の主要な構成機器である

- 商品選択機器
- 商品処理機器

のモジュールの記述例を示し，自動販売機の機能を上記のモジュールを用いて記述する．

5.2.2 商品選択機器

「商品選択機器」は, 販売ボタン map によって構成されている. 販売ボタン map は, 商品の ID と販売ボタンとの写像である.「商品選択機器型」を「販売ボタン型」によって定義している. ID 型は「単純もの」で表されている.

```
types
  商品選択機器型::
    販売ボタン map: 販売ボタン map 型;

  ID 型 = token;
  販売ボタン map 型 = map ID 型 to 販売ボタン型;
```

販売ボタンには 3 種類のボタンが存在する. 次の 3 つである.

・販売可能ランプ・準備中ランプ・売切れランプ
これら 3 つのランプは以下のように定義している. 販売ボタン型は,「複合もの」としてレコード型で表されている.

```
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;
```

5.2.3 商品処理機器

「商品処理機器」はラック map と商品取り出し口によって構成されている. ラック map は, ラックの ID とラックとの写像である. 商品取り出し口にはセンサが取り付けられている.

```
商品処理機器型::
  ラック map: ラック map 型
  商品取り出し口: 商品取り出し口型;

ラック ID 型 = token 型;
ラック map 型 = map ラック ID 型 to ラック型;

商品取り出し口型::
  センサ: 検知未検知型;
```

「在庫数を減らす」は, 操作で表されている. ラックを前状態として取り, 後状態にラック内の在庫数を 1 つ減らしたものを返す.「在庫あり」は述語で表されている. ラック内の在庫数が 0 より多ければ,bool 値が「true」として返される.

```
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)
 pre 在庫あり(ラック);

 在庫あり:ラック型 -> bool
 在庫あり(ラック) == ラック.在庫数 > 0;
```

6 考察

6.1 可読性の向上と意味の明確化

仕様モデルを作成することで, 用語を 5 つの分類によって説明することが可能となった. 述語や操作に名前をつけ

ることで, 用語の意味を理解しやすくなった. また,VDM-SL で記述することで, 意味を明確にしている. それらを「販売」という操作の例を用いて説明する.

```
販売：商品型 * 自動販売機型 -> 自動販売機型
販売（商品，自動販売機） ==
  let
   販売後 =
     mu（自動販売機，
      商品 map  |->  商品 map の在庫更新（商品，
        -1，自動販売機．商品 map），
      残高  |->  自動販売機．残高 -
        自動販売機．商品（商品）．価格）
    in
      if  販売可能な商品がある（販売後）
      then
        mu（販売後，
          出力  |->  mk_出力型（商品，nil））
      else
        mu（販売額を格納（販売後），
          出力  |->  mk_出力型（商品，
            販売後．残高））
pre 販売可能（商品，自動販売機）；
```

「商品 map の在庫更新」は, 操作を表す.「販売可能な商品がある」は述語を表す. また,「販売可能な商品がある」の定義を以下のように示すことで, 用語の意味を明確にしている. 例を用いて解説する.

```
販売可能な商品がある:自動販売機型 -> bool
販売可能な商品がある（自動販売機） ==
  exists 商品 in set 自動販売機．商品集合 &
    販売可能（商品，自動販売機）;
```

6.2 記述されていない仕様の分析

仕様モデルを用いて,VDM-SL にそのまま記述することができなかったものが存在した. 以下に記述できなかったものと, それに対しての例を示す.

・時間
- （貨幣投入タイムアウト）紙幣，硬貨とも，投入後 10 分間何も操作が行われなかった場合，自動返金する

・ハードウェア
- 本製品では 30 個のラック（10 商品× 3 段まで販売可能）がある

・制御
- （商品取り出し口）商品取り出し口センサは独立した CPU で制御する

・詳細
- （金銭表示機）貨幣の受け付け，もしくは，貨幣返金の都度計算，表示する

これにより, 仕様モデルを用いて,VDM-SL に記述することができないものが明確になった. これを仕様モデルに加えることで, より良い仕様モデルが作れると考える.

6.3 抽象モデルの必要性

VDM-SL 記述について分析した結果, 全体を把握することが困難であるということが挙げられた. そこで VDM-SL 記述の抽象モデルを作成することで, その問題を解決することを考える. 以下はハードウェアの概念が書かれた記述である.

```
types
  自動販売機型::
    商品集合： 商品集合型
    商品 map:商品 map 型
    預かり金： 金額型
    残高： 金額型
    金庫： 金額型
    販売ボタン集合： 販売ボタン集合型
    販売ボタン map： 販売ボタン map 型
    商品機器： 商品機器型
    金銭機器： 金銭機器型
    販売時間： 時間集合型
    現在時刻： 時刻型
    出力： [出力型];
```

以下はハードウェアの概念を取り除いた抽象モデルである.

```
types
  自動販売機型::
    商品集合： 商品集合型
    商品 map:商品 map 型
    預かり金： 金額型
    残高： 金額型
    金庫： 金額型
    販売時間： 時間集合型
    現在時刻： 時刻型
    出力： [出力型];
```

「販売ボタン集合」「販売ボタン map」「商品機器」「金銭機器」はそれぞれハードウェアの概念である. ハードウェアの概念を取り除き, 抽象モデルを作成することで, 全体を把握することを容易にすることができる. また抽象モデルを具体的にしていくことで,VDM-SL 記述の指針になると考える.

7 おわりに

本研究では, 自然言語の仕様書から VDM-SL を用いた機能仕様書を作成するための仕様モデルを提案して作成指針を考察した. その結果,VDM-SL 記述の可読性が向上し, 用語の意味も明確にすることができた. 今後の課題は仕様モデルを用いて,VDM-SL にそのまま記述することができなかったものを仕様モデルに加え, より良い仕様モデルを作成することである.

参考文献

[1] 荒木啓二郎, 張漢明, プログラム仕様記述論, オーム社,2002.

[2] 玉井哲雄, ソフトウェア工学の基礎 , 岩波書店,2005.

[3] 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,
http://aster.or.jp/business/contest/doc/2015tdc-v1

[4] 岩田陽平, 岩瀬拓也，“VDM を用いた機能仕様記述に関する研究,”南山大学 2015 年度卒業論文,2016.