VDM を用いたシーケンス図の妥当性確認に関する考察 ―研究室名簿システムを事例として―

2017se010 原拓海
指導教員：張漢明

1　はじめに

ソフトウェア開発において，仕様は開発対象の妥当性確認と正当性検証で，重要な役割を担っている．妥当性確認においては，要求を基準にして，その基準を満たしているかの判断材料になる [1]．妥当性を確認するためには実行することが有効である．UML 記述は一般的に実行可能な言語ではない．VDM は実行可能な形式仕様言語である．本研究の目的は，UML ダイヤグラムのシーケンス図の妥当性確認を支援する技術を考察することである．UML のシーケンス図を対象としてシーケンス図に対応した VDM を記述することにより，仕様段階でのシーケンス図の妥当性確認を行うことができる．本研究の研究課題は，

1. シーケンス図を実行するために必要な情報は何か
2. VDN-SL 記述を支援するために必要な技術は何か

を明らかにすることである．本研究では，研究室名簿システムを事例に用いて，UML ダイヤグラのシーケンス図，クラス図と，VDM-SL 記述の間の，構文レベルと意味レベルの関係を考察する．構文レベルの関係では，

- UML 記述から形式的な VDM-SL 記述への変換

意味レベルの関係では，

- UML で記述されていないものは何か

の観点から考察する．

2　既存技術

2.1　シーケンス図

シーケンス図は「クラスやオブジェクト間のメッセージやり取り」を「時間軸に沿って」，図で表現する．シーケンス図を用いることにより，複雑なシステム処理の流れが視覚的になり，プログラム概要および処理手順について，早く正確に理解することができる [2].

2.2　VDM-SL

VDM-SL は実行可能な形式仕様言語の一つである．VDM-SL には，以下の 2 種類の使い方がある [3].

・関数スタイル

システムの機能を，関数で記述することで定義する．

・状態に基づくモデリング

操作の実行前後の状態と入力の関係を
記述することで，システムの機能を定義する．

さらに，上記の二つの使い方において，それぞれに以下の 2 種類の書き方がある．

・陰スタイル

関数や操作を入力出力と事前・事後条件を記述することで定義する．

・陽スタイル

状態を定義して状態の変化を操作として記述することで定義する．

本研究では「陽スタイルによる状態に基づくモデリング」を用いる．

3　事例:研究室名簿システム

3.1　研究室名簿システムの概要

研究室名簿システムは，研究室所属構成員を名簿に記録し，その名簿に対して，何らかの機能を有するシステムである．本要旨では機能「名簿登録」を例として取り上げる．

3.2　UML 記述

3.2.1　クラス図

研究室名簿システムのクラス図を**図 1** に示す．

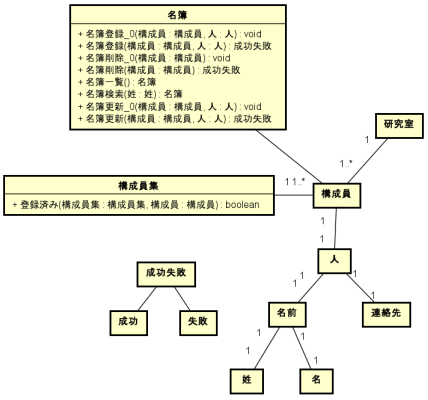

図 1　研究室名簿システムのクラス図

3.2.2　シーケンス図

名簿登録の処理の流れを記したシーケンス図を**図 2** に示す．

1. 利用者は，名簿に対し，メッセージ「名簿登録」を送る．
2. 1 のメッセージをきっかけにして，名簿は，構成員集に対して，メッセージ「登録済み」を送り，登録したい人が名簿に未登録かを調べる．
3. (a) すでに登録されていれば何も実行しない．
   (b) 未登録であれば，名簿は自身にメッセージ「名簿登録_0」を送り，名簿に構成員を登録する．
4. 名簿は登録が成功したがどうかを利用者に送信する．

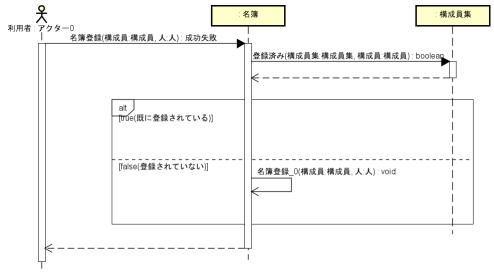

図 2 名簿登録のシーケンス図

3.3 VDM-SL 記述

UML 記述を基にした，VDM-SL 記述例を示す．

```
types
     構成員型 = token;
     人型 :: 名前 : 名前型 連絡先 : 連絡先型;
     名前型 :: 姓 : 文字列型 名 : 文字列型;
     連絡先型 = token;
     文字列型 = seq of char;
     名簿型 = inmap 構成員型 to 人型;
     構成員集型 = set of 構成員型;
      成功失敗型 = <成功> | <失敗>;

state 研究室 of
    名簿 : 名簿型
    itr : 名簿型
init s == s = mk_研究室 ({|->}, {|->})
end

operations
    名簿登録 : 構成員型 * 人型 ==> 成功失敗型
    名簿登録（構成員, 人）==
    if 登録済み（dom(名簿), 構成員）
    then return <失敗>
    else （名簿登録_0(構成員, 人);
          return <成功>);

    名簿登録_0 : 構成員型 * 人型 ==> ()
    名簿登録_0(構成員, 人）==
    名簿 := 名簿 munion {構成員 |-> 人}
    pre  構成員 not in set dom 研究室.名簿;

functions
    登録済み:構成員集型 * 構成員型 +> bool
    登録済み（構成員集, 構成員)==
    if 構成員 in set 構成員集
    then true
    else false;
```

上記の記述では，型を type，状態を state，操作を operations, 関数を functions でをれぞれ表現している．

4 考察

4.1 構文レベルの関係

クラス図

クラス図からは，クラス名，関連などが構文レベルの変換が可能であった．**表 1** は一部をまとめたものである．

表 1 クラス図における構文レベルの対応関係

クラス図		VDM-SL
クラス名	→	型の名前
関連 (1 対 1)	→	レコード型
操作名	→	操作 or 関数の名前

シーケンス図

シーケンス図から構文的に変換できる VDM 記述を以下に示す．

```
operations
    名簿登録 : 構成員型 * 人型 ==> 成功失敗型
    名簿登録（構成員, 人）==
    if 登録済み（???, 構成員）
    then ???
    else （名簿登録_0(構成員, 人);
          ???);

    名簿登録_0 : 構成員型 * 人型 ==> ()
    名簿登録_0(構成員, 人）==???

functions
    登録済み:構成員集型 * 構成員型 +> bool
    登録済み（構成員集, 構成員)==???
```

上記の ???は構文の変換からは得られない部分である．

4.2 意味レベルの関係

4.2.1 UML で記述されていないもの

UML は構文レベルの規定であるとみなすとクラスは型に対応する．抽象度の高い段階ではまだ決定していない型が存在する．例えば，構造を規定していないクラスや関連はまだ決まっていない．意味レベルの記述では，未決定のクラスの型に適切な型を割り当てる必要がある．VDM の記述は，自然言語で記述されているの操作の表現と，割り当てられた型に対する表現の間に対応関係があると考えられる．

4.2.2 記述支援方法

VDM 記述では，基本型にはその型に対する操作が定義されている．言語マニュアルにはその集合的な定義は示されているが，その利用方法は十分には説明されていない．そこで，支援方法の一つとして，自然言語に対する陽記述の記述方法をパターン化し，それらを体系化した，用語集を作成することが考えられる．例えば，「写像にデータを追加する」という表現は，**写像 munion {要素 A |-> 要素 B}** という記述になる．写像に領域の要素 A と 値域の要素 B を加えるときは，データ追加するときは要素 A は写像に含まれていないという事前条件がある．形式化で自然言語で表現されている暗黙知が表現される．

5 おわりに

本研究では，シーケンス図とクラス図から VDM-SL 記述を作成する上で, 変換可能な構文レベルの要素, UML には未記述な意味レベルの要素を明らかにした．また，意味レベルの記述に対する支援方法を考案した．今後の課題として，構文レベルでは，UML 記述から変換できる構造物の拡大，意味レベルではパターン集の蓄積が求められる．

参考文献

[1] 玉井哲雄：ソフトウェア工学の基礎 改訂新版，岩波書店，2022.

[2] Martin Fowler：UML モデリングエッセンス第 3 版，翔泳社，2005.

[3] 荒木啓二郎，張漢明：プログラム仕様記述論，オーム社，2002.