

要求仕様記述における形式手法の適用に関する研究

2000MT026 井上 真由美

指導教員 野呂 昌満

1 はじめに

ソフトウェア開発を手戻りなく進めるためには、仕様書、設計書を矛盾なく記述することが重要である。仕様書、設計書に矛盾があると依頼者と開発者との間に理解の違いが生じ、開発されたソフトウェアが要求されたソフトウェアと異なってしまう可能性がある。オブジェクト指向開発における一般的な仕様の記述には UML[2] を用いる方法がある。UML を用いると直観的には理解しやすいが、解釈が一意に決まらず、矛盾を生じやすい。

本研究の目的は、UML を用いて記述した図と図の間の整合性を検証する方法を提案することである。UML に形式手法を用いて記述する方法を定義し、整合性の検証をおこなう。本研究では、UML の中でもユースケース図、クラス図、シーケンス図の三つの図に着目して考える。この三つの図を選んだ理由は、ユースケース図はシステムの機能を外部から記述しており、本研究の基盤となる図であると考えた。システムを内部の視点からクラスとクラス間の関係を記述したクラス図とオブジェクト間のメッセージのやりとりを記述したシーケンス図に矛盾が生じやすいと考えたからである。ユースケース図を形式手法を用いて記述する方法を定義する。形式仕様記述言語の中でも、VDM-SL[1] と CSP[3] を用いる。VDM-SL はシステムの機能を記述し、状態の変化をさせる仕様記述に適しており、CPS は機能の順序を記述できるという理由から VDM-SL と CSP を取り上げる。

2 要求仕様記述

要求仕様記述はシステムを実現するさいにどのような機能が要求されているかを記述したものである。機能の一覧を表すユースケース図と各機能の内容を表すユースケースを用いて記述する。ユースケース図は利用者の視点に着目して記述されるのでユーザのマニュアルに記載される一般的な機能がユースケースとして定義される。

ユースケース図

ユースケース記述はユースケースの本質を以下の通りと考え記述方法を定義する。

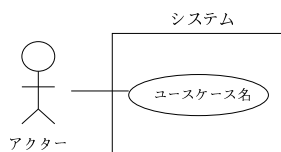


図1 ユースケース図

- ユースケース 1 : ユースケース名
 - 概要 ユースケースの概要
 - アクター ユースケースのアクター
 - 動作詳細 機能の詳細
 - 事前条件 ユースケースを実行するうえで必要な条件

3 形式記述の方法

ユースケースを VDM-SL, CSP の二つの形式仕様記述言語の適用についての方法を定義する。具体例として以下のユースケースを記述する。

- ユースケース 1 : 照明をつける
 - 概要 照明をつける
 - アクター 利用者
 - 動作詳細 スイッチを ON にして照明がつく
 - 事前条件 スイッチが OFF である

3.1 CSP を用いた記述方法

動作詳細から各動作を抽出し、イベント名として記述する。イベント名は分かりやすく「_」で単語ごとに区切って記述する。

ユースケースの動作詳細に記述された順序でイベントの間に「_」を記述し、動作の最後には正常終了の意味である SKIP を記述する。

具体例

動作詳細から各動作である「スイッチを ON にする」「照明がつく」をイベントとして記述する。ユースケース名の「照明をつける」に各動作を動作詳細に記述された順序で記述する。CSP 記述は以下ようになる。

```
channel on_switch, on_light
PUT_ON_LIGHT = on_switch -> on_light -> SKIP
```

図2 CSP 記述

3.2 VDM-SL を用いた記述方法

CSP と同様に、動作詳細から各動作を抽出し、operation 名として記述する。ユースケースは内部の記述をおこなわないので各動作の詳細は記述しない。ユースケース名も operation 名として、動作詳細の順序で「(A(); B(); C());」と operation の本体に記述する。事前条件がある場合には「pre」で記述する。

具体例

動作詳細から各動作である「スイッチを ON にする」「照明がつく」を operation 名で記述する。ユースケース名の「照明をつける」に各動作と事前条件を記述す

る.VDM-SL 記述は以下ようになる.

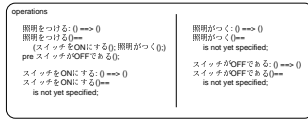


図 3 VDM-SL 記述

4 整合性の検証

ユースケースとクラス図, ユースケースとシーケンス図の二つの整合性の検証をおこなう. システムの内部と外部からの違う視点から記述された図と図の間の検証をおこなうので図と図の間の対応を考える必要がある.

4.1 ユースケースとクラス図の間の検証

VDM-SL を用いたユースケース記述の関数本体に記述された各 operation の A(),B(),C() がクラス図に記述されているかどうかを調べる. 事前条件がある場合, クラス図にも同様の条件が記述されているかどうかを調べる. 以上の検証項目が満たされていればユースケースとクラス図の間に機能要求の矛盾はないと言える.

具体例

具体例を用いて検証をおこなう.

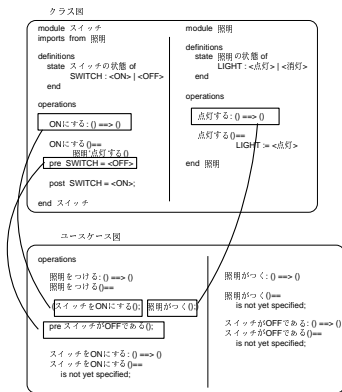


図 4 ユースケースとクラス図の対応

クラス図のモジュール名,operation 名,operation の本体からユースケースの operation 名との対応を検出する. ユースケースの operation の本体の「スイッチを ON にする」,「照明がつく」はクラス図のスイッチクラスの「ON にする」,照明クラスの「点灯する」に記述されている. 事前条件の「スイッチが OFF である」はクラス図のスイッチクラスの「ON にする」の事前条件で SWITCH = <OFF>と書かれている. 以上のことからユースケースとクラス図の間に機能要求の矛盾はないと言える.

4.2 ユースケースとシーケンス図の間の検証

CSP を用いたユースケースのイベントの順序とシーケンス図のメッセージのやりとりの順序に誤りがないかを調べる. 順序に誤りがなければユースケースとシーケンス図の間に機能要求の矛盾はないと言える.

具体例

具体例を用いて検証をおこなう.

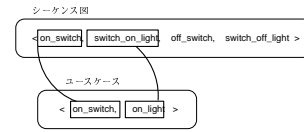


図 5 ユースケースとシーケンス図の対応

ユースケース記述の「on_light」はスイッチから light へのメッセージであるので,シーケンス図の「switch_on_light」に対応している. ユースケースとシーケンス図に順序の誤りはない. ユースケースとシーケンス図の間に機能要求の矛盾はないと言える.

5 考察

形式手法を用いた記述に関する考察

動作詳細を各動作ごとに抽出して記述するので機能が整理される. VDM-SL では,各動作の詳細は記述しない. ツールが整っており日本語の記述が可能である.CSP では,機能の順序を記述する. 以上のことから形式手法を用いて記述することで理解しやすい仕様を記述することができたと考える.

整合性の検証に関する考察

ユースケースはソフトウェア開発の分析段階で用いて設計段階で機能要求に矛盾がないかを検証することは一般的に活用されている. 実装段階で再度,機能要求に矛盾がないか検証することで開発の手戻りを削減させることが可能になると考える. 図によって記述する視点が異なるという UML の問題点は,図と図の対応を検出することにより解決されると考える.

6 おわりに

本研究では,ユースケースとクラス図,シーケンス図の間の整合性の検証をおこなった.UML の図と図の間の整合性の検証をするために形式手法を用いたユースケースの記述方法について定義した. 今後の課題として,具体的事例を用いて整合性の検証をおこなうことが挙げられる.

謝辞

本研究を進めるにあたり,2年間ご指導をいただいた野呂昌満教授,有益なアドバイスをいただいた張漢明助教授,蜂巢吉成講師,熊崎敦司さん,藤原泰昌さん,森貴彦さん,後藤修平さんに深く感謝いたします.

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] 渡辺博之,渡辺政彦,堀松和人,渡守武和記:組み込みUML,翔泳社(2002)
- [3] C.A.R.Hoare: Communicating Sequential Processes, Prentice-Hall International(1985)