

# 仕様記述言語 VDM++ を用いた 制限付きオブジェクト予約システムの記述

## Description of Restricted Object Reservation System using Specification and Description Language VDM++

牧田蒼斗† 和崎克己††  
Aoto Makita Katsumi Wasaki

### 1 はじめに

システム設計において、仕様を定義することはもっとも重要な工程である。現在主流である自然言語による仕様記述には、仕様自体の曖昧さ、欠陥発見の遅れによるコストの増大など多数の問題がある。そういった問題を取り除く有用な方法として、形式手法がある。

本研究では、UML 図を作成し、オブジェクトに期限・制約がかかった予約システムの関係性について考え、仕様記述言語の一種である VDM++ を用いてシステムを記述する。

### 2 形式仕様記述言語 VDM++ の概要

形式手法の一つである VDM とは、Vienna Development Method(ウィーン開発手法)の略であり、1970 年代中頃にウィーンの IBM 開発研究所が開発したモデルベースの形式手法である。この VDM 手法の基盤となる仕様記述言語 VDM-SL にオブジェクト指向拡張を行った言語を VDM++ と呼ぶ。不変条件・事前条件・事後条件・陰仕様記述などの形式手法を支援する機能だけでなく、クラス・多重継承といったオブジェクト指向プログラミングおよびパターン・内包表記・型変数・高階関数などの関数型プログラミング機能も持っている [1]。設計工程中で、VDM++ のようなモデリング言語を利用することにより、必要な機能性についての詳細な記述を行うことが可能になる。なお本研究でのクラス設計における記述は、文献 [2] の手法を参考とし、OvertureTool[3] を用いて実施した。

### 3 上位設計例

今回作成した「制限付きオブジェクト予約システム」のクラス関連図を図 1 に示す。

#### 3.1 システムの概要

本研究では、現在の数多くある予約システムの中で、期限や制約が付いた「制限付きオブジェクト」をもつ予約システムについて記述する。

「制限付きオブジェクト」とは、具体的には、災害時に支給される物資や、感染症対策のワクチンのことである。ホテルの予約や店の予約などの簡易的な予約システムは、システムに何らかの不具合が起きてもクリティカルな問題が起こる可能性は低い。しかし、ワクチン接種予約を例に挙げると、ワクチンの接種回数や接種するワ

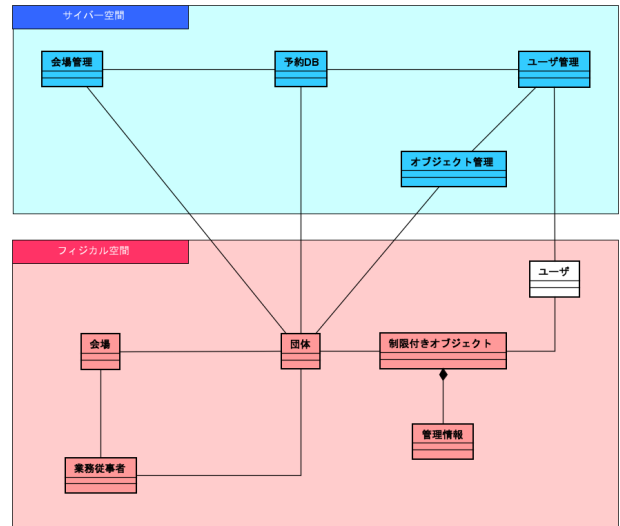


図 1 サイバー空間とフィジカル空間を結合した予約システムのクラス関連図

クチンの種類には制約があり、打ち手が必要となる。また、ワクチンには適温での保存や使用期限という制約が存在する。この制約下で先程述べた予約システムと同じようなシステムを作成したとすると、保存条件が不適切なワクチンを接種する可能性や、接種回数を誤り、健康に被害が及ぶなど、命に関わるクリティカルな問題に発展しかねない。この問題を防ぐため、VDM++ でオブジェクトの制約を仕様記述し、最適なシステムについて考える。

#### 3.2 要求仕様

今回記述した「制限付きオブジェクト予約システム」で満たされるべき要求は以下である。

- 十分な人数の業務従事者がいること
- 会場の利用可能時間内でイベントが行われること
- オブジェクトの制約を遵守すること

#### 3.3 UML 図

図 2 は、図 1 のクラス図をさらに詳細化したものである。図 1 より、サイバー空間内には「ユーザー管理」、「予約 DB」、「会場管理」、「オブジェクト管理」のクラスが存在する。この 4 つをサイバー空間の機能としてまとめるため、UML 設計ツール astah\*[4] の機能のパッケージ化を用いて、サイバー空間としてパッケージ化し、二重の構造化を図った。

† 信州大学大学院総合理工学研究科, Graduate School of Science and Technology, Shinshu University

†† 信州大学工学部電子情報システム工学科, Department of Electrical and Computer Engineering, Faculty of Engineering, Shinshu University

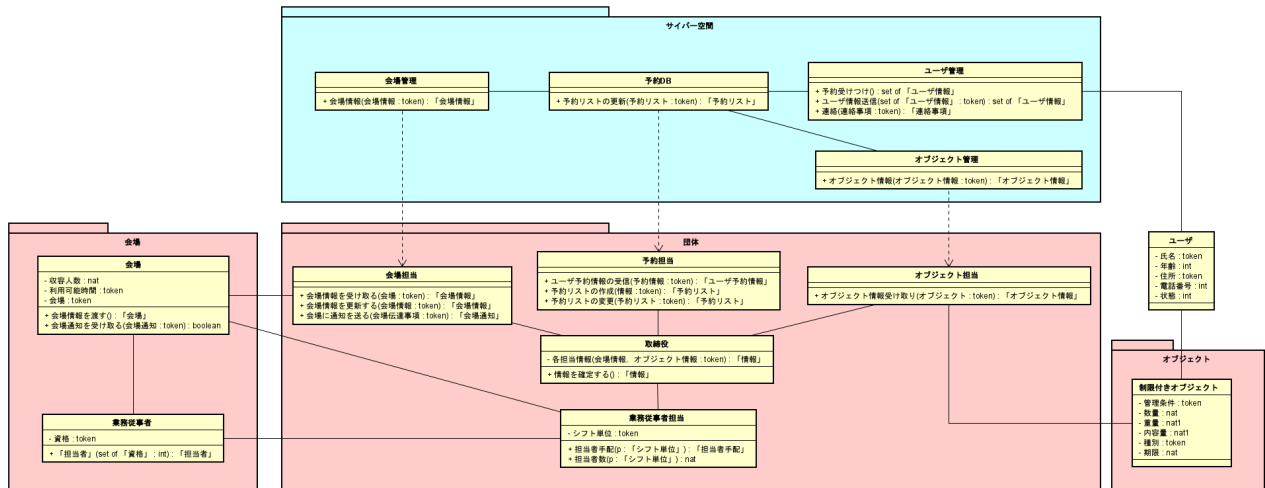


図 2 役割別に詳細化・パッケージ化した合成構造図

同様にフィジカル空間内でもパッケージ化を行い、団体、会場、オブジェクトの 3 つのパッケージに分けた。パッケージごとの説明は以下である。

- 会場パッケージでは、会場の情報を持つ「会場」クラスと、会場で業務をする「業務従事者」クラスがある
- オブジェクトパッケージには、「制限付きオブジェクト」クラスがある
- 団体パッケージとは、イベントの実行権を持ち、オブジェクトを提供する運営をまとめたものであり、「予約担当」クラス、「会場担当」クラス、「オブジェクト担当」クラス、「業務従事者担当」クラス、「取締役」クラスの 5 つのクラスがある。別のパッケージとの情報のやり取りを「取締役」クラスに渡すようになっている

このようにパッケージ化をすることで、パッケージ内のつながりや、各パッケージとのやり取りのインターフェースが見やすくなり、不備なども手早く見つけることができる。

#### 4 各機能とオブジェクトの VDM++ 記述

図 2 の UML 図の「取締役」クラスの関数定義部分の記述を図 3 に示す。図 3 は「取締役」クラスの関数の 1 つであり、予約の時間と人数が会場の利用可能時間内かつ収容可能人数内であるか判定する記述である。bool 型を返すことで、この関数でイベント実行を判断する。

次に図 2 の UML 図の「オブジェクト」クラスの記述を図 4 に示す。オブジェクトには制約を付けるため、型定義に合併型を用いて typeA, typeB と記述した。また今回、定数定義は適当な値を入れたが、期限はオブジェクトの使用期限の残り日数を表している。

#### 5 まとめと今後の課題

「制限付きオブジェクト予約システム」のクラス図を作成し、それに基づき VDM++ で記述した。VDM++ の記述をする前に、クラス図を作成し、さらに構造化することで、各クラス間のインターフェースが明確化された。そのため、システムの全体像が掴め、記述がしやす

```

functions
public イベント判定 : 予約リスト * set of 時間帯 -> bool
イベント判定(予約リスト,時間帯) ==
    予約リスト'時間帯 in set 時間帯'時間帯
    and
    予約リスト'人数 <= 時間帯'人数;
    
```

図 3 「取締役」クラス関数定義部分

```

class オブジェクト

types
public 「数量」 = nat;
public 「内容量」 = nat;
public 「種別」 = <typeA>|<typeB>;
public 「期限」 = nat1;
public 「管理条件」 = token;

values
数量:「数量」 = 1000;
内容量:「内容量」 = 5;
期限:「期限」 = 100;

end オブジェクト
    
```

図 4 「オブジェクト」クラス

くなった。

今後の課題は、上位設計からシステムの運用マニュアルを作成するために、どのような手法を用いれば良いか考えることである。上位設計から運用マニュアルを作成できれば、運用上のミスを極力防ぐことができ、品質向上につながる。

#### 参考文献

- [1] ジョン・フィッツジェラルド, 他: VDM++ によるオブジェクト指向システムの高品質設計と検証, 翔泳社 (2011)
- [2] 佐原伸: 事務処理用 VDM++ フレームワークの作成, 日本フィッツ株式会社情報技術研究所 (2004)
- [3] OvertureTool: <https://www.overturetool.org/>
- [4] 株式会社チェンジビジョン: <https://astah.change-vision.com/ja/>