

# モデル開発を支援するプロダクト管理システムの設計

5K-10

堀江 奈緒子

上田 賀一

茨城大学工学部情報工学科

## 1 はじめに

ソフトウェアのライフサイクルの工程で生成され、後工程に必要とされる情報をまとめて管理するソフトウェアリポジトリの考えを基礎として、モデル開発環境において生成されるモデルを管理するものを考える。ソフトウェアリポジトリとは、ソフトウェアに関する様々な情報を目的に適合する記法で表現・格納し、その変更を制御し、利用・維持することを助ける汎用の情報システムである。すなわち、ソフトウェア開発に固有の情報を管理するためのデータベースであるともいえるので、その構成要素は(1)情報モデル、(2)情報ベース、(3)情報ベース管理システム、(4)制御機能の4つの視点から分類できる[1]。

ここで考えるモデル開発工程ではメタモデルに基づくモデリング工程の連続として構成され、メタモデルの設計・構築も許すものとなっている。多数のユーザにより様々なモデルが生成される環境となっており、これらを結びつける様々な関係も扱わなければならない。本研究ではこれらの実体や関係を明確にし、それらを取り扱うためのプロダクト管理システムを検討した。

今回、このプロダクト管理システムの分析・設計について報告する。

## 2 プロダクト管理システム

プロダクト管理システムの目的とその概要を述べる。

### 2.1 システムの目的

モデルを管理・運用するためのシステムを設計するに当たってモデルの保守性を高めることおよびモデルの再利用性を高めることを主な目的としている。保守とはモデルの不当な扱いに対する防御策と考え、モデルの実体をシステムが管理して他のユーザのモデルアクセスを受け付けなくにするなどを考えている。ユーザがモデルを容易に再利用できるように、その情報の詳細を示すことにしたい。

Design of Product Management System for Supporting Model Development

Naoko Horie, Yosikazu Ueda

Department of Computer and Information Sciences, Faculty of Engineering, Ibaraki University

### 2.2 システムの検討

様々なツールから生成される様々なモデルに対応するシステムを検討し、その要求を満たすべく要求記述を書き出した。以下にその要約を述べる。

- 各モデルの設計者は一人とは限らないので、モデルごとのグループにまとめ、各ユーザは一つ以上のグループに存在する。
- ユーザID、グループID、プロジェクトIDを設定する。識別にはそのIDを用いる。
- 各モデルを生成する手段として用いられたツールも同様に管理する必要がある。
- 機能として更新、参照、追加(新規登録)、更新、削除などが考えられる。ただし、更新、参照、追加(新規登録)については全ユーザにとって使用可能である必要があり、更新、削除については作成した本人もしくはその本人を含むグループ内のユーザのみに限られたものでなければならない。
- 各ユーザが生成したモデルの登録の際、その登録日、ユーザID、グループID、モデル名、キーワード、コメント、関連ファイル名を記入できるようにし、それをそのモデルの情報とする。
- モデルの検索の際、各ユーザが記入したモデル名、ユーザID、グループID、キーワードを利用できるようにする。
- 同じグループの二人以上のユーザが同時に同じモデルの編集を行う場合が考えられる。そのため先に作業を行っている者以外は受け付けないようにするなどの制御が必要となる。
- モデル(ツール)名の重複はディレクトリもしくはユーザが異なる場合のみ許される。すなわち同名のファイルはそのディレクトリIDまたはユーザIDで区別する。
- モデルのバージョン管理やバックアップ処理などを行う。
- 管理の方法としては、モデルの実体そのものを一元管理で扱うようにする。

以上を要求記述とし、これに対する要求分析を行う。

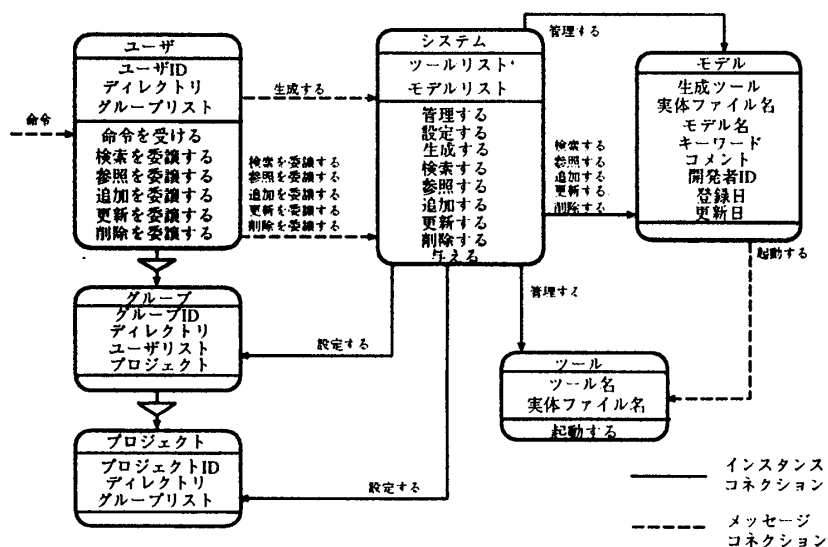


図 1. オブジェクト構造図

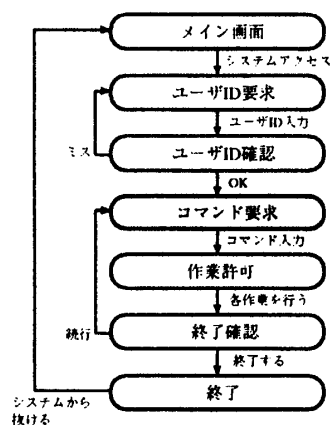


図 2. ユーザインタフェースの状態遷移図

### 3 オブジェクト指向分析・設計

自然言語からの記述というものは、たいてい曖昧であったり不完全であったりする。その記述を分析した分析モデルは明確で簡潔な要求表現を作り上げる。そこで、先に述べた要求記述を分析して要求仕様書を作成した。

Coad & Yourdon 手法 [2] に従い行った分析手順を以下に示す。

1. 要求記述からの単文の抽出
2. 名詞句・動詞句の抽出
3. クラス・属性候補の抽出
4. クラス・属性候補の検証
5. サービスの定義
6. 関係の定義

これらの作業を数回繰り返した。その後、設計として、

7. ユーザインタフェース関係、データ管理、タスク管理の定義

を行い、図 1 に示すオブジェクト構造図を作成した。

更に、この構造図上のオブジェクト「ユーザ」はユーザインタフェースの機能をサポートする必要がある、これを動的モデルである状態遷移図により図 2 に示す。

### 4 分析・設計法の使用感

オブジェクト指向分析法には幾つかのものがあるが、基本的には自然言語の要求記述からの名詞句、動詞句の抽出から始まり、この作業は困難なく行える。ところがこれらと、クラス、属性、メソッドといった概念に対応づけることは容易に判断できるものと困

難なものに明確に区別できるように感じられた。ある程度の判断指針が与えられているが、経験を積んでいないと判断しかねる指針であると感じられる。

今回取り上げた Coad & Yourdon 手法は手戻りの作業は多いが、成果物として主要なオブジェクト構造図と補足的なオブジェクト状態図、サービスチャートしかないため、利用者にとってはまとめ易いともいえるが、あらゆる情報をこれらにまとめなければならない。しかし、成果物が限定され基本となる概念が少ないため、設計段階や実装段階へのつながりがスムーズであると言える。

### 5 おわりに

今回、プロダクト管理システムについての要求記述とその分析、および設計について触れた。本研究では Coad & Yourdon 手法によるオブジェクト指向分析に主眼をおき開発に取り組んだ。その結果、設計への移行も容易にできたと考えられる。今後、実装に取り組む予定であるが、Coad & Yourdon 手法を用いたので比較的容易にできるものと思われる。

### 参考文献

- [1] 落水 浩一郎：ソフトウェア・レポジトリ、情報処理、Vol.35, No.2, pp.140-149 (1994)
- [2] 本位田 真一、山城 明宏：オブジェクト指向システム開発、日経 BP 社 (1993)