

フィーチャーモデリングとデータモデリングを統合したプロダクトライン開発手法

†鈴木 教之 †位野木 万里

‡東芝ソリューション株式会社

1. はじめに

ソフトウェア開発の現場では、開発コストの削減、リードタイムの短縮などを実現するために、プロダクトライン型開発が注目されている^[1]。プロダクトライン型開発では、あらかじめプロダクトラインの資産となるコアアセットを開発し、このコアアセットを組み合わせてプロダクトを開発する。

コアアセットの開発では、プロダクトの共通部分と可変部分の分析をフィーチャーモデリングで行う^{[2][3]}。ドメイン分析で得られた共通部分と可変部分を含む分析モデルを、ドメイン設計で設計モデルに反映する。ただ、どのようにして分析モデルを設計モデルに反映するかという手法が確立されておらず課題となっている。

本稿ではこの課題に対し、従来から利用されているフィーチャーモデルに加え、ユースケースモデル、データモデルの3種類のモデルから、ルールに従って設計モデルの要素を導出し、要素を統合することで、設計モデルを作成する手法を提案する。

2. 対象ドメインとアーキテクチャ

本稿で提案する手法が対象とするドメインを、画面や GUI を介したデータの登録管理システムのアプリケーションと定義し、アプリケーションの構築環境を JavaEE^[4] と定義する。

また、アプリケーションのクラス構造のアーキテクチャを図1のように定義する。

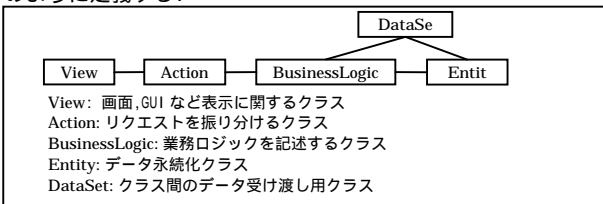


図1 ドメインのクラス構造

3. プロダクトライン開発手法

提案する手法は、ドメイン分析とドメイン設計の手法で構成する。

3.1 ドメイン分析

ドメイン分析では、ドメインの特徴、ドメインの使われ方、ドメインのデータの3つの観点から、次の(1)~(3)の技術により共通部分と可変部分を分析する。

(1) フィーチャーモデリング

ドメインを特徴の観点から共通部分と可変部分の分析を行い、フィーチャーモデルを作成する。

(2) ユースケースモデリング

ドメインを使われ方の観点から分析を行い、ユースケースモデルとユースケースシナリオを作成する。

(3) データモデリング

ドメインをデータの観点から分析を行い、データモデルを作成する。分析にデータ中心アプローチ(Data Oriented Approach: DOA)の手法を用いることで、ドメイン特有のデータの特徴が一気に抽出できる。

3.2 ドメイン設計

ドメイン設計では、ドメイン分析で作成した3種類のモデルに、次の5つのルールを適用して設計モデルの要素を導出する。

3.2.1 フィーチャーモデルからの設計モデル導出

フィーチャーモデルに以下のルールを適用し、設計モデルの要素を導出する。

ルール：可変フィーチャーを束ねるフィーチャーから BusinessLogic クラスを導出(図2)。

ルール：可変フィーチャーからメソッドを導出(図2)。

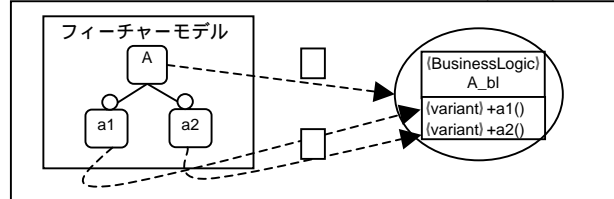


図2 フィーチャーモデルから導出した要素

3.2.2 ユースケースモデルからの設計モデル導出

ユースケースモデルに以下のルールを適用し、設計モデルの要素を導出する。

ルール：ユースケースで定義した画面数分の View クラスと、1個の Action クラス、1個の BusinessLogic クラス、1個の DataSet クラス、1個の Entity クラスから構成されるクラス群を導出(図3)。

ルールで導出した BusinessLogic クラス、DataSet クラス、Entity クラスは、フィーチャーモデルとデータモデルから導出した設計モデルの要素を統合するための中間的なクラスとする。

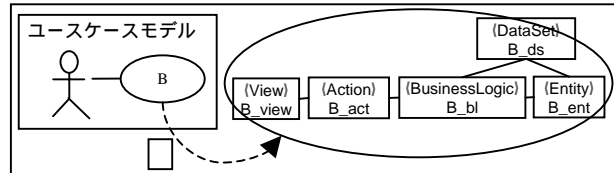


図3 ユースケースモデルから導出した要素

3.2.3 データモデルからの設計モデル導出

データモデルに以下のルールを適用し、設計モデルの要素を導出する。

ルール：各エンティティから DataSet クラスと Entity クラスを1つずつ導出(図4 -1, -2)。

ルール：親エンティティから BusinessLogic クラスを導出し、そのクラスにデータ処理に必要な登録、参照、更新、削除の4つの操作を加える(図4)。

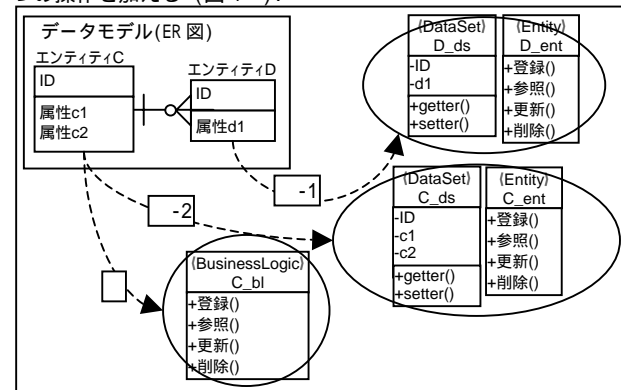


図4 データモデルから導出した要素

Product Line Development Method by Integrating Feature Modeling and Data Modeling.

†Noriyuki Suzuki, Mari Inoki

‡Toshiba Solutions Corporation

3.2.4 クラス構造へのマッピング

3.2.2 でユースケースモデルから導出したクラス群の中間的なクラスに対して、3.2.1 と、3.2.3 で導出した要素をマッピングし、設計モデルを作成する(図 5)。

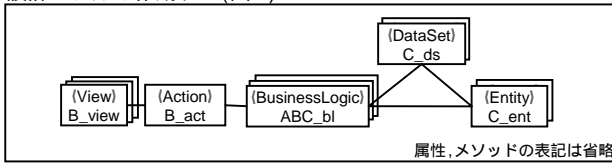


図 5 導出した要素から作成した設計モデル

4. 適用例

販売した製品に対する顧客の意見を登録管理する『ご意見管理システム』に本手法を適用し、ドメイン分析結果の各モデル(図 6(1)(2)(3))から設計モデルの要素を導出、統合し、設計モデル(図 6(4))を得た。

4.1 ドメイン設計

- ・フィーチャーモデルからの設計モデル導出
図 6(1)のフィーチャーモデルの「情報共有」フィーチャーに対して、3.2.1 で示したルール を適用し、図 6(4)(e)のもととなるクラスを導出し、「一括対応」フィーチャーと「逐次対応」フィーチャーにルール を適用し、「一括対応」メソッドと「逐次対応」メソッドを導出した。
- ・ユースケースモデルからの設計モデル導出
図 6(2)のユースケースモデルの「ご意見管理」ユースケースに対して、3.2.2 で示したルール を適用し図 6(4)の(a)(b)(c)(d)と、中間的な BusinessLogic クラス, DataSet クラス, Entity クラスを導出した。
- ・データモデルからの設計モデル導出
図 6(3)のデータモデルに 3.2.3 で示したルール を適用し図 6(4)の(f)(g)(h)(i)(j)(k)(l)(m)を、「ご意見エンティティ」にルール を適用し図 6(4)(e)のもととなるクラスを導出した。
- ・クラス構造へのマッピング
ユースケースモデルから導出した図 1 のクラス構造で構成されたクラス群に対し、フィーチャーモデルとデータモデルから導出した設計モデルの要素をマッピングし、設計モデルを作成した。

5. 考察

本手法を適用することで『ご意見管理システム』のコアアセットとして、設計モデルを作成することができた。これにより、本手法が設計モデルの作成に適用可能であることを示せた。

また、作成する分析モデルの構造を図 1 のように定義し、かつ分析モデルを構成する要素の導出方法をルール化したため、同じ分析モデルを入力とした場合、作業者が異なっても、ほぼ同じ設計モデルが作成できた。

なお、本手法を適用するにあたり、あらかじめ次の要素について組織で合意を得ておくこと効率的である。

- (1) ユースケースの粒度。
- (2) フィーチャーの粒度と種類。

フィーチャーモデルは機能要件だけでなく、非機能要件もフィーチャーで表現できるが、本手法では非機能要件を設計モデルへ反映するルールを定義していない。この点が今後の課題である。また、作成した設計モデルが、実装工程の入力ドキュメントとして妥当か検証が必要である。

6. おわりに

本稿では、コアアセット開発において課題となる、共通部分と可変部分を含む分析モデルを、5 つのルールに従って設計モデルに反映する手法を示した。今後は非機能フィーチャーを設計モデルへ反映するなどのルールの充実させる。

参考文献

- [1]Kyo C. Kang et al.,“Software Product Line Engineering”,COMMUNICATIONS OF THE ACM, Vol.49,No.12, pp.28-88(2006).
- [2]Kyo C. Kang et al., “Feature-Oriented Product Line Engineering”,IEEE Software, Vol. 19, No. 4, pp.58-65,2002).
- [3]Steffen Thiel et al., “Modeling and Using Product Line Variability in Automotive Systems” IEEE Software, Vol. 19, No. 4, pp.66-72(2002).
- [4]JavaEE ,<http://java.sun.com/javaee/>

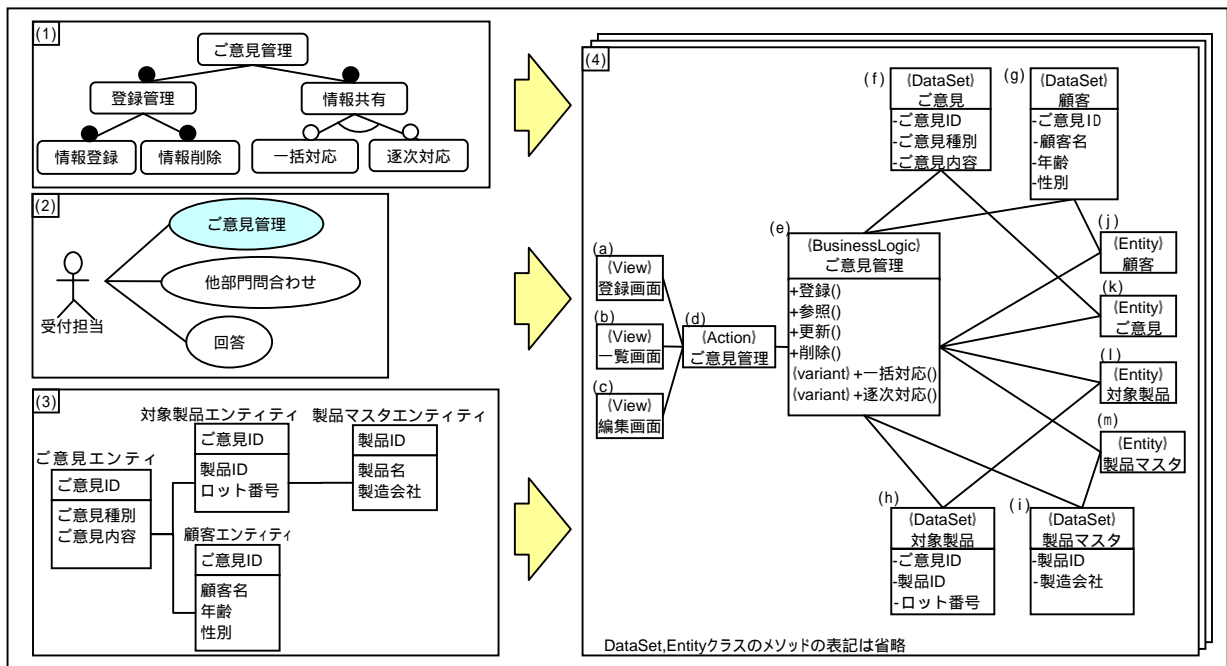


図 5 『ご意見管理』システムへの適用例