

MDAにおけるプラットフォーム分割手法の妥当性の検証

天川 美那† 松浦 佐江子‡

芝浦工業大学 システム工学部 電子情報システム学科‡

1 はじめに

ソフトウェアに対する要求はその幅や深度を増して年々高まり続けている¹⁾。その中で、オブジェクト指向に代表されるようなシステムの各部を独立させて開発や保守を行うという分割統治の概念が普及してきた。本研究で取り扱うMDA(Model Driven Architecture)も分割統治の考え方を強く反映するものであるが、既存のソースコード重視の開発とは異なり、オブジェクトモデルを主体とするシステム開発技法である。モデル主導の開発において分割統治を実現するうえで用いる「プラットフォーム分割」という手法の有用性について、本研究では実際にMDAを用いてシステム開発を行うことでその利点や課題を検証する。

2 MDAとプラットフォーム分割

2.1 MDAとは

MDAはモデリング言語であるUML(Unified Modeling Language)などの業界標準の制定で知られるOMG(Object Management Group)によって2001年に提唱されたシステム作成技法である。また、プラットフォームとはシステムが動作する環境の要素を総称したものである。また、プラットフォームを具体的に実装する技術をプラットフォーム実現と呼ぶ。言語をプラットフォームとした場合、JavaやC++などがプラットフォーム実現にあたる。

概念モデルの作成までは既存の工程と変わらないが、概念モデルを精細化し設計モデルへ移行する際にPIM(Platform Independent Model)と呼ばれるプラットフォームに依存しない形式のモデルを作成し、様々な側面からPIMを分割したのち、それぞれを独立したシステムモデルとしてその時々のプラットフォームに特化したPSM(Platform Specific Model)に変換し、最終的にそれらを接合して一つのシステムを作成する。

実際の工程ではPIMからPSMへの変換に際してマッピング関数という変換機構を用意した上で自動変換を行うが、本研究はプラットフォーム分割の工程に焦点を当てるため、PIMからPSMへの変換は既存の設計モデル開発と同様の手法で行うものとする。

2.2 プラットフォーム分割の利点

現在のシステム開発はシステムの動作するプラットフォームを全て定めた上で行われる。このため一度作成され

たシステムを異なる環境に再構築することが難しい。また、途中段階で作成される様々なモデルも一意の環境下で動作するシステムを想定しているため、環境が変化した場合と同じモデルをそのまま再利用することができない。MDAでは初期段階でシステムを互いに接合可能な仕組みを備えた上で分割するため、PIMに記述された必要機能を満たせばどのような実装で各部を作成しても正常に作動するシステムを作成できると考えられる。

現在でも既存のシステムからソースコードの一部を流用し、適した形に改変して再利用するといったことは技術者の中で自然に行われているが、どの部分をどう切り取って再利用するかという判断は人の手で行われている。つまりMDAは今現在も優れた技術者たちの間で行われている作業をノウハウ化し、広く一般に流布させる試みでもある。

3 ケーススタディ

3.1 要求

本研究では例題として研究室の所蔵書籍の賃借を管理するシステムを作成する。

システムの要件は以下のとおりとする。

- 主要機能
 - 貸し出し
 - 返却
 - 蔵書検索
- ユーザ区分
当システムではマルチユーザを想定しており、登録された各ユーザの区分によって使用できる機能を制限するものとする。
 - 管理者(ユーザ・書籍の追加削除など)
 - 利用者(本の貸し借り)
 - 閲覧者(蔵書の検索のみ)
- webベースの三層構造アーキテクチャ
- 言語はJava・DBはMySQLを使用

3.2 モデル分割

システム作成にあたってインタラクションを行う機構を一つのプラットフォームと考えて分割し、プラットフォーム実現としてキーボードによる標準入力と非接触型ICカードR/W機構であるSONYのFeliCaを用いるものとする。

インタラクションはユーザの入力に対してシステムが何らかの結果を出力するまでの一連の双方向コミュニケーションを指し、システムによって特に多様なインターフェイスが考えられる部分である。ここで入力機器をプラットフォームとしてインタラクション部をシステムから分割する事により、データ入出力時の

Feasibility evaluation of platform division technique in MDA

†Mina Amakawa

‡Saeko Matsuura

†‡Shibaura Institute of Technology

使用機器による形式の差異を意識しないシステムの開発が行えるものと考えられる。通常インタラクションのないシステムが存在しないことから、本研究ではこれをプラットフォーム分割実験の対象にふさわしいものと考えた。

一方、MDA は様々の側面からモデルの分割を行うことを想定しているため、ユーザ認証機能部をシステム要件部から分割しインタラクション部と同様に半独立の部品として開発を進める。なおここでのユーザ認証とは「ログインを試みるユーザのユーザ区分を確認し、それに応じて利用可能な機能を割り当てる」という一連の処理を指すものとする。ユーザ認証も多くのシステムに利用される機能であり、これを一個のシステムから独立させることは有効だと考えられる。

3.3 開発の手順

- 1 システム全体に必要な機能を洗い出し、概念モデルを作成する。
- 2 概念モデルの中からユーザ認証およびインタラクションに関わるフィールド・メソッドを抜粋し、それを元にユーザ認証部・インタラクション部・システム要件部の三種の PIM を作成する。(図 3.1)
- 3 各 PIM を実装に即した PSM に書き換える(図 3.2)このとき、インタラクション部の PSM はプラットフォーム実現が標準入出力の場合と FeliCa の場合の二種類を作成する

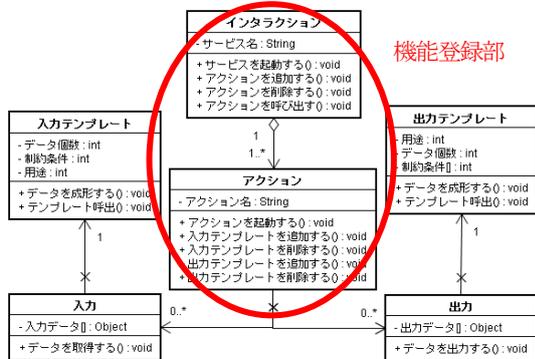


図 3.1 インタラクション部の PIM

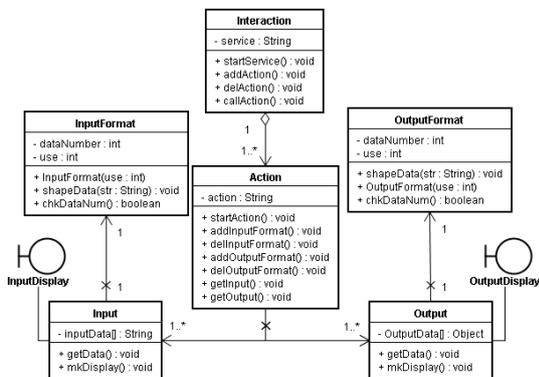


図 3.2 インタラクション部の標準入出力用 PSM

3.4 システムの接合

インタラクション部のモデルの「アクション」クラスはシステム要件部の機能をインタラクション部のモデル内で表す部分である。インタラクションクラス内で貸出・返却・書籍追加等の具体的な機能をアクションクラスのインスタンスとして登録し、各機能に必要な入力内容と形式をアクションクラス内で設定することでシステム要件部とインタラクション部の接合を果たしている。同様の仕組みがユーザ認証部にも存在しており、これを用いて各部品を接合する。

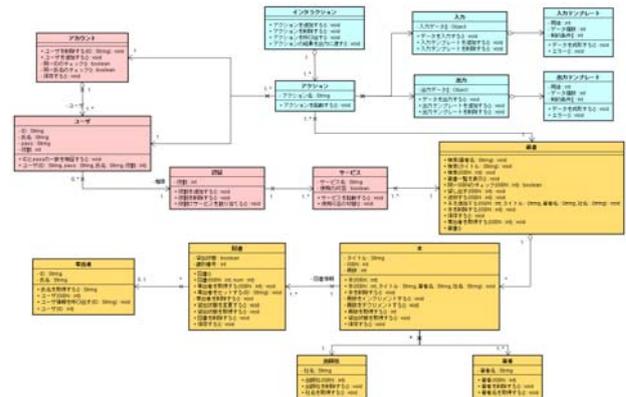


図 3.3 図書管理システム PIM

図 3.3 は今回作成するシステム全体の PIM である。アクションクラスは図書管理の主機能を持つ蔵書クラスと、ユーザの登録・認証を行うアカウントクラス及びユーザクラスに関連を持っている。同様にユーザ認証部のサービスクラスが蔵書クラスと関連を持つ。

4 まとめ

図 3.1 の機能登録部が正しく動作していれば、それがどのようなプラットフォーム実現で実装されても図書管理システムとしての機能を果たすのは自明である。それ以外にも、システムを分割して開発することでメソッドやフィールドの持つ役割をより明確にすることができるため、保守が容易になる。またインタラクション部を分割することで、入力値のエラー処理を考慮せずに要件部の開発を行うことができた。これらのことからプラットフォーム分割はシステム開発効率の上昇に有効であると考えられる。

本研究ではユーザ認証機構とインタラクション部の二つをシステム要件から分割したが、他にもエンティティから DB への接続部や言語をプラットフォーム境界とした場合のシステムの切り分けなど、様々な側面からの分割が考えられる。MDA を用いた開発におけるシステム拡張の利点や問題点の考察も進めていきたいと考えている。

5 参考文献

- [1] S.J.メラー, K.スコット, A.ウール, D.ヴァイセ: "MDAのエッセンス", 翔泳社(2004)