

定義・生成・検証の観点からのモデル駆動開発への取り組み

松浦佐江子[†] 青木善貴[†] 加藤真[†] 松井浩二[†] 川合怜[†] 小形真平[‡]

アブストラクト

モデル駆動開発は高品質なソフトウェアを効率よく開発する魅力的な技術である。この技術を活用するために、我々が取り組んできた研究を紹介し、実用化への問題点を議論する。

Toward Practical Model Driven Development - Definition, Generation and Verification -

Saeko Matsuura Yoshitaka Aoki Makoto Kato Koji Matsui Satoshi Kawai Shinpei Ogata

Abstract Model-driven development is an attractive technology to develop efficient high-quality software. In order to take advantage of this technology, this paper discusses the problems for putting it into practical use in industrial applications through our several researches on MDD.

1. はじめに

芝浦工業大学電子情報システム学科では、2002 年度より、オブジェクト指向技術を用いた 3 年後期に行われる実践的ソフトウェア開発実習を目標に Java および UML の教育を行っている。これらの経験を経て、本研究室では、数年来、モデル駆動開発に関わる研究を、学部から修士・博士課程までの研究テーマとして取り組み。その成果を、学部及び修士課程の授業におけるモデリング教育等に活用している。本稿では、本研究室の取り組みを紹介するとともに、モデル駆動開発の実用化へ向けた課題を議論する。

2. モデル駆動開発の研究事例

2.1. モデル駆動開発普及の問題点

モデル駆動開発は、多様なソフトウェア開発において「モデルコンパイラ」によりコードを生成するという高品質なソフトウェアを効率よく開発するための魅力的な技術である。しかし、コンピュータは翻訳された指示をそのまま実行するだけであるから、元となるモデルが適切に書かれていなければ役には立たない。また、モデルはそのままでは動作しないことや、モデルごとに抽象化の観点が異なることから、具体的にすべての要素を記述するコードに比べて、書かれていることを十分吟味できない学生が多い。モデルの良し悪しを判断する手段がないと、モデリングの必要性は感じて、現場で利用するには至らない。

2.2. 研究事例

開発の出発点である要求定義では機能要件を明らかにするために、ユースケース分析を行う、レビューするだけではなく、コンピュータ支援を行いやすくするため、本学科では、ユースケースの定義を UML のユースケース図・アクティビティ図・クラス図を用いて行っている。このモデリングの特徴は、アクティビティ図のアクション系列としての基本フローの定義、ガードの明記による例外フローの明確化、パーティションの利用による振舞いの主体の役割の明記、サブアクティビティによる全体モデルの定義、オブジェクトノードによる振舞いとデータの関連の定義である。

このモデルを要求分析モデルと呼び、下記の研究の出発点とし、要求定義の妥当性の確認、実現可能性の検証、実装への変換について研究してきた。

要求分析モデルからの WebUI プロトタイプの自動生成[1]では、モデルを実際に動作させることにより、モデルで定義された要求の妥当性を確認しながらモデルを洗練する。[2]では、要求分析モデルに対して、テストケースを定義し、要求分析モデルをシミュレーションすることで、テスト観点からのモデルの実現可能性を確認しながら、モデルを洗練する。[3]ではシステムのデータの普遍的な性質をステートマシン図でモデル化し、複数のユースケースの実行において、その性質が満たされているかを、モデル検査ツールを用いて自動的に検査し、要求分析モデルの実現可能性を検証する。[4]では、要求分析モデルから、複数の Web フレームワークを利用して実装する方法を検討し、要求分析モデルの十分性を確認した。

モデルの利点は、その関心事毎に分離してシステム

[†]芝浦工業大学 Shibaura Institute of Technology

[‡]信州大学 Shinshu University

の特性を考えることであるが、コードを書くように詳細な手順を書きすぎてしまうことがある。モデルを必要以上に複雑化しないことが重要である。そして、要求分析モデルではユーザとシステムのインタラクションを分離してモデル化するが、システムアーキテクチャによる違いは変換規則により、段階的に定義することが望ましい。モデリングのエディタはよいツールがあるが、統合したビューで確認することや、特定の性質を検査しながらモデルの構築を支援するツールがない。

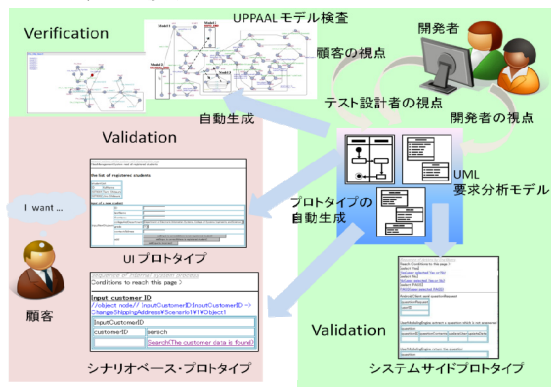


図 1 要求分析モデルの定義とプロトタイプ生成・検証

3. 実践事例

本学科では学部 2 年から 3 年次に Java 言語の講義 1 コマ×半期 2 回, Java 言語の演習 1 コマ×半期, UML の講義・演習 1 コマ×半期, 仕様変更やテストを含む応用演習 2 コマ×4 回, PBL (Project Based Learning) によるソフトウェア開発実習 2 コマ×半期を行っている。これらの授業においては, UML はソフトウェア開発を行う際の 1 つの道具として, 要求の定義や, ソースコードをリバースして, クラス図をレビューするといった方法で随時, 利用している。

3 年次の UML の講義・演習では, ユースケース記述の評価レポートを自動生成し, 学生にフィードバックしている[5]。モデルの形式的な誤りは, レポートにより改善傾向がみられる。この演習では, モデルから Java コードを記述し, 動作させる課題を通して, 実装に必要な情報を認識させる課題も実施している。

3 年次のソフトウェア開発実習[6]では半期の実習中約 3 分の 2 がモデル定義である。レビューを繰り返すことで, モデルの意味と効果を理解してもらう。

大学院授業では, プロトタイプ生成を用いた要求分析モデリング演習を実施した[7]。3 名のグループで, 生協の教科書販売システムを分析し, 顧客役の教員および TA によるレビューを実施した。

4. 今後の課題

モデルが「適切」であるかを確認できる方法および支援ツールが必要である。段階ごとに書くべき内容を整理し, モデルが理解しやすい程度に複雑化しないことも重要である。現在, スマートフォンアプリケーション開発への適用や, コード生成に向けた OCL の導入方法を検討している。

さらに, モデルを単なるお絵かきではなく, コードを開発の設計書として利用することができる技能育成が重要である。本研究室では総合研究の導入演習や学習教材の開発において, こうした教育を実施しているが, 多くの学生がこうした演習や開発に参加できる機会を提供したい。

参考文献

- [1] 小形, 松浦: UML 要求分析モデルからの段階的な Web UI プロトタイプ自動生成手法, 日本ソフトウェア科学会, コンピュータソフトウェア, Vol.27, No.2, pp.14-32, 2010.
- [2] 式見, 松浦, 要求定義の実現可能性保証のためのシミュレーションによるテスト設計手法, 電子情報通信学会和文論文誌 D, Vol.J97-D, No.3, pp. 427-436, 2014.
- [3] Y.Aoki, S.Ogata, H.Okuda and S.Matsuura, Data Lifecycle Verification Method for Requirements Specifications Using a Model Checking Technique, Proc. of The Eighth International Conference on Software Engineering Advances(ICSEA2013), pp.194-200..
- [4] Hirotaka Okuda, Shinpei Ogata, Saeko Matsuura, Experimental development based on mapping rule between requirements analysis model and web framework specific design model, SpringerPlus 2013, 2:123 doi:10.1186/2193-1801-2-123
- [5] 松井, 奥田, 野呂, 岡田, 加藤, 渡辺, 松浦, モデリング能力育成を目的としたユースケース記述の自動評価手法, 電子情報通信学会和文論文誌 D , Vol.J97-D, No.3, pp. 461-472, 2014.
- [6] 松浦: 実践的ソフトウェア開発実習によるソフトウェア工学教育, 情報処理学会論文誌, Vol.48, No.8, pp.2578-2595, 2007.
- [7] S.Ogata and S.Matsuura, Training of Requirements Analysis Modeling with UML-based Prototype Generation Tool, ISEC '12, pp.105-108, 2012.