

モデル駆動開発

平成25年度シラバス

2013年1月4日

国立情報学研究所

トップエスイープロジェクト

代表者 本位田 真一

1. 科目名

モデル駆動開発

2. 担当者

鷲崎 弘宜、鄭 顕志、久保秋 真

3. 本科目の目的

本科目では、モデル駆動開発がなぜ必要で、どのような開発方式なのか講義し、LEGO Mindstorms で制作したロボットとモデル駆動開発ツール BridgePoint を使って演習する。講義と演習を通じて、モデル駆動によるソフトウェア開発が、抽象度の高いモデルから抽象度の低いコードまでをシームレスに接続できる方法であることを実践を通じて理解する。また、モデル駆動開発が、生成・変換規則とモデルを再利用することで、効率的で属人性の少ない一定の品質を確保したシステムを構築する方式であることを理解する。

4. 本科目のオリジナリティ

モデル駆動開発が、属人的なプログラム作成方法を生成・変換規則とモデルの再利用に置き換えられることを強調し、単に自動化の取り組みとは捉えずに解説する。また、組込み向け商業ツールである MentorGraphics 社のモデル駆動開発ツール BridgePoint と、LEGO Mindstorms で制作したロボットを使用することで、実務レベルのモデル駆動開発による実際に動作する組込みシステムの開発演習を実施する。

5. 本科目で扱う難しさ

オブジェクト指向や UML の普及に伴い、モデルを用いたシステム開発の価値や有用性は広く認知されるに至った。しかしながら、システムを実装する現場では、プログラムを作成する人間が、設計されたモデルを読み込んでのち、理解した設計内容に自分の経験則やドメイン知識などを結びつけることでプログラムを書くという行為が続けられている。また、大学のシラバスや企業研修の講座を見ても、教育や技術導入の場面において、モデリ

ングの訓練とプログラミングの演習は実施していても、モデルとプログラムを結びつける技術に関する教育を体系化して実施している場合はほとんどない。

モデル駆動開発では、生成・変換規則を用いて新しい情報や条件を付与し、モデルの変換を繰り返すことで開発を進める。このことは、プログラマが咀嚼した設計内容に自身の技量や経験を付与してプログラムに変換する属人的なプログラム作成方法における暗黙的な知識活用から、そこに含まれている生成・変換の規則を陽に表出した機械化可能な知識として扱う方法への推移を意味する。

これらのことは、本科目が2つの難しさを取り扱うことを意味する。ひとつは、モデルとプログラム（あるいは中途の別のモデル）との間を結びつけるための技術が必要であることを認識することであり、いまひとつは、プログラミングにおける暗黙的な変換作業を形式的な知識として表出することである。

6. 本科目で習得する知識・技術

本科目では、モデル駆動開発におけるモデル変換の原理と、これを適用したツールを使った場合の実務的なシステム開発の方法を実践的に学ぶ。

モデル駆動開発、モデル駆動アーキテクチャの考え方、モデルとプログラムの接続やそこに使われるモデル変換の仕組みといったことに関する知識が得られる。モデル変換の技術について演習する場面では、モデル交換に使用する表記や、変換言語による変換規則の作成方法を学ぶことができる。また、商業的なモデル駆動開発支援ツールはまだ十分に普及しているとは言えないため、BridgePoint を使って実際にロボットを動かすことができるプログラムを生成することは、モデル駆動開発の実用性を体験する貴重な機会となっている。

7. 前提知識

本科目の受講生は、以下の項目を習得済みであることが望ましい。

- (1) オブジェクト指向
- (2) ソフトウェアモデルと UML
- (3) C 言語プログラミング
- (4) デザインパターン

このうち、項目(2)および(4)は、トップエスイー開講科目「ソフトウェア工学入門」で習得可能である。

8. 講義計画

概要

- 第1回： なぜモデル駆動開発が必要なのか
- 第2回： モデル駆動開発とコード生成
- 第3回： モデル変換の仕組み
- 第4回： モデル変換の実践
- 第5回： コード生成の環境
- 第6回： モデル駆動開発の実践
- 第7回： 実践およびまとめ

詳細

第1回：なぜモデル駆動開発が必要なのか

1. モデル駆動開発 (MDD) とは
2. モデル駆動アーキテクチャ (MDA) とは
3. モデル駆動エンジニアリング (MDE) とは
4. プラットフォームとモデル、メタモデル階層
5. モデル変換方式

第2回：モデル駆動開発とコード生成

1. 演習1：モデル・コード手動接続体験
2. 演習2：モデル・コード自動接続体験

第3回：モデル変換の仕組み

1. モデル変換のしくみ
2. モデルの表記：XMI
3. モデル変換言語：ATL
4. ATLによるモデル変換

第4回：モデル変換の実践

1. 演習3：ATLによる Singleton パターン適用
2. 演習4：ATLによる Observer パターン適用

第5回：コード生成の環境

1. BridgePoint に内包されている MDD 技術について
2. 演習5：BridgePoint でモデルを作る

第6回：モデル駆動開発の実践

1. 演習6：モデリングとコード生成による搬送システム開発

第7回：実践およびまとめ

1. これまでの演習を受けた討論
2. 演習のふりかえり
3. 関連技術の紹介
4. まとめ

9. 教育効果

本科目を受講することにより、システム開発にモデルを活用する効果的な方法を実践を通じて習得できる。また、自動/手動に依らず、生成・変換規則によるモデル変換を活用した開発の進め方の重要性と、モデルとプログラムを結びつける技術の必要性を認識できるようになる。さらに、生成・変換規則を活用したモデル変換によって、プログラマがプログラムを作るときの暗黙的な知識を表出して、機械化可能な情報として扱うことができることを理解できるようになる。

10. 使用するツール

- ・ MentorGraphics 社 BridgePoint (モデル駆動開発ツール)
- ・ LEGO Mindstorms NXT (ロボット制作キット)
- ・ nxt0SEK/JSP および関連ツール (クロス開発システム)
- ・ ArgoUML および関連ツール (モデル変換ツール)

11. 評価

講義・演習中の発表や議論、課題レポートを総合して評価する。

12. 実験および実習

ATL を使ったモデル変換演習を用意しており、実際に変換規則を作ることと、作成した変換規則をモデルに適用することを演習できる。また、BridgePoint を使って LEGO Mindstorms のロボットを動かすプログラムをモデルから生成する演習を用意している。構造のモデルと振舞いのモデル、振舞いのアクションを指示するスクリプトを作成し、自動変換によって得たプログラムで実際にロボットを動作させることを体験できる。

13. 教科書・参考書

教科書

- ・ 講義資料を用意する

参考書

- ・ 「オブジェクト指向の実装技法入門 多態・隠蔽・関連などを C 言語で実装する」, インターフェース, CQ 出版, 2002 年 5 月特集
- ・ 「Executable UML 実践入門 クラス・モデルをいかに作成するか」, レオン スター, CQ 出版, ISBN: 4789837149
- ・ 「Practical Statecharts in C/C++ Quantum Programming for Embedded Systems」, Samek, CMP Books, ISBN:1578201101

Web

- ・ [QVT01] Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification
- ・ <http://www.omg.org/docs/ptc/07-07-07.pdf>
- ・ [ATL01] ATL User Manual
- ・ [http://www.eclipse.org/m2m/at1/doc/ATL_User_Manual\[v0.7\].pdf](http://www.eclipse.org/m2m/at1/doc/ATL_User_Manual[v0.7].pdf)
- ・ [ATL02] ATL Basic Examples - Families to Persons
- ・ http://www.eclipse.org/m2m/at1/basicExamples_Patterns/
- ・ [ATL03] ATL Transformations - Book to Publication
- ・ <http://www.eclipse.org/m2m/at1/at1Transformations/>
- ・ [ATL04] IDL Modules

- <http://www.omg.org/docs/ad/99-10-14.doc>
- [MDA01] ビジネスコミュニケーション 最新技術トレンド
- <http://www.bcm.co.jp/site/2003/2003Oct/techo-trend/03techo-trend-mdal0.htm>
- [MDA02] ビジネスコミュニケーション UML の基礎と応用
- <http://www.bcm.co.jp/site/2002/uml/uml12.htm>
- [MDA03] IT アーキテクト MDA の基本思想とアーキテクチャ
- http://www.itarchitect.jp/technology_and_programming/-/12801-1.html
- [QVT02] INF 5120 - Lecture #5 Model transformations with ATL and other technologies
- http://www.uio.no/studier/emner/matnat/ifi/INF5120/v08/undervisningsmateriale/lecture5_INF5120.pdf
- [DIC01] Weblio
- <http://www.weblio.jp>

14. 本科目で使用する課題について

本科目で使用する演習課題の一部は、平成 22 年度文科省専修学校補助事業「産学連携による実践型人材育成事業－専門人材の基盤的教育推進プログラム－」に採択された「産業界と連携した高品質組込みソフトウェア技術者養成プロジェクト (SQEP)」(九州技教育専門学校主催) の演習課題を参考に作成している。同プロジェクトの取り組みに敬意を表すると共に、活用に必要な許諾を与えていただいたことに感謝する。