

ソフトウェアモデリング教育における モデル駆動開発の活用

赤山 聖子^{†1} 久保 秋 真^{†2} 久住 憲 嗣^{†3}
二上 貴夫^{†4} 北須賀 輝明^{†5}

モデル駆動開発 (MDD) は、モデル上での検証とコードの自動生成ができるため、モデリングに集中することができ、初学者のモデリングスキルの向上に役立つと考えられる。また、ソフトウェア開発におけるモデリング技術の活用方法などの習得には、PBL (Project Based Learning) の利用が効果的であると考えられる。本実践では、MDD と PBL 手法を組み合わせた初学者向けソフトウェアモデリング教育プログラムの開発を行った。総合的な能力が必要な PBL を初学者でもスムーズに取り入れられるように、ガイド付き短期 PBL とガイドなし長期 PBL の 2 段階 PBL 手法を取り入れた。開発した教育プログラムを用いた専門学校での実証講座を実施し、モデリングスキルやモチベーション、協調性の向上が図れることを確認した。

An Application of Model Drive Development (MDD) in Software Modeling Education

SEIKO AKAYAMA,^{†1} SHIN KUBOAKI,^{†2} KENJI HISAZUMI,^{†3}
TAKAO FUTAGAMI^{†4} and TERUAKI KITASUKA ^{†5}

Model Driven Development (MDD) can verify the software and automatically generate the source codes leading the programming beginners to focus on the modeling and improve their modeling skills. To apply the modeling technique in software developments, Project Based Learning (PBL) is effective method to learn the application of MDD. We proposed the software modeling educational program for beginners combined with MDD education and PBL method. In this project, we adopted the two-step PBL which consists of the short period PBL with a guide and the long period PBL without a guide in our software modeling education program. We confirmed that our proposed program led beginners to improve the modeling skills, motivation and communication skills.

1. はじめに

近年、組込みソフトウェアの設計品質の向上のため、設計にモデリング技術を活用することがソフトウェア開発における大きな流れになっている¹⁾。ソフトウェア開発では、UML を用いたオブジェクト指向モデリングを用いられることが多くなり、教育現場での教育の必要性も高まっている。しかし、UML によるモデリングの教育では、学習者が「このモデルは合っているのだろうか?」という疑問を持つことが多く²⁾、その解決が教育上の大きな課題となっている。

一方、産業界では、モデルを使う開発手法として、モデル駆動開発 (Model Driven Development :MDD) の実用化が進んでいる。MDD では、モデル上での検証とコードの自動生成が可能で、作成したモデル図をすぐに動作として確認することができるため、モデル図の評価手段の一つとして用いることができる。さらに、設計と実装を完全に分離することができるため、モデリングに集中して開発できる。従って、MDD を初学者のソフトウェア開発教育に活用することで、モデリングに重点を置いた教育が行え、モデリングスキルの向上が図れると考えられる。

MDD の教育への活用例としては、高校生や大学初級学年を対象としたプログラミングを前提としない、抽象思考の訓練があげられる³⁾⁴⁾。しかし、ソフトウェア開発を対象としたソフトウェアモデリング教育における UML やモデリング技術の教育への活用はほとんど行われていない。

また、オブジェクト指向モデリングの教育には数名のグループによる演習の実施が効果的であるとされており⁵⁾、大学におけるモデリング教育でもグループによる協調学習活動を取り入れている⁶⁾⁷⁾。グループでの教育として利用が増加している PBL (Project Based Learning) 手法では、それまでに獲得した知識・技術を実開発でどのように利用するのか?

^{†1} 九州技術教育専門学校
Kyushu Technical Education College

^{†2} 株式会社アフレル
Afirel Co.,Ltd.

^{†3} 九州大学システム LSI 研究センター
System LSI Research Center, Kyushu University

^{†4} 株式会社東陽テクニカ
Toyo Corporation

^{†5} 熊本大学大学院自然科学研究科
Graduate School of Science and Technology, Kumamoto University

ということを知得するための教育手法として有益であるため、ソフトウェア開発におけるモデリング技術の活用方法などの習得にも、PBLの利用が効果的であると考えられる。ただし、PBLでは、技術力だけでなく、プロジェクトマネジメント技術やコミュニケーション技術など、プロジェクトの進め方に関する知識も必要となる。先行実践においても主に大学3、4年生や大学院生などの上級学年での利用が行われており⁸⁾⁹⁾¹⁰⁾、初学者に取り入れるのは難しい手法である。

本実践では、ソフトウェア開発におけるモデリングスキル及びモデリング技術の活用方法の習得を目的としてMDDとPBL手法を組み合わせた初学者向けソフトウェアモデリング教育プログラム(以下、本プログラム)の開発を行った。総合的な力が必要なPBLを初学者でもスムーズに取り入れられるように、ガイド付き短期PBLとガイドなし長期PBLの2段階PBL手法を取り入れた。本論文では、開発した教育プログラムを専門学校生に対して実証講座を行った結果を報告する。

2. 教育目標

モデリングは、細かい実装方法を知らない方が習得しやすいと言われていること⁴⁾、習得には時間がかかること²⁾などを踏まえ、ソフトウェア開発に関する学習の早い段階でモデルを用いたソフトウェア開発手法の教育を行うこととした。教育対象は、ソフトウェアエンジニアを目指している初学者、具体的には情報系の専門学校及び情報系学部1、2年次に在籍する学生を対象とする。

ソフトウェアモデリング教育において初学者を対象とした場合の教育目標を次の3点とする。

- 静的・動的モデリングを行うための最低限のモデリング記法が分かる。
- ソフトウェア開発におけるモデリング技術の活用方法が分かる。
- チームメンバーの一員としてモデルを用いたソフトウェア開発ができる。

3. 教育プログラムのコンセプト

3.1 教育プログラムの要件

組込みソフトウェアエンジニアの育成においてモデリング教育の必要性が高まり、ソフトウェアモデリングを重視したロボットコンテスト¹¹⁾¹²⁾等も実施されている。九州技術教育専門学校(以下、本校)では、組込みソフトウェア教育の一環として、2008年からETロボコン¹¹⁾に参加しており、組込みソフトウェアモデリング教育を行ってきた。ただし、モ

デリングの教育は難しく、学習者がモデリングを行う上で「このモデルは合っているのだろうか?」という疑問や、最終的に必要なのは動くソースコードなのだから「モデリングなんて意味がない」という意見を口にし、学習途中で諦めてしまうことがある²⁾。本校のETロボコンの参加者も同様の傾向にあり、ETロボコンの評価の対象になっているという理由で仕方なくモデル図を描いている状態で、「なぜ、モデリングが必要なのか?」という根本の理解させることが困難であった。

また、小木ら⁷⁾は、モデリング演習における問題の原因として(1)モデル設計のノウハウが分からない(2)受講者のモチベーションが高くない(3)グループワークで協力できないなどをあげている。

このように、ソフトウェアモデリング教育においては、「理解性」「モチベーション」「協調性」に関する課題がある。さらに、ソフトウェアモデリングでは、モデリング、動作検証というサイクルを何度も繰り返すことで、モデリングスキルの向上が図れる。しかし、モデリング結果を動作として確認するためには、実装言語でのコーディングが必要であり、設計と実装という抽象度切り替えを何度も行う必要があり、その変換が初学者の理解の障壁になっている。

これらを解決するため、本プログラムの要件を下記のように設定した。

- モデルを作成したらすぐに動作を確認できる環境を提供すること
- モデルを使って開発することはプログラミングと同等以上の価値があると実感できること
- 学習の初期段階からチームで協働する演習を学習活動の中心とすること

3.2 コンセプト

本プログラムでは、ソフトウェアの初学者に対し、モデリングのスキル向上やモデルを中心に置いた開発方法の理解を目的としている。教育プログラムの要件を踏まえて、以下の3つを教育コンセプトとして、教育プログラムの開発を行った。

3.2.1 ソフトウェアモデリング教育にMDDを活用

MDDとは、設計段階で作成したモデルをツール等を使って動作シミュレーションを行うことで検証し、実際に動作するソフトウェアの実装コードを自動生成することを狙った開発手法である。MDDにおいては、モデルがソフトウェア・ライフサイクルのすべての局面において開発プロセスの中心となる¹³⁾。MDD手法をソフトウェア開発に用いることで、次のようなメリットがある(1)実装と設計を分離できるため、モデリングに注力して開発できる(2)モデル上でのシミュレーションができるため、開発早期での検証が可能となる。

(3) 実装工程を自動化することで、設計、テスト、改善のサイクルを短時間で繰り返すことができる。

MDD をモデリング教育に活用するメリットとしては、以下のようなものがある。

- モデリングに集中させることができる。
- 学習早期に一連の開発体験が行える。
- モチベーションの維持につながる。

MDD では、抽象度の高いモデルの記述のみで、動作検証を実施することができるため、抽象度の変更に伴う混乱を招きにくい。さらに、開発対象以外の部分をブリッジという形で提供することで、学習者はアプリケーションドメインのみを開発すればよく、プログラミングやハードウェアの詳しい知識がなくても、一連の開発体験を行うことができる。これらの特徴は、学習者のモチベーションの維持にもつながる。本プログラムでは、MDD の実現方法の一つである Executable UML¹⁴⁾ を利用した MDD 教育を実施した。実行可能モデルと称される Executable UML とは、実行可能なセマンティクスとタイミング規則を UML 表記法のサブセットと組み合わせたものである。Executable UML における基本的な構成要素は、クラス図、ステートマシン図、アクション記述の 3 つである。Executable UML を用いてモデル化する際の手順を次に示す。

- (1) システムの問題領域(ドメイン)を定義する。
- (2) ドメインに対するクラス図を作成する。
- (3) 各クラスのライフサイクルをステートマシン図で記述する。
- (4) ステートマシン図の各状態のプロシージャをアクション言語で記述する。
- (5) モデルを検証する。
- (6) モデルをコンパイルし、ソースコードを生成する。

なお、Executable UML での開発を行うツールとして BridgePoint (Mentor Graphics) を用いた。

3.2.2 要素技術と開発体験をスパイラル的に教育

教育項目を限定して最低限の要素技術を教育した後に、その技術を利用した一連の開発を実施させるという工程をスパイラル的に積み重ねることで、技術の必要性を理解させながら教育を行う。教育の概略を図 1 に示す。

この方法を用いることで、受講者のモチベーション維持につながるだけでなく、座学での講義により習得した技術の定着及びスキル化を図ることができる。開発体験においては、最初からすべての工程を開発することは困難であるため、受講者の開発対象領域を限定し

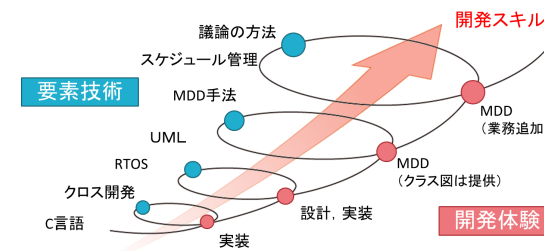


図 1 スパイラル教育の概略
Fig. 1 Spiral curriculum model.

て、それ以外の部分は開発環境として提供し、徐々に開発対象領域を拡張していくことでスムーズに開発体験が行える工夫を行う。

3.2.3 教育早期における PBL の活用

オブジェクト指向モデリングの教育には数名のグループによる演習の実施が効果的とされており⁵⁾、大学におけるモデリング教育でもグループによる協調学習活動を取り入れている⁶⁾⁷⁾。グループでの教育として利用が増加している PBL 手法では、それまでに獲得した知識・技術を実開発でどのように利用するのか?ということを知得するための教育手法として有益であるため、ソフトウェア開発におけるモデリング技術の活用方法などの習得にも、PBL の利用が効果的であると考えられる。また、総合的な技術力が必要なため、知識や技術の定着が期待できる。

さらに、PBL のもう一つの効果として、学習者自身が不足している知識や技術に気づくことができる点がある。その為、初級学年における PBL の経験は、次年度以降の講義や演習への取り組みに対する意欲の向上が期待でき、在学期間における教育効果の向上が見込める。ただし、PBL では、技術力だけではなく、プロジェクトマネジメント技術やコミュニケーション技術など、プロジェクトの進め方に関する知識も必要となる。先行実践においては、主に大学 3、4 年生や大学院生などの上級学年で半年~1 年を通しての利用が行われており⁸⁾⁹⁾¹⁰⁾、初学者に取り入れるのは難しい手法である。

本実践では、ソフトウェア開発の経験がない初学者に対する PBL とするため、通常実施されている長期の PBL の前にプロジェクトファシリテーションツールを用いたガイド付き短期 PBL¹⁵⁾ を実施する 2 段階 PBL とすることで、スムーズな PBL が行えるように工夫した。

4. 教育プログラムの構成

作成したカリキュラムは、基礎編、応用編、短期 PBL、長期 PBL である。UML やモデリング技術は、応用編で教育し、基礎編では組み込みソフトウェア開発を行う上での基礎知識の習得を目的とした。知識教育である基礎編・応用編の後に実施する PBL では、基礎編、応用編で習得した技術の定着と自律的にモデル中心の開発が行えるようになることを目標とする。なお、PBL では、MDD による開発の進め方及びモデリングスキルの向上を目的としたガイド付き短期 PBL とチームによる自律的な開発力の習得を目的とした長期 PBL の 2 段階構成とした。

また、基礎編、応用編、短期 PBL の各ステップで学習する基礎演習や総合演習の題材をできる限り同一のものにすることで、各ステップで学習する内容と事前のステップで学習した内容の関係を深めさせる工夫をした。各ステップでの教材間の関係を図 2 に示す。

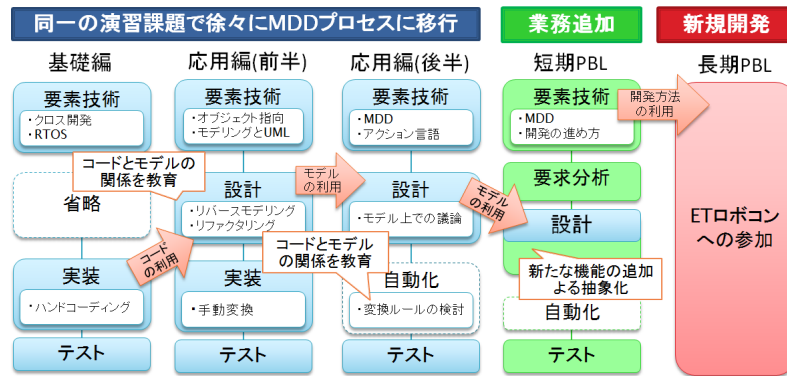


図 2 教材間の関係
Fig. 2 Relationship between educational materials.

4.1 基礎編・応用編

基礎編、応用編は座学による要素技術の講義と基礎演習、総合演習をセットで実施する。基礎編、応用編の実施項目を表 1 に示す。

基礎編及び応用編で実施した総合演習課題を紹介する。架空の運輸会社の自動搬送ロボットを開発するという業務であり、開発対象ロボットは LEGO Mindstorms NXT で製作し

表 1 基礎編・応用編の教育項目

Table 1 Education items basic and advanced classes.

編	要素技術	基礎演習	総合演習
基礎編	要素技術 A 基礎演習 A 総合演習 A	組み込みシステムの基礎、RTOS、クロス開発 センサ、モータを動かすプログラム、ライトトレース 一連の業務をハンドコーディングで実装	
応用編	要素技術 B 基礎演習 B 総合演習 B	UML、MDD 手法、モデリング手法、モデルからコードの生成技法 基礎演習 A のリバースモデリング、変換ルールに基づいてモデルからコードの手動変換 総合演習 A と同一業務を MDD 手法で開発	

た自律型車両ロボット（図 3 左）である。自動化する業務は、運搬業務、転送サービス、回送業務の 3 種類である。3 種類の業務は、配達先や荷物の有無により変更される。なお、配達先はロボット側面の側壁監視部（超音波センサ）、転送先の検知はロボット前面のバンパ（タッチセンサ）で検知する。課題はロボットの前方にあるライン監視部（光センサ）でコースの黒いラインをトレースし、配達先または転送先、車庫で停止し、それぞれに地点において適切な動作を行い、所定の位置に荷物を届けることである。

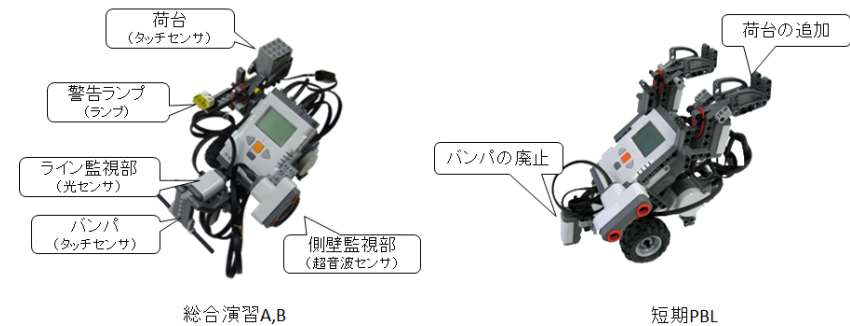


図 3 自動搬送ロボット
Fig. 3 Auto transport robot.

総合演習課題を MDD で開発する応用編では、図 4 に示すように、自動搬送システムの全体的な制御を行うクラス（AutoTransporter）、ラインに沿って走行するクラス（LineTracer）の 2 つ以外のクラス内のステートマシンは、実装済みのものを提供する。これにより、命名済みの 2 つのクラスの責務のみに着目して責務分割を検討する方法を学ぶことができる。また、各クラス内で起こる主要なイベントを定義した状態で配布し、各自が独自のイベント

を定義する際に適切な抽象度で定義するための指針となるようにした。さらに、MDD ツールを用いて、コードの自動生成を行うことで、モデル図のレビューを複数回行い、その度に動作確認をすることができるため、短期間でのモデリングスキルの向上が見込める。

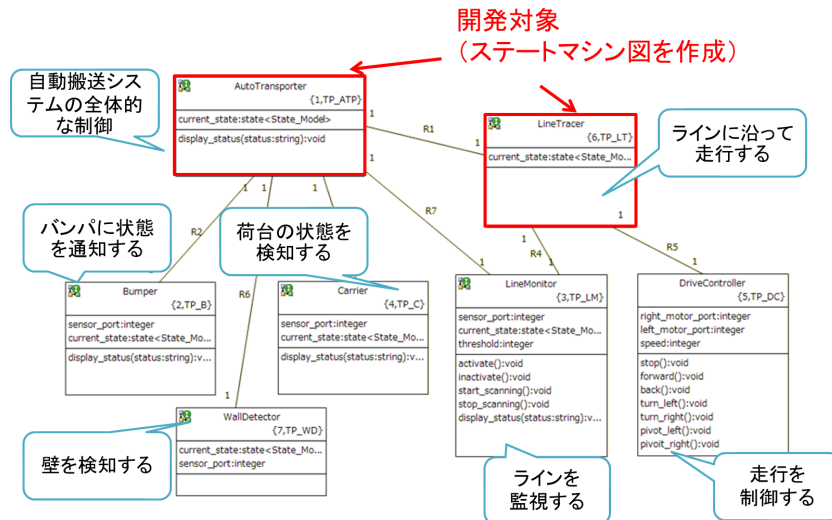


図 4 自動搬送システムのクラス図
Fig. 4 Auto transport system class diagram.

4.2 短期 PBL

短期 PBL では、応用編で開発した業務システムの仕様変更に対応したシステムの開発を MDD で開発させることで、MDD での開発手法の理解及びモデリングスキルの向上を図る。

4.2.1 課題内容

総合演習 B に対して業務を追加するソフトウェア開発案件を、MDD を用いてチーム開発を行う。これにより、モデルを利用して開発することで、仕様変更に対応しやすいことを実感してもらうことがねらいである。業務の追加、変更内容は、次の通りである (1) タッチセンサを使用したバンパを廃止し、荷台を 2 つにする。同じデバイスでも別の利用方法をした場合の名前の付け方の工夫や多重度の変更を期待する (2) 搬送コースを変更する。搬送コースが今後も変更する可能性があることを予想した設計になることを期待する (3) 基地

局と通信をして、搬送ルート及び搬送個数の変更を行う。通信機能を設けることで、新しいデバイスを利用することになるため、通信ブリッジが必要になることを気づいてもらう。

なお、ブリッジの開発に関しては、教育対象外であるため、ブリッジが必要だと気づいたチームには、提供をする。自動搬送ロボットと基地局の通信には、bluetooth を用いた。また、基地局は、PBL 期間中に実行モジュールを提供した。

4.2.2 PBL の進め方

本 PBL では、初学者がスムーズな PBL を行えるようにプロジェクトの進め方に関するガイドを設けた。ガイドとして、プロジェクトファシリテーション手法¹⁶⁾を用いた。ツールとして、本 PBL 用にアレンジした下記の 2 つを利用した¹⁵⁾。

タイムボックスかんばん: チーム内での進捗状況管理に使う表で、1 日のチームの目標と各タイムスロット毎の個人単位の目標、ToDo、うまくいったこと、ダメだったところを記載する。

KPTT 表: 夕会時に、Keep (良かったこと)、Problem (問題だと認識していること)、Try (試してみたいこと) を記載し、その中から ToDo (次の日に試してみること) を決める。

また、1 日の始まりには朝会、終わりには夕会を行うこと、議論した内容を記録として残すことを義務付けた。

14 日間の PBL 期間の全体の流れを表 2 に示す。

MDD では、モデルからの動作検証により、機能面の検証はできる。しかし、責務が過大なクラスがある、クラス名がおかしいなどの質の悪いクラス図でも、その内部にステートマシンを記述することで、動くソフトウェアを作ることにはできる。品質や再利用性に優れたモデル図になるよう PBL 期間中に中間レビューを入れることで対応した。なお、表 2 の提供資料は、チームが必要だと気づいて要求した場合に限り提供することとし、機能を作るにはどんなものがあつたかをチーム内で考えるように誘導した。

4.3 長期 PBL

ET ロボコン¹¹⁾ を題材とし、MDD での開発を行う約半年間の PBL を実施することで、ソフトウェア開発において、モデルを利用した自律的な開発スキルの習得をさせることを目標とする。

4.3.1 ET ロボコン

ET ロボコン¹¹⁾ とは、組込みソフトウェア分野の技術教育をテーマにしたロボコンである。すべてのチームが同一のハードウェア (走行体) を使うことで、ソフトウェアの優劣

表 2 短期 PBL の流れ
Table 2 Schedule of the short period PBL course.

項目	内容, 到達目標, 提供資料など
キックオフ	PBL の説明
1 日目	グループワークの進め方, ロールプレイ
キックオフ	業務説明
2 日目	ミニ PBL による環境構築
PBL2 日目	目標: 走行部分の完成 応用編のモデル図を提供
PBL3 日目	通信ブリッジの提供
PBL4 日目	基地局プログラム(実行モジュール)の提供
PBL5 日目	目標: 通信部分の完成
PBL7 日目	目標: 基本の処理完成 中間レビュー
PBL9 日目	目標: 業務完成
PBL10 日目	目標: システムテスト
成果発表準備	発表内容の説明 発表準備, リハーサル
成果発表	成果発表, デモ 講評, ふりかえり

を競うコンテストであり, UML 等で記述された, 走行競技システムの分析・設計モデル内容の評価を行う「モデル部門」と, 自律型ライントレース・ロボットの走行競技による性能の評価を行う「競技部門」からなる。実施期間は, 3 月の実施説明会から 9 月の地区大会までのおよそ半年間である。教育をテーマにしたロボコンであるため, ET ロボコン実行委員会により, 2 回の技術教育が用意されている。5 月の技術教育 1 では, 開発環境・要素技術(2 時間)及びモデリングの文法を中心とした基礎的な教育(3 時間), 6 月の技術教育 2 では, モデルを作成するときに必要な考え方やポイントなど(3 時間), モデルをコードや実機環境にどう落とししていくか(3 時間)の教育が各チーム 1, 2 名を対象に実施される。さらに, 本番と同一のコースで走らせることができる試走会が 7 月, 8 月の 2 回開催され, 同一環境でのテストを行うことができる。

4.3.2 PBL の進め方

短期 PBL では, 開発の進め方を演習として準備し, 演習を進めることで, システムが開発する仕組みになっているため, 進め方に関する自由度は低かったが, 長期 PBL では, 半年間の開発期間をどのようなスケジュールで進めるのかや, 問題の解決方法などをチーム内で話し合いながら自由に進めるように促し, 自主的な開発能力を育成することを目指し

た。そのため, 教育と学生の関わりは, チームからの要望があった場合に指導を行う, チームリーダーから週 1 回メールを通じた進捗報告を行うというルールにした。

参加者は, 短期 PBL を受講した者の中から希望者とし, 開発時間は授業時間 10 時間及び放課後, 休日とした。

5. 基礎編・応用編の評価

開発した教育プログラムの有効性を検証するために, 専門学校生に対する実証講座を行った。前提知識の違いにより, 習得の変化を見るために 1 年生及び 3 年生を対象に実証講座を実施した。それぞれ, 講座 1, 講座 2 と呼ぶこととする。

5.1 講座 1

基礎編, 応用編の講座を専門学校の 1 年生を対象に実証講座を実施した。講座 1 の概要を表 3 に示す。受講者は, 入学後半年間 C 言語及び IT スキル標準レベル 2 を目指すカリキュラムを受講した学生である。なお, 基礎編の受講から応用編受講までの期間がおよそ 1 ヶ月であった。

表 3 講座 1 の概要
Table 3 Outline of the No.1 trial.

基 礎 編	受講者	専門学校の 1 年生 21 名
	前提知識	ポインタ, 構造体を除く C 言語の文法
	演習形態	ペア(1 ペアに PC1 台, ロボット 1 台提供)
	期間	3 日間(1 日 7 時間)
応 用 編	テスト	なし
	アンケート	講座内容, 知識・スキルに関して
	受講者	専門学校の 1 年生 20 名
	前提知識	基礎編を受講したもの
用 編	演習形態	ペア(1 ペアに PC1 台, ロボット 1 台提供)
	期間	4 日間(1 日 7 時間)
	テスト	UML に関する知識(2 日目) MDD に関する知識(4 日目)
	アンケート	講座内容に関して

5.2 講座 2

専門学校の 3 年生で, C 言語, JAVA, SQL, UML のスキルを持っている, IT スキル標準レベル 2 程度の学生に対して, 応用編の実証講座を行った。講座 2 の概要を表 4 に示す。受講者 9 名中 8 名が ET ロボコンの経験者であり, ロボットの組込みプログラミングのス

キルもあった。UMLの前提知識があったため、講座1で利用した教材からUMLの知識を除いたものとした。講義時間は、講座1の応用編の28時間に対して、講座2は20時間とした。

表4 講座2の概要
Table 4 Outline of the No.2 trial.

受講者	専門学校の3年生9名
前提知識	C言語, JAVA, SQL, UML
演習形態	ペア(1ペアにPC2台, ロボット1台提供)
期間	4日間(1日5時間)
テスト	UMLに関する知識(2日目) MDDに関する知識(4日目)
アンケート	講座内容に関して

5.3 評価

基礎編・応用編実証講座における評価として、3.1小節で述べたモデリング教育の3つの課題のうち、「理解性」、「モチベーション」に関する評価を行う。

5.3.1 理解性

「理解性」に関する評価として、応用編実証講座におけるUMLの知識及びMDD手法に関する確認テストを行ったUMLの知識がまったくなかった講座1の受講者のUMLの知識に関する正答率は、平均65%程度であったが、演習で利用したクラス図、ステートマシン図に関する問いでは、平均75%の正答率があり、MDDツールを用いて動かしながら学べたことで理解度が向上したと考えられる。また、MDD手法に関しては、講座2の受講者の平均正答率が80%と高く、UMLの前提知識があったため、MDD手法及びモデルを中心にした開発方法の理解に注力することができたためであるといえる。

応用編総合演習では、10個のマイルストーンを定義し、達成率を評価した。演習課題の達成率を表5に示す。提供したクラス図の実装されていない2つのクラスの内部のステートマシンを作成することで、業務を実現するソフトウェアができる。ほとんどの受講者が4番まで達成できたが、5番のマーカ検知でつまずく受講者が多かったため、それ以降の課題内容を実装できていない受講者が多かった。ただし、つまずいた原因は、マーカ検知のパラメータ調整の部分であり、動的モデリングスキルを身に着けることはできたといえる。

5.3.2 モチベーション

「モチベーション」に関する評価を行う目的で、各講座の受講者の取り組み姿勢に関する

表5 応用編総合演習課題達成率

Table 5 Achievement rate of the advance class exercise.

マイルストーン	講座1	講座2
1) ライントレース出来る	100 %	100 %
2) 側壁を検知して止まる	78 %	100 %
3) 荷下ろしを検知して回送する	78 %	100 %
4) 車庫に入って止まる	89 %	100 %
5) マーカを検知して振る舞いを変える	56 %	75 %
6) 右コースへ転進する	44 %	75 %
7) ラインエッジをトレースする	33 %	75 %
8) マーカを無視する	33 %	75 %
9) 反転する	33 %	50 %
10) 全てのパターンで完走する	11 %	50 %

アンケートを実施した。その結果を表6に示す。

表6 取組み姿勢のアンケート結果

Table 6 Motivation survey results.

	講座1		講座2
	基礎編	応用編	
とても熱心であった	10 %	16 %	25 %
熱心であった	38 %	21 %	63 %
どちらかというと熱心であった	24 %	37 %	13 %
あまりやる気がなかった	29 %	26 %	0 %
やる気がなかった	0 %	0 %	0 %
全くやる気がなかった	0 %	0 %	0 %

基礎編の実証講座よりもMDDを教育した応用編の方が「とても熱心」に取り組んだ受講者が多く、MDD教育によりペアやチームでモデル図を見ながら一緒に開発できたこと、MDD手法という新しい方法を学べたことがモチベーションの維持につながったと考えられる。

5.4 考察

専門学校の1年生に対して実施した講座1では、「難易度が高かった」、「時間が短すぎた」という受講者のコメントが多かった。前提知識がほとんどない状態で、組込みソフトウェアの開発、モデリング、MDD手法など幅広い分野を短期間の実証講座に盛り込みすぎたことが原因であった。また、講座を集中講義として実施したことで、知識の定着する期間がな

かったといえる。ただし、短期間の実証講座期間に演習課題の一部の業務を、全受講者が実現できていたこと、モデル図をみんなで議論できていたこと、モデルの必要性を理解できるようになったことを踏まえると MDD による開発方法やモデリング手法を短期間で習得できるプログラムであったと考える。

一方、同一教材を用いて対象を変えて行った講座 2 では、受講者がプログラミング、モデリングの前提知識があったため、モデルを中心に開発する方法や、MDD 手法の理解が高かった。これらのことより、同一の演習課題であっても、受講対象に合わせた、情報提供をすることで、教育項目のバリエーションを増やすことができる。MDD では、ドメインを分割して開発することができるため、情報提供に変化を持たせやすいという利点を確認できた。

MDD をソフトウェアモデリング教育に活用する場合の問題点としては、受講者が MDD で評価できる機能面の完成に強く意識をとらわれることで、モデルの品質への意識が疎かになってしまうことである。本プログラムでは、レビューを入れることで対応したが、今後さらにモデルの品質に関して意識を持たせる工夫が必要である。今後の運用方法としては、通常の半期または通年のカリキュラムとして、毎週 1 コマ実施するなどの形態が望ましいと考える。

6. 2 段階 PBL の評価

6.1 PBL 実証講座の概要

基礎編・応用編の実証講座を受講した専門学校の 1 年生 17 名（1 チームあたり 4、5 名）に対して、短期 PBL を実施した。期間は、キックオフ 2 日間、PBL 期間 10 日間、成果発表 2 日間である。短期 PBL に参加した学生の中の希望者 8 名（1 チーム）が 1 年生の 3 月から 2 年生の 9 月までの約半年間長期 PBL に参加した。短期 PBL の実施工程は、表 2 の通り実施した。開発工程を学生に自由に設定させた長期 PBL の工程は、図 5 の通りである。月 1 回程度、ET ロボコン実行委員会が実施する教育や試走会などのイベントがあるため、それに合わせたスケジュールを 3 月に計画し、進捗状況により、適時スケジュール変更を行った。8 名の参加者のうち、主にモデル作成を担当するソフトウェア設計チームと走行戦略の立案とハードウェアの調整を担当する戦略設計チームに分かれて開発を進めた。なお、ET ロボコン用走行体でライトレースのみを行う BrigePoint モデルは、開始時に教員から配布し、改良して利用してもよいこととした。

6.2 評価

2 段階 PBL における評価として、3.1 小節で述べたモデリング教育の 3 つの課題に関する

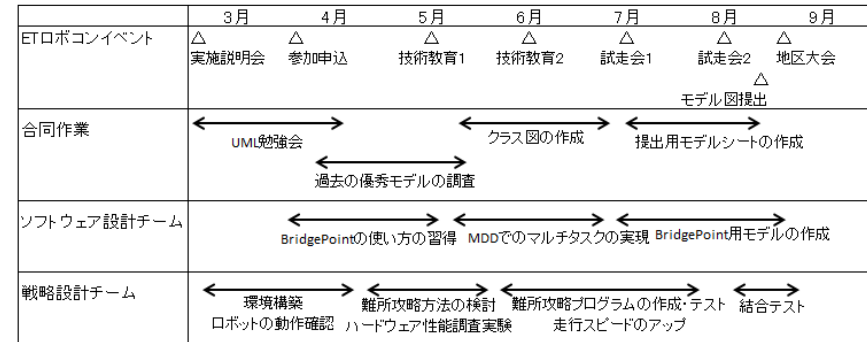


図 5 長期 PBL の実施工程

Fig. 5 Time schedule of long period PBL.

る評価を行う。

6.2.1 理解性

短期 PBL では、仕様の変更に伴い、クラスの責務を整理してクラス図を作成するかがメインのテーマであった。PBL の 7 日目に講師によるクラス図の中間レビュー、最終日に成果発表を行った。追加業務の内容を踏まえて、応用編のクラス図から変更を期待する項目を評価項目とし、達成率を評価した。評価結果を表 7 に示す。

表 7 短期 PBL 実証講座のクラス図評価結果

Table 7 Evaluation Results of class diagrams in the short period PBL trial.

評価項目	中間レビュー	成果発表
1) バンパクラスの削除	60%	80%
2) 荷台クラスの多重度の変更	0%	20%
3) 基地局との通信を行うクラスの追加	40%	80%
4) 搬送ルートを管理するクラスの追加	0%	60%
5) その他、責務に応じたクラスの追加	0%	80%

通信を行うクラスに関しては、Test という名前で実装済みのものを受講者に配布した。中間レビュー時には、Test という名前そのまま追加されているチームが多かったが、その場合は、3) 番未達成とした。また、応用編でバンパとして使用していたタッチセンサを PBL 編では荷台として利用しているにも関わらず、バンパクラスが残っている、責務分割がうまくなされておらず、クラス図のステートマシンが大きなものになっているなどが特徴的であった。

中間レビュー前は、課題の機能完成を急ぎ、クラス設計をせずに開発を進めているチームが大変多かった。しかしながら、中間レビュー以降は、各班が積極的にモデルのチーム内レビューを実施しており、成果発表時には、期待したクラス図に近いものになった。達成率の低い荷台クラスの多重度に関しては、多重度を増やすべきと認識しているチームが複数あったが、それに伴いアクション記述をどのように記述したらよいかかわからず、荷台クラスを2つ作ることで対応しているチームが多かった。この点に関しては、事前にアクション記述の教育する時間を設けることで、対応できると考える。クラス図レビューが終了3日前で定期的に遅かったが、その後クラス図の再検討、クラス内部の振る舞いの実装を行い、成果発表時には全チーム課題内容の7割程度の業務が実現できており、MDDを用いたことで、レビューから改善のスピードが早くなったことを確認した。MDD教育と適切なタイミングでのレビューとを組み合わせることで、短期間でモデリングスキルの向上が図れることが分かった。ただし、受講者アンケートによると全員が課題の難易度が高かったと答えており、期間内にシステム全体が完成する難易度にするすることで、受講者の満足度や達成感が高まると考えられるため、今後は事前指導の期間を増やすことや難易度を調整することを検討する必要がある。

長期PBLでは、終了後の参加者アンケートから、参加者の9割が設計やモデリングのスキルが向上したと答えている。さらに、ETロボコン九州地区大会では、モデルの正確性、理解性、設計品質や性能などに関する評価が行われ、B+評価(全ランク:A+ ~ D-の12段階)を得ることができ、企業や大学生のチームを含めた35チーム中3位で学生チームの中では1位であり、本教育プログラムにより、モデリングのスキルの向上が図れたといえる。

6.2.2 モチベーション

PBLの取組みに関するアンケート結果を行ったところ、短期PBLでは半数の学生が「熱心に取り組んだ」と答えており、「どちらかという熱心に取り組んだ」という学生を入れると、9割の学生がモチベーションを保っていた。また、長期PBLでは「熱心に取り組んだ」と答えたが学生が6割、「どちらかという熱心に取り組んだ」という学生を入れると全員がモチベーションを保って活動していた。さらに、授業時間以外の放課後や休日に積極的に活動をしていた。開発内容だけではなく、コンテストへの取組みという形態がモチベーションを保つ要因であると考えられる。

6.2.3 協調性

短期PBLでは、「チーム内の情報共有はできましたか?」という問いに対して、半数が

「できた」と答えており、「どちらかというできた」という学生を加えると7割の学生ができていた。タイムボックス制を基本としたガイド付きPBLに関しては、9割の学生がプロジェクトを円滑に進めるツールとして「効果的であった」と答えており、グループでの活動を支援することができたといえる。また、長期PBLでは、PBLツールの利用を特に強制しなかったが、長期・短期のガントチャートを作成する、タイマーを用いて時間管理をする、目標に合わせた計画を立て実行し、KPTT表を用いてふりかえりをするなどの活動を独自に実施しており、短期PBLでプロジェクトの進め方や計画の立て方を学ぶことができていたことが確認できた。さらに、長期PBL終了後のアンケートでは、学べた内容として技術的側面より、チームメンバーとの接し方やストレスコントロールの仕方、プロジェクトマネジメントなどプロジェクト運営に関わる部分をあげている学生が多く、チームの一員としての働き方を習得することができていたといえる。

6.3 考察

長期PBLで採用した、MDD及び2段階PBL活用の効果を評価するため、2008年~2010年過去3年間のETロボコン参加チームと比較して考察を行う。2008年~2011年のETロボコンへの取り組みの結果を表8に示す。

表8 2008年~2011年のETロボコンへの取組み結果
Table 8 The activity of ET robot contest from 2008 to 2011.

年度	2008年	2009年	2010年-1	2010年-2	2011年
チーム人数	2人	4人	4人	4人	8人
実装言語	C言語	C言語	C言語	C++	C言語
モデル作成の時期	8月	8月	7月, 8月	6月~8月	5月~8月
MDDの利用	なし	なし	なし	なし	あり
モデル審査評価	C	B	B	C+	B+

MDDを利用せず、長期PBLのみであった、過去3年間の参加チームでは、4名程度と比較的少人数であり、その中でも1,2名の学生がほとんどの活動を行う形態となり、チーム内で分担した活動はできていなかった。さらに、モデルを描く時期は、モデル審査の直前であり、出来上がったコードにあうようにモデルを描いていたため、モデルの効果的な利用が出来ていなかった。本論文で提案したMDD及び2段階PBLを実施した2011年の取組みでは、開発早期からモデル作成を行っており、モデルを利用した開発を行っていた。さらに、チーム内全員のスキル向上のため、クラス図を全員記述し、教員を交えたレビューの結果一番良いモデルを初期バージョンモデルに採用するという方式を実施し、特定のメンバー

だけがモデリングを行っていた過去の大会と比べて、効果的なモデリング教育が出来ていた。チーム内での作業分担がうまくいった要因としては、MDD を利用したことで、各クラス内の動的な振る舞いを独立して作成することができたことや短期 PBL でチームメンバー全員がプロジェクトの進め方や円滑に進めるためのツールの使い方を習得していたことがあげられる。

ET ロボコンに MDD を活用した取組みの問題点としては、ET ロボコンのような小規模な開発の場合、MDD の開発基盤の整備にかかる時間や BridgePoint 等の MDD ツールを使いこなすのにかかる時間が相対的に大きなものとなり、結果的にハンドコーディングの方が早く開発できてしまうため、MDD のメリットを感じることができないことである。参加した学生のアンケートでは、「MDD ツールの使い方に苦勞し時間がかかりすぎた」や「モデルコンパイルに時間がかかるため、ハンドコーディングの方がよかった」などの意見があった。

応用編や短期 PBL では、教員が開発環境をそろえて実施したため、小規模の開発でも問題なく実施できたが、開発手順や問題解決方法を学生の自由に開発させた長期 PBL では、MDD の開発基盤の整備も学生自ら行う必要があり、MDD の有用性を感じてもらいにくかったといえる。今後は、前年度モデルの再利用を積極的に促進することで、MDD の有用性を感じられる教育プログラムとする必要があると考える。

7. おわりに

本論文では、ソフトウェアモデリング教育における課題として、「理解性」、「モチベーション」、「協調性」をあげ、その課題の解決に MDD 及び 2 段階 PBL を活用することを提案した。専門学校での実証講座では、MDD と適切なタイミングでのレビューを組み合わせることで、モデリングスキルの向上が図れることを確認した。また、受講者へ提供するモデル図や開発環境を変化させることで、受講者にレベルに合わせた教育が行えることも分かった。さらに、過去の ET ロボコンの参加結果と提案教育プログラムを利用した 2011 年度の参加結果を比較し、MDD 及び 2 段階 PBL の活用の有用性を確認した。

今後は、実施期間、教育対象者と情報提供内容及び課題内容の組合せ、レビュー時期とレビュー項目の検討、PBL の効果的なガイド方法の検討等を行い、さらにモデリングスキル向上が行える教育プログラムの開発を行う。

謝辞 本教育プログラムの開発は、文部科学省研究拠点形成費等補助金（産学連携による実践型人材育成事業）による助成のもとで行った。

参考文献

- 1) 独立行政法人情報処理推進機構：組込みソフトウェア開発における品質向上の進め [設計モデル編]，アイティメディア (2006)。
- 2) 新井玲子：UML オブジェクト指向モデリング セルフレビューノート，ディー・アート (2005)。
- 3) 香山瑞恵，二上貴夫，Starrett, C.，今野篤志：Model Driven Development に基づく抽象化概念教育の提案 (2010)。
- 4) Starrett, C.: Teaching UML Modeling Before Programming at the High School Level, *Advanced Learning Technologies, IEEE International Conference on*, pp.713–714 (2007)。
- 5) 中尾信明：オブジェクト指向、UML に関する教育の視点と実践，情報処理学会研究報告，2004-CE-74， Vol.2004, No.49, pp.9–16 (2004)。
- 6) 高井久美子，渡辺博芳，佐々木茂，鎌田一雄：オブジェクト指向モデリング教育における個別学習と協調学習を組み合わせた授業の設計と実践，情報処理学会研究報告，Vol.2009-CE-101, No.1, pp.1–8 (2009)。
- 7) 小木曾禎，遠山紗矢香，湯浦克彦：学生向けモデリング演習支援システムの開発と評価，情報処理学会研究報告，Vol.2011-CE-109, No.5, pp.1–9 (2011)。
- 8) 沢田篤史，小林隆志，金子伸幸，中道 上，大久保弘崇，山本晋一郎：飛行性制御を題材としたプロジェクト型ソフトウェア開発実習，情報処理学会論文誌，Vol.50, No.11, pp.2677–2689 (2009)。
- 9) 松澤芳昭，塩見彰睦，菟川友宏，酒井三四郎：ソフトウェア開発の教員主導型 PBL における反復プロセスと EVM 導入の効果，情報処理学会研究報告，Vol.2009-CE-99, No.9, pp.1–8 (2009)。
- 10) 金田重郎，井上 明：実システム開発を通じた社会連携型 PBL の提案と実践，情報処理学会研究報告，Vol.2009-IS-107, No.32, pp.185–192 (2009)。
- 11) ET ロボコン実行委員会：ET ロボコン 2011. <http://www.etrobo.jp/2011/>。
- 12) 二上貴夫：I 見聞録：MDD ロボットチャレンジ 2010，情報処理， Vol.52, No.4, pp.572–576 (2011)。
- 13) Liggesmeyer, P. and Trapp, M.: Trends in Embedded Software Engineering, *IEEE Software*, Vol.26, No.3, pp.19–25 (2009)。
- 14) Mellor, S.J. and Balcer, M.J.: Executable UML-MDA モデル駆動型アーキテクチャの基礎-，テクノロジックアート (2003)。
- 15) 赤山聖子，久保秋真，久住憲嗣，二上貴夫：プロジェクトファシリテーションツールを活用した初学者向けソフトウェア開発 PBL，情報処理学会研究報告，Vol.2011-CE-111, No.14, pp.1–10 (2011)。
- 16) 平鍋健児，天野 勝：プロジェクトファシリテーション価値と原則編，Vol.8 (オンライン)，入手先(<http://www.objectclub.jp/community/pf/>) (2011)。