

# UML エディタからのコード生成によるロボット制御実習

A Practice on Robot Control Training using codes generated by UML Editor

柴田 敦也<sup>†</sup>  
Atsuya Shibata

神原 弘之<sup>‡</sup>  
Hiroyuki Kanbara

武苗 棟之<sup>§</sup>  
Muneyuki Takenae

## 1 はじめに

ソフトウェアの再利用性を高めるため、情報システムの開発にはオブジェクト指向言語が幅広く用いられている。情報システムの規模や複雑さは年々増大しており、システム開発に UML (Unified Modeling Language) を用いる需要が高まる事が予想される。UML は主に以下のような用途でソフトウェア開発の現場で幅広く用いられている。

- オブジェクト指向に基づくシステムのモデリングと要求分析
- 顧客と開発者あるいは開発者間で共通の理解を得るためのソフトウェアの内部設計資料
- 制御プログラムの入力インタフェース [1]

大学等の教育機関における本格的なソフトウェア開発実習 [2] や企業での新人教育では、UML を用いたシステム設計とオブジェクト指向言語によるプログラミングが実施されている。その一方、電気電子や機械工学を中心とした学部/学科では、前述したようなソフトウェア開発実習に十分な時間を配分することは難しい。実製品あるいは社会インフラにおいて組込みソフトウェアの果たす役割が増大している中、情報分野の専攻でない学生が卒業までに UML を利用できるためにどのようにして教育するかは、課題の一つとなっている。

本稿では、ソフトウェア工学の講義に UML 実習を組込んだ背景、UML 実習を実施するために開発した計算機上のソフトウェア環境、ならびに 2016 年度に初めて実施した UML 実習の概要を紹介する。

## 2 UML 実習を組込んだソフトウェア工学の講義

立命館大学理工学部電子情報工学科では、学部 3 回生の後期にソフトウェア工学の講義が実施されている。2015 年度までは、ソフトウェア工学の教科書 [3] を用いた 1 時間半の座学 15 回からなる講義が行われてきた。15 回の

講義の内 2 回をオブジェクト指向分析の講義に割当て、その内の 1 回で UML の解説を行ってきた。教科書中では、ネット販売システムを例題にして、ユースケース図、クラス図、シーケンス図といった UML の主要な図について俯瞰的な解説がなされており、情報システムの開発経験者にとってはわかりやすい例題が示されている。しかし、学生は日常的にネット販売システムを利用していても、情報システムとしての振る舞いを具体的に把握するにいたっていない。このため、ネット販売システムの例題を講義で解説するだけでは UML による図形表現の意味を理解する事が困難である学生が散見された。そこで 15 回の講義のうち 3 回を、計算機 (PC) を使用できる環境での実習に振替え、以下のような UML の導入教育を実施した。

- 学生が PC 上で実際に UML の入力を行う
- 自らが入力した UML によりシステムがどのように制御されるかを体験する

具体的には次の課題を実施した。

- 学生が UML の「クラス図」と「状態遷移図」を入力することでロボットの動作を制御する

この課題を実施するにあたり、PC 上のソフトウェアをどのように操作すべきかを、「シーケンス図」を用いて説明した。これにより UML の「クラス図」「状態遷移図」「シーケンス図」を理解するための「素地」を育むことを目指した。

実習に参加した学生の大半の知識は学部 2 回生前期に実施される C 言語プログラミング実習の内容に基いており、複雑なプログラミングの経験がない。さらに Java や C++ といったオブジェクト指向のプログラミング言語を使用した経験がないため、「クラス図」と「シーケンス図」を用いてソフトウェアを設計し、オブジェクト指向プログラミング言語を用いて実装する事は、本実習の対象から外した。また、ソフトウェア設計と実装に至る前段階での「ユースケース図」と「概念クラス図」からの「クラス設計の詳細化」も対象から外した。

<sup>†</sup> 奈良先端科学技術大学院大学, NAIST

<sup>‡</sup> 京都高度技術研究所, ASTEM RI

<sup>§</sup> 立命館大学, Ritsumeikan University

### 3 UML によるロボット制御実習用の計算機環境

#### 3.1 Roomba ロボットのシミュレータ

iRobot 社の Roomba は、室内を一定時間自律走行して床の掃除を行う。この通常モードの他に、以下の特別な動作モードを備えている [4]。

- シリアル通信ポート経由で外部から送られてきた制御コマンドに従い、走行の方向、速度、旋回、掃除等の機能を詳細に制御する
- 内蔵センサで取得した情報を、シリアル通信で外部の PC に送信する

合宿形式の組込みシステム開発の勉強会：LED-Camp (Learning Embedded software Development Camp) では、この動作モードを利用した以下のような実習課題を実施することで、UML による設計の正しさを評価している [5]。

- UML エディタで入力した状態遷移図から制御コマンドを自動生成する
- 自動生成された制御コマンドを実物の Roomba ロボットに入力し、その走行を制御する

しかし多数の学生が参加する大学の講義で、受講者の人数に応じたロボットを必要台数確保し、かつその走行を試行できる環境を用意する事は困難である。そこで本実習では、以下のような簡易版の環境を用意することにより、LED-Camp と同様の Roomba ロボット制御の実習を行う事にした。

- 限られた時間内で学生が把握できる簡易版の制御コマンドを再定義する
- 簡易版のコマンドに従って Roomba ロボットがどのように走行するかを PC 上で模擬できる、ソフトウェア・シミュレータを開発する

この簡易版の制御コマンドの一覧を表 1 に示す。また、開発したソフトウェア・シミュレータを起動した直後の画面を図 1 に示す。

図 1 に示す画面は、四角い部屋の真ん中に○印で示された Roomba ロボットが静止状態にあることを示している。この状態から「Forward」コマンドを送ると、黒丸印がついた方向に前進を開始する。黒丸印のついた面と部屋の壁が接触すると Roomba は前進できなくなり、同時に「GetSensor()」コマンドからの返り値は「NO\_HIT」から「HIT」に変化する。次章で説明する UML の状態遷移記述では、「後退」「停止」「左(右)回転」「左(右)旋回」などの別動作を行う次状態へのトリガー条件を、この壁にぶつかった状態を検知することで実現できる。この他にも、状態遷移後に内蔵タイマの値を「ResetTime」コマンドでリセットすることで、タイマの値がある一定値までインクリメントされたことを検知して、別のコマンドを送信する状態に遷移するトリガー条件も設定できる。

表 1 UML 実習用 Roomba の制御コマンド

走行の制御	
Forward	前進する
Stop	停止する
Backward	後退する
SpinLeft	左回転する
SpinRight	右回転する
TurnLeft	左旋回する
TurnRight	右旋回する
搭載センサからの情報の取得	
GetSensor()	壁にぶつかると HIT を返す それ以外では NO_HIT を返す
タイマを制御するコマンド	
ResetTime	内蔵タイマの値を 0 にリセットする リセットされないと、内蔵タイマの値はインクリメントし続ける

本シミュレータは、壁や障害物の存在をプログラム中に追加することにより、部屋のどの領域を Roomba が走行可能かを制限できる。通過可能な領域(迷路)を走破する条件を UML を用いて記述し、その走行結果をシミュレーションにより確認できる。さらに、部屋のどの箇所が清掃されたかを表示するため、Roomba ロボットの走行軌跡を画面上に残す機能も備えている。

Processing 言語を用いて実装された本シミュレータは、Windows あるいは MacOSX 上で動作する。

#### 3.2 UML エディタ

本実習では UML エディタに astah\* professional を使用した。前述したシミュレータ上の Roomba は実物と同様、外部から与えられた制御コマンドに応じてその走行を制御することができる。学生が UML 記述をエディタで編集することで、シミュレータにより解釈実行される制御コマンドも自動的に変化する点が、実習の効率面で望ましい。LED-Camp に倣い、シミュレータ上の Roomba に制御コマンドを送信するプログラムを UML の状態遷移図を元に自動生成することにした。このため、コマンド発行プログラムを自動生成する、astah\* のプラグイン・モジュールを新たに開発した。

### 4 UML によるロボット制御実習の実施

本章では、2016 年 12 月に実施した、UML を用いた Roomba ロボットのシミュレーション・モデルの制御を行う実習について述べる。

本実習は、1 時間半の実習 3 回からなり、ソフトウェア工学の科目を履修する立命館大学理工学部電子情報工学科の学生 85 名が参加した。Windows PC 1 台を学生 2 名が共有し、各自 UML エディタを操作して課題に取り組んだ。実習は、講師 1 名が前で操作手順を説明し、ティーチングアシスタント (TA) 2 名が学生のサポートを行った。実習の様子を図 2 に示す。

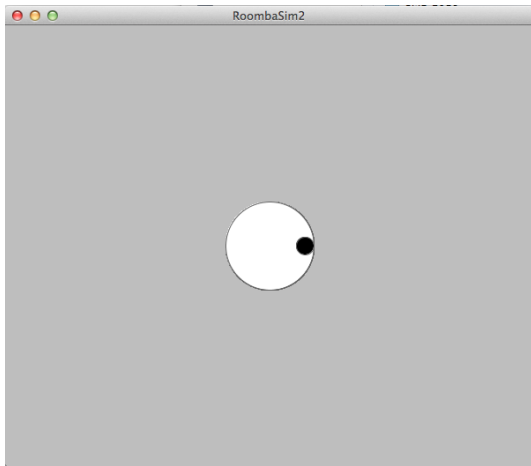


図1 Roombaのシミュレータ画面



図2 立命館大学での実習の様子

#### 4.1 第1回の実習内容

第1回の実習では、UML エディタでクラス図と状態遷移図の入力、制御プログラムの自動生成を行うことで、シミュレータ上の Roomba ロボットを制御する手順を習得する事を目標とした。最終のロボット制御までたどり着かない一部の学生に対しては、TA と講師が PC の画面を見て解決策を提示するといったサポートを行った。学生にとって UML エディタの操作は初めての体験であり、第1回の実習の前半では、講師と TA は学生のサポートに忙殺された。一方実習の後半には、課題を完遂した学生が PC を共有するもう1名の学生をフォローする事で、参加した学生全員が目標を達成することができた。この第1回の実習で学生が取り組んだ状態遷移図を図3に示す。

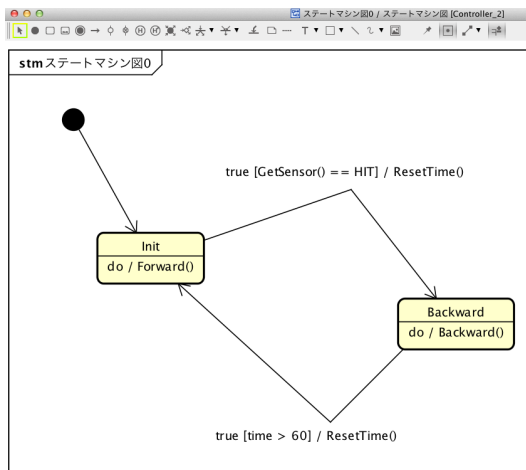


図3 UML エディタで入力する状態遷移図

#### 4.2 第2回の実習内容

第2回の実習では、第1回の実習と比べて複雑な動きを表現する以下のような仕様を満たす状態遷移図を作成し、Roomba を制御することを目標とした。

- 真っすぐ進んで、壁にぶつくと 180°回転し、再度走り出す動き (往復運動)

- 画面の四隅を經由して走行する動き (4つの頂点を必ず通過する)
- 十字に走行する動き
- 8の字に走行する動き

第2回の実習では UML エディタの操作に慣れた学生は、動きを実現する状態遷移図の考察およびその問題の解消に注力できるようになり、講師と TA の負担は第1回の実習と比べて大幅に軽減された。

#### 4.3 第3回の実習内容

第3回の実習では、以下の2点を目標とした。

- 第2回の実習で提示した課題をすべてクリアする
- すべてクリアした学生は、より複雑な迷路を走破する状態遷移図を作成する

学生は、実習室を利用できる時間一杯かけて迷路の走破を目指すなど、意欲的に実習に取り組んでいた。第3回の実習に提示した「より複雑な迷路」を図4に示す。

実習のまとめとして、以下の内容について述べるレポートを後日提出させ、これをもって実習完了とした。

- 作成した状態遷移図とそのアルゴリズムの詳細
- シミュレータ上のロボットの動作 (走行軌跡)
- 工夫した点

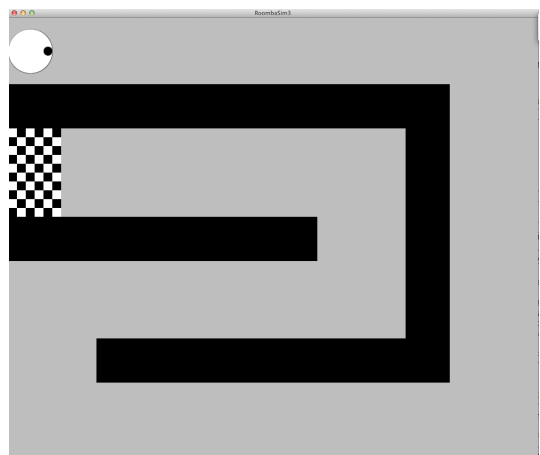


図4 迷路のシミュレータ画面

## 5 おわりに

情報工学の専攻でない学生を対象とした, UML を用いてロボットを制御する実習を計画し, そのためのソフトウェア環境を整備した. 本稿では, 実習における計算機環境と提示した課題の内容, 講師および TA のサポート体制の概要を報告した. 本実習は多くの学生が意欲的に取り組んでいたことから, UML の導入教育として一定の効果があったことが期待される. なお, 学生の UML の理解度向上に本実習がどれほど貢献したのかを, 定量的に測定可能な課題を設定することが将来の課題である.

### 謝辞

本実習を実施するにあたり計算機環境の整備にご協力頂いた, 立命館大学の富山宏之教授と熊木武志准教授に感謝いたします. 今回の実習を計画するにあたり, LED-Camp に参加することで多くの知見を得る事ができました. LED-Camp 実行委員会の皆様に感謝いたします. UML エディタからのコード生成方法について, 指導いただいた細合晋太郎氏 (チェンジビジョン) に感謝いたします.

### 参考文献

- [1] 水川真, 坂本武志, 大原賢一: "UML と RT ミドルウェアによるモデルベースロボットシステム開発," オーム社, (2009).
- [2] 松浦佐江子: "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 (2010).
- [3] 高橋直久, 丸山勝久: "ソフトウェア工学," 森北出版, (2010).
- [4] Tod E. Kurt: "Hacking Roomba," Wiley (2007).
- [5] "LED-Camp," <http://swest.toppers.jp/LED-Camp/> .