

モデル駆動開発方法論に基づく UML モデリング教育環境における汎用性を意識したロボット動作設計向けモデルコンパイラの開発

Development of a General Model Compiler on Model-Driven Development Environment for Designing Operation Control Method for Robot

但馬 将貴^{*1}, 香山 瑞恵^{*1}, 小形 真平^{*1}, 橋本 昌巳^{*1}
Masaki TAJIMA^{*1}, Mizue KAYAMA^{*1}, Shinpei OGATA^{*1}, Masami HASHIMOTO^{*1}

^{*1} 信州大学工学部

^{*1} Faculty of Engineering, Shinshu University

Email: 11t5041a@shinshu-u.ac.jp

あらまし：本研究の目的は、UML モデリング教育を支援する環境の拡張である。ここでは Web ブラウザ上で動作するモデル駆動開発ツール：clooca を対象に、ロボット動作をクラス図と状態遷移図とでモデル化する状況を想定する。モデリング対象のロボット毎に定義が必要な事柄である、学習者が記述する UML モデルの要素を規定するメタモデル、モデル化際の語彙を定義する DSL、UML モデルからプログラムコードを生成するテンプレートの 3 点に関して、汎用性を意識した設計を試みた。本稿では、これらの成果と授業実践への適用結果について記述する。

キーワード：モデル駆動開発, MDD, UML, モデリング教育, ロボット, モデルコンパイラ

1. はじめに

近年、情報系の大学や専門学校等におけるモデリング学習では、モデル駆動開発 (Model Driven Development, 以下 MDD) 方法論が取り入れられ始めている⁽¹⁾。この学習では、抽象度の低いソースコードを直接打ち込むプログラミングに代わり、抽象度の高いモデルを描く。このモデルからソースコードが生成されるため、プログラミングを行うことなく対象デバイスでの動作を視覚的に確認しながら、モデルを修正することで学習を進めることとなる。

本研究の目的は、MDD ツール：clooca⁽²⁾を用いて、モデリング学習環境の拡張を行うことである。clooca は現在、信州大学工学部情報工学科において、UML モデリング学習に利用されている⁽³⁾。本稿では、このモデリング学習におけるロボット動作設計向けの汎用性を意識したモデルコンパイラの開発について述べる。

2. MDD ツール：clooca

clooca は、IPA 未踏プロジェクトでの成果をもとに開発された、Web ベースのクラウド型 MDD ツールである。図 1 に、clooca の機能の概要を示す。ここでは、開発者がモデルコンパイラツールの設計を行い、利用者が設計されたツールを利用してモデル駆動開発を行う。開発者は、1 つのツールごとに、モデルの定義となるメタモデル、モデルを記述する際の語彙となる DSL (Domain Specific Language)、モデルからソースコードを生成する規則を記述するテンプレートの設計を、それぞれ行う。このとき、対象デバイスが異なっても、メタモデルと DSL はデバイスに非依存であるため、共通化できる。一方、テンプレートはデバイスに依存した設計になる。

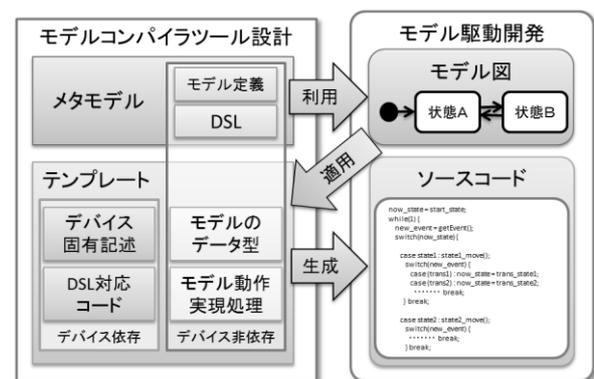


図 1 clooca の機能概要図

3. モデリング教育向けモデルコンパイラツールの設計

複数のデバイスを対象としたとき、DSL を含むメタモデルを共通とすれば、テンプレートのデバイス依存の部分のみ書き換えることで、異なるデバイスでも同様に動作するツールが提供できる。

3.1 メタモデル

メタモデルとは、モデル作成時に使うモデル図の記述要素の定義である。要素の名称、形、属性や、要素間の関連などを定義する。ここでは、簡易的な状態遷移図の定義を例として挙げる。記述要素は、「状態」と「遷移」とする。2 つの状態は遷移で関係付けられる。状態には開始状態、通常状態、終了状態の 3 種がある。開始状態の形は黒丸、終了状態は二重丸とする。通常状態の形は角丸矩形であり、動作という属性を与える。一方、遷移は単方向矢印とする。遷移の属性は名前とし、それをイベントと名付ける。この定義に従って、記述された状態遷移図の例を図 2 に示す。

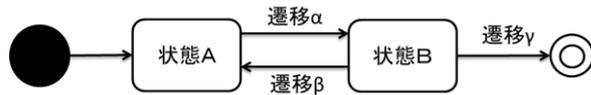


図2 状態遷移図のモデル例

表1 DSL の例
(a) 動作の DSL

動作	説明	対応コード
前進	両輪のタイヤを正回転	go()
停止	両輪のタイヤを停止	stop()
右旋回	左のタイヤのみ正回転	turn_r()
左旋回	右のタイヤのみ正回転	turn_l()

(b) イベントの DSL

イベント	説明	対応コード
黒線に入った	光センサ値340以下	getIR(PortC)<= 340
白地に入った	光センサ値680以上	getIR(PortC)>= 680

3.2 DSL

メタモデルのうち、記述要素の属性に対する語彙を定義するのが DSL である。状態遷移図の場合、状態の動作として「前進」や「停止」、遷移のイベントとして「荷物が載った」や「黒線に入った」などの語彙を、課題や学習者の習熟度に応じて定義する。

3.3 テンプレート

テンプレートとは、記述されたモデルからソースコードを生成する規則である。テンプレートを用いることで、モデル図の要素や構成から、ソースコードへの埋込や対応付けを行う。テンプレートには、デバイス依存の記述と非依存の記述がある。デバイス依存の記述には、デバイス固有の記述や DSL との対応コードを含む。非依存の記述には、モデルのデータ型やモデルの動作を実現するための処理を含む。

ここでは、対象デバイスに依存する部分が少ないテンプレートを例として挙げる。この例では、モデルの動作を実現する処理にキューを導入する。発生イベントの保持にキューを取り入れることで、センサ入力やタイマー、変数の比較を処理できる。そのため、対象デバイスや使用するセンサを変えても、デバイスに非依存な部分の処理アルゴリズムを変更する必要はない。しかし、キューの動作を表現するために処理アルゴリズムが複雑になり、テンプレート全体の可読性は低くなる。

4. 教育実践への適用

大学1年生を対象としたモデリング教育向けのモデルコンパイラツールを設計した。対象デバイスとしたロボットは、NXT Mindstorm LEGO (Mindstorm社)、iCreate Robot (iRobot社)、Robotist (Artec社)の3種類である。これらのロボットでは、外部センサの値をイベントとして入力し、モータ制御値の組み合わせを状態として出力する。ここでは、課題や学習者の習熟度に応じてメタモデルを定義した。

【メッセージを含む状態遷移図】

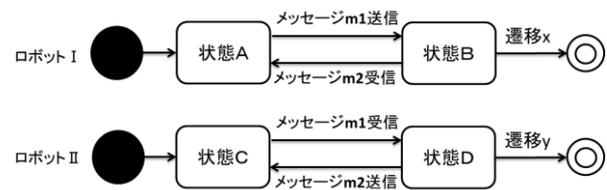


図3 段階的なモデル例

4.1 課題に応じたメタモデル定義の例

白紙の上の黒線に沿って走るラインレース課題を取り上げる。この課題におけるロボットの入力は光センサ、出力は左右のモータ制御値である。この課題のメタモデルでは、図2に示す状態遷移図を定義した。また、課題に応じたDSLとして、表1(a),(b)を定義した。表1中、対応コードは同行の状態名やイベント名が用いられた場合に、ソースコードに埋込まれる対象デバイスのライブラリ関数である。

3種類のロボットに対して、テンプレート中のデバイス依存の記述部分のみを、各ロボットのライブラリ関数を用いてそれぞれ作成した。その他の、メタモデルとDSL、およびテンプレートにおけるデバイス非依存の記述は共通のものを利用した。

4.2 習熟度に応じたメタモデル定義の例

2台のロボットが協調動作を行う課題を取り上げる。ここでは、ロボット同士がBluetoothにより通信する。この課題のメタモデルには、クラス図と状態遷移図を導入する。クラス図では、2台のロボットに相当するクラスと属性を定義する。状態遷移図では、名前と動作を属性とする状態および、他の状態遷移図へのイベントを記述できるメッセージを属性とする遷移を定義した。このメタモデルで記述した状態遷移図の例を図3に示す。これは、図2よりも難易度が高い記法である。学習者の習熟度に応じて簡易的な記法からフルスペックでのUML図まで定義することができる。

5. おわりに

本稿では、MDDツール: cloocaを対象として、汎用性を意識したロボット動作設計向けモデルコンパイラツールの開発と、その成果を述べた。今後は、センサ入力とモータ出力で動作するロボットだけでなく、さらに多様な対象物に対するツールを設計し、対象デバイスの入力と出力のタイプに応じたツール設計の方法論として整理したい。

参考文献

- (1) 赤山聖子他：オブジェクト指向モデリング教育におけるモデル駆動開発ツールの活用方法の検討，情報処理学会論文誌，vol.55，no.1，pp.72-84 (2014).
- (2) S. Hiya, et al. : clooca : Web based tool for Domain Specific Modeling, Proc. of MODELS2013, pp.1-5 (2013).
- (3) clooca for Shinshu Univ., <http://mdd.shinshu-u.ac.jp/> (accessed 2015/2/16).