

LEGO MINDSTORM EV3を用いた自律ロボットプログラミング教育の取り組み —ステートマシン図によるプログラミングツールの開発—

Endeavor for the Autonomy Robot Programming Education with LEGO MINDSTORM EV3

-Development of the Programming Tool with the State Machine Diagram-

北條 史華*1*a, 野瀬 裕昭*1*b

Fumika HOJO*1, Hiroaki NOSE*1

*1長野県工科短期大学校 情報技術科

*1Computer & System Engineering Dept., Nagano prefectural Institute of Technology

Email: n18a417*a / nose*b@cse.pit-nagano.ac.jp

あらまし：長野県工科短期大学校 情報技術科では、ものづくり教育の一環として「ICTサマーキャンプ 小惑星探査機をプログラミングしよう」と題した市民講座[1]を開催した。この中で、アルゴリズム的思考要素と組み込みプログラミング的要素は、別のツールを使って段階的に学習させる必要があると明らかになった。そこで本研究では、UMLにおけるステートマシン図を応用したEV3用プログラミングツールを開発し、より初学者にも入門しやすい講座の開発を目指す。

キーワード：LEGO MINDSTORM EV3, 自律ロボットプログラミング, プログラミング教育, ステートマシン図, オブジェクト指向

1. 緒言

1.1. 研究背景

LEGO MINDSTORM EV3 (以下, EV3と記す) を用いて組み立てた走行体と, 教育版EV3ソフトウェア (以下, EV3ソフトウェアと記す) を使用した子供向けプログラミング講座を通して, EV3ソフトウェアの問題点を発見した[1]。この中で, EV3による走行体を使用したプログラミング教育を実施する場合には, 下記に示すように目的を区別する必要があるということが明らかになった。

(1) アルゴリズム的思考の段階

提示された目的を達成するために, どのように走行体の動きを組み合わせれば良いかを考えさせる。

(2) 組み込みシステムの開発段階

走行体に装備されているセンサやモータの制御について, ポーリングなどの組込システム開発の要素も含めて考えさせる。

(2)は, EV3ソフトウェアを使うことで学ぶことが可能である。しかし, (1)のみの学習に適した教材でEV3に対応したものは存在しないため, 本研究では(1)の学習に適した教材を開発する。さらに, 段階によって教材を使い分け, より初学者にも入門しやすい講座を目指す。

1.2. ステートマシン図の適用

ある目的を達成するために走行体に求められる動きを, 「走行体がどのような状態の時にどのような事象が発生したら, 次にどのような状態へ変化するのか」として考えさせることで, (1)の要素を学ぶことができると考えられる。このような考え方を表す代表的なものとして, UMLのステートマシン図 (状態遷移図) がある。ステートマシン図とは, 状態と遷移を組み合わせ, 対象物の状態がセンサ値の変化や時間経過などのイベントによって, どのように遷移していくかを表現する図である。

図1に, EV3による走行体の簡単な動作の一例につ

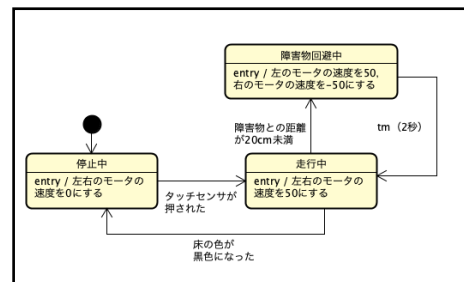


図1 走行体の動作を表すステートマシン図

いて, ステートマシン図で表したものを示す。実際の動作としては, 停止中にタッチセンサが押されると前進し, 前進中に障害物に近づくと右折し, また, 前進中に黒い床を検知すると停止する, というものである。走行体には停止している状態, 前進している状態, 右折している状態があるが, 図1では停止中, 前進中, 右折中と書かれた黄色の矩形で表されている。その状態の時に走行体が行う動作や処理について, 状態を表す矩形の下部に状態内アクションとして記述されている。また, 停止中状態から前進中状態に伸びている矢印が遷移を表す。

このステートマシン図を用いて, 目的を達成するための走行体の振る舞いを学習者に表現させることが可能である。この際, それぞれのイベントの原因となるセンサーの利用部分に関して, 「スイッチが押された」といった抽象的な表現を用いることにより, ポーリングなどの組込システム独特な処理を適切に隠蔽することが可能となる。そこで本研究では, (1)の学習に適した教材として, ステートマシン図を用いたEV3のプログラミングツールを開発する。

1.3. オブジェクト指向プログラミング教育への展開

ステートマシン図は, オブジェクト指向開発における振る舞いを記述するための図である。このため今回開発するツールは, 課題の難易度を適切に設定

することにより、高等教育機関におけるステートマシン図等UMLの実習においても利用が可能であると考えられる。そこで本研究では、ステートマシン図から生成される実際のソースコードを、ステートマシン図におけるオブジェクト（クラス）構成を理解しやすく反映したものとする手法に関して併せて提案する。

2. 提案手法

2.1. 開発するツールの概要

今回開発する教材には、ステートマシン図を記述する機能と、ステートマシン図からEV3上で実行可能なソースコードを作成するという2つの機能が必要となる。ステートマシン図は、システム設計におけるUMLで定義される一般的な図である。そこで今回は、ステートマシン図を記述する機能として、UMLモデリングツールであるastah*を利用する。

ステートマシン図からEV3上で実行可能なソースコードを作成する機能は、astah*のプラグインとして開発する。また、我々はEV3のプログラミング環境としてTOPPERS/EV3RTを使用しているため、生成する走行体用のソースコードはC++で記述する。

2.2. ステートマシン図からのソースコード作成

ステートマシン図からソースコードを作成する一般的な方法としては、状態を表す関数と状態遷移を管理する関数をそれぞれ作成するというものがある。状態の関数には、その状態が持つ状態内アクションを実行する処理とトリガーとなるイベントを監視する処理を持たせる。状態遷移を管理する関数は、現在の状態と発生したイベントの組み合わせからどの状態へ遷移すべきかを判断する処理を持たせる。この2種類の関数を使うことで、ステートマシン図で記述された振る舞いをソースコードへ変換することができる。

2.3. オブジェクト指向に基づいた実装

ただ2.2の方法で実装するソースコードは条件分岐が多くなり、可読性が低くなる。そこで、ステートマシン図と対応付けしやすいソースコードを作成する方法として、ステートマシン図における各要素をクラスとして定義し、オブジェクト指向に基づく実装を行う方法を提案する。これにより、各要素に対応する処理がソースコード上でオブジェクトとして定義されるため、よりステートマシン図と対応付けしやすいソースコードの作成が可能となる。

3. ソースコード変換処理の実装

生成されるEV3用ソースコードは、可読性の向上や構造化の観点からメインタスクとイベントタスクの2つに分け並列処理とした。

メインタスクでは、走行体の状態遷移やそれに伴うアクションの実行を行う。イベントタスクでは、トリガーとなるイベントの発生を遷移オブジェクトにそれぞれ伝達する処理を行う。イベントタスクにおけるイベントの発生を遷移クラスを通してメインタスクへ伝達し、そのイベントに該当する遷移オブジェクトの状態が、現在の走行体の状態であれば状態の遷移を実行する。この概念に基づいて作成したクラス概念モデルをメインタスクについては図2、イベントタスクについては図3に示す。

4. 結言

概念モデルに従ってステートマシン図からEV3用ソースコードを作成した結果、状態遷移の処理にク

ラス同士の関連を利用するため、条件分岐の多用を回避することができた。また、ステートマシン図における各要素をクラスとして定義することにより、ステートマシン図の内容とソースコードの対応付けが容易になった。以上のことから、ステートマシン図からソースコードを作成する場合は、オブジェクト指向を用いることにより可読性の向上が確認できた。さらに、このことから高等教育機関におけるステートマシン図等UMLの実習においても有用であることが確認できた。

今後は、ステートマシン図のアクション及びトリガーとソースコードを対応づけるDSLの整備を進める。さらに、対象年齢や学習対象に合わせたクラスの抽象度について検討を行う。また、現在はC++を使用しているためコンパイルが必要となるが、スクリプト言語であるPythonに変更することでコンパイルを不要とし、より使いやすいツールとしていく。

参考文献

- (1) 北條史華, 野瀬裕昭, 桃井貞美: “LEGO MINDSTROM EV3を用いた小学生向け自律ロボットプログラミング教育の実施報告”

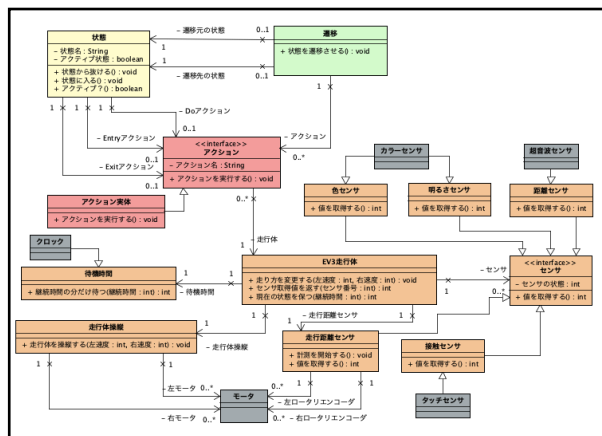


図2 メインタスクの概念モデル

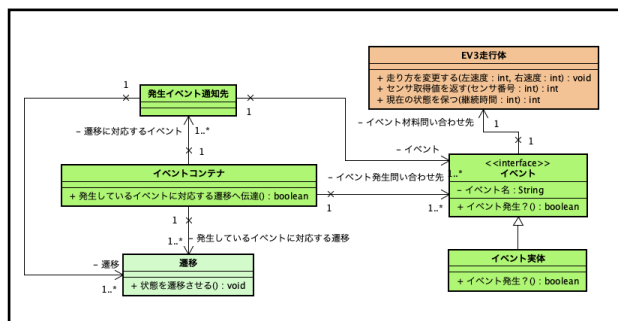


図3 イベントタスクの概念モデル