

組込みシステムのモデル駆動開発実験 A Model Driven Development Experiment on Embedded System

伊藤 邦彦[†] 松浦 佐江子[‡]
Kunihiko Ito Saeko Matsuura

1. はじめに

組込みソフトウェア開発では、システムへの要求をハードウェアとソフトウェアがどのように役割分担して実現するかを考慮して開発を行う必要がある。本稿では迷路探索ロボットを題材として、システムへの要求からトップダウンにそれぞれの役割を考慮したモデルを実行可能モデルを用いて開発するモデル駆動開発実験について報告する。

組込みソフトウェア開発では、ハードウェア・外部環境などによる物理的制約を考慮したモデルを構築し、シミュレーションによって動作を確認する。しかし、モデルの精度により、実機ではうまく動作しない場合がある。また、システムへの要求が十分把握されていない段階で精度の高いシミュレーションモデルを構築することは難しい。

本実験では、迷路を解くアルゴリズムを決定し、実機のセンサーによる外部環境の取得を入力、モーターによる動作を出力と考え、入出力を加えた迷路探索アルゴリズムを作成する。そして、仮定環境を想定し、シミュレーションを行い、ハードウェアに必要な能力をソフトウェアモデル上で議論しながら、システムへの要求をモデルに反映する方法について議論する。

2. モデル駆動開発

2.1 Executable UML

本稿ではExecutable UMLを用いてソフトウェアモデルを定義し、実行することでシミュレーションを行う。Executable UMLを記述するにあたりExecutable UMLツールとして、iUMLite2.20[2]を用いる。Executable UMLでは、データをクラス図、振る舞いをステートマシン図、振る舞いのアルゴリズムをアクション記述言語で定義する。Executable UMLにおいて、操作はステートマシンにおけるアクションから導き出される。これらを実行可能記述言語で定義されたテストケースを実行し、シミュレートする。

2.2 Executable UML を用いた開発例

Executable UMLを用いた開発例として迷路探索ロボットを用いる。迷路探索ロボットの制御は家電製品の組み込み用ソフトウェアと同程度の複雑さやハードウェアとの密接度を持っていると考えられる。また、シミュレートレベルで十分な仕様テストを行うには、ロボットの物理的な特性や、ロボットの外部環境のモデル化も必要となる[1]。開発

[†] 芝浦工業大学大学院 工学研究科 電気電子情報工学専攻
Department of Electronic Engineering and Computer Science,
Graduate School of Engineering, Shibaura Institute of Technology

[‡] 芝浦工業大学 システム工学部 電子情報システム学科
Shibaura Institute of Technology Department of
Electronic and Information Systems

例では迷路探索ロボットは車両型のロボットであるとする。また、前方にタッチセンサーを持っているものとする。迷路は壁と走路から構成される。開発例では迷路のスタートからゴールまで到達できることを目標とする。

2.3 初期の開発

探索方法として右手法を用いて探索を行う。右手法とは、右側の壁に手をつき壁沿いに進むという方法である。図1は右手法の手順をアクティビティ図で表した物である。

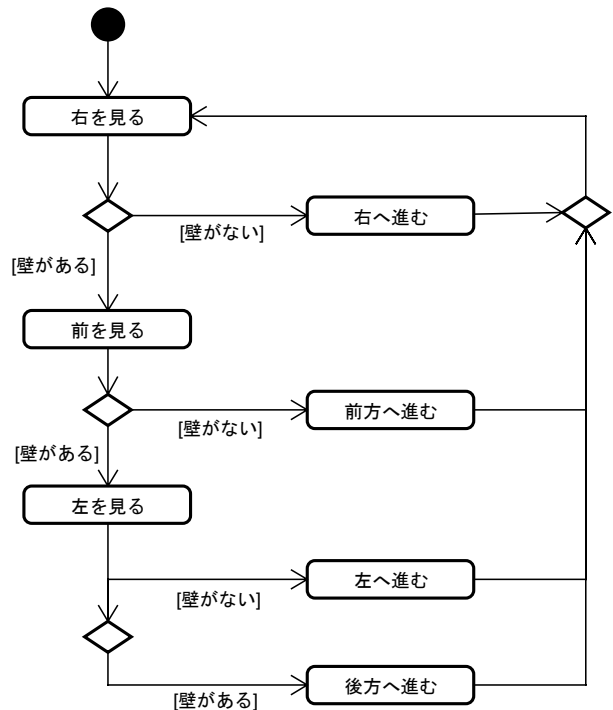


図1 右手法の手順

上記の手順をゴールまで繰り返すことにより迷路を探索する。また、初期段階でのロボットが行える動作を以下とする。

- ロボットは誤差なく一定距離を前進する
- ロボットは誤差なく90度の左右回転を行う
- タッチセンサーにより前方の壁の有無を判断する迷路は以下のように仮定する。
- 迷路の一マスは一定サイズである
- 走路幅は車体が旋回できる幅を持つ
- 迷路は必ず90度に曲がる

• 迷路上にはロボットの移動を妨げるものは何もない
以上の前提をもとにクラス図を定義する。入力役割を行うクラスとして「Sight」クラス、出力を行い移動するクラスとして「Body」クラス、探索を行うクラスとして「Search」クラス定義する。「Search」クラスは右手法の手順を行う。右手法の見ることを「Sight」クラスが行い、

進むことを「Body」クラスが行う。作成したクラス図が図2である。

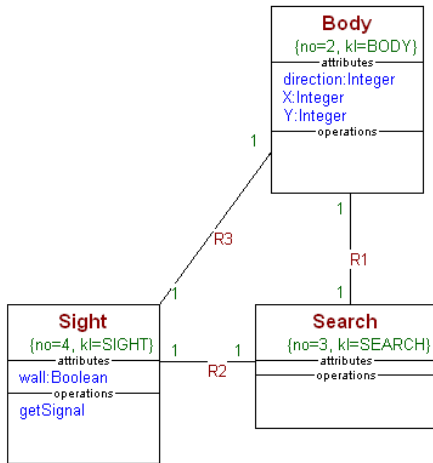


図2 基本動作のクラス図

「Body」クラスのステートマシン図は図3となり、動作可能なものを状態としてまとめている。状態は前進状態・右を向く状態・左を向く状態から構成される。また、ロボットの動作を加えて右手法の手順をステートマシンとすると、Searchクラスの状態は、スタート状態・右側壁チェック状態・右に壁がある状態・前に壁がある状態となる。ステートマシン図が図3のようになる。スタート状態はロボットの初期の位置を状態とした。右側壁チェック状態は車体を右に向け進むことにより壁を判定する。これにより右手法の1を行う。ただし、壁がある場合は壁により前に進まないため位置は変更されないものとしている。次に右に壁がある状態では左を向き進むことを行う。これにより、右手法の2を行う。さらに、前に壁がある状態では左を向き進むことを行う。左に壁がある場合は手順の2と3を繰り返すことにより後方に進むというを実現している。このように状態を定義することにより、見ることと進むことを制御することが可能となる。「Sight」クラスは現段階ではタッチセンサーのみを持っていると想定している。このクラスでは迷路のデータを基に走行体の前方に壁があるかどうかを判断している。データは図5を用いる。

このように定義されたモデルを用いてシミュレーションを行う。

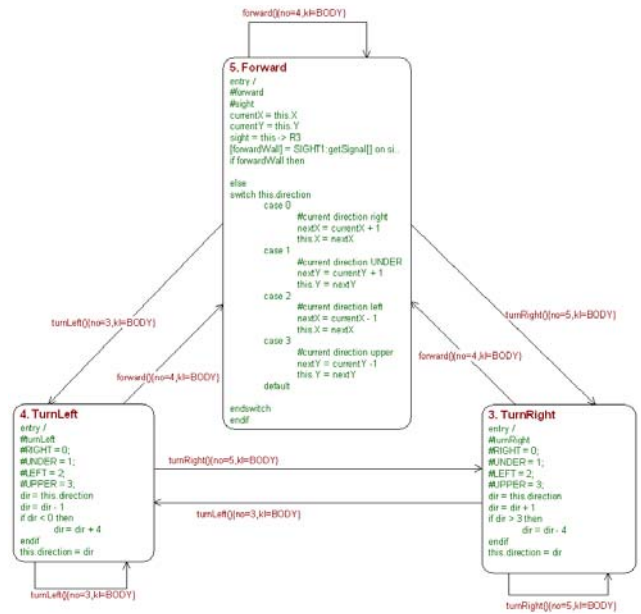


図3 Bodyクラスのステートマシン図

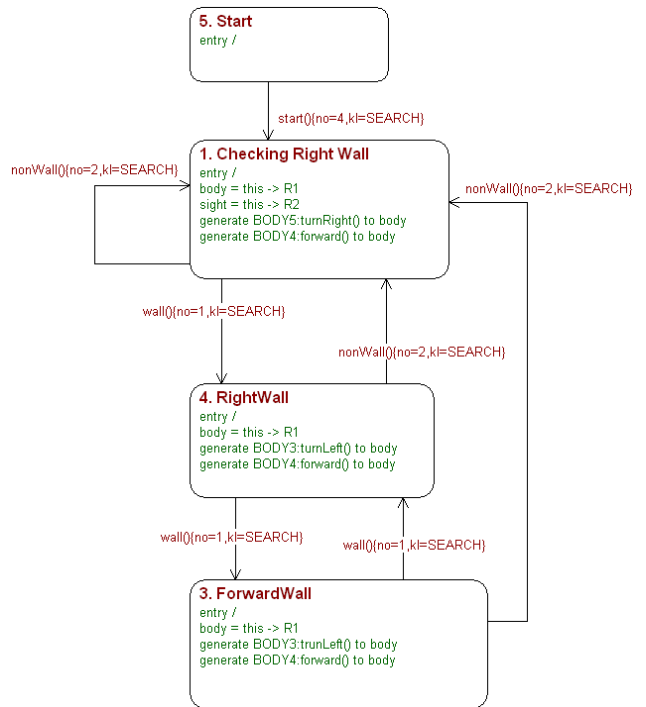


図4 Searchクラスのステートマシン図

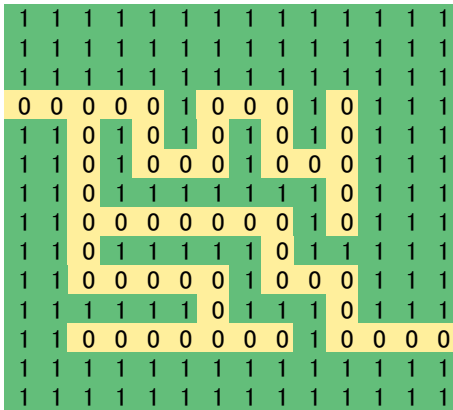


図5 1マスの迷路

3. シミュレーション結果

初期の開発で定義されたモデルに、初期設定を行い、実行する。初期設定ではロボットの絶対座標(x,y)、車体の方向、各ステートマシンの状態を設定する。実行時には「Search」クラスの start シグナルを送ることにより起動を行う。実行した結果、図5の右上から左下までを探索することに成功した。しかし、右手法のみではゴールを認識することができないため、特定の値が来た時にシミュレーションを終了とした。

3.1 問題点

シミュレーション結果より現在のモデルではゴールを判定できないという問題点が明らかとなった。ロボットを停止されるためには、ゴールを定義しなくてはならない。現在シミュレーションの終了を迷路のデータ配列の配列外に出た時に終了している。これは実際のゴールの定義にはならないため、ゴールを定義する方法としてハードウェアを用いる方法と、ソフトウェアを用いる方法があげられる。ハードウェアを用いるとしては、新たに光センサーを追加し、ゴール地点の走行路の色を変更する方法である。クラス図では「Sight」クラスに新たにどのようなセンサーを用いるかという「Sensor」クラスを付随させることが考えられる。一方ソフトウェアを用いる方法は、探索アルゴリズムに走行位置の履歴をとる仕組みを入れ、ゴール位置を指定する方法である。クラス図では「Search」クラスに新たに通過経路を記録するクラスを付随させることが考えられる。

3.2 前提条件による要素の考察

初期段階での開発では前提を用いてモデルを定義した。この前提を実際の外部環境、ハードウェアを考慮し変更を加える。例として走路の幅、ロボットの幅を変更することを考える。この変更により「Body」クラスに幅の属性が必要となる。また、走路の幅、ロボットの幅を考慮した動作を行わなくてはならないため、前進する距離を制御する必要がある。そのため、「Body」クラスに「駆動」クラス、「操舵」クラス等を追加することが考えられる。このように、前提条件をより実際の環境に近づけることにより、分析を進めていくことが可能となる。

4. まとめ

初期段階では仮定を用い外部環境、物理的な制約を簡略化し、ソフトウェアモデルを構築することが可能となる。これをシミュレートすることにより作成したモデルが正しく動作することを確認することができる。しかし、実際の動作では今回行った初期段階でのモデルだけでは足りないため、外部環境、走行体の要素を増やすことでどのような環境で用いることのできるソフトウェアであるか議論ができる。このようにモデルを議論することにより、問題をハードウェア、外部環境、またはソフトウェアのいずれを変更することが必要であるかを分析することができる。

参考文献

- [1]伊藤 恵, 久保秋 真, “組み込みソフトウェア開発のための仕様シミュレーション”, 情報処理学会研究報告, Vol.2002, No.64 (2002).
- [2]Kennedy Carter, “iUMLite” <http://www.kc.com/>