組込みシステムにおけるトップダウン設計方法の一考察

伊藤　邦彦†　　松浦　佐江子‡

芝浦工業大学大学院　工学研究科　電気電子情報工学専攻†

芝浦工業大学　システム工学部　電子情報システム学科‡

1. はじめに

組込みシステム開発では，既存のハードウェア・外部環境を基にボトムアップ開発を行うことが一般的である．しかし，要求から必要となるハードウェア・外部環境・ソフトウェアを明確にすることは困難である．そのため，MDA(Model Driven Architecture)のプラットフォームを切り分け，問題領域を明確にするという指針を基に，MDD(Model Driven Development)により要求に基づくトップダウン開発を行う．

本稿では，迷路探索ロボットを題材とし，Executable UML を用いてモデル上に外部環境の特性を段階的に取り入れることで，システムへの要求を満たすハードウェア・ソフトウェアを定義し，個々の役割を明確にする方法について述べる．上記題材を基に，組込みシステムにおける要求からのトップダウン設計方法を考察する．

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．これらをアクション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充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]．開発例では迷路探索ロボットは自律移動を行うロボットとする．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. 迷路探索ロボットの要求

ロボットは迷路を探索し，スタートからゴールまでを自律移動を行う．この要求を満たす探索方法として，スタートからゴールまでを壁伝いに移動する右手法が考えられる．右手法はアルゴリズムが簡単であり，自身の位置から右方に壁の有無を認識することで次の行動を決定できる．現在の状況を認識することでスタートからゴールまでを探索できるという利点から探索方法を右手法とする．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．

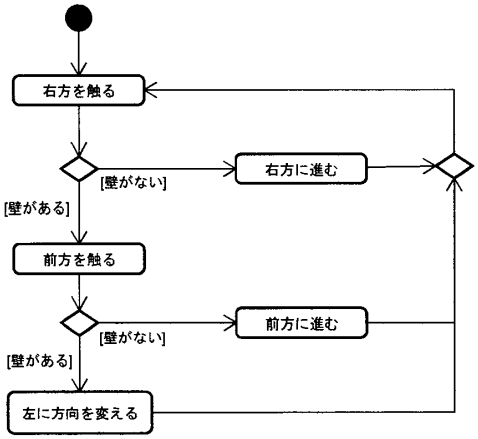

図 1　右手法の手順

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，この動作を行うにあたって最低限のハードウェア構成として前方にタッチセンサーがあり，左右回転・前進可能な車両型走行ロボットを想定する．ロボットの動作を以下に限定する．

- ロボットは誤差なく一マスを前進する
- 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

また，動作する外部環境としての迷路を以下のように仮定する．

- 迷路の走路幅は一マスサイズである
- 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

一マスは正方形の区画とする．以上の前提を基に図 1 の右手法の手順を入力・出力・探索に分割しロボットの入出力動作を分析する．分割したアクティビティ図が図 2 である．図 2 では入力を「壁の有無を調べる」としている．しかし，ここでは，前方にタッチセンサーがあり衝突しなくてはならないため，「壁の有無を調べる」を行うためには「前進する」動作を行わなくてはならない．また，「右方を触れる」を行うためには，「右に向く」動作を行わなくてはならない．そのため，入力を得るためには出力を伴わなくてはならないことが分かる．このことから図 2 をまとめたものが図 3 である．図 3 では壁がある場合には「前方を触れる」を行う．直前に右に方向を変更しているため，元の進行方向に変更する必要がある．そのため，出力は「左に向く」を行う．図 3 のレーンをクラスとして定義し，各アクティビティを状態と

A Study on Top Down Design Method for Embedded Systems

† Kunihiko Ito　‡ Saeko Matsuura

†Department of Electronic Engineering and Computer Science,

Graduate School of Engineering, Shibaura Institute of Technology

‡Shibaura Institute of Technology Department of Electronic and Information Systems

してとらえ各クラスに状態を定義した．クラス図では外部環境を認識する入力クラスとして「Sight」クラス，外部環境に影響を与え動作する出力クラスとして「Body」クラス，探索アルゴリズムを実行するクラスとして「Search」クラス定義する．

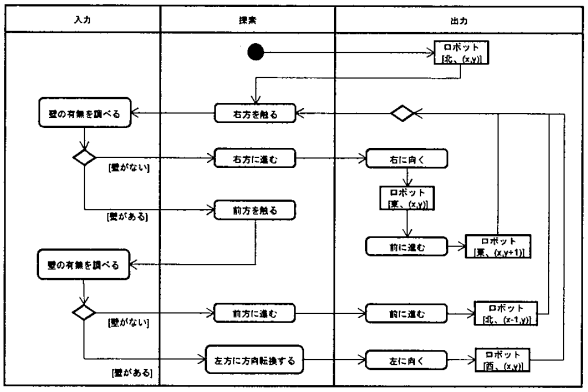

図２　右手法の入出力分析

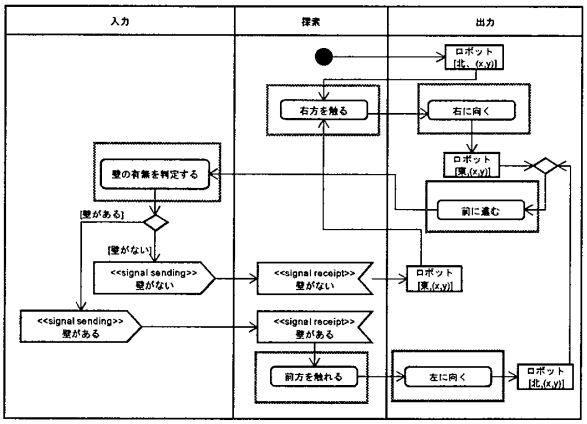

図３　入出力を分割した右手法の手順

2.4. ステートマシン図の作成

各クラスのステートマシン図を図 3 のアクティビティ図から記述する．状態をアクティビティとし，遷移をレーン内のアクティビティからアクティビティに入るフローを遷移とする．また，遷移条件を入力から得るものとし，アクティビティ図のシグナルを遷移条件として記述する．また，アクションはフロー先のアクティビティを実行する命令を記述する．「Search」クラスの状態は，「スタート」状態・「右方を触れる」状態・「前方を触れる」状態の 3 状態となる．「Search」クラスのステートマシン図が図４のようになる．「Search」クラスのステートマシンでは壁の有無の入力があった場合に遷移し，遷移後に状態のアクションを実行する．アクションはロボットの出力を記述している．「右方を触れる」状態では，「Body」クラスの「右に向く」，「前進する」の順に実行命令を送っている．また，「前方を触れる」状態では，「左に向く」，「前進する」の順に実行命令を送っている． 定義モデルを用いてシミュレーションを行う．

3. シミュレーション結果と問題

シミュレート実行した結果，スタートからゴールまでを探索することができた．次に走路幅の変更を行う．分岐路の発見のために，走路幅の変更に応じて前進幅についても考慮する必要がある．走路幅が車体幅よりも大きい時，隣のマスに壁があるとは限らないためロボットは即座に壁の有無を認識できない．そのため，壁の有無を認識するためにマスを移動しなくてはならなくなる．ここでは移動時に壁があるということを考慮しなくてはならない．

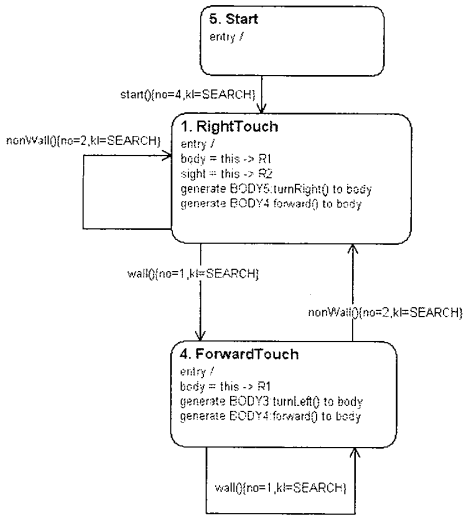

図４　Search クラスのステートマシン図

3.1. 問題点

外部環境の変更により移動時に壁にあたると実際には止まるがロボットの内部のデータではどこまで進んでいるかを判断することができない．これは入力のタイミングが問題である．入力は常に取得することができないため，データに不整合が出ない正しいタイミングで入力を得る必要がある．本実験では壁に衝突した瞬間に入力を得ることでデータの不整合が発生しないため衝突時が正しいタイミングとなる．しかし，入力を得るために前進するなどの出力を伴う場合，正しいタイミングで入力を得ることは困難である．

3.2. 解決案

上記の問題を解決する方法として，入力から入力の間に動作する距離を誤差としてとらえ，誤差を許容できるシステムを構築する方法である．誤差をシミュレーションで得るためにモデル上に外部環境を定義する．入力・出力ともに外部環境に関連付け，ロボットが把握できる位置とは別に，位置を保持させることで生じる誤差を計測する．

4. まとめ

システムの要求からアクティビティ図によりシステム全体の流れを記述した．アクティビティ図からステートマシン図を記述し，Executable UML を作成する．シミュレート実行することでシステム全体の流れが正しいことが確認できた．

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/