5M-5

協調設計による組込みソフトウェアの開発

長谷川　裕基†　松浦　佐江子‡

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

携帯電話をはじめとした情報機器の中にはマイクロコンピュータが搭載され, 中には機器の制御を行うことを役割とする組込みソフトウェアが入っている.

本学では学部３年生を対象にした「情報実験Ⅱ」という授業がある. 情報実験Ⅱでは 6～9 人のグループに分かれ, UML を用いたソフトウェアの開発を行う. この授業で著者は LEGO Mindstorms を使った「荷物の自動搬送システム」の開発を経験した.

１．１．組込みソフトウェア

センサ等から入力信号を受け, ソフトウェアによって処理を行い, 結果を基に駆動系等で制御することが役割である. 専用化されているので対象機器の種類の多さから種類毎に合わせた仕様にする必要があり, 多様性が求められる.

１．２．荷物の自動搬送システム

組込みソフトウェアの開発をテーマとして２台のロボットを集荷担当と配達担当に分け, 集荷担当が荷物の依頼を受け中継所まで図１のような楕円状の走行路を走行する. 走行路は黒い線で描かれている. 実際の運ぶ荷物はデータとして扱っており, 赤外線通信による荷物の引継ぎを行い配達担当に荷物を渡す. そして配達担当が配達先まで走行路を通って荷物を届けるシステムである.

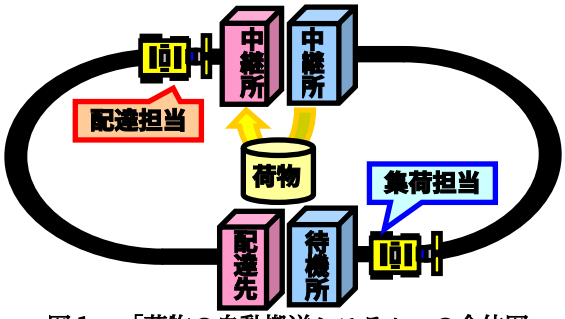

図１　「荷物の自動搬送システム」の全体図

１．３．LEGO Mindstorms

マイクロコンピュータを搭載した RCX と呼ばれるコントロールユニットを中心にブロックを使ってロボットを製作する. 光センサ, タッチセンサ, モーター, 赤外線通信等が使用できる. leJOS という小さな Java Virtual Machine にハードウェア制御用ライブラリが与えられており, そのライブラリを使うことで Java 言語によるオブジェクト指向に基づいたプログラムを作成する.

１．４．研究の目的

システムの開発終了後, 仕様変更要求があったがハードウェアの制約からソフトウェアのみで実現が困難となった時, 走行路等の外部環境を拡張することで要求を満たした. この経験からソフトウェア・ハードウェアに外部環境のモデルを加える必要があると考え, MDA(Model Driven Architecture)を参考に LEGO Mindstorms の仕様等の実装技術をプラットフォームとし, そこからの独立モデルと依存モデルに切り分けて考えることで, 従来よりも再利用性が高く, 仕様変更に強い組込みソフトウェアの設計モデルを考案した. 本研究ではソフトウェア・ハードウェア・外部環境の３点による協調設計の変更容易性を外部環境とハードウェアの観点からの変更を試みる.

Co -Design Method of Embedded Software Development
† Hiroki Hasegawa　‡ Saeko Matsuura
† ‡ Shibaura Institute of Technology
Department of Electronic Information Systems

２．MDA(Model Driven Architecture)の適用

モデリングを主体としたソフトウェア開発アプローチで, 本研究ではモデルの相互変換や PIM（Platform Independent Model）から PSM(Platform Specific Model)への自動生成[1]については扱わず, LEGO Mindstorms の仕様等の実装技術とは関係が無く, 走行する方法や概念等の機能やノウハウのような標準技術仕様について PIM として記述する. PIM を基に実装技術に依存したモデルを PSM として分けることで標準技術仕様について記述された PIM の再利用化を図る部分の考え方を参考にした.

３．情報実験Ⅱでの開発経験

この開発ではロボットが光センサを用いて黒い線の上を追跡しながら進み, タッチセンサの反応で目的地に到着したと判断し走行を中止するという方法でロボットの「走行」を実現した. しかし開発終了後, 走行路中に障害物が発生しそれを回避して目的地まで走行できる仕様に変更することになった.

変更の際, 課題として目的地と障害物の判別方法である. 両方共タッチセンサで感知できてもどちらなのかは判別できない. 光センサは走行路の追跡に使用している. ハードウェアの構成も固定されているのでこれ以上の判断基準を設けることができなかった. そこで目的地に黒い紙を敷き, タッチセンサが反応した時に首振り運動をさせて周囲の地面の状況を確認することとした. 障害物は走行路上に存在するため, 中央が黒でも周囲は白であると判定できるので, 周囲全体が黒であれば目的地であると判別できる.

このようにハードウェアやその中のソフトウェアだけで問題を解決するのではなく, それを取巻く外部の環境も利用することが組込みソフトウェアの開発における問題点の解決の切り口として有効であると考えた.

しかし LEGO Mindstorms に依存した設計を行ったため, ロボットの基本動作やセンサの値の意味を定義することなく, 値の大小でモーターの制御を変えて直接命令する形式になっているため変更箇所が集中した. また障害物という仕様外の要素が増えたことによって大幅な修正を必要とし, 修正後のプログラムも冗長でわかりにくかった.

このことから外部環境を仕様に組込むためにはあらかじめモデルに加えてソフトウェア・ハードウェア・外部環境の３点で協調した設計を行うことでモデル内部の役割も明確になり, 再利用性や仕様変更に強くなると考えた.

４．外部環境を加えた協調設計

ソフトウェア・ハードウェア・外部環境の３点を考慮した設計を行う. 上記で述べた MDA の参考部分を利用して「荷物の自動搬送システム」の再設計を行った.

４．１．プラットフォーム独立モデル（PIM）

今回の場合, プラットフォームをハードウェアモデル（LEGO Mindstorms）とし, ソフトウェア（ロボットの動作）と外部環境の定義の２つのモデル間で「荷物の自動

搬送システム」の設計を行うと図２のようになる.

ロボットの動作

・ ロボットの基本動作（「前進する」・「地面を見る」等）は BasicAction クラスで定義されている.
・ ロボットは荷物の取扱いを行う Delivery クラスと走行を行う Move クラスの２つがある. Robot クラスはその２つのモードの切り替えを行う.
・ Navigation クラスは外部環境で定義された Road クラスのインスタンスを全て持つ. Move クラスはこのクラスから現在の走行状態を確認して走行する.

外部環境の定義

・ Road クラスは走行路の定義が記述され, ロボットから受取った情報から走行路上か否かを判断する.
・ インターフェース Place を実装したクラスはロボットが現在地としてそれぞれのインスタンスを持つことでその場所に応じた動作ができるようにする. 上記で述べた Road クラスとの対応を Map クラスで行う.

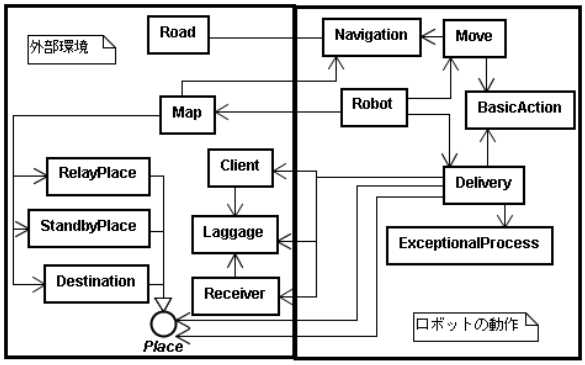

図２　PIM クラス図

４．２．走行方法

上記のロボットの動作と外部環境の定義から走行動作について具体的な説明を行う. 走行に関するクラスは Road, Move, BasicAction, Navigation の４つである.

Move クラスは BasicAction クラスを使ってロボットを前進させる等の動力の操作や視覚等の感覚を使って周囲の情報を取得させる. 取得した情報を Navigation クラスに与え, それを基にロボットが現在走行している Road インスタンスについての情報（走行路上を走行しているか否か）を確認する. 確認した Move クラスは走行路上ならばそのまま前進させ, 外れていれば進路修正を行うように BasicAction を使ってロボットの操作を行う.

４．３．プラットフォーム依存モデル（PSM）

図３の下部のような各種ハードウェア（センサ・モーター等）制御とその組合せについての記述, ロボットの動作モデルでハードウェアの操作をするためのクラスを定義して上記の PIM に加えて PSM を作成した.

PIM が「荷物の自動搬送システム」の方法や概念について記述しているので, システムは外部環境とロボットの動作で成り立っている. PSM でハードウェアモデルをロボットの動作モデルの中にとりこむことで抽象的だったロボットの動作方法や走行路の判別方法を具体的にしたモデルが作成できる. 黒い線による走行路の判別を光センサの明暗で行う方法から, 走行路の両端に壁を設置しタッチセンサで判別する方法に変更した場合, Road クラスの走行路上かどうかを判別するメソッドを変更すれば解決することから変更箇所を特定かつ変更しやすくなった.

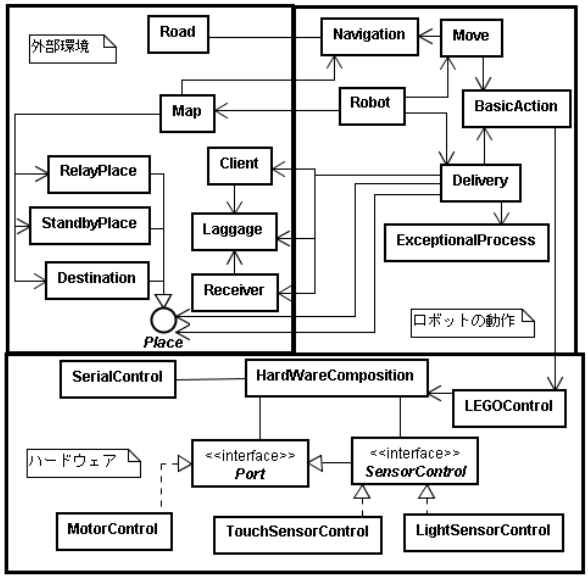

図３　PSM クラス図

５．研究室内の自動走行ゴミ箱システムの開発

「荷物の自動搬送システム」から走行路の形状やロボット同士の通信の内容が異なる「研究室内の自動走行ゴミ箱システム」の開発を今回のモデルを利用して行う.

５．１．システムの概要

自動走行するゴミ箱ロボットを親機とし, 座席の近い研究員同士で所持している子機を使用して規定の場所で待機している親機を呼び出しに行く. 親機は子機の種類（誰が所持する子機か）を判別し, その研究員の下へゴミ箱を運び, ゴミを捨てたことを確認した後再び規定の場所に戻る. ロボットは LEGO Mindstorms を使用する.

５．２．走行路の変化に対するモデルの変化

今回は走行方法に関するモデルの変化に注目し, 走行路が今まで楕円の形状から違う形状に変わったこと, 目的地によって走行ルートが変わるシステムになったので, その変更に対してモデルがどう変化するかを見る.

５．３．考察

ロボットの動作と外部環境の定義について「黒い線の上を走行する」という仕様の下で外部環境の定義を変更することなく実装ができた. よってロボットの動作と外部環境の定義についての協調は上手くいったと言える. しかしセンサの位置やロボットの大きさ等のハードウェア仕様の記述が不足し, センサの位置とロボットの中心とのズレによる正確な走行が出来なかった. PIM から PSM に変換する際にハードウェアとソフトウェアだけを繋げる形式ではなく, 記述の不足した部分である外部環境から見たハードウェアの特徴をモデル化する必要がある.

６．まとめ

本研究ではソフトウェア・ハードウェア・外部環境の協調を目指した設計を行った. ハードウェアモデルとロボットの動作モデルの関連だけでなく, ハードウェアモデルと外部環境の定義モデルも関連させることでハードウェアの大きさや構造を踏まえた設計が行えることがわかった.

７．参考文献

[1]吉田裕之:基礎からわかる MDA(モデル駆動型アーキテクチャ) －なぜモデリングするのか？, 日経 BP 社, 2006