

# LEGO MINDSTORMS™ の工学教育への導入のための機能拡張とその応用

Study for LEGO MINDSTORMS™ Application to Java Language Education.

今 成 一 雄

Kazuo IMANARI

## 1. はじめに

前回までの報告<sup>1,2)</sup>では、LEGO MINDSTORMS™ (以下、MINDSTORMS と称する。)のロボティクスおよび制御工学を主とした工学教育への導入の可能性を検討し、導入上必要な機能拡張の提案とその評価に関して報告した。今回は工学教育の視点を更に広げ、MINDSTORMS のコンピュータ言語教育への導入・応用を検討したので報告する。

本校の例を挙げるまでもなく、一般的なコンピュータ言語教育は、主に文法の理解に時間が割かれ、アルゴリズムやプログラミング技術に関する実用的な教育は、カリキュラムの終盤または別建の講義で実施されることが多い。それゆえ、文法を学習している間は命令の具体的な応用方法が理解しにくくなり、それが理解を妨げ、学生にコンピュータプログラミングは難しいとの印象を与えていると考える。またプログラミングの性質上、学習・演習はブラックボックスと化した机上のパソコンで行われるため、その理解は、学習者の習熟度とセンスとに依存する。ことに近年、その主流となっているオブジェクト指向言語 (Object Oriented Language : OOL) では、受講学生の反応・理解度から察するに、従来の手続き型言語 (Procedural Language) や関数型言語 (Functional Language) と比較して、この傾向が強いよう見受けられる。

これらの現状を踏まえて、OOLとして最も広く応用されているJava言語を取り上げ、これにMINDSTORMSを組み合わせた、OOP (Object Oriented Programming) を効率的・意欲的に学習できるシステムを提案する。

以下、第2章はMINDSTORMSの特徴と制約について、第3章は一般に入手可能な既存のJava言語学習教材について概観し、問題点を確

認する。第4章はMINDSTORMSで利用できるJVM (Java Virtual Machine) leJOSについて紹介し、その可能性に言及する。第5章は、学習を効率よく進めるための統合開発環境に要求される事項やその実現手段としてEclipseの導入について述べる。そして、第6章では、今回提案するMINDSTORMSを導入したJava言語学習教材を紹介、試用評価を報告する。第7章では、まとめと今後の展開の可能性について検討する。

## 2. LEGO MINDSTORMS™の特徴と制約<sup>3,4,5)</sup>

一連の報告<sup>1,2)</sup>で紹介しているが、ここで改めて、MINDSTORMSの特徴と制約について確認する。

### 1) Overview

1998年9月デンマークのLEGO社より発売されたMINDSTORMSは、対象年齢とシステムの規模により数種類発売されているが、ここでは対象年齢が12歳以上に設定されたROBOTICS INVENTION SYSTEM 2.0 (RIS 2.0)を取り上げる。図1は、RISのLEGOピース群を示す。RISは、RCX (Robotics



図1 RISのLEGOピース群

Command eXplorer )と呼ばれるマイクロコンピュータ・電源を内蔵したコントロールユニットを中心に、各種センサ・モータを含む700個余りの LEGO ピースで構成されている。全てのパーツは従来の LEGO Block との結合の互換性が確保され、従来のパーツも使用できる。

## 2) RCX

RCX は、図1 において左上にある、ひときわ大きな LEGO パーツである。RIS は RCX を中核とした作品を創作し、RCX にプログラムを転送することで自律動作を可能としている。表1 は、RCX の仕様を示す。パソコンや組み込みマイコンと比較し、RAM が非常に小規模である。複雑なシステムを構築する上で、これがボトルネックとなる。

表1 RIS の LEGO ピース群

マイクロプロセッサ	H8 / 3292 ( 8 bit ) ROM 16kByte RAM 32KByte
インターフェース	センサポート ×3 出力ポート ×3 赤外線通信ポート ×3 LCD ×3
電源	単 3 乾電池 ×6
外形寸法・重量	95×63×40mm, 115g ( 12×8×4 スタッド)



図2 IR タワー

(左：RS232C 接続、右：USB 接続)

## 3) 通信

RCX へのプログラムやデータの転送は、RCX 側は内蔵された赤外線ポートを、パソコン側は USB接続( RIS 1.0, 1.5 では RS232C 接続 )された IR タワーを使用する( 図2 )。赤外線ポートを使用した RCX 同士の双方向通信も可能である。IR タワーと RCX との双方向通信にはプロトコルの問題もあり、1Byteの通信に800ms 程度かかる。これが第2 のボトルネックである<sup>6,7)</sup>。

## 4) プログラム言語

RCX のプログラム言語は、RIS に付属の標準プログラム言語 RCX CODE やフリーウェアの NQC ( Not Quite C )<sup>8)</sup> が用意されている。

RCX CODE( 図3 )は初学者の使用を前提としたビジュアルな言語である。その開発環境や GUI ( Graphical User Interface )の使い勝手は、十分に工夫されている。しかし、プログラミングできる内容・機能にはかなりの制限が課されている。

NQC( 図4 )は、文法が C言語に非常に似通った言語である。本格的なプログラミングが行える一方で、開発環境は現在の目から見れば、最低限の機能しか持たない貧弱なものであるといえる。

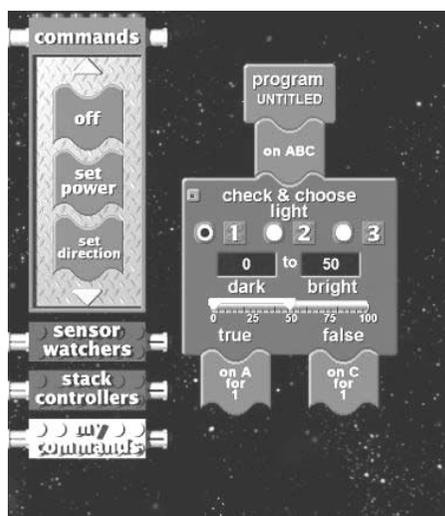


図3 標準プログラム RCX CODE の1例

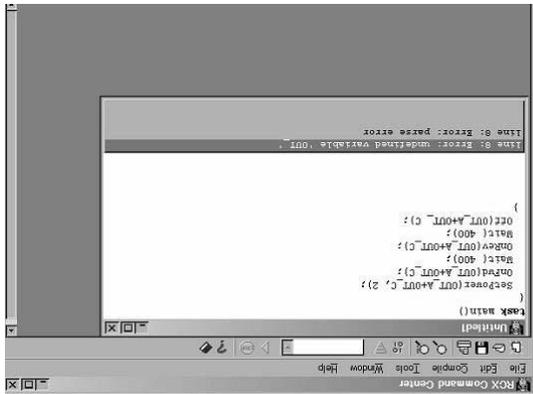


図4 NOCの1例



図5 RoboCode™の1例

いずれにしても、プログラミングに使用できる言語は手続き型言語や関数型言語であり、OOPの学習は出来ない。

### 3. Java言語学習教材

ここでは一般に入手可能な既存のJava言語学習教材を概観し、その問題点を明らかにする。

#### 1) RoboCode™(図5) 9)

IBM alphaWorksが、Javaプログラミング初學者向けに、プログラミングという分野を取り入れた学習環境である。API(Application Programming Interface)コードの呼び出し、Javadocの閲覧、継承、内部クラス、イベント処理などのOOPの基礎を学べるように設計されている。

内容は、戦車ロボット(オブリエクト)の目

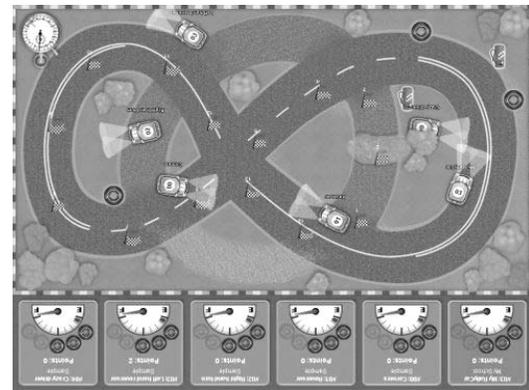


図6 CodeRallyレーストラック

2) CodeRally™(図6) 10)

IBM alphaWorksが、Javaプログラミングの学習スキルの上を自覚する方法として、プログラミングミニゲームを取り入れた学習環境である。目標を持ち一定の制限を意識した上で、継承、ポリモーフィズム、イベント処理などのオブリエクト指向プログラミングを学べるように設計されている。

内容は、仮想的な2次元ドリフト上で車(オブリエクト)の自律的な動きをプログラミングによって競うラリーカーゲームである。ラリーカーは、CodeRally™が提供するJavaAPIを使用してJavaクラスとして作成される。基本的には、RoboCode™に酷似しているが、走行コースの制約や外部環境の変化など考慮せねばならない項目が増え、より高いプログラミングスキルが要求される。

以上の2例のようにプログラミング言語学習教材という性格上、その形態はシミュレータである。これは手軽である反面、具体性・現実味に乏しく、学習者に対して想像力・洞察力などのスキルを要求する。ことにCodeRally™においては、ラリーカーの予期せぬ動きが、外部要因に拠るものな

か、単なるプログラムのバグなのか、の判断がつきにくい。さらに、それぞれの学習環境は、統合開発環境( Integrated Development Environment : IDE ) の形態をとってはいるもの、提供されている機能は、現在の目から見ると十分に満足できるものではない。

#### 4. leJOS<sup>11, 12)</sup>

前述のように、RCX に搭載されている RAM 容量は 32 Kbytes と大変小さく、そのままでは J2ME( Java2 Micro Edition ) の JVM でさえ実装は不可能である( J2ME : 128 Kbytes ~ 512 Kbytes )。記憶容量が容易に拡張できれば問題は解決するが、パッケージ化・パーツ化されている RCX には、望むべくもない。そこで RCX の容量に合わせて不要なパッケージやクラスを取捨選択し、一方で特有用な機能を提供した JVM を用意する必要が生じる。

leJOS は、Jose Solorzano らが開発した RCX に特化した JVM サブセットである<sup>11)</sup>。RCX の純正 Firmware を書き換えて implement する。図7は、leJOS のメモリマップである。RAM 32 Kbytes のうち約 4 Kbytes が ROM ルーチンで使用され、leJOS JVM が 16 Kbytes を占有する。よって、Byte Code 領域は残り 12 Kbytes とわずかである。しかし、55 行程度の class code から生成される Byte Code は約 3 Kbytes であることから、RIS 規模のシステムでは十分であると考えられる。表2は、leJOS 3.0 技術的仕様(原稿執筆時)を示す。自動ガーベージコレクション

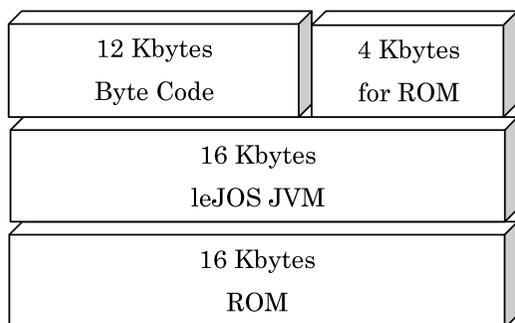


図7 leJOS Memory Map

を除き、必要であると考えられる最低限の機能は、その全てが網羅されている。

#### 5. 統合開発環境

現在、プログラミング学習・開発においては、それがストレスなく行える環境の整備を求められる。すなわち IDE の導入が一般的である。IDE に求められる機能としては

- 1) プロジェクト管理機能
- 2) 編集機能
- 3) 入力補完機能
- 4) エラーコンソール
- 5) アウトライン表示機能
- 6) Windows 標準 GUI
- 7) ユーザカスタマイズ機能

などが、あげられる。leJOS において、IDE は用意されていないが、シェアウェアの開発環境を使用することはできる。しかし、シェアウェアは、購入時の負担・メンテナンス契約などの費用がかかる上、GUI が独自のものである可能性があり導入には踏み切れない。一方で、先にあげた条件を全て満たしたオープンソース IDE が Eclipse である。

Eclipse は、1999年に OTI (Object Technology

表2 leJOSの技術的仕様

浮動小数点演算	利用可能 ( 32bit float 型 )
多次元配列	利用可能
再帰	利用可能 ( 10 level 程度 )
スレッド	利用可能 ( 理論値 : 255 )
イベント・オブジェクト	Timer , Button , Sensor
例外	Class と Signature によるコード表現
ガーベージコレクション	利用不可能
Java API	java.io ,            java.lang java.net ,            java.util javax.servlet.http josx.platform.rcx josx.rcxcomm josx.rcxcomm.remotecontrol josx.robotics ,    josx.util

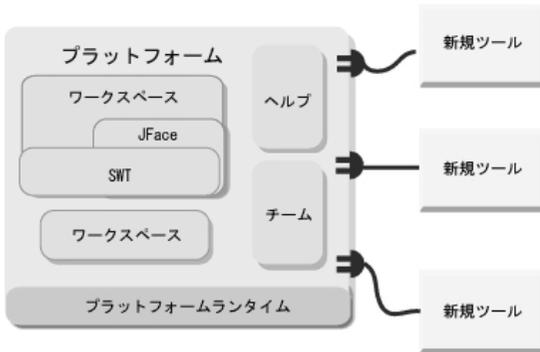
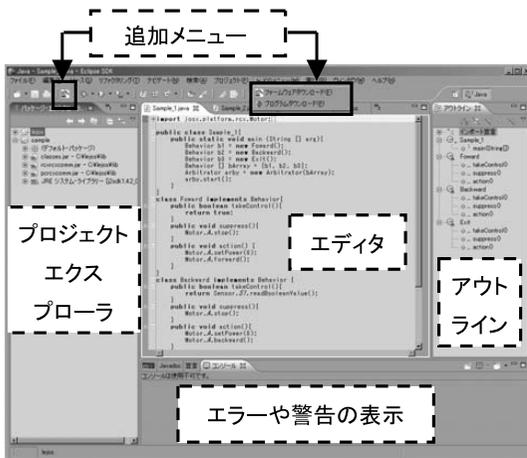
図8 Eclipse プラットフォームの構造<sup>15)</sup>

図9 Java 言語学習教材 Plug-in を導入した Eclipse の画面例

International) が開発していた IBM VisualAge for Java for MicroEdition が元となり、オープンソースとして公開されたものである。現在は Common Public License (CPL)、バージョンは 3.1 となっている<sup>13, 14)</sup>。

Windows, UNIX をはじめ、各種の OS に実装されており、先に記した 1)~5)の機能の充実はもとより、6) GUI についても申し分ない。特に 7) ユーザカスタマイズ機能に関しては、Plug-in方式を導入しており、学習・開発の環境・対象に応じて自由に機能の拡張・修正が行える(図8)。生来、Java 言語用のプラットフォームとして開発された Eclipse が、様々な環境で広く利用されている所以である。ここでも

Eclipse のこの特徴を最大限に利用する。

## 6. LEGO MINDSTORMSTM に leJOS と Eclipse とを導入した Java 言語学習教材

### 1) leJOS 教材と IDE

図9は、今回開発した Java 言語学習教材 Plug-in を導入した Eclipse の画面例である。本 Plug-in は、以下の機能を提供する。① RCX への Firmware (leJOS) のダウンロード、② RCX への Byte Code のダウンロード、③ RCX への通信速度の設定、④ クラスパスへの leJOS ライブラリの自動追加、⑤ leJOS API のヘルプ。①、②では、ダウンロードの進行状況がプログレスバーで表示される。

プログラムの開発手順は、以下のとおりである。

- 手順① 新規 Project の作成
- 手順② クラスファイルの作成
- 手順③ コンパイル
- 手順④ leJOS のダウンロード
- 手順⑤ Byte Code のダウンロード

つまり本教材では、パソコン上でクロス開発されたプログラムが、RCX の JVM 上で動作することになる。

### 2) 学習サンプル

図10は、学習教材例を示す。本教材は、センサ (プッシュスイッチ) の入力に応じて、テーブルの回転を制御する。センサ入力 that 全く無いと、テーブルは正転を続ける。センサ1 入力があるとテーブルは逆転する。センサ3 入力がある

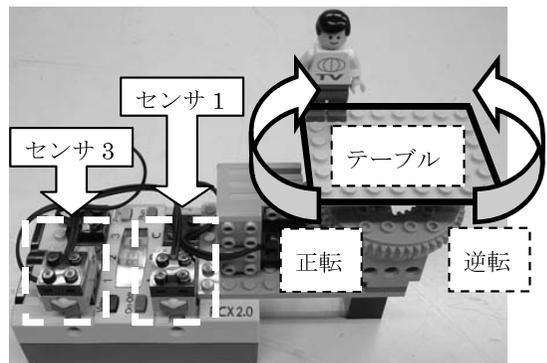


図10 学習教材例

## リスト1 プログラム ( クラスファイル ) 例

```

1:import josx.platform.rcx.Motor;
2:import josx.platform.rcx.Sensor;
3:import josx.robotics.Arbitrator;
4:import josx.robotics.Behavior;
5:public class Sample{
6:    public static void main (String [] arg){
7:        Behavior b1 = new Foward();
8:        Behavior b2 = new Backward();
9:        Behavior b3 = new Exit();
10:       Behavior [] bArray = {b1, b2, b3};
11:       Arbitrator arby = new
Arbitrator(bArray);
12:       arby.start();
13:    }
14:}
15://正転
16:class Foward implements Behavior{
17:    public boolean takeControl(){
18:        return true;
19:    }public void suppress(){
20:        Motor.A.stop();
21:    }public void action() {
22:        Motor.A.forward();
23:    }
24:}
25://逆転
26:class Backward implements Behavior {
27:    public boolean takeControl(){
28:        return
Sensor.S1.readBooleanValue();
29:    }public void suppress(){
30:        Motor.A.stop();
31:    }public void action(){
32:        Motor.A.backward();
33:    }
34:}
35://終了
36:class Exit implements Behavior{
37:    public boolean takeControl(){
38:        return
Sensor.S3.readBooleanValue();
39:    }public void suppress(){
40:    }public void action(){
41:        System.exit(0);
42:    }
43:}

```

ると、テーブルは停止し、プログラムが終了する。

リスト1 は、制御プログラム ( クラスファイル ) の例である。Java 言語の特徴である、API コードの呼び出し、継承、内部クラス、イベント処理などの OOP の基礎が随所にまんべんなく盛り込まれ、学習教材として効果的であるといえる。以前の教材 ( RCX が JVM を実行させているパソコンと通信をしてデータ処理をさせていた教材 ) では、スイッチを押してから反応が現れるまでに 1.5秒ほどのタイムラグがあった。しかし本教材は、センサ入力に対して即時に反応し、以前の教材で問題となっていた学習上のストレスは全く感じられない。

## 3) 試用評価

本教材を、本校電気情報工学科の 3,4年生が試用評価した。試験者は3つのグループに分けられる。

Aグループ：Java 初学者

Bグループ：Java 既学者

## リスト2 学習者の意見

## A グループ：Java 初学者

- ・ハードウェアの種類とサンプルソフトウェアを多数用意してほしい。
- ・Eclipse の画面表示が理解できなかった。
- ・RCX は外部電源にしたほうが良い。
- ・RCX を複数台同時に動かしてみたい。

## B グループ：Java 既学者

- ・遊び感覚で Java 言語を学習できる。
- ・プログラムの動作を目で確認できる。
- ・入力補助機能が、学習を容易にしている。
- ・error の指摘などの Eclipse による学習補助により、学習しやすい。

## C グループ：Java 既学者 (旧システム経験者)

- ・センサ入力から反応動作までのタイムラグがなくなり、ストレスを感じない。
- ・パソコンとの常時通信がなくなり、取扱いが容易になった。
- ・テンプレート機能のおかげで入力が容易になった。

Cグループ：Java 既学者(旧教材使用  
経験者)

学習内容は、MINDSTORMS でロボットを作成し、その制御プログラムを作成するものとし、ロボットの機能などは特に指定しなかった。

どのグループも、ロボット作成時間は約 1 時間、プログラム時間は約 30 分程度であった。初学者のグループは、多少プログラムに手間取っていたようであった。リスト 2 は、学習者の生の意見である。ここから、教材のユーザビリティの向上と、Eclipse 導入の有用性が読み取れる。一方で、Java 言語の十分な tutorials が用意されていないことから、初学者にとっては難易度が高くなったようである。

## 7. まとめ

本紀要は、LEGO MINDSTORMSTM をコンピュータ言語 (Java 言語) 教育に積極的に取り入れるための提案とその評価に関して報告した。

JVM には leJOS を、IDE には Eclipse を実装し、より直感的・具体的な学習教材にまとめることが出来た。試用評価にもあるように、授業で使用するには、まだ細かな問題が残ってはいるものの、導入の効果は非常に期待される。今後は、完成度を更に上げて、授業への導入を計りたい。

最後に、本報告をまとめるに当たり、実験・議論に参加・協力してくれた卒業研究生諸氏・電気情報工学科学生諸君に感謝する。

## 参考文献

- 1) 今成一雄, “LEGO MINDSTORMSTM の工学教育への導入の検討と提案” 小山工業高等専門学校研究紀要第 37 号 (2005) .
- 2) 今成一雄, “LEGO MINDSTORMSTM の工学教育への導入のための機能拡張の提案と評価” 小山工業高等専門学校研究紀要第 38 号 (2006) .
- 3) レゴ ジャパン, LEGO MINDSTORMS JAPAN Home page, <http://mindstorms.lego.com/japan/aboutus/lego/history.html>.
- 4) 古川 剛編, “LEGO MINDSTORMS パーフェクトガイド”, 翔泳社(1999).
- 5) Joe Nagata, “Joe Nagata のLEGO MINDSTORMSTM ロボット入門”, オーム社 (2000).
- 6) 石川 陽, “LEGO MINDSTORMSTM を導入した IDE 型 Java 言語学習教材の開発”, 卒業論文(2005).
- 7) 外池 英, “LEGO MINDSTORMSTM を導入した IDE 型 Java 言語学習教材の開発 II”, 卒業論文(2006).
- 8) NQC (Not Quite C) Home Page, <http://news.lugnet.com/robotics/rcx/nqc/> .
- 9) IBM Robocode ホーム - Japan, <http://www-06.ibm.com/jp/event/robocode/home/index.html>
- 10) dW Java technology CodeRally, [http://www-06.ibm.com/jp/developerworks/java/040813/j\\_j\\_coderally.html](http://www-06.ibm.com/jp/developerworks/java/040813/j_j_coderally.html) .
- 11) leJOS, <http://lejos.sourceforge.net/> .
- 12) Overview (leJOS API platform documentation), <http://lejos.sourceforge.net/apidocs/index.html> .
- 13) Eclipse.org home, <http://www.eclipse.org/> .
- 14) FrontPage - EclipseWiki, <http://eclipse-wiki.net/eclipse/> .
- 15) 的場聡弘・岡本隆史, @IT : Eclipse 徹底活用 (8), [http://www.atmarkit.co.jp/fjava/rensai2/eclipse2\\_08/eclipse08\\_1.html](http://www.atmarkit.co.jp/fjava/rensai2/eclipse2_08/eclipse08_1.html) .

