

## ロボットコンテストを利用した組み込み教育の実践

小倉 信彦<sup>†1</sup> 渡辺 晴美<sup>†2</sup>

近年、組み込みシステム開発の人材育成としてロボットコンテストが注目されつつある。我々は、ETソフトウェアデザインロボットコンテスト、MDDロボットチャレンジ等のコンテストに参加してきた。ロボットコンテストに向けた開発では、実験の繰返しによるドメイン理解とチューニングが不可欠であるため、試行錯誤を重ねながら開発する能力が育まれる。試行錯誤による開発では、開発対象の課題を発見し、部分課題へと分割、課題を解決する方法を考え、解決方法に従い実験・検証した結果からの考察により、次の課題を発見するというプロセスを繰り返す。これらを実践する能力は、技術者にとって必要であるが、一般の授業は、課題発見能力を養うことを主要な目標としていない。我々は、試行錯誤の能力を、ロボットコンテストを通じて効果的に養うために、課題発見、実験、要求分析から最終プロダクトまでのトレーサビリティを主要なテーマに、モデリングへの工夫、実験道具の開発等を行った。本論文では、これらについて紹介する。

### A Case Study of Embedded System Education Using Robot Contests

NOBUHIKO OGURA<sup>†1</sup> and HARUMI WATANABE<sup>†2</sup>

Recently, Robot Contests attract attention as an effective way of education for embedded software engineers. We, and our student, have participated in several robot contests, including ET Software Design Robot Contest, MDD Robot Challenge. In these contest, experiments and tunings play essential roles to comprehend application domain and to achieve competitive systems, hence students can learn the importance of developing process by trial and error cycle. The trial and error cycle is composed of the following processes: discovering subjects, decomposing sub-subjects, proposing method to solve each subject, discussing results of experiments and verifications with respect to the proposed method to discover next subjects. While the ability to handle the cycle is essential skill for engineer, many general lectures in undergraduate schools are insufficient to obtain the ability. To make effective training, the paper focuses on 3 themes: discovering subjects, experiments, and traceability from requirement analysis process to goal products.

### 1. はじめに

機器を制御する組み込みソフトウェアの開発では、センサやアクチュエータの特性を十分に把握する必要がある。この特性の把握は、単にセンサやアクチュエータの部品のハードウェア特性を調べるといふことにとどまらない。センサが計測する対象は、システムにより異なり、ソフトウェアからどのように見えるかについては、様々な視点で、試行錯誤を重ねながら理解、決定していく。したがって、試行錯誤を重ねて理解することは組み込みソフトウェア技術者にとって重要な能力の1つである。本論文において、試行錯誤とは、課題発見から、実験を行いその結果を考察することを繰り返し、最善の結果を導くプロセスとする。また、試行錯誤を重ねながら開発する能力とは、開発における課題を適切な部分課題へと分割する能力、課題の解決方法を考える能力、解決方法に従い実験した結果の考察に基づき、次の課題を発見する能力のこととする。なお、以下、試行錯誤を重ねながら開発する能力を「試行錯誤の能力」と呼ぶことにする。

近年、全国各地で様々なロボットコンテストが催されるようになり<sup>1)-3)</sup>、ロボットを利用したプロジェクト型の教育もさかんに行われるようになってきた。ロボットコンテストでは、実験の繰返しによるドメイン理解とチューニングが不可欠であるため、試行錯誤を重ねながら開発する能力が育まれる。この能力は、前記のとおり組み込みソフトウェア技術者にとって必要であるが、一般の授業では、養うことが難しい。ここで一般的な授業とは、大学学部生のプログラミングや実験を想定している。このような授業では、与えられた課題を1度解き、その結果について考察してレポートにまとめる。課題に特別に興味を持った学生が、徹底した実験・プログラミングを行った結果をレポートにすることはあるが、同じ課題を様々な方向から分析し、その結果に基づき、繰り返し実験を行うことは少ない。授業時間はコマ数で限られているため、試行錯誤という課題は本質的に扱いにくい。

試行錯誤の能力を効果的に養うために、我々は、課題発見、実験、要求分析から最終プロダクトまでのトレーサビリティに着目した。トレーサビリティが明らかであるということは、課題を解くために、どのような実験を行い、次の実験へ、そして完成したプログラムへ

<sup>†1</sup> 武蔵工業大学  
Musashi Institute of Technology

<sup>†2</sup> 東海大学  
Tokai University

とどのようにつながっていくのかが明らかであるということである。したがって、完成したプログラムの 1 行 1 行の根拠、なぜそのようにプログラムしたかということが明確であるということである。

試行錯誤を重ね、トレーサビリティのある結果を得るために、我々は、開発プロセス、ゴール指向分析、実験計画書・報告書、設計モデルを工夫し、実験計測器、簡易 MDD ツールを開発し、モデル検査を扱った。本教育により、受講生全員が、実験結果に基づくプログラムを行うことができるようになった。

以下、2 章では、ロボコンを利用した教育の目的、理念について明らかにする。3 章では、本論文の事例として紹介する ET ソフトウェアロボットコンテストについて概説する。4 章では、上記に記したロボコンを利用した教育を効果的に行うための個々の工夫について記す。5 章では、試行錯誤の能力育成に効果があったかどうかについてアンケートに基づき検討する。

## 2. ロボコンを利用した教育目的

ロボコンは、組み込みシステム開発の人材育成として期待されている。参加する学生達は学習の楽しさを知り、自ら学ぶ動機付けになればという程度に漠然と考えがちである。プログラミング能力の向上は、最も期待したいところであるが、ロボコンに参加した学生たちが、他の学生と比較し、際立ってプログラミングができるようになるとは限らず、実際、プログラミングの授業の成績において、ロボコン経験者と未経験者の間に成績に大差はなかった。ただし、ロボコンへ参加した学生の満足度は高く、アンケートでは、一様にプログラミング能力が向上したと答えている。

過去 3 年間の取り組みの中で、我々が特に効果があると実感したことは、「試行錯誤の能力」と「協調して開発する能力」の育成であることから、本研究では「試行錯誤の能力」に着目した。

ロボコンは競うことが本質であることから、チューニングが不可欠である。チューニングは直感的で単調な作業ではなく、緻密な原因分析が必要な作業である。スピードのチューニングは、速度を直感的に増やすという単純作業ではない。スピードが出ない原因をできるかぎり正確に把握し取り除く努力をしなければならない。ロボコンでは、どれほど原因をつかめたかによって、勝敗が決まることから、熱心に取り組まざるをえない。

1 章で述べたとおり、試行錯誤しながら開発する能力は組み込みソフトウェア技術者にとって重要である。産業界では、デミングの Plan Do Check Action が実践されている。また、

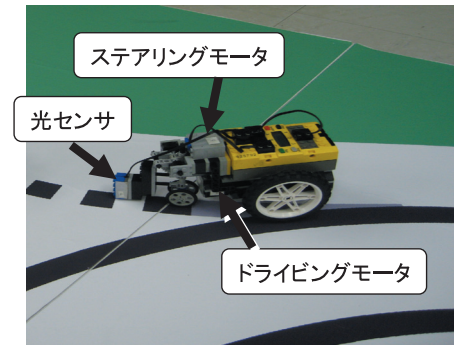


図 1 ET ロボコンの走行体  
Fig. 1 Target hardware in ET robot contest.

数学の問題を解くプロセスを記した古典である Polya の<sup>4)</sup>では、問題を理解する「理解」、分からない箇所を明確にしていく「計画」、問題を解いていく「実行」、得られた解の「検討」を繰り返すことを説いている。これらは試行錯誤の能力が、技術者にとってきわめて基本的な力であることを説いているといえる。

ソフトウェア設計の教育に LEGO を利用した事例集として、文献 6) がある。試行錯誤の能力で最も難しいのは、実験課題の発見である。課題を適切な課題に分割する能力が必要になる。このような能力の発見に、LEGO を用いた学習が優れていることが知られている<sup>7)</sup>。

## 3. ET ロボコン

ET ソフトウェアデザインロボットコンテスト (ET ロボコン) は、黒線で描かれたレーンをリアルタイムで検出しながら走行するライントラッキングレースであり、コース周回の走行時間を競うコンテストである。さらに、その開発時に作成したモデルの審査を行うことが特徴である<sup>4)</sup>。ソフトウェアとその作成過程を競うことを特徴としているため、走行体は、主催者側が規定した形状に構築し、LEGO MINDSTORMS RCX を使用する。走行体を図 1 に示す。走行体の前方の光センサにより、明るさを判定することにより、コース部分を識別し、ステアリングモータを用いて前方の操舵車輪の角度を制御し、ドライビングモータにより後輪の駆動力を制御することで、黒色線上を高速かつ安定に走ることが要求される。ET ロボコンでは、図 2 に示す難所が設定されており、開発に工夫を施さなければ、周回を完走することも難しい。

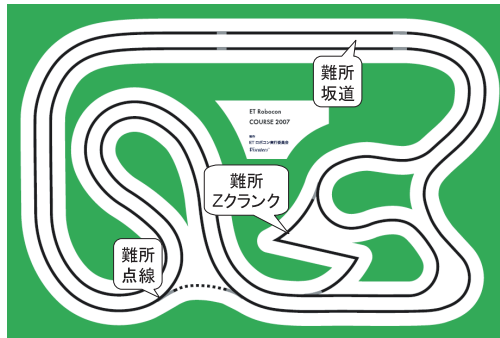


図 2 ET ロボコンのコース  
Fig. 2 Course in ET robot contest.

#### 4. 開発方法と教育

本章では、試行錯誤の能力を効果的に身につけるために行った工夫について紹介する。1章で述べたとおり、試行錯誤の能力育成のために、トレーサビリティに着目した。本論文でのトレーサビリティは、プログラムの1行1行について、なぜそのようにプログラムしたのかという根拠が明確であることを目指している。したがって、ドキュメント間の構成要素が対応しているだけでは不十分である。解くべき課題設定を行っても、直感的な実験では、論理的に、プログラムを構築する材料にはならない。また、次の課題を発見することも容易でなくなる。本研究では、(1) 試行錯誤のプロセス全体を管理する方法、(2) 開発するシステムを物理的に実験し、客観的に観察できる仕組み、(3) その結果を分析し、整理、記録する仕組みを教材とする。さらに、(4) 物理的に観察する方法とは違う視点を持つことで、課題発見能力に幅を持たせるようにする。(1) では、開発プロセスを工夫し、(2) と関連し、測定器具の開発を行った。(3) と関連し、ゴール指向分析を利用し、課題が展開していく様子を理解しやすくした。また、実験計画書、報告書を工夫した。そして、この試行錯誤が行いやすいように設計モデルを構築した。(4) に関してはモデル検査により、異なる視点を持つよう工夫した。

##### 4.1 教育対象・カリキュラム

本カリキュラムの対象は、主に大学学部3年生と4年生であり、本論文で対象とした受講者数は13人(5人+8人の2グループ)である。なお、過去3年間の類似した教育から構

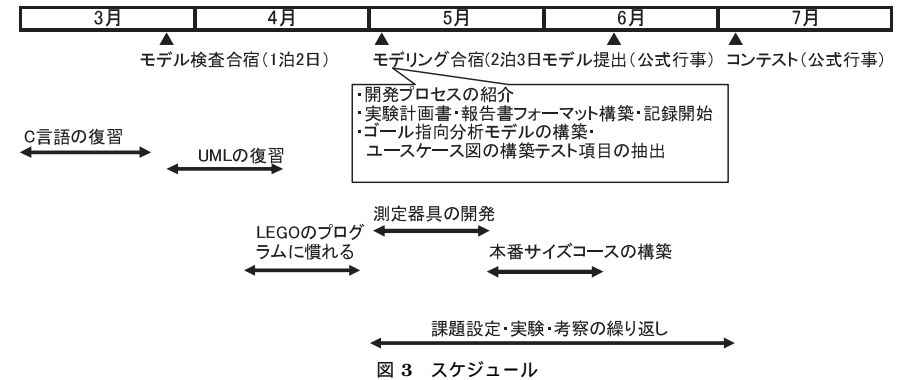


図 3 スケジュール  
Fig. 3 Schedule of the curriculum.

築したカリキュラムであり、過去3年間の延べ人数は38人であり、10人は、今回を含め、2度の類似した教育を受けている。本論文で対象としている13人の背景は次のとおりである。C言語で関数を利用したプログラムが作成できるレベルの学生が受講者の60%、C言語で制御構造を利用した数行のプログラムが作成できるレベルの学生が10%、初学者としての教育が必要なレベルの学生が30%である。UMLについては、文法の知識があり、読むことが可能なレベルの学生が70%である。図3に教育スケジュールを示す。

##### 4.2 開発プロセス

図4に我々が実践した開発プロセスを示し、表1に各プロセスの定義、テーマ、方法(目的とプロダクト)について示す。過去のコンテスト経験をもとに、学生が理解しやすいと思える単純な構造で、実験が中心となるようなプロセスを教員が構築した。なお、各プロセスでモデルを記述するのは学生である。

要求分析プロセスは、課題を明らかにするプロセスと定義し、競技規約や、過去のコンテストの分析から、ゴール図を作成し、サブゴールから実験項目を抽出する。また、ドメインを十分に理解するために、分析段階での実験を十分に行う。設計プロセスは、要求分析で明確化した課題の解決方法を明らかにするプロセスとした。分析段階で行った実験経験をもとに設計方針を立て、モデルにした。サブモジュールのアルゴリズムはアクティビティ図で示した。そのアクティビティ図を状態チャートで表し、ファームウェアとのインタラクションをおおまかに示した状態チャートとシーケンス図を作成した。これらの状態チャート、シーケンス図をもとにモデル検査も部分的に行った。実現プロセスは、設計で得

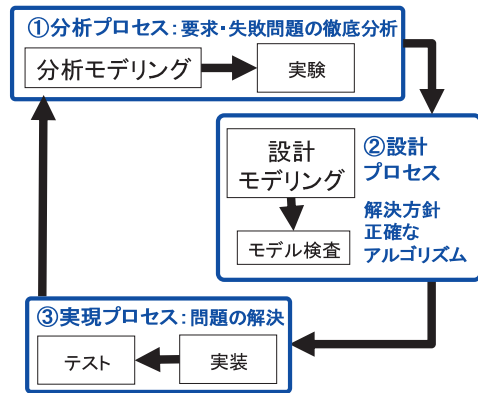


図 4 開発プロセス  
Fig. 4 Developing process.

た解決方法に従い課題を解決するプロセスと定義した。コンテスト本番用走行プログラムの作成とチューニングを行うプロセスである。なお、本プロセスでは、下記の 3 種類の実験・テストを行う。(1)と(2)は、要求分析プロセスで行い、(3)は設計プロセスで行う。

- (1) 事前準備のための実験：パスファインダーやコース、実験用システムの基本的な性能や使用方法を調べる。
- (2) 設計方針を決めるための実験：技術や部品を入手、開発、評価し、設計方針を決める。
- (3) チューニングのためのテスト：パスファインダの性能を極限まで引き出すことを目標に実験を行う。

学生にも理解しやすく、3章で示した実験に基づいた思考を養うことを目的とする開発プロセスとした。

#### 4.3 ゴール指向分析による課題の発見

要求分析においてゴール指向分析を取り入れ、分析の結果をゴール図として記述した。ユースケース図による機能抽出では、コンテスト主催者側が示す課題のみしか抽出できない。ゴール指向分析では、非機能要求、非正常系について、学生たちでも開発初期段階にある程度抽出可能である。たとえば、ユースケース図による要求分析では、図 2 の難所である点線、Z クランク、坂道等のみしかユースケースとして発見することができない。ゴール指向分析により、「通常走行を安定して走行する」というサブゴールが抽出でき、そこから、通常走行を安定させるためにはどうすればよいのかということを考えるきっかけにな

表 1 開発プロセスの定義

Table 1 Definition of the developing process.

①要求分析プロセス		
定義	課題を明らかにするプロセス	
テーマ	競技規約・問題点の徹底分析	
方法	目的	プロダクト
1	機能分析 共通理解	ユースケース図 コラボレーション 図
2	非機能要件&抜け漏れ	ゴール図
3	環境・システム性能 の分析	実験計画書 実験報告書
②設計プロセス		
定義	課題の解決方法を明らかにするプロセス	
テーマ	分析実験結果に裏付けられた設計 変更しやすい&正確なモデル	
方法	目的	プロダクト
1	変更容易性の向上	クラス図での変更 箇所を明確化
2	分析実験結果に裏付 けられたアルゴリズム 設計	アクティビティ図
3	正確なモデル	挙動をステートチ ャート、シーケ ンス図で記し、モ デル検査
③実現プロセス		
定義	解決方法に従い課題を解決するプロセス	
テーマ	上流プロセスとのトレーサビリティ	
方法	目的	プロダクト
1	クラス図と C 言語の関 係	クラスを関数群 に対応させる手 順の明確化
2	トレーサビリティ	ステートチャー トからプログラ ムへ
3	非機能・機能・正常系・ 非正常系のテスト・チュ ーニング	テストケース

る。そこから、学生たちは、さらなるサブゴールとしてステアリングモータを安定させることに気付く。このようにして、開発初期段階において、25 個の実験項目を発見した。なお、この 25 個は、実験の主要なテーマの個数であり、1 つの実験の目的を達成するために、おおよそ 5 段階程度の実験の種類が存在し、各々に複数の測定項目が含まれる。すなわち、ゴール指向分析により、25 個を起点とした多岐にわたる課題を発見することができ、ゴール指向分析モデルにより、発見した課題を整理することができた。

#### 4.4 実験計画書・報告書

効果的に実験を行うために、実験計画書と報告書を作成した。実験計画書・報告書は、これらのフォーマットを決めるにあたり、学生たちに指示したことは、授業の実験で作成するレポートをイメージして作成するという点である。実験計画書は、下記のとおりであり、授

業の実験で利用する形式に近い形式になっている。

- 目的：何をテスト・計測するのかについて記す。
- 背景：実験が走行にどのように役立つのか、モデルのどの部分と関連するかについて記す。
- 期待値：期待する実験結果について記す。
- 実験方法：具体的な実験条件や、実験装置を明記した方法を記す。

実験報告書の例を図5に示す。実験報告書は、授業の実験報告書と比べ、実験条件に特徴がある。この部分は、実験終了後のレビューにおいて、ロボコン経験者がつねに指摘する項目を集めたものである。実験計画書によって、実験の位置づけが明確化され、実験報告書の比較検討部分により、次の実験の課題発見が容易になると同時に、どのような過程で実験を行ってきたかをトレースすることが可能である。図5の実験は、走行体がコースを脱線したことを認識する処理に役立てるための基礎実験の1つである。脱線からの復帰は、完走するために、必ず入れる必要がある処理であるが、この復帰処理により、他チームの走行体との衝突、迷走することも多々ある。この実験では、走行開始時の加速時に、光センサが獲得する値が不安定になる範囲を測定した。この実験から、不安定になる範囲として1,000カウントという数値を得た。カウントは、光センサから値を獲得し、モータへ指令を送るループのカウント数である。この実験結果である1,000カウントの前後で、脱線の認識処理および脱線からの復帰処理を変えなければならないことを示している。

脱線からの復帰処理を実現するためには、ほかにも隣のコースを走行する走行体と接触しない位置や復帰可能な状態を保つためにどのような条件が必要か実験によって確認する必要があるが、前記実験のように実験結果より考慮すべき新しい課題が発見されることが少なくない。このように試行錯誤を重ねながら開発するプロセスを実験計画書・報告書によって実践した。

#### 4.5 測定器具の開発

組込みシステムにおいて所期の性能を達成するためには、ソフトウェア、マイコンの領域を超えたドメインを認知する能力が必要となる。そのためには、システムを動かす対象領域で何が起きているのか、調べる方法を獲得する必要がある。

場合によってできあいの測定装置で十分であることがあるし、適応する測定装置がない場合、測定器具を手作りしなければならない。ETロボコンの課題は0.1秒でも早く規定のコースを走りきることであり、速度を適切にコントロールすることで、安定な走行が実現されるため走行体の速度は、最も基本的な測定対象となる。

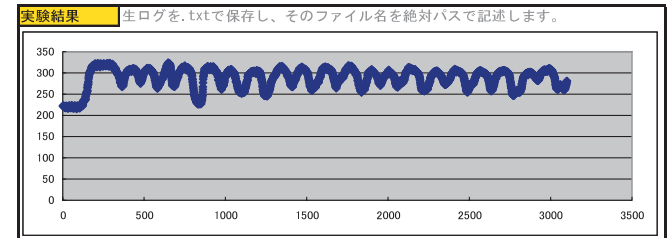
実験番号: 0 1 3 日付: 西暦 2 0 0 7 年 6 月 2 4 日 開始時刻 1 3 : 4 5

輝度: 6 3 4 温度: 2 9 [°C] 湿度: 6 8 [%]

実験責任者 樋口 大介 実験場所 4-208A

共同実験者 高瀬 雅織

<b>実験条件</b> 次の項目について、値を調べ、記入します。			
<b>【実験前】</b>			
光センサ番号	2	遮光スクートの番号	なし
光センサの高さ	5 [mm]	走行体の番号	1
ステアモータ番号	13	R C X の番号	623529
駆動モータ番号	14	タッチセンサ番号	なし
電池の種類	Panasonic/アルカリ /1.5 [V]		
テスターで測ったバスファインダーの電池の電圧			
1 本目	1.49 [mV]	2 本目	1.49 [mV]
3 本目	1.49 [mV]	4 本目	1.49 [mV]
5 本目	1.49 [mV]	6 本目	1.49 [mV]
使用したプログラム名			
log_interval.c			
<b>【実験後】</b>			
テスターで測ったバスファインダーの電池の電圧			
1 本目	1.49 [mV]	2 本目	1.49 [mV]
3 本目	1.49 [mV]	4 本目	1.49 [mV]
5 本目	1.49 [mV]	6 本目	1.49 [mV]



© Watanabe Lab. Sch. High-Tech. for Human Welfare Tokai University. All Rights Reserved.

<b>比較検討</b>
<p>等速度になるまでの加速時間はかなり短い事が分かった。                  等速度になってからは、規則的にカウントしているが、カーブ付近になるとカウントが不規則になっている。                  今回の実験の結果により、走行開始から1000カウントまではカウントが不規則なので、白カウントによる復帰動作は1000カウントが終わってから行ったほうが望ましい。</p>

図5 実験報告書の例

Fig. 5 An example of experiment result documents.

しかしながら、既製の測定装置で時々刻々と変わる走行体の速度を計測するのは容易ではない。速度を計測するための測定器具の製作によって、どのくらいの精度の計測が必要なのか、組込みシステム特有の外部環境からくる不確定で再現性のない現象を引き起こす原因が何であるのか、追求する姿勢と方法を学び、同時に今回の課題のドメインへの理解を深める。

図6に速度変化測定器具の構成、図7に速度測定の様子を示す。速度の計測を行うには様々な方法があるが、コース上の位置との対応付けが確実であり、周回時間の計測やETロボコン以外の組込み教育等他目的への使い回しが容易であることが重要と考え、透過型フォトインタラプタを用いた時間計測装置を制作させた。この方法では10cm間隔で並べたときに1%の精度の計測を行うためには1mmのセンサの設置精度が必要であり、うまくいかないとどのような現象が観測され、精度良く設置を行うにはどのような困難があり、いかなる工夫によって必要な精度が実現できるか、といった計測方法の工夫を通じて、試行錯誤に

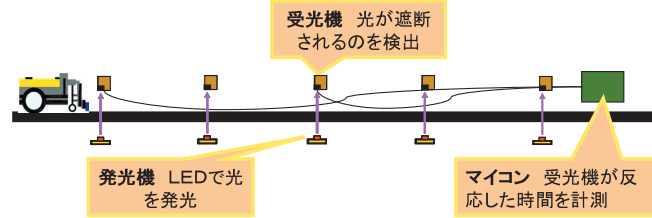


図6 速度変化測定器具の構成

Fig. 6 Overview of a measurement equipment.

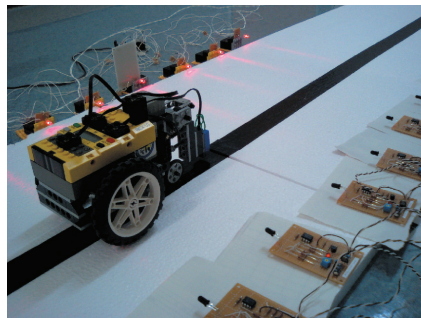


図7 速度測定の様子

Fig. 7 Measuring velocity in a test run.

よる課題解決能力を養うことを狙いとしている。

#### 4.6 設計モデルの工夫

図8に要求分析プロセスで行った実験で得た経験をもとに記述した設計クラス図を示す。変更の頻度に応じて修正ができるようなモデルを構築させることを目的とし、変更を頻繁に行う箇所の明確化を行うための変更レベル分け、作業分担の容易化を狙ったパッケージ化を指示した。学生の定義した変更レベルは下記のとおりで、それぞれをステレオタイプとすることで変更を頻繁に行う箇所を明確化した。

- 変更レベル1：コンテスト当日修正が必要な箇所：コンテスト当日には、当日のコース状況に応じて調整が必要である。その部分をこの箇所に記すようにした。
- 変更レベル2：頻繁にチューニング変更する箇所：安定して走行できるよう、当日まで

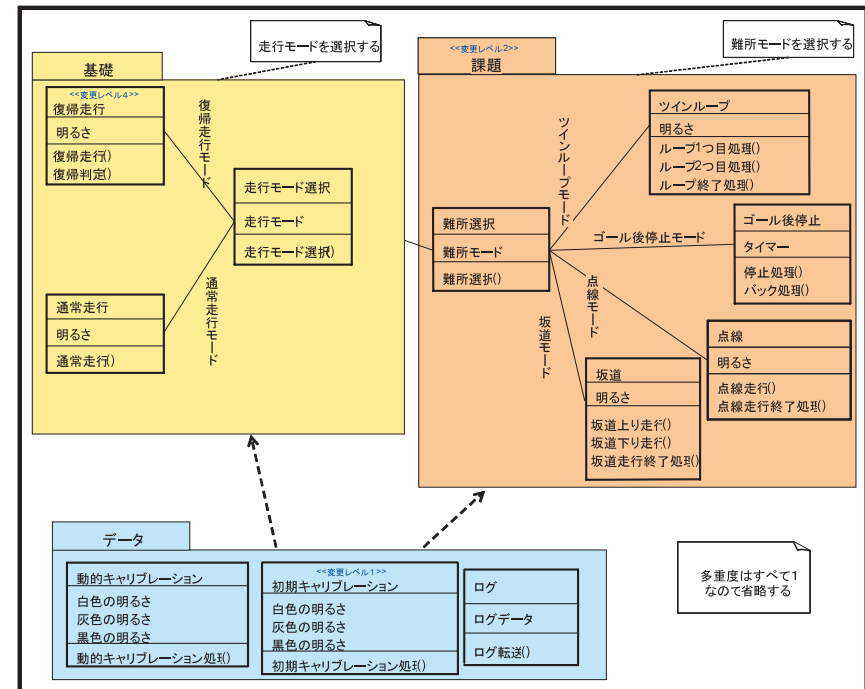


図8 設計クラス図

Fig. 8 A class diagram in design models.

チューニングを行う部分である。

- 変更レベル 3：来年以降、コンテスト課題変更等に応じて変更する箇所。

さらに、パッケージを「アプリケーション」と「ドライバ」とし、「アプリケーション」を「基礎」、「課題」、「マネージャ」とし、来年度以降変更になる可能性の高いところを「課題」パッケージにまとめた。「基礎」、「マネージャ」とパッケージすることで、開発順序の優先度を明確にした。「ドライバ」パッケージは、ファームウェアとのインターフェース部分であり、将来的にファームウェアの開発等を行った場合に備えてある。また、パッケージ、クラスの切り分けは、チューニング途中で、新たな走行アルゴリズムを思いついた場合に備え、容易に取り替えられるように工夫した。

#### 4.7 簡易 MDD ツールの開発

モデル間あるいはモデル・プログラム間のトレーサビリティに関しては、ロボコンに取り組んでから、つねに教育するように務めてきた。MDD ロボットチャレンジ 2005 では、状態チャートとプログラムを関連付けるためのフォーマットを用意し、状態チャートとプログラムのどの部分が関連付いているかを明確に示す仕組みを用意した。しかし、この取組みはあまりうまくいかなかった。そこで、2006 年度の MDD ロボットチャレンジでは、図 9 に示す簡易 MDD ツールを VBA で作成した。2007 年度に実施した本教育では、生成ターゲットの追加やモデル検査への応用のための機能追加を学生が行った。このツールでは、画面左部分にあるボタンを押すと図を作成できる仕組みになっている。作成開始ボタンを押すと画面右側に示されているフォームが現れ、モデルの詳細について入力できる。そして、このフォームの入力が完了すると、この状態チャートに対応した状態遷移表が作成される。さらに、状態遷移表から C 言語のプログラムが生成される。作成にあたり、教員が、状態遷移表からプログラムを生成するプロトタイプを作成した。学生たちは、このプロトタイプをリファクタリングし、図 9 のユーザインタフェース部分、状態チャートから状態遷移表への変換部分を作成した。現在は、3 種類のマイコンの C 言語プログラム、SPIN<sup>8)</sup> によるモデル検査のための PROMELA を生成することができる。作成に携わった学生たちは、モデリングの授業課題も自分たちのツールを使用している。

#### 4.8 モデル検査

モデル検査には、SPIN を用いた。モデル検査は、今後、発展が見込まれる技術であるが、卒業後に触れる機会は少ないと思われる。しかしながら、優秀な学生の学習動機付けに効果的であるため、モデル検査も含めたトレーサビリティをあえて開発に加えた。検査対象モデルは、システムの難所走行アルゴリズムを表す状態チャートである。図 10 は、難所の

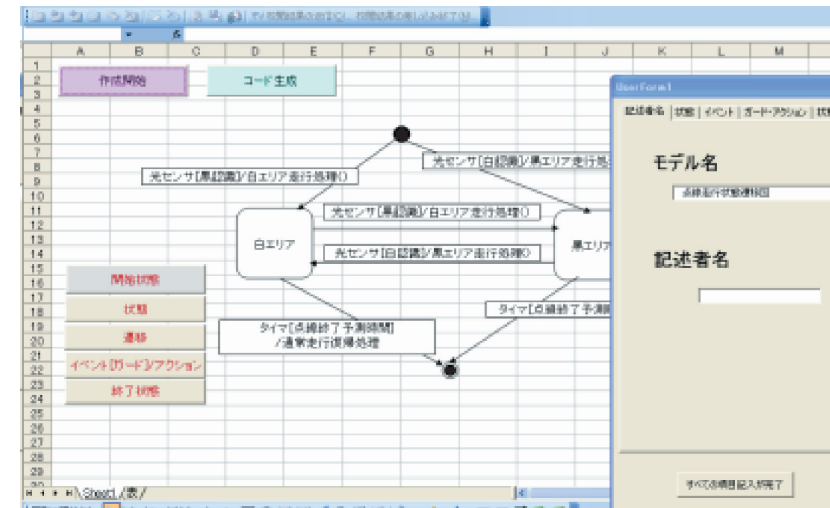


図 9 簡易 MDD ツール  
Fig. 9 A basic tool for MDD.

1 つである点線部分の PROMELA であり、図 11 は、図 10 の PROMELA を検査するための時相論理式 LTL である。点線を含む難所走行部分は、brickOS\_brightness を通じ、並行に動作しているファームウェアから値を得ている。ファームウェアは、複雑ではあるが、光センサ部分を定期的にポーリングする仕組みをとっているため、光センサから値を得ている状態とそれ以外の状態という 2 状態の PROMELA を記述し、点線部分と並行に動作するモデルとした。なお、UML 上では、点線の状態チャートとファームウェアの状態チャートが brickOS\_brightness を介して並行に動作する状況は、シーケンス図で表現する。シーケンス図のメッセージの状態が、LTL の論理式となる。本論文のテーマである「試行錯誤を重ねながら開発」とモデル検査の関係は、実験結果と論理値の比較にある。モデル検査において評価できたタイミングと実際の実験との誤差は 10%程度であった。これはファーム側や、アプリケーション側の負荷の見積りが不十分であることに起因する誤差である。たとえば、ある処理に必要なソフトウェアのためのループカウントは「3000 カウント程度がよい」という結果を出すには、相当な回数の実験が必要となるため、モデル検査により、可能性のある値を絞り込むことにより実験の回数を減らす可能性を見出した。

```

active proctype dashed_drive(){
do
  ::true->
    /* センサから取得した値のチェック */
    sensor_check_flag=1;
    brickOS_brightness?judged_color;
    if
      ::(judged_color == white)->
        motor_dir = left_side;
        white_count++;
        if
          ::(white_count > 100)->
            motor_speed = low_speed;
          ::else->
            skip;
        fi
      fi
      ::(judged_color == gray)->
        motor_dir = right_side;
        black_count++;
        if
          ::(black_count > 10)->
            motor_speed = normal_speed;
            black_count = 0;
            white_count = 0;
          ::else->
            skip;
          fi
        fi
      ::(judged_color == black)->
        motor_dir = right_side;
        black_count++;
        if
          ::(black_count > 10)->
            motor_speed = normal_speed;
            black_count = 0;
            white_count = 0;
          ::else->
            skip;
          fi
        fi
    fi
  od
}

```

図 10 点線部分の PROMELA

Fig. 10 PROMELA statement for dashed line.

## 5. 検 討

本章では、教育効果について検討する。開発開始当初は、すべての学生が、コンテストの課題として明示されている難所を攻略するプログラムを直接作成しようとする。プログラムの作り方も、直感的に 300 msec 間モータを動作させて止める。その結果、たまたまうまく

```

(r || p) U q
q が成り立つまでは、常に r または p である。ただし、
r: white_count <= 100
   (シーケンス図 3: 白カウント)
q: motor_speed == low_speed
p: black_count > 10
   (シーケンス図 7: 黒カウント)
以上から点線の白色から黒色への状態遷移を検査

```

図 11 点線部分の LTL

Fig. 11 LTL statement for dashed line.

く動いたか・動かないかによって良し悪しを判断しプログラムを作成していく。本教育が終了した段階では、まずどのように LEGO を動作させるか決定し、LEGO に装備したセンサが測定する輝度変化がどのようなになるべきか、目標値となる輝度のグラフの形状を考える。実際に、輝度の変化を調べ、測定値をグラフ化する。目標値と測定値を比較し、違いがあれば、その原因を考え、次の実験を考える。このようなプロセスを受講者全員が遂行できるようになった。

効果を評価するために、ロボコン教育受講者 13 人と、未受講者 15 人に対してアンケートを行った。アンケートは下記の質問について自由に記述する方式である。

- (1) LEGO の走行体が点線を通過するのに、必要なことは何か、何をすべきか。
  - (2) 飛行船ロボットで最初の風船に到達するのに、必要なことは何か、何をすべきか。
- (1) の質問のねらいは、受講者と未受講者に違いがあるかをみることにある。(2) のねらいは、類似した課題で、未着手の課題を受講者がどのように考えるかについて調査することである。なお、未受講者には、プログラミングを得意とする学生を選んだ。

(1) について、未受講者の 1 人は、進行方向の認識をどのようにすべきかを重要視した。また、別の未受講者はセンサの取り付け方法による性能の変化に興味を持った。未受講者全員が、点線の入り口の灰色を検出することを指摘している。受講者の 60% は、白を何秒間認識したら脱線と見なすか、センサの付いたヘッド部分が時間あたりにどれくらい動くか、というふうに、具体的な回答だった。アンケートの意図は説明しなかったが、学生たちは、意図を汲んで回答したようだ。経験がなければ、このような回答をすることは困難であるため、意図を汲んでいるかどうかは、試行錯誤の教育評価を阻害するものではないと考える。

アンケート結果において、意外な部分は、受講者の 80% が、コンテスト当日に奇抜な走



行をしたチームの走行について賞賛をこめて記していたことである。点線部分の標準的な走行は、点線に沿って走行するが、奇抜な走行は、1度、故意にコースを外れ、戻ってくるという方法である。点線での奇抜な走行は、一見、競技規約の抜け道を使って楽をするようにみえるため、開発を行っていない見学者の批判が多かった。実際には規約どおりに走るよりはるかに困難なこれらの走行を学生たちが賞賛していたことは、何度も実験を重ね苦労したことの表れと考える。

(2) に関しては、LEGO で電池の消耗を調査していた学生たちは、飛行船についても電池の消耗を指摘していた。このように、LEGO で自分が苦労した点については詳しく記されていたが、飛行船の挙動に関する指摘は未受講者と変わる部分はない。ただし、時間、人、設備等の環境面や、テストを重ねること、計画が重要と記した学生もいた。

受講者に関しては、さらに3番目の質問として、ロボコンで役に立ったことについても調査を行った。そのうちの1人は、開発対象と外部環境の関係性、影響力がどういったものか、という思考の重要性、また同時に解決の難しさを知り、どう解決すればよいかを考えたり調べたりする習慣が身に付いたと答えている。

以上から、トレーサビリティに着目した教育が、試行錯誤の能力を養うのに効果があったといえる。

## 6. まとめ

本論文では、ロボットコンテストを利用した組込み教育の一例を紹介した。試行錯誤を重ねながら開発する能力育成を目指し、トレーサビリティに着目した。試行錯誤の能力を効果的に身に付けるために「開発プロセス」、「ゴール指向分析による課題の発見」、「ドメインに適した実験計画書・報告書のフォーマット」、「測定器の開発」、「設計モデル」、「簡易 MDD ツールの開発」、「モデル検査」それぞれで行った試みについて概説した。

本教育の実施結果における、教育効果は予想以上に高かったが、今後も、様々な点で改善が必要である。たとえば、実験計画書・報告書は冗長な箇所が多いため、効率的でなく、学生の取り組む気持ちを妨げる要因になりかねない。教育効果の評価のために、教育目的をよりはっきりと定義することも必要である。特に、今回の評価方法は、曖昧で直観的であった。今後、評価方法についても洗練し、より効果的な教育活動になるように努めたい。

謝辞 本研究は、財団法人栢森情報科学振興財団の研究助成を受けたものである。

## 参考文献

- 1) 二上貴夫, 鷺崎弘宜, 小林靖英, 乾 裕紀, 大槻博之, 仲久保正人, 久保寺勇気, 川縁幸平, 羽田千織, 三橋祐仁, 沼里京介: MDD ロボットチャレンジ 2006 開催報告, 研究報告「ソフトウェア工学」, No.2007-SE-156, pp.79-86, 情報処理学会 (2007).
- 2) 鷺崎弘宜, 久保秋真, 小林靖英, 渡辺晴美, 小倉信彦, 飯田周作: MDD チャレンジにみる組込みソフトウェアモデル中心開発の工学と教育, 研究報告「ソフトウェア工学」, No.2007-SE-156, pp.95-102, 情報処理学会 (2007).
- 3) 山下博之: 小中学生を対象としたロボット競技会と総合理科教育, 情報処理, Vol.48, No.5, pp.502-511, 情報処理学会 (2007).
- 4) Polya, G.: *How to Solve It*, Princeton University Press (1971).
- 5) <http://www.etrobo.jp/>
- 6) Martin, F.: *Ideal and Real Systems: A Study of Notions of Control in Undergraduates Who Design Robots* (1994). <http://llk.media.mit.edu/papers.php>
- 7) Resnick, M. and Ocko, S.: *LEGO/Logo: Learning Through and About Design, Constructionism*, Harel, I. and Papert, S. (Eds.), Ablex Publishing, Norwood, NJ (1991).
- 8) Holzmann, G.J.: *The SPIN Model Checker: Primer and Reference Manual*, Addison-Wesley Professional (2003).

(平成 20 年 1 月 9 日受付)

(平成 20 年 7 月 1 日採録)



小倉 信彦

1971 年生, 1995 年東京工業大学工学部情報工学科卒業, 1997 年東京工業大学大学院理工学研究科電気電子工学専攻修士課程修了, 同年東京工業大学精密工学研究所助手。2004 年東京工業大学において博士(工学)を取得。2005 年武蔵工業大学環境情報学部講師, 2007 年同学部准教授, 現在に至る。信号処理, 最適化, 不動点についての研究に従事。電気情報通信学会および日本音響学会各会員。



渡辺 晴美 (正会員)

1990年東京工科大学情報工学科卒業。1990年から1993年まで日本電気マイコンテクノロジー(株)勤務。1995年東京工科大学大学院工学研究科システム電子工学専攻修士課程修了。1998年東京工業大学大学院情報理工学研究科計算工学専攻博士課程修了,博士(工学)。1998年北陸先端科学技術大学院大学情報理工学研究科リサーチアソシエイト,2000年立命館大学理工学部助手,2004年東海大学開発工学部講師,2006年同大学同学部准教授。2008年東海大学情報通信学部准教授,現在に至る。組み込みソフトウェアシンポジウム ESS2004 実行委員長(2004年),NHK ロボコン東海北陸大会審査委員(2006年)(社)全国工業高等学校校長協会第7回高校生ものづくりコンテスト全国大会電子回路組立部門審査委員長(2007年)。組み込みソフトウェア開発に関する研究,ロボット教育に興味を持つ。

---