

## 日本語プログラミングを用いた 教育軽減による開発プロジェクトの効率化

高瀬 えりか<sup>†</sup> 江見 圭司<sup>†</sup>

プロジェクトの効率化のために開発言語に日本語プログラミング「ドリトル」を用いて、C#で開発した場合と比較した。実施例として MDD ロボットチャレンジを取り上げる。MDD ロボットチャレンジとは小型飛行船を飛ばす制御のソフトウェアを開発し、その成果を競うものである。大会に向けた開発は、複数人によるチーム開発によって行われ、下記の3点を柱として行われた。

- ・オブジェクト指向による設計
- ・ソフトウェア開発
- ・ハードウェア開発 (HERO 社担当のため、本論文では割愛する)

開発期間が限られているため、開発に必要な知識習得とソフトウェア開発を両立させる必要がある。本論文では、知識習得のための教育軽減によってプロジェクトの効率化が可能かどうかを検証する。

## Making efficient development project by reduction of teaching hours by using Japanese language programing.

Erica TAKASE <sup>†</sup> Keiji EMI <sup>†</sup>

We compared using Japanese programming language "Dolittle" with using C#. In this study, we executed "MDD robot challenge". "MDD robot challenge" is a robot-controlling contest, to develop software to control the small airship. It's developed by the team, and it's important for the following three elements.

- ・ Object-oriented design.
- ・ Software Development.
- ・ Hardware development, (HERO Co., LTD who is responsible, in this project is omitted).

Because of the short development period, we have both learning and developing. In this paper, we verify that project could be efficient by reduction of teaching hours.

<sup>†</sup> 京都情報大学院大学

The Kyoto College of Graduate Studies for Informatics

### 1. はじめに

#### 1.1 研究の目的

”制御装置のプログラミング” (楠田達文著) [1]の中で、著者はソフトウェアをゲーム、情報処理、機械制御の3種類に分類している。組み込みソフトウェアがゲームや情報処理系のソフトウェアと異なる点は、現実に存在する機械を動かさなければならないことや機械の出来不出来や故障、周囲の環境など不確定要素が数多くあることである。

本研究では、この組み込みソフトウェア開発のプロジェクトを実施例に挙げて述べる。今回実施例に用いた開発プロジェクトでは、必要な開発知識の修得と開発を並行して行なう必要があり、開発期間が限られているために効率良くプロジェクトを進める必要があった。そのため、日本語でのプログラミングを用いて開発言語の修得時間及び開発時間の短縮を試みた。

#### 1.2 実践例

##### 1.2.1 MDD ロボットチャレンジの概要と審査基準

本研究では、京都情報大学院大学 (以下 KCGI)・京都コンピュータ学院 (以下 KCG)・株式会社ヒューマン エンジニアリング アンド ロボティクス社 (以下 HERO 社) の産学連携プロジェクトとして行った MDD ロボットチャレンジ 2007 年～2009 年度大会への参加を実践例 (以下、本プロジェクト) として取りあげる。

MDD ロボットチャレンジとは、小型飛行船を制御するソフトウェアをモデル駆動開発 (Model Driven Development 以下 MDD) に従って開発・実践・研究する機会と、組み込み技術者育成の場を提供している大会であり、毎年十数チームの参加がある。情報処理学会の組み込みシステム研究会が主催している [2]。チャレンジと名のつくように、この大会は完成された技術や知識を競うのではなく、新しい技術や未経験の問題に挑戦することが重要視されている。

審査基準はモデル審査と飛行競技の2つから成る。

モデルは、本プロジェクトでは Unified Modeling Language (以下 UML) を用いた。システムに要求する機能はユースケース、構造はクラス図 (必要に応じてオブジェクト図)、さらにシステムの構造をシーケンス図で表現する必要がある。また、作成したモデルからコーディングに至るまでのコード変換ルールを定めることも審査基準に含まれる。他チームでも UML を主に用いているが、Systems Modeling Language やアスペクト指向分析などを取り入れたモデリング手法も見られる。このモデルからコーディングを行う。開発言語は多くのチームが C++ を採用しているが、開発言語そのものに関する規定はない。

飛行競技は”自動航法競技”と呼ばれ、これは飛行船を出発地点から規定の立ち寄りポイントを通りながら目的地に着陸させる一連の動作を自動制御で行うことを競うものである。また、競技中に GUI で飛行の様子を表示する必要がある[3]。

総合評価点は、モデルの点数と競技飛行の点数の積で求められる。2008 年度の大会では我々のチームは飛行競技 1 位、モデル審査 6 位、2009 年度は飛行競技 13 位、モデル審査 6 位であった。

モデルと競技飛行の評価は同等に行われるのではなく、モデルの評価が特に重視される。これは大会の趣旨がモデル駆動開発でのソフトウェアを競うことであり、モデルの完成度と実際の飛行にモデル内容が反映されることが重要となるためである。さらに飛行競技では競技時の気温や気流に影響されやすく、運の要素も勝敗を左右するといえる。

### 1.2.2 プロジェクト概要

本プロジェクトでは、MDD に基づいたチームでの組み込み開発を実践する。また、下記の 2 つが柱として挙げられる。

#### a. ソフトウェア開発

この大会では、MDD に基づいて開発を行う必要がある。我々のチームではオブジェクト指向開発のための統一モデリング言語、UML を用いてモデル作成を行う。

そして、作成したモデルに基づいて飛行船制御のためのソフトウェア開発を行う。KCG, KCGI に所属するメンバーは主にソフトウェア開発を担当した。

開発言語には 2007 年、2009 年大会ではドリトル、2008 年大会では C# を用いた。

#### b. ハードウェア面の開発

こちらは産学連携プロジェクトの為、HERO 社が担当している。

## 2. 開発環境

### 2.1 本プロジェクトの遂行における問題分析

本プロジェクトでは、競技規定が発表される 5 月以降から大会が行われる 10 月の間が開発期間と言える。

プロジェクトの大まかなスケジュールは下記の通りである。

- ・4 月下旬～7 月～ 大会エントリー、勉強会等
- ・6～10 月下旬～ ソフトウェア開発および提出ドキュメント作成
- ・10 月下旬 大会

2007 年度に行ったプロジェクトでは、開始から大会までの期間が 8/24～10/19 で、大会エントリー開始の 4 月下旬～10 月下旬の半年間を通常考えられる開発期間として比較すると、約 35%であった[4]。さらにメンバーには C 言語を学習しただけという初心者もいた。

極めて短い開発期間で、初心者も含めたチームでのソフトウェア開発を行う必要があり、その手段のひとつとして開発言語に日本語プログラミング言語「ドリトル」を用いた[5]。

それ以降の 2008 年度、2009 年度のプロジェクトにおいても、実際にメンバーがプロジェクトに充てられる時間が少ないという問題が生じている。さらに、プロジェクトの初期段階では非経験者は必要な知識や技術の習得が最優先事項となる。本プロジェクトでは、前年度以前から引き続き参加しているメンバーを経験者、初めて大会に参加するメンバーを非経験者と定義した。

本プロジェクトでは

- ① オブジェクト指向によるモデリングのための知識
  - ② ソフトウェアの開発言語の知識
- の 2 つの知識が特に重要となる。

非経験者はモデリングやプログラミングの知識・技術を、勉強会への参加や KCG・KCGI の授業を受講することなどによって修得していく。ある程度知識が身に付いてきた時点で、実際に競技で用いるソフトウェア開発に携わっていくが、知識・技術の習得は大会当日・あるいは大会後まで続く。

経験者はプロジェクトの初期段階では非経験者の育成、またはその年度の大会でさらに必要となる知識・技術の習得を最優先で行い、ソフトウェア開発も並行して開始していく必要がある (図 1)。

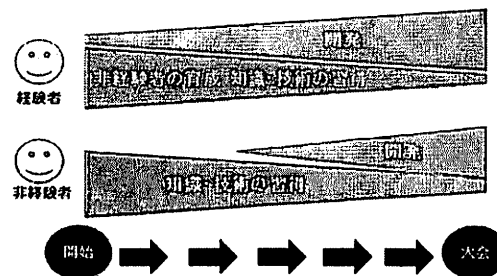


図 1 プロジェクトの進行

このため、短期間で効率的な学習・開発が必要となる。

MDD ロボットチャレンジは作品の出来栄や優劣を競う「コンテスト」ではなく大会名に「チャレンジ」と冠している通り、開発に挑戦することにも意義があるという大

会の趣旨を踏まえて、メンバー募集の際に技術や知識のレベルを問うことはしていない。さらに、モデリングに必要な知識・技術の習得やモデリングの工数を短縮することは今回は優先しなかった。MDD ロボットチャレンジはモデル駆動開発で開発を行うことを趣旨としており、モデルの完成度を重視するためである。

本プロジェクトではソフトウェアの開発言語修得の教育にかかる時間と、ソフトウェア開発（コーディング）時間を開発工数削減の対象とした。

## 2.2 開発言語

2007年度、2009年度のMDDロボットチャレンジの開発言語にはドリトルを用いた。ドリトルとは、大阪電気通信大学の兼宗進教授によって教育用に開発されたプログラミング言語である[6]。本プロジェクトでの開発におけるドリトルのメリットは以下の点が挙げられる。

- ・日本語で記述可能で、言語習得が容易であること。
  - ・ドリトルの文法や命令語は日本語を母国語として会話する日本人にとってなじみやすい。これが言語習得のハードルを下げると思われる。
  - ・GUIと組み込み制御の実装が容易である
- MDD ロボットチャレンジの競技規定では、飛行の様子を画面上に表示させる必要がある。このため、GUIの実装が容易である言語は開発しやすい。組み込み制御についても同様である。

## 3. 開発および言語修得の所要時間の比較

### 3.1 開発言語習得のための教育にかかる時間

ソフトウェア開発にあたっての基本的な言語習得の教育に必要な時間について述べる。ドリトルの所要時間は約18時間、C#の所要時間は約42時間となった。

ドリトルの言語習得の所要時間を約18時間とした根拠は、KCGで2009年3月に実施したスプリングコース、2009年9月に行われたサマーコースのドリトル入門講座のシラバスより、飛行競技に必要な要素を修得するための講義のみ抜粋すると、

1.5時間（1コマ）×12回=18時間となる。抜粋した講義を下記に示す。

- ・基本文法（1コマ）
- ・描画（2コマ）
- ・ミニゲーム作成（3コマ）
- ・音楽演奏（1コマ）
- ・チャット、ネットワークゲーム作成（3コマ）

- ・ロボット制御（2コマ）

C#の所要時間は42時間としているが、こちらの根拠はKCGで実施されている授業「C#実習1」と「C#実習2」のシラバスより、飛行競技に必要な要素を習得するための講義のみ抜粋すると

3時間（2コマ）×14回=42時間となるためである。

「C#実習1」「C#実習2」のシラバスを以下に示す。各項目につき、2コマ必要となる。

- ・基礎事項、変数、予約語、演算、簡単な入力操作
- ・分岐構造、ループ構造
- ・配列の仕組み
- ・メソッドの仕組み、再帰メソッド、クラスの基礎
- ・namespaceの仕組み
- ・Destruction, Garbage Collection, オーバロード
- ・プロパティ、継承、仮想関数、メソッドオーバーライド
- ・抽象クラス、sealクラス、Objectクラス、構造体、列挙体
- ・インデクサ、可変長引数、文字列と正規表現
- ・インタフェース、例外、ストリーム
- ・Stream, Delegate & Event
- ・Thread
- ・Generic Programming, Lambda Expression, Anonymous Type
- ・GUIプログラミングの基礎事項、GDI+による描画処理

### 3.2 ソフトウェア開発時間

ソフトウェア開発時間は、飛行船が飛ぶ状態のプロトタイプのソフトウェアが出来るまでの所要時間を比較した。ドリトルでは6時間、C#では11時間かかった。

ドリトルの場合、GUIの実装が容易であるため、開発時間を短縮できたと考えられる。

ドリトルの開発時間は2007年度大会、C#は2008年度の大会の開発に基づいている。飛行船は外的環境やハードウェアの状態など多くの影響を受けるため、それ以後の細かい数値等の調整はドリトル・C#ともに同程度の時間がかかっている。

2009年度の大会でも開発にドリトルを用いたが、2009年度はドリトルがバージョンアップしたため2007年度に使用したソフトウェアの改造を行った。2009年の開発時間は最初からソフトウェアを作成するのではなく、改造にかかった時間となるためC#での開発時間と比較するには妥当ではないため、2007年度の開発時間を用いた。

### 3.3 オブジェクト指向での設計から実装へ

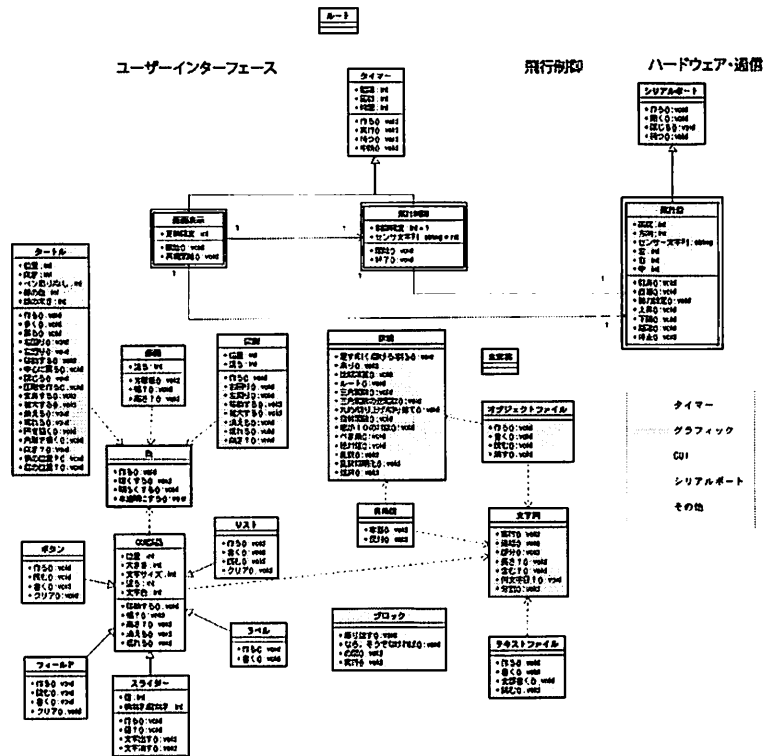


図2 飛行競技のクラス図

図(図2)は飛行競技用のソフトウェアに用いたクラス図である。ドリトルで予め用意されているライブラリを活用したため、新しく作成したクラスは「画面制御」、「飛行制御」、「飛行船」の3つのクラスのみであった。ライブラリを活用することで、コーディングも容易に行なうことができた。

### 4.おわりに

3.1 で述べたように、C#と比較した場合ドリトルの方が言語の知識習得の教育に必要な時間が少なかった。

これは、2.2 でも述べたように、ドリトルは GUI の実装を容易に行なうことができるため、教育に必要な時間やソフトウェア開発の時間の短縮につながったと考えられる。

また、ドリトルは本来持っている機能の数が C#よりも少ない。このため、教育に必要な時間を短縮できたとも言える。C#でも機能を限定して学習を行えば、ある程度所要時間を短縮できる可能性もある。

本研究の実施例では、飛行船を競技規定やチームの目的に合わせて飛行させるためのソフトウェアが必要であった。また、本プロジェクトの特徴として、必要な知識や開発言語の修得とソフトウェア開発を限られた時間で行なう必要があった。ドリトルは本プロジェクトに必要な要素を持ち合わせていた。

ドリトルは教育用、Java や C 言語はプロフェッショナルな製品開発用として開発されているが、開発するソフトウェアに適しており、プロジェクトの個々の状況に合った言語を選択するのであれば用途に限定される必要は無いと我々は考えている。

### 5.謝辞

ドリトルの使用にあたり、開発者である大阪電気通信大学 兼宗進教授、飛行船のハードウェア開発にあたり HERO 社の岡村社長をはじめとする社員の皆様、MDD ロボットチャレンジの参加にあたり KCGI・KCG の参加メンバーの皆様のご協力を賜りましたことを感謝いたします。

### 6.参考文献

- [1] 楠山達文 "装置制御のプログラミング", p13-20, CQ 出版社, 1997.
- [2] MDD ロボットチャレンジ2009" <http://sdlab.sys.wakayama-u.ac.jp/mdd2009/>
- [3] MDD ロボットチャレンジ2009 競技仕様書
- [4] 中村州男,ほか "教育用プログラミング言語を用いたプロジェクト工数削減と多段階開発の実践", 情報処理学会, SE-160 EMB-9, 2008.
- [5] 高橋修司, ほか "ドリトルを用いたモデル化・シミュレーション・オブジェクト指向開発の自学自習実践", 情報処理学会 第 94 回コンピュータと教育研究発表会, 2008.
- [6] (a) 兼宗進, 久野 靖:ドリトルで学ぶプログラミング-グラフィックス, 音楽, ネットワーク, ロボット制御, イーテキスト研究所(2008).
- (b) 兼宗進:プログラミング言語「ドリトル」 <http://dolittle.eplang.jp/>