

ドリトルを用いたオブジェクト指向チーム開発学習の実践と評価

高瀬 えりか 村上 智史 後藤 洋信 坂本 雅洋 江見 圭司
 京都コンピュータ学院・京都情報大学院大学

インスタンススペースのオブジェクト指向教育用プログラミング言語である「ドリトル」を用いて、小型飛行船を制御するためのソフトウェアをチームで開発した。ドリトルは、日本語で記述するプログラミング言語であるため、他の言語を用いるよりも学習・開発時間の短縮が可能であると我々は考えていることを報告する。

本稿は、MDD ロボットチャレンジ 2007 および 2008 相撲競技に参加し、実践したこれらの報告である。

Practice and Evaluation of Team Development Learning with Japanese Programing Language "Dolittle"

Erica TAKASE, Satoshi MURAKAMI, Hironobu GOTOH, Masahiro SAKAMOTO and Keiji EMI
 Kyoto Computer Gakuin and the Kyoto College of Graduate Studies for Informatics

"Dolittle" is an object-based language for education with Japanese language. We have developed software to control a miniature airship for contests.

We are going to report that we can save both developing time and learning time by using Japanese program language "Dolittle", compearing with other common programming language.

This paper is that we joined the "MDD robot challenge 2007" and "MDD robot challenge 2008 sumo wrestling game ".

1. はじめに

1.1. MDD ロボットチャレンジとは

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に従って開発・実践・研究する機会と、組込み技術者育成の場を提供している大会である[1]。開発に用いたモデル、飛行競技をコンテスト形式で競う。本研究では、2008 年に参加した MDD ロボットチャレンジの「相撲競技」における、ドリトルを用いたオブジェクト指向チーム開発学習の実践と評価について報告する。

1.2. 相撲競技とは

相撲競技は、飛行船を操作して1対1で対戦し、勝敗を競う。操作はコントローラや

マウス、キーボード、ゲームのコントローラなどを用いて行う。今回我々は飛行船操作のためのソフトウェア開発にあたって、ドリトルを用いた。

1.3. 本研究の目的

日本語プログラミング言語であるドリトル用いて開発を行うことで、学習・開発にかかる時間を短縮し、短期間でシステムを実装することを目的とした。また、ドリトルはインスタンススペースのオブジェクト指向言語である。オブジェクト指向による設計は、UML での記述により、分析・設計段階におけるドキュメントの標準化を図ることができる。

3.3. 第三段階

実装を行うため、クラス図とシーケンス図を作成した。(図3 参照,)

操作のためのボタンクラス他、飛行船と通信するシリアルポートクラス、シリアルポートを継承するモータ制御クラスなどから構成されている。

また、オブジェクト間のデータの流れを把握するためにシーケンス図を作成した。

3.4. 第四段階

モデルを基に、ドリトルでソースコードを実装した。ドリトルは日本語で記述するため、モデルから汎用言語でプログラムを作成する際に必要な、日本語から英語あるいはローマ字表記への変換作業を省略することができた[7]。実際のコーディングは筆者(村上)が行った。設計さえ固まっていれば、ドリトルでの実装は難しくないと考えられる。ソースコードと、実行後の操作画面は図の通りである(図4, 図5 参照)

3.5. 開発工数

ここで、開発工数の比較を行いたい。

MDD ロボットチャレンジ 2007 では、ドリトルを用いて自動航法競技に用いるソフトウェアを開発した。図(図6 参照)は開発項目ごとの開発工数である。

自動航法競技とは、飛行船を自動制御で目的地まで到着させることを競う競技である。

翌年の MDD ロボットチャレンジ 2008 では、自動航法競技に用いるソフトウェアはドリトルに代わり C#を用いて開発を行った。開発にかかった工数は、分析・設計には 6.6 人日、実装には 11.2 人日、計 17.8 日かかった。(ただしハードウェア調整・実験・シミュレーションにかかった日数は除く) ドリトルでの開発に計 12.6 日かかったことから、こちらの方が 5.4 人日分開発工数を短縮できたことが分かる。

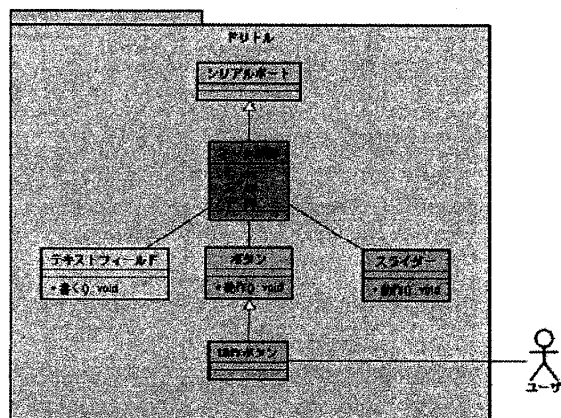


図3 クラス図

```

112 終了ボタン=ボタン! 『通信終了』
113 140 -175 位置 150 50 大きさ
114 終了ボタン: 動作=「モータ制御!
115 開始ボタン=ボタン! 『通信開始』
116 140 -175 位置 150 50 大きさ
117 開始ボタン: 動作=「! 消える。方
  
```

図4 ソースコード

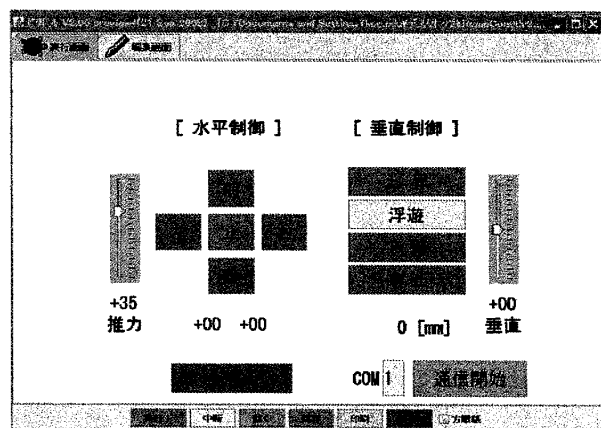


図5 操作画面

開発項目	人数[人]	工数[人日]
分析・設計	3	6.6
実装	2	6.0

図6 2007年自動航法競技の開発工数

実際の開発とは別に、開発言語の学習にかかる時間の比較も行う。

ドリトルでは本開発に必要なスキルを修得するのに2～6時間程度かかりが、C#では45時間以上かかった。ドリトルの場合は、インターフェースの開発について勉強する必要がない。このことが学習時間を大幅に短縮させる要素のひとつであると考えられる。コーディングとコーディングのための学習時間が少なく済んだことにより、チームで分析・設計することに注力できた。

ドリトルが日本語プログラミングであることは修得時間を短縮し、開発のためのコーディング時間を短縮することができたといえる。

4. おわりに

MDD ロボットチャレンジ 2008 相撲競技において、我々はドリトルを用いたオブジェクト指向開発を行った。UML のモデルを用いたドキュメントの標準化は、設計後の効率の良い実装につながった。

MDD ロボットチャレンジ 2007 ではドリトルを用いて開発を行い、自動航法競技用のソフトウェアを開発できた。また、翌年のC#での自動航法競技用ソフトウェア開発と比較した結果、開発言語の学習時間や開発工数の短縮を図れることも分かった。

これらのことから、ドリトルは教育用の日本語プログラミングであるが、教育にとどまらず、他にも実用可能なソフトウェアを効率よく開発できる可能性があると考えている。今後の課題は、中高生などプログラミング初心者向けへの教育の実践と実用可能なソフトウェアの開発である。

謝辞

ドリトルの使用にあたっては、兼宗進一橋大学准教授、にお世話になりました。

MDD ロボットチャレンジ 2007 および2008 参加・ハードウェアの開発にあたっ

ては、株式会社ヒューマンエンジニアリングロボティックスの岡村勝氏、西村憲二氏、高橋嘉也氏、竹内勇貴氏、松井委宏樹氏、岡部拓矢氏、古川彬氏、にお世話になりました。また、京都情報大学院大学の中村州男氏、高橋修司氏にもお世話になりました。

参考文献

- [1] 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」第94回コンピュータと教育研究発表会 (2008)
- [2] (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) プログラミング言語「ドリトル」
<http://dolittle.eplang.jp/>
- [3] 兼宗 進, "教育用プログラミング言語の動向", 情報処理, 特集 教育用プログラミング言語と授業利用, Vol.48 (2007)
- [4] 兼宗進, ほか 「初中等教育におけるオブジェクト指向プログラミングの実践と評価」 情報処理学会論文誌, Vol.42 (2002)
- [5] 竹政昭利 『はじめて学ぶ UML 第二版』(ナツメ社, 2007)
- [6] 高橋修司, ほか "MDD ロボットチャレンジ 2008", モデル審査資料 (2008)
- [7] 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」 社団法人情報処理学会 (2008)