

# 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

高橋 修司, 中村 州男, 江見 圭司 (京都情報大学院大学)  
村上 智史, 橋本 拓哉, 橋川 竜起 (京都コンピュータ学院)  
西村 憲二, 高橋 嘉也, 竹内 勇貴, 松井 宏樹, 岡村 勝  
(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)

**要約:** オブジェクト指向教育用日本語プログラミング言語であるドリトルを用いて, MDD ロボットチャレンジに参加した。参加するにあたって, 飛行のモデル化とシミュレーション・オブジェクト指向開発の自学・自習・実践を行った。本稿は, この報告である。

**We had studied for self-learning, self-teaching and practice modeling, simulation and object-oriented development using Dolittle.**

Shuji TAKAHASHI, Kunio NAKAMURA and Keiji EMI  
The Kyoto College of Graduate Studies for Informatics  
Satoshi MURAKAMI, Takuya HASHIMOTO and Ryuki HASHIKAWA  
Kyoto Computer Gakuin  
Kenji NISHIMURA, Yoshiya TAKAHASHI, Yuki TAKEUCHI, Hiroki MATSUI and Masaru OKAMURA  
Human Engineering and Robotics Co., LTD.

We have taken part in MDD Robot Challenge 2007 using "Dolittle : object-oriented educational programming language with Japanese language". This report is that We have practice modeling, simulation and object-oriented development at MDD Robot Challenge 2007 with self-learning and self-teaching.

## 1. はじめに

### 1.1. MDDロボットチャレンジとは

組込みシステムシンポジウム (ESS) の特別企画として, MDD (Model Driven Development モデル駆動開発) ロボットチャレンジを開催し, 組込みシステム開発を实践, 研究する機会と, 組込み技術者育成の場を提供している [1][2][3]。

### 1.2. MDDチャレンジの目的

MDD チャレンジの目的を「MDD チャレンジ参加へのお誘い」から引用しておく [4]。

#### 1.2.1 研究促進

研究促進: 組み込みソフトウェアとシステム工学推進のための研究用試料を開発する。

下記領域について開発実践しながら情報を蓄積し研究に役立てる。

ドメインテーマ (飛行船制御)

ハードウェア (電気, 電子, 機械, 計測など)

ソフトウェア (制御, 通信, 情報処理など)

評価技術 (モデル評価)

運営管理技術 (開発プロジェクト運営, 教育イベント運営)

#### 1.2.2 産業振興

産業振興: MDD というソフトウェア工学の開発パラダイムを学ぶ機会を提供する。

UML モデルレイヤーの 1 層 (実装), 2

層(アプリモデリング), 3層(メタモデリング)開発技術習得

開発した試料情報を開示/展示して産業振興に役立てます。

### 1.2.3 教育実践

教育実践: 情報技術教育のプラットフォームとしての模型飛行船開発を行う。

夢中になれる組込み技術(情報科学, 組込みシステム) → 中高校生

工学原理を適用した組込み技術(UMLモデリング, MDD) → 大学生

製品開発を改善する要素としての組込み技術(MDDの実用) → 社会人

### 1.3. 我々のチームの概要

MDDロボットチャレンジのチームは, 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(HERO)の産学協同チームとしてプロジェクトが発足した。主な役割として, KCGI・KCGはモデル開発及びソフトウェア開発, HEROは実機の開発である。本チームのメンバー構成は以下の通りである。KCGIから江見圭司・高橋修司・中村州男, KCGから, 橋本拓哉・村上智史・橋川竜起である。HEROから, 西村憲二・高橋嘉也・竹内勇貴・松井宏樹である。開発言語はドリトルを使用した。本チームは, 大会に参加することによって, 産学協同で「MDD」を習得と, 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を目的としている。

本MDDロボットチャレンジにおいて, 我々は下記の戦略に基づいてプロジェクトを推進することにした。

概要として, オブジェクト指向開発, モデルの段階的作成開発, 短期間によるチーム開発, 初心者による開発, モデル化とシミュレーションによる開発がある。

### 2. 何が自学自習か

MDDロボットチャレンジにおいて, 我々

は, 下記の領域を実践した。

- (A)オブジェクト指向開発
- (B)プログラミング
- (C)シミュレーション
- (D)プロジェクトマネジメント

### 2.1. オブジェクト指向開発

モデル開発などを担当する, KCGI・KCGの学生メンバーの全てが, オブジェクト指向開発, UML, MDD に対して初心者であった。

そのため, MDD については過去の大会の要旨集で[2][3], 学習した。

オブジェクト指向開発・UMLについては, 京都情報大学院大学の講義のオブジェクト指向システム設計を参考に自学自習した[5][6][7][8][9][10][11]。

### 2.2. プログラミング

我々のように, C言語(ANSI)を学習しただけという, 初心者の多いチームが従来のオブジェクト指向言語を用いて, 開発を行うのは困難であると考えた。そこで我々は, オブジェクト指向でありながら, 開発工数・開発期間を大幅に削減できる言語やツールを探すことにし, 結果, 教育用日本語プログラミング言語「ドリトル」に行き着いた。以下に, 開発工数・開発期間を削減できる理由を, 開発者である兼宗進氏の論文を参考に述べる。[12]

- ・理解しやすく習得の時間が短い, 簡潔な構文, 日本語との対応。
- ・動かしながら段階的に学べる, インクリメンタルなプログラミング。
- ・基本的な計算機の原理を学べる, テキストによる記述, アルゴリズムと構造化。
- ・日常使うソフトウェアの原理に結び付く, オブジェクト指向, プロトタイプ方式。
- ・ネットワークを体験できる。

#### 2.2.1 ドリトルとは

学校教育(小中高)を中心に, 大学生や社会

人にも適した入門用のプログラミング言語です。BASICやLOGOといった、30年以上前に設計された言語に代り、最新のオブジェクト指向の考え方でプログラムを学ぶことができるのが特徴である。[13]

## 2.22 ドリトルの習得方法

兼宗進氏のwebに載っている、ドリトル入門[14]・一橋大学 計算機概論(2006)[15]、この2つのコンテンツで習得した。初心者向けに、アンプラグドのテキスト[16]、上級者向けに兼宗進氏の博士論文がある。

## 2.23 RS-232C制御

RS-232C制御はミュウで習得した。[17]ミュウで制御プログラムを作成後、飛行船制御プログラムを作成した。

2段階に分けることにより、飛行船制御プログラム作成時に頓挫することなく開発が出来るようになった。

## 2.3 シミュレーション

実機を使用した実験を前に、シミュレーションを行った。これにより機体開発が遅れていたり、学業の都合上、実機実験に参加できない場合でも開発そのものができるようにした。

## 2.4 プロジェクトマネジメント

プロジェクトマネジメントを行った学生メンバーがおらず、京都情報大学院大学の講義のプロジェクト管理技法やIPAのプロジェクトマネージャ試験を元に自学自習した[18][19]。

## 3. オブジェクト指向開発

### 3.1. 飛行戦略

要求モデルを作成するにあたり、「一定時間経過後に必ず着陸する」という基本的な飛行戦略をたてた。これにより、制御不能による失格ということを防いだ。

### 3.2. 飛行制御の状態遷移表

飛行制御の状態遷移表を作成し、基本的な飛行戦略から、競技に耐えられる飛行戦略を段階的に作成した。

### 3.3. モデルの作成方法

モデルの作成は、ホワイトボードに記述し、デジカメで保存、清書のみを株式会社チェンジビジョン社のJUDE/Community[20]にて行った。このため、モデル作成に集中でき、効率よく短期間で進んだ。実機の開発状況と開発能力に対応し、飛行モデルを4段階に分けて作成した。

### 3.4. モデルの作成過程

飛行戦略である飛行制御の状態遷移表をもとにモデルの作成(ユースケース図、クラス図、シーケンス図、ステートチャート図)を行った。

### 3.5. モデルの内容

#### 3.5.1 概念モデル

概念モデルはユースケース図を作成した。

#### 3.5.2 設計モデル

設計モデルはクラス図を作成した。クラスを5つだけと非常に簡潔に作成した。ユーザインタフェース部分はドリトルにあるものを使っているため、クラス図に記載していない。

#### 3.5.3 実装モデル

実装モデルは、ドリトルに依存したモデルで、シーケンス図とステートチャート図で表現した。

シーケンス図は、ハードウェア間のデータフローのものと、飛行中のオブジェクト間の通信のものを作成した。

ステートチャート図は先に記述したとおり4段階に分けて作成した。

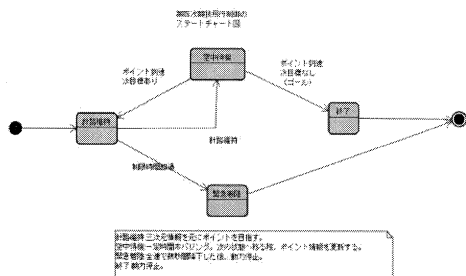


図1 第4次飛行モデルのシーケンス図

## 4. プログラミング

### 4.1. 飛行状況をGUIで表す

ドリトルには標準でグラフィック関係のオブジェクトやGUIオブジェクトがあり、簡単に作成することができる。例えば、ドリトルでは、「カメ太=タートル!作る。」でGUIオブジェクトが作成できる。

これがWin32APIでGUIオブジェクトを作成できるレベルになるためには、京都コンピュータ学院(KCG)の標準講義時間数で換算して、60時間かかる[21][22]。

### 4.2. シリアルポート制御

ドリトルでは簡単にシリアルポート制御ができる。例えば、ミュウを使用すると、「ロボ太=MYU!"com1"作る。ロボ太!ひらけごま消える。」でcom1ポートを指定してミュウロボオブジェクトを作り、パソコンとミュウとの通信が可能になる。

これをKCGの標準学生が、Win32API習得後できるようになる場合はさらに標準講義時間数で換算して60時間かかる[23]。

### 4.3. UMLからのプログラミング

UMLモデルからドリトルにする場合、日本語のモデルから英語コードへの変換がいらないため、すぐに変換ができた。

UMLモデルからC言語にする場合、KCGの標準学生の場合10時間かかる[24]。

### 4.4. ドリトルを使ったメリット

ドリトルを使用することにより、言語習得

が容易になり、開発工程の削減ができる。ドリトルを使用した場合、KCGの標準学生は合計20時間で習得ができ、約110時間の開発工数の短縮ができた。このため、MDDロボットチャレンジの本質である、モデル化に開発工数をとることができる。また短縮した開発工数をプロジェクトマネジメントの習得に当てることができるようになった。

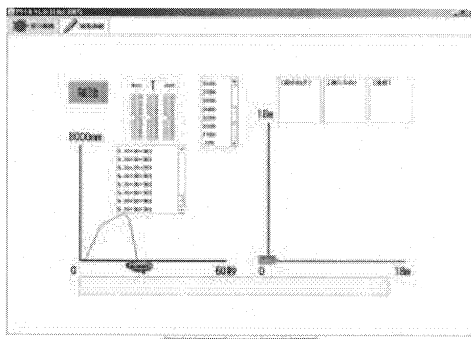


図2 第1次飛行モデルのモデル ドリトル実行画面

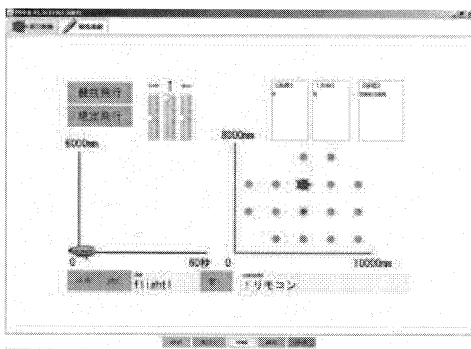


図3 第4次飛行モデルのモデル ドリトル実行画面

## 5. シミュレーション

モデル化して画面上にシミュレーションしたうえで、実体である飛行船を飛ばし、結果を画面上に表示している。通常他のMDDロボットチャレンジ参加者は、モデル化して実体である飛行船を飛ばし、結果を画面上に表

示している。事前にシミュレーションは行われていない。

## 6. プロジェクトマネジメント

本プロジェクトに対し、ガントチャートを作成し、プロジェクトを管理した。ガントチャートを利用することにより、行き当たりばったりな開発や、「デスマーチ」にならなかった。

### 6.1. スコープマネジメント

スコープを視覚化して、タスクを明確にした。これにより、役割分担が行われ、各個人の作業に集中できた。また現状の作業が全体の中でどのような役割を果たしているかを理解でき、効率的に開発が行われた。

### 6.2. タイムマネジメント

スケジュールの全体を視覚化して、開発を行った。これにより、全体のスケジュールの把握と、現在のタスクの状況がわかり、無理なく開発ができるようになった。

### 6.3. リスクマネジメント

リスクマネジメントは、スコープマネジメントを詳細化することにより解決した。具体的には、飛行戦略を簡略化して、制御不能による失格を防いだ。強制着陸を搭載することにより、プロジェクトの最終目的が必ず着地するというように決めたため、競技の失格という最悪の事態は防ぐことができた。

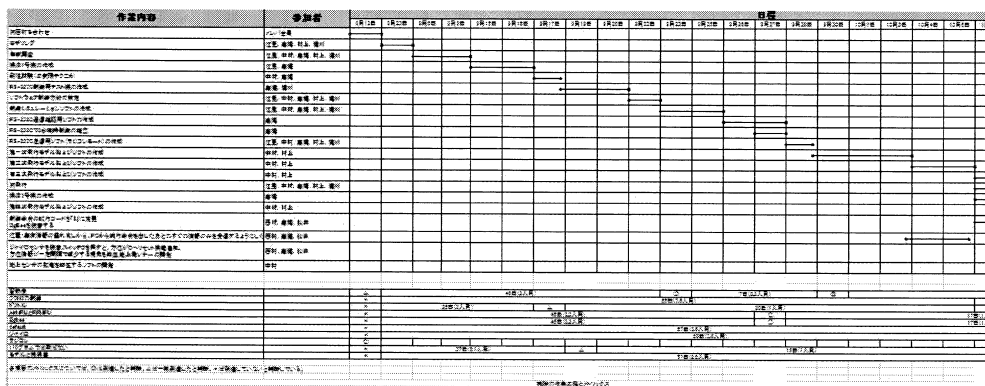


図 4 ガントチャート

### 6.4. ヒューマンリソースマネジメント

今回が初めての参加だが、次の世代を育てることも視野にいたれた人員配置を行った。これによりプロジェクトの持続を可能にした。

個人の自発的な取り組みを促すチーム作りを行い、個人の長所を生かす配置を行いプロジェクトを進めた。

キックオフミーティングを、メンバ全員で参加し意思統一を行った。

## 7. まとめ

ドリトルを用いて、MDD ロボットチャレ

ンジに参加することにより、以下のことを得た。言語習得に時間をかけずにオブジェクト指向開発は可能であるということを実証し、実践した。

ドリトルを使用したことにより全開発工程において以下のようなメリットを得た。

- ・日本語ですぐできる→日本語のモデルから英語コードへの変換がいらぬ。
- ・言語仕様が単純→言語習得が容易なのでモデル化に集中できる。
- ・シミュレーションを行い、開発工数削減の実現および、再現性を高めた。

・モデル化→シミュレーション→実機テストを行い、より効率的に実機テストを行うという繰り返し型開発を実践した。

ガントチャートを利用して、スコープマネジメント、タイムマネジメント、リスクマネジメント、ヒューマンリソースマネジメントなど、プロジェクトマネジメントの能力を得た。

## 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。

## 参考文献

- [1]MDDロボットチャレンジ  
<http://www.ertl.jp/ESS2007/mdd/index.html>
- [2]社団法人情報処理学会,「MDD ロボットチャレンジ 2004 産学連携による組み込みソフトウェア開発の実践」(社団法人情報処理学会, 2005)
- [3]社団法人情報処理学会,「MDD ロボットチャレンジ 2005 産学連携によるモデルベース組み込み開発の実践」(社団法人情報処理学会, 2006)
- [4]MDD チャレンジ参加へのお誘い  
<http://www.ertl.jp/ESS2007/mdd/file/MDDchallenge2007.pdf>
- [5]竹政昭利,「はじめて学ぶUML 第2版」(ナツメ社, 2007)
- [6]青山幹雄・中谷多哉子,「オブジェクト指向に強くなる—ソフトウェア開発の必須技術」(技術評論社, 2003)
- [7]山口淳一・江見圭司,「オブジェクトモデリング(UML)を用いた組み込みソフトウェア開発技術者養成プロジェクト(IPSJSymp.Vol.2004,pp.171-174)
- [8]江見圭司・石井充・矢島彰,「モデリングを中心にしたオブジェクト指向技術者養成カリキュラム」(IPSJ Symp. Vol.2002, No.12,pp.127-132)
- [9]牛尾剛・長瀬嘉秀,「オブジェクト脳のつくり方—Java・UML・EJBをマスターするための究極の基礎講座」(翔泳社, 2003)
- [10]平澤章,「オブジェクト指向でなぜつくるのか—知っておきたいプログラミング, UML, 設計の基礎知識—」(日経BP社, 2004)
- [11]株式会社テクノロジックアート,「UML辞典」(翔泳社, 2004)
- [12]兼宗進,「教育利用を目的としたオブジェクト指向言語の研究」,(筑波大学, 2003)

[http://kanemune.eplang.jp/data/kanemune\\_phd.pdf](http://kanemune.eplang.jp/data/kanemune_phd.pdf)

[13]ドリトルホームページ

<http://dolittle.eplang.jp/>

[14]ドリトル入門

<http://dolittle.eplang.jp/pukiwiki.php?%A5%C9%A5%EA%A5%C8%A5%EB%C6%FE%CC%E7>

[15]一橋大学 計算機概論(2006)

<http://kanemune.eplang.jp/index.php?cs06>

[16]兼宗進(翻訳),「コンピュータを使わない情報教育アンブレグドコンピュータサイエンス」(イーテキスト研究所, 2007)

[17]ロボット工房 スタジオ ミュウ

<http://www.geocities.jp/shuinoue/>

[18]岡村正司,「徹底解説!プロジェクトマネジメント—国際標準を実践で活かす」(日経 BP 社, 2003)

[19]金子則彦,「プロジェクトマネージャ完全教本 2007年版」(日本経済新聞出版社, 2007)

[20]株式会社チェンジビジョン社

JUDE/Community

<http://jude.change-vision.com/jude-web/index.html>

[21]江見圭司,「C言語の歩き方 =Windows 環境における C 言語教育=」,(CIEC 会誌コンピュータ&エデュケーション,vol8,pp.51-56,2000)

[22]ハーバート シルト (著), Herbert Schildt (原著), 山本 信雄 (翻訳)「Windows2000 プログラミング標準講座— Windows95/98/NT 共通の基本的なプログラミング技法と Windows2000 の新機能 (Programmer's SELECTION)」, pp.1-704, (翔泳社, 2000 年)

[23]Charles A.Mirho (著), Andre Terrisse (著), 梅原 系,「Windows95 通信プログラミング」,(アスキー,pp.1-308,1996)

[24]UML モデルをどうやって C 言語に落とし込むか

<http://monoist.atmarkit.co.jp/fembedded/robocon/etrobo06/road03/road03a.html>