社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－ＳＥ－160 (17)
2008－EMB－ 9 (17)
2008／6／20

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

中村州男[†] 高橋修司[†] 江見圭司[†]
村上智史[††] 橋本拓哉[††] 橋川竜起[††]
西村憲二[†††] 高橋嘉也[†††] 竹内勇貴[†††] 松井宏樹[†††] 岡村勝[†††]

「ドリトル」は教育用のプログラミング言語である．しかも，日本語で記述するオブジェクト指向言語でもある．これを使って，プロジェクト工程を大幅に削減することに成功した．そして，組込開発の進捗と学習者の習熟度に応じた多段階開発を行った．本稿は，MDD ロボットチャレンジ 2007 で実践したこれらの報告である．

Practice of Man Power Control and Multi-step Development by Using a Programming Language for Education

KUNIO NAKAMURA[†] SHUJI TAKAHASHI[†] KEIJI EMI[†]
SATOSHI MURAKAMI[††] TAKUYA HASHIMOTO[††]
RYUKI HASHIKAWA[††] KENJI NISHIMURA[†††]
YOSHIYA TAKAHASHI[†††] YUKI TAKEUCHI[†††]
HIROKI MATSUI[†††] MASARU OKAMURA[†††]

There is a programming language for education that is called “Dolittle.” It is also an object-oriented language with Japanese language. We succeeded in effective man power control by using “Dolittle.” The higher developer’s skill level and hardware level were getting, the higher our development steps were getting. So we call this “multi-step development.” In this paper, we are going to report on this practice in MDD (Model-Driven Development) robot challenge 2007.

1. はじめに *

1.1 MDD ロボットチャレンジ 2007 とは

モデル駆動開発(Model-Driven Development:MDD)とは，属人性が排除された変換ルールを用いて繰り返しモデルを変換していくことによってプログラムコードを導出する開発手法である．

MDD ロボットチャレンジは，小型飛行船を自動制御するシステムを MDD に従って開発し，その過程で得られるモデルや実際の飛行競技をコンテスト形式で競うことで，研究促進・産業振興・教育実践の 3 項目を達成することを目的とする．同チャレンジは，情報処理学会組込みシステム研究会の組込みシステムシンポジウムで開催されていて，今回で 4 回目である．

過去 3 回のチャレンジ開催で，モデルや飛行の質は向上しているものの，モデル上で審査員が予測したシステム品質と，飛行によって検証される実際の品質が必ずしも一致していなかった．そこで，今回の MDD ロボットチャレンジ 2007 では，開発工数を含む適切な測定法(メトリック)の継続適用を参加者(チャレンジャ)に義務付け，プロセスやプロダクトの品質の観点からモデル駆動・組込みリアルタイムシステム開発に取り組むことが奨励された．

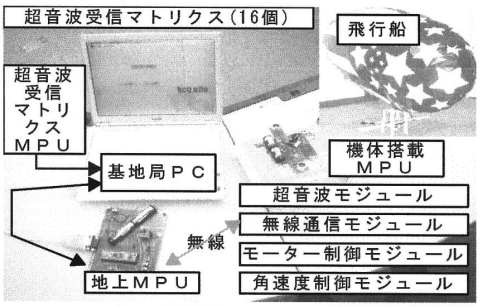

図 1 飛行船システムの概要

チャレンジャが用いるシステムの概要を図 1 に示す．チャレンジャは，基地局 PC と地上 MPU，機体搭載

*[†] 京都情報大学院大学
The Kyoto College of Graduate Studies of Informatics (KCGI)
[††] 専修学校京都コンピュータ学院
Kyoto Computer Gakuin (KCG)
[†††] 株式会社ヒューマン エンジニアリング アンド ロボティックス
Human Engineering & Robotics Co.,LTD. (HERO)

MPU に搭載されるプログラムを開発する.

チャレンジ審査の内容は, 次に示すモデル審査(予想配点は全体の85%), 規定飛行(予想配点は全体の5%), 航法飛行(予想配点は全体の10%)の3項目である.

(1) **モデル審査**

チャレンジャが作成して事前に提出したモデルそのものの内容を審査する.

(2) **規定飛行**

垂直, 水平, 回転, 停止など基礎的な動作について審査を行う.

(3) **航法飛行**

飛行船を出発地から離陸して立ち寄り点(WP)を通過しながら目的地に着陸させることの正確さを競う. 概要を図 2 に示す. また, チャレンジャはスクリーンに基地局の処理状況を投影する.

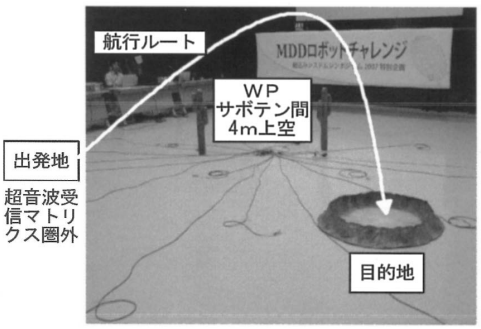

図 2 航行競技の概要

このように, 本チャレンジにおけるシステム構成および課せられる要求は非常に複雑なため, 開発早期における諸問題の整理検討や段階的な洗練/合成. および, ハードウエアへのプログラム搭載と実験の繰り返しを促す組込みモデリング/MDD の適用が重要となる. [1]

1.2 「kcg.edu ヒーロー」チームの形成

MDD ロボットチャレンジ 2007 に「kcg.edu ヒーロー」チームとして筆者らも参加することになった. しかし, 相当遅れてからの参加となった. 図 3 に示すように, 8/24～10/19 までと通常の 35%の期間である.

5月	6月	7月	8月	9月	10月
飛行船の調達		アーキテクチャの利用開始		飛行練習調整	
モデリング学習				変換の試み	
ｼｽﾃﾑ学習と試作		MDD研究・勉強会			

10/15モデル提出
10/18競技練習
10/19競技本番

我々のプロジェクト期間は通常の３５%

図 3 通常のプロジェクト期間との対比

筆者(中村,高橋修司,村上,橋本,橋川:学生)の参加者は, ブログ等の告知で集まった者がほとんどである. また, 以前より産学連携の関係のある組込み開発分野の筆者(西村,高橋嘉也,竹内,松井,岡村:企業)の参加が決まっていた.

8/24 の初会合で, 本チームの顧問である筆者(江見)より, 期間が短いので今回は来年の参加への布石とすることを目的としている. ①モデル駆動開発でなくてもよい, ②飛行船は飛ばなくてもよい, というチームの達成目標が示された. それは, 過去の参加で過酷な経験をしていることからの配慮であったことが後に判明する.

筆者(企業)が飛行船本体・ヘリウム・電池等消耗品提供, 地上 MPU 製作, 機体搭載 MPU 製作ならびに組込みプログラム開発を行うこと, 筆者(学生)が基地局 PC プログラム開発と機体搭載 MPU を飛行船本体に取り付ける筐体の製作を行うことが決まった. この段階では発注・調達ともに全く行われていない.

筆者(中村,高橋修司:大学院生)はプロジェクト推進等, コンピュータ専門学生である筆者(村上,橋川,橋本:専門学校生)はモデル資料作成と基地局 PC プログラム開発と大まかな分担も決まった. 基地局 PC のプログラム開発言語は未定である.

1.3 本研究の目的

筆者(中村,高橋修司)は参加するだけのチームの達成目標では物足りないと感じた. 筆者(企業)はさらに強く感じたと考える. そこで, 飛行船を飛ばすことを目的として, ①短期プロジェクトで難しいチームワークの形成, ②プロジェクト期間の短縮方法, ③チャレンジ開催日程という絶対的な納期を厳守する方法について筆者らと研究することにした.

2. チームワークの形成

組織全体として機能するためには, プロジェクトの目標のように, 共有できるビジョンが必要である. このようにビジョンを共有することによって問題解決や新しい価値の創造に立ち向かうことができる[2].

当然,「飛行船を飛ばそう」が筆者らで共有できるビジョンである. しかし, このような直接的に示すものでは, 短期プロジェクト期間内に醸成しきれず形式的なチームワークに留まってしまう.

そこで, ほとんどの人が知らないメタファー(隠喩)を使って表現した[2]. それが「空飛ぶドリトル」であ

る．ドリトルは一橋大学兼宗進准教授が開発した教育用オブジェクト指向の日本語プログラミング言語である[3]．中学校の授業に使っているところもある．しかし，筆者(江見，中村)以外は知らなかったと思う．筆者(中村)も2007年5月に初めて知った言語である．ドリトルのイメージキャラクタはタートル(海亀)である．実行すると直ぐ画面に現れる．

「空を飛ばないはずのタートルを飛ばそう．そのために基地局PCのプログラミング言語はドリトルである．空飛ぶドリトル！」という旨を筆者(中村)が9/8の第2回目の会合の席上で発言すると，筆者(高橋嘉也)が反応した．面白そうだが不安げな表情であった．これは8/26-28に兼宗准教授に確認済みの筆者(中村)を除く全員の感情だったと思う．

しかし，9/22の第3回の会合で地上MPU，機体搭載MPUの一部(モーター制御・無線通信モジュール)とその組込みプログラム開発を筆者(西村)が行い持参してきた．そして，筆者(竹内，松井，高橋嘉也)がPCと地上MPUを接続し，ドリトルで作ったテストプログラムで，飛行船のプロペラを回したのである．ドリトルで最初に飛行船を動かしたのは筆者(企業)であった．

これを契機に，筆者らは一丸となって困難な課題であるプロジェクトに立ち向かうチームワークの形成ができたと筆者(中村)は考える．翌9/23からは「空飛ぶドリトル掲示板」を筆者(中村)が開設して，筆者(村上，橋本，橋川)のドリトル習得が始まった．

なお，筆者(中村)が唯一不安だった点は，ドリトルがJavaベースのインタープリタ型言語で処理が遅いことである．はじめて9/8の第2回目の会合でリモコン制御の飛行船が飛ぶ様子を見て，コントロールが難しいことを理解した．これにより，過去3回のチャレンジでゴールに到達したのは1チームだけであることも理解できた．いくら精度を上げても困難であり，逆に1秒から2秒単位の制御に精度を下げても支障はない．したがって，ドリトルで問題ないと判断した．

3. プロジェクト期間の短縮方法

3.1 会合の回数の削減

内容が薄く，だらだらした会議は，メンバーのモチベーションを低下させ，プロジェクトの回転を低下させる直接的な要因である[4]．

筆者らは産学連携チームであり，筆者(企業)を休日に拘束することになる．したがって，毎週のように定期的に会うのではなく，進捗に合わせて次回の予定を決めた．その結果，8/24の第1回会合からチャレンジの日までの9週に5回だけの会合となった．

これを可能にしたのは，その間の連絡等をGoogle社の無料インターネットサービスGoogleグループを使ったからである．これは，筆者(高橋修司)の提案とチャレンジ開催事務局や産学の連絡を全て担ってくれたお陰である．

プログラム開発の助言などタイムリーに行わなければならないことはチャットを使い，そうでなければ共有文書にしているプログラムを直接修正し合う．また，同時編集機能はモデル清書時にも大いに役立った．

しかし，会合の回数削減は，単にインターネット利用するだけではなく，筆者(高橋修司)のように伝達すべき情報の取捨選択のできる人がいてはじめて実現できると考える．

3.2 組込みプログラム開発と基地局PCプログラム開発の役割分担による開発工数の削減

組込みソフトウエアの作業計画で留意すべき事項①に「ハードウエア開発プロセスなどとの整合性や連携点の折りこみ」とある[5]．

組込みプログラム開発と基地局PCプログラム開発の役割分担は大いに悩む部分ではあるが，プロジェクト期間の短縮のために，できる限り早い段階で切り分ける必要がある．もちろん，後戻りはしない．9/22の第3回の会合で地上MPUに実装すべきプログラム仕様が決定した．

幸い，地上MPU，機体搭載MPUとその組込みプログラム開発は，組込みのスキルの非常に高い筆者(西村)が中心となってくれた．そのお陰で筆者(学生)がオブジェクト指向言語で手続き型の処理をしたくない旨の一方的なお願いにも応えて頂けた．その結果，基地局PCのRS-232C回りのプログラム処理は単純化することができた．もちろん，筆者(西村)から本来ならば，処理能力もメモリも大量にあるPC側でやるべきとの意見も頂いた．これは角速度制御のやり取りで筆者(中村)と長時間議論したところである．しかし，来年度以降のチャレンジのときに，ドリトルを使わないことも想定され，そのときの労力が軽減されることが望ましいと判断した結果，合意が形成されたのである．

通常は，基地局PCに機体搭載MPUから送信された情報を継続して受け取ることになる．これだと基地局PCプログラム開発言語を変更すると通信制御に関わる工数が常に増えてしまう．そこで図4に示すように，機体搭載MPUは標準のものを使用し，地上MPUの組

込みプログラムを筆者(西村)に変更してもらった．地上MPUは，①基地局PCからのモーター制御コマンドの受信・機体搭載MPUへのデータ変換と送信・その時点で保持している角度・高度情報の基地局PCへの送信，②機体搭載MPUより頻繁に送信される角速度データの受信と累積による角度データ保持・高度データの受信と最新データ保持を行う．

これにより，基地局PCからは，モーター制御のコマンドを送信したときだけに，角度と高さ情報を受信することだけになり，基地局PCのプログラム開発の工数を削減することができる．

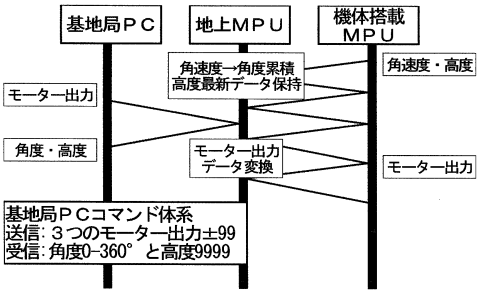

図 4 筆者(中村)制作の基地局PC・地上MPU・機体搭載MPU間のシーケンス図

また，基地局PCのプログラム開発の工数を削減するために，プログラム変更ができない会場備付けの超音波受信マトリクスMPUのデータ変換とファイル出力を行う基地局PC上の制御通信プログラムを筆者(中村)が作った．

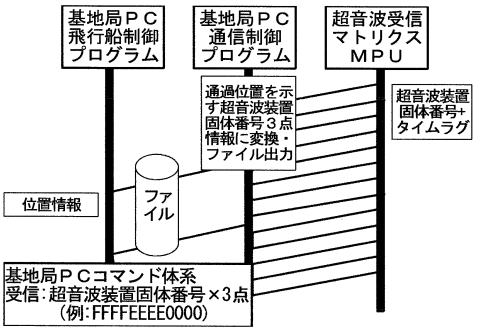

図 5 筆者(中村)制作の基地局PCと超音波マトリクスMPU間のシーケンス図

3.3 ドリトルによるモデル作成工数の削減

ドリトルは日本語で記述できるオブジェクト指向言語である．このため，モデルからプログラムを作成する際に必要な日本語から英語やローマ字表記への統一性のある変換作業が必要ない．それだけではなく，プログラム中にそのクラス概念の日本語表記ができるので，今回のような単一プログラムのシステムでは，モデル表記の清書としてプログラムソース(図 6)からクラス図等のモデル(図 7)を表現することができる．なお，この例題の実行結果は，図 8 に起動時，図 9 にボタンを押した時の画面を示す．したがって，大幅にモデル作成工数の削減ができる．

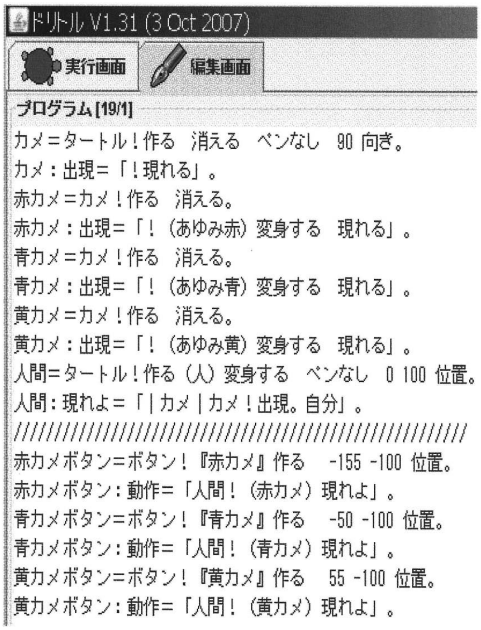

図 6 筆者(中村)制作「人間！カメ現れよ」プログラム

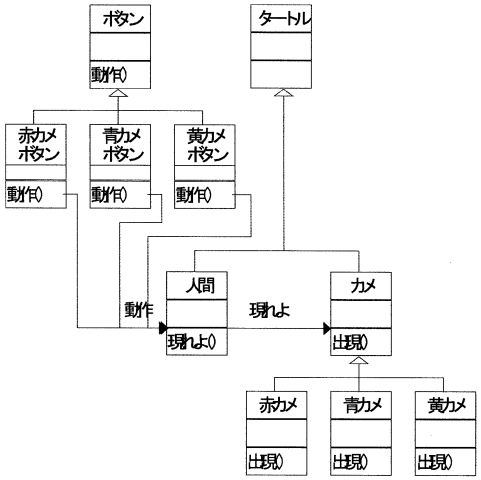

図 7 [6]を参考に筆者(中村)が作成した「人間！カメ現れよ」のクラス図

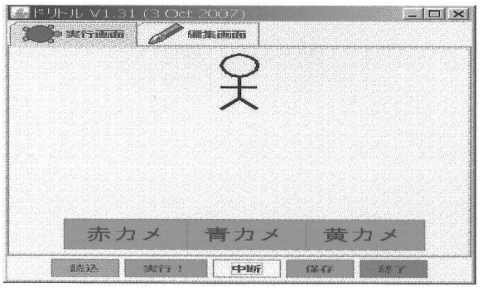

図 8 「人間！カメ現れよ」プログラム起動直後

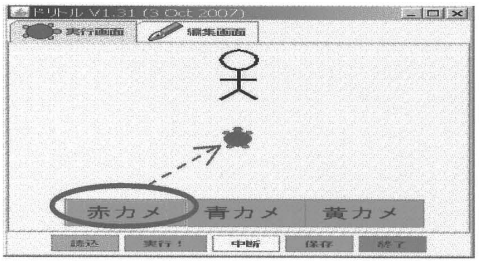

図 9 「赤カメ」ボタンを押した後

3.4 シミュレーションによる工数削減等

組込みソフトウエアの作業計画で留意すべき事項②に「作業上の順序関係などを考慮した作業の並行化などの工夫」とある[6].

筆者らは，通常の35%の期間で，かつ，飛行船本体も含めてハードウエアも整っていない状態からのスタートである．ハードウエアがなくてもソフトウエアの検証をしておく必要がある．このため，チャレンジでは審査項目にないシミュレーションを行うことで，質的にも一定のレベルを確保した作業の並行化による工数削減を筆者(中村)が考えた.

その際，ドリトルはシミュレーション画面の表示処理が簡単であることも重要であった.

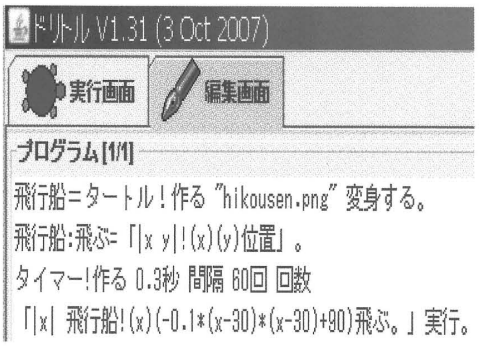

図 10 筆者(中村)制作「単純な飛行モデル」プログラム

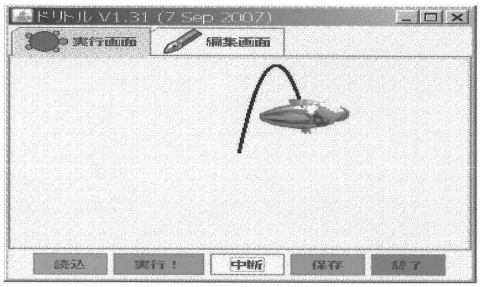

図 11 「単純な飛行モデル」のシミュレーション画面

図 10 は二次関数を使った単純な飛行モデルのわずか4行のドリトルプログラムである．これを実行すると図 11 に示すシミュレーション画面が表示できる．このように簡単にシミュレーション結果を表示できることから，想定されるテストデータを準備すれば，実機を動かすことなく一定レベルの検証ができる．また，実機を動かすと電池の消耗やヘリウムガス注入など経費が掛かるが，これを抑えることも可能となる.

筆者(中村)は，高校教科情報にもあるモデル化とシミュレーションの研究をしている[7]．図 12 は「数に強いこと」と「数学ができること」の違いを図解した畑村洋太郎の文献等[8][9]をもとにモデル化とシミュレーションによるモデルの検証・実体によるモデルの検証を示したものである．シミュレーションによるモデルの検証とフィードバックは大変重要であり，この過程をスキップして実体によるモデルの検証をすることは工数の肥大化に繋がると筆者(中村)は考える.

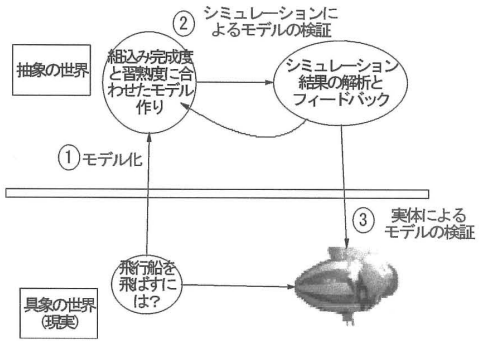

図 12 モデル化とシミュレーション・実体による検証の関係([8][9]をもとに筆者(中村)制作)

3.5 スキル等を意識した役割分担による工数削減

組込みソフトウエアの作業計画で留意すべき事項③に「個々の技術者のスキルなどを意識した作業の分

担・割り振り」とある[6].

筆者らは筆者(企業)は組込み関係，筆者(専門学校生)は基地局 PC 飛行船制御プログラム開発・文書作成と大まかな作業の分担を行っていた．筆者(大学院生)はこれらの活動が迅速に行われるようにプロジェクトを推進する．筆者(中村)はソフトウエア開発環境調整，筆者(高橋修司)はそれ以外の企業等連絡調達調整や筐体製作・文書作成調整である．

ドリトルでの基地局 PC プログラム開発をどのように行うかで作業分担が変わるため，その状況に応じて細かな作業分担を変えていく．つまり，当初に割り当てして固定化せず，臨機応変の人材配置をすることによって工数削減を図ることにした．

なお，リーダーシップを発揮したのは筆者(大学院生)であったが，明確なリーダーは任命されていない．筆者(橋川)は次期リーダーとして育成する方針とした．

最終的には，図 14 に示すとおり基地局 PC プログラム開発は筆者(村上)，文書作成は筆者(江見)の指導[10]の元，筆者(高橋修司,橋本,橋川)が行った．

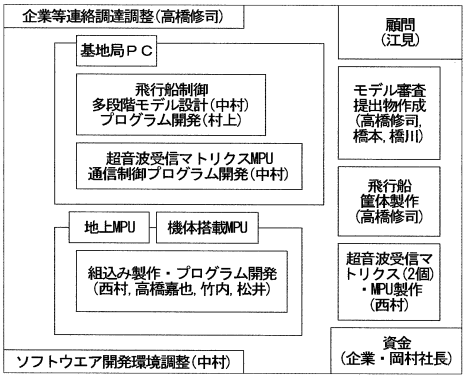

図 14　プロジェクトの役割分担

4.　納期を厳守する方法

4.1　飛行モデルの多段階開発

第 3 回会合の時点ではチャレンジ当日までに筆者(専門学校生)がドリトルでどの程度開発できるのか，組込みがどの程度完成するのかは未知数であった．

そこで，筆者(中村)はスパイラル開発とは異なる多段階開発を提案した．これは，両方のレベルに応じた段階的な飛行モデルを予め設定し，開発するものである．モデル提出期限の 10/15 時点での最良の飛行モデルで審査用書類の作成ならびに飛行することにした．

最終的には，図 13 の GanttProject[11]で作成したガントチャートに示すとおり，全ての環境が整った飛行モデルでチャレンジすることができた．

4.1.1 第 1 次飛行モデル(10/1-3，157 行，3549 文字)

図 13 に示す第 3 回会合直後の 9/23 に「空飛ぶドリトル掲示板」を制作し，図 10 図 11 のシミュレーションレベルのドリトルの習得支援を行った．この結果，

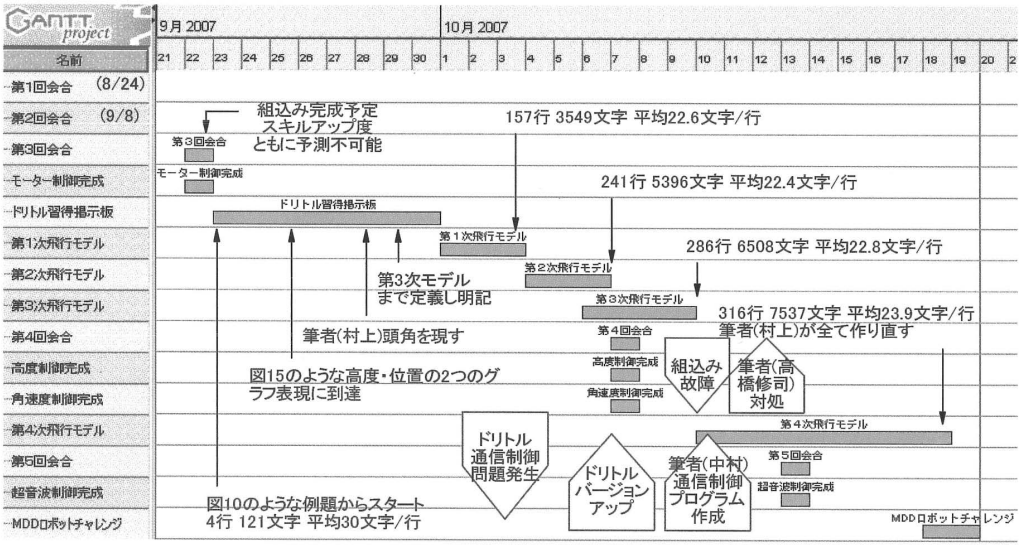

図 13 第 3 回会合以降の組込み完成度と習熟度による飛行モデルの多段階開発の経過

10/1 には第 1 次モデルの原型が出来上がった.

このモデルは高度制御・モーター制御だけである. 図 15 に示すとおり，離陸後，数メートル先の高度２m以上通過ポイントを超える高度に指定回転・指定方向を維持して前進しながら浮上し，水平飛行を数秒続けて，着陸する．なお，自爆タイマーを 60 秒にセットし，左右０・下 90 で指定秒数回し続けるものである.

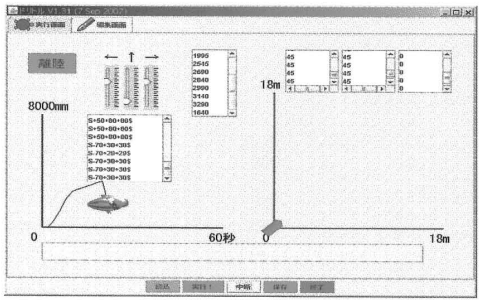

図 15 第 1 次飛行モデルのシミュレーション画面

4.1.2 第２次飛行モデル(10/4-6，241 行，5396 文字)

このモデルは高度制御・モーター制御・初回値のみの超音波センサー制御を行う．図 16 に示すとおり，離陸後，数メートル先の高度２m以上通過ポイントを超える高度に指定回転・指定方向を維持しながら前進浮上し 水平飛行し，地上センサー情報を１つ入手したら，そのセンサー位置ごとにセットされた左右回転時間と水平飛行時間秒飛行し，着陸する．なお，自爆タイマーを 60 秒にセットし，左右０・下 90 で指定秒数回し続けるものである.

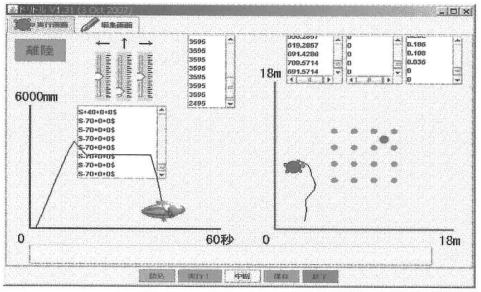

図 16 第 2 次飛行モデルのシミュレーション画面

4.1.3 第３次飛行モデル(10/6-9，286 行，6508 文字)

このモデルは高度制御・モーター制御・超音波センサー制御・角速度制御の完全な制御を行う．図 17 に示すとおり，センサー群情報を１回だけでなく継続して取得できるようになった場合を想定している．その際，図 5 の受信データ中の先頭のセンサー固有番号により，直前のセンサー位置と今回のセンサー位置の方向に船が進行していたと見なして，今回のセンサー位置に対応したゴールへの向き，直進時間で着陸するように軌道を修正し，ゴール到達まで繰返すものである.

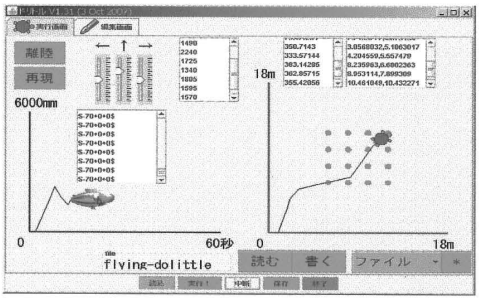

図 17 第 3 次飛行モデルのシミュレーション画面

4.1.4 第４次飛行モデル(10/10-19，316 行，7537 文字)

図 18 に示すこのモデルは，筆者(中村)の段階的な設計指導の結果，筆者(村上)が競技参加用に全面的にソースプログラムを書き直し，チャレンジに使用したものである．想定していなかった飛行モデルであり，筆者(村上)がドリトルを完全に理解したことを意味する.

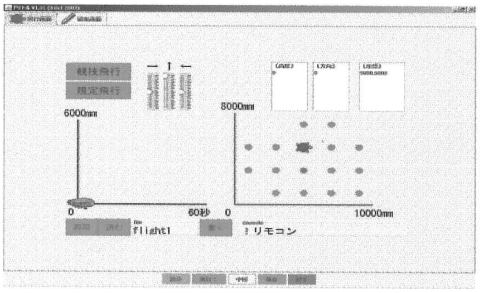

図 18 第 4 次飛行モデルのシミュレーション画面

4.2 代替案の準備

納期を厳守するためには不測事態の発生に備えて予め代替案を事前に準備した.

ドリトルで問題が発生した場合には，作者である兼宗進准教授に変更をお願いする．また，筆者(中村)が補完するプログラム開発をする．筆者(専門学校生)がプログラム開発できなかった場合には，筆者(中村)が開発することにしていた.

組込み関係については，筆者(西村)が中心に行っていた．筆者(岡村)はその企業の経営者であり産学連携に積極的で資金面でも多大な支援を頂いているので，筆者(西村，竹内，松井，高橋嘉也)の作業における諸問題は発生しても企業側で対処できると考えた.

5. 発生した問題と対処

基地局 PC は超音波受信マトリクス MPU と地上 MPU の制御に 2 つの RS-232C ポートを利用する．ドリトルでは同時に利用できなかったので，筆者(中村)の指示で 1 ポートずつ OPEN・CLOSE して対処した．しかし，ドリトルでは非同期で大量に流れてくる COM ポートのバッファ処理ができず筆者(中村)が超音波受信マトリクス MPU を制御する基地局 PC 通信制御プログラムを開発して対処した．また，頻繁に電池交換が必要となる機体搭載 MPU に故障が発生した．このため，筆者(高橋修司)が筆者(岡村)の会社に伺い，筆者(西村)より組込み基板情報の初期化方法や PC からのプログラム更新方法を習得し対処した．

チャレンジ当日の観客等による会場の熱気による気温上昇で規定飛行は離陸できなかった．また対流により，航行飛行は離陸時点で風に流され飛行船浮力が落ち離陸するのが精一杯であった．また，角速度情報がチャレンジ直前から左右反転する事態となっていた．これはチャレンジ終了直後わかったことである．これらの問題は，次回のチャレンジで解決することにした．

6. 評価

審査結果は，100 点満点換算で 36 点(優勝 80 点)で最下位ではなかった．通常のチャレンジ期間の 35%と一致しているが，単純に期間は圧縮できないので，効率よく工数を削減したといえる．また，チャレンジ前日の練習航行では飛行していたので，ドリトルでの秒単位の制御精度は誤りではなかったと考える．

また，来年度のチャレンジはハードウエア・ソフトウエアとも揃っていて，しかも基地局 PC での開発言語を変更しても通信制御が単純化されているので開発も容易であると考える．

人材育成の観点からは，筆者(高橋嘉也,竹内,松井)は意欲的に仕事に取り組むようになり，プロジェクトを任されるなど成長したと筆者(岡村)より報告を受けている．また，課外活動のため，能力のある学生の発掘ができたと考える．筆者(村上)は第 4 次飛行モデルで 2007 日本語プログラミングコンテストのドリトル部門で金賞[12]を受賞している．本学での大学院生と専門学校生との六年一貫教育[13]の成果でもある．

7. おわりに

教育用プログラミング言語のドリトルを用いてプロジェクト工数削減と多段階開発の実践と研究をした．ドリトルを使って基地局 PC 飛行船制御プログラムを開発することで筆者らが一丸となり各自のスキルを最大限発揮して，研究の目的を達成できたと考える．

ドリトルは中学校での情報教育に利用されている．理系文系を問わずソフトウエアの完成を必要とする短期プロジェクトの教育訓練にも活用が期待できる．

謝辞 ドリトルの開発ならびに短期間でのバージョンアップ対応をいただいた一橋大学兼宗進准教授，モデル化とシミュレーションの研究をしている筆者(中村)に実体でのモデル検証という新たな研究の機会を与えていただいた二上貴夫先生(株式会社東陽テクニカ)に，謹んで感謝の意を表する．

参考文献

1) 鷲崎弘宣,太田寛,久保秋真,佐藤啓太,鈴木茂,中村宏明,二上貴夫:MDD ロボットチャレンジ 2007 モデル講評,情報処理学会組込みシステムシンポジウム 2007 論文集,Vol.2007,No.8,pp.266-267(2007)
2) 西之園晴夫,岡本敏雄編著:情報科教育の方法と技術,ミネルヴァ書房(2007)
3)a 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)
3)b ドリトル,URL:http://dolittle.eplang.jp
4) 近藤哲生:はじめてのプロジェクトマネジメント,日経文庫(2005)
5) 独立行政法人情報処理推進機構ソフトウエア・エンジニアリング・センター編:組込みソフトウエア向け開発プロセスガイド,翔泳社(2006)
6) 牛尾剛,長瀬嘉秀:オブジェクト脳のつくり方, 翔泳社(2003)
7) 中村州男,江見圭司:モデル化とシミュレーションのビジュアル教材の開発, 情報処理学会,SSS2007 Vol.2007,No.6,pp.33-38(2007)
8) 江見圭司,南野公彦:グラフ電卓を用いた情報・数学・科学の統合的教育への提案,情報処理学会コンピュータと教育研究会,第 89 回研究会,pp.1-4(2007)
9) 畑村洋太郎:数に強くなる,p.220,岩波書店,東京(2007)
10)a 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,情報処理学会,SSS2004,Vol.2004, pp.171-174(2004)
10)b 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,情報処理学会,SSS2002,Vol.2002,No.12,pp.127-132(2002)
11) GanttProject,URL: http://ganttproject.biz/
12) 2007 日本語プログラミングコンテスト, URL: http://aoi-project.com/event/2007/jp-procon/
13) kcg.edu 六年一貫教育プログラム,URL: http://www.kcg.ac.jp/school_info/speciality/six_year_consistent_curriculum.html