

MDD ロボットチャレンジ 2006 開催報告

二上 貴夫¹ 鷺崎 弘宜² 小林 靖英³ 乾 裕紀⁴ 大槻 博之⁴ 仲久保 正人⁵
久保寺 勇氣⁵ 川縁 幸平⁵ 羽田 千織⁵ 三橋 祐仁⁵ 沼里 京介⁵

¹ 東陽テクニカ ² 国立情報学研究所 ³ アフレル ⁴ 東海大学 ⁵ 岩崎学園情報科学専門学校新横浜校

2006年10月に、小型飛行船を自動制御するシステムをモデル駆動開発に従って開発し、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、組込み開発におけるモデル駆動開発手法や関連するソフトウェア工学技術の研究および教育を促進する MDD ロボットチャレンジ 2006 が開催された。本稿ではチャレンジの内容を紹介し、運営側、審査側、参加側のそれぞれの立場で得られた知見を報告する。

Report on the Model-Driven Development (MDD) Robot Challenge 2006

Takao Futagami¹ Hironori Washizaki² Yasuhide Kobayashi³ Hironori Inui⁴
Hiroyuki Ohtsuki⁴ Masato Nakakubo⁵ Yuuki Kubotera⁵ Kohei Kawaberi⁵
Chiori Hada⁵ Masahito Mitsuhashi⁵ Kyosuke Numari⁵

¹Toyo Corporation ²National Institute of Informatics ³Afrel, Inc. ⁴Tokai University ⁵Iwasaki Gakuen

A robot design contest, called the "MDD Robot Challenge," which involves designing software to automatically control a small airship robot, was held in Tokyo, in October 2006. The challenge was intended to encourage research and education of model-driven development and related technologies in the field of embedded system development. In this paper, we give the results of the challenge from the viewpoint of steering committee, judges, and participants.

1 はじめに

モデル駆動開発 (Model-Driven Development: MDD) 技術は、産業的な意義としても、研究テーマとしても注目をされている。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を指す [1]。MDD の適用により、変換ルールが蓄積された問題領域における生産性の向上や、開発早期における機能的/非機能的特性の高い精度による検証などが期待できる。

情報処理学会ソフトウェア工学研究会および組込みシステム研究会が主催する組込みシステムシンポジウム (Embedded System Symposium: ESS) では、2004 年以来、MDD の学習や研究を目的として MDD ロボットチャレンジを続けてきた。参加者は、産業界の組込みソフトウェア技術者、ソフトウェア工学や制御、システム工学の研究者、そして学生な

どである。この3年間の経緯で顕著な変化点を振り返ってみよう。

- 2004 年: 集団で取り組む研究やソフトウェア開発実験においては、仮定の要求や仕様が利用されるが、概して現実味に乏しい。このような仮定要求を超えた、きわめて具体的な要求から実装用の仕様までを ESS の場で定義することができた。この要求仕様に対して、変換可能な分析モデルの提示がおこなわれた [2]。しかし、通信メカニズムの問題で自動飛行という肝心の目的は達成できなかった。
- 2005 年: 2004 年のメカニズム問題が天井カメラ、高信頼性無線通信などのアーキテクチャによって解決された [3]。並行して研究者からは緻密な UML [5] モデルが示されツールと一貫した手作業による変換の事例が示された。また、実際の航行では、1 チームが自動航行と定点着

陸に成功した。

- 2006年: 研究者, 産業, 専門学校など MDD の取り組みをおこなう層が拡大した。さらに, 制御システムの進化, トランスレーションの試みの拡大, 新規実験などがおこなわれた。実飛行においては, 定点着陸に 2 チームが成功した。実際の飛行の様子を図 1 に示す。

このように MDD ロボットチャレンジの取り組みは, ゆっくりと発展してきている。本稿では以降において, 2006 年における内容と結果を報告する。

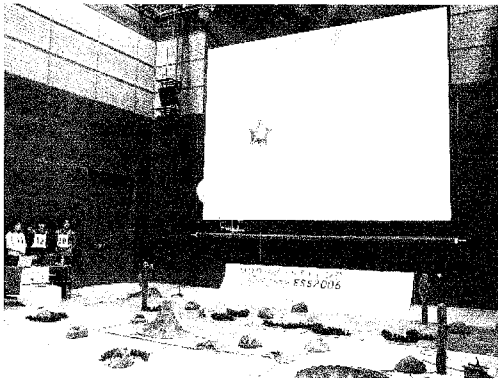


図 1: 競技当日における飛行の様子

2 チャレンジの内容

2006 年の MDD チャレンジは, 8 月~9 月の 2 ヶ月間のシステム開発を経て, 10 月の ESS 併設イベントとして競技が開催された。主要なスケジュールを表 1 に示す。

表 1: スケジュール

日程	事柄
6 月 3 日	UML/組込み開発教育
6 月 17 日	MDD 教育
7 月 29 日	ハードウェア解説
9 月 12 日	耐空検査
10 月 3 日	モデル提出
10 月 11 日	モデル審査
10 月 19 日	競技

以降において, MDD チャレンジの運営の観点で概説する。MDD チャレンジは, モデルの審査と飛行競技の 2 本立てでコンテストを主軸に現在まで進めてきた。

飛行競技の主眼は, 模型飛行船をあらかじめ定められた経路に沿ってソフトウェアのみで飛行できるか否かを競うことにある。これは, 実機でいうところの航法にあたる。模型の He 飛行船にマイクロプロセッサを搭載, 地上にある制御基地との間で位置や高度の情報をやりとり, 分散処理している。航法競技の概要を図 2 に示す。参加者 (チャレンジャ) には, 飛行船を出発地から目的地まで, 途中のバルーン (ウェイポイント) の近傍を経由して一定時間内に到着させるように制御することが求められる。目的地周辺には超音波受信アレイが配備され, 位置の把握に用いられる。

ただし, 航法だけでは難易度が高いので, 規定と呼ぶ単純な自動飛行も競技種目にしている。規定では, 制御結果 (位置や経路などの精度) よりも, ソフトウェアが制御動作を適切に実行しているかを評価する。規定競技の概要を図 3 に示す。規定競技では, 決められた地点からの上昇, 一定高度に達した後の回転, 前進, 下降, 着地後の停止という一連の動作の適切さが競われる。

これらの航法競技と規定競技をそれぞれ, 審判が観察し複数の評価点項目に沿って採点する。同時に, 「飛行に偶然成功した」場合と「的確な制御により成功した」場合を区別するため, チャレンジャには, 制御システムにおいて把握している飛行船の状態 (位置や方位など) および指令状況 (送受信コマンドなど) を航法制御 PC 画面 (スクリーン) 上にリアルタイムで可視化することが求められた。可視化の様子を図 4 に示す。

また, 自動飛行プログラムを開発する方法として MDD を利用し, その基点であるドメインモデルの適切さ, MDD の程度や質を評価しモデル審査とする。モデルの表現を UML に限定するような規則は設けないが, ISO19501 となっている UML 標準の考えかたとモデル概念レイヤに対応する形でモデルを分類している。

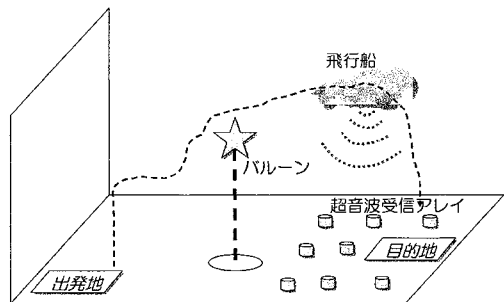


図 2: 航法競技の概要

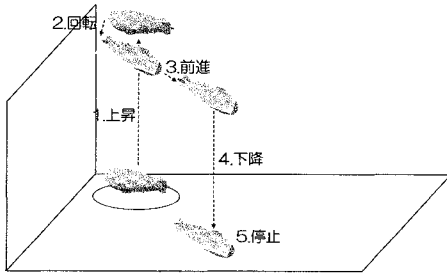


図 3: 規定競技の概要

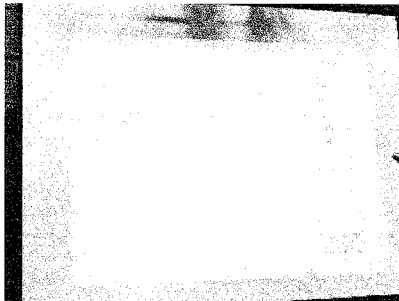


図 4: 競技当日における可視化の様子

ただし、コンテストのみでは、枠組みに当てはまらない技術の試行や研究の促進に支障がでる。このため、規定外の研究開発については、エキシビションという枠を設けて対応している。また、最近はどうような開発手法であろうとも組込み製品の信頼性や安全性を確保する作業が一般化してきている。このような問題を考えるきっかけとして耐空性審査という事前のシステム審査をコンテストの1ヶ月前に実施することとした。2006 年では一貫した審査は主催者側の準備不足からできなかったが、審査の存在そのものが、チャレンジに問題の存在を知らせることに役立った。

以降において、チャレンジにおけるハードウェアとソフトウェアをそれぞれ詳説する。

2.1 ハードウェア: 分散・組込みシステム

チャレンジには、ハードウェアを含めて開発するケース（システムチャレンジ）と主催者の提供するハードウェア環境に自分たちのパソコンを接続してパソコン上のソフトウェア開発に注力するケース（ソフトウェアチャレンジ）の2通りのチャレンジ方法がある。どちらの方法であっても、基本的な制御のフィードバックループは類似である。

図 5 に基本構成を示す。飛行船に搭載された AirMPU は、超音波ソナーと角速度センサーを使っ

て機体の高度と方位を得る。そのデータは、無線リンクで地上の航法制御パソコン（GR-PC）に送られる。位置検出プロセッサ（US-MPU）は、飛行船の発したソナー信号を地上に展開した超音波アレイでキャッチする。航法制御 PC は、US-MPU の信号から現在位置を計算し、次に必要な飛行船のプロペラ推力を計算して推進指令を生成する。指令は、無線リンクで AirMPU に指令され、AirMPU が受信に成功するとモータ駆動のプロペラ推力が更新され飛行船は上昇、下降、旋回などの動作を行なうことになる。

2006 年の主催者ハードウェアは、2005 年のそれよりも 80g あまり軽量化が進み、ヘリウムガスの消費量が減った。

また、地上超音波レーザの配備も見直した。無線通信の方式は、主催者システムでは 315MHz のオンオフ変調方式であった。システムチャレンジは、Bluetooth を利用したり、ZigBee モジュールを利用して通信の高速化やモジュール化を進めていた。

2.2 ソフトウェア: MDD による開発

上述のように、チャレンジにおけるプラットフォーム/環境/要求は非常に複雑であるため、組み込むソフトウェアの開発にあたり、早期の問題整理検討や段階的な洗練が必須である。さらには、最終的に必要なソフトウェアは一定以上の規模を有することとなるため、集団で意思疎通して計画・管理するために中間の成果物をモデルとして表していく必要がある。

そこでチャレンジでは、飛行船制御ソフトウェアの開発にあたって、モデル駆動アーキテクチャ（Model-Driven Architecture: MDA[4]）に代表される MDD を適用することを必須とした。MDA では、ドメインや要求を表す計算処理独立モデル（Computation Independent Model: CIM）を作成し、ツール等によって CIM を変換してプラットフォーム独立モデル（Platform Independent Model: PIM）を得た後に、ツール等によって PIM をプラットフォーム依存モデル（Platform Specific Model: PSM）に変換し、さらにその変換を経てプログラムコードを得る。用いるプラットフォームの情報を異なるモデル（プラットフォームモデル、Platform Model: PM）として記述し、PIM から PSM への変換時に用いることもある。MDA/MDD の適用により、異なる関心事の早期の分離と個別の検討、および、プログラムではなくモデルを中心とした意思疎通と段階的开发が可能となる。

チャレンジでは、MDA/MDD の適切な適用をチャ

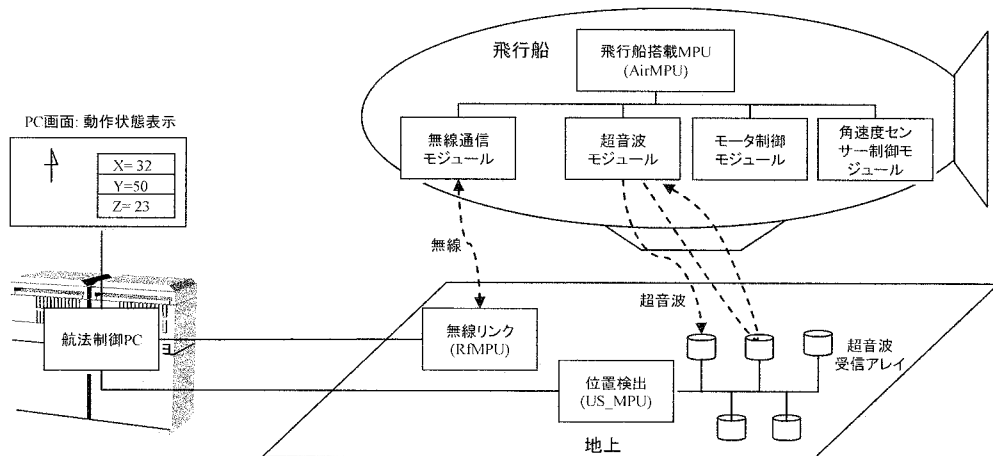


図 5: MDD ロボットチャレンジにおける電子システムの全体構成

レンジャに対して奨励するために、航法競技・規定競技の事前に以下の満足するモデルの提出を義務付けた。

- 競技規約に基づき飛行船システムによって競技を達成する方法、および、システムを構成するハードウェア制御の方法を表す
- CIM, PIM, PSM がそれぞれ明示されている
- モデル間の変換方法が明示されている
- システムの機能、構造、振る舞いが明確である
- ただし、モデルの表記法は自由とする。例えば UML, 状態遷移表, 図を補足する文章などを用いることができる。

審査員は、チャレンジャからの提出モデルが上記の規定を満足することを前提に、主に以下の観点で審査した。

- 競技規約に基づき飛行船システムによって競技を達成すること、および、システムの構成ハードウェアを適切に制御することを実現するために妥当に作成されていること。
- 簡潔で読みやすく意図が明確であること。
- その他の特徴や、規定された内容以外の記述がある場合、次につながる有効な特徴/記述と判断可能な場合は、加點評価する: 競技の達成やハードウェアの適切な制御, システムの効率の良い開発や保守, 実開発への展開

3 チャレンジの結果

チャレンジには、表 2 に示す計 11 チームの参加があった（ただしチーム T_{11} はエキシビジョン参加）。以降において、各チームの競技結果とモデル審査結果を示し、続いて、両結果を踏まえたチャレンジ結果への考察を述べる。

表 2: 参加チーム一覧 (* はエキシビジョン参加)

ID	チーム名	種別	所属
T_1	ムソムン考房	学, 産	金沢工業大学, 北陸日本電気ソフトウェア株式会社
T_2	新潮流	産	ニューウェイブシステムズ有限公司
T_3	電大非行船	学	東京電機大学
T_4	WEST	学	和歌山大学
T_5	MONTBLANC	学	東海大学 湘南
T_6	KAIT-NEXT	学	神奈川工科大学
T_7	アルゴノーツ	産	株式会社アルゴ 21
T_8	新横 AirShip	学	情報科学専門学校新横浜校
T_9	ふわっと	学	武蔵工業大学, 東海大学静岡
T_{10}	Tigermoth	独	(独) 宇宙航空研究開発機構
T_{11}^*	ねこねこ専 FU	学, 産	専修大学, 富士通株式会社

3.1 競技結果

競技当日における 10 チームの航法競技と規定競技それぞれの結果を表 3 に示す。各競技の結果は、構成動作の単位で難度に応じて得点化されて（例えば離陸 1 点, 着陸停止 2 点など）、最終的に合計された。

過去 2 年間で比較してチャレンジャの制御の精度は向上しており、特に上位 3 チーム (T_9, T_4, T_3) に

については、上昇に始まって目的地付近への着地と停止に成功した。これらのチームはスクリーン上へ飛行船制御の状態を適切に可視化しており、偶然ではなく、確かなソフトウェア自律制御に成功していることを確認した。

表 3: チーム単位による飛行競技の得点

ID	規定競技	航法競技	合計	飛行順位
T_1	6	3	9	4
T_2	4	3	7	7
T_3	8	6	14	3
T_4	6	9	15	2
T_5	0	1	1	9
T_6	0	0	0	10
T_7	4	3	7	7
T_8	6	3	9	4
T_9	8	10	18	1
T_{10}	6	3	9	4

3.2 モデル審査結果

チャレンジャから提出されたモデルは、競技とは独立して事前に審査員によって審査された。審査のプロセスを以下に示す。

(1) 審査シートの作成と修正: 審査員 1 名が、客観性と納得感のある審査を実施するため事前に、様式と品質から構成される審査シートを作成した。様式について以下の項目を挙げた。

- 種類: 適切な CIM, PIM, PSM の有無
- 変換方法: 各モデル間の適切な変換方法の有無
- 記述側面: 機能, 構造, 振る舞いの有無

また多様な品質特性について網羅的に評価するために、事前公開された審査基準を逸脱しない範囲で ISO9126-1 品質モデルに基づいて以下の項目を挙げた。

- 機能性: システムが競技達成/適切なハード制御のために妥当に構築されているか。モデル変換が妥当で動くシステムを導出できているか。
- 信頼性: 位置ロスト時や通信エラー時の対処はあるか。「無風」とはいえ実際には微風があるはずだがその対処はあるか。
- 効率性: 一定時間内に到着するように設計されているか。飛行船の小メモリに納まるように設計されているか。
- 保守性: 読みやすいか。仕様変化に対応できるか。テストできるか。

- 移植性: 他のプラットフォームに移植できるか。

また、審査団の全 5 名の審査員が集団で審査シートをレビューおよび修正した。

(2) 1 チームを合同審査しキャリブレーション: 審査シートによって審査の観点のブレは排除されたが、依然として各観点における良し悪しの程度にブレが生じる可能性があった。そこで、10 チームより 1 チームをサンプルとして選択し、同チームのモデルを審査員全員で議論しつつ審査し、付与する得点のキャリブレーション(較正)を実施した。

(3) 各自が残り全チームを審査し集計: 続いて審査員それぞれが残りの全チームを審査し、得点を集計した。

(4) 審査ポイントの最終決定: さらに、提出モデルに特筆すべき事柄がある場合は加点することとし、最終的に表 4 の結果を得た(上位 2 チームのみ抜粋)。加点事項として、開発プロセスの積極的な修整、独自 MDD ツールの開発と活用、独自シミュレータ開発と活用、プロダクトラインへの取り組みがそれぞれ見られた。審査結果を受けて得点の高い順に、チーム T_7 へ最優秀モデル賞、 T_{10} へ MDD 奨励賞、 T_1, T_3 へエンジニアリング奨励賞がそれぞれ授与された。

表 4 において、品質得点の高いチームが上位にランクインしており、それらの上位チームは同時に様式についても網羅していた。従って、審査員は様式について網羅されていたために、それぞれ高い品質特性を確認することができたと考えられる。また、品質特定の単位では全般的に機能性、信頼性、保守性の高さを定性的に確認できたが、効率性および移植性はモデルから把握困難であった。

例として最優秀モデル賞を受賞した T_7 について、PIM モデル内の構造、PIM から PSM への変換規則の一部、および変換の例示をそれぞれ図 6、図 7、図 8 に示す。図 6 において、システム化にあたりプラットフォームに依存しない形で、飛行船を中心とした本質的な要素と要素間の関係を分析できていることが分かる。また、図 7 においてモデル要素をテンプレートした形で利用プラットフォームに特化させる変換規則を定義し、(機械化されてはいないものの) 属人性を排した変換に成功していることが分かる。

3.3 モデルと飛行の関係

飛行競技とモデル審査の両結果の関係を分析した。運営側・審査員は、問題を早期に分離し整理検討することを促す MDD の適用により、モデルの評価が良いチームの飛行船ほど、適切に制御され良い飛行

表 4: チーム単位によるモデル審査得点 (上位2チームのみ抜粋)

ID	モデル			～への変換方法			側面			品質特性					加点	合計	モデル順位
	C	PI	PS	PI	PS	コード	機	構	振	機	信	効	保	移			
T ₇	2	2	2	2	2	1	2	2	2	17	14	11	18	15	5	97	1
T ₁₀	2	2	2	0	0	2	2	2	2	18	14	12	15	14	5	92	2

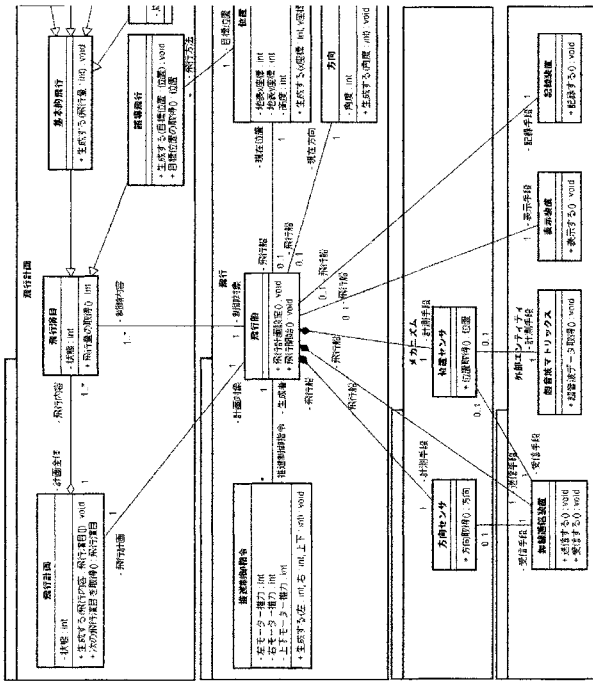


図 6: T₇: PIM 内の構造クラス図 (抜粋)

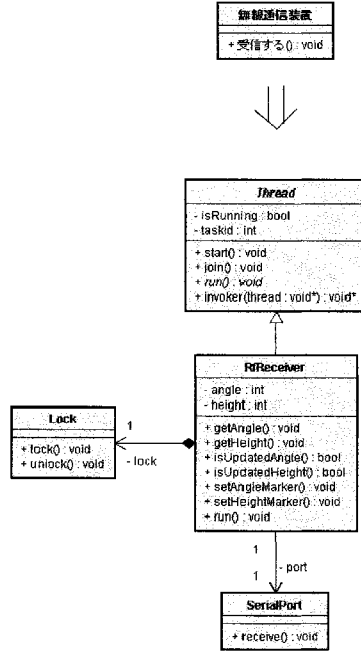


図 8: T₇: PIM から PSM への変換例

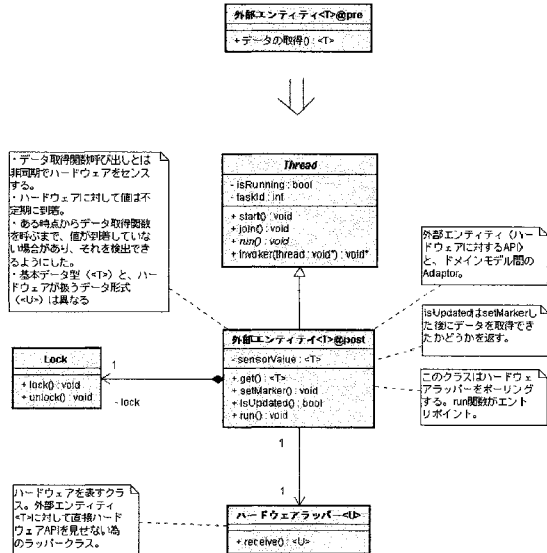


図 7: T₇: PIM から PSM への変換規則定義の一種

結果をもたらすことを期待した。両結果を合わせたグラフを図9に示す。両結果の相関係数は0.14となり、全10チームの結果を用いる場合にモデル評価と飛行結果は強い関係にないことが分かった。

ただし図9を詳しく分析すると、飛行結果が著しく低い2チームを外れ値として除外すれば、他のチームを概ね、モデルは中程度で飛行が良いA群、モデルも飛行も良いB群、モデルが良いが飛行が中程度のC群に分類できる。A群は、期待に反するものであり、モデリングよりも注力せずに実機テストや実装の修正に注力することで外界やプラットフォームの特性を的確に扱ったことに起因すると考えられる。C群も期待に反するものであり、反対にモデリングに注力したものの外界やプラットフォームの特性をモデル上での確に操作・検証できなかったことに起因すると考えられる。同様の結果は、組込みシステムに関する類似コンテストにおいても得られている[7]。

B群は期待に合致するものであり、実際に該当チームT₃は、PIMの段階における物理モデル検討

への注力と、シミュレータによる実装段階での制御処理の検証に注力していた。このようなモデル上での関心事の分離検討および実装段階における繰り返しの振る舞い検証により、モデルと飛行の両方についてよい結果をもたらしたと考えられる。今後は、振る舞いの検証を PIM や PSM の段階で実施して、同様の結果を得る取り組みが期待される。

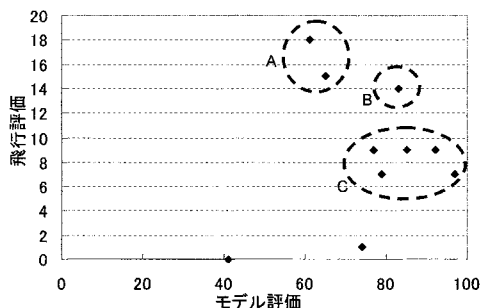


図 9: モデル評価と飛行評価の関係

4 個々のチャレンジ成果

各チャレンジの立場から成果・知見を述べる。

4.1 T_5 : MONTBLANC

MDD ロボットチャレンジに参加したことで、我々は非常に多くのことを学ぶことが出来た。ここではチーム MONTBLANC のチャレンジ成果として、学んだことや苦労したこと、そして次回に活かしたいことをまとめる。

学んだことの中で最も大きかったことは、多人数による共同開発の実践が行えたことである。例えば、タイムスケジュール等のマネジメント、分割作業を行なうための仕様の決定、他のメンバとのコミュニケーションの必要性などを学ぶことができた。このようなことは、我々学生にはなかなか学ぶことが出来ないようなことである。この経験は、この先も必ず役に立つであろう。

他に学んだこととしては、ハードウェアおよびソフトウェア両方の開発を学ぶことが出来た事、そして MDD による開発に挑戦できたことである。前者においては、これからの組込み分野での開発には欠かせない経験が出来たように思う。また、後者においては、組込み分野に限らず、開発のプロジェクトなどで役立つような経験を得られたように思う。

苦労した中でもっとも大変であったことは、実際の物理現象を常に考えなければならなかったことだった。例えば、飛行船を前進させるには左右のモータを共に逆回転させる必要が有る。システムからこの命令

を実行に移すのは、プログラムを書くことも含めてそう難しいことではなかった。しかし、左右のモータが正しく逆回転をしても、飛行船は当然のように真直ぐには前進してくれなかった。プロペラの向きの微妙な差異、会場の温度、人には分からないような空気の流れや電源電圧の微妙な低下等々、実に様々な要因で飛行船は真直ぐに前進したり蛇行をしたりしてしまっ。また、我々の飛行船は、エンベロープと Gondola 部分を複数箇所固定するようにして、プロペラの位置を微調整できる作りになっていた。そのため、エンベロープと Gondola 部分は完全に取り付けていなかったため、データを集めて完璧に直進するように出力を調整したとしても、翌日にはそのデータが役に立たないということがあった。さらに、エンベロープ内のヘリウムは、時間が経つにつれて減っていつてしまうため、その時々でプロペラの位置を微調整し、出来る限り真直ぐ前進するようにするしかなかった。

また、モデルの面でも苦戦を強いられた。開発の開始時に MDD を行ったことがあるメンバが 1 人であったため、まずはその習熟を行い、試行錯誤をしながら進めていくしかなかった。試行錯誤をしながら進めていく中で、特に苦戦したのが飛行戦略や機能等のハードウェアに直接影響して来る部分で、モデルからハードウェア構成を考えた結果エンベロープ 1 基では浮きそうもなくなってしまった。そこで十分な浮力を得るためにエンベロープを横に二連として開発を進めていった。しかしながら、ハードウェア設計に余裕が出来た事とひきかえに、航行時の機体制御が格段に難しくなってしまった。

コンテストにおける内容は芳しくない結果となってしまったが、見学することができた他のチームの競技や競技後の感想会などで、いくつか試してみたようなアイデアを得ることが出来た。これらのアイデアは次回参加時に活かして行きたい。

4.2 T_8 : 新横 AirShip

(1) プロジェクトの概要: 情報科学専門学校新横浜校 3 年制情報工学科 3 年次の卒業研究として取り組んだ。学生 5 名指導教官 1 名のチームである。4 月からスタートし 10 月のコンテストをはさんで 3 月の最終発表会まで 1 年間のプロジェクトである。長丁場なのでモチベーションの起伏をコントロールするのが一番のポイントであった。方向付けと意義付け、スケジューリングや学外での実験機会の確保調整などは指導教官が行い、モデリングから設計・実装・コンテスト後の改善改良までは全て学生が行った。

(2) 動機: パソコンの中のバーチャルなモノではな

く、パソコンの外のリアルなモノを動かす「組込み」と、モデリングからアプローチをする「MDD」に好奇心とチャレンジ精神をくすぐられた。

(3) 前提知識: 組み込みはまったくはじめてで、OAKS16 のサンプルプログラムではじめて組込み系独特のプログラムに出会った。プラットフォームとしての Linux は、サーバ構築演習で一通り触ったが Cygwin は聞くも触るもはじめてで、開発環境の gcc は未経験。シリアル通信はまったく初めてであり、C 言語、C++ 言語でのシリアル通信では苦勞し、ものにできなかった。VB で COM コントロールを使ってようやく通信部分を自由にできるようになった。プログラム言語は、C 言語、VB、Java は一通り経験済み。C++ はやっていない。VB ではちょっとした作品などを制作した経験はある。UML は一通りやってあり、図の意味はつかめる。

(4) 成果: モデリング部門では、怪しいながらも CIM、PIM、PSM へ段階的にモデルをすすめた。例として PIM 内の正常・異常処理の検討結果を図 10 に示す。PSM の段階で C++ から VB に切り替えることになったが、モデルを通してプログラムのイメージはできていたので、見通しよくコーディングできた。また、Cygwin と C++ を使った通信プログラムから取り掛かったが結局は撤退し、Windows 上の VB.NET2005 でオブジェクト指向プログラミングに切り替えた。基地局のプラットフォームが Windows であったことが幸いした。成果物の規模は、コンテスト時点で 10 クラス、全 2883 ステップであった。さらに飛行競技部門では、規定競技では何とか飛行できたが、航法競技では操縦不能になってしまい総合 4 位。VB での処理が重かったこと、飛行戦略に弱点があった。コンテスト後に VB の重さは分散処理で速さを 1/60 に改善した。また当初の飛行戦略では軽視していたジャイロの方位情報を位置情報として常に参照するように改善した。

(5) 難しかったこと: 組込み分野のモデリング [6] にあたってハードウェア、通信の仕組み、ドメイン知識などが重要であることを痛感した。通信制御系のドメイン知識がないので、モデリングのドメイン分析でつまづいたが、自分達のイメージできるオブジェクトでモデルを考え直した。現象から入り現実のデータを直視することにし「現象=>実体=>本質」を意識した。また、モデリングツールは Java ベースがほとんどであり、多少 C++ ベースのものがあり VB ベースのものは手に入るフリーのソフトはなかった。結局 Java ベースの Jude/Community で表現したモデルを VB.NET2005 でコーディングした。さらに、シミュレータと実機との動作速度の違いにはいつも絶望感を味わい実機での検証環境が欲しかった。ただし実機に頼りすぎるとモデルが弱

くなってしまうのではないかとも思え、今回はこれでよかったと思う。

注用機種の並列処理 (ハード)			並列制御 (制御)				
非決定	変異	非実性	セクタの互換	重要・方法 保持	位置情報	考えられる原因	対応
○	○	○	○	○	○	正常	何もない
○	○	×	○	○	×	JS センサーが異常	情報を捨てて再実行
			○	○	×	JS センサーが正常範囲外	大きな向きを以て JS センサーの上に乗す
○	×	○	○	×	○	正常範囲内	センサーが正常
×	○	○	×	○	○	正常範囲内	センサーの異常
○	×	×	○	×	×	センサーが故障	センサーの異常
×	○	×	×	○	×	センサーが正常	情報を捨てて再実行
×	×	○	×	×	○	センサーが正常	情報を捨てて再実行
×	×	×	×	×	×	故障	何もない

○ 正常 X 異常

図 10: T₈: PIM での正常系・非正常系の検討結果

5 おわりに

MDD2006 では、2004,2005 と 2 年続けた別冊集 [2, 3] の発行をおこなう代わりに、本稿のような形で知見を残すことにした。協力いただいている主催メンバー諸氏の負担を減らしつつ、貴重な情報を広く伝える手法として適切だと思う。

また、今年からは航空宇宙学会との協力が検討されている。航空宇宙学会でも最近になって飛行船の学生コンテストを実施し始めており、一部の制御は PE (プログラムドエレクトロニクス) のレベルに達している。我々は航法や空力に素人であり、彼らはソフトウェア開発に素人である。2 つの組織がうまく協調できれば素晴らしい活動に発展することが期待される。

参考文献

- [1] Bran Selic. The Pragmatics of Model-Driven Development, IEEE Software, Vol.20, No.5, pp.19-25, 2003.
- [2] 二上貴夫, 渡辺晴美編: MDD ロボットチャレンジ 2004 産学連携による組込みソフトウェア開発の実践, 情報処理学会, 2005.
- [3] 二上貴夫 編: 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込み開発の実践, 情報処理学会, 2006.
- [4] Object Management Group (OMG), MDA Guide Version 1.0.1, 2003, <http://www.omg.org/docs/omg/03-06-01.pdf>
- [5] Object Management Group (OMG), Unified Modeling Language (UML), Version 2.1.1, 2007, <http://www.omg.org/technology/documents/formal/uml.htm>
- [6] 渡辺博之, 渡辺政彦, 堀松和人, 渡守武和記: 組込み UML - eUML によるオブジェクト指向組込みシステム開発, 翔泳社, 2002.
- [7] Hironori Washizaki, et al.: Quality Evaluation of Embedded Software in Robot Software Design Contest, Progress in Informatics, Vol.4, 2007.