

ロケット電子制御システムへのモデルベース開発手法の適用

Applying Model-Based Development (MBD) to Rocket Electronic Control Systems

下 浦 美 那	株式会社 IHI エアロスペース	基盤技術部電子技術室
辻 本 圭 史	株式会社 IHI エアロスペース	基盤技術部電子技術室
森 田 康 志	株式会社 IHI エアロスペース	基盤技術部電子技術室 主査
大 依 仁	株式会社 IHI エアロスペース	基盤技術部電子技術室 室長 博士 (工学)

近年、モデルベース開発の適用は組込みシステムに対して広がりつつある。モデルベース開発は、上流工程でのシステムレベルの不具合検出や、要求トレーサビリティ確保の容易さなどの点において、開発をより効率化できることが期待される。本研究ではロケット電子制御システムのソフトウェアに対して開発をより効率的、効果的に進めるためのモデルベース開発の実用化について試行した。制御設計の開発で課題である制御モデルでの検証 (MILS) に対し、実機を用いた制御設計の検証 (HILS) との組合せ工程の時間短縮の効果測定によって、従来と比べて後戻りの少ない設計が可能であるモデルベース開発の有用性が確認された。次に、制御モデルから自動生成するコードは、手動によって実装したコードの性能上の有意差および、ハードウェア性能への影響が小さいことが確認された。そのため手動によるコードと同レベルの品質を確保しながら開発の短縮化などに貢献することが確認できた。本稿では、これらの評価結果を示しながら、株式会社 IHI エアロスペースにおけるモデルベース開発に対する取組みと、今後の課題について述べる。

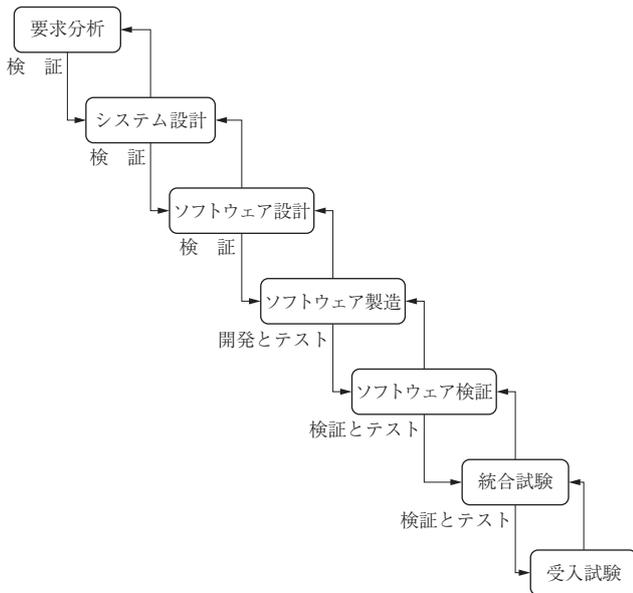
The use of Model-Based Development (MBD) for embedded systems is growing in recent years. MBD has the potential to make development processes more efficient because it makes detecting system level errors in upstream phases and ensuring that traceability requirements are met much easier. We have been exploring MBD to make the development process of electronic control systems for rockets more efficient. We first researched the possibility of applying MBD to rocket electromechanical actuator control software. From this research, we found that it is possible to reduce man-hours with a combination of Model-In-the-Loop Simulation (MILS, verification of control logic by using a plant model) and Hardware-In-the-Loop Simulation (HILS, verification of control logic by using an actual system). We then tried to use auto-code generation for electromechanical actuator control software. The difference in performance between man-made and auto-generated code was small, so automatic code generation is useful to reduce coding time and make coding more effective. In this paper, we describe our approach to MBD in detail.

1. 緒 言

今日、組込みシステム分野においてソフトウェアのシステムに占める役割は増大し、大規模、複雑化している⁽¹⁾。ロケットの領域でも同様にソフトウェアの役割は大きくなっており、加えてロケットの自律点検システムなど、ソフトウェアがこれまでになかった新たな付加価値を生み出してきている⁽²⁾。一方で、大規模、複雑化したシステムは、その全貌を把握しながら開発することに多くの時間と人員が必要になっている。このようなシステム開発の課題に対応するため、モデルベース開発の適用は組込みシステム分野に対して広がりつつある。

一般的な従来型のシステムやソフトウェアの開発の際、ウォータフォール方式に基づいた文書ベースの開発手法をとっていた。ウォータフォール方式とは、① 要求分析

② システム設計 ③ ソフトウェア設計 ④ ソフトウェア製造 ⑤ ソフトウェア検証 ⑥ 統合試験 ⑦ 受入試験、などそれぞれの工程にそのアウトプットを審査し、次工程へ進むという開発手法である。第 1 図にウォータフォール方式の概念図を示す。つまり、システム全体内でのソフトウェアの振る舞いを検証するタイミングが必然的に後工程になってしまうため、システムレベルの問題は、統合試験や受入試験まで工程が進まないため検出が難しかった⁽³⁾。しかし、統合試験や受入試験の段階で検出された問題は再設計-ソースコード修正-ソフトウェア検証と、追加のサイクルが必要になることが多く、その結果大きな後戻りとなってしまう開発効率を下げていた。さらに、株式会社 IHI エアロスペース (IA) の文書ベースによる開発では、設計要求、設計結果、試験仕様書などのトレーサビリティの確保を手動で行っていたため、特に大規模なシステムではトレーサビリティの維持・管理が困難になり、要求実装漏れや工数増大の原因になっていた。



第 1 図 ウォータフォール方式概念図
Fig. 1 Waterfall Model

これらの問題を解決する方法の一つにモデルベース開発がある。モデルベース開発とは、対象となるシステムをモデルによって記述することで、紙では記述できない動的な振る舞いを明確化し、モデルをよりどころとして開発を進めていくという手法のことである。その特徴として、実機製造前の振る舞いの動的検証、モデルからのコード自動生成、トレーサビリティの自動的保持が可能であることから、複雑なシステムの開発におけるシステムレベルの問題の早期発見や、コーディング時間の短縮、トレーサビリティ確保の問題などに対応することができる。モデルベー

ス開発は、近年すでに海外を中心とした航空宇宙分野の組込みシステムにも採用され、一定の効果がでていたため注目を集めている⁽⁴⁾。

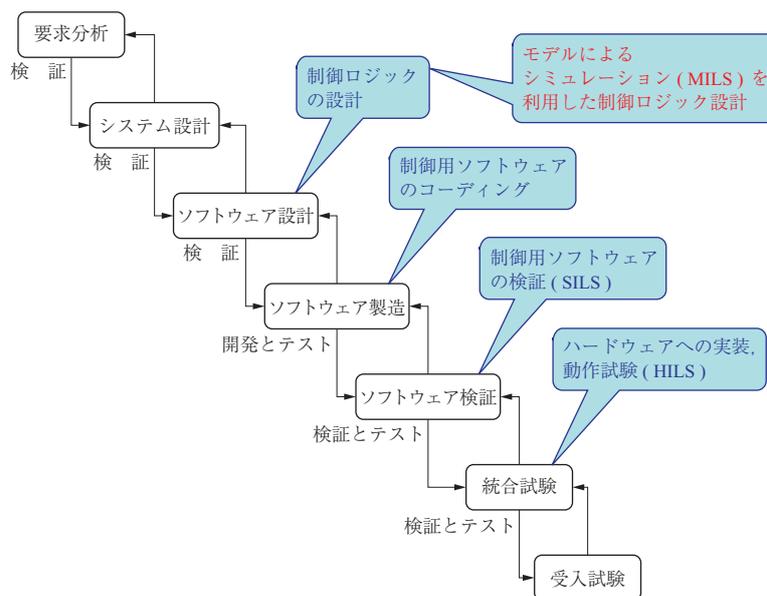
そこで、IA が取り扱うロケット電子制御システムにおいても、開発をより効率的、効果的に進めるため、モデルベース開発を試行することにした。

以下にモデルベース開発適用に向けたこれまでの取り組みである制御設計への適用に対する調査、自動コード生成に対する調査⁽⁵⁾について述べるとともに、今後の課題である上流のシステム要求分析、システム設計におけるモデルベース開発の適用について述べる。

2. モデルベース開発適用に向けたこれまでの取り組み

2.1 制御設計検証への適用

まずはじめに、ロケット電子制御システムにおいて、IA が従来開発を進めている電動アクチュエータ (EMA : Electro Mechanical Actuator) の制御ソフトウェア設計にモデルベース開発を試行した結果を示す。制御ソフトウェアの設計も通常の設計と同様に、① 制御ロジック設計 ② 制御ソフトウェアコーディング ③ 制御ソフトウェアの検証 (SILS : Software-In-the-Loop Simulation) ④ ハードウェア実装と統合試験 (HILS : Hardware-In-the-Loop Simulation) , といった開発プロセスを経る。制御設計へのモデルベース開発の適用には、制御ロジックへのモデル適用が最初のステップになる。第 2 図に制御ソフトウェアの設計を示す。

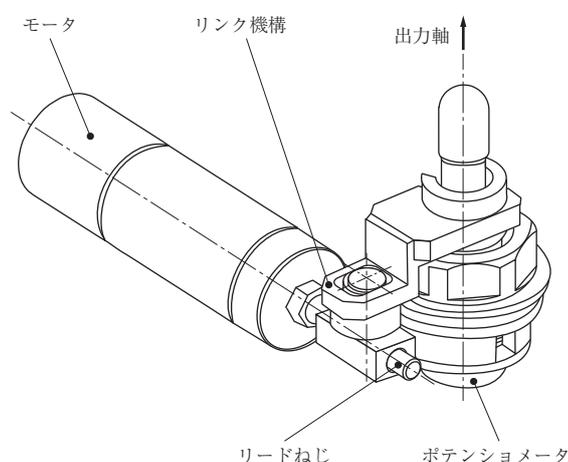


第 2 図 制御ソフトウェアの設計
Fig. 2 Development process of control software

実機製造前の段階で設計の後戻りを少なくするには、MILS を充実させることが有効である。MILS とは制御ロジックおよび制御対象物（以下、プラント）それぞれに対し特性を記述したモデルを作成し組み合わせることで、モデル上で制御の振る舞いを確認することであり、その利点は、組み立てた制御ロジック性能の確認、パラメータの決定を早期に実施できることである。また、一度プラントモデルを作成した後は、制御ロジックに変更が必要な場合の検討や、変更したロジックの動作確認が机上で容易に行える。さらに同種の機器の制御設計に再利用することも可能になる。しかし、上記で述べた MILS による有効性はプラントモデルの正確性に大きく左右されるという側面がある。この場合は、プラントモデルの熟成に時間を要するというジレンマがあり、新技術・新製品開発には必ずしも大きな効果を発揮しない場合がある。

第 3 図に今回調査に用いた、動翼用の低コスト電動アクチュエータシステムの試作品の概要図を示す。これは、①モータ ②リードねじ ③リンク機構 ④出力軸角度測定のためのポテンシオメータ、から構成される。

この EMA の制御設計は MATLAB/Simulink（アメリカ：MathWorks 社）による MILS で検証を試みた。しかし、設計初期段階では EMA の非線形要素のモデル化が不十分であったため、統合試験時に出力軸が振動するという事象が発生し、結果的には、ロジックの検討－コーディング－検証－実装・動作試験、の繰り返しといった制御設計のイタレーションが起こった。このコーディングから実装までのプロセスは、通常のプロセスを経ると時間が掛かるため、MILS による設計効率化は達成できていなかった。

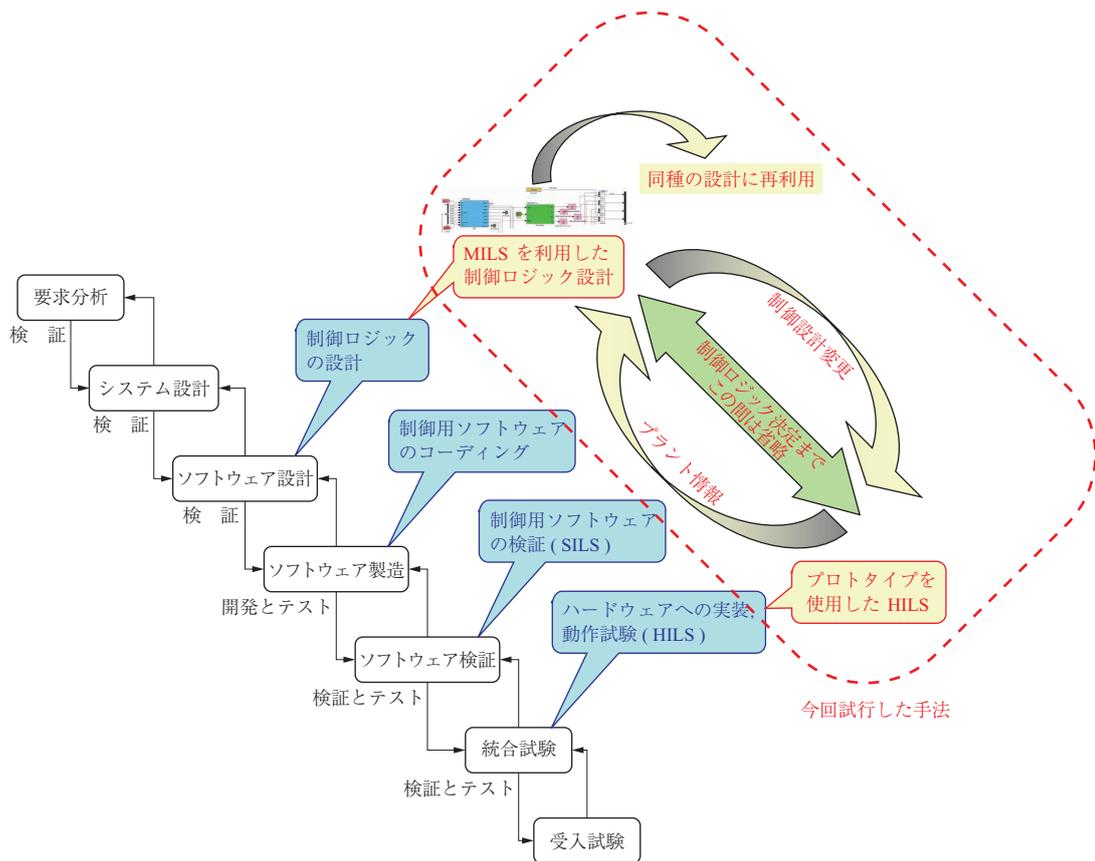


第 3 図 電動アクチュエータ概要図
Fig. 3 Electro mechanical actuator

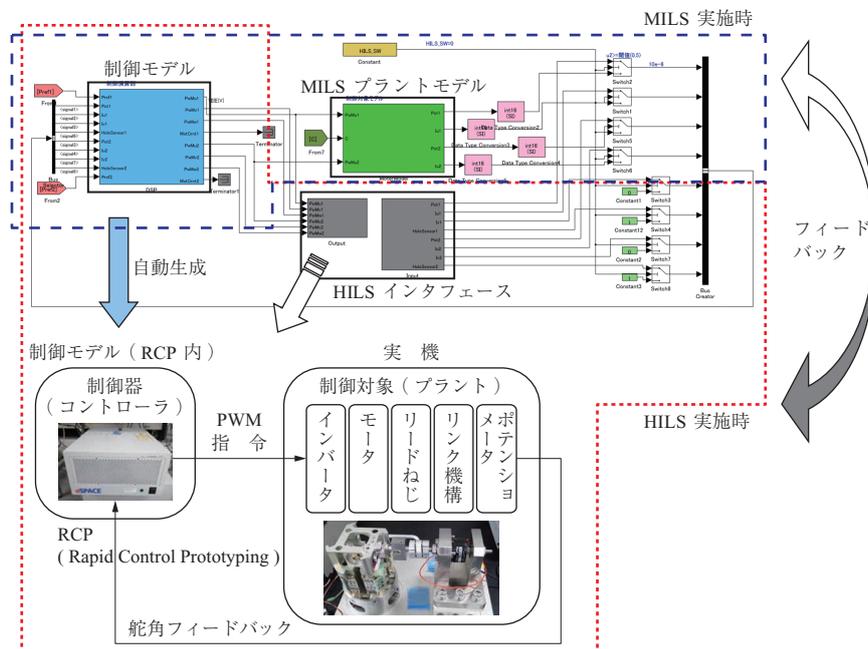
このように、制御対象物の理解が不十分のまま開発を進めると、MILS であらかじめ制御性能を確認していても、統合試験や受入試験時に不具合が発生し、問題解決のための後戻りが大きくなる可能性がある。今後、機能・性能を満足しつつも、必ずしも高価で高性能、精密加工された機構に頼らない最適な低コスト設計を実現するには、ばらつきや非線形性など設計が困難で避けてきた要素も積極的に制御で対応することが求められる。つまり、開発のできるだけ初期の段階でも非線形要素などを取り入れたプラントモデルを作成しておくことが課題になる。そこでこの課題を解決するため、次のような手法を試行した。この考え方は、制御設計の試行錯誤の効率化と精密なプラントモデル構築時間の削減のため、MILS と HILS を同形態でほぼ同時に実行させるというものである。従来は MILS - SILS（ソフトウェア上での検証）- HILS と段階的に構成し、ソフトウェア製造過程を繰り返していたが、早期の段階で試作レベルのハードウェアがあれば、MILS に取り込み、ロジックの検証や実機の細部の理解を進めるというソフトウェア製造過程を必要としない HILS を試行した。**第 4 図**に今回試行した手法の考え方（概要）を示す。

第 5 図に今回試行した MILS および HILS を同形態で対応可能とした手法を示す。MILS 実施の場合はこれまでと同様、制御対象物として MATLAB/Simulink のプラントモデルを使用する。HILS 実施の場合は、**第 5 図**の下に示すように、実機動作にモデルから自動生成した制御ロジックを組み込んだ RCP（Rapid Control Prototyping：実機制御を可能にする dSPACE 社（ドイツ）開発の装置）と実機を組み合わせる。

これによって、実機を動かす際にこれまでのようなソフトウェア製造、ソフトウェア検証などの実装工程を踏まわず、制御モデルと実機をリアルタイムに組み合わせることができると、実際に実機をリアルタイムに動作させながら、さまざまな内部パラメータ変更に対する評価を実施できる。また、容易にロジックを変更し動作させられることから、実機の詳細な性質の測定の効率化が図られた。この手法を実際に試行した結果、制御ロジックの変更に掛かっていた時間を半分以上に短縮することが可能になった。つまり、プラントモデルの正確性が十分でない場合、従来は制御設計を完了させるためには、制御・ソフトウェアのエンジニア各 1 名とオペレータの 3 名の技術者で 1 トライ最低 30 分の作業を制御設計が完成するまで繰り返す必要があったが、これによって制御設計者 1 名が数時間で制



第 4 図 今回試行した手法の考え方 (概要)
Fig. 4 Concept of this trial



第 5 図 電動アクチュエータの MATLAB/Simulink モデルと、HILS 形態
Fig. 5 MATLAB/Simulink model and HILS configuration of electro mechanical actuator

御設計が完了できるようになった。

先に述べたとおり、MILS 単独では、プラントモデルが実機の非線形性を正確に表現しきれていない場合、制御設計の検証に抜けがでてくる。したがって、たとえ試作レベ

ルのものであっても、HILS によって実機の非線形性を検出することが必須となる。その際、上記のような形態を用い、リアルタイムで制御ロジックの検証を行うことは非常に有効であることが分かった。

また、この形態で HILS を実施する利点として、制御ロジックを容易に変更し、動作させることができるため、プラントパラメータの収集が容易になることである。HILS で収集した実機データは、MILS で用いているプラントモデルに反映させ、制御ロジックやパラメータに変更が必要な場合の検討や、変更したロジックやパラメータの動作確認に使用する。さらには、作成した詳細なプラントモデルは同種の機器の制御設計に再利用し、実機製造前などの早期段階における制御性能の確認、パラメータの決定などといった MILS による初期検討の有効性を向上させる（第 4 図参照）。この HILS-MILS のサイクルを繰り返すことで、開発の初期段階から制御設計の検証が可能になり、制御設計を修正するような手戻り作業を軽減できると考えている。

2.2 コード自動生成の評価

コードの自動生成は人による作業を減らすことができ、コーディング時間の削減に非常に効果的である。組込みシステム開発のさらなる効率化のため、コード自動生成の評価を実施した。評価に用いたコードは電動アクチュエータの制御用のソフトウェアで、自動生成の手法として、TargetLink（dSPACE 社：量産コード自動生成ツール）および RTW（Real Time Workshop, MATLAB/Simulink で生成した制御ブロック図の自動コード生成ツール。現在は MATLAB coder および Simulink coder となっている）である。これらのツールを使用すると、モデルで表現された制御ロジックを瞬時に実装用のプログラムに変換するこ

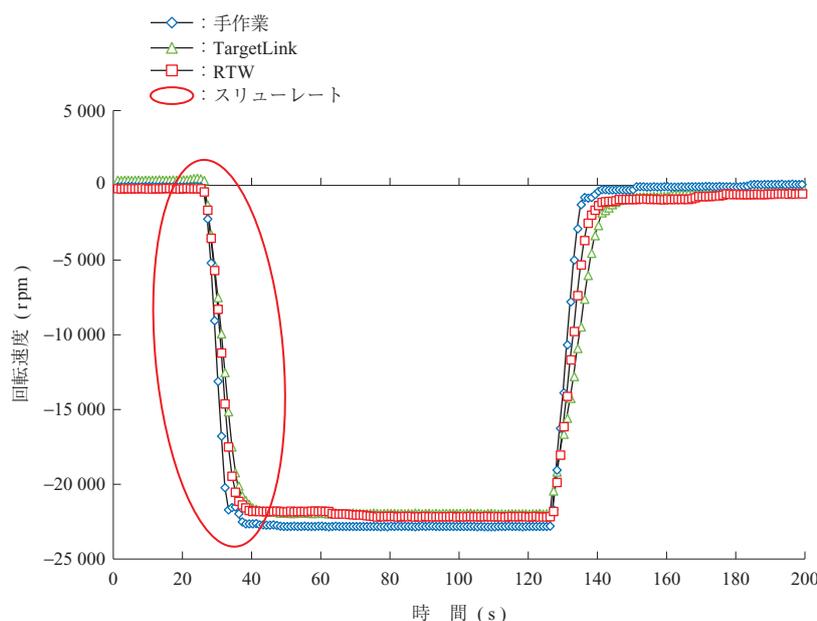
とが可能になる。本調査では自動生成されたソフトウェアと、従来どおり手作業で実装されたソフトウェアとを比較し、制御性能および、ハードウェア性能に対する影響の有無を評価した。

まず、手作業、自動生成それぞれの制御性能の比較を行った。第 6 図に手作業、TargetLink による自動生成、RTW による自動生成のソフトウェアそれぞれに対し、電動アクチュエータのステップ応答を比較したグラフを示す。青いマークは手作業、緑のマークは TargetLink、赤いマークは RTW の結果を示す。グラフから、自動生成したコードのスリューレート（応答速度を表す指標）は手作業のものとはほぼ一致しており、制御性能に有意差はない。

次に、手作業、自動生成によるソフトウェアのサイズ比較を行った。第 1 表はこれらのコードのメモリサイズをまとめたものである。比較の結果、2 種類の自動生成したコードのメモリサイズは、手作業のコードに対し数%の差に留まることが分かった。したがって、ソフトウェアを自動生成した場合も、メモリ容量を大きく圧迫するようなことは起こらず、ハードウェア性能への影響に対する問題はない。

以上の結果から、自動生成したコードは手作業のコードと性能にほとんど差がないと判断し、IA の具体的なプロジェクトへ投入を開始することにした。

このように、自動生成したコードを用いコーディング時間を削減することで、ソフトウェア開発をより効率的、効果的に進めていくことができる。さらに、コードの自動生



第 6 図 電動アクチュエータのステップ応答比較
Fig. 6 Step response of an electromechanical actuator

第 1 表 電動アクチュエータ制御コードのメモリサイズ
Table 1 Step response of an electromechanical actuator

メモリ区分	手作業コード	TargetLink 自動生成コード (%) *1	RTW 自動生成コード (%) *1
ROM	1	+1.6	-0.5
RAM	1	+4.1	+1.5

(注) *1: サイズは手作業コードを基準に示す。

成は、要求の変更によるコードの再生成が容易となるため、特に、複雑な制御プログラムや、大規模プログラムなど、コードの全貌の把握が難しいようなソフトウェアの変更への対応力が向上する。

3. 上流工程におけるモデルベース開発の適用

よく知られているとおり、組込みシステムの開発は V 字サイクルで表すことができる。第 7 図に IA のモデルベース開発適用によるソフトウェア開発プロセス向上の取組みを示す。2 章に述べた取組みは主に、第 7 図の「ソフトウェア設計」、「ソフトウェア製造」、「ソフトウェア検証」に対応しており、設計の比較的下流工程に対する MBD (Model Based Development) 適用の効果や留意点など、さまざまな知見を得ることができた。

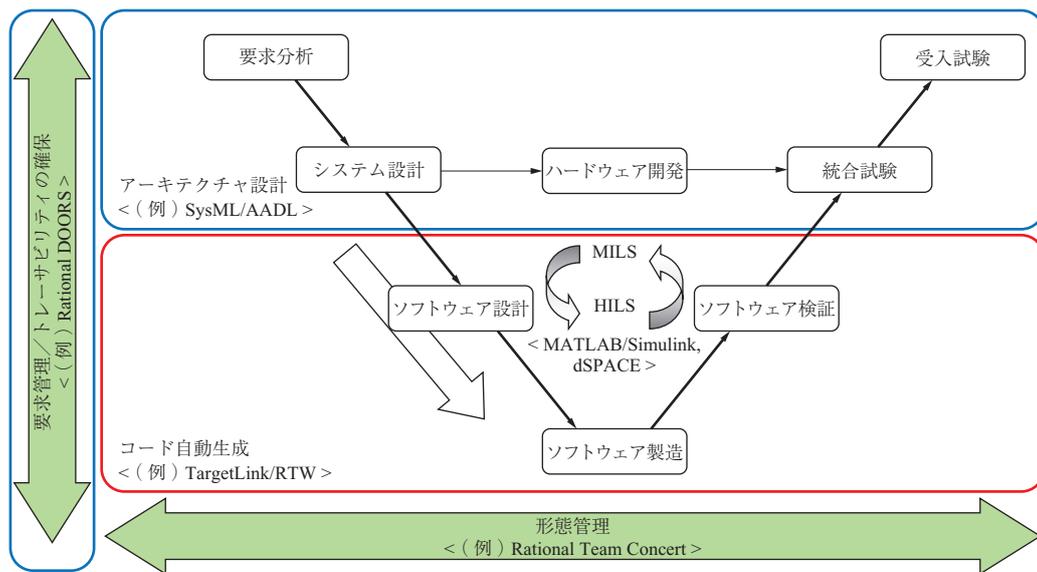
今後は、開発のより上流工程である「要求分析」、「システム設計」に対するモデルベース開発の適用、また開発プロセス全般にわたるトレーサビリティの確保について調査を実施していく必要がある。

今後の課題としては二つの側面があり、第 1 に、要求トレースを確実に実施できるような仕掛けを作る。また第 2 に、モデルを設計仕様として利用し、動作させることでシステム検証を実施できる仕組みを作る。要求トレーサビリティの確保は、要求、仕様、モデル、実装、テスト項目が漏れなくリンクされるようなツールを用いた管理を目指している。その際、特に複雑な振る舞いを示すシステムや、大規模なシステムに対してはモデルで振る舞いを記述し、可視性の高いトレースを可能にする。

これによって要求の実装漏れの防止、仕様の検証漏れの防止、要求変更があった場合の実装変更に対応できると考えている。さらに、モデルによるシステム検証では、システムに対する理解を深め、要求解釈の齟齬を減らし、設計欠陥のプロセス上流での対処を目指している。さらに、システム全体のモデル化によって、構成品それぞれの処理の流れが、システムとして整合性が取れているか確認できるようにする。以上の手法と、2 章で述べたこれまでの取組みである制御設計のモデルベース開発手法とを組み合わせることで、製品開発のライフサイクル全域にわたって、後戻りの少ない効率的な組込みシステム開発の実現を目指していく。

4. 結 言

本稿では、ロケット電子制御システムのモデルベース開発適用に関し、電動アクチュエータ制御設計への適用に対



(注) ①: 今後の課題
②: 本文の 2 章の範囲

第 7 図 モデルベース開発適用によるソフトウェア開発プロセス向上の取組み
Fig. 7 V process with MBD tools

する調査, 自動コード生成に対する調査について述べ, その結果, ソフトウェア設計や検証過程など比較的下流工程へのモデルベース開発適用は開発を効率的, 効果的に進めるために非常に有効であることを示した.

今後はより上流工程において, モデルベース開発を要求トレーサビリティの確保, システム成立性の早期検証という側面を中心に適用していき, モデルを活用した一連の開発プロセスの確立を目指すことが課題になる. さらには, このプロセスを必要に応じてほかのコンポーネントや, システム全体にまで拡張することによって, 大規模で複雑な組込みシステムをより効率的, 効果的に開発し, ロケット分野に対してソフトウェアが提供する付加価値をさらに高めていく.

参 考 文 献

(1) 刀川 眞: 我が国の社会的特性に着目した組込

みシステム開発の方向性 - エレクトロニクス化された耐久消費財におけるソフトウェア開発の強化策 - 科学技術動向 2011 年 125 号 pp.12 - 22

(2) Y. Morita : Innovative Concept of Epsilon Rocket and its Evolution JSASS-2013-S4001

(3) H. F. Peter and P. G. David : Model-Based Engineering with AADL Addison-Wesley (2012. 10)

(4) Imon Chakraborty et. al : Development of a Modeling and Simulation Environment for Real-time Performance Analysis of Electric Actuators in Maneuvering Flight AIAA 2013 (2013. 1)

(5) K. Tsujimoto et. al : Applying Model-Based Development (MBD) to embedded systems AIAA SciTech 2014 (2014)