

15-A-13

大規模システムへのモデルベース開発手法の適用¹

1. 適用した技術や手法の概要

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。

本事例では、大規模システムであるロケット電子制御システムのソフトウェア開発をより効率的、効果的に進めるためにモデルベース開発の適用可能性について検討を行い試行した。

制御システムの設計にモデルベース開発を適用する場合、MILS (Model In the Loop Simulation)² 実施時に、制御対象物のモデル化の正確性が不十分だと MILS による設計効率化はそれほど期待できないという課題があった。しかし、ある工夫を取り入れて MILS と HILS (Hardware In the Loop Simulation)³をほぼ同時に実行させることで、MILS によるモデル化の正確性が不十分な場合にも開発効率をかなり上げることができた。これにより、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。

また、組込みシステム開発のさらなる効率化のために導入したコード自動生成の評価を行った。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た結果、両者の制御性能には有意差がないこと、およびハードウェア性能に影響を与える使用メモリサイズにも有意差がないことが確認された。この事実により、自動生成されたコードが、手作業で実装された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。

本事例では、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（以下、IHI エアロスペース）におけるロケット電子制御システムへのモデルベース開発の適用に対する取り組みを紹介する。

¹ 事例提供: 株式会社 IHI エアロスペース 基盤技術部 下浦 美那 氏、辻本 圭史 氏、大依 仁 氏、ロケット技術部 森田 康志 氏

² ソフト開発におけるシミュレーション技術の一つ。シミュレーションに際し、実際のハードウェアは使用しない。ハードウェアをモデルのみでシミュレーションする

³ ソフト開発におけるシミュレーション技術の一つ。シミュレーションに際し、実際のハードウェアを使用する

2. 適用した技術や手法の導入に踏み切った理由や経緯

2.1. 背景と課題

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。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。

従来の開発ではシステムやソフトウェアの開発の際、ウォーター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。ウォーターフォール方式とは、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。

しかし、上記の背景と開発スタイルには、以下の(A)～(D)に示す課題があった。

- (A) 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。
- (B) ウォーターフォール開発では、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と検出が難しかった。
- (C) 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る。
- (D) IHI エアロスペース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。

2.2. 課題への対策

このような従来からの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が組込みシステム分野に対して広がりつつある。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、開発を進めていくという手法のことである。モデルとは、システムの振る舞い（仕様、制御アルゴリズム）をグラフィカルに表したもので、状態遷移図やフローチャートは基本的なレベルでのモデル例である。モデルによって記述することで、システムの内部構造や処理フローを直感的に理解できる。また、モデルベース開発の特徴として、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。近年では、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、一定の効果がでており注目を集めている。

そこで、本事例では、IHI エアロスペース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、モデルベース開発の適用可能性について検討を行い試行した。具体的には、電動アクチュエータ制御設計への適用に関する調査、コード自動生成に関する調査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も触れる。

3. 技術や手法の適用のための事前準備や工夫

3.1. 活動を開始する上での準備

ツール業界を広く調査し、よく利用されているツールを選択した。ツールベンダに対し技術支援を依頼し実施してもらった。具体的な技術支援は、以下の2点である。

- ① 電動アクチュエータ制御設計への適用では、既存の制御プログラムから、RCP (Rapid Control Prototyping : 実機制御を可能にする、dSPACE 社 開発の装置) へ適用するためのモデルの作成とモデルの評価についての支援を仰いだ。
- ② コード自動生成では、自動生成したソースコード評価についての支援を仰いだ。

3.2. 活動を開始するための工夫

社外の技術支援を適宜取り入れるとともに、社内の上層部に対し、有効性等の説明を実施し、支援体制を整えた。具体的には以下内容を上層部に説明しこの手法の有効性を理解してもらった。

- ① 現状分析として、制御システムの設計が高度化、複雑化してきていることの説明と、紙ベースの従来開発手法の問題点を説明した。
- ② 具体的方策の RCP やコード自動生成を、開発プロセスのどの部分で適用するかを説明した。
- ③ 具体的な効果として、効率化（工数半分）と高品質化（不具合流出防ぐ）をあげた。

3.3. 現場に合わせたカスタマイズやテーラリング等

実際のユーザーを想定し、製品に適用できるような環境を整備した。具体的には、RCP 実行用のホスト PC、リアルタイムシミュレータ、実プラントの環境を整備した。また、実装に向けては、コード自動生成機能のためのモデルチューニングを実施した。さらに、後工程での検証を容易にするため、実コントローラと実プラントを組み合わせることで応答特性、周波数特性が取得できるような構成を用意した。

上記のように環境を整備した後に、現場に向けて社内デモンストレーションを実施した。デモンストレーションで提起された要望（ツールのボタンや表示など）について対応し、環境に反映させた。

3.4. 使いこなせる人材の育成

導入体制は管理職 1 名、スタッフ 2 名の体制で開始した。社内調整は管理職が中心となり実施し、実行面はスタッフが対応した。環境構築からスタートしたため、2 名で 3 か月～半年の期間を要した。

育成については、ツール習得だけではなく、習得した技術を用いて実際にツールを使用し、製品へ適用した。教育は、OJT とスキルアップツールを利用し適宜実施した。

3.5. 電動アクチュエータ制御設計への適用時の工夫

ロケット電子制御システムにおいて、IHI エアロスペース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的设计と同様に、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。

実機製造前の段階で設計の後戻りを少なくするには、MILS（Model In the Loop Simulation）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。

この電動アクチュエータの制御設計は MATLAB/Simulink（米 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テレーションが起こった。このため、MILS による設計効率化は達成できていなかった。

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。今後は、機能・性能を満足しつつも低コスト化を実現するためには、精密加工された機構に頼らずに機構のばらつきや非線形性を制御で吸収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。従来は、MILS-ソフトウェア製造-ソフトウェア検証-HILS と段階的に構成しこの過程を繰り返していたが、今回試行した手法は、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に、ソフトウェア製造とソフトウェア検証の過程を踏まずに、MILS と HILS のみをほぼ同時に実行させるというものである。図 15-A-13-1 に今回試行した手法を示す。MILS 実行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 実行の場合は、図 15-A-13-1 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 と実機のプロトタイプを組み合わせる。

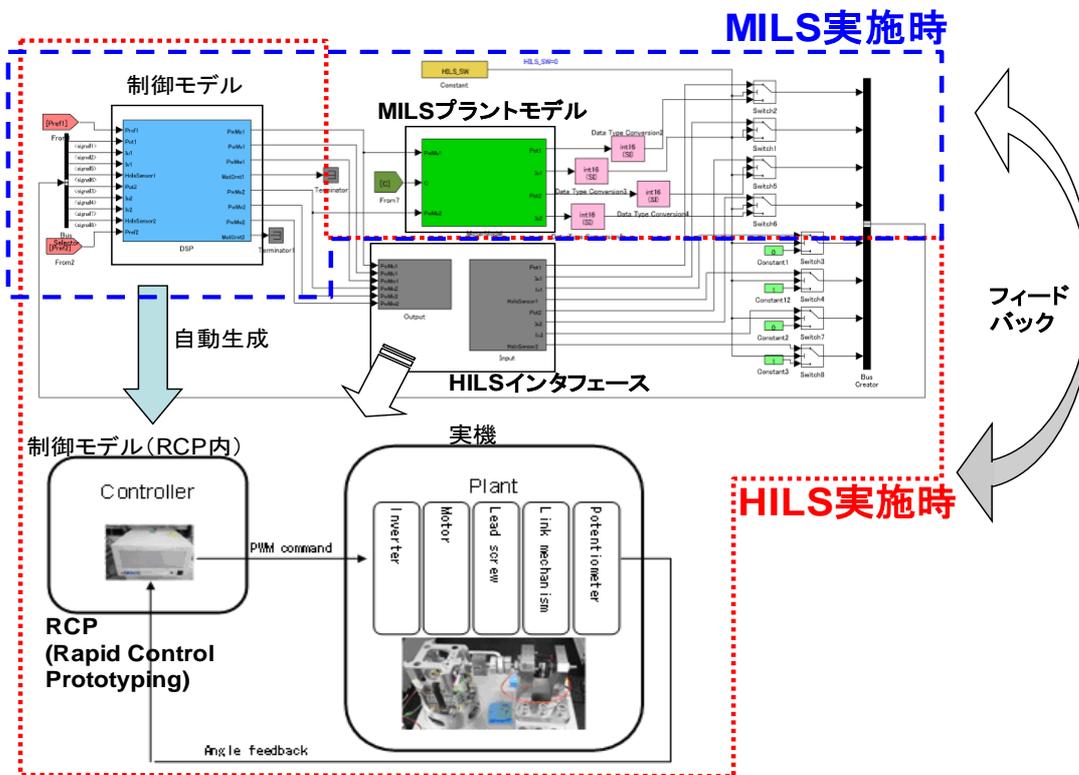


図 15-A-13-1 今回試行した手法

3.6. コード自動生成の事前評価

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。モデルベース開発を用いた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、TargetLink⁴および RTW⁵を使用した。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

⁴ dSPACE 社：量産コード自動生成ツール

⁵ Real Time Workshop、MATLAB/ Simulink で生成した制御ブロック図のコード自動生成ツール。現在は MATLAB coder および Simulink coder となっている

ことが可能になる。本評価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。図 15-A-13-2 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。

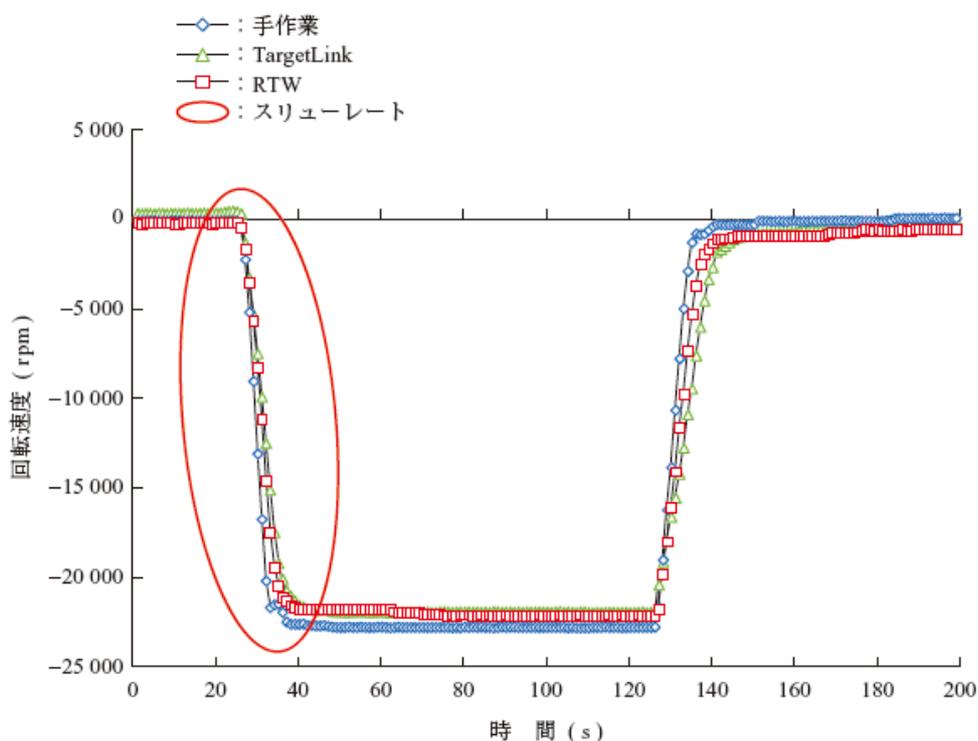


図 15-A-13-2 電動アクチュエータのステップ応答比較

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。表 15-A-13-1 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。

表 15-A-13-1 電動アクチュエータ制御コードのメモリサイズ

メモリ区分	手作業コード	TargetLink 自動生成コード(%) ^{注1}	RTW 自動生成コード(%) ^{注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

4. 適用した技術や手法の効果測定の方法とその結果

4.1. 電動アクチュエータ制御設計への適用の結果

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か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数回繰り返す必要があったが、この手法では制御設計者1名が数時間で制御設計を1回で完了できるようになり、組込みシステム開発の現場では慢性的な悩みとなっている要員不足問題を解決できる糸口がつかめた。

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。図 15-A-13-3 に示した MILS-HILS 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。

4.2. コード自動生成評価の結果

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、IHI エアロスペース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。

4.3. 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。図15-A-13-3にIHI エアロスペース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り組みを示す。3.5に述べた取り組みは主に、図15-A-13-3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。

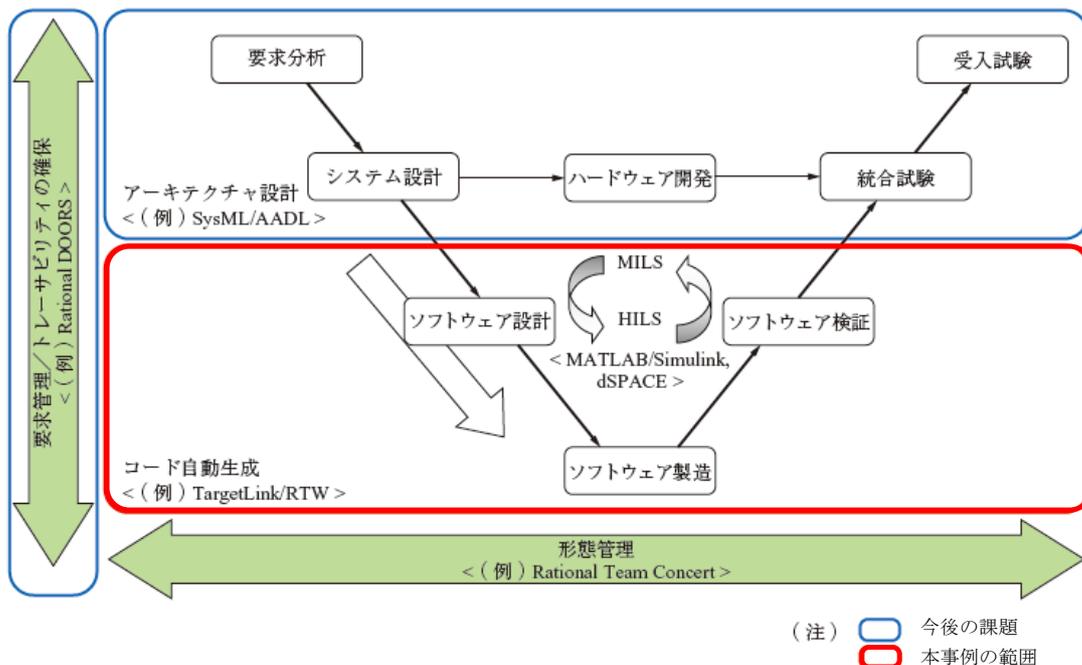


図 15-A-13-3 モデルベース開発適用によるソフトウェア開発プロセス向上の取り組み

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。

4.4. 品質と生産性

品質向上に対する効果としては、ハードウェアとソフトウェアの統合後の不具合が減少したこと、また不具合発生時の検証時間など対処にかかる時間が 1/10 以下に低減できたことが、あげられる。また、生産性に対する効果としては、「設計検証時間の短縮」、「コーディング工数の短縮」、「オートコーディング技術による単体試験工程の短縮」などが確認できた。

5. 導入時の工夫・苦労と適用途中での課題・対策

現場への導入時に具体的な以下の利点を認識してもらうことに苦労した。

- ・ 関連するシステムとのトレーサビリティを保証できる
- ・ システム固有の制御モデルや制御ロジックの検証ができる
- ・ ゲインチューニングを効率的に実施できる

現場適用途中での課題として、本手法を現場に適用していく上で、既存のプロセスと同居させながら、下記事項を検討することが重要である。

① 既存のソフトウェア開発プロセスの分析

本手法を適用するにあたり、既存プロセスを再分析、体系的に整理した上で、置き換え可能なプロセス群を識別し、具体的な置き換え方法の検討が必要である。

② 適用効果の評価指標の検討

本手法の適用効果について、ソフトウェア開発プロセス全体を俯瞰した評価指標の検討が必要である。

上記の課題に対し下記の対策を実施する。

- ・ 既存プロセスの再分析を実施し、整理する
- ・ 本手法に置き換えられる部分を具体的に検討する
- ・ プロセス全体を俯瞰した場合の評価指標を確立する
- ・ 本手法を実践し、上記指標に基づいた評価を実施する

6. 今後の課題と改善活動

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。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、3.5 で述べたこれまでの取り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によって、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。

さらに、従来からある開発プロセス等への組込みは現在検討中であり、手法を取り入れたプロセスづくりを実施中である。そのためには、従来の社内標準まで含んだ取り組みが必要と考えており、標準の改訂等を見据えたプロセスを社内でも実施している。また、IHI グループ全体への、活動内容の展開を実施しており、技術導入の検討が一部の分野でされつつある。

7. まとめ

本事例では、ロケット電子制御システムのモデルベース開発適用において、電動アクチュエータ制御設計への適用に関する調査、コード自動生成に関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。

具体的には、制御システムの設計にモデルベース開発を適用する場合、MILS の実施時に制御対象物のモデル化の正確性が不十分だと MILS による設計効率化はそれほど期待できないという課題があったが、ある工夫の採用で MILS と HILS をほぼ同時に実行させることで、MILS によるモデル化の正確性が不十分な場合にも開発効率をかなり上げることができた。

ある工夫とは、MILS 実行は標準通りに行うが、HILS 実行に当たっては、制御対象物には実機のプロトタイプを、それを動かすためのコントロールには dSPACE 社の製品である RCP を利用することである。この結果、ソフトウェア製造・検証などの実装工程が不要となり、開発効率をかなり改善できた。また、トレーサビリティについても追跡が可能になった。

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ことが期待される。

参考文献

- [1] 下浦 美那: 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、IHI 技報 Vol.54 No.1(2014)
- [2] 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
- [3] Y.Morita : Innovative Concept of Epsilon Rocket and its Evolution JSASS-2013-S4001
- [4] H.F.Peter and P. G. David : Model-Based Engineering with AADL Addison-Wesley、2012.10
- [5] Imon Chakraborty et. al: Development of a Modeling and Simulation Environment for Realtime. Performance Analysis of Electric Actuators in Maneuvering Flight AIAA 2013、2013. 1
- [6] K. Tsujimoto et. al: Applying Model-Based Development (MBD) to embedded systems AIAA. SciTech 2014、2014

掲載されている会社名・製品名などは、各社の登録商標または商標です。

独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター (IPA/SEC)