

電子制御 AWD システムにおける 新 MBD ソフトウェア開発プロセス

New MBD Software Development Process for Electronic Control AWD System

渡 部 知 樹*

Satoki WATANABE

塩ノ谷 陽 介*

Yosuke SHIONOYA

吉 野 弘 規*

Hiroki YOSHINO

関 孝 幸*

Takayuki SEKI

要 旨

電子制御 AWD システム開発において、従来の実行可能なモデル表現による制御設計 (Model Based Design: MBD) 仕様書を適用したソフトウェア開発プロセスに対して、仕様品質の向上を目的として、システム制御設計の強化をおこなう新 MBD ソフトウェア開発プロセスを構築した。その際、検証プロセスの変更やシステム制御設計とソフトウェア詳細設計との開発役割領域の見直しをおこなった。また、それに伴う仕様書の作成手法の構築や、仕様書の流用率を上げるためにアプリケーションのプラグイン構造化をおこなった。

このプロセスを、新規量産する電子制御 AWD システムのソフトウェア開発に適用した結果、従来のソフトウェア開発プロセスに対して仕様書変更量を 50% 削減できた。

ABSTRACT

A new model-based design (MBD) software development process was constructed for the development of electronic control all-wheel-drive (AWD) systems. This new process enhances system control design with the aim of increasing specification quality compared to the previous software development process that applies executable model-based control design specifications. The new process changes the verification process and adjusts the division of development duties between system control design and software detail design. In addition, a method for generating the related specifications was constructed, and plug-in structuring was performed for applications to increase the specification reuse rate.

The new process was applied to the development of software for a new mass-produced electronic control AWD system, and resulted in a 50% reduction in the volume of specification changes compared to the previous software development process.

1. まえがき

近年、車両に搭載される電子制御システムおよび制御ソフトウェアは、複雑かつ大規模化しているが、一方で制御信頼性向上、開発期間の短縮、開発の効率化が求められている。そこで、ソフトウェア開発および仕様書の記述手法として実行可能なモデルによる MBD 仕様書の導入が進められてきており、モデル記述のガイドラインも制定されている⁽¹⁾。

*四輪 R&D センター

電子制御 AWD システム開発では、2007 年の量産機種から、MATLAB/Simulink, Stateflow⁽²⁾に汎用のコード自動生成ツール⁽³⁾と Honda 内製のデータ設定ツール⁽⁴⁾を活用して、固定小数点データモデル表現による MBD 仕様書を適用した。この MBD 仕様書の適用は、システム開発者の仕様理解度の向上や MBD 仕様書とソフトウェアの不適合の削減、システム制御設計者が作成する仕様書からコード生成までのソフトウェアの設計を容易にすることを目的とした。モデルおよびコード生成に対する検証負荷

を軽減するために、MBD 仕様として使用できるブロックの種類の制限やブロックに適用するパラメータや入出力に対するデータ設定値の制約、データ名称など記述手法の規定をおこなった MBD ソフトウェア開発プロセスが構築された。

このプロセスを V 字プロセスで表現すると Fig. 1 のようになる。システム制御設計者は、MBD 仕様のモデルを作成するときに、モデル内で定義される固定小数点のデータ設定をおこなっていた。しかし、データ設定において、適切なスケリングをおこなうにはソフトウェア設計のスキルを要する。また、モデルへのデータ設定をおこなう工程は、データ設定だけでなくデータの説明などの付属情報の組込みもおこなうので MBD 仕様書の作成には多大な時間を要していた。本来、これらの設計はソフトウェア詳細設計の領域でもあり、システム制御設計とソフトウェア詳細設計の役割分担が不明確であった。

また、このプロセスにおいて、システム制御設計者は要求仕様とモデルの一致性を確認しながら MBD 仕様の作成をおこなうことができるが、仕様の確からしさを確認する手法は確立しておらず、V 字プロセスのソフトウェアを設計する工程において、動作確認結果を管理する仕組みもなかった。結果として、ソフトウェア開発の最終工程であるシステム検証でも MBD 仕様書作成時の動作確認時と重複したテストをおこなうという無駄の発生や、MBD 仕様作成時の動作確認で検出できなかった仕様不具合が、システム検証時に初めて発見されることもあった。そのときの仕様修正ループは大きいため、システム開発の後期に仕様変

更によるソフトウェア開発の割合が増大し、MBD による実行可能な仕様書のメリットを十分に活用できていないという課題があった。

2. 新 MBD ソフトウェア開発プロセスのねらい

従来の MBD ソフトウェア開発プロセスの課題を解決し、さらに、仕様品質の向上を実現することを目的として、新 MBD ソフトウェア開発プロセスでは以下のような手法を考えた。

- (1) 浮動小数点データモデルを用いたフロントローディング検証の導入
- (2) システム制御設計強化のための仕様書フォーマットの作成
- (3) MBD 仕様書のデータ抽出ツールの開発
- (4) 仕様流用率を向上させるアプリケーションのプラグイン構造化

これらにより、ソフトウェア開発効率を向上させ、仕様書修正の削減やシステム開発の後期のソフトウェア開発負荷割合の軽減を図った。

3. フロントローディング検証

3.1. 新 MBD ソフトウェア開発プロセスの V 字プロセス

Fig. 1 のような従来の V 字プロセスに対して、新 MBD ソフトウェア開発プロセスを Fig. 2 のような V 字プロセスとした。従来のプロセスからの変化点は以下の二点であ

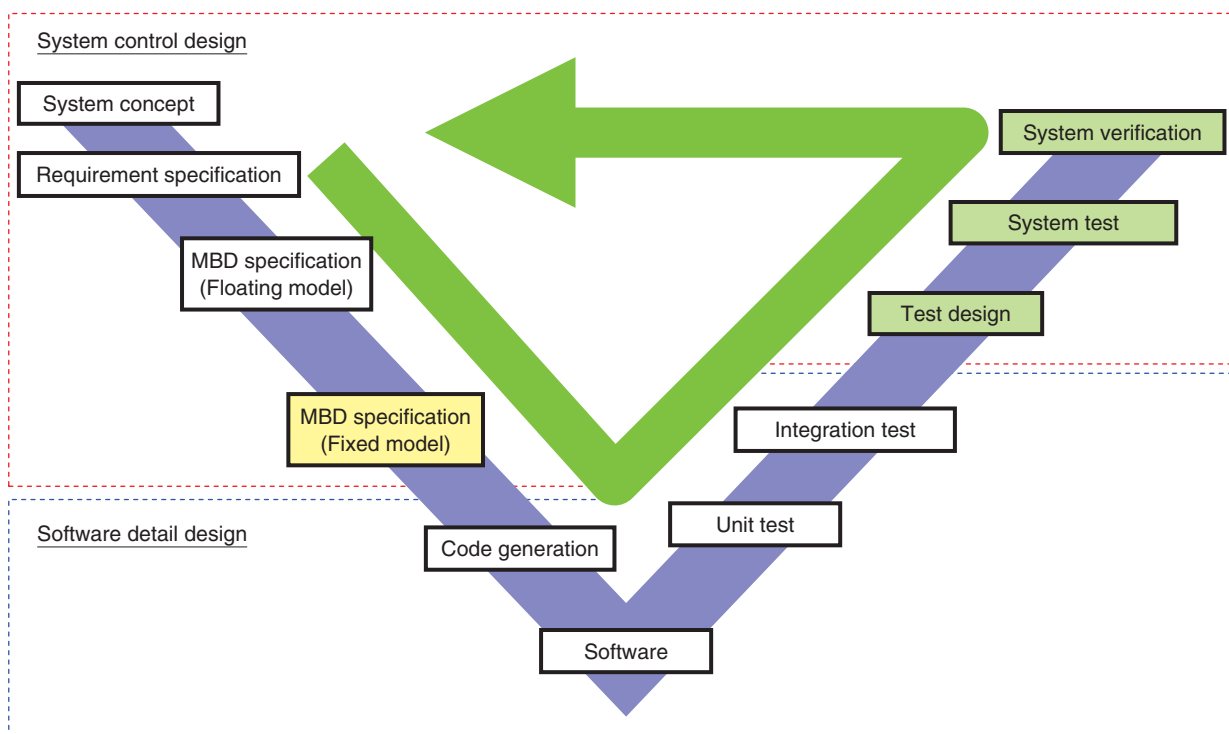


Fig. 1 V-process of previous MBD process

る。一点は、システム制御設計からソフトウェア詳細設計に移行する前に、フロントローディングで検証をおこなうことである。もう一点は、システム制御設計者が作成するMBD仕様書を浮動小数点データのモデルで表し、固定小数点データのモデルならびにデータスケールをソフトウェア詳細設計領域に移行させたことである。

ソフトウェア詳細設計に移行するまえにフロントローディングで検証をおこなう目的は、要求仕様やMBD仕様の不適合の下流工程への流出を防ぎ、修正ループを小さくすることで開発効率を向上することである。また、検証工程を分割しているため、システム開発の最終工程であるシステム検証の負荷を低減することができる。

フロントローディング検証をおこなうにあたっては、MBD仕様書を浮動小数点データモデルに留めておくことによりデータ設定工程を削減して、シミュレーションからフロントローディング検証までのループを効率的に循環させ、システム制御設計に集中した検証をおこなうプロセスとした。

3.2. 検証工程の分散

新MBDソフトウェア開発プロセスでは、Fig. 2のV字プロセスで示すように、検証工程をフロントローディング検証と実機によるシステム検証の二つに分けて検証負荷を分散させるプロセスとした。

フロントローディング検証は、定常的な制御動作確認や要求仕様とMBD仕様の一致性、パラメータ設計値の妥当性についてシミュレーションや机上検証を実施する。実機

によるシステム検証では、車両運動性能の評価や車両環境との適合性検証、過渡状態の検証を実施する (Fig. 3)。

新MBDソフトウェア開発プロセスでは、MBD仕様書を作成すると同時に、フロントローディング検証項目とシステム検証項目について、テスト設計仕様書を作成することにした。フロントローディングの検証イベントでは、それぞれの検証手法とフロントローディングの検証結果についてレビューをおこなう。このように、実機でのシステム検証手法も含め、フロントローディング検証でレビューをおこなうことにより、システム制御設計者や検証承認者がシステム検証工程のテスト内容について理解を深め、テスト内容の妥当性をチェックすることができる。これにより、システム開発の最終工程であるシステム検証で発見されることがある検証内容の不適合や仕様要求自身やソフトウェアとの不適合を削減する取り組みとした。

また、従来のプロセスでは、開発初期でソフトウェアに投入するデータは、厳密に設定されたデータではなく、実機でのテストをしながらパラメータの調整をおこなっていることが多く、パラメータの変更がシステム内の他仕様領域や他システムに影響を及ぼすこともあった。本プロセ

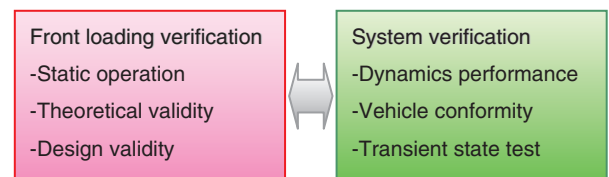


Fig. 3 Division of verification items

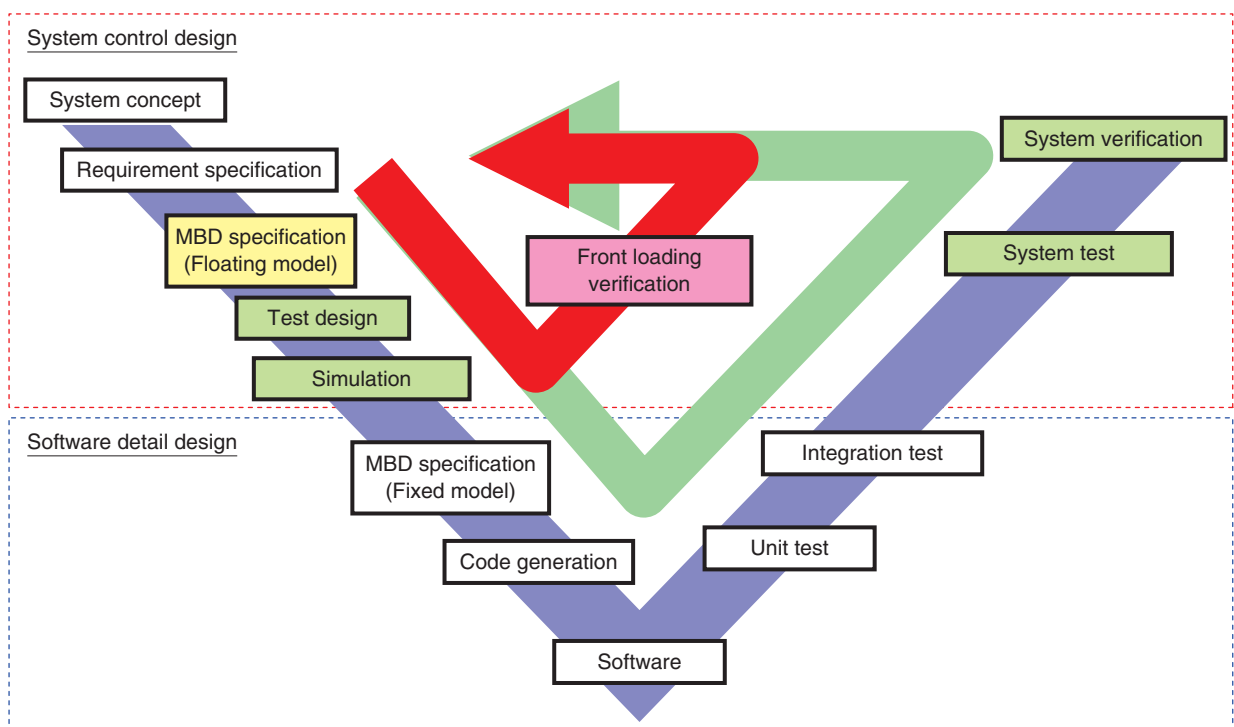


Fig. 2 V-process of new MBD process

スでは、開発初期であっても仕様検証され、かつ理論的なデータが設定されたソフトウェアであるため、パラメータ変更やそのソフトウェアの影響確認などの反復的な作業を削減している。

4. 仕様書フォーマット構成

新 MBD ソフトウェア開発プロセスでは、システム構想からシステム検証まで一貫した検証を推進するために、MBD 仕様書に加えて以下の四つのフォーマットを用意した (Fig. 4)。

- (1) システム構想書
- (2) 要求仕様書
- (3) テスト設計仕様書
- (4) テスト報告書

以下に各フォーマットの特徴を示す。

- (1) システム構想書

投入する機能の要求仕様に併記して、目的や要求の背景や構想内容、要求に至った経緯などのノウハウを記述するためのフォーマットである。

- (2) 要求仕様書

ソフトウェア詳細設計に仕様要求を明示する仕様書であ

り、システム構想書に対して背景やシステム構想の情報を取り除いて、投入する機能の目的や要求とその条件のみに記述された仕様書とする。なお、この要求仕様を詳細に具現化した仕様が MBD 仕様となる。

- (3) テスト設計仕様書

検証レビューでは、あらかじめ検証手法やパラメータ設定手法を記述し、それに則って検証した結果を示すが、要求と実現内容の不適合を容易にチェックできるように要求項目と検証手法や検証する観点、パラメータの設定手法を併記して記述する構成としている。このとき、検証手法やパラメータ設定手法は、フロントローディング検証項目とシステム検証項目を分割してそれぞれの内容を締結できるように記述する構成とした。

- (4) テスト報告書

フロントローディング検証結果とシステム検証結果をひとつのシートに記述するフォーマットにしており、フロントローディング検証の結果からシステム検証への引き継ぐ項目なども記述することにより、フロントローディング検証とシステム検証間の検証漏れを防止する構成とした。

5. モデル内データ情報の自動抽出

システム制御設計において、MBD 仕様書は浮動小数点データのモデル表現としているが、ソフトウェア詳細設計者がソフトウェアを製作するためには、データスケーリングの基準が必要である。また、システム制御設計者自身も、適切な制御をおこなうために、スケーリングなどの設定を要求したいデータが存在する。そこで、

- a. 各アプリケーションの最終出力の精度要求
- b. パラメータ設定のスケーリング要求

のように、ソフトウェア詳細設計者にデータ設定要求をおこなう。その手段として、MBD 仕様書とは分離して、上記二項目のデータ設定要求をおこなう仕様書 (以下データ設定要求書) を作成して、MBD 仕様書と一対でソフトウェア詳細設計へ提示する。そして、ソフトウェア詳細設計ではデータ設定要求書を基準として、要求するデータ以外のデータを含めて全データのスケールをおこなうソフトウェアを作成することとした。

ただし、データ設定要求書を作成する際、ハンドタイピングでデータの記述をおこなうと MBD 仕様とデータ設定要求書の間でタイピングミスなどの誤記によるデータ名称やデータ説明内容の不適合が発生する恐れがある。そこで、MATLAB/Simulink モデルにおいてデータ設定要求したいデータを抽出してデータ設定要求書に反映させるツールを開発して仕様書表現不備の低減をおこなった。

データ設定要求書を作成する自動抽出ツールは、

- (1) MBD 仕様書 (浮動小数点データモデル)
- (2) データ情報抽出プログラム

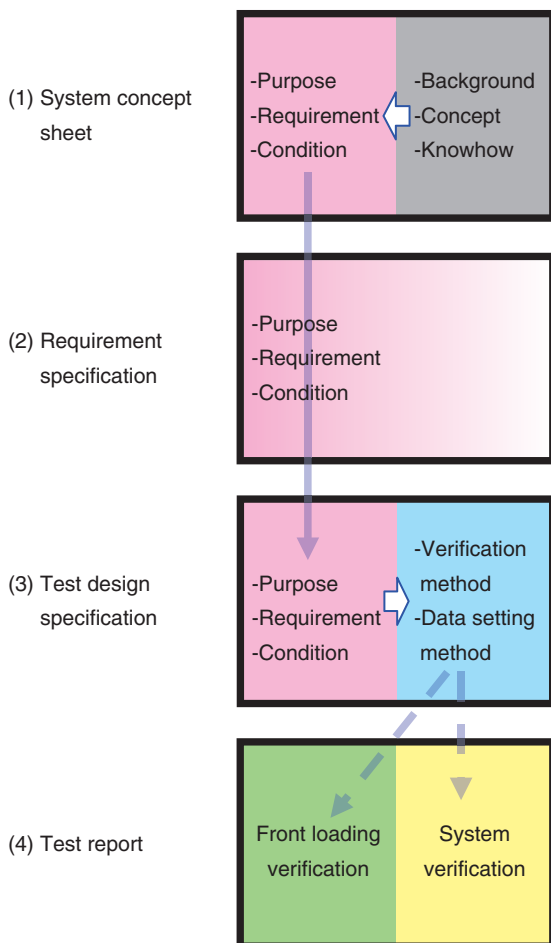


Fig. 4 Format of new process documents

(3) データ抽出要求および挿入プログラム

(4) データ設定要求書

の四つの要素で構成される。データ設定要求書は Fig. 5 のような流れで作成されるが、以下にその特徴を示す。

(1) MBD 仕様書 (浮動小数点データモデル)

データ設定要求書に定義されるべき変数およびパラメータが MBD 仕様書内のモデルブロックに表現される。また、MBD 仕様には、データ属性の認識を容易にするなど MBD 仕様の可読性向上や検証効率の向上のためにモデル内のブロック適用規定やデータ名称規定をおこなっている。データ設定要求書に抽出したいデータについても、MBD 仕様の記述規定により、プレフィックスの付加や抽出するデータを定義するためのプロパティの設定をおこなっている。

(2) データ情報抽出プログラム

データ情報抽出プログラムは、(1)でモデルに設定された情報をキーワードとしてモデルの階層構造を解析しながら検索して、モデル内に記述されたさまざまな情報からデータ名称やブロックに埋め込まれたデータ説明の情報とその階層のみを抽出するプログラムとした。

また、パラメータの抽出は、ひとつのブロックに対して複数のパラメータの設定を要求するデータがあるため、適

用するブロックに応じて、異なるパラメータ数を最適に要求できるプログラムとした。

(3) データ抽出要求および挿入プログラム

このプログラムは、データ設定要求書で起動され、このプログラムの要求によりデータ情報抽出プログラムが実行される。そして、データ情報抽出プログラムによって作成されたリストのデータ名称とその説明、データの階層をデータ設定要求書に挿入していくプログラムである。また、データ設定要求書に既存のデータがあり、改版を実行する場合は、データのマージ処理をおこなう機能を有する。

加えて、データ名称の重複チェックなどのデータチェックや、データ設定要求書の改版時に反映される情報の変化点が抽出できる機能を有することにより、MBD 仕様記述されるデータの不備を低減した。

(4) データ設定要求書

データ設定要求書には、上記プログラムによりモデルから抽出されたデータが整列するが、ソフトウェア詳細設計に対してデータ設定要求をおこなうために、各データに対してスケーリングや、データの単位を記述する。

設定要求を記述する際、タイピングミスによる誤記や複数の仕様作成者によるデータ設定要求のばらつきを削減するために、あらかじめデータの分類をおこない、データ設定要求値は分類に応じたりリストから選択するだけの構成とした。

6. アプリケーションのプラグイン構造

従来の MBD ソフトウェア開発プロセスのアプリケーション構成は、アプリケーション間のデータの入出力は直接変数を受け渡しする構成であり、アプリケーションの外部出力に変更が生じると、変数を参照しているすべてのアプリケーションは自身に変更がなくても参照元の変更の影響を受けて仕様書変更が必要となる。

新 MBD ソフトウェア開発プロセスでは、アプリケーションの仕様変更が他アプリケーションに与える影響を小さくして、仕様の流用率を向上することにより仕様書の作り直しを削減するアプリケーションの構成を考案した。これにより、定時間処理のソフトウェア処理構成でありながらアプリケーションの付け替えが容易にできるプラグイン構造となるように、変数が受け渡しされるアプリケーション構成とした。

そのプラグイン構造は、Fig. 6 のように OS データを各アプリケーションで演算するための物理値変換する Application Programming Interface (以下 API) と各アプリケーションの API を有することを特徴とする。各アプリケーションの API は、対になっているアプリケーションの出力が他のアプリケーションの入力として使用されるときに API からの入力の位置づけで使用できるように、汎用的

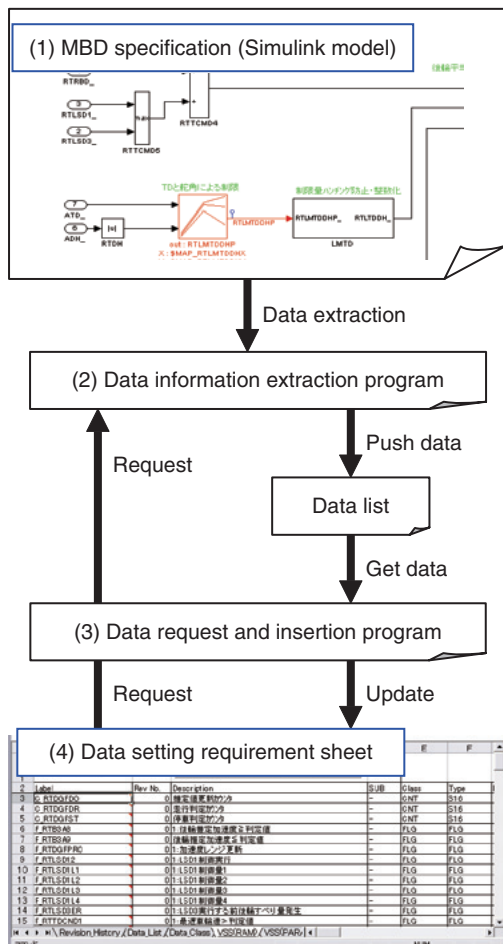


Fig. 5 Generation of data setting requirement sheet

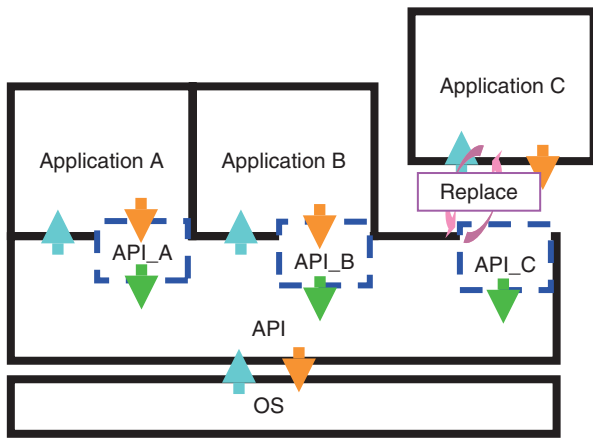


Fig. 6 Plug-in structure of applications

な変数に変換する。これによりアプリケーションの出力に変更が生じて各アプリケーションの API が変更を吸収するため、参照先のアプリケーションを変更しなくてもソフトウェアを生成することが可能となった。

例としては、通常の車両制御アプリケーションに対して、診断機による制御などの代替値制御のアプリケーションがあり、その制御値を通常の制御値に切り替えて使用したい場合がある。そのとき、車両制御アプリケーションに割り込ませて使用するのではなく、車両制御アプリケーションの API に制御値の切り替え処理をおこなう機能を持たせることにより、各アプリケーションは本来の目的の制御のみを要求する仕様を表現することができる。これにより、従来の開発では仕様をアドオンするとさまざまな領域に対しても実施しなければならない仕様変更を軽減したり、仕様書の流用率が向上でき、開発効率を上げることができた。

7. プロセスの評価

7.1. 仕様品質の評価

電子制御 AWD システム開発では、新 MBD ソフトウェア開発プロセスを新規量産システムのソフトウェア開発に適用した。システム制御設計者が提示した仕様書に対して、ソフト詳細設計者から発行された仕様整合書の発行枚数や記載内容を分析して、従来の MBD ソフトウェア開発プロセスを初めて新規システム開発に適用したときのものと比較をおこなった。

まず、仕様書量に対する仕様整合書の発行枚数の割合は従来の MBD ソフトウェア開発プロセスを適用した開発に比べ約 20% 減少した。

また、仕様整合書の内容を分析して、要求仕様と MBD 仕様の不適合や仕様書表現不備に関連する「記述ミス」「データ設定不具合」「仕様不適合」の項目とその他に分類した。その結果を Fig. 7 に示す。従来の MBD ソフトウェア

開発プロセスでは、要求仕様や MBD 仕様の不適合や仕様書表現不備の項目の割合が全体の 80% 以上を占めていたが、フロントローディング検証を実施した新 MBD ソフトウェア開発プロセスでは 40% 以下となった。そして、新 MBD ソフトウェア開発プロセスでは、総仕様書量に対する要求仕様や MBD 仕様の不適合や仕様不備項目による仕様書変更の割合が、従来の MBD ソフトウェア開発プロセス適用時に比べ約 50% 減少した。

7.2. ソフトウェア開発負荷の評価

システム開発日程に対して時系列で出図されたソフトウェア仕様書量の比較結果を Fig. 8 に示す。従来の MBD ソフトウェア開発プロセスでは、システム開発後期に仕様書出図がピークとなっているが、新 MBD ソフトウェア開発プロセスでは、開発終了に近づくにつれ仕様書の出図量が減少している。

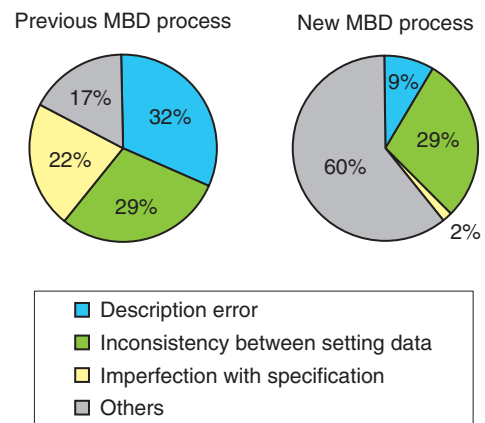


Fig. 7 Comparison of subject in requirement confirmation sheet

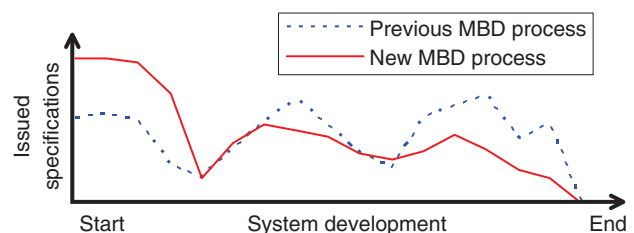


Fig. 8 Volume of issued specifications

8. まとめ

電子制御 AWD システムのソフトウェア開発プロセスにおいて、MBD 仕様書を浮動小数点データ表現とすることで、システム制御設計領域とソフトウェア詳細設計領域を分離した。これにより、フロントローディング検証が効率的におこなわれるようになり、仕様品質を向上できた。

新規量産の電子制御 AWD システムのソフトウェア開発に適用した結果、要求仕様と MBD 仕様の不適合や仕様書表現不備による仕様書の修正が減少したことにより、仕様作成工数および検証作業量が削減され、システム開発後期のソフトウェア開発負荷割合の軽減をおこなうことができた。

参考文献

- (1) <http://jmaab.mathworks.jp>
- (2) http://www.mathworks.co.jp/products/product_listing/index.html
- (3) <http://www.dspace.com/en/inc/home/products/sw/pcgs/targetli.cfm>
- (4) 箕輪信太郎, 廣田俊明, 横山浩紀, 石黒哲矢, 大塚訓: パワープラント制御ソフトの自動生成, Honda R&D Technical Review, Vol.16, No.1, p.243-248

■ 著 者 ■



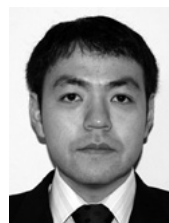
渡部 知樹



吉野 弘規



塩ノ谷 陽介



関 孝幸