

トランスミッション
ECU 向けの効率的な
ソフトウェア開発プロセス

Handling High Numbers of Variants

電子制御ユニット (ECU) ソフトウェア開発では、多数のバリエーションを処理しなければなりません。これは避けて通れない多くの課題の1つです。このため、ZF Friedrichshafen AG などの自動車サプライヤでは、重要な開発段階でツール支援による手法を採用しています。そのようなツールの1つがdSPACEの量産コード生成ツールである TargetLink です。



り新たな課題が生まれてきています。時には安全にも関わる多数のハイブリッド制御機能は多くの場合ネットワーク化されていますが、これらの機能は既存のソフトウェア環境に組み込み、限られたメモリや演算処理リソースを使用して ECU に実装する必要があります。そのため、ここで重要な課題となるのは、適切な開発手法や開発プロセスを導入することにより多数のトランスミッションバリエーションに対処しながら、ISO 26262 およびその他の安全関連規格を遵守し、さらには、新たな機能的要件に対応するソリューションを迅速かつ効率的に量産環境に組み込めるようにするという事です。もちろん、決定的な要素である品質を落とすわけにはいきません。

開発手法および開発プロセス

組込みシステム向けのソフトウェア開発では多くの場合、すべての要件を満たし、かつできる限り迅速にソフトウェアを用意できるようにするため、モデルベース開発手法が利用されています。これは特に自動車業界では一般的です。ここでは、どのような開発手法を用いるかによって、要件定義からソフトウェアリリースまでの開発プロセス全体に影響が生じます。モデルベース開発の利点を十分に生かすには、すべてのフェーズで一貫した手法を利用できる総合的な開発コンセプトが必要です。モデルベース開発によって達成できる主な目標には、品質の改善、開発期間の短縮、および工程全体の自動化の3つがあります。また、この手法では、安全関連機能のプログラミングをモデルベースで行うことも必要です。この場合、ツールやプロセスを多数の厳格な規格に準拠するよう設定し、かつ極めて低いエラー発生率を実現しなければならないため、ツールやプロセスに求められる要件は厳しいものとなります。また、効率性を向上させるには、モデルの再利用、ツールチェーン間の連携、および目的に応じたバリエーション管理が必要です。最終的な目標は、これらを通じてより多くのタスクを開発の初期段階にフロントローディングし、エラーを早期に検出することで、開発時間を短縮することです。

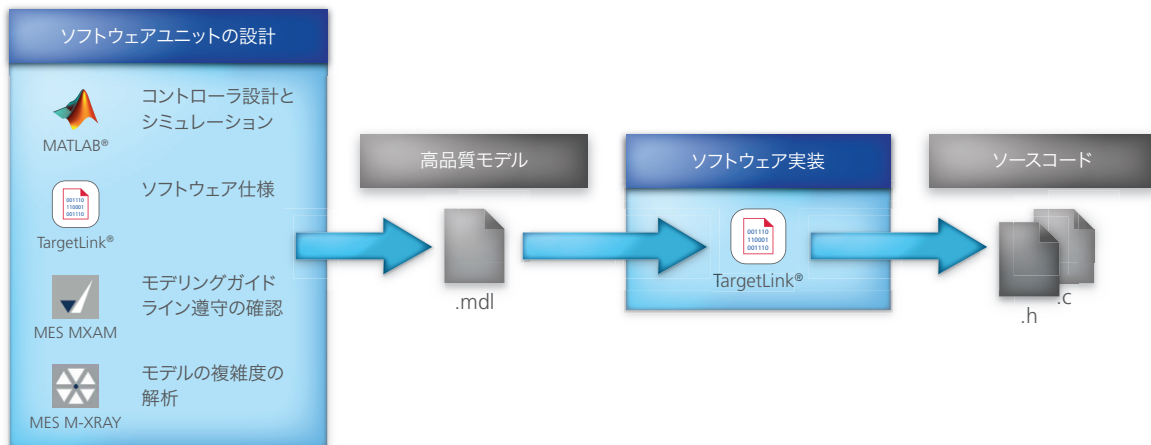
ツールチェーン

モデルベースのソフトウェア開発で最も重要なツールは、モデリングプラットフォームと量産コード生成ツールです。ZF 社で

>>

自 動車業界は、技術革新を常に行い、より新しい機能をより短期間で市場に投入しなければいけないという重圧を絶えず受けています。そのため、現在の市販車ではますます多くのバージョンやバリエーションが生産されています。この傾向はトランスミッション開発の分野でも同様です。トルクコンバータ式

オートマチックトランスミッションにおいてもますます多くのギヤが搭載されるようになってきており、ハイブリッドドライブまでに及ぶ広範な用途をカバーできるよう十分なテストを重ねる必要性も生じています。また、これらのテストに使用するソフトウェア開発分野にも、この多様なバージョンと新しいトランスミッションの機能によ



モデルベース開発に対応したツールチェーンおよびワークフロー

は、プロジェクトのほぼすべてにおいて、モデリングプラットフォームとして MathWorks® 社の MATLAB®/Simulink®, コード生成ツールとして dSPACE TargetLink® を使用しています。モジュールテストを実行する際は、TargetLink に統合されているシミュレーションコンセプトおよび Model Engineering Solutions GmbH (MES) 社の MTest ツールを使用します。また、モデリングガイドラインとの適合性チェックを自動的に実行したり、モデルの複雑さを解析したりする場合は、MES 社の MXAM および MXRAY ツールを使用します。このように、ZF 社では複数のツールを連携させることにより高品質

のモデルを生成できるようにしています。さらに、モデルの解析およびテストや、設定およびバリエーションの管理といったプロセスをサポートするために、その他のツールも積極的に活用しています。上記のツールチェーンは ZF 社の環境全体で使用されており、2008 年の導入以来、TargetLink で生成された量産コードが多数の製品で使用されています。

安全関連機能の開発

一般に、モデルベースで作成される機能は品質管理機能（従来のツールによる妥当性確認）だけではありません。同じことが安全関連機能にも当てはまります。量産

コード生成ツールである TargetLink の大きな利点の 1 つは、TÜV SÜD 社（ドイツの国際認証機関）による ISO 26262 規格の認定を受けていることです。そのため、たとえば ASIL-D までの安全関連システムを開発する場合、既にコードは規格に準拠していることになり、手間のかかるコードの見直しは一切不要になります。さらに、dSPACE ではリファレンスワークフローを提供することで、企業が安全関連システムにおけるモデルベースでのソフトウェア開発手法を社内の標準的なプロセスに容易に組み込めるようにしています。ZF 社では、プロジェクトに依存しない手法を確立したり、プロジェクト固有の手法

バリエーション処理のさまざまな方法



を特定したりするため、広く普及した規格に基づいた独自のモデリングガイドラインを使用しています。これにより、ISO 26262 規格で推奨される手法や項目をモデリングの段階から考慮することができます。

さまざまな方法でバリエーションを処理

ZF 社では、さまざまな段階で持ち越された未使用コードなどを複数のバリエーションで可能な限り多く活用できるようにするため、多様なバリエーション処理を実践しています。コード生成ツールである TargetLink の機能は、従来のプログラミングの機能に非常によく似ていますが、モデルベースの開発手法では、これらのコードを追加の手法によって拡張します。まず、コードを選択的に生成できるモデルバリエーションを使用して複数のバリエーションを処理する方法があります。ここでは、モデルの基礎部分は常に複数のサブモジュールで構成される全体モデルとして形成されます。モジュールの大半は、バリエーションが異なる場合でも同じであり、それはつまり、各モジュールは一度しか存在しないことを意味します。ただし、バリエーションに依存するモジュールについては、バリエーションごとに 1 つのモジュールが存在するように構成します。全体モデルが構築されると、選択したバリエーションに応じて関連するモジュールが使用されます。このようなバリエーション処理は、全体モデルのセットアップの際に使用されますが、いったん全体モデルが完成すると、異なるバリエーションを使用することはできなくなります。一方で、機能バリエーションスイッチやプリプロセッサスイッチを使用して複数のバリエーションを処理する方法もあります。機能バリエーションスイッチを使用すると、TargetLink からバリエーション固有のコード部分を生成することができます。この際、選択したバリエーションに無関係のソフトウェア部分は生成されません。これに対し、プリプロセッサスイッチを使用すると、従来のソフトウェア開発で得られるような効果を実現できます。この場合、TargetLink で生成された量産コードにはすべてのバリエーションが含まれており、コードをコンパイルする際はバリエーション固有の検証が行われます。また、TargetLink では、ソフトウェアアプリケーションで動的に制御するタイプのデータバリエーションを処理することも可能です。



SYNECT は、モデルベース開発専用開発されたデータ管理および連携用ソフトウェアであり、モデルや信号、パラメータ、依存関係、バージョン、バリエーション、さらには基礎的な要件の処理に活用できます。

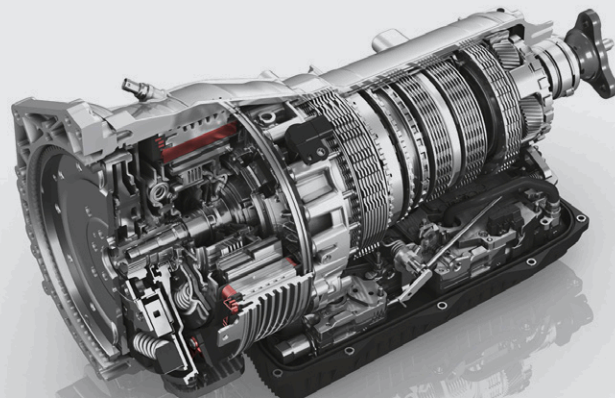
今後の展望：機能管理の活用

多くの企業が、さまざまな顧客から求められる要件の複雑化に対応し、増え続ける機能バリエーションをより適切に管理するため、将来的に機能管理システムを活用することを検討しています。このシステムでは、さまざまな機能モデルを抽出して生成したパラメータ設定セットを TargetLink モデルまたはコードで使用することができます。ただし、これには、プロジェクト全体と個別コンポーネントの両方に機能モデルを使用するマルチステージへのアプローチが必要です。このアプローチは非常に複雑であることが見込まれ、かつ高度なトレーサビリティの実現も必要となります。その

ため、多くの企業の開発部門では、dSPACE SYNECT® などのデータ管理システムを自社の要件管理システムと連携させ、自社のツールチェーンやプロセスに統合する可能性を模索しています。

ZF Friedrichshafen 社のご厚意により寄稿

8速プラグインハイブリッドドライブは、8つあるZF社のHPシリーズのうちの1つのバリエーションであり、TargetLinkで量産コードの生成を行っています。



出典：© ZF 社