

# AUTOSAR in the Development


開発プロセスへの AUTOSAR 適用事例

モデルベースによる AUTOSAR 準拠の制御ロジック開発を量産プロジ



# Process

プロジェクトに導入する手順 (Daimler 社)



AUTOSAR 規格は、ソフトウェアアーキテクチャ、ソフトウェアインターフェース、および交換フォーマットを定義して、ECU サプライヤに依存しない制御ロジックライブラリの構築を可能にします。Daimler 社は、制御ロジックライブラリのソフトウェアモジュールを標準化する取り組みの一環として、快適機能と車両インテリア機能の分野への AUTOSAR の導入を進めています。このプロセスでは、AUTOSAR ソフトウェアコンポーネントのモデリングとコード生成に、自動コード生成ツールの TargetLink が使用されています。

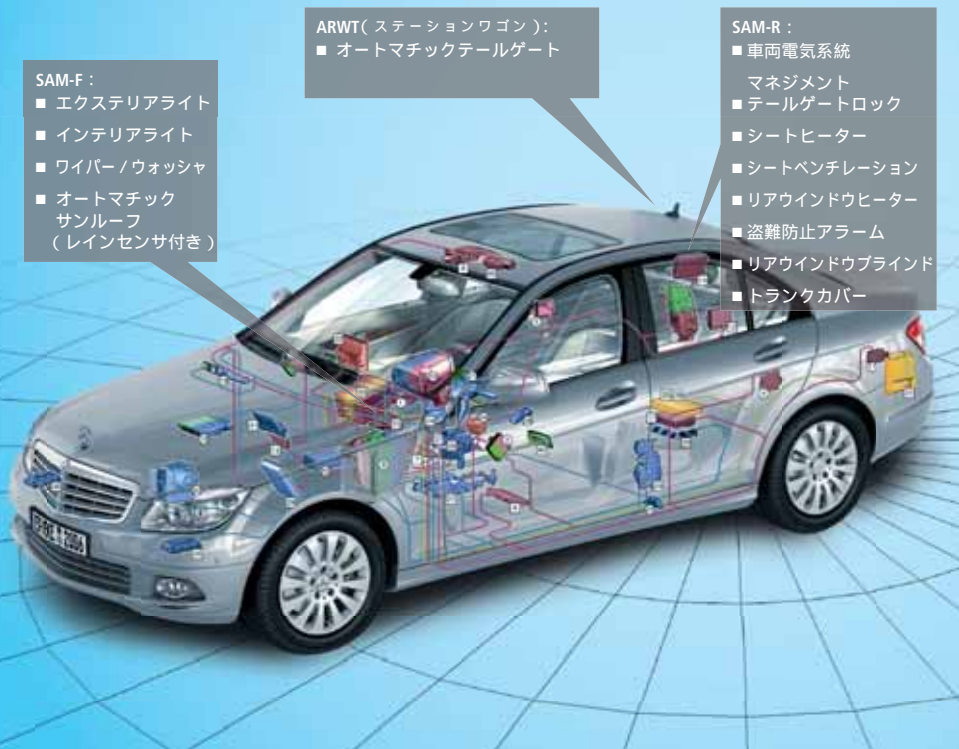


図1：モデリングおよび量産コードの自動生成により設計されたCクラス(204シリーズ)の快適機能(SAM：セントラルボディコンピュータ、ARWT：オートマチックテールゲート)

Daimler社は、乗用車の開発でモデルベース設計と量産コードの自動生成を行なっています。この事例では、どのようにしてモデルベースの開発プロセスを段階的にAUTOSAR準拠のアーキテクチャ対応へと移行させたか、およびこれまでに得られた経験についてご報告します。

#### 車両インテリア機能のモデルベースの開発

Daimler社はすでに何年も前から、車両制御ロジック、特に快適機能とインテリア機能の開発にはモデリング手法を採用しています。車両インテリア機能に関して、モデルベース開発の中心となるのは、セントラルボディコンピュータの主要制御ロジックです。これらの制御ロジックのひとつひとつには、さらに複数の制御ロジックが組み込まれています。最新のCクラス(204シリーズ)の開発では、モデルベース手法を大幅に取り入れました(図1)。モデルベース手法は、現在量産開発が進められているシリーズにも体系的に適用さ

れており、適用される機能の数は着実に増えています。

今回開発した機能は、様々なECUで使用されるアプリケーションソフトウェアの共通部分、もしくは競合製品と差別化をはかるために既存のソフトウェアに追加される付加価値機能の部分です。このビジネスモデルにおいては、サプライヤは当社が開発した制御モデルとテスト仕様書を受け取り、モデルに対応したソフトウェアを開発し、それをECUに実装する役割を担います(図2)。モデルベースの開発により、自動車メーカーには以下のメリットがあります。

- シミュレーションにより、設計の初期段階で詳細な検証が可能
- ECUがまだ未完成のときでも、車両上でのラピッドプロトタイピングにより、制御ロジックの再現、検証が可能
- 自動生成コードの直接実装

#### インテリア-CAN

1. 電子イグニッションスイッチ
2. ドアECU(フロント左)
3. ドアECU(フロント右)
4. ドアECU(リア左)
5. ドアECU(リア右)
6. シート調整(運転席)
7. シート調整(助手席)
8. ルーフコントロール
9. SAM/SRB(フロント)
10. SAM/SRB(リア)
11. タイアプレッシャーコントロール(米国)
12. トレーラ接続ユニット
13. パークトロンニックシステム
14. キーレスエントリー
15. オートマチックエアコン
16. インストルメントクラスタ
17. マルチファンクションECU
30. COMAND
37. 重量検知システム(米国)
40. パノラマサンルーフ

#### シャーシ-CAN

1. 電子イグニッションスイッチ
9. SAM/SRB(フロント)
16. インストルメントクラスタ
18. エンジン電子制御ユニット(ガソリン)
19. エンジン電子制御ユニット(ディーゼル)
20. 車両安定化プログラム(ESP)
21. ステアリングコラムジャケットモジュール
22. エアバッグARAMIS
23. リバースベルトテンション(フロント左)
24. リバースベルトテンション(フロント右)
41. アダプティブダンピングシステム

#### ピークルダイナミクス-CAN

20. 車両安定化プログラム(ESP)
25. センサクラスタ

#### ドライブトレイン-CAN

18. エンジン電子制御ユニット(ガソリン)
19. エンジン電子制御ユニット(ディーゼル)

26. 電子制御変速モジュール
27. 全的に統合されたトランスミッション制御
28. 燃料タンクECU(ガソリン)
29. トランスミッション電子制御ユニット

#### LIN-Bus

2. ドアECU(フロント左)
3. ドアECU(フロント右)
9. SAM/SRB(フロント)
10. SAM/SRB(リア)
18. エンジン電子制御ユニット(ガソリン)
19. エンジン電子制御ユニット(ディーゼル)
21. ステアリングコラムジャケットモジュール

#### テレマティクス-CAN

30. COMAND
33. セントラルディスプレイ
34. COMANDコントローラ

#### プライベート-Bus

15. オートマチックエアコン
42. オートマチックエアコン(リア)

#### MOST-Ring

30. COMAND
31. オーディオアンプ
32. デジタルラジオ(DAB)
39. SIRIUS衛星ラジオ(米国)

#### 診断-CAN

9. SAM/SRB(フロント)
38. 緊急呼出しシステム(米国)

#### フロントエリア-CAN

9. SAM/SRB(フロント)
35. キセノンECU
- ヘッドライトマスター
36. キセノンECU
- ヘッドライトスレーブ

- サプライヤに依存しない制御ロジックの開発と開発した制御ロジックに基づいたさらなる開発が可能
- 制御ロジックの再利用
- 知的財産権の保護

#### ソフトウェアアーキテクチャへの組込み

現状では制御ロジックがモデリングベースで作成された場合でも、サプライヤがその制御ロジックをECUへ組み込む際には、多くの手作業が必要となります。また、自動車メーカーがベーシックソフトウェアの通信部分の仕様を指定していたとしても、全体の作業量は各サプライヤが使用するソフトウェアアーキテクチャに大きく影響されます。場合によっては、ソフトウェアアーキテクチャを調整するか、特別に拡張する必要があります。完全に標準化されているソフトウェアアーキテクチャが存在しないため、何度も特定のサプライヤとの調整会議を行わなくてはならないこともあります。こうした調整が必要なのは、ソフトウェア

アーキテクチャだけではなく、自動車メーカーとECUサプライヤは、たとえば制御ロジックについてのインターフェースリストや、バス信号へのアプリケーション信号のマッピングなど、統合する制御ロジックに関するメタデータの記述を共同で定義しなくてはなりません。すなわち、モデルベースの開発を広範囲にかつプロセスセーフに行なうためには、サプライヤに依存しない統一されたソフトウェアアーキテクチャと、標準化されたメタデータの記述が必要条件となります。

#### AUTOSAR ソフトウェアアーキテクチャ

AUTOSAR 規格では、ECU のためのソフトウェアアーキテクチャ、組み手法、およびそれらに必要な交換フォーマットが定義されています。すなわち AUTOSAR は、上で述べたモデルベースの制御ロジックをプロセスセーフに統合するために必要なほとんどの要件に対応しています。AUTOSAR では、1 個の ECU のアプリケーションソフ

トウェアをいくつかのソフトウェアコンポーネント(SWC)に分割し、これらのコンポーネント間の通信にはミドルウェア(ランタイム環境すなわち RTE)を使用します。SWC によりソフトウェアをカプセル化してデータ型定義を与えることにより、データ交換は厳密に定義されたインターフェースを通じてのみ可能になります。

1 個の ECU への組み込みにはマッピングを 2 回行う必要があります。まず SWC インスタンスの ECU へのマッピング、続いて ECU 間で通信するために、データ要素のネットワーク信号へのマッピングを行います。

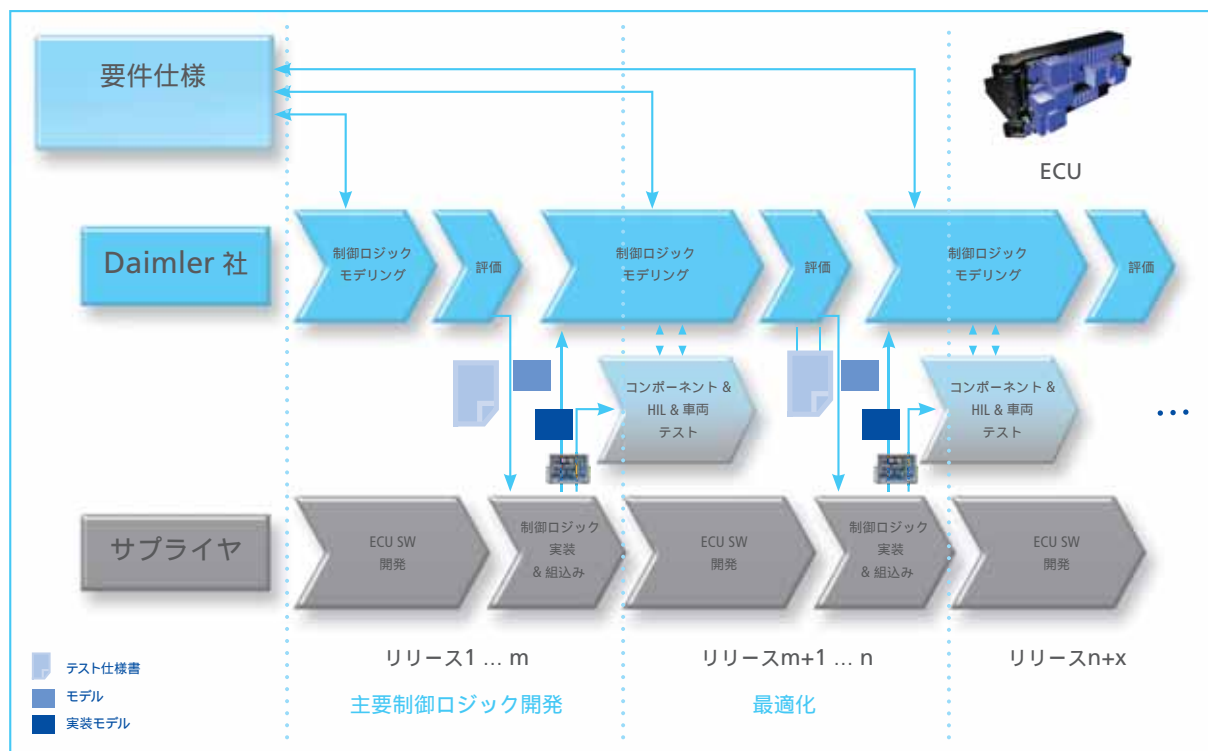
#### AUTOSAR 導入のための戦略

次世代のインテリア ECU において、Daimler 社は AUTOSAR アーキテクチャに向けた第一歩を踏み出しています。導入は、AUTOSAR ソフトウェアアーキテクチャの最上位レイヤから始め、下位レイヤへと順次進めます。つまり、

AUTOSAR 導入はアプリケーションソフトウェアコンポーネント(SWC)と RTE から開始します(図 3)。ここではソフトウェアアーキテクチャは、アプリケーション部と基本ソフトウェア部へと体系的に分割し、その間の通信は特有のインターフェースを介して行います。このインターフェースは AUTOSAR 規格に基づいて定義しますが、規格の定義自体にまだ不十分な部分があれば、逆により厳密に定義されている部分もあります。今回は第一段階として、標準ソフトウェアとして実績のある Daimler Standard Core をベースにし、そこにいくつかの AUTOSAR ソフトウェアサービス(NVRAM データのメモリ管理など)を追加します。

初期段階では、このように開発された ECU も、従来の手法で開発された ECU とネットワーク上は互換性を保つことができます。すなわち、AUTOSAR テクノロジーは段階的に導入することが可能です。

図 2 : 典型的な反復型開発サイクル



### 動作レベルとアーキテクチャレベルでのモデリング

AUTOSAR では、モデリングは 2 つのレベルで実施されます。動作レベルでは、制御ロジックの動作をモデル化します。もう一方のアーキテクチャレベルでは、SWC のインターフェースとその接続を指定された形式で記述しなくてはなりません。

トップダウン方式で新しい車両制御機能を開発する場合、まずその制御ロジックをいくつかの SWC に分割し、それからアーキテクチャレベルでインターフェースを定義すると便利です(図 4)。その後、生成された SWC の動作をモデル化します。

既存の制御モデルについては、ボトムアップ方式でモデルインターフェースから SWC の記述を生成します。その後、生成された SWC を、アーキテクチャレベルで相互に接続します。段階的に導入する場合は、車両全体の設計を完全にトップダウン方式で行うことはできません。また ECU レベルでは、すべての制御ロジックを最初

からモデルとして利用できるわけではありません。したがってここでは中間的な戦略が有効であり、各 ECU において以下のステップを実行します。

- SWC の記述を既存の動作モデルから導出
- 新しい車両制御機能については、アーキテクチャレベルで SWC ごとにインターフェースを最初に定義
- センサ / アクチュエータ用 SWC を、定義された規則に基づいて自動生成

生成された SWC はアーキテクチャレベルでコンポジションとしてグループ化され、相互にネットワーク化されます。接続されずに残されたポートは、コンポジションのポートとして後で外部と接続され、ECU の通信インターフェースとなります。ポート経由で参照されたデータ要素は、通信マトリクスによって ECU 専用の信号にマッピングされます。この方法により、ECU の SWC 構造を適正なコストで作成することが可能になります。

次に、これらの作業をより詳細に説明します。

### AUTOSAR インターフェースの定義

ソフトウェアコンポーネントの開発では、SWC 開発者が SWC ポートを定義するときに使用可能なインターフェースおよびデータ型の定義を行なうための、共有データベース、すなわち共通オブジェクトプール(COP)が基盤となります。このプールは通信マトリクスの信号定義を基にして作成されており、常に実装に使用する定義と内容を一致させる必要があります(図 8)。

インターフェース定義を転換する場合の規則として、1 個のインターフェースが 1 個のデータ要素を持ち、それぞれの通信マトリクス信号に対応するように作成します。これにより、ライブラリ内にある名前を指定して、それに該当するインターフェースを見つけるのも容易となり、現在の通信マトリクス信号の世界から AUTOSAR の世界へとスムーズに移行できます。既存の

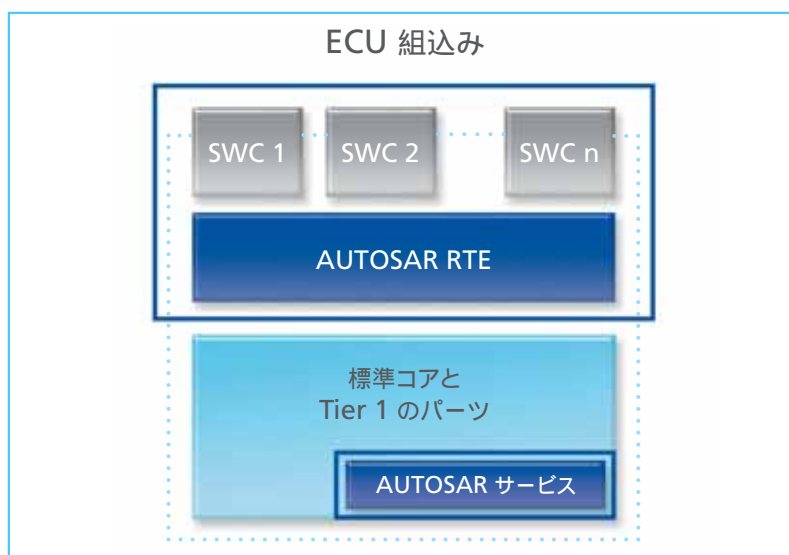


図 3 : AUTOSAR を段階的に導入する場合、まずアプリケーションソフトウェア(SWC)とミドルウェア(RTE)を AUTOSAR に準拠させて開発します。ベーシックソフトウェアとして Daimler Standard Core を引き続き使用し、これを選定された AUTOSAR ソフトウェアサービスによって拡張します。



## 用語解説

コンポジション コンポジションにより複数の SWC がグループ化され、上位コンポーネントと接続

DBC ファイル データベース格納ファイル。CAN 用の通信マトリクス記述ファイル

通信マトリクス ECU 間の信号 / データ交換を記述

NVRAM 診断情報などを格納する不揮発性のランダムアクセスメモリ

Runnable AUTOSAR 準拠の SWC 内で実行可能な要素。制御ロジックに相当

システムアーキテクチャツール 複雑な ECU アーキテクチャを計画、統合のために使用。AUTOSAR ファイルのインポート、エクスポート、変更が可能

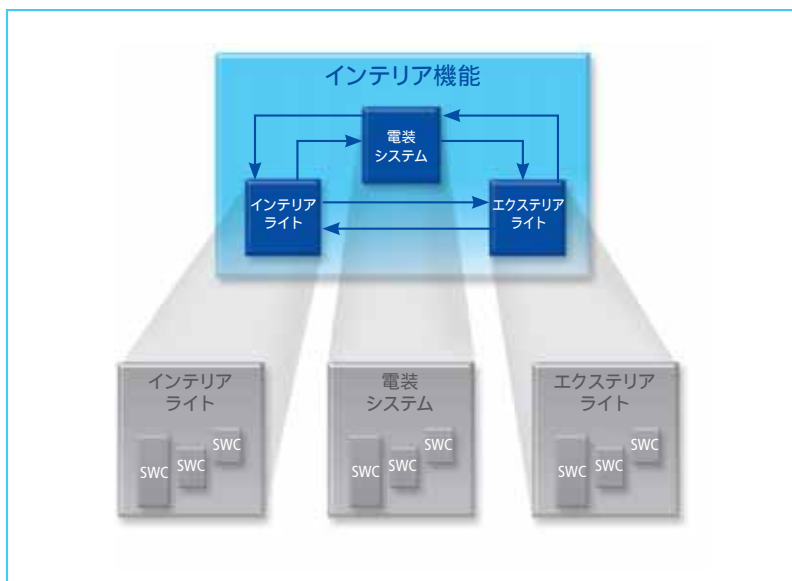


図 4 : トップダウン方式では、最初に制御ロジックをいくつかの SWC に分割し、続いてそれらのインターフェースをアーキテクチャレベルで定義します。

定義の転換ではなく、全く新しくインターフェースを作成する場合は、複数のデータ要素がひとつのインターフェースに収まるように AUTOSAR の構造オプションを使用します。この構造オプションは既存のインターフェースの転換には使用できません。SWC 間の互換性を確保するには、モデリング規則に従って、常に、プールにあるインターフェースのみを使用するようにします。そのためには、必要に応じて、プールにインターフェースを追加する必要があります。

### トップダウン方式による新たな車両制御機能の作成

新しい車両制御機能をモデリングする場合は、まずアーキテクチャレベルにおいて、上記の COP データベース内に SWC インターフェースを記述することから始めます。この作業には、インターフェースによってデータ型が定義されたポートの作成も含まれます。また、Runnable とその SWC インターフェースのサブセットも定義します。RTE イベントを使用して Runnable のサイクル時間を定義し、さらに必要に応じて、新しい信号の受信などによるイベントトリガも定義します。最後に AUTOSAR の SWC 記述を生成します。

この AUTOSAR の記述は動作モデリングのベースとなります。SWC 記述をデータディクショナリにインポートし、動作制御ロジックをモデリングするための出発点となるフレームモデルを生成します。フレームモデルでは、TargetLink AUTOSAR ブロックを使用することにより、関連する AUTOSAR インターフェースと Runnable を作成します( 図 5 )、続いてフレーム中に、標準ブロックを組み込み、実際のアルゴリズムをモデル化します。データベースに対する変更は、常にデータベース内で行います。

ボトムアップ方式による既存モデルの移行  
既存の動作モデルを移行する際、ポートを導出してデータ型を指定するためにインターフェース定義が使用されます。モデリング規則として、入出力変数に通信マトリクス信号の名義を使用することを取り決めていたため、90% のモデル移行作業を自動化することができました。最初にデータディクショナリ内に SWC を作成し、それから適切な AUTOSAR ブロックをモデルに組み込みました( 図 6 )、最後に、コード生成と同時に自動生成される XML ファイル( SWC 記述ファイル)



図 5 : TargetLink で生成されたフレームモデル

を、SWC データベースにロードします。これによってインターフェースをデータベース内に保存することができます。モデル化された制御ロジック(動作モデル)は、まったく変更を加えなくても完全に移行できましたが、モデルインターフェースだけは規格に適合させるための手作業が必要でした(図7)。

#### AUTOSAR ツールチェーン

AUTOSAR 準拠の車両制御ロジックを開発するために、既存のツールチェーンを拡張しました(図8)。AUTOSAR インターフェースと SWC 記述を格納するための共通オブジェクトプールは、開発環境における主要な要素です。この共通オブジェクトプールは、通信マトリクスを作成するために使われていた既存のデータベースアプリケーションを拡張することにより、実現しました。このプールには、AUTOSAR モデリング要素が含まれておりさらにパラメータと NVRAM 変数を記述し、値を設定することもできます。AUTOSAR インターフェースは通信マトリクスと類似しているため、通信マトリクス内のデータから容易に生成し、その後アップデートすることも可能です。これまでの通信マトリクス開発では、リリースするまで何度も試行を重ねてきましたが、その成果を引き続き活かすことができます。

SWC 記述は、AUTOSAR の交換フォーマットを介してインポートおよびエクスポートできます。同時に、データディクショナリおよびシステムアーキテクチャツールへの接続もサポートしています。AUTOSAR 準拠の制御モデルのモデルベース開発は、開発環境に組み込まれている MIL (Model-in-the-loop) と SIL (Software-in-the-Loop) のテストによってさらにサポートされます。最後に、AUTOSAR 準拠の量産コードを、一貫性のある AUTOSAR 準拠の SWC 記述ファイルと合わせて生成します。SWC をネットワーク化するには、システムアーキテクチャツールを使用します。大部分の操作は、専用のグラフィックエディタによって行います。ビジュアルな画面が、単なるフォームベースのデータベースアプリケーションよりも使い勝手に優れているためです。続いてこのシステムアーキテクチャツールを使用して、非ローカルのデータ要素をネットワーク信号にマッピングします。こうして作成された AUTOSAR システムテンプレートは、サプライヤが ECU を組込むために使用します。

#### まとめと展望

AUTOSAR は、モデルベースの制御ロジック開発のために記述フォーマットとインターフェースを標準化するという長年の要望に応えるものです。

自動車メーカーが ECU サプライヤへ制御モデルを手渡すときに、型安全な AUTOSAR 準拠の記述を採用することで、個別に開発される制御ロジックモデルの一貫性を開発プロセスの極めて早い段階で保証することができます。そのためサプライヤは制御ロジックの組込み作業を、はるかに効率的に行えるようになります。また、自動車メーカーとサプライヤの双方が AUTOSAR によって標準化された用語を使用することで、ソフトウェアアーキテクチャを議論するための調整会議がいっそう生産的なものになります。

現時点では、AUTOSAR 準拠の機能ネットワーク開発において、2種類の開発ツール(動作モデリング用ツールとインターフェース記述ツール)を使用する必要があり、また、各システムは、管理しやすくかつ論理的に使いやすいという特性を持ったソフトウェアコンポーネントへ分割する必要があります。将来的には、異なるベンダーから提供されるさまざまなモデリングツール間の移行を、より効率的に自由に行えるようにする必要があります。さらに、現在の AUTOSAR 開発環境は、アーキテクチャモデリングおよびシステム統合用のツールと、コード自動生成制御ロジックを内蔵した動作モデリング用のツールとに分割されていますが、これはツールのドメインに関係しており、ツールに依存しないシステムモデリングとリソースの最適化と



図6: TargetLink の AUTOSAR ブロックを使って制御ロジックをモデル化



図7: AUTOSAR への移行後に適切なインターフェースブロックを使って Runnable をモデル化

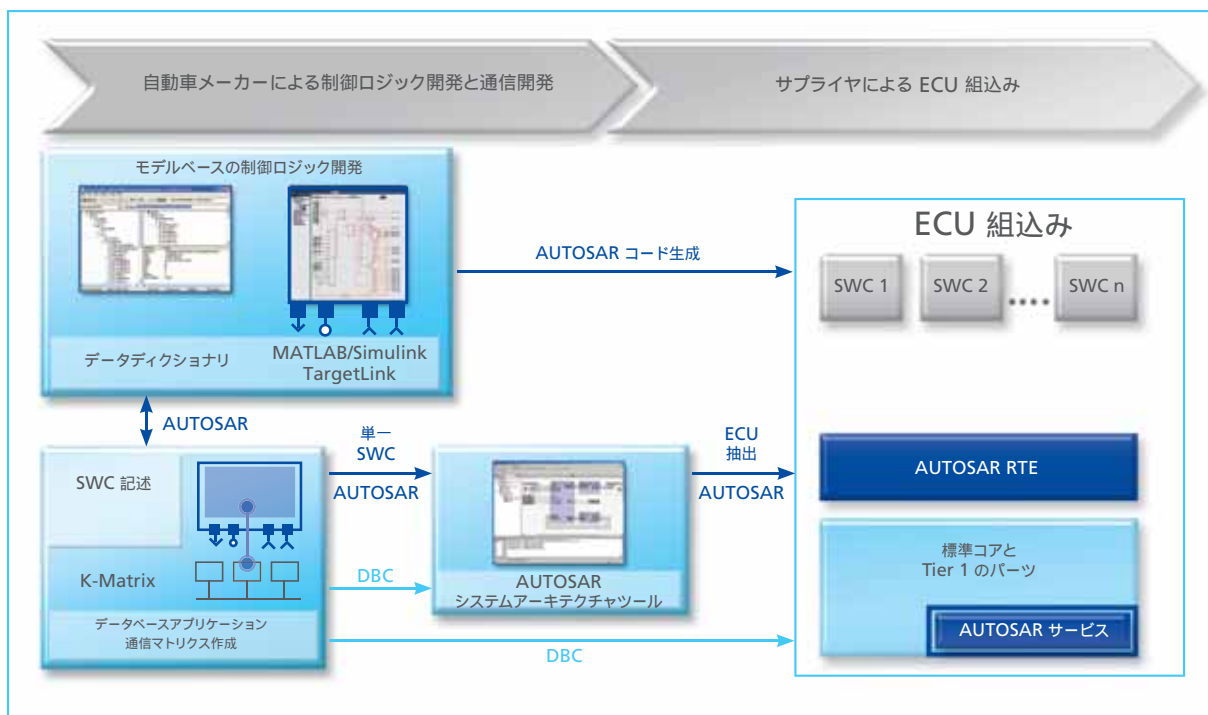


図 8 : AUTOSAR ソフトウェアコンポーネント (SWC) を生成するための AUTOSAR ツール環境。ツール間のデータ交換は AUTOSAR XML と DBC ファイルに基づく

いう点では制限事項となります。つまり、アーキテクチャと制御ロジック設計を最初の段階で慎重に決定する必要があります。AUTOSAR の記述対象が非常に広範囲に及ぶ一方で、個々のツールがまだ完全には対応していないため、開発の際は適用する規格のサブセットに対して、適切なアプリケーションを選択・判断することから始めなくてはなりません。

すでに見てきたように、AUTOSAR を開発プロセスに漸進的に導入できることは実証されました。段階的なアプローチで AUTOSAR を導入することによって、既存の制御モデルおよび実績のあるプロセスやツールチェーンを継続して使用しながら、徐々に最適化することが可能です。

今回の成功により、快適機能とインテリア機能の開発に制御モデルを使用することは、Daimler 社における標準手法となりました。このようなプロジェクトを通じて得られた経験の大部分は、AUTOSAR 準拠の開発でも活用できます。現在の手作業を、将来は自動化および標準化されたプロセスによって置き換えることができるようになるでしょう。

AUTOSAR 規格の定義はまだ不完全です。AUTOSAR への投資が将来確実に実を結ぶように、今後も標準化グループとツールメーカーとの密接な協力が必要です。■

Christian Dziobek, Dr. Florian Wohlgenuth  
Mercedes-Benz Cars Development  
Dr. Thomas Ringle,  
Group Research & Advanced Engineering  
Daimler AG, ドイツ

## まとめ

- Daimler 社はインテリア ECU の量産プロジェクトに AUTOSAR を導入
- 既存のツールとプロセスを AUTOSAR 規格対応に向けて拡張
- AUTOSAR 準拠のコード生成ツールを使用することにより既存のモデルを再利用