

SimulinkモデルによるAUTOSAR適用10箇条

David Jaffry, Holly Keener

目次

用語	2
はじめに.....	3
AUTOSAR向けSimulinkサポートの概要.....	4
AUTOSAR適用の際のベストプラクティス.....	6
関連情報.....	17

用語

AUTOSAR関連の説明では多くの略語が使用されます。このガイドで使用されている用語の簡単なリストを次に示します。

略語	意味
AAT	AUTOSARオーサリングツール
API	アプリケーションプログラミングインターフェイス
ARXML	AUTOSAR XML
AUTOSAR	AUTomotive Open System ARchitecture
BSW	ベーシックソフトウェア
CRL	コード置換ライブラリ (Embedded Coderの機能)
IDE	統合開発環境
PIL	プロセッサインザループ
RTE	ランタイム環境
SIL	ソフトウェアインザループ
SWC	ソフトウェアコンポーネント
V&V	検証
XML	拡張マークアップ言語

はじめに

AUTOSAR (AUTomotive Open System ARchitecture) は車両の製造業者、サプライヤーおよびその他の会社 (エレクトロニクス、半導体、ソフトウェア産業) で構成される世界的な開発パートナーシップです。AUTOSAR規格はソフトウェアの標準化、再利用および相互運用性を実現することを目的としています (図1)。

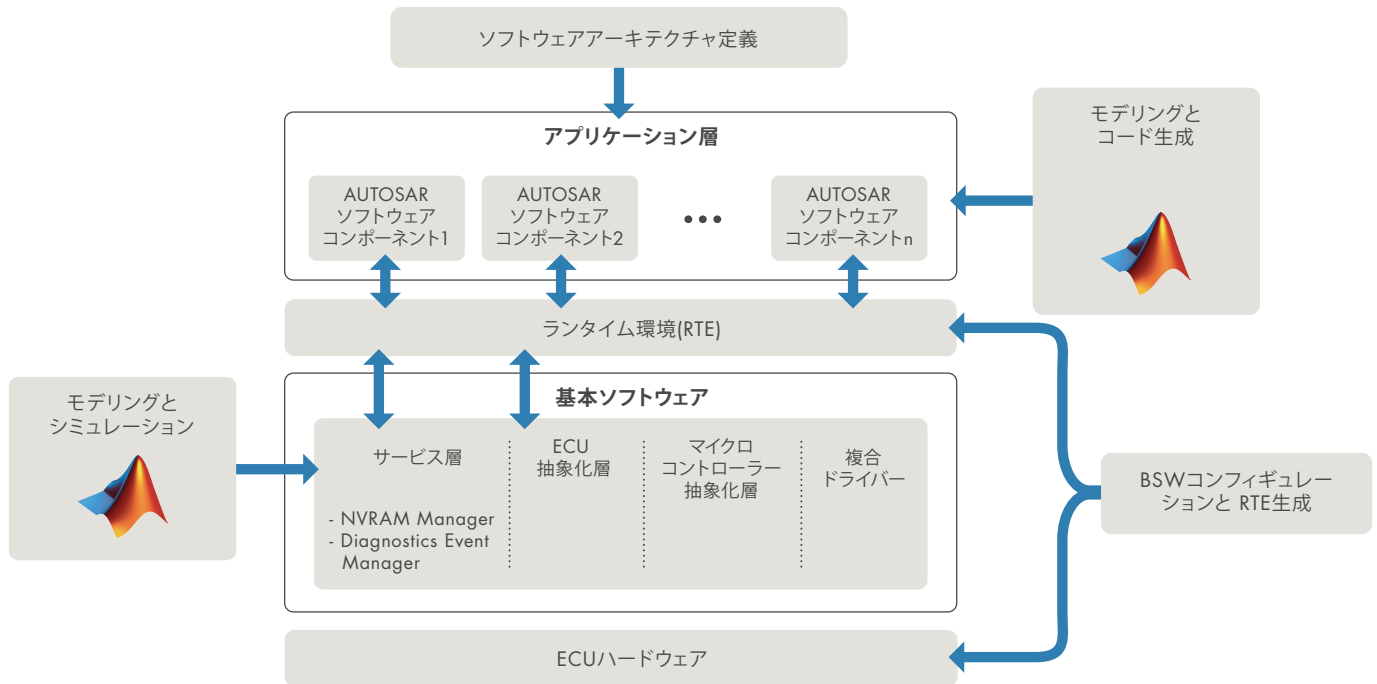


図 1. AUTOSARの階層アーキテクチャ。

MathWorksはAUTOSARプレミアムメンバーであり、この規格の発展に積極的に参加しております。特に自動車のECUにおいてAUTOSAR開発プロセスでモデルベースデザインを使用する方法に注力しています。Simulink®およびAUTOSAR Blockset™では、AUTOSARの機能に加えてワークフローの自動化のためのアプリケーションプログラミングインターフェイス (API) を提供しています。

このガイドは、モデルベースデザインを使用してAUTOSARを適用した多くの実例に基づいて、SimulinkによるAUTOSAR適用に関するベストプラクティスをまとめたものです。これらは、AUTOSARを企業全体で適用するための体系的なアプローチのプロセスを通して、コスト、リスク、問題の発生を最小限に抑えながら最大限の利益を実現しています。ベストプラクティスは、MATLAB®とSimulinkを使用したモデルベースのAUTOSAR適用において、自動車OEMおよびサプライヤーであるお客様の事例に基づくものです。

以下は、Simulinkモデルによる効果的なAUTOSAR適用の際のベストプラクティスのトップ10です。

1. 既存のSimulinkモデルをAUTOSARに移行する戦略を決定する
2. AUTOSARワークフローを1つに絞る
3. データ管理戦略を選択する
4. モデリング標準を確立する
5. シミュレーションしてからコード生成する
6. 量産コード生成機能を使用する
7. Simulink を使用してレガシーコードをAUTOSARに移行する
8. 徹底的に自動化する
9. ISO 26262の導入は事前に計画する
10. 将来を見据えた移行計画

SimulinkによるAUTOSARサポートの概要

まず、Simulink製品ファミリによるAUTOSARサポートについて簡単に説明しましょう。SimulinkとAUTOSAR Blocksetを使用してAUTOSARソフトウェアコンポーネント(SWC) の設計、実装、検証を行うことができます。各AUTOSARソフトウェアコンポーネントには1つ以上のエントリポイント関数を含めることができ、エントリポイント関数はAUTOSAR内でランナブルエンティティとして実装されます。AUTOSAR Blocksetは、SimulinkモデルをAUTOSARにマッピングして、Embedded CoderがAUTOSAR準拠のC/C++コードとAUTOSAR XML (ARXML) ファイルを生成します。

Simulinkモデルの特定の要素を使用することにより、AUTOSARコンセプトを実装します。AUTOSAR ターゲットを使用するようにSimulinkモデルを設定すると、図2に示すように、エディターを使用してAUTOSARの属性を管理し、SimulinkモデルをAUTOSARにマッピングします。

- **AUTOSARコンポーネントの設定**
- **Code Mappings Editor**を使用したコード生成のためのAUTOSAR要素のマッピング
- **AUTOSAR Component Designer**を使用した、量産組み込みプロセッサ用に、AUTOSARブロックを使用したモデルから、読みやすく、コンパクトで高速なCおよびC++コードの生成

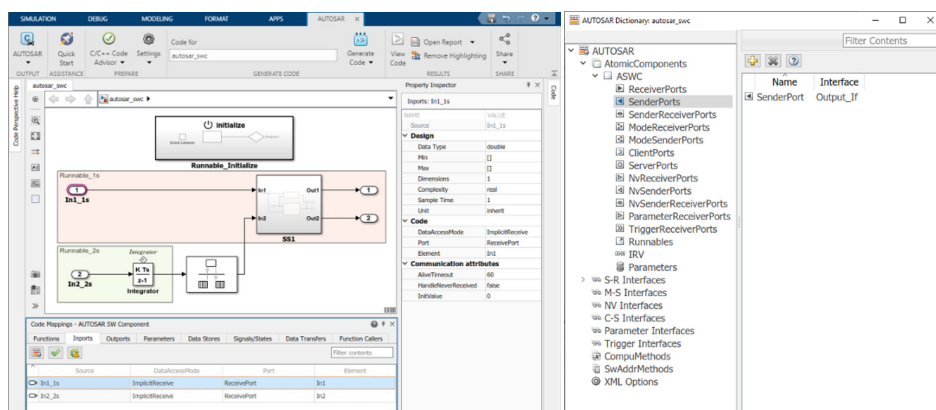


図 2. SimulinkからAUTOSARへのマッピング。

AUTOSAR適用の際には、アプローチとワークフローについて複数の選択肢があり、一般的にはトップダウンアプローチとボトムアップアプローチの2つがあります。(図3)

トップダウンアプローチ

AUTOSARオーサリングツール(AAT)でシステムのアーキテクチャを設計し、ARXMLファイルを生成してSimulinkにインポートして、モデルの外枠を生成します。次にSimulinkで内部動作を設計します。ソフトウェアコンポーネントを検証した後でCコードを生成します。AATがすべてのARXMLファイルを管理します。

ボトムアップアプローチ

Simulinkを使用して、ソフトウェアコンポーネントを作成、開発、検証することから始めます。System Composerをお持ちの場合は、AUTOSARコンポジションとコンポーネントを開発するためのソフトウェアアーキテクチャキャンバスを作成することができます。ベーシックソフトウェアサービスコンポーネントなどのシミュレーションビヘイビアを追加することができます。SWCを開発したら、SimulinkでのテストやAUTOSARランタイム環境への統合のために、ARXMLコンポーネント記述とアルゴリズムCまたはC++コードを生成します。

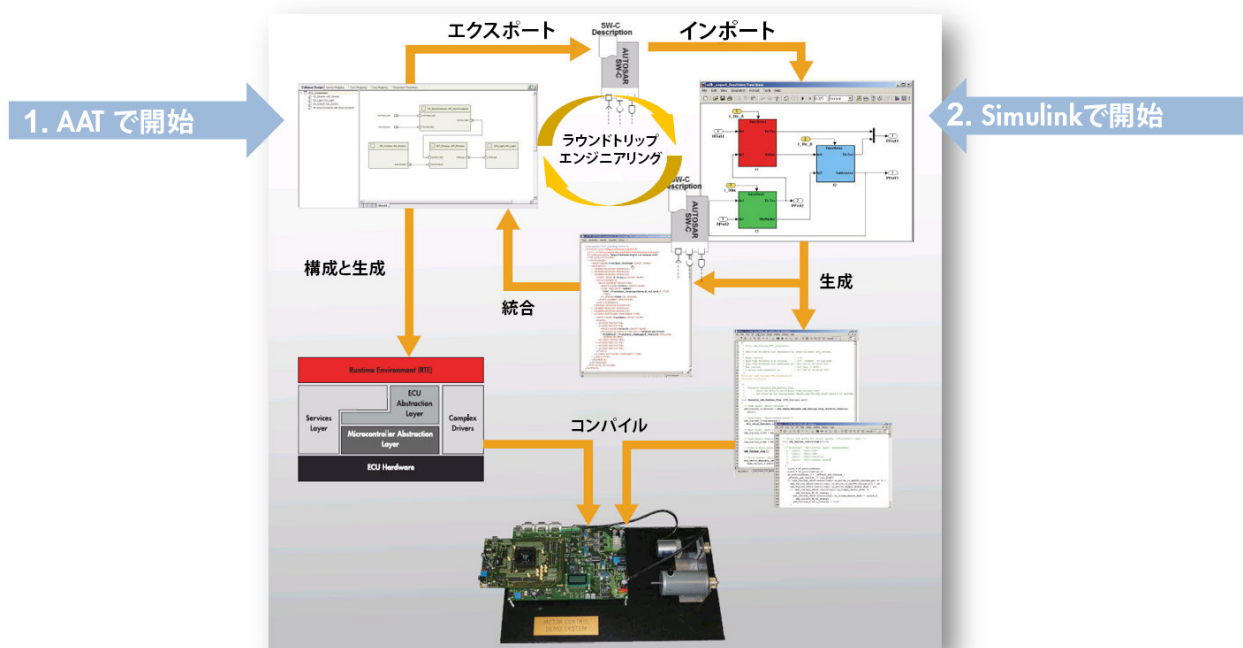


図 3. Simulinkを使用したAUTOSARのワークフロー。トップダウンアプローチはAUTOSARオーサリングツールから始め

ボトムアップアプローチはSimulinkから始めます。

AUTOSAR適用の際のベストプラクティス

1. 既存のSimulinkモデルをAUTOSARに移行する戦略を決定する

一般的なワークフローではAUTOSARを採用する前にSimulinkでソフトウェアアルゴリズムを設計します(図4)。非AUTOSARからAUTOSARへ設計を移行する際に、開発チーム内でアプローチを統一しておくことが重要です。これにより、チーム内で共通のプロセスとツールを使用できるようになります。AUTOSARへの移行の際、次の3つの選択肢があります。

- 白紙状態から
- AUTOSARへの完全移行
- 1つのモデルをAUTOSARと非AUTOSARで共有

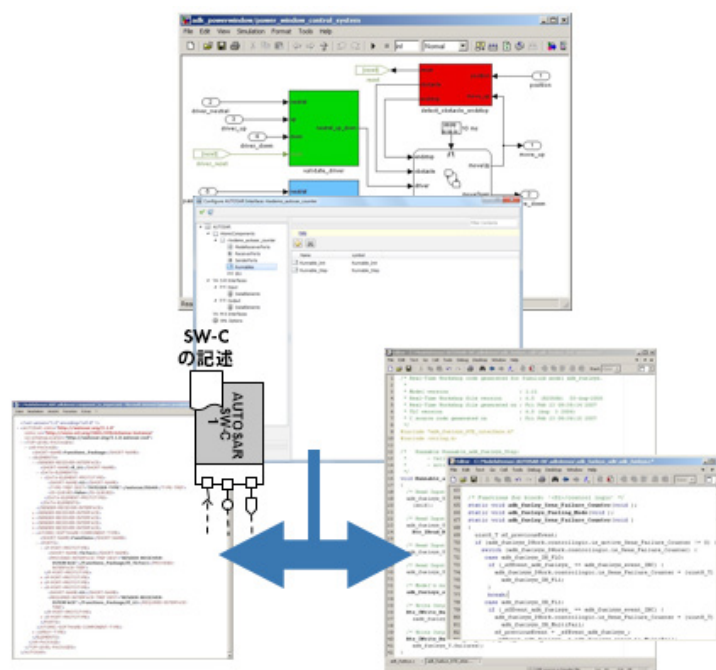


図 4. Simulinkからのコード生成。

白紙状態から

アルゴリズムを大幅に再設計する場合は“白紙状態”から始めるのが最善の方法です。要件に従って設計プロセスを開始し、AUTOSARに適した新規Simulinkモデルを最初から開発します(図5)。この方法ではレガシー設計の制約がないため、AUTOSAR、モデルベースデザインを柔軟かつ最大限に活用できます。ただし、レガシー設計の再利用が最小限にとどめられる場合にのみ、このオプションは有効です。

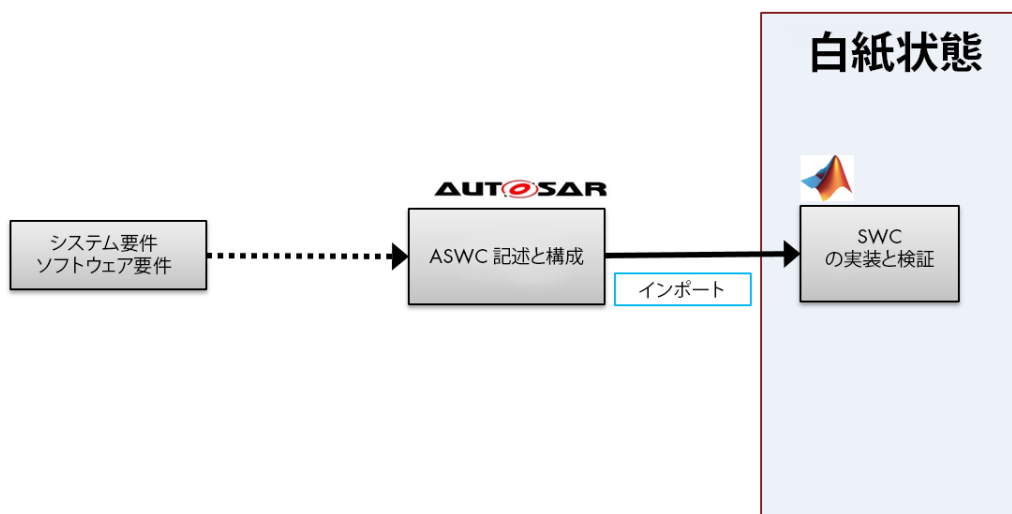


図 5. 白紙状態からの移行。

AUTOSARへの完全移行

既存のSimulinkアルゴリズムを再利用して、ソフトウェアアーキテクチャはAUTOSARで置き換える場合、AUTOSARに最適なモデリングスタイルにSimulinkモデルを変換または移行することが最善の方法です。このアプローチでは、AUTOSAR用の最適なSimulinkモデリングスタイルと機能セットを選択した上で、モデルの移行を計画します(図6)。必要に応じて、Simulink実装を更新し、モデリングスタイルを反映させます。Simulinkモデルの変更の前には、リグレッションテストを実施し、機能的な変化がないことを確認します。この方法は、レガシー実装を再利用しながらモデルベースデザインによるAUTOSARも活用するための、効果的な折衷案です。

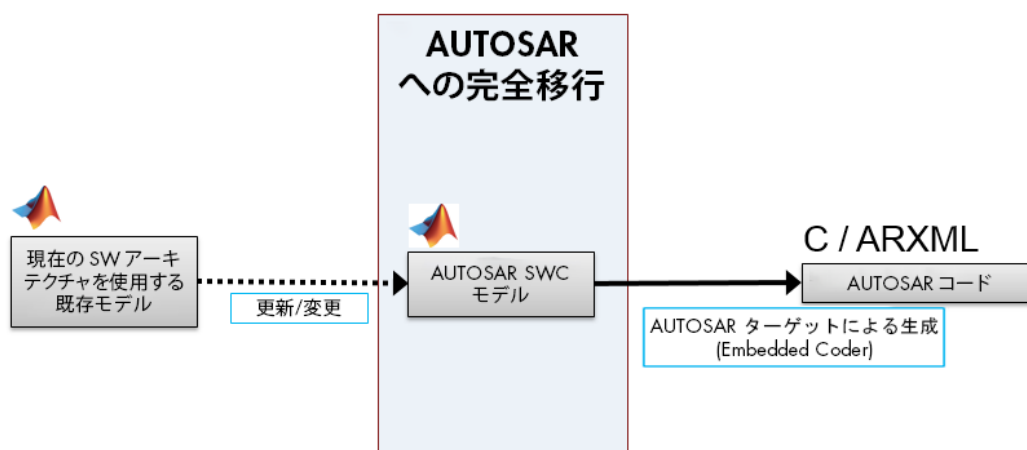


図 6. AUTOSARへの完全移行。

1つのモデルをAUTOSARと非AUTOSARで共有

SimulinkとEmbedded CoderでAUTOSAR適用する主要なメリットの1つは、単一のモデルからAUTOSAR用と非AUTOSARの用途のコードを生成できることです(図7)。この方法は、共通の実装からAUTOSAR用と非AUTOSARのアプリケーションを並行して開発するとき有効です。完全移行の際と同様に、モデリングスタイルと機能セットの選択は一律に行います。そうすることで、設計がAUTOSARで必要となるSimulinkのモデリングスタイルや制約と互換であることを保証します。この方法においても、リグレッションテストは移行プロセスの中で非常に重要です。1つの実装モデルを複数のアーキテクチャで使用する場合、AUTOSARの適用度合いに制限が生じることがあります。しかし、単一のモデルを設計、テストして2つのターゲットアーキテクチャに展開することは、効率面で非常に大きなメリットがあります。

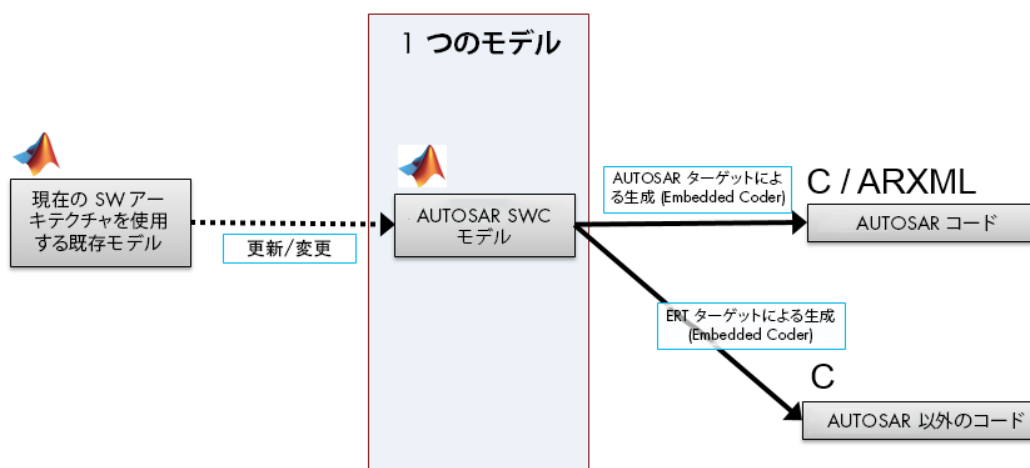


図 7. 1つのモデルによる移行。

いずれの方法でも、移行プロセスの範囲を適切に見定め、小規模かつ典型的なSimulinkモデルのパイロットプロジェクトから始め、より全体的なシステムへとスケールアップしていきます。組み込みシステムは一般的に複数のドメインにまたがります。モデルベースデザインの対象となるモデリングドメインでコンポーネントを明確に表現できることが重要です。

2. AUTOSARワークフローを1つに絞る

AUTOSAR規格にはツールの相互運用性に関する要件が含まれており、これによって効果的なトップダウンおよびボトムアップワークフローが可能になります(図8)。各ワークフローの長所と短所を理解し、AUTOSARを採用する本来の目標に最も適したワークフローを選択します。プロジェクト内に複数のワークフローを混用すると、データや設計の一貫性を保つために不必要な反復作業や確認が必要になり、混乱や、やり直し、無駄が生じる可能性があります。

特にラウンドトリップ(反復開発)では、データのオーナーを明確に1人にしておくことが最善となります。

ワークフローを決めたら、SimulinkモデルにAUTOSAR構成を適用する最も簡単な方法を選択します。トップダウンワークフローでは、AUTOSAR Blocksetの機能で、ARXMLをインポートし、Simulinkモデルを作成もしくは更新して、AUTOSAR構成を適用します。ボトムアップワークフローでは、AUTOSAR Blocksetが提供するグラフィカルもしくはプログラミングによる方法で、AUTOSAR構成を適用、更新できます。AUTOSARプロパティの管理、構成に複雑なカスタマイズをすると、保守作業が必要となるため、お勧めしません。重要なワークフローの大半は市販のツールで十分に対応可能です。

選択したワークフローと、重視するAUTOSARコンセプトに適したツールを選んでください。

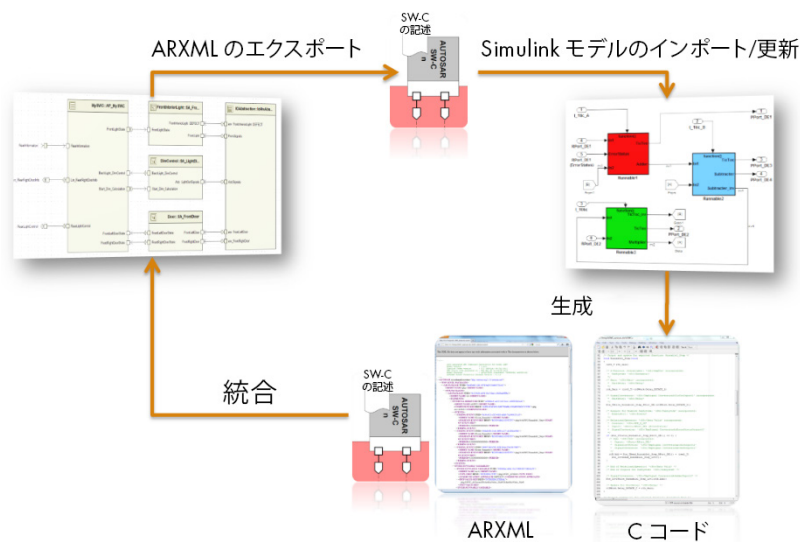


図 8. AUTOSARワークフローのラウンドトリップ(反復開発)。

3. データ管理戦略を選択する

AUTOSARでは重要な設計データを指定するための豊富なスキーマが提供されており、重要な設計データの管理方法はAUTOSAR採用戦略の中で非常に重要な役割を持ちます。データはSimulinkやAUTOSARオーサリングツール、もしくは外部ツールで管理でき、異なるレベルや、複数のオーナーで管理することもできます。変更管理もまた、データ管理戦略の不可欠な要素なのです。データ管理戦略が、選択したAUTOSARワークフロー、データの所有権に関する戦略、レガシーデータの管理方法、既存のSimulink設計をAUTOSARにの移行するための戦略に即していることが肝要です。

SimulinkのAUTOSAR Component Designerアプリ (図9) を使用すると、AUTOSARソフトウェアコンポーネントのデータを管理することができます。パラメーターと変数データは内部データであり、コンポーネントにスコープされていますが、システム全体の定数はコンポーネント間で共有されています。Component Designerアプリは、内部コンポーネントデータを管理するためのコードマッピングエディターとプロパティインスペクターを提供します。これらのSimulinkツールは、別のツールによるAUTOSAR特有データの管理を補完します。

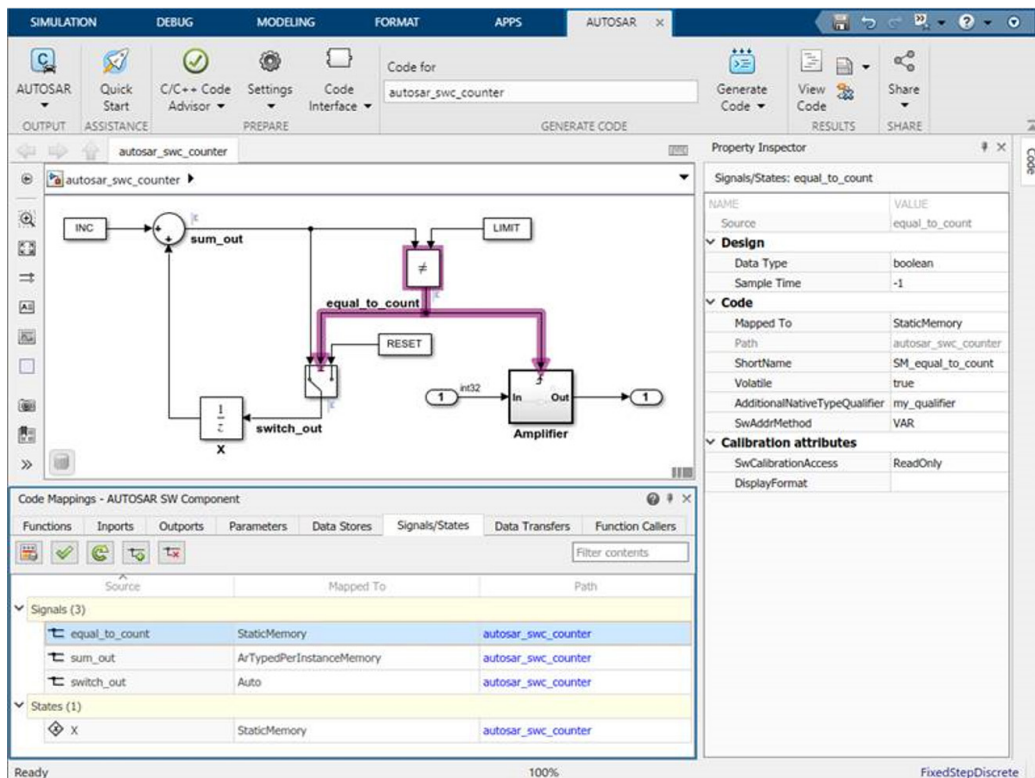


図 9. AUTOSAR Component Designerアプリ。

さらに、要求に応じてコンポーネントデータをランタイムキャリブレーション用や測定用に構成することができます。

4. モデリング標準を確立する

SimulinkとAUTOSARのモデリング標準を確立することで、組織全体で採用を統一的に行うことができ、複数のプロジェクト間での再利用や、大規模なシステムやチームへの拡張が可能となります。モデリング標準はモデルベースデザインの展開には必須であり、また、ISO 26262安全規格の要件でもあります。モデリング標準は、Simulinkを用いたAUTOSAR採用方法、選択した機能セット、ワークフロー、およびデータ管理戦略に即している必要があります。

AUTOSARソフトウェアのコンポーネントのモデリングの際に有効なのは

- データのカプセル化：AUTOSARソフトウェアコンポーネント内で、パラメーターと変数データをカプセル化します。Simulinkパラメータ、データストア、信号、状態を使用して、AUTOSARコンポーネントの内部データをモデル化します。データのカプセル化により、コンポーネントのインスタンスをコンポーネント階層やシステムレベルのシミュレーションに柔軟に含めることができ、複数のプロジェクトで使用することができます。
- インターフェース管理：AUTOSARコンポーネントやプロジェクト間でインターフェースとタイプを共有します。AUTOSAR Blocksetでは、AUTOSARソフトウェアコンポーネントがインポートや作成するためのインターフェースとタイプを事前に定義することができます。例えば組織内で、以下のような物を事前定義して共有することが可能です。
 - AUTOSARコンポーネントのポート・インターフェースとその入出力データ・タイプ
 - AUTOSARコンポーネント内部データ型(IncludedDataTypeSets)
 - AUTOSARデータ型と関連要素(AUTOSAR XML packages)
 - AUTOSARシステム定数

AUTOSAR ソフトウェアコンポーネントの動作のモデリングパターン (図 10) については、ドキュメンテーションを参照してください。

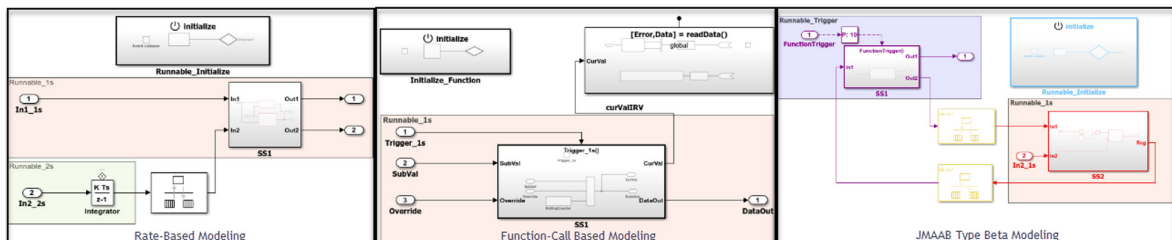


図 10. モデリングパターン。

5. シミュレーションしてからコード生成する

AUTOSARを導入すると、新しいワークフローや、ARXMLファイルの正しさ、ソフトウェアへの影響に注目しがちです。しかし、ソフトウェアコンポーネントをSimulinkで設計する一番の意義は、ARXMLの生成や、AATやRTE生成ツールとの統合のはるか前に、実装が確実に要件を満たすことです。Simulinkでは、広範なAUTOSAR規格をサポートするために導入された各機能についても効率的にシミュレーションを行うことができます (図11)。

シミュレーションによって次のことも可能になります。

- 設計プロセスの初期に、ソフトウェアコンポーネントの実装が正しいことを確認できます。
- コードを統合する前にSimulink内で複数のSWC(AUTOSARコンポジション)をまとめてシミュレーションできます。
- ソフトウェアインザループ(SIL)およびプロセッサインザループ(PIL)のテストを再利用して、RTE生成の前に単体レベルで生成コードを検証できます。

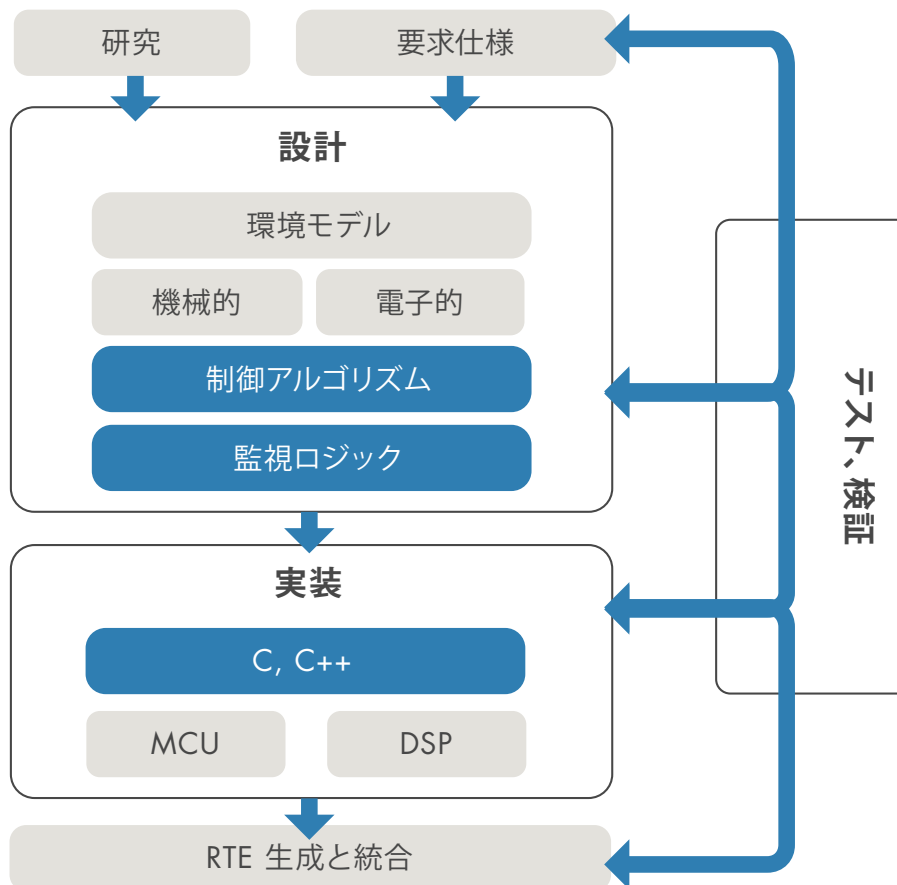


図 11. モデルベースデザインのアクティビティ。

6. 量産コード生成機能を使用する

AUTOSAR規格では、AUTOSAR基本ソフトウェア(BSW)、RTE、ARXMLファイルなどはツールを使用して生成されます。そのため、ソフトウェアコンポーネントのコードも自動生成すべきか否は自ずと明らかとなります。ソフトウェアコンポーネントの実装には、量産コード生成機能を使用するのがお勧めです。図12に示すように、APIの複雑度を考えると手動でAUTOSARをコーディングするのは現実的ではありません。AUTOSAR BlocksetとEmbedded Coderを使うことでSimulinkでの設計と実装をほぼ一致させることができるため、シミュレーションや検証を効率的に行うことができます。

```
void Runnable_simple_alg_Step(void)
{
    real_T rtb_Gain;
    real_T rtb_Delay;
    real_T rtb_Delay1;
    real_T rtb_TmpSignalConversionAtFast_i;
    if (simple_alg_M->Timing.TaskCounters.TID[1] == 0) {
        Rte_Receive_Fast_in_Fast_in(&rtb_TmpSignalConversionAtFast_i);
        rtb_Delay = simple_alg_DWork.Delay_DSTATE;
        rtb_Delay1 = simple_alg_DWork.Delay1_DSTATE;
        rtb_Gain = simple_alg_DWork.Delay2_DSTATE;
        rtb_Gain = (((rtb_TmpSignalConversionAtFast_i + simple_alg_DWork.Delay_DSTATE) + simple_alg_DWork.Delay1_DSTATE) +
            rtb_Gain) * simple_alg_P.Gain_Gain;
        if (simple_alg_M->Timing.TaskCounters.TID[2] == 0) {
            simple_alg_B.RateTransition = rtb_Gain;
        }
        simple_alg_DWork.Delay_DSTATE = rtb_TmpSignalConversionAtFast_i;
        simple_alg_DWork.Delay1_DSTATE = rtb_Delay;
        simple_alg_DWork.Delay2_DSTATE = rtb_Delay1;
    }
    if (simple_alg_M->Timing.TaskCounters.TID[2] == 0) {
        Rte_IWrite_Runnable_simple_alg_Step_Out1_Out1(simple_alg_B.RateTransition
            + Rte_IRead_Runnable_simple_alg_Step_Slow_in_Slow_in());
    }
    rate_scheduler();
}

```

```
...
<RUNNABLE-ENTITY UUID="aef16585-a355-494f-accd-1a548ca22e27">
<SHORT-NAME>Runnable_simple_alg_Step</SHORT-NAME>
<MINIMUM-START-INTERVAL>0</MINIMUM-START-INTERVAL>
<CAN-BE-INVOKED-CONCURRENTLY>false</CAN-BE-INVOKED-CONCURRENTLY>
<DATA-READ-ACCESS>
<VARIABLE-ACCESS>
<SHORT-NAME>IN_Slow_in_Slow_in</SHORT-NAME>
...
</RUNNABLE-ENTITY>
...

```

```
...
<SENDER-RECEIVER-INTERFACE>
<SHORT-NAME>Out1</SHORT-NAME>
<IS-SERVICE>>false</IS-SERVICE>
<DATA-ELEMENTS>
<VARIABLE-DATA-PROTOTYPE>
<SHORT-NAME>Out1</SHORT-NAME>
...
</VARIABLE-DATA-PROTOTYPE>
</DATA-ELEMENTS>
</SENDER-RECEIVER-INTERFACE>
...

```

図 12. AUTOSAR APIを使用して生成されたコード

Embedded Coderは量産用C コードをSimulinkから生成します。コードはAUTOSAR準拠にも、AUTOSAR準拠以外にも対応可能です。

7. Simulinkを使用してレガシーコードをAUTOSARに移行する

十分に検証され広く使用されているAUTOSAR以外のソースコード (レガシーコード) が、AUTOSARの初期採用の段階で再利用されることはよくあります。この場合、AUTOSAR準拠にするために、コードに変更を行う必要があります。コードの自動生成に加えて、レガシーコード用のAUTOSAR準拠のラッパーを自動生成することも有効です。C S-FunctionブロックやC Callerブロックを使用してこのコードをSimulinkに取り込むことで、モデルをAUTOSARにマッピングできるようになります。この方法を使用すると、CまたはC++コードをSimulinkに統合して、シミュレーションや量産コード生成を行うことができます。

コードをSimulinkに統合すると次のことが可能になります。

- テストハーネスの作成
- 出力の可視化と解析
- 統合されたレガシーコードを含むモデルカバレッジの取得

量産コード生成では、Embedded Coderはレガシーコードのラッパーとして、必要となるAUTOSAR RTE APIアクセスポイントおよびランナブルエンティティを生成します。結果として得られる生成コードの例を図13に示します。

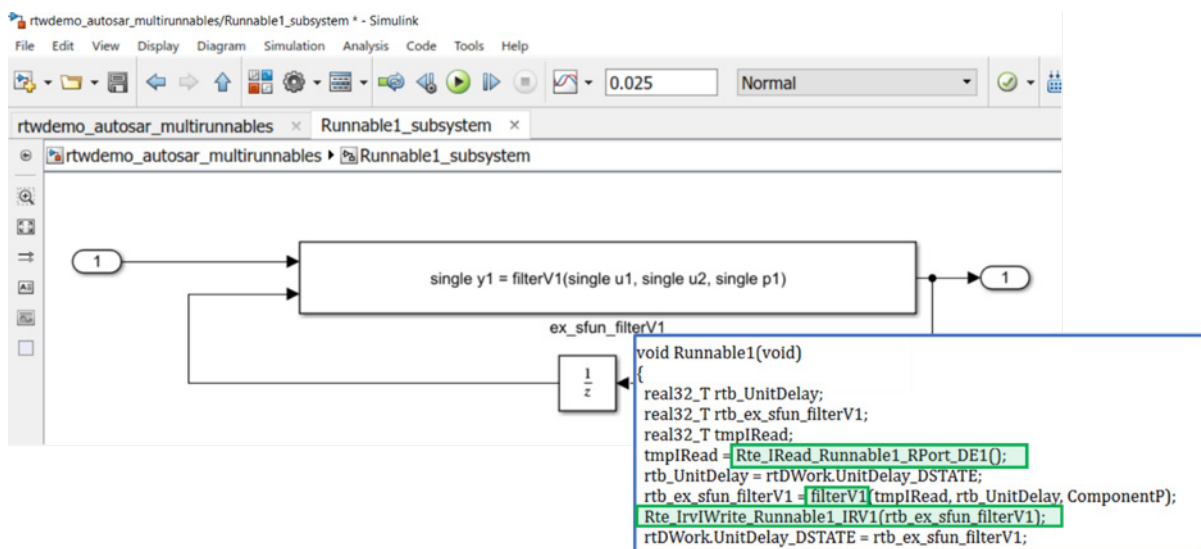


図 13. AUTOSAR APIを使用したレガシーコードの統合。

8. 徹底的に自動化する

一般的なAUTOSARワークフローでは、チーム内で繰り返し行われる共通プロセスがあります。AUTOSARの構成と展開を手動で行うと、以下が困難となります。

- 命名規則などの標準が複雑
- AUTOSARの反復する作業サイクル
- 複雑なコードAPIおよびXMLファイル定義

ワークフローの自動化にはAPIを使用することが有効です。図14に示すようなドキュメント化されたMATLAB APIを使用して、SimulinkでSWCを構成できます。これらのタスクの自動化機能を作成するときには、各タスクが十分にドキュメント化されていること、また各タスクが検証可能であることに注意してください。そうすることで、MATLABの新しいリリースへのアップグレードがスムーズに行えるようになり、新しいリリースでは不要なカスタムソリューションを排除できるようになります。

将来、新しいMATLABのリリースへの移行がスムーズに行えるよう、ドキュメント化されたAPIのみを使用するようにしてください。

```
%% Setup AUTOSAR Configuration Programmatically
model = 'autosar_swc_counter';

% Modify AUTOSAR Properties
autosarProps = autosar.api.getAUTOSARProperties (model);
set (autosarProps, 'Input', 'IsService', true);
set (autosarProps, 'XmlOptions', 'ArxmlFilePackaging', 'SingleFile');
```

図 14. AUTOSARターゲット用のMATLAB API。

9. ISO 26262の導入は事前に計画する

ISO 26262は自動車の機能安全に関する世界的な規格です。AUTOSAR対応と同じく、この規格への準拠も、開発プロセスと実装の両方に大きく影響します。AUTOSAR適用のための変更を行うのと同時に、ISO 26262準拠についても評価および計画を行っておくと良いでしょう。これにより、ISOのような安全規格への対応の必要が生じた場合にも、AUTOSAR向けのプロセスとツールを拡張できるようになります。

IEC Certification Kitは次の製品のツール認証の提供により、ISO 26262をサポートしています。

- Embedded Coder
- Simulink Check™
- Simulink Coverage™
- Simulink Design Verifier™
- Simulink Requirements™
- Simulink Test™
- Polyspace Bug Finder™
- Polyspace Code Prover™

IEC Certification Kitで提供されるアーティファクトはTÜV SÜDによって認証されています(図15)。ISO 26262規格の詳細についてはこちらをご覧ください。

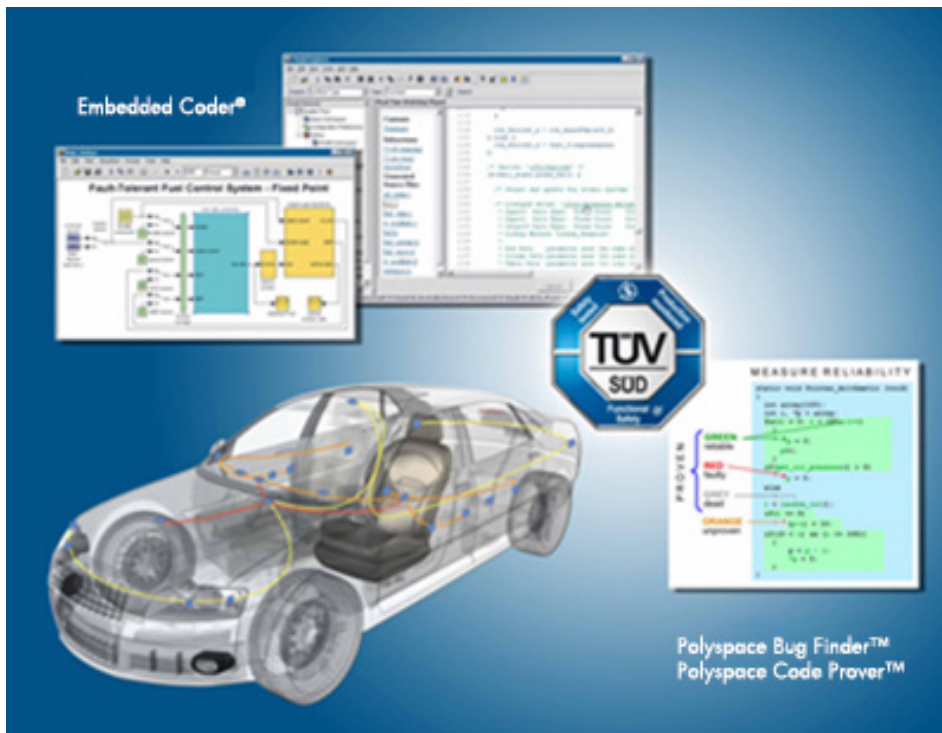


図 15. ISO 26262用の IEC Certification Kit。

10. 将来を見据えた移行計画

自動車業界における組み込みシステムの開発と比べて、AUTOSARは比較的新しい規格であり、現在も発展段階にあります。MathWorksでは製品リリースのたびにAUTOSAR向けのツールのサポートを拡大しています。SimulinkおよびEmbedded Coderでは、6か月ごとに提供される新しいバージョンごとに、AUTOSAR向けの重要な新機能が提供されます。

このため、AUTOSAR、Simulink製品の新バージョン対応戦略も必要となります。具体的には、アップグレードを行う頻度や、新しいバージョンを選択する条件などを指します。

一般的にMathWorksでは、継続的なアップグレードという考え方を推奨しています。継続的にアップグレードを行うことで、アップグレードへの対応は容易になっていきます。MathWorksでは、プレリリーステストや業界モデルテスト、セミナー、Web セミナー、カンファレンスなどで最新情報を提供しております。

アップグレードプロセスの管理の詳細については、ホワイトペーパー「[MATLAB and Simulink Version Upgrades for Large Organizations](#)」を参照してください。

関連情報

以下のリソースを利用してAUTOSARの開発を始めてください。

- [ドキュメンテーション](#)
- [トレーニング](#)
- [技術コンサルティング](#)

著者について

David Jaffryは、MathWorks Consulting Servicesの上級コンサルティングエンジニアとして、自動車、航空宇宙、生命工学、産業オートメーションの各作業分野の企業のモデルベースデザインの実装、検証の手法、組み込みシステムの開発を支援しています。MathWorksに加わる前は、組み込みシステムのソフトウェア開発およびコード検証に従事していました。プレスト国立工科大学でコンピューターシステム工学の修士号を取得しています。

Holly Keenerは、ミシガン州に拠点を置くMathWorks Consulting Servicesのマネージャーとして、自動車、航空宇宙、防衛、通信、医療機器の各作業分野のさまざまな事業で業界をリードする各企業と業務を行っています。組み込みシステムの開発や検証へのモデルベースデザインの採用を支援しています。Hollyは世界中の自動車OEMやサプライヤーにおけるAUTOSARのモデルベース デザインの適用を進めて来ました。MathWorksに加わる前は、自動車業界において組み込みシステムの開発を行っていました。Hollyはミシガン大学で電気工学の学士号と医用生体工学の修士号を取得しています。