



はじめてのAUTOSAR

ビギナー向け資料「はじめてシリーズ」

発行元：ベクター・ジャパン株式会社

ベクター・ジャパン株式会社の書面による許可なしに、本書の内容を転載、複製、複写することを禁じます。
記述されている内容は予告なく変更されることがあります。

©Vector Japan Co., Ltd.

目次

第1章 AUTOSARとは何か？	02
はじめに ~AUTOSAR 採用の増加~	02
AUTOSAR 普及の背景	03
技術面の概要	05
開発の流れの概要：ベクター製品を使用したときの例	08
第2章 AUTOSARの導入形態	11
導入事例	11
導入事例から見える現実	12
第3章 AUTOSAR導入のための準備	13
改めて、AUTOSARとは？	13
導入にあたっての心構え・準備	15
おわりに	16

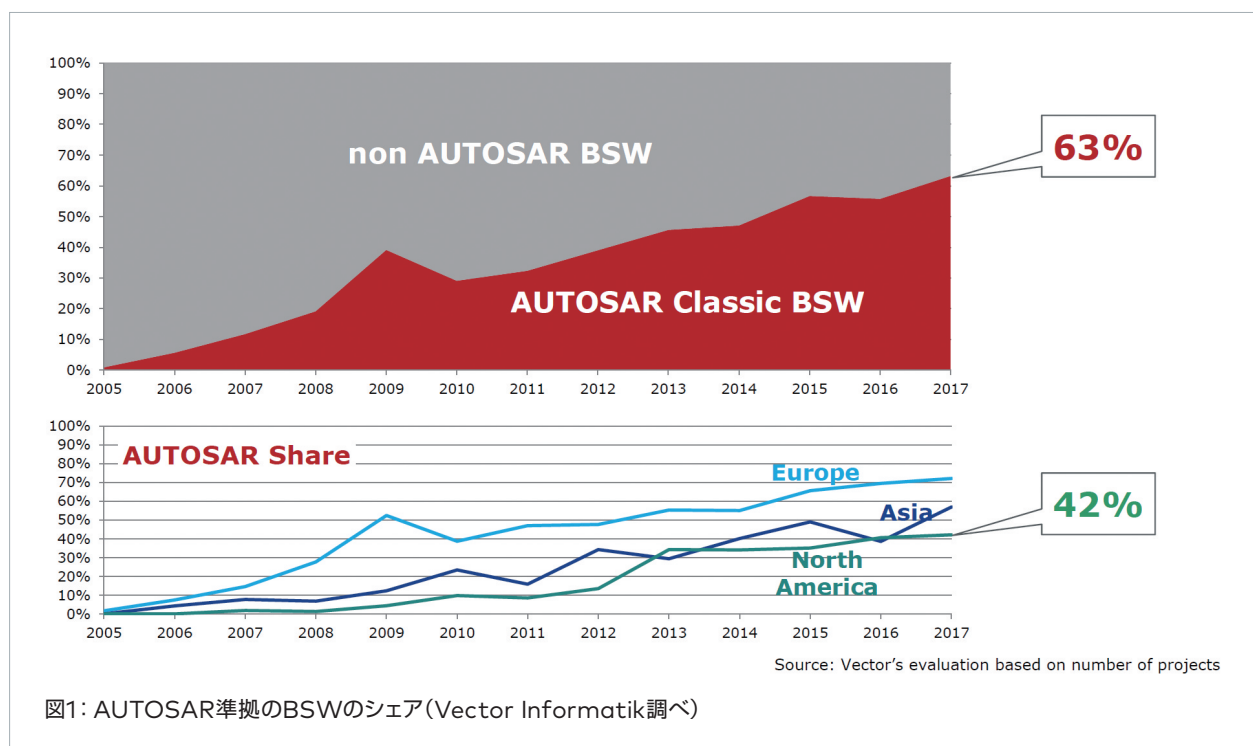
はじめてのAUTOSAR

第1章 AUTOSARとは何か？

はじめに ~AUTOSAR 採用の増加~

ベクター・ジャパン（以下、ベクター）では、AUTOSAR採用にあたって必要なサービスや製品を、2005年頃から約10年以上提供してきた。その間に、車載組込ソフトウェア開発に携わる方々の中では、「AUTOSAR」の名前は、すっかり市民権を得たように感じる。

実際、Vector Informatikの調査によると、AUTOSAR準拠のBSWのシェアは2017年に約63%に達した（図1）。



このように、量産開発の現場でAUTOSARが利用される機会は大幅に増え、ベクターでも、おかげさまで、多くのお問い合わせと実際のビジネスの機会を頂いており、まさに嬉しい悲鳴を上げている状態である。

しかし、その導入にあたっての、お客様側での事前の予想と現実には大きなギャップがあるようで「予想以上に準備が大変だった」「単にモジュールを入れ替えるくらいだと思っていたが、実際には大違いだった」などの声が多く寄せられている。その結果、特に、納期に間に合わせるためのサポートでは、お客様と一緒に、つらい悲鳴を上げざるを得なくなる場合も少なくない。また、これから導入しようとするお客様からは、「仕様書の内容を、本当に全部やらなければならないの?」「仕様書を読んだだけでは、そもそも、どんな準備が必要なのか、想像しようがない」という不安をぶつけられることも多い。

本資料では、「実際にAUTOSARに移行(migration)をするにあたって理解しておくべきこと」に関するヒントを中心に整理していきたい。

AUTOSAR 普及の背景

たとえ同じ仕様書に基づく同一の機能であったとしても、異なるECUサプライヤーにより、アプリケーション部分のソフトウェアの「実装」が、それぞれ個別に行われてしまうことは珍しいことではない。

その場合には、実装と評価の作業（場合によっては、さらに仕様詳細のすり合わせ作業）の重複が発生してしまう。もちろん、それを防ぐために、仕様書だけではなく、自動生成のためのモデルやソフトウェアのコードがセットで配布されることも実際に行われている。

しかし、マイコン性能に応じた対処や、ソフトウェアプラットフォーム（あるいは土台部分）とのインターフェイスの合わせ込みのためのコード書換えなどの理由により、ECUサプライヤー側での組み込み作業工数が当初予想よりも多く掛かることも少なくはないようである。

そもそも、ECUの構造は、自動車業界全体で共通の認識が存在しているとは言い難い。ソフトウェアの土台部分についても、自動車メーカー側から指定されたものを使用するケースがすでに一般的であるが、ECUサプライヤー各社が独自に用意し、各自動車メーカー向けに合わせ込みを行うケースもまた一般的である。

そのため、さまざまなモジュールを統合して正しく動作させるためのいわゆる「インテグレーション」の作業では、自動車メーカーにより指定される土台部分の相違などによって、アプリケーション部分と土台部分との関係（接続方法や機能的な分担など）に関する予想外の重複や隙間への対処が必要となってしまうこともある。当然、当初から想定している事態ではないため、対処のためにごく限られた時間しか割くことができないということもありえる。このことは、たとえば、モデルベース開発がすでに一般的であって、かつ、その都度ソフトウェアを新規開発するよりも、各種パラメーターの適合などを行う比率が高いというような分野であっても、同様である。

また、構造や土台における相違は、複数の自動車メーカー向けにECUを供給しようとする場合の障害の1つともなっている。もちろん、いくら汎用のものであったとしても、最終的には個別の要求に合わせ込まなければならない。そして、言うまでもなく、広い範囲で使うことができるという汎用性と、それを利用する上で必要とされる資源やコストとの間には、一般にトレードオフの関係があるため、個別のケースでの利用可能な資源やコストに関する要求が厳しい自動車分野では、「いかなるケースにも適用可能な汎用品」が常に利用可能であると期待することはできない。

そもそも、「個別の要求への合わせ込み」は、汎用製品を利用あるいは提供し続けていく上での永遠の課題でもあるが、構造や土台における相違は、その上に載せるものにいかに広い汎用性を持たせるか、ということを考える上で、「周囲の状況として想定すべきもの」を特定しにくい状況を生み出している。

量産後も、新機能追加、あるいは、より安価な電子部品への切り替えといったことも、「ソフトウェア」に関連する「制約」（実装内容の変更や再インテグレーションなどの工数など）により、さほど自由に行うことができるわけではない。

このように、ソフトウェアの開発においては、実装やインテグレーション、評価など多くの段階で多数の課題が存在している。

さらに、自動車に搭載される機能は年々高度になり、それを実現するためのECUの数やそれぞれの複雑さ、開発規模が大幅に増大している。さらに、マイコンなどの電子部品の世代交代サイクルが短くなる

なかで、車両のライフサイクルがさまざまな要因によってむしろ長くなる傾向が見られるなど、開発を取り巻く環境においても、対処が必要な課題は多く存在する^[1]。

これらの例を含む現在の自動車業界の技術面の課題に対しては、企業単位でのさまざまな取り組みだけではなく、OSEK/VDX^[2]やHIS^[3]などの団体によるソフトウェア関連の標準化が進められ、多くの成果を残してきた。

そして、2003年に設立された開発パートナーシップであるAUTOSAR^[4]では、従来の成果を利用しつつ、

Cooperate on standards, compete on implementation
（標準化においては協力、実装／利用にて競争）

のスローガン^[5]の下で、自動車業界全体にさらに範囲を広げて標準化に取り組んでいる。この活動には、自動車分野に関わる世界各国の多数の企業など（自動車メーカー、ECUサプライヤー、マイコンベンダー、ツールベンダー、ソフトウェアベンダーや研究機関他）が、さまざまな形で参加している。

AUTOSARの仕様書は、2005年発行のRelease 1.0からすでに多くの版を重ねており、2018年4月現在での最新はRelease 4.3である。また、Release 3.0から急速に量産車開発への採用が進んでおり、現在はRelease 4.xが主に利用されている。

-
- [1] 財団法人自動車検査登録情報協会:わが国の自動車保有動向
<http://www.airia.or.jp/number/index2.html> (last access: 20180404)、
 社団法人電子情報技術産業協会:自動車用半導体補給部品供給の円滑化に向けて
<http://semicon.jeita.or.jp/news/docs/050328.pdf> (last access: 20180404) など。
- [2] Offene Systeme und deren Schnittstellen fuer die Elektronik im Kraftfahrzeug.
<http://www.osek-vdx.org/> ([4]のページに自動遷移last access: 20180404)
- [3] Herstellerinitiative Software.
<http://www.automotive-his.de/> ([4]のページに自動遷移last access: 20180404)
- [4] AUTomotive Open System ARchitecture.
<http://www.autosar.org/> (last access: 20180404)
- [5] AUTOSAR Release 3.1 Technical Overview 仕様書 sec. 1.4

技術面の概要

AUTOSARでのECUソフトウェアは、大きく分けて、3つの部分から構成される。その構造を図1.1に示す。

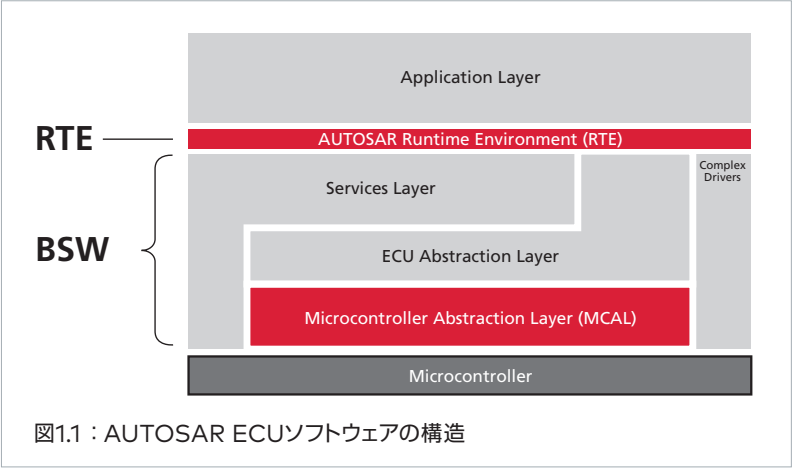


図1.1：AUTOSAR ECUソフトウェアの構造

一番下の、AUTOSARベーシックソフトウェア（Basic Software, BSW、詳細は図1.2）は、簡単に表現するならば、「どの自動車メーカー向けの、どのECUサプライヤーが作った、どの種類のECUにでも入っている、土台部分の基本ソフトウェア」であり、OS、通信スタック（CAN、LIN、FlexRay、Ethernetなど）、不揮発性メモリースタック、診断、入出力、モード管理などの各種機能を提供するソフトウェアモジュール群から構成される。

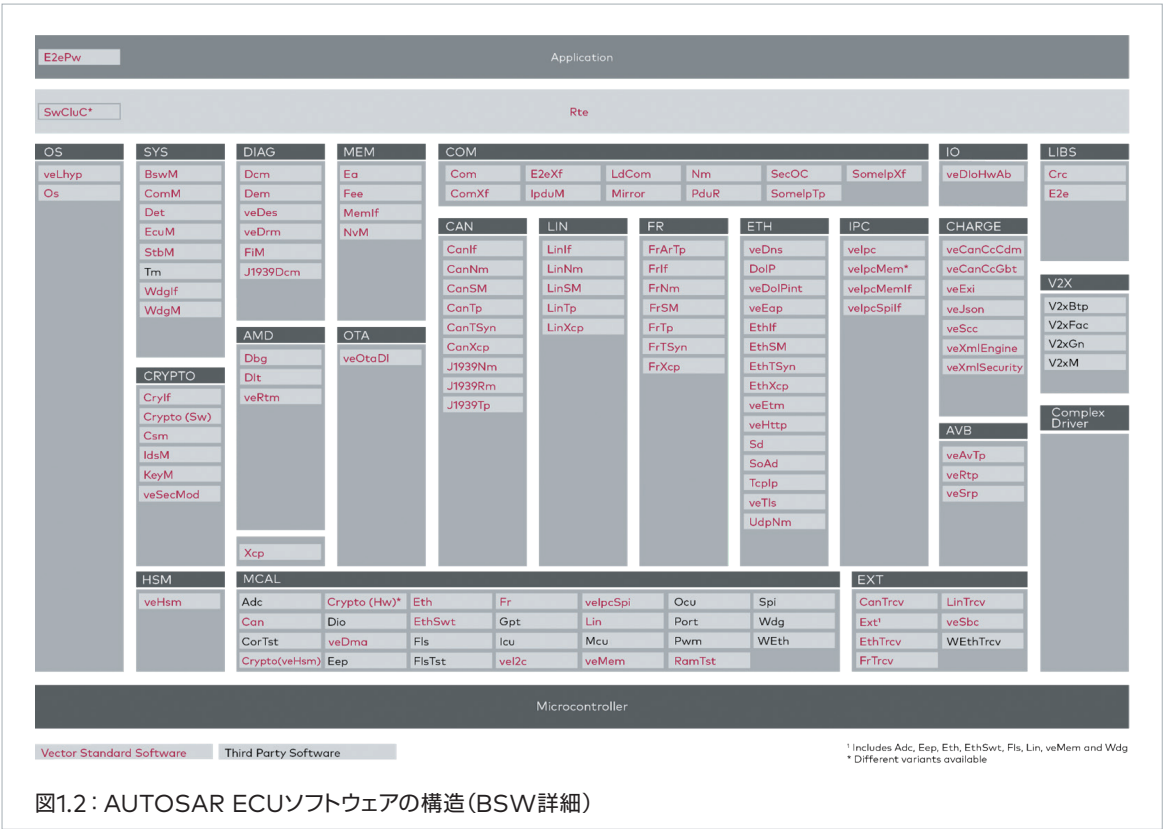


図1.2：AUTOSAR ECUソフトウェアの構造 (BSW詳細)

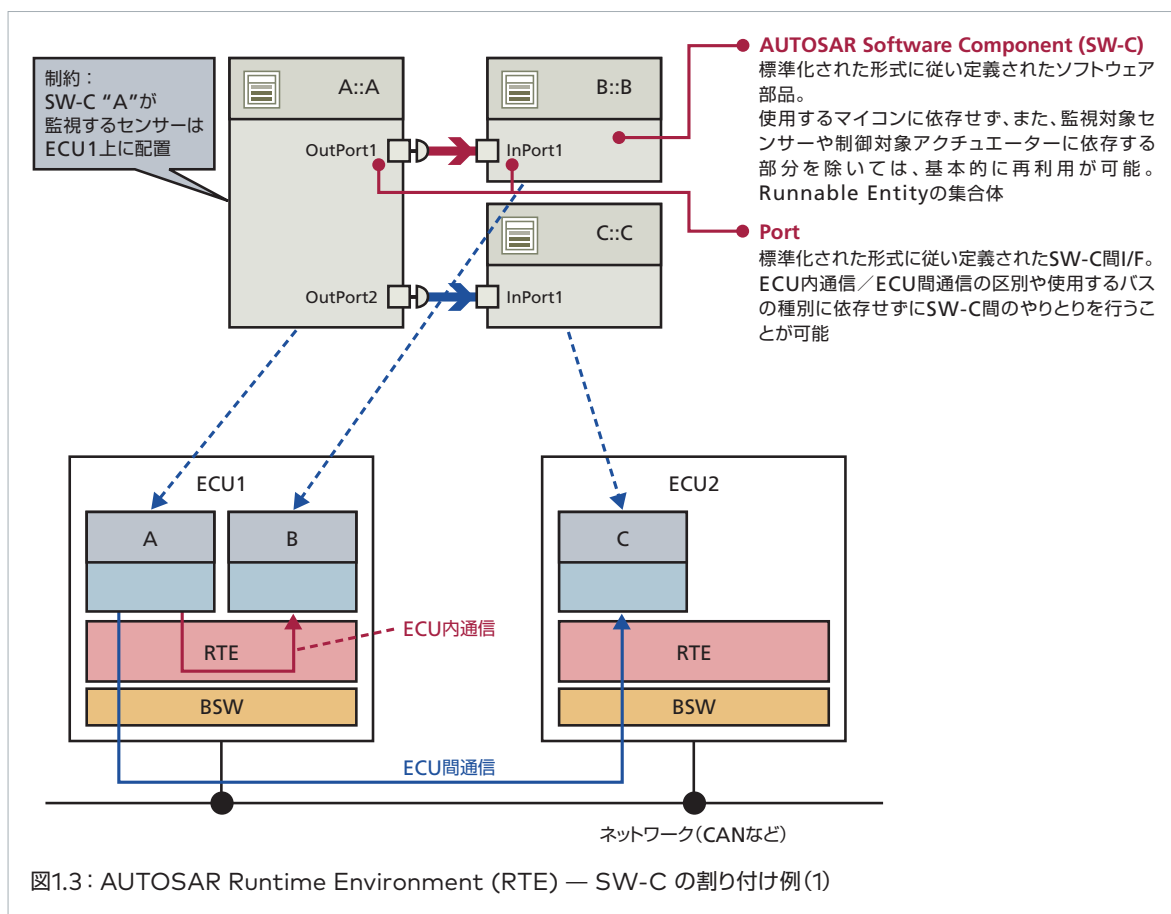
なお、ソフトウェアアーキテクチャー上、ハードウェア依存となるものは、マイコンに固有のMCAL(Microcontroller Abstraction Layer)、OSおよびComplex Driverと、マイコンに外部接続されるデバイス用のドライバーのみである。これ以外はハードウェア非依存であるため、ハードウェア変更に伴う影響を局所化することができる。

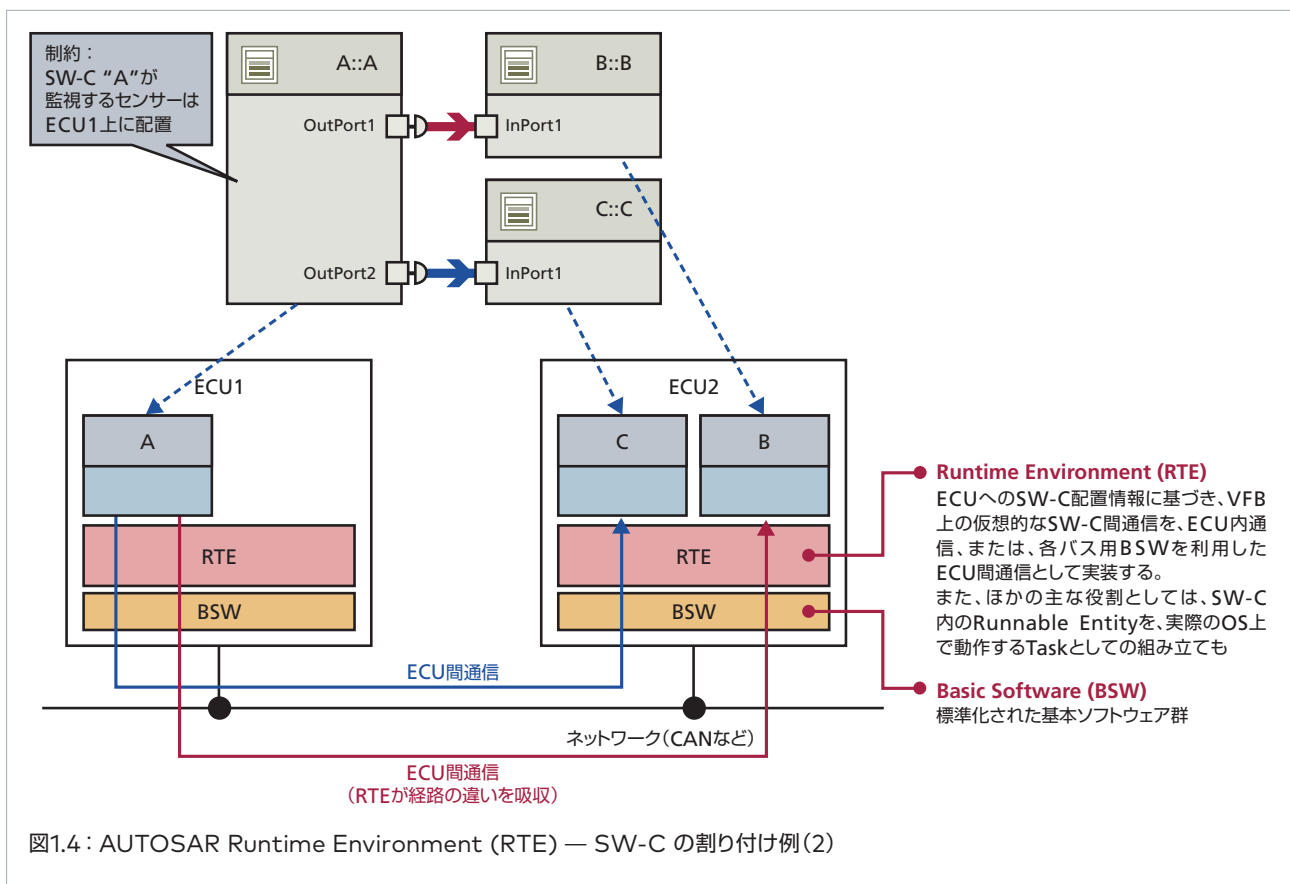
一番上の、個別のECUの機能を実現するためのアプリケーションレイヤーの部分は、通常複数のソフトウェアコンポーネント(Software Component, SW-C)から構成される。

ある機能を実現するために、車両内の複数のECUが関係することはすでに珍しくないが、車両を1つのシステムと見たとき、センサーを監視する入力部分、アクチュエーターを操作する出力部分、そして、その間の演算部分の3つの部分に分けることができる。そして、それぞれに対応するSW-Cの間は、「Port」により接続される(注:ここでのPortは、入出力ポートや、BSWのPORTドライバーとは異なる)。

SW-Cは、機能を実現するために必要なセンサー、アクチュエーター、ROM/RAMなどの資源と、時間制約を考慮した上で、それぞれのECUに割り付けられる。この割り付けにより、Port間の通信がCAN通信用BSWなどを利用したECU間通信になるのか、あるいは、ECU内通信になるのかが決まるが、その際に、SW-Cそのものには一切手を加える必要がないように、あたかも電話交換機のようにつなぎ変えてくれるのが、RTE(Runtime Environment)の主な役割の1つである。このことにより、アプリケーションレイヤーのトップダウン型開発が、従来よりも容易となる。

また、SW-Cの実際の処理は、周期イベントなどの各種イベント発生時に実行される処理である、「Runnable Entity」により実現される。このRunnable EntityをOSのタスクとして構成するのも、RTEの役割である。RTEの概要を図1.3、1.4に示す。





なお、BSW/RTEともに、実装依存の部分は少なくなく、また、実用面あるいは特定用途の要求に合わせるための仕様拡張もしばしば行われている。したがって、同じ仕様に基づく製品であっても、品質や実装方法だけではなく、機能面でも相違がある可能性があることに、注意が必要である。

また、AUTOSARでは、上記のようなソフトウェアアーキテクチャだけではなく、データ交換形式の標準化も行っている。これは、設計資産の利用可能期間をできるだけ長くしたり、インテグレーション作業の自動化の範囲を広げたりする上で、非常に重要なものである。AUTOSARではXMLを利用した標準データ交換形式を規定しており、そのうち、特に重要なものを以下に示す。

- > **System Configuration Description:** 車両システム全体の設計情報。
- > **ECU Extract of System Configuration Description (EcuEx):** System Descriptionのうち、特定ECUに関する情報のみを抽出したものであり、自動車メーカーとECUサプライヤーの間の設計情報授受の媒体として使用される。
 なお、従来、CANネットワークに接続されるECUの開発では、DBCファイルがその媒体として一般的に使用されてきた。EcuExは、それを置き換え、さらに、多くの情報が記述可能なものである。
- > **ECU Configuration Description (EcuC):** ECU上のSW-Cの情報や、RTEやBSWの設定情報などが含まれる。
- > **Software Component Description (SW-C Description):** SW-Cに関する設計情報。
 PortやRunnable Entityに関する情報や、どのBSWのどの機能を利用するか、などが含まれる。
- > **Basic Software Module Description (BSWMD):** 各BSWの設定項目などの情報。

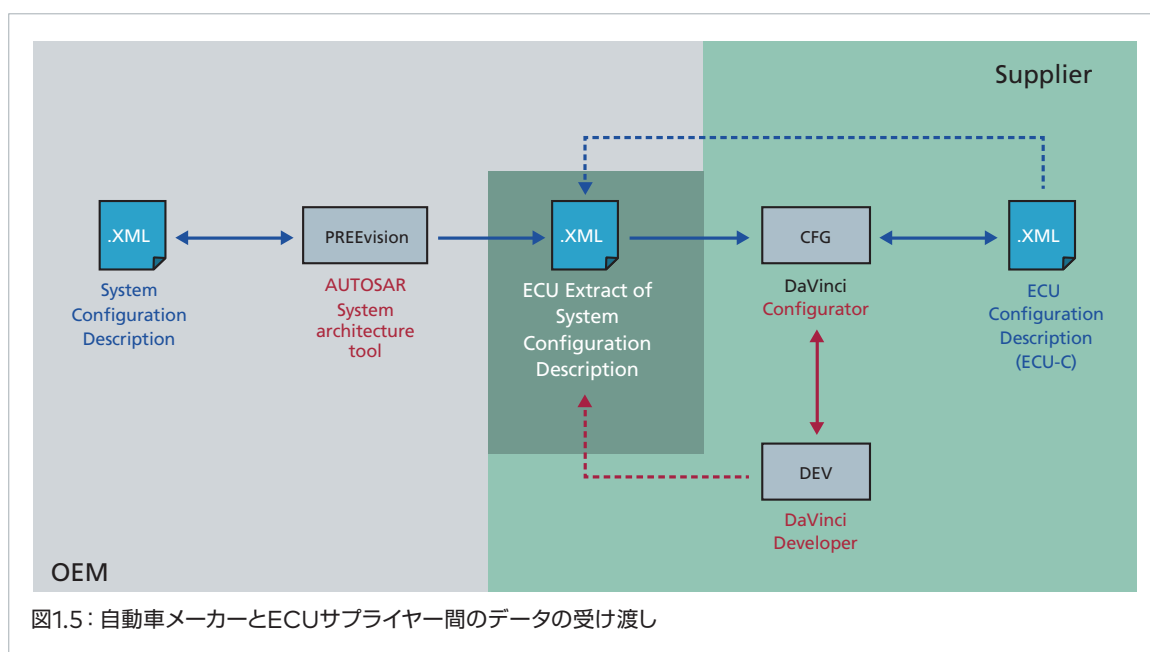
開発の流れの概要：ベクター製品を使用したときの例

AUTOSARの仕様書を読んだとしても、開発の流れの具体的なイメージを持つことは難しい。

これは、いわゆる「標準」(および「汎用製品」)は、「ある範囲においては、あんなこととこんなことが可能」という、枠組みとそこにおける一定の選択肢(明示的であるとは限らない)を提供するものであり、一般には、それを特定の状況に当てはめるための「選択」「このように使うという意思決定」と一体となって利用されるものだからである。もちろん、AUTOSARは「どの立場の人や企業が、どの作業をするべきか」「どの立場の人が使うツールが、どの作業をカバーすべきか」といったことの規定を、標準化範囲に含んでいない。

とはいえ、具体的なイメージを把握するためには、実例は有用である。ここでは、ベクター製品などを使った場合の、実際のECU開発の流れの一例を紹介する。

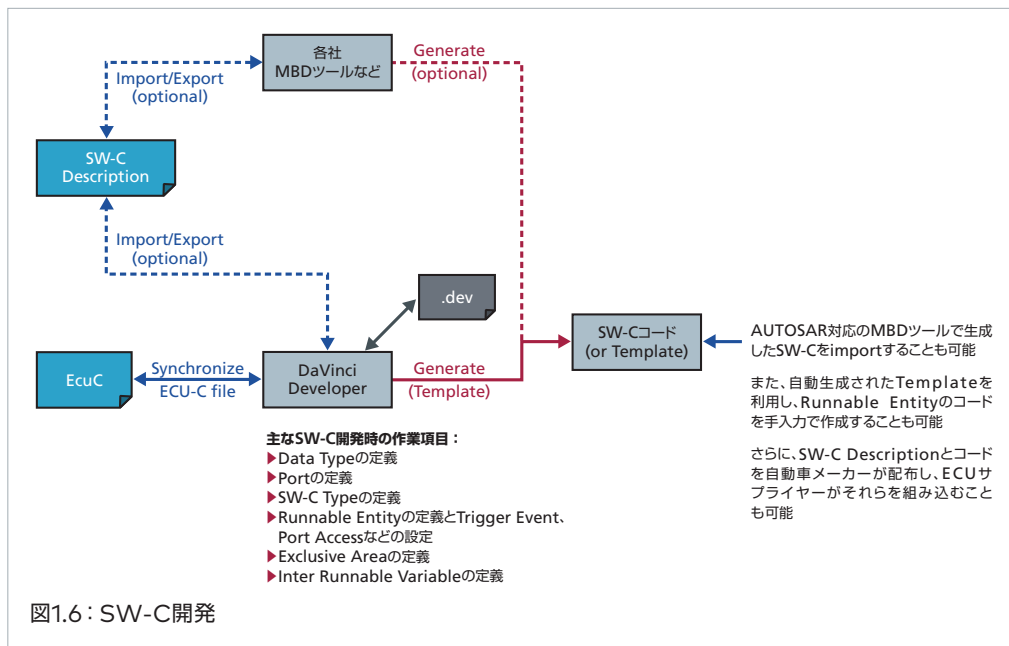
まずは、自動車メーカーとECUサプライヤー間のデータの受け渡しを、図1.5に示す。



前述のとおり、自動車メーカーとECUサプライヤーの間の設計情報授受の媒体としては、ECU Extract of System Configuration Description (EcuEx)が使用される。これは、車両システム全体の設計情報が記述されたSystem Descriptionのうち、特定ECUに関係する情報のみを抽出したものである。

このEcuExの内容は、RTEやBSWの設定情報などを記述したECU Configuration Description (EcuC)に引き継がれる。

次に、SW-Cの開発の流れを、図1.6に示す。

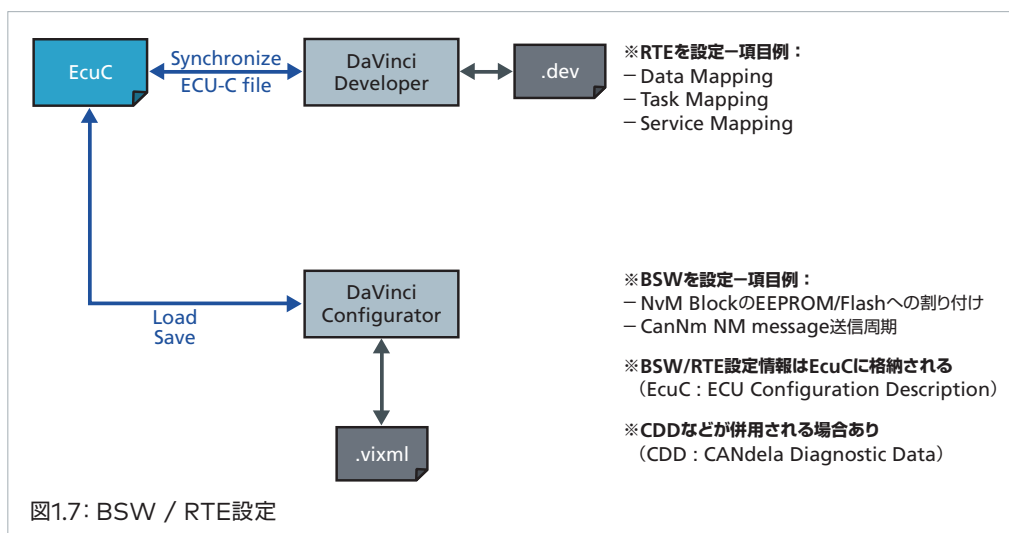


手書きコードとして、ある機能を実現するために必要なSW-Cを実装する際には、SW-Cの間をつなぐ「Port」や、周期イベントなどの各種イベント発生時に実行される処理である「Runnable Entity」の定義、設定が主な作業を行った後に、Runnable Entityの実際の中身（一般的にはCの関数）の実装を行う。モデルベース開発(MBD)の場合には、コード自動生成のために必要な設定を行うことで、これらの多くが行われる。

なお、Runnable Entityを起動するイベントやPortなどの情報は、後のRTEの設定の際に必要なとなるが、Runnable Entityの実際の中身（コード）は、この時点で作成されていなくとも良い。

また、SW-C構造やPortに関する定義が自動車メーカー側から提供される場合や、複数のMBDツールを併用する場合もあるため、この段階でのツール側での作業の流れは、import → 実装 → exportとなるケースや、最初から実装 → exportとなるケースなど、さまざまなパターンが考えられる。

次は、SW-C、BSWとRTEの統合のための各種設定の段階である（図1.7）

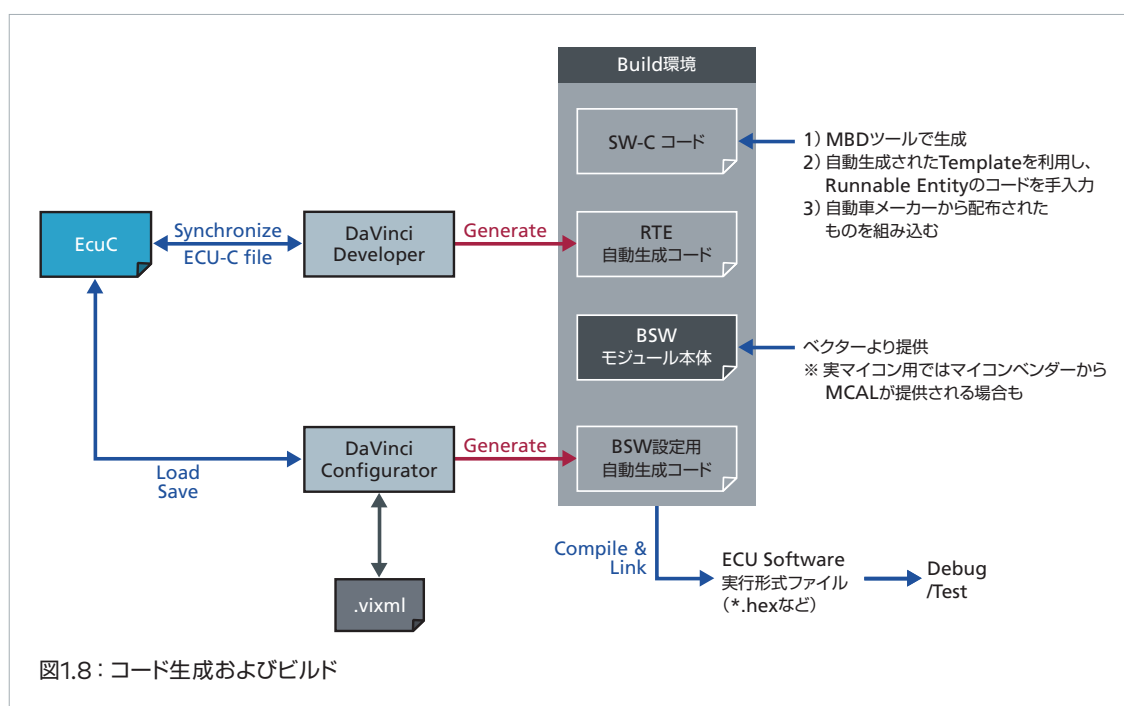


RTEの設定にあたっては、「どのようなSW-CがこのECUに載るのか」という情報が必要となる。

なお、高い汎用性を実現するために、BSWの設定項目は非常に多くなっている(10,000を超える場合もまれではない)。しかし、そのうちの多くは、自動車メーカーからの指示により決まるのが一般的である。その指示の手段は、AUTOSAR XMLの場合もあれば、仕様書などの書面の形式の場合もあるが、当然ながら、後者では、この段階でより多くの作業時間が必要となる。

また、汎用性の高いものを特定用途の要求に合わせるためには、設定内容が適切か否かの判断が不可欠であるため、AUTOSARのみならず、車載組込ソフトウェア開発全般にわたる広い知識が必要になる。また、ある動作を実現するための設定内容が一意に決まるとは限らないことにも、注意が必要である(たとえば、OSのタスクに対するRunnable Entityの割り付けなど)。

最終的に実装が完了したSW-C、設定が完了したBSWとRTEのコード生成とビルドが行われる(図1.8)。



なお、ランタイムのトラブル解析にあたっては、場合によっては、AUTOSARのみならず、車載組込ソフトウェア開発全般にわたる広い知識が求められることもある。

第2章 AUTOSARの導入形態

導入事例

AUTOSARの概要については、公開されている仕様書やいくつかの解説記事、セミナーなどから情報を得ることができる。しかし、量産開発現場でのAUTOSAR導入の実態については、ごく限られた情報しか得られない。

ここでは、現在のAUTOSARの導入形態について説明する。

図1.2には、多数のモジュールが登場するが、現時点では、実際には、BSWやRTEのすべてが、どのECUでも利用されているわけではない。

いくつか導入における特徴を挙げてみよう。

1. 通信／診断スタックのみの利用

他のECUとの通信／診断面の互換性の確保のため、通信／診断スタックのみを導入する。RTEを使用せず、従来のOSEK/VDX OSのタスクや、従来のスケジューラー型での開発スタイルを適用。車両内の全ECUにおける仕様共通化による、品質担保を目的としている。

2. 通信／診断スタックに加えて、OSとRTEを利用

自動車メーカーからのSW-Cの配布に備えて、OSとRTEも導入。

3. 既存モジュールの最大限の流用

他のECUとの通信面の互換性の確保や、自動車メーカーからのSW-Cの配布に備えて、通信／診断スタック、OS、RTEなどの最小限のものだけを利用し、他は可能な限り、既存モジュールを再利用したComplex Driverにて対応。

4. 最小限のMCALの利用

MCALは、一般的にマイコンメーカーから提供されるものを利用することが多いが、それがまだ存在しないような場合もある。そのような場合には、新規開発量をできるだけ減らすために、最小限のMCAL(またはそのサブセット)のみを利用し、他はComplex Driverで対応。

5. 自動車メーカー独自仕様の拡張

一部機能について、自動車メーカー独自仕様を追加。モジュールの改造かComplex Driverで実現。

もちろん、これらがすべてではなく、また、上記のそれぞれが完全に独立のものでもない。

次に開発の流れについての特徴も挙げてみる。これまでと大きな違いはなく、自動車メーカーからのSW-Cの配布があるのか否か、配布されてくるSW-CがどのBSWの提供するサービスを利用するのか、という前提はさまざまである。したがって、特に複数の自動車メーカーにECUを供給するような場合、配布されるSW-Cの組み込みのためにコスト面の制約のもとで、どのようなBSWを購入することが必要かという判断が、サプライヤーに求められる。

さらには、自動車メーカーとECUサプライヤーの間の設計情報授受の媒体であるECU Extract of System Configuration Description (EcuEx)についても、これまでと大きな違いはなく、その中身をどこまで自動車メーカーが記述するのかという点においては、さまざまな運用が行われている。そして、EcuExのかわりに、従来のCAN用DBCファイルや、診断仕様であればcddファイルを利用する場合も多い。さらには、Excelファイルで仕様作成して、ツールで読み込ませる形にコンバーターを使って変換している例もある。

導入事例から見える現実

先に説明した導入事例から、自動車メーカー、サプライヤーそれぞれのAUTOSARに対する思惑、狙いを見ていこう。

自動車メーカー： 車両として、多くのECUに共通の仕様となりえるモジュール（通信、診断など）を中心に導入をすすめる一方、必要なモジュールには独自仕様を織り込み、戦略的にAUTOSARへの移行を推進。非競争領域の工数低減を図り、競争領域へのリソースのシフトを狙っている。

サプライヤー： 規模の大きいサプライヤーは、複数の自動車メーカーを相手にしているため、自動車メーカー要求の違いにより、柔軟な対応を求められる。そこで、自社の標準PFを、AUTOSAR準拠のBSWと、要求仕様による違いに対応する吸収層モジュールを組み合わせることで、開発の効率化を狙う。また、従来から存在するECU（＝既存資産で開発しており市場実績のあるもの）には、前で述べたような通信／診断スタックなど他ECUと共通化するとメリットが出やすい部分を導入していきながら、徐々にAUTOSARにシフトしていく一方、新しい機能のECUは新規開発となるため、全面的にAUTOSAR仕様にシフトする場合もある。

このような状況を踏まえて、これから新しくAUTOSARを導入するためには、どのような準備が必要か、次章で考えてみる。

第3章 AUTOSAR導入のための準備

改めて、AUTOSARとは？

ここでは、AUTOSARに関しても、その導入のための適切な準備のために、もう一度、基本的な性質について見直してみたい。

1. あくまで標準「規格」であり、それ自身は強制力を持つものではない

しばしば、「これは、AUTOSARに対する違反ではないのか？」という質問を頂くことがある。これは、標準と名が付くものには法的拘束力があるのではないかと、という不安によるもののようであるが、実際には、法規制や個別の契約上の指定事項となった場合を除いては、規格単体では強制力を持たない（ISOなども同様）。

2. あくまで標準「規格」であり、標準「製品」ではない

AUTOSAR仕様書には多数の実装依存部分がある。また、その実装（製品）には、それぞれ仕様に対する拡張が加えられる場合がある。また、実装には品質など各種要因が付け加わる。したがって、同じ規格に基づくものであったとしても、製品における差は存在するものと仮定しなければならない。このことから、複数のモジュール供給元からの製品を組み合わせる使用することは、単一の供給元からのものを組み合わせる使用するよりも難しくなる可能性がある（トラブル解析など）。なお、AUTOSARには、いわゆるリファレンス実装（reference implementation）は存在しないが、自動車メーカーごとのものが存在することもある。

3. あくまで「手段」「道具」であり、「目的」ではない

AUTOSARを使うことそのものは、最終的な目的ではない。それを利用して、最終的な目的を実現するための手段あるいは道具である。

当然、AUTOSARでは足りない部分があれば、それに対応するための処置を、AUTOSARで想定された範囲に対する「拡張」として個別に講じていく必要がある。

4. 汎用性は高いが、個別最適にできるとは限らない

さまざまなユースケースに対応できるようにするために、多くのパラメーターが用意されている。しかし、設計思想の異なるそれぞれのマイコンに対する、処理性能の限界を引き出すための個別最適化よりも、汎用性を相対的に重視したソフトウェアアーキテクチャーとなっている。

そのため、個別最適化が全く不可能というわけではないが、専用品と同等の個別最適化が可能とは限らない。マイコンごとに個別最適化された専用品と比較した場合には、汎用品であるBSWは、処理時間

やROM/RAM消費量などの面で見劣りがする場合もあるが、これを無駄と見るかどうかは、考え次第である(汎用性とパフォーマンスのトレードオフ)。

また、特定のユースケースに合わせ込むためには、多くのパラメーターを設定するための手間が発生するという点でもある。もちろん、代表的なユースケースごとに、パラメーター設定内容の雛型のようなものを用意することは可能であり、作業効率向上のためにも一定の価値があるが、それが容易あるいは可能な部分と、そうではない部分があること、そして、その雛形が適用可能なユースケースには限界があることに注意が必要である。

5. 汎用性は高いが、万能ではない

前項で登場した個別最適化された専用品で何とか機能が実現されていたようなものを、AUTOSAR BSW/RTEを利用して実現しようとした場合に成立しない、という場合は十分に考えられる。

また、世の中のすべてのユースケースが想定されているわけではない。前述のとおり、AUTOSARでは足りない部分があれば、それに対応するための処置を、AUTOSARで想定された範囲に対する「拡張」として個別に講じていく必要がある。

6. 変化し続ける

2018年現在も、AUTOSAR仕様書そのものの開発は継続されており、当面それは続く。当然、その仕様書に基づき実装された製品も変化することとなる。

製品そのものも、一般的には、新たに登場したものがある程度安定するまでには時間がかかる上、使いやすさの向上などを目的とした機能拡張など、開発が継続的に行われており、進化し続けている(少なくとも、ベクターの組込ソフトウェア製品である「MICROSAR」や、ツール製品である「DaVinci Developer」「DaVinci Configurator Pro」は、この種のものに該当する)。

しかし、変化し続けるとは言っても、すべてが変化するわけではなく、既にAUTOSAR自体は成熟期に入っており、量産適用可能な完成度には達しているといつてよい。したがって、完全に仕様FIXするのを待つのではなく、その時点のものを「実際に導入してみて、見えてくること」を積み重ねて変化に対応することが重要ではないだろうか。

7. 「AUTOSAR対応」という表現は一意ではない

たとえば、構成要素や、Release Numberなど、さまざまな面での相違が発生しうる。

どのBSWコンポーネントを使う必要があるのか? RTEは必須なのか?どこからが「AUTOSAR対応」なのかという線引きは?など、本質的には同じ内容の質問をさまざまな表現方法で頂くが、AUTOSAR仕様書上ではこれらについては触れていない。

むしろ、AUTOSARという枠で考えて立ち止まるよりも、まずは、その時点で対応が必要な個別のユースケースごとに、「目的を達成するために、どの構成要素が必要なのか?」ということ判断していく方が、現実的である。

必要な構成要素を判断するための代表的なキーワードとしては、他のECUとの通信面の互換性、SW-C 授受の面での互換性などが挙げられる(これ以上の詳細はここでは割愛する)。

なお、「MCAL が存在する」という言葉が意味するものも一意ではない。対応コンパイラ、マイコン固有の機能がどの程度サポートされているか、などの点で、意味が全く異なる可能性がある。また、同じEcuExと名の付くものであっても、中に含まれるデータの項目は一意ではない。

一意に解釈できるものか、あるいは、そうではないのかを、注意深く見極めなければならない。

8. AUTOSARは不可欠か？

実際に、これまではAUTOSARなしで開発を行うことができたのだから、技術的成立性の面で必要不可欠かといえば、必ずしもそうではない。ただし、どの段階にどの程度時間をかけることができるかという現実の制約の中での成立性に目を向けた場合に、必要不可欠か否かの見方が分かれてくる。

もちろん、すぐに狙った効果が得られるわけではない。これは、「再利用」の効果を得ようとする場合全般に言えることだが、必要な初期投資の回収は、再利用の準備ができて、かつ、それが実際に行われるときまで待たされることとなるため、長期的視点が必要である(すぐに回収できなければならないならば、必要不可欠ではないと判断されるかもしれない)。

ただ、昨今多くの自動車メーカーがAUTOSAR準拠を要求しているというのは紛れもない事実であり、要求を受けたサプライヤーにとっては否応無く、AUTOSARは「必要」となってくる。

AUTOSARは、本質的には「土台」という地味な存在であるため、導入の結果として、分かりやすい即時の効果を期待することは難しい。さらに、実際にAUTOSARを導入するにあたっては、本章で紹介したように、「適切ではない理解や期待」や、それに伴う「準備の不足」「かかる手間の見積り誤り」によって、より多くの困難を招いてしまう可能性もある。

では、長期的にAUTOSARと付き合い、恩恵を得るためには、実際にどのような心構え・準備が必要なのであろうか？次章で紹介していきたい。

導入にあたっての心構え・準備

1. 自分たちのやりたいこと、目的を明確にする

先ほど述べたとおり、AUTOSARは「目的」ではなく手段である。したがって、「開発すべきものは、本質的にはどのようなものか？何を実現せねばならないのか？」を明確にしたうえで、その「目的」のために、AUTOSARの何が利用可能か、そして、何が不足かということを見渡していく必要がある。これも前章で述べたように、AUTOSARは汎用性は高いが万能ではないため、カバーされない範囲の制約に対する対処を考えなければならない場面がある。そのときに「目的」に立ち返り、AUTOSARありきではなく、自前の資産の活用も含めて最適な実現手段を判断するべきである。目的に対する適切なAUTOSARの導入範囲が、確度の高い工数見積りをもたらし、スムーズな開発に繋がるのではないだろうか。

2. 実際に試してみる

自分たちの目的を明確にし、AUTOSARの適用範囲とそうでないところを事前に分析することは、量産開発の前には必須である。しかし、その一方、「まずは試してみる」ことも有効であると考ええる。

ひとつの例ではあるが、弊社が提供する「MICROSAR」では、「Evaluation Bundle」と呼ばれる評価版のパッケージが存在する。これにはAUTOSAR準拠のBSWがすべて含まれており、実際のソースコードを見たり、パラメーターのコンフィギュレーションをツールで行ったりできる。これらを使ってあるモジュールの役割をコードで追ってみたり、パラメーターの設定をしたりと、実際にトライアルしてみると、具体的な導入イメージが掴めるのではないだろうか。特に導入初期では、開発工数を見積もることは非常に困難であるが、自分の手を動かしてみることでよりリアルに近い見積もりが可能になるのではないだろうか。

おわりに

以上、3章にわたって「実際にAUTOSARに移行(migration)していくにあたり理解しておくべきこと」に関するヒントを述べてきた。

冒頭でも述べたが、万能な解はもちろん存在せず、具体的な対処は、個別の要件、つまり、使用する製品や規格そのものに関するものと、応用に関するものの両方を考慮しながら決めていくこととなる。もちろん、実際の個別の応用に関しては、当然ながら、お客様の方がベクターよりも豊富な知識をお持ちである。そして、AUTOSARの規格自体を一人ですべて把握することは、非常に困難だが、製品提供者であるベクターの知識を「プラグイン」として組み合わせ利用すれば、「あなたにとってのAUTOSARとは？」を効率的に確立していくことができる。

また、大規模なものを利用する場合には、外部リソースを利用することがすでに一般的になっているが、ベクターでも、導入支援やその他各種の技術サービスを行っており、必要なツールなどの準備がまだ完全に整っていないくとも、部分的に開発に着手できるような支援なども可能である。

今後も、VectorはAUTOSAR導入およびそれを利用した開発のサポートにより、皆様のECU製品の品質、開発効率の向上に大きく貢献することを目指す。

[This page intentionally left blank.]

