

# 車載向けソフトウェア・プラットフォームAUTOSARを利用したソフトウェア開発

Software Development Applying Automotive Software Platform AUTOSAR

針本 修次\*      池田 浩\*  
Shuji Harimoto      Hiroshi Ikeda

近年、車載ソフトウェアの開発規模は増大を続けおり、ソフトウェアを部品として汎用化し再利用することで開発を効率化する動きが広がりを見せている。本稿ではソフトウェアを汎用化するためのソフトウェア・プラットフォームであるAUTOSAR (Automotive Open System Architecture) の特徴とAUTOSARを活用した開発方法について概説する。

Recently, the scale of automotive software development has been dramatically increasing, and hence the idea to develop software effectively by re-using it as general parts has started to spread widely. We outline the software platform AUTOSAR that can be used to generalize automotive software and describe its development in this report.

## 1. 車載ソフトウェアの開発動向

近年、自動車の高機能化、低燃費化、軽量化、電動化が進み、これらの制御を実現する車載ソフトウェアの開発規模は年々増大を続けている。さらに、規模の増大に加えコスト削減や納期短縮の要求にも対応する必要もあり、安全性や品質を確保しつつ開発を効率化することが強く求められている。上記の課題を解決する方法の1つとして、ソフトウェアを部品として扱い再利用するという取り組みがある。この取り組みでは、自社開発のソフトウェア部品に加え、他社で開発したソフトウェア部品も対象とすることが可能となり、再利用性を向上させることができる。

本稿では上記のソフトウェアを部品化する際に、共通のプラットフォームとして利用するAUTOSAR<sup>(注1)</sup> [1]の特徴とAUTOSARを活用した開発方法について解説する。

## 2. AUTOSAR概要

AUTOSARとは、AUTomotive Open System ARchitectureの略で2003年に車載ソフトウェアの標準仕様の策定を目的に設立された組織、およびこの組織で策定された仕様、あるいはこの仕様に従って作成されたソフトウェアそのものの名称である。この組織は仕様の標準化のみを行いソフトウェアの提供は行わないため、AUTOSARを利用した開発をする場合は、ソフトウェアベンダーからAUTOSAR仕様に対応したソフトウェアを購入するのが一般的である。

\* オートモーティブ&インダストリアルシステムズ社  
オートモーティブ事業開発センター  
Automotive Business Development Center,  
Automotive & Industrial Systems Company

仕様の策定は、欧州の車メーカーや部品メーカーが中心となり、既に活動していたOSEK/VDX<sup>(注2)</sup> [2]などのリアルタイムOSのプロジェクトや、HIS[3]などの団体によるソフトウェア関連の標準化活動の成果を取り入れている。またAUTOSARは、「標準化においては協力、実装/利用にて競争」[1]の方針で策定された仕様である。その結果、現在では企業の枠を超えた標準化を推進し、世界中で180社を超える企業がAUTOSARに参画している。AUTOSARの仕様は2015年現在も更新活動が続いており、いくつかのバージョンが存在している。なかでも2008年にリリースされたR3.0は欧州での量産採用が拡大し、2009年にリリースされたR4.0からは自動車向け機能安全規格ISO26262[4]の考え方を取り入れ始めたことから、欧州だけでなく日本を含めた世界中の車メーカーでの採用が加速している。

## 3. AUTOSAR仕様の技術的な特徴

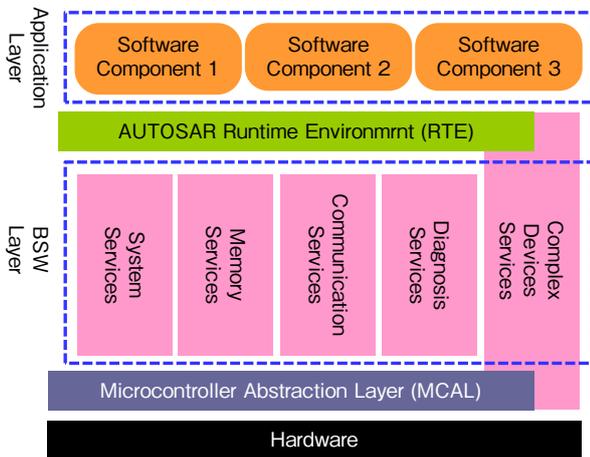
### 3.1 AUTOSARアーキテクチャによる階層分離

AUTOSARは、部品化するソフトウェアが特定のハードウェアに依存せずに再利用できるように、階層的に抽象化をするアーキテクチャを採用している。第1図にAUTOSARのアーキテクチャを示し、それぞれの階層の役割を第1表に示す。AUTOSARのアーキテクチャでは下位層から上位層に向かうほど抽象度が上がるため、最上位層のアプリケーション層はハードウェアに非依存の構成となる。このアプリケーション層内に特定の機能を有するSWC (SoftWare Component) を配置し、AUTOSARの仕様で標準化された機能を提供するBSW (Basic SoftWare)

(注1) AUTOSAR GbRの登録商標または商標。

(注2) Continental Automotive GmbHの登録商標または商標。

と合わせてシステムを構築する。つまりAUTOSARでは、1つのSWCを1つの部品として汎用化し、さまざまな機能をもったSWCを追加、変更することで、目的とするシステムを構築することが可能となる。



第1図 AUTOSARのアーキテクチャ  
Fig. 1 AUTOSAR architecture

第1表 各層の役割  
Table 1 Layer role

階層名	主な役割
Application層	システムの独自機能を実現する階層
RTE層 (Run Time Environment)	システムや通信インターフェースを抽象化する階層
BSW層 (Basic Software)	OS, メモリ, 通信機能など車載で利用される標準仕様を実現する階層
MCAL層 (Microcontroller Abstraction Layer)	ハードウェアのインターフェースを抽象化する階層

### 3.2 仮想機能バス (VFB : Virtual Function Bus)

AUTOSARでは、ハードウェアへの依存性が低いソフトウェア開発を実現し、ソフトウェアの再利用性を高めるための仮想機能バス (VFB) の考え方を取り入れている。すべてのSWCはVFBをもつ1つのシステム上で動作し、SWC間のインターフェースが、開発対象のシステム内で実現されるのか、外部のシステムとCAN (Controller Area Network) 通信などによって実現されるのかといった通信方式を意識せず、SWC間の接続関係のみを定義する。そのうえで実際のシステム構築の際は、全体の構成や制約などを考慮し、1つのシステムに配置するSWCを確定させる。これによりSWCはシステム環境に依存せずに自由に組み込み、再利用することが可能になる。

## 4. AUTOSARを適用した製品開発

### 4.1 新規システムへのAUTOSAR適用

新規にAUTOSARを利用してシステムを開発する場合、AUTOSARアーキテクチャに従って各階層内のソフトウェアを生成していく。実際の開発ではアプリケーション層とそれ以外の階層で大きく開発手法が異なる。

アプリケーション層の開発は、既に部品化されたSWCを利用する部分とSWCを新規に生成する部分に分かれる。既存のSWCを利用する場合はSWCを配置することで完了する。SWCを新規に生成する場合は、従来どおりのCコードの実装やMBD (Model Based Development) による自動生成コードを利用する。車載ソフトウェアで 사용되는標準的な機能 (スケジューリング機能, 通信機能, メモリ保存や保護機能など) はRTE以下の階層で提供されるため、ソフトウェア開発者は標準化されたインターフェースを使用することで、目的の機能が利用可能となる。このことからAUTOSARに準拠したインターフェースを有するSWCは、ハードウェアに依存しない高い移植性と保守性をもつこととなり、部品として汎用的に利用することが可能となる。

一方でアプリケーションより下の階層ではConfigurationと呼ばれる項目を設定することで開発を進める。アプリケーション層のSWCはハードウェアに依存しないが、下位層になるほどハードウェアに依存するため、一部のBSWやMCALは利用するハードウェアに合わせてConfigurationを設定し最適化する必要がある。Configurationはさまざまなカーメーカーの要望や既存の仕様を取り入れてきた経緯から、システムによって異なるが、およそ450種類以上の項目が存在し膨大な項目数を設定する必要がある。特にBSWのConfigurationではSWCに合わせた機能を提供するための設定が必要になるが、複数のConfigurationに依存関係が存在し、AUTOSARで実現する機能の仕様に準じたConfiguration設定になっていない場合は、ソフトウェアが動作しないこともあるため注意が必要である。基本的にAUTOSARでは、このConfigurationの設定をツールで実施する。ツールは入力されたパラメータ、スイッチ、関数名などのConfigurationから目的とするソフトウェアのCコードを自動生成する。Configurationは、従来のCコードベースの開発における実装だけでなく、ソフトウェアの設計行為にも当たるため、個々の設定内容や関連項目との整合性を有識者がレビューし、その妥当性を検証する。

### 4.2 既存システムへのAUTOSAR適用

既存のシステムにAUTOSARを適用する場合は、そのシ

システムアーキテクチャを考慮し効率的な開発を行う必要がある。以下ではAUTOSAR適用で検討すべき代表的な項目として、ICC (Implementation Conformance Class) とCDD (Complex Device Driver) について記載する。

ICCは、BSWで提供する機能に応じて3段階で定義されるAUTOSARのアーキテクチャの適応レベルである。例えば、RTE層のみを利用しAUTOSARのBSW層を利用せず、RTE以下をブラックボックスのように使用する場合は、ICC1を選択する。第1図に示すようなAUTOSARのサービスを構成する多くのBSWを使用する場合はICC3を選択する。なお、ICC2は、ICC1とICC3の中間に当たるが、制約が厳しく再利用性が低いため利用されることは少ない。ソフトウェア開発者はICCを適切に設定し汎用性と開発効率を両立させることが必要になる。

CDDは第1図に示すとおり、AUTOSARのインターフェースであるRTEやMCALを使わずにハードウェアからSWCまでを直接制御することが可能である。通常AUTOSARアーキテクチャへ既存ソフトウェアを搭載する場合は、既存システムを評価し、SWCで実現する機能とBSWで実現する機能に分割し、ソフトウェアをAUTOSARのアーキテクチャに再構成するため、CDDを利用することは少ない。ただし、実際にはソフトウェア改変時のリスクや処理速度を検討した結果、AUTOSARのアーキテクチャを利用することが困難な場合がある。そこで、既存のソフトウェアをCDD内で利用することによって、大きな修正をすることなく、AUTOSARを利用したシステムに組み込むことが可能となる。しかし、ハード依存性の高いCDDの利用は、開発するソフトウェア部品の再利用性を低下させることになるため、適用対象や範囲については慎重に検討する必要がある。

### 4.3 機能安全の適用

機能安全は、ハードウェアやソフトウェアの機能的工夫を追加することによって安全を確保するものである。そのため、安全に関連したソフトウェアが、誤動作を起こしたソフトウェアの影響を受けないように、安全部分とそれ以外の部分が独立し、干渉しないこと、すなわち、無干渉=FFI (Freedom From Interference) を保障する必要がある。その安全方策 (safety measures) の1つとして機能安全規格要求に対応したR4.0以降のAUTOSARを利用することも可能である。AUTOSARでよく利用される機能について第2表に示す。ただし、機能安全の観点から具体的な対応はシステムごとに異なるため方策決定には注意が必要である。

第2表 AUTOSAR機能を活用したFFI方策例

Table 2 Examples of FFI measures provided by AUTOSAR function

干渉	利用機能と方策例
タイミングと実行	周期実行処理監視機能、デッドライン監視機能、実行順序監視機能を利用して、ソフトウェアが意図した時間範囲内で動作していることを確認する。
メモリ	メモリ保護機能を利用して、ソフトウェアが意図した領域へのみアクセスすることを確認する
情報の交換	E2E (End-to-End) 通信機能を利用して、ソフトウェアが介在する通信に遅れ、ブロック改ざんなどが生じないことを確認する。

## 5. 今後の展望

AUTOSARは現在も更新活動を続けており、今後も新たな機能が標準化されAUTOSAR仕様として追加されることが考えられる。当社もAUTOSARのプレミアムメンバーとして最新の技術を取り入れ、部品化したソフトウェアを利用し、大規模ソフトウェアの開発と効率化に対応していく。また当社は大小さまざまな規模の車載システムを開発しているが、AUTOSARはある程度のハードウェアリソース (メモリー・CPU能力) を必要とすることから、現在のところAUTOSAR搭載は、比較的規模の大きいシステムが中心である。今後は安心、安全システムの観点から、商品群に合わせてAUTOSARをソフトウェアのプラットフォームとして利用し、展開していく予定である。

### 参考文献

- [1] “AUTOSAR,” <http://www.autosar.org/>, 参照 Apr. 16, 2015.
- [2] “OSEK VDX Portal,” OSEK/VDX, <http://www.osek-idx.org/>, 参照 Apr. 16, 2015.
- [3] “HIS - Hersteller Initiative Software,” <http://www.automotive-his.de>, 参照 Apr. 16, 2015.
- [4] ISO 26262シリーズ：Road vehicles - Functional safety, Ed. 1 (Part 1-9：2011, Part 10：2012) .