



自動車ソフトウェアの標準仕様 “AUTOSAR” の評価

堀川 健一*・目崎 元太・渡辺 章代
加 檜 武・松本 達治

Evaluation of Automotive Software Standard “AUTOSAR” — by Kenichi Horikawa, Genta Mesaki, Akiyo Watanabe, Takeshi Karo and Tatsuji Matsumoto — Automotive Open System Architecture (AUTOSAR) has established standard specifications of automotive software including layered software architecture and development methodology. We have made a prototype of our body ECU software in accordance with AUTOSAR standard specification to compare properties, such as ROM and RAM sizes and execution speed, with software produced by our current platform. In this report, model-based development methods are also evaluated along with AUTOSAR standard specification.

Keywords: AUTOSAR, RTE, basic software, model based development, automotive, ECU, standard specification

1. 緒 言

近年、自動車の電子制御化が急速に進んでいる。それに伴い、自動車1台に搭載されるソフトウェアの開発規模はこの20年間で約1,000倍に膨れ上がった⁽¹⁾。しかし、ソフトウェアの大規模化・複雑化は、生産性と品質の低下を招き、自動車産業界全体の深刻な課題となっている⁽²⁾。

この課題の解決のため、標準化団体AUTOSAR^{*1}が設立された。AUTOSARはソフトウェアアーキテクチャや開発メソッドロジなどの標準仕様を規定した⁽³⁾。

AUTOSARの標準仕様は非常に先進的であり、現状の課題をうまく解決できている。しかし、一方で各社が現在使用している基本ソフトウェアよりもオーバヘッドが大きくなるため、AUTOSARに移行するとマイコンの処理速度やRAMなどのメモリ容量が従来よりも多く必要になり、部品コストが増加すると考えられる。

我々は当社が製造しているボディECU（Electronic Control Unit、電子制御ユニット）のソフトウェアを、AUTOSAR標準仕様に準拠して開発した場合の、メリット・デメリットの評価が必要であると考え、試作評価を行った。本稿では、この評価結果について述べる。

2. 背 景

2-1 自動車のソフトウェア 近年、自動車では下記のような環境性能・安全性・快適性の向上が急速に進んでいる。

- ①環境性能：燃費向上・排出ガス低減のためのエンジン制御、ハイブリッド車等
- ②安全性：エアバッグ、アンチロックブレーキ等
- ③快適性：オートエアコン、自動変速機、遠隔制御ドアロック等

これらは電子制御により実現され、電子制御はボディ制

御、エンジン制御、パワートレイン制御、走行制御、情報システムなど多岐に渡り、①②③はその一例である。電子制御の中心であるECUの内部にはマイクロコントローラ（以下、マイコン）と呼ばれるコンピュータが組み込まれている。高機能化に伴い、1台の自動車に70個以上のECUが搭載されている場合もある。

このECUのソフトウェアは基本ソフトウェアとアプリケーションソフトウェア（以下、アプリケーション）から構成される。アプリケーションはECUの機能そのものと言ってよく、ボディECUのアプリケーションはドアロック、ヘッドライト、ワイパー等である。近年ではオートライト、オートワイパー、セキュリティ制御などとアプリケーションの多機能化が進んでいる。一方、基本ソフトウェアはECUそのものの機能というよりは、マイコンの管理サービス、リアルタイムオペレーティングシステム、通信サービスなど、アプリケーションを動作させるためのインフラストラクチャ的な部分である。

近年、自動車の高機能化が進んでおり、自動車の総原価に占めるエレクトロニクス関連部品の比率はハイブリッド車の場合では半分近くにまで高まっており、ECUの開発においてソフトウェアの開発工程が全行程の8割以上とも言われている⁽²⁾。このように自動車におけるソフトウェアは製品開発において重要な位置を占めるようになってきている。

そして、高機能化を実現するため、ソフトウェアの大規模化・複雑化が進展している。例えば当社製品のソースコード行数はこの10年間で10倍近い増加を示しており、一つのECUのソフトウェアのソースコードが20万行に達するものもある。大規模化と複雑化は生産性と品質の低下を招き、業界全体の深刻な課題となっている。例えば、20万個の部品からなる機械がECUのソフトウェアであると考えれば、その開発において生産性と品質を維持することの困

難さは想像できるであろう。

2-2 自動車ソフトウェア開発が抱える課題 続いて、自動車のソフトウェア開発が抱える課題について具体的に述べる。

①基本ソフトウェアの開発：従来、基本ソフトウェアは部品メーカーやカーメーカーにより独自に開発されており、当社も独自に開発している。しかし、自動車の商品力は基本ソフトウェアの優劣よりも、アプリケーションの優劣で決まる部分が大きく、差別化が図りにくい基本ソフトウェアの独自開発は負担になっていた。そこで、基本ソフトウェアを標準化して業界全体で共有することが考えられた。こうすることで、基本ソフトウェアの開発に投じていた人的資源をアプリケーションの開発に投入できる。過去に、欧州ではOSEK/VDX⁽⁴⁾によるオペレーティングシステムや通信サービスの標準化が行われたが、より広い範囲での標準化による開発の更なる効率化が必要と考えられた。

②アプリケーションの可搬性の確保：自動車のシステム開発では多くの場合、直近の同車種や他車種のシステムをベースにして、ECUの個数、搭載箇所、ECUへのアプリケーションの割付を見直す形で開発される。例えば、ベースとした車種では2個のECUに分割して割付けていたアプリケーションを、新しい車種では1個のECUにまとめる割付をして開発する場合がある。また、逆に分割して開発する場合もある。ECUの数・規模・搭載機能数が増加するにつれて、アプリケーションの割付の変更規模と複雑さが増す。このような割付変更の際、従来のソフトウェア開発ではアプリケーションの修正が必要だった。もし、アプリケーションを修正することなく、アプリケーションのECU間での割付を変更できれば生産性を大きく向上させることができる。ソフトウェア工学では、ソフトウェアの異なる環境への移しやすさを可搬性 (Portability)^{*2}という。

③企業間での仕様の伝達：従来、カーメーカーと部品メーカー間では、仕様書と呼ばれる文書により仕様が伝達されていた。自然言語主体の仕様書では、曖昧な部分や不完全な部分がどうしても発生し、それが不具合となり、手戻りや品質の低下を招いていた。

2-3 AUTOSARの特徴と利点 これらの課題を解決するため、欧州では自動車関連メーカーが中心となって、AUTOSARという標準化団体が設立された。AUTOSARは自動車のソフトウェアのアーキテクチャや開発メソッドの標準化を行うことで生産性と品質の課題を解決しようとした。以下に、標準化の概要とメリットを述べる。

①基本ソフトウェア：OSEK/VDXでは前述のように基本ソフトウェアの一部であるオペレーティングシステムと通信サービスについてのみ標準化された。AUTOSARでは基本ソフトウェア全般にわたり標準化を進めた。例えば、CPU動作クロックなどのマイコン

管理、AD変換やPWM出力、不揮発データの管理、後述する省電力制御機能なども標準化された。これにより、部品メーカーはアプリケーションの開発に更に注力できるようになった。

②RTE (Run Time Environment)：従来のアプリケーションは制御ロジックとアプリケーション外部とのインタラクションからなる。アプリケーションの可搬性を向上させるには、アプリケーションを制御ロジックのみにして、アプリケーション外部とのインタラクションは別モジュールに移すことが考えられる。アプリケーションの割付変更の際は、そのインタラクションのモジュールだけを修正すればよく、アプリケーションの修正は不要である。これはソフトウェアの分野では一般的な手法であり、当社独自の基本ソフトウェアでも既に同様の仕組みを導入している。AUTOSARではこの仕組みをRTEと呼び、標準化した。他にオペレーティングシステム、グラフィカルな開発ツール、ソースコードの自動生成機能なども併せて導入した。

③仕様記述の標準化：AUTOSARではアプリケーションのインタフェースを定義する定義ファイルの書式が標準化され、AUTOSAR準拠の開発ツールでグラフィカルに記述できる。従来の自然言語主体の文書の代わりに、この定義ファイルを授受することで、カーメーカー、部品メーカー間での仕様の伝達が明確になり、開発効率を向上することができる。

これらは当社の基本ソフトウェアよりも優れていると考えられ、実際にボディECUを試作して評価することが必要と考えた。

3. ボディECUにAUTOSARを適用した場合の問題点

一方、ボディECUにAUTOSARを適用した場合の問題点も考えられ、以下にそれらについて述べる。

3-1 計算量、ROMサイズ、RAMサイズの増加 製品の競争力を維持するためには、部品コストの低減が必要不可欠である。マイコンはECUの中で最も高価な部品であり、一般的にその価格はマイコンの動作クロック周波数が低いほど、ROMサイズ、RAMサイズが小さいほど低価格である。動作クロック周波数は単位時間あたりの計算量に比例する。計算量を低減できれば、動作クロック周波数を低く抑えることができ、低コスト化できる。そこで、当社の基本ソフトウェアも計算量、ROMサイズ、RAMサイズを小さくすることを重要な要件の一つにして開発している。

また、AUTOSARでは仕様の標準化に伴うオーバーヘッドが大きい。そのため、マイコンの処理能力を高く、メモリ容量を大きくする必要があり、大幅なコスト増につながると考えられる。

更に、AUTOSARではオペレーティングシステムが必須

の構成要素になっており、これもオーバーヘッドが大きくなる要因と考えられる。オペレーティングシステムの利用により、大規模なECUソフトウェアを開発する場合には生産性を向上させることができる。しかし、小規模ECUでは大規模ECUよりも、コスト増の影響が大きくなるため、オペレーティングシステムを搭載しないのが一般的である。当社の製品でも小規模ECUではオペレーティングシステムを搭載していない。

3-2 省電力制御の実現性 ボディECUの特徴には低消費電流モードによる省電力制御機能がある。エンジンECUなどのECUは基本的にはエンジン動作時にだけ動作すればよい。一方、ボディECUはドアロック、室内灯、ヘッドライト、セキュリティなどの機能を持つが、これらの機能はエンジン停止時にも動作が必要である。しかし、エンジン停止時は発電機が発電しないため、バッテリー電力のみで動作しなければならない。ドアロックやセキュリティに至っては、ユーザが車両を離れている間も動作を続ける必要があり、長期間バッテリーの電力のみで動作することが求められる。しかし、バッテリーを過度に消耗してしまうと、次回のエンジン始動時にエンジンがかからなくなってしまう問題点がある。

従って、エンジン停止時のボディECUの消費電流は小さいほどよい。この時、消費電流の大部分はマイコンによるものである。そしてマイコンの消費電流は、マイコンの動作クロック周波数が高いほど大きい。動作クロックを停止するか、極めて低い周波数にすることで消費電流を大幅に低減でき、この状態を低消費電流モードと呼ぶ。

そこで、ボディECUではバッテリーの充電量を温存しつつ、長期間の動作を続けるために、マイコンの処理が不要と判断した時に、低消費電流モードに移行する。そして、通常の動作が必要になった場合は通常の動作クロック周波数に復帰する。

AUTOSARでは基本ソフトウェアで省電力制御機能を用意している。しかし、前述したように、標準仕様に準拠した基本ソフトウェアにはオーバーヘッドがある。我々はこのオーバーヘッドにより、通常モードへの復帰判断にかかる時間が増大するおそれがあると考えた。この時間を省電力制御時の応答時間と呼ぶことにする。この応答時間が増加すれば、ユーザの行った操作に対する応答が遅くなる。ある程度以上遅くなると、ユーザは使い心地が悪く感じたり、故障が発生したと感じたりすることになり、商品性の観点から好ましくない。

以上のように、省電力制御時の消費電流と応答時間は省電力制御の重要な評価項目である。

4. 評価項目の検討

試作評価にあたって、まず、評価項目の検討を行った。評価項目の整理には、ソフトウェア品質の評価に関する国

際規格であるISO-9126⁽⁵⁾を用いた。ISO-9126では品質モデルを機能性 (functionality)、信頼性 (reliability)、使用性 (usability)、効率性 (efficiency)、保守性 (maintainability)、可搬性 (portability) の6特性と約30の副特性で表現している。我々はこの品質モデルを用いて、ボディECUソフトウェアの評価項目を検討した(表1)。今回は新規開発に限定した評価項目を抽出したため、評価項目が機能性、信頼性、効率性に集中している。実際の開発では仕様変更や、他の車種への流用作業が発生するため、保守性や可搬性が重要になり、これらの評価も必要になる。

表1 ボディECUソフトウェア評価項目(概略)

品質特性	副特性	評価項目の例
機能性	適切性	入力SWのノイズ除去の適切性
	正確さ	処理周期のばらつき(最大遅れ時間)、モータ出力時間・モータ停止時間の正確性、断線判定の精度
信頼性	フォールトトレランス	通信異常時に誤動作しないこと
	復元力	通信・電源復帰時に動作回復するまでの時間
効率性	時間挙動	起動時間、各機能の応答時間、低消費電流モードでの各機能の応答時間
	資源活用度	CPU負荷率、ROM使用量、RAM使用量、低消費電力モード時の消費電流

また、評価項目の検討とあわせて今回の試作開発での試作ECUの仕様を検討し、ドアロックとヘッドライトの制御機能の搭載を決定し、詳細仕様を定めた。

5. 評価結果

図1に試作したソフトウェアの設計の概念図を示す。また、写真1の確認用のハードウェアを開発し、これを用いてテストと測定を行った。

表2に主な評価結果を示す。

製品版のECUでは消費電流が小さくなるように電子部品を選定して、最適なハードウェア設計にするが、今回の評価用ハードウェアでは設計時間の問題からその様な設計にできなかった。そのため、今回は消費電流の評価を行っていないので別途行う必要がある。

5-1 処理周期のばらつき 処理周期の正確さは機能の安定した動作には欠かせないので、理想的な周期に対する最大遅れ時間を機能性の正確さの項目の一つとした。測定したところAUTOSAR導入後も0.1ミリ秒であり大きな遅れは発生しなかった。従って、ボディECUに適用しても問題ないと考えられる。なお、従来品は測定精度の0.1ミリ秒未満であった。この差はオペレーティングシステムの導入により発生したものと考えられる。

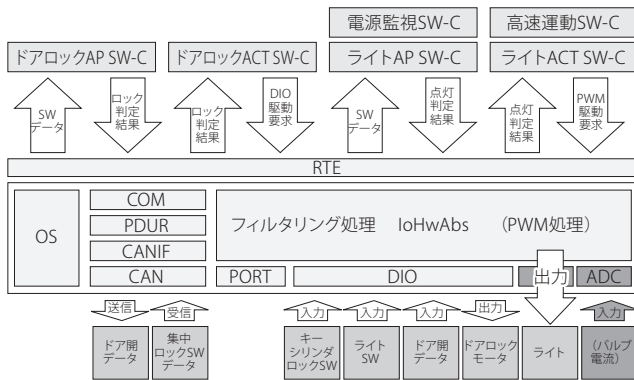


図1 試作したソフトウェアの設計の概念図

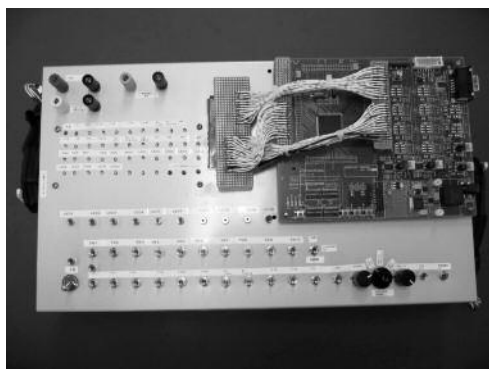


写真1 開発したボディ ECU 評価用ハードウェア

表2 主な評価結果

品質特性	副特性	評価項目	測定結果
機能性	正確さ	処理周期のばらつき (最大遅れ時間)	従来品: 0.1ms未満 (測定不能) 測定値: 0.1ms
効率性	時間挙動	低消費電流モードでの各機能の応答時間	従来品の約5倍 (通信によるウェイクアップの場合) 従来品の約3倍 (スイッチ入力によるウェイクアップの場合)
		マイコン負荷率	従来品の1.8倍 (定常時) 従来品の2.5倍 (高負荷時)
	資源効率性	ROM使用量	従来品の3.5倍
		RAM使用量	従来品の2.2倍

5-2 資源効率性 ROMサイズ、RAMサイズ、マイコン負荷率はAUTOSAR導入により約2~4倍に悪化した。そこで詳細に分析すると、アプリケーションはほとんど増加しておらず、基本ソフトウェアの増加が原因であることが分かった。具体的には、アプリケーションのROMサイズは従来品もAUTOSAR版も約3Kバイトであり差が

なかった。そして、基本ソフトウェアによる増分の大部分はオペレーティングシステムであった。

従って、オペレーティングシステムの導入により、開発が容易になるメリットはあるものの、AUTOSARの基本ソフトウェアを搭載、即ちオペレーティングシステムの搭載により、以前よりも多くのメモリ資源が必要になり、現状ではこのコストアップは受け入れられない。将来、ソフトウェアの規模が大きくなれば、AUTOSARによる必要メモリの増加は相対的に小さくなり、コストアップも小さくなるので、導入の可能性が増すと思われる。

5-3 低消費電流モードでの応答時間 低消費電流モードでの応答時間を測定したところ従来品に比べて3~5倍も長く、応答性がかなり悪くなっていることが判った。その理由を分析したところ、二つの原因が分かった。

①排他制御の処理時間の増加

基本ソフトウェアやアプリケーションでは、割込み処理による異常動作を防ぐために排他制御を行う必要がある。AUTOSARの基本ソフトウェアはいかなる使い方でも異常動作が発生しないよう、従来品の基本ソフトウェア以上に排他制御を多用している。さらに、排他制御の実現にオペレーティングシステムが提供する排他制御機能を利用しており、その1回あたりの排他制御の処理時間が従来品の基本ソフトウェアよりも長い。

さらに、低消費電力モード中はマイコンの動作クロック周波数を低く設定しており、排他制御を動作クロック周波数が低い状態で実行するため、実行時間が大幅に増加していた。

②マイコンの動作クロックを扱うドライバの問題

マイコンの動作クロック周波数の切替は基本ソフトウェアが行い、今回評価したAUTOSARではあらゆる周波数への切替を保証している。一方、半導体メーカーはマイコンの安定動作のため、切替時の安全な処理順序を開示している。特定の組み合わせの切替では、これらの指示に従わなくてもよい場合もあるが、AUTOSARの基本ソフトウェアでは半導体メーカーの指示に忠実に従った切替処理が用意され、組み込まれている。従来の基本ソフトウェアでは使用する周波数で必要な指示のみに対応し、必要かつ最小の処理で実現されているのにと比べると、AUTOSARはその点オーバーヘッドが大きい。また、AUTOSARでは切替処理の途中で安全な切替のため我々が使用している低消費電流モードの動作クロック周波数(約32KHz)よりもさらに低い周波数(約8KHz)で動作させていた。以上により、切替処理にかかる時間が大幅に増加していた。

実際の開発ではこれらのオーバーヘッドを軽減させて、応答性を向上しなければならない。そのために、基本ソフトウェアの開発元と協力して、基本ソフトウェアを最適化(カスタマイズ)する作業が重要と考えられる。

6. AUTOSARにモデルベース開発手法を併用した時の評価

6-1 モデルベース開発手法 現行のソフトウェア開発手法は設計書を中心とした開発手法である。仕様書からソースコードに至るまでに数種類の設計書を記述するが、設計書は動作させて確認することができない。ソースコードが完成した後に初めて動作させながら設計の正しさを確認することができるようになる。

一方、モデルベース開発手法では開発の初期段階から実現すべき機能をモデルと呼ばれる設計図を作成し、その後の各設計段階でもモデルを作成する。モデルはツールを用いて動作させることができ、それにより設計の正しさを確認することができる。開発の初期段階から動作に基づく検証が可能になるため、不具合の早期発見が可能になり、開発効率や品質を向上させることができる。また、モデルから自動コード生成を行うこともできるので、ソースコード作成工程の省力化も可能である。

6-2 AUTOSARとモデルベース開発手法 AUTO SARは前述のように、設計データの受け渡しの書式を標準化しており、アプリケーションのインタフェース定義も標準化されている。設計者はAUTOSARのシステム記述ツールを用いてアプリケーションのインタフェースを記述し、そのデータをAUTOSAR標準仕様に準拠したモデルベース開発手法のツールに取り込んで使うことができる。開発を進めていく中で、モデルベース開発手法のツールとの連携がAUTOSARの強みであることが分かり、モデルベース開発手法併用の効果検証も必要と考えた。そこで、これまで報告した試作と同一仕様のアプリケーションをモデルベース開発手法併用の開発環境を用いて設計して、連携の効果を検証した。

6-3 モデルベース開発手法併用の評価結果 モデルベース開発手法併用評価の測定結果を表3に示す。

表の「従来手法」とは当社製の基本ソフトウェアを用いて、従来手法によりソフトウェアを開発した場合である。

表の「モデルベース開発手法」とはAUTOSARの基本ソフトウェアとモデルベース開発のツールを併用した場合である。

表の「ハンドコード行数」はプログラマが直接記述したソースコードの行数である。「従来手法」では基本ソフトウェアは当社で開発しているので、その行数に含んでいる。一方、AUTOSAR版では、基本ソフトウェアは基本ソフトウェアを開発を専業とする企業から購入して利用したためその行数に含んでいない。

従来手法のハンドコード行数が22,641行であるのに対して、「モデルベース開発手法」ではその約1%の305行であり、ハンドコードの必要がほとんどないことが確認できた。今回、プログラマがソースコードを記述したのは、電源投入時のマイコンの初期化などごく一部に限られた。

この削減は以下のような理由による。

①基本ソフトウェアを購入したため、基本ソフトウェア

のハンドコードが不要であった。なお、基本ソフトウェアはオブジェクト供給であり、行数は不明である。そのため、表ではハンドコード以外の行数は記載していない。

②アプリケーションはモデルベース開発技法の自動コード生成によりハンドコードが不要だった。

③基本ソフトウェアとアプリケーションの中間に位置するRTEは、AUTOSARのRTE生成機能によりソースコードが自動で生成できたため、ハンドコードが不要だった。

ROM、RAMサイズをアプリケーション部分に限定して比較すると、表4のとおり、従来手法よりも若干小さくなっている。なお、ソフトウェア全体では表3に示すとおり、従来の2～3倍と非常に大きくなっており、この要因は主にAUTOSARの基本ソフトウェアである。

表3 モデルベース開発手法併用時の測定値 (全体)

	従来手法	モデルベース開発手法
ハンドコード行数	22,641行	305行
ROMサイズ (Kbytes)	18.4	53.6
RAMサイズ (Kbytes)	3.9	7.9

表4 モデルベース開発手法併用時の測定値 (アプリ部分)

		従来手法	モデルベース開発手法
ソースコード行数	ハンドコード	2,524行	0行
	自動生成	0行	4,537行
	合計	2,524行	4,537行
ROMサイズ (bytes)		3,144	3,056
RAMサイズ (bytes)		92	76

アプリケーションのROM、RAMサイズが従来手法よりも若干だが小さい理由は以下のように考えられる。

プログラマがソースコードを直接記述する従来手法の開発では、不具合になりやすいソフトウェア設計を排除する目的などのため、コーディング規約と呼ばれるソースコードの記述ルールが設けられている。例えばC言語の1つの関数の行数に上限を設定している。大きすぎる関数は、可読性が低下するため、不具合を発見、除去しにくくなり、保守性が低下するからである。コーディング規約を適用した場合、一般にはより多くのROMが必要になる。例えば1つの大きな関数を複数の小さい関数に分割して記述することが必要になる。そして、一般的には関数間での関数呼び出しが増加するためプログラムのROMサイズが増加するのである。

一方、自動コード生成の場合、ソースコードの作成は自動コード生成ツールにより行われ、ソースコードの作成の過程で人為的な不具合が入ることはない。また、保守の際も基本的には自動コード生成ツールに入力するモデルに対して改変を行うため、直接ソースコードを読む必要がない。これらの理由から、自動コード生成による開発では一般的にコーディング規約を適用する必要がない。但し、自動コード生成ツールには様々な設定項目があり、これを調整することでコーディング規約にある程度従った、可読性の高いソースコードを生成させることも可能である。

今回は自動コード生成でそのような調整をしなかったため、可読性は若干低下したが、その分ROM、RAMサイズを抑えられた。なお、現在の自動コード生成ツールでも人間のプログラマに劣る場合もあり、そのような場合ではROM、RAMサイズが増加する。我々は他の事例でもアプリケーションの試作評価を行っており、その事例では自動コード生成のROMサイズが従来手法よりも大きくなる場合があることを確認している。

また、今回の試作では、ツール間の連携の有効性も確認できた。即ち、AUTOSARのシステムを記述するツールとモデルベース開発ツールの間の設計データの交換がAUTOSARによるデータ形式の標準化により、電子的に行えることが確認できた。連携する複数の開発ツールのことをツールチェーンというが、今後の自動車のソフトウェア開発の分野ではこうしたツールチェーンによる開発が急速に発展し、更に効率的な開発が進められていくものと考えられる。

7. 結 言

我々は自動車ソフトウェアの国際的な標準仕様であるAUTOSARに準拠したボディECUソフトウェアを試作した。実際にソフトウェアを開発することで、世界最先端の標準仕様とソフトウェア開発技法を獲得することができた。また、マイコン負荷率などを定量的に測定し、部品コストの増加につながるため、現状のソフトウェア規模ではAUTOSAR導入によるメリットは少ないと感じた。だからこそ、基本ソフトウェアのオーバーヘッドを軽減させるなどしてコストを削減することが重要であると考えている。しかし、ECUソフトウェアの開発経験の少ない開発チームでもソフトウェアを開発できたことから、優れたアーキテクチャや開発手法により生産性を向上させられることを実感した。さらに、本稿後半で述べたように、モデルベース開発手法との親和性の良さも確認できた。これらの評価結果は、当社の今後のECU製品のソフトウェア開発に生かしていく所存である。

用語集

※1 AUTOSAR

Automotive Open System Architectureの略。自動車用ソフトウェアの部品化および共通化を目的とした団体。ドイツのDaimlerChryslerやBMW AG、Robert Bosch GmbHなどが中心になり設立。現在は、日本の多くの自動車関連企業も参加している。

※2 可搬性

移植性という訳語を使う場合もあるが、本稿では可搬性で統一している。

・OSEK/VDXは、ドイツContinental AGの米国及びその他の国における商標または登録商標です。

参 考 文 献

- (1) 小川計介、「標準化で開発効率を高める車載ソフト巨大化に立ち向かう」、日経Automotive Technology 2007年11月号、pp82-97
- (2) 徳田昭雄、田村太一、「車載ソフトウェアの標準化とAUTOSARの動向」、立命館経営学、第45巻第5号、PP.153-169, Jan (2007)
- (3) “AUTOSAR specification” R2.1, 3.0, 3.1 (AUTOSAR発行の標準仕様書)、<http://www.autosar.org>からダウンロード可能
- (4) “OSEK/VDX specification” (OSEK/VDX発行のOSEK OSなどの標準仕様書)、<http://www.osek-vdx.org/>からダウンロード可能
- (5) JISハンドブック2003 JIS X 0129-1:2003 (ISO-9126)、日本規格協会

執 筆 者

堀川 健一* : (株)オートネットワーク技術研究所
ソフト開発センター
自動車ECUのソフトウェア開発に従事



目崎 元太 : (株)オートネットワーク技術研究所 ソフト開発センター
渡辺 章代 : (株)オートネットワーク技術研究所 ソフト開発センター
加槽 武 : (株)オートネットワーク技術研究所 ソフト開発センター
主任研究員
松本 達治 : (株)オートネットワーク技術研究所 ソフト開発センター
センター長

*主執筆者