WHITE
PAPER

AUTOSAR コーディングガイドライン
導入のすすめ

www.prqa.com

PRQA
Programming Research

はじめに

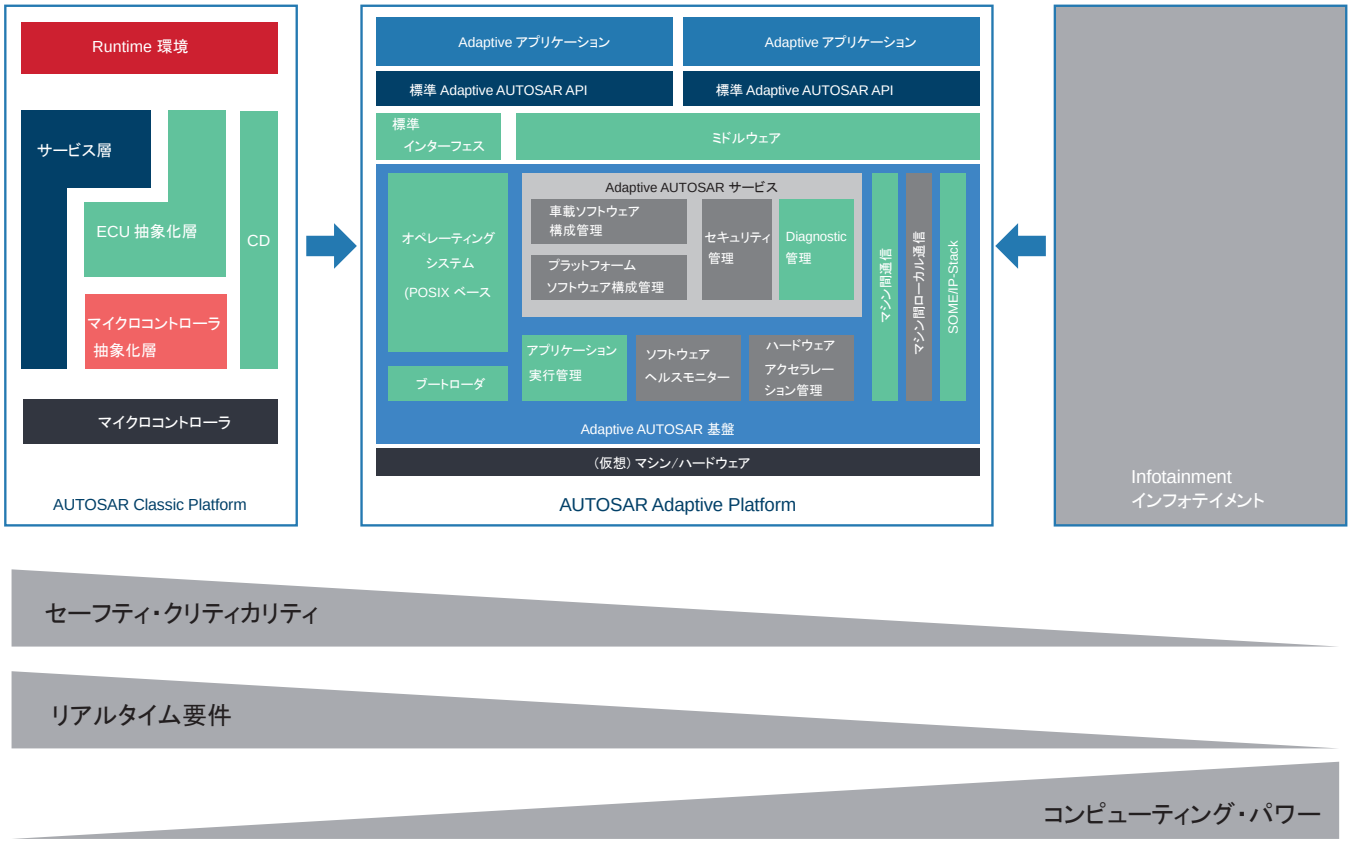
自動車業界においてのソフトウェア開発の重要性は、年々高まっている。安全性・環境性・利便性に対する要件は厳しさを増し、その結果として、自動車に使用される電子機器の数は急激に増加している。新しい技術の90%はソフトウェア駆動の電子部品に頼っており、それら部品の開発費は自動車の開発費全体の40%にも及んでいる。早くなる一方の開発ペース、新しい機能や制御ユニットの継続的インテグレーションに対する要求に対応することは、自動車メーカーにとっての大きな課題となっている。 このホワイトペーパーは、AUTOSARによって新たに策定されたコーディングガイドラインの概要を説明し、そのガイドラインに準拠するための方法について助言するものである。

AUTOSARとは?

AUTOSAR (AUTomotive Open System ARchitecture)は、標準的に使用でき、かつ将来においても有効なソフトウェアの基本要素、インターフェース、バスシステムを作り出すことで、自動車メーカーが開発コストを抑えつつ、複雑化の一途を辿っている車載システム要件に対応できるよう支援することを目指している団体である。車載ECU用の標準化オープンソフトウェア・アーキテクチャを開発している。

180以上の自動車メーカー、サプライヤ、ツール/半導体ベンダーがAUTOSARパートナーとしてこの活動に参加しており、主要メンバーには、BMW、ボッシュ、コンチネンタル、ダイムラー、フォード、ゼネラルモーターズ、PSA、トヨタ、フォルクスワーゲンが名を連ねている。

AUTOSARによって最初に開発されたオープンアーキテクチャである「Classic Platform」は、ベーシック・マイクロコントローラに実装されるセーフティクリティカルで厳格なリアルタイム要件を持つ自動車機能用である。それに対して、AUTOSARが新たにコネクテッドカーと自動運転車用に開発した「Adaptive Platform」は、急激に増大を続けるコネクテッドカーや高度自動運転技術に対する市場の要求に対応することを目的としている。「Adaptive Platform」には、外部メモリ付き並列処理・高帯域幅通信型の高性能32-/64-ビットのマイクロプロセッサが使用されている。



「Adaptive Platform」規格に従って開発されたソフトウェアは 、「Classic Platform」規格に従ってビルドされた既存システムへの統合が可能である。

「Classic Platform」は、明示的にC言語、C++言語およびJava言語で書かれたソースコードへの実装が考慮されていたが、主に使用されたのはC言語であった。それに対して、「Adaptive Platform」内で使用されるAPIは、C++言語で定義されており、AUTOSARが「Adaptive Platform」のコンポーネントにおいてC++言語の使用を推奨していることを示唆している。

C言語およびC++言語は、車載向け組込みシステム開発で最も使用されている言語である。主な理由として、これらのプログラミング言語が直接的・決定的なハードウェア制御を可能とする点や、開発者が柔軟に扱える点が挙げられる。しかし、これらのメリットは同時にリスクにもなりえる。なぜなら、未定義の動作を含むコードや、想定された以外のハードウェア上でコンパイル・実行された際に動作が保証されないコードがコンパイルされてしまう可能性があるからである。経験豊富な開発者であっても、不注意から問題となるコードを書いてしまう可能性はある。

AUTOSARコーディングガイドラインとは?

AUTOSAR softwareを実装して書かれたコードの安心・安全を確保するために、PRQA社は、AUTOSARから開発パートナーとして“クリティカルな安全関連システムにおけるC++14 言語使用についてのガイドライン”（以下、「ガイドライン」）[1]策定のためのワーキンググループへの参画を求められた。

AUTOSARで唯一の静的解析ツール開発パートナーとして、PRQA社はC++言語に関する専門知識と過去30年以上に渡る経験から得られたソフトウェア開発のベストプラクティスを活かし、この活動に貢献した。

AUTOSARガイドラインでは、342個のコーディングルールが定義されている。そのうち154個には、すでに広く使われているMISRA C++ 規格のコーディングルールが利用されており、131個はその他の良く知られたコーディング規格（PRQA社のHigh Integrity C++など）をもとに作られている。残りの57個のルールは、AUTOSARの研究やその他の資料がベースとなっている。過去のいくつかの規格では禁止されていたが、このガイドラインでは許可されている言語機能（動的メモリ、例外、テンプレート、継承、仮想関数など）も存在する。その代わりに、ガイドラインではこれらの言語特徴の安全な利用を確実にするためのルールが設定されている。

AUTOSAR 開発原則の1つに、標準化と並行しての複数仕様の有効化がある。Adaptive Platformは、C++言語で記述されたAUTOSARの内部実装（Adaptive Platform Demonstrator）を利用して検証されている。Demonstratorのソースコード品質を確保し、コーディングガイドラインへの適合を確認する目的で、PRQA社のQA·C++解析ツールが利用された。

なぜAUTOSARコーディングガイドラインが必要なのか？

AUTOSARガイドラインが策定される以前は、セーフティクリティカルなソフトウェア開発における新しいC++言語規格(C++11 と C++14) の使用に関する適切なコーディングガイドラインが存在していなかった（不完全であったり、古いC++言語対応であったりと、セーフティクリティカルな用途には適さないものしか存在していなかった。）自動車業界で最も広く使われていたC++コーディングガイドライン（MISRA C++:2008）も、C++03対応で14年以上も前に策定されたのだった。

C++03 が導入されてから以下の様な変化があり、これらを鑑みて、AUTOSARプロジェクトでは、MISRA規格の妥当性が低く評価された。

1. C++言語の進化
2. コンパイラの改善
3. テスト、検証および解析ツールの改善
4. ISO 26262 自動車機能安全規格の誕生
5. 以下のような追加のガイドラインによって、安全・安心に関する専門知識がより広く浸透：
   - High Integrity C++ (PRQA)[2]
   - Joint Strike Fighter Air Vehicle C++ (Lockheed Martin)[3]
   - CERT C++ (Carnegie Mellon)[4]
   - C++ Core Guidelines (Bjarne Stroustrup and Herb Sutter)[5]

AUTOSAR は既存のMISRA C++規格の拡張版としての使用を念頭にガイドラインを作成しており、新たにガイドラインで追加されたルール、更新されたルールおよび時代遅れになっているMISRAルールを明記している。

誰がAUTOSARコーディングガイドラインを必要とするのか？

ガイドラインの目的は、次のように書かれている: 「主な用途は自動車向けであるが、そのほかの組込み用途にも用いることができる・・・ AUTOSAR C++14コーディングガイドラインがターゲットとするのは、POSIXやその種のOSを用いて32/64ビットマイクロコントローラ上で起動し、C++14言語を効率的かつ網羅的にサポートする高度な組込みマイクロコントローラである。」

そのため、PRQA社ではC++14で組込みソフトウェアを開発しているすべての組織でこれらのガイドラインの導入を検討することを推奨している。

どのようにしてコードを確実にAUTOSARガイドラインに準拠させるか？

従来、エンジニア達は必要な規格に従ってコードが書けているかを、かなりの工数をかけて手動でコードレビューを行うことで確認していた。しかし、この方法は間違いが起こりやすく、また、今日の大規模で複雑なコードベースを扱うことは想定されていなかった。幸いにも、これらのチェックは現在ではツールを利用することで自動化することができる。静的解析ツールはこの用途のために作られたものであり、コーディングガイドライン違反を報告するだけでなく、詳細にコードを分析することにより未定義、未規定、コンパイラの処理系依存の動作を特定する機能を備えている。全ての考えられるプログラムの実行パスを解析し、ランタイムエラーの可能性をユーザに知らせる。テストですべての実行パスを網羅するの現実的ではないため、テストでは発見できない問題がツールでは発見されることが多い。安心・安全で信頼性のあるソフトウェア開発のためには、静的解析ツールは欠かせない。

PRQA社の静的解析ツール（QA·C++）が、AUTOSARのDemonstrator ソースコードの品質確保とコーディングガイドラインへの適合のために使用されたことにより、いくつもの価値ある学びが得られた。それらの学びとPRQA社のガイドライン策定への貢献に基づいて開発に成功したのが、AUTOSARガイドライン準拠のソフトウェア開発に最適化された唯一の静的解析ソリューション（PRQA社のAUTOSAR C++コンプライアンスモジュール）である。

AUTOSAR C++コンプライアンスモジュールは、QA·C++の拡張機能として提供され、特別な設定無しにAUTOSARガイドラインへの適合度を評価することができる。開発チームが中規模・大規模である場合は、コード品質を集中的に管理できるPRQA社のQA·Verifyを利用することで、プロジェクトの期間中、コード品質の追跡とレポート化に加えて、すべてのチームメンバーが確実にコーディングガイドラインを適用できるようになる。

まとめ

AUTOSAR規格は、現状の機能領域間の障壁を最小化することによって、次世代自動車アプリケーションの基盤となるだろう。関連するハードウェアからほぼ独立した形にし、システム内の”機能”と”異なる制御ノードに至る機能ネットワーク”へのマッピングを可能とすることで、この最小化を実現する。自動車業界向けに開発されたガイドラインではあるが、C++14を利用して組込みソフトウェア開発を行っている組織であれば、業界を問わず利用できる。どのようなアプリケーションであっても、PRQA社の静的解析ツール（QA·C++）は、ミスなくコーディングガイドラインに適合したコードが確実に書けるよう支援する。

参照

1. **High Integrity C++:** http://www.codingstandard.com
2. **Joint Strike Fighter Air Vehicle C++:** http://www.stroustrup.com/JSF-AV-rules.pdf
3. **CERT C++:** https://www.securecoding.cert.org/confluence/pages/viewpage.action?pageId=637
4. **C++ Core Guidelines:** https://github.com/isocpp/cppcoreguidelines
5. **AUTOSAR Guidelines:** https://www.autosar.org/fileadmin/user_upload/standards/adaptive/17-03/AUTOSAR_RS_CPP14Guidelines.pdf

PRQA社について

1985年の設立以来、PRQA社は自動車・航空・防衛・輸送・金融・医療機器・エネルギー業界において、ソフトウェアコーディング・ガバナンスのパイオニア企業です。小規模なベンチャー企業から世界的大企業に至るまで、様々な企業のソフトウェア開発を幅広くサポートしており、高度なコード解析・堅牢な不具合検出・様々な分野（機能安全からアプリケーションセキュリティまで）における業界標準コーディング規格および独自コーディング規格への準拠支援のためのソリューションを提供しています。

PRQA社の業界トップクラスのソリューションであるQA·CとQA·C++は、一般的なプログラミング言語の静的解析を非常に細部まで実施します。各マシンで解析する場合であっても、品質管理システムのQA·Verifyを利用して中央サーバーで集約的に解析する場合であっても、不具合の早期発見・修正をデスクトップとサーバーの両方で可能にします。また、不具合修正の進捗・履歴を可視化し、プロジェクトマネージャーによる徹底した管理をサポートします。

www.prqa.com