



특집 06

차량전장용 소프트웨어 프레임워크 AUTOSAR 테스트 동향 분석



김광훈 (한국정보통신기술협회)

-
- 목 차 »
1. 서 론
 2. AUTOSAR 소개
 3. 차량전장용 소프트웨어 테스트 특징
 4. AUTOSAR 테스트 동향
 5. 결 론
-

1. 서 론

현대의 자동차는 사람들에게 더욱 편리하고 지능적인 기능을 제공하기 위해 전자화(Electronic)되고 있다. 이전에는 엔진, 자동변속기, 브레이크 같은 자동차 부품을 기계적으로 제어했으나 최근에는 ECU(Electronic Control Unit)라고 불리는 전자제어 장치를 통해 자동차 부품을 전자적으로 제어하게 되는 것이다. 일례로 그랜저 XG나 TG의 경우 30여종 이상의 ECU가 사용되고 최신 제네시스 모델의 경우 40여종 이상의 ECU가 사용된다. 해외에서도 LEXUS나 BMW 최신 모델의 경우 100여종 이상의 ECU들이 분산 네트워크를 구성하고 있다^[1]. 첨단기술을 많이 적용한 자동차일수록 더 많은 ECU를 내장하는 것을 볼 수 있다.

자동차에 탑재되는 ECU 개수가 증가하고 자동차 구조가 복잡해 짐에 따라 이를 제어하기 위한 소프트웨어의 크기와 복잡도 또한 함께 증가

하고 있다. BENZ의 경우 5백만 라인 이상의 코드가 사용되고 있으며^[2], GM의 경우 천만 라인의 코드를 사용한 모델을 출시하기도 했다^[3]. 차량용 소프트웨어 복잡도의 증가는 소프트웨어 개발 비용과 시간의 증가를 가지고 오게 되었다. 이러한 비용과 시간을 줄이기 위해 유럽의 자동차 업체들은 AUTOSAR(AUTomotive Open System ARchitectue)라는 표준화된 통합 소프트웨어 프레임워크를 만들었다. AUTOSAR는 하드웨어와 소프트웨어의 분리를 통해 소프트웨어의 재사용성, 확장성을 향상시켜 복잡한 소프트웨어를 빠르고 신뢰성 있게 개발할 수 있게 해준다.

소프트웨어 복잡도의 증가는 필연적으로 버그의 증가, 즉 품질의 저하로 이어지기 쉽다. 일례로 TOYOTA의 하이브리드 자동차인 Prius는 소프트웨어 버그 때문에 가솔린 엔진이 정지하는 문제가 있었고, 이를 해결하기 위해 총 75,000대를 리콜하였다. 또한 BMW의 745i도 엔진 밸브의 타이밍을 조정하는 ECU의 결함으로 주행 중

엔진이 꺼지는 문제가 있었고, 이로 인해 총 5,470대를 리콜한 바 있다⁴⁾. 이렇게 차량내의 소프트웨어의 비중이 커지고 이에 따라 소프트웨어 오류가 차량 고장에 상당한 원인을 차지하게 되면서 차량용 소프트웨어 테스트 및 품질보증에 대한 관심이 커지고 있다.

이에 본 고에서는 차량전장용 소프트웨어 프레임워크인 AUTOSAR에 대해 알아보고 AUTOSAR의 테스트 동향을 살펴보고자 한다. 먼저 2장에서 AUTOSAR의 탄생배경과 AUTOSAR의 목적, AUTOSAR의 구조에 대하여 알아본다. 그리고 3장에서는 차량전장용 소프트웨어 테스트 시 고려해야 할 특징에는 무엇이 있는지 알아본다. 4장에서는 AUTOSAR 테스트의 종류, 관련 표준, 테스트 도구에 대해 살펴본다. 그리고 마지막으로 5장에서는 본고의 결론을 맺도록 한다.

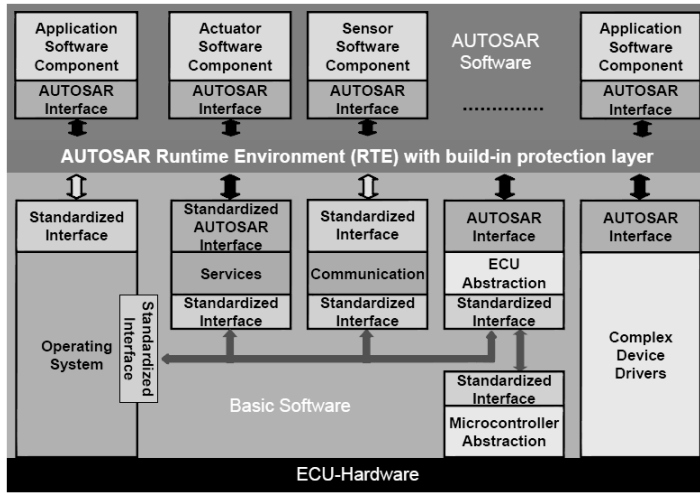
2. AUTOSAR 소개

자동차에 탑재되는 전장용 부품이 증가하고 그 구조가 복잡해짐에 따라, 자동차 업계는 기존 소프트웨어를 개발하는 방식으로는 전장용 소프트웨어의 개발 비용과 시간의 한계에 부딪히게 되었다. 그리고 전장용 소프트웨어는 하드웨어에 종속되어 있기 때문에, 하드웨어의 종류나 사양이 달라지면 관련된 소프트웨어를 수정하거나 재개발하는 경우도 빈번하게 발생한다. 이에 유럽계 자동차 업체들은 복잡하고 거대해지는 차량용 소프트웨어를 효과적으로 개발하기 위해 표준화된 통합 소프트웨어 플랫폼의 필요성을 느끼고 2003년 6월 AUTOSAR(AUTomotive Open System ARchitectue)라는 단체를 만들었다. AUTOSAR 단체는 차량용 소프트웨어 명세(Specification)를 표준화하여 AUTOSAR 표준안이라는 이름으로 발표하고 있다. AUTOSAR 표

준안은 2005년 6월 Release 1이 발표된 이래, 2009년 12월 Release 4까지 발표되었다.

AUTOSAR의 핵심 목적은 차량용 소프트웨어를 ECU나 네트워크 같은 기반 하드웨어 구조에 의존하지 않고 독립적으로 개발할 수 있도록 지원하는 것이다. 하드웨어와 소프트웨어의 분리는 소프트웨어의 독립성을 지원하고 재사용성, 확장성을 향상시켜 궁극적으로 소프트웨어 개발에 소요되는 비용과 시간을 단축시켜 준다. 또한, 소프트웨어 개발자는 응용 소프트웨어 개발에만 집중할 수 있어 보다 높은 품질의 소프트웨어를 만들어 낼 수 있다.

AUTOSAR 구조는 크게 SWC(Software Component), RTE(Run-Time Environment), BSW(Basic Software)의 3가지로 나누어지는 컴포넌트 기반 소프트웨어 개발(CBD: Component-Based Software Development) 형태를 가진다. 첫 번째 구성요소인 SWC는 (그림 1)에서 보는 것처럼 가장 상단에 위치하며 엔진, 자동 변속기, 브레이크 제어와 같은 응용 소프트웨어의 기능을 수행한다. SWC는 VFB(Virtual Functional Bus)라는 가상의 네트워크 상에 배치되어 적절한 ECU에 맵핑되어 동작한다. 따라서 SWC 개발자는 ECU나 네트워크 구성에 독립적으로 SWC를 개발하여 개발 비용과 시간을 줄일 수 있다. 두 번째 구성요소인 RTE는 SWC와 BSW 사이에 위치하며 SWC 사이 및 SWC와 BSW 사이의 데이터 교환을 위한 인터페이스를 제공한다. RTE는 SWC가 하드웨어에 종속되지 않는 독립성을 제공하기 위한 계층으로 모듈간의 통신 연결부 역할을 수행한다. 마지막 구성요소인 BSW는 가장 아래에 위치하며 OS(Operating System), Device Driver, Communications 같은 SWC가 필요한 작업을 수행하기 위한 서비스를 제공한다. 그 중 가장 중요한 OS 모듈은 자동차용 실시간 운영체제 표준인



(그림 1) AUTOSAR 구조도

OSEK OS^[5]를 기반으로 설계되었다. OS 모듈은 SWC나 RTE가 멀티태스킹이나 태스크 동기화 등 운영체제가 제공해야 할 서비스를 필요로 할 때 사용되며 AUTOSAR OS는 OSEK OS에 스케줄 테이블(Schedule Table) 개념과 OS-Application 개념이 추가되었다^[6].

3. 차량전장용 소프트웨어 테스트 특징

차량전장용 소프트웨어 테스트는 일반적인 PC용 소프트웨어 테스트와 다른 여러 가지 특징을 가지고 있다.

첫째로, 테스트 자동화 도구를 적용하기가 어렵다. 차량용 소프트웨어는 차량의 모델에 따라 상이한 하드웨어를 대상으로 하고 다양한 요구조건을 가지고 있기 때문에 하나의 일반화된 테스트 모델을 제공할 수 없다. 그래서 일반화된 테스트 케이스를 제작할 수 없으며 자동화 도구를 제작하고 그것을 테스트에 적용하는데 많은 어려움이 따른다.

둘째로, 결함의 위치를 파악하기가 어렵다. 차

량용 소프트웨어는 하드웨어, 미들웨어, 응용 소프트웨어 등 여러 계층의 컴포넌트가 밀접하게 관련되어 있다. 그래서 테스트 시 결함의 존재는 확인 가능하지만 결함의 원인은 어플리케이션뿐만 아니라 운영체제나 하드웨어일 수도 있다. 그래서 블랙박스 테스트를 수행하더라도 결함 발생 시 소스코드를 확인하는 화이트박스 테스트가 되기도 한다.

셋째로, 테스트 결과를 확인하기가 어렵다. 일반적인 PC용 소프트웨어의 경우 테스트 결과 확인을 위해 자체의 GUI를 이용하거나 테스트용 화면을 이용한다. 하지만 차량용 소프트웨어는 동작의 모니터링을 위한 별도의 소프트웨어나 장비가 없는 경우가 많아 테스트 결과를 확인하는데 어려움이 있다.

넷째로 차량용 소프트웨어 테스터는 하드웨어적 지식을 충분히 가지고 있어야 한다. 차량용 소프트웨어는 앞서 말한 것처럼 하드웨어와 밀접한 관계를 맺으며 동작한다. 테스터가 하드웨어에 대한 지식이 충분하지 않으면 오류 발생 시 오류가 발생한 위치가 어디인지 오류의 원인이 무엇

인지 판단하기 어려울 수 있다.

마지막으로 차량용 소프트웨어 테스트는 기능적 테스트뿐만 아니라 안전성 및 신뢰성 테스트에도 큰 비중을 두어야 한다. 일반적인 PC용 소프트웨어와는 달리 차량용 소프트웨어에 의한 자동차의 오작동은 사람의 생명과 직결되는 문제이기 때문에 안전성 및 신뢰성 테스트의 중요성은 더욱 강조된다.

4. AUTOSAR 테스트 동향

3절과 같은 차량용 소프트웨어 테스트의 어려움을 감소시키고자 AUTOSAR 단체에서는 AUTOSAR 테스트를 표준화한 적합성 테스트(Conformance Test) 표준을 만들었다. 그리고 여러 학계 및 기업에서는 다양한 테스트 기법 및 테스트 도구를 개발하고 있다. 이번 절에서는 AUTOSAR 적합성 테스트 표준, 테스트 기법 및 테스트 도구에 대하여 알아보도록 한다.

4.1 AUTOSAR 테스트 대상별 구분

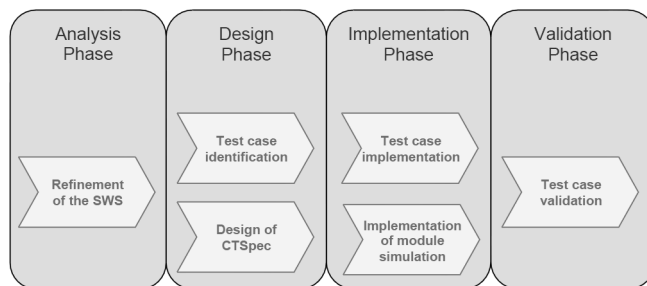
4.1.1 AUTOSAR 적합성 테스트

AUTOSAR 적합성 테스트는 AUTOSAR 표준을 기반으로 제작된 BSW가 AUTOSAR 표준을 준수하고 있는지를 확인하기 위한 테스트이다.

AUTOSAR 적합성 테스트는 BSW의 기능적인 요소만을 대상으로 하며 비기능적인 요소는 배제한다. 실제 테스트에서 소스 코드를 기반으로 한 화이트박스는 수행하지 않고 입출력 파라미터를 체크, 분석하는 블랙박스 테스트 기법만을 수행한다. AUTOSAR 적합성 테스트의 테스트 케이스는 파라미터 개수는 맞는지, 파라미터 타입은 정확한지, 입력 파라미터에 대한 출력 파라미터가 올바른 값을 반환하는 지를 확인한다.

AUTOSAR 적합성 테스트 프로세스는 (그림 2)와 같다. 테스트 프로세스는 테스트 분석 단계(Analysis Phase), 테스트 디자인 단계(Design Phase), 테스트 구현 단계(Implementation Phase), 테스트 검증 단계(Validation Phase)의 총 4단계로 이루어진다.

첫 번째 테스트 분석 단계(Analysis Phase)에서 테스트 디자이너(Test Designer)는 AUTOSAR 표준문서, AUTOSAR SWS(SoftWare Specification) 문서, 구현에 사용되었던 라이브러리 파일, 모델링 파일 등의 문서들을 기반으로 테스트에 필요한 요구조건을 담은 문서를 작성한다. 요구조건 문서 작성 시 비기능적 요구조건을 제외한 기능적 요구조건을 대상으로 하고 테스트 케이스 제작 작업을 용이하게 하기 위해 테스트 카테고리를 구분하여 테스트 아이템 항목을 도출한다. 두 번째 테스트 디자인 단계(Design Phase)에서 테



(그림 2) AUTOSAR 적합성 테스트 프로세스

스트 디자이너는 테스트 케이스 식별 도구를 이용해 분석 단계에서 도출한 테스트 아이템 항목을 가지고 테스트 케이스를 작성한다. 그리고 테스트 디자이너는 테스트 케이스 실행 환경인 테스트 아키텍처를 구성하고 테스트 대상 모듈에 대한 인터페이스 정의와 파라미터 정의를 정리한다. 세 번째 테스트 구현 단계(Implementation Phase)에서는 테스트 스크립트 언어인 TTCN-3으로 테스트 케이스를 구현하고 BSW 모듈의 시뮬레이터를 구현한다. 테스트 구현자(Test Implementer)는 TTCN-3 API 모듈과 TTCN-3 환경설정 모듈, 적합성 테스트 스펙에 기반한 UML 모델 등을 가지고 TTCN-3 테스트 케이스를 구현한다. 그리고 테스트 검증 구현자(Test Validation Implementer)는 SWS 문서를 기반으로 시뮬레이터에 대한 코드를 구현한다. 마지막인 검증 단계(Validation Phase)는 구현단계에서 만든 TTCN-3 테스트 케이스를 가지고 테스트를 수행한다. 테스트 평가자(Test Assessor)는 시뮬레이터를 가지고 TTCN-3 테스트 케이스를 수행하고, 시뮬레이션 테스트가 정상적으로 완료되면 차량전장용 하드웨어를 가지고 테스트를 수행한다. 테스트 후에는 테스트 케이스 수행 결과를 정리하여 테스트 보고서를 작성한다.

4.1.2 AUTOSAR SWC 테스트

테스트 대상에는 SWC를 하드웨어 및 ECU에 독립적으로 동작시키기 위한 BSW 뿐만 아니라 SWC 자체도 대상이 된다. AUTOSAR는 SWC에 대해서는 BSW처럼 테스트를 위한 표준은 만들지 않았다. 하지만 SWC에 대한 테스트도 자동차의 무결성 및 안전성을 위해서는 필요하기에 학계 및 기업에서는 SWC를 테스트하기 위한 여러 방법들을 제안하고 있다.

먼저 SWC를 테스트하기 위한 프로세스를 수립한 연구가 있다. SWC 기능의 최소 단위인 런어블(Runnable)의 기본기능과 소스코드를 검증하고 ECU에 맵핑되는 단위인 Atomic의 테스트를 수행한다. 그리고 SWC의 통신을 담당하는 RTE를 검증하고 SWC의 인터페이스를 테스트한다. 이렇게 개별 SWC와 RTE에 대한 충분한 검증이 이루어지면 Composition SWC와 Sensor/Actuator SWC 등 모든 컴포넌트를 통합하여 통합 테스트 및 기능 테스트를 수행한다. 마지막으로 명세서 기반의 시스템 레벨 테스트를 수행함으로써 하드웨어를 배제한 PC 기반의 소프트웨어 테스트를 마친다. 위와 같은 과정을 통해 소프트웨어가 제대로 구현되었다고 판정되면 프로토타입 단계에서는 실험용 하드웨어에 소스코드를 삽입하여 점차적으로 실제 하드웨어로 전환해 가며 최종 제품으로 개발해 간다^[7].

Vector 사에서는 SWC 테스트 시 테스트 대상인 SWC 외에 RTE, BSW, 실 차량 같은 부가 요소를 준비해야 하는 번거로움을 줄이기 위해 PC 기반의 에뮬레이터를 이용하는 방법을 제안한다. 테스트 수행 시 소스코드 수준의 테스트 케이스를 제작한 후 PC 기반의 에뮬레이터 상에서 테스트 케이스를 배치 형태로 입력하면 각각 테스트 케이스의 성공 여부가 확인된다. 성공 여부 확인은 테스트 케이스 각각에서 대상 API 호출 후 SWC 내의 상태 값이 예상 값과 같은지 비교하며 확인한다. SWC는 단독 컴포넌트 혹은 여러 컴포넌트의 조합 모두 적용 가능하며 가능한 테스트 항목은 1)Runnable Entity의 실행순서, 2)입력 값에 대한 출력 값의 정상여부, 3)Timeout, Queue Overflow 같은 예외상황 적용 후 SWC의 견고성 체크, 4)SWC의 RTE API 호출가능 여부가 있다^[8].

4.2 AUTOSAR 테스트 관련 표준화 동향

4.2.1 ISO 26262

ISO 26262는 IEC 61508을 기반으로 자동차에 탑재되어 있는 SW오류로 인한 사고를 미연에 방지하기 위해 제정한 기능 안전 규격이다. 원전, 항공, 의료, 철도, 장치산업 같은 안전 결정적(Safety-Critical) 시스템을 위한 안전 수준을 평가하는 국제표준으로는 IEC 61508이 가장 널리 활용되고 있다. 하지만 IEC 61508은 자동차를 사용하는 ‘소비자 관점의 안전’이 아니라, 안전이 보장된 제품을 제공해야 하는 ‘공급자 중심의 제품 안전’에 초점을 맞추고 있다. 이에 국제 자동차 기업들은 범용 전기전자장치의 안전성에 대한 표준인 IEC 61508을 기반으로 하여 SPICE(ISO/IEC 15504) 표준에 따라 도로를 주행하는 차량에 한정하는 ISO 26262 표준을 만들게 되었다.

ISO 26262는 IEC 61508의 핵심 개념인 안전 지속등급(SIL: Safety Integrity Level)과 하드웨어 중심의 안전생명주기(Safety Lifecycle)을 개선하여, 차량 안전지속등급(ASIL: Automotive Safety Integrity Level)과 시스템 중심의 안전생명주기를 도입하고 있다. ASIL은 자동차 제품의 특성을 반영한 3가지 핵심 요소인 1)위험 상황에 노출될 확률(Probability of Exposure in the Operational Severity), 2)위험의 잠재적 심각 수준(Potential Severity), 3)제어 가능성(Control-lability)에 따라 차량 안전지속 등급을 결정하도록 한다. 기존 IEC 61508에서 제시하는 SIL 결정 방식이 일반적인 재난분석과 위험심사를 통한 포괄적인 기준이라면, ISO 26262의 ASIL은 보다 현실적이고 구체적인 요소를 반영한 것이다. IEC 61508은 하드웨어의 안전 요구사항이 결정된 후 소프트웨어의 안전 요구사항을 추출하여 접근하는 전형적인

하드웨어 중심의 시스템 개발 생명주기형태를 가진다. 이에 반해 ISO 26262는 하드웨어와 소프트웨어 구성요소를 모두 고려한 시스템 수준의 개발 생명주기를 가지는 것이다.

자동차 업계에서는 ISO 26262가 향후 전통적인 형식승인을 대체하는 새로운 차량 안전 관리 규정으로 지정될 것으로 보고 있다. 차량용 전자장치의 기능이 복잡해지고 분산처리가 요구되면서 소프트웨어 기능도 복잡해졌고, 수십, 수백만 라인에 이르는 차량용 소프트웨어를 코드 수준에서 검증하는 것만으로는 자동차의 안전성과 신뢰성을 보장할 수 없기 때문이다.

4.2.2 Automotive SPICE

Automotive SPICE(이하 ASPICE)는 SPICE(Software Process Improvement and Capability Evaluation)를 기반으로 한 자동차 도메인의 프로세스 평가 모델이다. SPICE는 소프트웨어 프로세스 전반을 심사하여 소프트웨어 개발 프로세스를 개선하고 개발자의 능력을 향상시켜 소프트웨어에서 발생할 수 있는 위험을 사전에 예방하기 위한 목적으로 ISO와 IEC에서 추진하는 소프트웨어 품질 평가 모델이다. 자동차는 수많은 기능이 소프트웨어로 이루어져 있으며 각각의 기능이 어떤 프로세스로 관리되는가는 차량의 최종 품질을 좌우하는 중요한 요소가 되고 있다. 그래서 범용적인 도메인을 가지는 SPICE를 SPICE 사용자 그룹을 중심으로 한 자동차 SIG(Special Interest Group)에서 자동차 도메인 관점에서 재해석하여 ASPICE를 만들게 되었다.

ASPICE는 기존의 SPICE에서 21개의 프로세스를 제거하고 5개의 프로세스를 추가하며 자동차 도메인에 맞게 수정되었다. ASPICE는 자동차 메이커에 부품을 공급하는 공급자들의 능력을 심

사하고 그 결과를 공유하여 공급자의 프로세스 및 품질 능력을 높이는 것을 목표로 한다. 여러 메이커에 부품을 공급하는 공급자는 심사를 한번만 하면 되기 때문에 심사에 대한 부담을 줄일 수 있으며, 자동차 메이커는 합의된 심사 결과 및 형식을 공유할 수 있음으로써 공급자에 대한 능력 판단을 쉽게 할 수 있게 된다.

4.2.3 MISRA-C

MISRA-C는 MISRA(Motor Industry Software Reliability Association)에서 개발한 차량전장용 소프트웨어 코드 인스펙션을 위한 C 언어 표준이다. AUTOSAR에서는 자동차용 소프트웨어 개발 언어로써 C, C++ 및 Java를 명시하고 있지만 대부분의 경우 안전성 검증이 상대적으로 수월한 C 언어로 개발하는 것이 일반적이다. MISRA-C 문서는 차량전장용 소프트웨어를 개발하는 개발자들이 차량전장용 소프트웨어 개발 시 지켜야 할 코드 안전성, 이식성을 보장하는 내용들이 141개 가이드라인 형태로 정리되어 있다. MISRA-C의 141개의 가이드라인은 주로 차량전장용 소프트웨어의 컴파일 단계 이전에 소스코드를 검사하는 정적 코드 분석을 통해서 검사 가능한 규약으로 구성되었고, 개발자가 컴파일 이전에 개발도구를 이용하여 규약의 준수여부를 검사할 수 있다.

4.3 AUTOSAR 테스트 툴

이번 절에서는 AUTOSAR 테스트를 도와주는 대표적인 툴을 소개한다. 앞서 소개한 4.1절의 테스트 대상별 구분과 마찬가지로 테스트 툴도 BSW 관련 적합성 테스트 툴과 SWC 테스트 툴로 나뉘어 진다.

먼저 BSW 적합성을 검증하는 대표적인 테스트 툴로는 Volcano Vehicle Systems Tester

(VST)^[9]가 있다. VST는 Mentor사에서 만든 테스트 플랫폼인 Volcano VSx tool chain의 구성요소 중 하나로, AUTOSAR 적합성 테스트를 위한 도구이다. 테스트 디자인, 실행, 보고서 작성을 위한 단일 환경을 제공하며, 이클립스 플러그인 형태로 제공된다. PC와 타겟 머신 두 환경 모두에서 테스트 수행을 할 수 있고, TTCN-3 기반으로 동작하여 이식성과 호환성이 뛰어나다.

SWC를 테스트하는 대표적인 툴로는 DaVinci Component Tester^[8]가 있다. DaVinci Component Tester는 Vector사가 제작한 소프트웨어 컴포넌트(SWC) 검증도구이다. 실제 차량 하드웨어와 AUTOSAR의 BSW와 RTE 없이도 SWC를 검증 가능토록 하기위해 PC 상에서 BSW와 RTE를 에뮬레이션한다. 개별 소프트웨어 컴포넌트에 대한 단위 테스트부터 복잡한 기능에 대한 통합 테스트까지 PC 기반으로 확인할 수 있어 BSW를 개발하거나 하드웨어를 준비해야 하는 비용을 줄일 수 있다.

5. 결론

본 고에서는 차량전장용 소프트웨어 프레임워크인 AUTOSAR와 관련 테스트 동향을 알아보았다. AUTOSAR는 2005년 첫 번째 버전을 만든 이후 지속적인 활동을 하며 2009년 네 번째 버전을 배포하였고, FORD, TOYOTA, BMW 등 AUTOSAR의 9개 핵심 멤버들이 2012년까지 단계적으로 AUTOSAR 프레임워크를 적용하기로 공표하는 것처럼 차량전장용 소프트웨어의 대표 표준으로 자리매김을 하였다. 한국 정부도 2010년에 발표한 WBS(World Best Software) 프로젝트에 ‘AUTOSAR 기반 사용 설계도구 및 플랫폼 상용화 기술 개발’을 포함하며 차량전장화 기술 확보를 도모하고 있다.

이러한 AUTOSAR가 자동차에서 정해진 기능을 정확히 수행하고 안전하게 동작하기 위해 여러 단체와 기업 및 학계가 노력을 기울이고 있다. AUTOSAR 단체는 AUTOSAR 적합성 테스트 표준을 만들어 BSW 모듈 동작의 무결성을 보장하고 있다. 학계와 기업은 SWC가 AUTOSAR 프레임워크상에서 오류 없이 해당 기능을 동작할 수 있는지 테스트하는 기법을 제안하였다. 그리고 BSW와 SWC의 기능동작 검증뿐만 아니라, ISO 26262, ASPICE 등 기존의 개발방법론을 차량용으로 변형하여 차량용 소프트웨어의 안전성을 향상시키고 개발 프로세스를 견고하게 하려는 움직임이 있었다. AUTOSAR 테스트 과정에서 BSW 적합성 검증 도구와 SWC 테스트 도구를 사용해 테스트에 드는 노력을 최소화하는 노력도 진행 중이다.

위와 같은 테스트 활동을 통해 기능성과 안전성을 검증 받은 AUTOSAR는 우리의 자동차 생활의 많은 부분을 변화시킬 것이다. AUTOSAR를 기반으로 미래의 자동차는 운전자의 운전 패턴과 주행 상황을 분석하여 최적의 연비를 낼 수 있도록 차량의 세팅을 조정할 수 있고, 승객의 위치와 나이를 구분해 자동차의 에어백 같은 안전 시스템을 작동하거나 혹은 적은 힘으로 작동하도록 지원할 것이다. 초음파나 적외선 레이저를 통해 차간 거리를 유지하며 적합한 속도로 가속하는 적응성 크루즈 컨트롤(Adaptive Cruise Control) 시스템이나, 공간을 인지하면서 핸들을 자동 조절하는 자동 주차시스템(Automatic Parking System)도 AUTOSAR의 지원 아래 가능한 기술들이다. 자동차 생활을 더욱 편리하고 안전하게 만들어줄 AUTOSAR가 체계적이고 효율적인 테스트를 거쳐 보다 빨리 우리의 자동차에 탑재되기를 기대해본다.

참고 문헌

- [1] 박미룡, 김재영, "자동차 전장 SW 플랫폼 규격 (AUTOSAR) 표준화 동향", TTA Journal, 2008. 05.
- [2] 홍성수, 박지용, 유지석, "OSEK과 AUTOSAR를 중심으로 본 차량용 OS와 미들웨어 기술 동향", Journal of the Korean Society of Precision Engineering, Vol.21, No.00, 2006. 09.
- [3] http://www.cbsnews.com/8301-501465_162-20021427-501465.html
- [4] 유종훈, 박지용, 홍성수, "AUTOSAR 소프트웨어의 컴포넌트 모델의 분석", Control, Automation, and Systems Symposium, 2007. 05.
- [5] OSEK, "OSEK/VDX Operating System Version 2.2.3", 2005.
- [6] AUTOSAR, "Specification of Operating System Version 3.0.2", 2008.
- [7] 박광민, 금대현, 이성훈, 원웅재, 정우영, "Test methods of the AUTOSAR application software components", ICROS-SICE International Joint Conference, 2009. 08.
- [8] http://www.vector.com/vi_davinci_component_tester_en.html
- [9] <http://www.mentor.com/products/vnd/autosar-products/volcano-system-tester/>

저자 약력



김 광 훈

이메일 : nuly17@tta.or.kr

- 2006년 인하대학교 컴퓨터공학과(학사)
- 2008년 한국과학기술원 전산학과(석사)
- 2008년~현재 한국정보통신기술협회 선임연구원
- 관심분야: 소프트웨어 테스트, 컴퓨터 가상화, 서비스 디스커버리