

# AUTOSAR

## 소프트웨어 컴포넌트의 유닛 테스트 방법

### (Unit Testing for the AUTOSAR Software Component)

금대현<sup>\*</sup>      이성훈<sup>\*</sup>  
(Daehyun Kum)      (Seonghun Lee)

박광민<sup>\*\*</sup>      손병점<sup>\*\*</sup>  
(Gwangmin Park)      (Byeongjeom Son)

**요약** 최근 세계 자동차 업계는 AUTOSAR 소프트웨어 플랫폼 표준을 제정하여 소프트웨어의 신뢰성과 재사용성을 향상시키고 있다. 그러나 테스트 시스템 및 테스트 프로세스의 신뢰성 확보와 테스트에 소요되는 기간 단축을 위하여 테스트의 표준화 및 자동화가 절실하다. 특히 RTE 및 베이직 소프트웨어 모듈은 표준화를 만족하는 설계 도구를 사용함으로써 최소한의 신뢰성이 보장되지만, 여전히 응용 소프트웨어 컴포넌트에 대해서는 철저한 테스트가 필요하다. 본 논문에서는 표준화된 AUTOSAR 설계 정보를 이용하여 소프트웨어 컴포넌트의 유닛 테스트를 위한 테스트 시스템의 생성 및 테스트 케이스 생성 자동화 방법을 제안하였으며, 테스트 시스템은 TTCN-3 테스트 표준을 적용하였다. 본 논문에서 제안한 방법을 적용함으로써 차량 소프트웨어의 개발 기간 단축 및 신뢰성을 향상시킬 수 있었다.

키워드 : AUTOSAR, TTCN-3, 소프트웨어 컴포넌트, 테스트 자동화, 테스트 케이스

\* 본 연구는 교육과학기술부에서 지원하는 대구경북과학기술원의 기관고유 사업비로 수행하였음

\*\* 이 논문은 2010 한국컴퓨터종합학술대회에서 'AUTOSAR 소프트웨어 컴포넌트의 유닛 테스트 방법'의 제목으로 발표된 논문을 확장한 것임

<sup>\*</sup> 정 회 원 : 대구경북과학기술원 미래산업융합기술연구부 선임연구원  
kumdh@dgist.ac.kr  
shunlee@dgist.ac.kr  
(Corresponding author)

<sup>\*\*</sup> 정 회 원 : 대구경북과학기술원 미래산업융합기술연구부 연구원  
ggangmin@dgist.ac.kr  
easypass@dgist.ac.kr

논문접수 : 2010년 8월 18일  
심사완료 : 2010년 9월 16일

Copyright©2010 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제11호(2010.11)

**Abstract** AUTOSAR, a standard software platform for automotive, has been developed to manage software complexity and improve software reuseability. However Automated and standardized testing is needed to improve reliability and reduce time and effort on testing. Fundamental functionality of AUTOSAR RTE and basic software modules is guaranteed by using an AUTOSAR tool, but application software components have to be tested thoroughly. In this paper, we suggest a test system for the AUTOSAR software component using TTCN-3, a standardized testing language. Test execution system and test cases for the software component are generated automatically from AUTOSAR XML containing software design information. With the proposed testing techniques we can reduce time and effort to build the testing system.

Key words : AUTOSAR, TTCN-3, Software Component, Automated Testing, Test Case

## 1. 서론

세계 자동차 업계는 복잡한 소프트웨어의 재사용성과 신뢰성 향상을 위하여 AUTOSAR 컨소시엄을 구성하여 차량 소프트웨어 플랫폼과 개발 방법을 표준화하였다. 또한, 최근 소프트웨어가 복잡해져서 테스트 실행 환경과 테스트 케이스 설계에 많은 시간과 노력이 필요하며, 테스트 단계에서 오류가 발생할 확률도 높아졌다. 소프트웨어 플랫폼의 표준화만으로는 신뢰성 향상에 한계가 있으며, 테스트 신뢰성 향상과 테스트 기간 단축을 위해서는 표준 플랫폼을 활용한 테스트의 표준화와 자동화가 절실히 필요한 시점이다.

AUTOSAR 소프트웨어의 구조는 응용 소프트웨어 컴포넌트, RTE, 베이직 소프트웨어로 이루어져 있으며, RTE와 베이직 소프트웨어의 신뢰성은 표준을 만족하는 설계 도구를 사용함으로써 최소한의 신뢰성은 확보할 수 있다. 하지만, 응용 소프트웨어 컴포넌트의 구현 코드가 요구 사항과 소프트웨어 설계를 만족하는지에 대한 철저한 테스트가 필요하다.

본 논문에서는 TTCN-3 테스트 표준을 적용한 테스트 시스템의 구조를 설계하였으며, 표준화된 소프트웨어 설계 정보로부터 테스트 환경 및 테스트 케이스를 생성하는 방법을 제안하고자 한다. 소프트웨어 모델의 설계와 동시에 테스트 환경과 테스트 케이스가 자동으로 생성되어서 즉시 테스트가 가능하고, 테스트 표준을 적용함으로써 테스트 환경 및 테스트 데이터의 재사용성을 향상시키고자 하였다.

특히, 소프트웨어 컴포넌트의 유닛 테스트를 위한 실행환경의 자동 생성 방법과 각 소프트웨어 컴포넌트의 입출력 인터페이스 검증 및 기능 검증을 위한 테스트

케이스 생성을 자동화하였다.

본 논문에서 제안한 유닛 소프트웨어 컴포넌트의 테스트 방법을 적용함으로써 하드웨어 및 베이직 소프트웨어의 설계가 완료되지 않은 개발 초기 단계에서 쉽게 소프트웨어 컴포넌트를 검증할 수 있는 환경을 제공할 수 있다.

## 2. 관련 연구

최근 차량 소프트웨어는 모델 기반 방법을 적용하여 테스트를 자동화하고 있다. [1]과 [2]에서는 소프트웨어의 기능 모델로부터 테스트 케이스를 생성하여 테스트 수행을 자동화하는 방법과 프로세스를 제안하고 있다. 모델 기반 테스트는 주로 모델로부터 테스트 케이스 생성 방법에 관한 연구이며, 테스트 시스템의 생성은 고려하지 않고 있다. 그리고 설계 도구에서 제공하는 시뮬레이션 환경을 이용하거나 설계 도구에 종속된 테스트 도구를 사용함으로써 테스트 환경과 테스트 케이스의 재사용에 한계가 있었다. 본 논문에서는 표준화된 AUTOSAR XML 설계 명세를 이용하여 설계 도구에 독립적인 테스트 시스템과 시뮬레이션 환경을 생성하는 기술을 제안함으로써, 테스트 환경 및 테스트 데이터의 재사용성을 높이고자 하였다.

TTCN-3은 테스트 대상 시스템의 구현 언어 및 환경에 독립적인 테스트 케이스와 테스트 시스템 설계를 위한 표준이며[3], 최근에는 AUTOSAR의 적합성 테스트 표준으로 채택되었다. [4]는 AUTOSAR 설계도구의 적합성 테스트 시스템을 개발하였으며, [5]는 AUTOSAR 모델을 단순히 TTCN-3 코드로 변환하는 기술을 제안하고 있다. 기존의 연구는 TTCN-3 테스트 시스템의 구조와 타겟 시스템의 생성방법에 대한 연구는 언급하지 않았다. 본 논문에서는 TTCN-3 테스트 시스템의 구조를 제안하고 있으며, 또한 소프트웨어 컴포넌트의 시뮬레이션 환경과 테스트 데이터의 생성 방법을 제안하고 있다.

차량 소프트웨어의 기능 검증을 위한 테스트 케이스의 생성을 위하여 분류 트리 기법(classification tree method), 등가 분할(equivalence class partitioning), 경계 값 분석(boundary value analysis), 결정 테이블 테스트(decision table testing), 페어와이즈 조합 테스트(pairwise testing) 등을 사용하고 있다. [6-8]은 임베디드 소프트웨어를 위한 테스트 케이스 생성 방법과 차량 소프트웨어에 적용한 연구 사례를 보여준다. 본 연구에서도 소프트웨어 컴포넌트의 기능 검증을 위한 테스트 케이스 생성을 위하여 등가 분할, 경계값 분석, 페어와이즈 조합 테스트(pairwise testing) 등을 적용하였다.

## 3. 테스트 시스템

### 3.1 소프트웨어 컴포넌트의 유닛 테스트

AUTOSAR는 응용 소프트웨어의 재사용성 향상을 위하여 소프트웨어 컴포넌트 단위로 설계하고 검증한다. AUTOSAR 소프트웨어 컴포넌트는 기능 수행을 위하여 하나 이상의 런어블로 구성되며, 포트, 인터페이스, 내부 런어블 변수(Inter Runnable Variable)를 사용하여 데이터를 주고받는다. 그리고 소프트웨어 컴포넌트의 설계 정보는 표준화된 XML 형태로 저장되며, 그림1은 소프트웨어 컴포넌트의 설계와 명세 정보의 일부를 보여준다.

일반적으로 유닛 테스트는 소프트웨어의 구조적 결함과 기능적 결함을 발견하기 위한 가장 기본적이고 중요한 테스트이다. AUTOSAR 시스템도 결함 발견을 위하여 소프트웨어 컴포넌트는 가장 철저한 유닛 테스트가 요구된다. 특히 소프트웨어 컴포넌트를 제외한 RTE, 베이직 소프트웨어 모듈의 코드는 설계도구로부터 자동 생성되므로, 호환성 검증된 표준 설계 도구를 사용함으로써 최소한의 신뢰성은 보장되고, 소프트웨어 컴포넌트의 통합도 RTE에 의해서 신뢰성이 보장된다. 하지만, 각 소프트웨어 컴포넌트의 입출력 인터페이스와 기능에 대한 신뢰성은 유닛 테스트를 통해서 검증해야 한다.

소프트웨어 컴포넌트의 유닛 테스트는 AUTOSAR 인터페이스 테스트, 소프트웨어 컴포넌트의 기능 검증을 위한 블랙박스 테스트, 내부 소스 코드의 구조에 대한 화이트 박스 테스트의 3가지로 구분할 수 있으며, 본 논문에서는 소프트웨어 컴포넌트의 XML 설계 정보를 이용하여 테스트 자동화할 수 있는 AUTOSAR 인터페이스 테스트와 블랙박스 테스트에 대한 방법을 연구하였다.

소프트웨어 컴포넌트는 RTE를 통해서 다른 소프트웨어 및 베이직 소프트웨어와의 통신하며, 이때 RTE API를 호출한다. 소프트웨어 컴포넌트가 호출하는 RTE API에 대한 정보는 AUTOSAR XML에 모두 저장되어 있으며, 이 정보를 이용하여 RTE API가 정확히 사용되고 있는지를 테스트할 수 있다. 이처럼 소프트웨어 컴포넌트 내부의 RTE API 호출에 대한 테스트를 AUTOSAR 인터페이스 테스트라고 정의하였다.

소프트웨어 컴포넌트의 기능이 요구 사항을 만족하는지를 검증하기 위하여 블랙박스 테스트를 한다. AUTOSAR XML로부터 테스트에 필요한 입출력 정보를 추출한 후에 경계 값 분석, 등가 분할, 페어와이즈 테스트 케이스 생성 기법 등을 복합 적용하여 테스트 케이스를 생성한다.

### 3.2 테스트 시스템 구조

본 테스트 시스템은 개발 초기 단계에 소프트웨어 컴

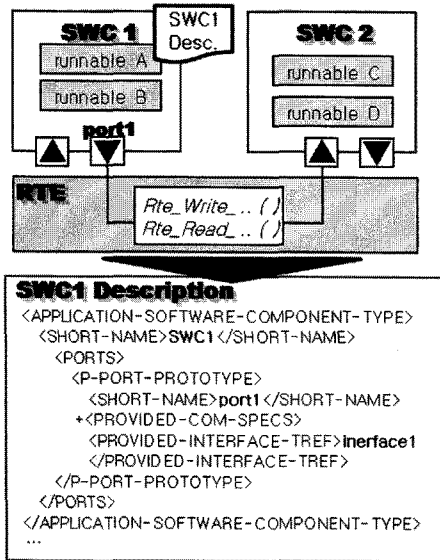


그림 1 소프트웨어 컴포넌트 XML 설계 명세

포넌트의 유닛 테스트를 위한 테스트 시스템과 테스트 케이스 생성을 자동화하는 데 목적이 있다. 테스트 시스템은 TTCN-3 표준을 적용함으로써 신뢰성과 재사용성을 향상 시키고자 한다. 테스트 시스템은 소프트웨어 컴포넌트의 표준화된 XML 명세 정보를 활용하여 자동으로 생성함으로써 설계 변경 또는 오류 수정으로 인하여 반복적으로 수행해야 하는 테스트에 소요되는 노력과 시간을 최소화할 수 있다.

그림 2와 같이 AUTOSAR XML 설계 정보로부터 즉시 실행 가능한 테스트 시스템과 소프트웨어 컴포넌트 시뮬레이션 환경을 생성한다. 테스트 케이스 생성기는 AUTOSAR 인터페이스 테스트를 위한 테스트 케이스와 기능 검증을 위한 블랙박스 테스트 케이스를 생성하며, TTCN-3에서 실행 가능한 테스트 데이터 형식으로 생성한다.

테스트 시스템은 TTCN-3 테스터와 타겟 시스템으로 나누어지며, 두 시스템은 UDP/IP 네트워크 통신을 이용하여 테스트 입력과 실행 결과를 서로 주고받는다. TTCN-3 테스터의 기본 모듈은 테스트 컴포넌트, 코덱 및 통신 인터페이스로 이루어진다. 테스트 컴포넌트는 테스트 데이터에서 기술된 테스트 입력을 타겟 시스템으로 보내고 소프트웨어 컴포넌트에서 실행된 결과를 수신하여 예상 결과 값과 비교하여 합격 여부를 판단하는 핵심 모듈이며, TTCN-3의 함수, Altstep, 테스트케이스를 사용하여 구현한다. 코덱 및 통신 인터페이스는 타겟 시스템과 테스트 데이터 교환을 위한 모듈이며 TTCN-3 TRI, TCI를 사용하여 구현한다.

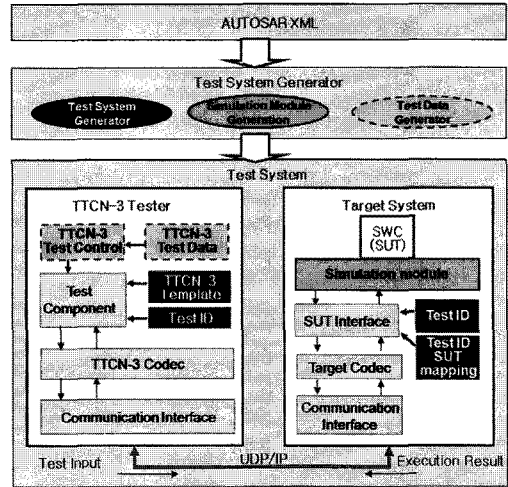


그림 2 테스트 시스템 구조

테스트 ID 및 TTCN-3 템플릿 모듈은 소프트웨어 컴포넌트의 입출력 관련 XML 정보로부터 생성하고, 테스트 데이터와 테스트 컨트롤 모듈은 테스트 데이터 생성 시스템으로부터 생성된다.

타겟 시스템은 소프트웨어 컴포넌트의 실행을 위한 시뮬레이션 환경과 테스트 입력을 수신하고 결과 값을 송신할 수 있는 인터페이스를 제공한다. 타겟 코덱과 통신 인터페이스는 테스터와 UDP/IP 통신으로 테스트 입력과 실행 결과를 주고받는다. 그리고 테스트 데이터의 송수신을 위해서 TTCN-3 테스터와 동일한 테스트 ID 모듈을 생성하고, SUT 맵핑 모듈은 각 테스트 ID를 소프트웨어 컴포넌트의 입출력 포트에 연결시킨다. 시뮬레이션 모듈은 소프트웨어 컴포넌트의 실행에 필요한 RTE 및 OS 환경을 제공한다.

## 4. 테스트 방법

### 4.1 AUTOSAR 인터페이스 테스트

소프트웨어 컴포넌트는 AUTOSAR 인터페이스만을 통해서 외부와 통신하므로, 소프트웨어 컴포넌트가 설계 명세에 충실하게 구현되었는지 검증하는 AUTOSAR 인터페이스 테스트는 소프트웨어 컴포넌트 테스트에서 가장 먼저 해야 할 테스트이다.

AUTOSAR XML을 이용하면 AUTOSAR 인터페이스 테스트를 위한 테스트 시스템과 테스트 케이스 생성을 모두 자동화할 수 있다. 각 언어블에 해당하는 테스트 ID와 테스트 케이스를 생성하고, AUTOSAR 인터페이스를 검증할 수 있는 시뮬레이션 모듈을 생성한다.

주요 RTE API는 소프트웨어 컴포넌트 사이의 Sender-Receiver 통신 방식에 사용되는 Rte\_Write, Rte\_Read,

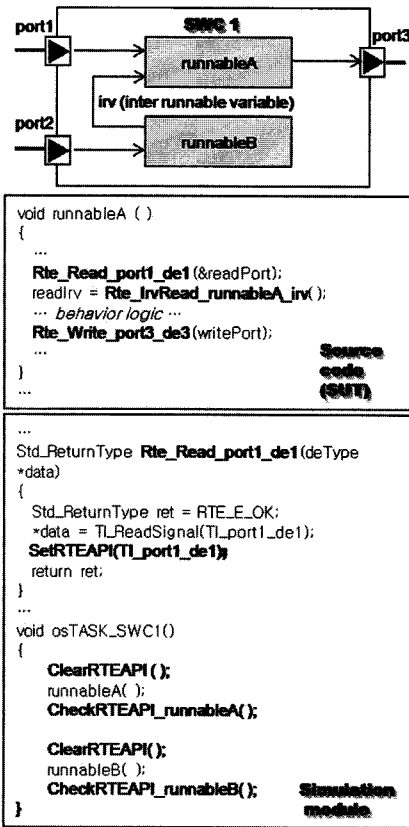


그림 3 AUTOSAR 인터페이스 테스트 방법

```

/* TestID for AUTOSAR Interface Test */
const integer sigRunnInterfaceTest := 205;
const integer sigRunnInterfaceTestResult := 206;

/* Runnable ID */
const integer RID_runnableA := 1;
const integer RID_runnableB := 2;

/* TTCN-3 Template for AUTOSAR Interface Test */
template SIGNAL setRunnInterfaceTest(integer runnID) := {
  signalID := sigRunnInterfaceTest
  signalLength := 1,
  signalValue := runnableID,
}

template SIGNAL getRunnableIOTest() := {
  signalID := sigRunnInterfaceTestResult,
  signalLength := ?,
  signalValue := TESTPASS,
}

/* Test Data for AUTOSAR Interface Test */
/* runnableA */
const TI TL1 := { setRunnInterfaceTest (RID_runnableA) };
const ER ER_1 := { getRunnableIOTest( ) };
/* runnableB */
const TI TL2 := { setRunnInterfaceTest (RID_runnableB) };
const ER ER_2 := { getRunnableIOTest( ) };

/* TTCN-3 Test Control */
execute( TC (TL1, ER_1) );
execute( TC (TL1, ER_2) );

```

그림 4 TTCN-3 테스트 데이터 및 테스트 컨트롤 모듈

Client-Server 통신 방식에 사용되는 Rte\_Call, 런어블 사이의 통신에 사용된 Rte\_IrvRead, Rte\_IrvWrite, Exclusive Area에 사용되는 Rte\_Enter, Rte\_Exit 등이 있다.

소프트웨어 컴포넌트는 런어블로 구성되며, RTE API의 호출은 런어블 내부에서 이루어진다. 그러므로 AUTOSAR 인터페이스 테스트도 런어블 단위로 수행하였으며, 런어블의 실행 중 호출된 RTE API를 분석하여 검증하였다. 모든 검증은 시뮬레이션 모듈에서 이루어지며 테스트를 위해서 소프트웨어 컴포넌트 내부에 코드 삽입은 없다.

테스트 방법은 다음과 같다. 먼저 시뮬레이션 모듈의 태스크가 소프트웨어 컴포넌트 내부의 런어블을 차례로 실행한다. 런어블의 실행 전에 함수에서 RTE API의 호출 여부를 판단할 수 있는 플래그를 해제하고(Clear\_RTEAPI()), 런어블 실행 중 RTE API가 호출될 때 해당 플래그를 설정한다(Set\_RTEAPI()). 런어블의 실행이 완료되고 나서 호출된 API를 확인하여 모든 호출이 정확히 실행되었는지를 검증한다(Check\_RTEAPI()). 시뮬레이션 모듈은 AUTOSAR 인터페이스를 테스트할 수 있는 환경을 고려하여 AUTOSAR XML로부터 생성한다.

소프트웨어 컴포넌트에 포함된 각 런어블마다 테스트 ID와 테스트케이스를 생성한다. 런어블에 해당하는 테스트 ID를 타겟 시스템으로 송신하면 타겟 시스템은 해당 런어블의 실행 중 호출된 RTE API를 확인하여 합격 여부를 TTCN-3 테스트 측으로 회신하고, 모든 RTE API가 정상적으로 호출되었으며 합격으로 처리한다. 그림 4는 AUTOSAR 인터페이스 테스트를 위한 테스트 ID, TTCN-3 템플릿, 테스트 데이터, 테스트 컨트롤 모듈의 예를 보여준다.

#### 4.2 블랙박스 테스트

소프트웨어 컴포넌트의 외부 인터페이스에 대한 검증을 완료 후 소프트웨어 컴포넌트의 기능 검증을 위하여 블랙박스 테스트를 하며, 요구 사항 검증을 위한 다양한 테스트 케이스 설계 기법을 적용해서 설계한다.

블랙박스 테스트를 위한 테스트 케이스 생성 기법은 우선 AUTOSAR XML로부터 소프트웨어 컴포넌트의 임출력 신호의 포트, 데이터 엘리먼트, 데이터 타입, 데이터 타입의 등가 클래스 등에 대한 정보를 추출한 후 경계 값 분석 및 블랙박스 테스트 케이스를 생성한다. 경계 값 분석은 각 등가 클래스의 경계 값을 테스트하며, 테스트 입력 및 등가 클래스의 조합을 이용하여 블랙박스 테스트 케이스를 생성한다. 이때 효과적인 테스트 케이스 생성을 위하여 페어와이즈 테스트 알고리즘을 적용하였다. 그림 5는 블랙박스 테스트 케이스 생성 방법을 나타

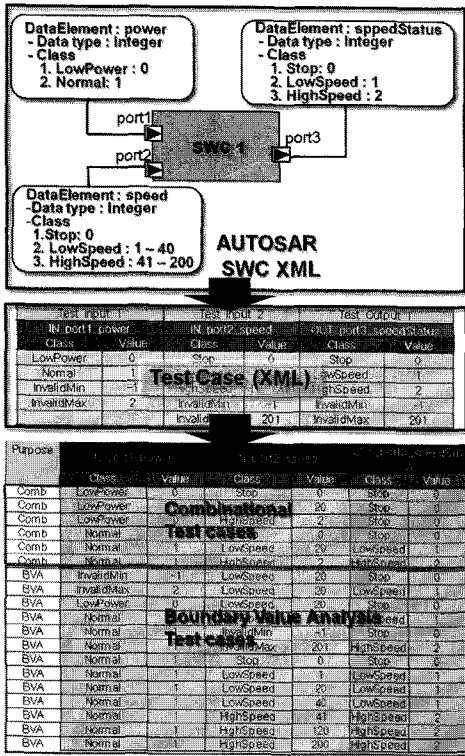


그림 5 블랙박스 테스트 케이스 생성 방법

내고 있다. AUTOSAR XML로부터 테스트 케이스 생성에 필요한 정보를 추출하여 테스트 케이스 관리를 위한 XML로 저장한다. 테스트 케이스 XML을 이용하여 다양한 테스트 케이스를 생성하고 관리하게 된다.

5. 결론 및 고찰

AUTOSAR 표준 플랫폼을 적용하면 RTE와 베이스 소프트웨어 모듈, 소프트웨어 통합에 대한 최소한의 신뢰성 보장되지만, 소프트웨어 컴포넌트는 여전히 철저한 유닛 테스트가 필요하다.

본 논문에서는 AUTOSAR XML 소프트웨어 설계 정보를 이용하여 소프트웨어 컴포넌트의 유닛 테스트를 위한 테스트 시스템 및 시뮬레이션 환경을 자동으로 생성하였으며, AUTOSAR 인터페이스 테스트와 블랙박스 테스트를 위한 TTCN-3 테스트 케이스 생성 방법을 제안하였다.

제안한 방법을 차량 바디 시스템에 적용한 결과 소프트웨어의 설계 정보만 이용하여 테스트 실행 환경과 테스트 케이스를 생성함으로써 테스트 시간 및 노력을 최소화할 수 있었다. 특히 개발 초기 단계에 하드웨어 및 베이스 소프트웨어의 설계가 완료되기 전에 쉽고 빠르

게 테스트를 수행할 수 있는 환경을 제공할 수 있었으며, 응용 소프트웨어 컴포넌트의 신뢰성 향상에 많은 도움이 되었다. 그리고 TTCN-3 테스트 표준을 적용함으로써 재사용성 향상을 향상 시키고자 하였다.

참고 문헌

[1] D. Kum, J. Son, J. Son and M. Kim, "Automotive Embedded System Software Development and Validation with AUTOSAR and Model-based Approach," *Journal of Control, Automation, and System Engineering*, vol.13, no.12, pp.1179-1185, 2007. (in korean)

[2] R. Baillargeon and R. Flores, "Model Driven Testing," *SAE World Congress*, 2008.

[3] J. Grabowski, D. Hogrefe, G. Réthy, I. Schieferdecker, A. Wiles and C. Willcock, "An Introduction to the testing and test control notation (TTCN-3)," *Computer Networks* 42, pp.375-403, 2003.

[4] A. Gilberg, "AUTOSAR Conformance Testing using TTCN-3," *TTCN-3 User Conference 2009*, 2009.

[5] J. Großmann, I. Schieferdecker, "Mapping AUTOSAR Interfaces to TTCN-3," *TTCN-3 User Conference 2009*, 2009.

[6] M. Conard, and A. Krupp, "An Extension of the Classification-Tree Method for Embedded Systems for the Description of Events," *Electronic Notes in Theoretical Computer Science*, vol.164, no.4, pp. 3-11, 2006.

[7] K. C. Tai and Y. Lie, "A Test Generation Strategy for Pairwise Testing," *IEEE Transactions on Software Engineering*, vol.28, no.1, pp.109-111, 2002.

[8] A Mjeda, P. McElligott, K. Ryan and S. Thiel, "Model-Based Testing Design for Embedded Automotive Software," *SAE World Congress*, 2009.