TTCN-3을 이용한 차량 소프트웨어 컴포넌트의 테스팅 자동화 방법

(Automated Testing Techniques for Automotive Software Components with TTCN-3)

금 대 현 † 이 성 훈 †
(Daehyun Kum) (Seonghun Lee)
박 광 민 † 조 정 훈 ††
(Gwangmin Park) (Jeonghun Cho)

요 약 최근 차량 소프트웨어의 신뢰성 및 재사용성 향상을 위하여 AUTOSAR 표준을 제정하였다. 그러나 소프트웨어의 표준화만으로는 높은 수준의 신뢰성 확보와 개발 기간 단축에 한계가 있으며, 소프트웨어 테스팅 표준화 및 자동화가 필요하다. 본 연구에서는 TTCN-3 테스트 표준을 적용한 AUTOSAR 소프트웨어 컴포넌트를 위한 테스트 자동화 방법 및 테스트 시스템을 제안하였다. 테스트 표준을 적용함으로써 테스트에 관련된 정보 교환 및 의사소통이 명확해 지고, 테스트 재사용성을 향상시킬 수 있다. 그리고 설계 모델로부터 테스팅 모델을 자동 생성함으로써 개발 기간 단축 및 신뢰성을 향상시킬 수 있다.

키워드 : AUTOSAR, TTCN-3, 소프트웨어 컴포넌트, 테스팅 자동화, 최악 응답 시간

Abstract AUTOSAR, a standard software platform for automotive, has been developed to manage software complexity and improve software reuseability. However reuse of test system is difficult because it is dependant on implementation language and test phase. In this paper, we suggest a test system generation method for AUTOSAR software component using TTCN-3, a standardized testing language. TTCN-3 test system is generated automatically from AUTOSAR XML containing software design information. The test system consists of TTCN-3 tester and target system and tests functionality and worst case response time of software under simulation environment. With the proposed testing techniques we can reduce time and effort to build the testing system and reuse testing environment.

Key words : AUTOSAR, TTCN-3, Software Component, Automated Testing, Worst Case Response Time

· 본 연구는 교육과학기술부에서 지원하는 기관기요사업비로 수행하였음.
· 이 논문은 제36회 추계학술발표회에서 'TTCN-3를 이용한 차량 소프트웨어 컴포넌트의 테스팅 자동화 방법'의 제목으로 발표된 논문을 확장한 것임

† 정 회 원 : 대구경북과학기술원 미래산업융합기술연구부
kumdh@dgist.ac.kr
shunlee@dgist.ac.kr
ggangmin@dgist.ac.kr
†† 정 회 원 : 경북대학교 전자전기컴퓨터학부 교수
jcho@ee.knu.ac.kr
(Corresponding author임)

논문접수 : 2009년 12월 23일
심사완료 : 2010년 2월 11일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제5호(2010.5)

1. 서 론

최근 복잡한 차량 소프트웨어의 신뢰성 및 재사용성 향상을 위하여 AUTOSAR 표준 소프트웨어 플랫폼을 제정하였지만, 소프트웨어의 표준화만으로는 테스트 테스트에 소요되는 많은 시간과 노력을 줄이기에는 한계가 있다. 소프트웨어가 복잡할수록 테스트 환경 설계에 많은 시간과 노력이 필요할 뿐만 아니라, 테스트 단계에서 오류 발생 확률도 높아진다. 현재의 차량 소프트웨어 테스팅 환경은 개발 언어 및 개발 환경에 종속적이며, 테스트 단계마다 다른 테스트 시스템을 사용해야 하는 문제가 있다.

최근 차량 소프트웨어의 테스팅을 위한 다양한 연구가 진행되고 있지만, 표준화 및 자동화 측면에서 부족한 면이 있다. AUTOSAR 소프트웨어 표준 플랫폼과 더불어 테스트 표준을 도입하여 테스트 환경의 재사용성을 향상시키고, 테스팅을 자동화하여 신뢰성 향상 및 개발 기간을 단축할 필요성이 있다.

본 논문에서는 TTCN-3 테스트 표준을 적용한 AUTOSAR 소프트웨어 컴포넌트의 테스트 환경 자동 생성 방법 및 테스터와 타겟 시스템으로 구성된 테스트 시스템의 구조를 제안하고자 한다. 제안된 테스트 방법은 AUTOSAR XML을 이용하여 기능 테스트와 타이밍 분석이 동시에 가능한 테스트 모델을 생성한다.

표준화된 테스트 환경을 적용함으로써 소프트웨어 표준 플랫폼의 활용도를 더욱 높이고, 테스트 환경 및 데이터의 재사용성을 향상시킬 수 있다. 또한, 하드웨어 개발이 이루어지지 않은 개발 초기 단계에 설계 모델로부터 즉시 실행 가능한 테스트 모델을 자동 생성함으로써 개발 기간 단축 및 신뢰성을 향상시킬 수 있다.

2. 관련 연구

최근 차량 소프트웨어의 테스트는 모델 기반 테스팅 방법을 적용하여 테스트를 자동화하고 있다. [1]과 [2]에서는 소프트웨어의 기능 모델로부터 테스트 케이스를 생성하여 테스트 수행을 자동화하는 방법과 프로세스를 제안하고 있다. 지금까지의 연구는 모델기반의 테스트 케이스 생성에 초점이 맞추어져 있었으며, 테스트 시스템의 생성은 고려하고 있지 않았다. 그리고 설계 도구에서 제공하는 시뮬레이션 환경을 이용하거나 설계 도구에 종속된 테스트 도구를 사용함으로써 테스트 환경과 테스트 케이스의 재사용에 한계가 있었으며, 스케줄링이나 네트워크 통신 환경을 고려한 테스트 환경을 고려하지 않았다. 본 논문에서는 표준화된 AUTOSAR XML 설계 명세를 이용하여 설계 도구에 독립적이고, 스케줄링과 네트워크 통신을 고려한 테스트 시스템과 시뮬레이션 환경을 생성하는 기술을 제안함으로써, 테스트 환경 및 테스트 데이터의 재사용성을 높이고 소프트웨어의 기능과 성능 테스트를 동시에 가능하도록 하였다.

TTCN-3은 테스트 대상 시스템의 구현 언어 및 환경에 독립적인 테스트 케이스와 테스트 시스템 설계를 위한 표준이며[3], 최근에는 AUTOSAR에서 적합성 테스트 표준으로 채택하였다. [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 구조와 새로운 자동 생성 방법을 제안하고 있으며, 또한 타겟 시스템의 시뮬레이션 환경의 구조와 생성 방법을 제안하고 있다.

3. 테스트 시스템

3.1 테스트 자동화 방법

본 테스트 방법은 개발 초기 단계에 하드웨어 및 네트워크 통신의 도움 없이 AUTOSAR 응용 소프트웨어 컴포넌트를 테스트할 수 있는 환경을 생성하고, 테스트 환경의 표준화 및 자동화에 목적이 있다.

TTCN-3 테스트 표준을 적용함으로써 테스트 케이스 및 테스트 실행 환경의 공유가 쉬우며 타겟의 구현 방법에 독립적으로 재사용 가능하다. 그리고 AUTOSAR의 설계 모델로부터 TTCN-3 테스터와 타겟 시스템을 자동 생성함으로써 설계 변경 또는 오류 수정으로 말미암아 반복적으로 수행해야 하는 테스트에 소요되는 노력과 시간을 최소화할 수 있다.

그림 1과 같이 AUTOSAR XML로부터 테스트 시스템 및 소프트웨어 컴포넌트 시뮬레이션 환경 생성을 위한 정보를 추출한다. 추출된 정보를 이용하여 기능 테스트와 최악응답시간 테스트가 동시에 가능한 테스트 시스템을 생성한다. 그리고 테스트 케이스를 위한 XML 스키마를 정의하고 설계함으로써, 테스트 케이스를 TTCN-3 테스트 데이터로 변환하여 즉시 실행 가능하도록 자동화하였다.

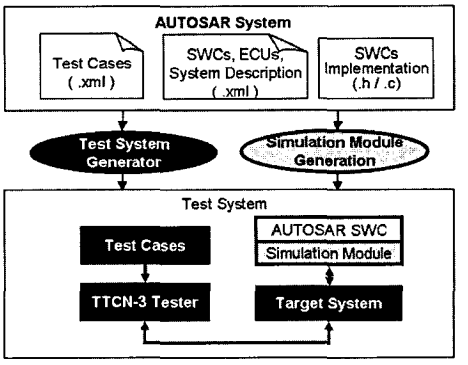


그림 1 AUTOSAR 시스템의 테스트 자동화 개념

3.2 테스트 시스템 구조

AUTOSAR 응용 소프트웨어 컴포넌트를 위한 테스트 시스템은 TTCN-3 테스터와 타겟 시스템으로 나누어지며, 두 시스템은 TCP/IP 네트워크 통신을 이용하여 서로 정보를 교환한다. 테스트 시스템의 상세구조는 그림 2와 같다.

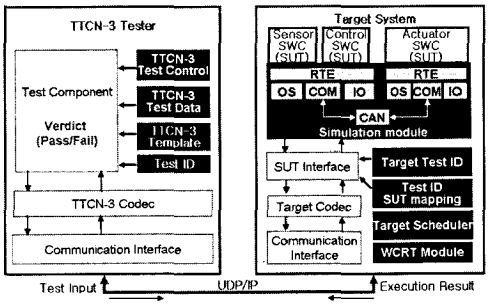
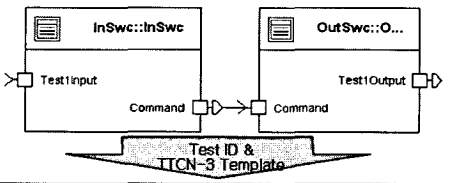


그림 2 테스트 시스템의 구조

TTCN-3 테스터는 테스트 컴포넌트, 코덱 및 통신 인터페이스, 그리고 테스트 케이스와 실행 순서 제어를 위한 테스트 데이터와 테스트 ID 등의 모듈로 구성된다. 테스트 컴포넌트는 테스트 데이터에서 기술된 테스트 입력을 송신하고 소프트웨어 컴포넌트에서 실행된 결과를 수신하여 예상 결과 값과 비교하여 합격 여부를 판단한다. 코덱 및 통신 인터페이스는 타겟 시스템과 테스트 데이터 교환을 위한 모듈이다.

타겟 시스템의 시뮬레이션 모듈은 AUTOSAR 소프트웨어 컴포넌트의 가상 실행 환경을 제공하며, SUT 인터페이스는 테스트 ID를 AUTOSAR 시스템과 맵핑시킨다. 코덱, 통신 인터페이스 모듈은 테스터와 테스트 데이터 교환을 위한 모듈이며, 타겟 스케줄러는 타겟 시스템 전체 스케줄링을 위한 모듈이다. 그리고 WCRT (Worst Case Response Time) 모듈은 테스트 입력 ID와 출력 ID에 대한 최악 응답 시간 분석하여 테스터로 송신하고, 테스터는 마감시간과 비교하여 합격 여부를 판단한다.



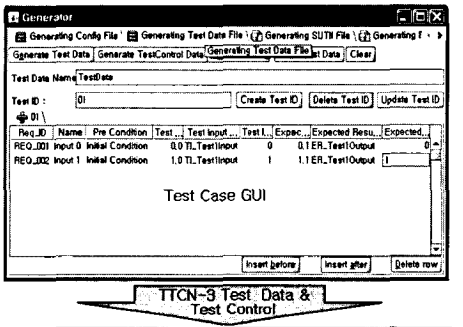

```
/*** Test input signal ID ***/
const integer Test1Input := 1;

/*** Test output signal ID ***/
const integer Test1Output := 2

/*** Test Template ***/
template TestSignal set Test1Input (integer Value) := {
  signalID := Input1Signal,
  signalLength := 1,
  signalValue := Value
  wcrt := 0
}
template TestSignal get Test1Output (integer Value,
integer Deadline ) := {
  signalID := Output1Signal,
  signalLength := 1,
  signalValue := Value,
  wcrt := Deadline
}
```

그림 3 Test ID와 TTCN-3 Template 모듈

3.3 TTCN-3 테스터

그림 2에서 테스트 ID, TTCN-3 템플릿 모듈은 AUTOSAR XML로부터 생성된다. AUTOSAR XML을 이용하여 시스템의 입력과 출력 신호를 검색하여 테스트 ID를 부여하고, 각 테스트 ID에 대한 TTCN-3 템플릿모듈을 생성한다. 템플릿은 결과 값과 최악 응답 시간을 동시에 확인할 수 있도록 설계하였다. 그림 3은 소프트웨어 컴포넌트로부터 생성된 테스트 ID와 TTCN-3 템플릿의 예를 간략히 보여준다.

TTCN-3 테스트 데이터와 TTCN-3 테스트 컨트롤 모듈은 테스트 케이스의 설계 및 실행에 관련된 제어를 한다. 미리 생성된 테스트 ID와 테스트 템플릿을 이용하여 테스트 데이터를 설계하며, 테스트 입력 값, 예상 출력 값, 마감시간 등을 입력한다. 테스트 컨트롤은 설계한 테스트 데이터의 실행 순서 등의 제어를 한다. 그림 4는 GUI에서 설계된 테스트 케이스로부터 생성된




```
/*** Test Data ***/
group TestData_01 {
  const TI TI_01 := { setTest1Input (0) };
  const ER ER_01 := { getTest1Output (0, 0.1) };
}
group TestData_02 {
 const TI TI_01 := { setTest1Input (0) };
  const ER ER_01 := { getTest1Output (0, 0.1) };
}

/*** Test execution ***/
control {
  execute ( TC (TI_01,ER_01,nTI_01,nER_01), 30.0);
  execute ( TC (TI_02,ER_02,nTI_02,nER_02), 30.0);
}
```

그림 4 TTCN-3 테스트 데이터 및 테스트 컨트롤 모듈

```
/*** Test function ***/
function funcSendTestInputs (TI TIs, integer nTIs)
runs on TestCompnent {
  var integer i;
  for(i=0 ; i<nTIs ; i:=i+1){
    pt.send(TIs[i]);
  }
}
function funcEvaluateResult(ER ERs, integer nERs)
runs on TestCompnent {
  var integer i;
  timerTC.start;
  for(i:=0 ; i<nERs-n ; i:=i+1) {
    alt {
      [ ] alt_evaluateResult(ERs, nERs) { }
      [ ] alt_timeGuard (timerTC) { }
    }
  }
}
/*** Test case ***/
testcase TC (TI TIs, ER Ers, integer nTIs, integer nERs)
runs on TestCompnent system System{
  map(mtc:ptTC, system:ptSystem);
  funcSendTestInputs(TIs, nTIs);
  funcEvaluateResult(ERs, nERs);
  all component.done;
  unmap(mtc:ptTC, system:ptSystem);
}
```

그림 5 TTCN-3 테스트 컴포넌트 모듈

TTCN-3 테스트 데이터 및 테스트 컨트롤 모듈의 일부를 보여준다.

테스트 컴포넌트는 테스트 입력 값을 타겟 시스템으로 송신하고, 수신된 결과 값과 최악 응답 시간의 합격 여부를 판단한다. 테스트 컴포넌트의 구현을 위해서 테스트 포트, 테스트 컴포넌트, Altstep, 테스트 함수, 테스트 케이스 함수 등을 설계한다. 그림 5는 TTCN-3 테스트 컴포넌트 모듈을 간략히 나타내었다.

3.4 AUTOSAR 타겟 시스템

타겟 시스템은 소프트웨어 컴포넌트의 실행을 위한 시뮬레이션 환경과 테스트 입력을 수신하고 결과 값을 송신할 수 있는 인터페이스를 제공한다. 그리고 테스트 입력이 수신되어서 출력이 발생하는 사이의 최악 응답 시간을 제공함으로써 기능 테스트와 타이밍 분석을 동시에 가능하게 한다.

개발 초기 단계에 응용 소프트웨어 컴포넌트의 빠른 검증을 위해서는 RTE(Runtime Environment), OS, COM, IO 모듈 및 CAN 통신 등 하드웨어 관련 모듈의 시뮬레이션이 필요하다. 시뮬레이션 환경은 AUTOSAR XML로부터 생성한다.

테스터와 테스트 데이터 교환을 위해서 AUTOSAR XML로부터 추출한 테스트 ID를 이용한다. SUT 인터페이스는 테스트 ID와 소프트웨어 컴포넌트의 입출력 신호를 맵핑시키고, 타겟 코덱과 통신 인터페이스는 테스터와 UDP/IP 통신으로 테스트 ID를 주고받는다. 타겟 스케줄러는 하나 이상의 ECU가 차량 네트워크에 연결된 시스템의 시뮬레이션과 테스트 데이터의 송수신 등 타겟 시스템 전체 스케줄링을 한다. 그림 6은 타겟 시스템의 테스트 ID, 테스트 ID SUT 맵핑, 타겟 스케줄러 모듈의 생성 예를 간략히 나타내었다. 최악 응답 시간 분석 모듈은 다음 절에서 상세히 설명하였다.

```
/*** Test Signal ID ***/
#define SID_Test1Input 1
#define SID_Test1Output 2
/*** Test Signal Mappping ***/
void MapSIGNALID( ) {
  BUF_TI_SIGNAL[0]= SID_ Test1Input;
  BUF_TO_SIGNAL[0]= SID_Test1Output ;
}
/*** TargetScheduler Configuration ***/
#define nENTITY 3
#define ENTITY1 Schedule_ECU1
#define ENTITY2 Schedule_ECU2
#define ENTITY3 Schedule_CAN
```

그림 6 타겟 시스템 생성 모듈

3.5 최악 응답 시간 분석 모듈

차량과 같은 경성 실시간 분산 제어 시스템은 시간 제약에 대한 요구 사항을 만족하지 못한다면 기능상의 오류뿐만 아니라 사고의 가능성도 있으므로, 시스템이 마감시간 내에 실행될 수 있도록 타이밍을 설계하고 예측하는 것은 중요하다. 일반적으로 응답 시간이 최대로 발생할 수 있는 최악 조건에서 시스템이 마감시간을 만족하는지 여부를 분석하는 최악 응답 시간 예측 방법을 사용한다[6,7].

최악 응답 시간 모듈은 각 테스트 입력 ID와 출력 ID에 대한 최악 응답 시간을 계산하여 저장한 2차원 배열이다. 테스트 입력을 수신하여 결과를 송신할 때, 입출력 테스트 ID를 참조하여 2차원 배열에 저장된 최악 응답 시간을 실행 결과 값과 함께 테스터로 송신한다. 그리고 테스터는 최악 응답 시간이 마감시간을 만족하는지를 판단한다. 그림 7은 최악 응답 시간 테스트 방법의 개념도를 보여준다.

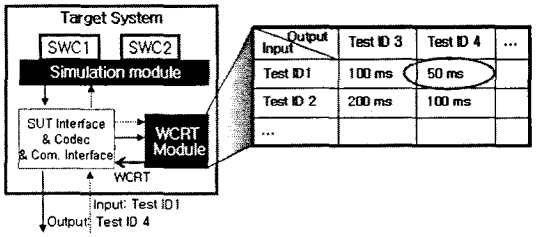


그림 7 최악 응답 시간 테스트 방법

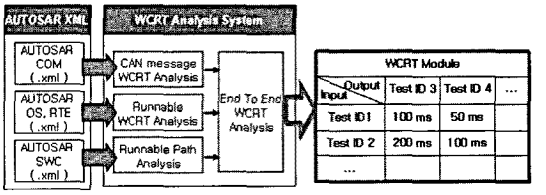


그림 8 최악 응답 시간 모듈 생성 방법

최악 응답 시간 모듈은 AUTOSAR XML로부터 생성하며 다음과 같은 단계로 나눈다. 먼저 AUTOSAR COM으로부터 각 네트워크 메시지의 최악 응답 시간을 계산하고, AUTOSAR OS, RTE로부터 각 런어블의 최악 응답 시간을 계산한다. 다음은 각 테스트 입력 ID와 테스트 출력 ID에 대한 런어블의 실행 경로를 계산하고, 경로 상에 있는 런어블과 네트워크 메시지의 최악 응답 시간과 지터 시간 등을 고려하여 테스트 입력부터 출력까지의 최악 응답 시간을 계산하여 2차원 배열을 완성한다. 그림 8은 최악 응답 시간 모듈의 생성 방법을 보여준다.

4. Case Study - 차량 실내등 시스템

4.1 차량 실내등 테스트 시스템의 생성

본 논문에서 제시한 테스트 자동화 방법을 차량 실내등 시스템에 적용하였다. 실내등 시스템은 6개의 소프트웨어 컴포넌트로 구성되어 있으며, 2개의 ECU가 CAN 네트워크에 연결되어 있다. 입력 신호는 7개의 스위치와 8개의 CAN 신호이며, 출력은 5개의 램프로 구성된다.

그림 9는 테스트 자동화 시스템의 툴 체인을 보여준다. 테스트 데이터 생성기에서 테스트 케이스를 설계하고 TTCN-3 테스트 데이터와 테스트 컨트롤 모듈을 생성한다. TTCN-3 테스터 생성기는 실내등 시스템의 XML로부터 시스템의 입출력을 검색하여 테스트 ID 모

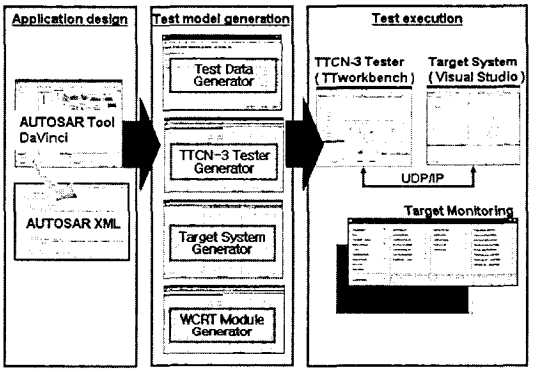


그림 9 테스트 자동화 시스템 툴 체인

듈 및 TTCN-3 템플릿 모듈을 생성한다. 타겟 시스템 생성기도 마찬가지로 테스트 ID 모듈 등을 생성하고, 소프트웨어 컴포넌트를 실행 환경을 위하여, OS, COM. RTE 등으로 구성된 시뮬레이션 모듈을 생성한다. 최악 응답 시간 모듈은 각 런어블 및 CAN 메시지의 최악 응답 시간과 테스트 입출력 ID 사이의 실행 경로를 분석한 결과를 바탕으로 End-to-End 최악 응답 시간 모듈을 생성한다.

4.2 테스트 결과 및 고찰

테스트 시스템을 위한 추가적인 시간과 노력의 투자 없이 소프트웨어의 기능 테스트뿐만 아니라 ECU의 스케줄링 및 네트워크 환경을 고려한 테스트 시스템을 즉시 생성하고 실행할 수 있었다.

일반적으로 소프트웨어의 개발 과정에서 기능 모델의 수정 또는 타이밍 향상을 위하여 스케줄링의 설계 변경, 네트워크 메시지의 설계 변경이 여러 차례 발생하게 된다. 테스트 시스템 생성을 자동화하여 테스트 환경 설계에서 발생할 수 있는 오류를 방지함으로써 소프트웨어의 결함 분석에 대한 신뢰성을 향상시킬 수 있었다. 또한 개발 초기 단계에 시스템의 기능뿐만 아니라 타이밍 테스트를 동시에 수행함으로써 시스템 전체 구조의 검증이 개발 초기 단계에 가능하였다.

시스템의 기능 수정 및 향상을 위하여 소프트웨어 컴포넌트의 배치를 수정할 경우, 마감 시간의 만족 여부를 기능 테스트와 함께 검증할 수 있었다. 타이밍 성능을 만족하지 못하는 경우 네트워크 메시지의 송신 주기를 변경하거나 태스크의 설계를 변경하고 그 결과를 바로 검증할 수 있었다.

5. 결 론

본 연구에서는 AUTOSAR 응용 소프트웨어 컴포넌트를 위한 TTCN-3 기반 표준 테스트 시스템 및 테스트 자동화 방법을 제시하였다. 소프트웨어 설계 정보로부터 즉시 실행 가능한 테스트 시스템을 자동으로 생성함으로써 테스트에 소요되는 시간과 노력을 획기적으로 줄일 수 있었다. 그리고 테스트 표준을 적용함으로써 테스트 케이스 및 테스트 환경의 재사용이 쉬운 장점이 있다.

본 연구에서 제안한 방법을 사용함으로써 테스트 기간을 단축하고 신뢰성을 향상시켜서 차량 소프트웨어 개발의 경쟁력을 높일 수 있을 것이다.

참 고 문 헌

[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] K. Tindell and J. Clark, "Holistic Schedulability analysis for distributed hard real-time systems," *Microprocessors and Microprogramming*, 40(2-3), pp.117-134. 1994.

[7] D. Kum, S. Lee and W. Jung, "Scheduling Analysis in the AUTOSAR Software Architecture," *Proceedings of KSAE 2008 Annual Conference*, pp.2174-2179, 2008. (in korean)