ISSN 1229-800X The Transactions of the Korean Institute of Electrical Engineers Vol. 64P, No. 3, pp. 138~142, 2015
http://dx.doi.org/10.5370/KIEEP.2015.64.3.138

철도신호 소프트웨어 테스팅을 위한 MSC 기반 테스트케이스 생성 모듈

MSC-based Test-case Generation Module for Railway Signaling Software Testing

황 종 규† · 백 종 현*
(Jong-Gyu Hwang · Jong-Hyun Baek)

Abstract - Most of the existing studies on functional safety testing for the railway signaling system software have focused on verifying the functional safety through the monitoring of internal memory embedded railway signaling system. However, the railway signaling system is one of the typical embedded control system in the railway sector, and the embedded software has a characteristic of generating an appropriate outputs through the combination of internal processing in consideration of the current internal status and external input .Therefore, the test approach of using the interface communication channel can be effective way for the functional testing for railway signaling system software in consideration of these characteristic. Since a communication interface specification of the railway signal system has a the properties of the sequence input and output signals, test-case for software testing is the most effective methodology by MSC (Message Sequence Chart) language, one of the graphic language. The MSC-based test-case generating methodology for signaling system software was proposed in this paper.

Key Words : MSC, TTCN-3, Railway signaling system, Software testing

1. 서 론

철도신호시스템은 철도분야에서 가장 대표적인 임베디드 시스템의 하나로서, 다른 제어장치들과 인터페이스를 통한 입출력 정보들이 상호 교환되고 있다. 더군다나 최근 들어 통신기술이 철도신호시스템에 적극적으로 활용되면서 철도신호시스템은 기존보다 더 다른 제어장치들과 인터페이스를 통한 철도신호 기능을 수행하는 시스템으로 진화해가고 있다. 즉, 기존에는 철도신호장치 하나하나별 각기 별개로 고유의 기능을 수행하여왔지만, 최근 들어서는 이들 각 철도신호장치들 또는 다른 철도제어장치들과의 인터페이스를 통한 보다 다양하고 복잡한 기능을 수행하는 철도신호시스템으로 개발 및 운용되고 있다[1, 2].

임베디드 소프트웨어의 검증을 위한 테스팅 도구는 크게 명세기반 테스팅 도구, 소스코드기반 테스팅 도구, 테스팅 관리도구 등으로 구분될 수 있으며, 명세기반 테스팅 도구에는 Quality Architect, CARSE 등이 있으며, 소스코드 기반 테스팅 도구는 정적분석(Static Analysis) 도구와 동적 테스팅(Dynamic Testing) 도구로 구분할 수 있다[1]. 정적테스팅 도구는 소스코드의 코딩규칙의 준수여부 등을 검사하는 도구로서 LDRA[3], CodeScroll Inspector™[4], 철도용 정적 테스팅 도구[5] 등이 있으며, 동적 테스팅은 소스코드 분석을 통해 각종 테스트 커버리지를 분석하는 것으로 Coverage Validator, CodeScroll™ 등이 있다.

명세기반 테스팅 도구는 시스템 소프트웨어의 명세를 바탕으로 데이터를 자동으로 조합하는 기능을 제공하는 테스팅 도구로서, 경계값 분석, 등치영역분할 기능 등 소프트웨어 블랙박스 테스팅 도구들이 대부분이며, 이 중 CARSE는 소프트웨어 신뢰성 분석을 지원하기 위한 도구이다. 이러한 명세기반 소프트웨어 테스팅 도구들은 철도신호 소프트웨어와 같은 임베디드 소프트웨어 블랙박스 테스팅에 활용이 가능하지만, 임베디드 소프트웨어 메모리의 억세스를 통한 테스팅이 대부분이어서 적용에 있어서는 사용하기 매우 어렵기 때문에 실제 적용에는 많은 제약이 있다. 정적분석 도구는 명세기반 테스팅과는 달리 소스코드 자체에 대한 결함을 검증하는 것으로 기능안전성 검증을 위한 블랙박스 테스팅을 수행할 수 없다. 또한 동적 테스팅 도구 임베디드 소프트웨어의 기능안전성 검증이 불가능하다. 따라서 철도신호시스템과 같은 임베디드 소프트웨어의 검증을 위해서는 정적분석, 동적 테스팅 및 기능안전성 테스팅이 모두 필요하며, 이 중 기능안전성 테스팅은 기존의 상용도구의 단점을 극복하기 위한 사용자 편의성을 고려한 새로운 접근방법이 필요하다.

철도신호시스템의 임베디드 소프트웨어는 외부의 입력신호를 받아 철도신호장치의 그 순간의 내부 상태정보들과 융합하여 처리 후 적절한 출력을 하나 이상 생성시키는 특징을 갖는다. 따라서 이러한 임베디드 소프트웨어의 동작특성을 고려하여 철도신호시스템이 실제 사용하고 있는 외부 인터페이스 통신 채널을 통해 기능안전성을 테스팅 하는 방법

† Corresponding Author : Railroad Safety Certification Center, Korea Railroad Research Institute, Korea
E-mail : jghwang@krri.re.kr
* Metropolitan Transportation Research Center, Korea Railroad Research Institute, Korea
접수일자 : 2015년 7월 24일
수정일자 : 2015년 8월 12일
최종완료 : 2015년 8월 16일

이 기능안전성 검증에 효과적인 방법일 수 있다. 철도신호 시스템과 같은 임베디드 시스템 소프트웨어의 검증을 위한 다양한 테스팅 도구들이 있지만, 앞에서 설명한 특성을 갖는 임베디드 소프트웨어의 특성을 잘 고려하지 못하고 있다. 이러한 특성을 고려한 테스팅 도구가 임베디드 시스템이 실제 사용하고 있는 인터페이스 통신 채널을 이용하여 테스트 데이터를 입력 및 피드백을 통한 방법이 최근 들어 연구되고 있다[6, 7].

이러한 임베디드 소프트웨어의 동작특성을 고려하여 철도신호시스템이 실제 사용하고 있는 외부 인터페이스 통신 채널을 통해 기능안전성을 테스팅 하는 방법이 기능안전성 검증에 효과적인 방법일 수 있다. 하지만 기 연구되고 있는 실사용 인터페이스를 활용한 테스팅 방법은 임베디드 소프트웨어 특성을 고려한 적절한 접근 방법이지만, 인터페이스의 실제가 통신이라는 특성에 대한 고려가 부족한 측면이 있다. 인터페이스는 기본적으로 입출력 신호 순서별 시퀀스 특성을 가지게 되는데, 기 연구되고 있는 테스팅 도구는 입출력 인터페이스 신호를 테스팅의 매개체로 활용하면서 시퀀스 특성을 일부만 고려한 방식이다. 즉, 통신프로토콜의 적합성 시험에 적합하도록 개발된 TTCN-3(Testing and Test Control Notation Ver. 3) 스크립트 언어로 테스트케이스를 생성하는 방식으로, 테스트케이스 생성 단계에서 TTCN-3 표준을 사용함으로 시퀀스적인 특성이 고려되었지만, 최종 사용자인 테스터 측면에서는 이러한 특성이 다소 부족한 측면이 있다[8-11]. TTCN-3는 ISO와 ITU에서 표준화한 이동통신 프로토콜, 인터넷 프로토콜의 적합성 시험 등을 지원하기 위한 국제 표준 테스팅 스크립트 언어로서, 기 연구되었던 실사용 인터페이스를 활용한 임베디드 소프트웨어 테스팅을 위해 적합하지만 TTCN-3 스크립트 언어 자체에 대해 지식이 없는 최종사용자인 테스터에게는 다소 어려울 수 있다.

더군다나 철도신호시스템의 외부 인터페이스는 입출력 신호별 순서적인 시퀀스를 가지는 특성이 있으므로, 입출력 신호를 통한 임베디드 소프트웨어 테스팅을 위한 테스트케이스는 시나리오에 따른 시퀀스를 표현하는 그래픽 언어의 하나인 MSC(Message Sequence Chart)로 나타내는 것이 사용자 측면에서 쉽고, 또한 통신프로토콜을 가장 효과적이고 정확하게 표현할 수 있는 방법이다[12, 13].

MSC는 ITU의 Z.120에 의해서 표준화된 그래픽 명세 언어의 하나로서, 통신시스템의 요구명세를 기술하기 위한 전형적인 언어로서 분류되고 있어 MSC를 이용할 경우 프로토콜 명세를 통해 가장 정확하게 테스트케이스를 생성할 수 있고 동시에 최종 사용자인 테스터도 테스트케이스를 쉽게 생성할 수 있다[13-15]. 본 논문에서는 이와 같이 MSC 기반의 테스트케이스의 생성 및 테스트 결과 출력을 통한 철도신호시스템 임베디드 소프트웨어 테스팅 방법을 제안한다. MSC 기반의 테스팅을 지원하기 위해 개발된 도구들은 대부분 통신 소프트웨어 개발 분야에 적합하도록 개발되어 있으며, 대표적인 것인 Tau[16], ObjectGEODE[17]이며, 그 외에 국내에서 이와 관련된 알고리즘에 대한 연구 사례가 있다[18, 19]. 상용화되어 있는 도구인 [16, 17]은 MSC 기반으로 테스트케이스를 생성하기는 하지만 실제 데이터 생성에 있어서 제약이 있어서 다양한 형태의 데이터 입력이 불가능하며, [18, 19]는 MSC 기반의 실험실에서 연구용으로 개발한 것으로 철도신호시스템과 같은 특정 응용에 적용하기에는 GUI 측면 등에서 다소 어려움이 있다. 또한 이러한 툴이나 연구사례들은 모두 테스트케이스 입력은 MSC 기반이지만 테스트 결과는 스크립트 형태로 되어 있어 테스터가 쉽게 사용하기에는 어려움이 있다. MSC 기반의 테스트케이스 생성과 이를 TTCN-3 스크립트 언어로 자동변환을 통한 최종적인 테스트케이스의 생성 및 테스팅을 수행하는 방법은 철도신호시스템과 같이 외부 입력과 내부 상태를 바탕으로 출력을 하는 특성을 갖는 임베디드 소프트웨어의 검증을 위한 보다 높은 정확도를 갖는 테스트케이스의 생성이 가능하고 또한 사용자가 쉽게 사용할 수 있는 장점이 있다.

2. MSC를 이용한 테스트케이스

임베디드 소프트웨어에서 하나의 프로세스는 대부분 전에 수행되었던 결과 값이 반영되어 수행되는 경우가 많다. 예를 들어 현재 열차의 제한속도를 계산하여 출력하기 위해서는 선행열차의 위치 정보가 필요하다. 여기서 선행열차의 위치정보는 현재 열차의 제한속도 값을 출력하기 위한 바로 전 수행결과에 해당할 수 있다. 이처럼 한 프로세스가 수행되기 위해서는 바로 전 프로세스의 결과가 순차적으로 반영되어야 하고 프로세스 수행 후 출력되는 그 결과 값 또한 또 다른 프로세스의 입력으로 이용되어지게 된다. 즉, 각 프로세스들 간에는 순서가 존재하며, 프로세스들의 출력정보는 다른 프로세스의 입력으로 활용하고, 동시에 메시지를 송수신하는 프로세스 사이에도 순서 관계가 있는 특징을 가진다.

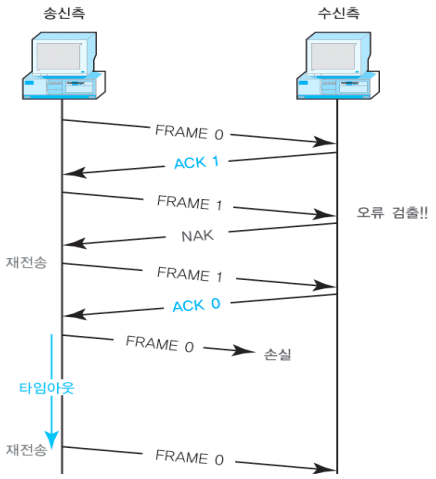

그림 1 MSC로 표시되는 프로토콜 명세 예
Fig. 1 Example of MSC based typical protocol specification

MSC는 앞 절에서 설명한 바와 같이 Z.120으로 표준화된 그래픽 기반 명세언어로서, 통신 프로토콜 명세에 가장 보편적으로 사용되고 있다. MSC는 통신에 참여하는 대상들과, 시간의 흐름을 나타내는 세로축, 그리고 통신 대상들 사이의 시간대별 데이터의 흐름과 데이터의 내용을 간단한 그래픽으로 표현한 방법으로, 쉽게 작성할 수 있으며 또한 통신 프로토콜을 가장 효과적으로 표현할 수 있다.

그림 1은 통신 프로토콜 명세의 일부로서 MSC의 전형적인 표현을 나타낸다. 그림에서와 같이 두 객체 사이에 시퀀스 적으로 데이터의 교환이 이루어지며, MSC에서는 전송되는 정보의 내용이 함께 표기되어 있다. 앞 절에서 설명하였듯이 철도신호시스템의 실사용 인터페이스를 활용한 임베디드 소프트웨어 검증을 위한 테스트케이스는 MSC 형태로 생성이 효과적이며, 생성된 MSC기반의 테스트케이를 국제표준 테스팅 언어인 TTCN-3 스크립트 언어로 자동으로 변환하여 임베디드 소프트웨어 테스팅의 수행하는 방법을 제안 및 지원도구를 개발하였다.

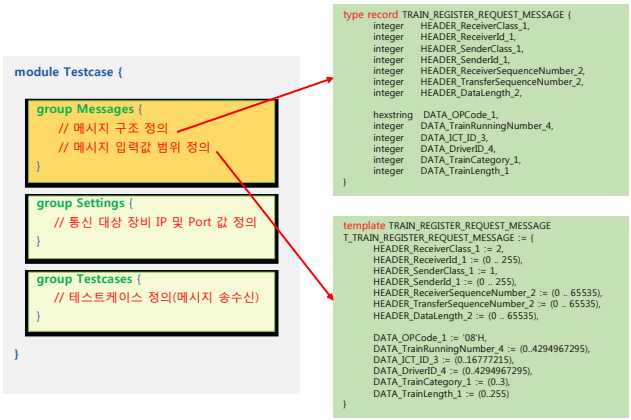

그림 2 TTCN-3 스크립트의 구조
Fig. 2 Structure of TTCN-3 script

기 연구된 테스팅 도구[4]에서의 테스트케이스는 TTCN-3 스크립트로 되어 있다. 그림에서와 같이 테스트케이스는 크게 Message, Settings, Testcase의 3가지로 구성되어 있다. Message 그룹은 메시지 구조의 정의 및 메시지 입력값 범위를 정의하고 있으며, Settings 그룹에서는 인터페이스 대상 장치의 통신정보(IP, 포트번호), 그리고 Testcase 그룹에서는 실제 테스팅을 위한 전송프레임 각 필드의 값 등이 포함되게 된다. 그림 2는 TTCN-3 스크립트로 이루어진 테스트케이스의 구조 및 Messages 그룹의 스크립트 예를 보여주고 있다. 그림에서와 같이 TTCN-3 스크립트에 의한 테스트케이스의 경우 테스터가 필요에 의해 수정하거나 편집하기에 다소 난해할 수 있고, 직관적으로 해당 테스트케이스가 어떠한 내용인지 파악하기 어렵다. 따라서 최종적인 테스트케이스의 생성에 많은 시간과 노력을 필요로 한다.

본 논문에서는 이러한 사용자의 편의성 및 테스트케이스의 직관적인 이해도 향상을 위해 사용자에게 익숙하고 통신 프로토콜 명세에 최적화되어 있는 MSC 기반의 테스트케이스를 입력 및 이를 통한 자동으로 TTCN-3 스크립트로 변환되는 모듈을 개발하였다. 그림 3은 MSC로 테스트케이스를 작성할 경우 프로그램 내부에서 MSC를 구성하는 각 구성요소들을 'Property' 창에 구분하여 저장하는 것을 도식화한 것이다. 즉, MSC 테스트케이스에서 메시지명, 시스템명, 메시지 파라미터를 각각 별도의 속성으로 인식하여 해당하는 부분으로 자동 매칭 되도록 하는 방식으로 저장되도록 하였다. 그림에서와 같이 시스템명이나 메시지명이 위치하는 좌표를 인식하고, 이 좌표정보를 바탕으로 메시지의 순서를 정렬하여 텍스트 파일로 변환된다. 그림 3은 이러한 MSC 형태의 테스트케이스가 텍스트 파일로 저장되는 과정을 나타낸 것이다.

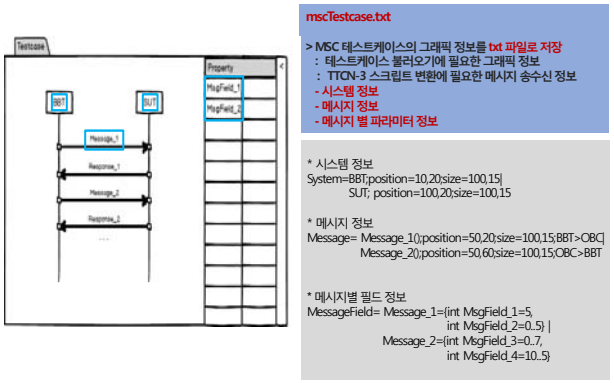

그림 3 MSC 기반 테스트케이스의 텍스트파일 저장
Fig. 3 Text file saving of MSC based test-case

이러한 형태의 MSC로 편집된 테스트케이스가 테스팅 도구에서 입력되는 과정은 [18]의 연구를 활용하였다. MSC에서는 송수신 개체 사이의 일정 순서를 가진 발생 이벤트들로 이루어져 있으며, 이들 이벤트들은 메시지 정보에 해당하며, 이들 이벤트들의 순서를 별도의 파라미터로 관리하는 저장하고, 이 파라미터에서 각 전송메시지를 참조하는 형태이다. 이에 대한 자세한 내용은 [18]을 참조한다.

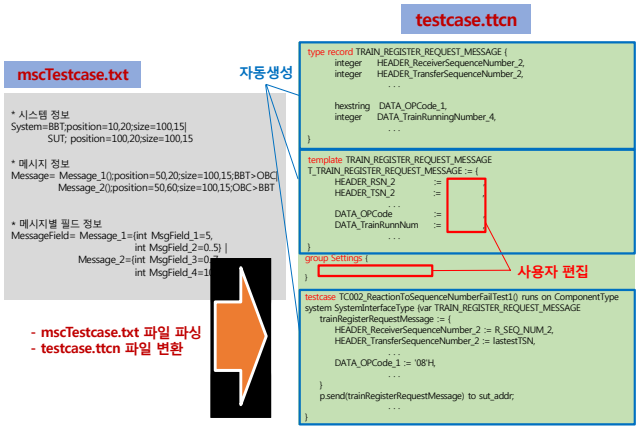

그림 4 TTCN-3 스크립트 파일로의 변환
Fig. 4 Conversion of TTCN-3 script file

그림 3과 같이 MSC 파일이 테스트 파일로 저장되면, 개발한 자동변환 모듈에 의해 이 테스트파일을 파싱하여 TTCN-3 스크립트의 해당 부분으로 그림 4와 같이 자동 변환시키게 된다. 그림에서와 같이 MSC에서 각 필드들의 값을 입력하면, 이 부분이 자동으로 스크립트로 변환되는데 이 부분을 포함한 일부분은 생성된 TTCN-3 스크립트에서 편집할 수도 있다.

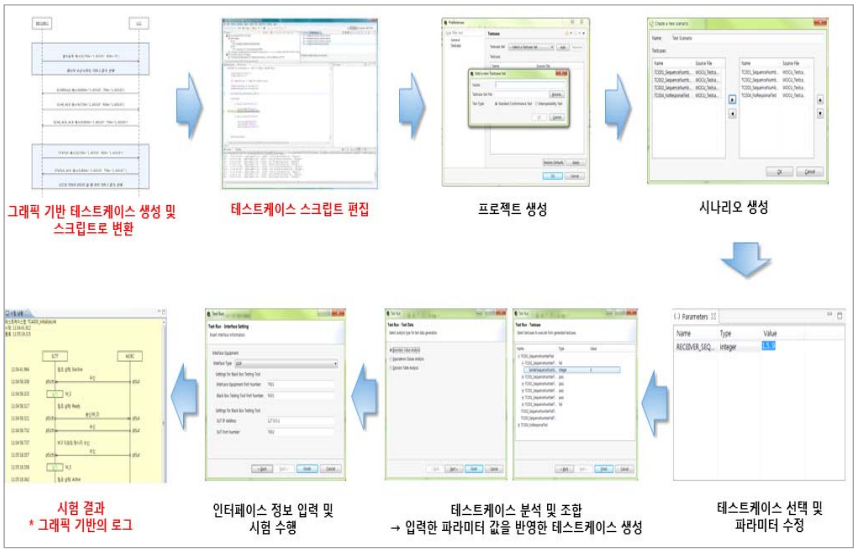

그림 5 테스트케이스 생성을 포함한 테스팅 절차
Fig. 5 Procedure of test-case generation procedure and test results

3. 도구의 개발 및 적용결과

앞 절에서 설명한 철도신호시스템 소프트웨어 블랙박스 테스팅을 위한 테스팅 도구의 MSC 입력 기반의 테스트케이스 자동생성 모듈이 개발되어 기 연구되었던 도구에 적용되었다. 그림 5는 MSC 기반 테스트케이스의 입력 및 스크립트 언어로의 자동생성과 편집, 테스트케이스에 의한 테스팅의 수행 및 테스결과의 표시까지의 일련의 과정을 나타낸 것이다. 그림에서와 같이 MSC 형태로 테스트케이스가 입력되고, 출력 또한 입력과 동일한 MSC 형태가 출력되도록 하였다. 즉, 본 논문을 통해 개발한 부분은 MSC 기반의 테스트케이스 입력 및 출력 모듈로서, 입력과 출력 모두 테스터가 직관적으로 이해할 수 있고 사용하기 편리한 형태로 되어 있다.

그림 6과 그림 7은 MSC 기반의 테스트케이스 편집 윈도우와 출력 윈도우를 나타낸 것이다. 편집윈도우에서는 오른쪽 윈도우에서 해당하는 속성을 선택하여 적절한 곳 위치시

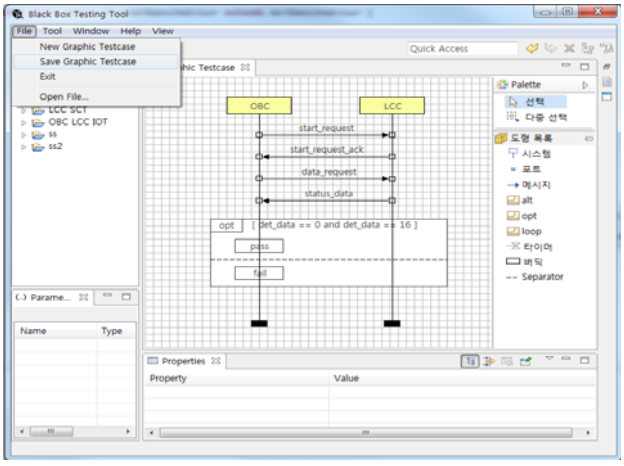

그림 6 MSC 기반 테스트케이스 편집 윈도우
Fig. 6 Windows for MSC-based test-case editing

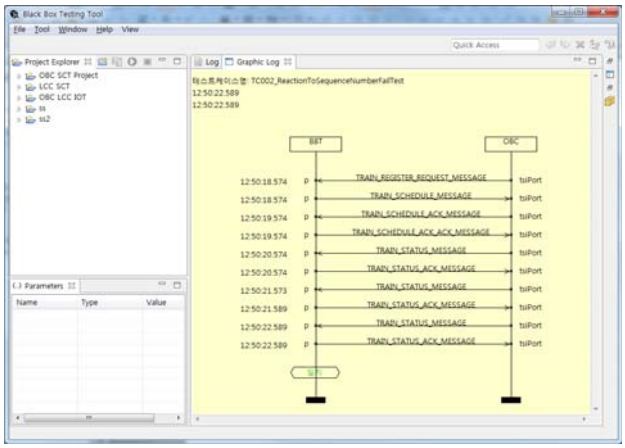

그림 7 MSC 기반 테스트케이스 출력 윈도우
Fig. 7 Windows for MSC-based test results output

키고 이들 각 객체들의 속성을 입력하면, 스크립트 형태로 자동변환 되게 된다. 즉, 사용자는 기존의 스크립트 형태로 테스트케이스를 생성하는 것보다 훨씬 간단하고 쉽게 생성할 수 있고, 이에 따라 생성과정에서의 오류도 줄어들 수 있다. 그림 6의 오른쪽에는 MSC의 다양한 객체들을 나타내고 있다. 그림 7은 테스팅 결과를 MSC 형태로 도식화하여 출력한 것으로, 테스트케이스 명, 테스팅 시작 및 종료시간을 확인할 수 있으며, 그림의 왼쪽은 실제 테스팅을 수행하면서 해당 메시지를 송수신 한 시간을 나타내고 있고, 아래 부분의 "일치"라고 되어 있는 부분은 테스트케이스와 일치하는 것을 나타낸 것으로 해당 테스트케이스에 의한 테스팅 결과 타켓 시스템이 정상동작하고 있음을 의미한다.

4. 결 론

기존의 철도신호시스템 블랙박스 테스팅 도구에서는 TTCN-3 스크립트 기반으로 테스트케이스를 편집 및 생성한다. 하지만 테스트케이스는 스크립트 언어로 생성할 경우

오류가 발생할 수도 있고, 또한 이 스크립트 언어기반의 테스트케이스는 사용자에게 다소 어렵게 느껴질 수 있다. 철도신호시스템과 같은 피시험체의 통신채널을 기반으로 한 테스팅을 위한 테스트케이스는 기본적으로 MSC 형태로 명세화 한다. 따라서 테스터 입장에서는 스크립트 언어로 테스트케이스를 편집하는 것 보다는 직관적으로 이해할 수 있는 MSC 형태로 생성하는 것이 오류도 줄일 수 있는 등 장점을 가진다. 이에 따라 본 논문에서는 용자에게 친숙한 그래픽 언어의 하나인 MSC기반의 테스트케이스를 편집 및 생성하고 자동으로 스크립트 언어로 변환을 통한 테스팅이 수행되도록 하는 모듈을 개발하였다. 또한 테스트 수행 결과인 테스트 로그도 MSC 형태 및 스크립트 형태 두 가지로 출력될 수 있도록 설계 및 개발하였다. 그래픽 에디터를 통한 테스트케이스의 편집으로 테스트케이스의 재사용을 가능하게하고 텍스트 기반의 로그를 사용자가 분석하기 편한 형태로 도식화하여 제공하였다. 개발된 모듈의 적용성 시험 수행 결과 MSC 형태로 입력한 테스트 케이스가 정상적으로 스크립트 언어로 변환됨을 확인하였고, 또한 이 자동 변환된 테스트케이스를 통해 테스팅 수행을 통해 개발도구의 적용성이 확인되었다. 시험결과의 로그도 MSC 형태와 스크립트 형태의 두 가지 형태로 표출되고, 그 두 가지 테스트 로그의 비교결과 정상적으로 출력되었음을 확인하였다.

감사의 글

본 연구는 2015년도 한국철도기술연구원 주요사업의 연구비 지원으로 수행되었습니다.

References

[1] Korea Railroad Research Institute, “Development of safety assessment technology for railway signaling system”, MOCT R&D research report, June 2011.
[2] Korea Railroad Research Institute, “Development of technology for safety and efficiency improvement of train operation", KRRI research report, Dec. 2014.
[3] LDRA, http://www.ldra.com/
[4] CodeScroll Inspector^{TM}, http://www.surecofttech.com
[5] J.G.Hwang, H.J.Jeong and B.H.Kim, “Results of Coding Rules Testing of Train Control System Software”, International Journal of Software Engineering and Its Applications, Vol.7, No.3, pp.249-257, 2013.
[6] J.G. Hwang, J.H.Baek, and H.J.Jo, “Interoperability test methodology for train control system using interface channels”, CMEM 2015 international conference proceeding, May. 1015.
[7] J.G. Hwang, et al, “Architecture of software testing tool for railway signaling through actual use interface channel”, The Journal of Korea Information and Communication Society, Vol.39A, No.9, pp.880-886, 2014.
[8] John Wiley & Sons, Ltd., “An introduction to TTCN-3”
[9] Colin Willcock, “An introduction to TTCN-3, 2nd Edition”, Wiley, 2011
[10] http://www.ttcn-3.org/index.php/downloads/standards
[11] B. K. Song and S. H. Lee, “The development of interoperability tester for the IEC 61850 based on TTCN-3”, Jouranl of IKEEE, Vol. 14, No. 3, 2010.
[12] H. S. Kim, H. S. Bae, and et al, “Development of a testing environment for parallel programs based on MSC specifications”, KIISE Transactions on Computing Practices, Vol.6, No. 2,pp. 135-149, 2000.
[13] E. Rudolph, J. Grabowski, and P. Graubmann, “Tutorial on Message Sequence Charts(MSC'96)”, Tutorial of FORTE/PSTV'96 conference proceedings, Oct. 1996.
[14] ITU-T, “Recommendation X.120 : Message Sequence Chart(MSC)”, Setp. 1994.
[15] J. Graowski, D. Hogrefe, I. Nussbaumer, and A. Spichiger, “Test case specifications based on MSCs and ASN.a,” Proc. of the Seventh SDL Forum 1995, pp.307-322, 1995.
[16] Telelogic Tau, http://www.telecogic.com
[17] ObjectGEODE, http://www.csverilog.org
[18] H.S Bae and Y. R.Kwon, “Validation of timing anf communication constraints in real-time parallel programs”, Phd. Dissertation in Computer Science, KAIST, 1999.
[19] N. H. Lee and et al, “MSCTest : an automated testing tool for embedded software”, Journal of Korea Information Science Society Computing Practice, Vol. 6, No. 2, pp.187-195, April 2000

저 자 소 개

황 종 규 (黃 宗 奎)

1994년 : 건국대학교 전기공학과 졸업
1996년 : 동 대학원 석사졸업
2005년 : 한양대학교 전자통신전파공학과 박사졸업
1995년~현재 : 한국철도기술연구원 책임연구원, 철도신호기술사
2011년~2012년 : Virginia Commonwealth Univ. 방문연구원

백 종 현 (伯 鐘 鉉)

1995년 : 전북대학교 제어계측공학과 졸업
1997년 : 광주과학기술원 메카트로닉스공학과 석사 졸업
2009년 : 전북대학교 메카트로닉스공학과 박사 졸업
1997년~현재 : 한국철도기술연구원 책임연구원

Copyright ⓒ The Korean Institute of Electrical Engineers
This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.