

열차제어시스템 바이탈 소프트웨어 안전성 평가를 위한 테스트 도구의 검토

황종규, 조현정, 윤용기, 정락교
한국철도기술연구원 열차제어연구팀

Review of Automatic Testing Tool for Vital Software Safety Test and Assessment of Railway Signalling Systems

Jong-gyu Hwang, Hyun-jeong Jo, Yong-gi Yoon and Rak-gyo Jeong
Korea Railroad Research Institute (KRRI)

Abstract - The portion of embedded software in railway signaling system is increased by the advance of computer and communication technology. And the safety requirements for railway system are standardized by int'l std., such as IEC 62278 and IEC 62279. According to these circumstances for software safety problems, we are deduced the testing items for software safety test and assessment for railway signaling. Testing tools used for embedded software are surveyed to find a feasible safety assessment architecture. And the architecture of automatic test tool is proposed to test the deduced safety testing items in this paper.

1. 서 론

최근의 컴퓨터 기술의 발달에 따라 열차제어시스템도 컴퓨터 기술에 의해 S/W에 의해 자동화되어감에 따라 시스템화 및 지능화가 이루어지고 있다. 이러한 지능화 및 자동화를 위해 열차제어시스템의 소프트웨어가 더욱 복잡해지게 되면서, 소프트웨어의 비중의 더욱 증대되고 있다. 이에 따라 임베디드화 된 열차제어시스템의 소프트웨어의 안전성이 열차제어시스템의 안전성으로 인지되어가고 있으며, 열차제어시스템은 열차의 안전한 운영에 있어서 매우 바이탈한 설비로 분류되고 있다. 더군다나 최근 들어 국제규격에 의해 열차제어시스템의 소프트웨어의 안전성 확보 및 검증이 더욱 중요해지고 있다 [1][2].

열차제어시스템에서 소프트웨어의 의존도 및 중요도가 증가함에 따라 소프트웨어의 안전성에 대한 테스트 및 검증이 중요한 부분이 되고 있으며, 특히 자동화된 테스트에 대한 필요성이 증대되고 있다[3].

현재 상용화된 임베디드 소프트웨어의 테스팅 자동화 도구에는 국소수이며, 특히 철도시스템의 소프트웨어의 안전성 관련 규격인 IEC62278과 IEC62279의 규격에 준하는 테스팅 도구는 현재 외국에도 존재하지 않고 있다. 상용화된 테스팅 도구를 사용하여 열차제어시스템 소프트웨어의 안전성 테스팅을 위해서는 상용화된 도구 몇 가지를 병행해서 사용하여야 되며, 이 또한 철도 소프트웨어 안전성 관련 국제규격 [1], [2]에서 요구하는 테스팅 항목 모두를 포함하는 것이 불가능한 것으로 분석되고 있다.

본 연구에서는 [1], [2]를 중심으로 철도시스템 소프트웨어의 안전성 평가를 위해 필요한 항목을 분석하였으며, 이를 바탕으로 테스팅 자동화를 위한 항목을 분석하였다. 그리고 열차제어시스템과 같은 임베디드 시스템 소프트웨어의 테스팅을 위해 필요한 요구사항을 분석하였다. 상용화된 테스팅 자동화 툴에 대한 조사를 통해 본 연구를 통해 도출한 테스팅 항목과 비교분석 결과를

본 논문에 간략하게 제시하였다. 최종적으로 본 논문에서는 개발하고자 하는 열차제어시스템 소프트웨어 안전성 평가를 위한 새로운 자동화된 테스팅 도구의 기능 및 구조를 제안한다.

2. 열차제어시스템 소프트웨어 테스팅

2.1 임베디드 소프트웨어 특징

열차제어시스템은 임베디드 제어시스템의 대표적인 예로서, 일반적인 열차제어시스템 소프트웨어의 경우도 다음과 같은 일반적인 임베디드 시스템의 특징을 모두 가지고 있다.

- Event-driven : user-driven 기능과 event-driven 기능이 혼재됨
- Time critical : 시간 제약사항이 있는 경우가 존재함
- Platform stability : 플랫폼(H/W, OS 등)이 매우 다양함
- Monitoring media : 모니터링을 위한 별도의 장비가 없는 경우가 많음
- Memory : 메모리 제약이 극심함
- Hard disk : 하드 디스크가 있는 경우가 별로 없음
- OS facilities : 타이머, 메모리 보호 등 OS 기본 기능이 취약함

열차제어시스템 임베디드 소프트웨어는 위에서 설명한 일반적인 임베디드 소프트웨어의 특징을 모두 가지고 있으으면서, 또한 열차제어시스템 도메인 고유의 요구사항이 요구되고 있다.

- 실시간성 : 정해진 시간 내에 정확한 제어명령이 각 제어장치간에 인터페이스 되어야 안전한 열차의 간격제어, 열차의 안전한 진로의 제어 및 확보 등이 가능. 즉, 소프트웨어는 정해진 시간 내에 기능들을 수행해야 하며, 또한 실행속도는 예측성(Predictability)을 보장하여야 함.
- 안전성 : 열차제어시스템 전체의 안전성을 보장하기 위하여 소프트웨어의 안전성이 확보되어야 함. 이를 위해 최근에 IEC에 의해 철도시스템 소프트웨어가 갖추어야 할 안전성 요구사항이 [1][2]와 같이 국제 규격화 되어 있음.

2.2 임베디드 소프트웨어 테스팅

열차제어시스템 임베디드 소프트웨어의 품질향상 및 안전성 확보를 위해서는 막대한 비용이 소요된다. 소프트웨어의 개발초기에 테스팅 과정을 통해 버그를 확인하여 품질비용(Cost of Quality)을 낮출 수 있다. 하지만 임베디드 소프트웨어는 하드웨어에 의존적이어서 자동화된 테스팅이 필요함에도 불구하고 매우 어렵다.

따라서 임베디드 소프트웨어의 상용화된 테스팅 자동화 툴이 일부 소개되고 있지만, 특정항목에 대한 테스팅 자동화 툴에 그치고 있다. 그럼 2는 임베디드 소프트웨

어 테스팅의 분류를 나타낸 것으로 다양한 형태의 테스팅이 필요하다. 현재 대부분의 테스팅 툴은 정적 테스팅의 자동화에 중점을 두고 있으며, 일부 코드기반 테스팅 툴이 존재하지만 IEC 규격에서 요구하는 열차제어시스템 안전성 확인을 위한 항목들을 포함하지 못하고 있다.

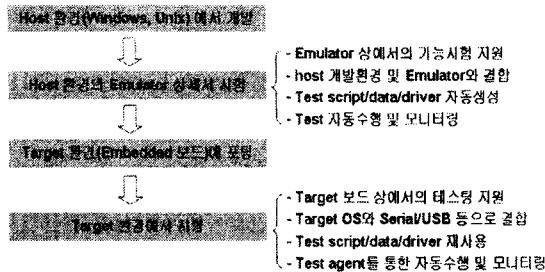


그림 1. 임베디드 소프트웨어 개발 및 테스트 과정

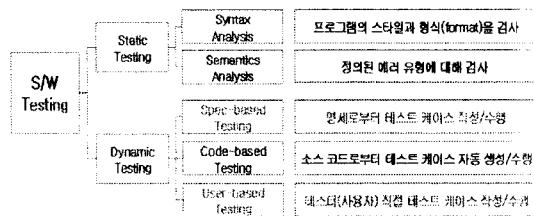


그림 2. 임베디드 소프트웨어 테스팅

표1은 철도시스템 소프트웨어의 안전성 규격인 IEC62278과 IEC62279 규격의 분석을 통해 도출한 열차제어시스템 소프트웨어 안전성 평가를 위해 도출한 14개의 테스팅 항목과 해당 항목의 시험이 가능한 현재 상용화된 툴을 나타낸 것이다. 표에서와 같이 열차제어시스템 소프트웨어의 테스팅을 위해서는 다양한 항목의 시험이 필요하지만 이 시험항목을 모두 커버할 수 있는 툴이 없으며, 또한 상용화된 몇 개의 툴을 동시에 사용하여야 테스팅이 가능한 실정이다.

표 1. 열차제어시스템 S/W 테스팅 항목 및 상용 테스팅 도구 비교

테스팅 항목	기존도구
① Performance Testing	TestRealTime, Load Tester, Pure Load, Intel Vtune
② Boundary Value Analysis	CodeScroll
③ Equivalence Classes	
④ Design & Coding Standard	CodeScroll
⑤ Control Flow Testing	CodeScroll
⑥ Data Flow Testing	CodeScroll
⑦ Fagan Inspection	ReviewPro
⑧ Symbolic Execution	C++ Test
⑨ Checklist	CodeWizard
⑩ Metric 지원	CMT++
⑪ Decision Table	
⑫ Fault Tree Analysis	Relax
⑬ Probabilistic Analysis	
⑭ Impact Analysis	Changer Miner

본 연구에서는 국제규격에서 요구하는 열차제어시스템 소프트웨어의 안전성 평가를 위해서 관련된 국제규격을 분석하였으며, 이 분석을 통해 자동화 가능한 테스팅 항목을 도출하였다. 그리고 바탕으로 열차제어시스템 소프트웨어 안전성 평가를 위한 자동화된 툴의 구조를 제시

하였다.

3. S/W 안전성 평가 테스트 도구

기존의 임베디드 소프트웨어 안전성 평가에 관한 연구들은 평가의 관점이 아닌, 개발초기에 적용이 가능한 수동적 분석방법론들이 주류를 이루고 있으며, 이들을 종합한 결과 기존의 연구결과나 판매되고 있는 도구들은 안전성 평가용으로 활용되기에 부족한 점이 많다고 판단된다. 기존의 안전성 분석기법들의 적용사례들을 볼 때, 대부분이 설계 및 개발과정에 집중되어 있으며, 인증시에는 자동화된 도구를 이용하기 보다는 개발업체에서 제공하는 안전성 활동 데이터를 이용하여 인증 및 평가가 이루어지고 있다[3]. 또한 몇 개 개발과정에서 이용하고 있는 안전성 분석 방법들도 대부분 개발자들의 전문성에 의존해야 하는 수동적인 과정으로 이루어지고 있다.

본 연구에서는 이를 결과들을 종합하여 열차제어시스템 소프트웨어 안전성 평가기법으로 대상시스템을 직접 이용한 평가방법을 기본으로 하였으며, [1], [2] 두 개의 국제규격의 요구사항 분석을 통해 표1과 같은 14개의 자동화된 테스팅 항목을 도출하였다[4]. 도출된 테스팅 항목은 열차제어시스템 소프트웨어의 안전성 평가를 위해 필요한 항목으로서, 이러한 테스팅을 위한 자동화된 도구를 제안하였다. 제안한 도구는 궁극적으로 소프트웨어의 안전성 평가 및 인증 단계에 적용을 목표로 하지만, 소프트웨어의 설계 및 개발단계에서도 동일한 테스팅의 수행을 통해 소프트웨어 안전성 및 품질 향상의 효과가 있어 개발단계에서도 적용이 가능하도록 제안하였다.

테스트 전단계 자동화

테스트 케이스 생성 단계
테스트 수행 단계
테스트 관리 단계
디버깅 단계

다양한 테스팅 기법 제공

체수형 기반 테스팅
데이터 기반 테스팅
소스코드기반 테스팅
스트래스 테스팅

지원하는 시험단계

소프트웨어 모듈 테스팅
소프트웨어 통합 테스팅
하드웨어와 소프트웨어 통합
소프트웨어 안전성 검증
소프트웨어 변경검증

다양한 시험방법 제공

회이트 백스 테스팅
소스코드 분석 테스팅
블랙박스 테스팅

그림 3. 테스팅 자동화 도구의 요구기능

그림 3은 제안한 테스팅 자동화 도구의 요구사항을 나타낸 것으로 6 단계에서 적용이 가능하도록 하였으며, 소스코드 분석, 회이트 백스 테스팅 및 블랙박스 테스팅 모두가 가능하도록 제안하였다.

제안한 열차제어시스템 소프트웨어 안전성 평가를 위한 테스팅 도구는 그림 4와 같이 기본적으로 ‘테스트 케이스 자동 생성 모듈’, ‘테스트 자동수행 및 모니터링 모듈’, 그리고 ‘타겟 테스팅 에이전트 모듈’로 구성되어 있다. 열차제어시스템은 대부분 임베디드 제어시스템으로 되어있어, 용용 S/W가 포팅된 실제 제어보드의 테스팅에 이전트 프로그램을 통해 테스팅 및 모니터링 되는 S/W 테스트 도구의 구조가 설계되어야 한다. 즉, 목표시스템인 테스트 도구는 소스코드와 입력자료 변환모듈을 이용하여 평가대상 소프트웨어의 안전성 분석 데이터를 변환하여 입력받고, 입력받은 소스코드와 안전성 분석 데이터를 바탕으로 테스트 데이터 자동생성모듈과 테스트 시나리오 자동생성모듈을 이용하여 테스트 데이터 및 시나리오를 생성한다. 이 생성된 테스트 케이스를 테스트 자동수행 및 모니터링 모듈과 타겟 테스팅 에이전트

를 통해 테스팅을 수행하여 시험결과를 분석하고, 그 결과를 화면 및 파일로 저장하는 구조를 가진다.

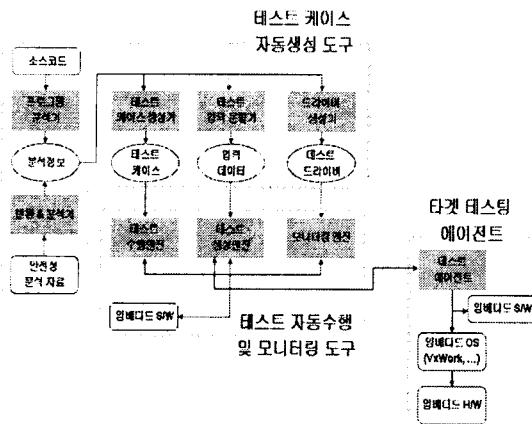


그림 4. 제안 시스템 구조

테스팅 도구는 전체적으로 테스트 단계와 프로젝트 창, 메뉴 바와 옵션들로 구성되어 사용자에게 테스팅이 쉬운 GUI를 제공하도록 하고, 각 시험 항목별 테스팅 자동화를 위해 다음과 같은 대략적인 처리 단계를 갖도록 구성한다.

- 프로그램 분석 : 소스코드 분석을 통해 얻은 정보기반으로 데이터 타입별로 입력 데이터 분할에 대한 정보를 생성.
 - 테스트 케이스 생성 : 테스트케이스 및 테스트 시나리오 생성하는 단계로서, 사용자가 추가 생성할 수 있도록 함.
 - 컴파일 & 빌드 : 테스트 대상 소스코드와 테스트 엔진을 연결하는 드라이버와 테스트가 수행될 프로그램을 생성.
 - 테스트 수행 : 테스트를 수행하고 테스트 커버리지, 구간별 테스트 내역 및 테스트 결과를 파악할 수 있도록 구성.
 - 결과분석 : 모든 테스트 정보와 결과를 사용자의 응선에 따라 리포트를 생성하는 단계

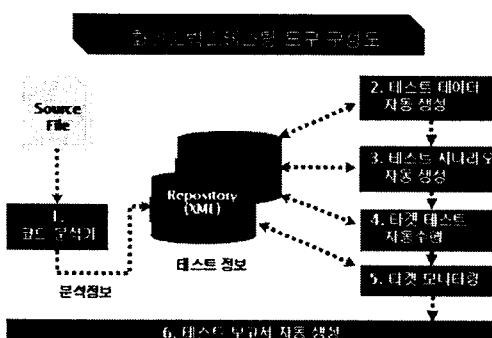


그림 5. 화이트박스 테스팅 모듈의 구조

이러한 열차제어시스템 소프트웨어 안전성 평가 도구는 크게 화이트박스 테스팅, 소스코드 분석 그리고 블랙박스 테스팅 모듈로 나눌 수 있다. 이 중 소스코드 분석 모듈은 소스코드의 코딩이 국제표준을 만족하는지 등에 대한 체크를 하는 모듈로서, 표1에서 제시한 테스팅 모듈 중 ④, ⑦, ⑧, ⑨, ⑪, ⑭ 항목에 해당한다. 또한 화이트박스 테스팅은 ①, ⑤, ⑥ 항목에 해당하는 테스팅을

수행하는 모듈이며, 블랙박스 테스팅은 ②, ③, ⑪, ⑫, ⑬ 항목을 테스팅 하는 모듈이다. 그럼 5와 6은 이중 화이트박스 테스팅 모듈과 소스코드 분석 모듈의 내부 구성을 나타낸 것으로 소스코드 분석 모듈은 그림처럼 소스코드 자체의 코딩 기준 등을 분석하는 기능만 필요하므로 이 모듈은 타겟 시스템 등이 필요 없다. 하지만 화이트박스 테스팅 모듈은 소스코드의 분석을 통한 테스트케이스를 생성하고 또한 타겟 에이전트를 통한 수행을 통해 테스팅을 수행하는 그림 5와 같은 구조를 가지고도록 한다.

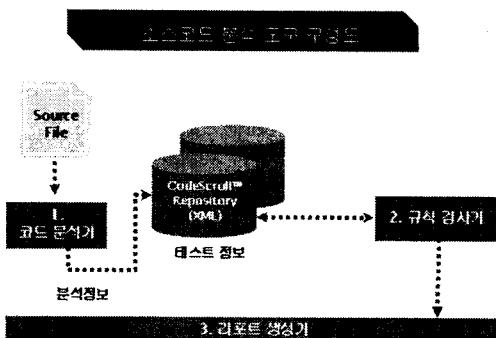


그림 6. 소스코드 분석 모듈의 구조

3. 결 론

최근 들어 열차제어시스템에서 소프트웨어의 비중이 증대되고 있으며, 또한 IEC에 의해 철도시스템 소프트웨어의 안전성관련 요구사항이 국제 규격화되고 있다. 따라서 본 논문에서는 관련된 국제규격의 안전성 요구사항 분석을 토대로 열차제어시스템 소프트웨어 안전성 평가를 위한 테스팅 항목을 도출하였으며, 또한 도출된 테스팅 항목을 자동화할 수 있는 테스팅 자동화 도구를 제안하였다. 열차제어시스템 소프트웨어 안전성 평가를 위한 도구의 자동화를 통해 테스팅 기간이 단축되고, 개발 단계 및 인증단계에서에 오류의 주입 테스팅이 가능하고, 또한 수작업에 비해 테스트 커버리지가 행상되며, 철도 RAMS관련 국제규격에 준하는 안전성 검증 테스팅이 가능하게 될 것으로 기대된다.

[참 고 문 헌]

- [1] IEC 62278, "Railway Applications :The specification and demonstration of RAMS", 2002.
 - [2] IEC 62279, "Railway Applications : Software for railway control and protection systems", 2002.
 - [3] 한재중, 김형신, 황종규, 조현정, 외, "열차체어시스템 소프트웨어 안전성 평가기법", 한국철도학회 춘계학술대회, 2007. 4.
 - [4] 철도안전종합기술개발사업 연구보고서, "열차체어시스템 안전성능 평가 및 사고방지기술 개발", 한국철도기술연구원, 2007. 8