열차제어 S/W 안전성 평가를 위한 절차 및 관리 방안 연구

Study on Procedure & Management for Railway Signaling S/W Safety Assessment

조현정† Jo, Hyun-Jeong　　황종규* Hwang, Jong-Gyu　　김용규** Kim, Yong-Kyu

ABSTRACT

Recent advances in computer technology have brought more dependence on software to train control systems. Hence, the safety assurance of the vital software running on the railway system is very critical task and yet, not many works have been done. While much efforts have been reported to improve the safety of electronic hardware, not so much systematic approaches to evaluate the safety of software, especially for the vital software running on board train controllers.

In this paper, we propose the guide line of actually applicable software safety assessment in train control systems. And we demonstrate the results of developing a new tool based on web system version for software assessment management.

1. 서론

열차제어시스템은 고속으로 동작하는 철도차량의 속도 및 진로를 실시간적으로 제어해야 하는 열차의 핵심 시스템이다. 열차제어시스템은 최근 기존의 기계적 장치로부터 임베디드 시스템을 이용한 컴퓨터시스템으로 전환되고 있으며, 특히 안정화된 하드웨어 보다는 소프트웨어에 그 의존성이 급격하게 증가하고 있다. 특히, 차상 탑재 컴퓨터의 소프트웨어는 열차운행의 자동화 추세에 따라 그 역할이 더욱 중요해 지고 있으며, 따라서 소프트웨어가 열차제어시스템 전체에 미치는 영향도 커지고 있다. 차상 탑재 컴퓨터는 마이크로프로세서 시장의 급속한 발달에 따라 점차 고성능화 하고 있으며, 사용하는 프로그래밍 언어도 단순한 어셈블리어로부터 최근 상위수준 언어가 이용되기도 하였다 [1].

이러한 열차제어시스템 소프트웨어는 임베디드 소프트웨어로서 특징은 실시간성과 고신뢰도, 그리고 안전성을 들 수 있다. 열차제어시스템 소프트웨어가 실시간성과 고신뢰도를 만족하지 못하는 경우 시스템은 위험원이 발생할 수 있으며, 이에 따라 시스템의 안전성이 위협받게 된다. 따라서 이러한 열차제어시스템에 내장되는 제어 소프트웨어의 안전성을 향상시킬 수 있고, 나아가서 효과적으로 관리할 수 있는 기술의 필요성이 더욱 증가하고 있다 [2][3]. 소프트웨어의 안전성을 평가하는 기술은 유럽과 미국의 표준화 기구들이 다양한 안전성 표준들로 규정해 놓고 있으나, 열차제어시스템 소프트웨어와 같이 특정 산업분야의 임베디드 시스템에 적합한 안전성평가 기준에 대한 체계적이고 종합적인 분석은 이루어지지 않았다 [4][5].

† 책임저자 : 정회원, 한국철도기술연구원, 열차제어통신연구실, 주임
E-mail : hjjo@krri.re.kr
TEL : (031)460-5458　FAX : (031)460-5449
* 정회원, 한국철도기술연구원, 열차제어통신연구실, 선임, 공학박사
** 정회원, 한국철도기술연구원, 열차제어통신연구실, 책임, 공학박사

우리나라의 경우, 철도차량 및 시스템의 국산화가 상당부분 이루어져 해외로의 수출을 앞두고 있으나, 이러한 선진국의 소프트웨어 안전성 평가 기준에 부합하는 소프트웨어를 개발하지 못한 실정이다. 그 이유로는 국내에는 소프트웨어 개발 체계를 감리하고, 선진국의 표준에 준하는 안전성 평가기준이 없기 때문이다. 따라서 본 논문의 2절에서 국내 열차제어시스템의 소프트웨어 안전성을 평가하기 위한 가이드라인 형태의 구체적인 방법을 제안할 계획이다. 또한, 3절에서는 이를 체크리스트 형태로 효과적으로 관리하기 위해 개발한 툴을 제시할 것이다. 마지막으로 4절에서 본 논문의 연구결과를 결론짓는 순서로 구성한다.

2. 열차제어시스템 소프트웨어 안전성 평가를 위한 가이드라인 제시

이 절에서는 열차제어시스템 소프트웨어의 안전성 평가 가이드라인을 만들기 위해 소프트웨어 안전 관련 표준에서 제시하는 안전성 요구사항 및 평가 절차를 확인한다 [6]. 또한, 그를 바탕으로 각 평가절차별 체크리스트를 생성한다.

2.1 열차제어시스템 소프트웨어 안전성 평가 방법

소프트웨어 안전성 인증은 피인증 소프트웨어가 설계과정에서 도출된 목표 안전도 등급을 어느정도 달성하였는가를 평가하고, 그 결과에 의거하여 목표의 달성 여부를 판단하는 과정이다. 본 논문에서는 일반적인 전기전자관련 안전성 표준인 IEC61508과 열차분야의 안전성 표준인 IEC62279를 평가의 기준으로 사용하였다. 인증을 위하여 피인증 소프트웨어의 기능 안전성을(Functional Safety) 평가하고, <그림 1>과 같은 소프트웨어 개발과 관련한 개발수명주기에서의 전반적인 안전성 활동의 수준을 평가하여, 피인증 소프트웨어의 기능안전성(Functional Safety)과 안전무결성등급(Safety Integrity Level, SIL)을 검증하고, 그 결과가 목표로 한 소프트웨어 안전성에 얼마나 부합했는지(Conformance)에 따라 인증여부를 결정한다.

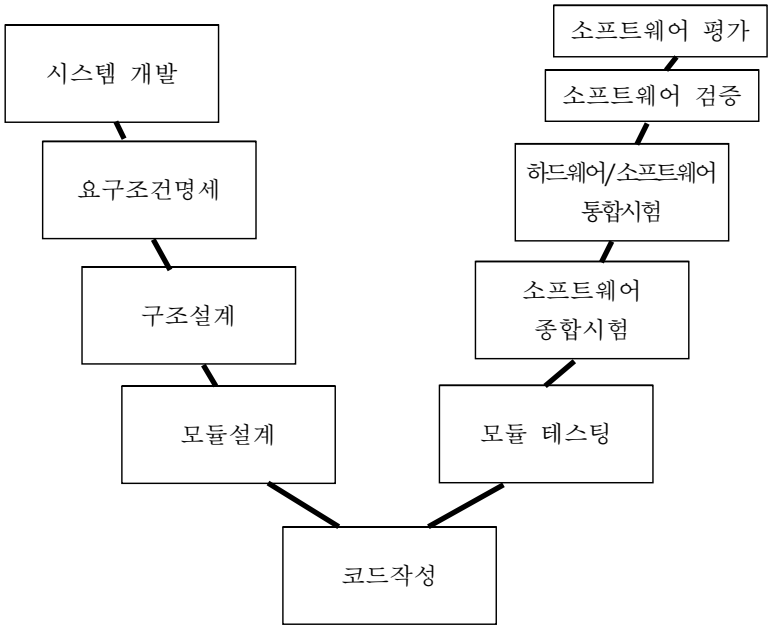

그림 1. 소프트웨어 개발 수명 주기

소프트웨어 안전성 인증은 피인증 소프트웨어가 설계과정에서 도출된 목표 안전도 등급을 어느정도 달성하였는가를 평가하고, 그 결과에 의거하여 목표의 달성 여부를 판단하는 과정이다. 본 논문에서는 일반적인 전기전자관련 안전성 표준인 IEC61508과 열차분야의 안전성 표준인 IEC62279를 평가의 기준으로 사용하였다. 인증을 위하여 피인증 소프트웨어의 기능 안전성을(Functional Safety) 평가하고, 소프

트웨어 개발과 관련한 개발수명주기에서의 전반적인 안전성 활동의 수준을 평가하여, 피인증 소프트웨어의 기능안전성(Functional Safety)과 안전무결성등급(Safety Integrity Level, SIL)을 검증하고, 그 결과가 목표로 한 소프트웨어 안전성에 얼마나 부합했는지(Conformance)에 따라 인증여부를 결정한다.

소프트웨어 안전 무결성 수준(Software Safety Integrity Level, SWSIL)은 소프트웨어 자체적으로 정의되지 않으며, 소프트웨어를 적용하는 시스템의 SIL과 동일하도록 결정된다. 그러나 소프트웨어의 오류가 시스템으로 전파되지 않도록 할 수 있는 경우에는 이보다 낮은 수준으로 정할 수 있도록 하였다. SWSIL은 시스템의 위험도에 따라 다음과 같이 5등급으로 분류한다.

SWSIL	안전 무결성
4	매우 높음
3	높음
2	보통
1	낮음
0	비안전성 등급

소프트웨어 개발 프로세스는 <그림 1>과 같이 개발 프로세스와 검증 프로세스들로 구성되어 있다. 표준에서는 각 개발 단계마다 만족해야 하는 요건들을 제시하고 있으며, 주요 요건들에 대해서는 SWSIL에 따라 의무(M, Mandatory), 강한권고(HR, Highly recommend), 권고(Recommend), 비권고(Not Recommend) 등으로 구분하여 요건의 평가방법을 제시하고 있다.

열차제어시스템의 소프트웨어 안전성 평가는 앞서 언급한 국제표준에서 요구하는 사항들에 대한 만족도를 평가하여 최종 그 결과에 따라 인증여부를 결정하도록 한다. 정의하고 있는 요구사항은 크게 문서화, 소프트웨어 품질관리 체계, 소프트웨어 안전수명주기 요구조건, 기능안전성 평가의 네 가지 분야로 구성된다. 따라서 인증은 이들 네 가지 분야에서의 평가결과를 근거로 이루어지도록 한다.

첫 번째, 문서화 만족도 평가는 소프트웨어 개발과정에서 문서들이 체계적으로 작성되고 있으며, 관리되고 있는지를 평가한다. 문서들은 소프트웨어의 기능 안전성과 SIL 판정을 위하여 요구되는 근거자료들을 체계적이며 명료하게 기술하고 있어야 한다.

두 번째, 소프트웨어 품질관리체계 평가는 소프트웨어 개발주기 동안의 관리 및 기술적 활동에 대한 만족도를 평가한다. 소프트웨어 개발 전주기에 대한 안전성활동 계획을 정의해야 하고, 소프트웨어 형상관리의 적절성 여부도 주요 평가대상이다.

세 번째, 소프트웨어 안전수명주기 요구조건평가는 안전성 생명주기의 각 프로세스마다 피인증 소프트웨어의 개발이 표준에서 요구하는 절차를 어느 정도 충족하고 있는지를 평가한다.

마지막으로 네 번째인 기능안전성 평가는 안전성 생명주기의 각 프로세스마다 평가하는 소프트웨어의 목표 SIL에서 요구되는 항목들의 만족도를 평가한다.

소프트웨어 안전성 인증과정은 특정 기관이 개발한 소프트웨어의 기능 안전성 및 안전무결성을 평가하고, 소프트웨어가 표준에서 요구하고 있는 문서화 및 품질보증절차를 따라 개발되었는지를 평가한다. 인증평가는 평가 목적에 부합될 수 있는 문서화된 평가 단계들로 구성된다.

소프트웨어의 안전성을 평가 받기 위한 업체는 인증기관이 필요로 하는 모든 자료를 제공해야 한다. 인증기관에서는 그 자료와 본 문서를 바탕으로 소프트웨어의 안전성을 4단계로 나누어서 평가한다. 소프트웨어 안전성 등급은 최종 생산물인 소프트웨어 자체의 안전성과 소프트웨어 개발 단계의 완전 무결성을 통합하여 평가된다. 필요에 따라서 인증기관은 본 문서에서 소개되지 않은 평가 기법을 적용할 수 있다. 하지만, 그 평가 기법의 적용이 소프트웨어의 안전성 등급에 큰 영향을 끼칠 경우, 그 이유와 적용이 타당함을 객관적으로 입증해야 한다. 평가자는 소프트웨어 안전성 평가를 위해 다음의 정량적 기준을 이용하여 요구조건들의 만족도를 4단계로 평가한다.

a) F(Fully achieved) 85% 초과 100% 이하 만족
b) L(Largely achieved) 50% 초과 85% 이하 만족
c) P(Partially achieved) 15% 초과 50% 이하 만족
d) N(Not achieved) 0% 이상 15% 이하 만족

2.2 열차제어시스템 소프트웨어 안전성 평가 가이드라인의 작성 절차

<그림 2>는 소프트웨어 안전성 평가를 위한 가이드라인의 작성 과정을 나타낸 것이며, 문서화 과정의 사례를 제시한 결과이다. IEC61508에서 정의된 소프트웨어 개발의 문서화 요건을 바탕으로 피인증 소프트웨어의 문서화 정도를 인증하기 위한 만족도 평가항목을 제안한다. 여기에서 문서화에서 문서로 인정되는 매체는 스크린이나 디스플레이로 표현될 수 있는 컴퓨터 파일, 종이, 필름 등이다.

소프트웨어 개발 단계 전체의 각 단계에 대한 문서기록이 충분한 정보를 포함했는지 확인해야 한다. 각 단계의 문서는 후속 단계와 검증 활동을 효과적으로 수행하기 위해 필요한 정보를 담고 있어야 한다. 또한, 기능 안전성의 관리에 필요한 정보를 충분히 담고 있어야 한다. 각 단계의 문서들은 제안하는 가이드라인을 이용해서 실시할 소프트웨어 안전성 인증의 바탕자료가 되므로 인증기관에서 모든 문서를 이용할 수 있도록 작성되고 관리되어야 한다.

각 문서들은 문서의 범위를 나타내는 제목이나 명칭이 부여되어 있어야 한다. 또한, 색인 기능을 제공해야 한다. 문서의 구조는 해당 회사의 절차와 업무 관행을 반영할 수 있으므로 인증기관은 문서의 구조를 파악해야 한다. 문서들은 개정 번호를 포함해야 한다. 이는 해당 문서의 다른 버전을 확인할 수 있도록 하는 요소이다. 또한, 문서들은 최신 개정 번호에 의해서 관리되고 있는지 확인해야 한다. 모든 관련 문서가 개정, 수정, 검토, 승인을 받아야 하며, 적절한 문서 통제 체계에 의해 관리되고 있는지 확인해야 한다.

이와 같이 소프트웨어 안전성 평가를 위한 가이드라인을 만들기 위한 절차로서 우선적으로 안전관련 표준에서 제시하는 안전성 요구사항 및 평가절차를 확인하는 과정을 거쳐야 한다. 사례로서 제시한 소프트웨어 안전성 평가를 위한 문서화 작업이외에 품질관리시스템, 개발 단계별 안전성 평가, 소프트웨어 기능 안전성 평가와 같은 각각의 분야에 따라 적용되는 방법에 대한 설명과 함께 평가절차별 최종 체크리스트를 생성하도록 한다.

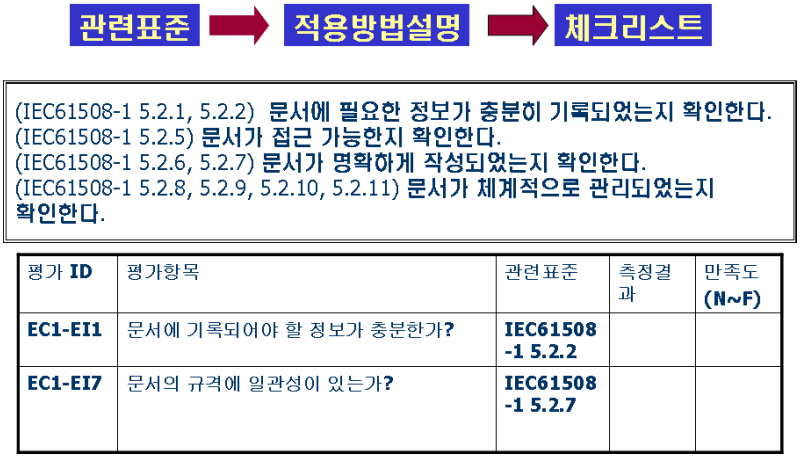

그림 2. 소프트웨어 안전성 평가 가이드라인 작성 예제

3. 열차제어 소프트웨어 안전성 평가 관리를 위한 체크리스트 도구 제안

본 절에서는 열차제어시스템의 소프트웨어 안전성 평가 관리를 자동화하여 효과적으로 수행하기 위해 2절에서 언급한 가이드라인의 산출물인 체크리스트를 활용하는 도구를 개발하였다. 체크리스트는 소프트웨어의 기능 안전성을 평가하기 위한 하나의 평가 기법으로서 최종적으로 시스템 적합성을 평가

하는데 적용할 수 있는 기법이다.

요구사양 명세서에 기초한 것보다 시스템의 모든 관점에 대한 중요한 평가를 제공하기 위한 시험 기법으로 질문에 대한 사람에 의한 체크리스트 수행으로 완성된다. 대부분의 질문은 각각의 시스템 적합성을 평가하기 위한 일반적인 사항이다. 이 기법은 소프트웨어의 다양성과 시스템의 검증에 용이하며, 대부분의 체크리스트는 많은 형태의 시스템에 적합한 질문을 포함한다. 결과적으로 체크리스트의 질문을 사용하여 시스템이 반드시 피해야 하는 사항과 연관이 없음을 확인하는 것이며, 시스템에 직접 명확한 질문의 체크리스트를 적용하기 위해서는 각각의 시스템 별로 체크리스트가 요구되는 것이다.

이러한 체크리스트는 최대한 간결해야 하며, 추가적인 정의가 요구되는 경우에는 반드시 참조된 문서를 추가해야 한다. Pass, Fail 그리고 결정할 수 없음 또는 이와 유사한 결정들은 반드시 각각의 질문에 대하여 기록되어야 한다. 이런 간결함이 체크리스트를 사용한 평가결과의 최종목표이며, 그 결과는 다른 기법을 사용한 것과 일치하여야 한다. 본 논문에서는 체크리스트를 Web System 상에서 효과적으로 관리하기 위한 도구를 개발하였으며, 그 결과는 다음과 같다. <그림 3>은 개발한 도구의 실행 화면 중, 항목별 평가 진행 과정을 보여주는 그림이다.

그림 3. 제안하는 도구에서의 소프트웨어 안전성 평가 실행 화면

먼저 <그림 3>에서와 같이 관리자가 체크리스트의 항목들을 관리할 수 있도록 소프트웨어 안전성 평가 기준에 따라 평가 ID, 평가항목, 관련표준 등을 입력하고 각 항목 별로 수정 및 삭제가 가능하도록 하였다. 또한 더불어서 항목들이 일정한 기준에 의거하여 큰 변화가 일어날 경우는 체크리스트의 항목들을 묶은 Set 별로 버전 관리가 가능하다. 평가자 입장에서는 관리자가 정의한 체크리스트의 항목을 바탕으로 각 평가 대상들에 대한 평가를 Web System을 이용하여 진행할 수 있도록 만들었으며, 각 항목 별 평가 내용을 입력, 수정, 조회 및 삭제가 가능하도록 개발하였다. 이는 최종적으로 종합 관리되어 체크리스트 평가 결과물에 대한 전체 평가가 또다시 이루어질 수 있도록 하였다. 개발한 체크리스트 도구의 기능을 정리하면 다음과 같다.

- 체크리스트를 활용한 평가 진행, 평가 내용 수정 (항목별 N~F 만족도 평가, 평가에 대한 의견 입력, 수정, 삭제 등)

- 프로젝트 내 체크리스트 Category 별 결과 확인
- 프로젝트 별 체크리스트 진행 현황
- 프로젝트 별 평가 결과 보고서 작성

4. 결론

최근 컴퓨터 기술의 발달에 따라 열차제어시스템들이 컴퓨터 소프트웨어에의 의존성이 급격하게 증가하고 있으며, 이러한 기술발전에 따라 열차제어시스템 소프트웨어에 높은 신뢰성과 안전성이 요구되고 있다. 따라서 본 논문에서는 열차제어시스템의 안전성 확보를 위해 소프트웨어 안전성 평가의 구체적인 절차, 즉 가이드라인을 제시하였다. 2장에서 언급한 열차제어 소프트웨어 안전성 평가 가이드라인의 평가 항목들 및 체크리스트를 이용한 평가 기법은 국내 열차제어시스템 소프트웨어 분야에 미비한 수준인 평가기준이나 인증체계를 구축하는데 도움이 될 수 있을 것이다.

또한, 열차제어시스템의 소프트웨어 안전성 평가 관리를 효과적으로 수행하기 위해서 가이드라인에서 제시하고 있는 체크리스트를 활용한 도구를 개발하였다. 개발한 도구를 사용하면 Web System 상에서 관리자나 평가자나 모두 손쉽게 소프트웨어의 기능 안전성을 평가하고 관리할 수 있으므로 나아가 시스템 적합성을 평가하는 것에도 사용할 수 있을 것으로 기대된다. 본 논문에서 제시하는 열차제어시스템 소프트웨어 안전성 평가를 위한 항목 및 가이드라인과 이를 위해 개발한 도구가 향후 열차제어시스템의 안전성을 개선하는데 기여를 할 것이며, 활용가치가 매우 크다고 내다본다.

참고문헌

1. Lawson H. W. et al (2001), "Twenty years of safe train control in Sweden", Engineering of Computer Based Systems, Proceedings of 8th Annual IEEE International Conference and Workshop, pp. 289.
2. EN5IEC 62279 (2002), "Railway Applications : Software for railway control and protection systems".
3. IEC61508 (1998), “Functional safety of electrical/electronic/programmable electronic safety-related systems”.
4. Ian B. Pirie (1999), "Software –How do we know it is safe?", Proceedings of the 1999 ASME/IEEE Joint, pp.122-129.
5. J.D. Lawrence (2000), "Software qualification in safety applications", Reliability Engineering & System Safety, Vol. 70, No. 2., pp. 167-184.
6. 황종규, 조현정, 김형신 (2008), “열차제어시스템 소프트웨어 안전성 평가도구의 설계”, 한국철도학회 논문집, 제11권 제2호, pp.139-144.