안전 필수 철도 시스템 개발을 위한 요구 사항의 정형 명세 작성

(Development of the Formal Requirements Specification of the Safety-critical Railway Systems)

이 진 호 † (Jeanho Lee) 황 대 연 † (Daeyon Hwang) 김 진 현 † (Jinhyun Kim) 박 준 길 † (Junkil Park)

최 진 영 †† (Jin-Young Choi) 황 종 규 ††† (Jong-Gyu Hwang) 윤 용 기 ††† (Yong-Ki Yoon) 조 현 정 ††† (Hyun-Jeong Jo)

요 약 철도 제어 시스템은 대표적인 안전필수 시스템이다. 국제 규격의 컴퓨터 기반의 철도 제어 시스템을 개발하기 위해서 정형 기법을 이용한 요구사항 명세와 검증이 요구된다. 본 논문에서는 정형 기법을 사용하여 요구사항 명세를 작성하는 지침서(guideline)을 개발하고, 컴퓨터 기반의 열차 제어 장치 시스템에 대한 실제 적용 사례를 제시한다. 정형 명세를 위해 상태차트(statechart)와 Z를 사용하고, 정형 명세의 일치성(consistency)과 완전성(completeness)을 검증한다.

키워드 : 정형명세, 요구사항 공학, 일치성, 완전성, 철도제어 시스템, 상태차트, 제드, IEC 61508, IEC 62279

Abstract A railway control system is one of the typical safety-critical systems. It is required to use formal methods for the requirements specification and verification in order to develop the global-standard railway control systems based on the computer systems. In this paper, we develop a guideline for requirements specification using formal methods, and present a case study of the development of a computer-based railway control system through the application of the proposed guideline. We use the Statechart and the Z method for the formal requirements specifications and verify the consistency and completeness of the formal specifications of the requirements.

Key words : formal specification, requirements engineering, consistency, completeness, railway control system, statechart, Z, IEC 61508, IEC 62279

· 이 논문은 2008년도 2단계 두뇌한국(BK)21 사업에 의하여 지원되었음.
· This work was supported by the second stage of the Brain Korea 21 project in 2008

† 학생회원 : 고려대학교 컴퓨터학과
jhlee@formal.korea.ac.kr
dyhwang@formal.korea.ac.kr
jhkim@formal.korea.ac.kr
jkpark@formal.korea.ac.kr
†† 종신회원 : 고려대학교 컴퓨터학과 교수
choi@formal.korea.ac.kr
††† 비 회 원 : 한국철도기술연구원 열차제어통신연구실 선임연구원
jghwang@krri.re.kr
ykyoon@krri.re.kr
hjjo@krri.re.kr

논문접수 : 2008년 7월 8일
심사완료 : 2008년 10월 31일

정보과학회논문지 : 소프트웨어 및 응용 제35권 제12호(2008.12)

1. 서 론

안전 필수 시스템의 개발을 위해서 다양한 분야에서 정형 기법의 사용이 의무화되거나 권고되고 있다[1]. 철도 관련 시스템은 고장이나 오류의 발생 결과가 인명과 재산, 환경에 위해를 가져올 수 있는 안전 필수 시스템이다.

정형기법은 신뢰성있는 시스템을 만들기 위한 방법으로서, 수학적 기반의 언어와 개념을 도입해서 시스템을 명세하고 검증하는 기법과 도구를 가리킨다[2]. 정형기법은 소프트웨어 공학의 시스템 개발 주기에 적용되어 초기 단계에 해당하는 요구사항 명세와 시스템 모델링 단계에서 주로 사용되고 있다[3]. 정형기법을 사용하는 것이 개발할 시스템의 정확성을 보장하지는 않는다. 시스템의 체계적인 개발 단계에서 정형 기법을 사용함으로써, 요구사항 공학(requirements engineering)에서 강

조하는 요구사항의 불일치성(inconsistency), 애매모호성(ambiguity), 불완전성(incompleteness)을 추출해내고, 요구사항 단계에서 이런 문제점들을 제거하여 개발할 시스템의 보다 정확한 이해와 기능적 정확성을 높일 수 있다[4].

철도 시스템과 관련하여 여러가지 국제 표준이 개발되어 사용되고 있다. 현재 철도 시스템 관련 국제 표준은 상위등급을 인증받기 위해 정형기법의 사용을 명시하고 있으나 비실용적인 측면이 있다. 첫째 최근의 철도 제어 시스템은 컴퓨터 혹은 임베디드 시스템을 기반으로 하고 있지만 이것을 반영한 체계적인 열차 제어 시스템 개발 프로세스가 없다. IEC 62425[5]의 경우 열차 제어가 아닌 통신 처리를 위한 전자 시스템에 대해 안전성 활동 프로세스의 내용을 기술하고 있다. IEEE 1474.1-2004[6]의 경우 통신 기반 열차 제어 시스템의 성능과 기능 요구사항에 대해 기술하고 있다. 둘째, 시스템 개발에서 정형기법의 사용에 대한 구체적인 가이드라인이 제공되어 있지 않다. IEC 61508[7]의 경우 전기 전자 시스템 개발 주기에서 요구사항 명세와 디자인과 개발 단계에서 정형기법을 사용할 것을 권고하고 있지만, 구체적인 활용 사례나 지침에 대한 내용은 없다. 셋째 컴퓨터 정보통신 분야의 IEEE 830[8]처럼 철도시스템의 요구사항을 명세하기 위한 표준 가이드라인이 없다.

본 논문에서는 컴퓨터 기반의 철도 제어 시스템을 개발하기 위한 요구사항을 정형 명세하는데 적합한 가이드라인을 제시하고, 실제 철도 제어 시스템을 정형 명세를 통해 요구사항의 완전성과 일치성을 분석한다. 본 논문은 정형 명세 도구로서 상태차트(statechart)[9]와 Z 명세기법[10]을 사용하였다.

논문의 구성은 다음과 같다: 2장에서는 컴퓨터 기반 철도 제어 시스템을 위한 표준들을 살펴보고, 정형기법의 특징과 요구사항 엔지니어링의 특징을 설명한다. 3장에서는 컴퓨터 기반의 철도 제어 시스템을 개발하기 위한 요구사항의 정형명세 작성을 위한 가이드라인을 제시한다. 4장에서는 제안한 가이드라인의 적용 사례로서, 현재 한국철도기술연구원(KRRI)가 개발하고 있는 컴퓨터 기반 철도 제어 시스템의 핵심 부분중 일부인 DCM (distance control module)에 대해 상태차트와 Z를 이용하여 정형명세를 수행하고 결과를 제시한다. 5장에서는 가이드라인에 따라 정형명세된 요구사항의 완전성과 일치성을 분석한다. 6장에서는, 결론 및 향후 연구에 대해 언급한다.

2. 관련 연구

2.1 철도 시스템 관련 표준

철도 시스템 관련 표준에서 다루는 내용은 주로 열차(train) 제어와 운용(operation)에 초점을 두고 있다.

철도 시스템의 안전성 활동에 대한 IEC 62278[11], 철도 제어 시스템의 소프트웨어 개발에 관한 IEC 62279 [12], 철도 시스템의 통신 및 신호처리에 관한 IEC 62280[13]과 IEC 62425[5], 전기전자 안전성 관련 시스템의 기능적 안전성을 위한 IEC 61508[7] 등이 있다.

현재의 철도 제어 시스템은 컴퓨터 혹은 임베디드 시스템이 주요 요소로 사용되고 있다.

컴퓨터 시스템 관련 표준은 주로 소프트웨어 시스템에 초점을 맞추어서 개발주기 활동을 명시하고 각 개발 단계에서의 프로세스와 산출물(deliverables) 들에 대해 기술하고 있다. IEEE 828-1998은 소프트웨어 형상관리 계획(configuration management plan)에 관해, IEEE 12207-1997[14]은 소프트웨어 개발 주기(development life cycle)에 관해 기술하고 있다. IEEE 830-1998[8]은 소프트웨어 요구사항 명세에 대한 지침을, IEEE 1233-1998[15]은 컴퓨터 시스템 요구사항 명세에 대한 지침을 서술하고 있다.

관련된 표준들을 표 1에서 비교하였고, 크게 2가지 유형으로 분류하였다. 절차지향적 표준은 주기 활동에서 각 단계의 처리활동 내용을 기술하고, 산출물 지향적 표준은 처리활동보다는 각 단계에서 생성해야 할 결과물 내용을 기술한다.

표 1 철도 제어 시스템 관련 국제 표준과 규격의 비교

표준 / 특성	IEC 61508-1998	IEC 62278	IEC 62279	IEC 62280	IEC 62425	IEEE 12207-1996	IEEE 1233-1998	IEEE 830-1998
영역	전기/전자	철도	철도	철도	철도	컴퓨터	전기/전자/컴퓨터	컴퓨터
종류	표준	표준	표준	표준	표준	표준	개발 가이드	개발 가이드
내용	전기/전자/임베디드 안전성 관련 시스템의 기능적 안전성	철도 시스템의 안전성 활동	철도 시스템의 소프트웨어 개발	철도 시스템의 통신 및 신호 처리 -안전 통신	철도 시스템의 통신 및 신호 처리 -안전 시스템	정보통신 컴퓨터와 소프트웨어의 개발 주기 프로세스	시스템 요구사항 명세	소프트웨어 요구 사항 명세
유형	절차지향적	절차지향적	절차지향적	절차지향적	절차지향적	절차지향적	산출물지향적	산출물지향적

2.2 정형 기법

철도 제어 시스템을 위해 사용된 정형기법 언어와 도구는 대표적으로 B-메소드[16], Z 명세[17], Petri-Net [18], Statechart[9], SCADE[19], SPIN [20]등이 있다. 기존 연구에서는 정형기법을 이용하여 시스템을 설계하고 모델링하는 방법론과 정형명세한 시스템 모델의 안전성 검증에 초점을 맞추고 있다.

본 논문에서는 요구사항의 작성을 위해 정형 기법을 사용하고, 요구사항의 정형 명세 도구로써 Z 명세와 상태차트를 사용한다.

상태 차트는 상태기반(state-based) 시각적 명세 언어로서 시스템의 정적인 측면인 시스템 구성과 동적인 측면인 시스템의 기능과 행위를 모델링하고 도식적으로 표현할 수 있다[21]. 시스템 행위를 하드웨어 특성적인 이벤트 주도(event-driven)형과 환경에 대응하는 반응형 시스템(reactive system), 소프트웨어 특성적인 프로시져(procedure-like)형을 모두 지원하기 때문에 주로 임베디드 시스템의 설계와 구현에 널리 이용되고 있다.

Z 명세 언어는 일차 술어 논리(first-order logic)와 집합론(set theory)의 개념과 표기법을 사용하며, 시스템의 데이타를 정의하고 시스템의 작업(operation)을 명세한다[10]. 작업은 선조건(pre-condition)과 후조건(post-condition)을 사용하는 스키마(schema)로 표현한다.

2.3 요구사항 공학(requirements engineering)

소프트웨어 공학의 소프트웨어 개발 주기의 요구사항 분석 단계에서 하나의 방법론으로 제시된 체계적이고 반복적인 요구사항 분석 방법으로서 전통적인 시스템 공학의 요구사항 분석 단계에도 적용되고 있다[22,23]. 요구사항 공학의 목적은 시스템의 사용과 개발에 관련된 참여구성원의 요구사항을 합의를 통해 충실히 반영하기 위함이다. 요구사항 공학에서 사용하는 시스템에 대한 정의와 범위에 따라, 소프트웨어 시스템인 경우[24]와 소프트웨어 시스템과 하드웨어 시스템 모두 포함하는 경우[22,25]로 나누어 연구되었다.

Hull이 제안한 반복적 요구사항 분석기법은 초기 요구사항을 분석하고 모델링을 사용하여 초기 요구사항으로부터 파생된 요구사항을 생성하고, 이것을 요구사항 변경 절차에 반영시키고, 다시 요구사항 분석을 반복적으로 시행하는 절차를 가진다. 요구사항의 정확성을 판단하기 위해 완전성과 일치성을 평가 기준으로 삼고 있다.

본 논문에서는 요구사항의 정확성 판단 기준으로 Hull이 제시한 평가기준인 완전성과 일치성을 채택하고, 실제 정형명세한 철도시스템의 요구사항의 정확성을 분석한다.

3. 철도 시스템을 위한 정형 요구사항 명세 지침서

국제 규격의 컴퓨터 기반의 철도 제어 시스템을 개발하기 위해, 본 논문에서 제안한 철도 시스템의 요구사항의 정형 명세 지침서는 다음의 사항을 고려하였다. 철도 제어 컴퓨터 부분을 개발하기 위해 컴퓨터 개발 프로세스 표준을 준수해야 하고, 국제 철도 규격의 인증을 받기 위해 철도 시스템 관련 표준을 따라야 하고, 실제 개발자가 활용할 수 있어야 한다.

컴퓨터 개발 프로세스 표준은 IEEE 1233-1998을 기본으로 삼았다. 국제 철도 인증 규격을 따르기 위한 표준으로 IEC 62278의 안전성 활동의 안전성 보고서(safety case)를 사용한다는 것을 가정으로 전제하고, IEC 62279의 시스템 개발 주기를 채택하였다. 실제 개발자가 활용하도록 하기 위해 정형명세를 해석한 자연어 문장을 정형명세와 함께 기술하는 방식을 사용하였다.

3.1 정형 요구사항 명세 지침서

그림 1은 컴퓨터 기반 철도 제어 시스템을 개발하기 위한 지침서의 목차를 보여주고 있다. 지침서의 내용을 살펴보면 다음과 같다. 1장에서는 개발할 시스템의 목적과 범위 그리고 개요에 대한 설명과 요구사항 문서를 읽고 이해하는데 필요하도록 문서 안에서 사용된 용어나 약어들을 정리한다.

2장은 시스템의 기능적인 설명을 주로 다루고 있다. 시스템의 구성요소와 주변 환경 사이의 관계, 시스템이 가지는 기능과 기능에 필요한 조건이나 제약 사항들, 시스템을 개발하거나 시스템의 기능적인 측면에서 반영해야 하는 가정이나 현실적인 제약 사항들을 명시한다. 시스템을 사용하는 사용자의 특성과 시스템이 사용되는 대표적인 시나리오를 서술한다.

3장은 시스템의 기능적인 측면보다는 성능이나 개발 관리 측면에서의 설명을 주로 다룬다. 시스템이 지녀야 할 물리적인 특성과 달성해야 할 구체적인 수량적 성능의 특징을 서술한다. 시스템의 개발 기간에 필요한 보안 사항이나 정보관리를 명시한다. 시스템을 개발할 때 준수해야 할 정부의 정책이나 규제를 명시하고, 시스템 개발 주기 단계를 명시하여 시스템이 개발 주기에 따라 개발되고 있는지를 알려주도록 한다. 시스템이 개발되어 실제로 운영될 때 필요한 사항도 함께 서술한다.

4장에서는 시스템이 외부 주변 환경과 상호작용할 때 주고 받는 데이타의 흐름을 서술한다. 5장은 명세문서를 기술하는데 보완할 사항이 있으면 추가로 서술할 수 있다.

6장은 시스템의 안전성과 관련된 요구사항을 기술한다. 먼저 시스템이 추구하는 안전성의 목표와 목적을 기술한다. 6.2절의 총괄적 안전성 요구사항에서, 안전성 보고서(safety case)에 포함된 위험원 분석(hazard analysis)과 위험성 분석(risk analysis) 결과에 따라, 각 위험원(hazard)마다 수용할 위험성(risk)을 결정하고 안전성 기능들을 나열한다. 각각의 안전성 기능마다, 안전성

그림 1 정형 명세 지침서 목차

기능 단위로 할당된 시스템 내부의 하위 시스템을 나열한다. 할당된 하위 시스템들 중에서 E/E/PES(전기/전자/임베디드 시스템)에 대해 안전성 기능에 관한 요구사항과 안전성 무결성에 관한 요구사항을 각각 6.3 절과 6.4 절에 나누어 서술한다.

4. 정형 명세 기법의 적용

4.1 철도 제어 시스템

KRRI에서 개발중인 안전한 열차 제어 시스템의 진로 제어 분야의 모의 열차 운행(mock-up)시스템은 연동 기능과 열차 간격 제어를 위한 CRD(control route distance) 시스템과 자동 열차 감시 장치인 ATS(automatic train supervision) 시뮬레이터, 실제 모의 열차 운행을 위해 차상 장치 기능을 가상으로 구현하는 차상 시뮬레이터(ATP automatic train protection)와 현장 신호 설비 기능을 구현하기 위한 신호 설비 제어 시뮬레이터 그리고 신호설비(선로전환기) 시뮬레이터로 구성된다. 모의 열차 운행 시스템의 개략적인 구성도는 그림 2와 같다. 요구 사항 명세를 적용할 대상 시스템은 CRD 시스템의 일부분인 DCM(distance control module)이다.

DCM은 5가지 기능을 수행한다: 열차의 간격 제어, 열차의 위치 확인, 열차 이동과 방향의 감시, 열차의 임시 속도 제어 명령의 처리, 블록의 개방과 폐쇄. 열차 간격 제어 기능은 DCM이 열차 진로의 각 블록에 3가지 허용이동권한 (PMA, permission of movement authority)중 한가지를 설정하여 ATP 에게 전송하고, ATP 로 하여금 PMA 에 따라 열차 이동을 제어함으로써 열차 사이의 안전거리를 확보한다. 열차 위치 확인 기능은 열차가 블록에 진입할 때 ATP 가 자신의 현재 블록을 DCM 에게 전송하고, DCM 은 해당열차의 점유 블록 정보를 ATS 에게 전송하는 작업 순서로 이루어진다. 열차 이동과 방향의 감시 기능은 DCM이 ATP로부터 수신한 현재의 열차의 위치 정보와 기존에 가지고 있던 열차의 위치 정보와 비교하여 열차의 정상 운행을 확인한다. 열차의 임시 속도 제어 기능은 ATS 가 특정 블록에 대한 임시속도 제한 명령을 DCM에게 전송하면, DCM 은 정보의 유효성을 판단하여 해당 블록에 임시 속도 제한을 설정하여 ATP 에게 전송하는 것으로 이루어진다. 블록의 개방과 폐쇄는 2가지 경우로 나누어 수행한다. 첫째 ATS가 특정 블록의 폐쇄 명령을 DCM에게 전송하는 경우, DCM은 해당 블록의 PMA를 진입금지 의미인 적색(red)으로 설정하여 관련 열차의 PMA를 설정하여 전송함으로써 블록을 폐쇄시킨다. 둘째 열차의 위치가 파악이 되지 않는 비상상황의 경우, DCM은 분실된 열차의 진로 상에 있는 관련 블록들의 PMA를 적색으로 설정하여 블록을 폐쇄시킨다. 블록 해제는 오직 ATS 의 블록 해제 명령만으로 이루어지며, DCM은 해제 명령을 수신하여 해당 블록의 PMA 설정을 변경시킴으로써 실제 블록이 해제된다.

4.2 철도 제어 시스템의 정형 명세

본 논문에서는 요구사항을 파생시키기 위해 구조적

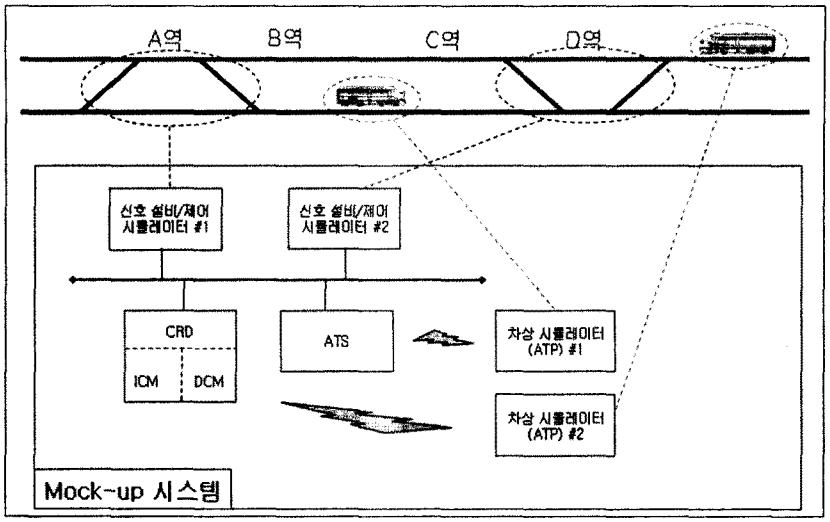

그림 2 열차 제어 시스템 구성도

분석 방법(structured analysis)에 속하는 기능 기반 시스템 구성 기법(function-based decomposition)을 사용하였다[9]. 상태 차트와 Z 명세 2가지 정형 기법을 사용하는 이유는 모델링의 다양한 관점을 확보하기 위해서이다. 상태차트는 시스템 행위에, Z는 시스템의 기능에 주로 중점을 두고 시스템을 분석한다.

4.2.1 상태 차트

기본적인 가정은 시스템의 신호처리와 통신은 무결성 정보가 보장되는 환경으로 가정하고, DCM의 기능을 개념적으로 모델링하였다.

필요한 기능을 순차적으로 또는 조건에 따른 실행으로 시스템을 기술하였다. DCM의 5가지 기능을 정형 명세한 결과는 다음과 같다.

그림 3에서 열차 간격 제어 상태 차트를 보여주고 있다. 열차 간격 제어 기능은 프로시져(procedure) 기반의 행위로 모델링하였고, PMA 설정 과정은 상세 설계 부분에서 구현될 알고리즘으로 구현될 부분으로 남겨두고 PMA 설정에 필요한 입력과 출력 정보를 기술하였다.

그림 4는 열차 위치 확인 상태 차트를 나타내고 있다. 먼저 ATP와 통신 연결이 이루어진 상태와 통신 연결이 끊어진 상태로 나누고, 통신 연결이 이루어진 상태에서 ATP로부터 열차 위치 정보를 수신하여 확인하는 행위를 모델링하였다. 통신 연결이 끊어진 상태는 열차 분실 상황에 해당하는데, 이에 대한 명확한 요구사항이 기술되어 있지 않았고, 통신 두절상태에서 통신 복구가 되는 과정이나 절차에 대한 명확한 요구사항도 서술되어 있지 않아서 반드시 마련되어져야 할 사항으로 기술하였다.

그림 5는 열차 이동과 방향 감시 기능을 나타내는 상태 차트이다 열차 이동과 방향 감시 기능은 반응 시스템(reactive system)의 행위처럼 모델링해서 주기적으로 ATP로부터 수신한 열차의 블록 정보를 기존의 열차

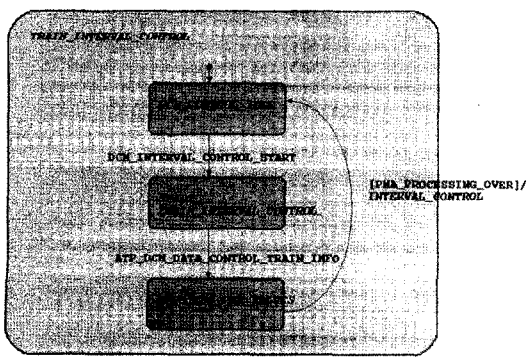

그림 3 열차 간격 제어 상태 차트

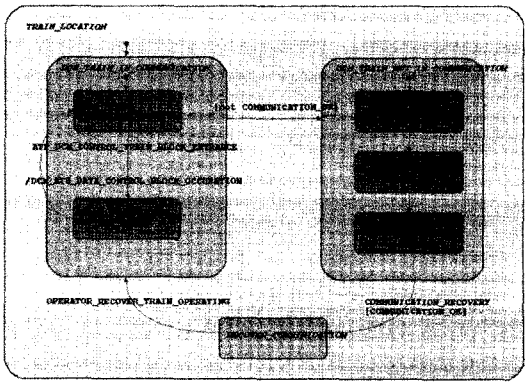

그림 4 열차 위치 확인 상태 차트

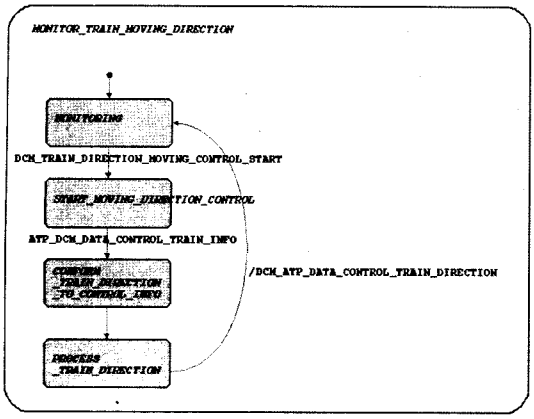

그림 5 열차 이동과 방향 감시 상태 차트

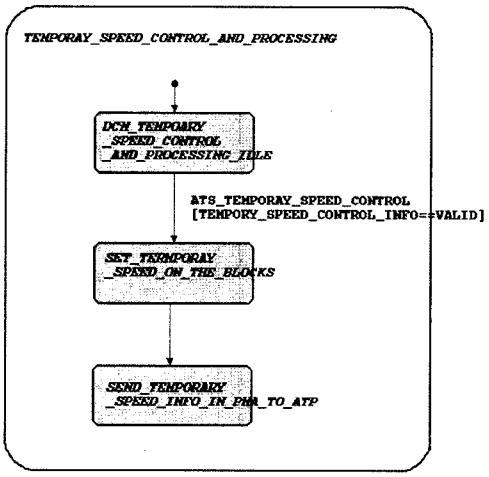

그림 6 임시 속도 제한 상태 차트

이동 방향 정보와 비교하여 감시하는 행위를 모델링하였다.

그림 6은 임시 속도 제한 기능을 나타내고 있다. ATS로부터 임시 속도 제한 명령을 수신받아 명령의 유효성을 검사하여 해당 블록의 임시 속도 제한을 설정하고 ATP 에게 해당 블록 정보를 PMA와 함께 전송한다. 유효하지 않은 명령인 경우에 DCM이 처리해야 할 행위가 요구사항에 명시되어 있지 않아서 상태 전이를 기술하지 않았는데 반드시 기술이 필요한 사항이다.

그림 7은 블록 개방과 폐쇄 기능을 나타내는 상태차트이다. 왼쪽 하위 상태 차트는 블록 폐쇄를, 오른쪽 하위 상태 차트는 블록 개방을 기술하고 있다. 블록 폐쇄는 ATS의 명령이나 비상상황의 경우에 이루어지는 것으로, 프로시져 형태의 행위로 모델링하였다. 블록 개방은 오직 ATS의 블록 개방 명령을 수신했을 때에만 수행하며 관련된 블록들의 PMA 설정을 변경시키는 것으로 블록 개방 행위를' 모델링하였다.

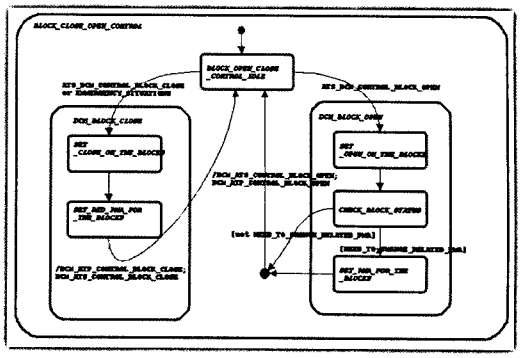

그림 7 블록 개방과 폐쇄 상태 차트

4.2.2 Z 명세

Z 명세를 사용하여 DCM 기능의 선조건과 기능 수행에 필요한 상태 변수, 기능의 후조건들을 기술하였다. DCM 기능의 Z 명세는 아래의 그림과 같다.

그림 8에서는 DCM이 처리해야 하는 정보를 기술하고 있다:

- 현재 열차의 점유 블록
- 열차의 운행 방향
- 모든 블록의 PMA 설정 상태
- 모든 블록의 임시 속도 설정 유무와 내정된 임시 속도
- 장애물로 설정된 블록
- DCM과 ATP 통신 연결 상태

DCM의 열차 간격 제어는 그림 9에서 기술하고 있고, DCM의 PMA 설정과 ATP로의 PMA 전송으로 달성된다.

그림 10에서 열차 위치 확인은 DCM이 ATP로부터 열차 정보를 받아서 ATS에게 열차 점유 블록 정보를 전송하여 이루어진다. 열차 방향 확인은 DCM이 ATP

DCM

$current : Block$
$direction : Direction$
$pma : Block \rightarrow PMA$
$speed : Block \nrightarrow Speed$
$obstacle : \mathbb{P}\, Block$
$redBlock : \mathbb{P}\, Block$
$yellowBlock : \mathbb{P}\, Block$
$greenBlock : \mathbb{P}\, Block$
$connection : ConnectionStatus$

$(redBlock, yellowBlock, greenBlock)$ partition $Block$
$behindOf(redBlock, direction) \cap greenBlock = \emptyset$
$\forall b : redBlock \bullet pma(b) = Red$
$\forall b : yellowBlock \bullet pma(b) = Yellow$
$\forall b : greenBlock \bullet pma(b) = Green$

그림 8 DCM 시스템 명세

sendPMA

ΞDCM
$b0, b1, b2, b3 : Block$
$pma0!, pma1!, pma2!, pma3! : PMA$

$b0 = current$
$b1 = next(b0, direction)$
$b2 = next(b1, direction)$
$b3 = next(b2, direction)$
$pma0! = pma(b0)$
$pma1! = pma(b1)$
$pma2! = pma(b2)$
$pma3! = pma(b3)$

그림 9 DCM의 PMA 전송

receiveLocation

ΔDCM
$locationFromATP? : Block$

$connection = Connected$
$current' = locationFromATP?$

receiveDirection

ΔDCM
$dir? : Direction$

$direction' = dir?$

그림 10 열차 위치와 방향 확인

sendSpeed

ΞDCM
$temporarySpeed! : Speed$

$current \in \mathrm{dom}\ speed$
$temporarySpeed! = speed(current)$

sendNothing

ΞDCM

$current \notin \mathrm{dom}\ speed$

$$TSendSpeed \mathrel{\widehat=} sendSpeed \lor sendNothing$$

그림 11 임시 속도 제한 명령 처리

receiveLocation

ΔDCM
$locationFromATP? : Block$

$connection = Connected$
$current' = locationFromATP?$

그림 12 블록 개방과 폐쇄

로부터 운행 방향 정보를 수신하여 갱신한다.

그림 11에서 임시 속도 제한은 현재 블록에 임시 속도 제한이 설정되어 있는 경우에만 임시 속도를 전송하고, 설정되어 있지 않은 경우에는 전송하지 않는다.

블록 개방과 폐쇄는 그림 12에서 기술하고 있으며, 블록을 폐쇄할 경우 해당 블록을 장애물로 처리하여 장애물에 추가하고, 블록을 개방할 경우 해당 블록을 장애물에서 삭제한다.

5. 분 석

정형 명세는 자연어가 가지는 애매모호성을 없애는 효과가 있으며, 정형 명세를 통해 자연어 요구사항이 가지는 모순성의 존재 여부와 완전성의 여부를 확인하게 해준다[2]. 본 논문에서는 정형 명세를 통해 발견된 요구사항 단계에서의 오류를 요구사항 판단 기준인 정확성(correctness)의 2가지 요소인 완전성(completeness)과 일치성(consistency) 측면에서 기술하고, 오류를 수정한 보완된 요구사항을 제시한다.

5.1 완전성(completeness)

예제 1. 기능의 불충분한 동작 조건. 그림 13은 열차 이동 방향 감시 기능에 관한 정형 명세로서, 원래 요구사항에서 기능의 시작조건이 기술되어 있지 않음을 보여주고 있다.

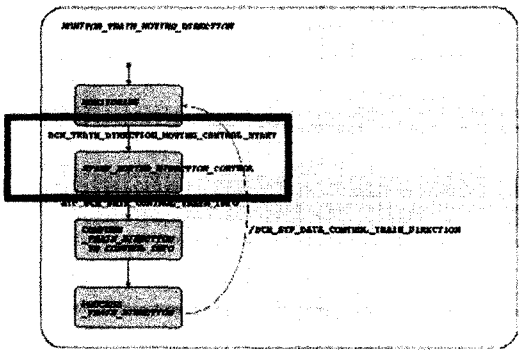

그림 13 기능 명세의 불완전성 예제

DCM의 열차 이동 방향 감시 기능에 관한 요구사항 (초기)
• DCM은 ATP로부터 열차의 진행 방향 정보를 전송받는다.
• DCM은 ATP로부터 수신한 열차 위치 정보와 기존에 DCM이 가지고 있던 해당 열차의 열차 위치 정보와 비교하여, 해당 열차의 진행방향이 정상적인지 판별한다.

초기의 자연어 명세에서는 열차 이동 방향 감시 기능의 시작 조건과 종료 조건이 명확하게 기술되어 있지 않다. 이것을 해결하기 위해 "dcm_train_direction_moving_control_start"라는 신호(signal)을 보완하여 기능의 시작을 명시하였고, 그림 13에 표시하였다.

이외에도, DCM 기능 중에서 임시 속도 제한 명령 처

리와 열차 간격 제어 기능이 초기 상태와 시작 조건이 초기 자연어 요구사항에는 명시되어 있지 않았다.

예제 2. 기능 사이의 불분명한 연관성. 임시 속도 제한 명령 처리에 관한 정형 명세에서 나타난다.

임시 속도 제한 명령 처리 (초기)
• Dcm 은 열차가 특정 구간을 통과할 때 특정 블록에 임시 속도 제한을 설정할 수 있다. • 임시 속도 제한의 대상은 특정 블록 혹은 모든 블록에 설정할 수 있다. • Dcm 은 ATS 로부터 특정 블록에 대한 임시 속도 제한 명령을 받았을 때만 기능을 수행한다. • Dcm 은 정보의 유효성을 판단하여, 유효한 경우에만 임시 속도 제한을 설정하며 해당 블록의 임시속도 제한 정보를 ATP 에게 전송한다.

초기 자연어 명세에서는 하위 처리 기능의 순서나 연관 관계, 선후 관계가 표현되어 있지 않은데, 반드시 기술되어야 한다. 그림 6의 정형 명세에서는 처리 순서를 정하여 표현하고 있다.

예제 3. 기능 수행에 필요한 변수와 조건. Z 명세에서는 기능 수행에 필요한 변수와 조건을 순서대로 기술해야 하며, 타입(type)이 필요없는 변수와 타입이 명시되어야 하는 변수가 생긴다. 그림 14에서 블록 정보와 속도 정보는 시스템이 실제 사용할 타입 혹은 가지게 될 정보이고 요구사항 단계가 아닌 설계나 구현 단계에 속하기 때문에 임의의 타입인 block 과 speed 로 표현하였다.

[*Block*]

[*Speed*]

그림 14 미확정 타입의 변수

PMA ::= Green | Yellow | Red

Direction ::= Upward | Downward

ConnectionStatus ::= Connected | Disconnected

그림 15 확정 타입의 변수

그림 15에서는 PMA 설정, 열차 이동 방향, 통신 연결 상태의 값은 구체적으로 표현하였다. 시스템이 사용하는 변수에 대한 개념이나 특성에 대해 구체적으로 명시해야 한다.

예제 4. 기능 수행을 위한 상태 기술의 불완전성. 기능을 수행하기 위한 시스템의 조건과 상태를 명시해야 한다.

DCM의 열차 간격 제어 기능에 대한 요구사항에서, DCM이 ATP에게 전방 4블록의 PMA를 보내야 한다고 기술되어 있지만, 열차 진로상 전방에 남아있는 블록의 개수가 4개가 되지 않는 경우에도 대한 시스템의 동작이나 행위가 명시되어 있지 않다. 열차 사이의 충돌을 방지하기 위해 열차 사이의 간격 제어를 하는 것이고, 열차 간격 제어를 위해 PMA를 설정하는 것인데, PMA 설정을 위한 모든 경우를 고려해보면 초기 요구사항의 PMA 설정 조건의 기술이 불충분하고 기능의 목적과 충돌이 생기는 것을 알 수 있다. 그림 9에서 열차의 현재 진행 방향으로 다음 4개 블록이 존재하는 상황만을 가정했는데, 단순한 진행 방향의 다음 블록의 고정된 개수가 아니라 속도 제어 측면에서의 다음 진행 방향의 블록의 개수를 가변적으로 설정할 수 있어야 한다.

5.2 일치성(consistency)

예제 1. 기능의 상호 연관성 무시. 자연어의 요구사항에서는 별개의 기능으로 분류되었던 기능들이, 정형 명세에 기술된 기능들 사이의 상호 운영성(interoperability)을 고려하면 포함관계나 하위 기능에 대한 공유관계 같은 기능 사이에서 일종의 계층적 구조를 파생시킬 수 있다.

그림 6과 7을 보면, 임시속도 제한 명령 처리 기능과 블록 개방 폐쇄기능의 경우, 주기적으로 발생하는 것이 아니라, 외부(ATS)로부터 DCM이 기능관련 요청이나 명령을 수신했을 때 수행하는 것이다.

그림 3, 4, 5를 보면, 열차 간격 제어, 열차 위치확인과 열차 이동방향 감시 기능은, 주기적으로 수행해야 하는 행위이다. 3가지 기능은 공통적으로 ATP로부터 전송받은 열차 정보(ATP_to_dcm_data_control_train_info)를 가지고 기능을 시작하고 있다. 이것은 3가지 기능이 병렬적으로 진행될 수 있고, 반드시 동일한 열차 정보에 대해 작업이 이루어져야 하므로 동기화될 필요가 있다는 점을 알 수 있다.

6. 결 론

본 논문에서는 컴퓨터 기반의 안전 필수 철도 제어 시스템을 개발하기 위한 요구사항을 정형 명세하는데 알맞은 요구사항 지침서를 제시하고, 실제로 KRRI에서 개발중인 철도 제어 시스템을 대상으로 직접 적용해 본 결과를 소개하고 있다. 정형 명세를 통해 기술된 요구사항의 완전성과 일치성을 분석하였고, 개선 사항을 제시하였다.

본 논문에서 제시한 정형 명세를 위한 요구사항 지침서는 2가지 측면에서 유용하다. 형식적으로는 국제 표준에 근거하고 있기 때문에 향후 제품의 국제 규격에 해당하는 인증 절차에 부합되고, 높은 수준의 품질 등급을 획득하는데 도움이 된다. 방법론적인 면에서는, 실제 요구사항 작성을 위한 정형 명세 도구로서 상태차트(statechart)와 Z 명세기법을 사용하는 예제를 제시하였는데, 자연어와 정형 명세 2가지 형태로 요구사항을 작성

하는 방법이 개발할 대상 시스템을 보다 정확하고 깊이 이해할 수 있게 해줌으로써, 보다 오류를 줄이는데 기여한다.

향후 연구과제로서 정형 기법을 보다 쉽게 사용할 수 있도록 정형 명세용 상위 수준의 스크립트 언어 도구를 개발하는 주제를 계획하고 있다.

그리고, 본 논문에서는 기능적인 요구사항만을 다루었는데, 향후 연구과제로서 안전성 요구사항도 정형명세를 적용하는 연구를 진행중에 있다.

참 고 문 헌

[1] Bowen, J.: Formal methods in safety-critical standards, Proc. 1993 Software Engineering Standards Symposium, IEEE, pp. 168-177, 1993.

[2] Clarke, E.M. & Wing, J.J., Formal Methods: State of the Art and Future Directions, ACM Computing Surveys, 1996.

[3] Monin, J-F, Understanding Formal Methods, Springer-Verlag, 2003.

[4] Zowghi, D. & Gervasi, V.: The three Cs of Requirements: Consistency, completeness, and Correctness, Proc. Of 8th International Requirements Engineering: foundation for software quality, 2002.

[5] IEC Std. 62425, "Railway applications-Communication, signaling and processing systems - safety-related electronic systems for signaling," 2005.

[6] IEEE Std 1474.1-2004, "Standard for Communications-Based Train Control Performance and Functional Requirements".

[7] IEC Std. 61508, "Functional safety of electrical/electronic/programmable electronic safety-related systems," 1998.

[8] IEEE Std. 830, "IEEE Recommended Practice for Software Requirements Specifications," 1998.

[9] Harel, D. & Politi, M.: Modeling Reactive Systems with Statescharts, 1998, McGraw-Hill.

[10] Potter, B., Sinclair, J., Till, D.: Introduction to Formal Specification and Z, 1996, Prentice-Hall.

[11] IEC Std. 62278, "Railway applications-Specification and demonstration of reliability," availability, maintainability and safety(RAMS), 2002.

[12] IEC Std. 62279, "Railway applications-Software for railway control and protection systems," 2003.

[13] IEC Std. 62280, "Railway applications-Communication, signaling and processing systems," 2002.

[14] IEEE Std 12207, "Standard for Information Technology - Software life cycle processes," 1996.

[15] IEEE Std. 1233, "IEEE Guide for Developing System Requirements Specifications," 1998.

[16] Lecomte, T., Servat, T., & Pouzancre, G.: Formal Methods in safety-critical railway system, Proc. Of Brazilian Symposium on formal methods (SBFM) 2007, Outo-Preto, Brazil.

[17] Janota, A.: Using Z specification for railway interlocking safety, Periodica Polytechnica Ser. Transportation Engineering, vol.28, no.1-2, pp. 39-53, 2000.

[18] Horste, M.M.: Modelling and simulation of train control systems using Petri nets, FM 1999, LNCS 1709, pp.720, Springer-Verlag, 1999.

[19] Abdulla, P.A., Deneux, J., Stalmarck, G., Argen, H., & Akerlund O.: Designing Safe, Reliable systems using SCADE, ISOLA 2004, LNCS 4313, pp. 115-129, 2006.

[20] Cimatti A., et al: Model Checking safety critical software with SPIN: an Application to a Railway Interlocking System, SAFECOMP, LNCS 1516, pp. 284-293, Springer-Verlag 1998.

[21] Harel, D.: On Visual Formalisms, CACM(31): No.5, pp. 514-530, ACM, 1988.

[22] Hull, E., Jackson, K., & Dick, J.: Requirements Engineering, 2nd edition, 2005, Springer.

[23] Grady, J.: System Requirements Analysis, 2006, Elsevier.

[24] Hartley, D., Hruschka, P,& Pirbhai, I.: Process for System Architecture and Requirements Engineering, 2000, Dorset House.

[25] Bjorner, D.: Software Engineering Vol.3, 2006, Springer.

이 진 호

연세대학교 전산학과 학사. 고려대학교 컴퓨터학과 석사. 2004년~현재 동대학원 박사 과정. 관심분야는 정형기법, 안전필수 시스템, 소프트웨어 공학, 정보보호

황 대 연

고려대학교 컴퓨터학과 학사. 동대학원 석사 졸업. 2005년~현재 동대학원 박사 과정. 관심분야는 정형기법, 수리 논리, 내장형 시스템, 소프트웨어 공학

김 진 현

한국외국어대학교 컴퓨터공학과 학사. 고려대학교 컴퓨터학과 석사 졸업. 2001년~현재 동대학원 박사과정. 관심분야는 정형기법, 실시간 시스템, 내장형 시스템, 소프트웨어 공학

박 준 길

고려대학교 컴퓨터학과 학사. 2005년~ 동대학원 석박사 통합과정. 관심분야는 정형기법, 소프트웨어 자동 검증

최 진 영

서울대학교 컴퓨터공학과 학사. Dept.of Mathmatics & Computer Science, Derexel University 석사. Dept. of Computer & Information Science, University of Pennsylvenia 박사. 1996~현재 고려대학교 컴퓨터학과 교수. 관심분야는 계산 이론, 수리 논리, 실시간 컴퓨팅, 정형기법, 프로그래밍 언어, 프로세스 대수, 소프트웨어 공학, 프로토콜 공학

황 종 규

건국대학교 전기공학과 학사. 동대학원 석사. 한양대학교 전자통신전파공학과 박사. 1995년~현재 한국철도 기술 연구원 열차제어 연구팀 선임연구원

윤 용 기

충북대학교 전기공학과 학사. 동대학원 석사. 2005년~현재 한양대학교 전자전기제어계측공학과 박사과정. 1995년~현재 한국철도 기술 연구원 열차제어 연구팀 선임연구원

조 현 정

한국항공대학교 항공전자공학과 학사. 광주과학기술원 정보통신공학과 석사. 2005년~현재 한국 철도 기술연구원 열차제어연구팀 선임연구원