Z와 Statechart에 의한 열차제어시스템 바이탈 소프트웨어 개발 방법 분석

論 文
57P-2-1

Applying Methodology for the Safety-Critical S/W Development of Railway Signaling with the Z and Statechart Formal Method

趙 賢 庭* · 黃 宗 奎† · 尹 用 基**
(Hyun-Jeong Jo · Jong-Gyu Hwang · Yong-Ki Yoon)

Abstract - Recently, many critical control systems are developed using formal methods. When software applied to such systems is developed, the employment of formal methods in the software requirements specification and verification will provide increased assurance for such applications. Earlier error of overlooked requirement specification can be detected using formal specification method. Also the testing and full verification to examine all reachable states using model checking to undertake formal verification are able to be completed. In this paper, we propose an eclectic approach to incorporate Z(Zed) formal language and 'Statemate MAGNUM' which is formal method tools using Statechart for applying to the railway signaling systems.

Key Words : 열차제어시스템, 정형명세, 정형검증, Zed, Statechart

1. 서 론

최근 들어 컴퓨터 및 통신 기술의 발달에 따라, 철도시스템에서 기존의 기계 및 전기식 열차제어시스템(Train Control Systems)들이 전자식으로 변경되어가고 있다. 이처럼 열차제어시스템들이 컴퓨터화 되어감에 따라 시스템의 안전성을 확보하기가 매우 어려워지게 되었다. 특히, 열차제어시스템은 열차의 안전운행을 책임지는 매우 바이탈한 장치들로서 높은 안전성이 요구되어진다. 이러한 문제의 해결을 위한 방안으로 관련 국제규격에서는 열차제어시스템의 개발사양 명세화 및 설계에 정형기법(Formal Method)의 적용을 권고하고 있으며 [1][2], 철도선진국, 특히 유럽을 중심으로 정형기법을 적용하기 위한 연구 및 노력이 진행되고 있다 [3]-[6].

이 중 프랑스 파리 지하철 14호선의 열차제어시스템 정형명세(Formal Specification) 적용사례는 실제 상용시스템에 정형기법을 적용한 최초의 사례로 소개되고 있다 [3]. [3]에서의 B method 뿐만이 아니라 Z(Zed), VDM(Vienna Development Method), Statechart 등과 같은 여러 정형기법 방법들을 열차제어시스템에 적용하기 위한 연구가 이미 진행되었거나 현재 진행 중에 있으나, 실제 철도 현장의 개발에 활용하기엔 아직 미비한 실정이다. 왜냐하면, 현재 연구결과로 제시되고 있는 정형기법 적용방안은 지나치게 복잡한 수학적 수식이나 이론에 바탕을 두고 있어서, 그 분야의 전문가가 필요하다는 것과 개발 시간이 매우 길게 요구되는 등의 문제점들을 가지고 있기 때문이다.

더군다나 국내에서는 열차제어시스템의 개발에 정형기법을 적용하기 위한 연구단계는 선진국에 비해 더욱 많이 뒤처진 상황이며, 철도신호용 프로토콜의 설계에 일부 적용한 사례가 있지만 [7], 열차제어시스템에 적용한 사례는 아직 없는 실정이다. 현재 국내에서는 정형기법이 소프트웨어의 안정성 확보를 위해 반드시 필요한 방법이라는 것을 연구차원에서는 이미 인식하고 있지만, 산업계에서는 적용 방안이 실제로 거론되지 않고 있어 철도분야에 적용할 수 있는 현실성 있는 구체적인 절차 및 방법에 대한 연구를 필요로 한다. 따라서 본 논문에서는 국내 처음으로 철도분야의 열차제어시스템에 실제로 정형기법을 적용하는 방안 및 절차를 제시하였다.

이를 위해 정형기법 적용의 단점 중 하나인 지나친 수학적 수식에 의존하기 보다는 그래픽 정형명세 언어 중의 하나인 Statechart를 사용하면서, 반드시 필요한 부분만 수학적 형식을 기반으로 하는 Z를 적용하는 접근법을 제안하였다. 본 논문의 순서는 다음과 같다. 2장에서 정형기법을 구성하는 정형명세와 정형검증(Formal Verification)에 대한 설명 및 필요성을 서술한 후에, 3장에서 우리가 제안하고자 하는 Z와 Statechart의 절충안을 이용한 국내 열차제어시스템 적용 방안을 구체적으로 제시할 것이다. 마지막으로 제안하는 방식을 이용한 정형명세의 결과를 보여준 후에 4장에서 결론을 맺고자 한다.

2. 정형명세와 정형검증

열차제어시스템은 열차의 속도제어 및 진로제어 등을 담

* 正 會 員 : 韓國鐵道技術研究院 主任研究員
** 正 會 員 : 韓國鐵道技術研究院 先任研究員 · 工博
† 교신저자, 正會員 : 韓國鐵道技術研究院 先任研究員 · 工博
E-mail: jghwang@krri.re.kr
接受日字 : 2008年 2月 25日
最終完了 : 2008年 4月 3日

당하며, 특히 열차의 충추돌 방지의 기능을 담당하여 최종적으로 안전운행을 책임지는 바이탈 시스템이다. 최근 들어 전자, 컴퓨터, 통신 기술이 발달하면서 열차제어시스템에 쓰이는 열차제어장치들도 과거의 기계식, 전기식에서 전자식으로 바뀌어 가고 있다. 이에 따라 열차제어장치들의 논리회로의 구성이 종래의 계전기를 이용한 것에서 마이크로프로세서를 이용한 것으로 이동되어지고 있다. 마이크로프로세서를 이용한 구성에는 하드웨어에 의한 안전성, 신뢰성의 향상방법이 일부에서는 적용 가능하지만, 많은 경우는 이것을 소프트웨어에 의존하도록 되어있다. 최근의 컴퓨터화된 열차제어시스템의 사용이 증가함에 따라서 소프트웨어의 복잡성으로 인한 오류로부터 장치들의 고장이 유발되어 대규모 인명피해나 경제적 손실이 발생할 수 있다.

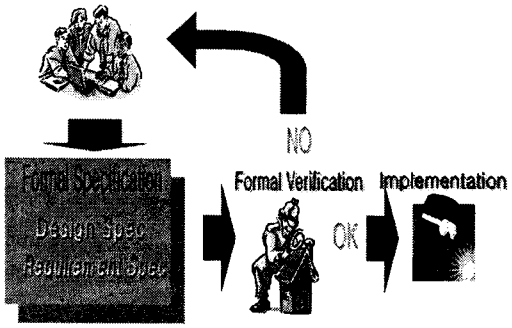

그림 1 정형기법 절차
Fig. 1 Procedure of formal methods

따라서 철도분야와 같은 안전성이 매우 중요한(Safety-critical) 시스템에서는 소프트웨어의 정확성과 신뢰성을 보장하기 위한 한 가지 방법으로 정형기법을 연구하고 있다. 정형기법이란 소프트웨어 공학의 일종으로 오류가 없는 시스템을 설계하여 시스템의 신뢰성을 높이려는데 목적이 있다 [8]. 이러한 정형기법은 수학적 기호를 사용함으로 시스템의 명세를 작성하는데 있어서 자연어가 일으킬 수 있는 애매모호함이나 불확실성을 최소한으로 줄일 수 있고, 설계된 사용자의 요구조건과 동일한지 수학적인 성질을 이용하여 증명할 수 있는데, 크게 정형명세와 정형검증으로 나눌 수 있다. 정형명세는 시스템이 달성해야 하는 요구 사항과 그러한 요구 사항을 만족시키는 설계를 명세화 하는데 그 목적이 있고, 정형 검증은 그 설계가 요구사항을 만족하는지를 검사하는 일련의 과정을 의미한다.

정형 명세는 수학적 논리 기호식이나 의미가 명확한 정형 다이어그램 등을 사용해 명세화하여 자연어 명세의 애매모호함과 불확실성을 제거하는 방법을 말한다. 정형명세 언어로는 크게 논리식이나 프로세스 대수와 같은 수학적 기호와 문자를 사용하는 언어와, State-diagram과 같은 도식적인 명세 언어를 들 수 있다.

정형검증은 Fig. 1과 같이 정형명세를 적용하여 작성된 시스템 설계 및 요구사항의 만족여부를 수학적, 논리적 증명 방법을 사용하여 검사하는 것을 말한다. 정형검증 방법에는 크게 모델체킹(Model Checking) 방법과 정리증명법(Theorem Proving)이 있다. 모델체킹 방법은 유한 상태의 병렬 시스템을 자동으로 검증해 주고 항상 원하는 결과가 나옴을 수학적으로 증명해 주는 기법이다. 또한, 정리증명법은 논리수학식을 사용해서 수학적으로 증명하는 기법인데, 사용하는데 있어서 시간을 많이 소모하고 전문가들을 요구한다는 단점이 있어서 일반적으로 정리증명법보다 사용하기 수월한 모델체킹 방법이 더 선호되고 있다.

3. 국내 열차제어시스템을 위한 정형기법 적용방안

정형기법들은 각자 고유의 적용 가능한 영역이 있으며, 이 영역에 맞게끔 사용해야만 최대한의 효과를 거둘 수 있다. 또한 앞 절에서도 언급하였듯이 실제 적용성을 위해 사용상 편의성을 고려하여야 하며, 정형기법을 지원하는 신뢰성 있는 도구의 선정 등 다양한 고려사항들이 필요하다. 본 논문에 앞서 Fig. 2의 정형명세 언어와 이의 지원을 위한 도구 선정 기준을 바탕으로 국내 열차제어시스템의 바이탈 소프트웨어에 적합한 정형기법을 채택하기 위한 여러 정형기법 방법들을 비교, 분석하는 연구를 수행한 바가 있다 [9][10].

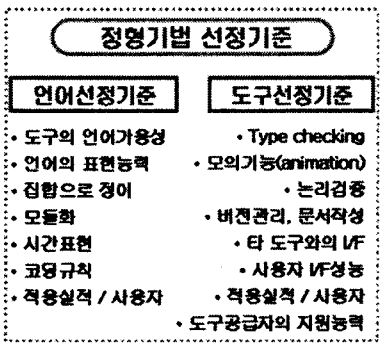

그림 2 정형기법 선정기준
Fig. 2 Guideline of formal method selection

이러한 정형기법 선정 기준을 근거로 비교분석한 결과로서 본 논문에서는 정형명세를 위한 방법으로 Z와 Statechart를 선정하였으며, 정형검증을 위해서는 Statechart를 기반으로 한 'Statemate'를 선정하였다. 본 논문에서 제안하는 Z와 Statechart의 절충안은 기존의 다른 방법에 비해서 정형명세 단계에서 시스템 정의와 같은 요구사항 명세 작성시 매우 높은 완전성을 이룸과 동시에, 그래픽 기반의 정형명세언어로 인한 사용의 편의성으로 개발시간을 단축할 수 있는 커다란 이점을 지니고 있다.

3.1 Z와 Statechart 절충안 분석

Z는 정형명세 작업에서 정확성이 높다는 큰 장점이 있는 반면, 모델링과 시뮬레이션과 같은 작업에 있어서는 데이터 흐름을 시간적인 개념을 포함하여 나타내기 힘들다는 단점을 지니고 있어, 이를 보완할 수 있는 상태천이 기반의 Statechart를 복합적으로 사용하는 방법이 효과적이라는 점을 고려하였다. 이에 따라, 정형명세 및 검증까지 지원되는 Statechart 도구로 가장 널리 알려진 툴인 Statemate와 Z를 함께 적용하는 Fig. 3과 같은 방식을 제시하였다.

먼저 시스템 요구사양과 개발사양을 나타내는 정형명세화 단계를 거쳐야 한다. 정형명세 단계에서는 자연어로 되어 있는 사양서를 수학적 논리 기호식이나 의미가 명확한 다이어그램 등의 그래픽 표현을 사용하는 정형명세 언어로 명세를 작성하기 때문에 정확성을 높일 수가 있다. 본 논문에서 적용 제안한 Z는 수학적 논리와 집합을 이용하므로 시스템 정의와 같은 요구사양의 명세 작성에는 매우 높은 완전성을 이룰 수 있게 해준다 [8][11]. 명확한 의미로 표현이 가능하다는 장점을 갖는 Z는 보통 정의하고자 하는 시스템의 데이터 타입을 체킹해주고, 그런 데이터의 흐름을 효과적으로 표현하여 시스템을 제어할 수 있게 해준다.

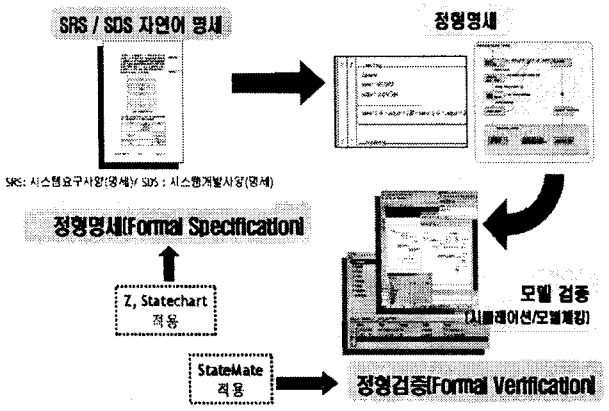

그림 3 철도신호시스템 바이탈 S/W를 위한 정형기법 적용 방안

Fig. 3 Formal method application for vital S/W of train control systems

Z언어 명세의 특성을 잘 나타내 주는 것이 바로 Fig. 4의 스키마(Schema)이다. 스키마는 여러 다른 표현으로부터 Z 명세를 구별하게 해주는데, 명세하고자 하는 시스템에 변수들을 도입하고 그 변수들 사이의 관계를 기술하는 구조를 갖는다. Fig. 4의 처음 스키마 박스에서 확인할 수 있듯이 그 구조는 먼저 스키마 이름(Schema name), 가운데 선의 위 부분에는 시스템 상태를 구성하는 변수 선언(Schema signature), 선의 아래 부분은 그 변수의 값에 대한 제약을 나타내는 술부(Schema predicate)로 구성되어 있다. Fig. 4의 아래 부분은 '텍스트 에디터의 문자 삽입기능'을 모델링한 스키마 작성의 간단한 예를 나타낸 것이다.

일반적으로 에디터에서 이 동작은 키보드에서 문자나 숫자 키 하나를 누르면 실행되는데, 제어문자가 아닌 프린트 가능한 문자에 대해서만 적용된다. Fig. 4에서 'Insert'로 스키마 이름이 선언되어 있고, 'Insert'내의 변수선언 부분은 'Insert'가 스키마 'Editor' 상태를 변화시키는 동작임을 나타내고 있고, 삽입되어지는 입력변수를 'CHAR'로 선언하였다 (ch?:CHAR). Z에서는 입력 변수를 항상 변수의 끝에 물음표(?)를 붙이는데, 이는 단지 변수를 구별하기 위한 표현이다. 술부는 에디터 상태가 어떻게 바뀌는지를 알려준다. 술부의 첫 번째 줄(ch? printing)은 선조건(precondition)을 나타낸 것으로, 다음 동작이 발생하기 전에 반드시 참이 되어야하는 조건을 기술한다. 선 조건은 '문자삽입' 동작이 프린트 가능한 문자 입력에 대해서만 발생할 수 있다는 것을 알려준다. 술부의 나머지 부분은 후 조건(postcondition)으로 동작이 끝난 후의 상태를 표현하고 있다. 즉, 'left' = left ch?'는 새로운 문자 입력이 커서의 왼쪽에 더해진다는 것을 의미하며, 'right' = right'은 커서의 오른쪽 부분은 동작 전후에 변화가 없음을 의미한다.

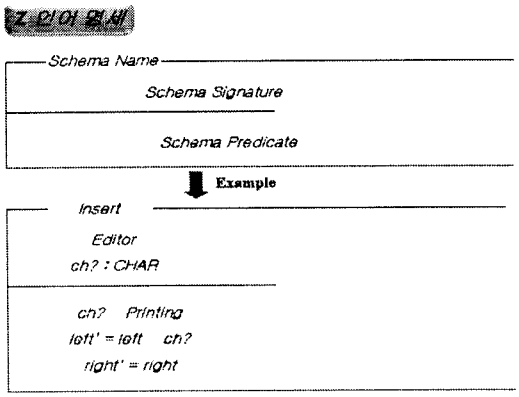

그림 4 Z 언어 명세의 스키마 표현 및 예제

Fig. 4 Schema expression of Z language specification & example

이처럼 Z언어는 명세하고자 하는 시스템의 데이터 타입 정의 및 체킹, 그리고 데이터 흐름을 효과적이고 명확하게 표현할 수 있지만, 집합론과 논리에 대한 정확한 지식과 경험이 요구된다는 단점을 가지고 있다. 또한, 데이터 흐름이 시간적인 개념을 포함하고 있어서, Z를 이용한 개발 사양에서는 구체적인 시스템 모델링과 시뮬레이션과 같은 작업에 있어서는 표현의 한계가 존재한다는 문제가 있다.

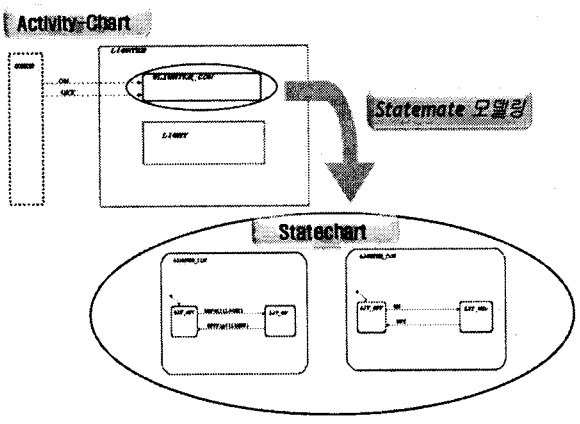

그림 5 Statemate의 Activity-chart와 Statechart를 사용한 손전등 동작의 명세화 사례

Fig. 5 Specification instance of a flashlight operation using Activity-chart & Statechart in Statemate

Statechart는 상태기반의 시각적 명세언어로 도식적으로 표현되기 때문에 이해하기가 편하고, 시스템의 행위적인 측

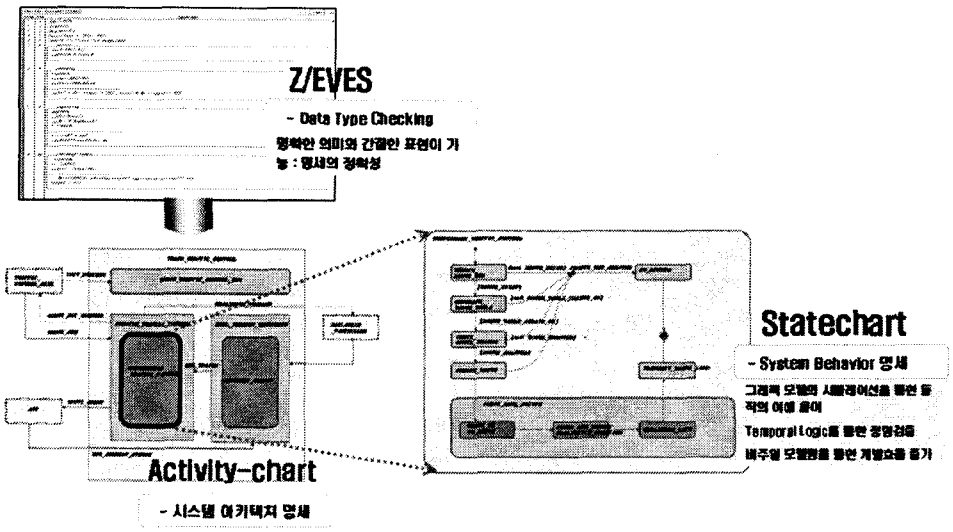

그림 6 정형명세의 적용방법
Fig. 6 Application of formal specification

면을 표현하는데 장점을 가지기 때문에, 임베디드 시스템의 설계 및 구현에 많이 적용되고 있다 [12]. 본 논문에서는 정형명세 언어의 하나인 Statechart를 기반으로 한 정형명세 및 정형검증 모두를 지원하는 Statemate를 활용하여 대상 시스템의 구조(architecture)를 툴에서 지원하는 Activity-chart를 사용하여 비쥬얼하게 나타내고, 시스템의 구체적인 행위(behavior)는 그래픽 모델의 시뮬레이션을 통해 이해하는데 용이하도록 한다. Activity-chart는 시스템의 상태를 표현하는 기능을 갖고 있어, 여러 계층으로 연결되어 있는 데이터의 흐름과 모듈간 인터페이스 사이의 작동을 확인할 수 있다.

Statechart는 시스템의 모든 설계를 도식적으로 표현하는데, 시스템의 입출력은 시그널과 데이터의 입출력으로 하고, 시스템의 행위는 상태도의 천이로 표현하게 된다. Fig. 5는 Activity-chart와 Statechart를 이용하여 손전등의 동작을 명세화한 예를 나타낸 것으로, 먼저 상위 그림에서 손전등에 대한 Activity-chart로 전체적인 손전등의 동작구조와 외부(사용자 : 'USER')와의 인터페이스 구조를 나타내고 있다. Activity-chart에서 확인할 수 있듯이 외부의 'USER'는 점선으로 표시된 'ON/OFF' control flow를 통해 '@LIGHTER_CON' 모듈을 통제하게 된다. 이 '@LIGHTER_CON' 모듈은 그림과 같이 하부에 Statechart 다이어그램을 포함하게 된다.

이처럼 Activity-chart를 이용하여 시스템의 구조 및 입출력 인터페이스를 표시하고, Activity-chart의 하부모듈인 Statechart를 통해서는 시스템의 기능 및 동작 상태를 표현하게 된다. Fig. 6은 본 논문에서 제안한 정형명세 방법을 설명한 그림으로 데이터 타입 정의 등은 Z에 의해 명세화하여 그 적절성 등을 검증하고, 이후 시스템의 구조 및 인터페이스는 Activity-chart에 의해, 시스템의 기능 및 데이터 흐름과 같은 시스템 행위를 Statechart에 의해 명세화하는 과정을 설명하고 있다. 특히 Activity-chart와 Statechart를 지원하는 Statemate 툴을 이용하여 Fig. 6과 같이 모델링된 그래픽기반 명세가 시뮬레이션을 통해 시스템을 어떻게 동작시키는지 가시적으로 확인할 수 있게 해준다.

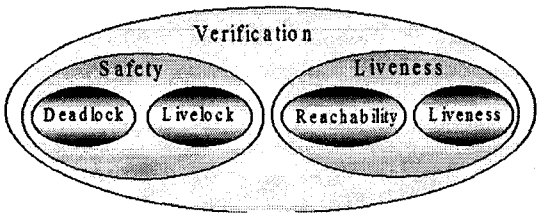

그림 7 정형검증을 위한 항목(상태)
Fig. 7 Formal verification items

시스템 요구사항은 Statemate에서 시뮬레이션을 통해 검증하고, 또한 정형검증을 위해서는 Fig. 7과 같은 상태들(Deadlock, Livelock, Reachability, Liveness)이 모델링 내부에 없음을 확인하여야 하는데, 이를 위한 정형검증 모듈의 항목들이 Statemate에 포함되어 있다. 다른 정형명세 언어와 비교해서 Statemate처럼 정형명세와 정형검증을 모두 지원하는 도구는 찾아보기 힘들다. 즉, Statemate의 가장 큰 장점은 비주얼 모델링, 시뮬레이션 코드생성, 도큐먼트 생성, 테스트 계획 등을 하나로 통합함으로써 복잡한 시스템의 명세 작성, 설계, 검증까지 모두 가능하게 해준다는 점이다.

3.2 제안한 방식에 따른 정형명세 결과

본 논문에서 바이탈 열차제어시스템 설계를 위한 정형기법의 적용절차 및 방법을 다양한 요소들을 고려하여 제안하였다. 제안한 방법의 타당성 검증을 위해 바이탈 열차제어시스템의 대표적인 열차진로 제어알고리즘에 대하여 요구사양을 정형명세화 하는 적용성 연구를 수행하였다. 열차 진로제어시스템은 역 구내에 진입하는 열차에 필요한 진로를

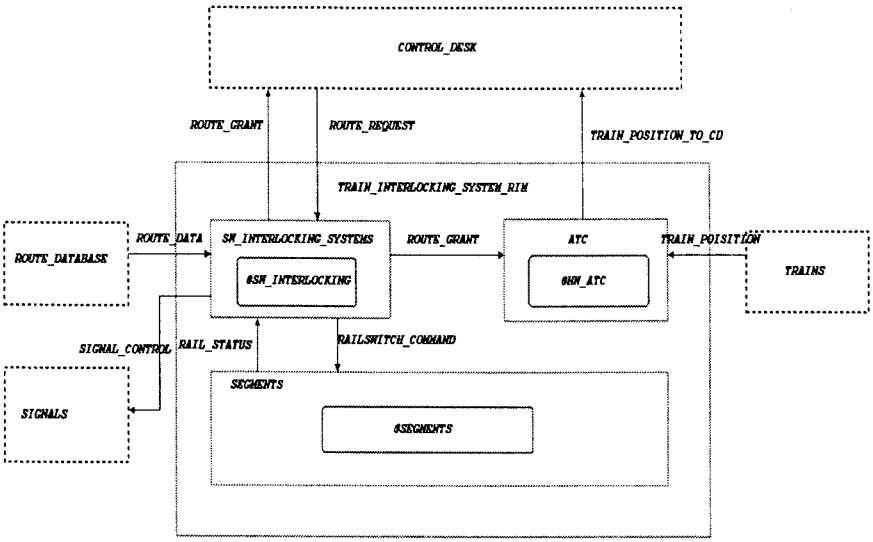

그림 8 Statemate의 Activity-chart를 이용한 열차제어시스템의 아키텍처
Fig. 8 System architecture using Activity-chart of Statemate

설정 및 보호하여, 진입하는 열차가 다른 열차와 충돌하거나 분기구간에서 탈선하지 않도록 하는 바이탈 제어장치이다.

이러한 열차진로제어시스템의 기본적인 요구사항에 맞추어 대상 열차제어시스템의 아키텍처를 작성한 결과가 바로 Fig. 8이다. Fig. 8은 Statemate의 Activity-chart를 이용하여 작성한 것이며, 작성 결과에서와 같이 장치 또는 소프트웨어를 바탕으로 하는 복잡한 시스템을 그래픽 모델로 효과적으로 나타내었음을 확인할 수 있다. 더 나아가 Statemate를 이용하여 대상 시스템을 설계할 수 있는 작은 단위로 분해하는 것도 가능하다.

Fig. 8의 Activity-chart를 토대로 하여 Z에서는 각 상태가 사용하는 데이터 형식을 검증하도록 하였는데, 이는 Z가 시스템의 입출력 인터페이스와 입출력 데이터의 타입을 기술하기에 적합하기 때문이다. 이러한 Z를 이용하여 열차제어시스템에서 선로전환기를 대상으로 다음과 같이 정형 명세 작업을 시행하였다. 선로전환기에서는 선로전환기 상태표의 내용을 토대로 하여 선로전환기 제어를 처리한다. 선로전환기 상태표는 선로전환기가 설치되어 있는 역명, 해당 선로전환기의 ID, 제어명령, 제어상태, 표시상태 및 연결되어 있는 폐색 등에 대한 정보를 저장해야 한다. 다음 Fig. 9는 해당하는 내용을 Z를 적용하여 명세화한 결과이다.

본 논문에서는 Z를 통해 검증된 데이터 형식을 사용하여 Statemate에서 시스템 제어와 알고리즘을 검증하도록 하였다. 이것은 Statemate의 Statechart가 상태기반의 언어로서 정해진 조건에서 시스템이 어떠한 반응을 하는가에 대한 행위를 표현하는데 적합하기 때문이다. 따라서 열차제어시스템의 기본적인 진로제어 기능을 담당하는 연동장치 소프트웨어에 대한 행위를 Statemate의 Statechart를 이용하여 작성해 보았다. 열차진로 제어프로그램의 제어흐름에 맞추어 진로를 탐색하는 부분과 진로상태표를 생성하는 부분, 선로전환기를 제어하고 진로를 설정하는 부분들로 분해하여 각각의 제어 행위를 알아보기 쉽도록 하는 Fig. 10과 같은

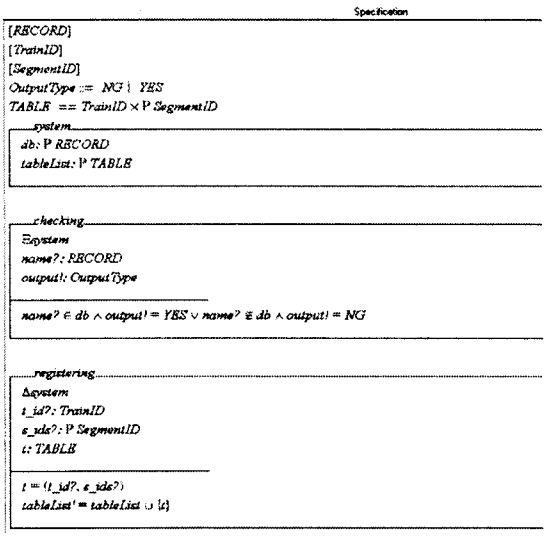

그림 9 Z 언어를 이용한 정형명세 결과
Fig. 9 Formal specification result using Z language

Statechart 결과를 얻을 수 있었다.

이러한 Z에 의한 데이터 타입 정의와 Activity-chart와 Statechart에 의한 그래픽기반의 정형명세화를 바탕으로 Statemate에 의한 시뮬레이션을 통해 진로제어시스템의 명확한 입출력 데이터의 흐름 및 기능 동작을 확인할 수 있었다. 이러한 정형명세화 과정을 통해 연구 초기에 작성한 자연어로된 요구사양이 보다 구체적이고 명확하게 재작성되었으며, 이러한 정형명세를 거친 요구사양이 시스템의 상세한 설계 및 제작을 위해 필요한 시간 단축 및 안전성 향상을 가져올 것으로 예상된다.

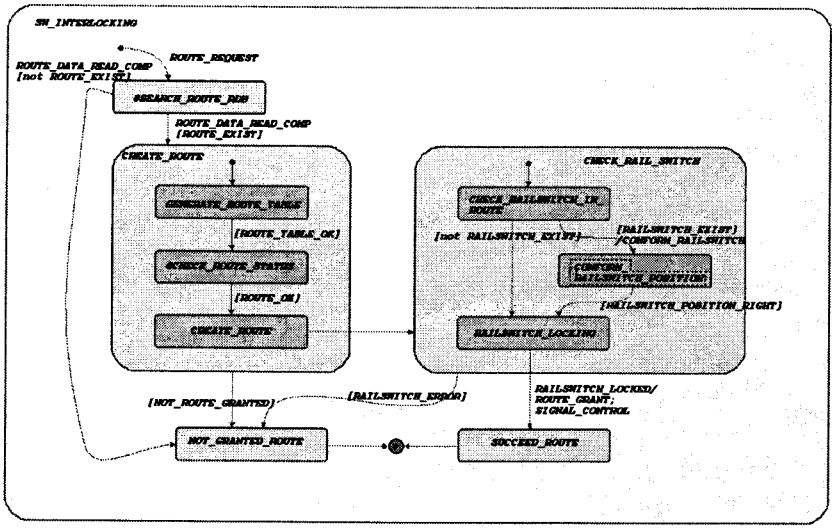

그림 10 연동장치 소프트웨어의 Statechart 작성 결과

Fig. 10 Statechart result of Interlocking S/W

4. 결 론

중대한 시스템에 사용되는 바이탈 소프트웨어를 개발할 때, 정형기법에 해당되는 정형명세와 정형검증을 사용하면 개발된 제어 시스템들의 안전성이 효율적으로 증가된다. 그러나 정형기법을 실제의 산업분야에서 활용하기에는 그 연구결과가 아직 미흡한 상황이며, 특히 국내에서는 실제 적용 사례가 철도 분야에서는 전무한 상태이기 때문에 열차제어 시스템을 위한 정형기법 적용의 구체적인 절차가 반드시 필요하다. 대부분의 국외 정형기법의 연구 사례에서처럼 열차제어시스템의 정형명세 및 검증을 위해 B method나 VDM 같이 수학적 논리식을 활용한다면, 이러한 정형명세 언어의 복잡성으로 인해 고도의 전문지식이 필요하여 현재와 같이 실제 철도시스템에 적용되지 못하고 실험실 수준의 연구에 그치고 말 것이다.

하지만 본 논문에서 제안한 Z와 그래픽 기반의 Statechart를 복합적으로 적용한 정형기법 적용방법은 이러한 문제점을 극복할 수 있는 실제적인 대안이 될 수 있다. 특히, 본 논문을 통한 정형명세 적용성 연구를 통해 제안한 방법에서는 대부분의 기능 및 데이터 흐름을 그래픽 형태의 정형명세 언어를 적용함으로 인해 정형기법에 대한 고도의 전문지식의 필요성이 줄어들게 되어 실제 적용가능성을 높일 수 있을 것으로 기대된다. 이에 따라 본 논문에서 제안한 방법을 적용할 경우 기존의 연구에서 정형명세언어 자체에 대한 전문적인 지식보다는 열차제어 로직 자체에 대한 분석에 많은 노력을 기울일 수 있게 됨으로서, 바이탈한 열차제어 시스템의 보다 높은 안전성 확보가 가능할 것으로 기대된다. 또한, 이와 함께 철도 RAMS 관련 국제규격에서 명시하고 있는 요구사항도 충족할 수 있을 것으로 예상된다.

참 고 문 헌

[1] IEC 62278, "Railway Applications - The specification and demonstration of RAMS", 2002.

[2] IEC 62425 Ed. 1, "Railway Application: Communications, signaling and processing systems - Safety related electronic system for signaling", 2005.10.

[3] Alain Faivre and Paul Benoit, "Safety Critical Software of Meteor Developed with the B Formal Method and the Vital Coded Processor", World Congress on Railway Research(WCRR), 1999.

[4] O. Lahlou, P. Bon and L. Allain, "Formalisation and Simulation of Operating Rules Using Coloured Petri Nets", Computers in Railway X, pp. 329-340, 2006.

[5] 福岡 博, 福田 光芳, 'ペトリネツトによる連動仕様の檢証', RTRI Report , Vol. 9, No. 11, pp. 19-24, 1995.

[6] J. L. Boulanger, P. Bon and G. marianom "From UML to B - A Level Crossing Case Study", Computers in Railway X, pp. 351-362, 2006.

[7] J. G. Hwang, H. Lee, and G. T. Park, 'Performance Evaluation and Verification of Communication Protocol for Railway Signalling Systems', Computer Standards & Interfaces, vol. 27, pp. 207-219, 2005.

[8] Kotonya, G., and Sommerville, I, "Requirements Engineering: Process and Techniques", Wiley, 1998.

[9] H.J.Jo and J.G.Hwang, "The Analysis of Formal Methods for Applying to Vital S/W in Train Control Systems", ITC-CSCC 2007, Jul. 8-11, 2007.

[10] H.J.Jo and Y.K.Yoon, "Formal Method Application with Safety Guarantee in Railway Signaling Control Systems", APSS 2007, Oct. 30 - Nov. 2, 2007.

[11] Jonathan Jacky, "The Way of Z", Cambridge, 1997

[12] David Harel and Ammon Naamad, "The STATEMATE Semantics of Statecharts", ACM Trans. Soft. Eng. Method, Oct. 1996.

저 자 소 개

조 현 정 (趙 賢 庭)

2003년 한국항공대학교 항공전자공학과 졸업. 2005년 광주과학기술원(GIST) 정보통신공학과 졸업. 2005년~현재 한국철도기술연구원 열차제어연구팀 주임연구원.
Tel : 031-460-5458
Fax : 031-460-5449
E-mail : hjjo@krri.re.kr

황 종 규 (黃 宗 奎)

1994년 건국대학교 전기공학과 졸업. 1996년 동 대학원 석사졸업. 2005년 한양대학교 전자통신전파공학과 박사졸업. 1995년~현재 한국철도기술연구원 열차제어연구팀 선임연구원.
Tel : 031-460-5438
Fax : 031-460-5449
E-mail : jghwang@krri.re.kr

윤 용 기 (尹 用 基)

1994년 충북대학교 전기공학과 졸업. 1996년 동 대학원 석사졸업. 2005년~현재 한양대학교 전자전기제어계측공학과 박사과정. 1995년~현재 한국철도기술연구원 열차제어연구팀 선임연구원.
Tel : 031-460-5440
Fax : 031-460-5449
E-mail : ykyoon@krri.re.kr