상태차트를 이용한 열차제어시스템 요구사항의 명세 및 검증1)

이혁*, 황종규**, 최진영*

*고려대학교 컴퓨터학과
{hlee, choi}@korea.ac.kr
**한국철도기술연구원
{jghwang}@krri.re.kr

Verification of Railroad Control System using Statecharts

Hyuk Lee*, Jin-Young Choi*
*Dept. of Computer Science, Korea University
Jong-Gyu Hwang**
**Korea Railroad Research Institute (KRRI)

요 약

시스템 또는 소프트웨어의 개발에 있어서 요구사항은 가장 중요한 산출물 중 하나이며, 요구사항의 오류를 수정하는 비용은 프로젝트가 진행됨에 따라 급격히 증가하기 때문에 잘 작성된 요구사항은 개발비용의 절감효과를 가져올 수 있다. 자연어의 모호성으로 발생될 수 있는 오류들을 줄이기 위해 정형적인 언어를 사용하여 요구사항을 명세하고, 요구사항의 완전성을 높일 수 있다. 본 논문에서는 열차제어시스템의 핵심 기능 중 하나인 간격제어모듈의 요구사항을 상태차트로 명세 및 검증 하였다.

1. 서론

시스템 또는 소프트웨어의 개발에 있어서 요구사항은 가장 중요한 산출물 중 하나이다. 개발에 앞서 무엇을 개발할 것인지에 대한 정확한 판단은 개발하고자 하는 시스템 또는 소프트웨어의 성공과 실패를 좌우한다. 요구사항에서 발생한 오류가 발견되지 못하고 시스템이 상용된 후 발견되면 이에 대한 수정비용은 오류가 발생한 단계에서의 수정비용보다 30배 혹은 그 이상을 초래하게 된다. 그러나 초기단계에서 요구사항을 완벽하게 파악하는 것은 사실상 불가능에 가깝다. 따라서 반복적인 요구사항의 검토와 수정작업이 요구되며 개발이 진행 될수록 이전 단계의 산출물과의 확인 작업이 필요하다. 하지만, 자연어로 작성된 요구사항은 의미적 모호함이란 한계를 갖고 있어서 요구사항의 오류를 모두 찾는 것은 힘든 작업이다. 이렇게 작성된 요구사항이 갖는 모호성을 없애기 위한 방법으로는 정형적인 언어, 즉 모호한 표현이 불가능한 제한된 형태로 요구사항을 표현하는 것이다.

2. 관련연구

하렐(Harel)에 의해 제안된 상태차트(Statechart)는 우수한 가독성을 특징으로 가지며 시스템의 행위적인 부분을 효과적으로 명세할 수 있는 정형명세 언어로 매우 직관적이며 동시성 및 계층성을 표현하기에 좋다[2]. 상태차트는 시스템의 요구와 설계사항을 상태들과 전이들로 표현하며, 상태간의 전이는 E[C]/A 형태로 표기된다. E는 전이를 유발시키는 이벤트이며, C는 상태 변화의 조건, A는 Action으로 해당 전이가 일어나면 데이터 값이나 조건 값을 바꾸거나 이벤트들을 발생시킨다. 상태차트가 갖는 특징은 다음과 같다.

Statechart = State-diagrams + Depth + Orthogonality + Broadcast Communication

3. 열차제어시스템의 요구사항 명세

열차제어시스템의 핵심부분 중 열차간의 안전한 간격을 유지하게 하는 간격제어모듈(Distance Control Module: DCM)에 대한 요구사항을 상태차트로 명세를 하였고, 명세도구로는 I-Logix社의 STATEMATE를 사용하였다.
간격제어모듈이 수행하는 기능은 다음과 같다.

- BLC_OPEN_CLOSE (블록 개방 및 폐쇄)
- SET_TEMP_SPEED (임시 속도제한명령 처리)
- MONITOR_TRAIN_DIRECTION (열차 이동/방향 감시)
- VERIFY_TRAIN_POSITION (열차 위치 확인)
- TRAIN_DISTANCE_CONTROL (열차 간격제어)

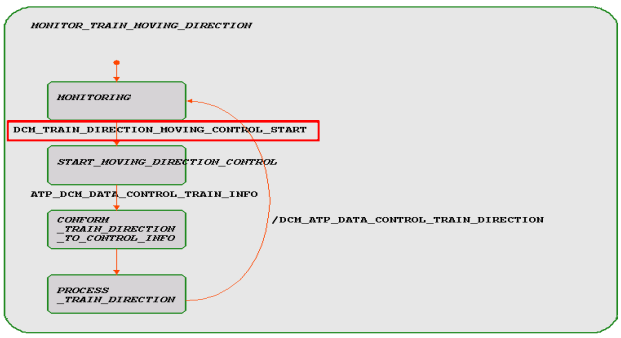
상태차트를 통해 자연어 요구사항을 명세하고, 시스템의 행위를 확인하기 위한 시뮬레이션을 수행하였다. 일차적으

1) 감사의 글 : 본 논문은 국토해양부가 출연하고 한국건설교통평가원에서 위탁시행한 철도종합안전기술개발사업의 결과입니다.

로 자연어 요구사항을 상태차트로 명세하는 과정에서 요구사항내의 모호성과 불완전성을 발견할 수 있었고, 시뮬레이션을 통해서 요구사항내의 오류를 발견할 수 있었다.

다음은 자연어로 작성된 열차이동방향감시 기능에 대한 요구사항의 한 부분이다.

- DCM은 열차의 열차운행정보(진행방향)을 차상시뮬레이터에게서 전송받는다.
- DCM은 수신한 현재열차위치정보와 기존에 DCM이 저장하고 있던 해당 열차의 열차위치정보를 비교 분석하여 해당 열차가 열차진행방향으로 정상적으로 이동하고 있는지를 감시한다.



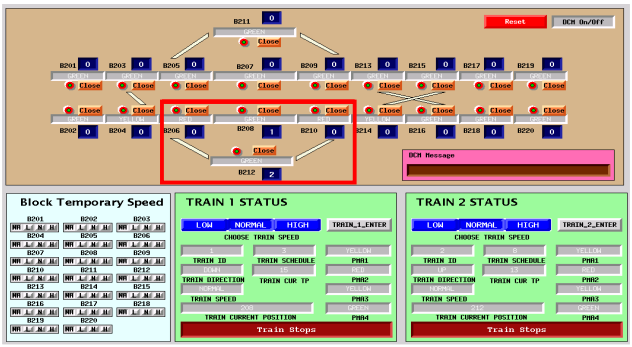
<그림 1> 자연어 요구사항을 표현한 상태차트 명세

열차이동방향감시 기능에 대한 자연어 명세에서는 열차이동방향감시가 언제 이루어지는지 명세하지 않고 있다. 상태차트 명세에서는 이러한 명세가 필요함을 보여주고 있으며, 이를 보완하기 위해 trigger 이벤트를 통해 시스템을 시작하고 있다. 이러한 부분에 대한 자연어 명세의 보완이 필요하다. 수정 및 보완된 자연어 요구사항은 다음과 같다.

(1) 열차 이동 방향 감시는 기능이 시작되면 열차 방향 감시 상태에서 대기한다.
(2) 열차 이동 방향 감시는 DCM의 열차 방향 감시 제어 신호가 들어오면, 열차 이동 방향 제어를 시작한다.
(3) 열차 이동 방향 감시는 열차 이동 방향 제어 중, ATP로부터 열차 정보가 들어오면, 열차 방향 확인을 한다.
(4) 열차 이동 방향 감시는 열차 방향 확인 후, 열차 방향 처리를 한다.
(5) 열차 이동 방향 감시는 열차 방향 처리 후, ATP에게 열차 방향을 보낸다.
(6) 다시 열차이동방향감시 기능을 시작한다.

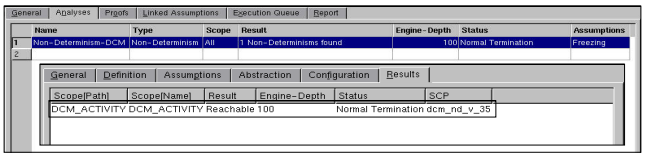
4. 검증 및 분석

<그림 2>는 두 열차가 교차할 때 더 이상 이동하지 못하는 교착상황이 발생하는 모습으로, 요구사항 내의 불충분한 예외사항 처리규칙들로 인한 결과이다. 자연어 요구사항에는 이러한 부분에 대한 언급이 없었으며, 명세 및 시뮬레이션을 통해 요구사항의 오류를 발견할 수 있었다. 이와 같은 불충분한 요구사항은 자연어 요구사항 분석 시 확인하기 힘든 부분이며, 명세언어와 도구의 사용으로 어렵지 않게 찾아 낼 수 있었다.



<그림 2> 두 열차의 교착 상태

<그림 3>은 명세한 상태차트들의 상태공간을 탐색하여 특정 속성에 대한 검사를 수행하는 기능인 모델체크(Model Check) 기능으로, 명세된 요구사항에 대해 비결정적 행위에 대한 검사를 수행한 결과, 우선순위가 정해지지 않은 두 개의 행위가 충돌하는 것을 확인할 수 있었다.



<그림 3> 비결정론적 행위 탐지

이 두 행위는 동시에 발생 가능한 행위들로, 상대 행위로부터 보호가 되지 않으면 비결정론적인 행위가 가능하며, 예상치 못한 시스템의 결과를 가져올 수 있는 부분이다. 요구사항의 수정을 통하여 명세상의 비결정론적 행위를 제거하고, 더 이상의 비결정론적 행위가 없음을 확인할 수 있었다.

	Name	Type	Scope	Result	Engine-Depth	Status	Assumptions
1	Non-Determinism	Non-Determinism	All	No Non-Determinism	inf	Normal Termination	None
2							

<그림 4> 비결정론적 행위 제거

5. 결론

본 논문에서는 열차제어시스템의 핵심기능인 간격제어 모듈의 기능적 요구사항을 상태차트로 명세하였고, 자연어 요구사항에서 찾지 못하였던 요구사항의 오류 또는 미완성된 부분을 시뮬레이션과 모델체크를 통해 확인할 수 있었다. 이로써 요구사항에 대해 더욱 정확히 파악할 수 있으며, 상세설계나 구현과 같은 단계를 시작하기 전에 좀 더 완성된 요구사항을 얻을 수 있었다. 이와 같이 요구사항 단계에서의 정형화 혹은 준정형화 명세언어의 사용은 자연어 요구사항의 모호함을 없애고 각 단계간의 이해관계를 높일 수 있다.

참고문헌

[1] Jame F. Peters, Witold Pedrycz, "Software Engineering - An Engineering Approach", Wiley, 2000

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