폴트 트리의 상태차트 변환 방법론 연구

이혁*, 이진호, 김진현, 최진영
고려대학교 컴퓨터학과
e-mail : {hlee, jhlee, jhkim, choi}@formal.korea.ac.kr

Study of methodology for converting Fault Tree to Statecharts

Hyuk Lee*, Jean-Ho Lee, Jin-Hyun Kim, Jin-Young Choi
Dept. of Computer Science, Korea University

요 약

안전성 분석 활동으로 널리 쓰이는 폴트 트리 분석은 그 원인들의 관계를 논리게이트로 표현하고 원인을 자연어로 설명한다. 폴트 트리에서 자연어를 사용하여 사고의 원인을 나타내는 것은 폴트 트리 자체의 검증뿐만 아니라, 동일한 대상의 시스템 명세에도 적용하기에 어려운 부분이다. 본 논문에서는 폴트 트리를 상태차트로 변환하는 방법을 제안 하였으며, 상태차트로 변환된 폴트 트리를 상태차트로 명세 된 시스템의 기능적 요구사항과 결합함으로써 상태차트로 표현된 기능 명세가 안전성 분석활동을 통해 얻은 폴트 트리에 순응 되는지 여부를 볼 수 있었고, 폴트 트리에서의 분석 대상인 사고가 발생하기 까지를 추적할 수 있었다.

1. 서론

폴트 트리 분석 (Fault Tree Analysis)[1]은 안전성 분석 방법 중 하나로서 시스템의 특정 사고에 대해 트리를 그려나가면서 사고의 원인들을 분석하는 방법이다. 폴트 트리 분석은 시스템의 고장 및 결과, 시스템의 기능들에 대한 지식을 기반으로 분석을 한다. 분석을 통해서는 특정한 사고에 대한 원인이 되는 사상(Event)들의 조합을 결과로 얻을 수 있다. 폴트 트리 분석은 시스템의 안전성을 분석하는데 유용한 도구로 쓰이고 있다.

하지만, 이렇게 널리 쓰이는 폴트 트리 분석은 대개 수작업으로 이루어지고, 사고의 원인들이 자연어로 표현되는 것이 일반적이며, 폴트 트리 자체의 검증을 위해서는 다른 정형 명세언어로의 변환이 불가피함 또한 나타내고 있다. 기존 연구들이 보여주고 있는 폴트 트리의 검증에 대한 내용은 주로 시제논리의 형태를 띄고 있고, 이러한 시제논리 형태로 변환된 폴트 트리는 트리 자체의 검증만을 위한 변환일 뿐, 다른 형태의 언어로 명세 된 시스템과의 비교/분석을 위해서는 추가적인 변환을 필요로 하고 있다.

본 논문에서는 안전성 분석 방법 중 가장 널리 쓰이는 폴트 트리 분석의 대상으로 열차 건널목 시스템에서 열차와 자동차의 '충돌'사고를 선정하였다. 시스템의 기능적 요구사항을 상태차트로 명세했다는 가정하에 폴트 트리를 상태차트로 변환하는 방법을 제안한다. 대상 시스템으로는 열차 건널목 시스템을 선정하였고, 상태차트로 모델링 된 시스템의 기능적 요구사항 명세와 병행 구성함으로써 폴트 트리상의 사고 경위를 시스템의 기능 명세에서 추적이 가능하고, 또한, 보완이 가능할 수 있음을 보였다.

본 논문의 구성은 다음과 같다. 다음 장에서는 본 논문의 연구배경인 상태차트와 폴트 트리 분석에 대해 살펴보고, 3 장에서는 폴트 트리를 상태차트로 변환하는 방법에 대해 설명하고, 마지막 장인 4 장에서는 결론과 향후 연구방향에 대해 설명한다.

2. 연구 배경

2.1 상태차트

상태차트는 하렐 (Harel)에 의해서 처음 제안되었고 뛰어난 가독성을 가지며 복잡한 반응형 (Reactive) 시스템의 행위를 효과적으로 명세할 수 있는 정형명세 언어이다. 상태 차트는 직관적인 이해가 매우 편하며 동시성 및 계층성을 표현하기에 매우 적합한 언어이다[2]. 상태차트는 시스템의 요구와 설계사항을 상태들과 전이들로 표현하고, 상태들은 더 이상 내부에 하위 상태를 가지지 않는 상태와 내부에 하위 상태를 가지는 상태들이 있다.

상태들간의 전이는 E[C]/A 형태로 표기되는데, E 는 전이를 유발시키는 이벤트이다. 이러한 이벤트들은 Broadcast 되는 특성으로 특정 전이를 대상으로 발생하지 않고, E 의 범위 내에 있는 모든 전이가 동시에 발생된 이벤트를 감지한다. C 는 상태 변화의 조건(Condition)이 된다. 즉, C 에 제시된 상태들이 만족되고 E 에 해당되는 이벤트들이 발생하게 되면 전이가 이루어지게 된다. A 는 Action 으로 해당 전이가 일어나면 데이터 값이나 조건 값의 변화를 발생시키거나 이벤트들을 발생시킨다. E, C, A 모두 필수가 아닌 선택적이며 E 와 C 가 모두 없으면 무조건 다음 시간 단위에 전이를 하게 된다.

상태차트의 특징은 다음과 같이 표현될 수 있다.
Statechart = State-diagrams + Depth + Orthogonality + Broadcast-communication

2.2 폴트 트리 분석

폴트 트리 분석은 시스템 안전공학의 대표적인 방법으로 1960 년대 초반에 제안되었고, 그 이후로, 전자 및 원자력 산업등 전반적인 산업분야에서 널리 사용되고 있다. 폴트 트리 분석은 특정 사고에 초점을 맞춰서 이 사고에 대한 원인을 규명하는 방법을 제공하는 연역적 방법이며, 사고를 발견하는 도구가 아니라 특정 예상 사고에 대한 원인을 Top-Down 방식으로 검토/분석하여 정성적/정량적 안전성을 평가 및 진단하는 방법이다.

폴트 트리는 시스템의 중요한 사고를 나타내는 정상사상 (Top Event)를 가지게 되고, 이것은 먼저 다른 방법에 의해 확인되어야 한다. 정상사상이 확인되면 이에 대한 가능한 원인을 찾기 위해 시스템을 분석한다. 폴트 트리 분석은 사고 발생의 원인이 되는 오류들의 조합을 나타내기 위해 이진논리 (Boolean Logic)를 사용하며, 트리의 각 레벨은 상위사상을 일으키는 필요 충분한 더 기본적인 사상들을 열거한다. 정상사상과 기본사상 (Basic Event) 사이의 중간사상 (Intermediate Event)들은 기본적인 사상들의 단순한 조합 또는 집합이다. 트리가 구성되고 나면 이것은 다시 이진논리식으로 쓰여질 수 있고, 정상사상의 원인이 되기에 충분한 기본사상들의 조합으로 간단히 표현될 수 있다.

3. 대상 시스템과 폴트 트리의 변환

본 장에서는 상태차트로 표현된 기능 명세의 안전성속성 만족 여부를 알아보기 위해 폴트 트리를 상태차트로 변환하여 기능 명세에 적용시키는 과정을 소개한다.

3.1 대상 시스템

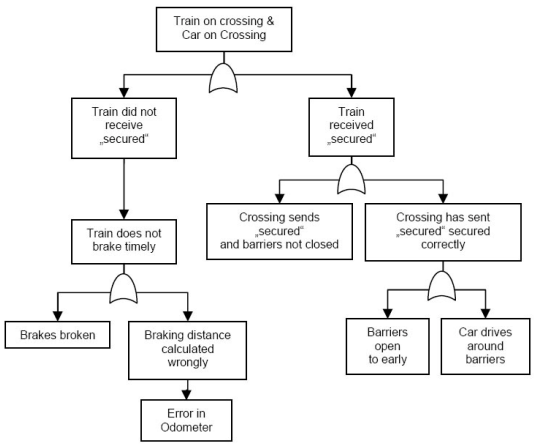
제안하는 방법의 적용을 위해 간단한 무선 신호 기반의 건널목 시스템을 대상으로 정하였다.

건널목 시스템은 크게 열차와 건널목으로 구성이 되어 있다. 열차는 특정 위치에 도달하게 되면 건널목에게 차단기 차단 요청 신호를 보내게 된다. 그리하면, 이 신호를 받은 건널목은 차단기의 상태를 확인하고 차단기를 내린 후에 열차에게 차단 완료 신호를 보낸다. 만약, 건널목이 차단 완료를 실패할 경우 (열차가 차단 완료 신호를 못 받은 경우), 열차는 건널목에 도달하기 전에 정지할 수 있는 최단제동거리에서 제동을 시작하게 된다. 건널목 시스템의 기본적인 행위와 시나리오는 위와 같다.

3.2 특정 사상에 대한 폴트 트리

안전성 활동에서는 발생할 수 있는 사고를 찾아내고, 폴트 트리 분석을 통해서 찾아낸 사고(들)에 대해서 원인을 분석해내는 기법이다. 하지만, 발생할 수 있는 사고들의 원인들은 매우 다양할 수 있다. 사고의 원인들은 시스템(하드웨어, 소프트웨어)상에서의 결함일 수도 있고, 시스템을 사용하는 사람에게 있을 수도 있으며 심지어는 환경적인 요인에 의해 발생 할 수도 있다.

본 논문에서는 폴트 트리의 범위를 상태차트로 명세할 시스템 수준까지로 정하였고, 그 외의 범위에 대해서는 다루지 않는다.



(그림 1) 예제 시스템의 특정 '사고'에 대한 폴트 트리

상태차트로 변환할 폴트 트리의 모델은 열차와 자동차의 충돌을 정상사상으로 하는 모델이다.

위의 폴트 트리는 다음과 같이 해석할 수 있다.

정상사상: 열차와 자동차의 충돌

1) 열차가 건널목으로부터 '차단완료' 신호를 미수신

1-1) 열차가 제때에 정지를 하지 못함

1-1-1) 브레이크의 고장
1-1-2) 제동거리 계산 오류 (정차 지점 계산 오류)

1-1-2-1) 주행거리계의 오류

2) 열차가 건널목으로부터 '차단완료' 신호를 수신

2-1) 건널목에서 차단기가 내려지지 않은 상태로 차단완료 신호를 열차에게 송출

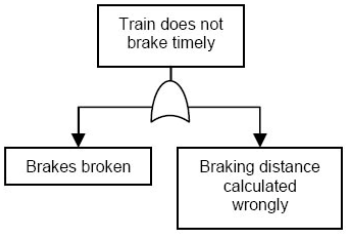
2-2) 건널목에서 차단완료 신호를 열차에게 올바르

게 송출

2-2-1) 차단기가 정해진 시간보다 일찍 올라감
2-2-2) 자동차가 내려진 차단기를 무시하고 건널목을 우회해서 통과

3.3 폴트 트리의 변환

아래 (그림 2)는 위에서 살펴본 '열차와 자동차의 충돌' 폴트 트리의 일부분이다. 트리의 해석은 다음과 같다.



(그림 2) 예제 폴트 트리

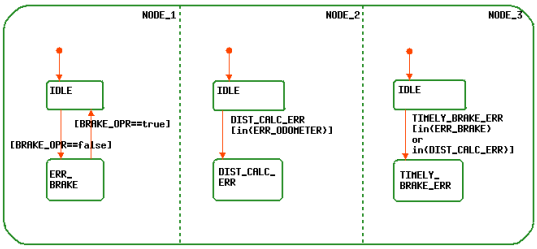
열차가 제때에 제동하지 못하였고, 이것에 대한 원인은 브레이크의 고장이거나 제동거리의 계산이 잘못된 경우이다.

폴트 트리를 상태차트로 변환하기 위해서는 트리에서 각각의 사상들이 나타내는 상태에 대한 정확한 이해가 필요하고, 이를 적절한 상태와 상태의 변화로 표현을 해야 한다.

```
사상_변환 (사상) {
  if (하위사상개수 == 0) {
    발생한 상태 생성
    발생하지 않은 상태 생성
    전이 및 조건 생성
  }
  else {
     발생한 상태 생성
     발생하지 않은 상태 생성
     전이 및 조건생성
     if (하위사상개수 == 1) {
        조건=현재조건 && (하위사상 발생됨)
     }
     else {
        if (하위사상들이 논리곱으로 구성) {
            조건=현재조건 && ((하위사상 a 발생됨) && (하위
            사상 b 발생됨))
        }
        elseif (하위사상들이 논리합으로 구성) {
            조건=현재조건 && ((하위사상 a 발생됨) || (하위
            사상 b 발생됨))
        }
     }
  }
```

(그림 3) 변환을 위한 의사코드(pseudo code)

본 논문에서는 특정 사상에 대한 폴트 트리를 상태차트로 변환하기 위해서 (그림 3)과 같은 규칙을 두고 변환을 하였다. 단, 폴트 트리의 모든 논리게이트는 이진 게이트만 사용한다고 가정한다



(그림 4) 상태차트로 표현된 예제 폴트 트리

(그림 4)에서 각각의 차트는 (그림 2)의 폴트 트리와 대응되며, NODE_1 은 'Brakes broken', NODE_2 는 'Braking distance calculated wrongly' 그리고 NODE_3 은 'Train does not brake timely'에 대응되는 차트이다.

NODE_1 차트를 살펴보면, ERR_BRAKE 상태는 BRAKE_OPR 이 거짓일 경우 활성화 된다. 폴트 트리 상에서 '브레이크 고장'은 더 이상의 원인을 규명할 수 없는 기본사상으로 되어 있지만, 이것은 시스템 디자인의 정도에 따른 폴트 트리 분석이기 때문에 상태차트로 변환할 시에는 이를 야기시키는 원인 또는 조건이 필요하다.

NODE_1 에서 NODE_3 으로 가기 위해서는 NODE_3 이 만족되어야 한다.

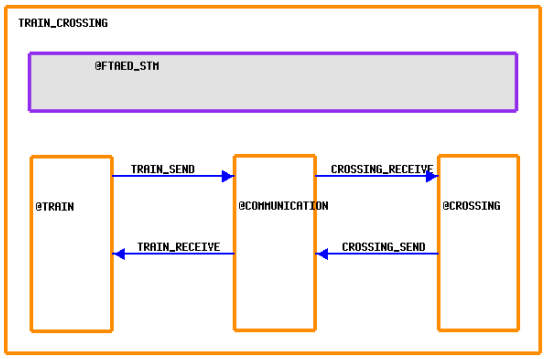
Top-Down 방식으로 살펴보게 되면:
1. 열차가 제때에 제동을 하지 못했다.
2. 원인은 브레이크의 고장이다.

이것을, Bottom-Up 의 방식으로 보게 되면:
1. 브레이크가 고장이 났다.
2. 열차가 제때에 제동을 하지 못했다.

Bottom-Up 방식의 접근에서 '브레이크 고장' 이 만족되었다고 해서 '열차가 제때에 제동을 하지 못했다'가 만족되는 무조건 적인 관계가 이루어 지지는 않는다. 브레이크가 고장 날 경우와 제동이 제때에 이루어 지지 않을 경우는 확률적인 문제이다. 다만, 브레이크가 고장 났을 때에, 제동이 제때에 이루어 지지 않을 경우에만 상위사상(Upper Event)로 올라 갈수 있음을 인지해야 한다.

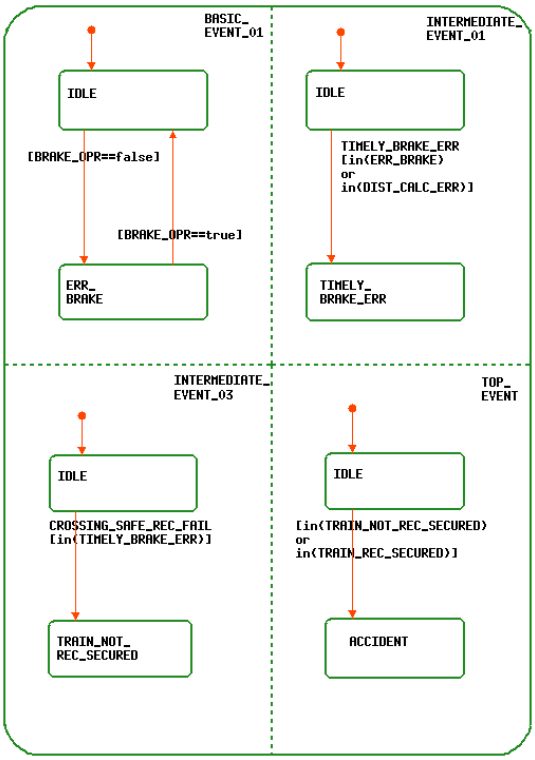
(그림 5)는 상태차트 명세의 최상위 레벨인 엑티비티차트(Activity Chart)로서 각 컴포넌트 간의 상호관계를 나타내는 차트이다. 상태차트로 표현된 기능 명세에 위의 방법으로 만들어진 폴트 트리 상태차트를 기

능 명세의 최상위 레벨 (각 컴포넌트와 동일한 레벨)에 넣음으로 병행 구성을 이루도록 한다. 상태차트로 표현한 폴트 트리를 컴포넌트들과 같은 레벨에 구성함으로써 시스템의 전체적인 행위와 조건들의 변화를 감지 할 수 있다.



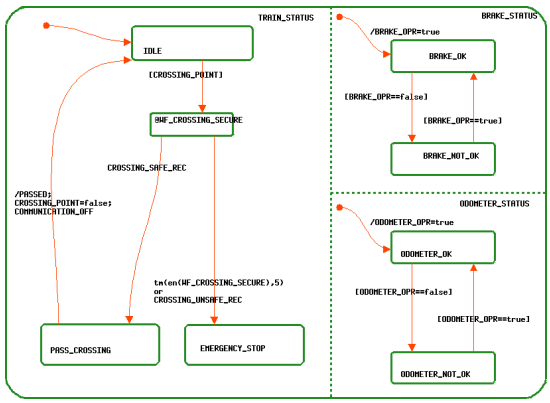
(그림 5) 상위레벨에서 폴트 트리 차트 구성

(그림 6)은 (그림 5)에서 FTAED_STM 엑티비티의 내부에 존재하는 상태차트(예제로 보인 (그림 1)의 폴트 트리로부터 변환된 차트)에서 'Brake Broken' 기본 사상에서 정상사상에 이르는 경로만 잘라낸 부분이다.



(그림 6) 특정 경로를 통한 정상사상 발생 경로

(그림 7)은 열차의 행위를 명세한 상태차트이다. 열차의 행위 명세에서 브레이크가 고장 날 경우, 이를 폴트 트리 차트에서 한눈에 확인할 수 있다. 더 나아가서 위험 회피, 탐지 또는 제거를 위해 시스템의 명세가 수정이 되었을 때에, 폴트 트리 상태차트를 이용하여 수정된 명세가 원하는 안전 속성을 만족시키는지 여부를 쉽게 파악할 수 있을 것으로 예상된다.



(그림 7) 열차의 행위를 나타낸 상태차트

4. 결론 및 향후 연구

본 논문에서는 안전성 분석 활동에서 많이 쓰이는 일반적인 폴트 트리 분석에서 자연어로 표현된 폴트 트리를 상태차트를 이용하여 표현하였고, 상태차트로 표현된 시스템의 기능 명세와 병행 구성 함으로써 변환된 폴트 트리상에서 원인-결과 추적이 한눈에 알아볼 수 있게 되었다. 또한, 차후에 안전성 요구사항에 부합하기 위한 시스템 명세의 보완을 도울 수 있음을 보였다. 본 논문에서는 시스템의 기능 명세 범위에 대해서만 폴트 트리의 적용 및 사용이 가능하였지만, 폴트 트리를 좀더 다양한 방면으로 연구하여 다른 각도로의 접근이 가능할 것으로 보인다.

참고문헌

[1] W. E. Vesely, Joanne Dugan, Joseph Fragola, Joseph Minarick III, and Jan Railsback. Fault Tree Handbook with Aerospace Applications. National Aeronatics and Space Administration, August 2002.

[2] Jame F. Peters, Witold Pedrycz, "Software Engineering – An Engineering Approach}, Wiley, 2000

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

[4] Gerald Luttgen and Michael von der Beeck and Rance Cleaveland, "A Compositional Approach to Statecharts semantics", Report 12, Institute for Computer Applications in Science and Engineering (ICASE 2000)