



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년06월27일
 (11) 등록번호 10-1993635
 (24) 등록일자 2019년06월21일

(51) 국제특허분류(Int. Cl.)
 G06F 17/50 (2006.01) B60W 30/14 (2006.01)
 (52) CPC특허분류
 G06F 17/5009 (2013.01)
 B60W 30/14 (2013.01)
 (21) 출원번호 10-2018-0030825
 (22) 출원일자 2018년03월16일
 심사청구일자 2018년03월16일
 (56) 선행기술조사문헌
 KR1020130088450 A
 “클러스터링 활용 국부 메타모델을 이용한 대용량 데이터 기반 성능 예측”, 대한기계학회 춘추학술대회(pp. 2102-2104), 2016년 12월
 “지능형 서비스 로봇을 위한 온톨로지 기반의 동적 상황 관리 및 시-공간 추론”, Journal of KIISE Vol. 43 No. 12(pp. 1365-1375), 2016년 12월
 KR1020130112507 A

(73) 특허권자
 충북대학교 산학협력단
 충청북도 청주시 서원구 충대로 1 (개신동)
 (72) 발명자
 홍장의
 충청북도 청주시 흥덕구 1순환로1137번길 79, 915-703 (분평동, 주은프레지던트)
 남승우
 충청북도 청주시 서원구 예체로19번길 7, 204호 (개신동)
 (74) 대리인
 김정현

전체 청구항 수 : 총 8 항

심사관 : 박승철

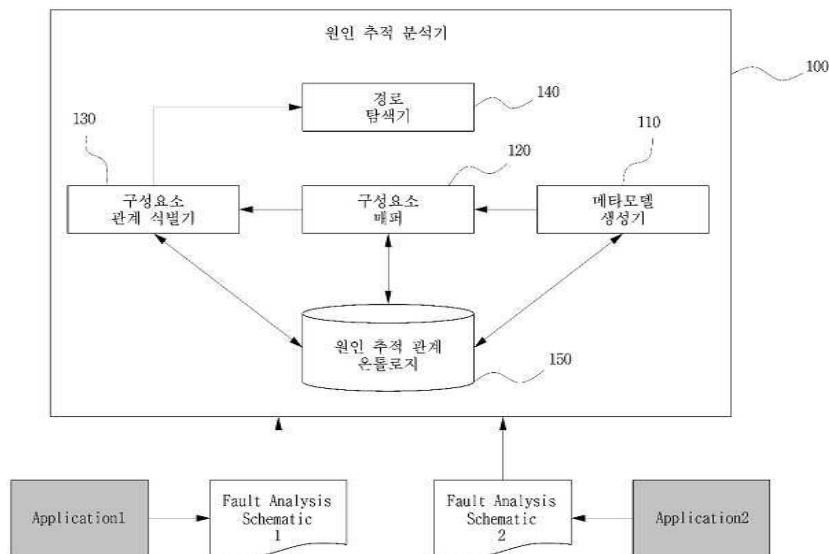
(54) 발명의 명칭 **지능형 자율 시스템에서의 사고 원인 추적 시스템**

(57) 요약

본 발명의 지능형 자율 시스템에서의 사고 원인을 추적하기 위한 사고 원인 추적 시스템에서, 각 결함분석 기법 산출물에 대한 메타모델 생성, 구성요소 매핑, 그리고 관계를 식별하기 위한 지식 정보가 저장되어 있는 원인 추적 관계 온톨로지, 상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 지능형 자율 시스템을 구성

(뒷면에 계속)

대표도 - 도1



하는 각 시스템에서 수행된 결함분석 기법 별 결함분석의 산출물을 입력받고, 이에 대한 메타모델을 클래스 다이어그램으로 생성하고, 원인 추적 분석에 사용할 메타모델 요소를 생성하기 위한 메타모델 생성기, 상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 각 기법의 메타모델 요소와 원인 추적 분석에 사용할 메타모델 요소를 결합하고, 이에 대한 CTFB(Cause Tracing Functional Block)을 생성하기 위한 구성요소 매핑, 상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 각 기법의 메타모델 요소들이 가지고 있는 정보들의 관계를 설정하고, 부분 안정성 평가가 가능하도록 이펙트 분류 기준에 관계의 영향을 지정하고, 이벤트가 트리거될 때, 블록 내부 또는 블록 간에 발생하는 이펙트 정보를 토대로 원인추적분석을 수행하는 구성요소 관계 식별기, 상기 구성요소 관계 식별기를 통해 상호 영향 관계가 정의된 두 시스템 간에 의도되지 않은 이벤트에 대한 추적정보를 입력으로, 원인을 유발하는 노드와 이펙트 명을 CTFBD(Cause Tracing Functional Block Diagram)에 나타내는 경로 탐색기를 포함한다.

이 발명을 지원한 국가연구개발사업

과제고유번호 2017M3C4A7066479

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 차세대정보컴퓨팅기술개발사업

연구과제명 CPS 어플리케이션의 복합적인 안전성 평가 지원을 위한 통합 추적성 분석 기술(Integrated Traceability Analysis Technique for Compositive Safety Assessment of CPS Application)

기여율 1/1

주관기관 충북대학교

연구기간 2017.08.01 ~ 2018.05.30

명세서

청구범위

청구항 1

지능형 자율 시스템에서의 사고 원인을 추적하기 위한 사고 원인 추적 시스템에서,

각 결합분석 기법 산출물에 대한 메타모델 생성, 구성요소 매핑, 그리고 관계를 식별하기 위한 지식 정보가 저장되어 있는 원인 추적 관계 온톨로지;

상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 지능형 자율 시스템을 구성하는 각 시스템에서 수행된 결합분석 기법 별 결합분석의 산출물을 입력받고, 이에 대한 메타모델을 클래스 다이어그램으로 생성하고, 원인 추적 분석에 사용할 메타모델 요소를 생성하기 위한 메타모델 생성기;

상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 각 기법의 메타모델 요소와 원인 추적 분석에 사용할 메타모델 요소를 결합하고, 이에 대한 CTFB(Cause Tracing Functional Block)을 생성하기 위한 구성요소 매핑;

상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 각 기법의 메타모델 요소들이 가지고 있는 정보들의 관계를 설정하고, 부분 안정성 평가가 가능하도록 이펙트 분류 기준에 관계의 영향을 지정하고, 이벤트가 트리거 될 때, 블록 내부 또는 블록 간에 발생하는 이펙트 정보를 토대로 원인추적분석을 수행하는 구성요소 관계 식별기;

상기 구성요소 관계 식별기를 통해 상호 영향 관계가 정의된 두 시스템 간에 의도되지 않은 이벤트에 대한 추적 정보를 입력으로, 원인을 유발하는 노드와 이펙트 명을 CTFBD(Cause Tracing Functional Block Diagram)에 나타내는 경로 탐색기를 포함하는 사고 원인 추적 시스템.

청구항 2

청구항 1에 있어서,

상기 메타모델 생성기는 FTA(Fault Tree Analysis)와 FMEA(Failure Mode and Effects Analysis)를 지정된 메타모델에 맞게 생성하고, 이 두 메타모델 객체를 입력으로 받는 CTA(Cause Tracing Analysis) 메타모델 객체를 생성하는 것을 특징으로 하는 사고 원인 추적 시스템.

청구항 3

청구항 2에 있어서,

상기 구성요소 매핑은 FT(Fault Tree) 스키마틱(Schematic)을 분석하고, 노드를 순차적으로 탐색하여 CTFB를 생성하되, AND 게이트 노드를 베이스 노드로 하여 CTFB를 생성하기 위한 트리를 탐색하는 것을 특징으로 하는 사고 원인 추적 시스템.

청구항 4

청구항 3에 있어서,

상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드가 최상위에 위치하면서, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB가 아직 생성되지 않은 경우, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB를 생성하는 것을 특징으로 하는 사고 원인 추적 시스템.

청구항 5

청구항 4에 있어서,

상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 것이 아니거나, 베이스 노드인 AND 게이트 노드가 최상위에 위치하면서, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB가 이미 생성되어 있으면, 하위 노드가 노멀 이벤트 타입(Normal Event Type) 노드인 경우, 형제 노드가 있으면 형제 노드 바로 뒤에 위치하는 P-S-C 구성요소만 있는 말단 CTFB를 생성하는 것을 특징으로 하는 사고 원인 추적 시스템.

청구항 6

청구항 5에 있어서,

상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 것이 아니거나, 베이스 노드인 AND 게이트 노드가 최상위에 위치하며, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB가 이미 생성되어 있으면서, 하위 노드가 노멀 이벤트 타입(Normal Event Type) 노드가 아닌 경우, 하위 노드가 OR 게이트 노드이면, 이 OR 게이트 노드의 하위 노드가 AND 게이트 노드인지 여부를 확인하고, AND 게이트 노드이면, 다른 블록을 뒤에 연결할 수 있는 I-N-S 구성요소와 P-S-C 구성요소를 모두 가지고 있는 CTFB를 생성하는 것을 특징으로 하는 사고 원인 추적 시스템.

청구항 7

청구항 6에 있어서,

상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 것이 아니거나, 베이스 노드인 AND 게이트 노드가 최상위에 위치하며, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB가 이미 생성되어 있으면서, 하위 노드가 노멀 이벤트 타입(Normal Event Type) 노드가 아닌 경우, 하위 노드가 OR 게이트 노드이면, 이 OR 게이트 노드의 하위 노드가 AND 게이트 노드인지 여부를 확인하고, AND 게이트 노드가 아니면, 베이스 노드인 AND 게이트의 서브트리를 모두 한 블록으로 보는 I-N-S 구성요소와 P-S-C 구성요소를 모두 가지고 있는 말단 CTFB를 생성하는 것을 특징으로 하는 사고 원인 추적 시스템.

청구항 8

청구항 7에 있어서,

상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드의 자식 노드에 대하여 CTFB를 모두 생성하면, FT 노드들을 모두 탐색할 때까지 노드를 순차적으로 탐색하고, FT 노드들을 모두 탐색하면, FMEA 스케마틱을 분석하고, 각 노드에서 발생할 수 있는 이벤트를 유발하는 결함 유형(Failure mode) 및 원인 요소(Causal factor)를 FMEA 분석 결과의 목록과 비교하여 연관성을 확인하고, FMEA에서 가져온 상세한 결함 유형 및 원인 요소 정보를 P-S-C 구성요소에 저장하는 것을 특징으로 하는 사고 원인 추적 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 지능형 로봇, 자율주행 자동차와 같은 지능형 자율 시스템에 관한 것으로서, 더욱 상세하게는 지능형 자율 시스템의 개발 과정에서 매우 엄밀한 결함분석기법을 적용하여, 시스템들이 협업 운영되는 과정에서 발생할 수 있는 결함을 사전에 제거하기 위한 원인 추적 분석 기법을 제공하는 시스템에 관한 것이다.

배경 기술

[0003] 지능형 로봇이나 자율주행 자동차와 같은 지능형 자율 시스템의 안전한 운영을 위한 다양한 시뮬레이션이 이미

상용화되고 이용되고 있다. 이러한 시뮬레이션들의 최대 장점은 잘 알려지지 않은 운영 환경에 대한 시스템의 반응을 사전에 모니터할 수 있다는 점이다.

[0004] 그러나 이러한 경우에 있어서는 이기종의 지능형 자율 시스템 간의 협업을 이루기 위해 빈번한 시뮬레이션을 수행해야 하는 문제가 있으며, 또한 기존 시스템의 일부 구성요소가 바뀌었을 경우, 처음부터 다시 시뮬레이션을 실행해 협업을 이루기까지 많은 안전 분석 시간이 소요되어, 의뢰인의 시간적인 손실뿐만 아니라 금전적인 비용도 증가시키게 된다는 문제점이 있다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 대한민국 공개특허 10-2017-0077332

발명의 내용

해결하려는 과제

[0007] 본 발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로서, 본 발명은 시스템 폴트(fault) 및 기능 불량(malfunction)의 원인을 추적하고, 추적된 영향을 바탕으로 엔지니어가 조치하는데 필요한 비용을 최소한으로 하는 지능형 자율 시스템에서의 결합 분석 기법을 이용한 사고 원인 추적 시스템을 제공하는데 그 목적이 있다.

[0008] 즉, 본 발명은 안전 엔지니어(Safety Engineer)가 지능형 자율 시스템들의 협업 시 발생할 수 있는 의도되지 않은 이벤트를 가동(Operation) 전에 최대한 많이 분석하고, 최소한의 비용으로 조치를 취할 수 있도록 부분별 안전성 평가(Safety Evaluation)가 가능한 형태의 분석 방법을 제공하는 것이다.

[0009] 본 발명의 목적은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0011] 이와 같은 목적을 달성하기 위한 본 발명의 지능형 자율 시스템에서의 사고 원인을 추적하기 위한 사고 원인 추적 시스템에서, 각 결합분석 기법 산출물에 대한 메타모델 생성, 구성요소 매핑, 그리고 관계를 식별하기 위한 지식 정보가 저장되어 있는 원인 추적 관계 온톨로지, 상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 지능형 자율 시스템을 구성하는 각 시스템에서 수행된 결합분석 기법 별 결합분석의 산출물을 입력받고, 이에 대한 메타모델을 클래스 다이어그램으로 생성하고, 원인 추적 분석에 사용할 메타모델 요소를 생성하기 위한 메타모델 생성기, 상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 각 기법의 메타모델 요소와 원인 추적 분석에 사용할 메타모델 요소를 결합하고, 이에 대한 CTFB(Cause Tracing Functional Block)을 생성하기 위한 구성요소 매핑, 상기 원인 추적 관계 온톨로지에 저장된 지식 정보를 읽어들이고, 각 기법의 메타모델 요소들이 가지고 있는 정보들의 관계를 설정하고, 부분 안정성 평가가 가능하도록 이펙트 분류 기준에 관계의 영향을 지정하고, 이벤트가 트리거 될 때, 블록 내부 또는 블록 간에 발생하는 이펙트 정보를 토대로 원인추적분석을 수행하는 구성요소 관계 식별기, 상기 구성요소 관계 식별기를 통해 상호 영향 관계가 정의된 두 시스템 간에 의도되지 않은 이벤트에 대한 추적정보를 입력으로, 원인을 유발하는 노드와 이펙트 명을 CTFBD(Cause Tracing Functional Block Diagram)에 나타내는 경로 탐색기를 포함한다.

[0012] 상기 메타모델 생성기는 FTA(Fault Tree Analysis)와 FMEA(Failure Mode and Effects Analysis)를 지정된 메타모델에 맞게 생성하고, 이 두 메타모델 객체를 입력으로 받는 CTA(Cause Tracing Analysis) 메타모델 객체를 생성할 수 있다.

[0013] 상기 구성요소 매핑은 FT(Fault Tree) 스케마틱(Schematic)을 분석하고, 노드를 순차적으로 탐색하여 CTFB를 생성하되, AND 게이트 노드를 베이스 노드로 하여 CTFB를 생성하기 위한 트리를 탐색할 수 있다.

[0014] 상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드가 최상위에 위치하면서, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB가 아직 생성되지 않은 경우, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB를 생성할 수 있다.

- [0015] 상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 것이 아니거나, 베이스 노드인 AND 게이트 노드가 최상위에 위치하면서, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB가 이미 생성되어 있으면, 하위 노드가 노멀 이벤트 타입(Normal Event Type) 노드인 경우, 형제 노드가 있으면 형제 노드 바로 뒤에 위치하는 P-S-C 구성요소만 있는 말단 CTFB를 생성할 수 있다.
- [0016] 상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 것이 아니거나, 베이스 노드인 AND 게이트 노드가 최상위에 위치하며, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB가 이미 생성되어 있으면서, 하위 노드가 노멀 이벤트 타입(Normal Event Type) 노드가 아닌 경우, 하위 노드가 OR 게이트 노드이면, 이 OR 게이트 노드의 하위 노드가 AND 게이트 노드인지 여부를 확인하고, AND 게이트 노드이면, 다른 블록을 뒤에 연결할 수 있는 I-N-S 구성요소와 P-S-C 구성요소를 모두 가지고 있는 CTFB를 생성할 수 있다.
- [0017] 상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 것이 아니거나, 베이스 노드인 AND 게이트 노드가 최상위에 위치하며, I-N-S 구성요소만 있고 다른 블록을 연결할 수 있는 CTFB가 이미 생성되어 있으면서, 하위 노드가 노멀 이벤트 타입(Normal Event Type) 노드가 아닌 경우, 하위 노드가 OR 게이트 노드이면, 이 OR 게이트 노드의 하위 노드가 AND 게이트 노드인지 여부를 확인하고, AND 게이트 노드가 아니면, 베이스 노드인 AND 게이트의 서브트리들 모두 한 블록으로 보는 I-N-S 구성요소와 P-S-C 구성요소를 모두 가지고 있는 말단 CTFB를 생성할 수 있다.
- [0018] 상기 구성요소 매핑은 베이스 노드인 AND 게이트 노드의 자식 노드에 대하여 CTFB를 모두 생성하면, FT 노드들을 모두 탐색할 때까지 노드를 순차적으로 탐색하고, FT 노드들을 모두 탐색하면, FMEA 스케마틱을 분석하고, 각 노드에서 발생할 수 있는 이벤트를 유발하는 결함 유형(Failure mode) 및 원인 요소(Causal factor)를 FMEA 분석 결과의 목록과 비교하여 연관성을 확인하고, FMEA에서 가져온 상세한 결함 유형 및 원인 요소 정보를 P-S-C 구성요소에 저장할 수 있다.

발명의 효과

- [0019] 본 발명에 의하면, 지능형 자율 시스템을 개발하기 위한 분석 및 설계단계에서 사용된 결함 분석 기법을 그대로 이용하여 사고 원인 추적 시스템을 제공함으로써, 안전 엔지니어(Safety Engineer)가 이벤트를 발생시킨 시스템의 원인 분석을 블록 별로 즉석에서 수행할 수 있어서, 보다 편리하고 신속하게 시스템의 원인을 분석할 수 있는 효과가 있다.
- [0020] 또한, 본 발명에 의하면 다른 결함분석기법 간의 결함 사용을 제공함으로써, 기존의 결함분석기법을 이용하여 안전 분석이 수행된 시스템에서도 별다른 프로세스 없이 사용할 수 있으므로, 유지보수를 위한 노력과 비용을 절감할 수 있는 효과가 있다.

도면의 간단한 설명

- [0022] 도 1은 본 발명의 일 실시예에 따른 사고 원인 추적 시스템에 대한 정적 구조를 나타내는 도면이다.
- 도 2 내지 도 4는 본 발명의 일 실시예에 따른 메타모델 생성기의 메타 모델 구조 생성을 나타내는 도면이다.
- 도 5는 본 발명의 일 실시예에 따른 사고 원인 추적 시스템의 전반적인 동작 절차를 나타내는 흐름도이다.
- 도 6은 본 발명의 일 실시예에 따른 구성요소 관계의 정의 기준을 나타내는 도면이다.
- 도 7은 본 발명의 일 실시예에 따른 원인 추적 관계 정보로 활용하기 위한 구성요소 간의 스키마 변환을 나타내는 도면이다.
- 도 8은 본 발명의 일 실시예에 따른 각 노드에서 발생한 원인이 다른 노드에 미치는 영향을 나타내는 도면이다.
- 도 9 내지 도 12는 본 발명의 일 실시예에 따른 경로 탐색 및 원인 분석을 예시한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0023] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.
- [0024] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가

아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서 상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

- [0025] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 갖고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 갖는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0026] 또한, 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조 부호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다. 본 발명을 설명함에 있어서 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다.
- [0027] 본 발명은 지능형 자율 시스템 개발 과정에서 협업 시 발생하는 시스템 결함 및 기능 불량(Malfunction) 분석 평가를 효과적으로 지원하기 위하여, 부분적 평가가 가능한 원인 추적 분석을 제공하고, 기존 결함분석기법과의 결합 사용을 가능하게 하여 시스템 개발에서 수행되는 안전성 평가 비용을 최소한으로 하는 것이다.
- [0028] 도 1은 본 발명의 일 실시예에 따른 사고 원인 추적 시스템에 대한 정적 구조를 나타내는 도면이다.
- [0029] 도 1을 참조하면, 사고 원인 추적 시스템(100)은 각 시스템(Application1, 2)에서 수행된 결함분석의 산출물(Fault Analysis Schematic1, 2)을 입력받아 원인추적 분석 및 평가를 제공한다.
- [0030] 사고 원인 추적 시스템(100)은 메타모델 생성기(Meta-model Generator)(110), 구성요소 매핑(Element Mapper)(120), 구성요소 관계 식별기(Element Relation Identifier)(130), 경로 탐색기(Path Navigator)(140), 원인 추적 관계 온톨로지(Cause Tracing Relation Ontology)(150)를 포함한다.
- [0031] 메타모델 생성기(110)는 결함 트리(Fault Tree)와 같은 기법의 산출물을 입력으로 받아서, 이에 대한 메타모델을 클래스 다이어그램으로 생성하는 역할을 한다. 또한, 원인 추적 분석에 사용할 메타모델을 추가로 생성한다.
- [0032] 구성요소 매핑(120)은 각 기법의 메타모델 요소와 원인 추적 분석에 사용할 메타모델 요소를 결합하고, 원인 추적 분석에 사용할 CTFB(Cause Tracing Functional Block)을 인스턴스화 한다.
- [0033] 구성요소 관계 식별기(130)는 각 기법의 메타모델 요소들이 가지고 있는 정보들의 관계를 설정하는 역할을 한다. 관계가 식별되었으면, 부분 안전성 평가가 가능하도록 하는 이펙트(Effect) 분류 기준에 관계의 영향을 지정한다. 이펙트(Effect)는 {Immediate Effect, System Effect, Command Effect, Secondary Effect}로 구분한다. 이벤트가 트리거 될 때, 블록 내부 또는 블록 간에 발생하는 이펙트(Effect) 정보를 토대로 원인추적분석을 수행한다.
- [0034] 경로 탐색기(140)는 구성요소 관계 식별기(130)를 통해, 상호 영향 관계가 정의된 두 시스템 간에 의도되지 않은 이벤트에 대한 추적정보를 입력으로, 원인(Cause)를 유발하는 노드와 이펙트(Effect) 명을 CTFBD(Cause Tracing Functional Block Diagram)에 나타낸다.
- [0035] 원인 추적 관계 온톨로지(150)는 각 기법 산출물에 대한 메타모델 생성, 구성요소 매핑, 그리고 관계를 식별하기 위한 지식 정보가 저장되어 있다.
- [0037] 도 2 내지 도 4는 본 발명의 일 실시예에 따른 메타모델 생성기의 메타 모델 구조 생성을 나타내는 도면이다.
- [0038] 도 2는 결함트리분석(FTA, Fault Tree Analysis) 기법으로부터 얻은 메타모델이다. 여기서 각 컴포넌트들이 의미하는 바는 다음과 같다.
- [0039] FtaSystem은 시스템에서 발생할 수 있는 이벤트들의 분석정보를 가지고 있는 컴포넌트이다.
- [0040] TopEvent는 사고를 유발하는 최상위 이벤트이다.
- [0041] PrimaryEvent는 시스템의 사고 발생에서 가장 처음 단계에 위치하는 기본적인 이벤트로, 식별된 기본이벤트인 BasicEvent와 추후에 명세될 기본이벤트인 UndevelopedEvent로 구분할 수 있다. 이 컴포넌트에서는 서브시스템의 디자인 정보를 가지고 있으며, 공통 원인이 두 개 이상의 시스템 컴포넌트를 고장에 이르게 하는 Ccf(Common Cause Failure) 모델과 개별 실패 모델인 FailureModel의 정보를 가지고 있다.

- [0042] DerivedEvent는 PrimaryEvent에서 단계가 거둬지는 이벤트로, 리스크 관리 평가가 수행 되어야 하는 이벤트로서, 리스크 관리에 필요한 DerivedResult, Importance, MinimalCutSet 정보를 가지고 있다.
- [0043] Gate는 논리적인 추론을 가능하게 하는 컴포넌트로서, AndGate, OrGate, VoteGate로 구분할 수 있다.
- [0044] 도 3은 결합유형 및 영향 분석(FMEA, Failure Mode and Effects Analysis) 기법의 메타 모델을 나타내며, 이 메타모델의 구성 요소에 대한 의미는 다음과 같다.
- [0045] FmeaSystem는 기능적 및 구조적 시스템의 결합분석 정보를 가지고 있는 컴포넌트이다.
- [0046] Block은 결합 및 영향 분석을 할 수 있는 최소 단위로, 관점에 따라 구조적 관점이면 SubsystemBlock, 기능적 관점이면 FunctionBlock으로 분류된다. 이 블록은 분석정보인 BlockData와 시스템 실패모드인 FailureMode를 가지고 있다.
- [0047] SystemBlock은 블록들 중에서 시스템 측면을 드러낼 수 있는 블록이다.
- [0048] FailureMode는 결합 분석의 근거가 되는 상태로, 관점에 따라 분류된 블록들의 특성에 따라 SystemBlock에서는 SystemFailureMode, SubsystemBlock에서는 SubSystemFailureMode, FunctionBlock에서는 FunctionFailureMode를 갖는다.
- [0049] Phrase는 블록으로 표현되는 정보에 대한 상세한 구문이다.
- [0050] 도 4는 CTA(Cause Tracing Analysis)를 통해 얻은 메타모델간 통합을 통해 얻어진 메타모델이다. 이는 메타모델의 구성요소들에 대한 의미적 관계를 기반으로 생성된다.
- [0051] 본 발명에서는 결합분석에서 해저드(Hazard)를 유발시키는 이벤트 식별을 가능하게 하는 대표적인 연역적 추론 기법인 FTA(Fault Tree Analysis)와, 자세한 이펙트(Effect) 분석을 가능하게 하는 대표적인 귀납적 추론 기법인 FMEA(Failure Mode and Effects Analysis)를 결합하고 추적할 수 있도록 Cause Tracing 메타모델을 제안한다. 이는 FTA에서 수행하기 어려운 이펙트(Effect) 분석과 FMEA에서 수행하기 어려운 이벤트(Event) 식별을, 원인 추적을 통하여 상관관계로 맺었다는 특징이 있다. 그리고, Derived Event는 FTA와의 결합을 가능하게 하고, Item 및 Failure Mode는 FMEA와의 결합을 가능하게 한다.
- [0052]
- [0053] 도 5는 본 발명의 일 실시예에 따른 사고 원인 추적 시스템의 전반적인 동작 절차를 나타내는 흐름도이다.
- [0054] 도 5를 참조하면, 메타모델 생성기(110)에서 입력받은 두 어플리케이션의 Fault Analysis Schematics를 기반으로 결합분석기법 간의 결합을 위한 메타모델 객체를 생성하며, FMEA와 FTA를 예제로 사용한다(S101~S105).
- [0055] 메타모델 생성기(110)는 두 어플리케이션으로부터 Fault Analysis Schematics을 입력받고(S101), 두 어플리케이션을 각각 분석하는데 사용한 FMEA와 FTA를 지정된 메타모델에 맞게 생성하고(S103), 이 두 메타모델 객체를 입력으로 받는 CTA(Cause Tracing Analysis) 메타모델 객체를 생성하여, 각 기법 간의 원인추적 분석을 가능하게 한다(S105).
- [0056] 그리고, 구성요소 매핑(120)는 FTA로 할 수 있는 연역적 추적과 FMEA로 할 수 있는 귀납적 추적 모두를 가능하게 하는 CTFB(Cause Tracing Functional Block)을 해당 알고리즘을 통해 생성하고, FTA의 Hazard Analysis 정보와 FMEA의 Hazard Analysis 정보를 CTFB를 구성하고 있는 두 컴포넌트 I-N-S와 P-S-C에 저장한다(S201~S237).
- [0057] 여기서 I-N-S 구성요소는 “What is Immediate, Necessary, and Sufficient to cause the event?” 의 개념을 의미하며, 즉각적인 이벤트에 대한 분석을 가능하게 한다. 그리고, P-S-C 구성요소는 “What are the Primary, Secondary, and Command causes of the event?” 의 개념을 의미하며, 구체적인 Causal factors의 분석을 가능하게 한다. 이들은 SS-SC 개념을 통해 “state of the system” (SS)를 I-N-S 로직의 입력과 AND 게이트 유형으로 fault event를 분류하고, “state of the component” (SC)를 P-S-C 입력과 OR 게이트의 유형으로 fault event를 분류하게 된다. 이는 FTA에서 사용하는 개념들로, 본 발명에서 제시하는 원인 추적 기법에서 이러한 개념들의 지원은 의도되지 않은 최상위 이벤트의 원인에 대한 논리적인 추론을 가능하게 한다.
- [0058] 이러한 구성요소 매핑(120)에서의 일련의 과정(S201~S237)을 구체적으로 설명하면 다음과 같다.
- [0059] 구성요소 매핑(120)는 FT Schematic을 분석하고(S201), 노드를 순차적으로 탐색한다(S203).

- [0060] 먼저, 베이스 노드가 AND 게이트 노드이면, CTFB를 만들기 위한 트리를 탐색한다(S205, S207).
- [0061] 탐색 결과, 베이스 노드인 AND 게이트 노드가 가장 상위에 위치한 것이면(S209), I-N-S 구성요소만 있고 다른 블록을 뒤에 연결할 수 있는 CTFB를 이미 생성하였는지 여부를 확인한다(S214).
- [0062] I-N-S 구성요소만 있고 다른 블록을 뒤에 연결할 수 있는 CTFB가 없으면, I-N-S 구성요소만 있고 다른 블록을 뒤에 연결할 수 있는 CTFB를 생성한다(S215).
- [0063] 베이스 노드인 AND 게이트 노드가 가장 상위에 위치한 것이 아니거나(S209), 베이스 노드인 AND 게이트 노드가 가장 상위에 위치한 것이면서 I-N-S 구성요소만 있고 다른 블록을 뒤에 연결할 수 있는 CTFB가 이미 생성되었으면(S214), 하위 노드가 Normal Event Type 노드인지 여부를 확인한다(S211).
- [0064] 하위 노드가 Normal Event Type 노드인 경우, 형제 노드가 있으면 형제 노드 바로 뒤에 위치하는 특성을 가진 P-S-C 구성요소만 있는 말단 CTFB를 생성한다(S217).
- [0065] 하위 노드가 Normal Event Type 노드가 아닌 경우, 하위 노드가 OR 게이트 노드인지 여부를 확인한다(S213).
- [0066] 하위 노드가 OR 게이트 노드이면, 이 OR 게이트 노드의 하위 노드가 AND 게이트 노드인지 여부를 확인한다(S221).
- [0067] OR 게이트 노드의 하위 노드가 AND 게이트 노드이면, 다른 블록을 뒤에 연결할 수 있는 I-N-S 구성요소와 P-S-C 구성요소를 모두 가지고 있는 CTFB를 생성한다(S223).
- [0068] 반면, OR 게이트 노드의 하위 노드가 AND 게이트 노드가 아니면, 베이스 노드인 AND 게이트의 서브트리를 모두 한 블록으로 보는 I-N-S 구성요소와 P-S-C 구성요소를 모두 가지고 있는 말단 CTFB를 생성한다(S225).
- [0069] S213 단계에서 하위 노드가 OR 게이트 노드가 아니면, 추후에 만들어질 CTFB의 특이사항으로 들어갈 네 가지 타입(Primary Failure, Secondary Failure, Condition, Transfer)을 구분하여 저장한다(S219).
- [0070] 여기서, S207 단계부터 S225 단계까지의 과정을 Loop A라 명명하기로 한다.
- [0071] 다음, 베이스 노드인 AND 게이트 노드의 자식 노드에 대해 모두 CTFB를 생성할 때까지 Loop A를 반복한다(S227, S229).
- [0072] 베이스 노드인 AND 게이트 노드의 자식 노드에 대해 모두 CTFB를 생성하면, FT 노드들을 모두 탐색할 때까지 노드를 순차적으로 탐색한다(S231).
- [0073] FT 노드들을 모두 탐색하면, FMEA Schematic을 분석한다(S233).
- [0074] 그리고, 각 노드에서 발생할 수 있는 Event를 유발하는 Failure mode 및 Casual factor를 FMEA 분석 결과의 목록과 비교하여 연관성을 확인한다(S235).
- [0075] 그리고, FMEA에서 가져온 상세한 Failure mode 및 Casual factor 정보를 P-S-C 구성요소에 저장한다(S237).
- [0076] 다음, 이벤트 트리거 소프트웨어 스크립트를 실행한다(S239).
- [0077] 그리고, 구성요소 관계 식별기(130)는 특정 이벤트를 입력으로 받았을 때, 네 가지로 분류된 이펙트(Effect)에 따른 원인 경로를 추적하게 된다(S301). 이펙트에는 두 어플리케이션의 시스템 상태(SS)로만 분석하는 Command Effect와 컴포넌트 상태(SC)의 분석이 되어야 할 수 있는 Immediate Effect, System Effect, Secondary Effect가 있다.
- [0078] 그리고, 경로 탐색기(140)는 사용자에게 CTFB와 그들의 연결 관계를 보여주는 CTFBD(Cause Tracing Functional Block Diagram)을 내부가 보이는 화이트박스 또는 내부가 보이지 않은 블랙박스 중 하나를 선택해서 보여준다(S401). 여기서, 네 가지 Effect는 하이라이트로 표시되어 사용자에게 보여 지게 된다.
- [0079] 그리고, 경로 탐색기(140)는 경로를 추적하여 CTFBD에 표시한다(403).
- [0081] 도 6은 본 발명의 일 실시예에 따른 구성요소 관계의 정의 기준을 나타내는 도면이다.
- [0082] 도 6은 원인 추적을 위해 모델링된 CTA(Cause Tracing Analysis) 기법에서 사용하는 CTFB(Cause Tracing Functional Block)의 구성요소가 기존 FTA 기법의 구성요소와 의미적으로 같은 부분을 매핑하면서, CTFB의 구성요소들을 통해 평가가 가능한 평가 요소 항목을 보여준다.

- [0083] 도 6을 참조하면, 기능적인 부분만 고려한 FTA 기법은 Functional Safety를 평가할 수 있고, 시스템 분석에 베이스를 둔 FTA는 FMEA와 함께 System Safety를 평가할 수 있다. 그리고, 디테일 분석에 사용되는 FMEA 자체로는 Reliability를 평가할 수 있다. 따라서 FTA와 FMEA 기법 간의 연결로 인한 Hazard Analysis의 지원 범위는 Functional Hazard Analysis, System Design Hazard Analysis, Subsystem Design Hazard Analysis이다.
- [0084] 본 발명에서 제시하는 Cause Tracing Analysis Technique에서는 FTA와 FMEA 연결로 인한 Hazard Analysis Type의 SD-HAT(System Design Hazard Analysis Technique) 및 DD-HAT(Detail Design Hazard Analysis Technique) 지원 외에 협업 환경에서 시스템 간 미션 수행의 평가를 위한 Functional Hazard Analysis를 추가 지원하여, 지능형 자율 시스템의 기능 수행의 영향으로부터 발생할 수 있는 Fault를 분석하고 평가할 수 있도록 한다.
- [0085] 도 6에 도시되어 있는 Evaluation Factor의 고려는 Safety Engineer가 안전을 위한 조치를 수준별로 취할 수 있도록 한다.
- [0086] Immediate Effect 또는 Secondary Effect로 발생한 의도되지 않은 이벤트는 Reliability 측면에서 조치를 취할 수 있다. “서버의 다운 시에 발생할 수 있는 Hazards를 서버 Redundancy를 허용하여 예방” 이 대표적인 Reliability Measures이다.
- [0087] System Effect로 발생한 의도되지 않은 이벤트는 Safety 측면에서 조치를 취할 수 있다. “기존시스템의 오작동을 예방하기 위한 다른 시스템 및 소프트웨어 추가” 가 대표적인 Safety Measures이다.
- [0088] Command Effect로 발생한 의도되지 않은 이벤트는 Functional Safety 측면에서 조치를 취할 수 있다. “시스템들의 협업 환경에서, 의도한 태스크를 수행하도록 각 시스템의 시간 제약사항 설정 및 시간 지연” 이 대표적인 Functional Safety Measures이다.
- [0089] 이처럼, 본 발명에서는 각 Cause Tracing Functional Block 및 Effect를 통해 원인 추적을 가능하게 하고 조치 수준을 결정할 수 있도록 한다.
- [0091] 도 7은 본 발명의 일 실시예에 따른 원인 추적 관계 정보로 활용하기 위한 구성요소 간의 스키마 변환을 나타내는 도면이다.
- [0092] 도 7을 참조하면, 본 발명에서 제시하는 CTFBD(Cause Tracing Functional Block Diagram)에서 FTA의 논리적 추론을 가능하게 함으로써, 각 블록별 Effect 분석을 가능하게 한다.
- [0093] 그리고, 시스템 상태를 나타내는 하나의 I-N-S 구성요소가 여러 블록들에서 겹치게 분석되도록 허용함으로써, 시스템의 특정 상태에서 나오는 Event의 추적을 가능하도록 한다. 예를 들어, I-N-S 구성요소와 같은 의미로 매핑 되는 AND게이트 노드 “Pwr Output From Arm Sw” 는 Block5, Block6, Block7의 공통의 원인추적 분석대상 노드이며, 이곳에 의도되지 않은 이벤트가 발생했을 때, Block5, Block6에서 어떤 Effect로 기인한 것인지 또는 이 이벤트가 Block9의 I-N-S 노드의 이벤트 발생에 영향을 주는지도 추적할 수 있다.
- [0094] 시간의 흐름과 연관지어 생각해 볼 때, 최상위 노드는 가장 마지막에 발생하는 최종적인 이벤트가 되고 최하위 노드들은 가장 먼저 발생하는 이벤트들이 된다. 여기서 Block1, Block3, Block9는 앞에서 정의한 CTFB(Cause Tracing Functional Block)과 다른 형태로 나타내었다.
- [0095] Block1과 Block3에서 특정 노드가 가지고 있는 Normal Event Type은 정상적인 시스템 작동의 일부로 발생할 것으로 예상되는 이벤트이다. 이 이벤트는 시스템 상태가 아닌 정상적인 시스템 작동의 일부로 나타나는 컴포넌트 상태에서 오는 이벤트이기 때문에 컴포넌트 상태와 매핑 되는 P-S-C 구성요소만 가지고 있다. Block9는 가장 최상위에 위치하는 최종적인 이벤트 이므로 시스템 상태와 매핑 되는 I-N-S 구성요소만 가지고 있다.
- [0096] 도 5 및 도 7을 참조하면, Block1과 Block3은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 노드가 아니면서, 하위 노드로 Normal event type 노드를 포함하고 있으며, 형제 노드가 있으므로, 형제 노드 바로 뒤에 위치하는 특성을 갖는 P-S-C 구성요소만 있는 말단 CTFB가 생성된다.
- [0097] Block2와 Block4는 베이스 노드인 AND 게이트 노드가 최상위에 위치한 노드가 아니면서, 하위 노드로 Normal event type 노드를 포함하고 있지 않으면서, 하위 노드가 OR 게이트 노드이고, 이 OR 게이트 노드의 하위 노드가 AND 게이트 노드가 아니므로, 베이스 노드인 AND 게이트의 서브트리 모두를 한 블록으로 보는 I-N-S 구성요소와 P-S-C 구성요소를 모두 갖고 있는 말단 CTFB가 생성된다.
- [0098] Block5는 베이스 노드인 AND 게이트 노드가 최상위에 위치한 노드가 아니면서, 하위 노드로 Normal event type 노드를 포함하고 있지 않으면서, 하위 노드가 OR 게이트 노드이고, 이 OR 게이트 노드의 하위 노드가 AND 게이트

트 노드이므로, 다른 블록을 뒤에 연결할 수 있는 I-N-S 구성요소와 P-S-C 구성요소를 모두 갖고 있는 CTBF가 생성된다.

- [0099] Block6는 베이스 노드인 AND 게이트 노드가 최상위에 위치한 노드가 아니면서, 하위 노드로 Normal event type 노드를 포함하고 있지 않으면서, 하위 노드가 OR 게이트 노드이고, 이 OR 게이트 노드의 하위 노드가 AND 게이트 노드가 아니므로, 베이스 노드인 AND 게이트의 서브트리 모두를 한 블록으로 보는 I-N-S 구성요소와 P-S-C 구성요소를 모두 갖고 있는 말단 CTBF가 생성된다.
- [0100] Block9은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 노드이므로, I-N-S 구성요소만 있고 다른 블록을 뒤에 연결할 수 있는 CTBF가 생성된다.
- [0101] Block7, Block8은 베이스 노드인 AND 게이트 노드가 최상위에 위치한 노드이면서, Block9에 의해 I-N-S 구성요소만 있고 다른 블록을 뒤에 연결할 수 있는 CTBF가 생성된 상태이며, 하위 노드로 Normal event type 노드를 포함하고 있지 않으면서, 하위 노드가 OR 게이트 노드이고, 이 OR 게이트 노드의 하위 노드가 AND 게이트 노드이므로, 다른 블록을 뒤에 연결할 수 있는 I-N-S 구성요소와 P-S-C 구성요소를 모두 갖고 있는 CTBF가 생성된다.
- [0103] 도 8은 본 발명의 일 실시예에 따른 각 노드에서 발생한 원인이 다른 노드에 미치는 영향을 나타내는 도면이다.
- [0104] 도 9 내지 도 12는 본 발명의 일 실시예에 따른 경로 탐색 및 원인 분석을 예시한 도면이다. 즉, 도 9 내지 도 12는 도 9에서 제시된 네 가지 이펙트에 대하여 경로 탐색기(140)에서 제공하는 경로 탐색 및 원인 분석을 각각 나타낸다.
- [0105] 도 9 내지 도 12에서 각 Effect에 대해 Cause 및 Event가 발생하는 위치와 실질적으로 Cause Tracing Functional Block Diagram 상에서 가시화되는 부분을 보여주며, 화이트 박스 가시화를 선택하면 Functional Block의 디테일한 내부 구조를 볼 수 있어 원인 추적을 엄격하게 수행할 수 있다.
- [0106] 도 8 내지 도 12를 참조하면, Immediate Effect는 DD-HAT(Detail Design Hazard Analysis Technique) 레벨에서, 하나의 컴포넌트 및 유닛에서 발생한 Failure Mode의 Effect가 Indenture Level 상위 관계에 있는 컴포넌트 및 유닛에 미치는 영향을 나타낸다. 이 Effect는 P-S-C 구성요소 안에서만 영향을 끼치며, 블록 이벤트로 바로 이어지지 않기 때문에 잠재적으로 System Effect에 기여해서 블록 이벤트로 나온다.
- [0107] System Effect는 SD-HAT(System Design Hazard Analysis Technique) 레벨에서 시스템 운영상에서 발생하며, 의도하지 않은 블록 이벤트에 직접적으로 기여할 수 있는 영향을 나타내고, 블록의 P-S-C 구성요소에서 I-N-S 구성요소로 이어지는 방향성을 가지고 있다.
- [0108] Command Effect는 F-HAT(Function Hazard Analysis Technique) 레벨에서 미션을 위한 각 시스템의 기능 수행 단계에서 발생한다. 시간의 흐름을 고려한 상황에서 Command Effect는 하위 블록 i의 시스템 상태인 I-N-S 구성요소에서 다음 상위 블록인 i+1의 I-N-S 구성요소로 전달되는 메시지를 통해 분석될 수 있다.
- [0109] 마지막으로 Secondary Effect는 DD-HAT(Subsystem Design Hazard Analysis Technique) 레벨에서, i번째 블록에 해당하는 서브시스템이 가지고 있던 Causal factors가 i+1번째 블록에서 발생하는 이벤트에 기여하는 것으로, i+1번째 블록에서 Hazard로 이어지는 시스템의 이벤트가 i+1번째 블록 자체에 원인이 없는 것을 말한다. 메시지 교환으로 발생하는 Command Effect로도 원인 분석이 되지 않을 때, 이전 블록의 Causal factors가 다음 블록의 Causal factors로 내재되어 있을 것이라고 추정한다. i번째 블록의 P-S-C 구성요소에서 i번째 블록의 I-N-S, i+1번째 블록의 P-S-C 순으로 이어져 i+1번째 블록의 System Effect와 함께 Event 발생에 기여한다.
- [0110] 도 9는 Immediate Effect에 대한 경로 탐색 및 원인 분석을 나타낸다.
- [0111] 도 9에서 보는 바와 같이, Immediate Effect는 Failure mode가 시스템 Fault로 바로 이어지지 않으며, 잠재적으로 System Effect에 기여한다.
- [0112] 도 10은 System Effect에 대한 경로 탐색 및 원인 분석을 나타낸다.
- [0113] 도 10에서 보는 바와 같이, System Effect는 Failure mode에서 발생하는 Effect가 시스템 Fault로 이어지는 현상으로서, 이벤트를 발생시킨다.
- [0114] 도 11은 Secondary Effect에 대한 경로 탐색 및 원인 분석을 나타낸다.
- [0115] 도 11에서 보는 바와 같이, Block i의 Secondary Effect는 Block i+1의 Causal Factor로 내재되어 있어, Block

$i+1$ 에서 발생하는 Event에 기여될 수 있다.

[0116] 도 12는 Command Effect에 대한 경로 탐색 및 원인 분석을 나타낸다.

[0117] 도 12에서 보는 바와 같이, Block i 에서 Block $i+1$ 로 이어진 Command Effect는 Block i 의 기능 실행이 Block $i+1$ 의 기능제공 실패로 이어진다.

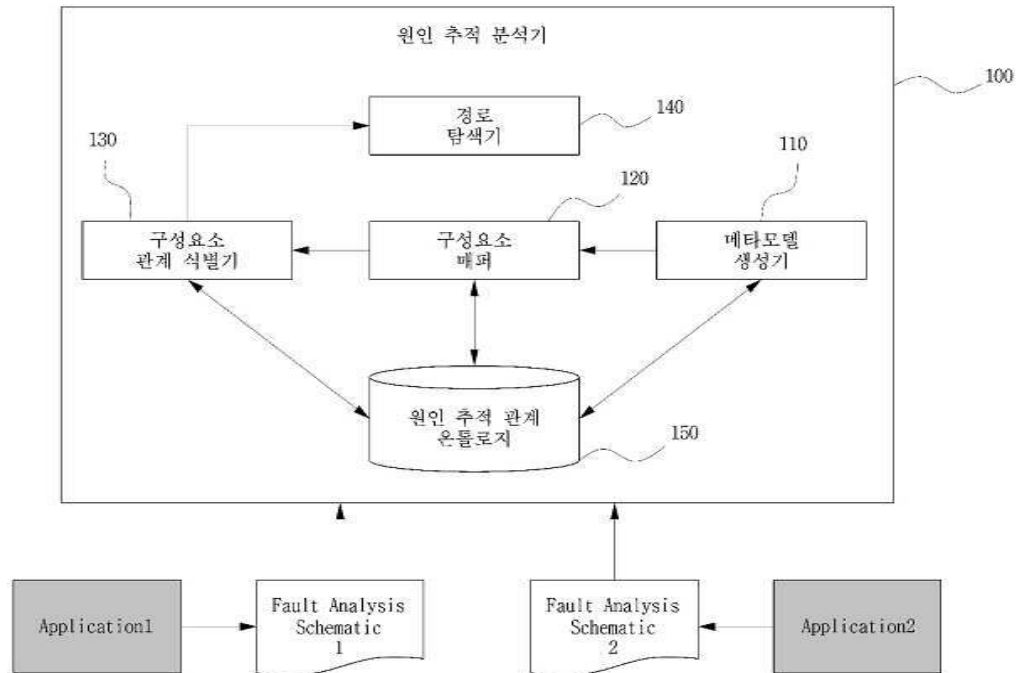
[0119] 이상 본 발명을 몇 가지 바람직한 실시예를 사용하여 설명하였으나, 이들 실시예는 예시적인 것이며 한정적인 것이 아니다. 본 발명이 속하는 기술분야에서 통상의 지식을 지닌 자라면 본 발명의 사상과 첨부된 특허청구범위에 제시된 권리범위에서 벗어나지 않으면서 다양한 변화와 수정을 가할 수 있음을 이해할 것이다.

부호의 설명

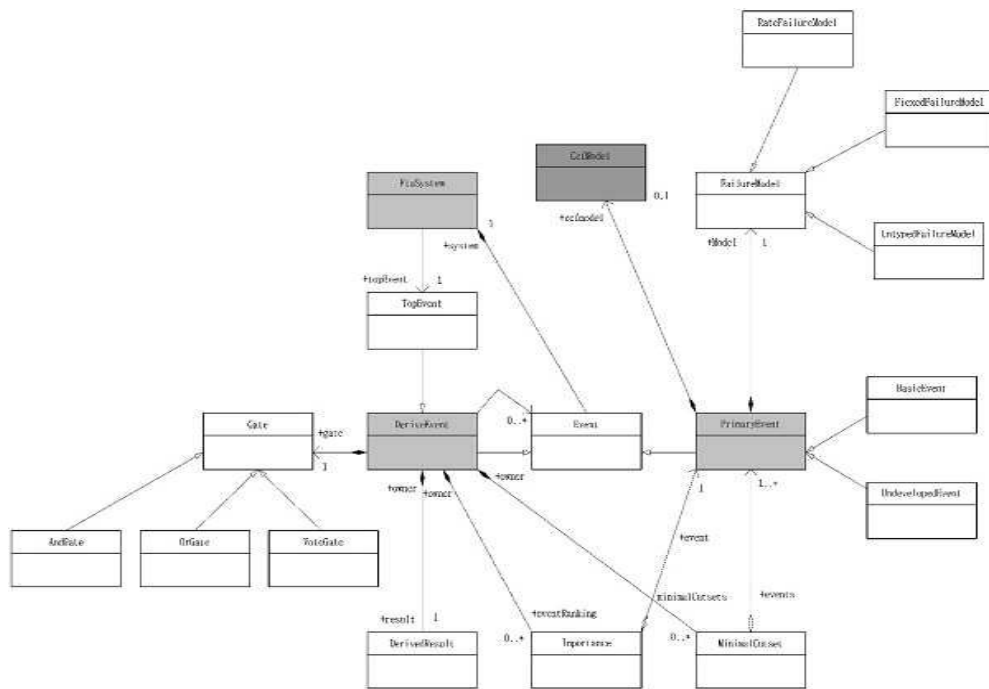
- | | | | | |
|--------|-----|--------------|-----|---------------|
| [0121] | 100 | 사고 원인 추적 시스템 | 110 | 메타모델 생성기 |
| | 120 | 구성요소 매핑 | 130 | 구성요소 관계 식별기 |
| | 140 | 경로 탐색기 | 150 | 원인 추적 관계 온톨로지 |

도면

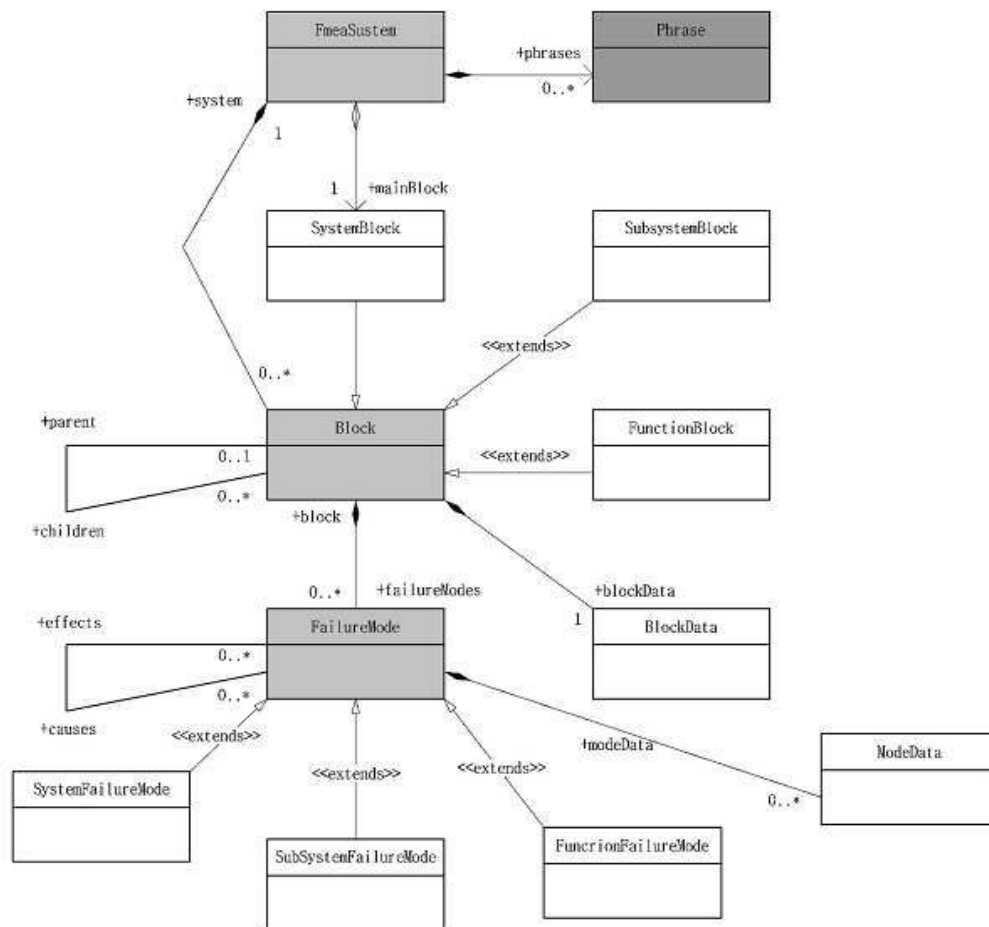
도면1



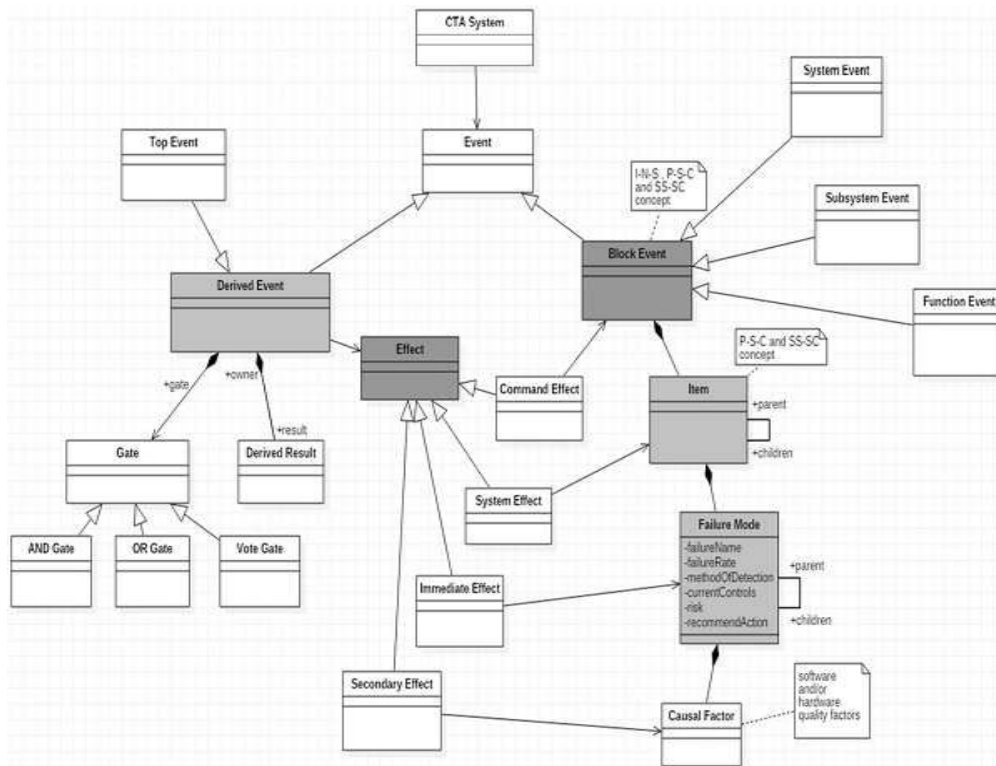
도면2



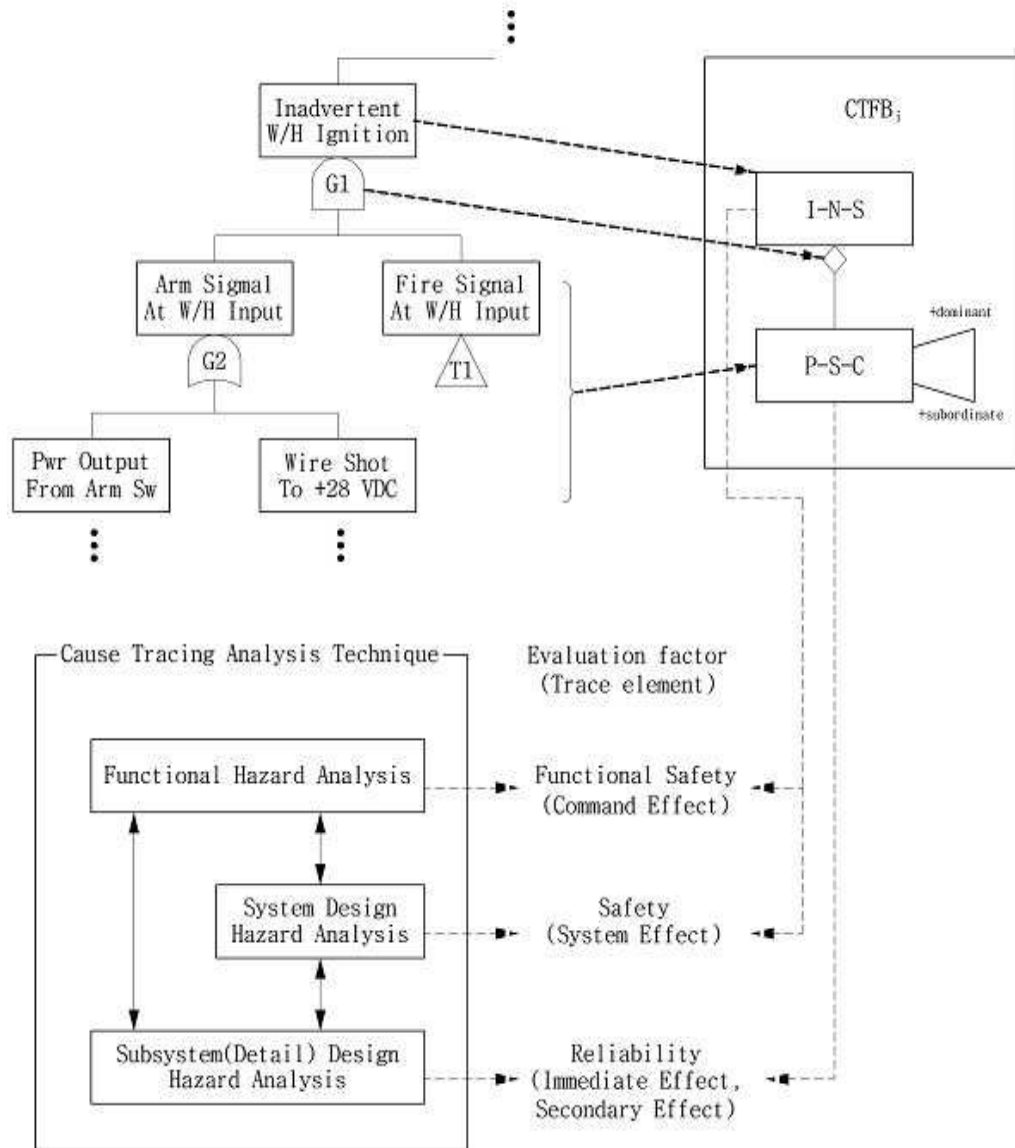
도면3



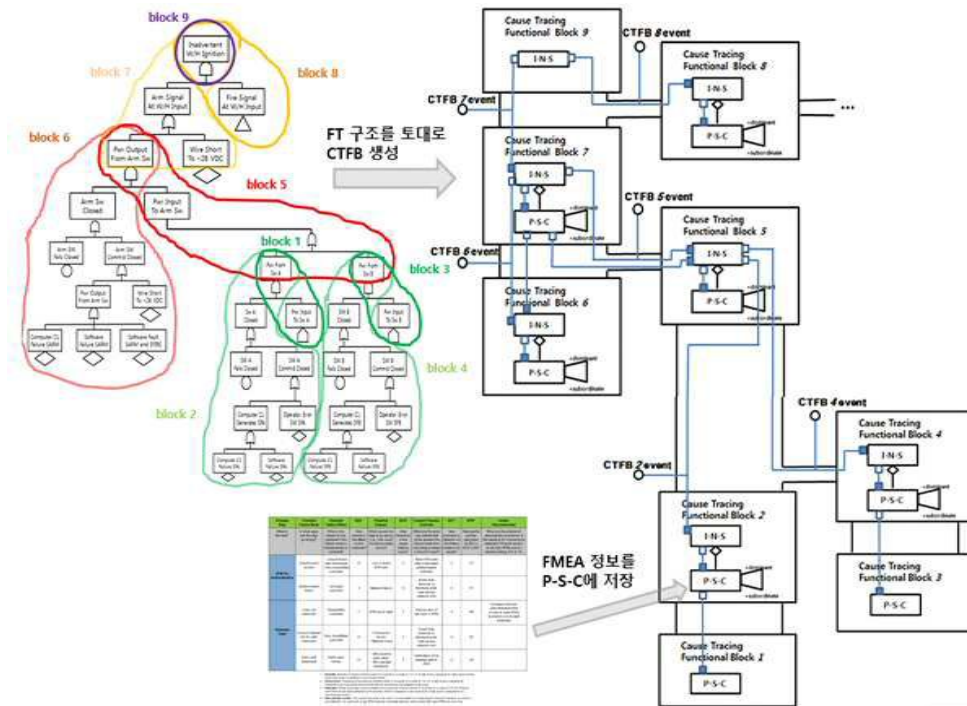
도면4



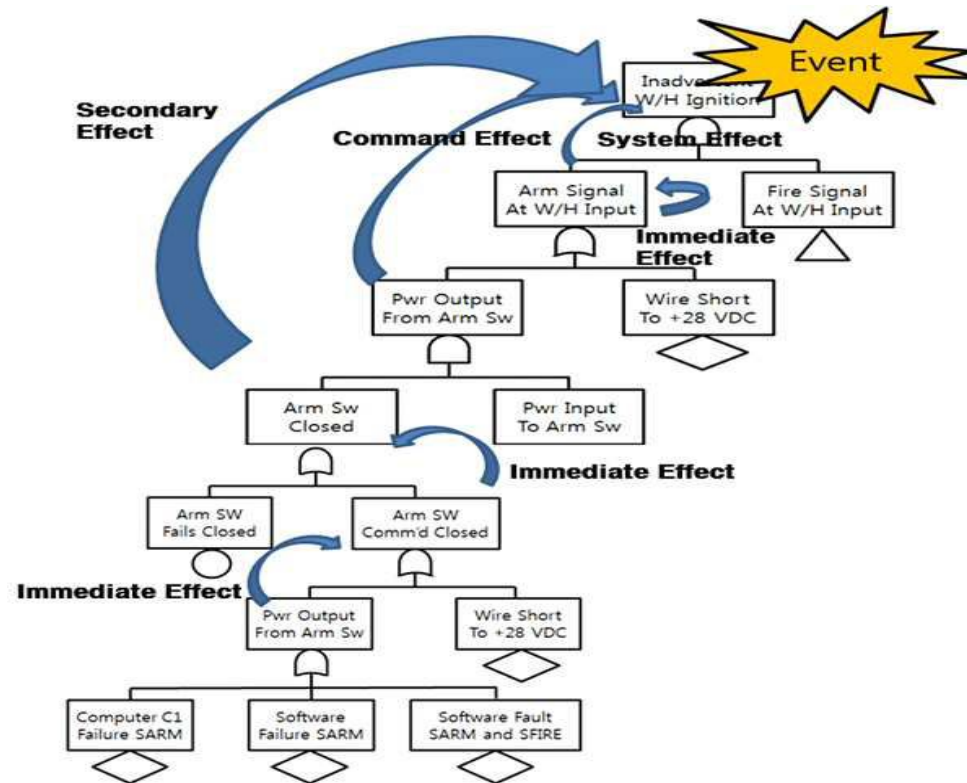
도면6



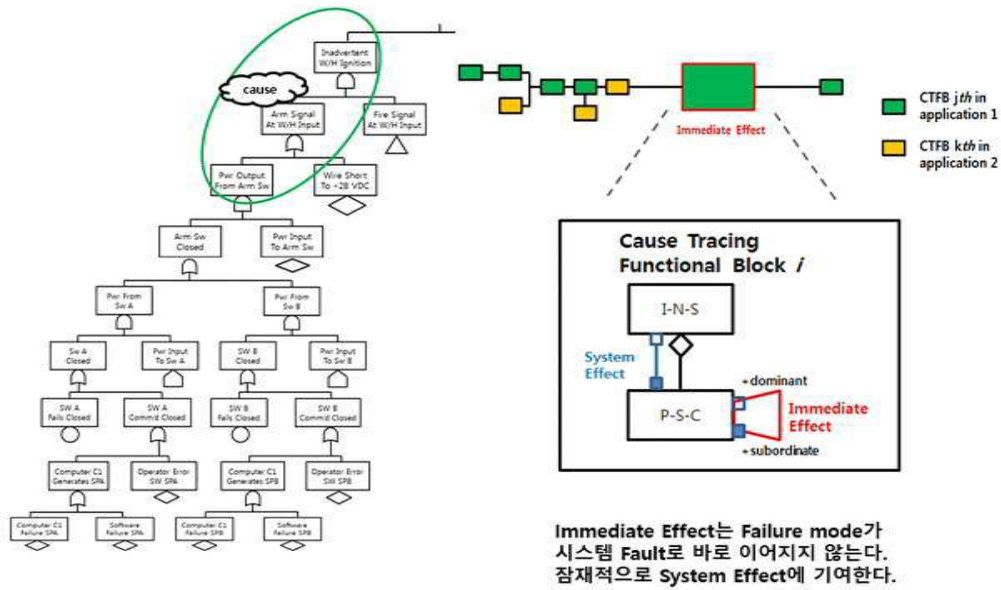
도면7



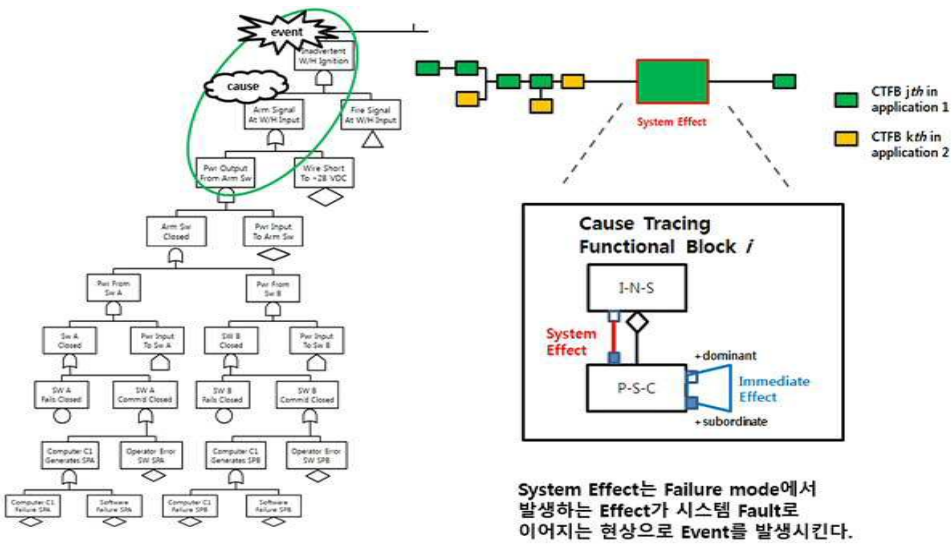
도면8



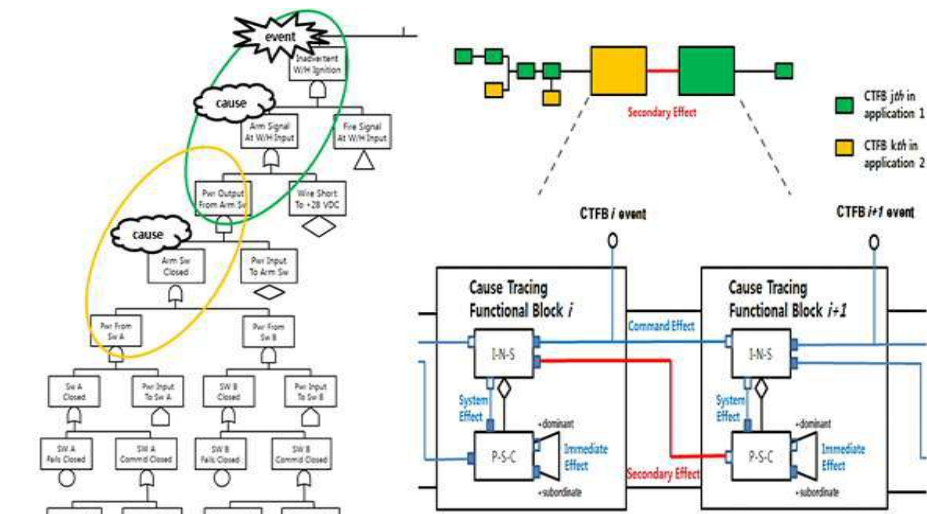
도면9



도면10

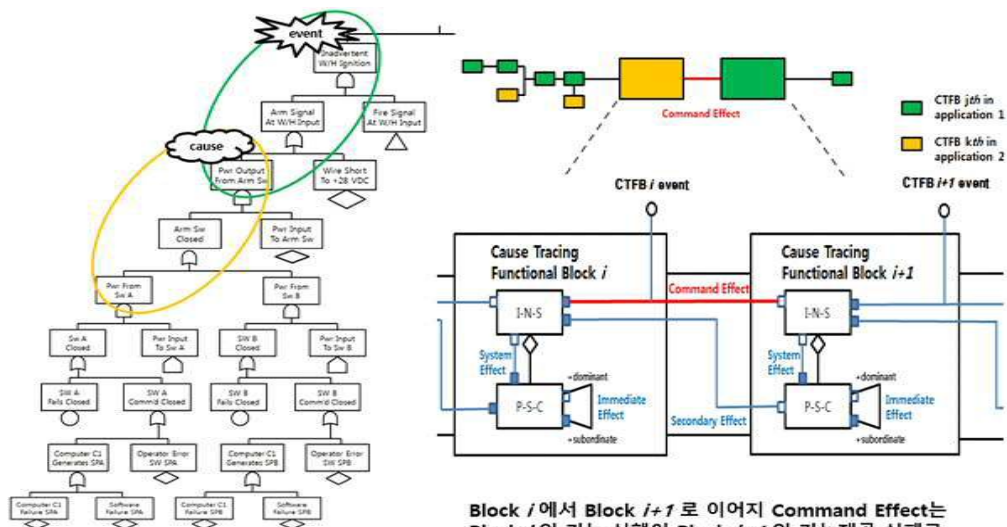


도면11



Block i 의 Secondary Effect는 Block $i+1$ 의 Causal Factor로 내재되어 있어, Block $i+1$ 에서 발생하는 Event에 기여될 수 있다.

도면12



Block i 에서 Block $i+1$ 로 이어지 Command Effect는 Block i 의 기능 실행이 Block $i+1$ 의 기능제공 실패로 이어진다.