J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

SysML을 이용한 STPA 기반의 위험원 분석 프로세스☆

Hazard Analysis Process Based on STPA Using SysML

최 나 연[1] 이 병 걸[2*]
Na-yeon Choi Byong-gul Lee

요 약

자동차, 원자력, 철도, 항공 등의 분야에서 발생하는 소프트웨어의 고장이나 사고는 바로 큰 재산 피해나 인명 피해로 연결될 수 있다. 이에 체계적이고도 효과적인 안전품질 관리의 필요성이 커지고 있으며, 최근 관련 산업 분야에서는 IEC 61508을 기반으로 안전 국제 표준이 제정되어 산업에 적용되고 있다. 국제 표준에서 명시하고 있는 안전 생명주기에 따르면 소프트웨어 안전성 품질을 확보하기 위해서는 개발 초기 단계에서 위험원 및 위험 분석(Hazard and risk analysis)을 통한 안전 요구사항을 개발하도록 권고하고 있다. 본 논문에서는 소프트웨어의 안전성 요구사항의 식별 및 정의를 위해 SysML을 활용한 STPA 기반의 위험원 분석 프로세스를 제안한다. 세부적으로는 SysML의 BDD과 IBD 다이어그램을 활용하여 기존 STPA 분석에서 활용되고 있는 제어구조도를 보다 명확하게 정의할 수 있도록 개선하였고, SD 다이어그램을 활용하여 안전 제약사항(요구사항)을 상세화할 수 있도록 하였다. 제안 방법의 적용 결과, STPA에서 누락되었던 위험원을 추가적으로 식별할 수 있었고, 위험원의 발생 시나리오도 상세하게 구체화할 수 있었다.

☞ 주제어 : 위험원 분석, STPA, SysML, 안전 요구사항

ABSTRACT

Today's software systems are becoming larger and more complicated, and the risk of accidents and failures have also grown larger. Software failures and accidents in industrial fields such as automobiles, nuclear power plants, railroad industries, etc. may lead to severe damage of property and human life. The safety-related international standards, such as IEC 61508 have been established and applied to industries for decades. The safety life cycle specified in the standards emphasize the activities to develop safety requirements through hazard and risk analysis in the early stage of software development. In this paper, we propose 'Hazard Analysis Process based on STPA using SysML' in order to ensure the safety of software at the early stage of software development. The proposed hazard analysis can be effectively performed minimizing the loss of hazard by using the BDD and the IBD of SysML to define the control structure of a system. The proposed method also improves the specification of the safety constraints(requirement) by using SD. As a result, it is possible to identify the hazard without missing and identify the hazard scenarios in detail, and safety can be sufficiently ensured in the early stage of software development.

☞ keyword : Hazard Analysis, Process, STPA, SysML, Safety Requirement

1. 서 론

소프트웨어 시스템의 규모가 점차 커지고 복잡해지면서 시스템의 오류나 고장으로 인한 사고위험도 함께 증가하고 있으며, 사고나 고장 발생 시 환경오염, 인명 피해, 재난 피해 등의 심각한 결과를 초래하는 경우가 많이 발생한다. 이에 따라 소프트웨어의 안전성 확보 문제가 시스템 산업 분야에서 중요한 이슈로 제기되고 있으며, 체계적인 안전성 품질 확보 방안이 다각적으로 연구되고 있다[1]. 특히 모든 산업 분야에서 사고 위험을 최소화하고 일정 수준의 안전성을 확보하기 위해 산업 도메인별로 안전 관련 국제 표준을 제정하고 있다. 산업 분야 안전 관련 국제 표준의 공통적인 기반이 되고 있는 IEC 61508[2]은 프로그램 가능한 전기/전자 기기의 안전 관리 표준으로, 안전 생명 주기, HW 및 SW에 대한 안전성 구현과 검증 방법을 제시하고 있다. 표준에서 제시하는 안전성 생명주기에 따르면, 시스템이나 장비의 안전성을 확보하기 위해서는 개발 초기 단계에서부터 위험원 및 위험 분석(Hazard and Risk Analysis) 활동을 통해 안전 요구사항을 초기에 식별 및 개발하도록 권고하고 있다. 위

1 Department of Information and Media, Seoul Women's University, Seoul, 01797, Korea.
2 Department of Information Security, Seoul Women's University, Seoul 01797, Korea.
* Corresponding author (byongl@swu.ac.kr)
[Received 15 November 2018, Reviewed 3 December 2018, Accepted 12 December 2018]
☆ 본 논문은 최나연의 학위논문을 바탕으로 작성되었음

험원 분석은 시스템에 내재되어 있는 위험원들을 식별하고 이들 위험원들에 의해 발현될 각 종 위험들을 미리 예상하고 평가하여 사고나 고장에 대한 대응을 수립하는 활동이다. 이와 같이 대부분의 안전성 관련 표준에서는 위험원 분석을 소프트웨어 개발 초기 단계에 수행함으로써 목표로 하는 안전 수준을 달성하도록 권고하고 있다.

본 논문에서는 소프트웨어 개발 초기 단계에서 위험원 분석 기법의 하나인 STPA(System Theoretic Process Analysis)[3,4]를 적용하여 위험원을 효과적으로 식별 및 분석할 수 있도록 하는 위험원 분석 프로세스를 제안한다. STPA는 STAMP(System Theoretic Accident Model and Processes)[5-7]에 기반하고 있는 위험원 분석 방법으로, 컴포넌트 사이의 상호작용에서 발생하는 고장의 원인을 식별하고 분석하는 기법이다. 하지만 기존 STPA 위험원 분석을 수행함에 있어서 두 가지의 어려운 점이 존재한다. 첫 번째로는 제어구조도의 명확한 정의가 어렵다[8]. 제대로 정의되지 않은 제어구조도를 통해 STPA를 수행하게 되면 식별해야 할 중요한 위험원이 누락될 수 있다. 두 번째로는 STPA에서 식별하는 UCA(Unsafe Control Action)가 모두 위험원으로 식별되지 않는다는 점이다. 개별 UCA는 시스템의 상태, 환경 상태 등에 따라서 위험원이 될 수도 있고, 그렇지 않을 수도 있다. Thomas[9]는 이러한 문제점을 보완하기 위하여 위험원 분석 과정에 UCA 판별 절차를 포함하는 개선된 STPA 분석 방법을 제시하고 있다. 제어구조도의 문제점을 개선하기 위한 연구[10,11]에서는 제어구조도 정의 단계에서 UML/SysML을 이용하거나, 본격적인 분석 활동 단계에서 SysML을 이용하는 연구들이 각각 진행되고 있다. 본 논문에서는 시스템 개발 초기 단계에서 SysML을 이용하여 제어구조도의 작성을 보다 상세화/구체화할 수 있는 위험원 분석 프로세스를 제안한다. 제안하는 프로세스의 효과성과 타당성 분석을 위해 IoT 가스락 시스템을 대상으로 제안 프로세스를 적용하고 그 결과를 분석하였다. 이를 통해 위험원의 누락을 최소화할 수 있었으며, 위험원 발생 시나리오를 더욱 상세하게 명세할 수 있었다.

2. 관련 연구

2.1 STAMP/STPA

STAMP[5-7]는 시스템의 위험 원인을 분석하기 위해 시스템 이론(System Theoretic Approach)을 적용한 모델이다. 시스템 이론은 이벤트 사이의 복잡한 관계를 분석하여 해당 이벤트가 왜 발생했는지를 파악할 수 있게 해주는 방법이다.

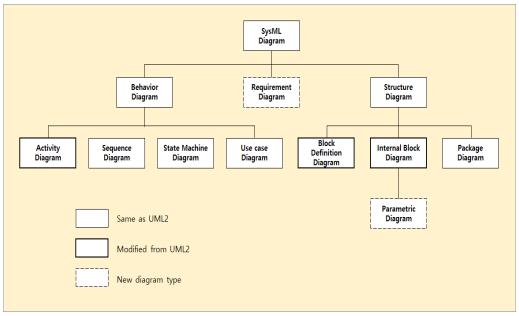
STPA는 STAMP에 기반한 위험원 분석 방법으로, 시스템을 구성하는 컴포넌트에 문제가 없더라도 컴포넌트 간의 상호작용에 의해 사고가 발생할 수 있다는 점을 고려한다. STPA는 3단계의 프로세스로 이루어진다. 1단계에서는 컴포넌트의 상호작용을 반영한 제어구조도를 정의하고, 2단계에서는 작성된 제어구조도를 통해 UCA(Unsafe Control Action)와 안전 제약사항(요구사항)을 식별한다. 3단계에서는 2단계에서 식별된 UCA에 대해서 왜 그러한 상황이 발생하게 되었는지 원인을 분석하고, 해당 원인으로 인한 발생 가능한 시나리오를 생성하게 된다.

2.2 SysML

SysML(System Modeling Language)[12,13]는 시스템을 구현할 때 적용할 수 있는 모델링 언어로, 시스템의 사양화, 분석, 설계, 타당성 확인/검증을 위해 사용되며, 현재 자동차, 항공우주, 통신 분야 등에서 광범위하게 활용되고 있다. SysML 모델링 언어는 UML을 기반으로 UML의 일부를 재사용하고, 여기에 시스템 엔지니어링에 필요한 내용을 추가하여 개발되었다. SysML 다이어그램의 종류로는 시스템의 동적 행위 분석을 위한 Bebavior Diagram, 시스템의 요구사항을 모델링하기 위한 Requirement Diagram, 시스템의 정적/구조적인 구성에 대한 분석을 위한 Structure Diagram으로 나누어진다. (그림 1 참고)

본 논문에서는 대상 시스템의 전체적인 구조를 파악하기 위해 SysML의 Structure Diagram 중 BDD(Block Definition Diagram)를 이용하고, 구성요소들 간의 상호작용을 파악하기 위해 IBD(Internal Block Diagram)를 이용한다. 또한 STPA기법을 통해 식별된 안전 제약사항(요구사항)을 더욱 상세화/구체화하기 위해 구성 요소들간의 시간적인 순서를 고려할 수 있는 Behavior Diagram 중 SD(Sequence Diagram)를 이용한다.

BDD [12,13]는 SysML의 구조 다이어그램 중의 하나로, UML의 Class 개념의 확장인 시스템과 그 구성요소를 나타내는 'Block'들 간의 관계를 정의하기 위한 다이어그램이다. BDD의 구성요소는 크게 시스템 및 구성요소를 나타내는 "Block", Block 간의 관계를 나타내는 "Relationship", Block 사이의 인터페이스를 나타내는 "Interface", Block과 Relationship 사이를 연결해주는 "Port", Block 사이에서의 정보 및 신호의 흐름을 나타내는 "Item Flow"가 있다. IBD [12,14]는 시스템 Block의 내



(그림 1) SysML 다이어그램
(Figure 1) SysML Diagram

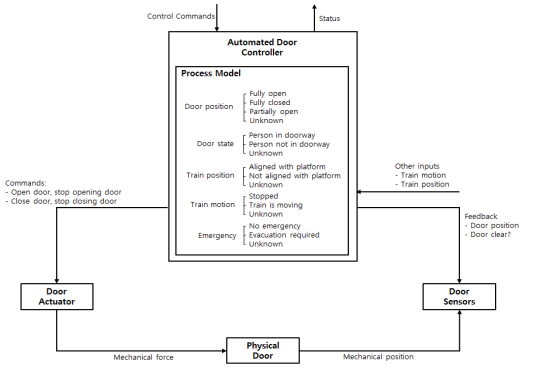
부를 정의하는 것으로, BDD로부터 인스턴스화된다. BDD에서의 Block은 IBD에서 Part로 표현되며, Part는 Port를 통해 Connector로 연결된다. SD [15]는 UML의 Sequence Diagram과 동일한 다이어그램으로, 여러 객체들이 어떻게 상호작용을 하는가를 표현한다. 즉 객체 간에 주고받는 메시지의 교환을 모델화하는 것으로, 메시지보다 메시지가 발생하는 순서에 초점을 맞추어 작성한다. SD의 수직 방향은 시간흐름을 나타내고 수평방향은 상호작용을 하는 객체들을 나타낸다. SD는 라이프 라인과 메시지로 구성되며, 라이프 라인은 네모난 직사각형으로 표현하며, 아래쪽 방향으로 점선 라인을 그리며 내려간다. 라이프라인의 이름은 역할 또는 인스턴스들을 나타낸다. 메시지는 객체에게 어떠한 일을 의뢰하기 위해 전달되며, 화살표로 표현한다 (표 1).

(표 1) SD의 메시지 종류
(Table 1) Message types of SD

종류	표기	의미
동기호출	———	메시지 실행이 종료 될 때까지 다음 메시지는 호출되지 않는다.
비동기 호출	———	메시지 실행의 종료와 상관없이 다음 메시지를 호출한다.
응답 (Return)	- - - - - - -	응답은 메시지가 종료한 것을 나타내며, 필수 표기가 아닌 생략이 가능하다.

2.3 Thomas의 STPA 제안방법

Thomas[9]는 기존 STPA 방법에 UCA(Unsafe Control Action) 판별 절차를 포함한 개선된 STPA 방법을 제시하고 있다. UCA는 대상 시스템의 상태 값에 따라서 위험원의 여부가 달라질 수 있다. 예를 들어 (그림 2) "기차의 도어가 열린다."와 같은 Control Action은 시스템의 상태 값에 따라서 위험원 여부가 달라질 수 있다. 플랫폼에 정상적으로 도착하지 않은 상태에서 도어열림 Control Action이 제공되면 도어열림은 위험원이 된다. 하지만 플랫폼에 정상적으로 도착한 상황에서 도어열림 Control Action이 제공된다면 도어열림은 위험원이 되지 않는다. 이와 같이 Thomas는 시스템의 모든 요소를 고려하여 위험원을 식별하기 위해 시스템의 상태 값을 구체적으로 표현할 수 있는 Process Model을 제어구조도에 반영하는 방법을 제안하고 있다.



(그림 2) Thomas의 Process Model을 포함하고 있는 제어 구조도
(Figure 2) Control Structure including Thomas's Process Model

3. SysML을 이용한 STPA 기반의 위험원 분석 프로세스

STPA를 체계적으로 수행하기 위해서는 Thomas가 제시한 방법과 같이 제어구조도의 각 요소를 정확히 파악하고, 구체적인 Process Model을 포함하여 정의하는 것이 중요하다. 하지만 이를 제대로 정의하기 위해서는 대상 시스템에 대한 전체적인 구조와 각 구성 요소들 간의 관계 및 상호작용을 제대로 파악해야 한다. 본 논문에서는 대상 시스템의 전체적인 구조를 파악할 수 있도록 BDD를 적용하고, 각 요소들 간의 상호작용을 파악할 수 있도록 IBD를 적용하여 제어구조도를 명확하게 정의할 수 있도록 돕는다. 또한 SD의 적용을 통해서 개발된 안전성 제약사항(요구사항)을 더 상세하게 명세할 수 있도록 지원한다.

3.1 기존 STPA 위험원 분석 프로세스

STPA는 표 3과 같이 3단계의 프로세스로 이루어져 있다. 1단계는 제어구조도를 정의하는 단계이며, 2단계는 UCA 및 안전성 제약사항(요구사항)을 식별하는 단계이며, 3단계는 2단계에서 식별된 위험원이 왜 발생하였는지 Casual Factor를 식별하는 단계이다. 다음은 기존 STPA 분석 프로세스를 IoT 가스락 시스템에 적용한 예이다.

(표 2) STPA 프로세스
(Table 2) STPA Process

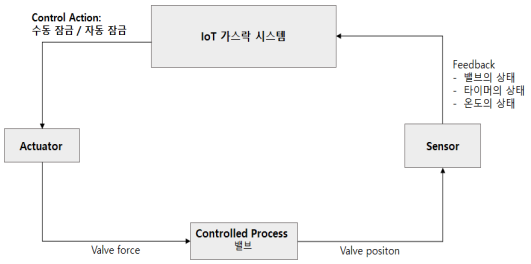
STPA 프로세스	
STEP 1	제어구조도 정의
STEP 2	UCA 및 안전 제약사항(요구사항) 식별
STEP 3	Casual Factor 식별

3.1.1 제어구조도 정의

STPA의 첫 번째 단계는 IoT 가스락 시스템의 제어구조도를 정의하는 단계이다. Controller는 IoT 가스락 시스템이 되며, 이때 Control Action은 수동잠금과 자동잠금이다. 잠금이 실행되는 Controlled Process는 밸브가 되며 센서에 의해서 밸브의 상태, 타이머의 상태, 온도의 상태값의 전달을 통해 IoT 가스락 시스템의 Control Action을 제어하게 된다. 이를 바탕으로 IoT 가스락 시스템에 대한 제어구조도를 정의하면 그림 3과 같다.

3.1.2 UCA 및 안전 제약사항(요구사항) 정의

STPA의 두 번째 단계는 첫 번째 단계에서 식별한 Control Action인 수동잠금과 자동잠금을 기반으로 UCA를 식별하고, 이에 대응하는 안전 제약사항(요구사항)을 식별하는 단계이다. 표 3의 "타이머 설정 상황에서의 자동잠금"의 예를 들면, ① 자동잠금을 제공하지 않았다 ② 자동잠금을 제공했다 ③ 자동잠금을 너무 늦게 또는 일찍 제공했다 ④ 자동잠금을 너무 짧게 또는 길게 지속했다와 같이 4가지 유형에 따라 위험원을 식별하고 각 위험원에 대응하는 안전 제약사항(요구사항)을 식별한다.



(그림 3) IoT 가스락 시스템의 제어구조도
(Figure 3) Control Structure of IoT Gaslock System)

3.1.3 Casual Factor 식별

STPA의 세 번째 단계는 두 번째 단계에서 식별한 위험원이 왜 발생하게 되었는지에 대한 Causal Factor를 식별하는 단계이다. STPA에서는 Causal Factor 식별을 위해 다음 4가지 기준을 제시하고 있다: ① 외부입력제어나 정보가 잘못되거나 없어짐 ② 제어알고리즘이 올바르지 않음 ③ 정보를 제공하는 시스템/센서가 잘못됨 ④ 수행대상이 실패함. 이 4가지 기준에 따라 표 3의 "H-1: 타이머의 설정이 완료가 되었는데 자동 잠금을 제공하지 않았다"에 대한 Casual Factor는 표 4와 같이 도출된다.

(표 3) UCA 및 안전 제약사항(요구사항)
(Table 3) UCA and Safety Constraints(Requirements)

Control Action	UCA(Unsafe Control Action)			
	Not Provided	Providing Causes	Too Late or Early	Too Soon or Long
자동 잠금	타이머의 설정이 완료가 되었는데 자동 잠금을 제공하지 않았다. (H-1)	해당 없음	타이머의 설정이 완료가 되었는데 자동 잠금을 너무 늦게 제공했다. (H-2)	타이머의 설정의 완료로 인한 자동 잠금이 너무 짧은 시간 동안 지속되었다. (H-3)
안전 제약사항 (요구사항)	타이머의 설정이 완료가 되면 밸브는 자동적으로 닫혀야 한다.	해당 없음	타이머의 설정이 완료가 되면 밸브는 5초 이내에 자동적으로 닫혀야 한다.	타이머의 설정이 완료가 되면 밸브는 자동적으로, 완전하게 닫혀야 한다.

(표 4) H-1에 대한 Casual Factor
(Table 4) Casual Factor for H-1

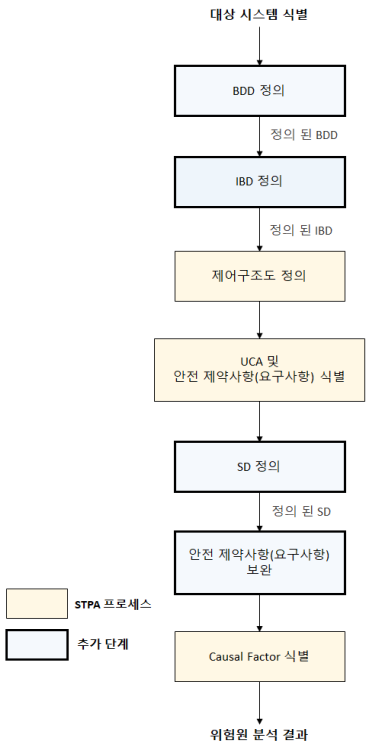
요소		기준	Casual Factor
Controller	IoT 가스락 시스템	①	-
		②	밸브의 Off 알고리즘이 잘못됨
			타이머와 밸브 사이의 Off 알고리즘이 잘못됨
		③	밸브의 상태를 Off로 잘못 인식
			타이머의 설정 상태를 제대로 인식하지 못함
Controlled Process	밸브	④	밸브 수행 실패
Actuator		④	Actuator 수행 실패
Sensor		③	밸브에 대한 정보 전달이 제대로 이루어지지 않음
			타이머에 대한 정보 전달이 제대로 이루어지지 않음

3.2 SysML을 이용한 위험원 분석 프로세스

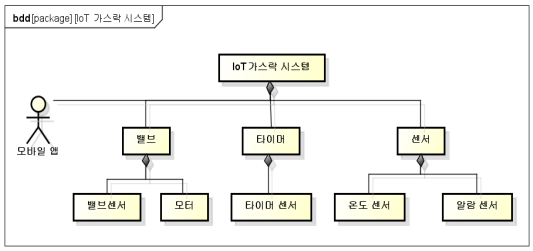
SysML을 이용한 STPA 기반의 위험원 분석 프로세스는 그림 4와 같이 1단계 BDD 정의, 2단계 IBD 정의, 3단계 제어구조도 정의, 4단계 UCA 및 안전 제약사항(요구사항) 식별, 5단계 SD 정의, 6단계 안전 제약사항(요구사항) 보완, 7단계 Casual Factor 식별 등 총 7단계로 이루어진다.

3.2.1 BDD 정의

IoT 가스락 시스템의 예에서, 원격 잠금을 실행하기 위한 모바일 앱, 잠금 대상인 밸브, 타이머의 설정을 위한 타이머, 주변 온도 및 경보음의 알림을 위한 센서 등이 BDD의 대상이 될 수 있다. 밸브는 실제로 밸브의 잠금을 구동시키는 모터와 이를 받아들이는 밸브 센서로 이루어져 있으며, 타이머는 타이머 센서, 센서는 온도 센서, 알람 센서로 이루어져 있다. 요소들은 BDD의 구성요소인 Block으로 표현되며 각 Block들은 합성 관계로 이루어진다. 이를 바탕으로 IoT 가스락 시스템에 대한 BDD를 정의하면 그림 5와 같다.



(그림 4) 제안하는 SysML을 이용한 STPA 기반의 위험원 분석 프로세스
(Figure 4) Proposed Hazard Analysis Process Based on STPA using SysML)

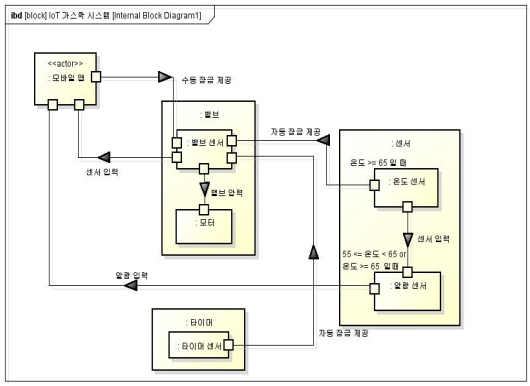


(그림 5) IoT 가스락 시스템의 BDD
(Figure 5) BDD of IoT Gaslock System

3.2.2 IBD 정의

두 번째 단계는 BDD를 기반으로 IBD를 정의하는 단계이다. 모바일 앱에서 수동 잠금 명령을 전달하면 밸브의 밸브 센서가 이를 전달받아 모터 구동에 의해 밸브가

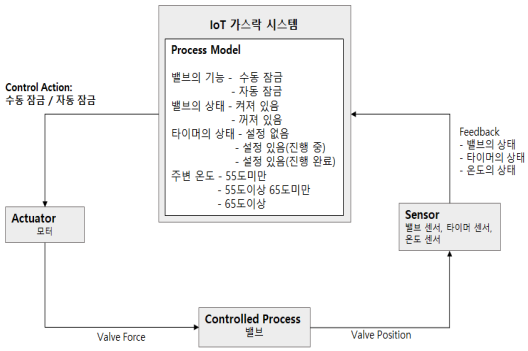
닫힌다. 또한 설정한 타이머의 완료를 알리는 타이머 센서에 의해서 또는 주변 온도가 65도 이상임을 알리는 온도 센서에 의해서 밸브는 자동으로 닫힌다. 이러한 각 구성요소들간의 관계 및 상호작용을 나타내는 IBD를 정의하면 그림 6과 같다.



(그림 6) IoT 가스락 시스템의 IBD
(Figure 6) IBD of IoT Gaslock System

3.2.3 제어구조도 정의

세 번째 단계는 IBD를 기반으로 제어구조도를 정의하는 단계이다. 누락 없이 위험원을 식별하여 STPA를 체계적으로 분석할 수 있도록 제어구조도의 각 요소를 정의하고, 구체적인 Process Model을 포함하여 제어구조도를 정의한다. Controller는 IoT 가스락 시스템이며 Process Model은 밸브의 기능, 밸브의 상태, 타이머의 상태, 주변 온도를 포함하고 있으며, 이에 따른 Control Action은 밸브의 기능으로 수동잠금과 자동잠금이다. Control Action인 잠금이 수행되는 Controlled Process는 밸브이며 Actuator의 요소는 모터이고, Sensor의 요소는 밸브 센서, 타이머 센서, 온도 센서이다. 이를 전부 포함하는 IoT 가스락 시스템에 대한 제어구조도는 그림 7과 같다.



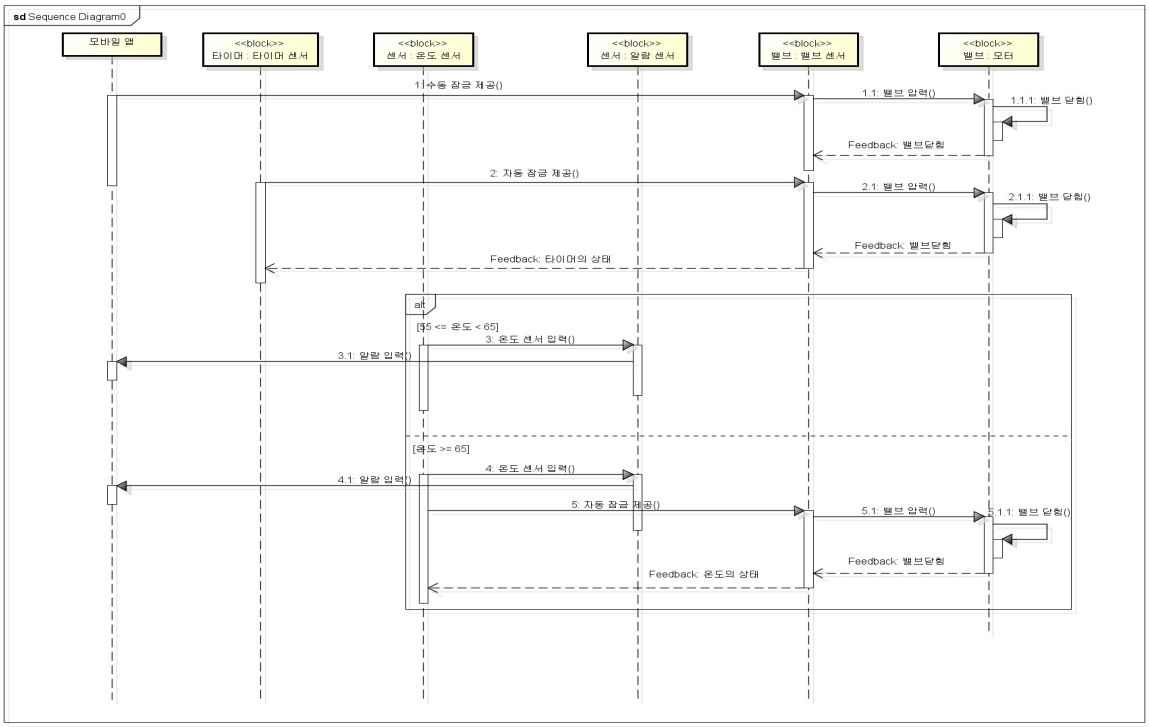
(그림 7) IoT 가스락 시스템의 제어구조도
(Figure 7) Control Structure of IoT Gaslock System

3.2.4 UCA 및 안전 제약사항(요구사항) 정의

SysML을 이용한 STPA 기반의 위험원 분석 프로세스의 네 번째 단계는 UCA 및 안전성 제약사항(요구사항)을 식별하는 단계이다. 이전 단계에서 식별한 Control Action

(표 5) UCA 및 안전 제약사항(요구사항)
(Table 5) UCA and Safety Constraints(Requirements)

Control Action	밸브 상태	타이머 상태		주변 온도	UCA(Unsafe Control Action)			
					Not Provided	Providing Causes	Too Late or Early	Too Soon or Long
자동 잠금	켜져 있음	설정 있음	진행중	관계없음	No	No	No	No
				55도 이하	No	No	No	No
				55도 이상 65도 미만	No	No	No	No
				65도 이상	Yes (H-1)	No	Yes (H-2)	Yes (H-3)
			진행 완료	관계없음	Yes (H-4)	No	Yes (H-5)	Yes (H-6)
				55도 이하	Yes (H-7)	No	Yes (H-8)	Yes (H-9)
				55도 이상 65도 미만				
				65도 이상	Yes (H-10)	No	Yes (H-11)	Yes (H-12)



(그림 8) IoT 가스락 시스템의 SD
(Figure 8) SD of IoT Gaslock System

인 수동잠금과 자동잠금을 기반으로 IoT 가스락 시스템의 UCA를 식별한다. (표 3 참고) UCA를 모두 식별하고 난 후, 각 위험원에 대응하는 안전 제약사항(요구사항)을 식별한다. (표 5 참고)

3.2.5 SD 정의

SysML을 이용한 STPA 기반의 위험원 분석 프로세스의 다섯 번째 단계는 두 번째 단계에서 정의된 IBD를 기반으로 IoT 가스락 시스템의 SD를 정의하는 단계이다. IBD에서 정의한 모든 구성 요소들이 SD의 각 객체가 되며, IBD에서 표현하고 있는 Item Flow가 SD의 각 메시지로 입력된다. IoT 가스락 시스템에 대한 SD를 정의하면 그림 8과 같다.

3.2.6 안전 제약사항(요구사항) 보완

SysML을 이용한 STPA 기반의 위험원 분석 프로세스의 여섯 번째 단계는 다섯 번째 단계에서 정의된 SD를 바탕으로 안전성 제약사항(요구사항)을 보완하는 단계이다. 예를 들어 “(H-4) 타이머의 설정이 완료되었는데 자동잠금을 제공하지 않았다.”와 같은 기존 위험원의 경우, 다섯 번째 단계에서 정의된 SD를 바탕으로 “타이머 센서에 의해 설정한 타이머의 완료를 확인하면, 자동잠금 명령이 밸브 센서로 전달되어 모터 구동을 통해 밸브는 자동적으로 닫혀야 한다.”와 같이 구성 요소들 간의 시간적인 순서를 고려하여 더 상세하게 보완된다. (표 6 참고)

3.2.7 Casual Factor 식별

SysML을 이용한 STPA 기반의 위험원 분석 프로세스의 일곱 번째 단계는 네 번째 단계에서 식별한 위험원이 왜 발생하게 되었는지 각 위험원에 대응하는 Casual Factor를 식별하는 단계이다. “타이머의 설정이 완료가 되었는데 자동잠금을 제공하지 않았다(H-1, H-4, H-7, H-10)”에 대한 위험원의 Casual Factor는 표 7과 같다.

(표 6) 기존 안전 제약사항(요구사항)과 보완 안전 제약사항(요구사항)
(Table 6) Existing and supplementary Safety Constraints(Requirements)

위험원	기존 안전 제약사항(요구사항)	보완 안전 제약사항(요구사항)
H-1	타이머의 설정이 완료가 되지 않았더라도, 주변 온도가 65도 이상이면 밸브는 자동적으로 닫혀야 한다.	타이머 센서에 대한 설정완료 알림이 없더라도, 온도 센서에 의해 주변 온도가 65도 이상임을 감지하면 자동잠금 명령이 밸브 센서로 전달되어 모터 구동을 통해 밸브는 자동적으로 닫혀야 한다.
…	…	…
H-4	타이머의 설정이 완료가 되면 밸브는 자동적으로 닫혀야한다.	타이머 센서에 의해 설정한 타이머에 대한 완료임을 확인하면, 자동잠금 명령이 밸브 센서로 전달되어 모터 구동을 통해 밸브는 자동적으로 닫혀야 한다.
…	…	…
H-7	주변 온도가 65도 이하일지라도, 타이머의 설정이 완료가 되었으면 밸브는 자동적으로 닫혀야 한다.	온도 센서에 의해 주변 온도가 65도 이하임을 감지하더라도, 타이머 센서에 의해 설정한 타이머에 대한 완료임을 확인하면, 자동잠금 명령이 밸브 센서로 전달되어 모터 구동을 통해 밸브는 자동적으로 닫혀야 한다.
…	…	…
H-10	타이머의 설정도 완료가 되었고, 주변 온도도 65도 이상이면 밸브는 자동적으로 닫혀야 한다.	타이머 센서와 온도 센서에 의해 설정한 타이머의 완료 및 주변 온도의 65도 이상임을 감지하면 자동잠금 명령이 밸브 센서로 전달되어 모터 구동을 통해 밸브는 자동적으로 닫혀야 한다.
…	…	…

4. 비교 분석

본 논문에서 제안하는 프로세스는 1단계인 BDD 정의와 2단계인 IBD 정의를 통해 기존 STPA 프로세스의 제어구조도를 명확하게 정의할 수 있도록 하였으며, 추가적으로 SD 정의를 통해 4단계에서 정의한 안전성 제약사항(요구사항)을 보다 상세하게 정의할 수 있도록 하였다. (그림 3 참고) 제안 프로세스를 IoT 가스락 시스템에 적용한 결과는 다음과 같다.

(1) 제어구조도 정의

SysML을 이용하여 제어구조도를 구체적으로 정의할 수 있었다. 밸브의 기능, 밸브의 상태, 타이머의 상태, 주변 온도를 포함하여 Process Model를 구체적으로 파악하고, Actuator의 요소에 해당하는 모터, Sensor의 요소에 해당하는 밸브 센서, 타이머 센서, 온도 센서를 구체적으로 식별하여 제어구조도를 보다 완전하게 정의할 수 있었다. (기존 제어구조도는 그림 3 참고, 본 논문에서 제안하는 프로세스의 제어구조도는 그림 7 참고)

(2) 식별한 위험원의 개수

제안하는 프로세스의 적용을 통해 기존 STPA를 통해서는 식별하지 못한 위험원까지 누락 없이 식별할 수 있었다. 표 3의 예에서, H-4, H-5, H-5은 기존 STPA 프로세스를 통해 식별한 위험원과 동일하지만, 이를 제외한 H-1 ~ H-12는 제안하는 프로세스를 통해 추가적으로 식별된 위험원이다.

(3) 안전성 제약사항(요구사항)의 보완

제안하는 SysML을 활용한 STPA 기반의 위험원 분석 프로세스를 통해서 네 번째 단계에서 정의한 안전성 제약사항(요구사항)을 보완하여 구체적으로 정의할 수 있었다. 예를 들어 “(H-4) 타이머의 설정이 완료가 되었는데 자동잠금을 제공하지 않았다.”의 위험원의 경우, 다섯 번째 단계에서 정의된 SD를 바탕으로 “타이머 센서에 의해 설정한 타이머에 대한 완료임을 확인하면, 자동 잠금 명령이 밸브 센서로 전달되어 모터 구동을 통해 밸브는 자동적으로 닫혀야 한다.”와 같이 구성 요소들간의 시간적인 순서를 고려하여 상세하게 정의할 수 있었다. (표 6 참고)

(4) Casual Factor

제안하는 SysML을 활용한 STPA 기반의 위험원 분석 프로세스를 통해 제어구조도의 각 요소를 정확히 파악하

(표 7) H-1, H-4, H-7, H-10에 대한 Casual Factor
(Table 7) Casual Factor for H-1, H-4, H-7, H-10
*괄호가 없는 것은 공통적인 부분이다.

요소		기준	Casual Factor
Controller	IoT 가스락 시스템	①	-
		②	밸브의 Off 알고리즘이 잘못됨
			온도와 밸브 사이의 자동적인 Off 알고리즘이 잘못됨 (H-1, H-10)
			타이머와 밸브 사이의 자동적인 Off 알고리즘이 잘못됨(H-4, H-7, H-10)
		③	밸브의 상태를 Off로 잘못인식
			주변 온도를 제대로 인식하지 못함(H-1, H-10)
			설정된 타이머로 인해 온도를 인식하지 못한 채 타이머에 대한 정보만 가지고 있음(H-1)
			설정한 타이머의 상태를 제대로 인식하지 못함(H-4, H-10)
			온도에 대한 정보만 가지고 있고, 타이머에 대해서 제대로 인식하지 못하고 있음(H-7)
Controlled Process	밸브	④	밸브 수행 실패
Actuator	모터	④	모터 수행 실패
Sensor	밸브 센서	③	밸브에 대한 정보 전달이 제대로 이루어지지 않음
	온도 센서		온도에 대한 정보 전달이 제대로 이루어지지 않음(H-1, H-10)
	타이머 센서		타이머에 대한 정보 전달이 제대로 이루어지지 않음(H-4, H-7, H-10)

고, 구체적인 Process Model을 포함하여 제어구조도를 정의할 수 있었다. 이를 바탕으로 위험원이 발생하게 된 Casual Factor 및 시나리오에 대해서도 구체적으로 식별할 수 있었다. 표 6에서 “타이머 설정 상황에서의 자동잠금을 제공하지 않은 위험원”의 예에서, 기존방법(표 4 참고)에서는 H-4에 대해서만 Casual Factor를 식별하였지만 제안방법(표 7 참고)에서는 위험원 H-1, H-7, H-10을 추가적으로 더 식별함으로써 구체적인 시나리오를 도출할 수 있었다.

5. 결 론

소프트웨어의 안전성을 확보하기 위한 국제 표준에서 권고하는 생명주기에 따르면 개발 초기 단계에서 위험원 및 위험 분석을 통해 안전 요구사항을 개발하도록 권고하고 있다. 본 논문에서는 개발 초기에 위험원이 누락되거나 식별되지 않는 문제점을 보완할 수 있는 개선된 위험원 분석 프로세스를 제안한다. 위험원을 누락하지 않고, STPA를 체계적으로 수행하기 위해서는 위험원 분석 전의 제어구조도를 제대로 정의하는 것이 중요하다. 본 논문에서는 제어구조도의 작성 시 위험원을 누락하지 않고 안전성을 확보하기 위해 SysML의 BDD와 IBD를 이용하였다. 또한 STPA를 통해 개발 된 안전 제약사항(요구사항)을 더 상세하게 보완할 수 있도록 SysML의 SD를 이용했다. 제안하는 프로세스의 타당성 분석을 위해 IoT 가스락 시스템에 적용하고 그 결과를 분석하였으며, 기존 방법과의 비교를 통한 분석 결과는 다음과 같다.

첫째, BDD를 통해 대상 시스템의 전체적인 구조를 파악하고, IBD를 통해 각 요소들 간의 상호작용을 파악하여 이를 바탕으로 제어구조도를 구체적이고 완전하게 정의할 수 있다.

둘째, 구체적으로 정의된 제어구조도를 바탕으로 기존 STPA 보다 더 많은 위험원을 식별할 수 있다.

셋째, 정의된 안전 제약사항(요구사항)을 SD를 통해 더 상세하게 구체화할 수 있다.

넷째, 위험원이 왜 발생하게 되었는지에 대한 Casual Factor 및 시나리오를 더 구체적이고 완전하게 명세할 수 있다.

향후 본 논문에서 제시하는 BBD, IBD, SD작성 프로세스를 PHA, FMEA 등 타 안전성 분석 프로세스에 확대 적용 가능한지 확인할 필요가 있다. 특히, PHA 분석 방법은 STPA 방법과 유사하게 개발 초기 단계에 적용할 수 있는 위험원 분석 방법이지만, 위험원 분석 활동을 제어구조도가 아닌 브레인스토밍 활동에 의존하고 있다. 본 논문에서 제시하는 방법인 BBD, IBD, SD, 그리고 제어구조도 작성이 PHA의 브레인스토밍 활동을 대체할 수

있을지, 아니면 보조적인 역할로만 사용되어야 할지 상세히 분석할 필요가 있다.

참고문헌(Reference)

[1] S. C. Huang, F. C. Cheng, and Y. S. Chiu, "Efficient Contrast Enhancement Using Adaptive Gamma Correction With Weighting Distribution," IEEE Transactions on Image Processing, Vol. 22, No.3, pp. 1032-1041, 2013. http://dx.doi.org/10.1109/TIP.2012.2226047

[2] "IEC 61508-1: Functional safety of electrical/ electronic/pro-grammable electronic safety-related systems: General requirements," IEC, April 2010.

[3] Young, William, and Nancy Leveson. "Systems thinking for safety and security," Proceedings of the 29th Annual Computer Security Applications Conference. ACM, 2013.

[4] Leveson, Nancy G. "Safety Analysis in Early Concept Development and Requirements Generation," 2018. http://dx.doi.org/10.1002/j.2334-5837.2018.00492.x

[5] Leveson, N.: Engineering a Safer World, Massachusetts Institute of Technology, 2011.

[6] Leveson, Nancy G. "A systems-theoretic approach to safety in software-intensive systems," IEEE Transactions on Dependable and Secure computing 1.1 (2004): 66-86, 2004.

[7] Leveson, Nancy, et al. "A systems theoretic approach to safety engineering," Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, 2003.

[8] Asplund, Fredrik, Jad El-khoury, and Martin Törngren. "Safety-Guided Design through System-Theoretic Process Analysis, Benefits and Difficulties," 30th International System Safety Conference. 2012.

[9] Thomas, J. "Performing hazard Analysis on Complex, Software-and Human-Intensive Systems J. Thomas, SM; Massachusetts Institute of Technology; Cambridge, Massachusetts, USA NG Leveson Ph. D.; Massachusetts Institute of Technology; Cambridge, Massachusetts, USA.

[10] Rejzek, Martin; Krauss, Sven Stefan; Hilbes, Christian, 2015. Safety Driven Design with UML and STPA - homepage 2019.04.15. https://www.zhaw.ch/no_cache/de/forschung/personen-publikationen-projekte/detailansicht-publikation/publikation/209168/

[11] Jensen, David C., and Irem Y. Tumer. "Modeling and Analysis of Safety in Early Design," Procedia Computer Science 16, 824-833, 2013. https://doi.org/10.1016/j.procs.2013.01.086

[12] Friedenthal, Sanford, Alan Moore, and Rick Steiner. A practical guide to SysML: the systems modeling language. Morgan Kaufmann, 2014.

[13] SysML Modelling: Block Definition Diagram (bdd) - homepage 2019.04.15. https://www.threesl.com/pages/reference/diagrams/sysml-block-definition-diagram.php

[14] SysML Modelling: Internal Block Diagram (ibd) - homepage 2019.04.15. https://www.threesl.com/pages/reference/diagrams/sysml-internal-block-diagram.php

[15] Sellami, Asma, et al. "A measurement method for sizing the structure of UML sequence diagrams." Information and Software Technology 59, 222-232, 2015.

◐ 저 자 소 개 ◑

최 나 연(Na-yeon Choi)
2015년 서울여자대학교 컴퓨터학과(공학사)
2017년 서울대학교 대학원 정보미디어학과(이학석사)
관심분야 : 소프트웨어 위험분석, 소프트웨어 안전성 프로세스, 소프트웨어 안전성 요구사항 etc.
E-mail : choiny@swu.ac.kr

이 병 걸(Byong-gul Lee)
1988년 University of Bridgeport 물리학과(이학사)
1996년 Auburn University 대학원 전산학과(공학석사)
1998년 Auburn University 대학원 전산학과(공학박사)
1998년~현재 서울여자대학 미래산업융합대학 정보보호학과 교수
관심분야 : 소프트웨어 보안, 소프트웨어 안전성, 소프트웨어 아키텍처, 소프트웨어 프로세스 etc.
E-mail : byongl@swu.ac.kr