Journal of KIIT. Vol. 21, No. 8, pp. 39-49, Aug. 31, 2023. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2023.21.8.39

SMC를 이용한 액츄에이터의 불확실성에 따른 STPA 위험 평가

박소희*, 권령구**, 권기현***

Risk Assessment of Actuators Uncertainty using STPA and SMC

Sohee Park*, Ryeonggu Kwon**, and Gihwon Kwon***

이 논문은 과학기술정보통신부 및 정보통신기술진흥센터의
고안전 SW 개발을 위한 안전 분석 및 검증 도구 기술 개발 사업의 연구 결과로 수행되었음(No. 2021-0-00122)

요 약

시스템 이론 기반으로 안전성을 분석하는 STPA(System-Theoretic Process Analysis)에서 액츄에이터는 컨트롤러가 하위 프로세스로 전달하는 제어 명령을 관리한다. 액츄에이터는 환경 조건, 외부 간섭 등에 의해 하위 프로세스에 잘못된 명령을 전달해 사고를 유발할 수 있다. 이러한 가능성을 고려하지 않으면 실제와 다른 위험 평가 결과를 도출할 수 있다. 따라서 본 논문에서는 PTA(Priced Timed Automata)와 SMC(Statistical Model Checking)를 이용하여 액츄에이터의 불확실성에 따른 STPA 위험 평가 방법을 제안한다. SMC는 기존 모델 검증과 달리 통계 기법을 이용해 추정치를 계산해 검증한다. 이러한 특징으로 상태 폭발 문제가 발생하는 PTA에도 적용할 수 있다. 제안한 방법을 철도 건널목 시스템에 적용하여 액츄에이터의 영향을 확인한 결과, 모든 시나리오에서 높은 위험도를 보였다. 이를 통해, 액츄에이터에 대한 손실 시나리오의 위험도 평가가 필요함을 확인하였다.

Abstract

In STPA(System-Theoretic Process Analysis), which analyzes safety based on system theory, actuators manage the control commands that the controller delivers to subprocesses. An actuator can send incorrect commands to a subprocess due to environmental conditions, external interference, etc. and cause an accident. Failure to consider these possibilities can lead to unrealistic risk assessment results. Therefore, this paper proposes a risk assessment method for STPA under actuator uncertainty using Priced Timed Automata (PTA) and Statistical Model Checking (SMC). Unlike traditional model verification, SMC uses statistical techniques to compute and verify estimates. This feature makes it applicable to PTAs with state explosion problems. The proposed method is applied to a railroad crossing system to determine the impact of actuators, and the results show high risk in all scenarios. This confirms the need for risk assessment of loss scenarios for actuators.

Keywords

actuator, stpa, risk assessment, statistical model checking, safety, uncertainty

* 경기대학교 SW안전보안학과 석사과정
- ORCID: https://orcid.org/0000-0003-0530-4946
** 경기대학교 컴퓨터과학과 박사과정(교신저자)
- ORCID: https://orcid.org/0000-0002-4942-247X
*** 경기대학교 AI컴퓨터공학부 교수
- ORCID: https://orcid.org/0000-0002-8221-4939

· Received: Jun. 25, 2023, Revised: Aug. 17, 2023, Accepted: Aug. 20, 2023
· Corresponding Author: Ryeonggu Kwon
Dept. of Computer Science, Kyonggi University, 154-42,
Gwanggyosan-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, South Korea
Tel.: +82-31-249-9666, Email: rkkwon@kyonggi.ac.kr

Ⅰ. 서 론

STPA(System-Theoretic Process Analysis)는 시스템 이론 기반의 위험원 분석 방법이다[1][2]. 전통적인 위험원 분석 기법인 SFMEA(Software Failure Modes Effects Analysis)와 FTA(Fault Tree Analysis)는 구성 요소의 결함이나 장애를 중심으로 분석하는 반면[3], STPA는 시스템의 구조, 동적 변화, 인간의 행동 등을 전체적으로 고려하여 시스템의 안전성을 평가하는 데 사용된다. 또한, STPA는 전통적인 기법에 비해 시스템의 복잡성과 다양성을 반영할 수 있고, 컴포넌트 간의 상호작용을 고려하여 불안전한 제어 명령(UCA, Unsafe Control Action)으로 일어날 수 있는 사고까지 분석 가능하다. 이러한 이유로 현대 시스템이 점차 복잡해져 위험원 분석 시 STPA 사용이 증가하고 있다[4].

그러나 STPA는 공식적으로 위험 평가 방법을 제공하지 않는다[5]. 시간적, 금전적인 요인으로 인해 모든 사고에 대해 안전 대책을 수립하는 것은 불가능하다. 따라서 제일 우선적으로 예방할 사고를 식별하기 위해서 STPA에 위험 평가 방법이 필요하다. 위험 평가 방법 중 하나인 정성적 방법은 대체로 전문가의 판단에 따라 위험의 발생 가능성과 심각도를 4~5단계로 나눠서 위험도를 높음, 중간, 낮음으로 분류하여 평가한다[6]. 정성적 방법은 간단히 위험도를 추정할 수 있다. 하지만 전문가마다 주관적인 평가가 이루어지기 때문에 간혹 시스템이 허용가능한 위험 수준을 만족하도록 평가가 이루어질 수 있는 가능성이 존재한다[7]. 이러한 이유로 정성적 방법뿐만아니라 정량적 방법도 필요하다. 정량적 방법은 통계 데이터와 사고 사례를 바탕으로 평가가 이루어져 정성적 방법보다 객관적인 평가가 가능하다. 그러나 이 방법은 시간과 비용이 많이 들고 불확실성을 다루기 어렵다는 한계가 존재한다.

따라서 이러한 한계를 극복하기 위해 STPA의 정량 위험 평가에 대한 다양한 연구가 진행되고 있다. Takata[8]는 소프트웨어 시스템의 안전성을 평가할 시 여러 분석 방법을 활용한 방안을 제시했다. 이 연구는 STPA와 FTA를 결합하여 각각을 보완하는 방법을 제시했다. FTA를 이용하여 사고를 발생시키는 결함을 추적한 후 각각의 발생 확률을 이용하여 사고 발생 가능성을 계산하였다. 이를 통해, 두 방법을 결합하여 위험원에 대한 안전 제약사항을 도출하는 것뿐만아니라 정량적 분석까지 충족하였다. Tsuji[9]는 통계적 모델 검증(SMC, Stochastic Model Checking)을 이용하여 손실 시나리오의 위험성을 분석하였다. STPA 분석 결과를 UPPAAL 모델로 변환하는 절차를 소개하고 철도 게이트 제어 시스템에 제안 방법을 적용해 손실 시나리오의 발생 가능성을 계산하여 위험도를 평가하였다.

기존 연구들은 STPA 분석 결과를 바탕으로 정량적 위험 평가를 수행하였다. 하지만 이 연구들은 액츄에이터의 오작동으로 인한 사고는 고려하지 않았다. 컨트롤러가 올바른 제어 명령(CA, Control Action)을 전달해도 액츄에이터가 명령을 제대로 수행하지 않는다면 하위 시스템인 제어되는 프로세스의 UCA를 유발하여 사고를 발생시킬 수 있다. 이러한 불확실성을 고려하지 않으면 실제 발생 가능한 위험을 과소평가나 과대평가할 수 있다. 따라서 액츄에이터의 오작동까지 고려한 위험 평가가 필요하다.

본 논문에서는 액츄에이터의 불확실성에 따른 STPA 위험 평가를 수행한다. 컨트롤러가 적절한 CA를 제공한 경우와 그렇지 않은 경우의 손실 시나리오를 모두 고려한 위험 평가를 한다. 시간에 따른 시스템의 동작을 표현하기 위해 PTA(Priced Timed Automata)를 이용한다. PTA는 일반적인 모델 검증을 사용하기에는 복잡한 모델이기 때문에 UPPAAL을 통해 SMC를 사용한다. STPA 수행 결과를 바탕으로 시스템 구조 및 동작을 UPPAAL로 모델링한다. 모델링 후 각 컴포넌트의 UCA이나 고장 등의 오작동 확률을 모델에 반영한다. 그 후, SMC을 이용하여 손실 시나리오의 발생 가능성을 계산한다. 이를 통해, 액츄에이터의 이상 행동으로 인한 사고의 위험성을 보이고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문의 배경 지식으로 STPA, SMC을 설명한다. 3장에서는 시스템의 모델링 방법과 각 시나리오의 쿼리 변환 방법에 대해 기술한다. 4장에서는 제안 방법을 철도 시스템에 적용한 결과를 보인다. 5장은 본 논문의 결론과 향후 연구에 대해 기술한다.

II. 배경 지식

2.1 STPA

STPA는 복잡한 시스템에서 잠재적인 사고 및 위험을 식별하고 예방하기 위한 시스템 이론 기반 위험 분석 기법이다[10][11]. STAMP(Systems Theoretic Accident Model and Processes) 기법의 확장으로 만들어졌으며, 시스템의 구조, 동작 및 컴포넌트와 사람 간의 제어 명령을 바탕으로 안전을 분석한다. STPA는 총 4단계로 구성되어 있다.

1) 사고 및 위험 정의

1단계에서는 분석 시스템의 범위를 결정하고 분석 시스템과 관련된 사고와 위험을 정의한다. STPA를 통해 어떤 사고(인명 피해, 임무 실패, 재산 손실 등)를 분석 및 예방할지 결정하여 시스템의 분석 범위를 결정한다. 그 후 시스템에 발생할 수 있는 사고를 정의하고 시스템 수준 위험(Hazard)을 도출한다.

2) 컨트롤 스트럭처 도식화

2단계에서는 컨트롤 스트럭처를 모델링한다. 컨트롤 스트럭처는 컨트롤러(Controller), 제어되는 프로세스(Controlled Process)와 각 컴포넌트간의 CA, 피드백(Feedback)으로 구성된다.

처음 모델링을 할 시에는 시스템을 추상화하여 모델링을 수행하고, 점차 시스템을 상세하게 모델링하여 컨트롤러나 제어되는 프로세스를 구체한다.

3) UCA 식별

3단계는 시스템의 위험을 유발할 수 있는 UCA을 식별한다. 2단계에서 도출한 CA을 바탕으로 분석하며 '제공하지 않음', '제공함', '너무 늦게 혹은 빨리 제공', '너무 이르게 혹은 너무 길게 제공'의 총 4가지 타입으로 분류된다.

4) 손실 시나리오 도출

4단계는 3단계에서 식별한 UCA가 발생하는 원인 및 손실 시나리오를 분석한다. 이때 손실 시나리오는 그림 1과 같이 크게 두 가지로 나뉘어진다.

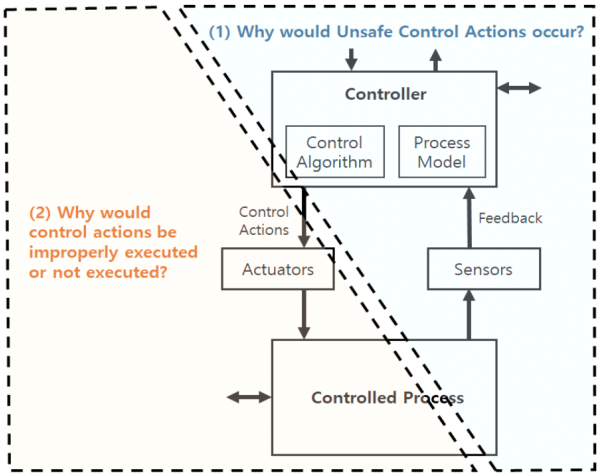


그림 1. 손실 시나리오 종류
Fig. 1. Type of loss scenarios

첫번째는 UCA로 인한 손실 시나리오이다. 컨트롤러에서 왜 UCA가 전송하였는지를 분석한 시나리오를 말한다. 두 번째는 적절한 CA이 제공되었지만 수행되지 않거나 부적절하게 수행된 시나리오이다. 액츄에이터의 오작동으로 인해 잘못 수행되지 못한 것이나 액츄에이터에서 CA를 전송하였지만 제어되는 프로세스로 잘못 전달된 것 대한 시나리오이다. 본 논문에서는 두 가지 손실 시나리오 중 액츄에이터의 오작동으로 CA가 잘못 수행된 경우에 대해 다룬다.

2.2 통계적 모델 검증

SMC은 통계 데이터를 기반으로 시스템의 동작을 분석하고 검증하는 방법이다[12]. 일반적인 모델 검증이 모델의 가능한 모든 상태를 전수 조사하여 확률을 계산하는 것과 달리 SMC는 시스템의 여러 번의 실행 결과를 바탕으로 전체적인 추정치를 계산한다. 이로 인해, 기존 모델 검증으로는 검증이 불가능한 대규모 상태 공간을 가지는 시스템에도 적용이 가능하다는 이점이 존재한다. 본 논문에서는 시간에 따른 시스템의 동작을 구현하기 위해 PTA를 사용한다. PTA는 상태 폭발 문제로 인해 일반적인 모델 검증으로는 사용할 수 없기 때문에 UPPAAL을 통해 SMC를 사용한다. PTA는 상태와 전환 모두에 비용과 시간이 할당된 오토마타이다[13]-[15].

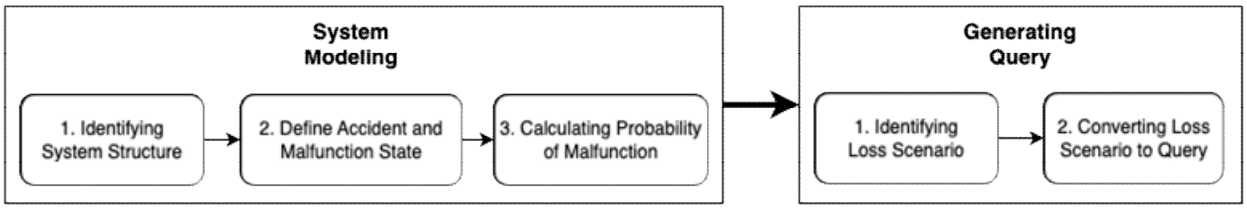


그림 2. 시스템 모델링 및 손실 시나리오 변환 단계
Fig. 2. Steps of system modeling and loss scenario conversion

Timed Automata를 확장한것으로 시계가 오토마타의 각기 다른 위치에서 서로 다른 속도를 가진 모델이다. PTA는 아래의 튜플로 정의된다.

$PTA = (L, l_0, X, \Sigma, E, R, I)$

- L: 위치(Location)들의 유한 집합
- l_0: 모델의 초기 위치
- X: Clock들의 집합
- $\Sigma = \Sigma_i \cup \Sigma_o$: 액션들의 집합
  Σ_i: 모델에 입력되는 액션 집합
  Σ_o: 모델에서 출력되는 액션 집합
- $E \subseteq L \times \mathcal{L}(X) \times \Sigma \times 2^X \times L$: 간선의 집합
  $\mathcal{L}(X)$: X에 대한 시간 제약사항들의 집합
- $R: L \rightarrow \mathbb{N}^X$: 각 위치의 clock 속도 할당 함수
- $I: L \rightarrow u(X)$: 위치에 대한 시간 제약사항을 할당하는 함수

PTA는 Clock X와 $\mathcal{L}(X)$ 통해 위치와 간선에 타이밍 제약 사항을 할당한다. 이러한 특징을 가져 PTA는 시간에 따른 시스템 동작을 구현할 수 있다.

UPPAAL 모델 체커의 통계 쿼리는 크게 4가지가 존재한다[16]. 그 중 확률을 추정하는 쿼리의 문법은 다음의 식 (1)과 같다.

$$Pr[SMCBounds]((' <> '|' [] ')expression) \quad (1)$$

SMCBounds는 쿼리 실행 수를 의미한다. 이를 사용하여 최대 실행 수를 원하는 정도로 제한할 수 있다. '<>'과 '[]'은 각각 expression이 한 번이라도 참일 경우와 항상 참일 경우를 나타내는 기호이다. '<>'는 언젠가 조건이 달성되는 경우를 구할 때 사용하는 것이고, '[]'는 초기 상태보다 시스템 실행이 종료될 때까지 상태가 만족되는 경우를 구할 때 사용된다. expression은 상태 경로를 나타내는 것으로 expression을 만족할 확률을 계산하는데 사용된다.

III. 불확실성을 반영한 시스템 모델링

본 논문에서는 SMC를 이용하여 액츄에이터의 오작동 확률을 따른 STPA 위험 평가를 수행하는 방안을 제안한다. [9]과 달리 결정적인 액션 수행과 비결정적 확률 전환을 모두 만족할 수 있는 방법과 손실 시나리오를 쿼리로 변환하는 법을 소개한다.

그림 2와 같이 총 5 단계로 시스템을 모델링하고, 손실 시나리오를 쿼리로 변환한 후 각 시나리오의 발생 가능성을 계산한다.

3.1 시스템 모델링 방법

1) 시스템 구조, 동작 및 제약사항 식별

시스템을 모델링하기 위해서 요구사항을 바탕으로 각 컴포넌트의 상태 공간과 제어 명령, 제약사항을 정의한다.

먼저, 각 컴포넌트의 상태 공간을 정의하기 위해 시스템의 요구사항을 바탕으로 컴포넌트가 수행할 수 있는 행동 혹은 명령을 파악한다. 여기서 행동은 전구의 ON, OFF와 같은 기능 동작을 말한다. 파악된 행동을 통해 컴포넌트가 수행하기 전 혹은 후의 상태를 도출하고, 이를 상태 공간 L으로 정의한다.

- L = {행동을 수행하기 전 상태} ∪
  {행동을 수행한 후 상태}

두 번째로 각 상태와 전환에 대한 시간 제약 사항을 결정한다. 요구사항을 바탕으로 컴포넌트의 위치가 변경되기까지 걸리는 시간을 식별한다.

- $\mathcal{L}(X)$ = {위치 전환에 걸리는 시간}

마지막으로 각 컴포넌트의 상태 전환를 식별하기 위해서 컨트롤 스트럭처를 바탕으로 각 컴포넌트의 연결과 주고받는 CA과 피드백들을 통해 액션들을 식별한다. 또한, 상태 공간과 CA들을 연결하여 컴포넌트의 상태 전환을 구체화한다.

- Σ_n = {타 컴포넌트로부터 수신받은 피드백} ∪ {타 컴포넌트에게 송신한 CA}
- $E \subseteq L \times \mathcal{L}(X) \times \Sigma \times 2^X \times L$

2) 사고 및 오작동 상태 정의

2단계에서는 1단계에서 정의한 모델을 수정하는 작업을 진행한다. 1단계의 모델에는 컴포넌트가 사고 및 오작동 상태에 도달하는 경우가 포함되어 있지 않다. 따라서 STPA 분석을 통해 얻은 사고와 위험원, UCA를 바탕으로 사고 상태 L_a, 오작동 상태 L_e와 오작동 액션 Σ_e를 정의한다. L_a는 STPA 분석 결과로 얻은 사고를 만족하는 상태로 정의된다. L_e는 UCA를 수행하고 나서 도달하는 컴포넌트의 오류 상태인 *stop_error* 상태와 시스템의 위험원을 만족하는 상태 *hazard*로 구성된다. Σ_e는 시스템의 UCA와 컴포넌트들의 오작동 *error*로 구성된다.

- L_a = {사고(loss)를 만족하는 상태}
- L_e = $\{stop\_errorvert(L, UCA) \rightarrow stop\_error\}$ ∪ $\{hazard | (L, error) \rightarrow hazard)\}$
- Σ_e = $\{UCA\} \cup \{error\}$

3) 오작동 확률 계산 및 반영

3단계에서는 오작동 확률을 계산하여 일반 상태에서 L_e로의 전환 및 확률을 정의한다. 오작동 확률은 구현하는 시스템의 사고 데이터를 바탕으로 적절한 수치를 결정한다. 상태간 확률 전환은 일반적으로 그림 3과 같이 구현한다.

L_e로의 전환은 제어 명령이 송신 혹은 수신과 동시에 확률에 따라 결정된다. 이로인해 그림 3과 같이 제어 명령 실행 후 확률에 따라 여러 분기점으로 나눠지게 구현해야 한다. 하지만 UPPAAL의 모든 제어 명령들은 항상 결정적(Deterministic)이게 수행되어야 한다. 따라서 이러한 점을 해결하기 위해 본 논문에서는 그림 4와 같이 제어 명령을 수행 후에 Committed 상태를 이용해 구현한다.

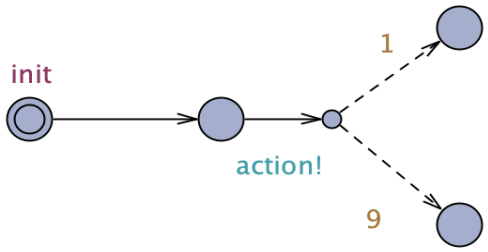


그림 3. 일반적인 확률 상태 전환
Fig. 3. General probability state transitions

Committed 상태에 도달하면 해당 상태의 전환부터 반드시 수행되어야 한다. 따라서 Committed 상태 후 확률 전환을 연결하면 제어 명령이 결정적으로 수행되게 하면서 확률 전환도 수행할 수 있기 때문에 그림 4와 같이 구현한다.

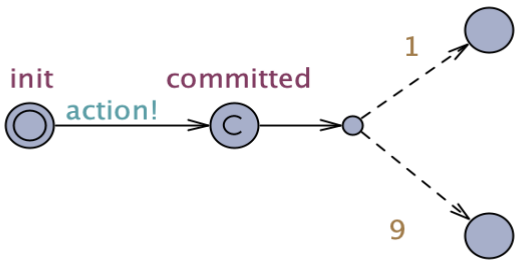


그림 4. Committed 상태를 이용한 확률 상태 전환
Fig. 4. Probability transition using committed state

3.2 쿼리 생성 방법

1) 손실 시나리오 식별

1단계에서는 STPA 분석 결과를 통해 생성된 손실 시나리오 중 2가지에 해당된 시나리오를 다룬다.

a) 액츄에이터의 오작동과 UCA 수행된 경우

a번에 속한 손실 시나리오는 액츄에이터의 오작동과 UCA가 동시에 수행된 경우를 말한다. 이 시나리오들은 UCA가 사고에 직접적으로 영향을 끼치는 것들이지만, 액츄에이터의 오작동도 간접적으로 영향을 미치는 시나리오이다. 따라서 컴포넌트들의 상호작용뿐만아니라 액츄에이터의 오작동을 포함한 위험 평가 수행을 위해 a번의 시나리오를 다룬다.

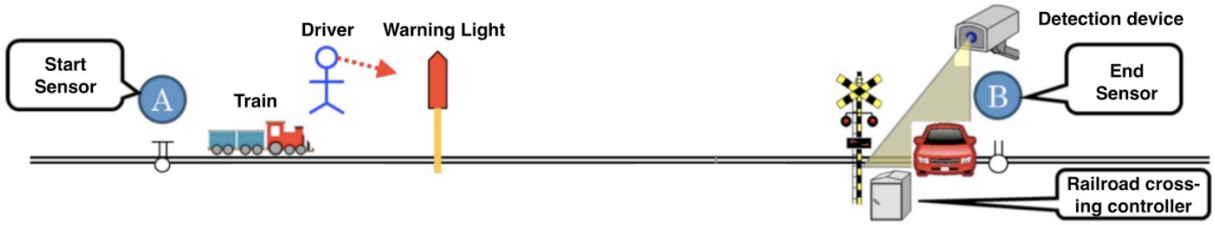


그림 5. 철도 건널목 시스템 구성도[17]
Fig. 5. Railroad crossing system configuration[17]

b) 액츄에이터의 오작동과 CA가 수행된 경우

b번에 속하는 손실 시나리오는 UCA가 아닌 CA가 생성되었지만 CA를 수신한 액츄에이터의 오작동이 발생해 사고가 발생하는 경우이다. 액츄에이터가 사고에 직접적으로 영향을 끼치는 시나리오이다. 따라서 액츄에이터의 오작동의 위험성을 파악하기 위해 b번의 시나리오를 다룬다.

2) 손실 시나리오 변환

2단계에서는 식 (1)를 바탕으로 손실 시나리오를 변환한다. 손실 시나리오는 컴포넌트의 UCA나 고장으로 인해 사고가 발생하는 경우를 기술한다. 따라서 여러 UCA가 수행되어 사고로 이어지는 경우를 쿼리로 표현해야 한다. 본 논문에서는 손실 시나리오를 쿼리로 변환하기 위해 오작동 상태 L_e와 사고 상태 L_a를 이용한다. L_e는 컴포넌트가 UCA 혹은 오작동을 수행 후 도달하는 상태로 구성된다. 따라서 L_e에 포함된 상태에 도달하면 UCA나 오작동이 수행되었다는 것을 알 수 있다. 따라서 손실 시나리오에 명시된 특정 UCA를 수행한 상태에 해당하는 L_e와 L_a를 모두 만족하면 동시에 손실 시나리오를 만족한다. 따라서 다음과 같이 상황을 가정한 경우 손실 시나리오를 식 (2)와 같이 변환한다.

[가정]

- $L_a = accident$
- $(L, error1) \rightarrow stop\_error1 \in L_e$
- $(L, UCA1) \rightarrow stop\_error2 \in L_e$
- 컴포넌트 T_1, T_2, T_3
- T_1의 오작동으로 인해 T_2의 UCA가 발생해 T_3가 사고에 도달한 경우

$$Pr[SMCBounds](<>(T_1.L_e \text{ and } T_2.L_e \text{ and } \text{ and } T_n.accident)) \quad (2)$$

Ⅳ. 철도 건널목 시스템의 위험 평가

본 논문에서 제안한 방법을 철도 건널목 시스템에 적용하였다. 철도 운송은 현대 사회의 필수적인 부분으로 사람과 물건을 안전하고 효율적으로 이동할 수 있는 수단을 제공한다. 하지만 IT 기술의 발전으로 철도 시스템의 복잡성이 증가하여 안전 사고 발생 가능성 및 위험도가 증가하였다. 이러한 이유로 본 논문에서는 제안한 방법을 적용하여 철도 시스템 중 하나인 철도 건널목 시스템에 대한 STPA 위험 평가를 수행하였다.

4.1 철도 건널목 시스템 모델링

본 논문에서 사용한 철도 건널목 시스템은 Yang[17]과 일본 IPA의 STAMP/STPA 가이드[18]를 참고하였다. 이 시스템은 기차가 건널목으로 접근할 경우 차단기를 내려 사람이나 차량의 진입을 통제한다. 그 후 감지기를 작동시켜 건널목 안에 차량 혹은 사람의 유무를 탐지해 비상 상황 시 경고등을 작동시켜 기차를 정지시키는 시스템이다. 철도 건널목 시스템의 구성도는 그림 5와 같다.

시스템은 시작과 마지막 센서, 경고등, 감지기, 컨트롤러, 차단기, 기관사, 열차, 차량/사람으로 구성된다. 본 논문에서는 사람의 행동이 아닌 SW, HW로 인한 손실 시나리오를 분석하는 것에 초점을 맞추었다. 따라서 그림 5의 구성 요소 중 기관사를 제외한다. 또한, 컴포넌트가 오작동하여 제어 명령을 아예 수행하지 못하거나 잘못 수행하는 두 가지만 다뤄 타이밍에 관련된 사고는 다루지 않는다.

1) 시스템 구조, 동작 및 제약사항 식별

시스템의 상태공간, 동작을 식별하기 위해 요구사항을 바탕으로 시스템의 구조와 동작을 분석한다. 본 논문에서는 기존 시스템과 달리 기관사를 제외한 시스템을 다루기 때문에 요구사항을 재정의하였다. 재정의한 시스템의 요구사항은 표 1과 같다.

표 1. 철도 건널목 시스템 요구사항
Table 1. Requirements of railroad crossing system

Requirements	
#1	When the train approaches the start sensor, the controller lowers the barrier and triggers the detector device.
#2	The detector device checks to see if there are vehicles or people in the crossing.
#3	If there are vehicles or people in the crossing, the detector device sends an ON signal to the warning light.
#4	The warning light turns on the stop sign when it is signaled ON.
#5	The train stops after a certain amount of time if the stop sign is on.
#6	When the train has passed the end sensor, the controller raises the barier and stops the detector device.
#7	People or vehicles can only enter the crossing when the barrier is open.

표 1의 요구사항을 통해 시스템의 상태 공간은 대부분 컴포넌트들의 차단기 열림, 닫힘 혹은 감지기 탐지, 미탐지 등 ON, OFF의 상태를 가지는 것을 알 수 있다. 다만, 기차의 위치에 따라 각 컴포넌트들이 동작하게 되므로 기차는 현재 자신의 위치를 상태로 가지게 된다.

시스템의 제어 명령과 피드백은 그림 6의 컨트롤 스트럭처를 바탕으로 아래와 같이 정의된다.

- 컨트롤러 → 감지기, 탐지기: ON, OFF
- 기차(센서) → 컨트롤러: COME, PASS
- 차량 → 감지기: IN, OUT
- 감지기 → 경고등: ON, OFF
- 경고등 → 기차: Brake

시스템의 시간 제약 사항은 요구사항 #5에 따라 기차가 브레이크 후 정지하는 데까지 걸리는 시간이 존재한다. 또한, 요구사항에 명시되지 않았지만 기차가 다음 상태 혹은 위치로 이동하는데까지 걸리는 시간이 존재한다.

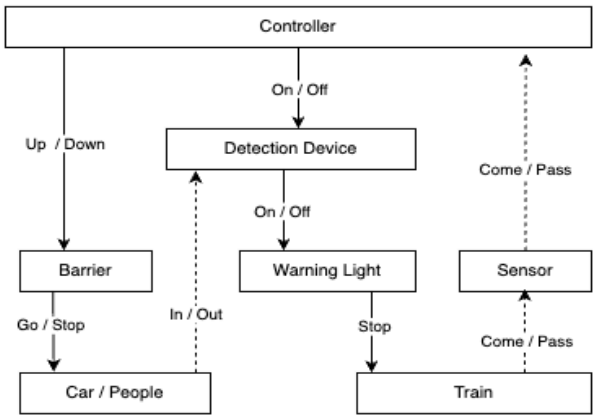


그림 6. 철도 건널목 시스템의 컨트롤 스트럭처
Fig. 6. Control structure of railroad crossing system

2) 사고 및 오작동 상태 생성

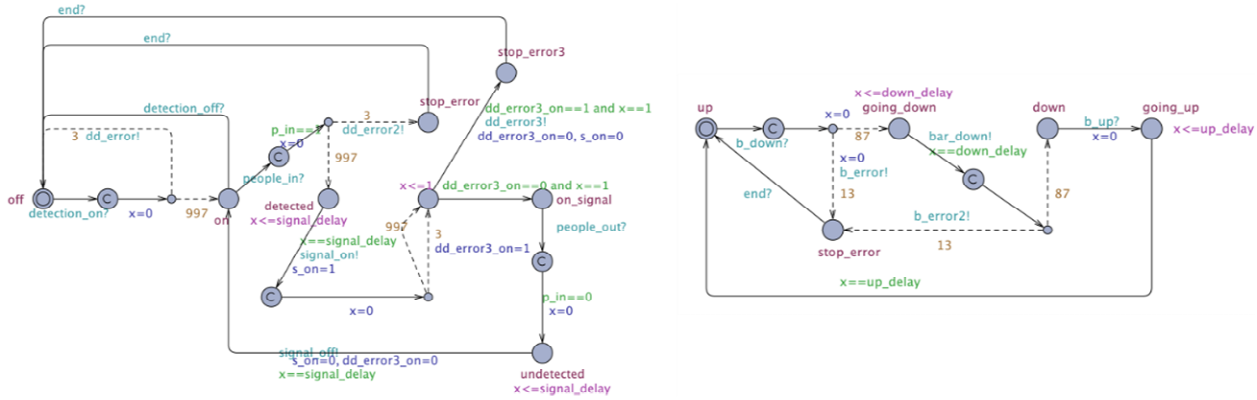
철도 건널목 시스템에 STPA를 분석하여 얻은 결과를 통해 L_e, Σ_e를 정의하였다. STPA로 분석한 결과 시스템의 사고는 기차가 사람 혹은 차량과 충돌하는 경우, 위험원은 경고등이 켜지지 않는 경우이다. 따라서 시스템의 L_e는 기차가 충돌한 상태와 경고등이 켜지지 않은 경우, 그 외 UCA를 수행한 후의 상태로 정의하였다. Σ_e는 시스템의 UCA와 감지기, 차단기, 경고등의 고장 액션으로 구성된다.

3) 오작동 확률 계산 및 반영

본 논문에서는 철도 시스템의 오작동 확률을 계산하기 위해서 미국의 FRA 철도 사고 보고서를 이용하였다[19]. 컴포넌트 별 전체 오작동 확률 P_e을 식 (3)를 이용해 계산하였다. 감지기의 오작동 확률의 경우, 감지기 정보는 사고 보고서에 따로 존재하지 않아 별도로 다른 확률을 적용하였다.

$$P_e = \frac{\#acc_c}{\#acc} \tag{3}$$

식 (3)의 $\#acc_c$는 컴포넌트가 원인인 사고 횟수를 나타낸다. $\#acc$은 전체 사고 수다. 또한, 식 (4)를 통해 컴포넌트의 각 오작동 확률을 계산하였다. 각 컴포넌트의 오작동 종류를 정의하고, P_e를 오작동 종류 수로 나눠 각 오작동에 균등한 확률을 부여했다.



(a) (b)

그림 7. 철도 건널목 시스템 모델링 결과 (a) 탐지장치 (b) 차단기

Fig. 7. Models of railroad crossing system (a) Detection device (b) Barrier

$$p_e = \frac{P_e}{\#error_c} \tag{4}$$

식 (3)을 통해 차단기의 P_e는 26%, 경고등은 12%로 계산되었다. 감지기는 FRA에 있는 여러 고장 확률 중 "전원 고장 확률" 2%로 지정하였다. 또한, 차단기의 오작동 종류 2개, 경고등의 오작동 종류 1개, 감지기의 오작동 종류 3개로 정의하여 식 (4)를 통해 각각 13%, 12%, 0.3%의 확률을 계산하였다. 계산된 확률들은 실제 사고 보고서를 바탕으로 했지만, 보고서에 사고 상황에 대한 구체적인 정보가 존재하지 않아 실제 확률과 다를 수 있다.

1, 2, 3단계를 종합하여 그림 7과 같이 철도 건널목 시스템을 UPPAAL로 모델링하였다. 그림 7의 (a)는 감지기를 UPPAAL로 모델링한 모습이다. 감지기의 오작동인 고장과 두 가지 감지 오류에 0.3%의 확률을 부여해 구현하였다. 그림 7의 (b)는 차단기를 모델링한 결과이다. 컨트롤러가 Down 명령을 내렸지만 Down 명령이 수행되지 않은 경우와 Up 명령을 내리지 않았는데 Up 명령을 수행한 경우의 오작동을 구현하였고, 각각 13%의 오작동 확률을 부여하였다.

4.2 실험 및 결과

본 논문에서는 타이밍과 환경으로 인한 손실 시나리오는 제외하였다. 따라서 다음과 같이 총 4가지의 손실 시나리오를 쿼리로 변환하여 각각의 발생 가능성을 계산하였다.

1) 액츄에이터의 오작동과 UCA가 수행된 경우

(HS2-N-2) 차단기가 내려가지 않았을 때 감지기의 인식 알고리즘 오류로 차량/사람이 감지되지 않아 경고등에 ON 지시를 내리지 않음:

$$Pr[\leq 1000](<>((dd.stop\_error)) and (cp.accident))$$

(HS2-N-2) 쿼리는 차량/사람이 건널목에 있지만 감지기가 인식하지 못한 경우를 나타낸다.

쿼리의 dd와 cp는 UPPAAL의 템플릿 이름으로 각각 감지기와 차량/사람을 의미한다. *dd.stop_error* 상태는 건널목 안에 차량/사람이 있지만 감지기가 처음부터 감지를 못하는 오작동이 발생한 상태를 말한다. *cp.accident*는 차량/사람이 기차와 부딪혀 사고가 발생한 상태를 의미한다. 따라서 손실 시나리오와 같이 처음부터 감지기의 인식 불량으로 경고등이 켜지지 않아 사고가 발생하는 경우를 나타낸다.

(HS3-P-1) 차단기가 내려가지 않았을 때 감지기의 인식 알고리즘 오류로 감지 중간에 차량/사람을 인식하짐 못해 경고등에 OFF 지시를 내림:

$$Pr[\leq 1000](<>((dd.stop\_error3)) and (cp.accident))$$

(HS3-P-1) 쿼리는 감지기가 차량/사람을 감지하여 경고등에 ON 신호를 보낸 후에 인식 오류로 차량/사람을 감지하지 못해 OFF 신호를 보낸 경우의 확률을 구하는 쿼리이다. *dd.stop_error3* 상태는 감지기

가 경고등으로 처음 ON 신호를 보낸 후 올바르지 않은 OFF 신호를 보내고 난 후에 도달하는 상태이다. 따라서 위의 쿼리와 같이 두 상태가 동시에 만족하는 경우는 손실 시나리오를 만족한다.

2) 액츄에이터의 오작동과 CA가 수행된 경우

(HS2-N-1) 차단기가 내려가지 않았을 때 컨트롤러가 ON 지시를 내렸지만 감지기의 고장으로 작동하지 않고 차량/사람을 감지할 수 없어 경고등의 ON 지시를 내리지 않음:

$$Pr[\leq 1000](<>((dd.off))\,and\,(cp.accident))$$

(HS2-N-1) 쿼리는 감지기의 고장이 발생한 후 사고가 발생할 확률을 의미한다. 사고가 일어날 때까지 *dd.off* 상태라는 것은 컨트롤러가 감지기를 작동시켰지만 고장이 나 켜지지 않았다는 것을 의미한다. 따라서 손실 시나리오와 같이 감지기의 고장으로 사고가 발생한 경우를 나타낸다.

(HS4-N-1) 차단기가 내려가지 않았을 때 경고등의 고장으로 정지 신호가 켜지지 않음:

$$Pr[\leq 1000](<>((wl.stop\_error))\,and\,(cp.accident))$$

(HS4-N-1) 쿼리는 감지기가 ON 신호를 보낸 후 경고등의 고장으로 인해 정지 신호가 켜지지 않는 경우를 나타낸다. 위 쿼리의 wl은 경고등을 말하는 것으로 *wl.stop_error* 상태는 감지기가 ON 신호를 보냈을 때 고장으로 인해 정지 신호를 키지 않을 때 도달하는 상태이다. 따라서 위의 쿼리와 같이 두 상태 만족되면 손실 시나리오 또한 만족된다.

위의 4가지 쿼리를 실행시켜 손실 시나리오의 발생 가능성을 계산하였다. 각 손실 시나리오의 발생 가능성 결과는 표 2와 같다. 액츄에이터의 오작동을 포함하여 계산된 각 시나리오의 발생 가능성은 표 3의 Probable와 Occasional에 해당하는 수치들이다. 이 수치들은 위험도가 2~4를 가질 수 있어 모든 시나리오가 높은 위험도를 가질 가능성이 존재한다. 따라서 이 손실 시나리오들은 높은 위험도를 가질 수 있기 때문에 액츄에이터의 오작동까지 고려한 위험도 계산이 필요하다.

표 2. 손실 시나리오의 발생 가능성 계산 결과
Table 2. Results of the likelihood of loss scenarios

Loss scenario	Likelihood
HS2-N-1	[0.0297222, 0.129219]
HS2-N-2	[0.0169465, 0.116203]
HS3-P-1	[0.00790082, 0.106991]
HS4-N-1	[0, 0.0981446]

표 3. 손실 시나리오 발생가능성 별 위험도
Table 3. Risk by likelihood of loss scenarios

Frequency classification	Probability	Risk levels
Frequent	$1e-1 \leq$	[3, 4]
Probable	$1e-2 \leq Pr < 1e-1$	[2, 4]
Occasional	$1e-3 \leq Pr < 1e-2$	[2, 4]
Remote	$1e-4 \leq Pr < 1e-3$	[1, 3]
Improbable	$1e-5 \leq Pr < 1e-4$	[1, 2]
Incredible	$< 1e-5$	1

V. 결론 및 향후 연구

STPA는 시스템 이론 기반의 위험원 분석 방법으로 시스템의 구조, 동적 변화 등을 고려하여 시스템의 위험원을 분석하는 데 사용된다. 이는 SFMEA, FTA와 같은 전통적인 위험원 분석 방법과 달리, 시스템의 고장으로 발생하는 사고뿐만 아니라 컴포넌트간의 UCA로 발생할 수 있는 사고까지 분석 가능하다. 이러한 이점으로 복잡성이 증가한 현대 시스템의 위험원 분석 시 STPA가 사용되고 있다. 그러나 STPA에는 공식적으로 제공하는 정량 위험 평가 방법이 따로 존재하지 않는다. 효율적인 안전 대책 수립을 위해서는 위험원들의 우선순위 부여가 필요하고, 이를 위해서는 각 위험원들의 위험성을 객관적으로 비교할 수 있는 수치가 필요하다. 이에 최근 STPA 정량 위험 평가에 대해 연구되고 있지만 액츄에이터의 오작동은 고려하지는 않는다. 액츄에이터의 오작동은 하위 시스템의 UCA를 유발할 수 있다. 따라서 액츄에이터의 오작동이라는 불확실성까지 고려한 위험 평가가 필요하다.

본 논문에서는 액츄에이터의 불확실성을 고려한 STPA 위험 평가를 위해 시스템의 오작동을 분석하고 PTA로 모델링하는 방법을 제안하였다.

제안한 방법을 철도 건널목 시스템에 적용하여 액츄에이터의 영향을 확인하였다. 이를 위해, 액츄에이터가 사고에 직접적 혹은 간접적으로 영향을 끼치는 경우에 속하는 손실 시나리오들에 한정하여 위험 평가를 수행하였다. 총 4가지의 손실 시나리오에 대해 평가를 한 결과, 4가지 시나리오 모두 최소 2에서 최대 4의 위험도를 가질 수 있었다. 이를 통해 액츄에이터의 오작동이 영향을 끼친 손실 시나리오들은 높은 시나리오를 가질 수 있고, 액츄에이터까지 고려한 위험도 계산이 필요함을 보였다.

본 논문에서는 기존에 UCA만 고려한 평가와 달리 액츄에이터의 불확실성까지 고려한 STPA 위험 평가를 수행하였다. 또한, 분석 대상 시스템을 UPPAAL로 변환하는 방법을 소개하였다. 이를 통해 기존보다 더 정확한 위험 평가 방법을 제안하였다는 기여가 존재한다. 하지만 논문에서 시스템의 각 오작동이 손실 시나리오 전체에 어느 정도의 영향을 끼치는 지 분석하지 않았다는 한계가 존재한다. 따라서 추후 논문의 한계에 대한 연구를 진행하고자 한다.

References

[1] N. G. Leveson and J. P. Thomas, "STPA HANDBOOK", MIT, Mar. 2018. https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf [accessed: Jun. 23, 2023]

[2] T. Ishimatsu, N. G. Levenson, J. Thomas, M. Katahira, Y. Miyamoto, and H. Nakao, "Modeling and Hazard Analysis Using Stpa", Proceedings of the 4th IAASS Conference, Making Safety Matter, May 2023.

[3] S. Gupta, G. V. Vinayak, and A. Gupta, "Software Failure Analysis in Requirement Phase", Proceedings of the 5th India Software Engineering Conference, pp. 101-104, Feb. 2012. https://doi.org/10.1145/2134254.2134272.

[4] Y. Qi, Y. Dong, X. Zhao, and X. Huang, "STPA for Learning-Enabled Systems: A Surveyand A New Method", arXiv preprint arXiv:2302.10588, Feb. 2023. https://doi.org/10.48550/arXiv.2302.10588.

[5] J. Zhang, H. Kim, Y. Liu, and M. A. Lundteigen, "Combining System-Theoretic Process Analysis and availability assessment: A subsea case", Proc. of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, Vol. 233, No. 4, Jan. 2019. https://doi.org/10.1177/1748006X18822224.

[6] N. Leveson, "Improving the Standard Risk Matrix using STPA", Journal of System Safety, Vol. 55, No. 2, pp. 13-22, 2019. https://doi.org/10.56094/jss.v55i2.44.

[7] T. J. Altenbach, "A Comparison of Risk Assessment Techniques from Qualitative to Quantitative", United States, 1995. https://www.osti.gov/servlets/purl/67753 [accessed: Aug. 12, 2023]

[8] T. Takata and H. Nakamura, "Applicability of Methods for Safety Analysis of Railway Signaling", Journal of the Korean Society for Railroad, Vol. 22, No. 7, pp. 538-549, Jul. 2019. https://doi.org/10.7782/JKSR.2019.22.7.538.

[9] M. Tsuji, T. Takai, K. Kakimoto, N. Ishihama, M. Katahira, and H. Iida, "Prioritizing Scenarios based on STAMP/STPA Using Statistical Model Checking", In 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp. 124-132, Oct. 2020. https://doi.org/https://doi.org/10.1109/ICSTW50294.2020.00032.

[10] Ministry of Science and ICT and TTA, "Risk Analysis Guide Using STPA", TTA, Dec. 2018. http://sw.tta.or.kr/fileDB/STPAGuide(TTA).pdf. [accessed: Jun. 23, 2023]

[11] J. Chang, R. Kwon, and G. Kwon, "STPA-RL: Analyzing Loss Scenarios in STPA with Reinforcement Learning", The Journal of Korean Institute of Information Technology, Vol. 21, No. 7, pp. 39-48, Jul. 2023. https://doi.org/10.14801/jkiit.2023.21.7.39.

[12] A. Legay, A. Lukina, L. M. Traonouez, J. Yang, S. A. Smolka, and R. Grosu, "Statistical model checking", Computing and software science: state of the art and perspectives, Springer International,

pp. 478-504, Oct. 2019. https://doi.org/10.1007/978-3-319-91908-9.

[13] P. Bulychev, A. David, K. G. Larsen, M. Mikučionis, D. B. Poulsen, A. Legay, and Z. Wang, "UPPAAL-SMC: Statistical model checking for priced timed automata", arXiv preprint arXiv:1207.1272, Jul. 2012. https://doi.org/10.48550/arXiv.1207.1272.

[14] G. Behrmann, K. G. Larsen, and J. I Rasmussen, "Priced timed automata: Algorithms and applications", Formal Methods for Components and Objects: Third International Symposium, FMCO 2004, pp. 162-182, 2005. https://doi.org/10.1007/11561163_8.

[15] A. David, K. G. Larsen, A. Legay, M. Mikučionis, D. B. Poulsen, J. V. Vliet, and Z. Wang, "Statistical model checking for networks of priced timed automata", Formal Modeling and Analysis of Timed Systems: 9th International Conference, FORMATS 2011, Vol. 6919, pp. 80-96, 2011. https://doi.org/10.1007/978-3-642-24310-3_7.

[16] UPPAAL, "Semantics of the Statistical Queries", https://docs.uppaal.org/language-reference/requirements-specification/smc_queries/ [accessed: Jun. 23, 2023]

[17] P. Yang, R. Karashima, K. Okano, and S. Ogata, "Automated inspection method for an STAMP/STPA-fallen barrier trap at railroad crossing", Procedia Computer Science, Vol. 159, pp. 1165-1174, 2019. https://doi.org/10.1016/j.procs.2019.09.285.

[18] Information-technology Promotion Agency (IPA), Software Reliability Enhancement Center (SEC), Software Reliability Enhancement Pro-Committee and System Safety and Reliability analysis WG, "Hajimete No STAMP/STPA, Jissenhen ~Beginning of thePracticing Part~", Mar. 2017. https://www.ipa.go.jp/sec/reports/20170324.html [accessed: Jun. 23, 2023]

[19] FRA, Homepage, https://safetydata.fra.dot.gov/OfficeofSafety/default.aspx. [accessed: Jun. 23, 2023]

저자소개

박 소 희 (Sohee Park)

2023년 2월 : 경기대학교 컴퓨터공학부(공학사)
2023년 3월 ~ 현재 : 경기대학교 SW안전보안학과 석사과정
관심분야 : 소프트웨어 공학, 소프트웨어 안전성, 머신 러닝

권 령 구 (Ryeonggu Kwon)

2011년 2월 : 경기대학교 컴퓨터과학과(이학사)
2013년 2월 : 경기대학교 컴퓨터과학과(이학석사)
2013년 3월 ~ 현재 : 경기대학교 컴퓨터과학과 박사과정
관심분야 : 기계 학습, 정형 합성

권 기 현 (Gihwon Kwon)

1985년 2월 : 경기대학교 전자계산학과(이학사)
1987년 8월 : 중앙대학교 전자계산학과(이학석사)
1991년 2월 : 중앙대학교 전자계산학과(공학박사)
1991년 2월 ~ 현재 : 경기대학교 컴퓨터공학부 교수
1999년 ~ 2000년 : 미국 CMU 전산학과 방문교수
2006년 ~ 2007년 : 미국 CMU 전산학과 방문교수
2014년 ~ 2016년 : 한국정보과학회 소프트웨어공학 소사이어티 회장
2021년 ~ 현재 : 경기대학교 SW중심대학 사업단장
2022년 ~ 현재 : 경기대학교 소프트웨어경영대학 학장
관심분야 : 소프트웨어 공학, 소프트웨어 안전성, 정형 검증및 정형 합성