STPA를 활용한 협업 가상물리시스템의 안전성 테스트 케이스 생성

Generation of Safety Test Cases for Cooperative Cyber-Physical Systems Using STPA

허윤아 유준범

Dependable Software Lab.

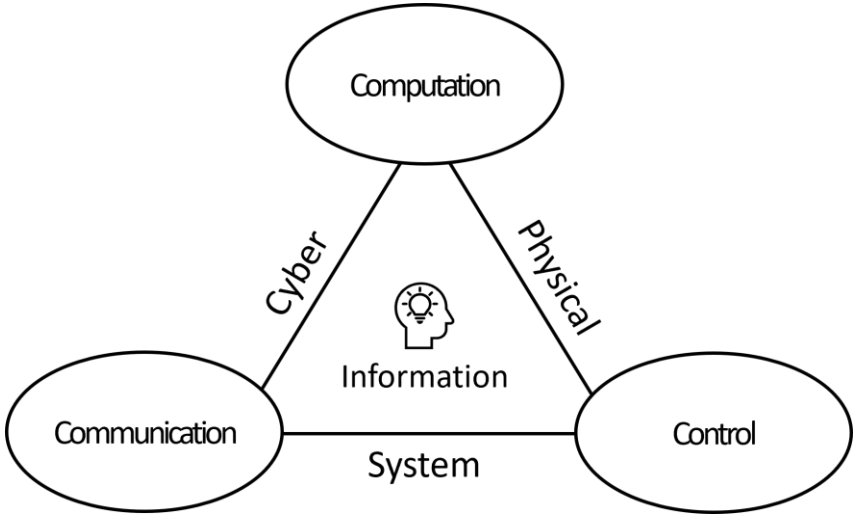
Konkuk University

2024.02.01

Table of Contents

Introduction

- 가상물리시스템 (Cyber-Physical System, CPS)



Real-time system

&

Safety-critical system

Introduction

- 서로 다른 여러 CPS의 협업 → 공동의 목표 달성
  - 협업 시 emergent property와 새로운 hazard가 나타날 수 있음

    * Emergent property: 시스템 이론에서, 구성 요소가 아닌 시스템 수준에서 발견되는 특성

- CPS ⊂ Safety-critical system

  → 개별 CPS 뿐 아니라 협업 CPS의 안전성 확보 필요
  - 안전성 확보를 위한 활동: 위험 분석, 리스크 평가, 안전 요구사항 정의, **안전성 검증** 등

Introduction

- 서로 다른 여러 CPS의 협업 → 공동의 목표 달성
  - 협업 시 emergent property와 새로운 hazard가 나타날 수 있음

    * Emergent property: 시스템 이론에서, 구성 요소가 아닌 시스템 수준에서 발견되는 특성

- CPS ⊂ Safety-critical system

  → 개별 CPS 뿐 아니라 협업 CPS의 안전성 확보 필요
  - 안전성 확보를 위한 활동: 위험 분석, 리스크 평가, 안전 요구사항 정의, **안전성 검증** 등

- 안전성 검증 중 안전성 테스팅을 위하여

  STPA 결과로부터 안전성 테스트 케이스를 수동으로 생성하는 프로세스 제안

Background

- Systems-Theoretic Process Analysis (STPA)
  - 위험 분석 (Hazard Analysis, HA) 기법의 한 종류
  - 시스템 이론에 기반한 causality model인 Systems-Theoretic Accident Model and Process (STAMP)를 위해 제안된 기법
  - 4-step Process

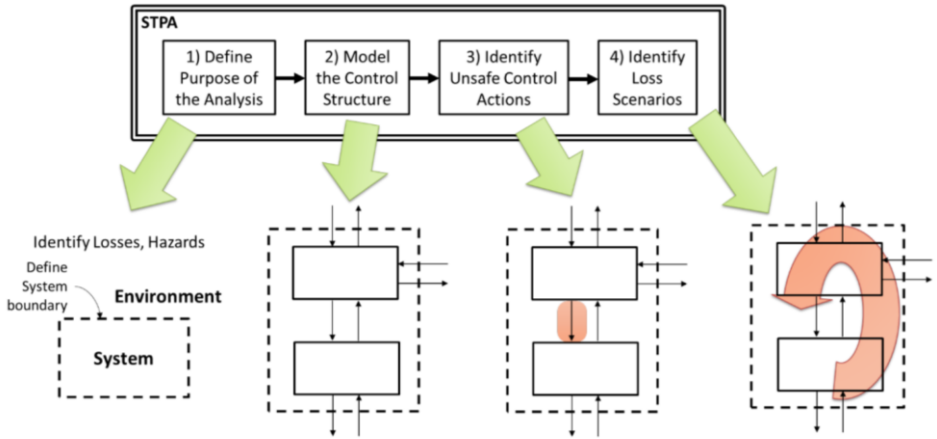
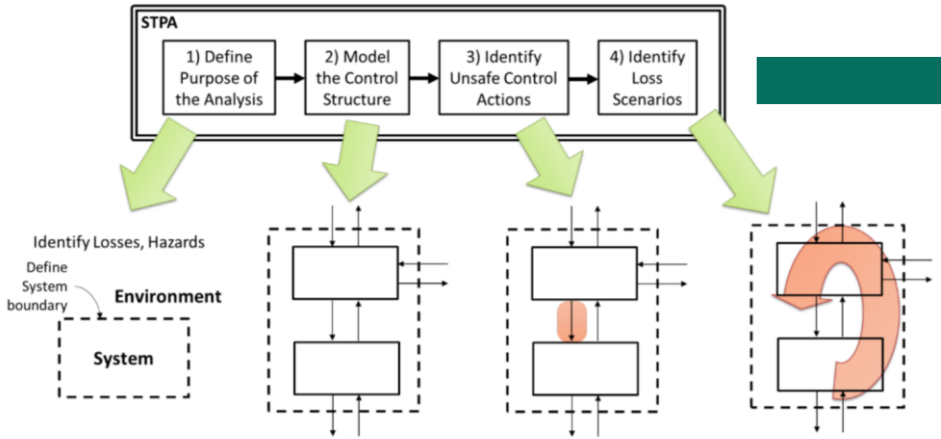


Figure 2.1: Overview of the basic STPA Method

*Ref: N. G. Leveson, J. P. Thomas. "STPA Handbook," 2018

Background

- Systems-Theoretic Process Analysis (STPA)
  - 위험 분석 (Hazard Analysis, HA) 기법의 한 종류
  - 시스템 이론에 기반한 causality model인 Systems-Theoretic Accident Model and Process (STAMP)를 위해 제안된 기법
  - 4-step Process



안전성을 확보하기 위한 후속 조치에 활용 가능

Figure 2.1: Overview of the basic STPA Method

*Ref: N. G. Leveson, J. P. Thomas. "STPA Handbook," 2018

Background

- Types of causal factors of loss scenario
  - UCA가 발생하도록 하는 원인
    1) 안전하지 않은 controller behavior
    2) 부적절한 feedback / information
  - CA가 잘못 실행되거나 실행되지 않아서 hazard가 발생하는 원인
    3) actuator를 포함하는 control path에서의 문제
    4) 실제로 control에 따라 동작하는 controlled process의 문제

Background

Quality Attribute Scenario (QAS)

Background

Quality Attribute Scenario (QAS)

QA: 시스템을 둘러싼 이해관계자들의 요구 (needs) 중
가치를 창출할 수 있으며 측정 가능한 (measurable) 비기능적 속성
(안전성 (safety), 신뢰성 (reliability), 확장성 (scalability), 사용성 (usability) 등을 포함)

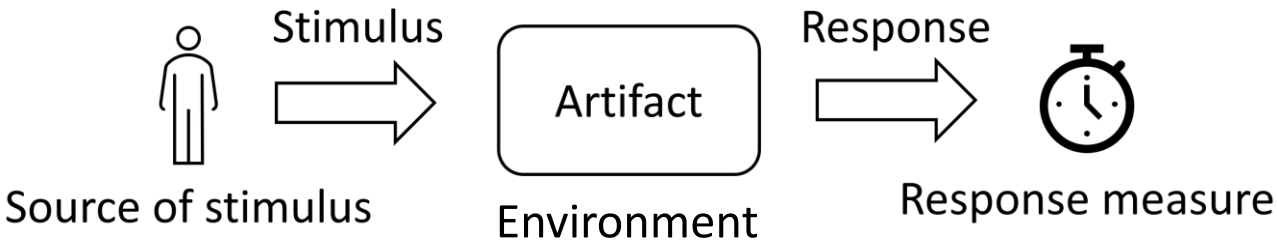
Background

Quality Attribute Scenario (QAS)

: QA에 대한 요구사항을 시나리오의 형태로 구체화한 것

Background

QAS의 구성



- Source of stimulus: 사람, 다른 시스템 등 stimulus를 제공하는 개체
- Stimulus: Source에 의해서 artifact에게 주어지는 이벤트
- Artifact: Stimulus가 도착하는 대상 시스템의 전체 또는 일부 혹은 시스템의 집단
- Response: Stimulus가 도착한 이후 artifact의 동작
- Response measure: 테스트를 위한, response에 대한 측정 가능한 값
- Environment: 시스템의 state 등 시나리오가 발생하는 환경의 집합

Background

- Safety Verification (안전성 검증)
  - HA의 후속 활동 중 하나
  - 일반적인 SW V&V (Verification & Validation)에서와 마찬가지로,
    동적 분석 (dynamic analysis)과 정적 분석 (static analysis)이 존재
- Safety Testing (안전성 테스팅)
  - 안전성 검증 중 동적 분석의 대표적 예시
  - HA의 결과물 → Safety Test Case (TC) 생성에 활용 가능

Previous Work

- 석사 학위논문에서 처음 제안

  (A Study on Identification of Quality Attribute Scenario for Safety of Cooperating Cyber-Physical Systems Using Systems-Theoretic Process Analysis)

- STPA의 결과를 활용하여 QAS 도출

  → 협업 CPS의 요구사항 / 설계 명세 등을 안전성 측면에서 보완

- 협업하는 각각의 CPS를 전체 CPS의 구성 시스템으로 간주

  → 각 구성 CPS와 전체 CPS에 대한 분석을 별도로 수행

- UCA와 Loss scenario를 활용하여 QAS를 식별하는 방안 제안

- 초기 형태의 guideword 제안 → QAS에서의 response measure로 활용

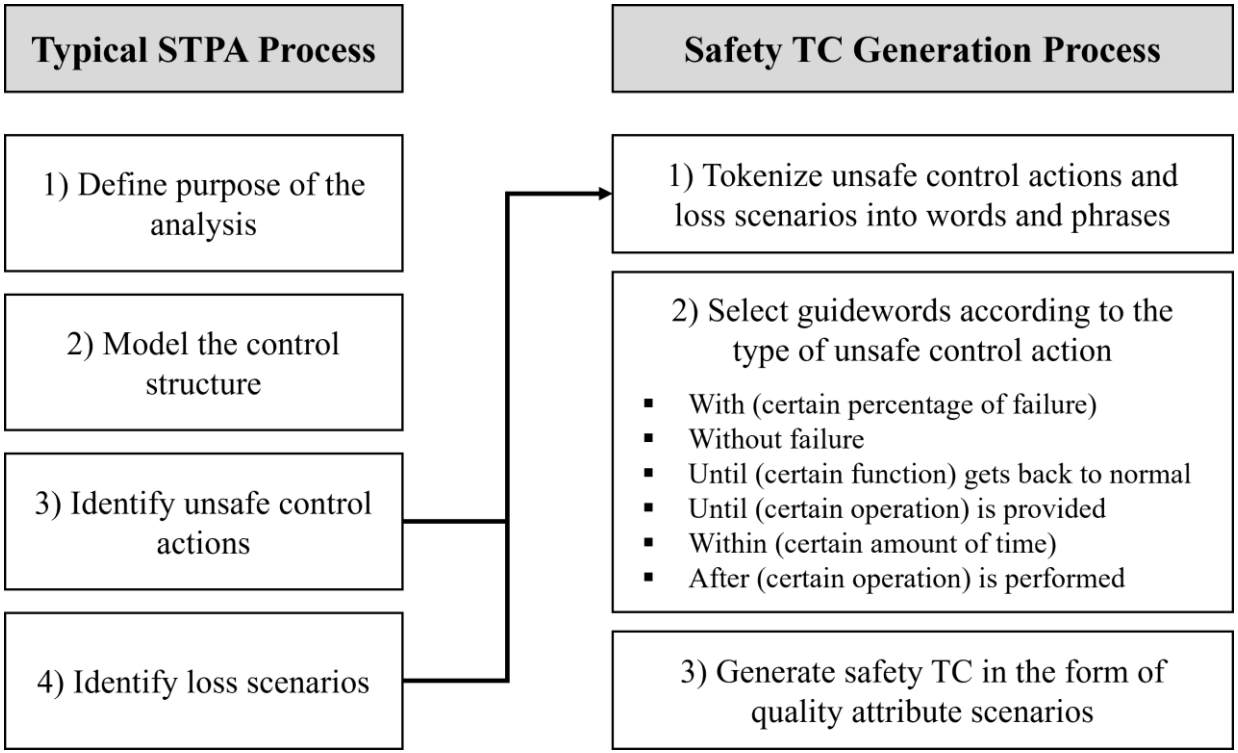
  - 'without failure, 'until (operation), 'within (time)', 'after (operation)'

Safety Test Case Generation Process Using STPA

본 연구의 목표

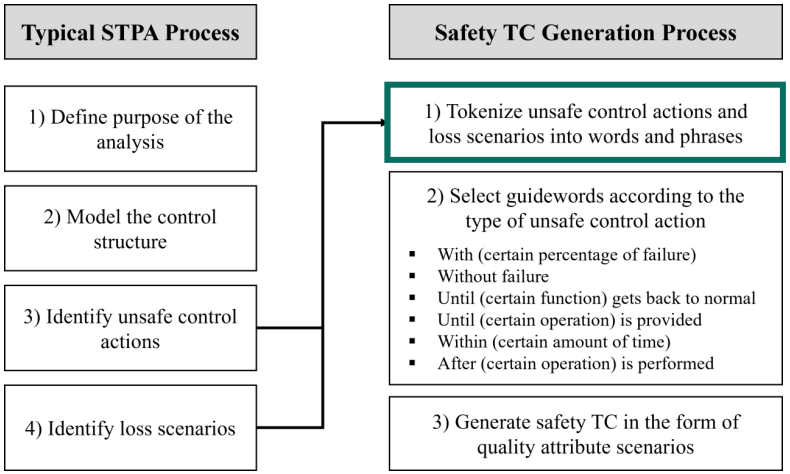
: **STPA를 활용한 Safety Test Case의 생성**

Safety Test Case Generation Process Using STPA

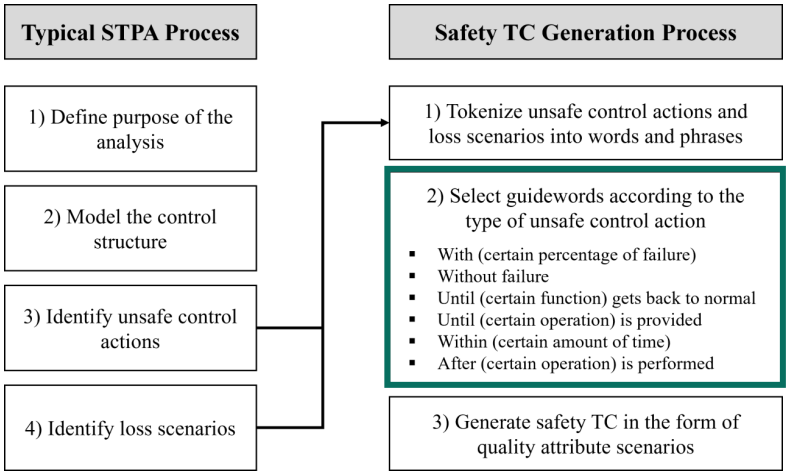


Safety Test Case Generation Process Using STPA

1) UCA와 loss scenario를 필요에 따라 단어 혹은 절의 형태로 토큰화 (tokenize)



Safety Test Case Generation Process Using STPA

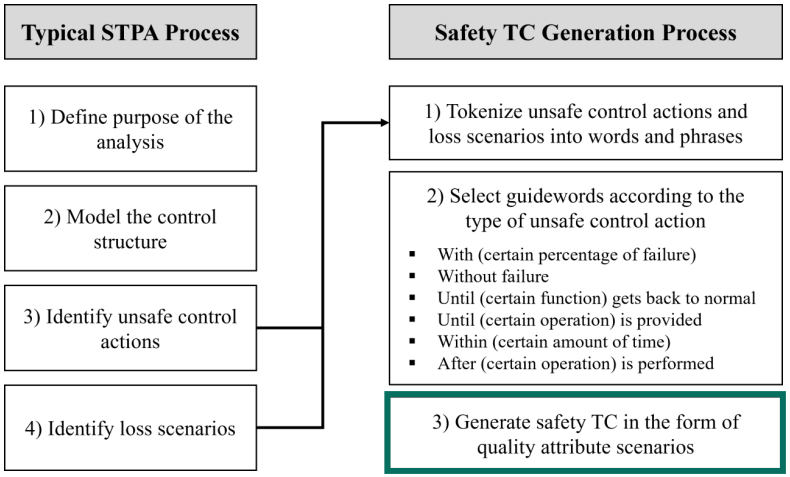


1) UCA와 loss scenario를 필요에 따라 단어 혹은 절의 형태로 토큰화 (tokenize)

2) 적절한 guideword 선정

- With (certain percentage of failure)
- Without failure
- Until (certain function) gets back to normal
- Until (certain operation) is provided
- Within (certain amount of time)
- After (certain operation) is performed

Safety Test Case Generation Process Using STPA



1) UCA와 loss scenario를 필요에 따라 단어 혹은 절의 형태로 토큰화 (tokenize)

2) 적절한 guideword 선정

- With (certain percentage of failure)
- Without failure
- Until (certain function) gets back to normal
- Until (certain operation) is provided
- Within (certain amount of time)
- After (certain operation) is performed

3) 1과 2의 결과를 활용하여 QAS의 형태로 safety TC 생성

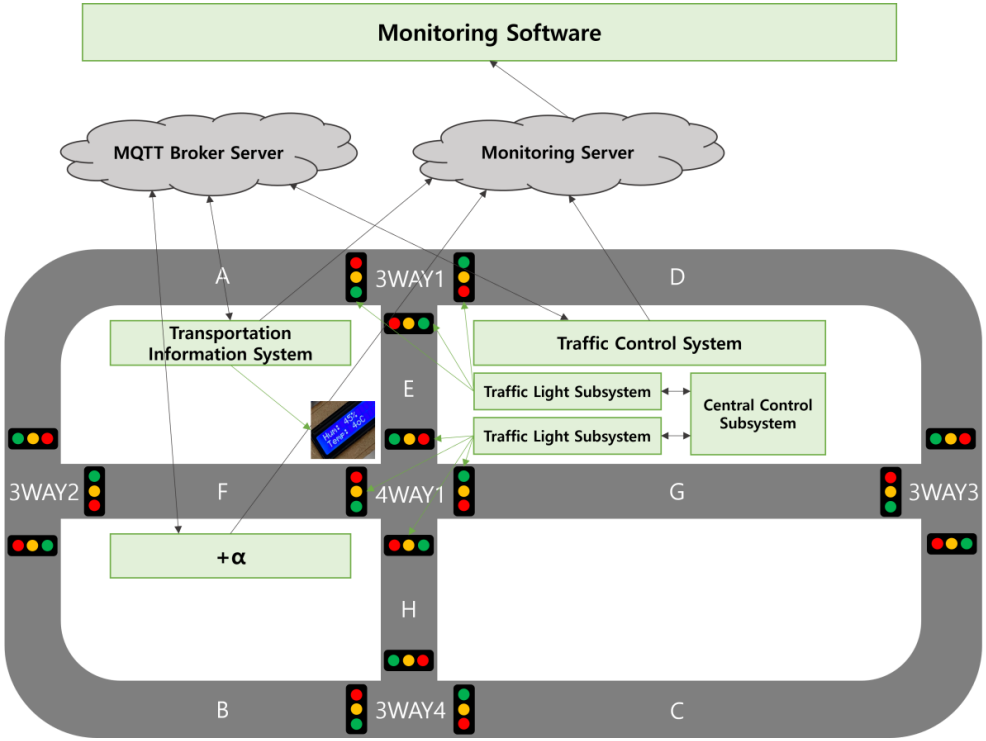
Safety Test Case Generation Process Using STPA

항목	사용 가능한 값
Source	▪ Operator ▪ Sensor ▪ I/O device ▪ Controller ▪ (internal/external) Data source
Stimulus	Source가 제공하는 데이터 혹은 제어명령
Environment	([1]에서 제공하는 safety general scenario) ▪ In normal operation ▪ In degraded operation ▪ In manual operation ▪ In recovery mode
Artifact	▪ Controller ▪ Controlled process ▪ I/O device
Response	토큰화한 STPA 결과 중 동작에 대한 부분의 내용에 반대되는 내용 또는 아래에 기술된 것처럼 [1]에서 제공하는 safety general scenario의 response 항목에 대한 값 중에서 선택한다. Unsafe state를 인식하고 다음 중 하나 이상을 따른다: ▪ Avoid the unsafe state ▪ Recover ▪ Continue in degraded or safe mode ▪ Shut down ▪ Switch to manual operation ▪ Switch to a backup system ▪ Notify appropriate entities (people or systems) ▪ Log the unsafe state (and the response to it)
Response measure	식별하는 사람의 판단에 따른 적절한 guideword 사용

Case Study

- Testbed system

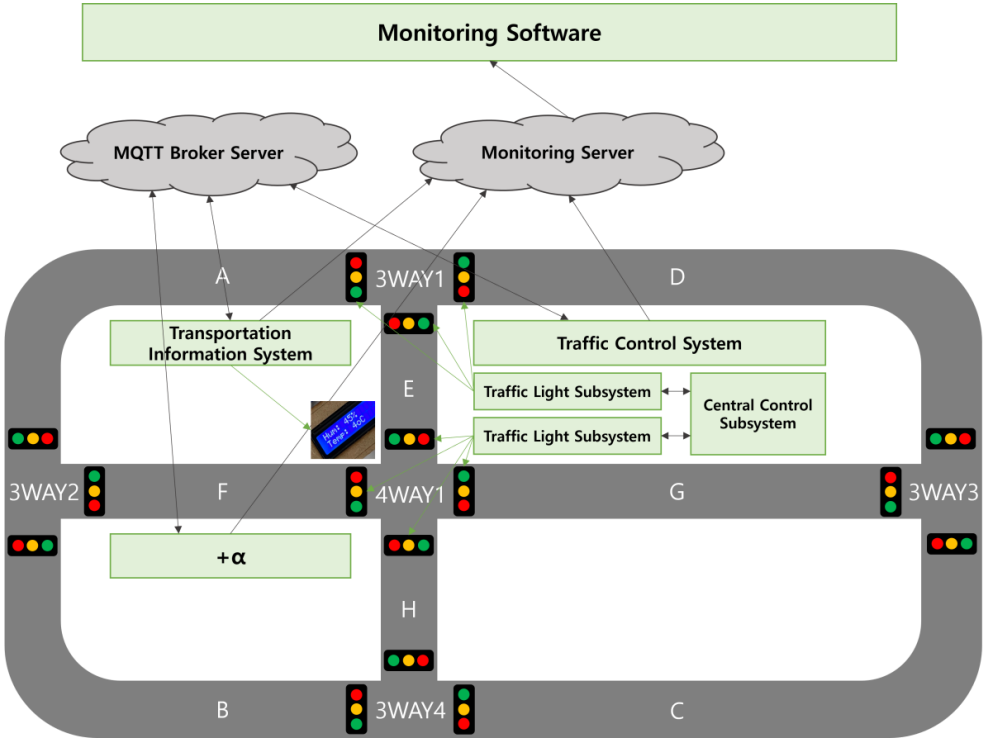
  : 간단한 Intelligent Transportation System의 예시



Case Study

- Testbed system

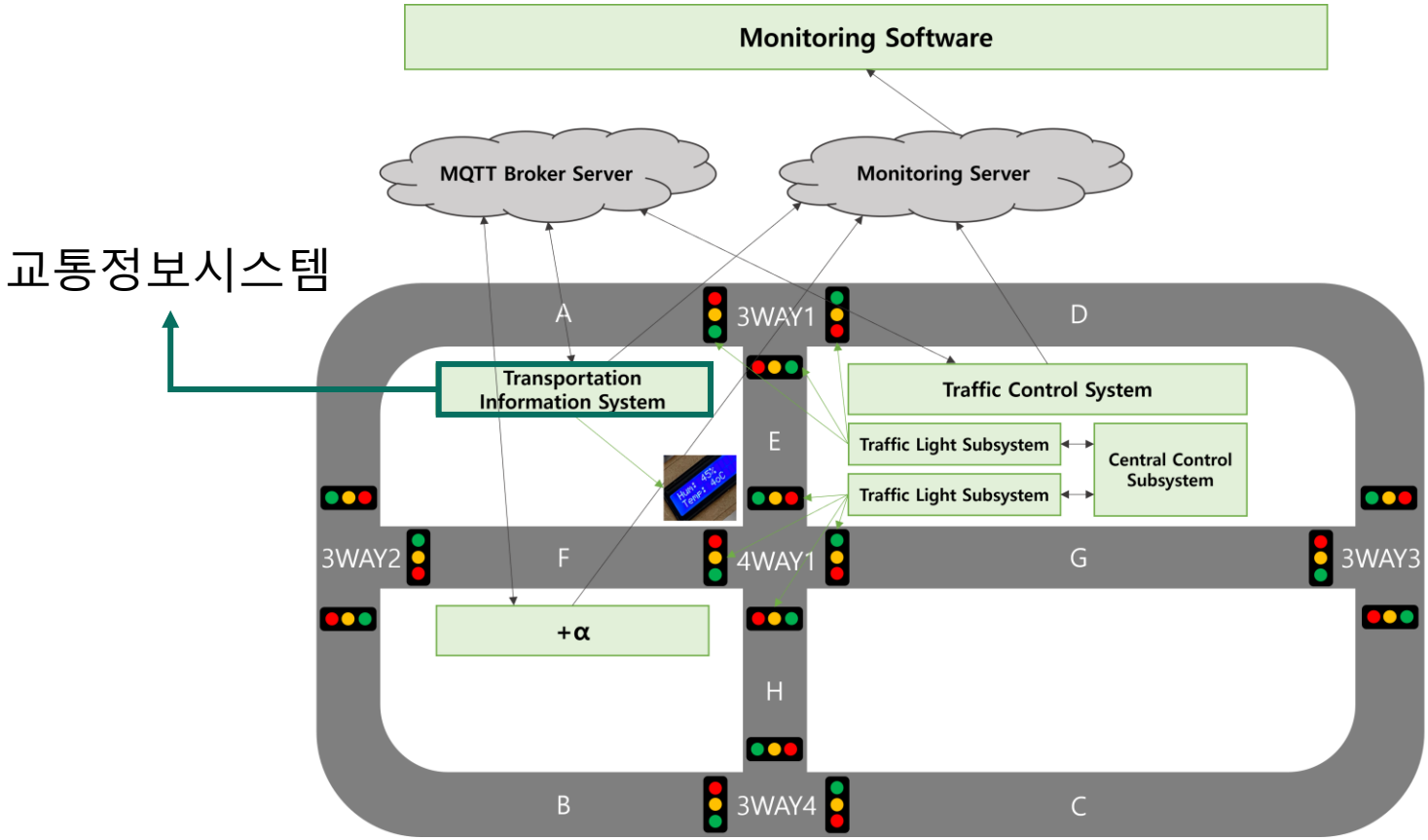
  : 간단한 Intelligent Transportation System의 예시



Common goal: 도로교통시스템을 외부 환경에 맞춰 적응형으로 제어한다

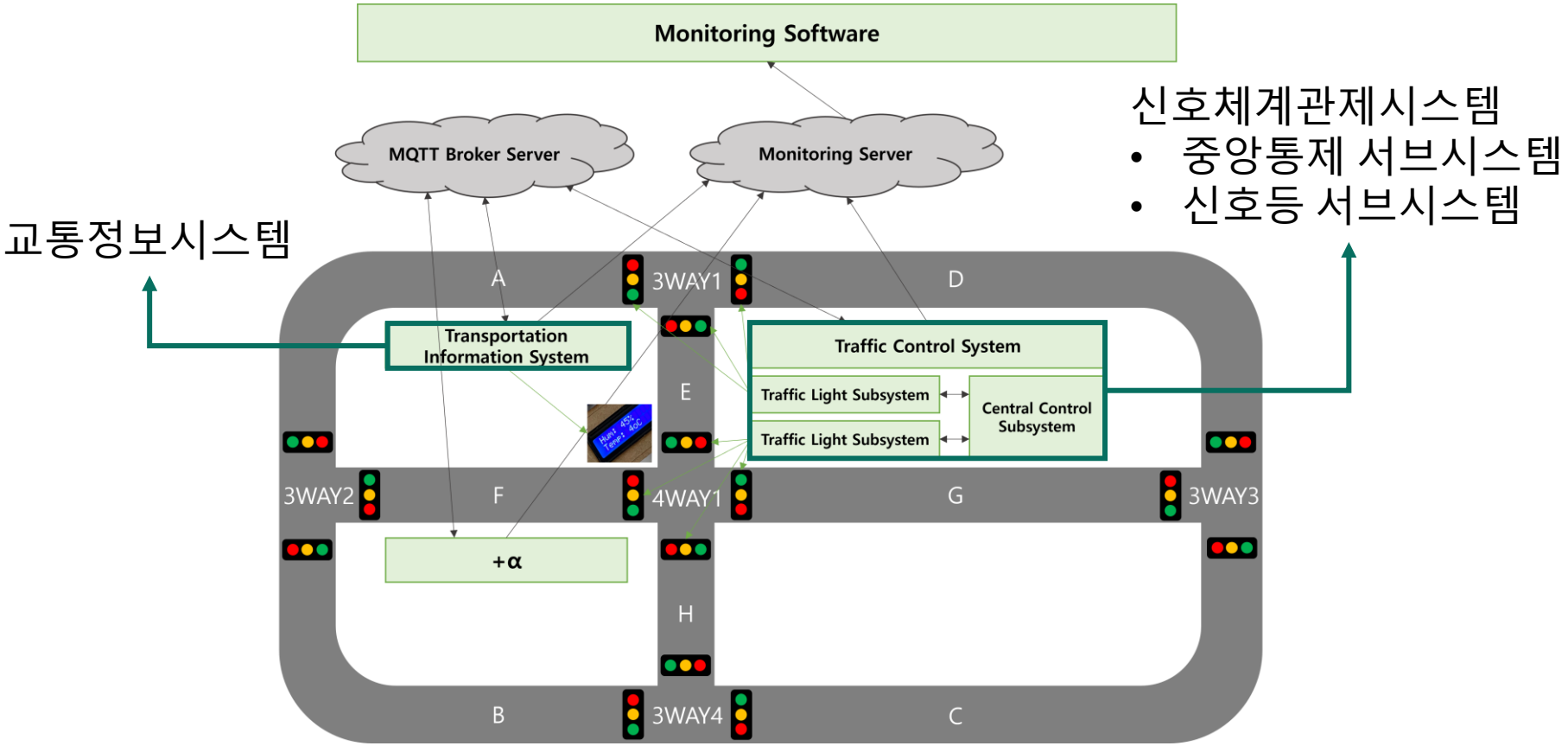
Case Study

- Testbed system



Case Study

- Testbed system



Case Study

- HW
  - 아두이노 보드 + 각종 모듈 (통신, 센서, 디스플레이 등)
- SW
  - 아두이노 보드 상의 SW 컨트롤러로 각 모듈 제어
- 정보 교환을 위한 MQTT 브로커 서버

  * MQTT: Message Queuing Telemetry Transport, IoT 시스템에서 통신 프로토콜로 활용
- 모니터링 시스템 (서버 + SW)

Case Study

STPA 수행 결과 예시

- UCA-13: 관리자가 교통정보시스템이 MQTT 서버를 통해 긴급 정보를 제공하지 않는 경우에 중앙통제 서브시스템에 수동 제어 명령을 제공하지 않는다.
  - 관련 hazard: WiFi 통신 오류, 제어 명령의 누락
- UCA-19: 중앙통제 서브시스템의 SW 컨트롤러가 교통정보시스템으로부터 MQTT 서버를 통해 긴급 정보가 제공된 경우에도 RF (Radio Frequency) 모듈에 제어 명령을 제공하지 않는다.
  - 관련 hazard: 시스템의 제어권 상실
- UCA-30: 신호등 서브시스템의 SW 컨트롤러가 중앙통제 서브시스템으로부터 수동 신호 제어 명령을 제공받은 경우에도 수동 신호등 제어 명령을 제공하지 않는다.
  - 관련 hazard: RF 통신 오류, 신호체계 오작동

Case Study

UCA별 loss scenario 예시

UCA-13: 관리자가 교통정보시스템이 MQTT 서버를 통해 긴급 정보를 제공하지 않는 경우에 중앙통제 서브시스템에 수동 제어 명령을 제공하지 않는다.

LS13-4: 교통정보시스템이 올바르게 동작하지 못하는 경우에도, 중앙통제 서브시스템의 SW 컨트롤러가 관리자로부터 올바른 수동 제어 명령을 입력 받았으나 이를 처리하여 전달하는 동작을 수행하지 못하였다.

Case Study

UCA별 loss scenario 예시

UCA-19: 중앙통제 서브시스템의 SW 컨트롤러가 교통정보시스템으로부터 MQTT 서버를 통해 긴급 정보가 제공된 경우에도 RF (Radio Frequency) 모듈에 제어 명령을 제공하지 않는다.

LS19-2: 도로 상에 긴급 상황이 발생했음에도 해당 상황에 대한 데이터를 제공받지 못했기 때문에 교통정보시스템이 긴급 정보를 제공하지 않아서 중앙통제 서브시스템이 잘못된 process model 값을 가지게 되었다. 긴급 상황에 대한 데이터를 교통정보시스템이 제공받지 못한 것은 센서 기능의 저하로 인해 정상적인 데이터를 수집하지 못했기 때문이다.

- 교통정보시스템으로부터 잘못된 정보를 제공받아 중앙통제 서브시스템이 잘못된 process model을 가지게 되었기 때문에 UCA-19가 발생했다고 분석
  → 교통정보시스템에 발생한 문제를 중점적으로 다룸

Case Study

UCA별 loss scenario 예시

UCA-30: 신호등 서브시스템의 SW 컨트롤러가 중앙통제 서브시스템으로부터 수동 신호 제어 명령을 제공받은 경우에도 수동 신호등 제어 명령을 제공하지 않는다.

LS30-1: 신호등 서브시스템의 SW 컨트롤러가 잘못된 control algorithm을 가지고 있어서 중앙통제 서브시스템으로부터 수동 신호 제어 명령을 제공받았으나 수동 신호등 제어 명령을 제공하는 데에 실패하였다.

Case Study

UCA와 loss scenario를 활용하여 Safety TC를 생성한 예시

UCA-13: 관리자가 교통정보시스템이 MQTT 서버를 통해 긴급 정보를 제공하지 않는 경우에 중앙통제 서브시스템에 수동 제어 명령을 제공하지 않는다.

LS13-4: 교통정보시스템이 올바르게 동작하지 못하는 경우에도, 중앙통제 서브시스템의 SW 컨트롤러가 관리자로부터 올바른 수동 제어 명령을 입력 받았으나 이를 처리하여 전달하는 동작을 수행하지 못하였다.

TC13-4: 관리자는 교통정보시스템이 올바르게 동작하지 못하는 경우 중앙통제 서브시스템을 수동 제어한다. 그러면 중앙통제 서브시스템의 SW 컨트롤러는 교통정보시스템이 올바르게 동작할 수 있도록 복구될 때까지 (until system function gets back to normal) 관리자가 제공한 수동 제어 명령을 받아들이고 그에 맞게 동작해야 한다.

Case Study

UCA와 loss scenario를 활용하여 Safety TC를 생성한 예시

UCA-19: 중앙통제 서브시스템의 SW 컨트롤러가 교통정보시스템으로부터 MQTT 서버를 통해 긴급 정보가 제공된 경우에도 RF (Radio Frequency) 모듈에 제어 명령을 제공하지 않는다.

LS19-2: 도로 상에 긴급 상황이 발생했음에도 해당 상황에 대한 데이터를 제공받지 못했기 때문에 교통정보시스템이 긴급 정보를 제공하지 않아서 중앙통제 서브시스템이 잘못된 process model 값을 가지게 되었다. 긴급 상황에 대한 데이터를 교통정보시스템이 제공받지 못한 것은 센서 기능의 저하로 인해 정상적인 데이터를 수집하지 못했기 때문이다.

TC19-2: 센서 기능의 저하로 인해 시스템 전체의 기능이 저하된 상황에서 센서는 긴급 정보와 일치하지 않는 잘못된 센서 데이터를 제공한다. 그러면 교통정보시스템의 SW 컨트롤러는 센서의 기능이 정상적으로 돌아올 수 있을 때까지 (until sensor function gets back to normal) 시스템의 모든 데이터에 대한 로그를 남기고 관리자가 이를 확인할 수 있도록 모니터링 서버로 전송해야 한다.

Case Study

UCA와 loss scenario를 활용하여 Safety TC를 생성한 예시

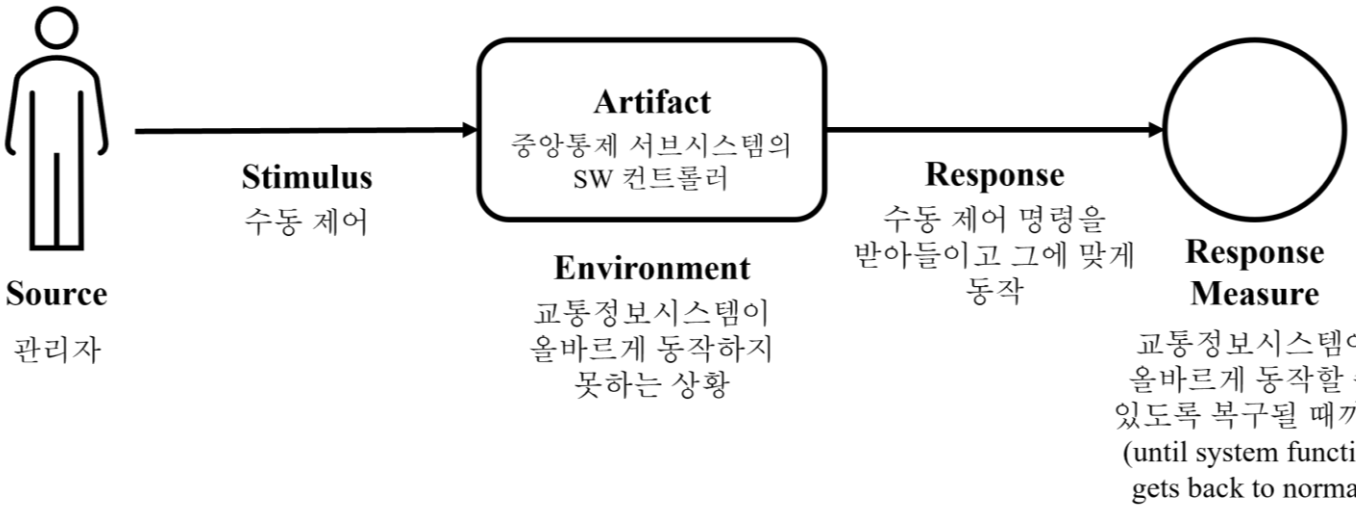
UCA-30: 신호등 서브시스템의 SW 컨트롤러가 중앙통제 서브시스템으로부터 수동 신호 제어 명령을 제공받은 경우에도 수동 신호등 제어 명령을 제공하지 않는다.

LS30-1: 신호등 서브시스템의 SW 컨트롤러가 잘못된 control algorithm을 가지고 있어서 중앙통제 서브시스템으로부터 수동 신호 제어 명령을 제공받았으나 수동 신호등 제어 명령을 제공하는 데에 실패하였다.

TC30-1: 정상 동작 시에, 신호등 서브시스템의 RF 모듈은 수동 신호 제어 명령을 SW 컨트롤러에 전달한다. 그러면 신호등 서브시스템의 SW 컨트롤러는 실패하지 않고 (without failure) 수동 신호등 제어 명령을 제공해야 한다.

Case Study

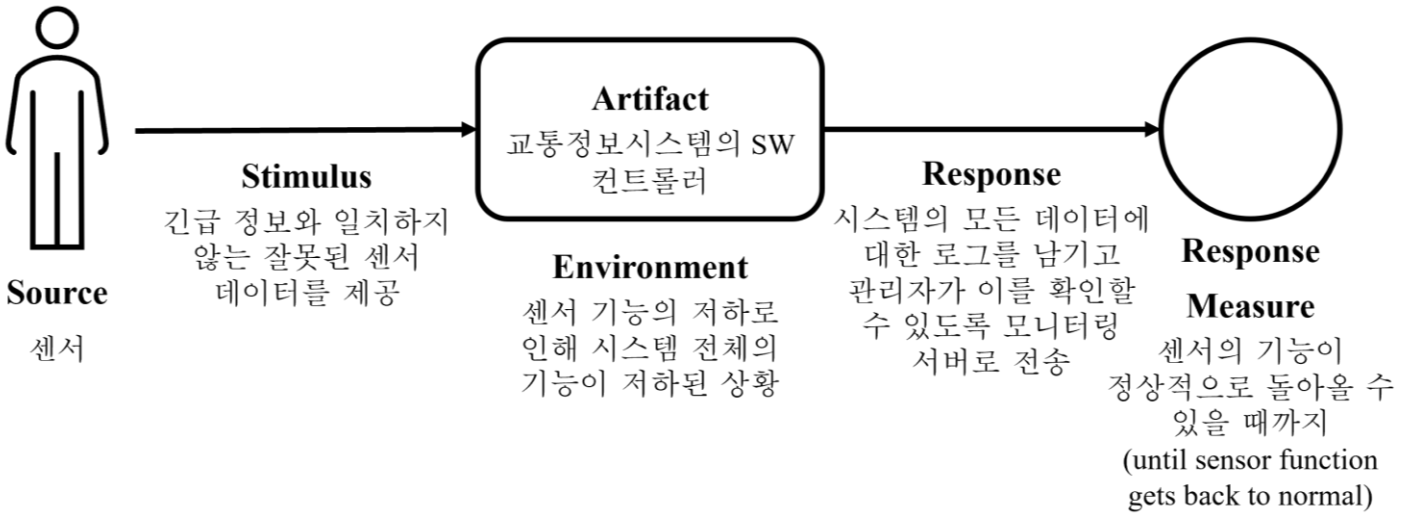
QAS의 형태로 나타낸 safety TC의 예시



TC13-4: 관리자는 교통정보시스템이 올바르게 동작하지 못하는 경우 중앙통제 서브시스템을 수동 제어한다. 그러면 중앙통제 서브시스템의 SW 컨트롤러는 교통정보시스템이 올바르게 동작할 수 있도록 복구될 때까지 (until system function gets back to normal) 관리자가 제공한 수동 제어 명령을 받아들이고 그에 맞게 동작해야 한다.

Case Study

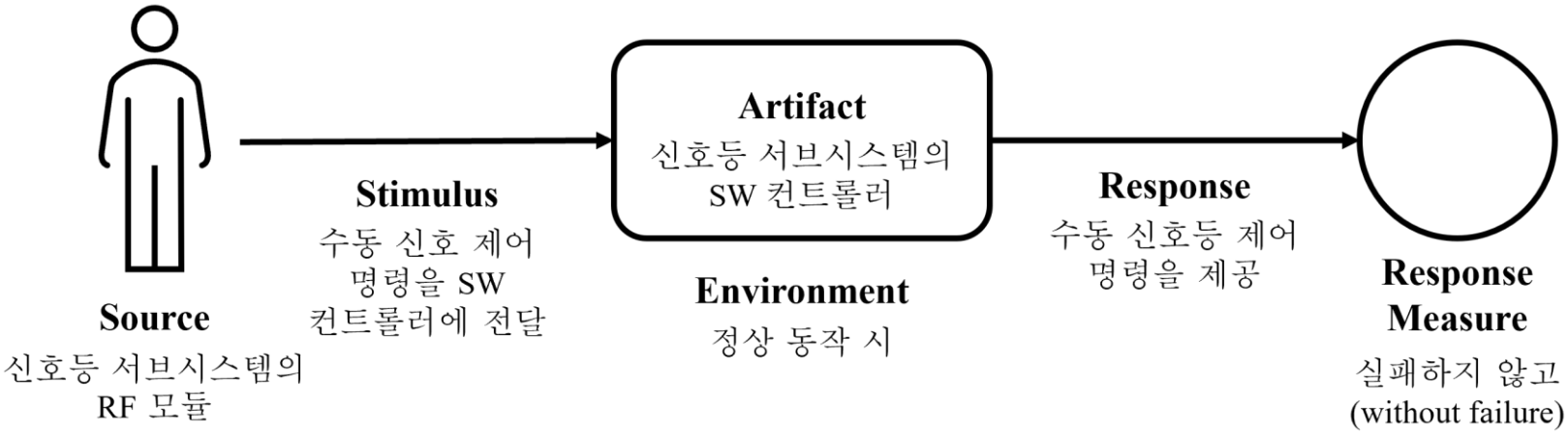
QAS의 형태로 나타낸 safety TC의 예시



TC19-2: 센서 기능의 저하로 인해 시스템 전체의 기능이 저하된 상황에서 센서는 긴급 정보와 일치하지 않는 잘못된 센서 데이터를 제공한다. 그러면 교통정보시스템의 SW 컨트롤러는 센서의 기능이 정상적으로 돌아올 수 있을 때까지 (until sensor function gets back to normal) 시스템의 모든 데이터에 대한 로그를 남기고 관리자가 이를 확인할 수 있도록 모니터링 서버로 전송해야 한다.

Case Study

QAS의 형태로 나타낸 safety TC의 예시



TC30-1: 정상 동작 시에, 신호등 서브시스템의 RF 모듈은 수동 신호 제어 명령을 SW 컨트롤러에 전달한다. 그러면 신호등 서브시스템의 SW 컨트롤러는 실패하지 않고 (without failure) 수동 신호등 제어 명령을 제공해야 한다.

Case Study

- STPA 결과
  - UCA: 43
  - Loss scenario: 110
- Safety TC 생성 결과
  - Safety TC: 110
- Safety Testing 결과
  - Pass rate: 100%

Case Study

- 단, STPA의 수행 정도에 따라 식별되는 UCA, loss scenario, 그리고 그에 따른 safety TC의 개수와 통과율이 달라질 수 있음을 인지해야 함
- 만일 safety TC를 통과하지 못했다면,
  어떤 원인으로 인해 통과하지 못했는지를 분석
  → 해당 원인이 이후 시스템의 안전성을 위협할 수 있다고 판단될 경우,
  시스템의 요구사항을 수정하고 이에 맞게 시스템을 개선하는 과정 필요

Case Study

- Pros

1) 체계적인 HA 기법으로 알려진 STPA의 결과를 활용하고 QAS의 형태를 빌려 safety TC를 생성하기 때문에 보다 체계적인 접근이 가능하다

2) STPA를 활용하기 때문에 loss scenario에 포함된 causal factor를 추후 시스템에 행해져야 할 점검에 대한 항목으로 고려할 수 있다

3) 여러 표준 [2, 3]에서 강조하는 기능 안전성 (functional safety)을 포함하여 제어 관점에서의 안전성 (control safety), 물리적 동작의 안전성 (physical safety) 등 안전성의 다양한 측면 [4]을 고려할 수 있다

Case Study

- Cons

1) 제안한 프로세스를 통해서 생성하는 safety TC가 다소 전형적일 수 있다

2) 사례 연구를 진행하는 테스트베드 시스템의 상호작용의 형태와 그 동작이 단순하다

3) 테스트베드 시스템이 지니는 특성 중 동일한 시스템의 여러 인스턴스가 존재한다는 특성에 대하여 다루지 못했다

Case Study

- Cons

1) 제안한 프로세스를 통해서 생성하는 safety TC가 다소 전형적일 수 있다
   - 각 부분에 대한 테스팅을 진행하고 통과 여부를 확인하는 것은 필요한 과정
   - 테스팅을 통해서 모든 경우를 완벽하게 다루는 것은 거의 불가능에 가깝다 (근본적 한계점)
     → 테스팅 이외의 검증의 수단을 추가로 마련하여 이를 보완할 수 있도록 하는 것이 필요

2) 사례 연구를 진행하는 테스트베드 시스템의 상호작용의 형태와 그 동작이 단순하다

3) 테스트베드 시스템이 지니는 특성 중 동일한 시스템의 여러 인스턴스가 존재한다는 특성에 대하여 다루지 못했다

Conclusion

- STPA 수행 결과를 활용하여 QAS의 형태로 시스템의 안전성을 테스트하기 위한 safety TC를 식별하는 방법을 제안
- 사례 연구 수행
  - 기개발한 테스트베드 시스템을 대상으로 함
  - 제안한 approach의 적용을 통하여 safety TC를 식별
  - safety testing을 수행하여 결과 및 적용 효과 분석

Future Work

- 기존의 테스트베드 시스템에 새로운 시스템 추가

  → 테스트베드 시스템의 상호작용에 대한 복잡도 증가, 기능적 단순함 극복

- 한 시스템의 여러 인스턴스가 존재하고 상호작용하는 경우에 대하여 다룰 수 있도록 제안한 프로세스 개선

  → 추가적인 guideword 제시 가능성

- STPA 이외에 다른 HA 기법의 추가적인 적용 고려 중

감사합니다

Q&A: hyoona1202@naver.com

References

[1] L. Bass, P. Clements, R. Kazman, “Software Architecture in Practice,” Fourth Edition, Addison-Wesley Professional, 2021.
[2] International Electrotechnical Commission (IEC), IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems, 2010.
[3] International Organization of Standardization (ISO), ISO 26262: Road Vehicles – Functional Safety, 2011.
[4] X. Lyu, Y. Ding, S-H. Yang, “Safety and Security Risk Assessment in Cyber-Physical Systems,” IET Cyber-Physical Systems: Theory & Applications, vol. 4, no. 3, pp. 221-232, 2019.