Demo

3조 KUTOKIT

201510436 허윤아
201611261 민지호
201611293 전다윤
201614158 장다혜
201710515 최연지

- 목차 -

01 프로젝트 제안 배경

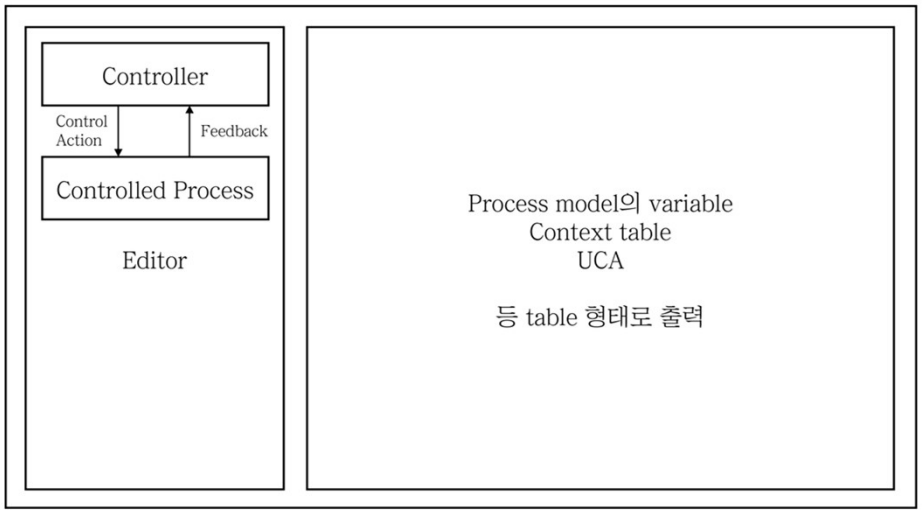
1. 과거와 달리 시스템이 점점 복잡해지고 규모가 커지면서, 더 이상 모든 문제의 원인이 component 에 있다고 단정짓기가 어려워졌고, 잠재적으로 어떤 문제가 발생 가능한지를 미리 예측에 어려움이 생김

2. 위와 같이 생기는 문제들을 해결하기 위해 기존의 것과 다른 관점을 가진 새로운 위험 분석 기법인 **STPA(System Theoretic Process Analysis)라는 위험 분석 기법의 제시**

3. 이때, STPA 수행에 있어서 인력이나 지원 도구들에 제한이 있는데, 이를 좀 더 효율적으로 수행하고자 **소프트웨어 컨트롤러 시스템의 formal SW specification 을 활용하여 STPA 의 수행을 지원하는 방법을 제안하고 이를 일부 자동화한 도구**를 개발하고자 함

02

프로젝트 소개 (KUtoKit : KU semi-auto Kit)

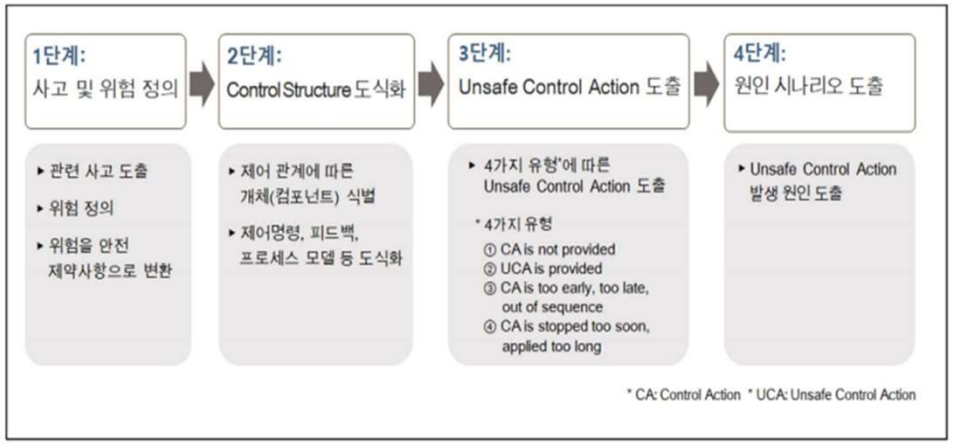


1. Controller System 을 STAMP 기반 프로그램 분석 기법인 STPA 에 맞춰서 효율적인 분석에 도움을 주는 프로그램

2. STPA 의 과정의 일부 자동화가 가능하도록 함

3. STPA 의 4 단계 과정 중 2 단계와 3 단계를 조금 더 쉽게 수행할 수 있도록 지원

03 STPA 개념



* 단, 이 때 1 단계는 이미 수행한 상태에서 이 프로그램을 사용한다는 가정이 필요
* 기본적으로 프로그램은 크게 두 가지 파트로 나눌 수 있는데,
기본적인 STPA 의 과정을 수행하기 위한 파트와,
NuSCR 이라는 Specification Language 를 이용하여 Supporting 하는 파트로 나뉘어짐

1 단계
STPA 에서는 우선적으로 시스템에 발생할 수 있는 accident(loss)를 분석 및
loss 발생시킬 가능성이 있는 hazard 와, 각 hazard 에 대한 System-level Safety Constraint 작성

2단계
Control Structure 는 시스템을 추상화하여 모델링
Control Structrue: Controller 와 Controlled Process, Control Action, Feedback로 구성

3단계
앞서 작성된 Control Structure 에서 도출된 Control Action 들을 기반으로
UCA(Unsafe Control Action)을 작성

4 단계
3 단계에서 도출해낸 UCA 를 기반으로 UCA 발생하는 원인을 분석하여
Causal Scenario 를 작

04

Project Functions

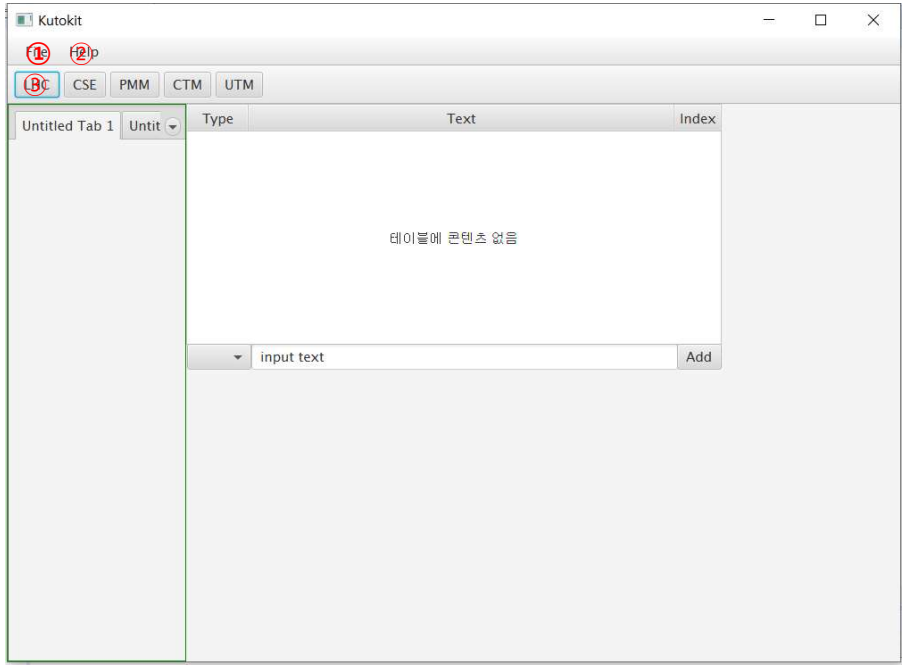
1. Get User Input	사용자의 입력(버튼 클릭, 드래그, 텍스트 입력)을 받는다.
2. Create Control Structure	Control Action 과 Feedback 을 입력 받고, 사용자의 입력을 통해 Control Structure 을 생성한다.
3. Get NuSCR File	NuSCR 로 작성된 파일을 가져온다.
4. Parse NuSCR File	NuSCR 로 작성된 파일을 분석하여, 분석하려는 CA 와 연관된 output variable 을 추출한다.
5. Get Essential Variable	output variable 추출에 필수적인 input variable, internal variable 들을 추출한다.
6. Create Process Variable	추출된 input/internal/output variable 들을 이용하여 Controller 별 Process variable 을 생성한다.
7. Create Process Model	추출된 input/internal/output variable 들을 이용하여 Controller 별 Process Model 을 생성한다.
8. Select MCS	NuFTA 를 통해 추출된 MCS 중 유의미한 것을 선별한다.
9. Create Context Table	선별된 MCS 를 이용해 Context table 을 생성한다.
10. Create UCA	생성된 Context table 에서 hazardous 하다고 선택된 context 들을 기반으로 Controller 별 CA 에 따라 UCA 후보군을 생성한다.
11. Create UCA Table	loss scenario 를 분석할 수 있도록 UCA table 을 생성한다.

05

현재까지 구현한 기능

1. Javafx를 이용한 어플리케이션 생성 및 레이아웃 제작
2. 어플리케이션 내에서 파일 불러오기
3. Process model에 필요한 variable을 가져오기 (PMM)
4. MCS 파일로부터 추출한 데이터로 Context Table 만들기
5. Loss , Hazard , Constraint 입력 (LHC)

1. Javafx를 이용한 어플리케이션 생성 및 레이아웃 제작



① File

- File의 생성, Open, Save 기능을 제공
- 전체 프로젝트를 저장하거나 각각 Control structure , Context Table , UCA를 저장하고 불러오고자 할 때 사용

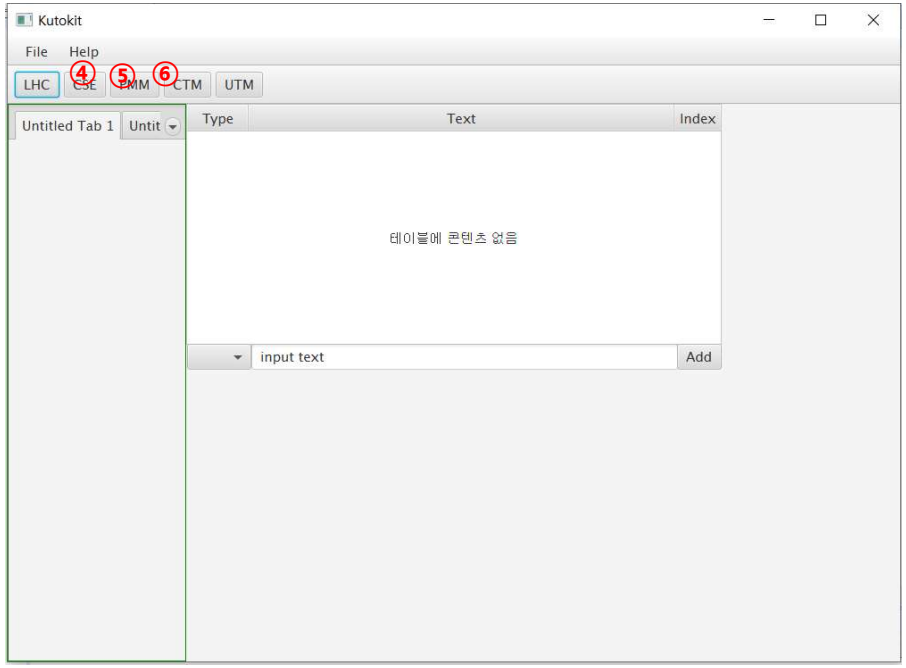
② Help

- 프로그램의 사용법 제공
  각 기능에 대한 설명과 전체 프로그램에 대한 설명 제공

③ LHC

- STPA 의 1단계에 해당하는 기능
- 사용자가 직접 시스템의 loss, Hazard, Constraint 를 분석해 입력 가능

1. Javafx를 이용한 어플리케이션 생성 및 레이아웃 제작



④ CSE

- Control Structure Editor
- Control Structure를 작성할 수 있는 editing 기능 제공

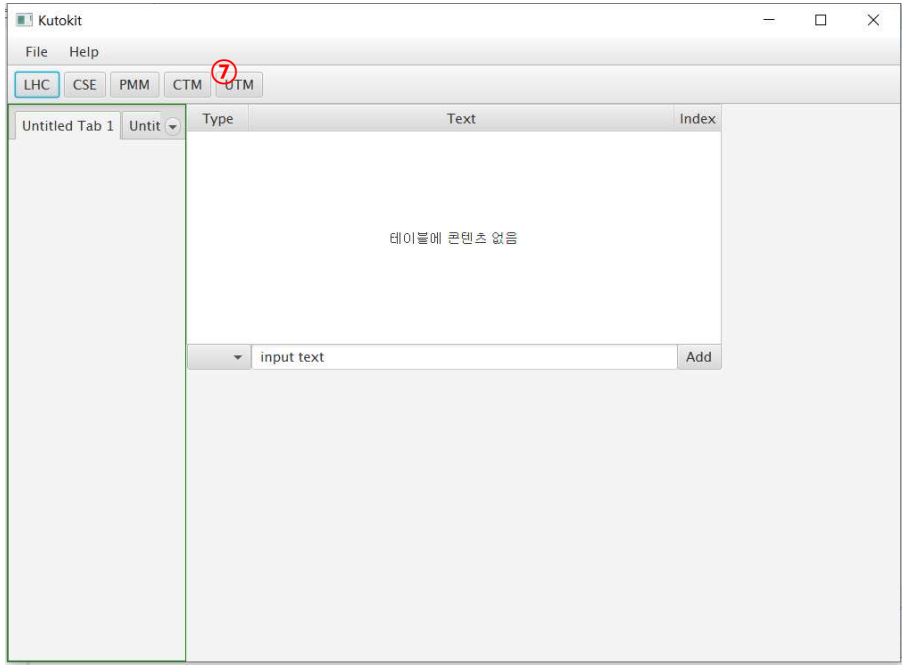
⑤ PMM

- Process Model Maker
- Controller마다 NuSCR로부터의 Process Model 추출 지원
- 사용자가 직접 입력, 수정, 삭제 가능

⑥ CTM

- Context Table Maker
- NuFTA를 통해 얻은 MCS로부터 Context Table 생성
- Hazardous 여부 선택 가능

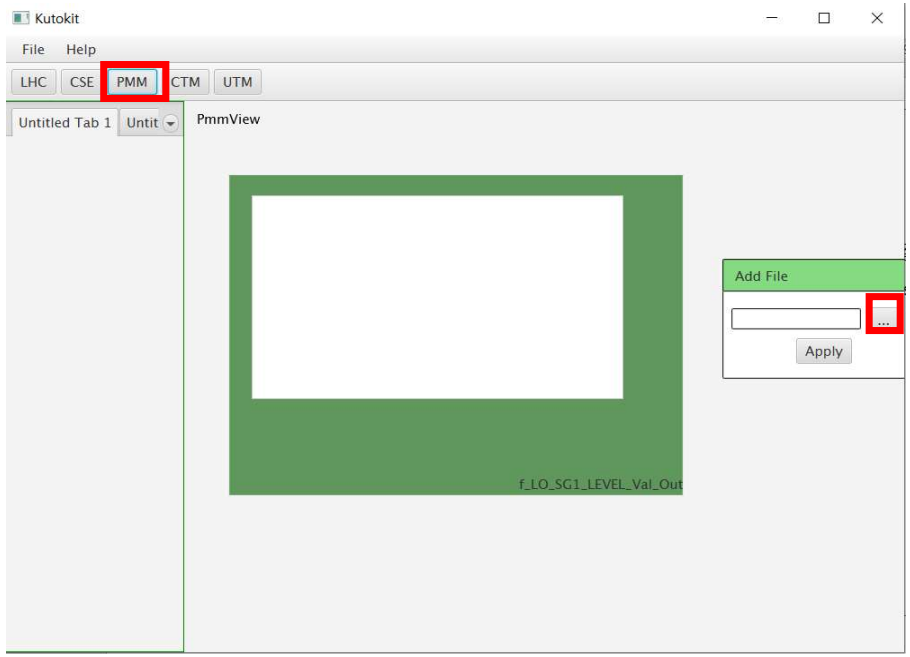
1. Javafx를 이용한 어플리케이션 생성 및 레이아웃 제작



⑦ UTM

- UCA Table Maker
- 사용자가 Context Table에서 Hazardous하다고 판단한 Context들을 기반으로 UCA Table 자동 생성

2. 어플리케이션 내에서 파일 불러오기 (PMM)

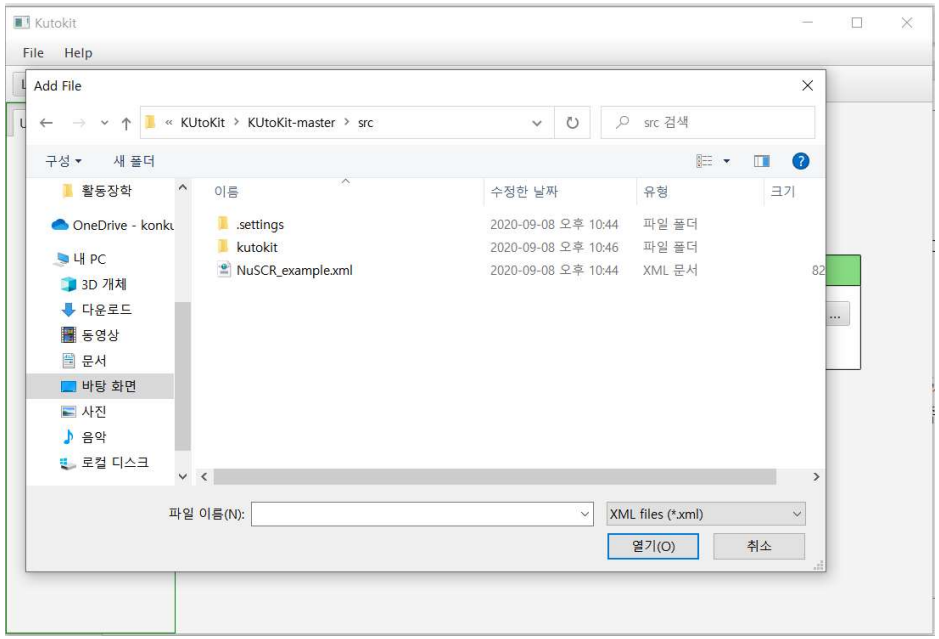


※ PMM 화면은 CSE와 연동되어야 함

CSE에서 작성한 Control Structure에서 Process Model을 추가하고자 하는 Controller를 선택

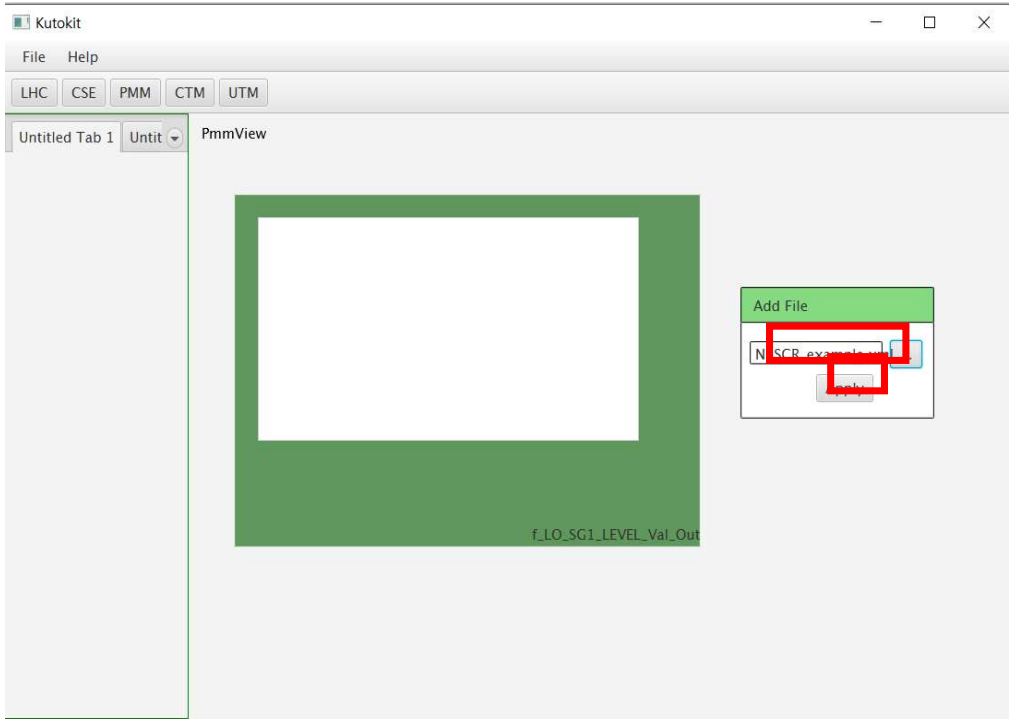
☞ 해당 Controller의 CA(Control Action)가 output으로 존재하는 NuSRS 파일 탐색 가능

2. 어플리케이션 내에서 파일 불러오기 (PMM)



AddFile 팝업에서 … 버튼을 눌러 분석에 필요한 NuSRS 파일(*.xml) 선택

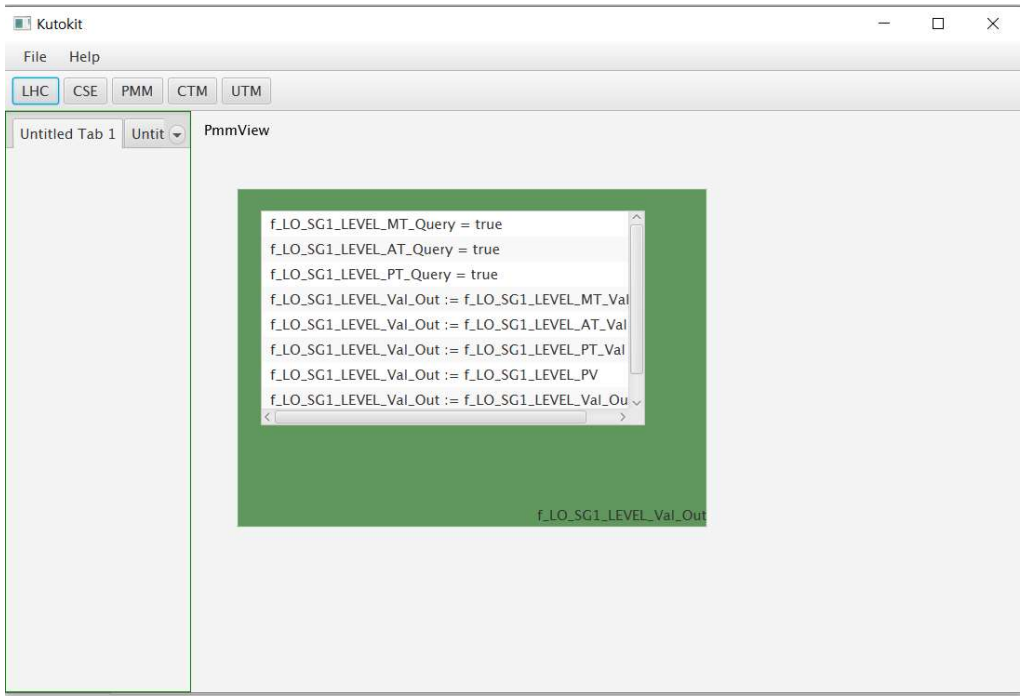
2. 어플리케이션 내에서 파일 불러오기 (PMM)



Apply 버튼을 눌러 적용

☞ 선택한 Controller에 Process Model이 자동으로 추출되어 적용

3. Process model에 필요한 variable을 가져오기 (PMM)



선택한 Controller의 CA에 알맞은 Process Model에

해당하는 variable과 input을 추출

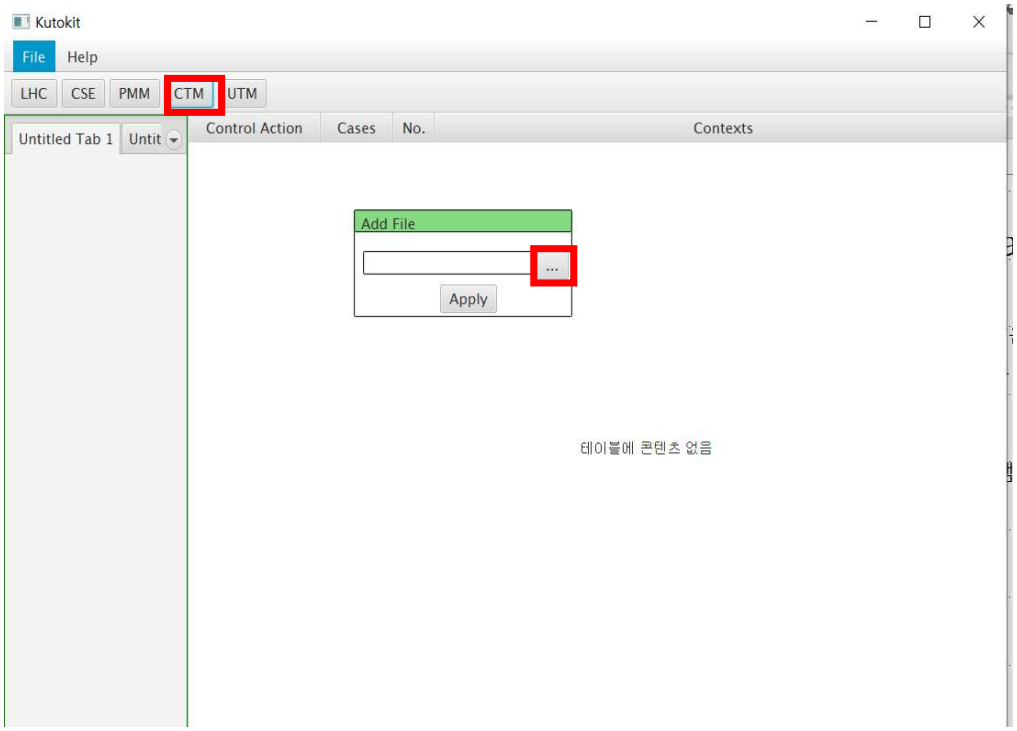
3. Process model에 필요한 variable을 가져오기 (PMM)

```
@FXML
public void ApplyFile() {
    if(selectedFile != null) {
        AddFile.getChildren().clear();
        // this.showList();
        reader = new xmlTest(selectedFile.getName());
        this.makeModel(reader.getNodeList(reader.getNode("f_LO_SG1_LEVEL_Val_Out"), "/condition"));
        this.makeModel(reader.getNodeList(reader.getNode("f_LO_SG1_LEVEL_Val_Out"), "/action"));
    }
}
```

```
// Make process model
public void makeModel(NodeList list) {
    for(int i = 0 ; i< list.getLength(); i++) {
        this.valuelist.add(list.item(i).getAttributes().getNamedItem("value").getTextContent());
    }
    PM.setItems(valuelist);
}
```

xml parsing code를 이용해 모든 variable, input을 parsing해오는 것이 아닌, CA에 해당하는 output과 연관성이 있는 내용들만 선택적으로 추출

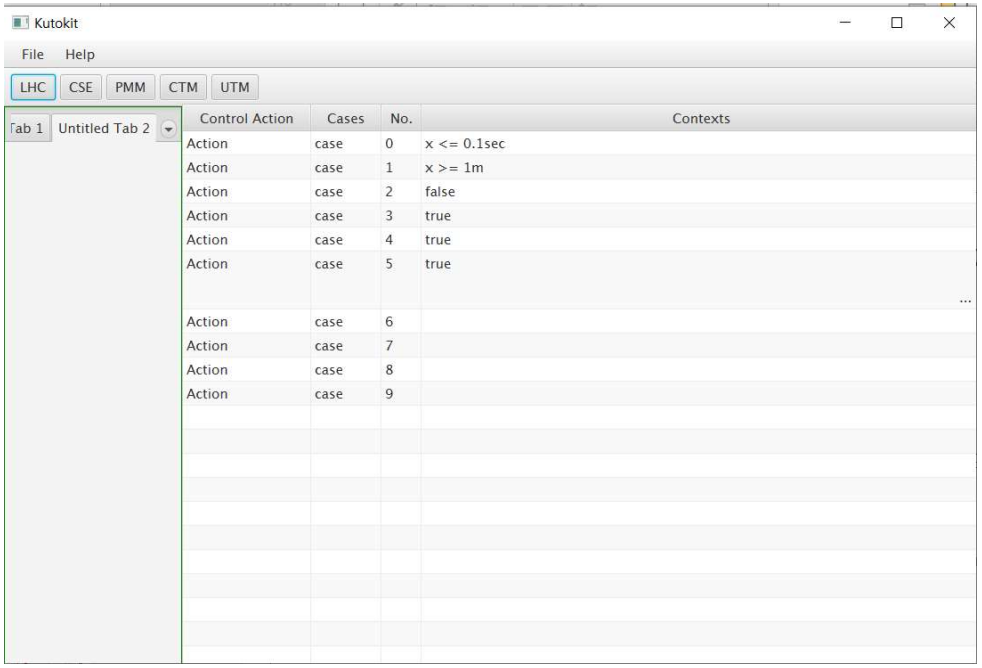
4. MCS 파일로부터 추출한 데이터로 Context Table 만들기



Add File 팝업에서 … 버튼을 선택, Context Table로 만들고자 하는 MCS 파일 선택 가능

※ MCS 파일은 NuFTA를 통해 Backward-Analysis된, 분석하고자 하는 시스템의 NuSRS

4. MCS 파일로부터 추출한 데이터로 Context Table 만들기



필요한 MCS 파일을 Apply

Process Model 작성 시와 마찬가지로 CA에 따른 variable과 input이 적용된 Context Table 자동 생성

5. Loss, Hazard, Safety Constraint 입력

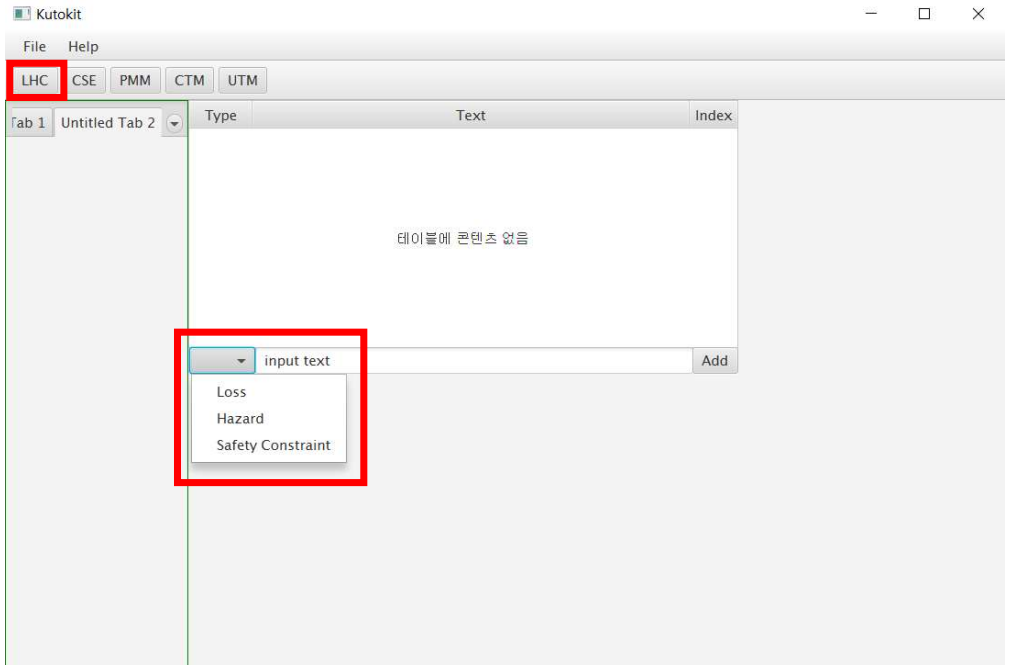


Table 좌측 하단의 Choice Box를 선택

☞ Loss, Hazard, Safety Constraint 를 drop down 방식으로 선택 가능

※ Tab의 형식으로 관리하는 것이 더 적합할 것 같으나, 아직 구현에 적절한 방법을 찾지 못함

06

향후 계획

1. Control Structure Editor - drag&drop, refactor(name, color….)
2. 전체적인 계층구조 시각화
3. 프로젝트를 한 파일로 저장 (.xml)
4. 구체적인 Context table 생성 로직
5. 구체적인 UCA table 생성 로직
6. Loss, Hazard, Safety Constraint와 UCA 각각의 indexing과 traceability 제공

07 프로젝트 진행 사항 (Github Project Kanban board)

https://github.com/minjyo/KUtoKit/projects/2