



SDS

5조 KUTOKIT

201510436 허윤아

201611261 민지호

201611293 전다윤

201614158 장다혜

201710515 최연지

- 목차 -

❖ High-level Design

1. System Architecture
2. Interface Design
3. System Sequence Diagram

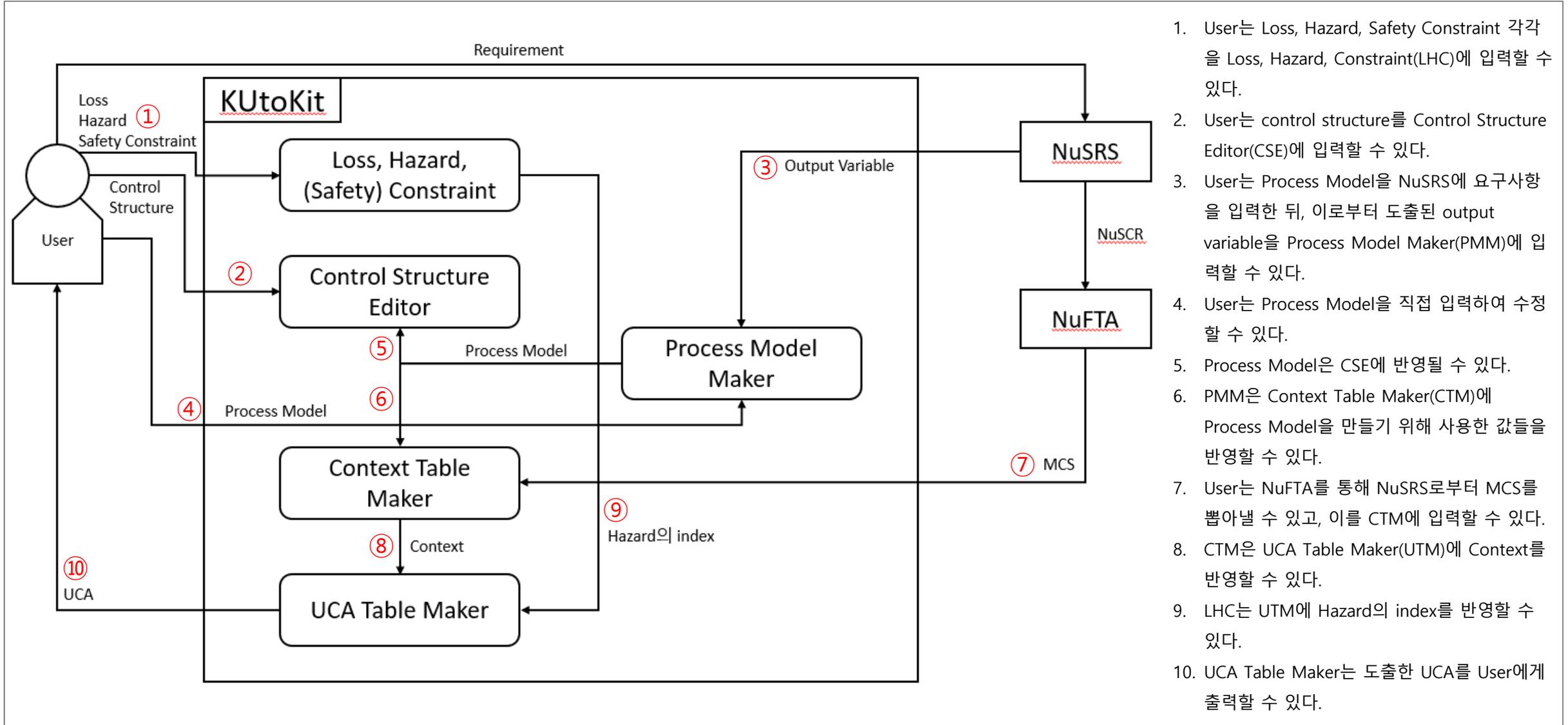
❖ Low-level Design

4. Class Diagram

❖ Traceability Matrix

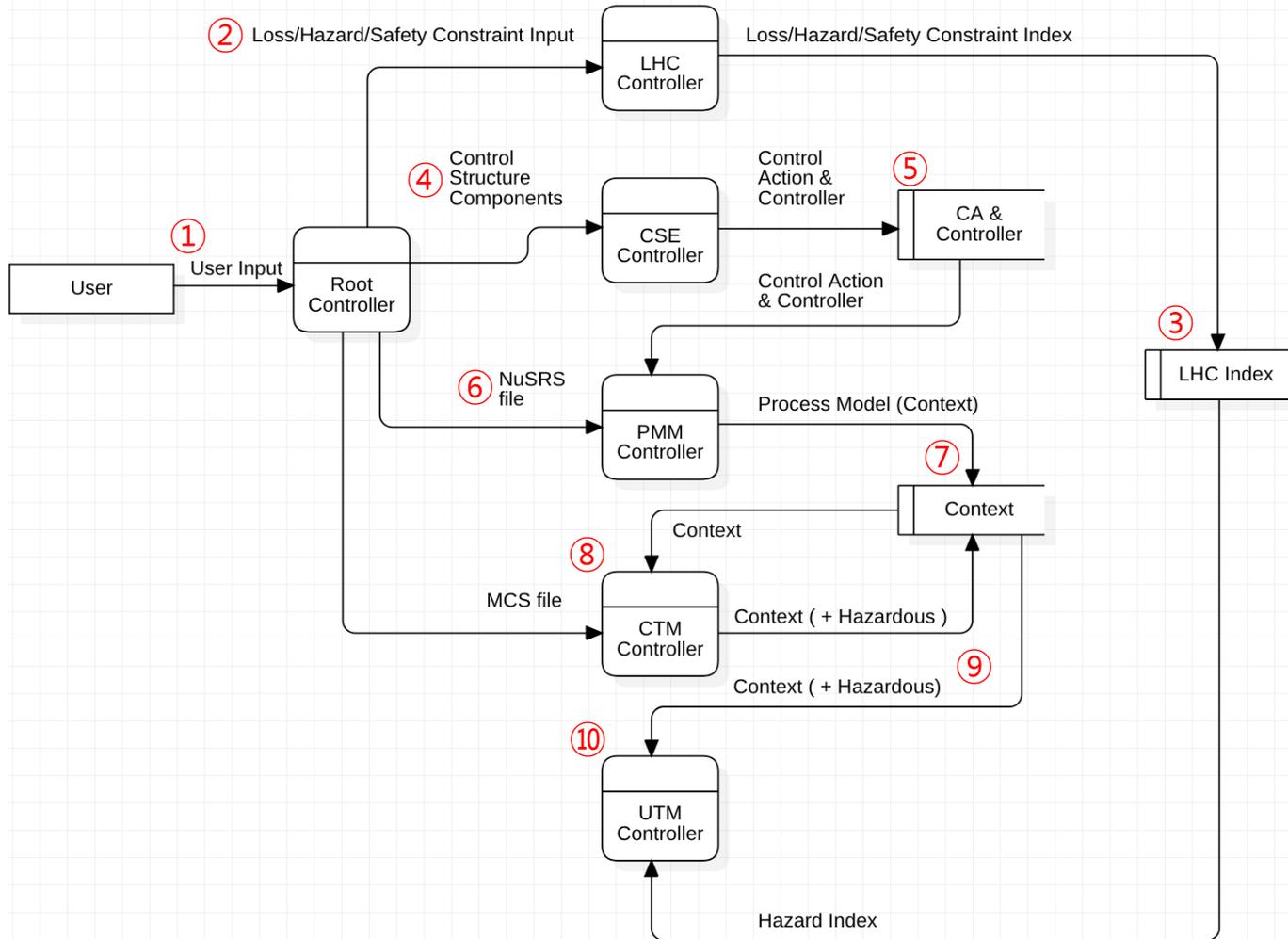
5. Traceability Matrix

System Architecture



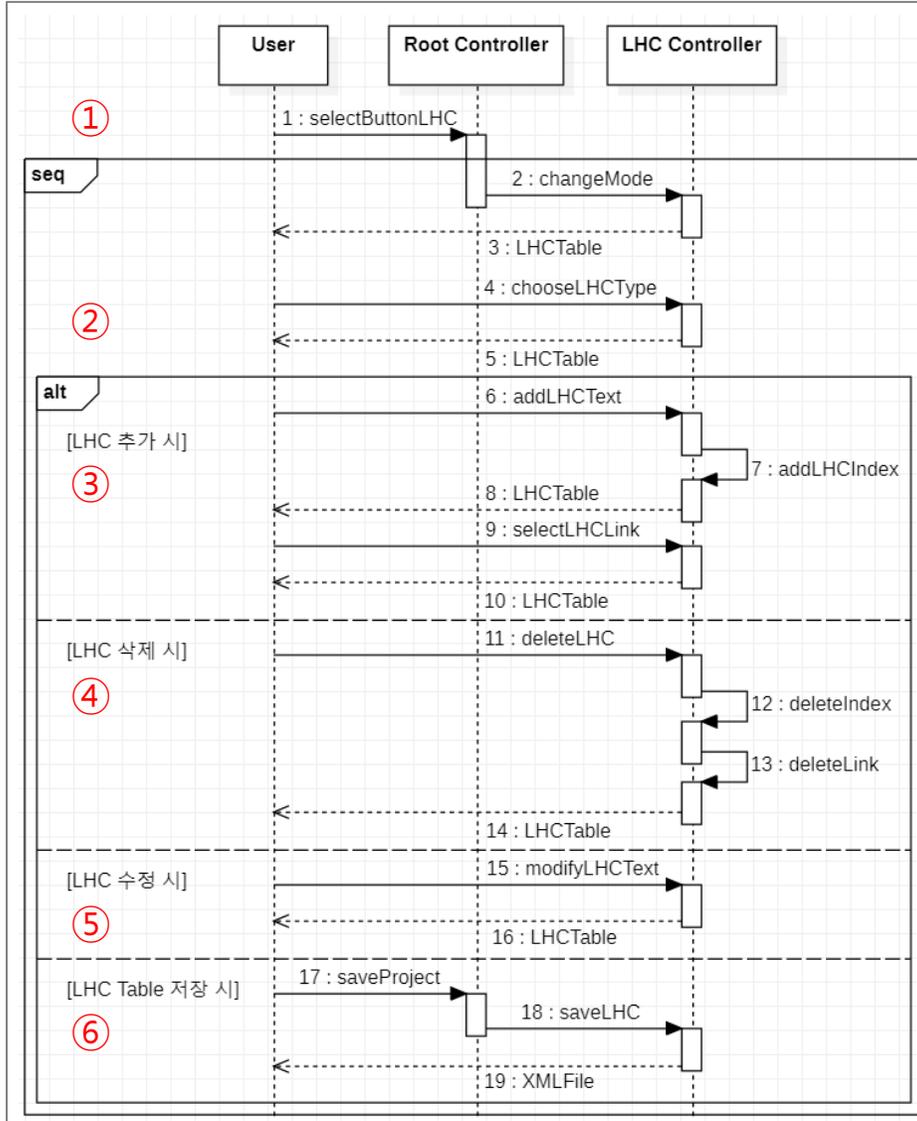
1. User는 Loss, Hazard, Safety Constraint 각각을 Loss, Hazard, Constraint(LHC)에 입력할 수 있다.
2. User는 control structure를 Control Structure Editor(CSE)에 입력할 수 있다.
3. User는 Process Model을 NuSRS에 요구사항을 입력한 뒤, 이로부터 도출된 output variable을 Process Model Maker(PMM)에 입력할 수 있다.
4. User는 Process Model을 직접 입력하여 수정할 수 있다.
5. Process Model은 CSE에 반영될 수 있다.
6. PMM은 Context Table Maker(CTM)에 Process Model을 만들기 위해 사용한 값들을 반영할 수 있다.
7. User는 NuFTA를 통해 NuSRS로부터 MCS를 뽑아낼 수 있고, 이를 CTM에 입력할 수 있다.
8. CTM은 UCA Table Maker(UTM)에 Context를 반영할 수 있다.
9. LHC는 UTM에 Hazard의 index를 반영할 수 있다.
10. UCA Table Maker는 도출한 UCA를 User에게 출력할 수 있다.

Interface



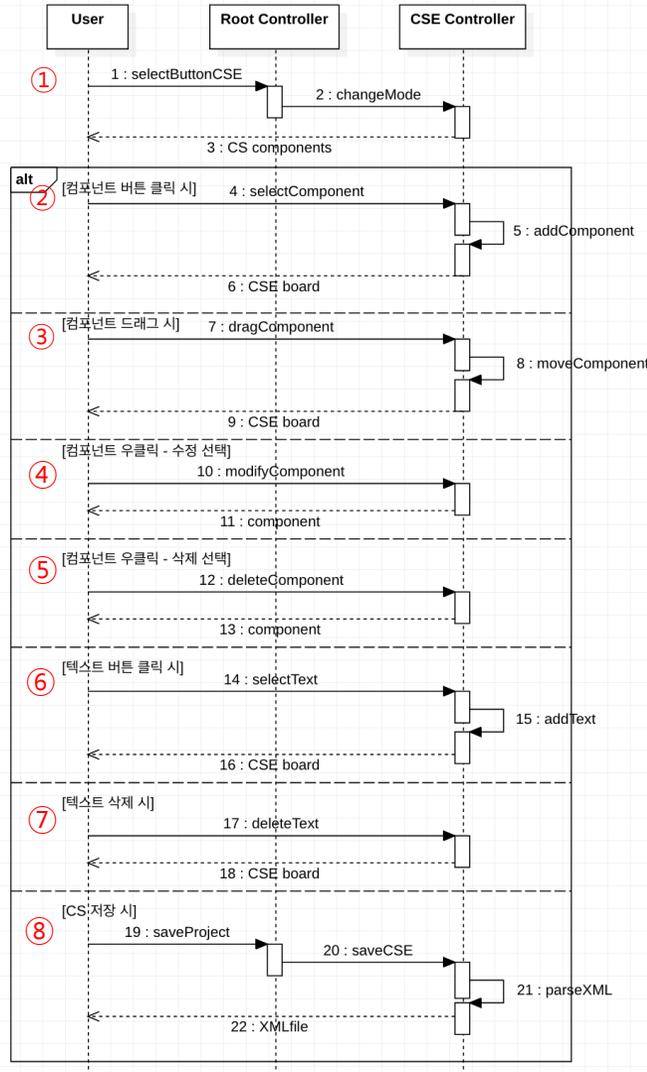
1. User는 Root Controller(프로그램의 프레임)를 통해 project의 생성, 저장 등 기본적인 input을 입력할 수 있다.
2. LHC Controller에는 Loss, Hazard, Safety Constraint에 관한 input들을 입력할 수 있다.
3. LHC Controller에서는 각각의 index 정보를 data store 형식으로 저장하고, link에 사용하거나 UTM Controller에서 Hazard의 index를 제공할 때 사용한다.
4. CSE Controller에는 Control Structure에 필요한 component들을 입력할 수 있다.
5. CSE Controller에서는 각 Controller의 이름과 그에 해당하는 Control Action에 대한 정보를 data 형식으로 저장하고, PMM Controller에 해당 정보를 전달한다.
6. PMM Controller에는 Process Model을 위한 parsing의 대상이 되는 NuSRS file을 입력할 수 있다.
7. PMM Controller에서는 Process Model을 context로 저장하여 이를 CTM Controller에게 전달한다.
8. CTM Controller에는 parsing의 대상이 되는 MCS file과, 그 기준이 되는 PMM Controller로부터의 context를 입력할 수 있다.
9. CTM Controller에서는 context와 hazardous 여부에 대한 값을 저장하여 이를 UTM Controller에 전달한다.
10. UTM Controller에는 CTM Controller로부터의 context와 hazardous 여부, 그리고 LHC Controller로부터의 Hazard index를 입력할 수 있다.

System Sequence Diagram(LHC)



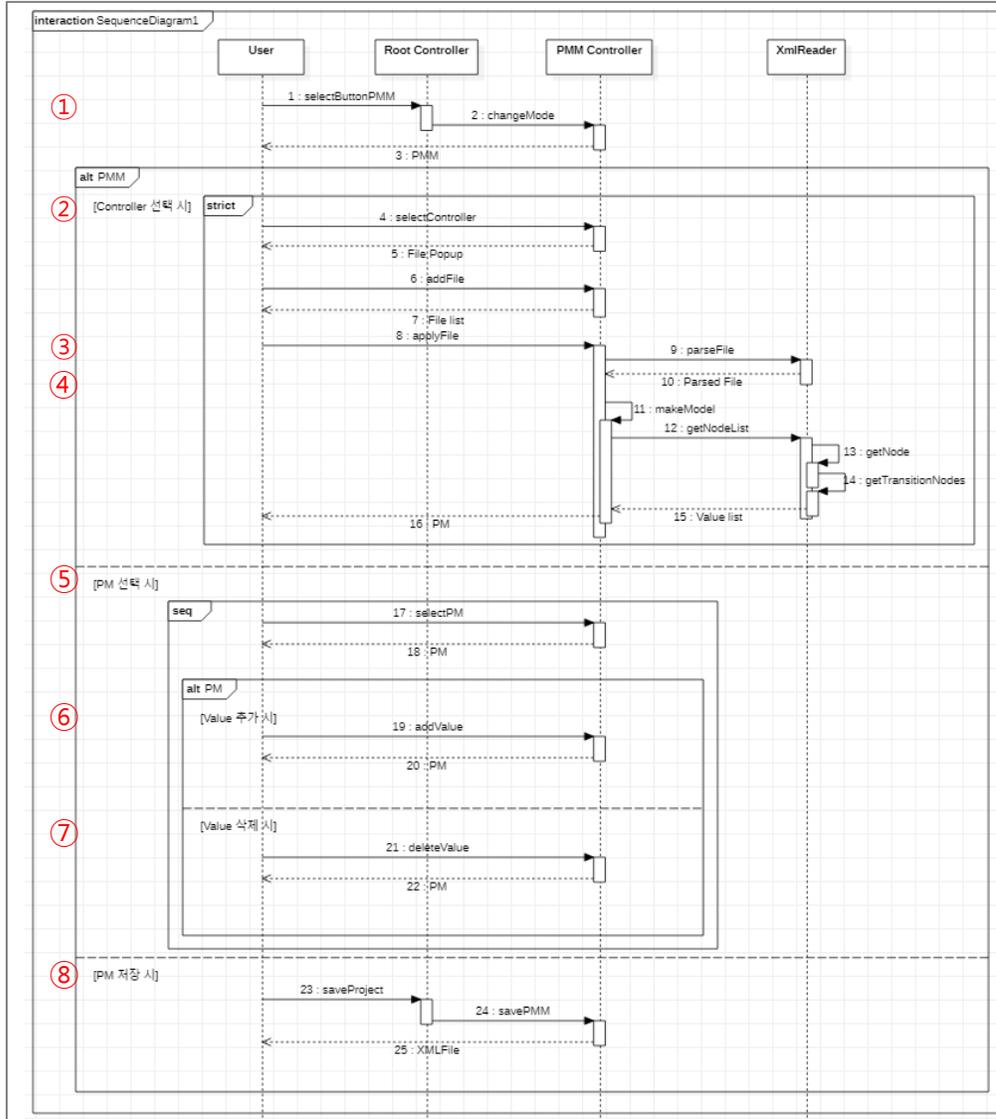
1. 모드 전환 : 메인 화면에서 LHC Table이 위치하는 곳을 클릭 시, LHC Controller로의 changeMode를 호출하여 비어 있는 LHC table을 보여주는 것으로 reply.
2. Type 선택 : LHC Table에서 상단의 tab 클릭 시 해당 tab으로 전환하여 화면을 보여주는 것으로 reply.
3. LHC 추가 : 선택된 tab의 type에 맞는 LHC의 내용을 사용자로부터 입력 받아, 자동으로 indexing을 해 주고 업데이트하여 table의 행을 채운 뒤 화면에 출력하는 것으로 reply.
Loss를 제외한 hazard와 safety constraint의 경우 link까지 설정해야 하고, 이후 설정된 link를 화면에 반영하여 출력하는 것으로 reply.
4. LHC 삭제 : 삭제하고자 하는 cell을 선택하여 우클릭한 뒤 삭제를 누르면 해당 cell의 text가 삭제되고, 이어서 자동으로 설정되었던 index와, 사용자가 설정해준 link가 삭제된다. 이를 화면에 반영하여 출력하는 것으로 reply.
5. LHC 수정 : 수정하고자 하는 cell을 선택하여 우클릭한 뒤 수정을 누르면 해당 cell의 text를 수정할 수 있고, 이를 화면에 반영하여 출력하는 것으로 reply.
6. LHC 저장 : 프로그램에서 '파일 - 저장' 또는 '파일 - 다른 이름으로 저장' 클릭 시, Root Controller에서 LHC Controller의 saveLHC를 호출하여 현재 LHC Table을 XML로 저장하는 것으로 reply.

System Sequence Diagram(CSE)



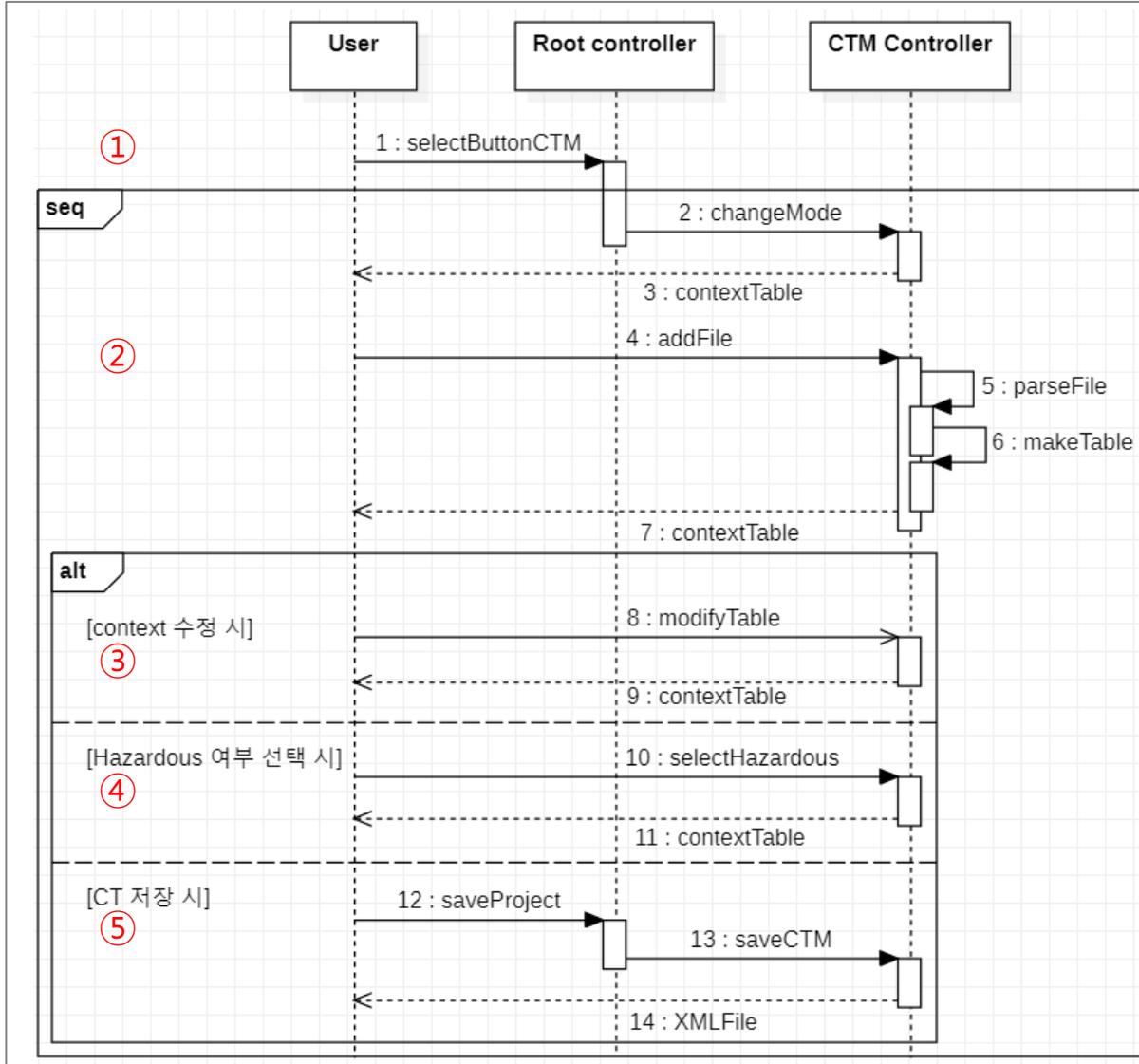
1. 모드 전환: 메뉴바의 CSE 버튼 클릭 시, Root Controller에서 CSE Controller로의 changeMode를 호출하여 CSE 모드로 전환하고, Control Structure의 컴포넌트들을 reply
2. 컴포넌트 추가: Side bar 의 컴포넌트 버튼 클릭 시, CSE Controller의 addComponent를 호출하여 해당 컴포넌트를 board 에 추가하고, 결과가 반영된 board 를 reply
3. 컴포넌트 이동: board 에 있는 컴포넌트를 드래그 시, CSE Controller의 moveComponent를 호출하여 해당 컴포넌트를 user 가 원하는 위치로 이동시키고, 결과가 반영된 board 를 reply
4. 컴포넌트 수정: board 에 있는 컴포넌트를 우클릭 하고 '수정'을 선택할 시, CSE Controller의 modifyComponent를 호출하여 해당 컴포넌트를 user가 입력한 정보로 수정하고, 결과가 반영된 컴포넌트를 reply
5. 컴포넌트 삭제: board 에 있는 컴포넌트를 우클릭 하고 '삭제'을 선택할 시, CSE Controller의 deleteComponent를 호출하여 해당 컴포넌트를 board에서 삭제하고, 결과가 반영된 board 를 reply
6. 텍스트 추가: Side bar 의 텍스트 버튼 클릭 시, CSE Controller의 addText를 호출하여 텍스트를 board 에 추가하고, 결과가 반영된 board 를 reply
7. 텍스트 삭제: board 에 있는 텍스트를 우클릭 하고 '삭제'을 선택할 시, CSE Controller의 deleteText를 호출하여 해당 텍스트를 board에서 삭제하고, 결과가 반영된 board를 reply
8. Control Structure 저장: 프로그램에서 '파일 - 저장' 또는 '파일 - 다른 이름으로 저장' 클릭 시, Root Controller에서 CSE Controller의 saveCSE를 호출하여 현재 Control Structure을 XML파일로 저장하고 해당 XML파일을 reply

System Sequence Diagram(PMM)



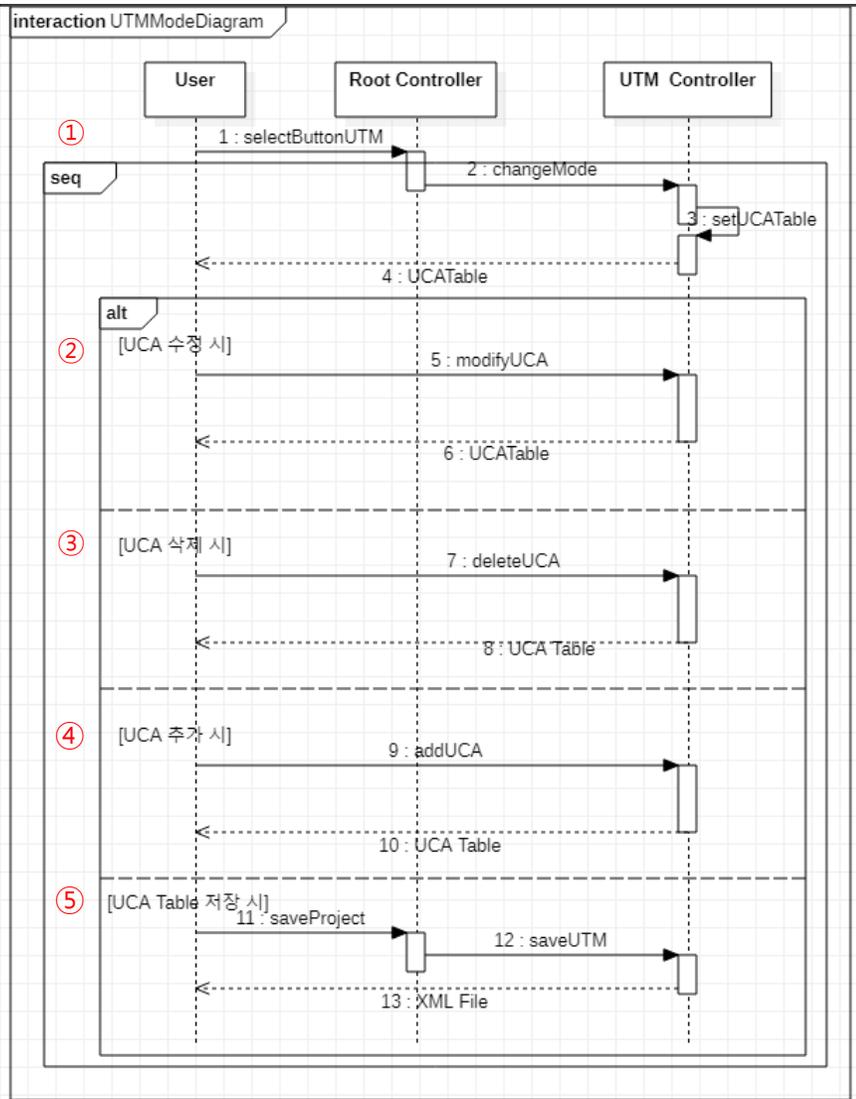
1. 모드 전환: 메뉴바의 PMM 버튼 클릭 시, Root Controller에서 PMM Controller로의 changeMode를 호출하여 PMM 모드로 전환하고, Controller와 PM을 reply
2. Controller 선택 : 원하는 Controller 선택 시, PMM Controller의 selectController를 호출하여 파일 선택 팝업창을 reply
3. Xml 파일 추가 : '.' 버튼 클릭 시, PMM Controller의 addFile을 호출하여 파일 list 를 reply
4. Xml 파일 추출 : 'Apply' 버튼 클릭 시, 선택된 Xml 파일이 XmlReader의 의해 parsing 되어 PMM Controller에 reply
 - 4-1. 연관 변수 추출 : XmlReader의 getNodeList 가 호출되고 , 선택된 Controller와 CA의 연관 변수들이 추출되어 PMM Controller에 reply
 - 4-2. PM 생성 : PMM Controller의 makeModel을 통해, 연관 변수들이 저장된 PM을 reply
5. PM 선택 : 원하는 PM 선택 시, PMM Controller의 selectPM을 호출하여 강조된 PM을 reply
6. 변수 추가 : 강조된 PM의 '+' 버튼 클릭 시, PMM Controller의 addValue를 호출하여 PM에 변수를 추가하고, 결과가 반영된 PM을 reply
7. 변수 삭제 : 강조된 PM의 '-' 버튼 클릭 시, PMM Controller의 deleteValue를 호출하여 PM으로부터 변수를 삭제하고, 결과가 반영된 PM을 reply
8. PM 저장 : 프로그램에서 '파일 - 저장' 또는 '파일 - 다른 이름으로 저장' 클릭 시, Root Controller에서 PMM Controller의 savePMM을 호출하여 현재 PM을 XML파일로 저장하고 해당 XML파일을 reply

System Sequence Diagram(CTM)



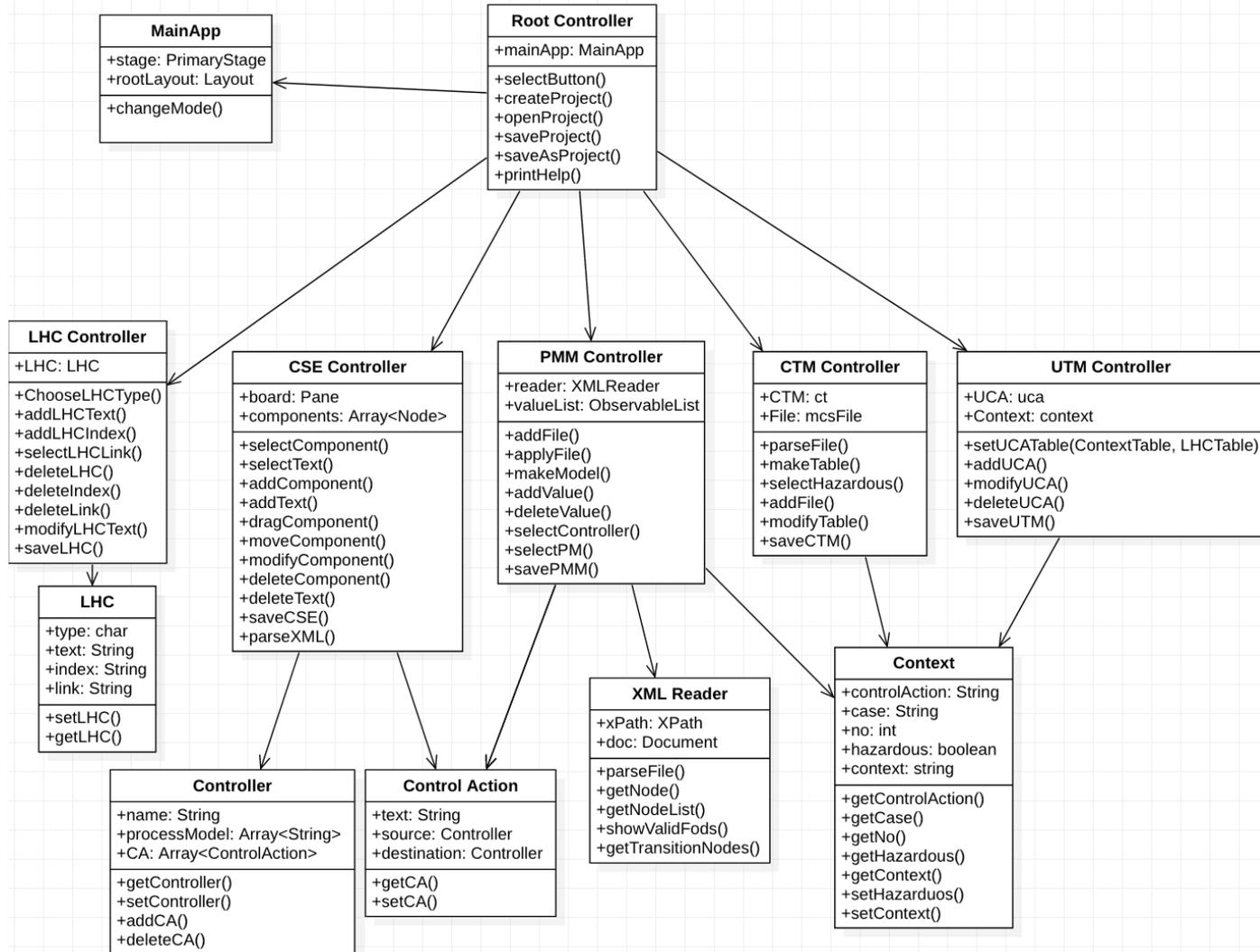
1. 모드 전환 : 메뉴바의 CTM 버튼 클릭 시, Root Controller에서 CTM Controller로의 changeMode를 호출하여 CTM 모드로 전환하고 비어 있는 Context Table을 보여주는 것으로 reply.
2. 파일 추가 : 파일 추가 팝업 창의 버튼 클릭 시 MCS File을 선택할 수 있고, MCS를 파싱하여 Table을 생성, CA에 따른 variable과 input이 적용된 Context Table을 reply
3. Context 수정 : context에 해당하는 cell을 더블클릭 시 context의 내용을 수정할 수 있고 update된 table을 reply
4. Hazardous 선택 : Table 좌측 하단의 Choice Box를 통해 Loss, Hazard, Safety Constraint를 drop down 방식으로 선택하여 update된 table을 reply
5. CT 저장 : 프로그램에서 '파일 - 저장' 또는 '파일 - 다른 이름으로 저장' 클릭 시, Root Controller에서 CTM Controller의 saveCTM을 호출하여 현재 CTM을 XML파일로 저장하고 해당 XML파일을 reply

System Sequence Diagram(UTM)



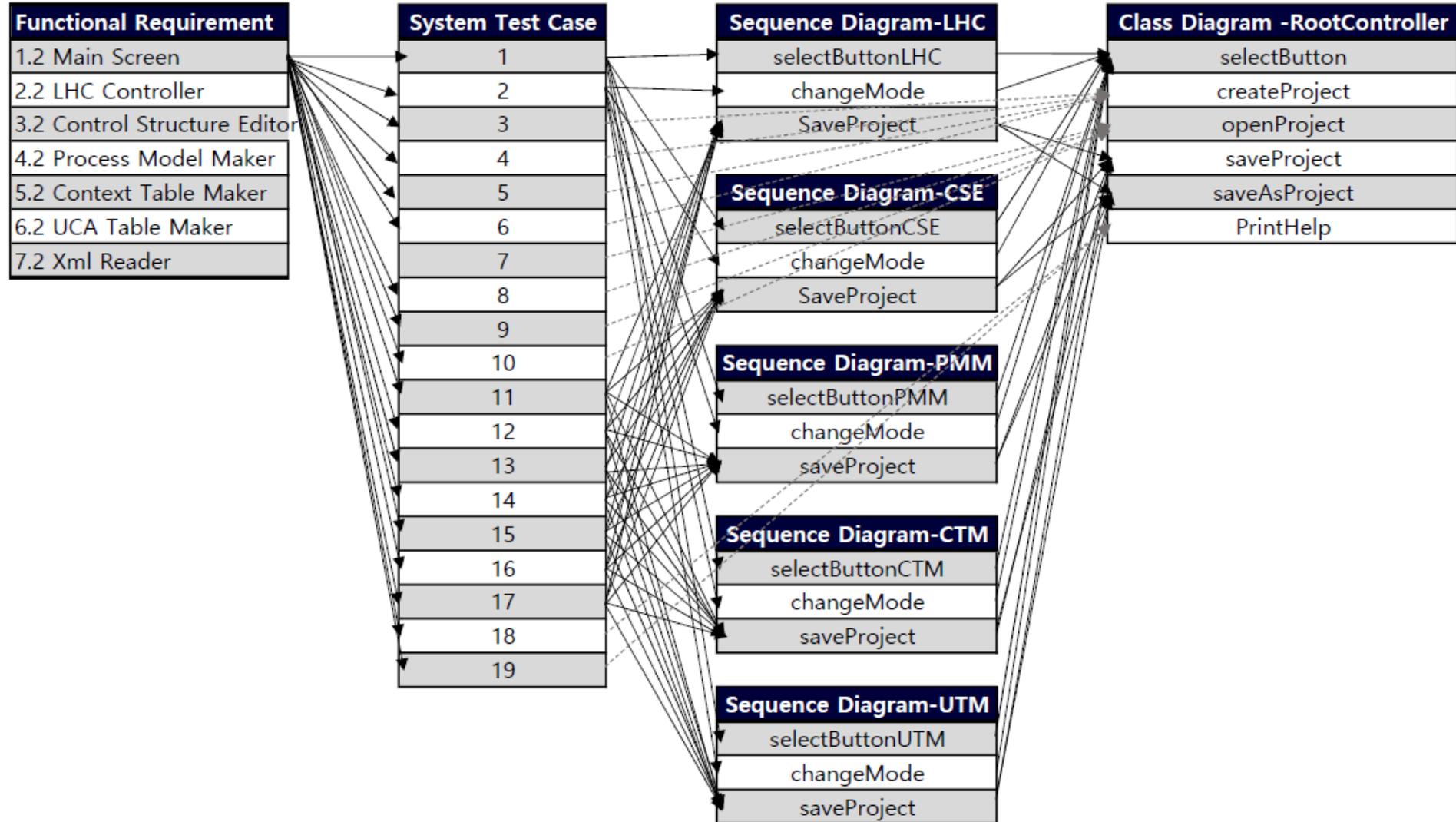
1. 모드 전환: 메뉴바의 UTM 버튼 클릭 시, Root Controller에서 UTM Controller 로의 changeMode를 호출하여 UTM 모드로 전환하고, UCA Table을 reply
2. UCA 수정: UCA Table 에 있는 UCA를 우클릭 하고 '수정'을 선택할 시, UCA Controller의 modifyUCA를 호출하여 해당 UCA를 user가 입력한 정보로 수정하고, 결과가 반영된 UCA Table을 reply
3. UCA 삭제: board 에 있는 UCA를 우클릭 하고 '삭제'을 선택할 시, UTM Controller의 deleteUCA를 호출하여 해당 UCA를 UCA Table에서 삭제하고, 결과가 반영된 UCA Table을 reply
4. UCA 추가: UCA Table 의 아래 텍스트를 입력하고 버튼 클릭 시, UTM Controller의 addUCA를 호출하여 입력한 텍스트를 UCA Table에 추가하고, 결과가 반영된 UCA Table을 reply
5. UCA Table 저장: 프로그램에서 '파일 - 저장' 또는 '파일 - 다른 이름으로 저장' 클릭 시, Root Controller에서 UTM Controller의 saveUTM를 호출하여 현재 UTM Controller의 UCA Table을 XML파일로 저장하고 해당 XML파일을 reply

Class Diagram

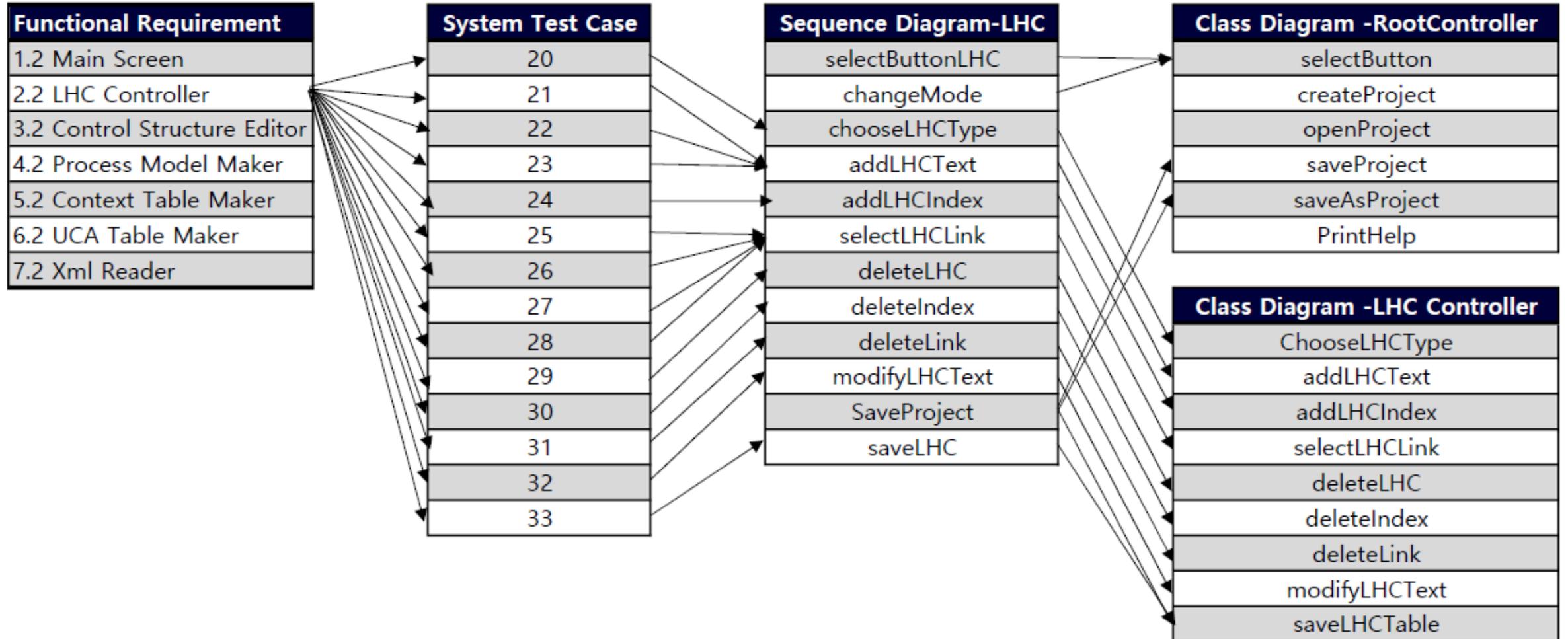


- Main App : 전체 프로그램을 구동시키기 위한 class(fxml 사용)
- Root Controller : 프로그램의 프레임을 제공해주는 class
- LHC Controller : Loss, Hazard, Safety Constraint를 추가, 수정, 삭제, 저장하는 기능을 제공해 주는 class
- LHC : LHC Controller를 위한 getter, setter가 존재하는 class
- CSE Controller : Control Structure를 작성, 수정, 삭제, 저장하는 기능을 제공해 주는 class
- Controller : Controller를 위한 getter, setter가 존재하는 class
- Control Action : Controller에 포함되는 Control Action을 위한 getter, setter가 존재하는 class
- PMM Controller : Process Model을 작성하기 위해 파일을 읽어오고, Process Model을 수정, 삭제, 저장하는 기능을 제공해 주는 class
- XML Reader : Process Model을 작성하기 위해 XML File을 파싱해 오기 위한 class
- CTM Controller : Context Table을 작성하기 위해 MCS File을 읽어 오고, Context table의 내용을 수정, 저장하는 기능을 제공해주는 class
- Context : Context를 위한 getter, setter가 존재하는 class
- UTM Controller : UCA Table을 작성하기 위해 Context를 받아와 설정하고, 추가, 수정, 삭제, 저장하는 기능을 제공해주는 class

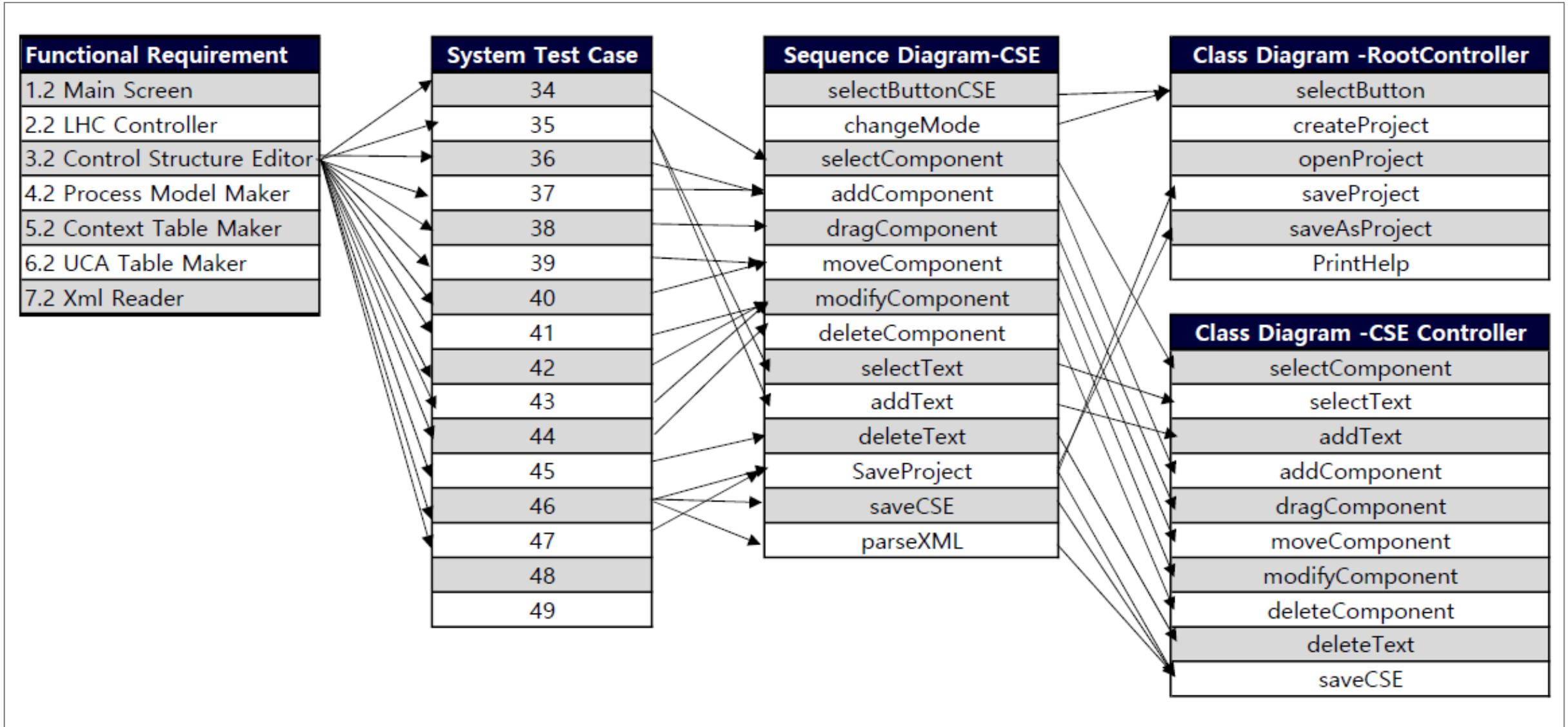
Traceability Matrix(Main)



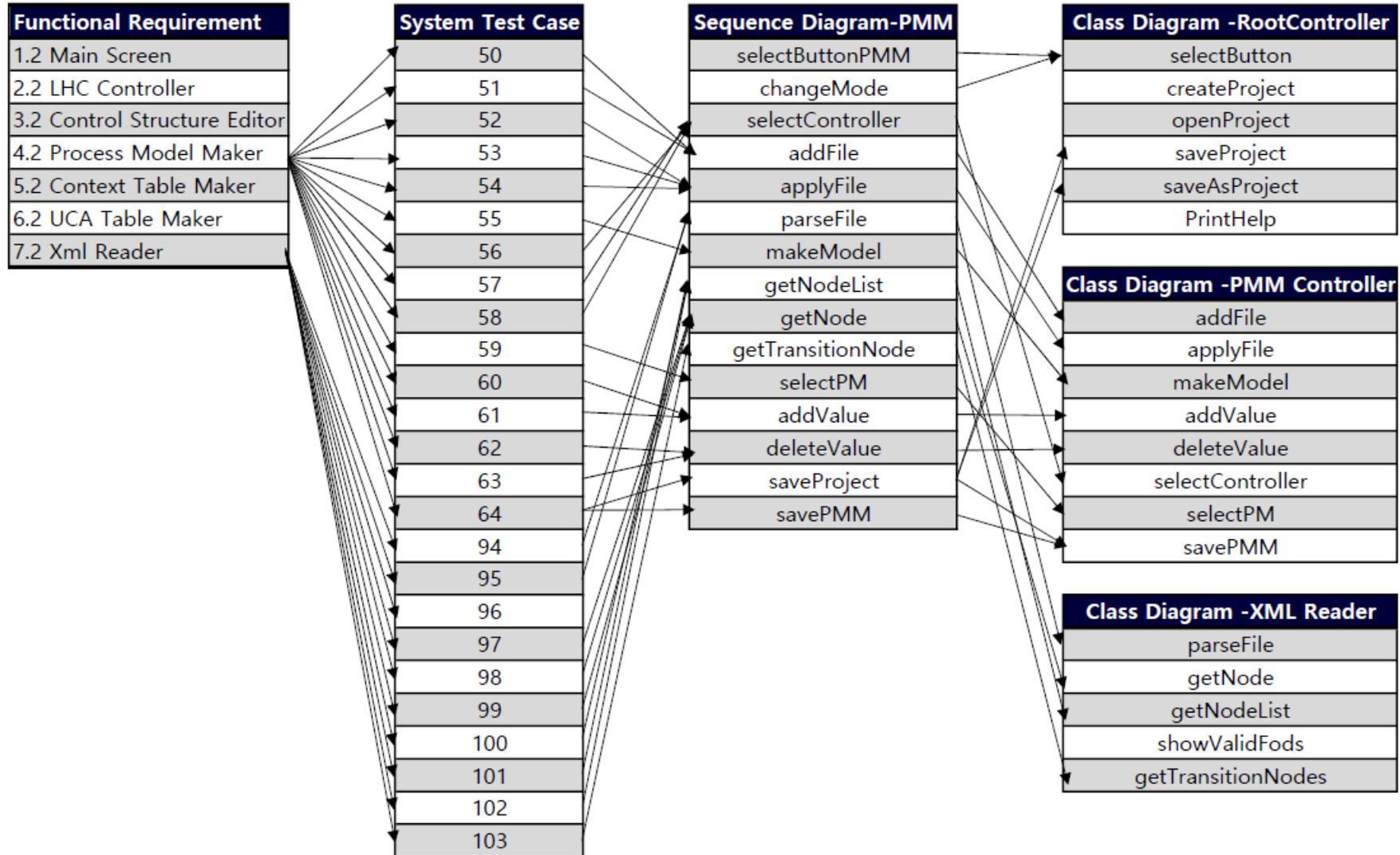
Traceability Matrix(LHC)



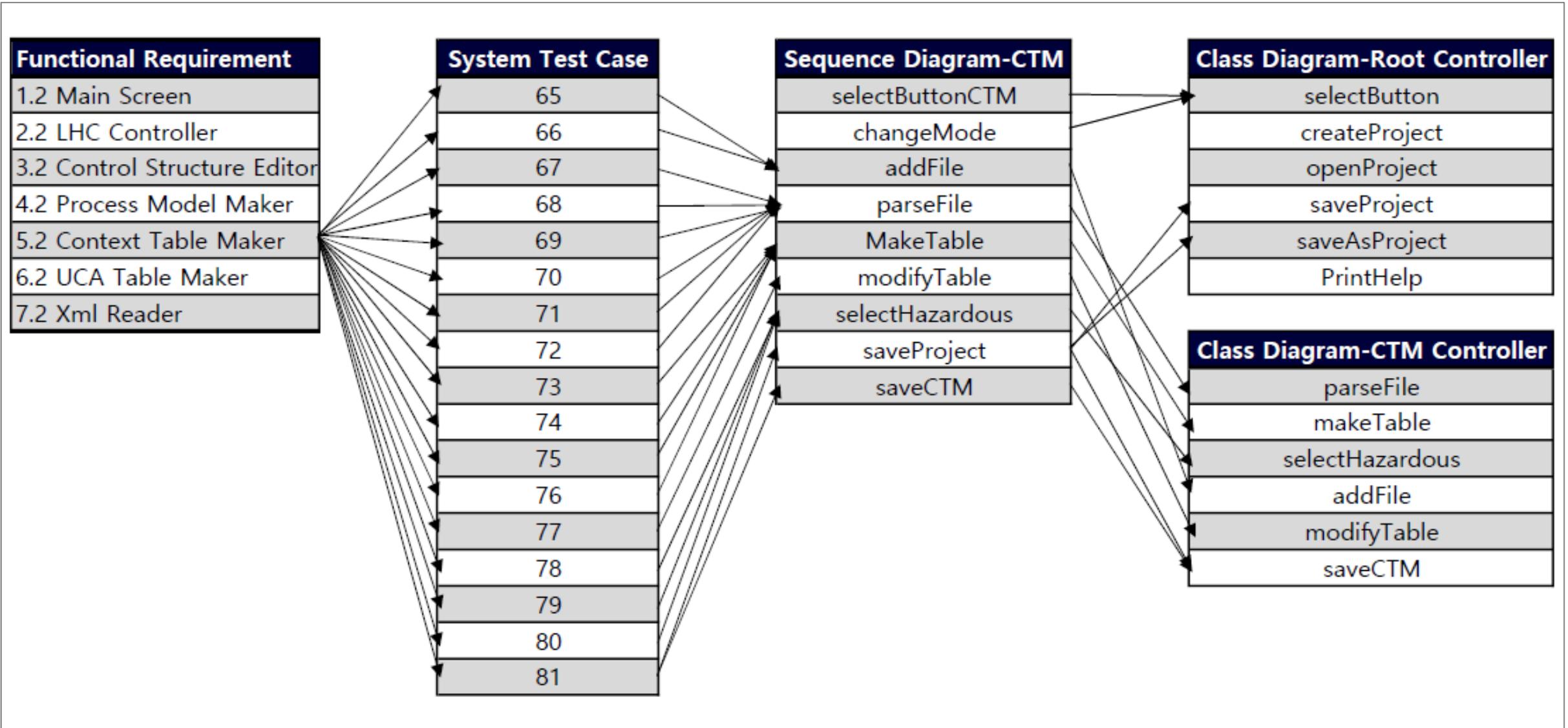
Traceability Matrix(CSE)



Traceability Matrix(PMM)



Traceability Matrix(CTM)



Traceability Matrix(UTM)

