



# 10강. 유스케이스 다이어그램 및 명세

컴퓨터과학과 김희천 교수



# 목차

- ① 유스케이스 분석
- ② 액터
- ③ 유스케이스
- ④ 유스케이스 명세





## Chapter. 1

# 유스케이스 분석

# 1. UML

## + UML은 객체지향 분석과 설계에서 모델 작성을 위한 **표준 표기법**

- × 1990년대 객체지향 방법론/표기법/도구의 난립
- × 프로세스는 환경과 상황에 따라 바뀌므로 표준의 제정이 어려움
- × 표기법은 표준이 존재할 수 있으며 공통의 의사소통 수단이 됨
- × UML은 방법론이 아닌 표기법, 즉 모델링 언어임
- × OMG는 1997년에 UML 1.1을 발표하였고 2015년에 2.5를 발표함

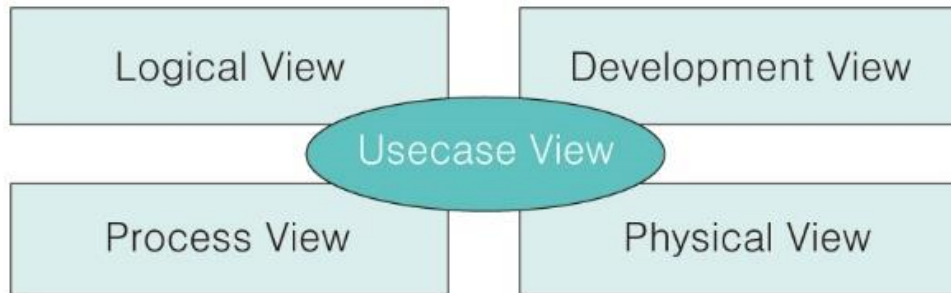
## 2. UML 다이어그램

- ✦ 설계나 구현 중인 시스템의 일부를 그래픽 형태로 표현한 것
  - 구조화된 텍스트 형태의 문서로 보완될 수 있음
- ✦ UML 다이어그램은 UML 모델의 요소와 에지로 구성됨
  - 각 UML 다이어그램마다 표현하고자 하는 주요 구성 요소가 있음
  - 클래스 다이어그램, 유스케이스 다이어그램, 시퀀스 다이어그램, ...
- ✦ UML 다이어그램의 분류
  - 구조 다이어그램은 시스템의 정적 구조를 보여주며, 추상화 수준에 따른 구성 요소들과 그들의 관계를 보임
  - 행위 다이어그램은 시스템 내 객체들의 동적 행위를 보여주며, 시간에 따른 시스템의 변화를 보임

### 3. UML과 4+1 뷰 (1/3)

#### + 4+1뷰

- × UML 모델들을 관점에 따라 분해하는 방법
- × 다양한 관점을 표현하는 여러 다이어그램이 존재함



### 3. UML과 4+1 뷰 (2/3)

#### 유스케이스뷰

- 외부에서 본 시스템의 기능을 표현하는 것
- 다른 네 가지 뷰의 기초가 됨

#### 논리뷰

- 기능 관점에서 시스템의 구성과 구성요소들 간의 상호작용을 추상화하여 표현하는 것
- 클래스 다이어그램, 시퀀스 다이어그램, 상태 머신 다이어그램

#### 프로세스 뷰

- 시스템의 실행에 초점을 맞추어 내부의 작업을 시각적으로 표현하는 것
- 액티비티 다이어그램, 상태 머신 다이어그램

### 3. UML과 4+1 뷰 (3/3)

#### 개발 뷰

- 프로그래머 관점에서 시스템의 아키텍처를 이루는 각 계층에서 시스템의 모듈들과 컴포넌트들의 구성을 표현하는 것
- 패키지 다이어그램, 컴포넌트 다이어그램

#### 물리 뷰

- 앞의 세 관점에서 표현된 시스템 설계 결과가 실세계 개체들로 어떻게 연결되는지 표현하는 것
- 최종적 시스템의 배치를 보여주는 배포 다이어그램



## 4. 유스케이스 개요

### 유스케이스



시스템이 사용자에게 제공하는  
유용하고 완전한 단일(단위) 기능

#### + 시스템이 제공하는 특정 기능의 동작을 설명한 시나리오

- × 액터와 시스템 간의 상호작용을 이벤트 흐름으로 기술
- × 사용자 관점에서 시스템을 모델링하기 위한 도구
- × 완성될 목표 시스템의 겉모습을 설명하는 것
- × 시스템의 내부 구조나 구현 방법을 기술하는 것이 아님

## 5. 유스케이스 분석(1/2)

### 유스케이스 분석



- 사용자 요구사항을 정리하여 구조화하는 과정
- 초기 단계에서 목표 시스템을 정확히 이해하고 세부 기능을 파악해야 함

#### + 유스케이스 분석의 결과는

유스케이스 다이어그램과 유스케이스 명세

#### + 유스케이스 분석의 효과

- 작업을 유스케이스로 구분하여 작업을 팀에 할당함
- 시스템의 기능에 관한 시나리오를 이해하면 관리에 도움을 줌
- 테스트 케이스 작성을 위한 기준을 제시

## 5. 유스케이스 분석(2/2)

### + 유스케이스 분석 전에 자연어로 된 사용자 요구사항을 분석함

× 구조화보다는 정확성, 완전성, 일관성 등을 검토

R.1. 의사는 등록되어 있는 환자에 대해 Subject Evaluation, Object Evaluation, Assessment, Plan 등의 상세한 내용을 포함한 전자의무기록(EMR)을 시스템에 생성할 수 있어야 한다.

EMR은 Electronic Medical Record의 약어이다.

### + 유스케이스 분석의 첫 단계는 시스템과 상호작용하는 개체를 찾는 일

× 액터는 요구사항에서 명사로 표현됨

× 동사는 유스케이스에 해당



## Chapter. 2

# 액터

# 1. 표기법

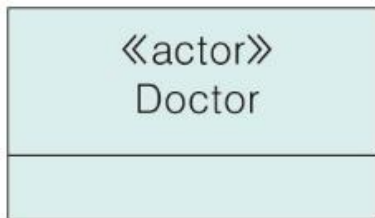
## + 액터의 경우 막대 인간으로 표현

✕ 또는 스테레오타입을 사용하여 개체의 이름과 함께 《actor》를 표시

## + 스테레오 타입의 사용

✕ 요소의 의미를 명확히 하기 위해 사용되는 UML 확장법으로 키워드를 《...》로 감싸 표현함

✕ 자주 사용되는 스테레오 타입을 아이콘으로 표시하기도 함



## 2. 액터 찾기

### 액터



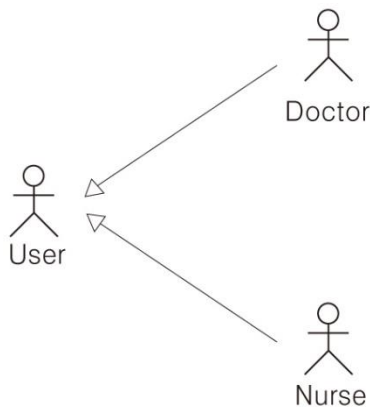
- 구현 대상은 아니며 외부에서 시스템과 상호작용하는 개체
- 시스템을 사용하는 사람이나 외부 시스템

### + 액터 찾기

- × 시스템과 상호작용하나 그 자체에 대한 분석이 필요 없는 개체
- × 사용자 외에도 관리자, 유지보수자, 시스템과 관련 있는 모든 사람이나 다른 시스템이 고려 대상임

### 3. 액터 간 관계

- + 액터들 간에는 일반화(또는 상속) 관계가 존재할 수 있음
  - × 공통의 요구사항을 가지는 여러 액터들을 일반화시켜 모델을 단순화 함
  - × HIS에서 Doctor과 Nurse를 일반화하여 User를 표현함
  - × 속이 빈 삼각형을 가진 실선으로 일반화 관계를 표현함





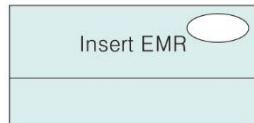
## Chapter. 3

# 유스케이스



# 1. 표기법과 유스케이스의 의미

## + 유스케이스는 대개 타원으로 표현함



시스템 사용에 관한  
시나리오를 나타내는 것

시나리오는 일련의  
사건의 흐름으로 표현됨

성공적 유스케이스만을  
의미하는 것은 아님



유스케이스는 유사 기능을 표현하는  
모든 시나리오를 포함하여  
일반화하고 구조화시켜 표현한 것

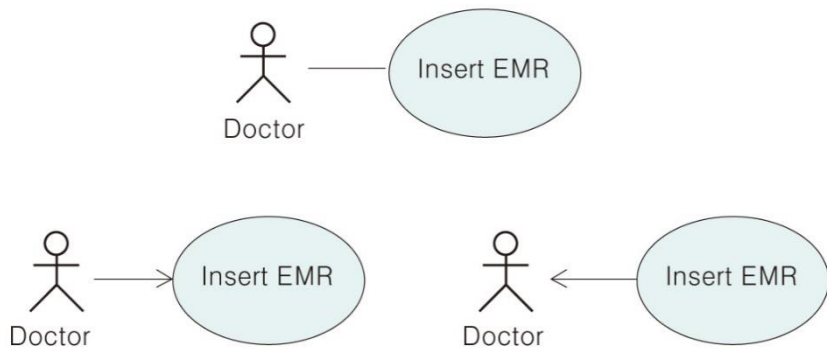
사용자 관점에서 명세되었을 때  
요구사항을 검증 할 수 있는  
기준이 됨

유스케이스의 이름은 타원의  
안에 두며 동사로 시작함

## 2. 유스케이스와 액터와의 관계

### + 액터와 유스케이스를 선으로 연결

- × 유스케이스를 시작시키는 액터로부터 유스케이스로 화살표가 함함
- × 결과를 제공하는 유스케이스로부터 제공받는 액터로 화살표가 함함
- × 왼쪽에 유스케이스를 시작시킨 액터를, 오른쪽에 결과를 받는 액터를 둬
- × 양방향의 경우는 화살표를 생략함



### 3. 유스케이스 간 관계 (1/4)

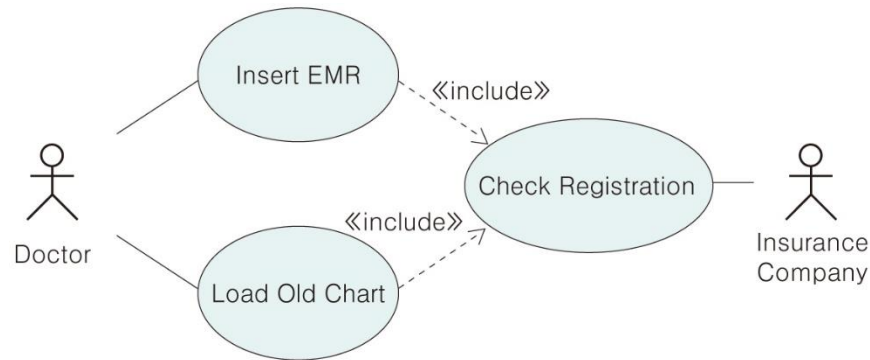
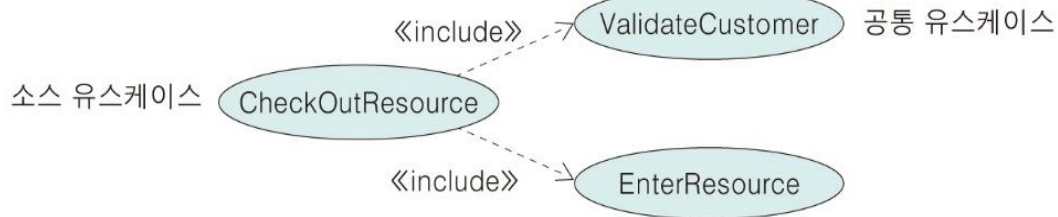
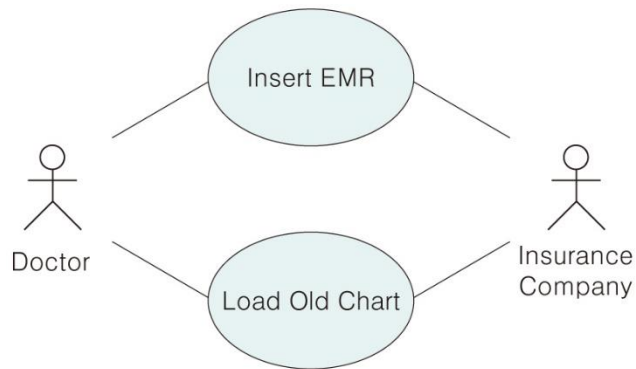
#### + 포함 관계 - 《include》

- × 두 유스케이스에서 중복되는 기능이 있는 경우, 중복된 부분을 별도의 유스케이스로 분리
- × 점선의 화살표를 사용하며 소스 유스케이스에서 공통 유스케이스로 향함
- × 소스 유스케이스는 공통 유스케이스 없이는 완전한 유스케이스가 되지 못함
- × 분리된 유스케이스로 만들면 유지보수성과 재사용성이 증가함

R.2. HIS시스템은 신규 환자 등록과는 별도로 기존 환자 차트를 화일로부터 로드하여 추가로 입력할 수 있다.

R.3. R1및 R2의 경우에는 보험회사를 통해 등록여부를 재확인하게 된다.

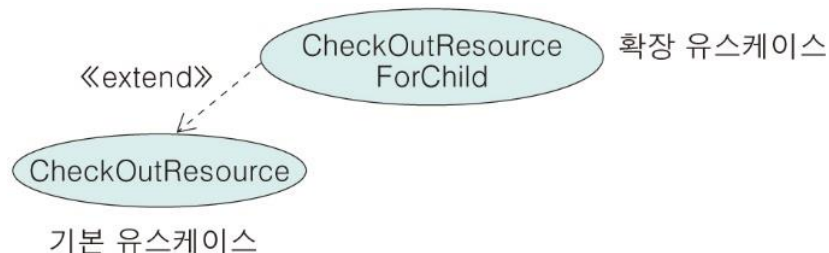
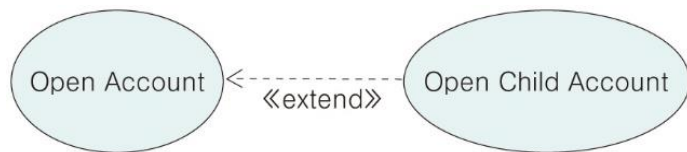
### 3. 유스케이스 간 관계 (2/4)



### 3. 유스케이스 간 관계 (3/4)

#### + 확장 관계 - 《extend》

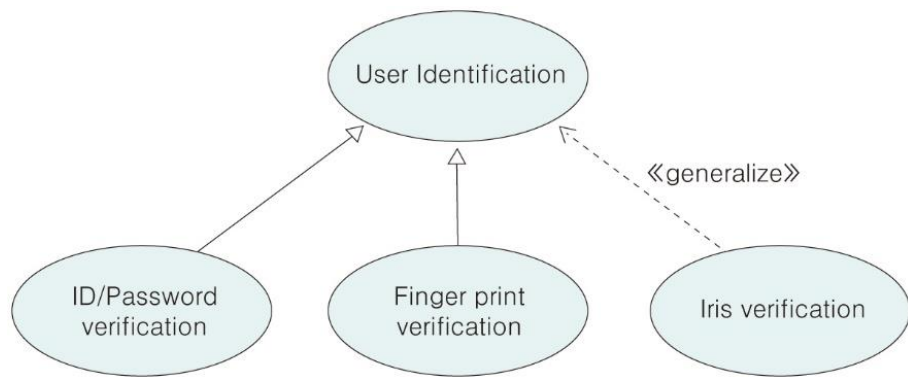
- × 특정 조건에서 선택적으로 사용되는 시나리오를 분리
- × 확장 유스케이스는 기본 유스케이스에서 **예외적이며 선택적인 사건의 흐름**을 떼어 내는 것
- × 기본 유스케이스는 그 자체로 완전한 유스케이스임
- × 점선의 화살표를 사용하며 확장 유스케이스로부터 기본 유스케이스로 화살표가 함함
- × 《include》 관계의 경우와 화살표 방향이 반대임



### 3. 유스케이스 간 관계 (4/4)

#### + 상속 관계 - 《generalize》

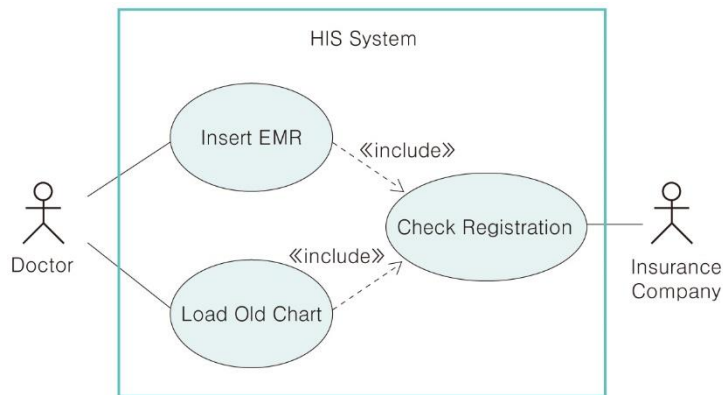
- × 전체적인 흐름은 동일하나 일부에서 구체적인 방법이나 내용이 틀린 경우
- × 자식 유스케이스는 부모 유스케이스에서 사용되는 흐름과 일치함
- × 사용자 인증 유스케이스가 부모가 되며 아이디/암호, 지문 인식, 홍채 인식을 사용한 인증을 자식 유스케이스로 둠



## 4. 시스템의 경계

### + 시스템 경계의 표시

- × 서브젝트 경계라고도 함
- × 시스템 내부에 존재하는 유스케이스 집합을 사각 영역 안에 위치시킴
- × 사각형 상단에 시스템의 이름을 표시
- × 시스템 내부는 분석과 설계를 통한 개발이 필요한 부분
- × 액터들은 사각형 외부에 위치함





Chapter. 4

# 유스케이스 명세



# 1. 유스케이스 다이어그램과 명세

## + 유스케이스 다이어그램

- 액터와 유스케이스 및 이들 간의 관계를 요약적으로 표현
- 요구사항을 구조화하고 고객과의 원활한 의사소통을 위한 수단

## × 유스케이스 명세

- 유스케이스 다이어그램만으로는 구체적이지 못하므로 상세한 설명을 추가해야 함

## 2. 유스케이스 명세 (1/2)

- + 유스케이스 명세는 시나리오의 구체적 사건 흐름을 텍스트 형식으로 기술한 것
- + 유스케이스 명세의 구성



## 2. 유스케이스 명세 (2/2)

### + 기본 흐름

- × 원하는 목적을 달성하는 성공적 시나리오
- × 번호와 제목을 달아 구조적으로 작성

### + 대체 흐름

- × 기본 흐름에서 벗어나는 다양한 상황을 기술
- × 기본 흐름의 번호를 이용하여 다른 상황에서 일어나는 요약된 시나리오로 기술함

### + 특수 요구사항

- × 시간, 성능, 품질 등과 관련된 비기능 요구사항을 기술
- × 도메인 요구사항이나 UI 들에 관한 요구사항을 자유롭게 기술

### 3. 유스케이스 명세의 예 (1/5)

유스케이스 : 진료 예약

관련 액터 : 환자, 의사 목록 시스템

요약 :

환자가 원하는 의사에게 진료 예약을 할 수 있도록 해준다. 뿐만 아니라, 진료 시점 24시간 전까지는 신청 내역을 확인하고 변경할 수도 있다. 진료 시스템은 예약이 가능한 모든 의사들의 목록을 보여준다.

선행 조건 : 환자가 진료 예약 시스템에 로그인 되어 있어야 한다.

종료 조건 :

환자가 진료 예약 시스템을 통해 신청한 요청이 서버에서 정상적으로 처리되어야 한다.

### 3. 유스케이스 명세의 예 (2/5)

#### <기본 흐름>

##### 1. 스케줄 생성

시스템은 환자가 사용 가능한 기능(스케줄 생성, 수정, 삭제)을 화면에 보여준다. 환자는 "스케줄 생성"을 선택한다.

##### 2. 의사 선택

시스템은 진료 예약 가능한 의사들의 리스트를 의사 목록 시스템으로부터 가져와서 화면에 보여준다. 환자는 예약 가능한 스케줄상의 진료를 신청할 수 있다. 환자는 최종적으로 스케줄 확정 버튼을 누르기 전까지 원하는 의사 및 스케줄을 확인하면서 추가 및 수정, 삭제를 할 수 있다.

### 3. 유스케이스 명세의 예 (3/5)

#### <기본 흐름>

##### 3. 스케줄 확정

환자는 스케줄 확정 버튼을 눌러서 진료 예약을 완료한다.

시스템은 해당 스케줄에 대한 확정 번호를 화면에 보여준다.

시스템은 환자의 스케줄 정보를 저장하고, 종료한다.

### 3. 유스케이스 명세의 예 (4/5)

#### <대체 흐름>

##### 1. 스케줄 수정

환자가 이미 스케줄을 가지고 있는 상황이다. 시스템은 기존에 확정되어 있는 진료 스케줄을 화면에 보여준다. 그리고, 그 상태에서부터 기본 흐름 2번으로 이어진다.

##### 2. 미확인 환자

환자가 로그인하지 않은 상태로 접근한 경우 에러 메시지를 화면에 보여주고 종료한다.

### 3. 유스케이스 명세의 예 (5/5)

#### <대체 흐름>

##### 3. 종료

진료 예약 시스템은 언제든지 환자가 종료 버튼을 누름으로써 시스템을 종료할 수 있도록 한다. 이 경우, 현재 상황을 저장 할 지의 여부를 확인하고, 그 여부에 따라 스케줄 상황의 저장 여부를 결정한 후에 종료한다.

##### 4. 진료 예약시간 마감

예약 마감시간 10분 전, 3분 전, 1분 전에 각각 경고 메시지를 보여주고, 강제적으로 시스템을 종료시킨다.

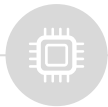
#### <특수 요구사항>

진료 예약은 동시 접속자수 1,000명까지 커버할 수 있어야 하며, 이 경우 최종 확정에 대한 서버 응답시간은 5초 이내이어야 한다.



## 4. 요구사항 명세 작성 팁

- + 유스케이스 분석은 유스케이스 다이어그램을 보완하는 추가적 설명을 텍스트로 제공함
- + 유스케이스 상세화 과정에서 초기에 발견하지 못했던 액터나 시나리오들을 발견할 수 있음
- + 고객과 반복적으로 의사소통함으로써 이전 작업의 수정을 통해 유스케이스 모델에 정확성을 가하게 됨



다음강의

# 11강. 액티비티 다이어그램

