

Korea  
Software  
Technology  
Association



## UML2.x 기초 다루기

**훈련기간: 2010.01.25 ~ 02.05**

**강사명: 손재현** -넥스트리소프트  
-jhsohn@nextree.co.kr

## □ 교육 목표 & 특징

- UML2.x의 이해
- 유스케이스 작성
- 객체모델링 이해
- UML2.x의 다양한 다이어그램 이해 및 활용
- 모델링 도구 사용법 습득

- 본 강의는 아래 기술에 대한 이해를 필요로 합니다.
  - 객체지향 언어(Java) 기초
  - 개발프로세스 이해

□ 교육은 매 회 4 시간씩 총 5회에 걸쳐 진행합니다.

1 일차	2 일차	3 일차	4 일차	5 일차
<ul style="list-style-type: none"> <li>- UML 개요</li> <li>UML 소개</li> <li>UML 역사</li> <li>UML 다이어그램분류</li> </ul>	<ul style="list-style-type: none"> <li>- 구조 다이어그램</li> <li>클래스</li> <li>객체</li> <li>컴포넌트</li> <li>배치</li> </ul>	<ul style="list-style-type: none"> <li>- 행위 다이어그램</li> <li>유스케이스</li> <li>액티비티</li> <li>상태기계</li> </ul>	<ul style="list-style-type: none"> <li>- 상호작용 다이어그램</li> <li>상호작용 Overview</li> <li>시퀀스</li> <li>커뮤니케이션</li> <li>타이밍</li> </ul>	<ul style="list-style-type: none"> <li>- 유스케이스 I</li> <li>유스케이스 개요</li> <li>유스케이스 내용</li> <li>유스케이스 다이어그램</li> </ul>
6 일차	7 일차	8 일차	9 일차	10 일차
<ul style="list-style-type: none"> <li>- 유스케이스 II</li> <li>유스케이스 목표수준</li> <li>유스케이스 명세</li> <li>유스케이스 패턴</li> </ul>	<ul style="list-style-type: none"> <li>- 유스케이스 III</li> <li>유스케이스 분석기법</li> <li>분석클래스</li> <li>제어클래스</li> <li>실체클래스</li> </ul>	<ul style="list-style-type: none"> <li>- 요구사항 모델실습 I</li> <li>유스케이스</li> <li>사용자 시나리오</li> <li>핵심개념 모델</li> </ul>	<ul style="list-style-type: none"> <li>- 요구사항 모델실습 II</li> <li>인터페이스 추출</li> <li>유스케이스 분석</li> <li>컴포넌트 식별</li> </ul>	<ul style="list-style-type: none"> <li>- 설계모델 실습</li> <li>컴포넌트 설계</li> <li>유스케이스 설계</li> <li>도메인 모델</li> </ul>

# 5일차 – 유스케이스

1. 유스케이스 명세
2. 유스케이스 패턴

ONE STEP AHEAD

# 1. 유스케이스 명세

ONE STEP AHEAD

- ❑ 유스케이스 데이터 작성
- ❑ 유스케이스 작성절차
- ❑ 특수한 유스케이스
- ❑ 작성 접근방법
- ❑ 유스케이스 작성 지침
- ❑ 유스케이스 패키지
- ❑ 전통적인 실수
- ❑ 유스케이스 워크샵

## □ 데이터 처리

- 얼마나 세부적으로 작성해야 하는가?
  - 적절한 정보를 유지하며 독자가 읽기 편한 정도로 작성하라!
- 데이터 처리 이슈
  - 데이터를 매우 상세하게 작성시 -> 유스케이스의 흐름을 읽기가 어려워짐
  - 데이터를 너무 개략적으로 작성시 -> 데이터와 관련된 모든 정보를 빼고 유스케이스를 작성한다면 유스케이스가 원하는 바를 제대로 전달하지 못함

# 유스케이스 데이터 작성(2/4)

## □ 예제-잘못된 수준

### 예제1 - 너무 간략한 경우:

1. 고객이 인출카드를 삽입하면 유스케이스가 시작된다.
2. 시스템은 인출카드에 대한 정보를 읽어 들이고, 고객에게 비밀번호를 요청한다.
3. 시스템은 사용 가능한 메뉴항목을 보여준다.
4. 고객은 현금인출을 선택하고, 인출금액을 입력한다.
5. 시스템은 입력된 금액을 지급하고, 영수증을 발급하고, 카드를 반환한다.
6. 고객은 현금과 카드와 영수증을 수령하고, 유스케이스는 종료된다.

### 예제2 - 너무 상세한 경우:

1. 고객이 인출카드를 삽입하면 유스케이스가 시작된다.
2. 시스템은 입력된 인출카드를 읽어 은행 식별자, 계좌 번호, 비밀번호를 얻는다. 그리고, 고객에게 비밀번호를 요청한다. 비밀번호는 6자리이고 반복되는 번호가 존재하면 안 된다.
3. 시스템은 입력된 비밀번호와 카드로부터 읽어 들인 비밀번호를 비교하여 서로 부합되는지 검증한다.
4. 입력된 비밀번호가 유효할 경우, 시스템은 메뉴 항목을 보여준다. 이는 현금인출, 입금, 이체, 잔액조회로 구성된다.
5. 고객은 현금인출을 선택하고, 인출금액을 입력한다.
6. 시스템은 시스템 내에 충분한 금액이 존재하는지 확인한다.
7. 시스템은 입력된 금액이 1만원 단위이고 20만원이 넘지 않는지 확인한다.
8. 시스템은 고객의 잔고에 충분한 금액이 있는지 확인하기 위하여 고객의 은행에 접속한다.
9. 고객의 잔고가 충분할 경우, 시스템은 입력된 금액을 지급하고 영수증을 출력한 후, 카드를 반환한다. 영수증 정보는 아래와 같다.  
- 날짜, 시간, ATM 위치, 은행 식별자, 지급금액, 지급 식별자
10. 고객은 현금과 카드와 영수증을 수령하고, 유스케이스는 종료된다



# 유스케이스 데이터 작성(3/4)

## □ 예제-적절한 수준(1/2)

1. 고객이 인출카드를 삽입하면 유스케이스가 시작된다.
2. 시스템은 입력된 **인출카드** 정보를 읽는다.
3. 시스템은 입력된 **비밀번호**와 카드로부터 읽어 들인 비밀번호를 비교하여 서로 부합되는지 검증한다.
4. 입력된 비밀번호가 유효할 경우, 시스템은 메뉴 항목을 보여준다. 이는 현금인출, 입금, 이체, 잔액조회로 구성된다.
5. 고객은 현금인출을 선택하고, 인출금액을 입력한다.
6. 시스템은 시스템 내에 충분한 금액이 존재하는지 확인한다.
7. 시스템은 입력된 금액이 1만원 단위이고 20만원이 넘지 않는지 확인한다.
8. 시스템은 고객의 잔고에 충분한 금액이 있는지 확인하기 위하여 **고객의 은행**에 접속한다.
9. 고객의 잔고가 충분할 경우, 시스템은 입력된 금액을 지급하고 **영수증**을 출력한 후, 카드를 반환한다.
10. 고객은 현금과 카드와 영수증을 수령하고, 유스케이스는 종료된다

유스케이스 명세서

**고객:** ATM이 지급 가능한 은행에 계좌를 가지고 있는 사람. 고객정보는 이름, 주소, 주민번호 등이 존재한다.  
**인출카드:** 금융관련 행위를 위하여 금융기관이 발행한 것으로 마그네틱에 다음과 같은 정보를 유지하고 있다.

- 은행정보: 해당기관에서 발행한 금융기관 고유 정보
- 고객정보: 금융기관이 고객에게 부여한 정보
- 비밀번호: 고객이 카드 발급 당시 정한 고유번호

**비밀번호:** 고객이 카드 보안을 목적으로 지정한 식별 번호. 이는 최대 6자리까지 허용되고 반복되는 번호는 허용되지 않는다. 고객정보를 식별하기 위하여 ATM은 항상 고객의 비밀번호를 요구한다.

**고객 은행:** 고객에게 은행 카드를 발급한 금융기관으로 고객은 해당 은행에 계좌를 보유하고 있다.

**계좌:** 금융기관이 발행한 것으로 고객의 현금이나 유가 증권 등에 대한 정보를 가지고 있는 것. 하나의 계좌는 고유한 계좌번호를 가지고 있고 ~

**영수증:** 특정 금융행위로 인하여 출력된 기록. 영수증은 다음과 같은 정보를 갖는다.

- 날짜 및 시간, ATM 위치, 은행 식별자, 지급금액, 지급 식별자

용어집 or 기타 산출물

## 유스케이스 데이터 작성(4/4)

### □ 예제-적절한 수준(2/2)

- 유스케이스를 읽을 때, 흐름에 초점을 두고 읽기가 매우 용이함
- 유스케이스를 작성 할 때, 필요한 정보는 위와 예제와 같이 볼드체로 표시하고, 용어집이나 기타 산출물에 따로 관리하는 것이 바람직 함
- 이와 같은 용어는 다른 유스케이스에서도 공통적으로 사용할 수도 있으므로 비용적인 측면에서도 매우 유용함
- 부가적으로 위와 같이 유스케이스 흐름과 데이터를 정리하다 보면, 이에 관련된 비즈니스 규칙이 도출 됨

## □ 기본 절차

- 시스템 범위와 경계 설정
- 주요 액터와 목표 찾기
- 유스케이스 찾기 및 각 유스케이스 서술

## 유스케이스 작성절차(2/2)

### □ 상세 절차

- 시스템 범위와 경계를 설정
- 시스템과 관련된 일차 액터 식별 - 브레인스토밍
- 시스템을 위한 사용자 목표 찾기 - 브레인스토밍
- 일차 액터에 대한 요약수준 유스케이스를 파악
- 요약수준 유스케이스에 대한 정제 - 목표를 추가, 삭제, 합병
- 상세 기술할 유스케이스를 선택
- 이해관계자와 이해관계, 선조건과 후조건을 파악
- 주요 흐름 작성
- 대안흐름 조건 나열 - 브레인스토밍
- 대안흐름 작성
- 복잡한 흐름을 하위 유스케이스로 작성, 사소한 하위 유스케이스는 합병
- 유스케이스 구조화를 실시 - 필요에 따라 추가, 삭제, 합병

## 특수한 유스케이스(1/2)

### □ 로그인 유스케이스

- 로그인 유스케이스는 우리가 찾고자 하는 사용자 목표수준의 유스케이스가 아님
- 로그인 유스케이스를 작성하는 방법과 표현에 대하여 고민하기보다는 사용자 목표 수준의 유스케이스를 기술하는데 초점을 두는 것이 현명한 방법
- 단순히 해당 사용자 목표 수준의 유스케이스의 선조건에 간단하게 작성하는 것이 바람직함
- 로그인 하는 절차가 복잡(이도 역시 기능수준의 유스케이스 임)할 경우에는 작성할 수도 있지만, 로그인 유스케이스의 작성을 추천하지는 않음

## 특수한 유스케이스(2/2)

### □ CRUD 유스케이스

- 데이터베이스에서 특정 데이터를 생성(Create), 검색(Retrieve), 갱신(Update), 삭제>Delete)하는 유스케이스를 CRUD 유스케이스라고 함
- 예제
  - (회원정보 생성, 회원정보 검색, 회원정보 갱신, 회원정보 삭제) vs (회원정보 관리)
- 적용
  - 프로젝트 초기에는 유스케이스의 숫자를 적게 하기 위하여 회원정보 관리 유스케이스로 처리하고, 작성 중 유스케이스가 점점 복잡해지면 새로운 하위 수준의 유스케이스로 분리

### □ Start up 유스케이스

- 종종 시스템을 시작 시키는 유스케이스를 유스케이스 다이어그램에 표현하기도 함
- 로그인 유스케이스와 마찬가지로 하위 기능 수준의 유스케이스이므로 무시해도 상관없음
- 아주 중요한 시스템(예; 생명과 관련된 실시간 시스템)에 있어서 시스템 시작 시, 시스템의 안정성 확인 등의 이유로 Startup 유스케이스를 표현하는 경우도 있음

## 작성 접근방법(1/4)

### □ 처음부터 상세하게 작업하지 않음

- 유스케이스의 작성은 특정 시점에서 세부적인 사항을 모두 작성하지 않음
- 시간의 흐름에 따라 그 정밀도를 높여가는 방식으로 작성

## 작성 접근방법(2/4)

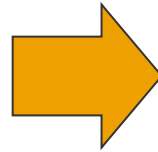
### □ 정밀도

- 1단계: 액터와 목표 작성
  - 시스템과 관련된 액터와 이의 목표를 작성
  - 정확성과 안정성을 위하여 이를 검토
- 2단계: 유스케이스 요약서 & 성공 시나리오 작성
  - 작성하려고 선택한 유스케이스에 대하여 간단한 요약을 작성하거나 주요 성공 시나리오를 작성
  - 시스템이 이해관계자의 이해관계를 정확히 반영하고 있는지 검토
- 3단계: 실패조건 작성
  - 주요 성공 시나리오를 완성하고 발생 가능한 모든 실패 상황에 대해서 브레인스토밍을 수행
- 4단계: 실패처리 작성
  - 시스템이 각 실패 상황에서 대처하는 방법을 작성
- 5단계: 기타항목 작성
  - 선조건이나 후조건, 특수 요구사항등 유스케이스에서 요구하는 기타 항목들에 대하여 작성
  - 이는 시나리오를 작성할 때 병행할 수도 있으나 위의 단계보다 비교적 우선순위가 떨어지는 작업

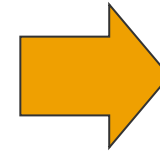


# 작성 접근방법(3/4)

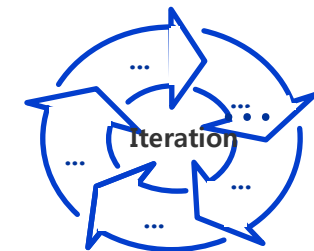
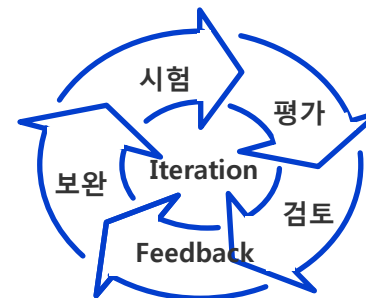
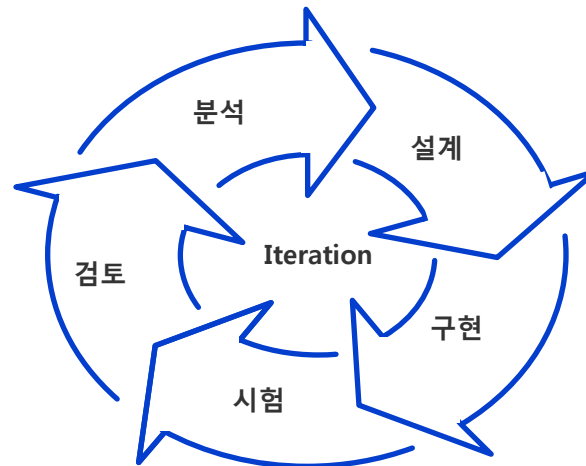
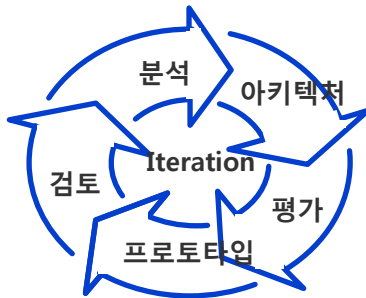
Draft
이름 : 액터 : 요약 :



1차 유스케이스
이름 : 액터 : 요약 : 기본 흐름 :



개선된 유스케이스
이름 : 액터 : 요약 : 선조건 : 후조건 : 기본 흐름 : 확장 흐름 : 특별 요구사항 :



## 작성 접근방법(4/4)

### □ 반복별 유스케이스 작성 예제

- 다음의 예제는 절대적인 것은 아니고, 반복적으로 유스케이스를 작성한다는 점에서 유스케이스 작성자의 이해를 돕기 위하여 참조 정도로 이용하도록 함
- 단, 도입단계가 끝나는 시점에서의 유스케이스는 10% 정도의 유스케이스(위험요소가 높거나 핵심 기능을 포함하고 있는 유스케이스를 아키텍트가 선정)가 상세하게 작성되고, 발단단계가 끝나는 시점에서의 유스케이스는 80~90% 정도가 상세하게 작성되어야 한다는 점은 주목할 사항

도입 반복 #1	발단 반복 #1	발단 반복 #2	발단 반복 #3	발단 반복 #4
대부분의 유스케이스 이름을 작성하고, 각 유스케이스는 짧게 요약되며, 10% 정도만 상세하게 작성한다.	유스케이스의 30% 정도를 상세하게 작성한다.	유스케이스의 50% 정도를 상세하게 작성한다.	유스케이스의 70% 정도를 상세하게 작성한다.	유스케이스의 80~90% 정도를 상세하게 작성한다.

# 유스케이스 작성 지침(1/7)

## □ 단순한 문법을 사용하라

- 문장구조는 '주어 ~ 목적어 ~ 부사 ~ 동사'와 같이 단순하게 작성

예제)

시스템에 ~ 일정액을 ~ 잔고에서 ~ 차감한다.

## □ 주체를 명확히 표현하라

- 해당 시나리오를 수행하는 주체를 명확히 표기
- 즉, 해당 액터와 시스템을 반드시 명기하도록 함

## □ 시스템 외부로부터 작성하기 시작하라

- 유스케이스를 시스템 내부에서 작성하는 것은 흔히 저지르는 실수
  - 특히, 프로그래머가 유스케이스를 작성할 경우

나쁜 작성법)

현금 인출카드를 받고 비밀번호를 입력 받는다.

계좌 잔고에서 금액을 차감한다

좋은 작성법)

고객은 현금 인출카드를 넣고 비밀번호를 입력한다.

시스템은 잔고에서 일정액을 차감한다 .

## 유스케이스 작성 지침(2/7)

### □ 액터의 움직임이 아닌 의도를 보여주어라

- 사용자 인터페이스에서 사용자의 움직임을 서술하는 것은 유스케이스 작성 시에 가장 흔히 저지르는 중대한 실수
- 이는 유스케이스를 읽기 어렵게 하고, UI 설계자의 UI 설계에 대한 작업을 제한할 수 있으며, UI가 변경되었을 때 유스케이스를 재작성해야 하는 문제점이 발생함

#### 나쁜 작성법)

1. 시스템이 이름을 묻는다.
2. 사용자가 이름을 입력한다.
3. 시스템이 주소를 묻는다.
4. 사용자가 주소를 입력한다.
5. 사용자가 “확인”을 클릭한다.
6. 시스템은 사용자 정보를 보여준다.

#### 좋은 작성법)

1. 사용자가 이름과 주소를 입력한다.
2. 시스템은 사용자 정보를 보여준다.

## 유스케이스 작성 지침(3/7)

### □ 상호작용을 일관적이고 합리적으로 작성하라(1/2)

- 유스케이스의 한 단계를 트랜잭션화 시켜서 표현하는 것이 일관적으로 보여짐
- 유스케이스 작성시 상세화 정도에 차이는 있겠지만 아래와 같이 작성하는 것을 권고
- 형식
  1. 일차 액터가 시스템에 데이터를 입력하고 이에 대하여 요청한다.
  2. 시스템은 입력된 데이터와 요청을 검증하고, 내부상태를 변경한다.
  3. 시스템이 이에 대한 결과를 가지고 액터에 응답한다.

# 유스케이스 작성 지침(4/7)

## □ 상호작용을 일관적이고 합리적으로 작성하라(2/2)

- 다음 5개의 예제는 어떤 식으로 작성해도 틀린 것은 아니지만, 3번째 방식을 선호함

예제1	1. 고객이 주문번호를 입력한다. 시스템은 그 번호가 이달의 당첨번호와 일치됨을 발견하고, 사용자와 주문번호를 이 달의 당첨자로 등록한 뒤, 판매 관리자에게 이메일을 보내고, 고객에게 축하 인사를 하고, 어떻게 경품을 받아가는지 안내한다.
예제2	1. 고객이 주문번호를 입력한다. 2. 시스템은 그 번호가 이달의 당첨번호와 일치됨을 발견하고, 사용자와 주문번호를 이 달의 당첨자로 등록한 뒤, 판매 관리자에게 이메일을 보내고, 고객에게 축하 인사를 하고, 어떻게 경품을 받아가는지 안내한다.
예제3	1. 고객이 주문번호를 입력한다. 2. 시스템은 그 번호가 이 달의 당첨번호와 일치됨을 발견한다. 3. 시스템은 사용자와 주문번호를 이 달의 당첨자로 등록한 뒤, 판매 관리자에게 이메일을 보내고, 고객에게 축하 인사를 하고, 어떻게 경품을 받아가는지 안내한다.
예제4	1. 고객의 주문번호를 입력한다. 2. 시스템은 그 번호가 이 달의 당첨번호와 일치됨을 발견한다. 3. 시스템은 사용자와 주문번호를 이 달의 당첨자로 등록한 뒤, 판매 관리자에게 이메일을 보낸다. 4. 시스템은 고객에게 축하인사를 하고, 어떻게 경품을 받아가는지 안내한다.
예제5	1. 고객이 주문번호를 입력한다. 2. 시스템은 그 번호가 이 달의 당첨번호와 일치됨을 발견한다. 3. 시스템은 사용자와 부문번호를 이 달의 당첨자로 등록한다. 4. 시스템은 판매관리자에게 이메일을 보낸다. 5. 시스템은 고객에게 축하 인사를 하고, 어떻게 경품을 받아가는지 안내한다.

# 유스케이스 작성 지침(5/7)

## □ 시기는 선택적으로 작성하라

- 대부분의 스텝은 이전 스텝의 다음에 바로 작성된다. 이렇게 뒤 따르는 행위 이외에 어떠한 시기에 관련된 행위가 유스케이스 시나리오의 흐름에 존재할 수 있다. 이러한 경우, 다음과 같이 작성하도록 함

예제)

스텝 3과 5사이에, 사용자는 ~ 한다.

사용자가 ~ 하자마자, 시스템은 ~ 한다.

## □ 관용구를 사용하라

- 관용구1) 사용자는 시스템 A가 시스템 B를 동작 시키도록 한다.
  - 이 경우는 대상 구축 시스템 A가 시스템 B와의 상호작용을 통하여 시나리오를 완성할 경우이다. 이럴 경우에는 다음과 같이 작성 하도록 함

예제)

사용자는 시스템으로 하여금 시스템 B로부터 원하는 데이터를 가져오게 한다.

- 관용구 2) 어떠한 조건이 이를 때까지 A~B 단계를 수행한다.

예제)

- 고객은 계좌번호, 이름, 주소를 제시한다.
- 시스템은 고객이 선호하는 물품 정보를 가져온다.
- 사용자는 구매할 품목을 선택하고, 구매하기 위해 표시한다.
- 시스템은 해당 품목을 고객의 '장바구니'에 추가한다.

고객은 쇼핑을 끝낼 때까지 스텝 3~4를 반복한다.

# 유스케이스 작성 지침(6/7)

## □ 다음과 같은 문장을 사용하여 일관성을 유지하라

예제)

~를 선택한다.	~를 요청한다.
~를 입력한다.	~를 검사한다.
~를 계산한다.	~를 수정한다.
~를 검색한다.	~를 삭제한다.
~를 보여준다.	

## □ 조건을 먼저 생각하라

- 대안흐름 작성시 가능한 모든 실패와 대안에 대하여 먼저 브레인스토밍 한 후에, 이에 대한 시스템의 대응 방법을 서술하는 것이 효과적이다. 조건은 스텝과 혼동되지 않게 다음과 같이 뒤에 "콜론:"을 두어서 작성하도록 함

예제)

비밀번호 오류:  
고객 무응답:  
현금 부족:  
고객 미등록:



# 유스케이스 작성 지침(7/7)

## □ 조건처리는 들어 쓰도록 하라

- 유스케이스의 가독성을 위하여 조건처리는 다음과 같이 들어 쓰도록 함

예제)

2a. 현금 부족:

2a1. 시스템이 고객에게 알리고, 금액을 새로 요구한다.

2a2. 고객이 금액을 새로 입력한다.

## □ 변동 목록을 이용하라

- 특정 흐름을 수행할 경우, 이를 수행하는 몇 가지의 다른 방법이 있을 경우에 변동목록을 이용할 수 있음
- 즉, 무엇이 일어나는가는 동일하지만, 어떻게 수행되는 지가 다양할 경우 이를 이용하도록 함
- 다음의 예제는 사용자 정보 및 계좌 정보를 입력하는 다양한 방식에 대한 변동 목록을 나타내고 있음

예제)

주요 흐름:

...

2. 사용자가 자신의 ID, 해당은행, 계좌번호를 입력한다.

...

변동 목록:

2a. 은행 카드, 안구 스캔, 지문 등을 이용한다.

# 유스케이스 작성 양식(1/7)

## □ 완전한 격식을 갖춘 양식

**제목 :** <제목은 반드시 짧은 동사구로 작성한 목표여야 한다.>

**이용 상황:** <목표를 더 길게 서술한 문장으로, 필요한 경우, 목표의 통상적인 발생조건>

**범위:** <설계 범위로, 설계 시 블랙-박스로 간주되고 있는 시스템>

**수준:** <요약, 사용자-목표, 하위기능 중 하나>

**일차 액터:** <일차 액터에 대한 역할 이름 혹은 설명>

**이해관계자와 이해관계:** <유스케이스의 이해관계자들과 주요 이해관계 목록>

**선조건:** <우리가 예상하는 이미 갖춰진 현실의 상태>

**최소 보증:** <모든 경우의 종료 시, 이해관계가 어떻게 보호되는가>

**성공 보증:** <목표가 달성되었을 때 현실의 상태>

**트리거:** <유스케이스를 시작하는 것으로, 시간 이벤트일 수 있다.>

**주요 성공 시나리오:**

<여기에 트리거로부터 목표 달성 그리고 그 후의 정리작업에 이르기까지 시나리오의 단계들을 기록한다.>

<단계 #><행동서술>

**확장:**

<여기에 한 번에 한가지씩, 각각 주요 시나리오의 단계를 참조하며 확장내용을 기록한다.>

<변경되는 단계><조건>:<행동이나 하위 유스케이스>

**기술과 데이터 변동 목록:**

<여기에 시나리오에서 분기점을 만드는 변동사항들을 기록한다.>

<단계 또는 변동 번호><변동 목록>

**관련된 정보:**

<프로젝트에 필요한 부가적인 정보는 무엇이든 기록한다.>

## □ 간결한 양식(예문)

일차 액터: 사용자

범위: 애플리케이션

수준: 하위기능

당사자임을 확인하기 위해, 사용자는 사용자 이름과 비밀번호를 입력하도록 요청 받는다. 시스템은 그 사용자 이름에 해당하는 가입자가 존재하는지, 그 가입자에 대해 비밀번호가 일치하는지를 검증한다. 그런 후에 사용자는 제출자를 위한 모든 명령에 접근할 수 있다.

그 사용자 이름이 관리자로 지정된 사용자와 일치하면, 사용자는 모든 사용자 명령문과 관리자 명령문에 접근이 허용된다. 만약 사용자 이름이 존재하지 않거나 비밀번호가 사용자 이름과 맞지 않으면, 사용자는 거부된다.

# 유스케이스 작성 양식(3/7)

## □ 한 개의 열을 갖는 표

유스케이스 번호	<제목은 짧은 동사구로 이루어진 목표이다>	
이용 상황	<이용 상황을 더 길게 서술한 문장으로, 필요한 경우에 서술>	
범위	<설계시 블랙박스로 간주되고 있는 시스템>	
수준	<요약, 일차 과업, 하위기능 중 하나>	
일차 액터	<일차 액터에 대한 역할 이름 혹은 설명>	
이해관계자와 이해관계	이해관계자	이해관계
	<이해관계자이름>	<여기에 이해관계자의 이해관계를 적는다.>
선조건	<우리가 이미 예상하는 현실의 상태>	
최소보증	<어떤 종료 시에도 보호되는 이해관계>	
성공보증	<성공적인 종료 시에 충족되는 이해관계>	

트리거	<제목은 짧은 동사구로 이루어진 목표이다>	
서술	단계	행동
	1	<여기에 트리거로부터 목표 달성 그리고 그 후의 정리작업에 이르기까지 시나리오의 단계를 기록한다.>
	2	<...>
	3	
확장	단계	분기 행동
	1a	<분기를 일으키는 조건>: <행동이나 하위 유스케이스의 제목>
기술과 데이터 변동	변동목록	

# 유스케이스 작성 양식(4/7)

## □ 두 개의 열을 갖는 표

고객	시스템
주문번호를 입력한다.	
	주문번호가 이 달의 당첨번호와 일치됨을 발견한다.
	사용자와 주문번호를 이 달의 당첨자로 등록한다.
	판매관리자에게 이메일을 보낸다.
	고객에게 축하인사를 하고, 어떻게 경품을 받아가는지 안내한다.
시스템을 종료한다.	

# 유스케이스 작성 양식(5/7)

## □ RUP 양식

### 1. 유스케이스 이름

#### 1.1. 요약 서술

... 본문 ...

#### 1.2. 액터

... 본문 ...

#### 1.3. 트리거

... 본문 ...

### 2. 이벤트의 흐름

#### 2.1. 기본흐름

... 본문 ...

#### 2.2. 대안 흐름

##### 2.2.1. 조건1

... 본문 ...

##### 2.2.2. 조건2

... 본문

### 3. 특별 요구사항

#### 3.1 플랫폼

... 본문 ...

#### 3.2 ...

### 4. 선조건

... 본문 ...

### 5. 후조건

... 본문 ...

### 6. 확장 지점

... 본문 ...

# 유스케이스 작성 양식(6/7)

## □ 템플릿 사례

### 1. 유스케이스 개요

유스케이스에 대한 간단한 설명과 가능하면 유스케이스를 구분할 수 있는 식별자를 부여한다.

### 2. 관계

#### 2.1 액터

– 액터의 이름과 간단한 설명을 한다. 이는 주 액터, 지원 액터, 비공식 액터로 구분하여 작성할 수 있다.

#### 2.2 선조건

– 유스케이스를 실행할 경우에 반드시 만족해야 하는 조건에 대하여 기술한다.

#### 2.3 후조건

– 유스케이스 종료 후 반드시 이루어야 하는 조건과 유스케이스 시나리오가 실패하였을 경우에도 만족해야 할 최소 조건에 대하여 기술한다.

### 3. 처리 흐름

#### 3.1 기본 흐름

#### 3.2 확장 흐름

### 4. 특별 요구사항

– 유스케이스에서 고려해야 하는 기능적 요구사항 이외의 사항들에 대하여 설명한다. 예를 들면, 해당 유스케이스에서 처리해야 할 성능이나 그 외의 비기능적인 요구사항에 대하여 언급할 수 있다.

### 5. 변동목록

– ‘유스케이스 기술 지침’ 의 ‘변동 목록’ 을 참조하여라.

### 6. 기타사항

– 부가적으로 설명해야 할 사항에 대하여 기술한다.

## □ 번호 매기기

기본 흐름

1.

2.

1-2 스텝을 사용자가 ~ 할 때까지 반복한다.

3.

4.

..

대안 흐름

\*a. <이는 기본 흐름 중 언제든지 발생할 수 있는 흐름에 대하여 작성한다.>

\*a1.

\*b.

2a. 조건1:

2a1.

2a2.

2b. 조건2:

2b1.

2b1a.

2b1b.

.1

.2

2b2.

3-5a. 조건1: <이는 기본 흐름 3-5에서 발생할 수 있는 흐름에 대하여 작성한다.>

3-5a1



## □ 패키징 방법

- 유스케이스의 숫자가 많을 경우에는 이를 패키지로 묶어서 관리하는 것이 효과적임
- 유스케이스를 패키지화 시키는 방법은 여러 가지가 존재할 수 있으나, 가장 권장되는 방법은 업무 영역별로 유스케이스를 패키지화하고 이를 팀별로 배포하는 것
- 예제)
  - 어떤 프로젝트가 고객 정보, 기획, 구매, 광고 영역을 가진다고 하면, 이를 패키지화하고 해당하는 유스케이스를 위치하도록 함

## □ 패키지 규모

- 얼마나 많은 단계의 패키지를 사용할 것인지를 결정해야 함
- 경험적으로 보면, 각 유스케이스 패키지는 대략 3개에서 10개의 더 작은 단위(유스케이스, 액터, 다른 패키지)를 포함
- 개략적인 가이드라인 - 권고
  - 0-15: 유스케이스 패키지가 필요 없음
  - 10-50: 한 단계의 유스케이스 패키지를 사용
  - 25이상: 두 단계 이상의 유스케이스 패키지를 사용

## 전통적인 실수(1/4)

### □ 잘못 작성된 유스케이스

- 액터가 없는 유스케이스, 시스템이 없는 유스케이스
- 사용자 인터페이스의 내용을 너무 상세히 담고 있는 유스케이스
- 너무 낮은 수준의 유스케이스
- 기술 용어를 언어를 사용하는 유스케이스
- 유스케이스의 목적과 내용이 서로 다른 유스케이스

### □ 잘못 작성된 유스케이스 예제(1/4)

- 시스템이 없는 유스케이스: 현금 인출

#### 수정 전:

1. 고객이 카드를 넣고 비밀번호를 입력한다.
2. 고객이 '인출'을 선택하고 금액을 입력한다.
3. 고객이 현금, 카드, 영수증을 받는다.
4. 고객이 떠난다.



#### 수정 후:

1. 고객이 인출카드를 카드 판독기에 통과시킨다.
2. 현금인출기가 은행 ID, 계좌번호, 암호화된 비밀번호를 카드로부터 읽어들이고, 은행 ID와 계좌번호를 주요 은행 시스템을 통해 검증한다.
3. 고객이 비밀번호를 입력한다. 현금 인출기는 그것을 카드로부터 읽은 암호화된 비밀번호와 비교하여 검증한다.
4. 고객은 '빠른 현금' 메뉴와 5달러의 배수로 인출금액을 선택한다.
5. 현금인출기는 주요 은행 시스템에 고객의 계좌번호와 인출된 금액을 알리고, 승인과 새로운 잔고 정보를 받는다.
6. 현금인출기가 현금, 카드, 새로운 잔고를 보여주는 영수증을 지급한다.
7. 현금 인출기가 거래를 로그에 남긴다.

## 전통적인 실수(2/4)

### □ 잘못 작성된 유스케이스 예제(2/4)

#### ■ 액터가 없는 유스케이스: 현금 인출

수정 전:

1. 신용카드와 비밀번호를 입력 받는다.
2. 거래 유형으로 '인출' 을 입력 받는다.
3. 원하는 금액을 입력 받는다.
4. 계좌에 충분한 잔액이 있는지 검증한다.
5. 돈, 영수증, 카드를 내어준다.
6. 화면을 초기상태로 설정한다.

수정 후:

'시스템이 없는 유스케이스'의 수정 후 예제와 동일하다.

#### ■ UI 세부사항이 너무 많은 유스케이스: 물품 구매

수정 전:

1. 시스템이 ID와 비밀번호 입력화면을 보인다.
2. 고객이 ID와 비밀번호를 시스템에 입력하고, '확인' 을 클릭한다.
3. 시스템이 사용자 ID와 비밀번호를 검증하고 개인 정보 화면을 보여준다.
4. 고객이 이름, 주소, 시, 도, 우편번호, 전화번호를 입력하고 '확인' 을 클릭한다.
5. 시스템이 사용자가 기존 사용자임을 검증한다.
6. 시스템이 이용 가능한 제품 목록을 제시한다.
7. 고객이 구매할 물건의 사진을 클릭하고, 옆에 수량을 입력하고, 일을 마치면 '완료' 를 클릭한다.
8. 시스템이 창고 보관 시스템을 통해 요청된 물건의 재고 수량이 충분한지 확인한다.

...

수정 후:

1. 고객이 ID와 비밀번호로 시스템에 접속한다.
2. 시스템이 사용자를 검증한다.
3. 고객이 이름, 주소, 전화번호를 입력한다.
4. 시스템은 고객이 기존의 고객임을 검증한다.
5. 고객이 제품과 수량을 결정한다.
6. 시스템이 창고 보관 시스템을 통해 요청된 물건이 재고 수량이 충분한지 확인한다.
- ...

# 전통적인 실수(3/4)

## □ 잘못 작성된 유스케이스 예제(3/4)

### ▪ 낮은 목표수준을 갖는 유스케이스: 물품 구매

#### 수정 전:

1. 사용자가 ID와 비밀번호로 시스템에 접속한다.
2. 시스템이 사용자를 검증한다.
3. 사용자가 이름을 입력한다.
4. 사용자가 주소를 입력한다.
5. 사용자가 전화번호를 입력한다.
6. 사용자가 제품을 선택한다.
7. 사용자가 수량을 결정한다.
8. 시스템이 사용자가 기존의 고객임을 검증한다.
9. 시스템이 제품 보관 시스템과 연결을 설정한다.
10. 시스템이 제품 보관 시스템에 현재의 재고 수준을 요청한다.
11. 창고 보관 시스템이 현재의 재고 수준을 알려온다.
12. 시스템이 요청된 수량이 재고에 있는지를 검증한다.
- ...



#### 수정 후:

1. 사용자가 ID와 비밀번호로 시스템에 접속한다.
2. 시스템이 사용자를 검증한다.
3. 사용자가 개인정보(이름, 주소, 전화번호)를 제공하고, 제품과 수량을 결정한다.
4. 시스템은 사용자가 기존의 고객임을 검증한다.
5. 시스템이 창고 보관 시스템을 통해 요청된 물건이 재고 수량이 충분한지 확인한다.

## 전통적인 실수(4/4)

### □ 잘못 작성된 유스케이스 예제(4/4)

- 목적과 내용이 서로 다른 유스케이스: 로그인

수정 전:

1. 사용자가 어플리케이션을 시작할 때 유스케이스가 시작된다.
2. 시스템은 로그인 화면을 보여준다.
3. 사용자는 사용자 이름과 비밀번호를 입력한다.
4. 시스템은 정보를 검증한다.
5. 시스템은 해당 화면을 보여준다.
6. 사용자는 주문하기를 선택한다.
7. 시스템은 주문에 필요한 정보를 보여준다.
- ...



이는 유스케이스의 목적과 내용이 서로 상이하다. 이는 '물품 주문하기' 유스케이스를 작성하여 물품 주문하기의 흐름을 서술하고, 로그인을 선조건에 명시하는 정도로 서술하면 될 것이다.

## □ 간단한 워크샵 수행

### 1. 액터 정의

- 누가 또는 어떤 것이 시스템을 사용할 지 식별함
  - ✓ 시스템을 사용할 실제 사람과 함께 초기에 워크샵을 시작
  - ✓ 각 사용자가 시스템을 사용할 때의 역할을 식별
- 액터를 식별할 때, 각 액터에 대해 간단한 설명이 있는지 확인함
  - ✓ 보통은 시스템과 관련하여 수행하는 액터의 역할과 액터가 시스템으로부터 필요한 것을 결정할 때, 도움을 줄 액터의 책임을 파악하도록 함
- 액터를 정의할 때, 시스템이 상호작용하는 다른 시스템을 항상 염두에 둬
  - ✓ 액터에 대한 아이콘은 사람으로 오해할 수 있지만, 액터의 개념은 상호작용하는 시스템을 포함한다는 것을 명심 하도록 함
  - ✓ 우선 '사람' 액터를 찾는데 중점을 두어야 함

#### <액터 정의시 유의사항>

1. 유스케이스 모델의 구조나 액터간의 관계에 대해서는 무시하여도 좋음
2. 단순히 시스템을 사용하는 사람 또는 사물을 파악 하도록 함
3. 많은 액터를 찾는데 중점을 두고, 액터들을 정제 시키는 것에 관하여도 너무 시간을 투자하지 않는 것이 좋음
4. 일반적으로 액터를 정의하는데 1~4시간 정도가 할당됨

### 2. 유스케이스 정의

- 액터의 목표에 의거한 유스케이스를 식별하고 상세화 함

## 유스케이스 워크샵(2/5)

### □ 워크샵 구성

- 유스케이스 워크샵은 브레인스토밍 회의방식
- 서로 다른 배경, 지식, 경험을 가진 사람들을 포함하는 그룹으로 구성함
- 워크샵 구성시에 소규모의 그룹으로 구성함
  - 일반적으로 5~6 명 이상의 규모는 오히려 회의를 방해하여 효율적인 결과를 이끌어 낼 수 없음
- 일반적인 구성원은 개발팀에서의 절반과 사용자 대표 그룹에서의 나머지 절반으로 구성되고, 이들 중간에는 보조자가 존재함
- 보조자는 중재자 역할(모든 아이디어와 요청에 대한 촉매 역할을 하는 사람)을 수행함

### □ 워크샵 도구

- 화이트 보드
- 보드마카(빨간색, 파란색, 검은색)
- 포스트 잇
- 디지털 카메라
- 녹음기
- 노트와 필기도구 등

## 유스케이스 워크샵(3/5)

### □ 워크샵 원칙

- 정시에 미팅을 시작하고 종료한다.
- 정해진 휴식 시간 후에 즉시 워크샵에 복귀하도록 한다.
- 한 번에 한 주제에 대해서만 진행한다.
- 주제에 대한 토론시간을 제한한다.
- 모든 사람이 회의에 자발적으로 참여한다.
- 서로 존중하고 개인이 아닌 회의 주제에 대해 언급하고 비판한다.

#### 주의!!!!!!

이와 같은 원칙에 워크샵 참석자들  
의 동의를 사전에 얻는 것이  
매우 중요하다.



## 유스케이스 워크샵(4/5)

### □ 워크샵 시간

- 평균적으로 유스케이스를 작성할 때에는 유스케이스 당 2-3시간 정도를 할당하여 워크샵을 진행할 수 있도록 계획을 작성함
- 워크샵 초기에 걸리는 시간보다 후반부로 갈수록 참여자들이 워크샵에 능숙해 지므로 시간이 줄어들 수 있음
- 시간 분배에 대한 계획을 세워 놓지 않는다면, 특정 주제에 대하여 너무 세부적으로 토의 하는 시간이 길어질 수 있어서, 다른 주제에 대한 토의를 적절히 진행하지 못할 가능성이 농후함

# 유스케이스 워크샵(5/5)

## □ 역할

- 회의 진행자
  - 전반적으로 회의를 주도하고, 화이트 보드에 유스케이스를 파악해서 얻어진 간단한 다이어그램들을 작성함
  - 처음에는 경험이 많은 컨설턴트가 본 역할을 수행하고, 점차적으로 워크샵에 참석한 구성원이 돌아가면서 본 역할을 맡을 수 있음
- 클래스 담당자
  - 워크샵의 해당 세션에서 다루는 클래스에 대한 모든 정보를 기록한다.
  - 본 역할은 두 세명 정도가 동시에 맡아서 수행하는 것이 바람직함
- 기록원
  - 화이트 보드에 작성되어 있는 다이어그램들이나 그 외의 주요한 사항들에 대하여 기록함
  - 디지털 카메라를 이용하여 화이트 보드의 내용을 찍어두는 것도 좋은 방법임

워크샵 진행 도중 위와 같은 책임을 가진 담당자들이 존재해야 한다. 모든 사람이 이러한 역할을 돌아가면서 맡아 진행하는 것도 좋은 방법이다.

## 2. 유스케이스 패턴

ONE STEP AHEAD

- ☐ 개요
- ☐ Business Rule
- ☐ Commonality
- ☐ Concrete Extension or Inclusion
- ☐ CRUD
- ☐ Multiple Actors
- ☐ Optional Service
- ☐ Orthogonal Views

## □ 유스케이스 모델러의 경험

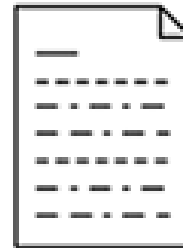
- 좋은 유스케이스 모델을 개발하는 방법에 대한 지식
- 유사한 시스템 경험을 바탕으로 간단하고 손쉽게 새로운 유스케이스 식별

## □ 경험의 재사용

- 문제의 이해, 즉 시스템 사용이나 디자인의 기초가 될 수 있는 해결책 마련
- 패턴을 적용할 때 상황에 대한 정보를 포함하고 예측되는 결과를 묘사

# Business Rule (1/2)

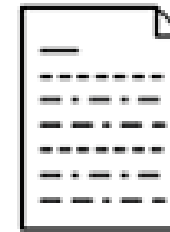
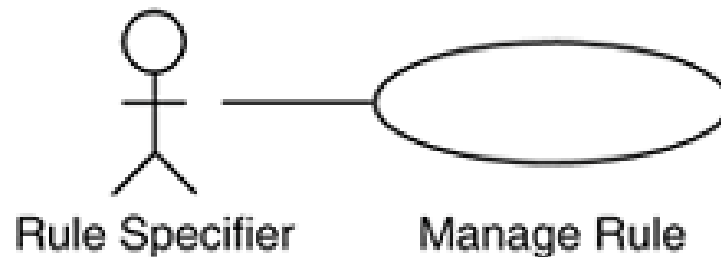
- 업무의 정책, 규칙, 표준으로부터 시작하여 정보를 추출하고 이를 기술
- Static Definition
  - 여러 유스케이스 내에 포함되어 있는 조직내의 업무 규칙
  - 유스케이스 모델의 구조에 영향이 없음



## Business Rule (2/2)

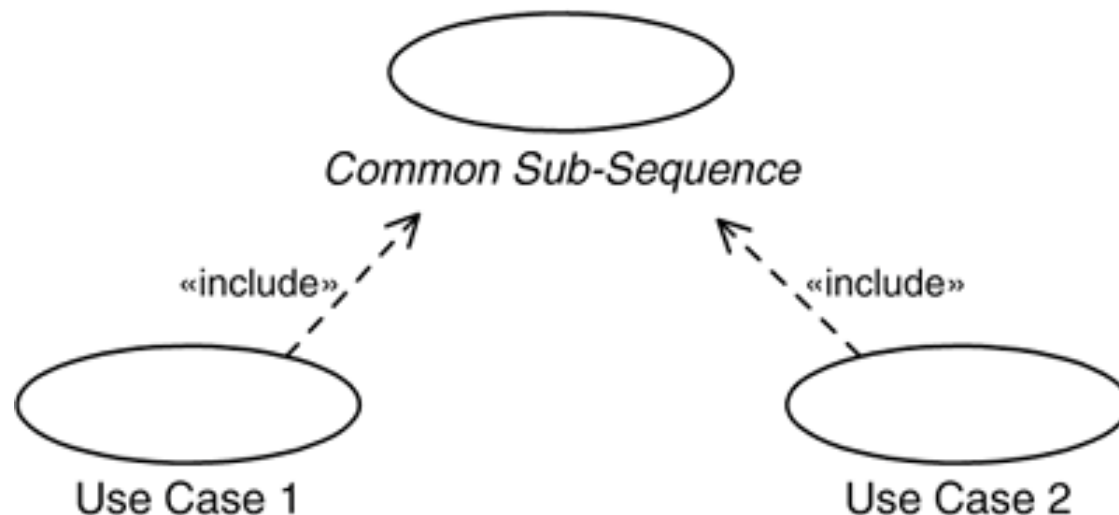
### □ Dynamic Modification

- 룰 관리를 위한 하나의 유스케이스를 포함
- 비즈니스 룰에 대한 생성, 수정 및 삭제 수행
- Manage Rule 이라고 부름



# Commonality (1/4)

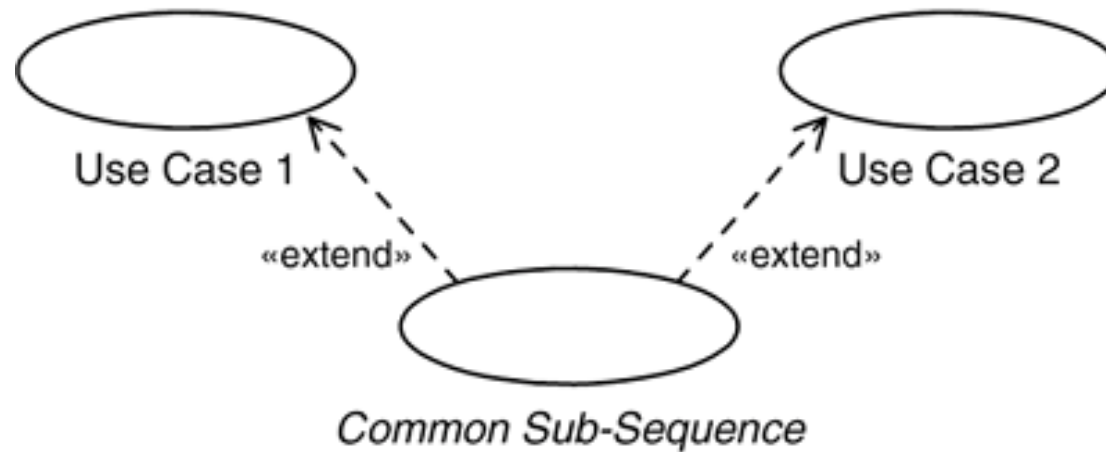
- 유스케이스 흐름의 여러 곳에 나타나는 흐름을 추출하고 분리하여 표현
- Reuse
  - 여러 유스케이스 내에 나타나는 활동의 순서를 모델화
  - Common Sub-Sequence 라고 부르며 다른 유스케이스는 이 유스케이스를 공유



# Commonality (2/4)

## □ Addition

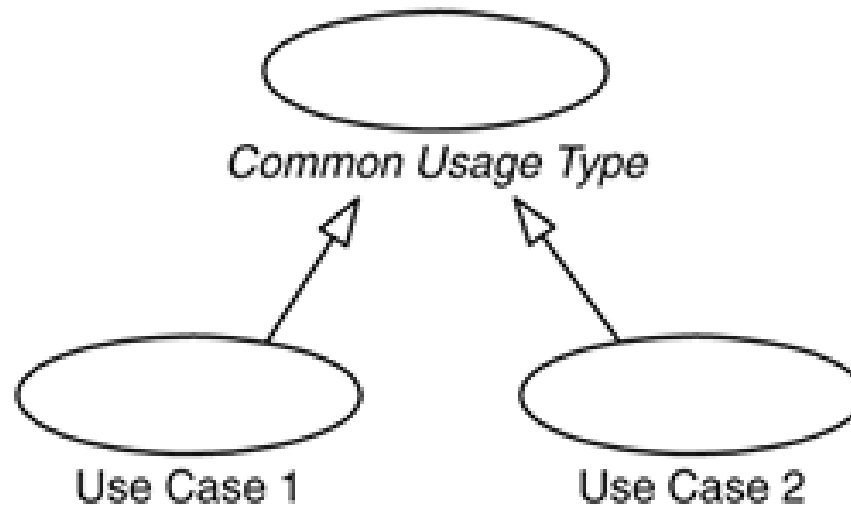
- 활동의 순서를 공유하는 다른 유스케이스로 확장
- 대안흐름에 대한 패턴





## □ Specialization

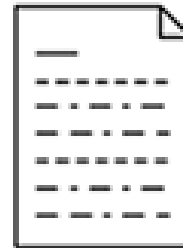
- 동일한 유스케이스 시나리오를 포함
- Common Usage Type 이라고 부르며 시나리오 흐름을 특수화 시켜 모델링
- 모든 유스케이스는 Common Usage Type 유스케이스를 상속받은 유스케이스로 흐름을 추가하거나 특정 흐름을 분화시킴



# Commonality (4/4)

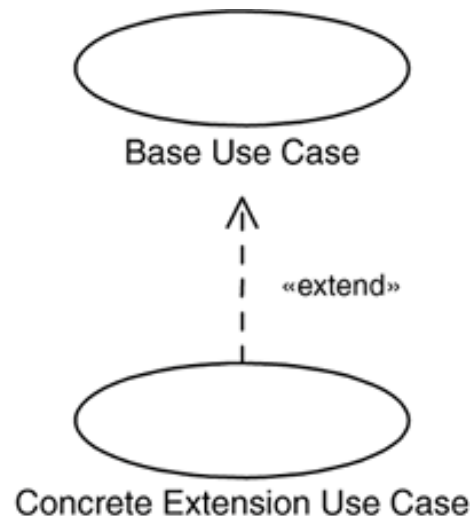
## □ Internal Reuse

- 한 유스케이스 내에서 다양한 곳에서 같은 시나리오 흐름을 사용
- 흐름을 추출하여 유스케이스와 분리할 필요가 없고 유스케이스 내에서 흐름을 분리하여 별도로 명세



# Concrete Extension or Inclusion (1/2)

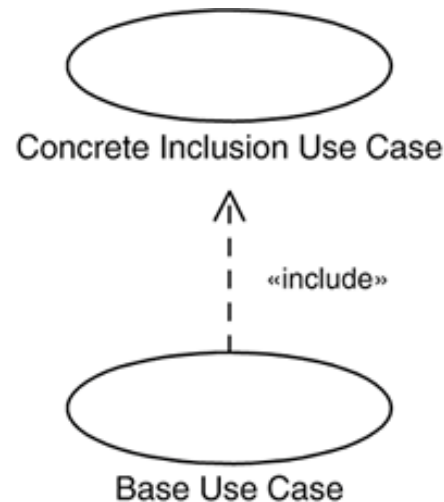
- 동일한 흐름의 있는 유스케이스를 분리하여 완전한 유스케이스를 모델링
- Extension
  - 확장은 구체적이고 기본 유스케이스와 같이 인스턴스화 됨



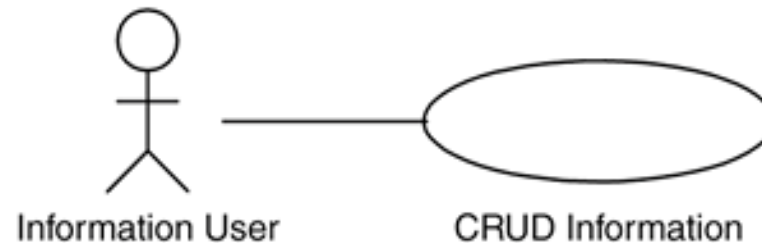
# Concrete Extension or Inclusion (2/2)

## □ Inclusion

- 기본 유스케이스로 부터 포함 유스케이스까지 포함관계를 표현
- 포함 유스케이스도 인스턴스화 됨

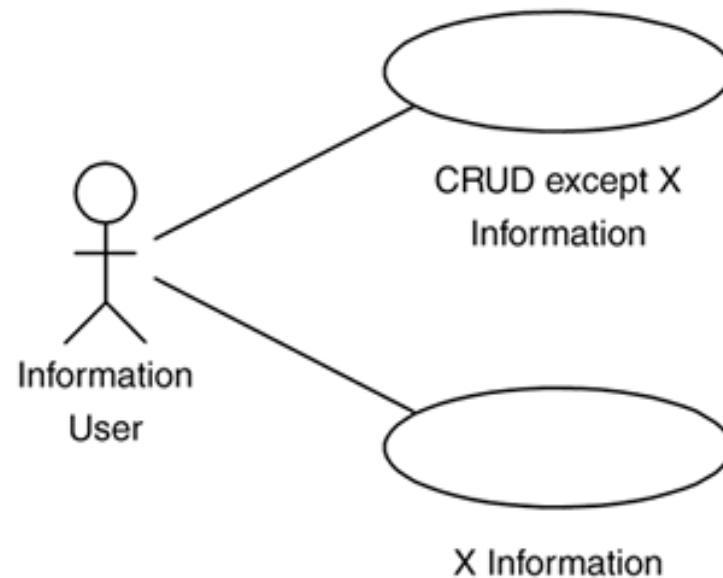


- ❑ 정보의 CRUD와 같은 단순한 작업을 짧고 간결한 유스케이스로 병합
- ❑ Complete
  - CRUD Information (Manage Information) 이라고 불림
  - 정보에 대해 CRUD와 같은 서로 다른 오퍼레이션을 모델링



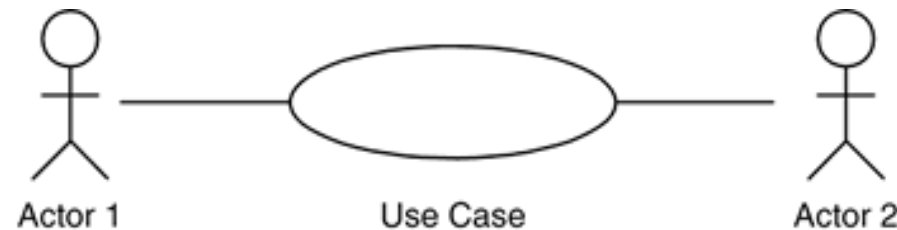
## □ Partial

- 유스케이스 대안흐름 중 하나를 분리하여 모델



# Multiple Actors (1/2)

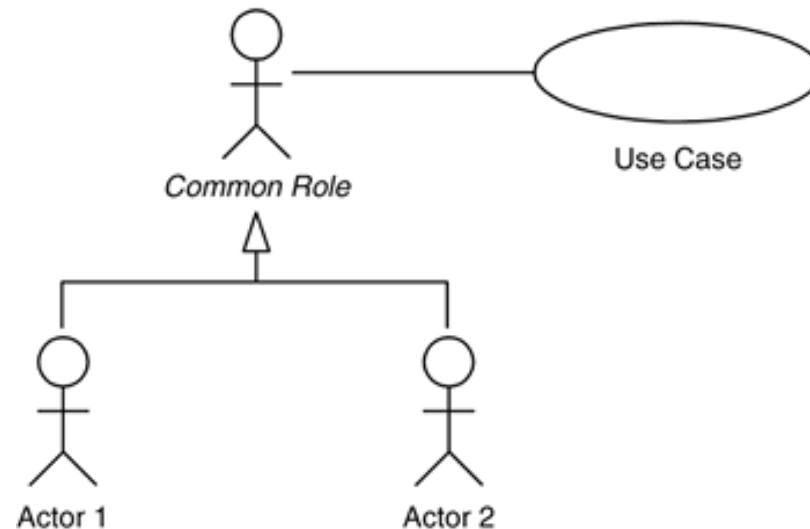
- 개별 역할을 분리하여 유지하는 동안 액터 사이의 공통사항 찾아냄
- **Distinct Roles**
  - 한 유스케이스에 두 액터로 구성
  - 액터 사이에 역할이 명확히 구분됨



# Multiple Actors (2/2)

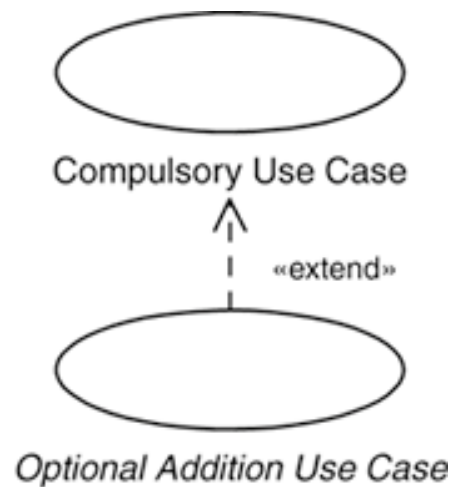
## □ Common Role

- 한 유스케이스에 같은 역할을 수행하는 두 액터로 구성
- 액터들은 공통된 역할을 공유하고 상속



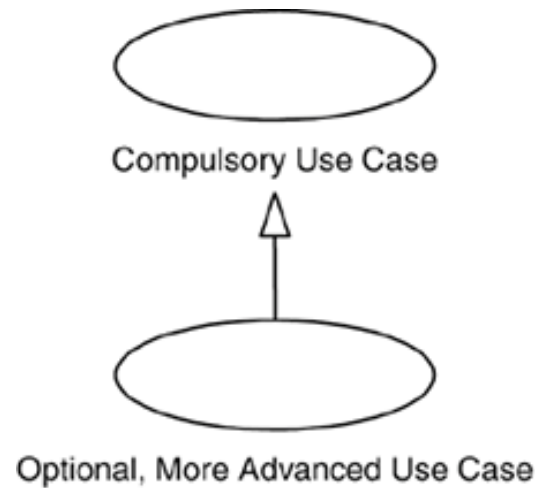


- 유스케이스에서 필수적인 부분과 선택적인 부분을 분리
- Addition
  - 확장 관계로 표현
  - 시스템의 필수적으로 사용되는 유스케이스에서 선택적인 추가 유스케이스를 포함관계로 모델링



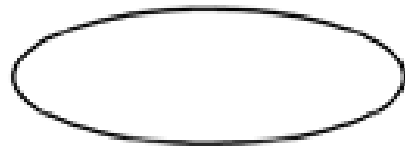
## □ Specialization

- 선택적인 부분이 포함된 유스케이스로 특수화 됨

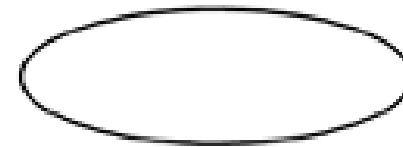


## □ Independent

- 선택적인 부분을 필수적인 유스케이스로부터 분리하고 관계가 없는 별개의 유스케이스로 모델링



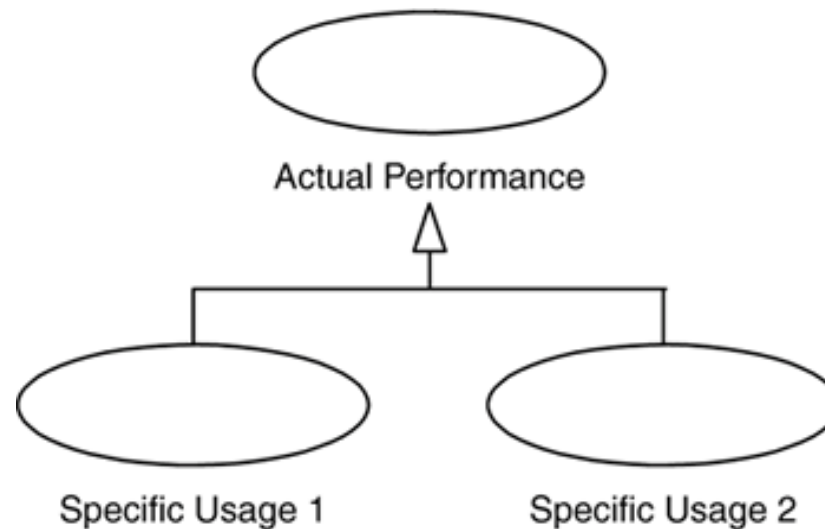
Compulsory Use Case



Optional Use Case

# Orthogonal Views (1/3)

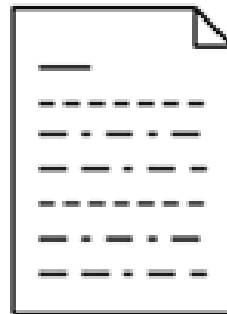
- 서로 다른 이해관계자에게 다르게 인식되는 시스템 흐름을 다른 뷰 제공
- Specialization
  - 상위 유스케이스는 시스템 내에서 수행되는 것을 모델링
  - 하위 유스케이스는 사용자에게 인식되는 특수화된 유스케이스로 모델링



# Orthogonal Views (2/3)

## □ Description

- 특별한 행위에 대해 단일 유스케이스의 분리된 영역에서 명세



# Orthogonal Views (3/3)

## □ 사용 예

