UML과 소프트웨어개발

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

과정 개요

❑ 교육 목표 & 특징

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

강의 요구 기술

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

교육 일정표

❑ 교육은 매 회 4 시간씩 총 5회에 걸쳐 진행합니다.

1 일차	2 일차	3 일차	4 일차	5 일차
- UML 개요 UML 소개 UML 다이어그램분류	- 유스케이스 유스케이스 개요 유스케이스 다이어그램 유스케이스 명세	- 분석모델 I 개념모델 구조 다이어그램 (클래스, 객체 ...)	- 분석모델II 컴포넌트 식별 인터페이스 식별 행위 다이어그램 (액티비티, 시퀀스 ...)	- 설계모델 컴포넌트 동적 설계 컴포넌트 내부 설계

2일차 – 유스케이스

ONE STEP AHEAD

1. 유스케이스 개요
2. 유스케이스 다이어그램
3. 유스케이스 명세

1. 유스케이스 개요

- ❑ 소프트웨어 개발과정
- ❑ 사용자 요구사항
- ❑ 유스케이스 개요
- ❑ 유스케이스와 시나리오
- ❑ 유스케이스 내용

소프트웨어 개발과정

- **요구사항 정의(Requirements) <- Our Focus**
  - 기능적, 비 기능적 요구사항의 집합으로 시스템 비전을 기술
- **분석/설계(Analysis & Design)**
  - 구현 단계에서 시스템이 어떻게 실체화 될지를 기술
- **구현(Implementation)**
  - 실행 가능한 시스템이 될 코드를 생성
- **테스트(Test)**
  - 전체 시스템을 검증
- **배포(Deployment)**
  - 시스템을 배포하고 사용자 교육

사용자 요구사항(1/2)

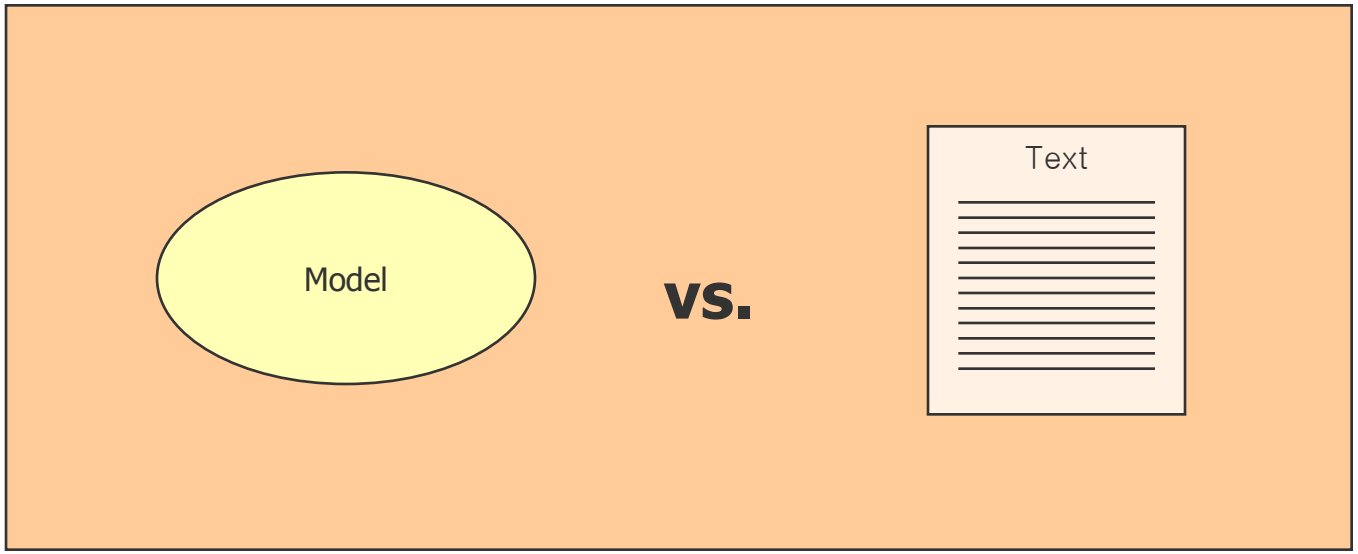
- **정의**
  - 시스템이 갖추어야 할 능력과 조건
  - 유스케이스는 그 자체로 요구사항
    - 시스템이 해야 하는 것에 대한 상세하고 정확하게 표현
    - 시스템의 기능적인 요구사항
  - 유스케이스가 요구사항의 전부는 아님
    - 외부 인터페이스, 데이터포맷, 비즈니스 규칙, 그리고 복잡한 공식과 같은 것은 자세히 다루지 않음
    - 요구사항 중 기능, 즉 행위관련 부분
- **요구사항 관리(Requirement Management)**
  - 최상의 실행 지침(Best Practice)
  - 변화하는 요구사항을 찾아내고, 조직화, 문서화, 추적하기 위한 체계적인 접근 방법

사용자 요구사항(2/2)

- 요구사항 타입
  - FURPS+
    - 기능성(Functionality), 유용성(Usability), 신뢰성(Reliability), 성능(Performance), 지원성(Supportability)
    - 구현(Implementation), 인터페이스(Interface), 운영(Operations), 패키징(Packaging), 법률요건(Legal)
  - 기능적/비 기능적 요구사항
- 유스케이스
  - 기능적인 요구사항 획득 및 서술

유스케이스 개요(1/4)

❑ 들어가기



유스케이스

유스케이스 개요(2/4)

- 역사
  - 1960년대 후반 야곱슨(Ivar jacobson)에 의해 개념이 수립
  - 1980년대 후반부터 요구사항 정립의 수단으로 각광을 받기 시작
  - 1990년대 유스케이스가 UML에 핵심적인 개념으로 추가
- 소개
  - 시나리오의 집합
    - 액터와 시스템간의 상호작용을 텍스트로 작성
    - 이해관계자들 중에서 일차 액터로 불리는 행위자의 요청에 대한 시스템의 응답
    - 플로우차트, 시퀀스차트, 프로그래밍 언어 등으로 작성될 수 있지만 기본적으로는 텍스트
  - 대상 시스템의 범위를 정하는 것을 도움
  - 고객과의 의사소통을 위한 도구

유스케이스 개요(3/4)

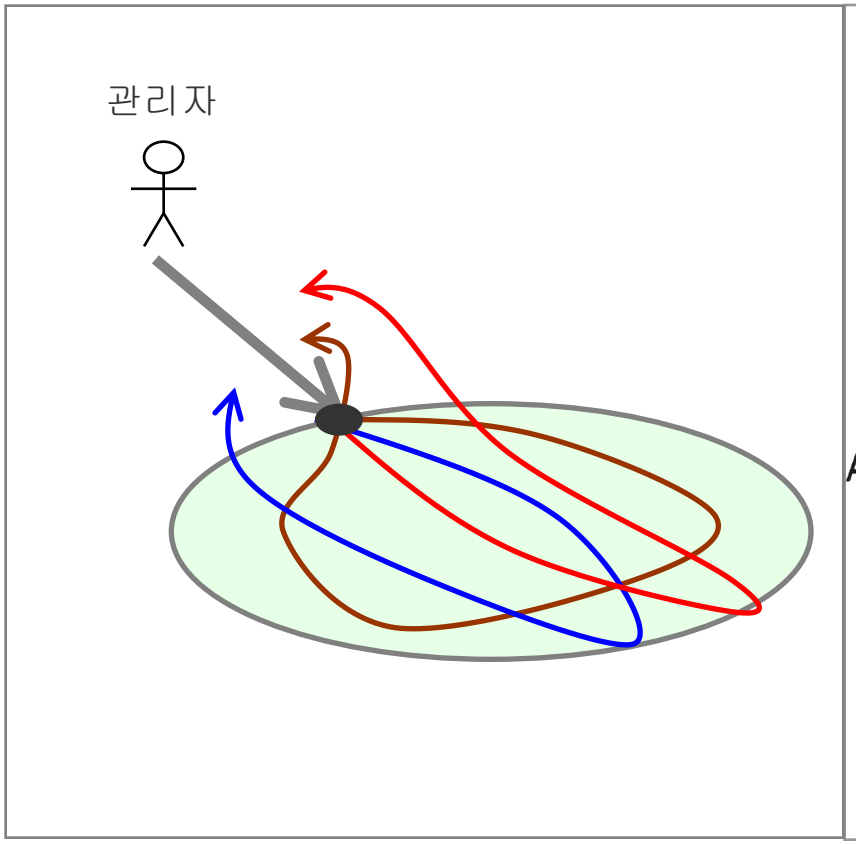
- **요구사항 관리**
  - 시스템의 기능적 요구사항(functional requirements) 획득 기법
  - 사용자와 시스템간의 상호작용 서술
  - 고객과의 의사소통 도구
- **시나리오(scenario)**
  - 사용자와 시스템 간의 상호작용을 서술하는 일련의 스텝들
  - 발생 가능한 시퀀스 중 하나
- **유스케이스(use case)**
  - 공통적인 사용자 목표에 의해 묶인 시나리오들의 집합
  - 기본적으로 텍스트
  - 유스케이스의 핵심은 사용자 목표(user goal)

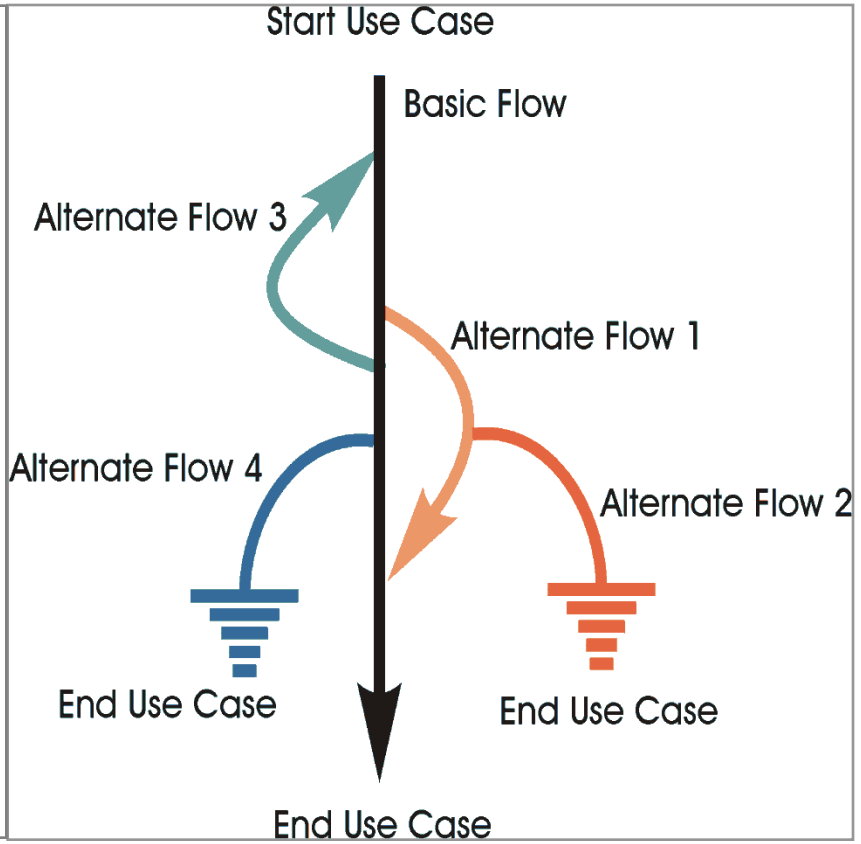
유스케이스 개요(4/4)

- **액터(actor)**
  - 사용자가 시스템에 대해서 수행하는 역할(role)
  - 여러 사람이 하나의 액터로, 하나의 사람이 여러 액터로 맵핑 가능
  - 액터가 반드시 사람일 필요는 없음
- **UML에서의 정의는 미약**
  - 유스케이스 다이어그램 표기법만을 정의
- **유스케이스의 가치**
  - 다이어그램이 아닌 내용(Content)
- **활용**
  - 요구사항을 도출하는 수단
  - 기능적 요구사항 획득
  - 근래의 객체지향 방법론이나 컴포넌트 기반 개발 방법론들은 유스케이스를 적극 사용

유스케이스와 시나리오

- 하나의 유스케이스는 여러 개의 가능한 시나리오를 내포함
- 유스케이스의 결과 성공일 수도 있고 실패일 수도 있음





유스케이스 내용 – 주 성공 시나리오

❑ 개요

- 기본 흐름은 모든 것이 제대로 작동하는 것처럼 작성: 주 성공 시나리오
- 기본 흐름은 분기나 대안 흐름이 없는 단순한 형태의 문장
- 스텝 작성 시 유의사항
  - 누가 스텝을 수행하는지 명확하게 표시
  - 사용자 인터페이스를 서술해서는 안됨
  - 액터가 수행하는 절차가 아니라 액터의 의도(intention)를 표시

Main Success Scenarios

1. 유스케이스는 고객이 주문하기를 선택하면 시작된다.
2. 고객은 이름과 주소를 입력한다.
3. 고객이 제품 코드를 입력하는 동안
   a) 시스템은 제품 설명과 가격을 제공한다.
   b) 시스템은 합계에 아이템의 가격을 추가한다.

End loop

4. 고객이 신용카드 지불정보를 입력한다.
5. 고객이 보내기를 선택한다.
6. 시스템이 그 정보를 확인하고, 주문을 저장하고, 지불 정보를 회계 시스템에 전달한다.
7. 지불이 확인되면, 주문은 확정된 것으로 표시되고, 주문 번호가 고객에게 전달되고 유스케이스는 종료된다.

유스케이스 내용 – 확장(Extensions)

❑ 개요

- 기본 흐름에 대한 대안을 제시
- 실패, 예외, 에러도 확장 흐름으로 문서화
- 시스템이 서로 다른 행위를 하게 하는 조건
- 확장 흐름은 언제든지 일어날 수 있는 일을 보여줌
  - 정상적인 이벤트의 흐름을 방해하는 것
- 주 성공 시나리오의 변동(variants)
- 확장은 성공할 수도 있고 실패할 수도 있음

Extensions.

1*. 보내기를 선택하기 전 언제든지, 고객은 취소를 선택할 수 있다. 주문은 저장되지 않고 유스케이스는 종료된다.

6a. 정보가 올바르지 않을 경우:

6a1. 시스템은 고객에게 정보를 수정하도록 한다.

7a. 지불이 확인되지 않을 경우:

7a1. 시스템은 고객에게 지불 정보를 수정하거나 취소하도록 한다.

.1 고객이 정보를 수정하면, 기본 흐름의 단계 4로 돌아간다.

.2 고객이 취소를 선택하면, 유스케이스는 종료된다.

유스케이스 내용 – 액터(Actor)

- **액터란 무엇인가?**
  - 이벤트를 완결하기 위해 시스템과 상호 작용하는 개체
  - 액터가 사람일 경우, 시스템과 상호 작용하는 사용자에 의해 수행되는 역할(Role)을 나타냄
  - 시스템과 상호작용하고, 역할을 수행을 위하여 실재하는 외부 개체
- **액터를 정의해야 하는 이유?**
  - 액터가 수행하는 역할은 유스케이스가 필요한 이유와 결과에 대한 관점을 제공
  - 액터에 초점을 맞춤으로써, 시스템이 어떻게 구현될 지가 아니라 시스템이 어떻게 사용될 지에 집중
  - 유스케이스 모델링에 포함될 필요가 있는 잠재적인 사용자 식별을 함

유스케이스 내용 – 일차액터

- 시스템의 서비스 중에 하나를 요청하는 이해관계자
- 시스템의 관점에서 볼 때, 시스템의 운용을 통해서 달성할 수 있는 하나의 목표를 가진 자
- 일차 액터가 유스케이스를 시작하지 않는 두 가지 경우
  - 궁극적 일차액터에 의한 시작
    - 예 : 서비스요청을 하는 고객의 요청을 실행하는 전화 교환원
    - 일차 액터 : 고객
  - 시간에 의한 시작
    - 예 : 매일밤 자정에 실행해야 하는 유스케이스
    - 일차 액터 : 해당 유스케이스에 관심을 갖는 이해관계자

유스케이스 내용 – 지원액터(Secondary Actor)

- 유스케이스를 수행하는 동안 시스템과 통신하는 다른 액터
- 시스템에 서비스를 제공하는 외부 액터
  - 예 : 고속프린터, 웹서비스, 사람
- 시스템이 사용할 외부 시스템과 인터페이스 간의 프로토콜 식별에 도움
- 다른 유스케이스의 일차 액터가, 또 다른 유스케이스에서는 지원액터가 될 수 있음

- 액터 실습 - 다음 중 액터는?
  - 현금 인출기(ATM), 고객, 현금 인출카드, 현금 인출기 화면, 은행 소유주, 서비스 담당직원, 프린터, 은행의 주 컴퓨터 시스템, 은행강도

유스케이스 내용 – 액터 찾기(1/2)

❑ 액터 찾기

- 시스템과 상호작용을 수행하는 외부 개체를 찾음
- 시스템과 상호 작용하는 Stakeholder 분석
- 요구사항 워크샵 초안
- 현재 프로세스와 시스템에 대한 사용자 매뉴얼과 훈련 지침서
- 가이드와 매뉴얼은 종종 잠재적인 액터의 역할을 직접적으로 나타냄
- 사용자와의 미팅 시 작성된 메모는 잠재적인 액터일 수 있는 사용자 식별을 도움

유스케이스 내용 – 액터 찾기(2/2)

❑ 액터를 찾기 위한 질문

- 특정 요구사항에 흥미를 가지는 사람은 누구인가?
- 조직 내부의 어디에서 시스템이 사용되는가?
- 시스템을 사용함으로 이득을 얻는 사람은 누구인가?
- 누가 시스템에 정보를 제공하고 정보를 사용하고 제거하는가?
- 누가(또는 무엇이) 시스템과의 이벤트를 시작하는가?
- 누가(또는 무엇이) 시스템이 이벤트에 응답하는 것을 돕기 위해 시스템과 상호 작용하는가?
- 누가 시스템을 유지보수 하는가?
- 시스템이 기존 시스템(legacy system)과 상호 작용하는가?
- 시스템은 외부 자원을 사용하는가?
- 시스템에 대해 이미 액터가 정의되어 있는가?
- 분석 동안 모델링 되어야 하는 시스템과 상호 작용하는 하드웨어 또는 소프트웨어 디바이스가 존재하는가?
- 만약 시스템 내에서 이벤트가 발생한다면, 외부 개체가 이 이벤트를 통지 받을 필요가 있는가? 시스템이 태스크 수행을 돕기 위해 외부 개체에 이를 요청할 필요가 있는가?

유스케이스 내용 – 추가 항목

❑ 추가 항목

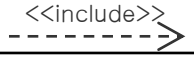
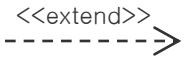
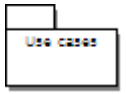
- 사전조건(pre-condition)
  - 유스케이스 시작 전에 시스템이 보장해야 하는 사항
- 보증(guarantee)
  - 유스케이스 종료 시에 시스템이 보장하는 사항
  - 성공 보증(success guarantee)은 성공 시나리오 종료 시 보장하는 사항 표시
  - 최소 보증(minimal guarantee)은 임의의 시나리오 종료 시 보장하는 사항 표시
- 트리거(trigger)
  - 유스케이스가 시작되도록 한 이벤트
- 유스케이스는 간략하게 유지하는 것이 중요
  - 긴 유스케이스는 가독성 저하

2. 유스케이스 다이어그램

- 유스케이스 다이어그램
- 유스케이스 수준
- 유스케이스 목표 수준
- 유스케이스 크기
- 유스케이스 분리

유스케이스 다이어그램(1/3)

❑ 표기법(Notation)

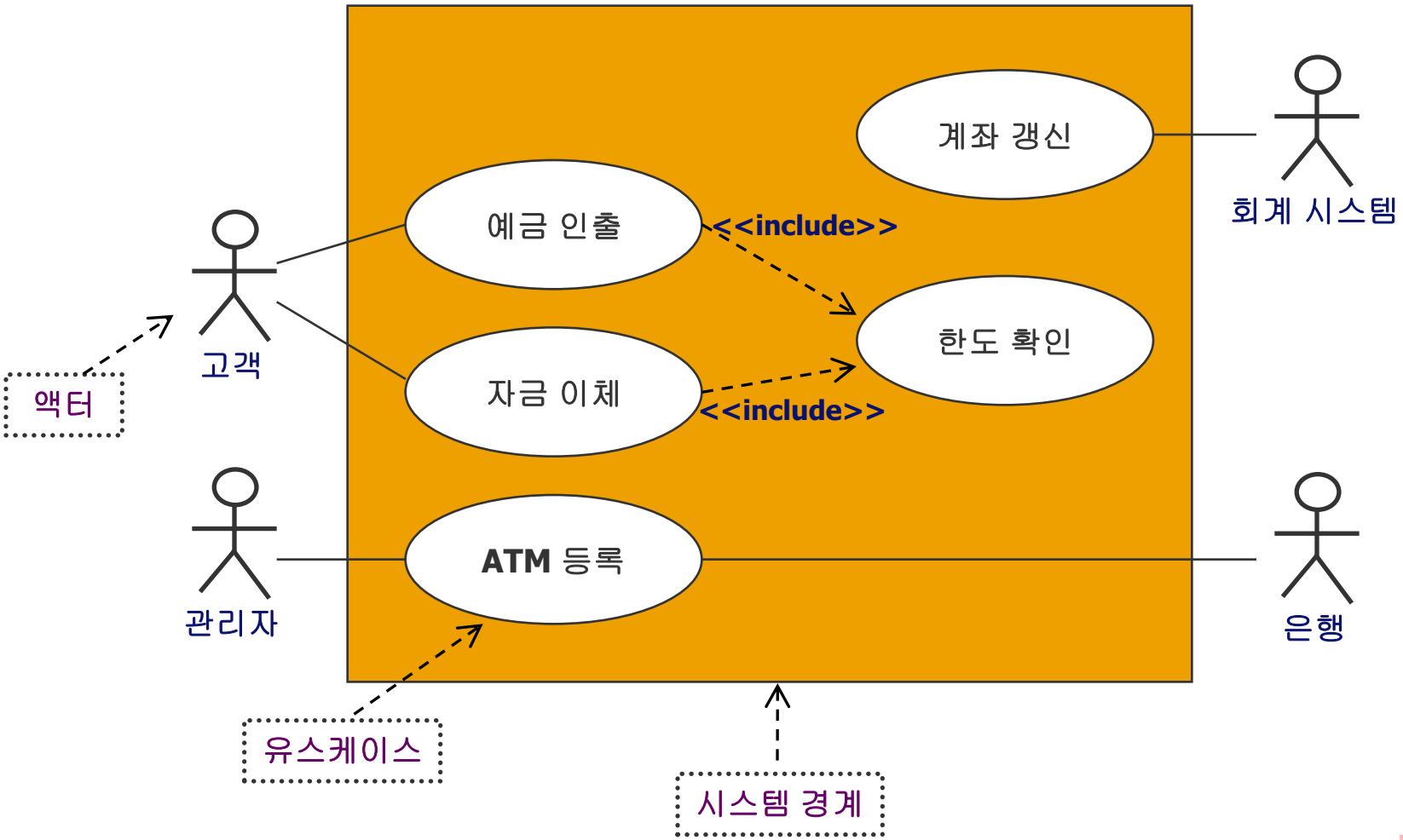
시스템 경계		시스템의 경계와 시스템을 위한 유스케이스를 포함. 액터들은 시스템 경계 외부에 위치한다.
연관관계		직접적으로 상호작용하는 액터와 유스케이스 간의 관계. 액터는 하나 이상의 유스케이스와 연관될 수 있으며, 유스케이스 또한 하나 이상의 액터와 연관될 수 있음.
단방향 연관관계		통신의 개시자(Initiator)를 나타내는 연관
포함 관계		하나의 유스케이스가 다른 유스케이스를 사용함을 나타내는 두 유스케이스 간의 관계
확장 관계		하나의 유스케이스가 다른 유스케이스의 행동을 추가함을 나타내는 두 유스케이스 간의 관계
일반화 관계		액터 간의 계승 관계를 나타내는 유스케이스
액터		시스템과 상호작용하는 사용자, 시스템, 다른 요소를 나타내는 외부 개체
패키지		유스케이스(그리고 UML의 다른 요소들)의 컨테이너. 다른 속성을 가진 임의의 수의 유스케이스 모델을 구성하기 위해 사용
유스케이스		시스템에 의하여 수행되는 행위들의 집합을 설명
우선순위 유스케이스		좀 더 긴급하거나 중요한 유스케이스를 구별하기 위해 우선순위가 부여된 유스케이스

유스케이스 다이어그램(2/3)

- **UML과 유스케이스**
  - 유스케이스 내용에 대한 언급은 없음
  - 유스케이스 다이어그램 형식만을 제공
  - 중요한 것은 유스케이스 내용
    - 유스케이스 다이어그램에 너무 많은 시간을 투자하지 말 것
- **유스케이스 다이어그램**
  - 유스케이스들의 목차의 그래픽적인 테이블 표현
  - 액터, 유스케이스, 액터와 유스케이스 간의 관계로 구성
  - 관계의 종류
    - 액터가 유스케이스를 실행
    - 유스케이스가 다른 유스케이스를 포함
  - 포함관계 이외의 유스케이스간 관계는 무시할 것
    - 대신 유스케이스의 텍스트 서술에 집중

유스케이스 다이어그램(3/3)

❑ 유스케이스 다이어그램의 예



유스케이스 수준(1/3)

- **유스케이스 수준 개요**
  - 요약 목표
  - 사용자 목표 <- FOCUS!!!
  - 하위 기능
- **요소 비즈니스 프로세스(EBP :Elementary Business Process)**
  - 사용자 목표 수준 유스케이스의 크기와 유사
  - 어떤 비즈니스 이벤트에 대처하기 위해서 한 사람이 한 장소에서 특정 시점에서 수행할 수 있는 작업의 단위
  - 비즈니스 가치를 생성할 수 있는 수준
    - 예) 주문하기: 이는 '문서 출력하기'나 '항목 삭제하기'와 같은 수준이 아니라 주요 흐름이 5~10 스텝 정도 존재하는 크기

유스케이스 수준(2/3)

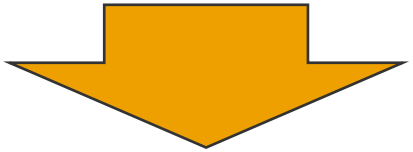
- **요소 비즈니스 프로세스 적용**
  - 유스케이스 작업을 끝난 후에 일을 했다는 만족도를 느끼는 정도
  - 한 사람이 한자리에서 수행하는 어떤 테스트를 수행하는 정도(2~20분)
  - 상급자에게 자신의 수행한 작업을 보고할 수 있는 수준 (Boss 테스트)
    - 예) '오늘 로그인 10번 수행하였습니다.' vs. '오늘 대출을 10건 수행하였습니다.'
  - 유스케이스 작업을 수행하면 연말 실적으로 반영되어 좋은 연봉 협상을 할 수 있는 근거가 되는 수준
- **사용자 목표 수준으로 볼 수 없는 예제**
  - '온라인 경매를 통한 구매완료': 온라인 경매는 일반적으로 수일 이상이 걸리는 작업(요약 수준)
  - '로그인': 연속 10회 로그인 한다는 작업은 시스템을 사용하는 사람의 업무 책임이나 목적을 만족시키지 못함(하위 기능 수준)

유스케이스 수준(3/3)

❑ EBP 가이드라인 적용 예제

다음은 분석가와 사용자(출납원)간의 요구사항 파악을 위하여 오갈 수 있는 대화이다. EBP 가이드라인을 적용하여 사용자 목표 수준의 유스케이스를 파악하는 방법을 알아보자.

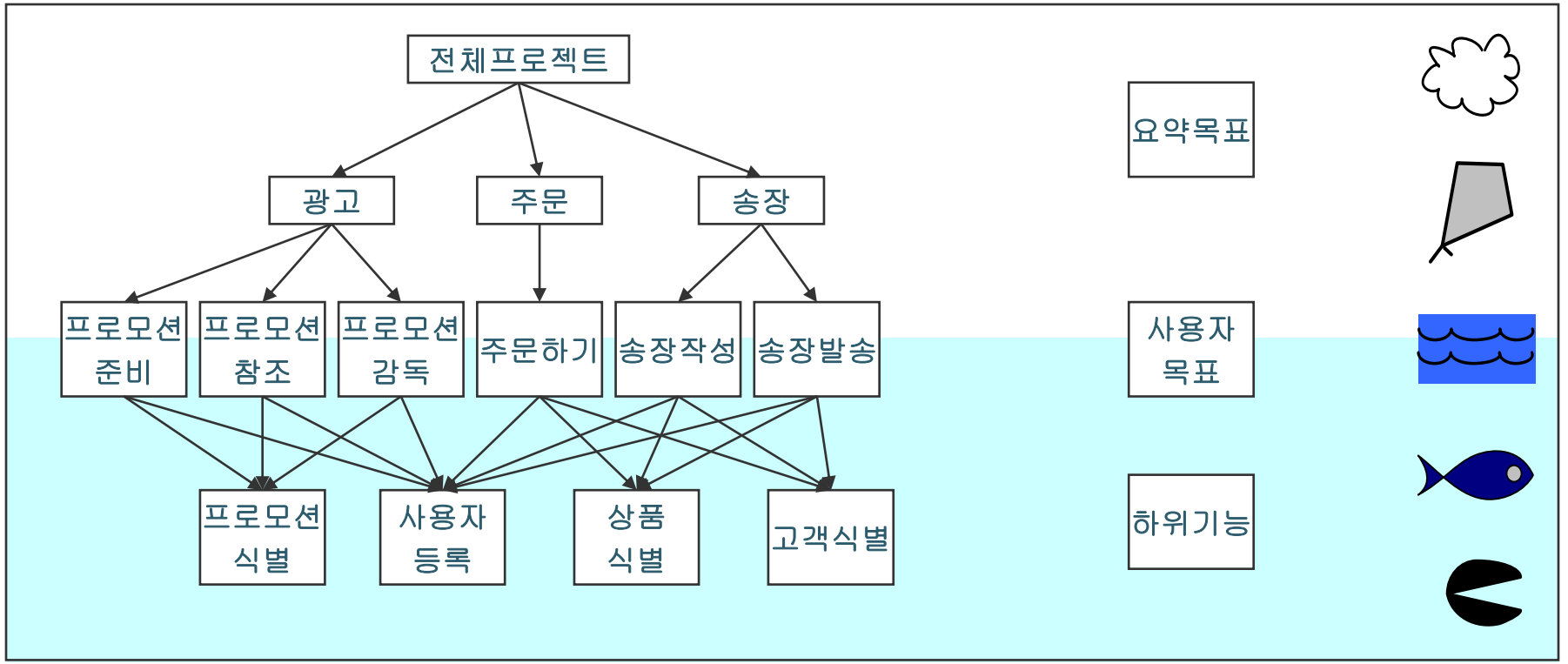
시스템 분석자: 시스템 사용을 할 때 당신의 목표는 무엇입니까?
출납원: 신속하게 로그인하는 것과, 판매를 파악하는 것입니다.
시스템 분석자: 더 높은 수준의 목표를 생각해 봅시다. 무엇 때문에 로그인이 필요하다고 생각하죠?
출납원: 시스템에 저의 신분을 확인시킵니다. 그러면, 판매 파악과 기타업무를 위해 제가 시스템을 이용하는 것이 허용되는가를 시스템이 검증할 수 있습니다.
시스템 분석가: 왜 그러한 일을 한다고 생각하시지요?
출납원: 도난을 방지하고, 데이터 변조를 막고, 회사의 정보 유출을 막기 위해서입니다.



"도난을 방지하고~" 부분은 사용자 목표보다 높은 수준이다. 대화 중 "신분을 확인 시키고 검증 받는다" 와 같은 목표가 사용자 목표 수준에 가깝다. 하지만, 이는 비즈니스 가치를 부가하지는 않을 것이다. 즉, "판매를 파악한다" 와 같은 내용이 EBP가이드라인을 따르는 사용자 목표 수준의 유스케이스로 적합하다고 보면 된다.

유스케이스 목표 수준(1/5)

- 하위 그리고 또 그 하위-하위 목표를 가지는 목표들
- 어떤 수준의 목표를 서술해야 하는지에 대한 유스케이스 작성시의 혼란
- 유용한 도구 : 목표수준에 이름 붙이기



유스케이스 목표 수준(2/5)

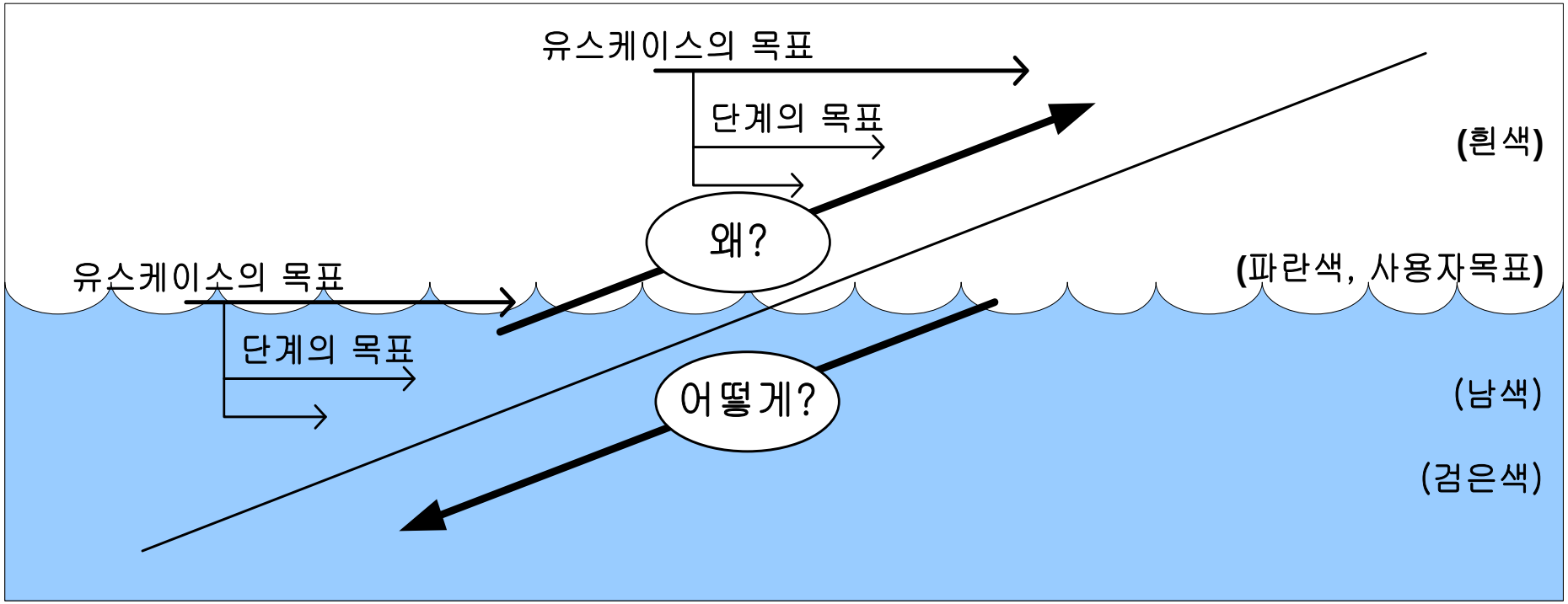
❑ 사용자 목표 찾기

- 어떤 사람이 시스템에서 무엇인가를 원한다. 그것을 얻은 뒤 그(녀)는 계속 하거나 다른 일을 한다. 지금 그녀가 여러분의 시스템에서 원하는 것은 무엇인가?
- 이 수준은 비즈니스 프로세스 모델링의 기초 비즈니스 프로세스, 유스케이스의 사용자 목표
- 사용자 목표를 찾기 위한 두 가지 질문
  - 일차액터가 정말로 원하는 것은 무엇인가?
  - 이 액터가 왜 이 일을 하는가?

유스케이스 목표 수준(3/5)

❑ 목표 수준 높이기와 낮추기

- 액터는 왜 이 일을 하는가? 어떻게 하는가?



유스케이스 목표 수준(4/5)

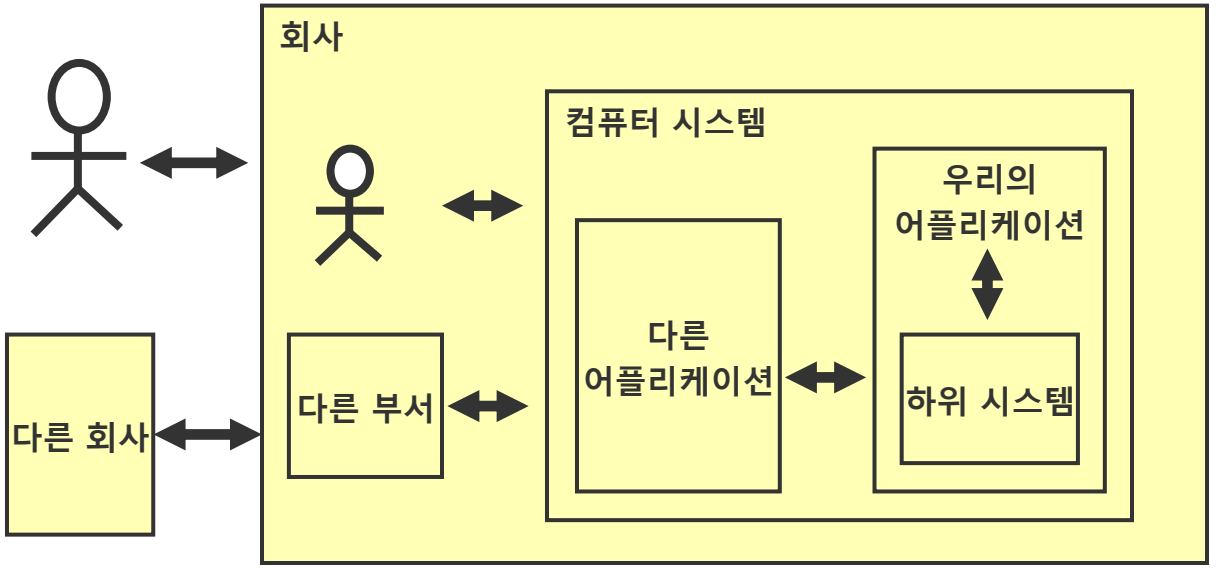
❑ 유스케이스의 길이

- 잘 작성된 유스케이스 대부분은 3~8단계. 최고 11단계
- 10단계 이상이라면 유저인터페이스 세부사항을 포함했다거나, 너무 낮은 수준의 행동단계를 기술했을 가능성이 크다. 이러한 경우:
  - 유저인터페이스의 세부사항을 제거한다. 액터의 움직임이 아닌 의도를 보여 줌
  - 한 단계 위의 목표 수준을 찾기 위해 "왜"라는 질문을 함으로써 목표수준을 높임
  - 단계를 통합
- 대부분의 유스케이스 길이는 두 페이지 정도가 적절

유스케이스 목표 수준(5/5)

❑ 설계 범위

- 시스템의 영역
- 설계나 논의에 대한 책임을 지는 하드웨어와 소프트웨어로 구성된 시스템의 집합 경계
- 모든 유스케이스의 관심 주제임



유스케이스 크기(1/3)

- **가장 바깥쪽 유스케이스**
  - 모든 유스케이스에 대해 가장 바깥쪽의 설계 범위
  - 작성시 장점
    - 적은 시간 안에 작성 가능함
    - 시스템의 컨텍스트를 명확히 제공함
    - 시스템의 가장 바깥쪽 사용자(주로 비즈니스 액터)에게 제공하는 궁극적 이익을 가시화함
    - 시스템의 행위를 전체적으로 훑어볼수 있도록 가시화함

유스케이스 크기(2/3)

❑ 사용자 목표 수준 유스케이스 - 개략적인 가이드라인

- Ivar jacobson의 제안
  - '하나의 특정 비즈니스 프로세스를 구현하고 있는 대형 시스템에 있어서 20개 이상의 유스케이스를 도출하지 않는 것이 가장 적절하다' -> 이는 포함, 확장, 일반화 관계 등을 제외한 숫자
  - 하지만, 유스케이스 숫자가 20개가 넘거나 적을 경우에 유스케이스 분해나 통합을 통하여 20개의 숫자로 조절하는 작업은 무의미 함
    - 5개의 유스케이스로 훌륭하게 표현된 상업용 시스템이 분명 존재하고, 40개정도의 유스케이스가 적절하게 도출되어 있는 시스템도 존재함
    - 하지만, 700개의 유스케이스가 존재한다면, 이는 분명히 잘못된 모델임

유스케이스 크기(3/3)

❑ 사용자 목표 수준 유스케이스 - 개략적인 가이드라인

- 유스케이스 포인트[UCP]
  - 1UCP = 1 Use Case Point
  - 1UCP를 구현하는데 드는 시간은 약 20 Man/Hour 정도
  - 간단한 유스케이스는 대략 5 UCP 정도로 추정
    - 이를 구현하기 위해서는 100M/H(2.5 M/W)가 소요
    - 즉 1명이 2.5 주 정도 걸려 구현할 수 있는 정도의 수준
    - 이는 절대적인 시간은 아니고, 상황에 따라서 많이 달라질 수 있음
    - 개발자의 수준이 매우 높다고 여겨지면, 간단한 유스케이스는 약 1주 이내에도 구현 가능함
- IBM 컨설턴트의 제안
  - 6명이 9개월 정도 수행하는 프로젝트는 보통 대략적으로 10~30개 정도의 유스케이스가 적절함

유스케이스 분리(1/3)

❑ 개요

- 다른 산출물과 마찬가지로 유스케이스를 작성하는 목적 중의 하나는 의사소통
- 작성된 유스케이스는 읽기에 불편함이 없어야 함

❑ 상황 및 분리 방법

- 대안흐름에 대한 브레인스토밍을 하다 보면 자연적으로 복잡한 확장 흐름이 발생
  - 이 경우 확장 흐름으로부터 하위 유스케이스를 분리하여 포함 관계로 유지
- 하위 유스케이스에 대한 일차 액터의 목표를 정의하고, 이 목표를 하위 유스케이스의 이름으로 명명
- 보통 사용자 목표수준의 유스케이스로부터 분리된 하위 유스케이스는 하위 기능 수준의 유스케이스가 됨

유스케이스 분리(2/3)

❑ 분리해야 할 경우

- 특정 흐름이 여러 유스케이스에서 사용
  - 특정 흐름을 하위 유스케이스로 만들어 이를 사용하는 유스케이스가 참조할 수 있게 작성 -> 이것이 하위 기능 수준의 유스케이스를 만드는 가장 유일한 이유
- 확장 흐름이 너무 복잡하여 유스케이스가 읽기 매우 어려움
  - 2페이지 이상의 유스케이스 시나리오가 작성된다던가 확장 흐름의 들여쓰기가 세 번 이상 반복된다면 하위 유스케이스로 분리

유스케이스 분리(3/3)

❑ 고려사항

- 유스케이스를 분리하면 프로젝트 관리상의 비용이 증가
  - 단지 이전에 언급한 상황에만 비추어 하위 유스케이스를 기계적으로 분리하는 것은 좋은 방법이 아님
  - 새로운 유스케이스에 대하여 추적하고, 이를 기반으로 프로젝트 일정을 잡고, 이를 검토하고, 유지 보수해야 하는 비용
- 가독성
  - 유스케이스 분리 시 하위 유스케이스의 표시는 밑줄을 그어 표기하는 것이 가독성을 높임
  - 예제

유스케이스: 주문하기

1. 사원이 고객을 식별하고, 시스템은 고객의 기록을 찾아낸다.
2. 사원이 주문 정보를 입력한다.
3. 시스템이 지불금액 계산을 한다.

…

유스케이스: 지불금액 계산

1. 시스템은 주문에 대한 기본 지불금액을 계산한다.
2. 시스템은 고객에 대한 할인금액을 계산한다.

…

실습 : 도서관리 시스템

1. 실습도메인 설명
2. 액터찾기
3. 유스케이스명세서
4. Use Case Model 실습

실습도메인 설명(1/4)

- 사용자는 자신이 보고 싶은 도서를 도서관리 시스템 (BookManagementSystem)을 통해 대출이 가능한지 조회한다.
- 대출자는 시스템에 계정을 통해 로그인 한다.
- 대출자가 자신이 대출한 도서에 대한 현황을 조회할 수 있다.
- 대출자는 관리자에게 도서 대출을 요청한다.
- 관리자는 대출자의 정보와 도서 정보를 시스템에 등록함으로써 대출 신청을 처리한다. 대출자는 정해진 기한 내에 도서를 반납해야 한다. 대출 신청을 위해서는 관리자는 고객등록을 해야 한다. 도서를 대출, 반납할 수 있는 사용자를 등록, 수정, 삭제, 조회할 수 있게 한다.
- 관리자는 도서 대출에 통계를 확인할 수 있도록 한다. 대출 통계는 고객별 통계와 도서별 통계로 분류되며 고객별 통계의 경우 총 대출 회수, 연체회수, 현재 대출 회수를 보여준다. 도서별 통계의 경우 도서별 총 보유고, 대출 회수, 대출 순위를 표시한다.

실습도메인 설명(1/4)

- 도서 관리자는 도서의 고유한 속성들을 입력함으로써 새로운 도서를 관리 시스템에 등록한다. 도서가 등록되면 관리자는 등록된 도서를 조회할 수 있으며, 등록된 도서에 대한 통계를 볼 수 있다.
- 사용자는 관리자에게 도서를 반납한다. 관리자는 대출자의 정보와 도서 정보를 시스템에 등록함으로써 대출 반납을 처리한다. 대출자가 정해진 기한 내에 도서를 반납하지 못했을 경우 연체 처리된다

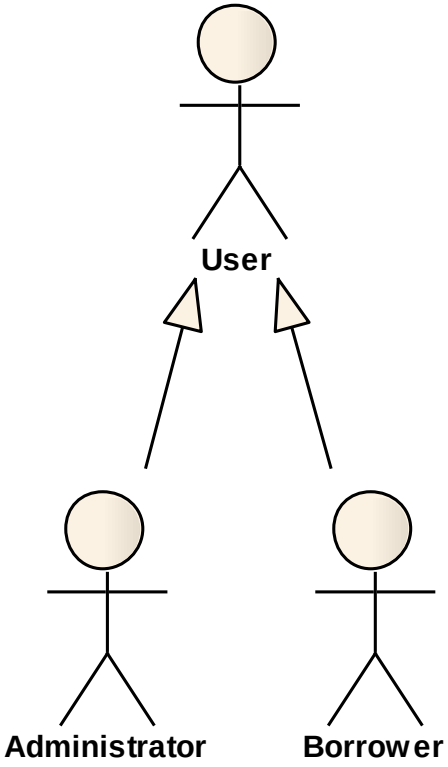
실습도메인 설명(3/4)

- 사용자는 자신이 보고 싶은 도서를 도서관리 시스템(BookManagementSystem)을 통해 대출이 가능한지 조회한다.
- 대출자는 시스템에 계정을 통해 로그인 한다.
- 대출자가 자신이 대출한 도서에 대한 현황을 조회할 수 있다.
- 대출자는 관리자에게 도서 대출을 요청한다.
- 관리자는 대출자의 정보와 도서 정보를 시스템에 등록함으로써 대출 신청을 처리한다. 대출자는 정해진 기한 내에 도서를 반납해야 한다. 대출 신청을 위해서는 관리자는 고객등록을 해야 한다. 도서를 대출, 반납할 수 있는 사용자를 등록, 수정, 삭제, 조회할 수 있게 한다.
- 관리자는 도서 대출에 통계를 확인할 수 있도록 한다. 대출 통계는 고객별 통계와 도서별 통계로 분류되며 고객별 통계의 경우 총 대출 회수, 연체회수, 현재 대출 회수를 보여준다. 도서별 통계의 경우 도서별 총 보유고, 대출 회수, 대출 순위를 표시한다.

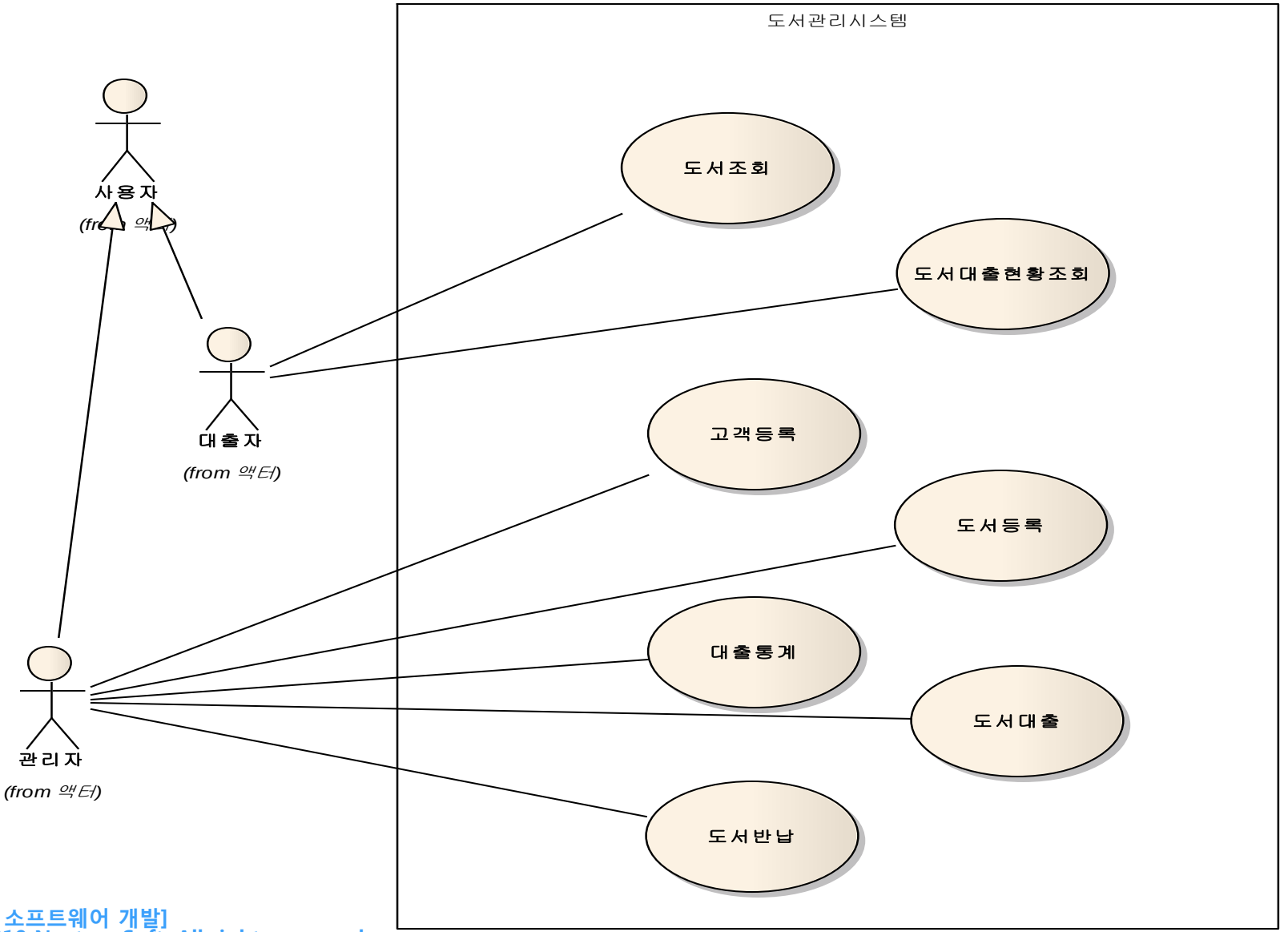
실습도메인 설명(4/4)

- 도서 관리자는 도서의 고유한 속성들을 입력함으로써 새로운 도서를 관리 시스템에 등록한다. 도서가 등록되면 관리자는 등록된 도서를 조회할 수 있으며, 등록된 도서에 대한 통계를 볼 수 있다.
- 사용자는 관리자에게 도서를 반납한다. 관리자는 대출자의 정보와 도서 정보를 시스템에 등록함으로써 대출 반납을 처리한다. 대출자가 정해진 기한 내에 도서를 반납하지 못했을 경우 연체 처리된다

UseCase Actors



Primary Use Cases



3. 유스케이스 명세

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

유스케이스 데이터 작성(1/4)

❑ 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)

유스케이스 명세서

1. **고객**이 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

용어집 or 기타 산출물

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

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

유스케이스 작성절차(1/2)

❑ 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

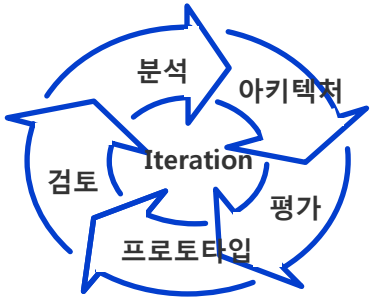
작성 접근방법(2/4)

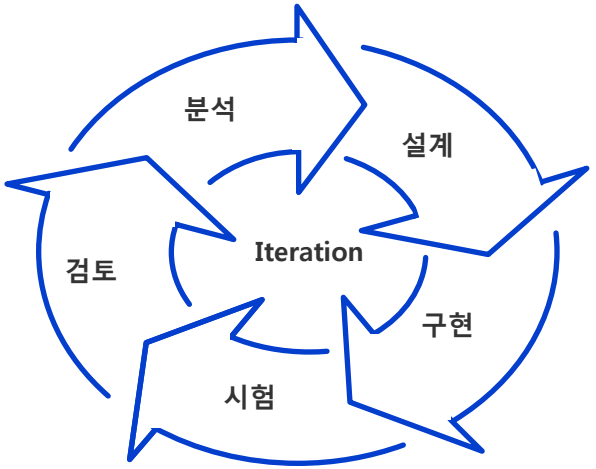
❑ 정밀도

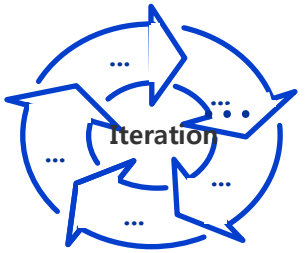
- 1단계: 액터와 목표 작성
  - 시스템과 관련된 액터와 이의 목표를 작성
  - 정확성과 안정성을 위하여 이를 검토
- 2단계: 유스케이스 요약서 & 성공 시나리오 작성
  - 작성하려고 선택한 유스케이스에 대하여 간단한 요약을 작성하거나 주요 성공 시나리오를 작성
  - 시스템이 이해관계자의 이해관계를 정확히 반영하고 있는지 검토
- 3단계: 실패조건 작성
  - 주요 성공 시나리오를 완성하고 발생 가능한 모든 실패 상황에 대해서 브레인스토밍을 수행
- 4단계: 실패처리 작성
  - 시스템이 각 실패 상황에서 대처하는 방법을 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	1차 유스케이스	개선된 유스케이스
이름 : 액터 : 요약 :	이름 : 액터 : 요약 : 기본 흐름 :	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.	유스케이스의50% 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)
1. 고객은 계좌번호, 이름, 주소를 제시한다.
2. 시스템은 고객이 선호하는 물품 정보를 가져온다.
3. 사용자는 구매할 품목을 선택하고, 구매하기 위해 표시한다.
4. 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 들여 쓰도록 하라

- 유스케이스의 가독성을 위하여 조건처리는 다음과 같이 들여 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.>

유스케이스 작성 양식(2/7)

❑ 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.

유스케이스 작성 양식(7/7)

❑ 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

유스케이스 패키지

❑ 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)

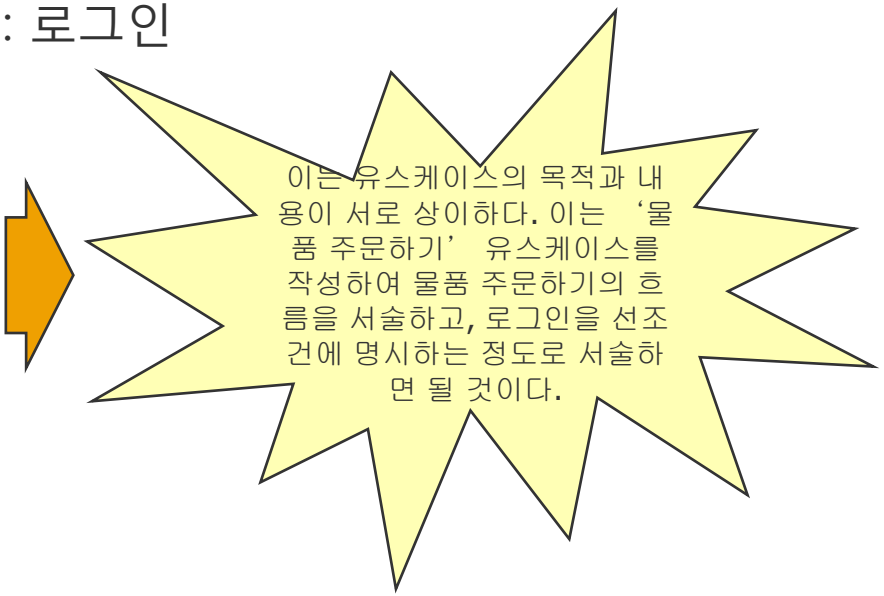
❑ 잘못 작성된 유스케이스 예제(4/4)

- 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.

…

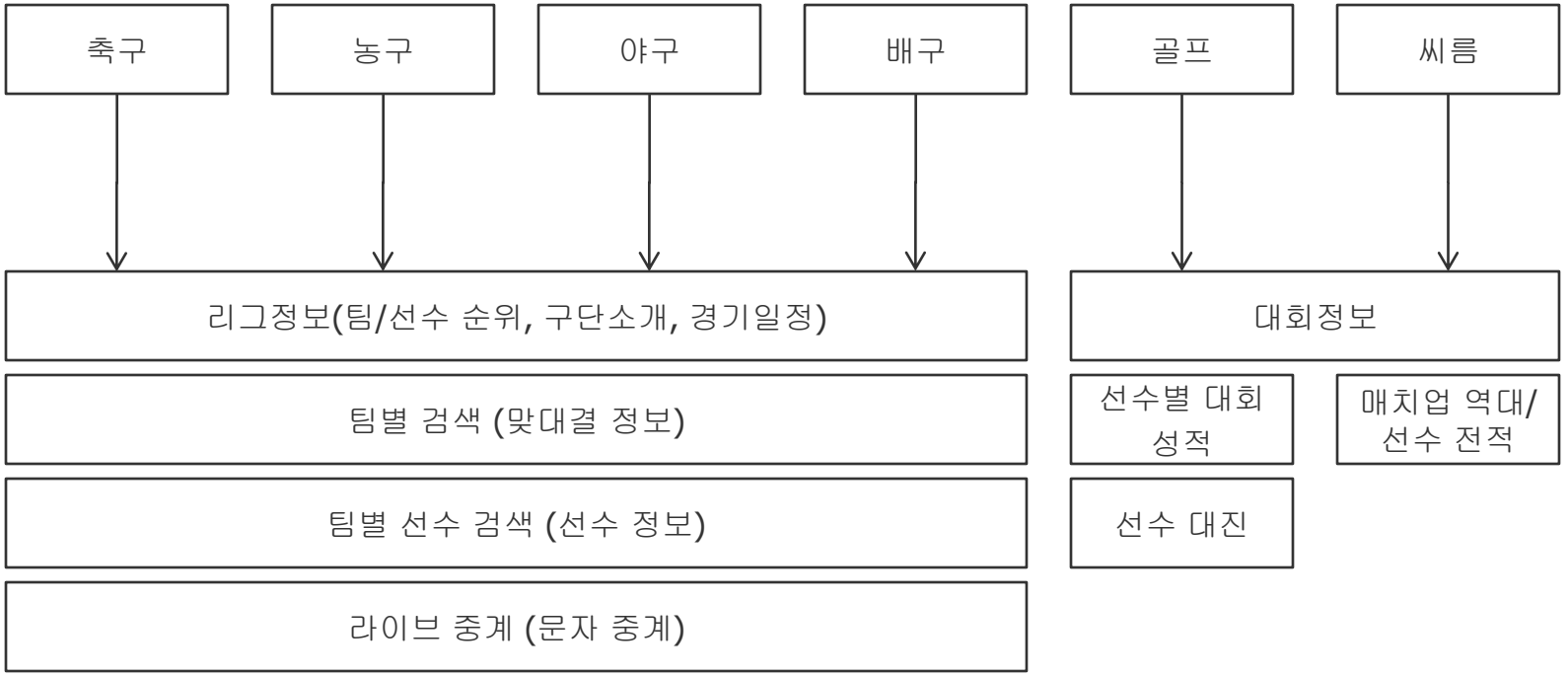


실습 : 유스케이스 명세

1. 비즈니스 개요
2. AS-IS 구조 분석
3. 사용자 요구사항

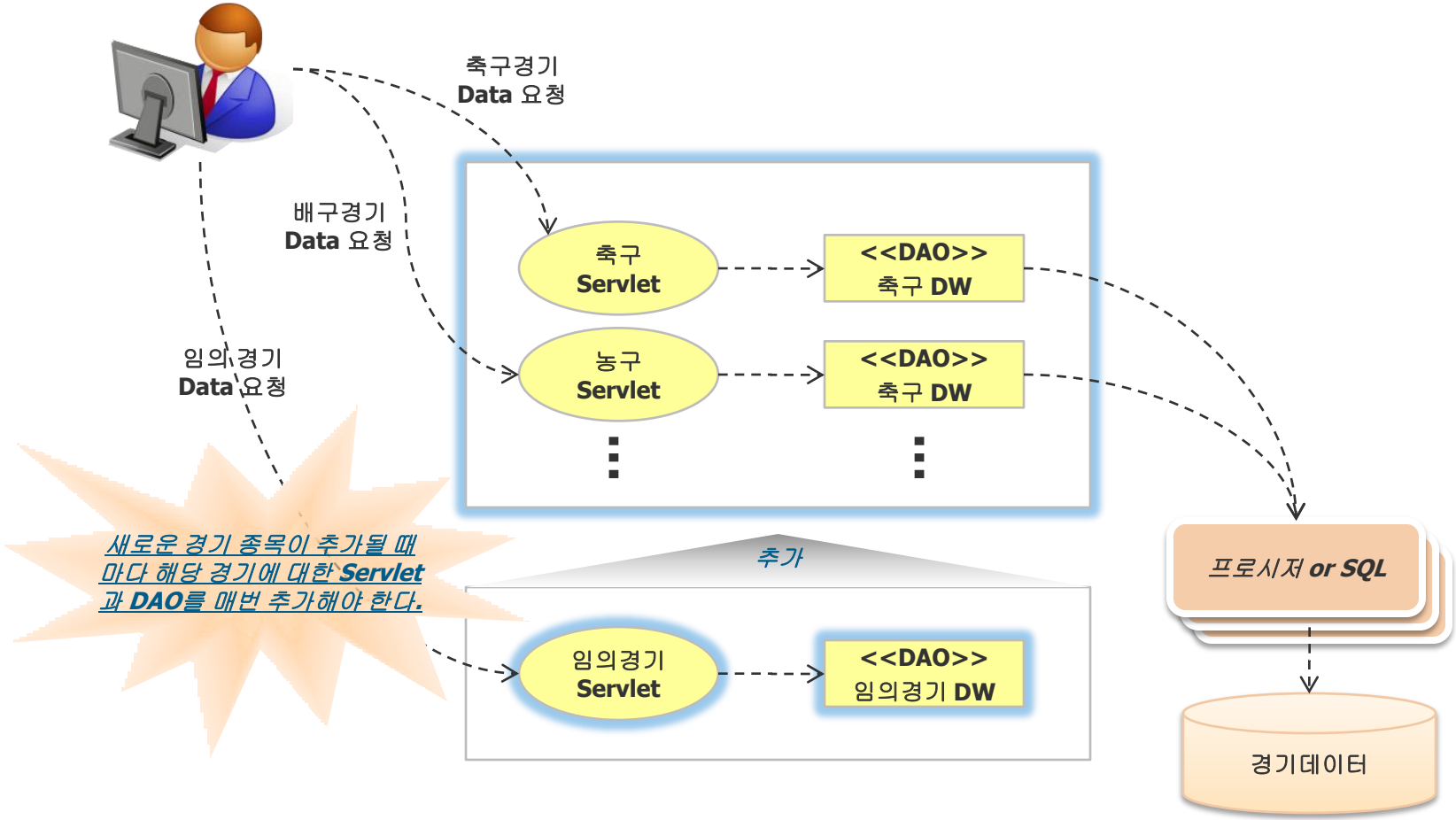
비즈니스 개요

- 경기정보를 사용자에게 제공
- 경기정보는 팀/선수 순위, 구단 정보, 선수 정보, 경기일정 및 맞대결 정보 등으로 구성



AS-IS 구조 분석

- AS-IS 경기정보는 DB 기반의 속성 중심의 구조
- 다른 경기 추가 시 유연성이 결여



❑ 경기정보 관리 시스템

예제 애플리케이션은 경기정보를 관리하는 온라인 경기정보 포탈시스템 이다.
사용자는 관심 있는 경기를 선택하여 원하는 정보를 얻고자 한다. 이 사이트는 축구, 야구, 농구, 배구, 골프, 씨름에 대한 경기정보 및 뉴스, 라이브(문자) 중계 등의 서비스를 제공한다.
사용자는 경기종목에 대한 각 리그 별로 경기일정과 팀 별 맞대결 전적, 팀 및 선수정보와 순위 등을 확인할 수 있다. 단, 경기일정과 순위를 제외한 정보를 원할 경우 로그인해야 하며 로그인 가능한 사용자가 아니라면 회원가입을 통해 로그인이 가능하도록 해야 한다.
경기정보는 데이터 제공업체를 통해 데이터가 입력이 된다. MLB, NBA, KBO, 유럽축구(EPL, 프리메가리가, 세리에A) 등의 정보는 '스포츠21' 업체에서 제공하고, K리그는 'I러브K리그, KBL은 'IT스포츠', WKBL은 '스포츠Easy', 배구는 '스파이크IT' 업체들로부터 제공 받는다. 골프 및 씨름은 경기일정이 많지 않아 관리자가 직접 입력한다.
관리자는 데이터 제공업체로부터 들어온 경기정보를 확인하여 필요한 경우 데이터를 보정할 수 있다. 또한 관리자는 해당 사이트에 사용자 관심을 높이기 위해서 이벤트를 게시한다. 사용자는 마감 전에 해당 이벤트에 참여할 수 있고 관리자는 추첨을 통해 이벤트 당첨자를 뽑은 후 당첨 메일을 메일발송 시스템을 통해 사용자에게 보낸다.
** 경기일정을 통해 중계일정(KBS Nsports, SBS Sports, MBC ESPN 등)을 확인이 가능
** 해외 경기에 대한 날짜 및 시간은 한국시간 기준으로 정보가 들어오고 야구일정의 경우 선발투수가 나와야 함
** 이벤트 참여는 각 경기 별로 조회가 가능