

Chap 04 요구 분석



목 차

4.1 요구

4.2 요구 추출

4.3 요구 분석

4.4 유스케이스

4.5 요구 명세

4.6 요구 검증

요구 분석

- 소프트웨어 개발의 실질적인 첫 단계
- 사용자의 요구에 대하여 이해하고 정리하는 작업
- 세 가지 작업
 - 요구 추출
 - 요구 분석 및 정의
 - 요구 확인

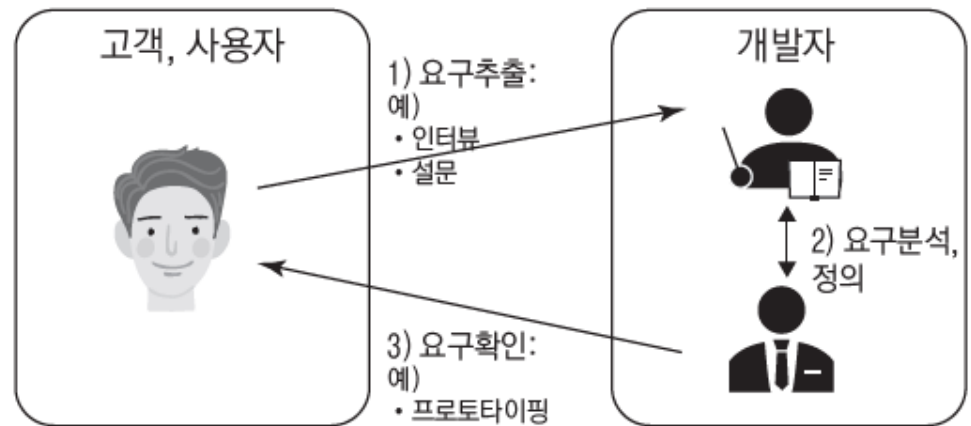


그림 4.1 요구 분석 단계의 활동

- 요구 변경은 파급효과가 큼

4.1 요구(Requirements)

- 요구
 - 시스템에 대한 고객의 요청을 확정한 것
- 진정한 요구를 찾는 일
 - 프로젝트 성공의 필수 조건
 - 여러 이해 당사자(stakeholder)의 이해 관계와 관련
- 제약 사항
 - 특정한 프로그래밍 언어, 특정한 제품 사용
 - 소프트웨어 시스템의 해결책을 제한

요구의 분류

고객이 요구하는 시스템이 처리할 기능
(업무 절차나 기계 동작을 실현한 것)

- 동사로 표현됨
- 쉽게 파악됨
- 제품 기능
- 사용사례로 정리

요청된 기능 이외에 시스템이 갖추어야 할 조건, 특징
(응답 속도, 고장에서 회복되는 기간, 보안 등)

- 형용사로 표현됨
- 파악하기 어려움
- 제품 속성
- 품질 속성 시나리오로 정리

기능 요구



비기능 요구

기능 요구

- 시스템이 외형적으로 나타내는 기능과 동작
 - 현금인출기: 현금의 인출, 잔금 조회, 계좌 이체, 현금 서비스
- 시스템과 외부 요소들 간의 인터랙션
 - 어떤 상태일 때 외부의 데이터나 명령을 받아들여 어떤 반응을 하는지 기술
- 입력 자료가 제공되어 이를 변환 처리하여 결과를 출력
 - '시스템은 ...을 해야 한다'

기능의 종류

분류	질문	사례(ATM)
기능	<ul style="list-style-type: none"> • 시스템이 무엇을 하는가? • 시스템이 언제 그 일을 하는가? • 시스템이 운용될 때 여러 가지 다른 모드가 있는가? • 언제 어떻게 시스템이 변경되거나 확장되는가? 	<ul style="list-style-type: none"> • 현금이나 수표를 인출, 조회, 입금, 송금 • 24시간 • 준비 중, 유지보수 중 • 지폐나 수표가 바뀔 때
자료	<ul style="list-style-type: none"> • 입력, 출력이 무엇이며 어떤 형태를 갖는가? • 얼마나 자주 자료를 받고 내보내는가? • 자료가 얼마나 정확하여야 하나? • 시스템에 유입되는 자료의 양 • 데이터는 일정 기간 동안 보관되어야 하나? 	<ul style="list-style-type: none"> • 현금 카드, 현금이나 수표 • 매일 수천 건 • 100%정확하여야 • 200바이트 × 수천 건 • 20년
입출력	<ul style="list-style-type: none"> • 다른 시스템에서 유입, 배출되는 입력은 무엇인가? • 데이터의 특정한 형태가 있는가? • 자료 전달에 사용되는 특정 미디어가 있는가? 	<ul style="list-style-type: none"> • 계좌 정보 • 문자와 숫자 • 없음
사용자	<ul style="list-style-type: none"> • 누가 시스템을 사용할 것인가? • 사용자가 여러 그룹인가? • 각 사용자 그룹의 컴퓨터 사용 경험은? • 각 사용자 그룹에 따라 필요한 교육은? 	<ul style="list-style-type: none"> • 은행 고객 • 3그룹(고객, 은행원, AS 엔지니어) • 초보 • 행원 교육

비기능적 요구

- 성능 – 시스템의 처리량, 반응시간, 실시간 처리, 자원 이용률
- 품질 – 신뢰성, 가용성, 사용시 오류 발생률
- 안전 – 의도하지 않은 오퍼레이션으로 인하여 원치 않는 상태에 있는 것을 방지하는 역량
- 보안 – 시스템의 자원을 악의적인 공격으로부터 보호할 수 있는 역량
- 사용성 – 인터페이스. 동작, 보고 느끼는 것(look and feel)

4.2 요구 추출

- 요구추출의 어려움

- 개발 팀이 **응용 도메인**에 대하여 충분히 알지 못함
- 고객과 사용자가 소프트웨어가 무엇을 하는지 또한 어떻게 요구를 표현할 지 **모름**
- **공통 배경지식 부족**으로 개발 팀과 사용자 사이의 **대화 장벽**이 생김
- 소프트웨어 요구에 대한 명세와 구현이 분리될 수 없어 정확히 명시하기 **어려움** : **What .vs. How**
- 요구 추출 작업을 관리자, 사용자, 개발자 모두 **과소평가**하는 경우가 많음
- **비기능적 요구**를 파악하고 이해하지 못함
- 요구가 **계속해서 변경**됨

4.2 요구 추출

- 추출 세 가지 단계
 - 응용에 대한 정보 출처 파악
 - 응용에 대한 정보 취합
 - 요구와 제한 사항의 정의 -> 요구분석명세서
- } 도메인분석

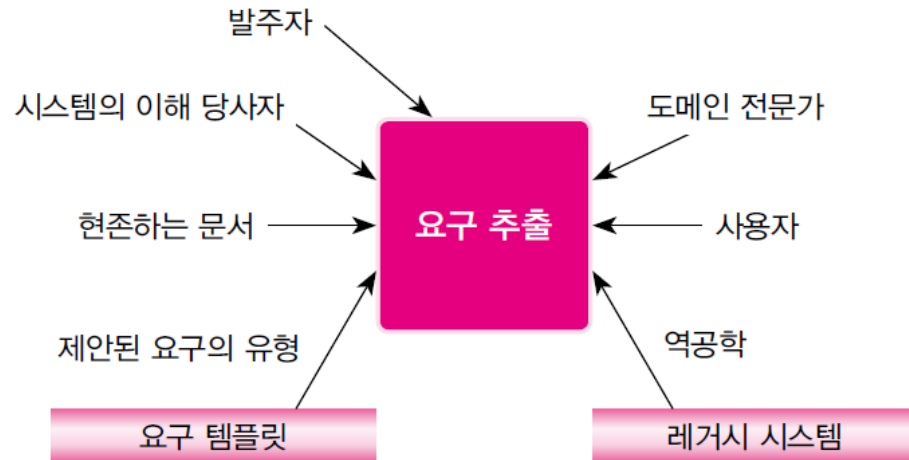
4.2 요구 추출

- 요구 우선순위(priority)에 따른 요구 구별
 - 절대적으로 필요한 요구
 - 요구되나 꼭 필요한 것은 아닌 요구
 - 요구로 판단될 수 있으나 제외될 수도 있는 요구
1. Showstopper : ultimate
 2. Required
 3. Useful
 4. Helpful
 5. Indifferent

요구 정보 출처

- 정보 출처 유형

- 고객
- 도메인 전문가 – 비즈니스 도메인을 지원하는 시스템을 구축하기 위하여 필요한 사람(예, 회계 시스템을 구축하기 위하여 회계사가 필요)
- 이해당사자(stakeholder) – 시스템 운용으로 인하여 영향 받는 사람
- 사용자 – 시스템을 직접 사용하는 사람
- 역공학



4.2 요구 추출

- 정보 수집 방법
 - 고객의 발표
 - 문헌, 양식 조사
 - 인터뷰
 - 설문
 - 브레인스토밍 회의
 - 관찰과 작업 분석
 - 프로토타이핑

고객의 발표

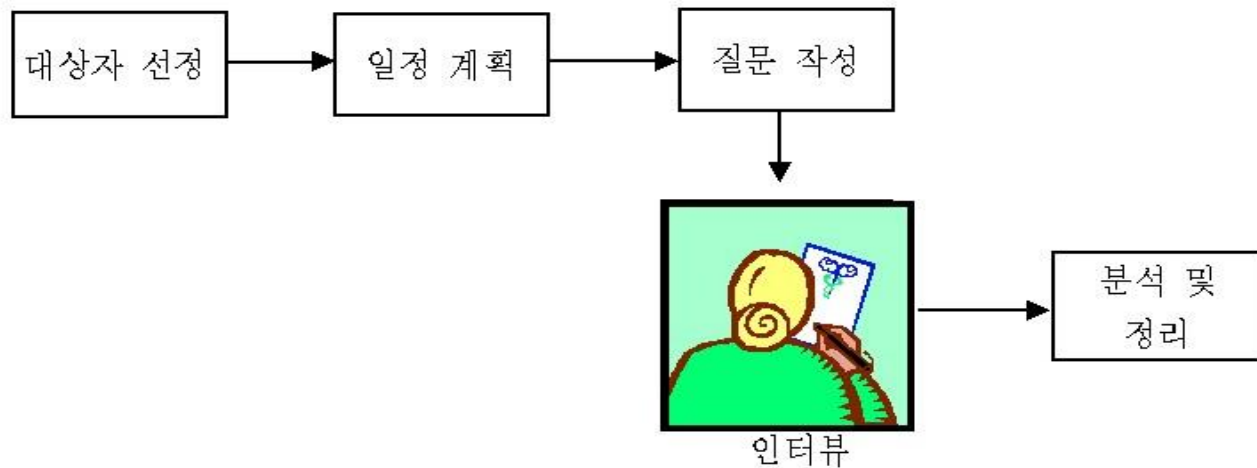
- 개발팀이 구축하는 시스템에 대하여 초기에 개념을 잡을 수 있음
- 효과적인 가이드라인
 - 고객 업무를 잘 알고 있는 운영 책임자나 관리자가 발표
 - 발표하기 전 개발 팀원이 필요한 정보가 있는지 검토
 - 의심이 가는 부분을 질문하여 명확히 할 것
 - 구현과 관련된 토의는 배제
 - 발표 내용의 복사본을 팀원과 공유
 - 2시간 이상의 발표회는 지양

문헌 양식 조사

- 유사한 프로젝트를 조사
 - 현재 개발할 시스템에 대한 통찰 제공
- 업무 문서나 양식을 조사
 - 현재의 업무나 시스템 정보에 대해 깊은 이해 가능
- 산업 및 기업 표준 조사
- 관련 정부 정책/규제 조사

인터뷰

- 인터뷰 수행 가이드 라인
 - 가능하면 많은 당사자와 인터뷰
 - 여유로운 인터뷰 일정
 - 인터뷰 약속 시간을 넘기더라도 여유롭게
 - 중요한 관련자와는 여러 차례 인터뷰



인터뷰

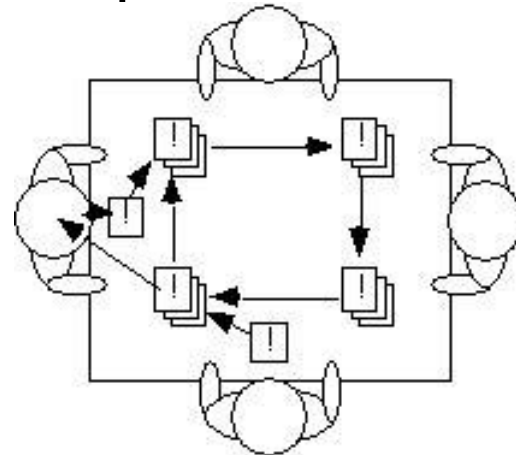
- 반드시 포함해야 할 질문 또는 행동 유형
 - 최대, 최소, 예외 규칙, 예상되는 변동 등 자세한 사항
 - 시스템에 대한 미래의 비전
 - 문제에 대한 최소한의 허용 가능한 솔루션이 무엇인지
 - 다른 정보원은 없는지
 - 인터뷰 대상자에게 다이어그램을 작성하게 함

설문

- 관리자나 사용자와 같은 이해 당사자를 대상
- 이해 당사자들이 의사결정 과정에 포함
- 무기명 설문
 - 이해 당사자들의 관심과 내부정보, 개선 의견 도출
 - 감추어진 정보를 끌어내기 쉬움
- 유의사항
 - 질문은 간단하고 중요한 이슈에 집중
 - 적절하고 잘 기술된 질문

브레인스토밍

- 아이디어를 낼 목적으로 여러 명으로부터 정보를 얻기 위한 회의
- 훈련된 요원이 주재
- 토론보다는 아이디어를 쏟아놓는 회의, 익명성 보장
- 서로 자극이 되어 열정을 가지고 아이디어를 창안
- JAD(Joint Application Development) – 집중 브레인스토밍 세션



브레인스토밍 과정

1. 관련자 모두가 참여하는 회의 소집
2. 경험 많은 사람을 회의 주재자로 선정
3. 테이블에 참석자를 배석시키고 종이 준비
4. 토론을 유도할 질문을 정함
5. 질문에 대하여 답을 종이에 적되 한 장에 하나의 아이디어만 적은 후 참석자에게 돌려 봄
6. 5번 단계를 5~15분간 반복
7. 간단한 설명
8. 모든 아이디어를 칠판에 적은 후 우선순위를 정하기 위하여 투표를 할 수도 있음

프로토타이핑

- 프로토타입
 - 최종 시스템의 예상 기능 중 일부를 빠르게 구현한 프로그램
- 가장 단순한 형태: paper prototype
 - 무엇이 일어날지 설명한 그림을 순서대로 그린 것
 - 병행하여 만들기 적합
- 가장 흔한 형태: 모의 사용자 인터페이스
 - 프로토타이핑 언어로 작성
 - 컴퓨팅, 데이터베이스 접근, 다른 시스템과의 상호작용은 불가능
 - 시스템의 특별한 측면을 프로토타이핑 하기도 함
 - 알고리즘, 데이터베이스 등

4.3 요구 분석

- 요구 후보를 분석하고 결정하여 요구로 확정
- 요구 품질
 - 원자적(atomic)
 - 완전성(complete)
 - 비모호성(unambiguous)과 통일성(consistent)
 - 추적성(traceable)
 - 우선순위화(prioritize)
 - 테스트 가능성(testable)

도메인 분석

- 도메인이란?
 - 요구의 배경
- 설계 모델링에 필요한 여러 개념과 비즈니스 룰을 파악
- 응용 분야에 존재하는 개념을 잘 정의하고 분석하여 시스템에 존재하는 개념으로 정립하는 단계
- 방법
 - 도메인 개념 찾기
 - 도메인 사전 작성
 - 비즈니스 규칙 정리

도메인 정의

- 업무 작업 영역을 파악하고 범위를 규정
 - 정보 시스템을 구축하는데 필요한 개념적인 프레임워크 제공
- 정보 시스템의 서브시스템 개념이 되는 프레임워크 제공
- 넓은 범위의 개념을 더 좁은 범위의 지식들로 체계화 하는 작업

도메인 개념

- 도메인의 목적, 구조, 동작을 구성하는 객체, 프로세스, 사람, 룰 같은 것
- 도메인 개념 발견을 위한 주의사항
 - 요구의 핵심을 발견해야 함
 - 요구가 해결될 것 같은 문제를 발견
 - 문제의 요소를 발견
 - 관련된 도메인의 개념을 발견

도메인 사전

- 도메인 개념을 조직화한 결과물
- 각 항목은 용어가 사용될 때는 언제든지 같은 의미로 통하게 하는 간결한 정의
- 요구, 인터뷰, 매뉴얼로부터 추출
- 도메인 정의에서 표현된 문장, 어절, 제목에 초점을 두고 개념 추출
 - <예> 의료 정보 시스템에서 도메인 개념 추출
 - 진료와 검사 의료 서비스의 성격에 따라 의사, 간호사, 검사원
 - 이 환자에게 적절한 예약된 의료 서비스를 제공
 - 도메인 개념: 의료 서비스, 의사, 간호사, 검사원, 환자, 예약

도메인 사전

- 사전 양식
 - 표로 구성
 - 세가지 항목 포함
 - 명칭
 - 타입
 - 설명

표 4.5 의료 정보 시스템의 도메인 사전

명칭	타입	설명
예약	프로세스	환자가 의료 서비스를 받는 일정. 예약 담당 직원에 의하여 처리됨
예약	객체	환자에게 의료 서비스를 제공하기 위하여 약속된 날짜와 시간
예약 담당 직원	역할	환자를 위하여 예약한다.
의료 서비스	객체	의료진이 환자에게 제공하는 의학적 서비스의 총칭: 진료, 처방, 조제, 검사 등
의료 서비스	기능	의료진이 환자에게 의학적 서비스를 제공하는 행위
예약	프로세스	의료 서비스를 제공하기 이전에 이루어짐. 프로세스는 새로 오는 환자 또는 기존 환자의 개인 정보와 보험 정보를 수집한다. 이 과정에 병원 진료 카드가 발급된다. 예약 담당 직원이 수행한다.
예약 담당 직원	역할	예약을 실시한다.

비즈니스 규칙

- 업무에서 지키기로 한 규정
- 기업이 운영되는 자세한 정책, 규정, 절차, 가이드라인, 표준의 집합
- 사용자에게 요구해도 준비된 전체 목록을 받기 어려움
- 비즈니스 규칙 종류
 - 사실(fact) – 개념이 무엇인지 설명
 - 추론(inference) – 다른 사실로 부터 얻은 사실
 - 액션 구동자(action enabler) – 조건이 일치되면 액션이 수행
 - 제약(constraints) – 시스템이나 외부 요소가 수행할 또는 수행하지 않을 제약을 가하는 규칙
 - 계산(computation) – 공식이나 알고리즘

비즈니스 규칙

표 4.6 의료 정보 시스템의 비즈니스 규칙

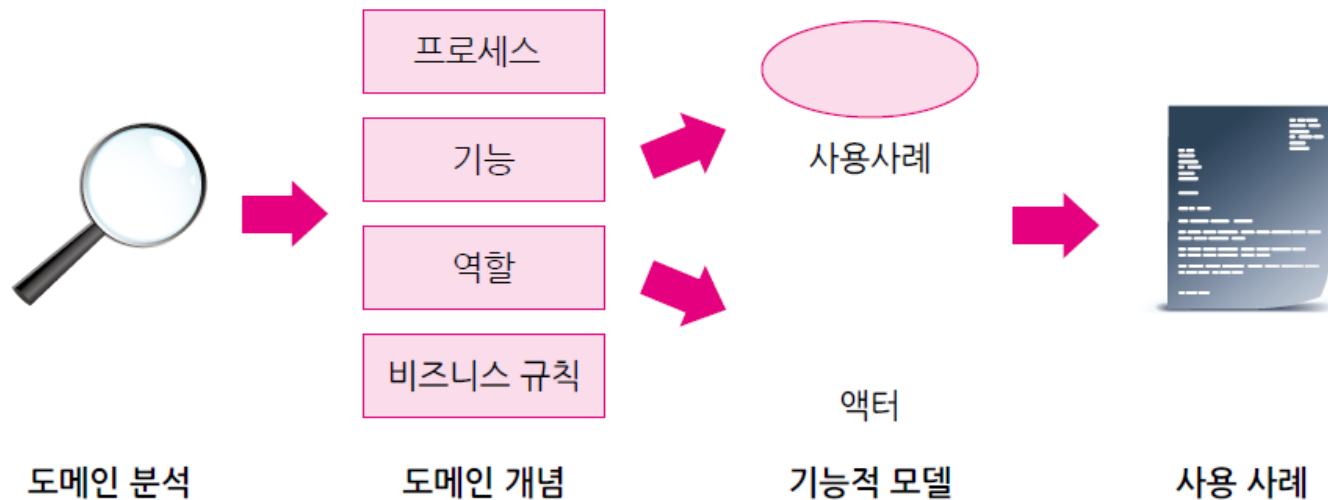
ID	정의	타입	소스
001	환자는 질병으로부터 고통받는 사람이며 의사나, 응급실, 다른 의료기관에서 진료를 의뢰받는다.	사실	도메인 사전
002	19세 이하의 환자는 보호자나 응급 센터와 동행하여야 한다.	제약	병원 정책
003	<p>예약할 때는 환자에 대한 다음 정보가 보관되어야 한다.</p> <p>ID 데이터</p> <p>004-01 이름</p> <p>004-02 주소</p> <p>004-03 전화번호1</p> <p>004-04 전화번호2</p> <p>004-05 주민등록번호</p> <p>004-06 의료카드 ID</p>	제약	도메인 전문가와 의 인터뷰
005	접수 직원은 환자의 개인 정보를 입력, 수정하고 필요하면 의료카드를 발행한다.	사실	도메인 사전
006	의료비 청구가 30일 이내에 수납되지 않으면 연체료를 물린다.	유추	병원 정책

시나리오 기반 분석

- 다양한 사람들이 참여하여 다양한 용어와 개념을 전달하여 요구를 도출
- 커뮤니케이션의 장벽 해소
 - 시나리오 기반(5W1H)
- 사용자 스토리
 - < 사용자/역할(who) >는
 - < 목표/혜택/이익(why) >를 얻기 위하여
 - < 행위/작업(what) >을 원한다.

4.4 유스케이스

- 도메인 분석과 모델링 사이의 관문
- 도메인 분석의 결과를 액터, 사용사례, 관계들로 구성된 시스템 명세로 매핑하는 작업



유스케이스 소개

- 유스케이스 요소

- 액터, 시스템 범위, 유스케이스, 관계

- 유스케이스 분석 과정

- 1) 액터 찾기

- 2) 유스케이스 찾기

- 3) 유스케이스 사이의 관계 찾기

} 유스케이스 다이어그램

- 유스케이스 명세

- 시스템이 제공하여야 할 서비스를 시간이 경과되는 순서로 정렬하여 기술한 것

유스케이스 다이어그램

- 시스템의 기능을 나타내기 위하여 사용자의 요구를 추출하고 분석하는데 사용
- 구성
 - 유스케이스(use case): 시스템 기능
 - 액터(actor): 시스템과 상호작용 하는 것(사용자, 시스템)

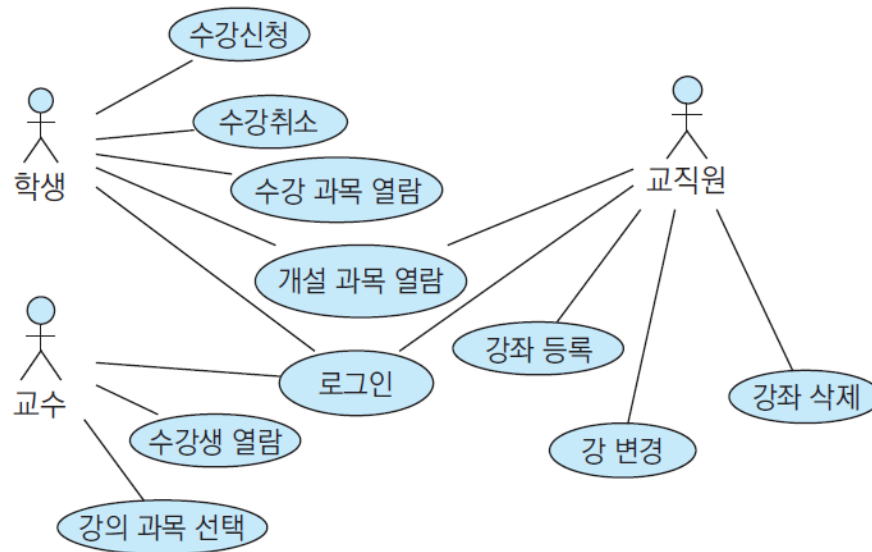
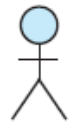


그림 4.11 유스케이스 다이어그램(수강 신청 시스템)

액터

- 액터
 - 시스템과 상호작용 하는 외부 엔티티
 - 구별되는 이름과 설명이 필요
- 액터가 될 수 있는 것
 - 사용자가 맡은 일
 - 다른 시스템



액터

액터는 시스템 밖에 있는 요소로 대상 시스템과 상호 작용하는 사람이나 다른 시스템에 의한 역할이다.

액터는 대상 시스템에게 서비스를 제공하거나 제공 받는다.

액터 찾기

- 액터를 찾기 위한 질문
 - 어떤 사용자 그룹이 작업을 수행하기 위하여 시스템의 지원을 받는가?
 - 어떤 사용자 그룹이 시스템의 주요기능을 사용하는가?
 - 어떤 사용자 그룹이 유지 보수와 관리 등의 부수적 기능을 사용하는가?
 - 시스템이 다른 외부 하드웨어나 소프트웨어 시스템과 동작하는가?

유스케이스 찾기

- 여러 개별 시나리오를 묶은 것
 - 정상적인 흐름
 - 오류, 예외 케이스
- 시나리오로부터 유스케이스 형성

계좌 이체

유스케이스

유스케이스는 시스템이 실행하는 일련의 동작으로 특정 액터에게 관찰 가능한 실행 결과를 제공한다.

유스케이스는 시나리오의 집합을 정의한 것이다.

유스케이스 찾기

- 개발자와 사용자가 함께 작성
- 현재의 응용 도메인에 대하여 기술한 여러 문서를 이용(지침서, 절차 매뉴얼 등)
- 필요한 질문
 - 시스템이 어떤 작업을 수행하기를 액터가 원하는가?
 - 액터가 원하는 정보는 무엇인가?
 - 누가 데이터를 생성하는가? 데이터는 조작, 삭제될 수 있는가? 이런 작업이 누구에 의하여 행해지는가?
 - 액터가 시스템에 정보를 알리는데 필요한 것은? 얼마나 자주 또 언제 이런 작업이 일어나는가?
 - 액터가 시스템으로부터 정보를 알아내는데 필요한 이벤트는? 이런 사건의 빈도는?

유스케이스 명세

시스템 제목	수강 신청 시스템
유스케이스 이름	수강 신청
액터	학생
시작 조건	학생이 재학 중이고 로그인 되어야 함.
기본 흐름	<ol style="list-style-type: none">1. 학생이 “수강 신청”을 선택한다.2. 시스템이 일정표 양식을 디스플레이한다.3. 시스템이 강좌 목록에서 수강 가능한 강의 리스트를 검색한다.4. 학생이 강의 리스트에서 꼭 수강하려는 4개의 강좌와 2개의 대안 강좌를 선택한다. 선택이 완료되면 ‘제출’버튼을 누른다.5. 선택된 각 강좌가 일정표에 추가된다.6. 시스템이 일정표를 저장한다.
대안 흐름	<p>3A</p> <ol style="list-style-type: none">1. 시스템이 강좌 목록을 접근할 수 없다면 일정한 횟수로 반복 시도하고 그래도 불가능하면 오류 메시지를 디스플레이한다.2. 학생이 오류 메시지를 인식하고 유스케이스를 마친다.
종료 조건	수강 신청된 강좌가 스케줄로 저장된다.

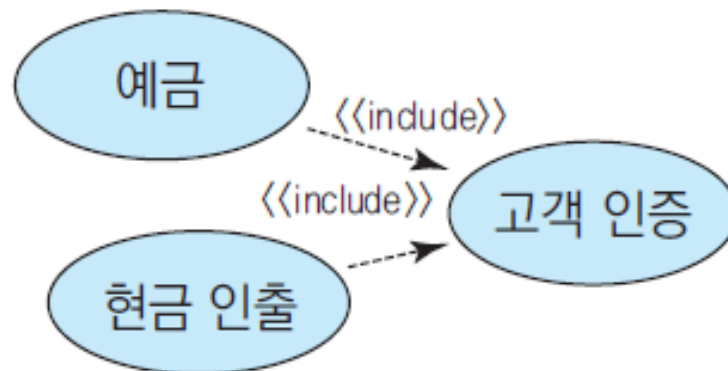
유스케이스 관계 찾기

- 관계를 이용하여 모형의 복잡도를 줄이고 이해도를 높인다.
- 관계 종류
 - 대안 흐름: 기본 유스케이스에서 이벤트의 흐름이 역간 변형되거나 선택, 예외인 경우
 - 포함 관계: 다른 유스케이스에서 재사용 할 수 있도록 캡슐화하려는 경우
 - 확장 관계: 기본 유스케이스의 기본 이벤트 흐름이 특정 조건에 만족되었을 때 분리 확장된 것

포함 관계

- 유스케이스 사이의 중복을 제거함
- 어떤 유스케이스가 다른 유스케이스를 포함하는 관계
- 공통된 동작을 떼어 낼 수 있다.

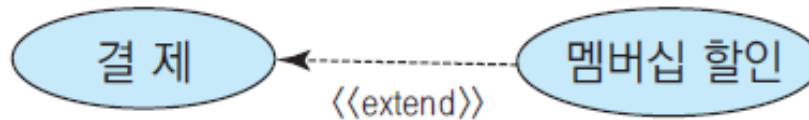
<예> 포함 관계의 예



확장 관계

- 유스케이스가 일정한 조건 아래 확장된 동작을 포함한다면 다른 유스케이스를 확장하는 관계에 있다.

<예> 결제 과정에 멤버십이 있는 경우 할인 적용



- 기본 유스케이스인 '결제'는 자체로 완전한 유스케이스
- 멤버십에 가입되었다는 조건에 만족하면 '멤버십 할인' 유스케이스가 삽입

4.5 요구 명세

- IEEE 830
- 요구분석명세서
- SRS
(Software Requirement Specification)

1. 개요

1.1 시스템의 목적

1.2 범위

1.3 정의, 약어

1.4 참조

2. 기능적 요구

2.1 외부 인터페이스 요구

2.1.1 사용자 인터페이스

2.1.2 하드웨어 인터페이스

2.1.3 소프트웨어 및 통신 인터페이스

2.2 기능 요구

2.2.1 기능 #1(사용 사례 #1)

2.2.2 기능 #2(사용 사례 #2)

...

3. 기타 요구 및 제약 사항

3.1 성능 요구(반응 시간, 처리 소요 시간, 처리율)

3.2 H/W 요구(기억 장치 규모, 통신수용도)

3.3 예외 조건 및 이의 처리

3.4 자원, 인력에 대한 제약 조건

4. 인수 조건

4.1 기능 시험 및 성능 시험

5. 참고 자료

작성 방법

- 사용자와 개발자간의 이해를 돕기위함
- Gilbert가 제안한 요구 분석명세서 작성시 주의 사항
 - 요구분석명세서는 사용자와 개발자 모두가 쉽게 이해할 수 있도록 써야 한다
 - 요구분석명세서에 기술된 조건은 개발자와 사용자가 모두 동의한 것이어야 한다.
 - 요구분석명세서는 목표 시스템에 의하여 수행될 모든 기능을 정확히 기술하여야 한다.
 - 요구분석명세서는 목표 시스템에 영향을 주는 모든 제약 조건을 기술한다.
 - 요구분석명세서는 시스템의 인수를 위한 테스트 기준을 제공하여야 한다.
 - 요구분석명세서는 원하는 시스템의 품질과 상대적인 중요도 및 품질을 측정하는 방법이 기술 되어야 한다.

요구 검증

검증 사항	설명
이해용이성(Comprehensibility)	요구 명세서를 읽을 때 요구의 의미를 잘 이해할 수 있는가?
중복(Redundancy)	필요없이 중복된 부분이 없는가?
완전성(completeness)	빠진 요구가 없는지, 요구를 기술하는데 빠진 정보가 없는가?
일관성(consistency)	요구사항이 서로 모순되지 않는가?
모호성(ambiguity)	요구분석의 내용이 모호함 없이 모든 참여자들에 의해 명확하게 이해될 수 있는가?
검증 가능성(verifiable)	요구분석 명세서에 기술된 내용이 사용자의 요구를 만족하는가? 개발된 시스템이 요구사항 분석 내용과 일치하는지를 검증할 수 있는가?
추적 가능성(traceable)	시스템 요구사항과 시스템 설계문서 및 구현과 매핑되어 추적할 수 있는가?

희망이 역경을 몰아냅니다

- 우리의 운명은 겨울철 과일 나무와 같다.
- 그 나뭇가지에 다시 푸른 잎이 나고 꽃이 필 것 같지 않아도
- 우리는 그것을 꿈꾸고, 그렇게 될 것을 잘 알고 있다.
- 요한 볼프강 폰 괴테
- 하늘이 무너져도 솟아날 구멍이 있다고 했습니다. 궁즉통(窮則通)입니다.
- 극단의 상황에 이르면 도리어 해결할 방법이 생기게 되어있습니다.
- 밤이 어두울수록 새벽이 가까운 법입니다.
- 역경은 희망에 의해 극복됩니다.

