
객체지향 소프트웨어공학

3 장 요구 분석

목 차

- 01 도메인 분석
- 02 문제 정의와 범위 설정
- 03 요구 추출
- 04 요구 추출 방법
- 05 사용 사례 분석
- 06 요구 문서화
- 07 요구 검토

요구 분석 과정

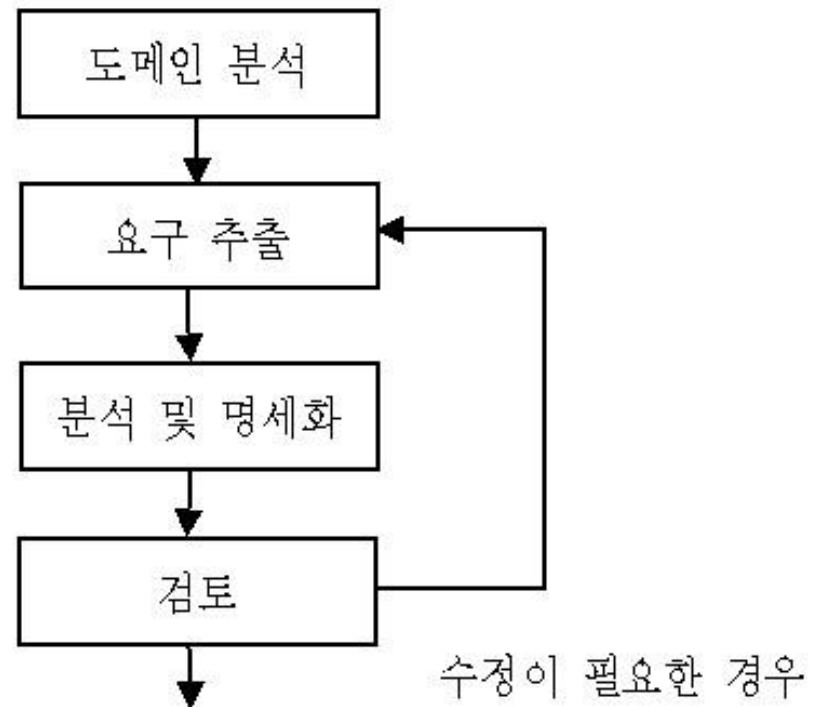
- ‘어떻게’보다는 ‘무엇을’에 관점을 두어야 함

문제의 배경과
성격, 범위를 파악

사용자가 소프트웨어에
대하여 무엇이 필요한지 도출

도출된 요구사항을 문서로 정리

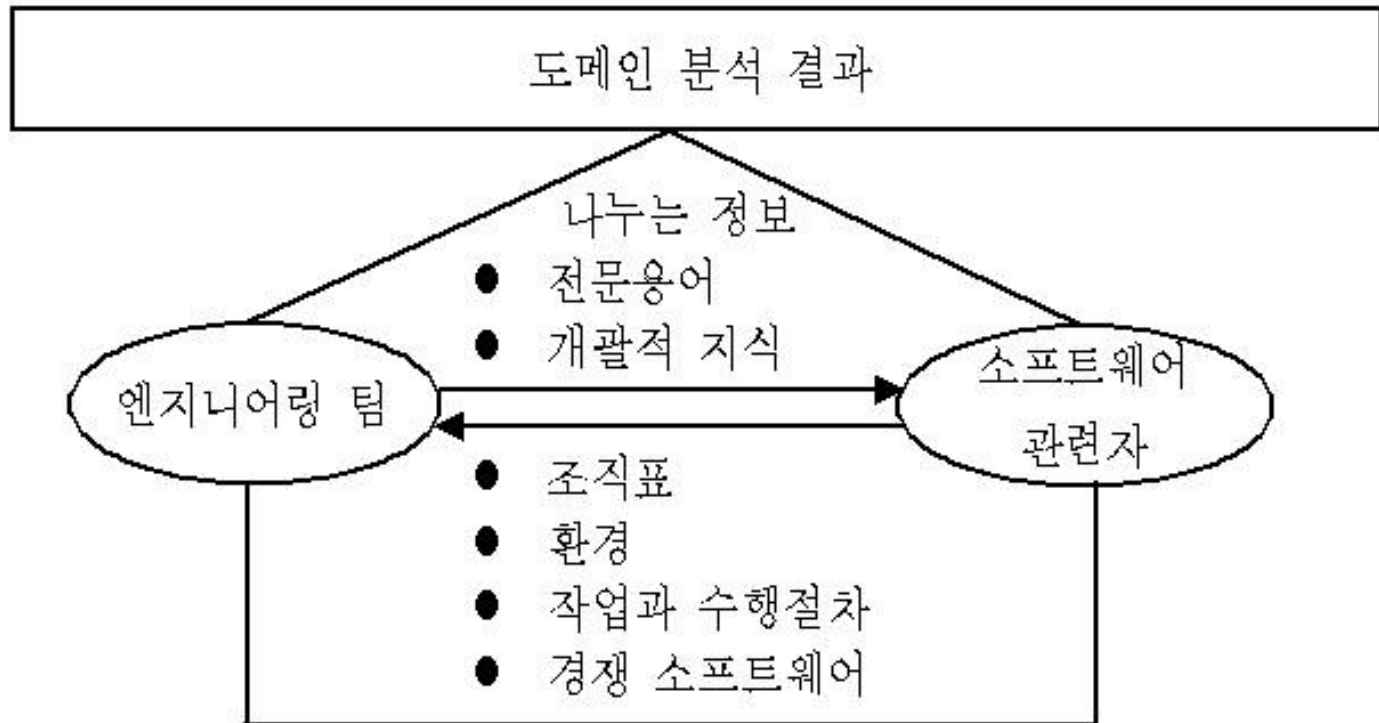
사용자가 요구하는 바로 그것
인지 체크



01 도메인 분석

- 소프트웨어 엔지니어가 문제를 더 잘 이해하기 위하여 도메인에 대하여 알아가는 과정
 - 도메인이란 소프트웨어를 사용할 것으로 예상되는 고객이 일하는 분야의 비즈니스나 기술
 - 도메인 전문가란 응용 분야에 깊이 있는 지식을 가진 사람
- 도메인 분석을 수행하는 이점
 - 빠른 개발
 - 더 좋은 시스템
 - 확장을 예견

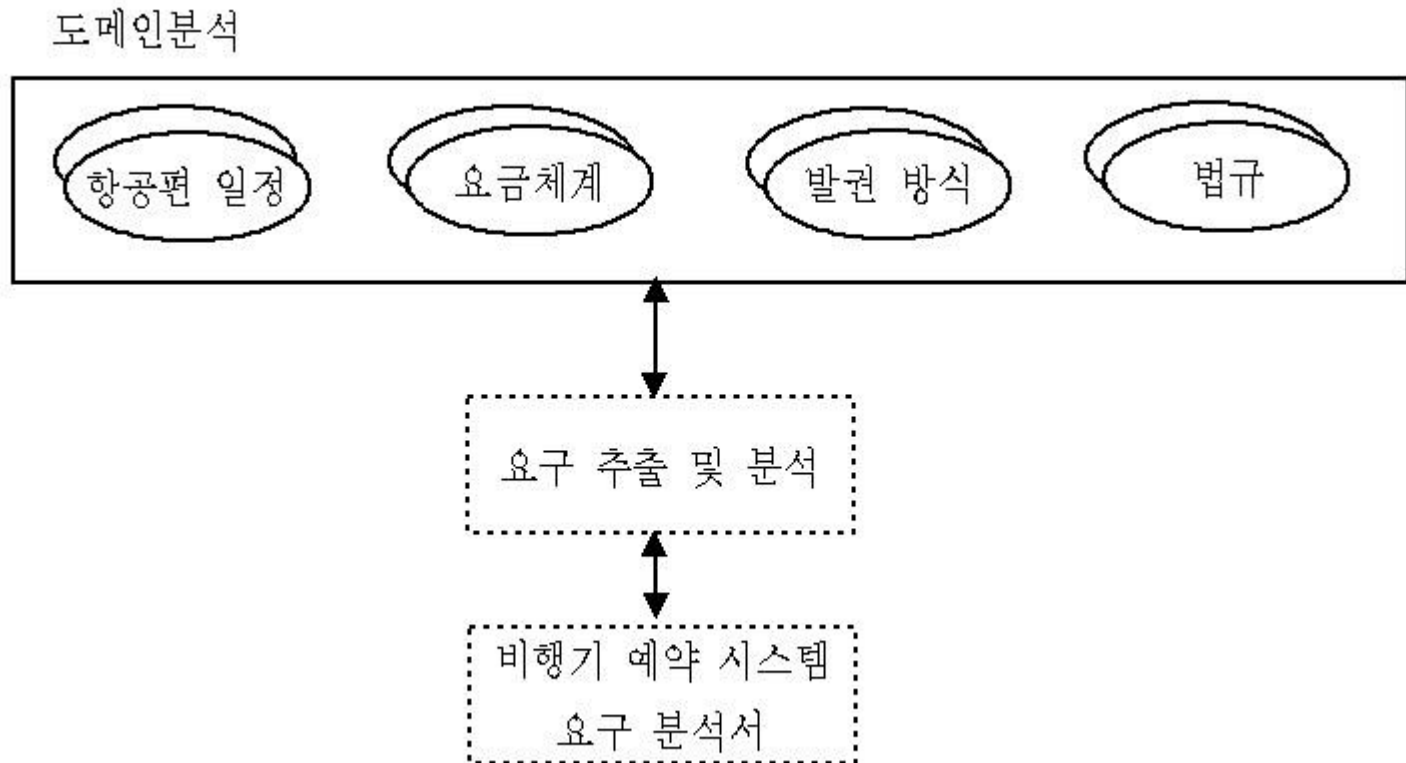
도메인 분석 과정



도메인 분석서

1. 개 요
2. 용 어
3. 개괄적 지식
4. 고객과 사용자
5. 환 경
6. 경쟁 소프트웨어
7. 도메인과 조직의 유사도

항공 예약 시스템의 도메인 분석



프로젝트 착수

요구가 아직
결정되지 않음 고객이 이미 요구를
결정함

신규 개발 -

A

B

현재 시스템
확장 -

C

D

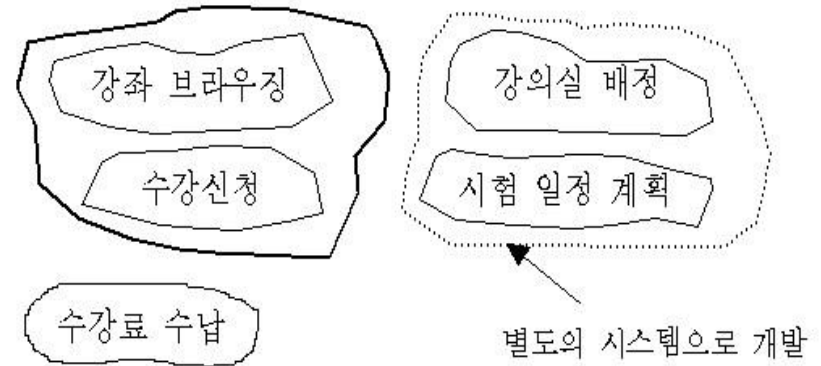
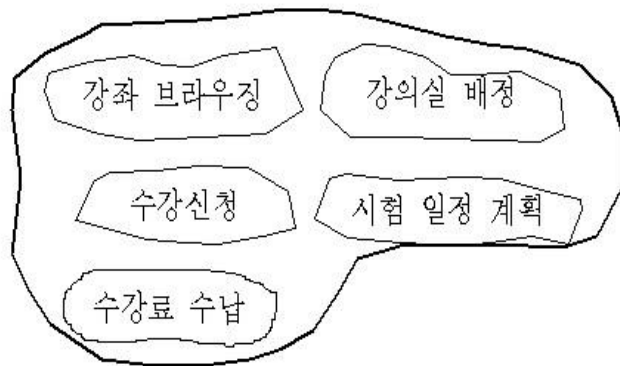
02 문제 정의와 범위설정

- 문제란?
 - 고객이나 사용자가 직면한 어려움
 - 생산성이나 매출을 높일 수 있는 기회
- 문제의 해결은 일반적으로 소프트웨어의 개발을 필요로 한다.
- 좋은 문제 정의는 짧고 간결

범위 설정

- 문제를 상세히 정의하여 범위를 축소
 - 시스템이 할 수 있는 모든 일을 상상하여 리스트 작성
 - 범위가 너무 넓으면 일부는 배제
 - 범위가 너무 좁으면 상위 목표를 설정

예) 대학 수강신청 시스템



03 요구 추출(requirement)

- 요구: 제안된 시스템이 무엇을 하는가를 나타낸 문장으로
고객의 문제가 적절히 해결되기 위하여 관련자들이
동의한 것
 - 짧고 간결한 정보
 - 시스템이 무엇을 하는가를 나타냄
 - 관련자 모두가 동의한 것
 - 고객의 문제를 적절히 해결한 것

요구를 모아놓은 것이 *요구 분석서*

요구의 유형

- 기능적 요구(functional requirements)
 - 시스템이 무엇을 하여야 하는지를 기술한 것
 - 예) 현금 인출, 잔금 조회, 계좌 이체, 현금 서비스 등의 외형적 기능
- 비기능적 요구(nonfunctional requirements)
 - 개발 과정에 고수하여 할 제약 조건
 - 예) 하드웨어 자원의 제약, 소프트웨어 품질 특성에 대한 수준

기능적 요구

- 시스템이 입력으로 무엇을 받아들이는가?
- 시스템이 무엇을 출력하는가?
- 시스템이 어떤 데이터를 다른 시스템이 사용할 수 있도록 저장하는가?
- 시스템에서 어떤 계산이 이루어지는가?
- 위 모든 사항들의 다이밍과 동기화

비기능적 요구

- 모든 사항이 검증 가능
- 세 가지 종류
- 1. 다음 사항을 반영하는 카테고리: 사용성, 효율성, 신뢰성, 유지보수성, 재사용성
 - 반응시간
 - 처리량
 - 자원 사용량
 - 신뢰도
 - 가용성
 - 고장에서의 회복
 - 유지보수와 확장의 허용
 - 재사용성의 허용

비기능적 요구

2. 시스템의 환경과 기술에 관한 카테고리

- 플랫폼
- 사용 기술

3. 프로젝트 계획과 개발 방법에 관한 카테고리

- 사용하는 개발 프로세스(방법론)
- 비용과 납기일
 - 시스템 개발 계약서나 별도의 프로젝트 계획서에 표시

04 요구 추출 방법

- 요구를 효과적으로 추출하고 분석하는 체계화된 기술

요구추출 방법	작업 방법	특 징
관 찰	사용자의 업무를 관찰하며 메모	감추어진 문제를 잘 드러냄
인터뷰	여러 관련 당사자를 만나 준비된 질문과 대답	정확한 요구추출 요구에 대한 오해를 줄일 수 있음
브레인스토밍	여러 사람이 모인 그룹에서 아이디어를 쏟아 놓는다	효과적인 정보 추출
프로토타이핑	시범적으로 시스템을 구현	요구에 대한 빠른 피드백
사용사례 분석	시스템 외부 기능 파악	체계적 요구 구성

관찰

- 관찰 방법
 - 문서를 읽고 사용자와 함께 요구에 대하여 논의한다
 - 잠재적인 사용자들이 수행하는 복잡한 일을 관찰
 - 사용자가 하는 일을 자세히 설명해 달라고 요구
 - 비디오 촬영
 - 예) 도매상에서 점원이 사려는 고객과 물건을 매매하는 과정
 - 시간이 많이 소요

인터뷰

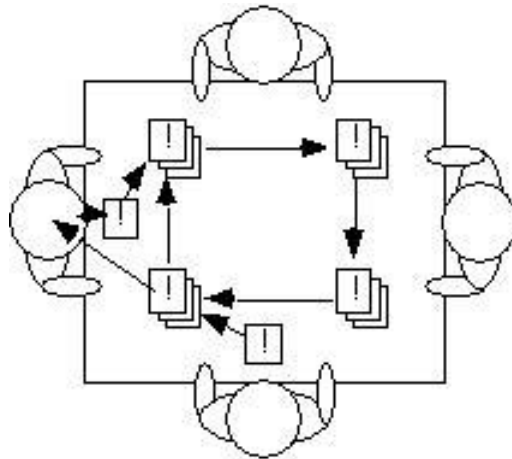
- 미리 잘 계획하여야 많은 정보를 얻을 수 있음
- 가능하면 많은 당사자와 인터뷰
- 관련자 이외의 다른 사람도 인터뷰
 - 경쟁 제품의 사용자, 마케팅 담당자 등
- 여유 시간 할애

인터뷰

- 여러 번의 인터뷰 준비
 - 구체적인 설명 요구
 - 최대, 최소, 예외 규칙, 예상 되는 변동 등
 - 관련자의 시스템에 대한 미래 비전
 - 어떤 융통성을 가져야 하는지 제안될 수도 있음
 - 다른 아이디어가 있는지 질문
 - 최소한의 허용 가능한 솔루션이 무엇인지 질문
 - 다른 정보원이 없는지 질의
 - 다이어그램 작성을 요구

브레인스토밍

- 아이디어를 낼 목적으로 여러 명으로부터 정보를 얻기 위한 회의
- 훈련된 요원이 주재
- 토론보다는 아이디어를 쏟아놓는 회의, 익명성 보장
- 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

05 사용사례 분석

- 개발한 소프트웨어를 가지고 사용자가 무엇을 할 수 있는지를 분석하는 체계적 방법
- 사용사례 분석
 - 시스템을 사용할 수 있는 사용자의 부류(actor)를 결정
 - 액터가 시스템을 사용할 필요가 있는 작업을 결정
- 액터 찾기
 - 서비스 받는 사용자
 - 연결된 외부 시스템
 - 시스템과 관련된 외부 요소(센서, 장치 등)

사용사례

- 일반적으로 사용사례는 작업의 시작부터 끝까지 전 단계를 커버하여야 함
- 사용자가 시스템과의 상호작용을 나타냄
 - 시스템이 실행하는 계산이 아님
- 사용사례는 특정 사용자 인터페이스 설계와 독립적으로 작성되어야 함
- 액터가 컴퓨터와 상호작용 하는 액션만을 포함하여야 함

사례: 도서관 시스템

- 대출자
 - 제목으로 검색
 - 저자로 검색
 - 주제로 검색
 - 다른 사람이 대출한 책을 예약
 - 대출자의 개인 정보와 대출한 책의 리스트를 체크
- 대출 직원
 - 대출자가 책을 대출
 - 반납된 책을 반환
 - 책의 대출 기간을 연장
 - 지불된 연체료를 기록

사례: 도서관 시스템

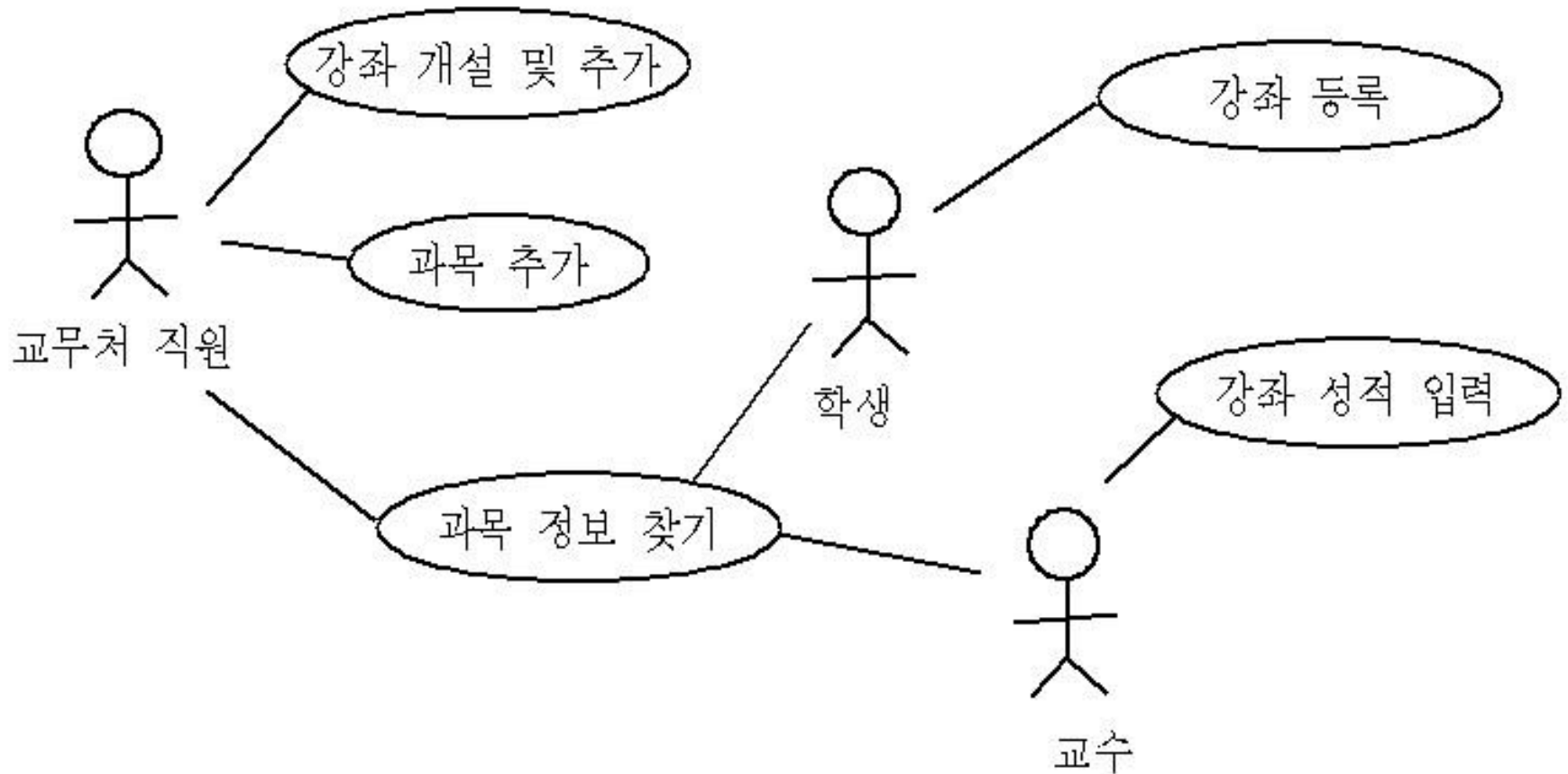
- 대출자를 추가
- 대출자의 정보, 주소, 전화번호 등을 갱신
- 사서
 - 새 책을 도서관 장서로 추가
 - 책을 도서관 장서에서 삭제
 - 책에 대하여 시스템이 기록한 정보를 변경
- 회계 시스템(별개로 작동됨)
 - 대출자가 지불한 연체료를 전달

사용사례 다이어그램

사용사례는 사용자가 주어진 작업을 완료하기 위하여 수행한 일련의 액션이나 이벤트

- 사용 사례를 분석하는 목적은 시스템을 모델링 하는 것
 - 특정 목적을 달성하려고 할 때
 - 사용자가 시스템과 어떻게 상호작용 하는지에 대한 관점으로
- 사용사례 모델은
 - 사용사례 집합과
 - 사용사례에 대한 설명, 사용사례 사이의 관계로 구성

사용사례 다이어그램



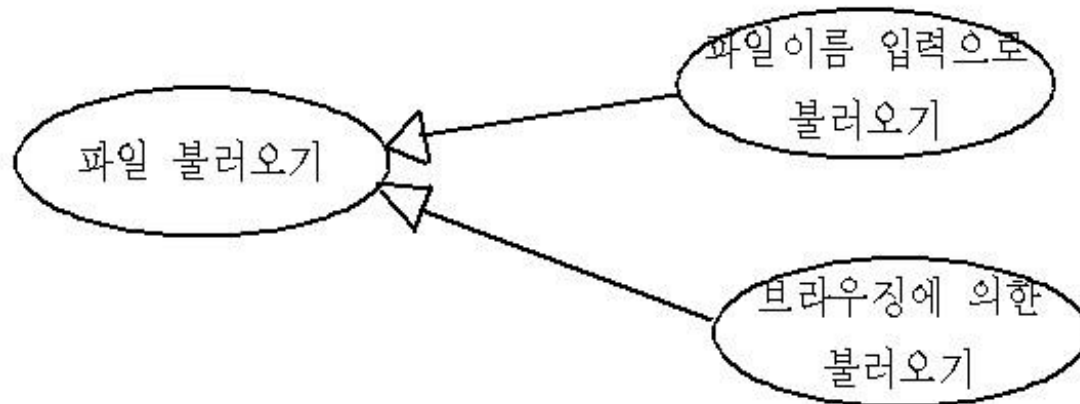
확장(extension)

- 선택적인 인터랙션을 명시적으로 나타낼 때 또는 예외적인 사례를 다룰 때 사용
- 사용사례 확장을 분리함으로써 기본적인 사용사례의 표현이 간단해 진다.
- 사용사례 확장도 사용사례의 처음부터 끝까지 모든 단계를 나열하여야 함
 - 특수한 경우의 처리도 포함



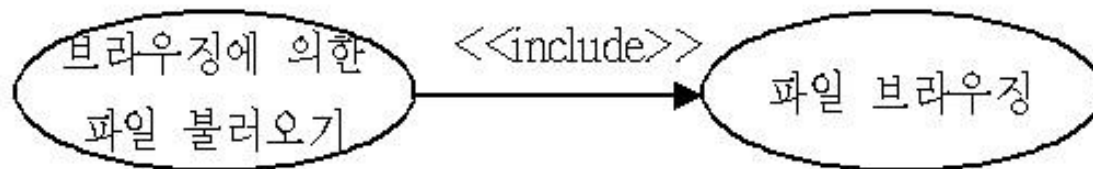
일반화(generalization)

- 클래스 다이어그램에서 슈퍼 클래스와 유사
- 일반화된 사용사례는 여러 유사 사용사례를 표현
- 상세화 된 여러 사용사례가 유사 사용사례의 상세한 내용을 제공

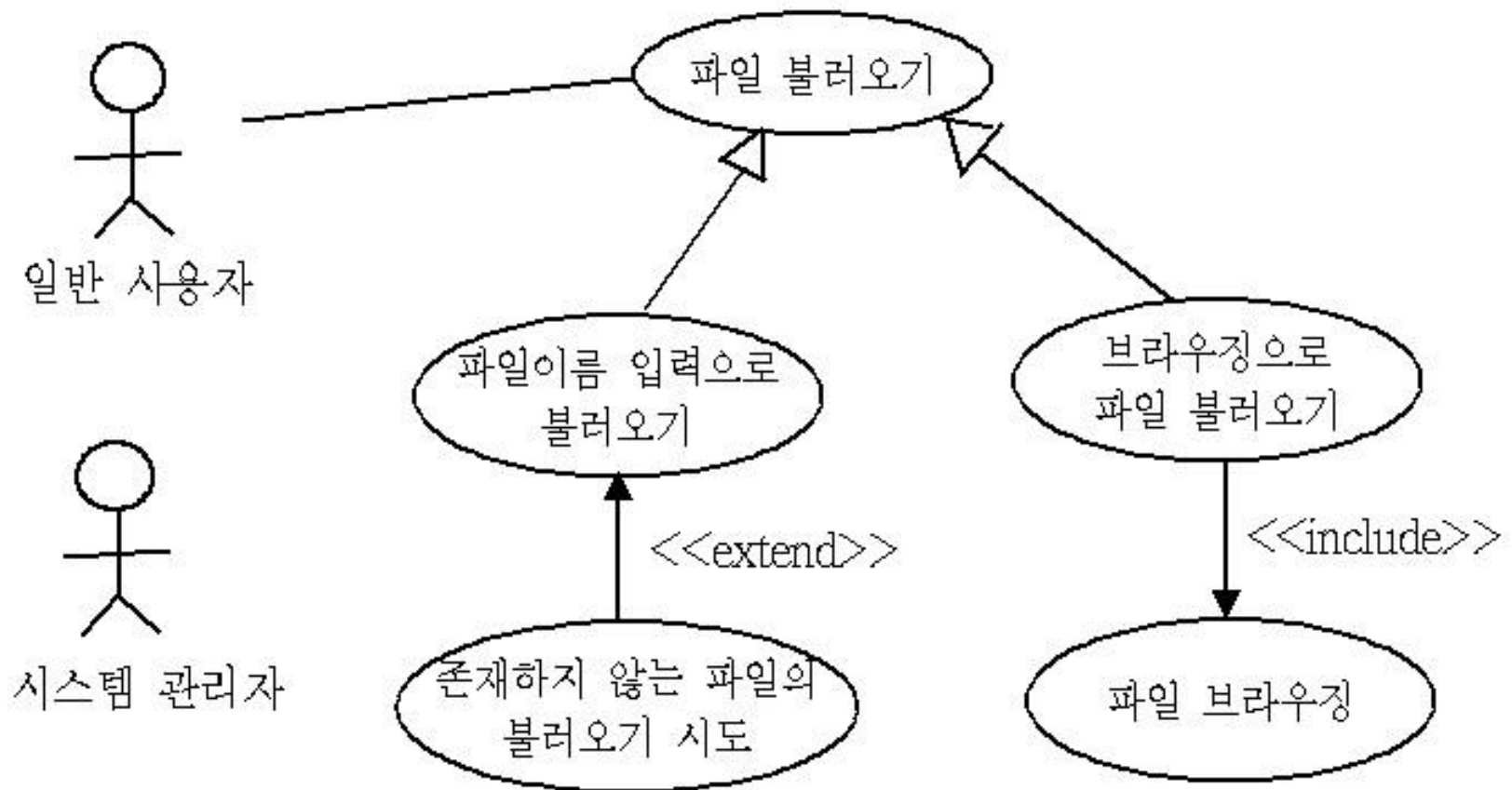


포함(inclusion)

- 여러 사용사례들 사이에 공통적인 부분을 표현
- 다른 사용사례들 안에
 - 일련의 액션을 공유
 - 다수의 사용사례 사이에 중복을 피함
- 하위 수준의 작업의 수행을 하위 수준의 목표로 표현



사용사례 다이어그램 예



사용사례 작성 예...

사용사례 : 파일 불러오기

관련 사용사례: 다음 두 사용사례를 일반화 한 것임

- 파일 이름을 주고 불러오기
- 브라우징으로 불러오기

사건의 흐름:

액터측 액션

1. ‘불러오기....’ 명령을 선택

3. 파일 이름을 명시

4. 선택을 확인

시스템측 반응

2. ‘파일 불러오기’ 다이얼로그를 창을 디스플레이

5. 디스플레이에서 다이얼로그 창을 삭제

사용사례 작성 예...

사용사례 : 파일 이름을 주고 불러오기

관련 사용사례: 파일 불러오기의 상세화

사건의 흐름:

액터측 액션

1. ‘불러오기....’ 명령을 선택

3a. 텍스트 필드를 선택

3b. 파일 이름을 입력

4. ‘불러오기’를 클릭

시스템측 반응

2. ‘파일 불러오기’ 다이얼로그 창을
디스플레이

5. 디스플레이에서 다이얼로그를 삭제

사용사례 작성 예

사용사례 : 브라우징으로 파일 불러오기(include)

관련 사용사례:

- 파일 불러오기 사용사례의 상세화
- 파일 브라우징 사용사례를 포함

사건의 흐름:

액터측 액션

시스템측 반응

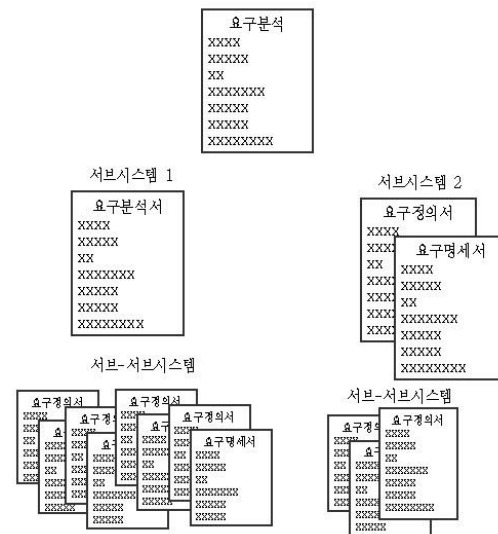
- | | |
|----------------------|------------------------------|
| 1. ‘불러오기....’ 명령을 선택 | 2. ‘파일 불러오기’ 다이얼로그를 창을 디스플레이 |
| 3. 파일을 브라우징 | |
| 4. 선택을 확인
창을 삭제 | 5. 디스플레이에서 다이얼로그 |

사용사례의 장점

- 시스템의 범위를 정하는데 도움이 됨
- 개발 과정을 계획하는데도 사용됨
- 요구를 개발하고 검증하는데 사용 됨
- 테스트케이스를 정의하는데 기초가 됨
- 사용자 매뉴얼 구성하는데 사용될 수 있음

06 요구 문서화

- 두 가지 극단적 형태
 - 몇 문단의 글이나 다이어그램으로 요구의 아웃라인을 정의한 것
 - 요구 정의(requirement definition)
 - 수천 페이지의 복잡하고 자세한 명세 리스트
 - 요구 명세서(requirement specification)
- 대규모 시스템을 위한 요구 문서는 계층적으로 정리



요구분석 상세 수준

- 얼마나 자세히 기술하여야 하는가는 다음 사항을 고려하여 결정
 - 시스템의 크기
 - 다른 시스템에 대한 인터페이스 요구
 - 목표로 하는 독자
 - 개발을 위한 계약
 - 요구 취합 단계
 - 도메인 및 기술에 대한 경험 수준
 - 잘못된 요구에 대한 비용

요구분석서

1. 문제
2. 배경 지식
3. 환경 및 시스템 모델
4. 기능적 요구
5. 비기능적 요구

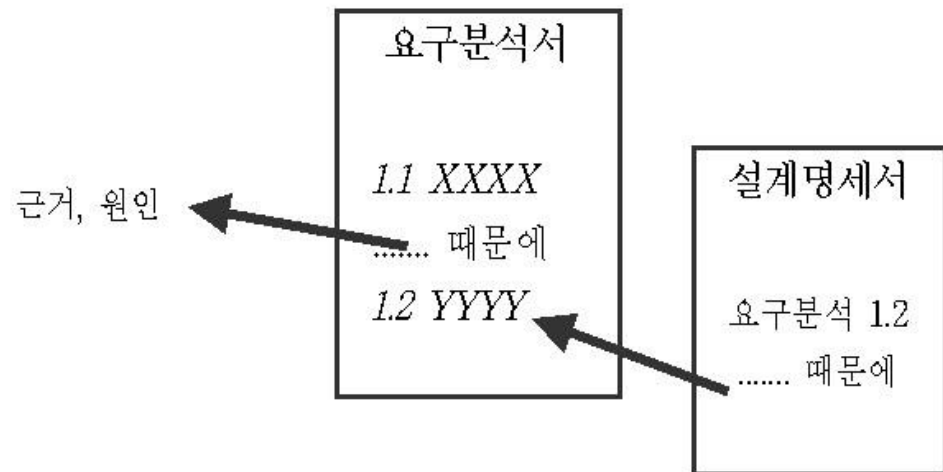
07 요구 검토

- 각 요구 사항이 다음을 만족하였는지
 - 개발 비용을 능가하는 이익이 있는지
 - 현재 당면한 문제가 해결되었는지
 - 명확하고 일관성 있는 표현이 되었는지
 - 모호한 점이 없는지
 - 논리적으로 일관성이 있는지
 - 시스템의 충분한 품질을 유도하는지
 - 가용 자원으로 실현가능한지
 - 검증 가능한지
 - 식별 가능한지
 - 설계를 과다하게 제한하고 있지 않는지

요구 문서는

- 문서는
 - 충분히 완벽하여야 하며
 - 조직화 하고
 - 명료하고
 - 관련자에 의하여 동의되어야

- 추적 가능성



요구분석서의 변경관리

- 요구는 다음 이유로 계속 바뀜
 - 비즈니스 프로세스의 변경
 - 기술의 변경
 - 문제 이해의 향상
- 요구분석은 중단될 수 없음
 - 고객과 사용자와 계속 대화
 - 변화의 이익은 비용을 초과
 - 작은 변화는(UI) 적은 비용으로 빠르고 신속히 가능
 - 대규모 변화는 신중하게 접근
 - 예상하지 않은 변화를 부분적인 시스템으로 구축하면 설계가 부실해지고 완성이 늦어질 수 있음
 - 어떤 변경은 겉만 향상
 - 시스템의 규모를 키우는 것은 피하고 오직 더 좋은 시스템이 되도록 노력