

## 4주 2강

# 요구 분석 절차와 표현방법



# 이번 주차에는...

## 소프트웨어 요구 분석

- 요구 분석 절차와 요구 사항 종류
- 요구 사항의 표현

# 1. 요구 분석 절차와 요구 사항 종류

## ▪ 요구 분석 절차



- ① 자료 수집: 현행 시스템 파악, 실무 담당자와 인터뷰, 현재 사용하는 서류 검토
- ② 요구 사항 도출: 수집한 자료 정리 및 분류    개발에 반영할 요구 사항 도출
- ③ 문서화: 요구 분석 명세서 작성
- ④ 검증: 요구 분석 명세서 검토    모순 사항, 빠뜨린 사항 등 점검

## 2. 요구 사항 분류

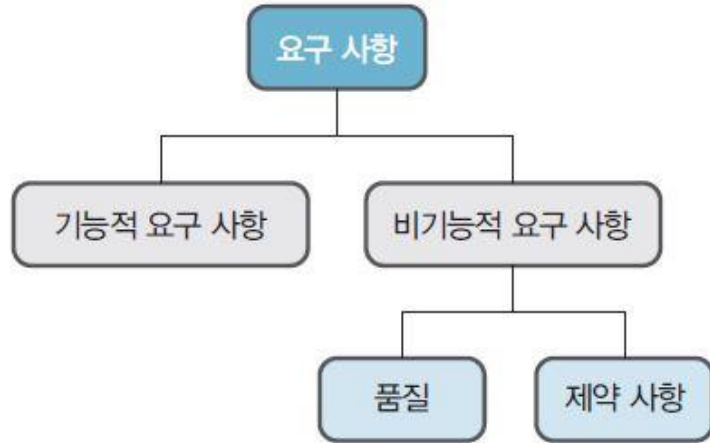


그림 4-5 요구 사항 분류



### 3. 비기능적 요구 사항

- 비기능적 요구 사항
  - 수행 가능한 환경, 품질, 제약 사항
- 제약 사항의 예
  - ① 자바 언어를 사용해 개발하고, CBD 개발 방법론을 적용해야 한다.
  - ② 레드햇 리눅스 엔터프라이즈 버전에서 실행해야 한다.
  - ③ 웹로직 서버WebLogic Server를 미들웨어로 사용해야 한다.
  - ④ 윈도우즈 운영체제와 리눅스 운영체제에서 모두 실행할 수 있어야 한다.
- 품질
  - 신뢰성, 성능, 보안성, 안전성, 사용성

## 4. 품질(1)

### ■ 신뢰성reliability

- 신뢰: 소프트웨어를 믿고 사용할 수 있는 것
  - 사용자가 주어진 시간과 환경에서 고장 없이 사용할 수 있어야 한다는 것
- 신뢰도: 장애 없이 동작하는 시간의 비율
  - (예) 신뢰도 98%(0.98): 100번 수행했을 때 오류 없이 동작하는 회수가 98번

### ■ 신뢰도 측정

- 고장 간 평균 시간(MTBF)과 이용 가능성(가용성)을 척도로 사용
- $MTBF = MTTF + MTTR$ 
  - MTBF고장 간 평균 시간, Mean Time Between Failure : 고장에서 다음 고장까지의 평균 시간
  - MTTF평균 실패 시간, Mean Time To Failure : 수리한 후 다음 고장까지의 평균 시간
  - MTTR평균 수리 시간, Mean Time To Repair : 고장 발생 시점에서 수리 시까지의 평균 시간
- 이용 가능성availability =  $MTTF / (MTTF + MTTR) \times 100\%$ 
  - 이용 가능성(가용성): 주어진 시점에서 프로그램이 요구에 따라 작동되고 있을 가능성

## 5. 품질(2)

### ■ 성능performance

- 사용자가 시스템에 어떤 요구를 했을 때 해당 기능을 정상적으로 수행하는 것은 물론, 사용자가 원하는 조건(응답 시간, 데이터의 처리량 등)을 만족시키는 것

(예 1) 대학 종합정보시스템은 수강 신청 시 동시 접속자수 10,000명은 가능해야 한다.

(예 2) 도서 관리 시스템에서 사용자가 책을 검색한 결과를 2초 이내로 보여주어야 한다.

### ■ 보안성security

- 인증을 받지 않은 사람이 시스템에 접근하는 것을 처음부터 막아 시스템과 데이터를 보호

### ■ 안전성safety

- 작동하는 모든 시스템이 소프트웨어 오류로 인해 인명 피해가 발생하지 않도록

(예 1) 공장의 생산 라인에 사람 손이나 물체가 인식되면 절단 작업이 중단되어야 한다.

(예 2) 전동차 출발 시 문에 가방 같은 물체가 끼면 출발을 멈춰야 한다.

## 6. 품질(3)

### ■ 사용성 usability

- 소프트웨어를 사용할 때 혼란스러워하거나 사용하는 순간에 고민하지 않게 하는 편의성
- 의미가 같은 단어 두 개를 혼용하여 사용자를 혼란스럽게 하는 경우
- 주민번호를 입력할 때 ‘-’를 넣어야 할지 말아야 할지 명확하게 제시해주지 못하는 경우



## 7. 사용자 요구 사항과 시스템 요구 사항

### ■ 사용자를 위한 도면

- 건축주와 일반인에게 설계 의도를 정확히 전달할 목적으로 작성된 도면
- 사용자가 알고 싶어하는 내용을 중심으로 작성
- 사용자 요구 분석 명세서 또는 요구 사항 정의서

### ■ 시공자를 위한 도면

- 전기 배선 기술자가 전기선을 포설하기 위해 필요한 도면
- 각 분야의 기술자가 작업하는 데 필요한 도면
- 실시 설계도: 사용자용 기본 설계도를 바탕으로 실제로 집을 짓는 데 필요해 작성한 도면
- 시스템 요구 분석 명세서에 해당

## 8. 사용자 요구 사항

- 사용자 요구 분석 명세서
  - 사용자 요구 사항을 정리하여 작성한 문서
  - 건축의 사용자를 위한 도면과 유사
  - 목적: 사용자와 대화 시 거부감을 줄이고 충분히 이해할 수 있도록 쉽게 작성
  - 방법: 사용자가 이해할 수 있도록 전문 용어보다는 쉬운 용어 사용
  - 도구: 다이어그램 사용
  - 사용자와 분석가가 서로 충분한 대화를 나누며 함께 작성
- 사용자 요구 분석 명세서 작성 방법
  - 유사한 프로젝트 경험을 가진 분석가 선정
  - 표준 양식 사용
  - 수집한 요구 사항에 대한 근거(출처) 마련

## 9. 시스템 요구 사항

### ■ 시스템 분석 명세서

- 시스템 요구 사항을 설계하는데 도움이 되도록 기술적 용어나 전문적 표현 사용하여 작성
- 건축에서 시공자를 위한 도면과 유사
- 설계를 위해 사용되므로 완전하고 일관성 있게 작성

### ■ 사용 도구

- 구조적 방법론의 구조적 언어
- 객체지향 방법론의 유스케이스 다이어그램
- 검증에 강한 Z 명세와 같은 정형화된 수학적 명세 언어

# 10. 요구 사항의 표현

- 표현

- 음악(작곡, 작사, 노래), 미술(그림, 조각), 아이들의 그림, 수학

- 모델

- 정의: 어떤 복잡한 대상의 핵심 특징만 선별하여 일정한 관점으로 단순화시켜 기호나 그림 등을 사용해 체계적으로 표현한 것
  - 악보, 수학 공식, 모델 하우스, 플라스틱 자동차나 로봇, DNA분자 모델 등

- 모델의 필요성

- 모델을 통해 실제 모습을 생각하고 확인해볼 수 있기 때문
    - (예) 모델하우스, 장난감 자동차, DNA 분자 모델

# 11. 모델

## ■ 건축의 모델

- 하나의 사물을 여러 관점에서 바라보는 건축의 도면
- 조감도: 건물 위에서 바라본 모습
- 평면도: 건물을 수평으로 절단했을 때 바라본 모습
- 배치도: 건물과 부지의 위치 관계를 나타낸 모습
- 배선도: 전기 공사를 위해 전기선의 연결 관계를 나타낸 모습

## ■ 소프트웨어 개발에서의 모델

- 여러 설계 도면을 보고 건물을 시공하는 것처럼 SW 개발 시에도 여러 관점의 모델 사용
- UML의 다양한 다이어그램을 통해 소프트웨어의 범위나 개략적인 구조와 기능을 이해

## 12. SW에서 모델 사용의 장단점

- 장점

- 개발될 소프트웨어에 대한 이해도 향상, 이해 당사자 간의 의사소통 향상
- 유지보수 용이

- 단점

- 과도한 문서 작업으로 인한 일정 지연 가능성
- 형식적인 산출물로 전락할 가능성

# 1. 모델링

- 모델링

- 모델을 제작하는 과정 또는 작업
- 현실 세계를 단순화하여 표현하는 기법

(예 1) 연필로 종이에 간단한 콘티를 작성하는 것

(예 2) 스토리보드같이 자기가 표현하고 싶은 것을 문장으로 서술하는 것

(예 3) 오션지에 음표를 그리는 것

- 소프트웨어 개발에서의 모델링

- UML 다이어그램을 이용하여 표현

(예 1) 요구 사항 표현 시 UML의 유스케이스 다이어그램 사용

(예 2) 자연어 사용

(예 3) 형식 언어의 형식적 표기법 사용

# 13. 모델링 언어

- 모델링 언어

- 애매모호한 표현 등의 문제점 해결하기 위해 모델링을 할 때 사용하는 기호, 표기법, 도구  
(예) 악보 기호, 수학 기호, UML 다이어그램, Z 언어

- 개발 방법론에 따른 모델링 언어

- 구조적 방법론: DFD<sup>Data Flow Diagram</sup>, DD<sup>Data Dictionary</sup>, 프로세스 명세
- 정보공학 방법론: DB 설계 시 표현은 ERD<sup>Entity-Relationship Diagram</sup>
- 객체지향 방법론: UML 표기법



# 14. DFD

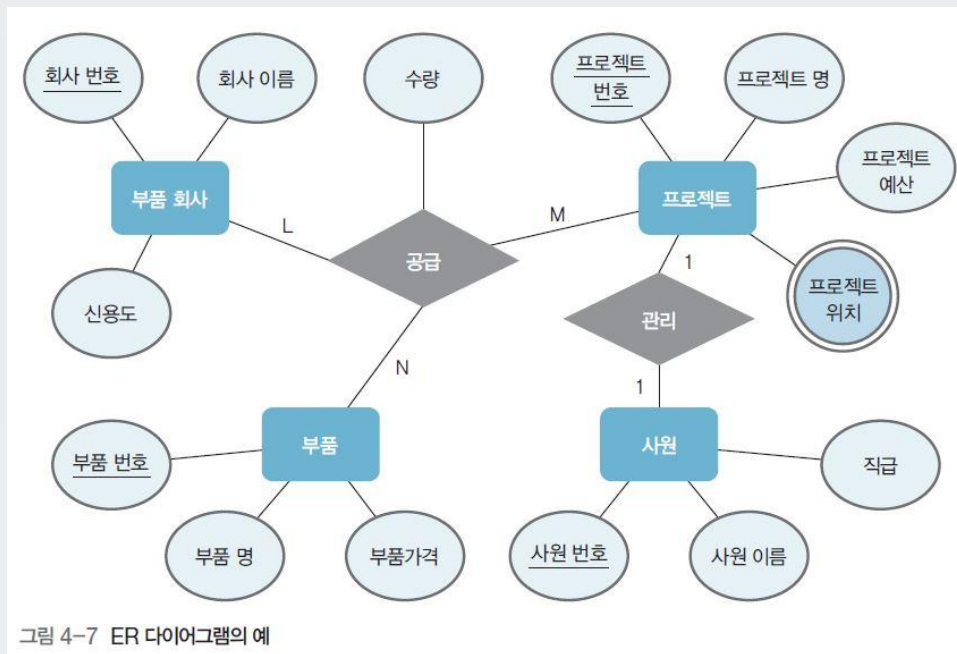


- Terminator: 출원지, 목적지를 나타냄(학생, 교수)
- data flow: 자료의 흐름을 나타냄(화살표)
- data store: 자료가 저장되는 곳을 나타냄(데이터베이스(테이블))
- Process: 자료를 입력 받아 처리하는 알고리즘(성적 계산)

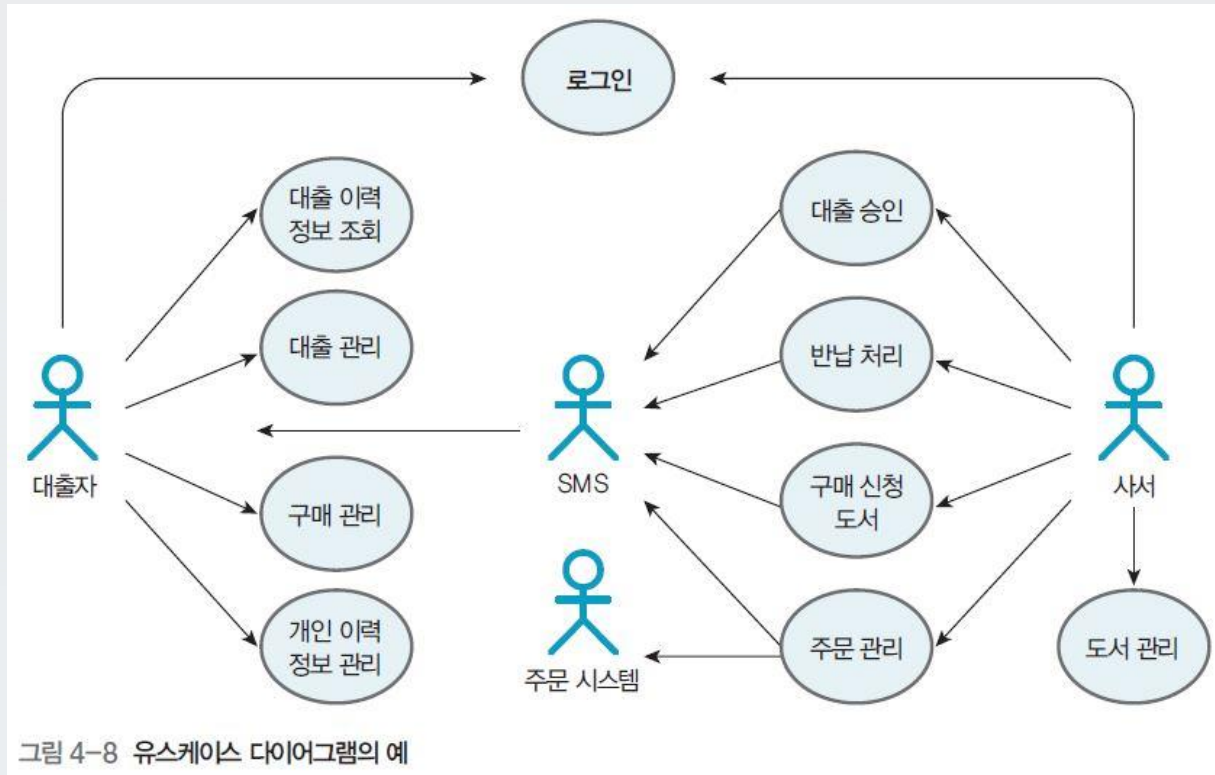
# 15. ERD

## ■ ER 다이어그램

- 데이터베이스에 저장할 데이터를 개체entity와 관계relationship를 중심으로 작성



# 16. 유스케이스 다이어그램(1)



# 17. 유스케이스 다이어그램(2)

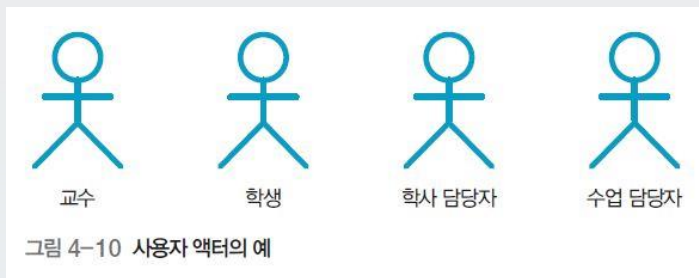
- 유스케이스
  - 사용자의 요구를 나타내는 기능
  - 실제로 코딩할 수 있을 만큼의 가장 작은 단위의 기능



# 18. 유스케이스 다이어그램(3)

## ■ 사용자 액터

- 시스템을 사용하는 사람(역할)



## ■ 시스템 액터

- 본 시스템과 데이터를 주고받는 등 서로 연동되는 또 다른 시스템
- 해당 프로젝트의 개발 범위에 속하지 않고 이미 다른 프로젝트에서 개발된 시스템  
(예) 대학 종합정보시스템에서 등록납부 현황 자료를 받아오는 은행 시스템  
(예) 입학생들의 기초 자료를 교육부로부터 받아오는 교육부시스템



다음 시간

# 요구사항의 문서화



송실사이버대학교

송실사이버대학교의 강의콘텐츠는  
저작권법에 의하여 보호를 받는바, 무단  
전재, 배포, 전송, 대여 등을 금합니다.

\* 사용서체 : 나눔글꼴