

객체지향소프트웨어공학

8주차-3 : UML 기반 시스템 분석 설계

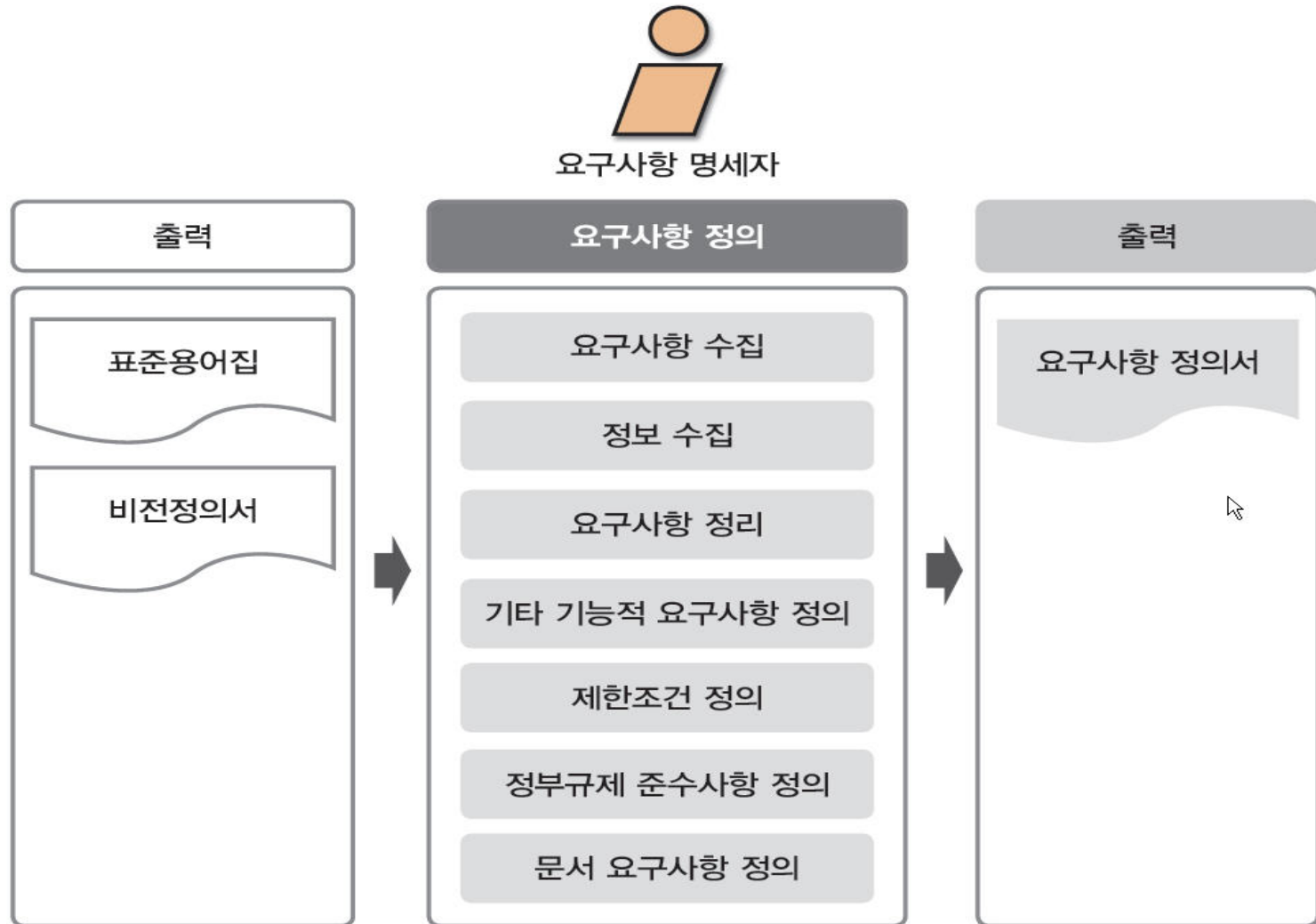
교재#2 : I-05. 요구사항 정의

1. 요구사항 정의의 의의
2. 유스케이스 모델 작성
3. 유스케이스 관련 명세서
4. 유스케이스 다이어그램의 관계 표현
5. 한국 IT 대학 유스케이스 모델
6. 관련 산출물 사례 및 작성 지침
7. 요약

5장 목차

1. 요구사항 정의의 이해
2. 유스케이스 모델 작성 절차 이해
3. 유스케이스 관련 명세서 작성방법 설명
4. 유스케이스 다이어그램의 관계 표현 방법 설명
5. 한국 IT 대학 유스케이스 모델 사례 설명
6. 관련 산출물 사례 및 작성 지침 설명 및 이해
7. 교재 P49~52에 대한 상세 설명 및 사례 설명

- 요구사항 정의 워크플로우의 주요 활동



■ 요구사항 정의 워크플로우 상세

• 요구사항 수집

- 조직도 및 업무분장표를 참조하여 관련 조직 및 사용자를 파악하고 면담계획 수립
- 요구사항 수집을 위한 원천을 결정하고 우선순위 부여
- 우선순위에 따라 자료조사를 위한 방법 및 이에 따른 계획 수립
- 요구사항 수집을 지원할 수 있는 조직 및 사용자를 식별하고, 파악된 사용자를 대상으로 면담, 설문 또는 워크숍 계획 수립
- 사용자 식별 방법
 - 시스템의 사용자는 누구인가?
 - 경제적인 측면에서 시스템의 구매자는 누구인가?
 - 시스템이 산출하는 결과물에 영향을 받는 사람은 누구인가?
 - 납품되어 설치될 때 시스템을 평가하게 될 사람은 누구인가?
 - 시스템을 필요로 하는 어떤 다른 내.외부 사용자는 누구인가?
 - 시스템을 유지보수 하게 될 사람은 누구인가?
 - 그 외 다른 관련자는 없는가?

■ 요구사항 정의 워크플로우 상세

• 정보 수집

- 사전에 수립된 계획에 따라 사용자와의 면담, 설문, 워크숍 등을 실시하거나 자료 수집
- 현행 업무 수행 시 사용하는 화면, 보고서 및 서식을 조사 방법
 - 면담계획에 따라 사용자와의 면담, 설문 또는 워크숍 실시
 - 자료조사 계획에 따라 사용자 조직의 자료를 수집
 - 현행 시스템이 존재하는 경우에는 이와 관련된 문서 수집

• 요구사항 정리

- 요구사항을 분류하고 정리
- 요구사항에 대한 우선순위 부여

■ 요구사항 정의 워크플로우 상세

- 기타 기능적 요구사항 정의
 - 특정 유스케이스에 적용할 수 없는 기능적 요구사항
 - 보고, 감사, 인쇄, 보안, 지원/집행 라이선스 부여, 인증 요구사항 등
 - 이러한 기능적 요구사항은 보충 명세서에 문서화
 - 상당한 수의 시스템 전반의 기능적 요구사항이 있는 경우 이 섹션의 조직 고려
 - “FURPS+” 모델
 - 요구사항을 기능성(F : Functionality)
 - 유용성(U : Usability)
 - 신뢰성(Reliability)
 - 성능(P : Performance)
 - 지원성(S : Supportability)
 - 기타(+)로 분할하여 식별

■ 요구사항 정의 워크플로우 상세

- 시스템 품질 정의

- 비기능적 요구사항에는 품질과 제한조건이 모두 포함

- 품질에 대해 논의

- 품질은 “FURPS+” 모델에서 “URPS”로 표시

- 가용성(Usability)

- 사용자 인터페이스의 미적 요소 및 일관성 표현

- 하부 항목 구성

- » 일반 사용자 및 고급 사용자가 특정 오퍼레이션에서 생산적이 되기 위한 필수 훈련 시간

- » 일반 작업에 대한 측정 가능한 작업 시간

- » 공통 사용성 표준(예: IBM의 CUA 표준 또는 Microsoft의 GUI 표준)을 준수하기 위한 요구사항

■ 요구사항 정의 워크플로우 상세

- 시스템 품질 정의

- 신뢰성(Reliability)

- 시스템 계산의 정확성 및 시스템의 장애에서 복구하는 능력 표현

- 하부 항목 구성

- » 가용성 : 사용 가능한 시간의 백분율(xx.xx%), 사용 시간, 유지보수 액세스, 성능 저하 모드 오퍼레이션 등 지정

- » 평균 실패 시간 간격(MTBF) : 대개 시간 단위로 지정되지만 일, 월 또는 년의 항목으로 지정 가능

- » 평균 복구 시간(MTTR) : 시스템이 실패한 후 동작하지 않을 수 있도록 허용되는 시간 지정

- » 정확성 : 시스템 산출물에서 요구되는 정밀도(분해능) 및 정확성(몇 가지 알려진 표준에 의한) 지정

- » 최대 오류 또는 결함 비율 : 일반적으로 오류/KLOC(1000행의 코드) 또는 오류/기능 점수의 항목으로 표현

- » 오류 또는 결함 비율 : 사소, 중요 및 위험 오류의 항목으로 분류

■ 요구사항 정의 워크플로우 상세

- 시스템 품질 정의

- 성능(Performance)

- 처리량, 응답 시간, 복구 시간, 시작 시간 및 종료 시간 능력 표현

- 하부 항목 구성

- » 트랜잭션의 응답 시간(평균, 최대)

- » 처리량(예: 초당 트랜잭션)

- » 용량(예: 고객 수 또는 시스템이 처리하는 트랜잭션 수)

- » 성능 저하 모드(시스템이 어떤 방식으로 성능 저하되는 경우 허용 가능한 오퍼레이션 모드)

- » 자원 사용: 메모리, 디스크, 통신 등

- » 성능 요구사항을 문서화할 때 반드시 특정 응답 시간을 포함하고 해당되는 경우 관련 유스케이스를 이름으로 참조

■ 요구사항 정의 워크플로우 상세

- 시스템 품질 정의
 - 지원성(Supportability) : 시험 용이성, 적응성, 유지보수성, 호환성, 구성 가능성, 설치 용이성, 확장성 및 현지화 용이성 능력 표현
 - 하부 항목 구성
 - 코딩 표준
 - 이름 지정 규칙
 - 클래스 라이브러리
 - 유지보수 액세스
 - 유지보수 유틸리티

■ 요구사항 정의 워크플로우 상세

- 제한 조건 정의
 - 빌드되는 시스템에 대한 모든 디자인 제한조건 문서화
 - 제한조건
 - 솔루션 제공에 있어서 갖고 있는 자유도에 대한 제한
 - “FURPS+”에서 ‘+’로 표시되며 추가로 분류 가능
 - 디자인 제한조건
 - » 시스템 구조화 또는 디자인을 위한 옵션을 지정하거나 제한
 - 구현 제한조건
 - » 시스템의 코드 또는 구현/구축(Construction)을 지정 또는 제한
 - 인터페이스 제한조건
 - » 시스템이 상호작용해야 하는 외부 항목 지정
 - » 그런 상호작용 내에서 사용되는 형식 또는 기타 요소 제한
 - 실제 제한조건
 - » 시스템을 내장하는 데 사용되는 하드웨어에 부과되는 실제 제한조건 지정

■ 요구사항 정의 워크플로우 상세

• 제한 조건 정의

- 시스템에 대해 문서화해야 하는 제한조건의 수에 따라서 각 제한조건 유형에 대한 하위 섹션을 제공 가능

- 요구사항이 써드파티 컴포넌트의 구입을 포함하는 경우

- 요구사항이 특정 인터페이스 요구사항을 포함하는 경우 위의사항

- 사용자 인터페이스의 경우 소프트웨어가 구현할 사용자 인터페이스 설명

- 하드웨어 인터페이스의 경우 논리 구조, 실제 주소, 예상 동작 등 포함

- 소프트웨어 인터페이스의 경우 소프트웨어 시스템의 다른 컴포넌트에 대한 인터페이스의 설명 포함

- 개발하려고 하는 시스템의 행위 중 기능적인 사항들
 - 유스케이스 모델을 통해 문서화
 - 유스케이스 모델은 액터, 유스케이스, 액터와 유스케이스의 관계 설명
 - 액터와 유스케이스 기호



액터



유스케이스

- 유스케이스 모델의 가장 중요한 역할은 의사소통
 - 고객, 사용자, 개발자들이 시스템의 기능과 행위에 대해서 토론할 때 사용되는 좋은 도구

■ 액터(Actor)

- 사용자
 - 인간뿐만 아니라, 시스템과 상호작용을 하는 시스템도 포함
- 개발되는 시스템과 상호작용하는 어떤 사람 또는 제안된 시스템의 외부에 존재하는 다른 시스템과 같은 것들
- 자신의 목적을 달성하기 위해서, 시스템과 상호작용하는 어떤 사람 또는 어떤 것의 역할
- 시스템과 상호작용하는 형태에 따른 액터의 분류
 - 시스템에 정보를 입력하기만 하는 액터
 - 시스템으로부터 정보를 제공받기만 하는 액터
 - 시스템에게 정보를 입력하고, 시스템으로부터 정보를 제공받는 액터

■ 액터(Actor)

• 액터를 찾는 데 도움을 줄 질문

- 어떠한 요구사항에 관심을 가지는 사람은 누구인가?
- 조직 안에서 시스템이 사용되는 곳이 어디인가?
- 누가 이 시스템을 사용함으로써 이익을 얻는가?
- 누가 시스템에 이 정보를 제공하고 사용하고 제거시키는가?
- 누가 시스템을 관리하고 지원할 것인가?
- 시스템은 외부 자원을 사용하는가?
- 한 사람이 여러 역할을 하는가?
- 여러 사람이 같은 역할을 하는가?
- 시스템이 기존 시스템과 상호 작용하는가?

■ 액터(Actor)

- 비전 문서의 작성
 - 이해당사자의 리스트와 이해당사자의 니즈(Needs)가 포함
 - 비즈니스와 관련된 이해당사자들의 대표적인 예
 - 고객, 공급자, 시스템 개발 의뢰자, 동종의 업무를 수행하는 타 회사, 개발 시스템과 관련을 맺는 시스템들
- 액터를 찾을 때, 너무 포괄적이거나 너무 상세하게 찾는 경우의 예
 - “복학생과 신입생이 같은 학생인가 아니면 다른 학생인가?”
 - 액터를 지나치게 세분화하는 경우로써 사람들이 수행하는 모든 역할에 따라 액터를 만드는 것

■ 유스케이스(Use-case)

- 액터와 시스템 간의 대화
- 시스템에 의해 제공되는 기능
- 유스케이스의 형식적인 정의
 - 특정 액터를 위해 측정 가능한 가치를 결과로서 제공하기 위해 시스템에 의해 수행되는 일련의 트랜잭션
- 시스템의 유스케이스를 찾기 위한 질문
 - 각 액터의 업무는 무엇인가?
 - 어떤 액터가 시스템에서 정보를 생성, 저장, 변경, 제거, 읽을 것인가?
 - 어떤 유스케이스가 이 정보를 생성, 저장, 변경, 제거, 읽을 것인가?
 - 어떤 액터가 갑작스러운 외부변화에 대한 정보를 시스템에게 알릴 필요가 있는가?
 - 시스템 내의 어떤 현상에 대해 통지를 받을 필요가 있는 액터는?
 - 어떤 유스케이스가 시스템을 지원하고 관리할 것인가?
 - 모든 기능적 요구사항이 유스케이스에 의해 수행될 수 있는가?

■ 유스케이스(Use-case)

- 유스케이스의 유용성
 - 유스케이스는 일반적으로 시작으로부터 끝까지가 하나로 완료되는 하나의 중요 기능을 나타내야 함
 - 하나의 유스케이스는 반드시 액터에게 어떤 가치를 제공해 주어야 함
 - 서로 다르기는 하지만 함께 속하는 것으로 보이는 기능을 묶어 주는 것이 좋음
- 유스케이스를 찾는 방식
 - bottom-up
 - top-down

■ 유스케이스(Use-case)

• 액터를 통해서 유스케이스를 찾는 방식

- 시스템에 관련된 이해당사자들을 찾음
- 이해당사자들의 니즈(Needs)를 찾고, 니즈를 통해 피쳐(Features)를 찾음
- 이해당사자들 중, 시스템을 직접적으로 사용하는 이해당사자들을 사용자로 전환
- 사용자가 시스템을 사용하는 경우를 목록화
- 사용자의 시스템을 사용하는 경우를 가지고 액터 정의
- 액터의 정의에서 목적에 해당하는 부분을 유스케이스로 바꿈

■ 스토리보드(Story board)

- 시스템 사용을 가시적으로 표현
- 유스케이스 스토리보드를 사용하는 사람들
 - 유저 인터페이스 설계 담당자
 - 유저 인터페이스의 모델을 생성하기 위해
 - 설계 담당자
 - 유스케이스 스토리보드에 있는 경계 객체의 설계 담당자
 - 유스케이스 안에 객체의 역할을 이해하고 객체들이 어떻게 상호작용하는지를 이해하기 위해
 - 설계 담당자는 경계 객체를 설계하고 구현하기 위해서 정보를 사용.
 - 시스템의 차기 버전 설계자
 - 시스템이 경계 객체에 의하여 사건 흐름을 어떻게 처리하는지 이해하기 위해
 - 테스터
 - 시스템의 유스케이스를 시험하기 위해 사용
 - 관리자
 - 분석과 설계 작업을 계획하고 추적하기 위해 사용

■ 스토리보드(Story board)

- 스토리보드의 작성
 - 벤치마킹을 통해서 기본적인 형태 작성
 - 이를 통해 사용자와 대화를 시작
 - 실제 데이터를 넣어 보면서 사용자와 대화
 - 스토리보드에 사용되는 정보에 대한 제약사항을 찾고 표시
- 스토리보드에 의한 제약 사항 표시



로그인
로그인이 필요한 서비스입니다. 로그인 해주세요.

☐ 아이디 기억 ☐ 보안접속

아이디

비밀번호

[가입](#) | [아이디 찾기](#) | [비밀번호 찾기](#)

한글	영문 필드명	유효값	입력 조건
아이디	ID	영문 12자리 이하	필수
비밀번호	Password	영/숫자 8자리 이상	필수
기능	시스템을 사용하고자 하는 교직원 및 학생들이 시스템에 대한 사용 권한을 인증 받는다.		

■ 스토리보드(Story board)

- 유스케이스 사건 흐름
 - 각 유스케이스는 사건 흐름에 의해 문서화
 - 유스케이스에 요구된 행위를 달성하기 위해 필요한 사건에 대한 설명
 - 사건 흐름의 포함 사항
 - 유스케이스의 시작과 종료 시기와 방법
 - 액터와 시스템의 상호작용
 - 유스케이스에 의해 요구되는 데이터
 - 유스케이스에 대한 정상적인 사건 흐름
 - 대안 혹은 예외적 흐름에 대한 설명

■ 스토리보드(Story board)

- 유스케이스 사건 흐름

유스케이스 기술서 양식

x <유스케이스 이름> 유스케이스 사건 흐름

x.1 선행조건(**Preconditions**)

x.2 메인 플로우(**Main Flow**)

x.3 서브 플로우(**Sub Flow**)

x.4 대안 플로우(**Alternative Flows**)

- 유스케이스 기술서(use case description)
 - 사건 흐름 문서를 만들기 위한 표준 템플릿의 일반적인 형태
 - x는 각각의 유스케이스마다 붙여지는 일련 번호
 - 일반적으로 유스케이스 기술서는 워드 프로세스를 사용하여 작성
 - 대부분의 UML 모델링 도구들은 유스케이스와 유스케이스 기술서와 같은 문서들을 상호 링크하여 관리 할 수 있는 기능 제공

■ 관계(Relation)

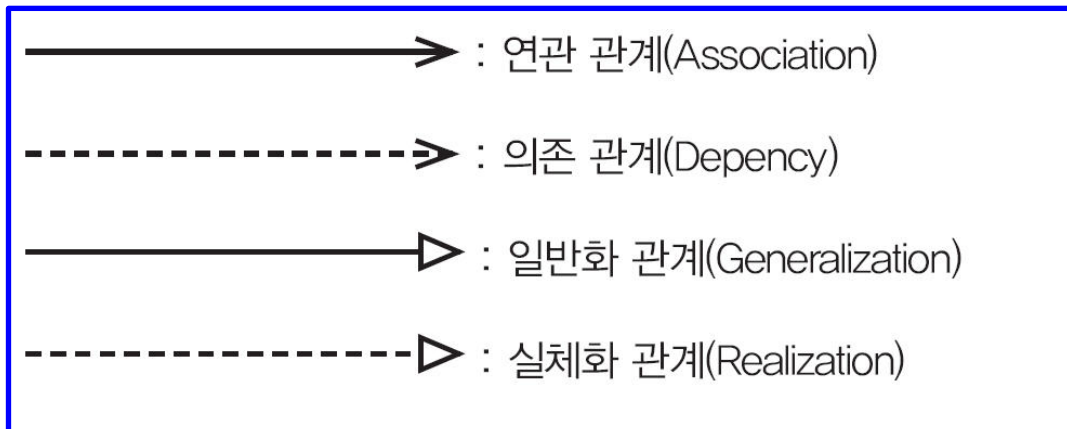
- 액터와 유스케이스 간에는 연관 관계들이 존재
 - 양방향으로 진행(navigate)되거나 단방향으로만 진행 가능
- 유스케이스와 액터의 관계



- 연관의 진행 방향은 누가 커뮤니케이션을 시작하는지 표현

■ 관계(Relation)

- 관계 기호와 의미



- 유스케이스 간에 존재할 수 있는 관계
 - 포함(include)과 확장(extend)
- 각각의 유스케이스마다 작성되는 것보다 분리해서 작성하는 것이 더욱 효율적
- 주의할 점
 - “유스케이스 상호 간에는 연관 관계를 정의할 수 없다”

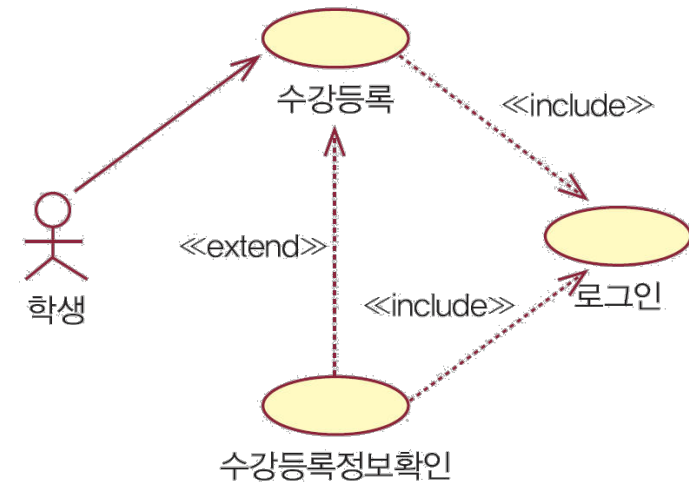
■ 관계(Relation)

- 확장 관계의 사용
 - 선택적 행위
 - 알람을 울리는 것처럼, 특정 상황에서만 수행되는 행위
 - 액터의 선택에 의해 수행되는 몇 가지 상이한 흐름

- 스테레오타입(Stereotype, 기본)
 - 새로운 요소를 만들기 위해 기본 모델링 요소들에 대한 의미의 확장 제공
 - 스테레오타입의 이름
 - <<>>안에 쓰임
 - 스테레오타입에 포함 관계(include)를 정의하는 경우
 - 여러 유스케이스들 사이에 공유되는 기능 표현
 - 향후 재활용하고자 활용
 - 스테레오타입에 확장 관계(extend)를 정의하는 경우
 - 유스케이스의 부수적이고 선택적인 동작을 묘사하기 위해 사용

■ 스테레오타입을 적용한 유스케이스 관계

- 스테레오타입에 의한 관계 확장
 - 수강등록 유스케이스는 학생 액터에 의해서만이 서비스 개시 가능
 - 수강등록 유스케이스에 의해 서비스된 정보나 데이터는 학생 액터에게 전달
 - 수강등록 유스케이스의 흐름에는 반드시 로그인 유스케이스를 활용
 - 경우에 따라서는 수강등록정보확인 유스케이스가 수강등록을 위해 수행 가능
 - 수강등록정보확인 유스케이스의 흐름에는 반드시 로그인 유스케이스 활용



- 액터 찾기
 - 문제 정의서

문제 정의서

학생은 수강등록을 하고자 할 때와 수강등록이 완료된 후, 자신의 등록정보를 확인하기 위해서 시스템을 사용한다. 교수는 해당 학기에 가르칠 강좌를 개설하고, 수강등록이 완료된 후, 자신의 개설강좌에 등록한 학생들의 정보를 확인하기 위해서 시스템을 사용한다. 강등록관리자는 신규 학생이나 교수에 대한 정보를 등록하고, 학생정보나 교수정보의 변경내용을 수정하고 삭제할 때와 수강등록기간이 종료되었을 때, 시스템에게 수강신청완료를 알리기 위해서 시스템을 사용한다. 지불시스템은 수강신청기간이 종료되어 수강등록시스템으로부터 수강신청 정보를 제공받을 때와 수강등록기간이 종료되어 지불 정보를 수강등록시스템에게 제공할 때 시스템을 사용한다.

■ 액터 찾기

- 문제 정의서에 대해 액터를 찾기 위하여 앞서 제시된 질문을 적용하여 답을 획득
 - 학생들은 개설강좌에 등록하기 원함
 - 교수는 가르칠 과목을 선택 (강좌개설) 하기 원함
 - 등록 담당자는 커리큘럼을 만들고, 학기에 해당되는 카탈로그(종합시간표와 같은 것)를 만듦
 - 등록 담당자는 과목, 교수, 학생에 대한 모든 정보 관리
 - 등록금과금시스템은 지불 정보를 시스템으로부터 받고 등록금 지불정보 제공
- 위 답을 기준으로, 학생, 교수, 등록 담당자, 등록금과금시스템을 액터로 결정 가능

■ 액터 찾기

- 이해당사자의 시스템 사용 리스트를 기반으로 액터에 간략한 기술
 - 학생 : 시스템을 사용하는 사람
 - 수강을 원하는 과목을 신청함
 - 수강등록을 하기 위해서 수강등록이 완료되었을 때, 자신의 수강등록 정보 확인
 - 교수 : 시스템을 사용하는 사람
 - 해당 학기에 가르칠 과목을 선택
 - 강좌를 개설하기 위해서 수강등록이 완료되었을 때, 자신의 개설강좌에 등록한 학생정보 확인
 - 등록관리자 : 시스템을 사용하는 사람
 - 신규 학생정보, 교수정보나 개설과목 정보를 입력함으로써 학생정보, 교수정보나 개설과목 정보 등록
 - 원하는 학생정보, 교수정보, 개설과목 정보를 선택하고 수정하거나 삭제하기 위해서, 수강신청완료를 시스템에 알리기
 - 등록금과금시스템
 - 수강신청정보 제공
 - 지불정보 제공

■ 유스케이스 찾기

- 액터에 대한 간략한 기술에서 목적 부분 도출
- 유스케이스
 - 수강등록
 - 수강등록 정보 확인
 - 강좌 개설
 - 개설강좌 정보 확인
 - 교수 정보 관리 학생 정보 관리
 - 개설과목 정보 관리

■ 유스케이스 사건 흐름 작성 사례

1. 강좌개설 유스케이스의 사건 흐름

1.1 액터

1차 액터 : 교수

2차 액터 : 없음

1.2 선행조건(Preconditions)

교수가 강좌개설을 선택하고 로그인 함으로 시작한다.

1.3 사건 흐름(Event Flow)

1) 액터는 승인정보 (id, password)를 입력하고 시스템에게 승인을 요청한다.

2) 시스템은 액터가 입력한 승인정보가 유효하면 해당 학기의 개설과목 리스트와 액터가 선택한 개설강좌 리스트를 제공한다.

[강좌개설 요청 시]

3) 액터는 개설과목 리스트에서 개설을 요청할 과목을 선택하고, 시스템에게 강좌개설을 요청한다.

4) 시스템은 요청한 과목을 개설강좌 리스트에 추가하고, 액터에게 갱신된 개설강좌 리스트를 제공한다.

[강좌개설 취소 시]

5) 액터는 개설강좌 리스트에서 취소할 강좌를 선택하고 시스템에게 강좌개설 취소를 요청한다.

6) 시스템은 요청한 과목을 개설강좌 리스트에서 삭제하고, 액터에게 갱신된 개설강좌 리스트를 제공한다.

[시스템 -강좌개설 마감일이 되었을 시]

7) 시스템은 학사일정에 강좌개설이 마감되었음을 표시한다.

1.4 대안플로우(Alternative Flows)

A-2-1 : 입력한 승인정보가 오류가 난 경우 재입력 받는다.

A-2-2 : 입력한 승인정보가 회원 DB에 존재하지 않는 경우 회원 가입 화면을 활성화한다.

A-5-1 : 대학원의 연구 과목의 경우 지도교수의 허가를 얻은 후 취소할수 있다.

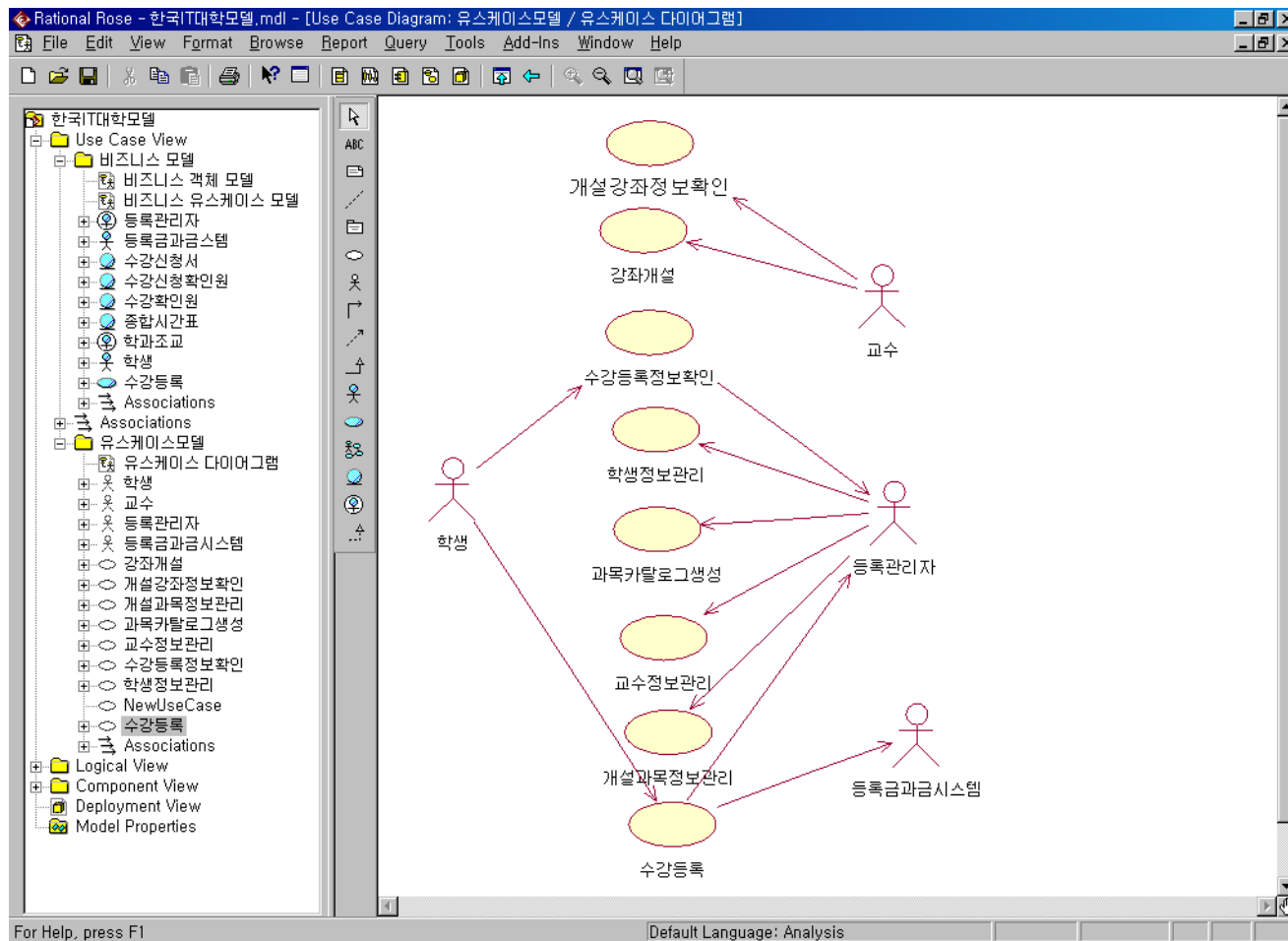
1.5 후행조건(Postcondition)

교수가 요청한 강좌가 개설강좌에 추가되거나, 취소한 강좌가 개설강좌에서 삭제된다. 강좌개설마감이 되었을 때, 학사일정에 강좌개설이 마감되었음을 표시한다.

1.6 제약사항

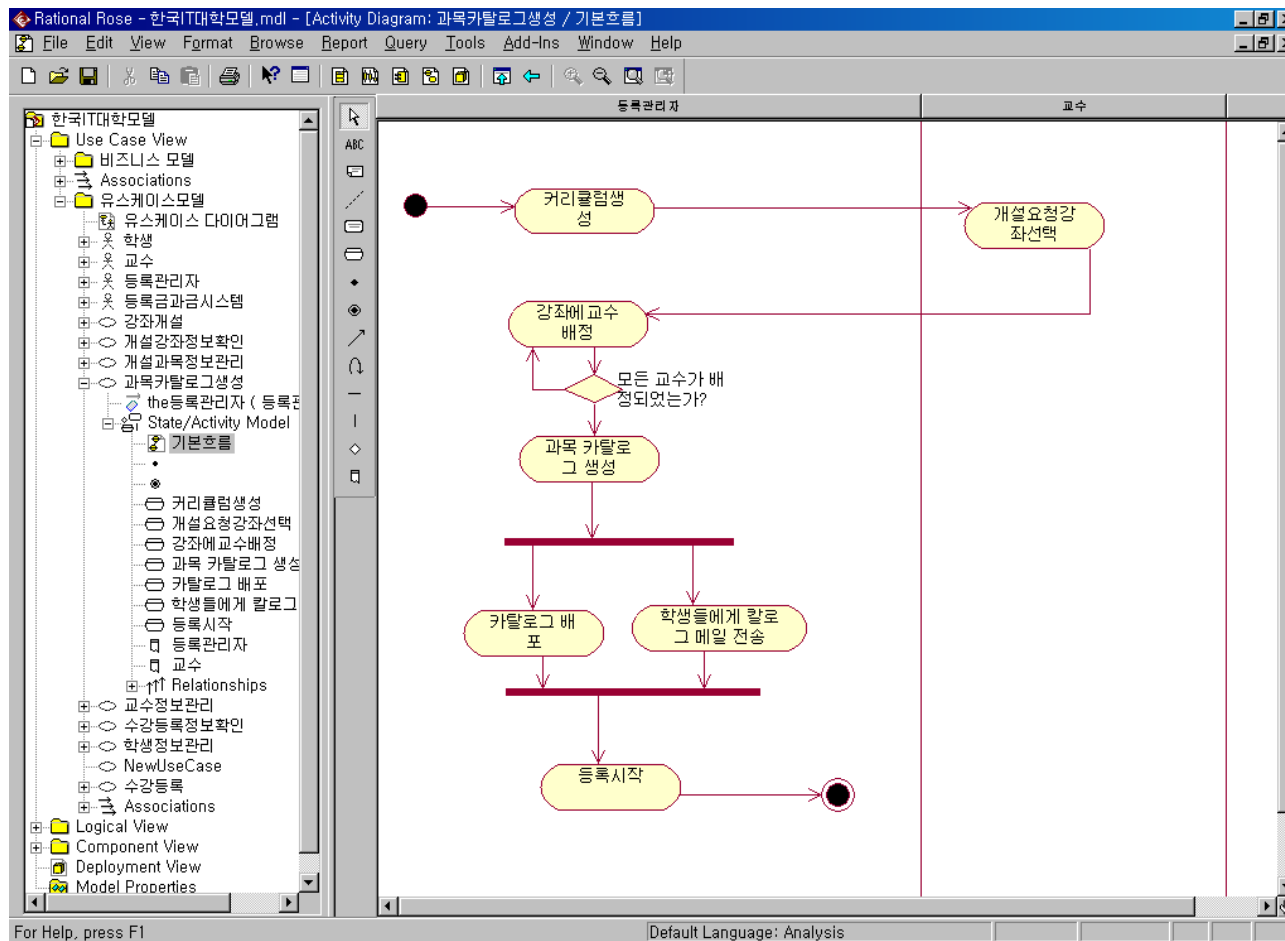
■ 유스케이스 모델

- 수강등록 시스템 유스케이스 모델



■ 유스케이스 모델

- 과목카탈로그생성 유스케이스의 기본 흐름 액티비티 다이어그램



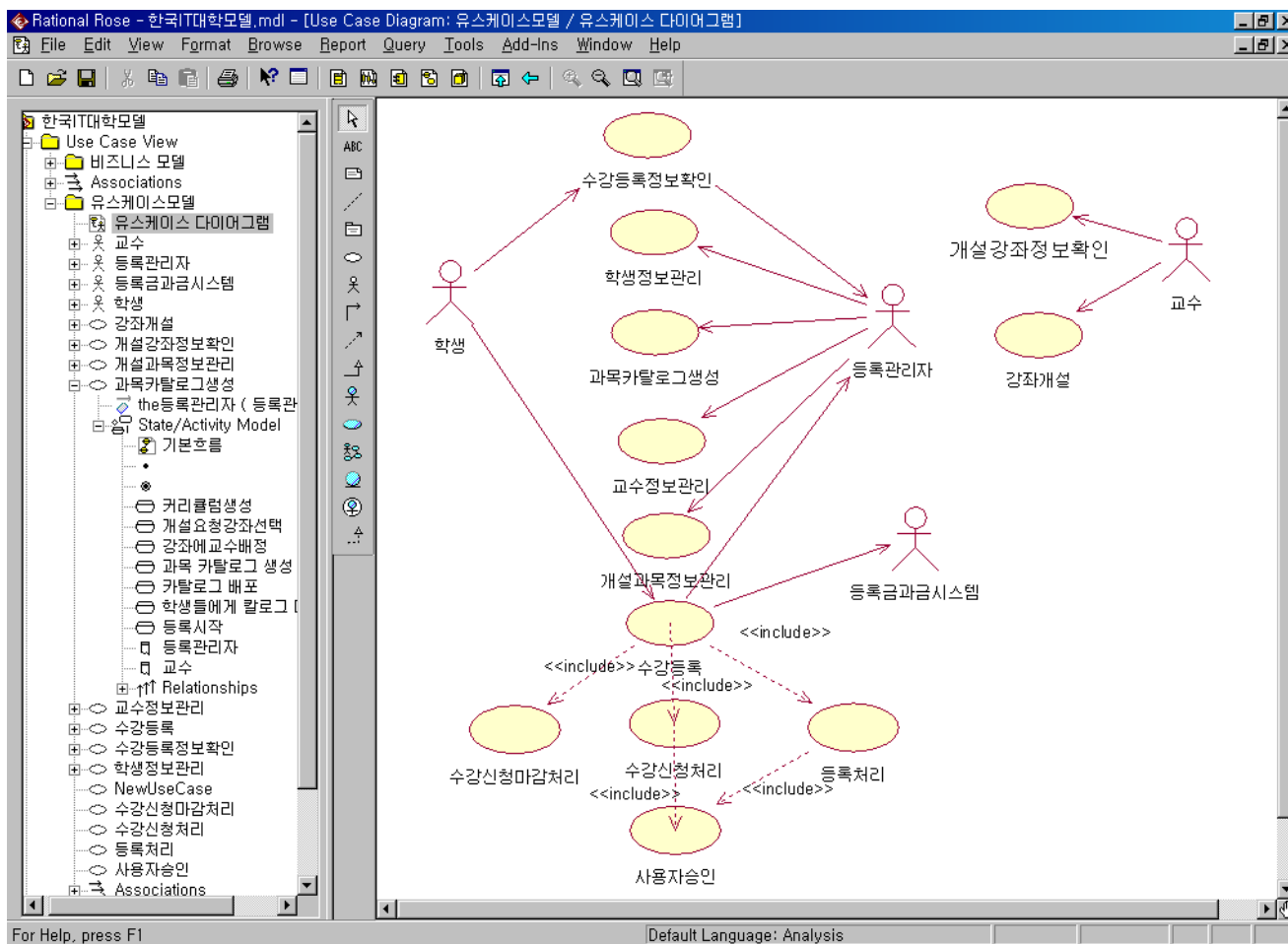
■ 유스케이스 구조화

문제 정의서

수강등록 유스케이스는 2차 액터로 등록금과금시스템과 등록관리자를 갖는다. 수강등록 사용자 레벨 유스케이스를 개발자 레벨 유스케이스로 분할해 보면, 등록금과금시스템에 관련된 부분은 등록금 납입 기간이 완료될 때, 학생들의 등록금 납부 정보를 수강등록 시스템에 제공하기 위한 “등록금납부정보 제공”으로 등록관리자와 관련된 부분은 신청 등록 기간이 완료되었음을 시스템에 알리기 위한 “수강 등록마감”으로, 시스템의 지원 작업에 관련된 부분은 수강신청기간이 완료되었을 때, 신청을 마감하기 위한 수강신청마감이 된다. 이러한 분리된 유스케이스를 가지고 사용자가 한자리에서 끝내기를 원하는 단위가 무엇인지에 대해서 대화를 시작한다.

■ 유스케이스 구조화

- 구조화된 수강등록 시스템 유스케이스 모델



■ 용어집

용어집		작업흐름		R00 요구사항		
시스템명		작성자		작성일자		
		문서번호	DR010	버전		쪽

용어집	영문법	설명	비고
예금 잔액	Account balance		
만기일	Due date		
예금주	Account owner		

■ 용어집

• 작성지침

- 용어집은 프로젝트 시작부터 지속적으로 작성되는 산출물
- 해당 업무 조직에서 사용하는 중요한 일상용어 정리
- 시스템을 기술하는 데 사용되는 용어의 표준화 및 업무 도메인에 대한 이해를 돕는 것
- 분석 설계 단계에서 객체를 식별하는데 사용
- 용어집에 기술되는 용어는 업무 도메인, 시스템 도메인의 주요 용어가 해당

• 용어명, 영문명, 설명, 비교

- 용어명, 설명 외에도, 필요한 경우 해당 용어의 영문명(또는 영문 약어), 실제 문서 양식이나 근거가 되는 업무 문서에 대한 참조를 비교란에 기술

■ 사용자 요구사항 정의서

용어집		작업흐름		R00 요구사항		
시스템명		작성자		작성일자		
		문서번호	DR010	버전		쪽



- 개요 : 하단부 작성 지침 참조
 - 면담 결과를 이용하여 개략적으로 기술
 - 지역 구분 : 서울 /
 - 업무 구분 : 보상 / 산재 보상

■ 사용자 요구사항 정의서

요구사항				현행 처리 방법	수용 여부	미반영 사유	비고
ID	유형	구분	내역				
1	제도개선 및 조회 출력 개선		구 상 금 납 부 통지서를 출력토록 조치(통지서 양식이 없음)	기능 없음	개발 반영		
2	제도개선 및 프로세스 개선		수급권자의 계좌 정보를 입력 및 변경 하고 나 면 장해연금 지급시에 바로 적용이 되게 함	본부에서 만 일괄 적용	개발 미반 영	금융 사고의 위험이 있으므로 현 처리대로 결재를 득한 후에 처리하는 것이 타당	

- 사용자 업무 정의 : 작성 지침 참조
- 현 시스템 분석 : 작성 지침 참조

■ 사용자 요구사항 정의서

- 작성지침

- 고객, 시스템 최종 사용자, 마케팅 담당자, 전산실 직원 등 개발 시스템과 관련한 다양한 관련자 집단으로부터 면담이나 설문지, 요구사항 워크숍 등을 통해 추출한 요구사항을 분석하여 정리하기 위한 목적으로 작성

- 개요

- 현업 면담 결과에 대한 전반적인 분석 결과를 간략하게 기술
- 일반적으로 개발팀의 대표가 작성
- 개요에 언급되면 좋은 사항
 - 목적, 추진 일정, 주요 추진 내용, 방문 지사, 업무별/직급별 면담 인원, 유형별/기능별 요구사항 현황, 주요 쟁점 사항``

■ 사용자 요구사항 정의서

- 면담 결과
 - 지역 구분
 - 방문한 (지역별) 지사 명을 기술 함
 - 업무 구분
 - 업무 구分的 범위는 구체적으로 세분화
 - 대분류/중분류/소분류의 순으로 작성
 - ID
 - 해당 요구사항을 식별하게 하는 식별 번호 명시
 - 요구사항 유형
 - 기술한 요구사항이 어떤 유형의 요구사항인지 명시
 - (예) 업무 개선 사항/제도 개선, 시스템 개선 사항/프로세스 개선, 시스템 개선 사항/입출력 개선, 관리항목 추가, 조회 출력 개선
 - 요구사항 구분
 - 기술한 요구사항이 어느 기능에 해당하는 것인지를 명시
 - (예) 계좌번호, 급여징수

■ 사용자 요구사항 정의서

- 면담 결과

- 현행 처리 방법

- 기술한 요구사항에 대해 기존 시스템이 존재하는 경우
 - » 기존 시스템을 통해 처리되는 현재 업무 진행 방식
 - 시스템이 없는 경우
 - » 매뉴얼 한 방식의 업무 처리 진행 방식 기술

- 수용 여부

- 현업 면담 결과 추출한 요구사항에 대해 개발팀과 고객 측의 개발 반영/미반영 여부 식별

- 미반영 사유

- 미반영한 요구사항의 경우, 미반영 사유 작성

- 비교

- 요구사항 추출과정에서 얻어낸 추가적인 정보 기술

- 사용자 요구사항 정의서
 - 사용자 업무 정의
 - 시스템 개발 범위에 해당하는 사용자 그룹의 직무 기술
 - 현 시스템 분석
 - 기존 시스템을 사용하는 경우 현 시스템의 문제점이나 개선 요구사항 기술

■ 보충명세서

• 소개

보충명세서		작업흐름		R00 요구사항		
시스템명		작성자		작성일자		
		문서번호	DR010	버전		쪽

- 보충명세서의 “소개”는 전체 문서에 대해 설명하는 부분
- 여기에는 보충명세서의 목적, 범위, 정의, 축약어, 약어, 참조, 개요 등 포함
- 시스템의 비기능적 요구사항 식별
- 유스케이스 모델의 유스케이스를 식별하지 않음
- 보충명세서에 포함되는 요구사항
 - 법률과 요구사항의 조정 담당자는 애플리케이션 표준
 - 시스템의 품질 속성은 유용성, 신뢰성, 성능, 요구사항 지원가능성 등
 - 다른 요구사항은 운영체제와 환경, 호환성 요구사항, 설계 제약사항 등

■ 보충명세서

- 목적
 - 보충명세서의 목적 기술
- 범위
 - 보충명세서의 범위를 간결하게 기술
 - 프로젝트에서 이 문서의 영향과 작용을 연계시킴
- 정의, 축약어, 약어
 - 보충명세서를 적절히 설명하기 위해 요구되는 모든 용어의 정의, 축약어, 약어를 제공
 - 용어집의 참조에 의해 제공
- 참조
 - 보충명세서에서 참조된 모든 문서에 대한 모든 목록 제공
 - 날짜, 조직, 제목, 보고서 번호 등으로 각 문서 식별
 - 참조가 이루어질 수 있는 곳의 출처 지정
 - 이 정보는 다른 문서나 부록에 참조로 제공될 수 있음

■ 보충명세서

- 개요
 - 보충명세서가 무엇을 포함하는지에 관해 기술
 - 문서가 어떻게 구성되는지에 관해 설명
- 기능성
 - 자연어 스타일로 표현된 시스템의 기능성 요구사항 기술
 - 애플리케이션, 소프트웨어 요구사항 명세서 패키지의 크기, 섹션의 조직 등으로 구성
 - 기능성 요구사항은 특성, 역량, 보안 등
- 유용성
 - 유용성에 영향을 미치는 모든 요구사항
 - 유용성에 포함되는 항목 항목들
 - 일반적인 사용자와 숙련된 사용자를 위해 요구된 교육 기술
 - 일반적 작업을 위한 작업시간을 측정하여 기술
 - 일반적인 유용성 표준을 따르는 요구사항 기술

■ 보충명세서

- 신뢰성

- 시스템의 신뢰성을 위한 요구사항 기술

- 신뢰성 기술 항목

- 유용성

- » 유용한 시간, 사용 시간, 관리 접근, 감소시킨 모드 오퍼레이션 등의 비율 명시

- 실패 간의 평균 시간(Mean Time Between Failures (MTBF))

- » 일반적으로 시간으로 명시되나 날짜나 달, 년으로 명시되기도 함

- 복구 시간(Mean Time To Repair (MTTR))

- » 시스템이 문제로부터 복구하는 데 걸리는 소요 시간 기술

- 정확성

- » 시스템 입력에 요구되는 정확성, 명확성

■ 보충명세서

- 신뢰성
 - 신뢰성 기술 항목
 - 최대 오류 수나 결함 비율
 - » 일반적으로 코드의 각 천 라인마다 나타나는 오류나 기능적인 포인터마다 일어나는 오류
 - 오류나 결함 비율
 - » 중요하지 않은 오류, 중요한 오류, 치명적인 오류의 식으로 구분
 - » 요구사항은 어떤 “치명적” 오류가 생길 수 있는지 정의
- 성능
 - 시스템의 수행성능에 대한 특징 기술
 - 성능 기술 항목
 - 트랜잭션을 위한 응답시간 (average, maximum)
 - 효율성 (예제, transactions per second)
 - 역량 (예제, 고객 또는 시스템이 처리할 수 있는 트랜잭션의 수)
 - 성능저하 모드
 - 리소스 사용
 - » 메모리, 디스크, 통신

■ 보충명세서

- 지원가능성
 - 시스템의 지원가능성, 유지보수성 강화, 코딩 표준, 이름 짓는 규약, 클래스 라이브러리, 유지보수 제어, 유지보수 유틸리티 요구사항 등 서술
- 설계 제약사항
 - 생성될 시스템의 설계 제약사항 기술
 - 디자인 결정을 표현
- 온라인 사용자 문서화와 도움말 시스템 요구사항
 - 요구사항, 온라인 사용자 문서화, 시스템 도움말, 공지 도움말 등 기술
- 구매한 컴포넌트
 - 구매한 컴포넌트의 시스템 사용, 라이선스 또는 제한사항, 호환성/상호운용성, 인터페이스 표준 등 기술

■ 보충명세서

• 인터페이스

- 애플리케이션을 지원하는 인터페이스 정의
- 소프트웨어가 개발될 수 있고, 인터페이스 요구사항에 대하여 검증할 수 있기 위한 적합 특이성, 프로토콜, 포트와 논리주소 등 포함
- 사용자 인터페이스
 - 소프트웨어에 의해 실행되는 사용자 인터페이스 기술
- 하드웨어 인터페이스
 - 논리적인 구조, 물리 주소, 예상되는 행동 등을 포함하는 소프트웨어를 지원하는 하드웨어 인터페이스 정의
- 소프트웨어 인터페이스
 - 소프트웨어의 다른 컴포넌트에 대한 소프트웨어 인터페이스 기술
- 통신 인터페이스
 - 시스템에서의 커뮤니케이션 인터페이스, 근거리 통신망 등과 같은 것을 기술

- 보충명세서
 - 라이선스 요구사항
 - 라이선스 시행 요구사항, 취급 제한 요구사항 등 정의
 - 적용 가능한 표준
 - 적용 가능한 표준 기술


- 객체지향 기법을 적용한 개발 프로젝트에서 요구사항 정의 방안 수립 및 실행능력 배양
- 팀 프로젝트에서 실행방법 모색 및 적용
- 시스템 개발을 위한 요구사항 정의 및 산출물 작성
- 관련 개념 정리

강의 계획 피드백 (8주차)

주차	강의주제	강의내용	과제	평가
1주차	객체지향 패러다임	과목 소개 및 객체지향 방법론의 전반적인 개요		
2주차	프로젝트 관리1	프로젝트 계획 및 팀 편성/프로젝트 과제 제시		
3주차	소프트웨어 개발방법론과 UML	기존의 소프트웨어 개발방법론과 객체지향방법론 차이점 이해	과제1 : 프로젝트 현장 및 계획서 제출(5)	
4주차	Use Case와 UML	UML 특성 이해		
5주차	UP(Unified Process) 방법론	UP 방법론 이해		
6주차	비즈니스 모델링 및 요구사항 정의	사례를 통한 비즈니스 모델링 및 요구사항 정의 방법 이해	과제2 : 요구사항 정의 결과 제출(5)	
7주차	분석 모델링 및 UML 다이어그램 (분석)	객체지향 분석 방법 이해 및 분석용 UML 다이어그램 작성 방법 이해		
8주차	분석 결과 문서화 및 설계 모델링	분석 산출물 작성 방법 및 객체지향 설계 방법 이해	과제3 : 분석 결과 제출(10)	
9주차	UML 다이어그램(설계)	설계용 UML 다이어그램 작성 방법 이해		
10주차	객체 설계	객체설계 및 세분화		
11주차	설계 결과의 문서화 및 프로젝트 관리 2	시스템 설계 결과의 문서화 방법 이해 및 형상관리/검증과 확인 방법 이해	과제4 : 설계 결과 제출(10)	
12주차	시스템 구현	객체지향 프로그래밍의 기본 개념 및 기법		
13주차	시스템 테스트 및 구현/시험 결과의 문서화	객체지향 테스트 기법 및 구현/시험 산출물의 문서화 방법 이해	과제5 : 구현/시험 결과 및 유지보수 계획 제출(20)	
14주차	프로젝트 관리3	소프트웨어 품질관리와 프로세스 개선 방법 이해		
15주차	최종 결과 문서화 및 발표	최종 산출물 문서화 방법 이해 및 개발 결과 발표	과제6 : 최종보고서 제출 및 발표(10)	

다음 주(9주차) 강의계획

- 교재#2의 3~5장, 교재#1의 11장 내용 **철저히** 이해하기
- 교재#2의 6장 **꼭** 읽어오기
- 팀 프로젝트 진행 : **계획단계-분석단계**
 - **계획단계**
 - 프로젝트 헌장(PC) 작성
 - 1차 제출(2016.09.21, 18:00)
 - 최종 제출(2016.09.28, 18:00)
 - 프로젝트 관리 계획서(PMP) 작성
 - 1차 제출(2016.09.28, 18:00)
 - 최종 제출(2016.10.05, 18:00)
 - **분석단계**
 - 요구사항 정의서 작성
 - 1차 제출(2016.10.05, 18:00)
 - 최종 제출(2016.10.12, 18:00)
 - 최종 제출(수정본, 2016.10.30, 24:00)
 - **요구사항 명세서(SRS) 작성**
 - 1차 제출(2016.10.30, 24:00)

 다음 주(9주차)는 **교재#2의 6장 강의**