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

ONE STEP AHEAD

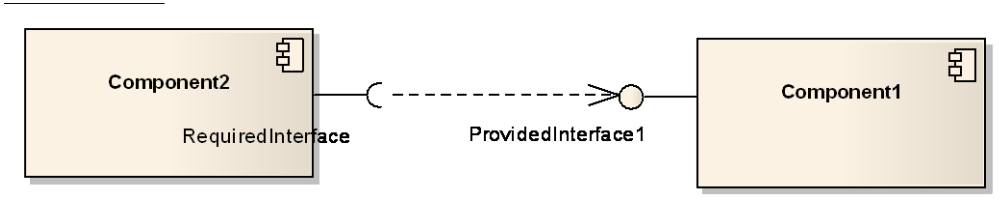
1. 컴포넌트 다이어그램
2. 개발자가 읽어야 할 도서

1. 컴포넌트 다이어그램

- ❑ 개요
- ❑ 구성요소
- ❑ 작성 및 주의사항

개요 (1/3)

- **시스템의 구현과점**
  - 실행모듈(컴포넌트)을 정의하고 실행모듈 간의 정적 상호작용을 정의한 모델
  - 물리적 구성요소들로 실행모듈(컴포넌트) 구성되고 그들간의 연관성을 정의
- **SW 분야에서 사용되는 컴포넌트의 광역적 의미**
  - 시스템의 재사용 가능한 구성요소
  - 시스템의 교체단위이자 업그레이드 단위
  - 인터페이스를 통한 기능 사용 및 독립적으로 인도되는 기능조각

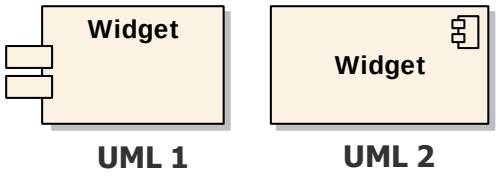


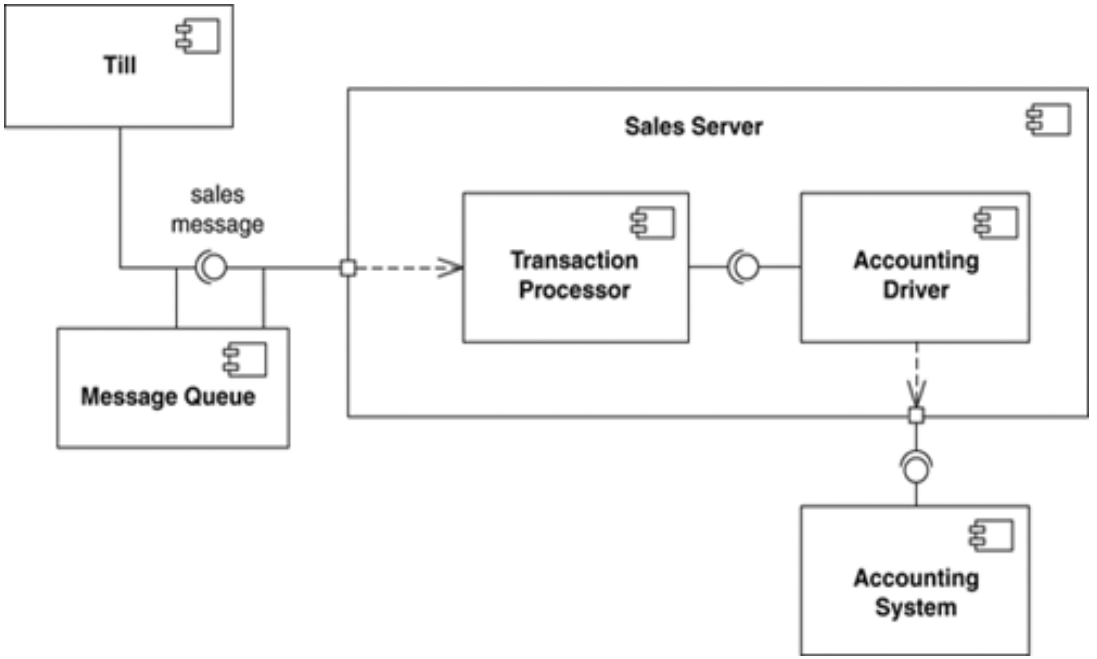
개요 (2/3)

❑ 표기법 변경

- UML1.x의 표기법이 UML2.x에서 변경

표기형식의 변경





개요 (3/3)

❑ 작성목적

- 시스템의 실행모듈(컴포넌트)들을 정의
- 컴포넌트 간 의존성*Dependency* 정의
- 실행모듈뿐 아니라 소스코드, 데이터베이스 등의 상호작용 모델링

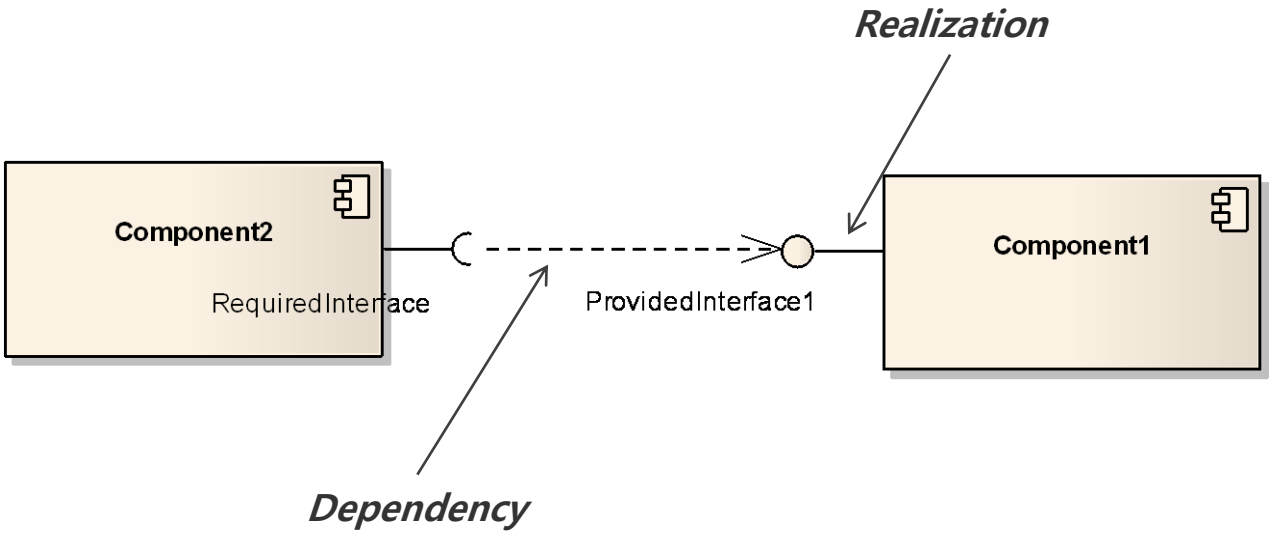
❑ 작성시기

- 모든 클래스가 물리적으로 완전히 정의되고 상호관계도 정의된 후 작성
- 시스템 설계단계 후반에 주로 작성

구성요소 (1/5)

❑ 컴포넌트 다이어그램의 구성요소

- 요소: 컴포넌트*Component*, 인터페이스*Interface*
- 관계: 의존성*Dependency*, 실현*Realization*



구성요소 (2/5)

❑ 컴포넌트*Component*

- 독립적으로 배포되고 교체되며 재사용될 수 있는 SW 조각
- 소스코드, User Interface, 분석, 설계 산출물 등을 포함한 광의적 정의
- 용어를 표현하는 사람의 의도에 따라 달라지므로 컨텍스트 파악이 중요

컴포넌트 예)

보험시스템	고객, 사원, 계약, 상품 등
오픈 마켓 시스템	신용카드 결재, 상품, 우편번호 검색 등

구성요소 (3/5)

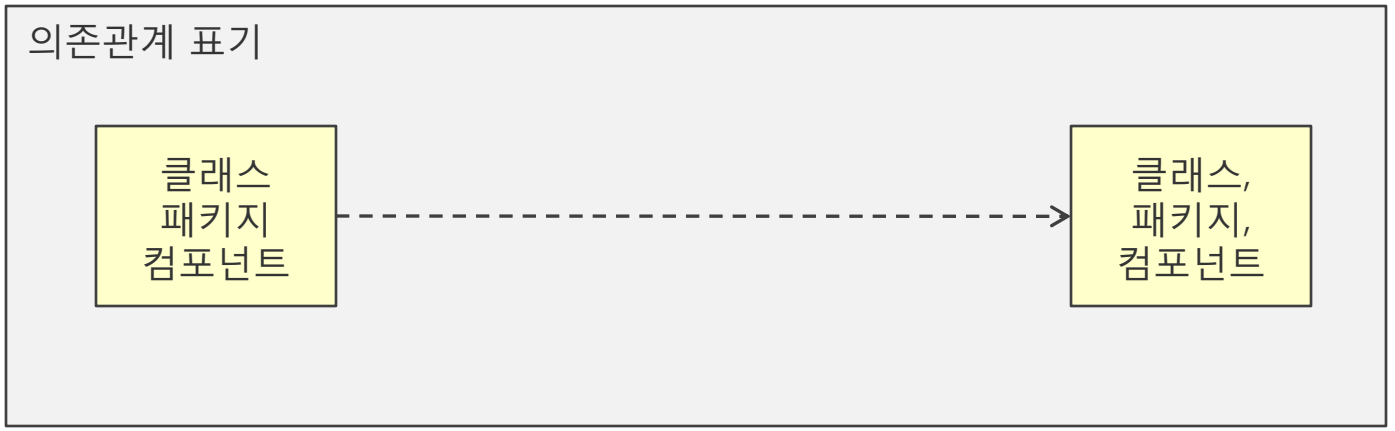
❑ 인터페이스*Interface*

- 클래스의 일종
- 컴포넌트의 기능을 외부에 공개할 목적으로 사용
- 클래스나 컴포넌트는 선언뿐인 인터페이스의 구현을 담당

구성요소 (4/5)

❑ 의존관계*Dependency*

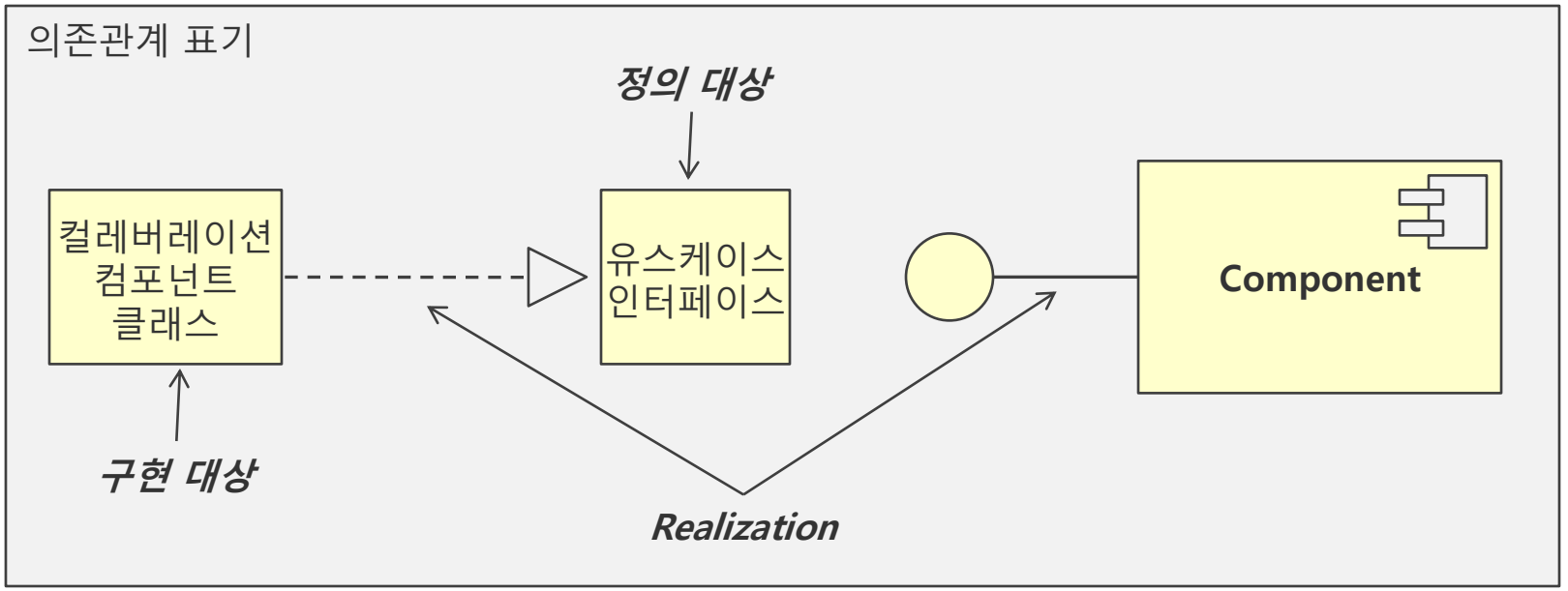
- 객체나 컴포넌트가 다른 객체나 컴포넌트의 실행을 요구할 경우 실행 또는 참조관계 표현
- 클래스, 패키지, 컴포넌트 사이에 주로 사용되는 관계
- 때로는 클래스 – 패키지 – 컴포넌트 상호 간에도 사용



구성요소 (6/5)

❑ 실체화 관계*Realization*

- 정의하는 인터페이스와 이를 구현하는 클래스 간의 표현하는 관계
- 실체화 유형
  - 유스케이스(정의) – 컬레버레이션(구현)
  - 인터페이스(정의) – 컴포넌트(구현)
  - 인터페이스(정의) – 클래스(구현)



작성 및 주의사항 (1/2)

❑ 컴포넌트 다이어그램 작성 단계

- 컴포넌트 대상 정의
- 컴포넌트 식별
- 컴포넌트 배치
- 의존관계 정의

단계	내용
컴포넌트 대상 정의	컴포넌트 다이어그램을 그리기 전에 무엇을 컴포넌트로 표현할 지 클래스를 구성요소로 하는 실행모듈로 할지, 소스코드를 정의할 지 기타 무엇을 컴포넌트로 표현할 지를 정해야 한다.
컴포넌트 식별	컴포넌트 다이어그램에 등장할 컴포넌트를 정한다. 소스파일일 경우 그 대상은 쉽게 식별되지만 실행모듈일 경우 간단하지 않으므로 여러 가지 방법으로 컴포넌트를 식별해 내는 작업을 수행해야 한다.
컴포넌트 배치	컴포넌트 다이어그램에 컴포넌트를 배치하고 이름을 정의한다. 인터페이스가 필요한 경우 인터페이스를 정의하고 컴포넌트와 실체화 관계로 연결한다.
의존관계 정의	컴포넌트 간 의존관계를 분석하여 Dependency를 정의한다.

작성 및 주의사항 (2/2)

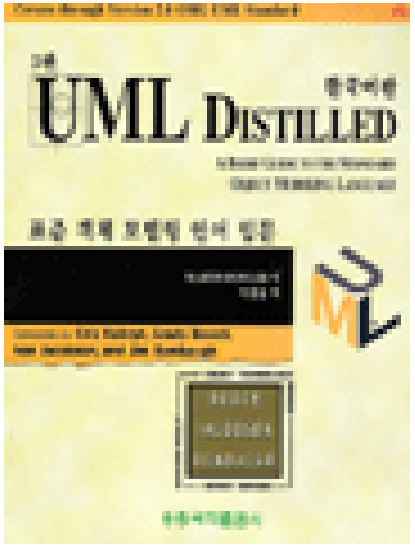
❑ 컴포넌트 다이어그램 작성 시 주의사항

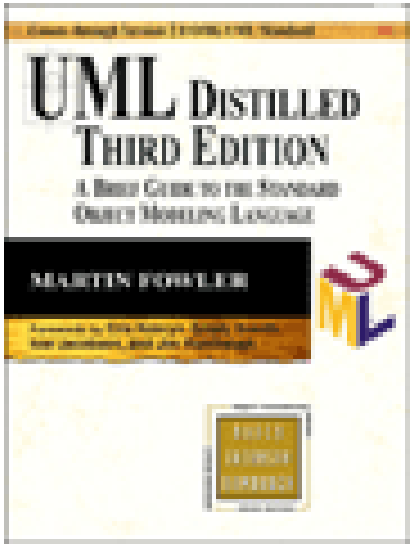
- 컴포넌트는 응집도는 높고 결합도는 낮은 단위로 정의
- 컴포넌트 크기*Granularity* 일관성 유지
- 추상화 수준에 맞는 상세성을 일관되게 제공
- 목적을 전달할 수 있는 명칭 부여

순서	내용
응집도와 결합도	실행모듈로서의 컴포넌트를 식별할 때, 컴포넌트는 다른 컴포넌트와 독립적이고 기능 차별성을 갖추는 단위로 정의되어야 합니다. 즉, 기능 측면에서 컴포넌트 내부는 강한 유사성을 갖는 단위들로 구성되어야 하고(높은 응집도), 다른 컴포넌트에 강하게 종속되지는 않는(낮은 결합도) 단위로 정의되어야 한다.
크기	한 시스템에서 컴포넌트의 크기에 너무 차이가 나면 바람직하지 않으므로 컴포넌트의 크기는 기술구조와 시스템 특성들이 고려되어 적절한 크기로 정의해야 하며, 그 크기도 되도록 많이 차이 나지 않도록 해야 한다.
상세성	모든 모델이 마찬가지 이지만 한 장의 모델에는 동일한 상세화 레벨이 유지하고 서로 다른 추상화 레벨의 컴포넌트가 섞여 있으면 의미 파악이 어려우며 소스와 실행모듈을 한 장에 정의한 컴포넌트는 좋은 예가 아니다.
명칭	모호한 명칭으로 정의하면 혼란만 야기 시키는 결과가 나오므로 명확한 표현을 사용해야 한다.

개발자가 읽어야 할 도서

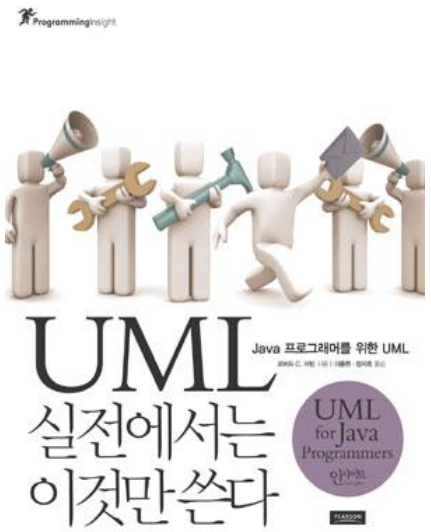
- UML Distilled: 표준 객체 모델링 언어 입문 (3/E)
- 저자: Martin Folwler

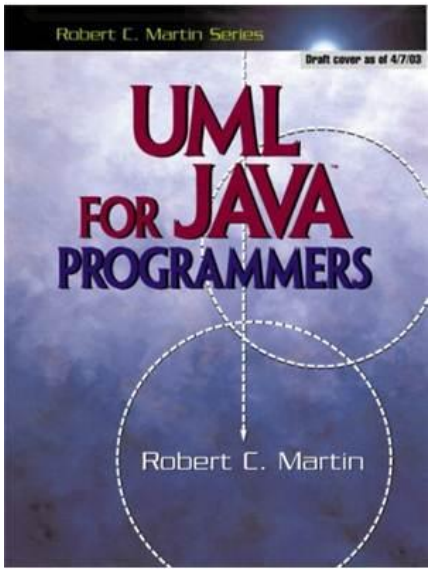




UML for Java Programmers

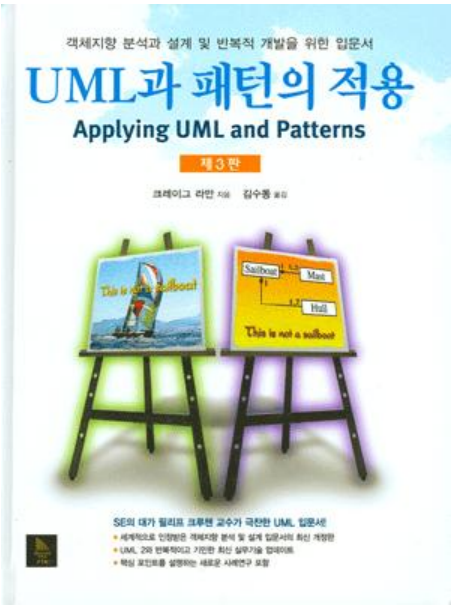
- UML, 실전에서는 이것만 쓴다
- 저자: Robert Cecil Martin

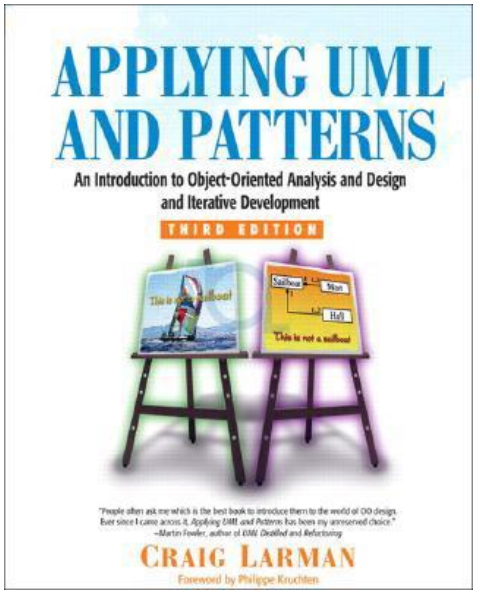




Applying UML and Pattern

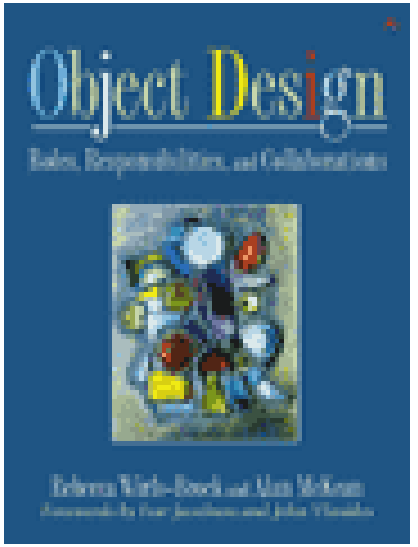
- UML 적용과 패턴 (3/e)
- 저자: Craig Larman





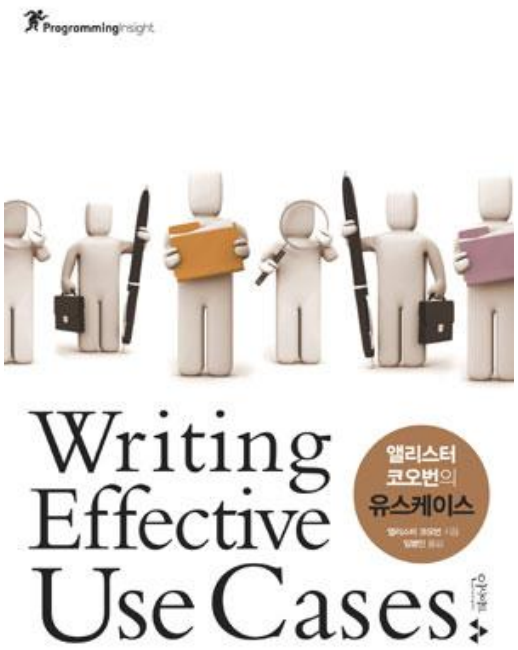
Object Design: Roles, Responsibilities and Collaborations

- 오브젝트 디자인
- Rebecca Wirfs-Brock, Alan McKean



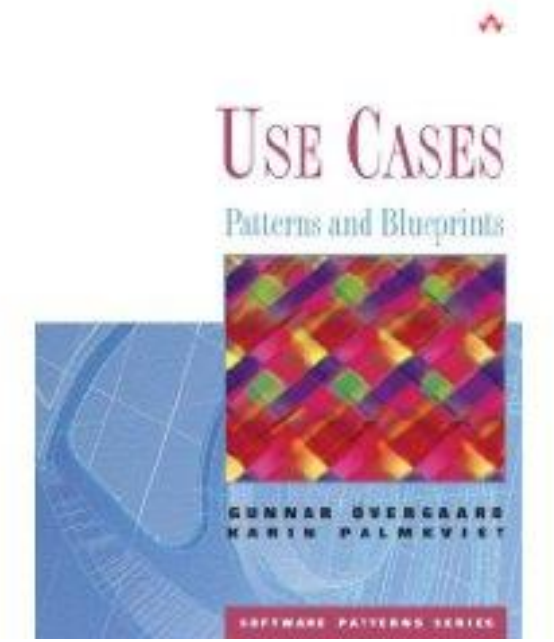
Writing Effective Use cases

- 유스케이스 바로쓰기
- Alistair-Cockburn



Usecases Patterns and Blueprints

- Gunnar Overgaard, Karin Palmkvist
- Addison Wesley Professional, 2004



- Refactoring : Improving the Design of Existing Code Addison-Wesley Professional, Martin Fowler, Addison-Wesley, 1999
- Domain Driven Design: Tackling Complexity in the Heart of Software, Eric Evans, Addison-Wesley, 2003
- 기타 객체지향 서적...