노동부 Ministry of Labor 중소기업직업훈련컨소시엄

Korea Software Technology Association

UML2.x 기초 다루기

훈련기간: 2010.01.25 ~ 02.05

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 역사 UML 다이어그램분류	- 구조 다이어그램 클래스 객체 컴포넌트 배치	- 행위 다이어그램 유스케이스 액티비티 상태기계	- 상호작용 다이어그램 상호작용 Overview 시퀀스 커뮤니케이션 타이밍	- 유스케이스 I 유스케이스 개요 유스케이스 내용 유스케이스 다이어그램
6 일차	**7 일차**	**8 일차**	**9 일차**	**10 일차**
- 유스케이스 II 유스케이스 목표수준 유스케이스 명세 유스케이스 패턴	- 유스케이스 III 유스케이스 분석기법 분석클래스 제어클래스 실체클래스	- 요구사항 모델실습 I 유스케이스 사용자 시나리오 핵심개념 모델	- 요구사항 모델실습 II 인터페이스 추출 유스케이스 분석 컴포넌트 식별	- 설계모델 실습 컴포넌트 설계 유스케이스 설계 도메인 모델

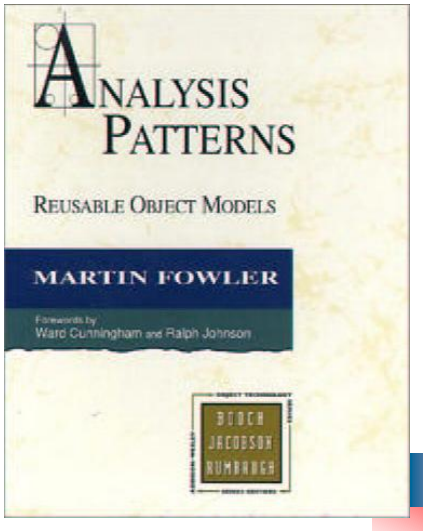
8일차 – 요구사항 모델 실습 I

ONE STEP AHEAD

1. 핵심 개념 모델

핵심 개념 모델

- ❑ 개요
- ❑ 도메인 모델



개요 (1/2)

- Martin Fowler의 저서 'Analysis Pattern – Reusable Object Model' [Addison-Wesley, 1996] 에서 처음 사용
- 객체 개발의 중요한 원칙
  - 소프트웨어의 문제를 반영한 구조의 디자인 수행
  - 모델은 분석과 설계의 마무리된 결과
  - 대부분의 개발자는 분석과 설계의 차이에 대한 인식이 없음
- Mental Model
  - 요구사항에 대한 지속적인 문제를 머리 속으로 모델링 하여 회고
  - 이해하고 있는 문제에 대한 Mental Model을 개념모델로 생성

개요 (2/2)

- **도메인 전문가*Domain Expert***
  - 개념모델링 수행 시 도메인 전문가를 포함하는 것이 필수적
  - 전문가의 지식은 좋은 분석 모델의 중심이 됨
- **분석 기술은 소프트웨어 기술에 독립적**
- **개념 모델은 소프트웨어 구현보다 소프트웨어 인터페이스에 관계가 있음**

모델링 원칙

- 모델은 옳고 그름이 아니고 보다 더 유용한가 아니면 덜 유용한가 이다.

- 개념모델은 구현(classes)이 아니고 인터페이스(types)와 연결 된다.

도메인 모델

❑ 정의

- 객체지향 분석 단계에서 도메인을 중요한 개념이나 객체로 분해
  - 도메인에서 개념적인*conceptual* 클래스나 실제 상황*real-situation* 객체의 시각적 표현
  - 개념*conceptual* 모델, 도메인 객체 모델, 분석 객체 모델
    - ➢ 도메인 모델은 데이터 모델이 아님.
- UP의 정의
  - 소프트웨어 객체가 아닌 실제 상황 개념적인 클래스의 표현
  - 소프트웨어 클래스를 기술하는 다이어그램들, 혹은 소프트웨어 아키텍처의 도메인 계층*layer*, 혹은 책임성이 있는 소프트웨어 객체들을 의미하지 않음.
  - 비즈니스 모델링 원칙에서 생성될 수 있는 산출물 중에 하나로 간주
    - ➢ "비즈니스 도메인에 중요한 '사물' 이나 제품을 설명하는데 초점을 맞춘" UP의 비즈니스 객체 모델BOM의 특수화된 형태
    - ➢ 하나의 도메인에 초점

도메인 모델

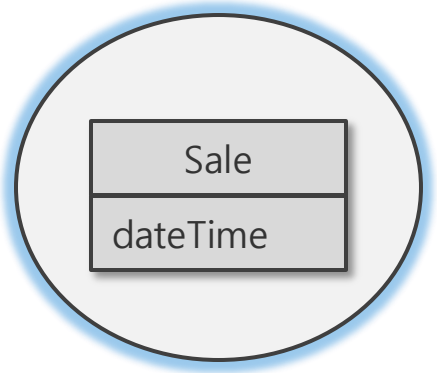
❑ UML 표기

- 오퍼레이션이 정의되지 않은 클래스 다이어그램으로 묘사
  - 개념적인 관점*conceptual perspective* 제공
  - 도메인 객체 혹은 개념 클래스
  - 개념 클래스 간의 연관*association*
  - 개념 클래스의 속성
- 시각적인 사전*visual dictionary*
  - UML을 사용한 정보는 평문으로 표현될 수 있음.
  - 도메인의 중요한 추상화, 도메인 어휘*vocabulary*, 정보 내용*information content*에 대한 시각적인 사전

도메인 모델

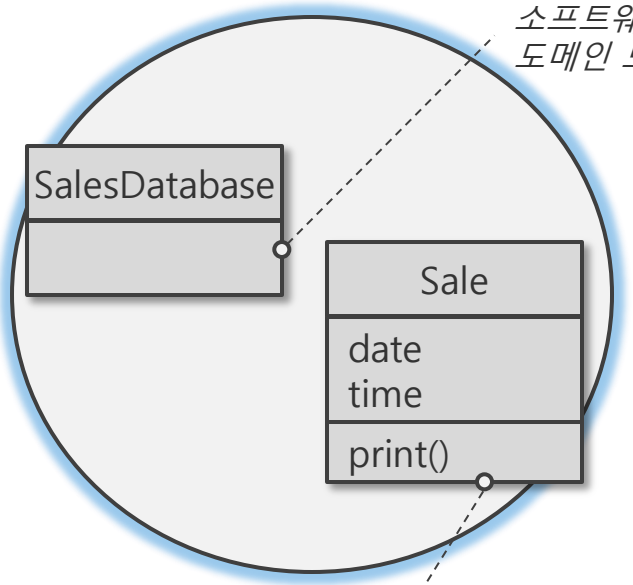
❑ 도메인 모델이 아닌 요소

- Java나 C# 클래스와 같은 소프트웨어 객체나 책임성을 가진 객체가 아닌 관심이 있는 실제 상황 도메인의 사물에 대한 시각화
- 소프트웨어 비즈니스 객체의 사진
  - 윈도우나 데이터베이스와 같은 소프트웨어 산출물
  - 책임성이나 메소드



관심 도메인의
실세계 개념의 시각화

소프트웨어 클래스의 사진이 아님



소프트웨어 산출물 :
도메인 모델 요소가 아님

소프트웨어 클래스 :
도메인 모델 요소가 아님

도메인 모델

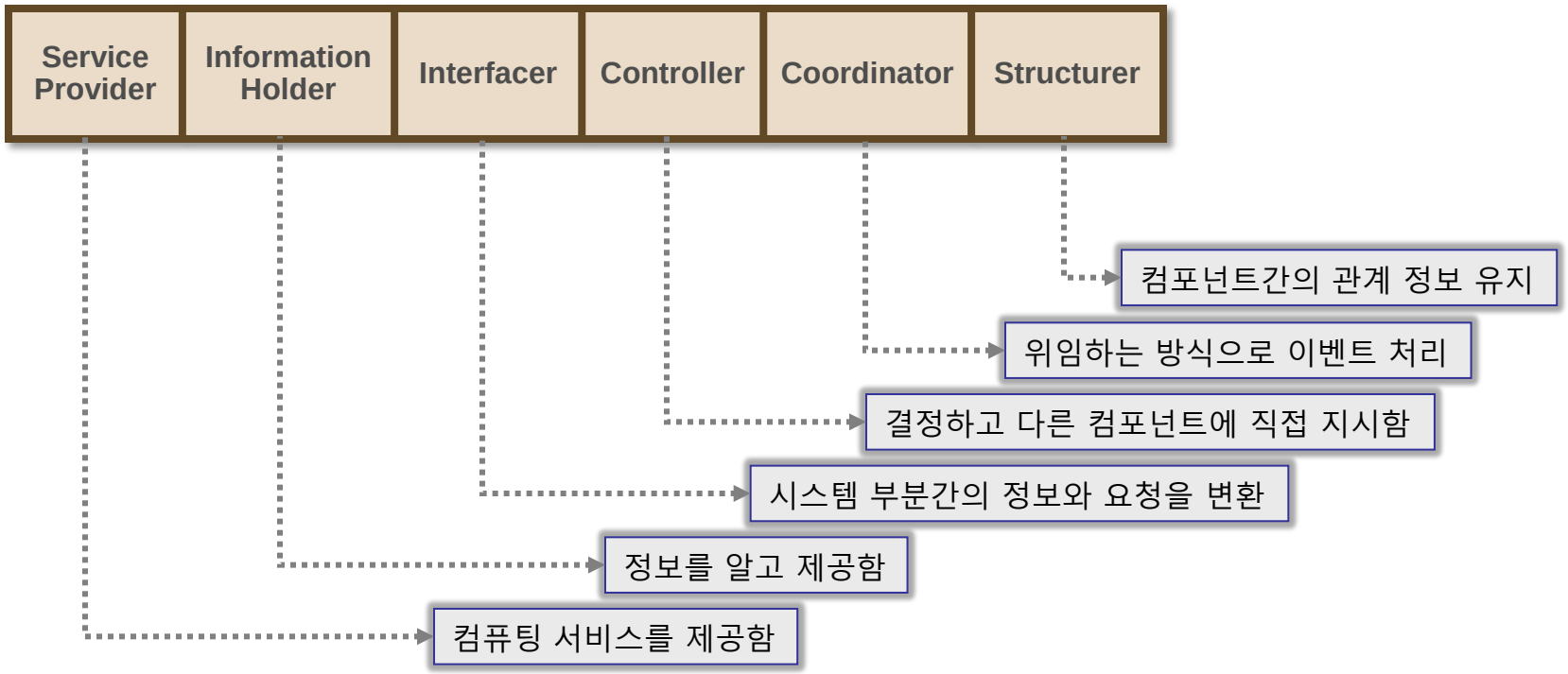
❑ 도메인 모델의 격리*isolation*

- 시스템의 다른 기능function과 분리
  - 도메인 개념을 소프트웨어 기술과 관련된 다른 개념과 혼동 회피
  - 도메인의 직관력 소실 회피
- 도메인 모델 설계 접근방법
  - 책임성 기반 설계*responsibility-driven design*
  - 계약에 의한 설계*design by contract*
  - 객체지향 설계*object-oriented design*
  - 모델 기반 설계*Model-driven design*

도메인 모델

❑ 책임성 기반 설계*responsibility-driven design*

- 역할에 의한 객체 분류



도메인 모델

❑ 계약에 의한 설계*design by contract*

- 계약contract의 의미
  - 법적인 표현 형태
    - ➢ 계약 주체 당사자가 의무obligation를 수용하고, 권리right를 획득
  - 객체지향 관점
    - ➢ 객체의 책임성이 명확하고 애매모호하지 않게 수행됨
    - ➢ 객체는 측정 자극(권리)가 충족될 때에만if and only if 서비스(의무)를 실행하는 책임이 있음.

주체	의무	권리
고객	2000원 지불 80g 이하의 물건 서울내 배송 주소 지정	4시간 내 전달되는 물건
배송 서비스 회사	4시간 내에 물건 배송	배송 주소는 서울 내 2000원 받음 모든 물건은 80g 이하

도메인 모델

❑ 모델 기반 설계*model-driven design*

- 분석은 이해되고 의미있는 방법으로 도메인에서 기본적인 개념을 포함
- 설계는 대상 배포 환경에 효과적으로 수행되고 어플리케이션이 해결하는 문제를 정확하게 푸는 프로젝트에 사용되는 프로그래밍 도구와 같이 구축될 수 있는 객체들의 집합을 지정