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

Copyright © 2009 Nextree Soft, All rights reserved

과정 개요

❑ 교육 목표 & 특징

- UML2.x의 이해
- 유스케이스 작성
- 객체모델링 이해
- UML2.x의 다양한 다이어그램 이해 및 활용
- 모델링 도구 사용법 습득

Copyright © 2009 Nextree Soft, All rights reserved

강의 요구 기술

- **본 강의는 아래 기술에 대한 이해를 필요로 합니다.**
  - 객체지향 언어(Java) 기초
  - 개발프로세스 이해

Copyright © 2009 Nextree Soft, All rights reserved

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

Copyright © 2009 Nextree Soft, All rights reserved

3일차 – 행위 다이어그램

1. 유스케이스 다이어그램
2. 액티비티 다이어그램
3. 상태기계 다이어그램

Copyright © 2009 Nextree Soft, All rights reserved

1. 유스케이스 다이어그램

ONE STEP AHEAD

- ❑ 개요
- ❑ 구성요소
- ❑ 작성 및 주의사항

Copyright © 2009 Nextree Soft, All rights reserved

개요

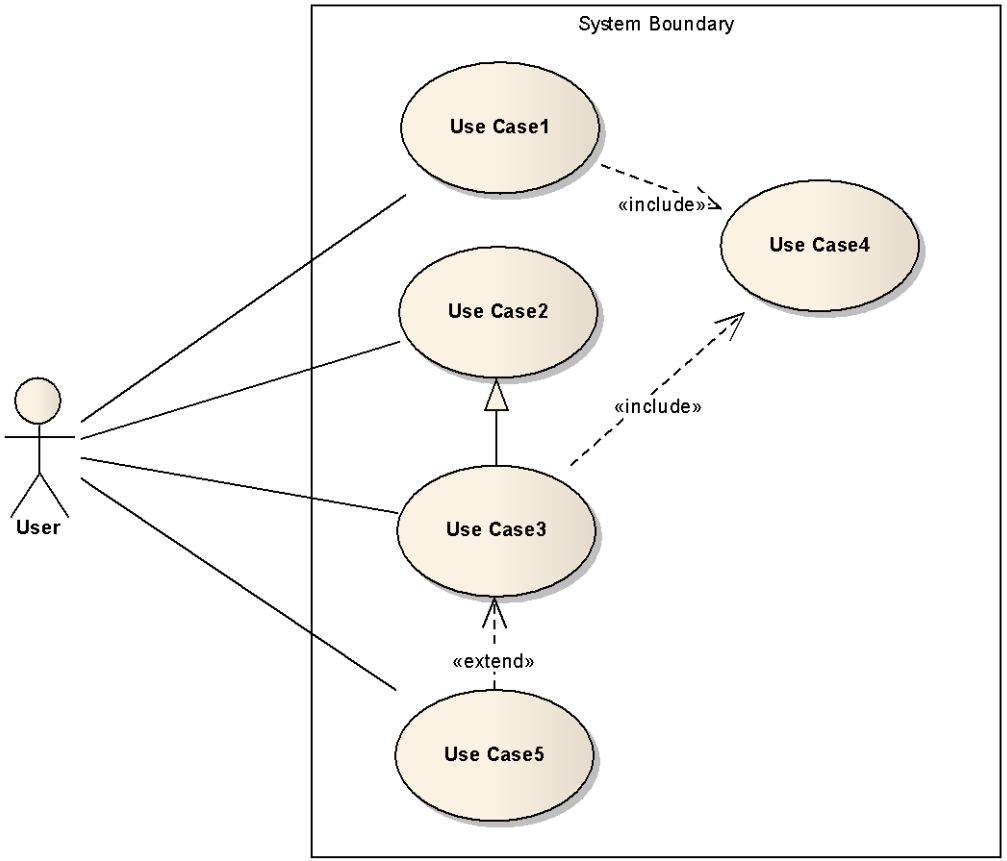
- **유스케이스의 정의**
  - 사용자 시각에서 소프트웨어 시스템의 범위와 기능을 설명하고 정의한 모델
  - 소프트웨어 시스템의 기능적 요구사항에 대한 베이스라인
- **작성시기**
  - 소프트웨어 프로젝트의 개발범위를 정의하는 단계
  - 소프트웨어에 대한 요구사항을 정의하는 단계
  - 소프트웨어의 세부기능을 분석하는 단계
  - 소프트웨어가 아닌 업무영역을 이해하고 분석하는 단계

Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (1/7)

❑ 유스케이스 다이어그램 구성요소

- 요소: 액터*Actor*, 유스케이스*Usecase*
- 관계: 커뮤니케이션*Communications*, 포함*Include*, 확장*Extend*, 일반화*Generalization*



Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (2/7)

❑ 액터*Actor*

- 시스템의 외부에 존재하면서 시스템과 교류 혹은 상호작용하는 것
- 시스템이 서비스를 해 주기를 요청하는 존재
- 시스템에게 정보를 제공하는 대상

액터의 예)

보험시스템	고객, 사원, 관리자, 결재자, 환자 등
오픈 마켓 시스템	회원, 구매자, 배달자, 관리자, 배송시스템 등
병원관리 시스템	의사, 간호사, 환자, 수납책임자 등

Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (3/7)

❑ 유스케이스 *Usecase*

- 시스템이 제공하는 서비스 혹은 기능
- 시스템이 액터에게 제공하는 사용자 관점의 기능단위
- 액터의 요청에 반응하여 원하는 처리를 수행하거나 정보를 제공
- 액터와 한 번이상의 상호작용을 통한 의미있는 묶음의 시스템 행위
- 의미있는 자기완결형의 서비스 단위
- 사용자관점에서의 정의가 필요

유스케이스의 예)

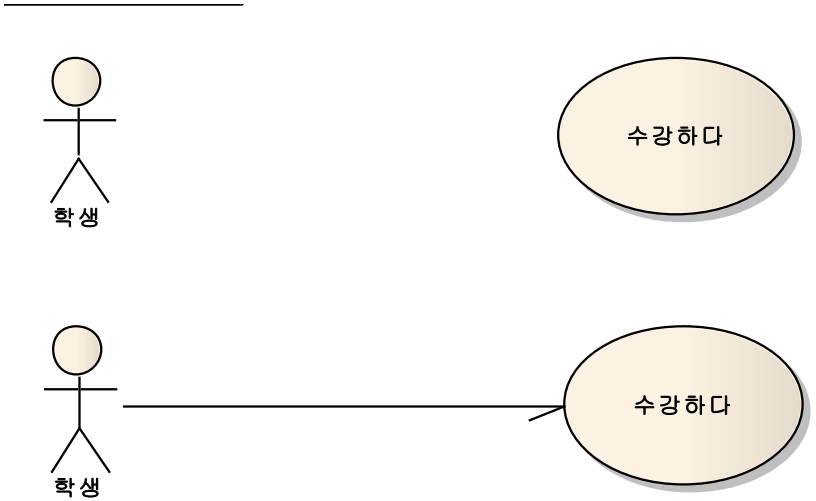
보험시스템	상품구성하다, 계약하다, 청구하다 등
오픈 마켓 시스템	물품정보조회, 주문, 배송, 결재 등
병원관리 시스템	과거병력조회, 진료기록 등록, 진료비계산 등

Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (4/7)

❑ 커뮤니케이션 *Communication*

- 액터와 유스케이스 사이에 정의되는 관계
- 일반 상호작용 관계가 존재하는 것을 의미
- 즉, 액터가 특정 사용목적을 가지고 유스케이스와 상호작용할 때 정의
- 액터를 정보를 통보받거나 요구
- 유스케이스는 정보를 제공

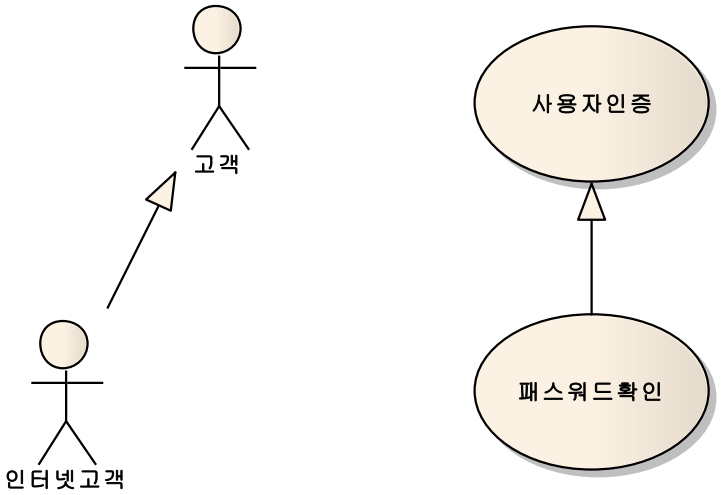


Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (5/7)

❑ 일반화*Generalization*

- 액터와 액터, 유스케이스와 유스케이스 사이에 정의
- 두 개체가 일반화 관계에 있음을 의미
- 보다 보편적인 것과 보다 구체적인 것 사이의 관계 (is-a 관계)
- 상속의 특성을 지님

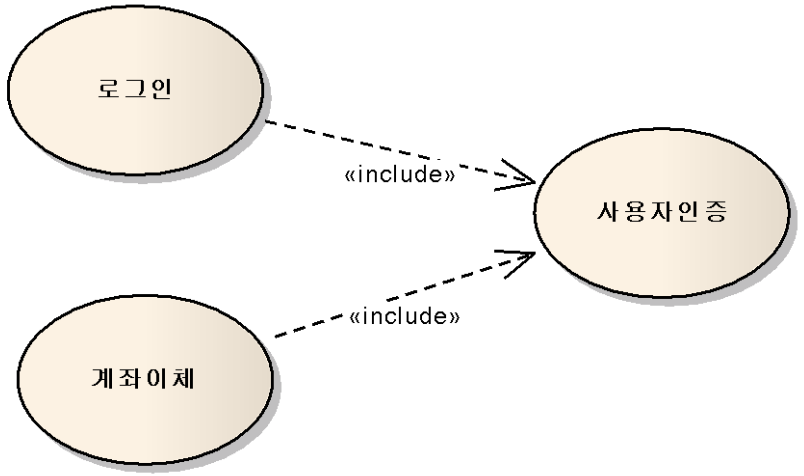


Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (6/7)

❑ 포함*Include*

- 유스케이스와 유스케이스 사이에 정의되는 관계
- 한 유스케이스가 다른 유스케이스의 서비스 수행을 요청하는 관계
- 즉, 한 유스케이스가 자신의 서비스 수행 도중에 다른 유스케이스의 서비스 사용이 필요할 때 정의
- 포함되는 유스케이스는 공통 서비스를 가진 존재

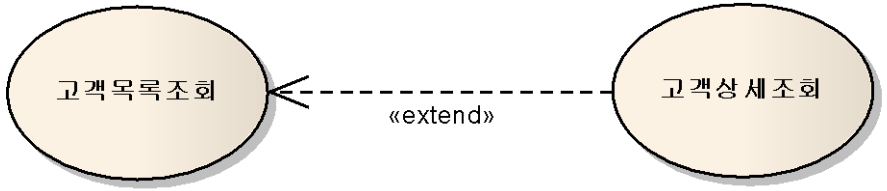


Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (7/7)

❑ 확장*Extend*

- 유스케이스와 유스케이스 사이에 정의되는 관계
- 포함관계와 동일하게 서비스 수행을 요청하는 관계
- 포함관계와 달리 서비스가 수행되지 않을 수 있음
- 수행 요청 조건을 확장점*Extension Point*이라고 함

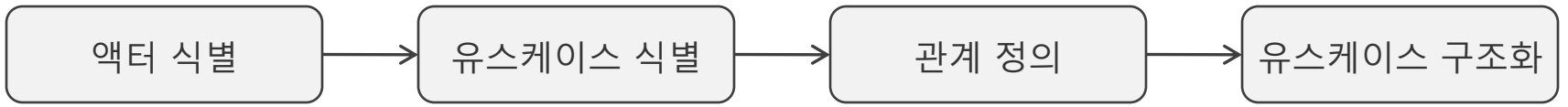


Copyright © 2009 Nextree Soft, All rights reserved

작성 및 주의사항 (1/2)

❑ 유스케이스 다이어그램 작성 단계

- 액터 식별
- 유스케이스 식별
- 관계 정의
- 유스케이스 구조화



- 모든 사용자 역할 식별
- 상호작용하는 타 시스템 식별
- 정보를 주고받는 하드웨어 및 지능형 장치 식별

- 액터가 요구하는 서비스 식별
- 액터가 요구하는 정보 식별
- 액터가 시스템과 상호작용 하는 행위를 식별

- 액터와 액터 관계분석 정의
- 액터와 유스케이스 관계분석 정의
- 유스케이스와 유스케이스 관계분석 정의

- 두 개이상의 유스케이스에 존재하는 공통서비스 추출
- 추출된 서비스를 유스케이스 정의
- 조건에 따른 서비스 수행부분 분석하여 추출
- 추출된 서비스를 유스케이스로 정의
- 추출된 서비스를 사용하는 유스케이스와 관계 정의

Copyright © 2009 Nextree Soft, All rights reserved

작성 및 주의사항 (2/2)

❑ 유스케이스 다이어그램 작성 주의사항

- 액터와 유스케이스 명은 직관적으로 이해 가능
- 사용자 관점이라는 전제조건 인식
- 모든 유스케이스는 반드시 하나의 이상의 액터와 관계를 가짐
- 유스케이스의 추상화 레벨은 비슷한 수준으로 함
- 유스케이스의 크기단위는 일정

전체정보를 피드백한다(X), 활동을 탐색한다(X), 신규회원정보를 등록한다(O)
Data Window 출력(X), 회원정보 검색(O)
include/extend 관계의 유스케이스는 직접 연결하지 않을 수도 있다.
고객을 관리하다 VS 고객정보를 입력한다 회계시스템과 인터페이스하다 VS 과정수강정보를 회계시스템에 제공
파일관리 VS 파일저장/파일열기

Copyright © 2009 Nextree Soft, All rights reserved

2. 액티비티 다이어그램

ONE STEP AHEAD

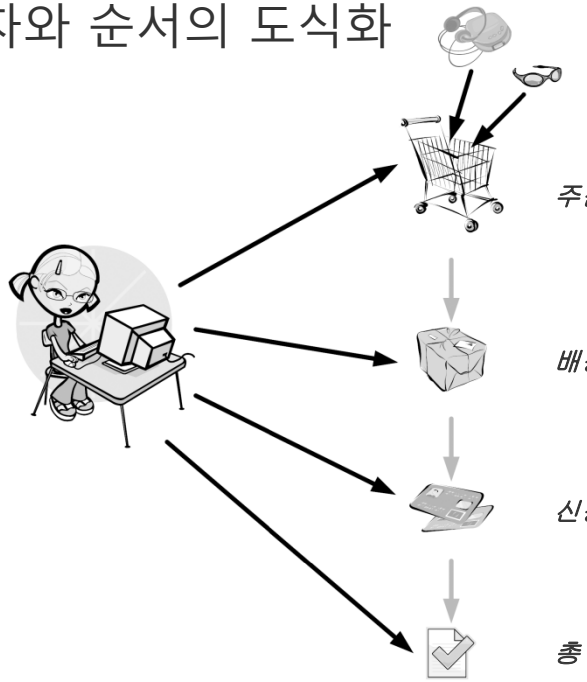
- ❑ 개요
- ❑ 구성요소
- ❑ 작성 및 주의사항

Copyright © 2009 Nextree Soft, All rights reserved

개요 (1/4)

❑ 절차와 순서의 메타포*Metaphor*

- 물품구매 시 장바구니 기능
  - 주문할 품목을 추가하고 이 단계를 수행하여 품목을 계속 추가 가능
  - 배송방법, 목적지 주소, 보증 등에 대한 배송정보 작성
  - 신용카드 정보 및 청구서 주소 등의 결재정보 입력
  - 총 결재 금액 등을 포함한 주문완료 및 주문내역을 재검토
- 절차와 순서의 도식화



주문할 품목을 추가한다. 사용자는 이 단계를 수행하여 품목을 계속 추가할 수 있다.

배송 방법, 목적지 주소, 보증 등에 대한 배송정보를 작성한다.

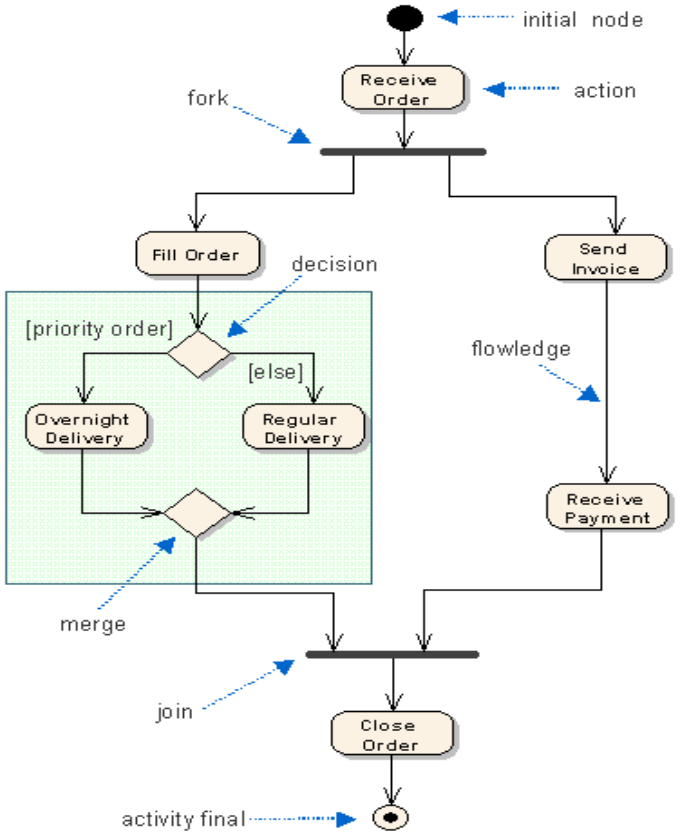
신용카드 정보 및 청구서 주소 등의 결재정보를 입력한다.

총 결재 금액 등을 포함한 주문 완료 및 주문내역을 재 검토한다.

Copyright © 2009 Nextree Soft, All rights reserved

개요 (2/4)

- 업무영역이나 시스템영역에서 다양하게 존재하는 각종 처리로직
- 조건에 따른 처리흐름을 순서에 입각하여 정의한 모델



Copyright © 2009 Nextree Soft, All rights reserved

개요 (3/4)

❑ 작성목적

- 대상에 상관없이 처리 순서를 표현하기 위해 작성
- 비즈니스 프로세스를 정의
- 프로그램 로직을 정의
- 유스케이스 실현*Realization*

❑ 작성시기

- 업무 프로세스 정의 시점
- 유스케이스 정의 작성 시점
- 오퍼레이션 사양 정의 시점
- 기타 처리흐름이나 처리절차가 필요한 시점

Copyright © 2009 Nextree Soft, All rights reserved

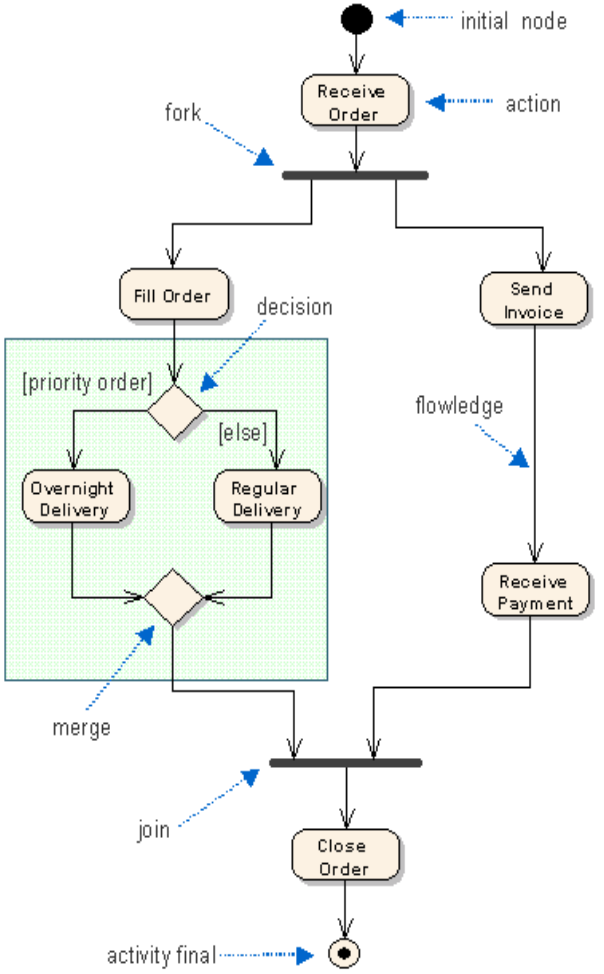
개요 (4/4)

- **UML 1.X**
  - 상태 다이어그램의 특별한 형태로서 액티비티 다이어그램을 취급
  - 절차적인 로직이나 비즈니스 프로세스, 워크 플로우를 표현하기 위한 기법
- **UML 2 : fork와 join이 반드시 매칭 되어야 한다는 룰 제거**
  - 상태의 변화보다 flow에 의해 액티비티 다이어그램을 이해
  - 새로운 표기법으로 표현
  - 타임 시그널(time signals), 수신 시그널(accept signals), 매개변수(parameter), 조인 명세서(join spec.), 서브다이어그램 표기(subdiagram rakes)
- **다차원을 나타내는 swim lanes는 파티션(partition)으로 불림**

Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (1/4)

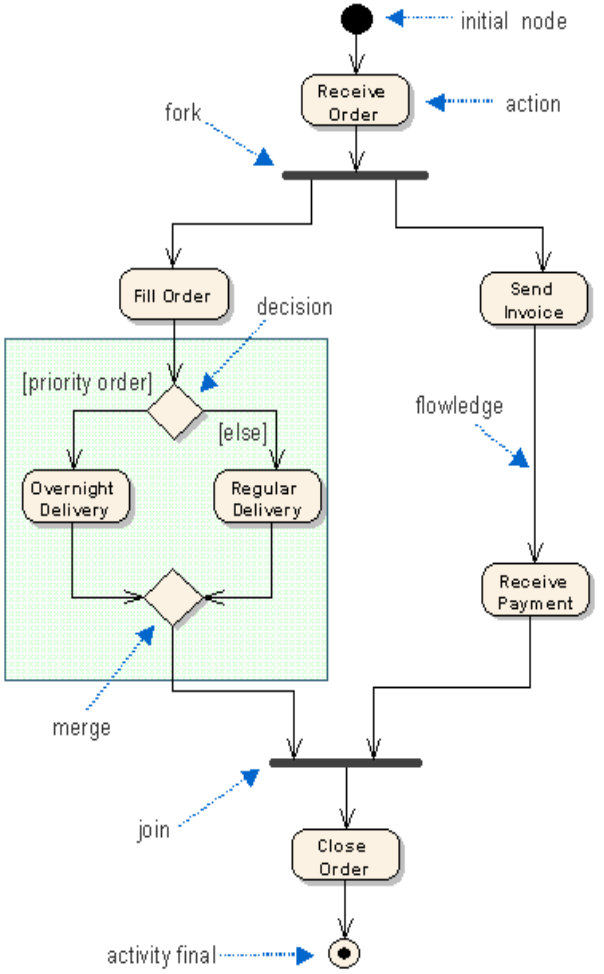
- **액티비티 다이어그램의 구성요소**
  - UML 1: 액티비티(activity), 분기(branch)
  - UML 2: 액션(action), 결정(decision)
- **상호작용 다이어그램과 액티비티 다이어그램**
  - 상호작용 다이어그램은 객체에서 객체로의 제어흐름 강조
  - 액티비티 다이어그램은 액션에서 액션으로의 제어흐름 강조
- **액티비티 다이어그램과 플로우차트(Flowchart)**
  - 플로우차트: 순차처리만 표현
  - 액티비티 다이어그램: 병렬처리 역시 표현



Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (2/4)

- **조건행동**
  - 결정(decision): 하나의 입력과 조건에 의한 다수의 출력전이로 구성
  - 병합(merge) : 여러 입력과 하나의 출력전이로 구성
- **병렬행동**
  - 포크(Fork): 하나의 입력전이와 여러 출력전이를 동시에 처리
  - 조인(Join): 여러 입력전이와 하나의 출력전이, 포크의 마무리

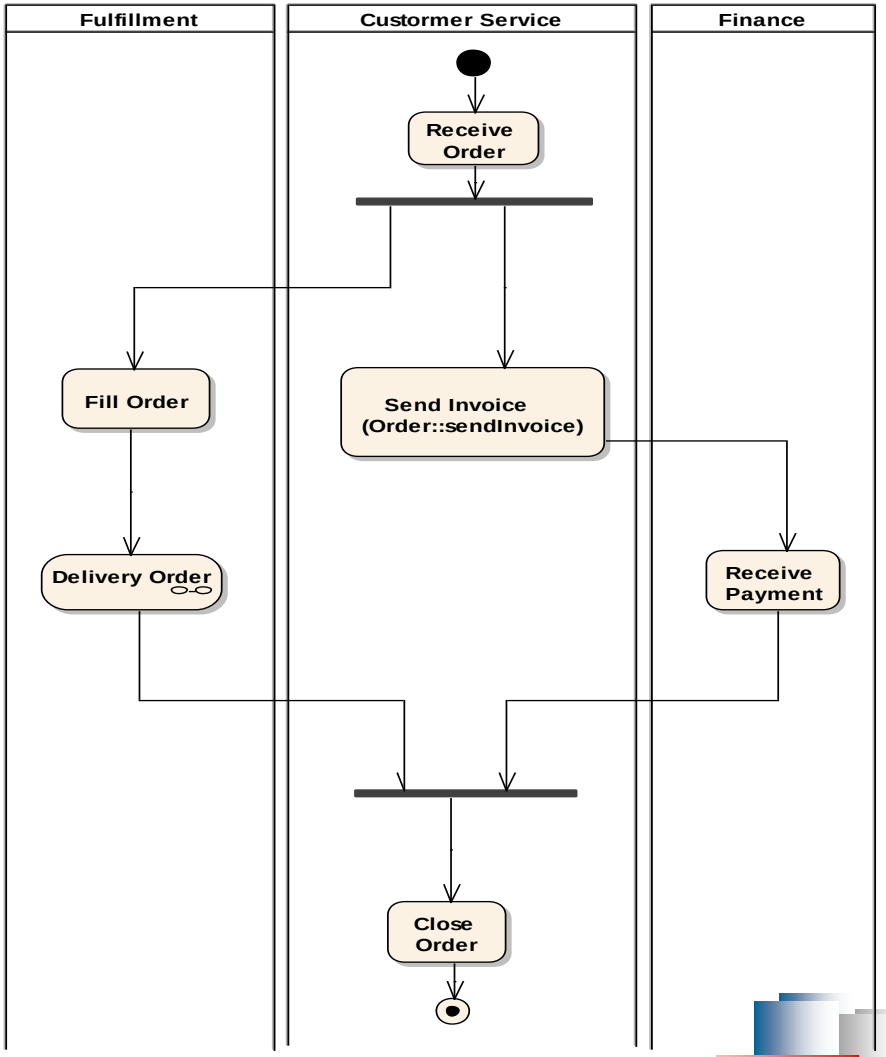


Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (3/4)

❑ 파티션 *Partition*

- 일반적 이슈
  - 액티비티 다이어그램은 어떠한 액션을 수행하는지 알 수 있지만 누가 수행하는지 모호
  - 프로그래밍 관점에서 어느 클래스가 어떤 액션을 수행하는지 그 책임을 알 수 없음
  - 비즈니스 프로세스 모델링 관점에서 각 액션을 책임지는 사람이나 부서를 알 수 없음
- 해결책
  - 각 액션의 책임 클래스 또는 사람을 표기
  - 파티션을 사용하여 책임성 부여
- 표기법
  - UML 1.X : swimlanes
  - UML 2 : partitions

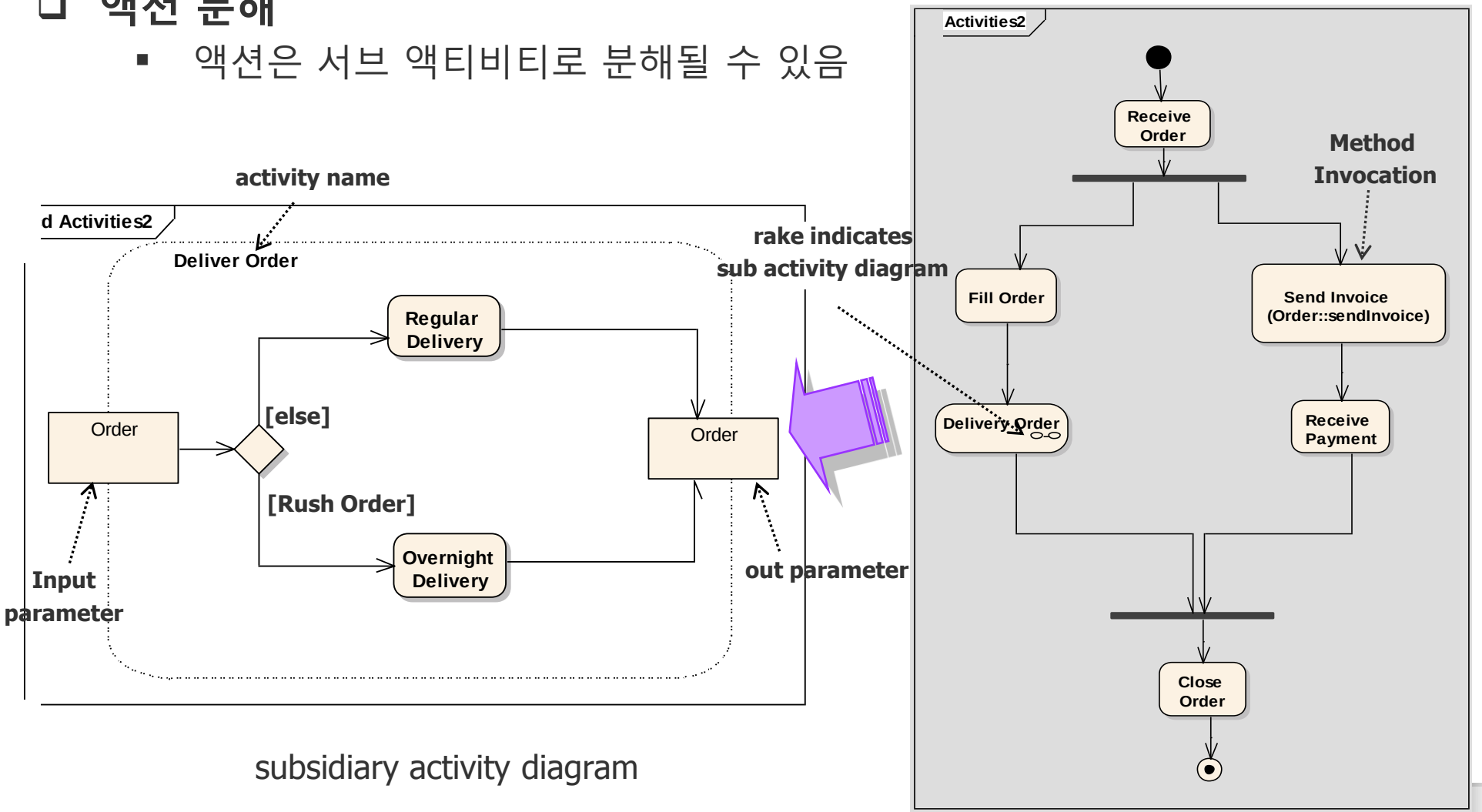


Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (4/4)

❑ 액션 분해

- 액션은 서브 액티비티로 분해될 수 있음

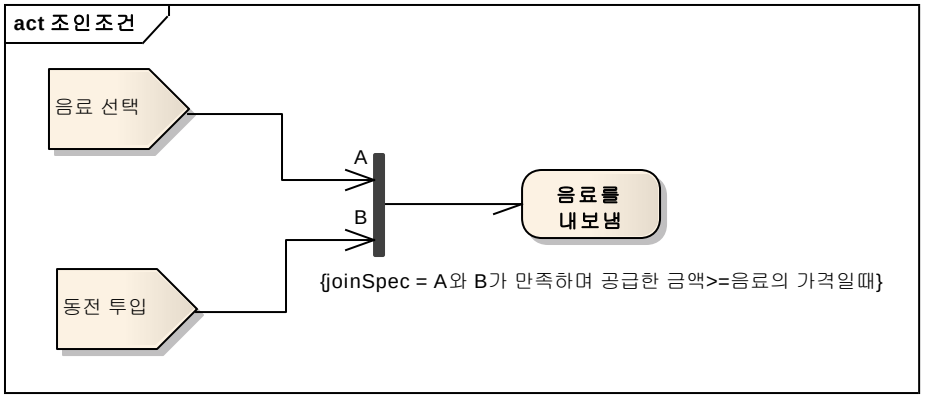


subsidiary activity diagram

Copyright © 2009 Nextree Soft, All rights reserved

❑ 조인 조건*Join specification*

- 조인은 모든 입력 플로우가 조인에 도착했을 때 나가는 방향의 플로우 실행
- 토큰이 조인에 도착할때마다 조인 조건을 검사

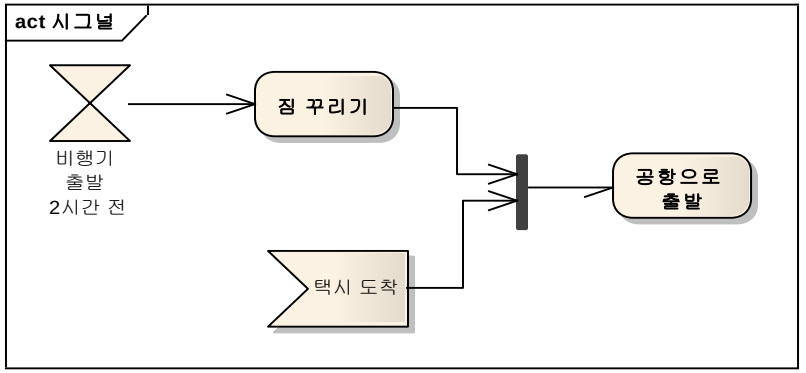


Copyright © 2009 Nextree Soft, All rights reserved

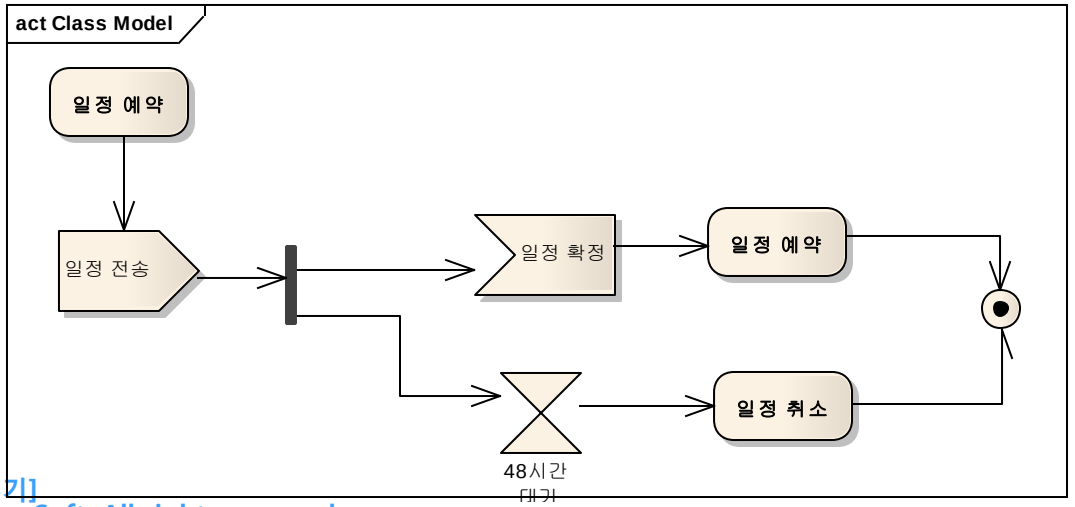
구성조건^시간 ^시그널(Time signal)

❑ 시간 시그널 *Time signal*

- 시간의 흐름에 따라서 발생



- 시그널은 액티비티가 외부의 프로세스로부터 이벤트를 수신함을 표현



Copyright © 2009 Nextree Soft, All rights reserved

작성 및 주의사항 (1/2)

❑ 액티비티 다이어그램 작성 단계

- 작성 대상을 선정
- 필요한 경우 파티션 정의
- 액티비티를 사용하여 처리절차 모델링

단계	내용
작성대상 선정	액티비티 다이어그램의 작성 대상을 선정한다. 대부분의 경우 액티비티 다이어그램은 업무 프로세스를 모델링 하거나, 오퍼레이션 사용을 정의하는 용도로 사용된다.
파티션 정의	대상영역에 명확한 역할을 정의할 수 있을 경우, 역할을 식별하여 파티션으로 표현한다. 파티션은 필수적으로 정의 해야하는 대상은 아니다.
처리절차 모델링	처리절차를 모델링 할 경우 시작점과 끝점이 표현되어야 하고 처리흐름이 도중에 끊겨 미아상태가 되지 않아야 한다.

Copyright © 2009 Nextree Soft, All rights reserved

작성 및 주의사항 (2/2)

❑ 액티비티 다이어그램의 작성 시 주의사항

- 해당 부분을 이해하는데 필수적인 요소들만 표현
- 추상화 수준에 맞는 상세성을 일관되게 제공
- 중요한 의미를 이해하기 적절한 단위로 표현
- 목적을 전달할 수 있는 명칭의 부여
- 주 흐름으로 부터 시작하여 전이, 분기, 동시성을 표현
- 교차선이 최소화하도록 요소를 배치
- 중요한 부분은 노트, 색 등을 이용하여 시각적 효과를 사용

Copyright © 2009 Nextree Soft, All rights reserved

3. 상태기계 다이어그램

ONE STEP AHEAD

- 개요
- 구성요소
- 작성 및 주의사항

Copyright © 2009 Nextree Soft, All rights reserved

개요 (1/3)

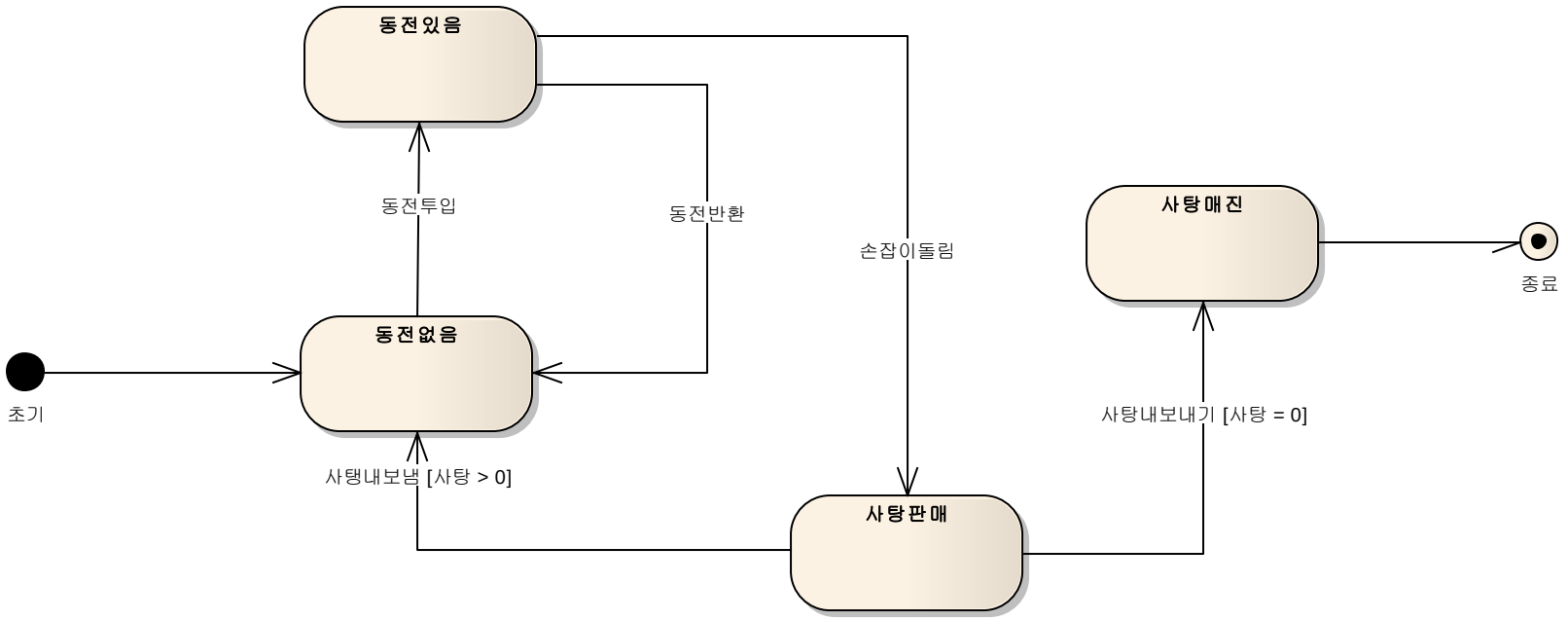
❑ 메타포*Metaphor*

- 사탕 뽑기 기계 상태
  - 사탕을 뽑기 위해 동전을 투입하여 동전있음 상태로 전이
  - 동전이 있는 상태에서 반환버튼을 누르면 동전없음 상태로 바뀜
  - 동전이 있는 상태에서 손잡이를 돌리면 사탕판매가 됨
  - 사탕판매 상태에서 사탕을 내보냄
  - 사탕이 나온 후에 캡슐 내에 사탕이 존재하면 동전없음 상태로 전이
  - 사탕이 나온 후에 캡슐 내에 사탕이 존재하지 않으면 사탕매진 상태로 됨
- 시스템 내의 객체 상태를 파악하고 도식화

Copyright © 2009 Nextree Soft, All rights reserved

개요 (2/3)

- 단일 클래스의 생명주기 동안 객체의 상태의 변화를 분석
- 객체의 상태와 함께 객체 상태 변화를 유발하는 이벤트와 동작*action/activity* 도 함께 정의



Copyright © 2009 Nextree Soft, All rights reserved

개요 (3/3)

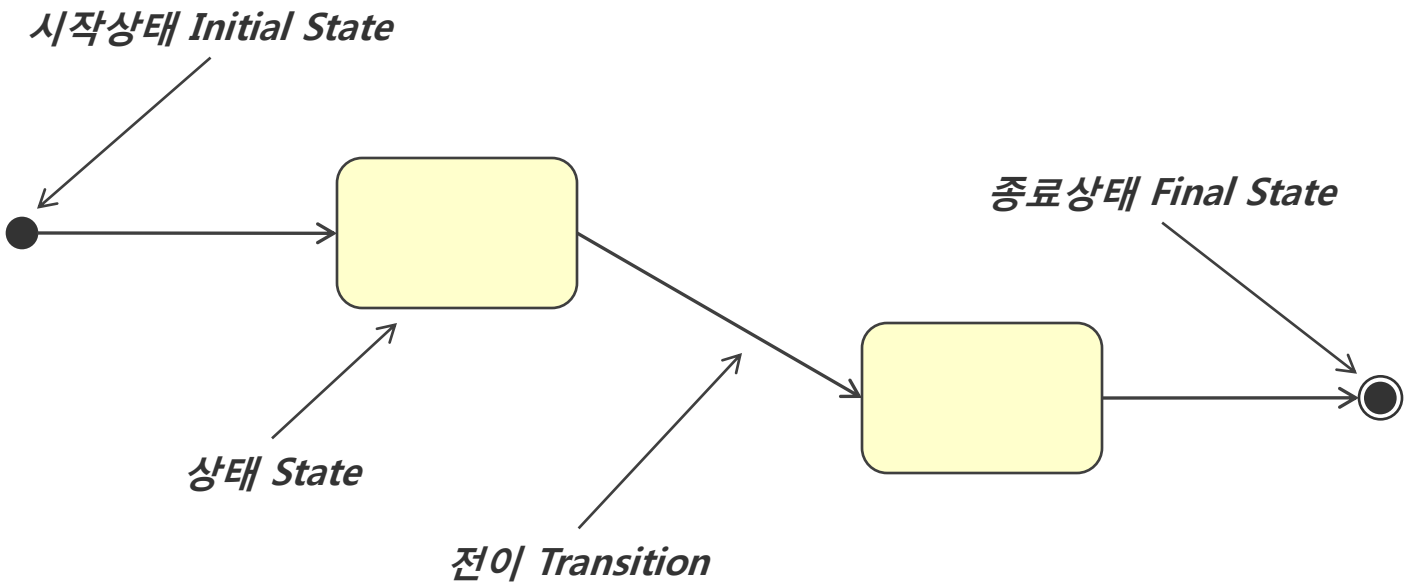
- **작성목적**
  - 객체의 상태변화를 상세히 분석
  - 이벤트에 의한 객체의 반응을 분석
  - 객체의 속성이나 오퍼레이션을 검증
- **작성시기**
  - 특정 시기를 한정하기 어려움
  - 일반적으로 클래스 정의 후 클래스 다이어그램이나 시퀀스 다이어그램 작성이 완료되면 작성됨

Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (1/6)

❑ 상태기계 다이어그램 구성요소

- 요소: 상태*State*, 시작상태*Initial State*, 종료상태*Final State*
- 관계: 전이*Transition*



Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (2/6)

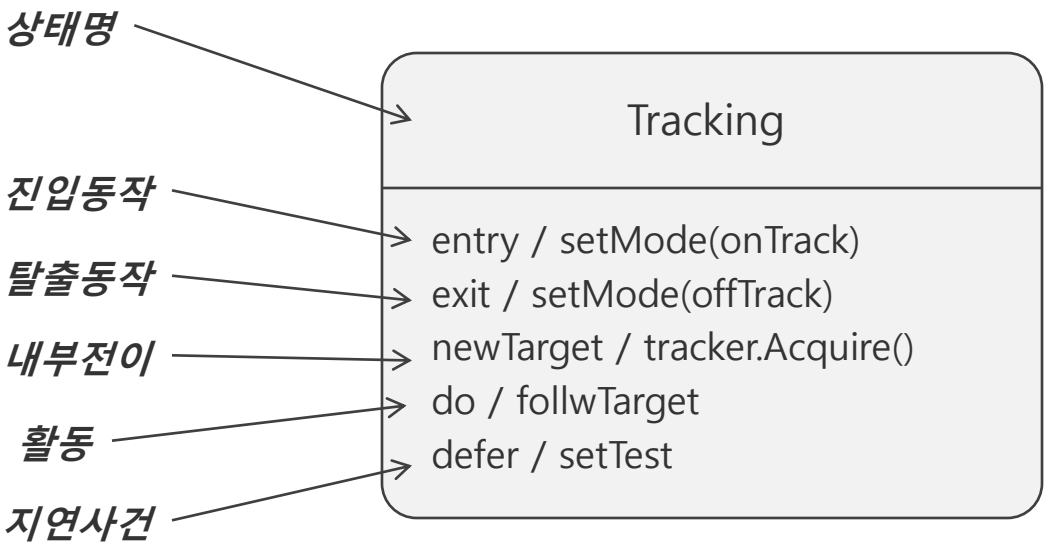
- **시작상태** *Initial State*
  - 속이 꽉 채워진 원으로 표기
  - 객체의 상태변화가 시작되는 곳을 의미
  - 일반적으로 객체의 생성시점이 시작상태가 됨
- **종료상태** *Final State*
  - 속이 꽉 채워진 원에 바깥의 또 다른 원이 둘러싸고 있는 모양으로 표기
  - 객체의 상태변화가 종료되는 곳을 의미
  - 일반적으로 객체의 소멸시점이 종료상태가 됨

Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (3/6)

❑ 상태*State*

- 객체가 가질 수 있는 조건이나 상황
- 생명주기 동안 객체의 상태는 변화
- 상태는 객체의 속성값으로 표현
  - 예) 자동차의 객체 상태: 주차, 주행, 정차, 수리 …



Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (4/6)

❑ 상태*State*

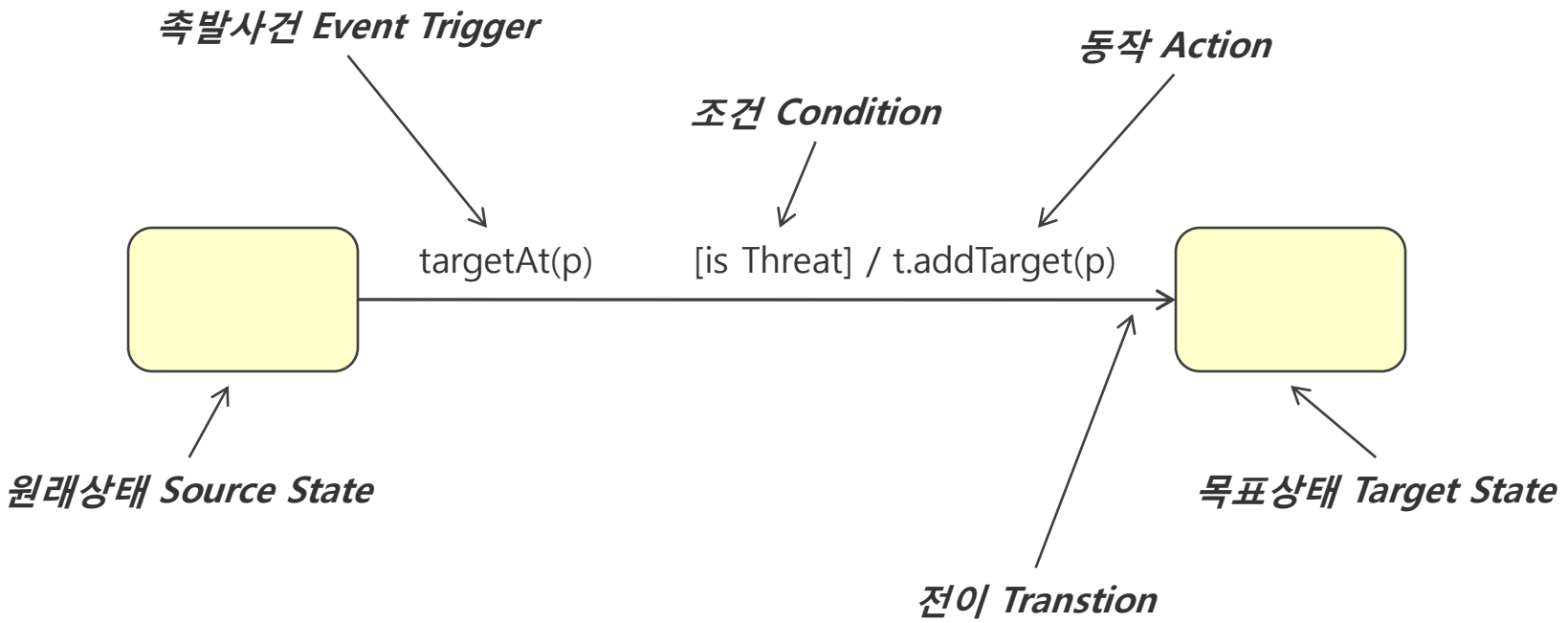
- 진입동작
  - 상태에 들어올 때 수행되는 동작을 정의
- 탈출동작
  - 상태에 나갈 때 수행되는 동작을 정의
- 내부전이
  - 현재 상태에서 처리할 수 있는 이벤트가 발생할 경우 상태를 떠나지 않고 해당 이벤트를 처리하는 경우
- 활동
  - 현 상태에서 수행할 동작을 표현
- 지연사건
  - 현 상태를 빠져 나갈 때 발생한 것처럼 그 효과를 지연시킨 이벤트
  - 예에서 Tracking 상태에서 setTest 이벤트가 발행하면 이것을 메이지 큐에 저장했다가, Tracking 상태에서 벗어난 순간 이벤트가 활성화 됨

Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (5/6)

❑ 전이 *Transition*

- 하나의 상태에서 다른 상태로 변화하는 것
- 상태 간의 관계를 표시



Copyright © 2009 Nextree Soft, All rights reserved

구성요소 (6/6)

- 전이 *Transition*
  - 원래상태
    - 전이가 실행되기 전의 객체 상태
  - 촉발사건
    - 전이를 촉발시키는 사건
  - 전이조건
    - 전이 촉발 시에 검토되는 Boolean 식
    - 참일 경우에만 전이가 수행됨
  - 동작
    - 전이 도중 실행되는 행위 또는 오퍼레이션
  - 목표상태
    - 전이가 완료된 후의 객체 상태

Copyright © 2009 Nextree Soft, All rights reserved

작성 및 주의사항 (1/2)

❑ 상태기계 다이어그램 작성 단계

- 작성 대상 객체를 선정
- 객체가 가지는 상태를 정의하여 나열
- 상태와 상태 간의 전이를 정의
- 전이와 상태의 상세한 부분을 정의

단계	내용
대상 객체 선정	상태기계 다이어그램의 작성 대상 객체를 선정한다. 분석할 객체는 다소 복잡한 상태변화를 가지는 객체이다.
상태 정의 및 나열	객체의 생명주기를 분석하면서 어떤 상태를 가지는 지를 먼저 식별한다. 식별된 상태는 여러 번의 검토를 통해 합쳐지거나 더 상세한 것으로 분리한다.
전의 정의	하나의 상태를 기준으로 해서 전이할 수 있는 상태를 정한다. 전이는 모든 상태에서 연결되어야 한다. 일반적으로 어떠한 상태에서도 전이를 통해 종료점까지 도달할 수 있다.
상세부분 정의	전이의 경우 촉발사건, 전이조건, 동작 등을 정의하고 상태의 경우 진입동작, 탈출동작, 내부전이, 활동, 지연사건 등을 정의한다. 상화에 따라 상세내용은 생략하거나 필요한 것만 정의할 수 있다.

Copyright © 2009 Nextree Soft, All rights reserved

작성 및 주의사항 (2/2)

❑ 상태기계 다이어그램의 작성 시 주의사항

- 객체 하나에 대한 상태 변화를 표현
- 블랙홀 상태를 주의
- 클래스 다이어그램 및 시퀀스 다이어그램과의 일관성에 유의

단계	내용
상태 변화	상태기계 다이어그램의 작성 중에 상태에 집중하다 보면 객체라는 한계를 벗어난 경우가 발생한다. 객체 하나의 상태변화와 여기에 관련된 이벤트들을 모델링하는 것에 충실해야 한다.
블랙홀 상태	상태기계 다이어그램에 표현되는 상태는 들어오는 전이와 나가는 전이가 모두 정의되어야 한다. 만약, 들어오는 전이만 있고 나아가는 전이가 없을 경우, 그 상태는 블랙홀이 된다. 이런 실수를 하면 객체가 종료 상태에 이르지 못하고 무한 루프를 수행하는 오류를 범하게 된다.
일관성 유의	상태기계 다이어그램을 작성 완료한 후 새롭게 정의된 오퍼레이션과 속성은 클래스 다이어그램과 시퀀스 다이어그램에 반영되어 일관성을 유지해야 한다. 그런 작업이 없을 경우 모델에 대한 신뢰성이 떨어지게 된다.

Copyright © 2009 Nextree Soft, All rights reserved