

# 시스템 분석과 설계

개정판

효과적인 비즈니스 정보시스템 개발

## Chapter 14 UML 모델링

# 목차

**01** UML의 구조

**02** 정적 모델링 도구

**03** 동적 모델링 도구

**04** UML 모델링 절차

# 학습목표

- UML의 구성요소를 이해한다.
- UML의 정적 모델링 도구를 학습한다.
- UML의 동적 모델링 도구를 학습한다.
- UML을 활용한 분석 설계 과정을 이해한다.

- 가시화
- 명세화
- 구축
- 문서화

## ■ UML의 구성요소

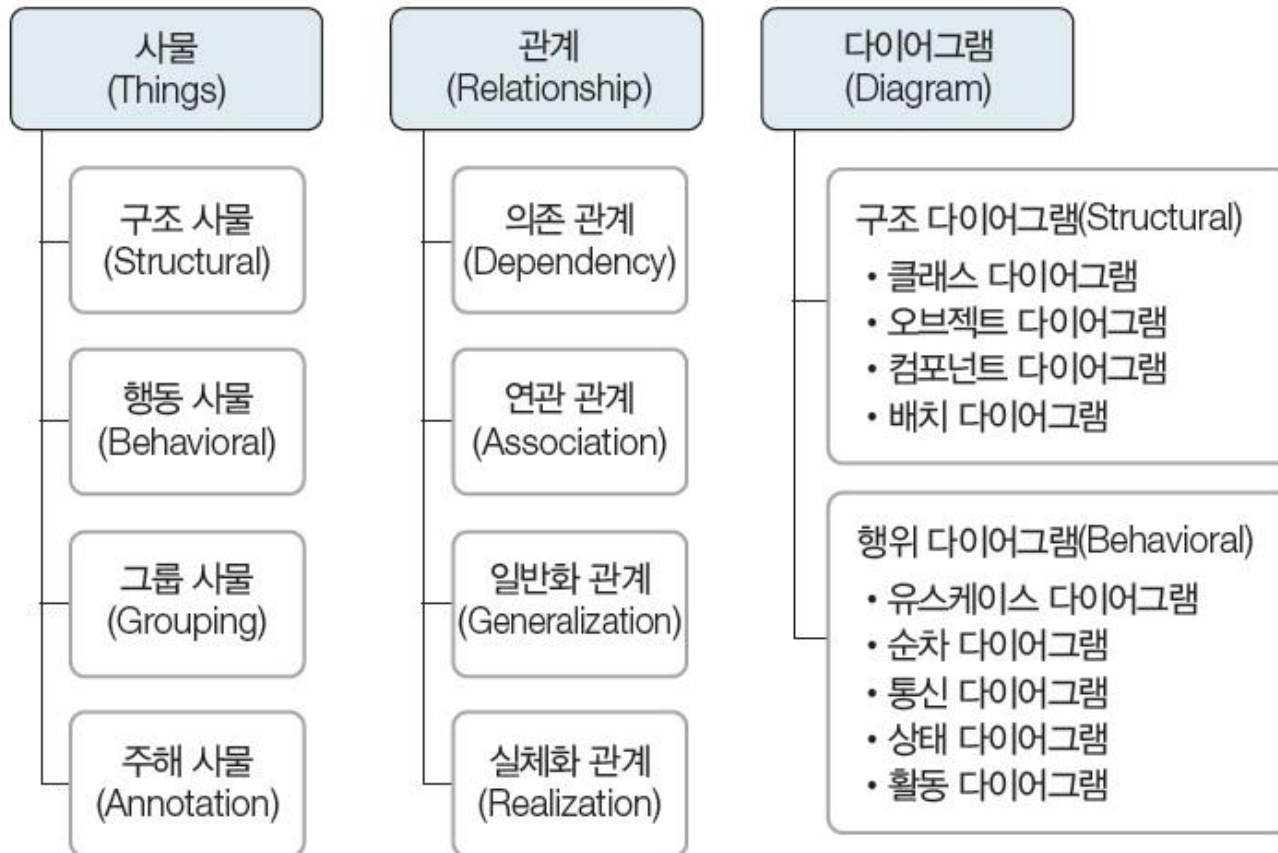


그림 14-1 UML의 구성요소

### ■ 사물

- ① 구조 사물(Structural Things)
- ② 행동 사물(Behavioral Things)
- ③ 그룹 사물(Grouping Things)
- ④ 주해 사물(Annotation Things)

### ■ 관계

#### ① 의존 관계(Dependency Relationship)

예) TV와 리모컨

#### ② 연관 관계(Association Relationship)

예) 자동차와 부품

#### ③ 일반화 관계(Generalization Relationship)

'is-a' 관계라고도 함

예) 부모 클래스로서의 자동차와 자식 클래스로서의 택시, 버스, 트럭 등

#### ④ 실체화 관계(Realization Relationship)

한 객체가 다른 객체에 의해 오퍼레이션을 수행하도록 지정하는 것

예) TV의 행동 중 일부가 리모컨의 행동을 '실체화(Realize)' 함

클래스(TV)와 인터페이스(리모컨)가 가지는 관계가 실체화 관계

### ■ 다이어그램

- 구조 다이어그램 : 정적 모델링을 위한 다이어그램
- 행위 다이어그램 : 동적 모델링을 위한 다이어그램



### ■ 네 가지 측면

- 정적 측면 : 모델링 대상 범위에서 객체의 구조와 관계를 나타냄
- 동적 측면 : 이벤트가 발생함에 따라 객체의 상태 변화를 나타냄
- 기능적 측면 : 사용자의 측면에서 본 시스템의 행동을 나타냄
- 물리적 측면 : 시스템 실행을 위해 필요한 컴퓨터와 기억 매체의 공간적인 배치 기술

### ■ 세 가지 레벨

- 개념 레벨 : 문제 영역(도메인)의 해석을 기록
- 사양 레벨 : 설계 작업
- 구현 레벨 : 개발 작업

# 1.3 UML 모델링의 이해

## ■ UML 다이어그램의 작성 분류

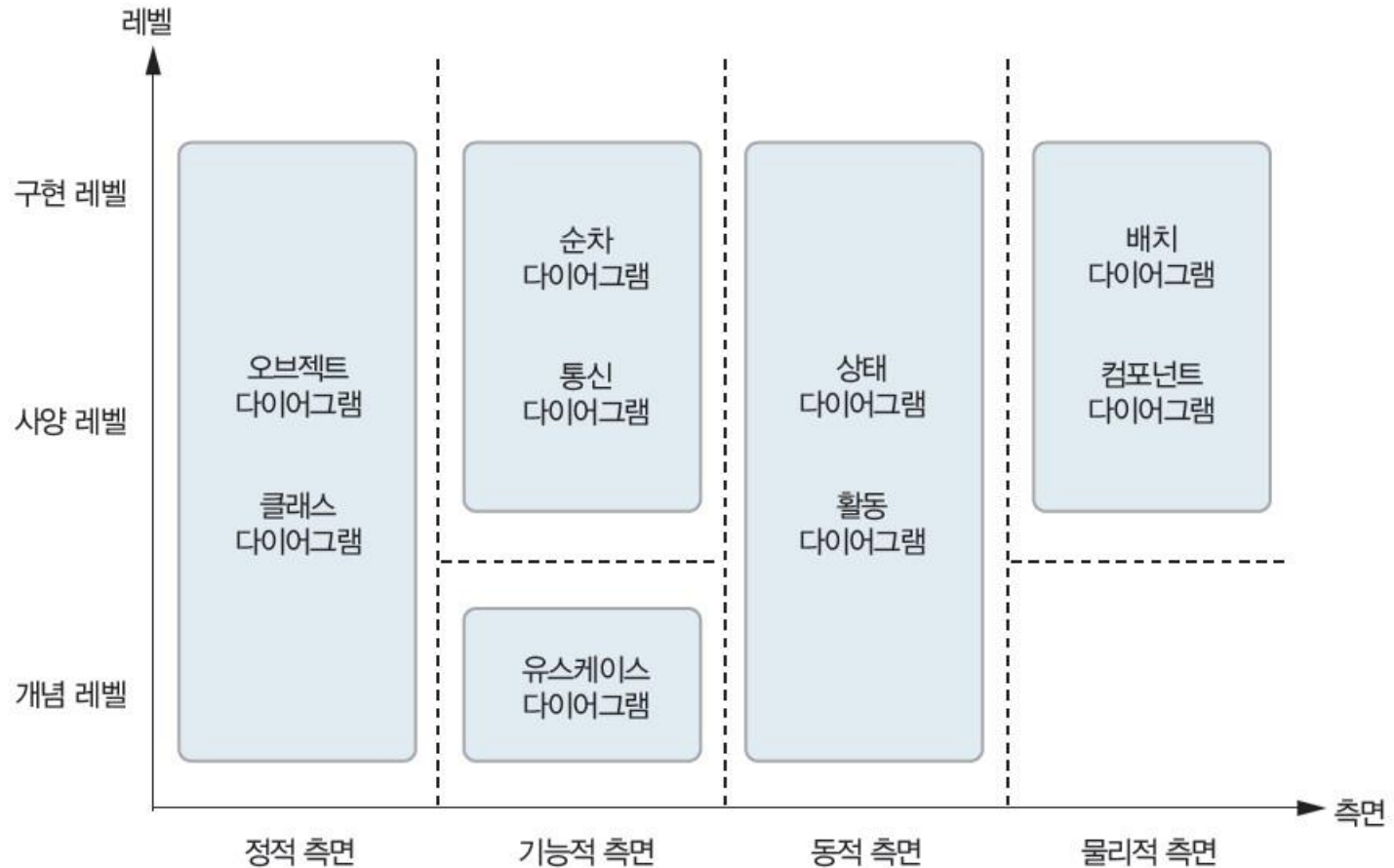


그림 14-2 UML 다이어그램의 작성 분류

## ■ UML 1.x와 UML 2.0의 비교

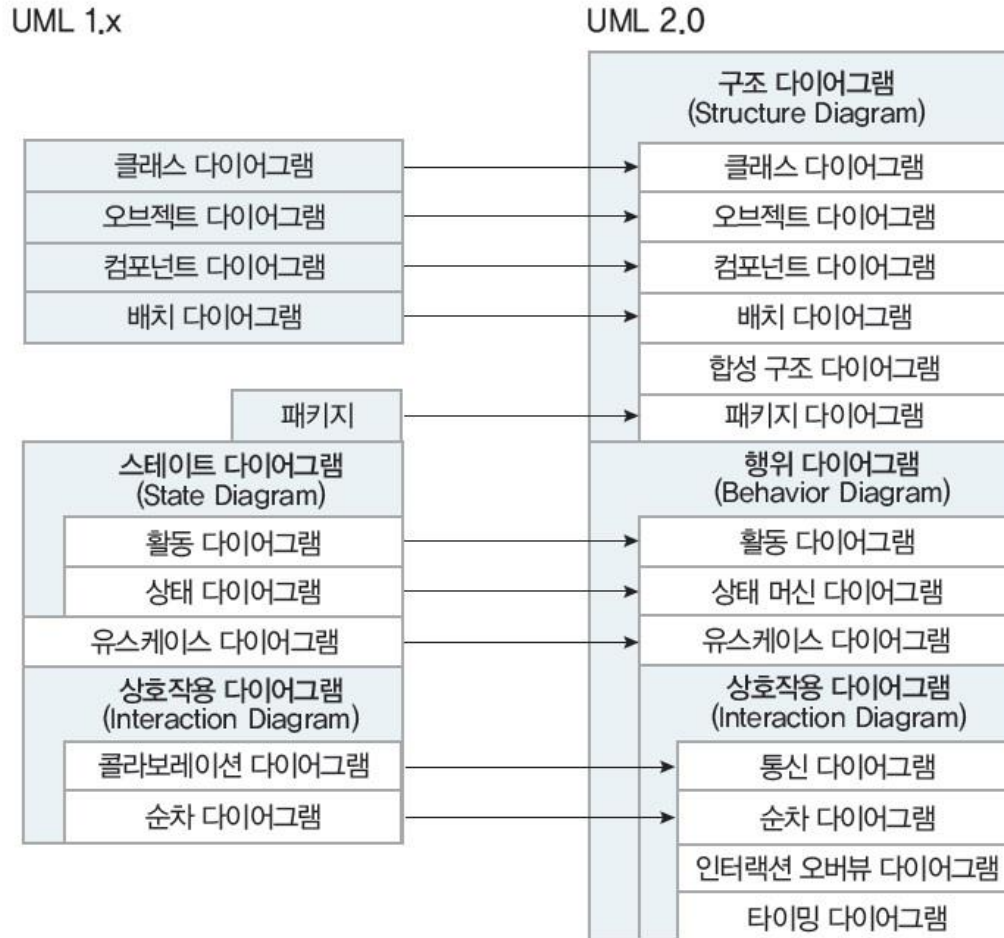


그림 14-3 UML 1.x와 UML 2.0의 비교

### ■ 정적 모델링 도구

- UML의 구조 다이어그램에 해당
- 시스템의 지속적이고 정적인 측면을 모델링하는 특징이 있음

### ■ 클래스의 구성요소

- 이름
- 속성
- 메서드(오퍼레이션)

이름	은행계좌
속성	예금주 : 문자열 잔고 : 원 = 0
메서드	입금하다(예금액) 출금하다(출금액)

그림 14-4 클래스 다이어그램의 예

### ■ 클래스 다이어그램 작성 절차

- ① 클래스 명세를 결정하고,
- ② 메서드(오퍼레이션)를 찾아낸 후,
- ③ 필요한 속성을 추출한다.

### ■ 클래스 다이어그램 작성 예시

#### 공공도서관 도서대출 업무 명세

- 도서를 대출하기 위해서는 회원자격을 획득해야 한다. 회원이 되려면 회원가입 절차를 통해 자신의 기본정보를 입력하고 도서관 이용규정에 동의해야 한다.
- 회원가입이 완료되면 회원증이 발급되며, 회원증을 이용해 도서를 대출할 수 있다.
- 회원은 한 번에 최대 5권의 도서를 대출받을 수 있으며, 대출기간은 2주로 제한한다.
- 회원은 원하는 도서를 찾아 대출창구에 회원증과 함께 제출하면 대출받을 수 있다. 도서를 반납하려면 해당도서를 반납창구에 제출하여 확인을 받으면 된다.

### ① 클래스 후보 추출

- 클래스 후보를 추출하기 위한 가장 일반적인 방법은 업무 명세서에서 명사를 찾아내는 것

회원, 회원가입, 회원증, 도서, 대출, 반납



### ② 클래스와 클래스 사이의 관계 규정

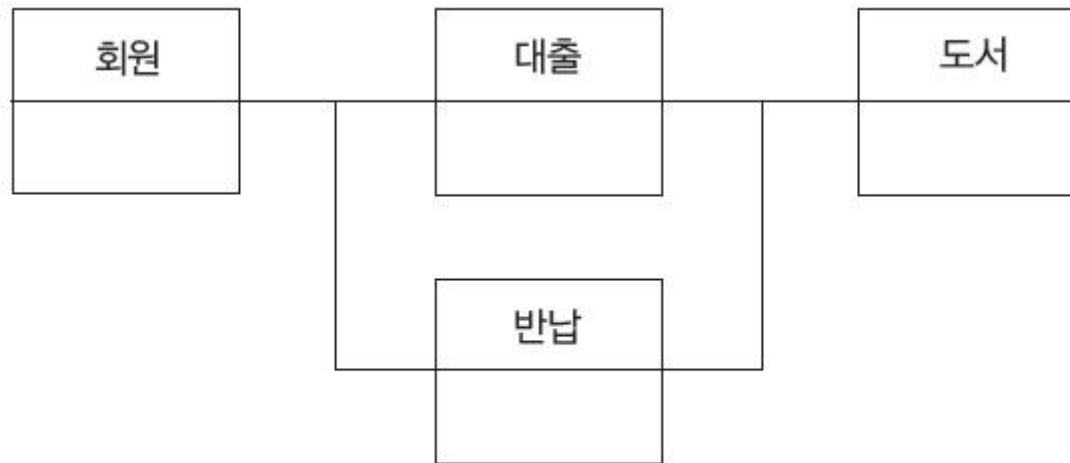


그림 14-5 클래스 간 관계도

### ③ 클래스의 메서드와 속성 규정



그림 14-6 클래스 속성과 메서드

### ③ 클래스의 메서드와 속성 규정

#### ■ 개선된 클래스 다이어그램

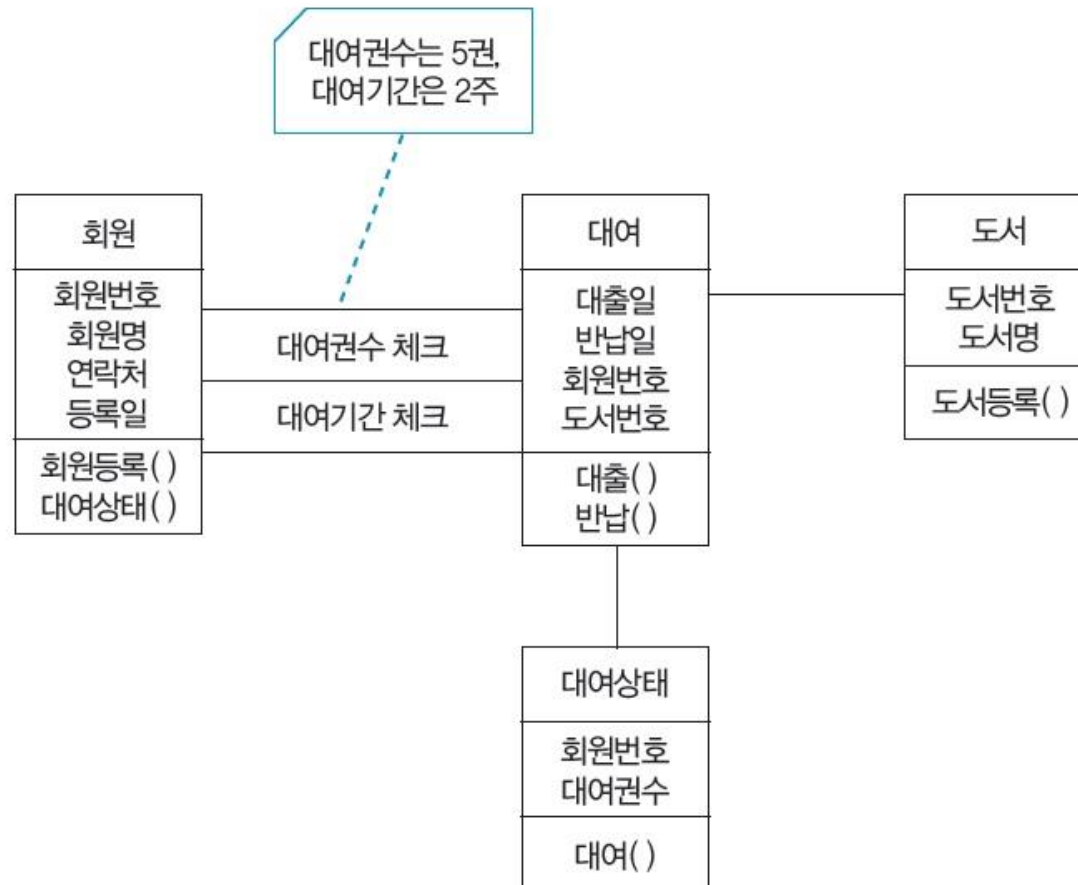


그림 14-7 제약사항을 반영한 클래스 다이어그램

### ■ 오브젝트 다이어그램

- 오브젝트 다이어그램은 특정 시점의 오브젝트들의 구조적 상태를 표현

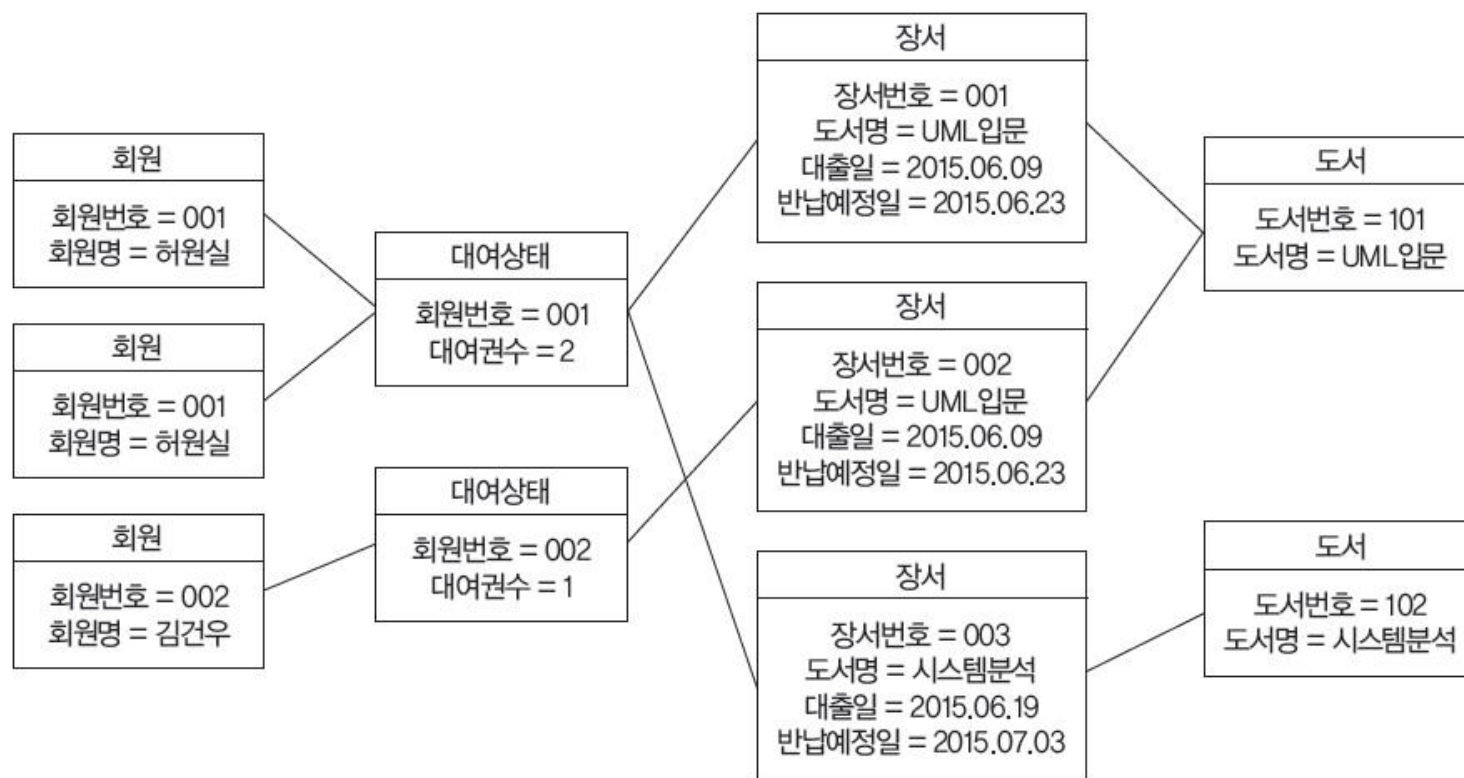


그림 14-8 도서대출 시스템의 오브젝트 다이어그램 작성 예

### ■ 컴포넌트

- 인터페이스에 의해서 기능이 정의된, 독립적으로 개발 · 배포 · 조립 가능한 시스템의 구성단위
- 예) J2EE 플랫폼의 JAR 파일, 닷넷 플랫폼의 DLL 파일

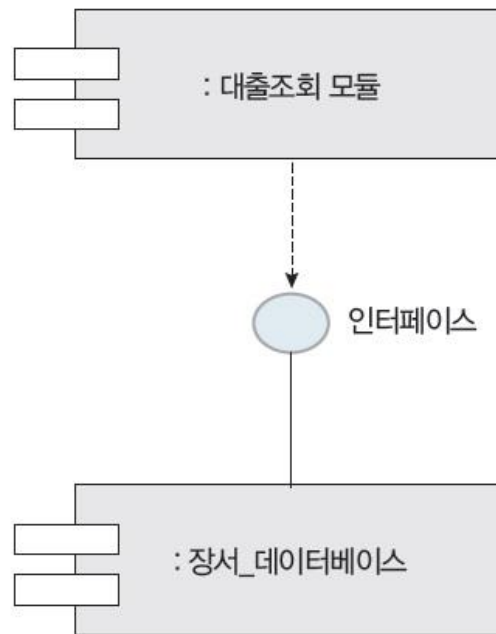


그림 14-9 컴포넌트 다이어그램 예시

### ■ 배치 다이어그램

- 노드를 입체적으로 표현하고, 그 사이를 의존 화살표와 접속 관계를 나타내는 실선으로 연결해 이들 간의 통신 관계를 나타낸 것
- 노드란 시스템 실행 시 필요한 물리적인 처리 자원과 실행 모듈, 소프트웨어 컴포넌트의 인스턴스 등을 뜻함

## 2.4 배치 다이어그램

### ■ 배치 다이어그램의 작성 예시

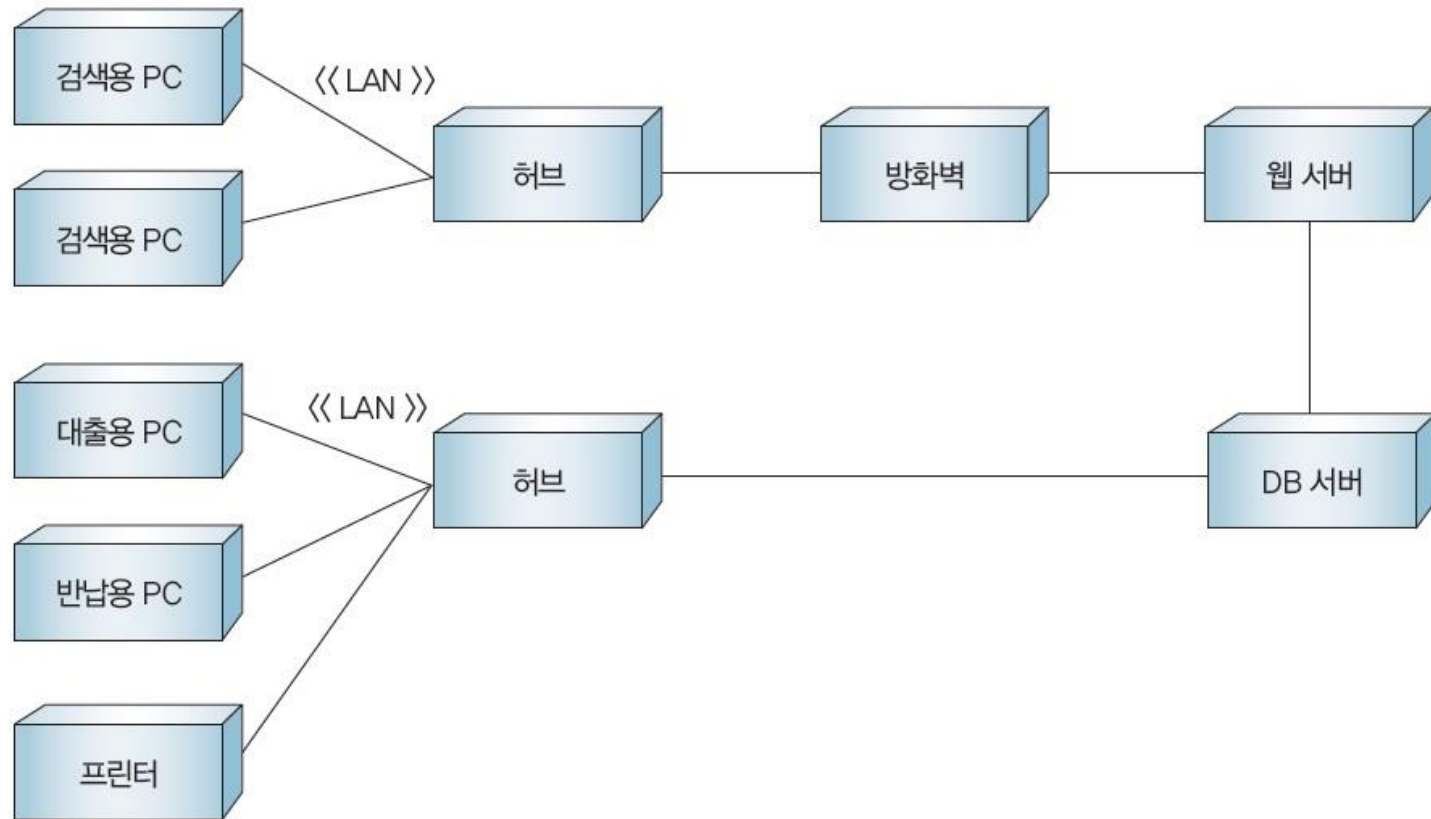


그림 14-10 배치 다이어그램 예시

### ■ 동적 모델링 도구

- UML의 행위 다이어그램과 인터랙션 다이어그램에 해당
- 시간의 흐름에 따라 유동적으로 변하는 객체의 상태나 행위, 객체 간의 상호작용 등을 표현



## 3.1 유스케이스 다이어그램

### ■ 유스케이스 다이어그램

- 액터의 관점에서 본 시스템의 기본적인 행동을 기술한 것

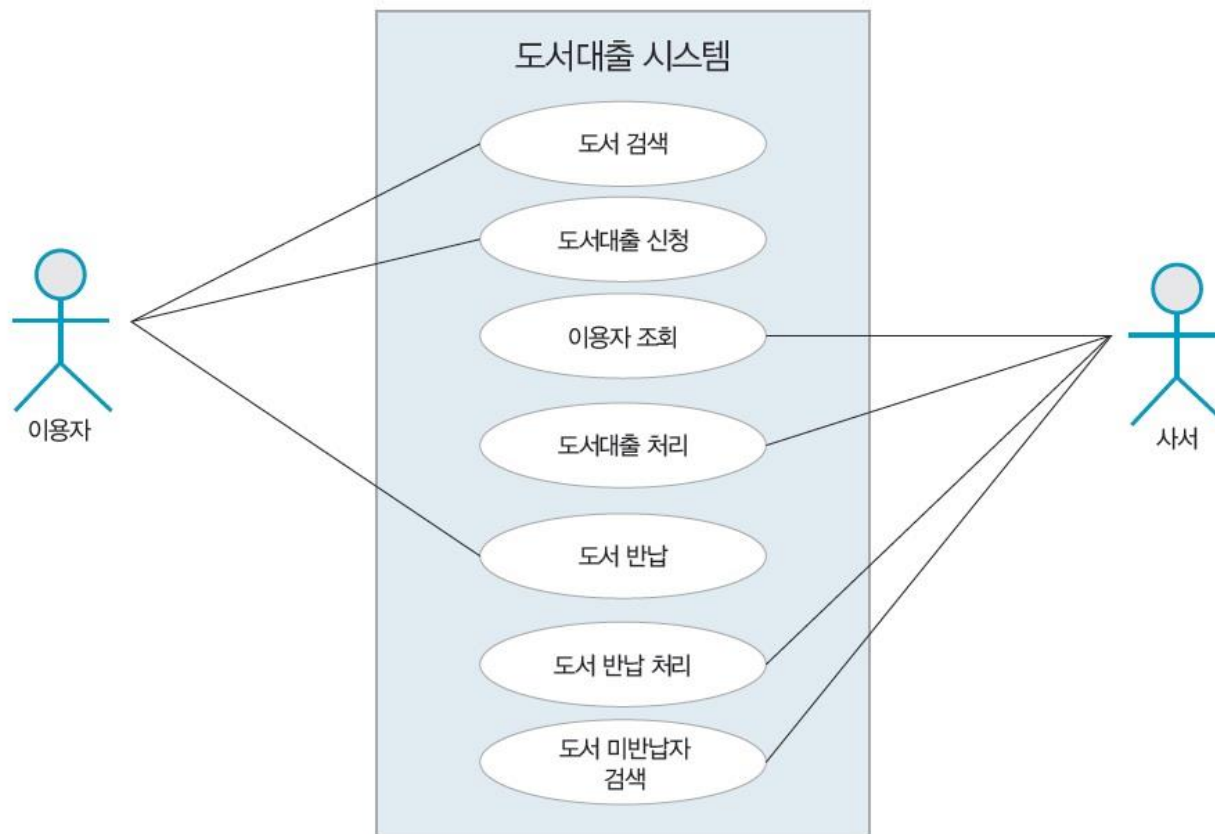


그림 14-11 유스케이스 다이어그램 예시

### ■ 순차 다이어그램

- 객체 간의 동적 상호작용을 시간의 흐름에 따라 나타낸 것

### ■ 순차 다이어그램의 특징

- 객체의 메서드(오퍼레이션)와 속성을 상세히 정의
- 객체의 책임 : 순차 다이어그램의 객체는 다른 객체가 의뢰하는 일을 처리
- 유스케이스를 실현

## 3.2 순차 다이어그램

### ■ 순차 다이어그램의 예시

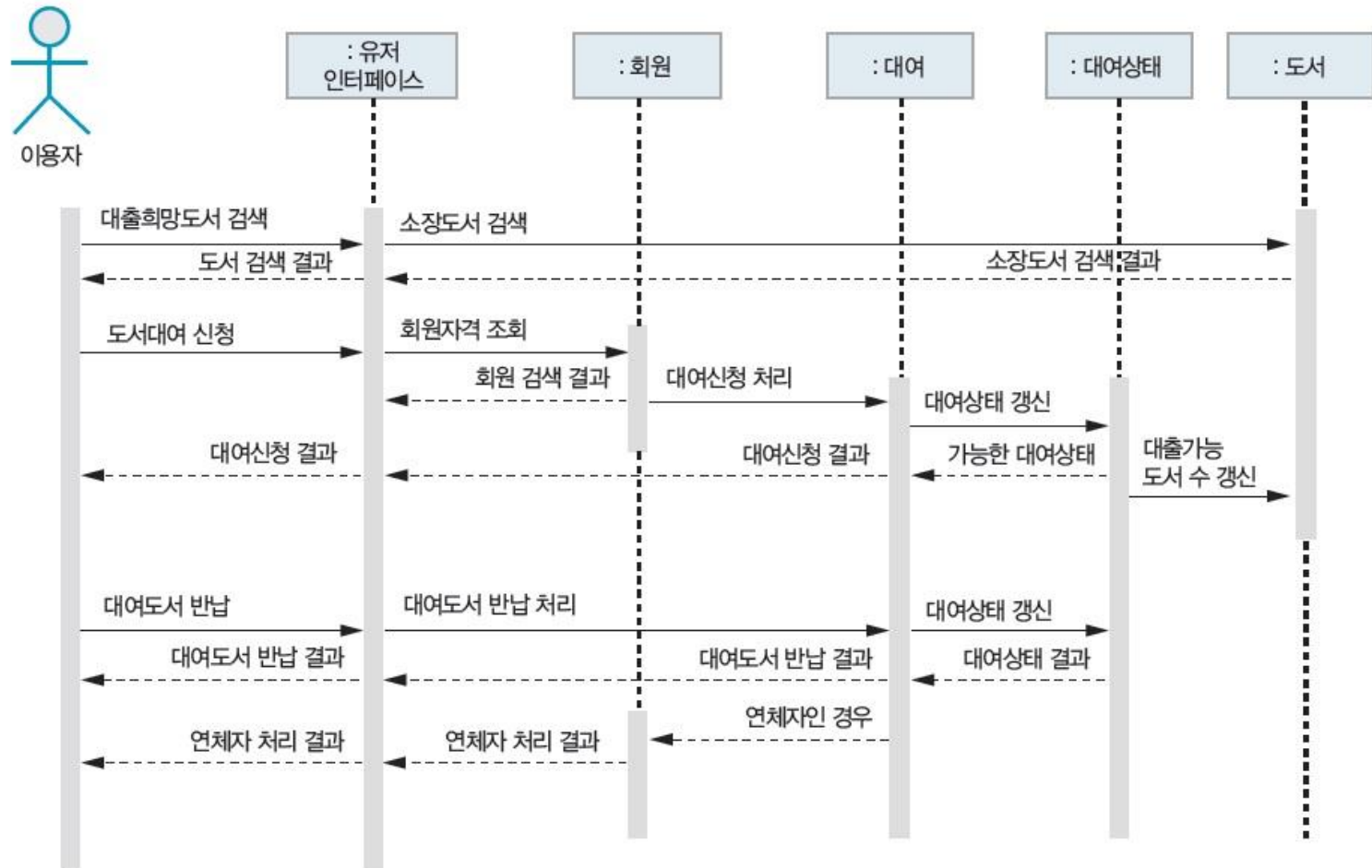


그림 14-12 순차 다이어그램 예시

## 3.3 통신 다이어그램

### ■ 통신 다이어그램

- 객체들 사이에 주고받는 메시지를 표현

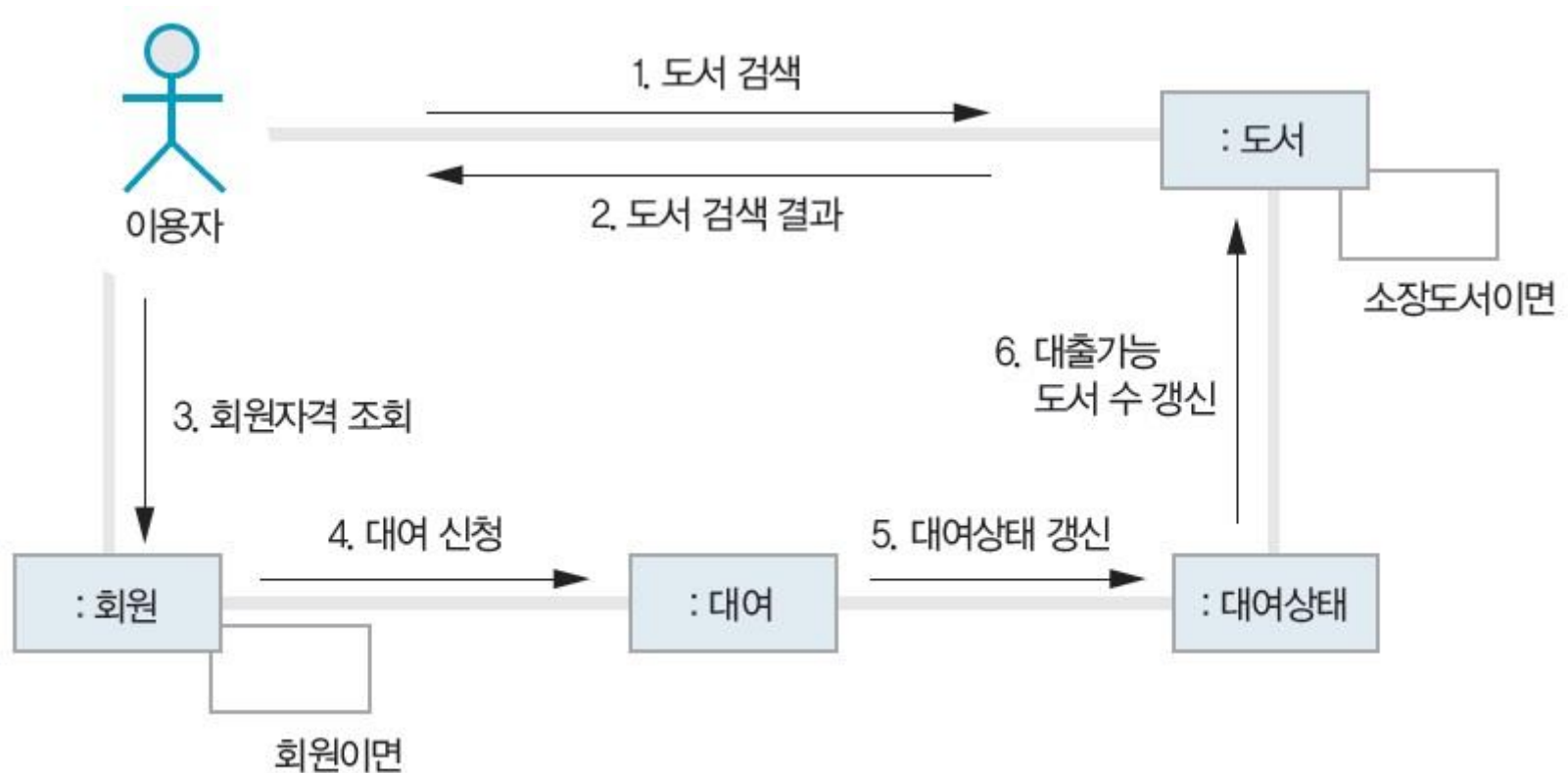


그림 14-13 통신 다이어그램 예시

### ■ 상태 다이어그램

- 객체의 상태가 이벤트의 발생 혹은 시간의 경과에 의해 어떻게 변화하는지를 나타낸 것
- 특정 객체가 생성하여 소멸할 때까지의 라이프 사이클을 모델화함

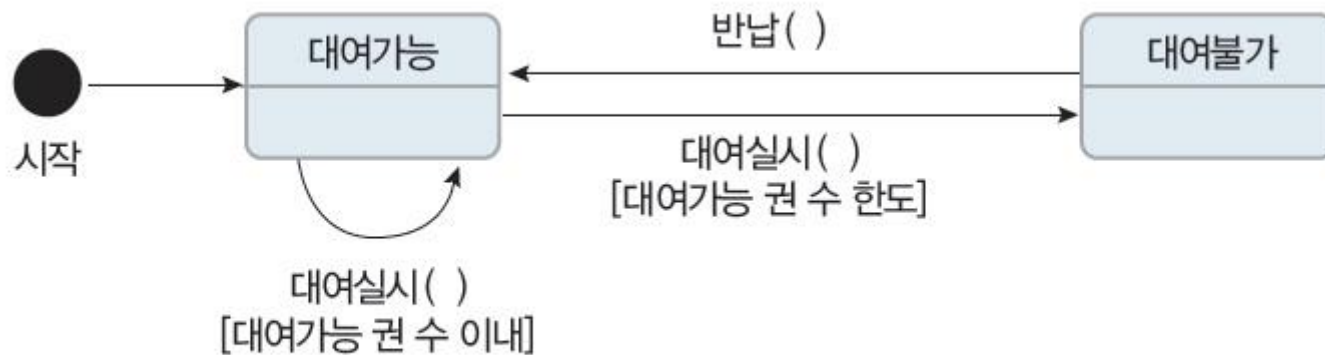


그림 14-14 상태 다이어그램 예시

### ■ 활동 다이어그램

- 예전부터 사용해왔던 순서도와 모양이 매우 비슷
- 순서도(어떤 행위에 대한 워크플로우를 표현하는 대표적 수단)를 객체지향 스타일로 개선시킨 형태

## 3.5 활동 다이어그램

### ■ 활동 다이어그램의 예시

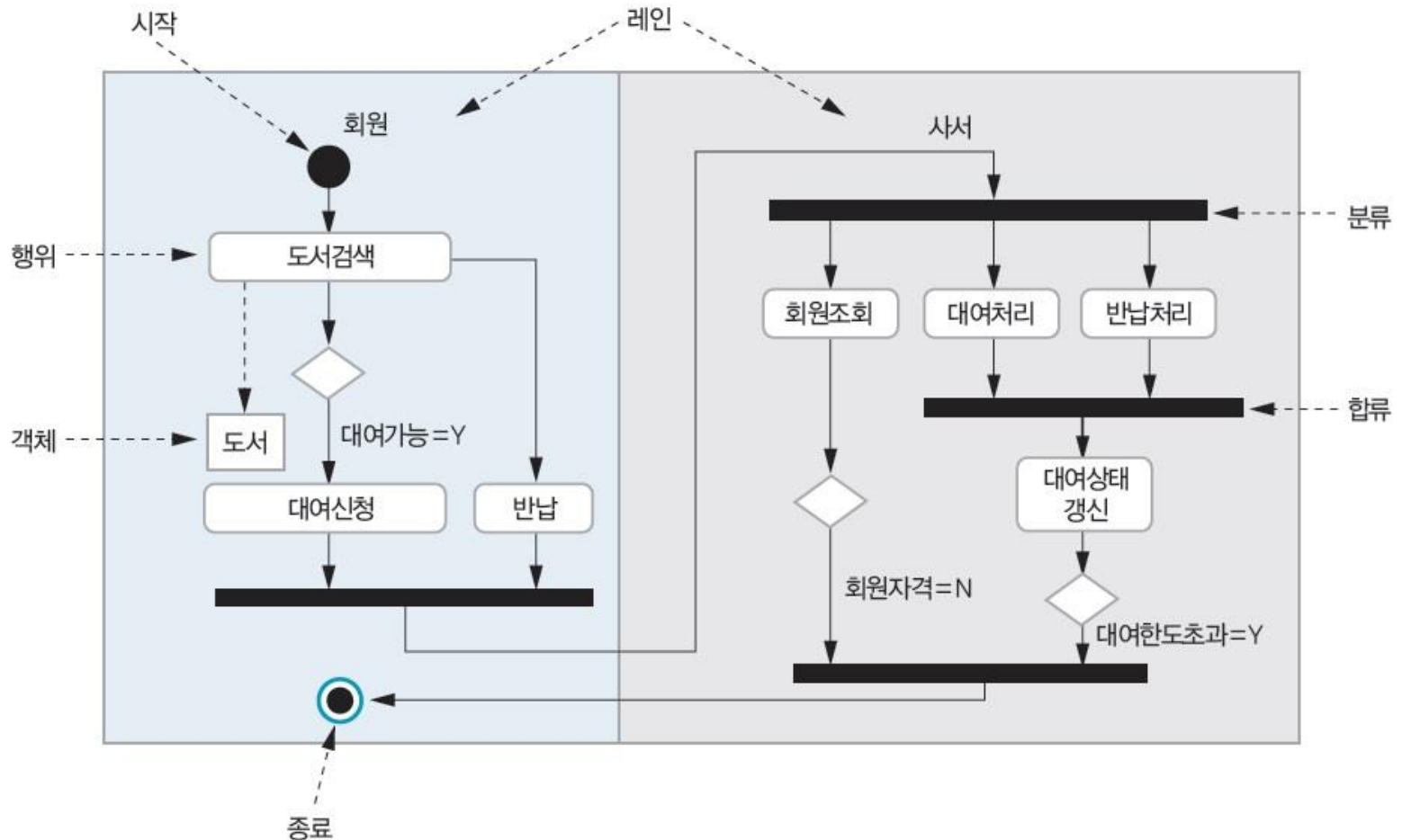


그림 14-15 활동 다이어그램 예시

### ■ UML 모델링의 절차

① 초기 클래스 다이어그램 작성



② 유스케이스 다이어그램 작성



③ 클래스 다이어그램 변경



④ 순차 다이어그램과 통신 다이어그램 작성



⑤ 인터페이스를 식별(클래스 다이어그램 변경)



⑥ 활동 다이어그램과 상태 다이어그램을 작성



⑦ 컴포넌트 다이어그램 및 배치 다이어그램 작성





Thank You

---