

UML

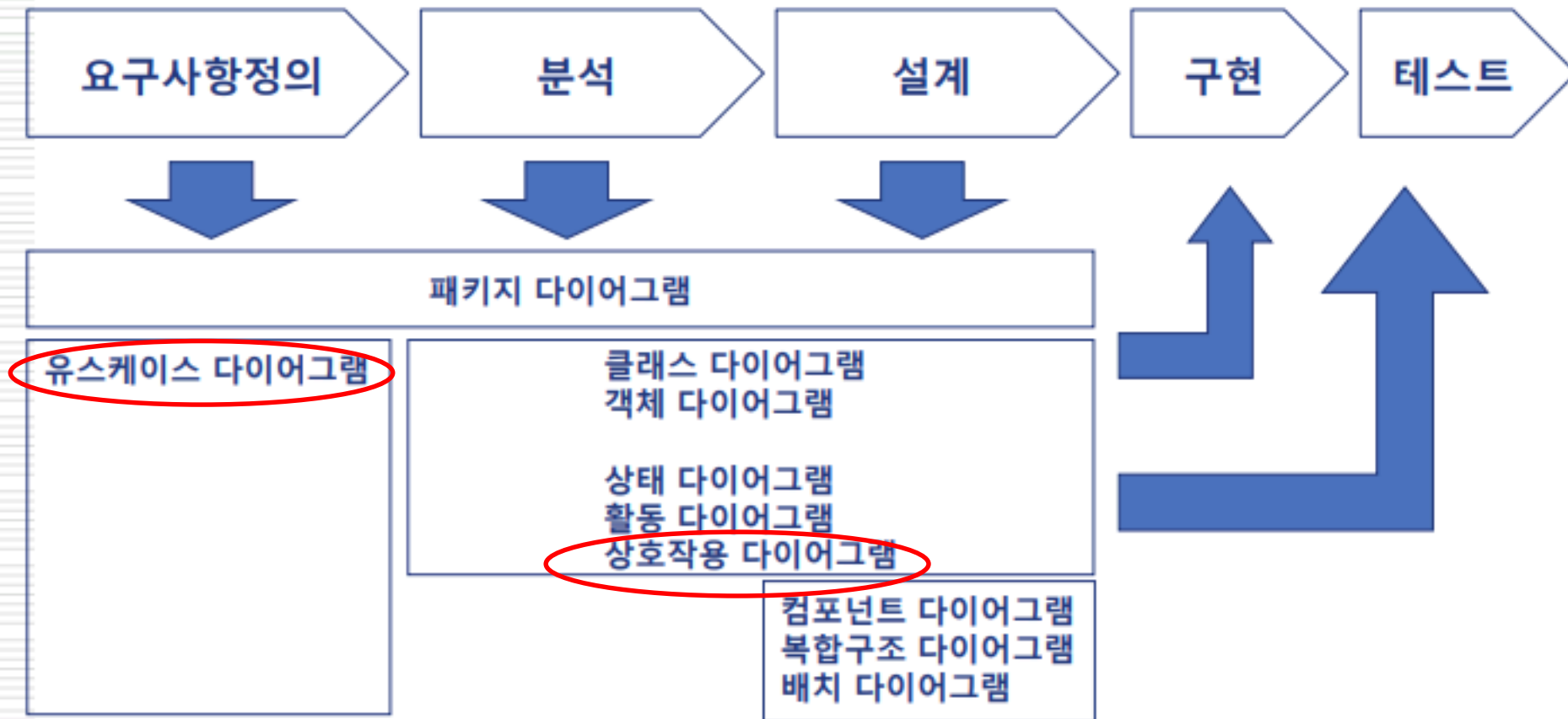
❖ 개발의 전 과정에서 사용됨



시스템 개발자가 구축하고자 하는 소프트웨어를 코딩하기에 앞서 표준화되고 이해되기 쉬운 방법으로 소프트웨어를 설계하여 다른 사람들과 효율적으로 의사소통할 수 있는 메커니즘을 제공한다.

UML

❖ 개발 단계 별로 주로 사용되는 다이어그램이 있음

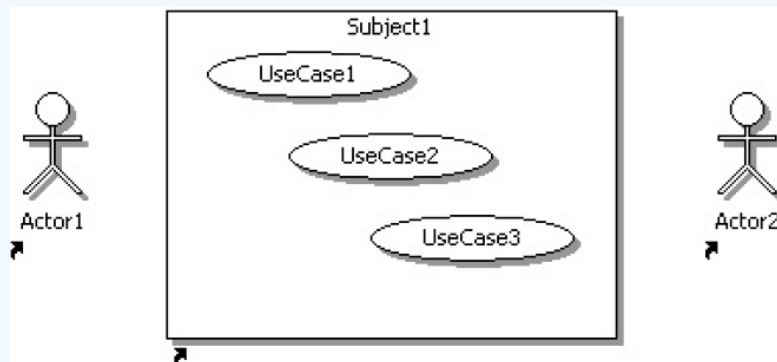




유스케이스 다이어그램

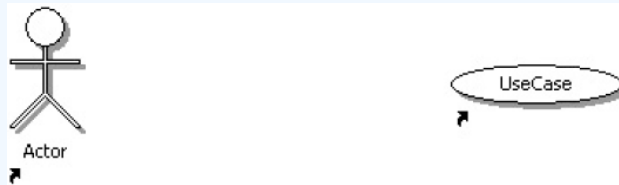
- 유스케이스 모델링과 UML 표기법
- 유스케이스와 유스케이스 관계
- 액터 사이의 일반화관계
- 액터 사이의 중복관계
- 유스케이스 모델링 절차
- 재고관리 시스템 유스케이스 모델링

- 시스템 개발에 참여하는 사람들을 큰 부류로 나누면,
 - ✓ 의뢰인, 개발팀, 사용자 (User)
 - 사용자의 관점을 빨리 이해해야만 쓸모 있고 유용한 시스템을 만들 수 있다.
 - 개발자는 가능한 한 모든 요구사항을 파악하여 사용자의 승인을 받아야만 후일 요구사항 변경에 대한 위험부담을 줄일 수 있다.
-
- [그림 3-1]
 - ✓ 네모난 창은 시스템의 경계
 - ✓ UseCase1, 2, 3은 구축할 시스템의 기능을 표현
 - ✓ 유스케이스 모델링에서 개발할 시스템 외부 존재를 Actor라는 개념으로 정의
 - ✓ 시스템의 범위에 해당되어 개발될 시스템의 단위 기능을 Usecase 개념으로 정의



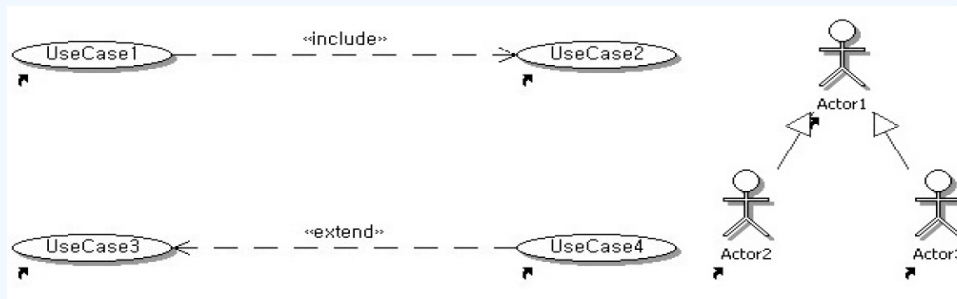
[그림 3-1] 시스템의 경계

- 유스케이스의 역할 : **사용자의 시점에서 시스템을 모델링**한다는 것
- 유스케이스 : 시스템에 대한 시나리오의 집합
- 이벤트의 흐름 : 사람, 시스템, 하드웨어, 혹은 시간의 흐름에 의해 시작
- 액터 : 이벤트 흐름을 시작하게 하는 객체



[그림 3-2] 유스케이스와 액터

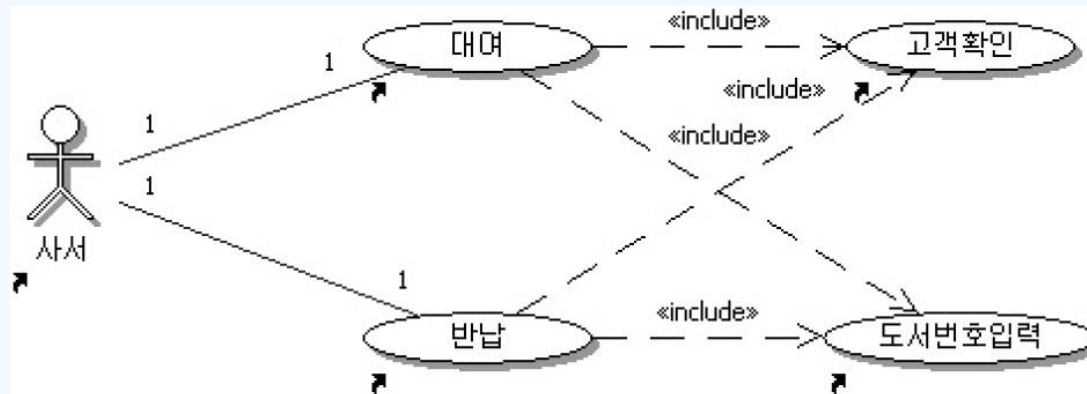
- 유스케이스 사이에는 일반적인 연관관계 이외에 또 다른 관계
 - ✓ 포함(Include)관계 : 다른 유스케이스에서 기존의 유스케이스를 재사용하는 관계
 - ✓ 확장(Extend)관계 : 기존의 유스케이스에 진행단계를 추가하여 새로운 유스케이스를 만들어내는 관계
 - ✓ 액터와 유스케이스에 대한 일반화관계



[그림 3-3] 유스케이스 관계

● 유스케이스간 포함관계

- 포함관계 : 유스케이스를 수행할 때 다른 유스케이스가 반드시 수행되는 것
- 유스케이스 다이어그램에서는 다른 유스케이스가 나타내는 이벤트 흐름을 포함(include)하는 관계를 유스케이스간에 표현



[그림 3-4]도서관 시스템의 포함관계

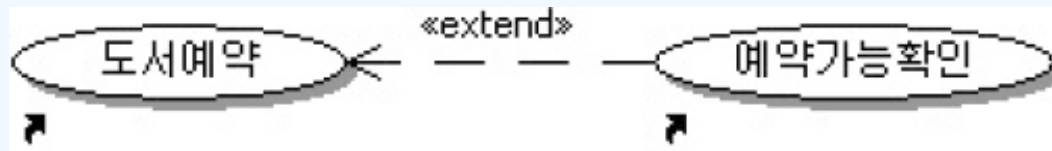
● 유스케이스간 확장관계

- 확장관계의 유스케이스는 포함관계처럼 여러 유스케이스에 걸쳐 중복적으로 사용되지 않고, 특정 조건에서 한 유스케이스로만 확장되는 것을 의미
- 확장(extend)하는 유스케이스는 상위 유스케이스로부터 어떠한 특정 조건에 의해 수행됨을 의미
- 포함관계와 확장관계의 차이점
 - ✓ 포함관계 : 여러 유스케이스에서 **공통**적으로 발견되는 기능을 표현
 - ✓ 확장관계 : 한 유스케이스에서 추가되거나 **확장**된 기능을 표현



[그림3-5] 금액계산 유스케이스의 확장

- 확장 유스케이스는 특정 조건이 만족되는 상황에서만 확장 유스케이스의 이벤트 흐름이 수행
- 확장 유스케이스의 이벤트 흐름의 수행 여부를 결정짓는 조건은 확장 유스케이스를 포함하는 기준 유스케이스가 아니라 확장 유스케이스에 표현되는 점을 주목

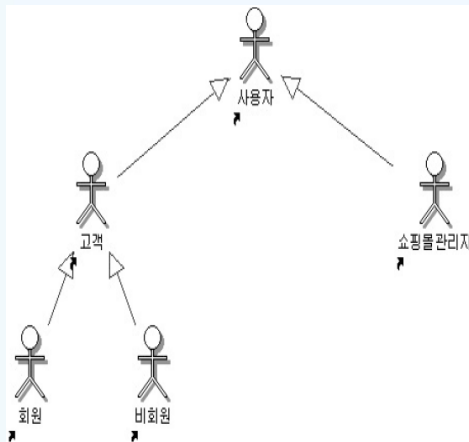


[그림3-6] 도서예약 유스케이스의 확장

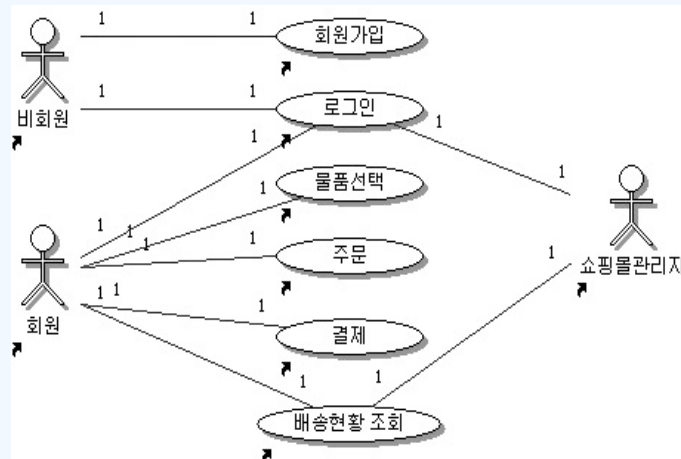
[표 3-3] 포함관계와 확장관계의 비교

	포함관계 <<include>>	확장관계 <<extend>>
목 적	<ul style="list-style-type: none"> 여러 유스케이스에 <u>공통적인</u> 기능을 표현하기 위해 사용된다. 	<ul style="list-style-type: none"> 기준 유스케이스에 부가적으로 <u>추가된</u> 기능을 표현하기 위해 사용된다.
이벤트 흐름	<ul style="list-style-type: none"> 포함된 유스케이스로의 이벤트 흐름 분기가 필수적이다. 기준 유스케이스 이후의 이벤트 흐름이 포함된 유스케이스의 수행 결과에 의존한다. 	<ul style="list-style-type: none"> 확장 유스케이스는 확장 유스케이스에 기술된 조건에 따라 선택적으로 수행된다. 기준 유스케이스 이후의 이벤트 흐름이 확장 유스케이스의 결과에 의존하지 않는다.

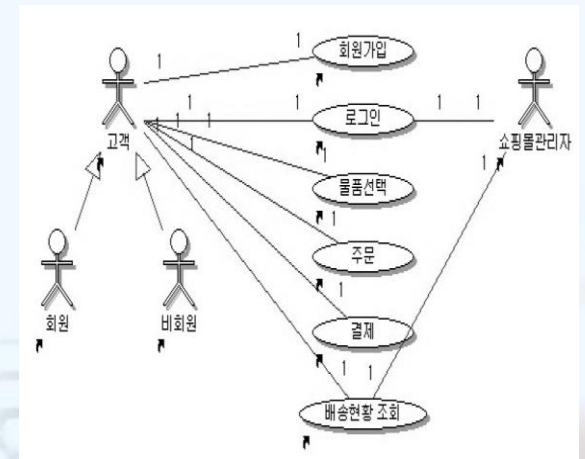
- 액터 사이의 일반화(Generalization)는 클래스 사이의 일반화와 비슷한 개념
- 추상적인 액터와 좀 더 구체적인 액터 사이에 맺어주는 관계로서
 - ✓ “한 액터가 다른 액터의 일종이다” 또는
 - ✓ “한 액터도 다른 액터에 해당된다”의 의미가 되는 두 액터를 일반화관계로 연결
- 일반화관계를 액터에 적용하면 유스케이스 다이어그램에서 사용되는 여러 액터들의 의미를 좀 더 명확하게 하고 다이어그램도 보다 간결하게 작성



[그림3-9] 액터의 일반화관계



[그림3-10] 일반화를 사용하지 않은 경우



[그림3-11] 일반화관계를 적용한 경우

● 유스케이스 다이어그램을 만드는 단계

- 1단계 : 시스템 상황을 확인
- 2단계 : 액터 식별
 - ✓ 행위자와 그들의 책임을 확인
- 3단계 : 유스케이스 식별
 - ✓ 특정한 목적의 관점에서 볼 때 쓰임새와 시스템의 특성을 확인
- 4단계 : 유스케이스 다이어그램 작성
 - ✓ 유스케이스에서 <<include>> 의존성이 있는지 평가(공통 부분)
 - ✓ 유스케이스에서 <<extend>> 의존성이 있는지 평가(확장부분)

▶ [예제] 재고관리 시스템

■ 1단계 : 시스템 상황 분석(문제 기술서 작성)

- ✓ 한 인터넷 쇼핑몰에서 원활한 창고의 재고관리를 위해 재고관리 시스템 구축
- ✓ 재고관리 시스템은 크게 입고관리 기능, 출고관리 기능, 현황관리 기능 제공

① 입고관리 기능 :

- 창고로 입고된 상품을 현황관리에 추가
- 입고는 업체로부터 새로운 상품을 입고 받거나 고객의 반품에 의한 것
- 입출고 담당자는 입고된 제품 상태를 파악하고 불량조치하여 업체에 반품

② 출고관리 기능 :

- 창고로 출고된 상품을 현황관리에서 뺀다.
- 출고는 고객이 구매한 상품을 발주하는 것과 판매하고 남은 상품을 통신 업체로 반품한 것

③ 현황관리 기능 :

- 입고와 출고된 현황을 실시간으로 인터넷 쇼핑몰에 업데이트하는 기능
- 입출고 담당자는 현황을 조회할 수 있고 현황관리 담당자는 재고현황을 관리하여 협력업체에 주문 혹은 반품을 요청하고 쇼핑몰에 업데이트

■ 2단계 : 액터 식별

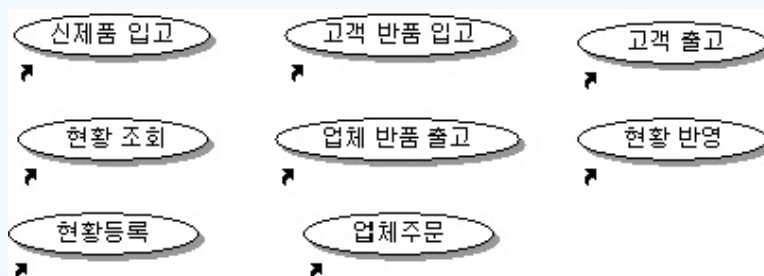


[그림3-20] 재고관리 시스템의 액터

■ 3단계 : 유스케이스 식별

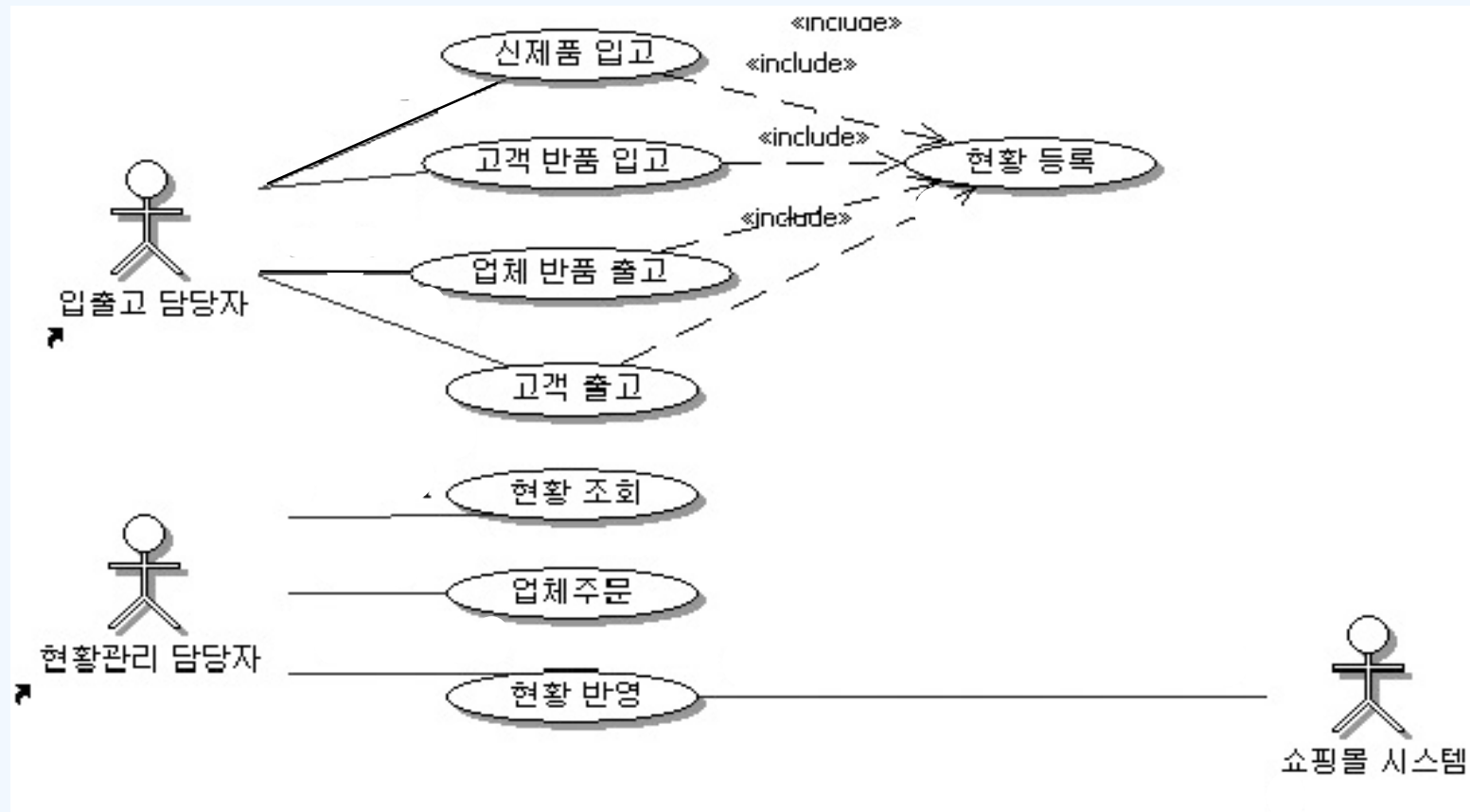
기능 범주	사용자	기능(유스케이스)
입고관리	입출고 담당자	통신업체 제품 입고 기능 고객 반품 입고 기능
출고관리	입출고 담당자	통신업체 반품 출고 기능 고객 출고 기능 발주 기능
현황관리	입출고 담당자	현황 조회
	현황관리 담당자	현재 현황 등록 기능(업데이트) 통신업체 주문 기능
	쇼핑몰 시스템	실시간 현황 반영 기능

[표 3-7] 재고관리 시스템의 사용자별 기능



[그림3-21] 재고관리 시스템의 유스케이스

■ 4단계 : 유스케이스 다이어그램 작성



[그림3-22] 재고관리 시스템의 유스케이스