

유스케이스 다이어그램

* 학습목표

- 유스케이스의 개념을 이해한다.
- 유스케이스의 관계를 학습한다.
- 유스케이스 모델링을 학습한다.
- 유스케이스 다이어그램을 실습한다.

01. 유스케이스 모델링과 UML 표기법

02. 유스케이스와 유스케이스 관계

03. 액터 사이의 일반화관계

04. 액터 사이의 중복관계

05. 유스케이스 모델링 절차

06. 재고관리 시스템 유스케이스 모델링

요약

연습문제



집을 설계할 때는 집의 용도를 정확히 파악해야 하듯이 소프트웨어 개발에서도 요구사항을 정확히 파악했는지에 그 성공 여부가 달려 있다. 프로젝트가 실패하는 주된 원인을 찾아보면 사용자 요구사항을 정확히 파악하지 못한 것이 대부분이다. 설령 요구사항의 중요성과 문제점을 인지한다 하더라도 다양한 사용자 요구사항을 주어진 기간 내에 완벽하게 도출하기는 사실상 어렵다. UML에서는 이러한 요구사항을 더 쉽고, 효과적으로 도출하기 위하여 유스케이스 다이어그램을 사용한다.

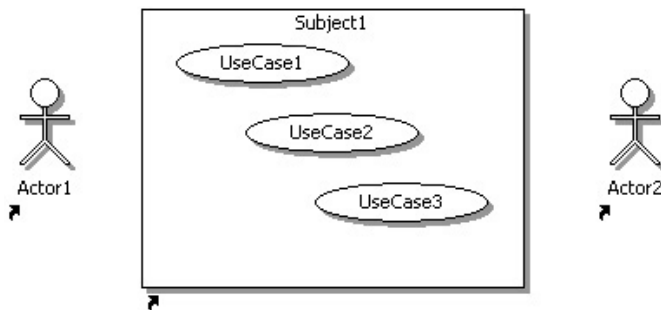
유스케이스 다이어그램은 스웨덴의 이바 야콥슨에 의해 처음 개발된 다이어그램으로, 유스케이스 다이어그램은 사용자 요구사항들 가운데 주로 기능 요구사항을 유스케이스라는 단위로 표현한다.



유스케이스 모델링과 UML 표기법

시스템 개발에 참여하는 사람들을 큰 부류로 나누면, 의뢰인과 개발팀으로 나눌 수 있다. 하지만 또 하나의 중요한 부류가 바로 사용자(User)이다. 사용자의 관점을 빨리 이해해야만 쓸모 있고 유용한 시스템을 만들 수 있다. 그러므로 개발자는 가능한 한 모든 요구사항을 파악하여 사용자의 승인을 받아야만 후일에 요구사항 변경에 대한 위험부담을 줄일 수 있다. 이를 위해 요구사항 정의 활동이 개발과 설계에 있어서 커다란 비중을 차지하고 있다는 것을 알 수 있다.

예를 들어 [그림 3-1]에서 네모난 창은 시스템의 경계를 나타낸다. 따라서 이 경계 내부에 위치한 UseCase1, 2, 3은 구축할 시스템의 기능을 표현한 것이다. 반면에 시스템 경계의 외부에 위치한 Actor1, 2는 구축할 시스템의 범위 밖이다. 즉, 개발 대상에 해당하지 않는다.



[그림 3-1] 시스템의 경계

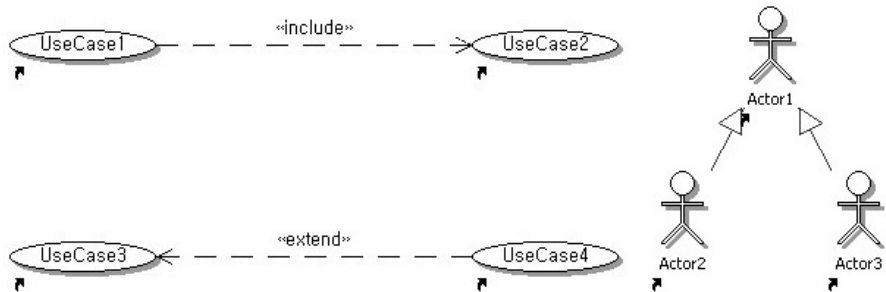
유스케이스 모델링 기법에서 개발할 시스템 외부의 존재를 액터(Actor)라는 개념으로 정의하고, 시스템의 범위에 해당되어 개발될 시스템의 단위 기능을 유스케이스라는 개념으로 정의한다.

사용자의 시점에서 시스템을 모델링한다는 것, 이것이 바로 유스케이스의 역할이다. 유스케이스는 시스템에 대한 시나리오의 집합으로 볼 수 있다. 각 시나리오에는 발생하는 이벤트의 흐름이 나타나 있다. 각 이벤트의 흐름은 사람, 시스템, 하드웨어, 혹은 시간의 흐름에 의해 시작된다. 이때 이벤트 흐름을 시작하게 하는 객체가 액터이다. 이 흐름은 해당 액터 혹은 다른 액터에게 어떤 의미를 가지는 결과를 제공한다. 유스케이스와 액터는 [그림 3-2]와 같은 UML 표기법을 갖는다.



[그림 3-2] 유스케이스와 액터

유스케이스 사이에는 일반적인 연관관계 이외에 또 다른 관계를 가질 수 있다. 첫째는 포함(Include)관계로서 다른 유스케이스에서 기존의 유스케이스를 재사용할 수 있는 관계를 나타낸다. 둘째는 확장(Extend)관계로서 기존의 유스케이스에 진행단계를 추가하여 새로운 유스케이스를 만들어내는 관계이다. 또한 액터들 간의 일반화관계가 있을 수 있다. [그림 3-3]은 유스케이스 다이어그램에서 사용되는 관계를 그림으로 나타낸 것이다.



[그림 3-3] 유스케이스 다이어그램에서 사용되는 관계들



유스케이스와 유스케이스 관계

유스케이스 다이어그램에서 유스케이스간의 관계는 의존(Dependency, --->)관계로 표현되며, 스테레오타입으로 그 관계를 명확하게 표현한다. 스테레오타입은 포함과 확장을 의미하는 <<include>>와 <<extend>>로 표현된다.

① 유스케이스 간 포함관계

포함관계는 하나의 유스케이스를 수행할 때, 같은 기능을 가진 다른 유스케이스가 반드시 수행되는 것을 말한다.

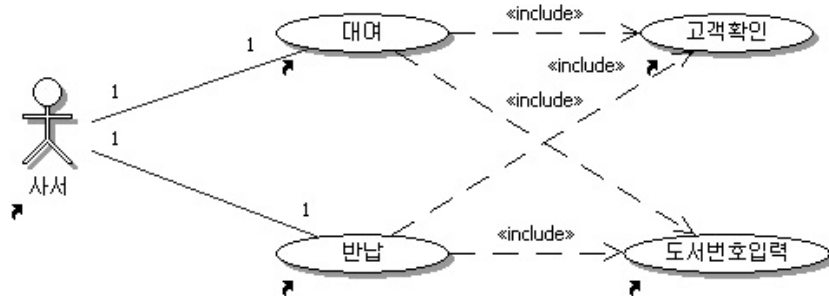
유스케이스 다이어그램에서는 다른 유스케이스가 나타내는 이벤트 흐름을 포함(include)하는 관계를 유스케이스간에 표현할 수 있다. 새롭게 만들어진 ‘고객확인’ 유스케이스와 ‘도서번호입력’ 유스케이스를 이용하여 ‘대여’, ‘반납’ 유스케이스와의 관계를 포함관계로 설정한 것이 [그림 3-4]에 나타나 있다. 이 그림은 ‘대여’ 유스케이스와 ‘반납’ 유스케이스 각각이 ‘고객확인’ 유스케이스와 ‘도서번호입력’ 유스케이스를 포함함을 의미한다.

여기서 잠깐



액터와 유스케이스와의 관계를 설정하기 위해서는 해당 액터와 정보를 주고받는 유스케이스를 찾아서 연관(Association)관계를 설정한다. 액터와 유스케이스를 연결하는 실선을 연관관계라고 하는데, 이는 액터와 유스케이스 사이의 교류가 이루어지고 있다는 것을 의미한다.

**액터와 유스케이스
관계**



[그림 3-4] 도서관 시스템의 포함관계

[표 3-1]은 '대여' 유스케이스와 '반납' 유스케이스가 '고객확인' 유스케이스와 '도서번호 입력' 유스케이스를 이용할 때의 이벤트 흐름을 보여준다. [표 3-5]에서는 '대여' 유스케이스와 '반납' 유스케이스에서 중복된 부분이 '고객확인' 유스케이스와 '도서번호입력' 유스케이스를 이용하도록 수정되었다.

이와 같이 여러 유스케이스에 나타난 공통적인 이벤트 흐름을 별도의 유스케이스로 표현함으로써 이벤트 흐름의 중복을 방지할 뿐만 아니라 시스템의 공통 기능을 추출하여 효율적으로 개발하는 것이 가능하게 된다.

[표 3-1] 포함관계를 이용한 '대여'와 '반납' 유스케이스의 이벤트 흐름

대여 유스케이스 이벤트 흐름	반납 유스케이스 이벤트 흐름
1. '고객확인' 유스케이스를 포함한다. 2. 사서는 '대여'를 선택한다. 3. '도서번호입력' 유스케이스를 포함한다. 4. 도서관 시스템은 고객이 대여가 가능한지 확인한다. 5. 고객에게 대여를 가능 여부를 표시한다. 6. 도서관 시스템은 도서를 대여 처리한다.	1. '고객확인' 유스케이스를 포함한다. 2. 사서는 '반납'을 선택한다. 3. '도서번호입력' 유스케이스를 포함한다. 4. 도서관 시스템은 도서를 반납 처리한다.

② 유스케이스 간 확장관계

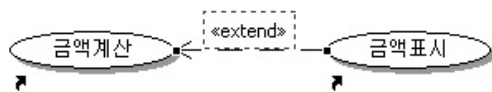
확장관계의 유스케이스는 포함관계처럼 여러 유스케이스에 걸쳐 중복적으로 사용되지 않고, 특정 조건에서 한 유스케이스로만 확장되는 것을 의미한다. 따라서 확장(extend)하는 유스케이스는 상위 유스케이스로부터 어떠한 특정 조건에 의해 수행됨을 의미한다.

포함관계가 여러 유스케이스에서 공통적으로 발견되는 시스템의 기능을 표현하기 위하여 사용되는 반면에 확장관계는 한 유스케이스에서 추가되거나 확장된 기능을 표현하기 위하여 사용된다.

예를 들어 자동판매기 시스템에서 사용자는 동전을 투입하고 음료수를 선택하여 원하는 음료수를 마실 수 있다. 여기에서 동전을 투입할 때 자동판매기 시스템은 현재의 금액을 표시한다. 또한 음료수를 선택하면 현재 금액에서 뺀 나머지를 표시한다.

즉 자동판매기 시스템이 현재의 금액을 계산하면 사용자에게 보이는 금액표시기가 현재 금액을 나타낸다. 금액표시는 금액계산 유스케이스에 추가된 부가적인 기능이라고 볼 수 있다. 그리고 금액표시 유스케이스의 기능과 성격이 다르므로 금액계산 유스케이스의 이벤트 흐름에 포함시켜서 기술하는 것보다 별도의 유스케이스로 표현하는 것이 바람직하다.

[그림 3-5]는 금액계산 기능을 ‘금액계산’ 유스케이스로 표현하고, ‘금액표시’ 유스케이스로 확장시킨 그림이다. 이때 ‘금액계산’ 유스케이스를 기준(Based) 유스케이스라 하고, ‘금액표시’ 유스케이스를 확장(Extending) 유스케이스라 부른다. [표 3-2]는 ‘금액계산’ 유스케이스와 ‘금액표시’ 유스케이스의 이벤트 흐름을 보여준다.



[그림 3-5] 금액계산 유스케이스의 확장

[표 3-2] 금액계산 유스케이스와 금액표시 유스케이스의 이벤트 흐름

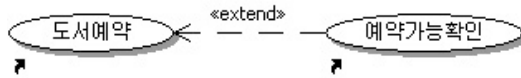
금액계산 유스케이스 이벤트 흐름	금액표시 유스케이스 이벤트 흐름
1. 사용자가 자판기에 동전을 투입하거나 음료수를 선택한다. 2. 현재 금액에 투입된 동전만큼 액수를 추가한다. 3. 금액표시 유스케이스 확장	1. 금액계산 유스케이스로부터 금액을 받는다. 2. 받은 금액을 표시한다.

확장 유스케이스는 일반적으로 특정 조건이 만족되는 상황에서만 확장 유스케이스의 이벤트 흐름이 수행된다. ‘금액계산’ 유스케이스의 경우에는 사용자의 동전 투입이나 음료수 선택 등의 행위에 의해 금액을 계산한 후에 ‘금액표시’ 유스케이스의 이벤트 흐름이 수행된다.

확장 유스케이스의 이벤트 흐름의 수행 여부를 결정짓는 조건은 확장 유스케이스를 포함하

는 기준 유스케이스가 아니라 확장 유스케이스에 표현되는 점을 주목한다. 즉 [표 3-2]에서 보면, 확장 유스케이스인 ‘금액표시’ 유스케이스의 이벤트 흐름 1번 항목에 수행 여부 조건 (금액이 계산되었을 경우)이 나타나 있다.

[그림 3-6]은 확장관계의 또 다른 예로서, 도서예약 시스템에서 예약을 할 때 예약이 가능한지를 확인하고 예약 기능을 수행한다. 즉 도서예약 기능을 기본적으로 수행하는 도중에 예약이 가능한지를 추가적으로 수행한다.

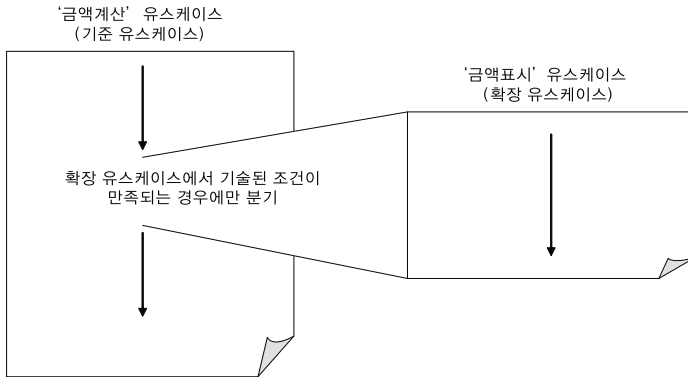


[그림 3-6] 도서예약 유스케이스의 확장

③ 확장 유스케이스와 포함 유스케이스

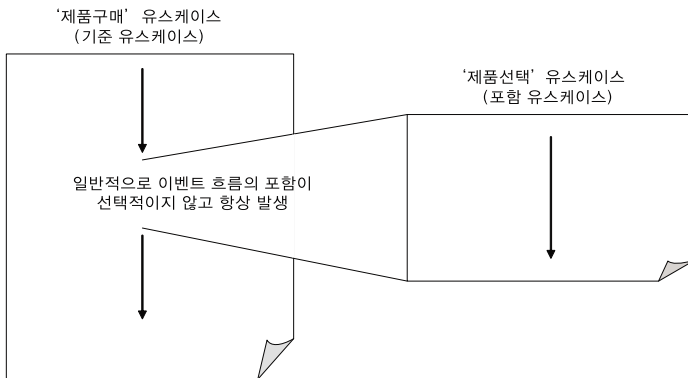
확장관계에 있는 유스케이스 사이의 이벤트 흐름은 포함관계에 있는 유스케이스 사이의 이벤트 흐름과 유사하다. 즉 기준 유스케이스의 이벤트 흐름이 수행되었다가 확장점을 만나면 지정된 유스케이스의 이벤트 흐름으로 분기된다.

[그림 3-7]은 기준 유스케이스와 확장 유스케이스 사이의 이벤트 흐름 수행관계를 보여준다. 기준 유스케이스인 ‘금액계산’ 유스케이스에 기술된 이벤트 흐름이 차례로 수행되다가 확장점에서 확장 유스케이스인 ‘금액표시’ 유스케이스로 이벤트 흐름이 분기된다. 확장 유스케이스에 기술된 이벤트 흐름의 수행이 완료되면 다시 기준 유스케이스로 되돌아와서 이후의 이벤트 흐름을 수행한다.



[그림 3-7] 확장관계의 이벤트 흐름

[그림 3-8]은 포함관계가 있는 두 유스케이스 사이의 이벤트 흐름을 보여준다. 기준 유스케이스인 '제품구매' 유스케이스에 기술된 이벤트 흐름이 차례로 수행되다가 특정 지점에서 포함된 유스케이스('제품선택' 유스케이스)로 분기가 발생한다. '제품선택' 유스케이스의 이벤트 흐름이 모두 수행되면 다시 '제품구매' 유스케이스의 이벤트 흐름으로 돌아와 이후의 이벤트를 수행한다.



[그림 3-8] 포함관계의 이벤트 흐름

확장 유스케이스와 포함된 유스케이스의 이벤트 흐름이 수행된 후에 다시 원래 유스케이스로 돌아와서 이후의 이벤트 흐름이 수행되는 것은 확장관계와 포함관계 모두 동일하다. 그러나 포함관계의 유스케이스는 이벤트 흐름이 피포함 유스케이스로 항상 분기되는 반면에

확장 유스케이스에서는 확장 유스케이스에 기술된 조건이 만족되는 경우에만 확장 유스케이스의 이벤트 흐름으로 분기가 발생한다.

예를 들어 ‘제품구매’ 유스케이스에서는 사용자가 제품을 선택해야 하기 때문에 항상 ‘제품선택’ 유스케이스로 분기가 되어야 한다. 그러나 ‘금액계산’ 유스케이스의 경우에는 동전이 투입되거나 음료를 선택하여 금액계산이 일어날 때만 ‘금액표시’ 유스케이스로 분기가 발생한다.

즉 포함된 유스케이스로의 분기는 필수적인 반면에 확장 유스케이스로의 분기는 선택적이다. 포함 또는 확장 유스케이스로부터 다시 돌아와 이후의 이벤트를 수행할 때 필수와 선택의 차이점이 드러난다. 포함된 유스케이스는 필수적으로 분기가 발생하며, 대부분의 경우에 포함된 유스케이스의 수행 결과에 따라서 기존 유스케이스의 이벤트 흐름이 영향을 받는다. 즉 ‘제품선택’ 유스케이스에서 제품을 선택하였는지 하지 않았는지에 따라 ‘제품구매’ 유스케이스 이후의 이벤트 흐름에 영향을 주게 된다.

반면 확장 유스케이스는 선택적으로 이벤트 흐름이 수행된다. 따라서 기존 유스케이스 이후의 이벤트 흐름은 확장 유스케이스의 수행 결과에 의존하지 않는다. 비록 ‘금액계산’ 유스케이스에서 ‘금액표시’ 유스케이스로의 분기 이후에 다른 이벤트 문장이 없지만 만약 있다 하더라도 이 이벤트 문장은 ‘금액표시’ 유스케이스의 수행 결과에 영향을 받지 않는다. [표 3-3]은 유스케이스간의 포함관계와 확장관계를 주요 목적과 이벤트 흐름 측면에서 비교한 것이다.

[표 3-3] 포함관계와 확장관계의 비교

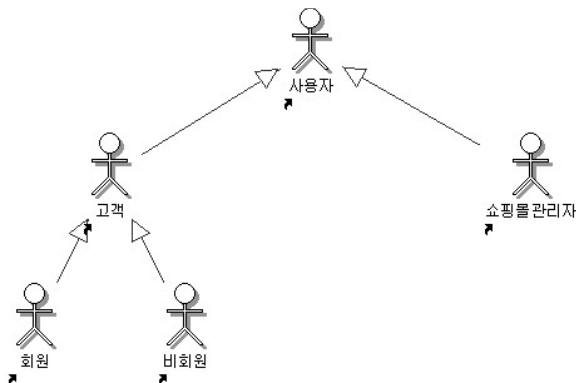
	포함관계	확장관계
목 적	• 여러 유스케이스에 공통적인 기능을 표현하기 위해 사용된다.	• 기존 유스케이스에 부가적으로 추가된 기능을 표현하기 위해 사용된다.
이벤트 흐름	• 포함된 유스케이스로의 이벤트 흐름 분기가 필수적이다. • 기존 유스케이스 이후의 이벤트 흐름이 포함된 유스케이스의 수행 결과에 의존한다.	• 확장 유스케이스에 기술된 조건에 따라 선택적으로 수행된다. • 기존 유스케이스 이후의 이벤트 흐름이 확장 유스케이스의 결과에 의존하지 않는다.



액터 사이의 일반화관계

액터 사이의 일반화(Generalization)는 클래스 사이의 일반화와 비슷한 개념이다. 추상적인 액터와 좀 더 구체적인 액터 사이에 맺어주는 관계로서 “한 액터가 다른 액터의 일종이다” 또는 “한 액터도 다른 액터에 해당된다”라는 의미가 만족되는 두 액터를 일반화관계로 연결한다.

예를 들어 쇼핑몰 시스템에는 ‘고객’ 액터와 ‘쇼핑몰관리자’ 액터가 있다. ‘고객’ 액터는 ‘회원’과 ‘비회원’이 될 수 있다. 그리고 ‘고객’ 액터와 ‘쇼핑몰관리자’ 액터는 모두 쇼핑몰 시스템의 ‘사용자’라고 볼 수 있다. 따라서 이들 액터 사이의 일반화관계를 [그림 3-9]와 같이 나타낼 수 있다.

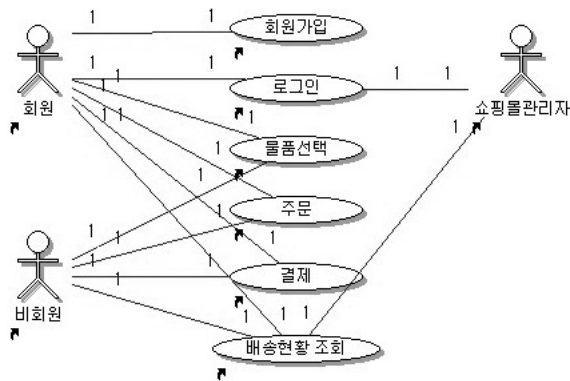


[그림 3-9] 쇼핑몰 액터들 사이의 일반화관계

이러한 일반화관계를 액터에 적용하면 유스케이스 다이어그램에서 사용되는 여러 액터들의 의미를 좀 더 명확하게 하고 다이어그램도 보다 간결하게 작성할 수 있다.

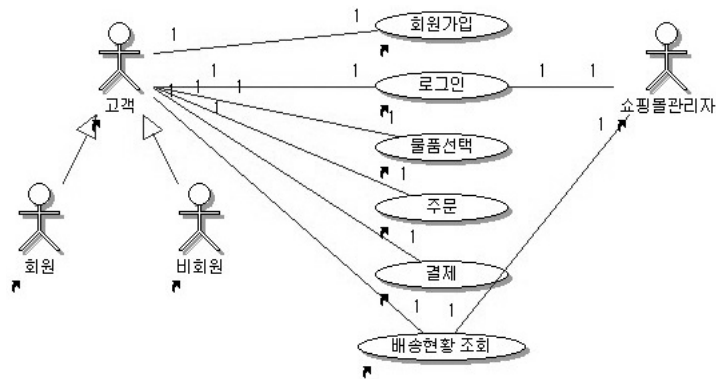
[그림 3-10]의 유스케이스 다이어그램은 ‘고객’ 액터를 이용하지 않고, ‘회원’, ‘비회원’, ‘쇼핑몰관리자’ 액터만을 이용해서 나타낸 것이다. 쇼핑몰 시스템에서 ‘회원’, ‘비회원’,

‘쇼핑몰관리자’ 액터는 시스템의 기능 중 ‘로그인’ 기능을 이용할 수 있다. ‘회원’, ‘비회원’, ‘쇼핑몰관리자’ 액터에서 ‘로그인’ 유스케이스로 연관을 각각 만들어야 한다



[그림 3-10] 일반화를 사용하지 않은 경우

이렇게 작성된 유스케이스는 매우 복잡하게 느껴진다. 다이어그램이 불필요하게 복잡하면 개발자들이 쉽고 정확하게 이해하기가 어려워지므로 바람직하지 않다. 이를 해결하기 위해 일반화관계를 이용하면 다이어그램을 간결하게 표시할 수 있다. [그림 3-11]은 [그림 3-10]을 액터 사이의 일반화관계를 이용하여 표현한 유스케이스 다이어그램이다.



[그림 3-11] 액터 사이의 일반화관계를 적용한 경우

일반화관계의 정의에 의하면 하위 액터는 상위 액터의 일종이다. 따라서 명시적으로 표현되지는 않았지만 상위 액터와 연관된 유스케이스는 하위 액터와도 연관을 맺는다. 예를 들어 [그림 3-11]의 유스케이스 다이어그램에서 ‘고객’ 액터는 ‘로그인’ 유스케이스와 연관을 맺고 있다. 그리고 ‘회원’과 ‘비회원’ 액터는 ‘고객’ 액터와 일반화 관계를 맺고 있다. 따라서 ‘회원’과 ‘비회원’ 액터도 ‘로그인’ 유스케이스와 연관 관계를 맺게 된다. 결과적으로 ‘고객’ 액터의 하위 액터인 ‘회원’, ‘비회원’은 ‘로그인’ 유스케이스와 묵시적으로 연관 관계를 맺고 있는 것이다.

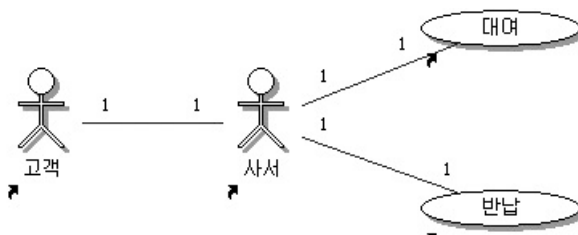
[그림 3-11]의 유스케이스 다이어그램은 [그림 3-10]의 유스케이스 다이어그램과 동일한 정보를 표현하면서도 보다 간결하고 명확하다. [그림 3-11]의 다이어그램이 간결해진 이유는 [그림 3-10]의 유스케이스 다이어그램에 비해 적은 수의 연관이 표시되었기 때문이다.

[그림 3-10]의 다이어그램에서 표현된 ‘회원’, ‘비회원’, ‘쇼핑물관리자’에서 ‘로그인’ 유스케이스로의 연관이 [그림 3-11]의 다이어그램에서는 ‘고객’ 액터에서 ‘로그인’ 유스케이스로의 연관 하나만으로 표현되었다. 이와 같은 방법으로 ‘물품선택’ 유스케이스와 ‘주문’ 유스케이스로의 연관이 각각 하나만으로 표현이 가능하다.



액터 사이의 중복관계

유스케이스 모델링 과정에서 2개 이상의 유스케이스 이벤트 흐름에서 중복적인 부분이 발생할 수 있다. 이러한 경우에 유스케이스간 포함관계를 설정하여 해결한다. 예를 들어 도서관 시스템에 대해 생각해보자. 이에 대한 간략한 유스케이스 다이어그램은 [그림 3-13]과 같다.



[그림 3-13] 도서관 시스템의 간략한 유스케이스 다이어그램

도서관 시스템의 대여 유스케이스와 반납 유스케이스의 이벤트 흐름을 살펴보자.

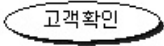
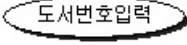
[표 3-4] 대여 및 반납 유스케이스의 이벤트 흐름

대여 유스케이스 이벤트 흐름	반납 유스케이스 이벤트 흐름
1. 사서는 고객의 아이디와 패스워드를 도서관 시스템에 입력한다.	
2. 도서관 시스템은 아이디와 패스워드로 고객의 정보를 확인하고 업무 종류를 선택할 수 있는 화면을 보여준다.	
3. 사서는 '대여'를 선택한다.	3. 사서는 '반납'을 선택한다.
4. 도서관 시스템은 사서에게 도서 번호를 입력하기 위한 화면을 보여준다.	
5. 사서는 도서 번호를 입력한다.	
6. 도서관 시스템은 도서 목록에서 도서 번호가 유효한지 확인한다.	
7. 도서관 시스템은 고객이 대여가 가능한지 확인한다.	7. 도서관 시스템은 도서를 반납 처리한다.
8. 고객에 대한 대여의 가능 여부를 표시한다.	
9. 도서관 시스템은 도서를 대여 처리한다.	

[표 3-4]를 살펴보면 대여 유스케이스와 반납 유스케이스에서 상당한 부분이 중복되는 것을 볼 수 있다. 즉 1번, 2번, 4번, 5번, 6번 이벤트의 부분이 동일한 이벤트 문장이다. 이와 같이 유사한 기능을 가지는 여러 유스케이스에서 동일한 이벤트 문장이 중복적으로 발생하면, 중복되는 불필요한 부분을 중복적으로 개발하게 되기 때문에 비효율성을 초래한다. 그러므로 공통적인 이벤트 문장을 추출해서 별도의 유스케이스로 한꺼번에 표현하는 것이 바람직하다.

위에서 비슷한 이벤트인 1번과 2번 문장을 표현하는 유스케이스를 따로 만들고, 마찬가지로 비슷한 이벤트로 묶여 있는 4번, 5번, 6번의 문장을 표현하는 유스케이스를 따로 만들어 새롭게 2개의 유스케이스를 생성하면 [표 3-5]와 같다.

[표 3-5] '고객확인' 유스케이스와 '도서번호입력' 유스케이스

	1. 사서는 고객의 아이디와 패스워드를 도서관 시스템에 입력한다. 2. 도서관 시스템은 아이디와 패스워드로 고객의 정보를 확인하고 업무 종류를 선택할 수 있는 화면을 보여준다.
	1. 도서관 시스템은 사서에게 도서의 번호를 입력하기 위한 화면을 보여준다. 2. 사서는 도서 번호를 입력한다. 3. 도서관 시스템은 도서 목록에서 도서 번호가 유효한지 확인한다.



유스케이스 모델링 절차

① 유스케이스 다이어그램을 만드는 단계

1단계 : 시스템 상황을 확인한다.

2단계 : 액터 식별

- 행위자와 그들의 책임을 확인한다.

3단계 : 유스케이스 식별

- 특정한 목적의 관점에서 볼 때 쓰임새와 시스템의 특성을 확인한다.

4단계 : 유스케이스 다이어그램 작성

- 행위자와 유스케이스에서 정제할 부분이 있는지 평가한다.
- 유스케이스에서 <<include>> 의존성이 있는지 평가한다.
- 유스케이스에서 <<extend>> 의존성이 있는지 평가한다.
- 행위자와 유스케이스를 일반화(또는 공유)할 수 있는지 평가한다.

5단계 : 유스케이스 명세서 작성

- 유스케이스명, 액터명 및 개요를 기술한다.
- 사전 및 사후 조건과 제약사항들을 식별한다.
- 작업(정상, 대치, 예외)흐름과 시나리오를 도출한다.
- 유스케이스 흐름에서 포함이나 확장 유스케이스로 구조화한다.

6단계 : 유스케이스 실체화

- 구현 시스템의 논리적 구성 요소인 클래스를 식별하고 통신관계를 파악하는 데 중점을 두는 과정이다.

위의 모델링 단계를 예를 들어 설명하면 이해가 빠를 것이다.

예제 3-1

비디오숍 관리 시스템 문제

1단계 : 시스템 상황 분석(문제 기술서 작성)

비디오숍 관리 시스템에서 고객은 반드시 회원가입을 해야만 비디오를 대여할 수 있다. 회원은 가입된 이름과 전화번호를 통해 확인한다. 비디오숍 관리 시스템에는 대여, 반납, 연체 확인 기능이 있다.

대여는 고객이 비디오를 선택하면 비디오 코드를 확인하여 시스템에 입력한다. 현재 고객이 대여 중인 비디오가 있으면 표시하고 대여 기간이 지났으면 연체료를 계산하여 보여준다. 연체 중인 고객은 비디오 대여를 할 수 없다. 그러나 연체료를 납부하면 대여할 수 있다. 대여료와 연체료는 현금이나 신용카드를 통해 이루어진다. 대여된 비디오는 대여목록에 비디오 코드와 고객명이 등록된다.

관리자는 연체관리 기능을 통해 현재 연체 중인 회원과 연체된 비디오를 확인한다. 연체금 표시 기능을 통해 오늘 날짜에 해당하는 연체금을 표시한다. 반납 기능은 반납한 비디오의 코드를 입력하여 대여목록에서 삭제한다.

2단계 : 액터 식별

액터는 외부에서 시스템에 접근할 수 있는 사람이나 시스템과 관련된 외부 시스템을 의미한다. 그래서 액터의 명칭은 특정 사람의 이름이라기보다 역할을 의미하는 명칭을 사용한다. 예를 들어 John, Sandy 등과 같은 지정된 명칭은 사용하지 않으며, 고객, 관리자, 시스템과 같은 추상적 명칭을 사용해야 한다. 액터는 요구 명세서에서 시스템에 접근하거나 시스템으로부터 정보를 받는 외부 시스템에 해당하는 사람 또는 사물로 접근할 수 있다.

| 따라하기 |

‘[예제 3-1] 비디오숍 관리 시스템’에서 액터를 추출하면 ‘고객’, ‘관리자’, ‘신용카드 승인시스템’ 등이 액터로 정의될 수 있다. 특히 ‘신용카드 승인시스템’은 비디오숍 관리 시스템에서 개발될 시스템이 아니라 신용카드회사에서 개발되는 시스템이기 때문에 외부 시스템으로 정의할 수 있다. [그림 3-14]는 식별된 액터를 UML로 표현한 것이다.



[그림 3-14] 액터 식별 예

3단계 : 유스케이스 식별

유스케이스는 개발을 위한 시스템의 기능을 의미한다. 즉 시스템을 수행하는 일련의 행위들이다. 이는 시스템에서 제공해야 하는 독립적인 기능을 의미한다. 또한 외부 시스템과 상호작용하는 행위들의 기능만을 의미한다. 따라서 유스케이스는 행위 자체만을 표현할 뿐이며, 행위 과정은 기술할 필요가 없다. 행동에 대한 구체적인 수행 방법은 규정하지 않는다.

유스케이스를 식별하기 위해서는 먼저 개발할 시스템의 기능 요구사항 명세서를 보고, 액터를 중심으로 시스템에 대해 원하는 기능들이 무엇인지 찾아서 이를 유스케이스로 추출하여야 한다.

| 따라하기 |

‘[예제 3-1] 비디오숍 관리 시스템’의 경우, 비디오숍 관리 시스템에서 제공하여야 할 주요 기능인 ‘회원가입’, ‘회원확인’, ‘대여’, ‘반납’, ‘연체관리’, ‘결제’, ‘신용카드지불’, ‘등록’, ‘삭제’ 등이 유스케이스로 추출될 수 있다.

여기에서 ‘신용카드지불’ 유스케이스의 경우에도 신용카드 승인시스템에서 처리하는 일련의 기능을 제공하기 위해 별도의 유스케이스로 도출하였다. 따라서 [그림 3-15]는 식별된 9개의 유스케이스를 UML로 표기한 것이다.



[그림 3-15] 유스케이스 식별 예

4단계 : 유스케이스 다이어그램 작성

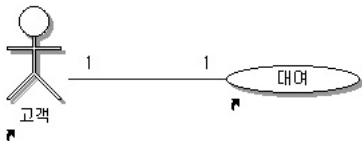
유스케이스 다이어그램은 액터와 유스케이스 사이, 또는 유스케이스와 유스케이스 사이의 관계를 설정함으로써 표현할 수 있다.

■ 액터와 유스케이스 관계

액터와 유스케이스와의 관계를 설정하기 위해서는 해당 액터와 정보를 주고받는 유스케이스를 찾아서 연관관계를 설정한다.

| 따라하기 |

[예제 3-1]의 경우, ‘고객’이 ‘대여’ 유스케이스를 수행한다는 관계를 [그림 3-16]과 같이 표시할 수 있다.



[그림 3-16] 액터와 유스케이스 사이의 관계 설정

■ 유스케이스와 유스케이스 관계

유스케이스 다이어그램에서 유스케이스 사이의 관계는 의존관계로 표현되며, 스테레오타입으로 그 관계를 명확하게 표현한다. 스테레오타입은 포함과 확장을 의미하는 <<include>>와 <<extend>>로 표현된다.

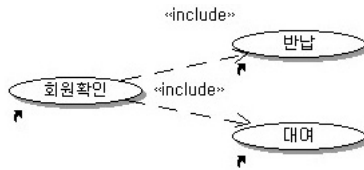
<<include>> 관계는 하나의 유스케이스를 수행할 때 같은 기능을 가진 또 하나의 유스케이스를 반복적으로 반드시 수행해야 할 경우를 말한다.

<<extend>> 관계의 유스케이스는 <<include>> 관계처럼 여러 유스케이스에 걸쳐 중복적으로 사용되지 않고, 특정 조건에서 한 유스케이스로만 확장되는 것을 의미한다. 따라서 확장하는 유스케이스는 상위 유스케이스로부터 어떠한 특정 조건에 의해 수행됨을 의미한다.

| 따라하기 |

<<include>> 관계

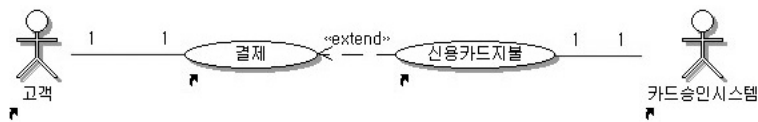
[예제 3-1]의 경우, ‘대여’와 ‘반납’ 유스케이스가 수행될 때는 반드시 ‘회원확인’ 유스케이스가 선행(포함)되어야 한다. 그러므로 ‘회원확인’ 유스케이스를 ‘대여’, ‘반납’ 유스케이스로 <<include>>하는 관계로 표시한다. [그림 3-17]은 <<include>> 관계를 나타낸 것이다.



[그림 3-17] <<include>> 관계

<<extend>> 관계

‘고객’이 ‘결제’ 유스케이스를 수행할 때, 신용카드로 결제하는 경우에는 ‘신용카드지불’ 유스케이스가 수행되기 때문에 ‘결제’ 유스케이스로부터 ‘신용카드지불’ 유스케이스가 확장되는 형태로 이루어져야 한다. 또한 신용카드지불 시에는 카드승인사에 카드승인을 요청해야 하므로, ‘카드승인 시스템’ 유스케이스와의 관계도 설정하여야 한다. [그림 3-18]은 <<extend>> 관계를 나타낸 것이다.



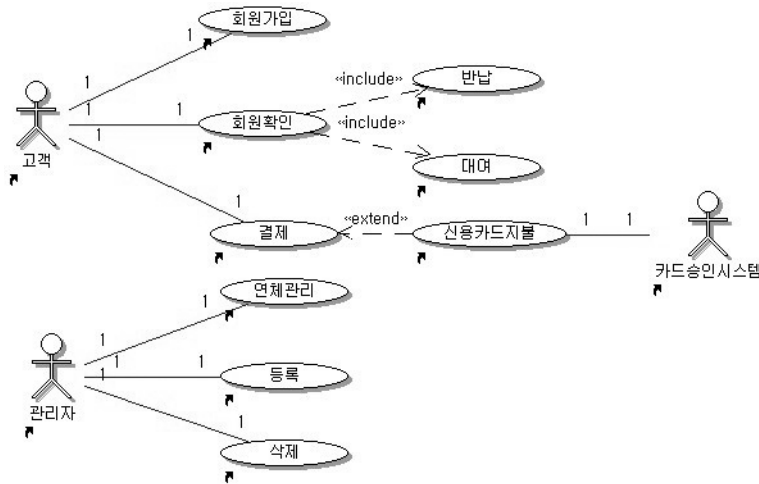
[그림 3-18] <<extend>> 관계

■ 전체적인 유스케이스 다이어그램 작성

요구명세서로부터 액터와 유스케이스의 관계 및 유스케이스 사이의 관계를 이용한 전체적인 유스케이스 다이어그램을 작성한다.

| 따라하기 |

비디오숍 관리 시스템의 전체적인 유스케이스 다이어그램은 [그림 3-19]와 같다.



[그림 3-19] 비디오숍 관리 시스템의 유스케이스 다이어그램

- ‘고객’은 ‘회원가입’을 통해 시스템에 접근할 수 있다.
- ‘대여’, ‘반납’을 수행할 때는 반드시 ‘회원확인’을 해야 한다.
- ‘결제’시에 신용카드를 사용하는 경우, ‘신용카드지불’ 유스케이스로 확장할 수 있으며, 신용카드지불이 요청되면 ‘카드승인 시스템’을 통해 카드승인을 수행한다.
- ‘고객’은 ‘회원가입’, ‘대여’, ‘반납’, ‘결제’ 업무를 수행할 수 있다.
- ‘관리자’는 ‘등록’, ‘삭제’, ‘연체관리’ 업무를 수행할 수 있다.

5단계 : 유스케이스 명세서 작성

유스케이스 다이어그램에서 유스케이스는 시스템이 수행해야 할 기능이 무엇인지만 표현하고, 유스케이스 내부에 대한 기술은 유스케이스 명세서를 통해서 기술할 수 있다. 유스케이스는 시스템의 기능을 표현하고 유스케이스 명세서는 각각의 유스케이스에 대해서 해당 유스케이스가 어떻게 수행되는지를 표현하는 것이다.

UML에서 유스케이스 명세서를 기술하는 방법이나 양식은 제공하지 않지만, 일반적으로 유스케이스 명세서에 기술되는 항목들은 다음과 같다. 특히 시나리오는 유스케이스의 특정

예를 구체적인 값을 통해 예시하는 것이므로, 가능한 한 다양한 업무유형을 알고 있는 업무 전문가가 작성하여야 하고, 시스템을 개발할 때 테스트 케이스로 유형을 충분히 파악하여 신중히 작성하는 것이 좋다.

- 유스케이스명
- 액터명
- 유스케이스 개요
- 이벤트 흐름
 - 정상 흐름(Normal Flow) : 유스케이스가 사건 주 흐름을 표현하는 절차이다.
 - 선택 흐름(Alternative Flow) : 유스케이스 내의 사건 주 흐름 외에 수행되는 절차를 표현한다.

| 따라하기 |

다음은 ‘회원가입’ 유스케이스 명세서 예이다.

- 유스케이스명 : 회원가입
- 액터명 : 고객(비회원)
- 유스케이스 개요 및 설명 : 고객이 비디오풀 관리 시스템을 사용하기 위해 회원가입을 하는 유스케이스이다.
- 사전 조건 : 회원에 가입되어 있지 않은 상태여야 한다.
- 이벤트 흐름
 - 정상 흐름
 - ① 기존 가입 회원인지 주민등록번호를 검색하여 확인한다(시스템).
 - ② 회원가입을 요청한다(액터).
 - ③ 회원약관을 보여준다(시스템).
 - ④ 회원약관에 동의한다(액터).
 - ⑤ 회원정보 입력항목을 보여준다(시스템).
 - ⑥ 회원정보, 항목(이름, 주민등록번호, 전화번호, 핸드폰번호, 이메일 등)을 입력하고 등록 요청을 한다(액터).
 - ⑦ 입력된 정보를 확인한다(시스템).
 - ⑧ 회원정보를 저장, 등록을 완료한다(시스템).
 - 선택 흐름
 - ▶ 기존에 가입되어 있는 회원인 경우 “이미 가입된 회원입니다”라는 메시지를 보여준다.
 - ▶ 회원약관에 동의하지 않을 경우 약관 동의하에 회원가입 가능 오류 메시지를 보여주고 동의를 요청한다.
 - ▶ 회원정보 입력항목 중 입력하지 않은 항목이 있을 경우 오류 메시지를 띄우고 재입력을 요청한다.
 - ▶ 등록번호의 형식이 틀렸을 경우 메시지를 보여주고 재입력을 요청한다.

6단계 : 유스케이스 실체화

유스케이스 실체화는 도출된 기능 중심의 유스케이스를 구현 시스템의 구성 요소로 구체화시키는 작업이다. 실체화 과정을 통해 명세서 중심의 유스케이스를 구현 시스템의 논리적 구성 요소인 클래스로 식별하고 이의 통신관계를 파악한다. 이를 위해 UML의 순차 다이어그램과 활동 다이어그램이 사용된다.

순차 다이어그램은 이벤트 흐름 모델을 나타내고, 활동 다이어그램은 화면 흐름 모델을 표현하기 위해 사용된다. 즉 유스케이스 다이어그램은 시스템 외부의 뷰를 표현하는 반면, 실체화를 위해 작성하는 다이어그램은 어떻게 유스케이스들이 실현되는지를 객체들간 메시지 흐름의 상호작용으로 설명함으로써 실체화된다고 볼 수 있다. 이처럼 요구사항 정의 활동의 산출물은 유스케이스 모델, 유스케이스 명세서, 이벤트 흐름 모델, 그리고 화면 흐름 모델이며, 이들을 통칭해서 요구사항 모델이라 정의한다. [표 3-6]은 요구사항 정의 활동의 각 산출물을 작성할 때 이용되는 UML 다이어그램과 산출물 작성에 대한 필수 여부를 요약한 것이다.

[표 3-6] 요구사항 정의 활동의 산출물

산출물		UML 다이어그램	필수 여부
요구사항 모델	유스케이스 모델	유스케이스 다이어그램	필수
	유스케이스 명세서	이용 안 함	필수
	이벤트 흐름 모델	순차 다이어그램	선택
	화면 흐름 모델	활동 다이어그램	선택

이번 장에서는 유스케이스 모델과 유스케이스 명세서 작성에 대해서 알아보았고, 유스케이스 실체화 부분의 이벤트 흐름 모델과 화면 흐름 모델에 대한 내용은 ‘5장. 순차 다이어그램’과 ‘7장. 활동 다이어그램’ 부분에서 자세히 알아보도록 한다.



6

재고관리 시스템 유스케이스 모델링

재고관리 시스템에 대한 유스케이스 모델링에 대하여 확실히 이해하도록 하자. 재고관리 시스템의 문제 기술서는 다음과 같다.

예제 3-2

재고관리 시스템

1단계 : 시스템 상황 분석(문제 기술서 작성)

한 인터넷 쇼핑몰에서 원활한 창고의 재고를 관리하기 위해 재고관리 시스템을 구축하고자 한다. 재고관리 시스템은 크게 입고관리 기능, 출고관리 기능, 현황관리 기능을 제공한다.

- 입고관리 기능 : 창고로 입고된 상품을 현황관리에 추가한다. 입고는 통신업체로부터 새로운 상품을 입고 받거나 고객의 반품에 의한 것이다. 입출고 담당자는 입고된 제품의 상태를 파악하고 불량 조치하여 통신업체에 반품할 수 있다.
- 출고관리 기능 : 창고로 출고된 상품을 현황관리에서 뺀다. 출고는 고객이 구매한 상품을 발주하는 것과 판매하고 남은 상품을 통신업체로 반품한 것이다.
- 현황관리 기능 : 입고와 출고된 현황을 실시간으로 인터넷 쇼핑몰에 업데이트하는 기능이다. 입출고 담당자는 현황을 조회할 수 있고 현황관리 담당자는 재고현황을 관리하여 협력업체에 주문 혹은 반품을 요청하고 쇼핑몰에 업데이트를 한다.

2단계 : 액터 식별

재고관리 시스템의 사용자로는 입출고 담당자, 현황관리 담당자를 생각할 수 있다. 또한 개발될 재고관리 시스템과 연동되는 다른 시스템에 해당하는 시스템 액터로는 쇼핑몰 시스템을 생각할 수 있다. 따라서 예제의 재고관리 시스템에 해당하는 시스템의 액터는 ‘입출고 담당자’, ‘현황관리 담당자’, ‘쇼핑몰 시스템’이 될 수 있다. [그림 3-20]은 재고관리 시스템의 액터를 나타낸 것이다.



[그림 3-20] 재고관리 시스템의 액터

3단계 : 유스케이스 식별

유스케이스는 시스템이 제공하는 하나의 단위 기능이다. 따라서 모든 유스케이스를 찾으면 시스템의 모든 기능을 찾는 것이고, 이는 결국 시스템의 모든 요구사항을 찾는 것이라고 볼 수 있다. 재고관리 시스템의 경우에는 [표 3-7]과 같은 시스템 기능, 즉 유스케이스를 생각할 수 있을 것이다. [그림 3-21]은 재고관리 시스템의 유스케이스를 나타낸 것이다.

[표 3-7] 재고관리 시스템의 사용자별 기능

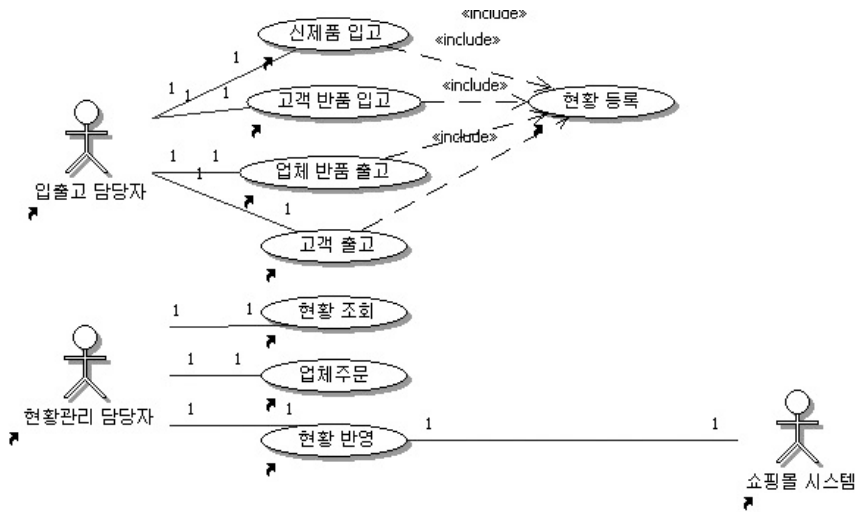
기능 범주	사용자	기능(유스케이스)
입고관리	입출고 담당자	통신업체 제품 입고 기능 고객 반품 입고 기능
출고관리	입출고 담당자	통신업체 반품 출고 기능 고객 출고 기능 발주 기능
현황관리	입출고 담당자	현황 조회
	현황관리 담당자	현재 현황 등록 기능(업데이트) 통신업체 주문 기능
	쇼핑몰 시스템	실시간 현황 반영 기능



[그림 3-21] 재고관리 시스템의 유스케이스

4단계 : 유스케이스 다이어그램 작성

액터와 유스케이스와의 관계는 연관관계로 표현하고 유스케이스 사이의 관계는 의존관계로 표현한다. 또한 재고관리 시스템의 경우, '신제품 입고', '고객 출고' 등 대부분의 유스케이스가 수행될 때는 반드시 '현황 등록' 유스케이스가 수행되어야 한다. 그러므로 '현황 등록' 유스케이스를 모든 유스케이스들의 포함관계로 표시한다. 이를 바탕으로 [그림 3-22]는 전체적인 재고 관리 시스템의 유스케이스 다이어그램을 나타낸 것이다.



[그림 3-22] 재고관리 시스템의 유스케이스 다이어그램

5단계 : 유스케이스 명세서 작성

유스케이스 명세서는 유스케이스별로 작성한다. 예를 들어 재고관리 시스템의 '신제품 입고' 유스케이스 명세서는 다음과 같이 작성될 수 있다.

- ▶ 유스케이스명 : 신제품 입고
- ▶ 액터명 : 입출고 담당자
- ▶ 유스케이스 개요 및 설명 : 입출고 담당자는 신제품이 입고되면 제품의 상태를 확인하고 입고 또는 반품시킨다.
- ▶ 사전 조건 : 현황관리 담당자가 통신업체에 주문한 제품이다.

- 이벤트 흐름
 - 정상 흐름
 - ① 통신업체로부터 제품 입고를 요구한다.
 - ② 입출고 담당자는 주문한 제품이 맞는지 확인한다.
 - ③ 제품의 상태를 파악한다.
 - ④ 제품을 입고하고 제품 목록을 현황관리 담당자에게 알린다.
 - 선택 흐름
 - ① 제품에 하자가 발생하면 현황관리 담당자에게 하자를 알리고 반품한다.



요약

1 유스케이스(Usecase)

유스케이스는 우리말로 쓰는새라고도 한다. 유스케이스라 함은 말 그대로 '쓰이는 경우' 라든가 '용도' 와 같은 의미이다. 유스케이스는 사용자 시각에 맞춘 분석을 의미하며, 어떤 시스템을 만드느냐를 사용자 입장에서 정리한 것이다. 유스케이스는 시스템의 행위를 결정하는 것이다. 구체적으로는 시스템의 기능을 정의하고, 범위를 결정함으로써 시스템과 외부 환경 변수를 구분하고, 상호 관계를 정립하는 것이라고 볼 수 있다. 유스케이스를 나타내는 유스케이스 모델(Model)은 유스케이스 다이어그램으로 표현된다. 유스케이스 다이어그램은 액터(Actor, 행위자)와 유스케이스, 그리고 관계(Relationship)로 표현한다.

2 포함관계(Include Relationship)

포함관계는 하나의 유스케이스가 실행할 때 반드시 실행하는 경우이다. 예를 들면 회원이 로그인할 때 반드시 아이디와 패스워드를 입력한다. 이는 반드시 해야 할 일이므로 포함관계이다.

3 확장관계(Extend Relationship)

확장관계는 하나의 유스케이스가 실행할 때 특정 조건에 한하여 실행하는 경우이다. 물건을 사고 돈을 지불할 때 카드로 할 경우 카드승인시스템에 접속해야 하는 유스케이스가 있다. 이 경우 구매와 신용카드 지불 유스케이스는 확장관계이다.

4 액터(Actor)의 의미와 식별

액터는 시스템의 일부가 아니다. 액터는 시스템과 상호작용을 하는 모든 것들을 나타낸다. 시스템을 사용하게 될 사람은 물론이고, 연관된 다른 시스템도 액터이다. 대체로 액터의 행위는 정보의 입력과 출력으로 살펴볼 수 있다. 정보를 입력하거나 출력하는 액터가 있고, 입출력을 모두 행하는 액터가 있을 것이다. 액터는 다이어그램 상에서 막대인간(stickman)으로 표현된다. 'Visual Modeling with Rational Rose and UML' 에 따르면 다음과 같은 질문들이 액터를 추출하는 데 도움을 준다고 한다.

- 특정 요구사항에 이해관계자는 누구인가?
- 어떠한 부서나 집단에서 시스템을 사용하는가?
- 시스템을 사용함으로써 이익을 얻는 이는 누구인가?
- 누가 시스템에 정보를 입력하고 사용하고 삭제하는가?
- 누가 시스템의 유지보수를 수행하는가?
- 시스템이 외부 자원을 사용하는가?
- 한 사람이 복수의 역할을 수행하는가?
- 여러 사람이 한 가지 역할을 수행하는가?
- 시스템이 레거시 시스템(Legacy System)과 상호 작용하는가?



연습문제

다음 5가지의 문제 기술서를 보고 각각의 유스케이스 다이어그램을 만들어라.

1 인터넷 쇼핑물 시스템 문제 기술서

- ① 우리는 인터넷을 통해 물품을 판매하는 시스템을 구축하고자 한다.
- ② 고객은 회원등록을 통해 본인 정보를 등록하고, 아이디, 패스워드를 입력함으로써 시스템에 접속할 수 있으며, 물품을 구입할 수 있다.
- ③ 고객이 로그인을 할 때 및 물품을 구입할 때에는 별도 암호화된 모듈을 통해 본인확인 절차를 거쳐야 한다.
- ④ 물품을 구매하여 결제할 때에는 일반 포인트로 결제할 수 있으며, 신용카드를 통해 결제할 수도 있다.
- ⑤ 신용카드를 통해 결제하고자 하는 경우에는 협약을 맺은 카드승인사에 카드승인을 요청하여야 한다.
- ⑥ 고객은 구매한 물품의 배송상황을 조회할 수 있으며, 직원도 고객 물품에 대해 배송추적을 할 수 있다.

2 주문관리 시스템 문제 기술서

- ① 주문관리 시스템은 고객확인을 통해 시스템에 접속할 수 있다. 고객확인은 회원가입을 할 때 입력한 패스워드와 아이디를 입력함으로써 확인한다.
- ② 접수관리 직원은 확보된 상품을 등록한다. 상품 확보는 재고관리 직원이 한다. 고객은 등록된 상품을 선택하고 장바구니에 등록한다. 장바구니에 등록된 상품은 카드나 계좌이체를 통해 결제를 하면 운송직원이 상품을 고객에게 배달한다.

3 비디오숍 시스템 문제 기술서

- ① 반드시 회원가입을 통한 회원만이 비디오를 대여할 수 있다. 회원은 가입된 이름과 전화번호를 통해 확인한다.
- ② 비디오숍 시스템은 대여, 반납, 연체 확인 기능이 있다.
- ③ 대여는 고객이 비디오를 선택하면 비디오 코드를 확인하여 시스템에 입력한다. 현재 고객이 대여 중인 비디오가 있으면 표시하고 대여 기간이 지났으면 연체료를 계산하여 보여준다. 연체 중인 고객은 비디오 대여를 할 수 없다. 그러나 연체료를 납부하면 대여할 수 있다. 대여된 비디오는 대여목록에 비디오 코드와



연습문제

고객명이 등록된다.

- ④ 관리자는 연체관리 기능을 통해 현재 연체 중인 회원과 연체된 비디오를 확인한다. 연체금 표시 기능을 통해 오늘 날짜에 해당하는 연체금을 표시한다.
- ⑤ 반납 기능은 반납한 비디오의 코드를 입력하여 대여목록에서 삭제한다.

4 항공기 예약 시스템 문제 기술서

- ① 고객은 예약 시스템을 통해 항공기 예약을 한다. 티켓 구입을 위해 고객은 항공기 운항 노선을 선택한다.
- ② 선택한 노선 중 원하는 항공사를 선택한다.
- ③ 선택한 항공사의 이용 가능 인원을 확인하고 예약할 인원을 입력한다.
- ④ 인원 확인 결과 가능하다면 예약을 한다. 예약 후 운임 납부를 위해 카드 또는 계좌이체를 통해 결제한다. 결제가 완료되면 예약 확인서를 다운받는다.

5 도서관리 시스템 문제 기술서

- ① 도서관리 시스템은 시스템에 등록된 사람들에게 책과 잡지를 빌려준다.
- ② 사서는 새로운 책이 있으면 구입하고 오래된 책과 잡지는 조건을 적용하여 제거한다. 구입하거나 제거한 책은 도서목록에 추가하거나 제거한다.
- ③ 고객은 원하는 책이 현재 도서관에서 이용이 불가능할 때 예약을 할 수 있고, 도서관으로 책이 반납되거나 새로 구매를 한 경우에 연락을 받아 이용할 수 있다. 고객이 책을 빌려갔거나 직접 취소를 원하면 예약을 취소한다.