

## UseCase 다이어그램 모델링하기

유스케이스 다이어그램에서 편집할 수 있는 요소들은 다음과 같습니다.

- Actor
- UseCase
- Association
- Directed Association
- Generalization
- Dependency
- Include
- Extend
- System Boundary
- Package

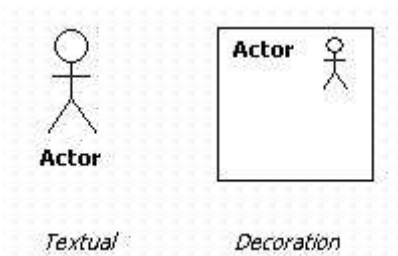
### Actor

의미:

액터(Actor)는 일반적으로 시스템 외부에 존재하면서 시스템과 상호작용하는 개체입니다. 액터는 사람이거나 기계 혹은 소프트웨어 등이 될 수 있습니다.

**Actor** 생성 방법:

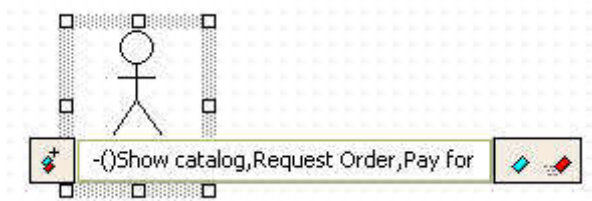
Actor를 생성하려면, **Toolbox>UseCase**의 **Actor** 버튼을 클릭하고 **Main** 윈도우창에서 Actor가 위치할 곳을 클릭합니다. Actor는 Stick Man 형태로 표현되지만, 사각형 모양에 오른쪽 상단에 아이콘이 포함된 **Decoration View** 형태로 사용되기도 합니다. Actor를 Decoration View 형태로 보여지도록 하기 위해서는 **[Format] -> [Stereotype Display] -> [Decoration]** 메뉴 아이템을 선택하거나 툴바의  콤보 버튼의 **[Decoration]** 항목을 선택합니다.



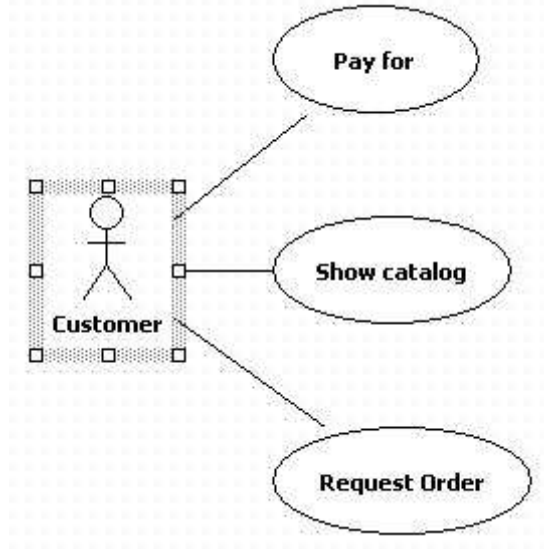
**Actor**가 사용하는 **UseCase**를 한번에 여러 개 생성하는 방법:

Actor가 사용하는 여러 개의 UseCase를 한꺼번에 만들려면 Actor의 단축 생성 구문을 사용합니다.

1. Actor를 더블 클릭해서 쿼다이얼로그가 나타나면, 쿼다이얼로그에서 "-()" 문자열 다음에 생성하려는 UseCase의 이름을 입력합니다. 각 UseCase 이름은 "," 문자로 구분해서 입력합니다.



2. 그리고 **[Enter]**키를 누르면 Actor와 연관 관계를 가지는 여러개의 UseCase가 수직으로 자동 배열되어 생성됩니다.




## UseCase

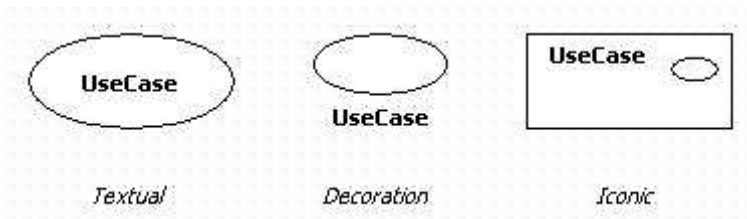
의미:

유스케이스(UseCase)는 시스템의 행위(behavior)를 정의하기 위해 사용하는 요소입니다. 일반적으로 유스케이스는 액터와 상호 작용합니다.

**UseCase**를 생성하는 방법:

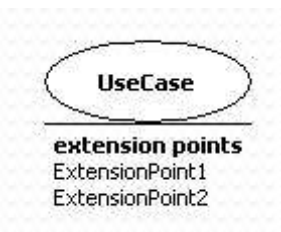
UseCase를 생성하려면, **[Toolbox] -> [UseCase] -> [UseCase]** 버튼을 클릭하고 Main 윈도우창에서 UseCase가 위치할 곳을 클릭합니다.

UseCase는 Textual, Decoration, Iconic의 3가지 형태로 표현 가능합니다. **[Format] -> [Stereotype Display]**의 하부 메뉴 아이템을 선택하거나  버튼의 아이템을 선택하면, UseCase의 스타일을 변경할 수 있습니다.

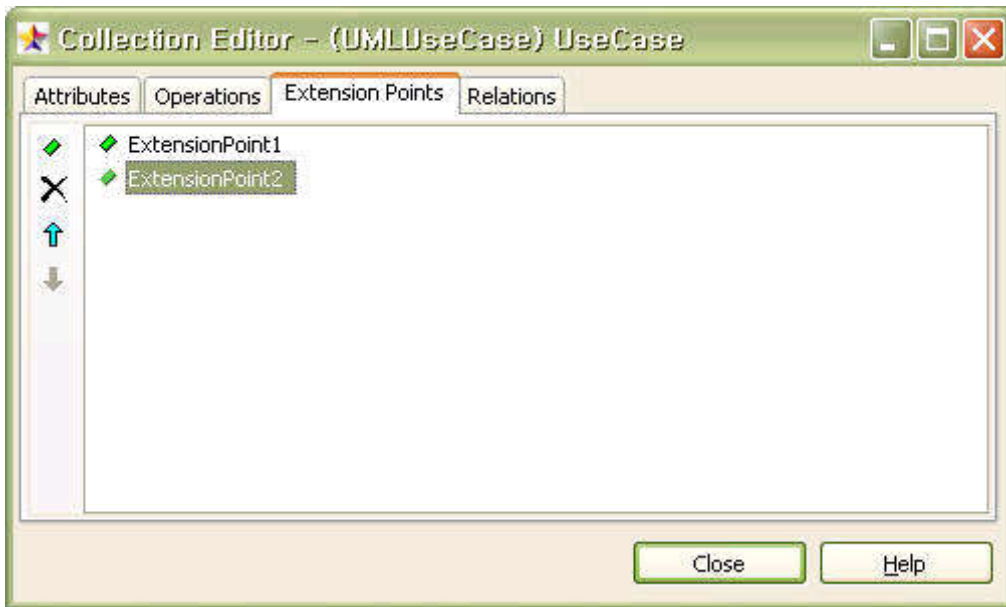


**Extension** 추가하는 방법:

확장점은 유스케이스에서 확장되어지는 하나 또는 여러개의 위치를 참조합니다.

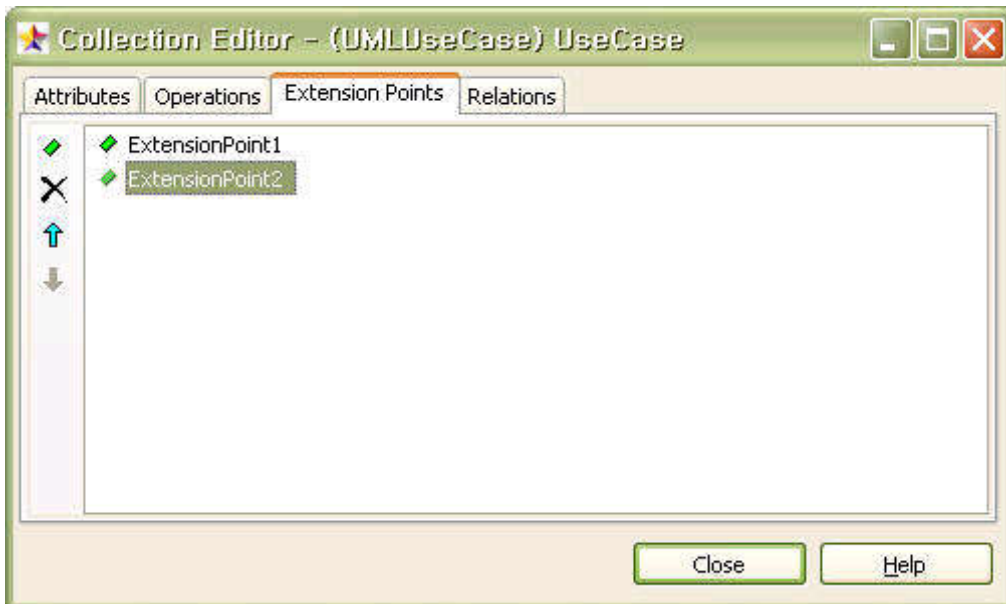


UseCase에 ExtensionPoints를 입력하려면 UseCase의 **[Collection Editor...]** 팝업 메뉴를 클릭하거나 UseCase의 ExtensionPoints 컬렉션 속성의  버튼을 클릭해서 **[Collection Editor]**에서 값을 수정합니다.



### UseCase Specification 속성 입력 방법:

UseCase 작성시 많이 사용되는 속성들인 BasicFlow, AlternativeFlow등을 입력하기 위해서는 **[Tagged Values...]** 팝업 메뉴를 선택하거나 Ctrl+F7 버튼을 클릭하여 Tagged Value Editor의 UseCaseSpecification을 선택하여 필요한 속성의 값을 입력합니다.



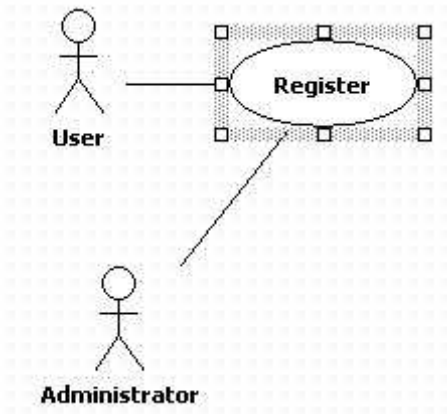
### UseCase로부터 Actor 생성하는 방법:

현재 선택된 UseCase와 연관 관계를 가지는 Actor 여러개를 한꺼번에 만들려면 UseCase의 단축 생성 구문을 사용합니다.

1. UseCase를 더블 클릭하거나 UseCase를 선택하고 **[Enter]**키를 누른다. Quick Dialog가 나타나면, Quick Dialog에서 "()-" 문자열 다음에 연관된 Actor의 이름을 입력합니다. 각 Actor 이름은 "," 문자로 구분해서 입력합니다.



2. 그리고 **[Enter]**키를 누르면 UseCase와 연관 관계를 가지는 Actor들이 생성됩니다.



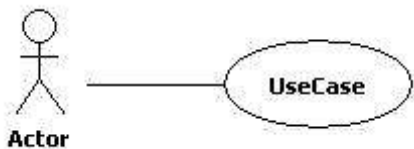
## Association / Directed Association

의미:

연관(Association)은 클래스류(Class, Interface, Enumeration, Signal, Exception, Component, Node, UseCase, Actor) 사이의 의미적 관계를 정의합니다.

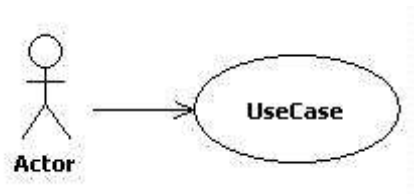
**Association** 생성하는 방법:

Association을 생성하려면, **[Toolbox] -> [UseCase] -> [Association]** 버튼을 클릭하고 Main 윈도우창에서 연결하려는 첫번째 요소에서 두번째 요소로 마우스를 누르고 드래그하면 됩니다.

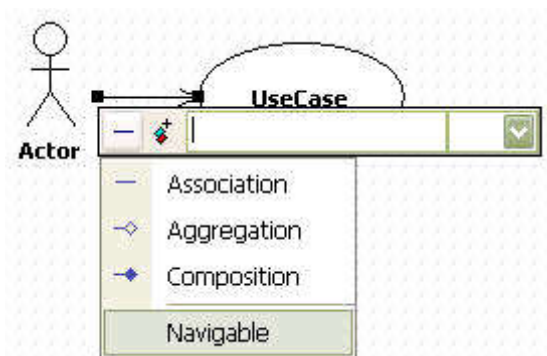


**Directed Association** 생성하는 방법:

Association 생성방법과 동일하며, 두 요소간 마우스 드래그를 화살표 방향으로 합니다.



또는 Association을 생성하고 Actor쪽 association의 끝을 클릭하고 Quick Dialog의 Navigable의 체크를 취소하면 DirectedAssociation으로 변환합니다.



요소로부터 **Association/Directed Association** 관계의 요소 생성하는 방법:

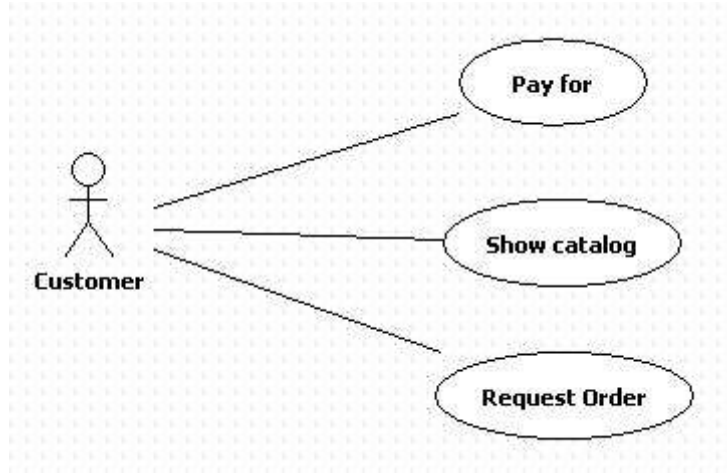
현재 선택된 요소로부터 Association/DirectedAssociation 관계를 갖는 요소를 만들려면 요소의 단축 생성 구문을 사용합니다.

1. 요소를 더블 클릭해서 Quick Dialog가 나타나면, Quick Dialog에서 "--" 또는 "->" 문자열 다음에 Association/DirectedAssociation 관계를 갖는 다른 요소의 이름을 입력합니다. 여러개의 요소와 관계를 맺기 위해서는

각 요소 이름은 "," 문자로 구분해서 입력합니다.



- 그리고 **[Enter]**키를 누르면 선택된 요소와 Association/DirectedAssociation 연관 관계를 가지는 여러 요소들이 생성되고 자동 배열되어 생성됩니다.



## Generalization

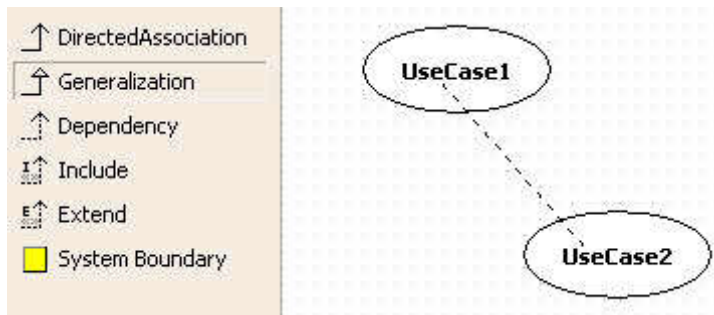
의미:

일반화(Generalization)">일반화(Generalization)는 더 일반적인 요소와 더 구체적인 요소를 연결하는 관계입니다.

**Generalization** 생성하는 방법:

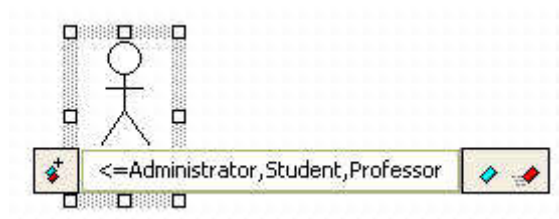
### Procedure for creating generalization

Generalization를 생성하려면, **[Toolbox] -> [UseCase] ->[Generalization]** 버튼을 클릭하고 Main 윈도우창에서 연결하려는 자식 요소에서 부모 요소로 마우스를 누르고 드래그하면 됩니다.

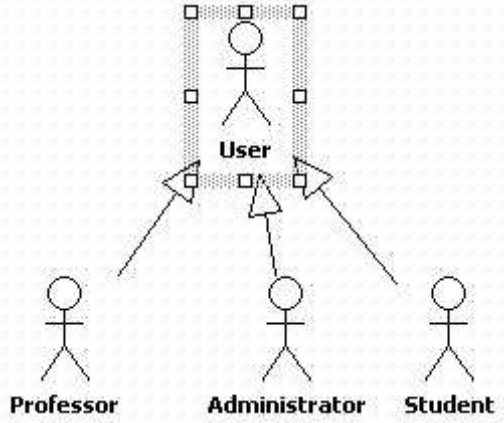


**Actor**를 상속하는 여러개의 자식 **Actor** 생성하는 방법:

- 특정 요소를 상속하는 하위 요소가 여러개일 경우에 Quick Dialog의 단축 생성 구문에서 다음과 같이 입력하면 현재 요소를 상속하는 여러개의 하위 요소를 한꺼번에 생성합니다.



2. 하위 요소들은 선택된 요소의 아래에 생성되면서 정렬됩니다.



만약 상속할 상위 요소가 여러개인 경우에는 Quick Dialog의 단축 생성 구문에서 "<=" 대신에 ">=" 문자열을 사용한다.

## Dependency

의미:

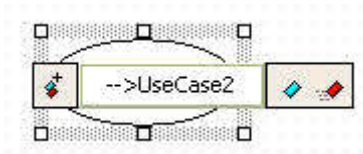
의존관계(Dependency)"> 의존관계(Dependency)는 어떤 요소의 구현이나 기능을 위해 다른 요소의 존재가 요구되어지는 의존적인 관계를 의미합니다.

**Dependency 생성 방법:**

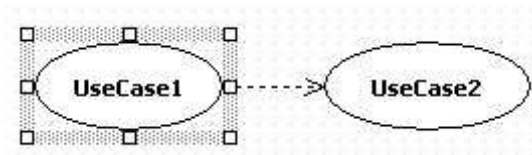
Dependency를 생성하려면, [Toolbox] -> [UseCase] -> [Dependency] 버튼을 클릭하고 Main 윈도우창에서 요소에서 의존 하는 요소로 마우스를 누르고 드래그하면 됩니다.

**UseCase로부터 의존하는 다른 UseCase 생성하는 방법:**

쿼다이어로그의 단축 생성구문을 다음과 같이 입력하면 됩니다.



그러면 다음과 같이 두 요소간의 Dependency가 생성됩니다.



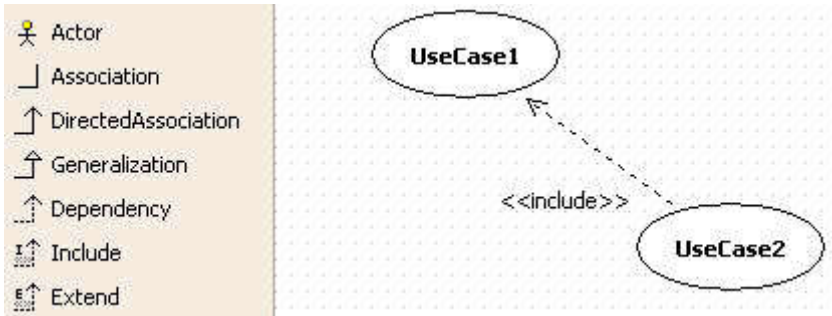
## Include

의미:

포함관계(Include)는 어떤 유스케이스가 특정 유스케이스의 행위를 포함한다는 것을 정의합니다.

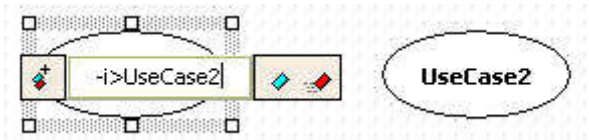
**Include 생성 방법:**

Include를 생성하려면, [Toolbox] -> [UseCase] -> [Include] 버튼을 클릭하고 Main 윈도우창에서 요소에서 포함할 요소로 마우스를 누르고 드래그하면 됩니다.

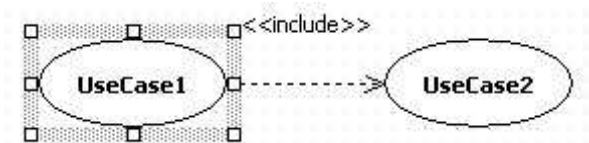


### UseCase로부터 Include 관계의 다른 UseCase 생성하는 방법:

쿼다이어로그의 단축 생성구문을 다음과 같이 입력하면 됩니다.



그러면 다음과 같이 두 요소간의 Include가 생성됩니다.



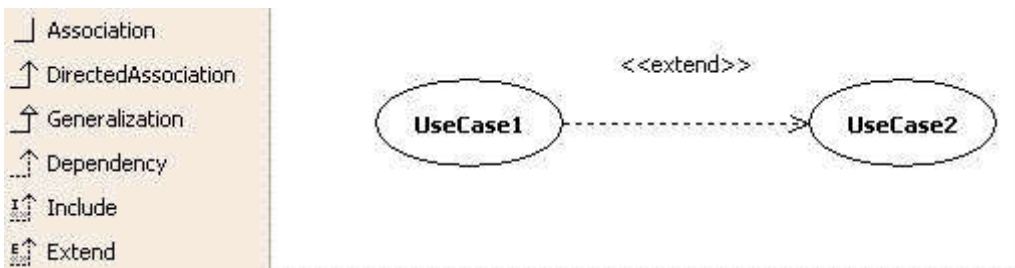
## Extend

의미:

확장관계(Extend)">확장관계(Extend)는 어떤 유스케이스가 특정 유스케이스에 정의된 행위로 추가 확장될 수 있다는 것을 나타냅니다.

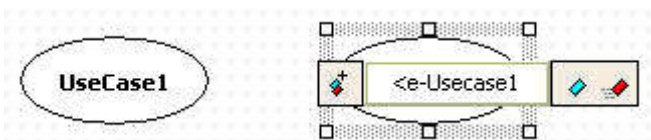
### Extend 생성 방법:

Extend를 생성하려면, [Toolbox] -> [UseCase] -> [Extend] 버튼을 클릭하고 Main 윈도우창에서 요소에서 확장할 요소로 마우스를 누르고 드래그하면 됩니다.

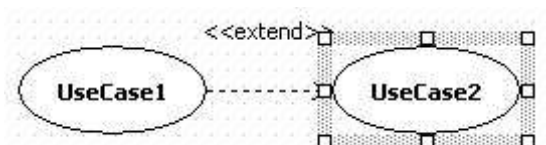


### UseCase로부터 Extend 관계의 다른 UseCase 생성하는 방법:

쿼다이어로그의 단축 생성구문을 다음과 같이 입력하면 됩니다.



그러면 다음과 같이 두 요소간의 Extend가 생성됩니다.

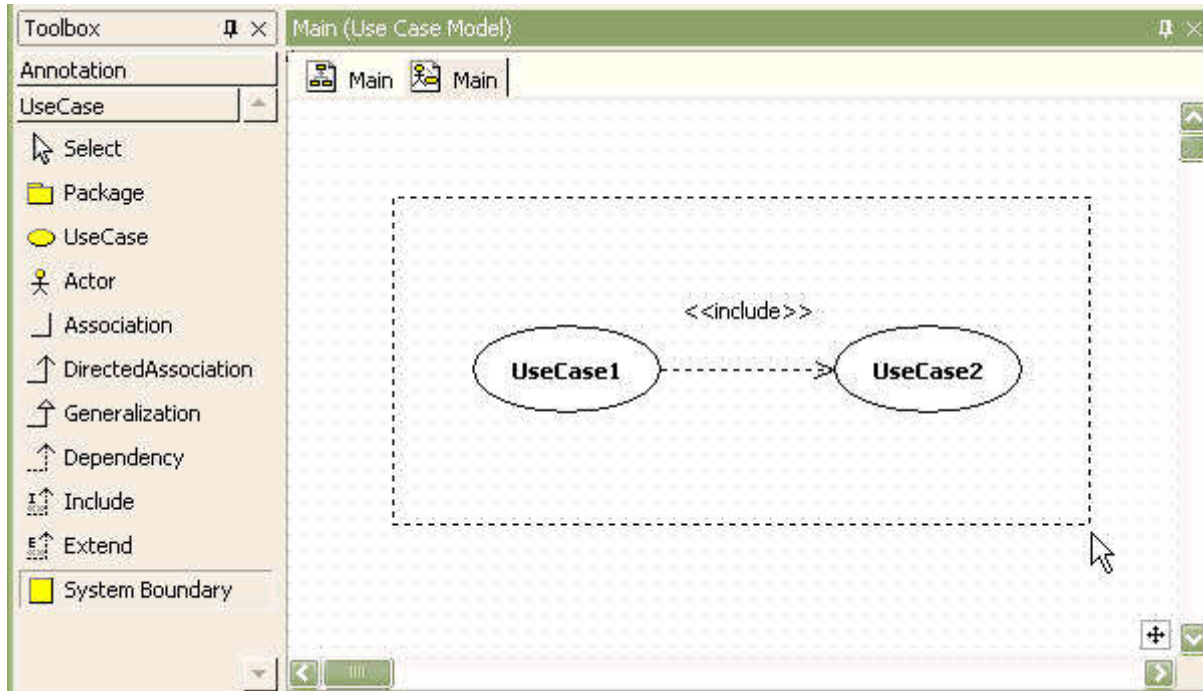


## System Boundary



**System Boudary**를 생성하는 방법:

System Boundary를 생성하려면, **[Toolbox] -> [UseCase] -> [System Boundary]** 의 System Boundary 버튼을 클릭하고 Main 윈도우창에서 System Boundary가 삽입될 위치에 마우스를 클릭하고 생성될 크기 만큼을 드래그합니다.

**Package**

의미:

패키지(Package)는 모델 요소들을 논리적으로 그룹화하여 관리하기 위한 요소입니다. 패키지는 요소들을 조직화하기 위한 어떠한 용도로 사용되어도 무방한 매우 일반적인 요소입니다. 패키지 대신 모델(Model), 서브시스템(Subsystem)의 더욱 특수화된 요소를 사용할 수도 있습니다.

**Package** 생성하는 방법:

Package를 생성하려면, **[Toolbox] -> [UseCase] -> [Package]** 버튼을 클릭하고 Main 윈도우창에서 Package가 위치할 곳을 클릭합니다.

