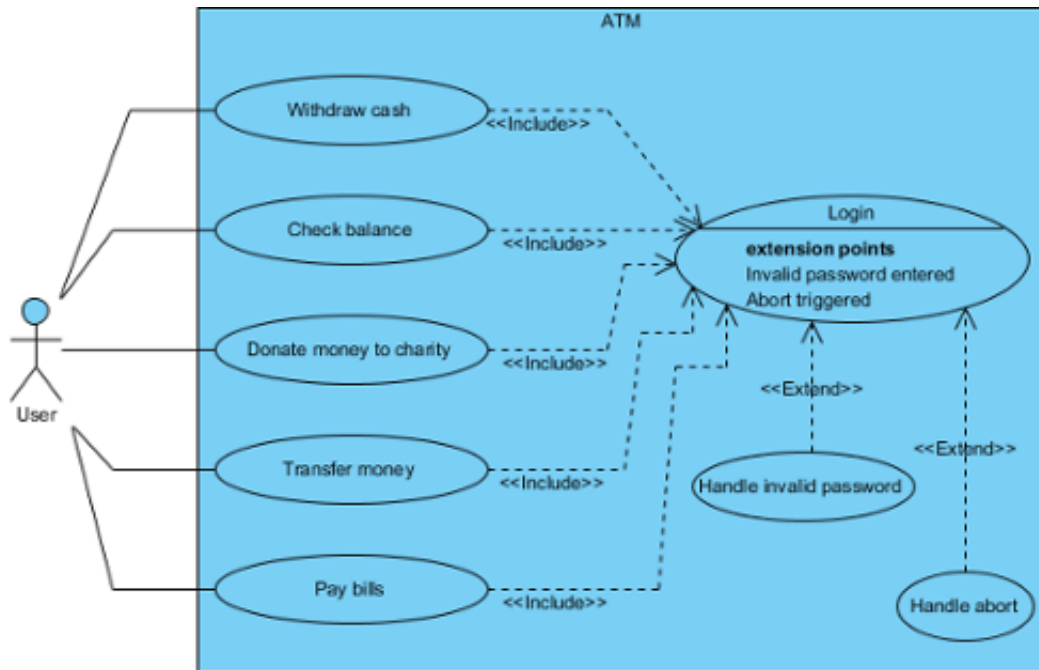


# Use Case Diagram notations guide



[Use case diagram](#) is a kind of [UML diagram](#). Here is a list of Unified Modeling Language (UML) notations supported in a UML use case diagram:

Icon	Name
	<a href="#">Use Case</a>
	<a href="#">Association</a>
	<a href="#">Actor</a>
	<a href="#">System</a>
	<a href="#">Include</a>
	<a href="#">Extend</a>
	<a href="#">Dependency</a>
	<a href="#">Generalization</a>
	<a href="#">Realization</a>
	<a href="#">Collaboration</a>

## Use Case



A use case represents a user goal that can be achieved by accessing the system or software application. In Visual Paradigm, you can make use of the sub-diagram feature to describe the interaction between user and system within a use case by creating a sub-sequence diagram under a use case. You can also describe the use case scenario using the Flow of Events editor.

## OMG UML Specification

---

What is a use case in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 606), use case is:

A use case is the specification of a set of actions performed by a system, which yields an observable result that is typically of value for one or more actors or other stakeholders of the system.

## Association



Actor and use case can be associated to indicate that the actor participates in that use case. Therefore, an association corresponds to a sequence of actions between the actor and use case in achieving the use case.

## OMG UML Specification

What is an association in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 36), association is:

An association describes a set of tuples whose values refer to typed instances. An instance of an association is called a link. A link is a tuple with one value for each end of the association, where each value is an instance of the type of the end. ... An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type. An end property of an association that is owned by an end class or that is a navigable owned end of the association indicates that the association is navigable from the opposite ends; otherwise, the association is not navigable from the opposite ends.

## Actor



Actors are the entities that interact with a system. Although in most cases, actors are used to

---

represent the users of system, actors can actually be anything that needs to exchange information with the system. So, an actor may be people, computer hardware, other systems, etc.

Note that actor represents a role that a user can play but not a specific user. So, in a hospital information system, you may have doctor and patient as actors but not Dr. John, Mrs. Brown as actors.

## OMG UML Specification

What is an actor in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1), actor is:

An actor

specifies a role played by a user or any other system that interacts with the subject. (The term "role" is used informally here and does not necessarily imply the technical definition of that term found elsewhere in this specification.)

.. An Actor

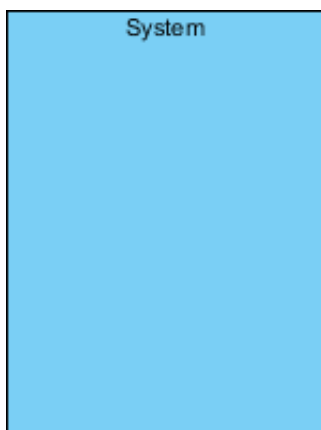
models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data) but which is external to the subject (i.e.

in the sense that an instance of an actor

is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor

does not necessarily represent a specific physical entity but merely a particular facet (i.e. "role") of some entity that is relevant to the specification of its associated use cases. Thus, a single physical instance may play the role of several different actors and conversely, a given actor may be played by multiple different instances.

## System



The scope of a system can be represented by a system (shape), or sometimes known as a system boundary. The use cases of the system are placed inside the system shape, while

---

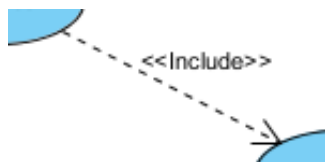
the actor who interact with the system are put outside the system. The use cases in the system make up the total requirements of the system.

## OMG UML Specification

What is a system in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 608), system is:

If a subject (or system boundary) is displayed, the use case ellipse is visually located inside the system boundary rectangle. Note that this does not necessarily mean that the subject classifier owns the contained use cases, but merely that the use case applies to that classifier.

## Include



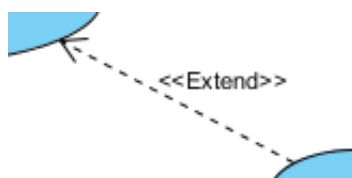
An include relationship specifies how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.

## OMG UML Specification

What is an include in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 604), include is:

An include relationship defines that a use case contains the behavior defined in another use case.

## Extend



An extend relationship specifies how the behavior of the extension use case can be inserted into the behavior defined for the base use case.

## OMG UML Specification

What is an extend in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 601), extend is:

---

A relationship from an extending use case to an extended use case that specifies how and when the behavior defined in the extending use case can be inserted into the behavior defined in the extended use case.

... This relationship specifies that the behavior of a use case may be extended by the behavior of another (usually supplementary) use case. The extension takes place at one or more specific extension points defined in the extended use case. Note, however, that the extended use case is defined independently of the extending use case and is meaningful independently of the extending use case. On the other hand, the extending use case typically defines behavior that may not necessarily be meaningful by itself. Instead, the extending use case defines a set of modular behavior increments that augment an execution of the extended use case under specific conditions.

Note that the same extending use case can extend more than one use case. Furthermore, an extending use case may itself be extended.

## Dependency



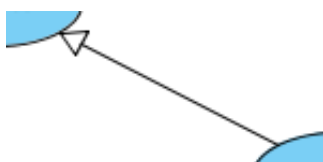
A dependency relationship represents that a model element relies on another model element for specification and/or implementation.

## OMG UML Specification

What is a dependency in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 61), dependency is:

A dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation. This means that the complete semantics of the depending elements is either semantically or structurally dependent on the definition of the supplier element(s).

## Generalization



A generalization relationship is used to represent inheritance relationship between model elements of same type. The more specific model element share the same specification with the

---

more general the model element but carries more details in extra.

## OMG UML Specification

What is a generalization in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 70), generalization is:

A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. Each instance of the specific classifier is also an indirect instance of the general classifier. Thus, the specific classifier inherits the features of the more general classifier.

## Realization



A realization is a relationship between a specification and its implementation.

## OMG UML Specification

What is a realization in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 131), realization is:

Realization is a specialized abstraction relationship between two sets of model elements, one representing a specification (the supplier) and the other represents an implementation of the latter (the client). Realization can be used to model stepwise refinement, optimizations, transformations, templates, model synthesis, framework composition, etc.

## Collaboration



## OMG UML Specification

What is a collaboration in UML? According to the OMG Unified Modeling Language (OMG UML) specification (UML Superstructure Specification version 2.4.1, page 174), collaboration is:

A collaboration describes a structure of collaborating elements (roles

---

), each performing a specialized function, which collectively accomplish some desired functionality. Its primary purpose is to explain how a system works and, therefore, it typically only incorporates those aspects of reality that are deemed relevant to the explanation. Thus, details, such as the identity or precise class of the actual participating instances are suppressed.