




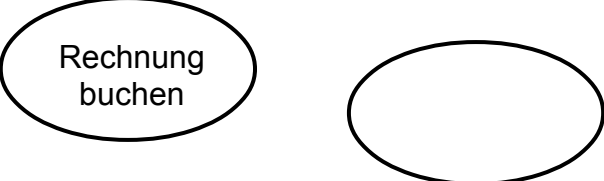
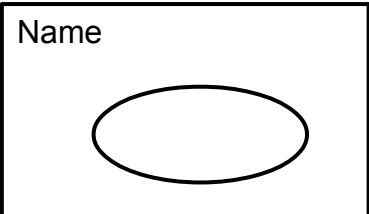
## Anwendungsfall-Diagramm

Das Anwendungsfall-Diagramm (Use-Case-Diagramm) stammt aus dem Bereich der Verhaltensdiagramme (Behaviour-Diagrams) und stellt mit grafischen Symbolen die Interaktionen von Akteuren mit einem Software-(Teil-)System anhand von von Anwendungsfällen dar.

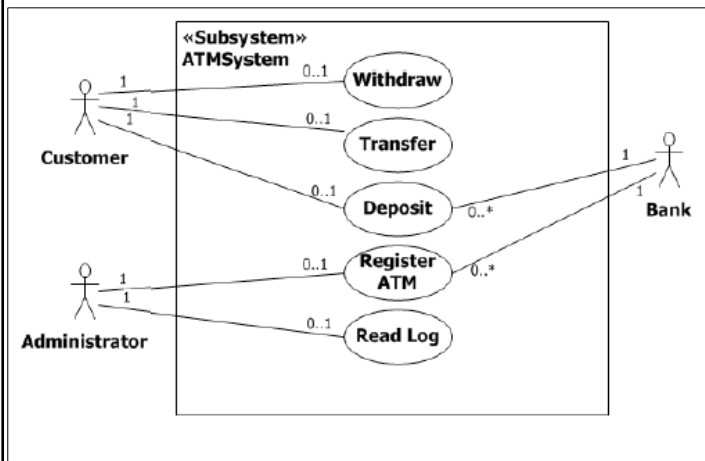
Mit dem Diagramm sollen die Fragen

- Wer interagiert mit dem Software-System? ► Akteure
- Welche Interaktionen sind geplant? ► Anwendungsfälle
- Welche Beziehungen unter welchen Bedingungen sind vorhanden? ► Assoziationen, include, extend

beantwortet werden. Ein Anwendungsfall-Diagramm kennt folgende grafischen Symbole<sup>1</sup>:

Symbol	Beschreibung
<p>Figure 18.6, Figure 18.7 and Figure 18.8 exemplify the three different notations for Actors.</p>  <p>Customer</p> <p>Figure 18.6 Actor notation using stick-man</p>  <p>Figure 18.7 Actor notation using Class rectangle</p>  <p>User</p> <p>Figure 18.8 Actor notation using icon</p>	<p>Akteure können als Strichmännchen oder als Klasse mit dem Stereotyp &lt;&lt;actor&gt;&gt; dargestellt werden.</p> <p>Weiterhin können Akteure auch mit Hilfe eines sonstigen Symbols dargestellt werden, die den Typ des Akteurs vermitteln.</p> <p>Akteure befinden sich außerhalb des Systems.</p> <p>Im Original: An Actor is represented by a "stick man" icon with the name of the Actor in the vicinity (usually above or below) the icon, as illustrated by the example in Figure 18.6.</p> <p>An Actor may also be shown as a Classifier rectangle with the keyword «actor», with the usual notation for all compartments, as illustrated by the example in Figure 18.7.</p> <p>Other icons that convey the kind of Actor may also be used to denote an Actor, such as using a separate icon for nonhuman Actors, as illustrated by the example in Figure 18.8.</p>
 <p>Auftrag erteilen</p>	<p>Ein Anwendungsfall wird als eine Ellipse dargestellt. Entweder wird der Name des Anwendungsfalls innerhalb oder unterhalb der Ellipse eingetragen.</p> <p>Im Original: A UseCase is shown as an ellipse, either containing the name of the UseCase or with the name of the UseCase placed below the ellipse. An optional stereotype keyword may be placed above the name.</p>
	<p>Das (Teil-)System, welches die Anwendungsfälle enthält, wird als Rechteck gezeichnet. Die Anwendungsfälle befinden sich innerhalb des Rechtecks. Oben steht der Name des (Teil-)Systems</p> <p>Im Original: A subject for a set of UseCases (sometimes called a system boundary) may be shown as a rectangle with its name in the top-left corner, with the UseCase ellipses visually located inside this rectangle.</p>

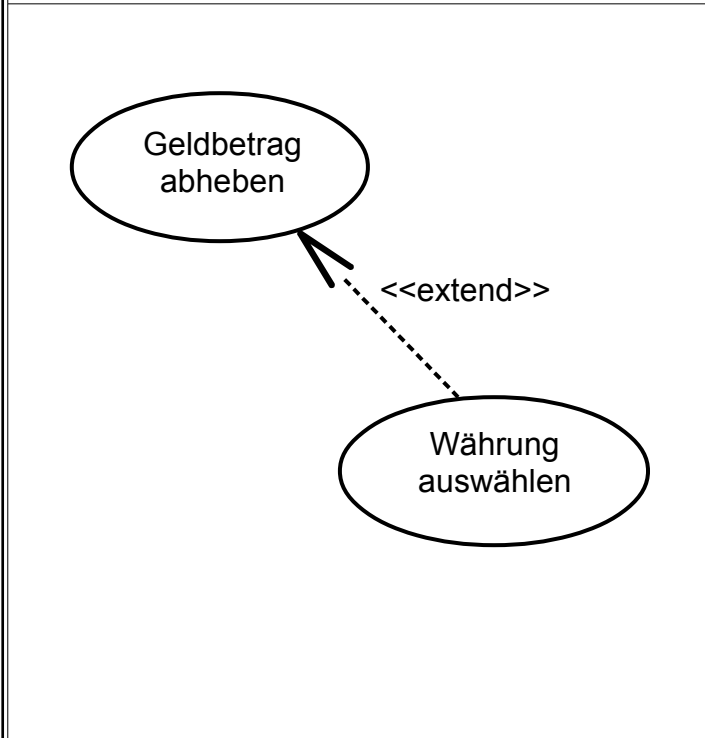
<sup>1</sup> Siehe Beschreibung der UML v2.5.1 im Dokument UML formal-17-12-05.pdf, S. 639 ff.  
© Uwe Homm Version vom 11. Oktober 2018



Eine **Assoziation** zwischen einem Akteur und einem Anwendungsfall wird als **Linie zwischen Akteur und Anwendungsfall** dargestellt.

An den Enden der Linie kann die **Multiplizität** der Beteiligten (Akteur bzw. Use-Case) dargestellt werden.

Im Original:  
 UseCases may have associated Actors, which describe how an instance of the Classifier realizing the UseCase and a user playing one of the roles of the Actor interact.



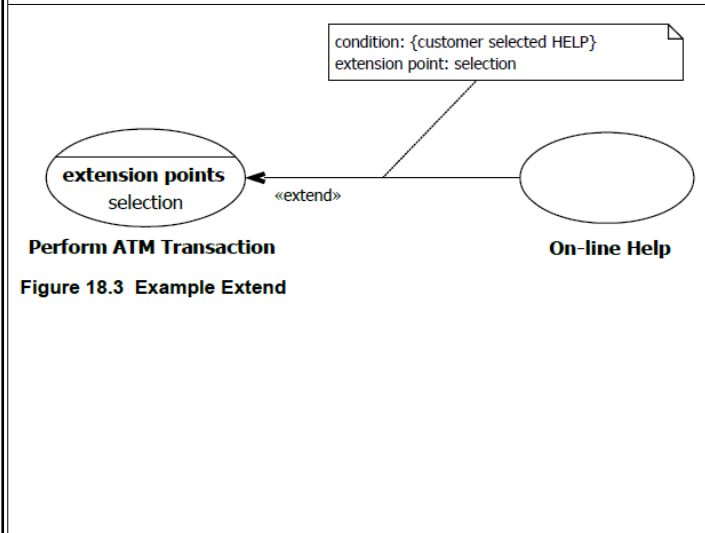
Durch eine Erweiterungs-Assoziation (Extend-Assoziation) kann ein Anwendungsfall durch einen weiteren separaten Anwendungsfall, der nur unter bestimmten Bedingungen notwendig wird, erweitert werden.

Die Erweiterungs-Assoziation wird mittels einer gestrichelten Linie mit einem offenen Pfeil am Ende des zu erweiternden Use-Case dargestellt. An der Linie steht der Text **<<extend>>**.

Im Original:  
 An Extend is a relationship from an extending UseCase (the extension) to an extended UseCase (the extendedCase) that specifies how and when the behavior defined in the extending UseCase can be inserted into the behavior defined in the extended UseCase. The extension takes place at one or more specific extension points defined in the extended UseCase.

Extend is intended to be used when there is some additional behavior that should be added, possibly conditionally, to the behavior defined in one or more UseCases.

An Extend relationship between UseCases is shown by a dashed arrow with an open arrowhead pointing from the extending UseCase towards the extended UseCase. The arrow is labeled with the keyword **<<extend>>**.



Ein Erweiterungspunkt (Extension-Point) beschreibt, unter welchen Umständen eine Erweiterung zur Ausführung kommt.

Grafisch wird dies durch den Text „extension points“ und der Angabe eines Namens im zu erweiternden Use-Case dargestellt.

Mittels einer Notiz an der extend-Linie kann die Bedingung näher beschrieben werden.

Im Original:  
 An ExtensionPoint identifies a point in the behavior of a UseCase where that behavior can be extended by an Extend relationship. Each ExtensionPoint has a unique name within a UseCase.

ExtensionPoints may be listed in a compartment of the UseCase with the heading extension points. Each ExtensionPoint is denoted by a text string within the UseCase oval symbol [...].

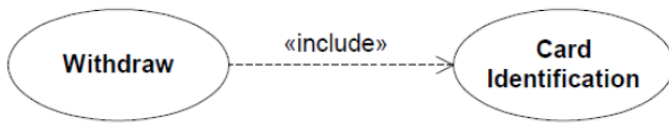


Figure 18.4 Example Include

Eine Beinhaltet-Assoziation (Include-Assoziation) wird verwendet, wenn mehrere Anwendungsfälle gleichartige gleiches Verhalten zeigen. Diese gleichen Teile des Verhaltens werden dann in einen separaten Anwendungsfall ausgelagert, der dann von den betroffenen Anwendungsfällen inkludiert wird.

Grafisch wird diese Beziehung durch eine gestrichelte Linie mit einem offenen Pfeil vom inkludierenden Use-Case hin zum inkludierten Use-Case dargestellt.

Im Original:

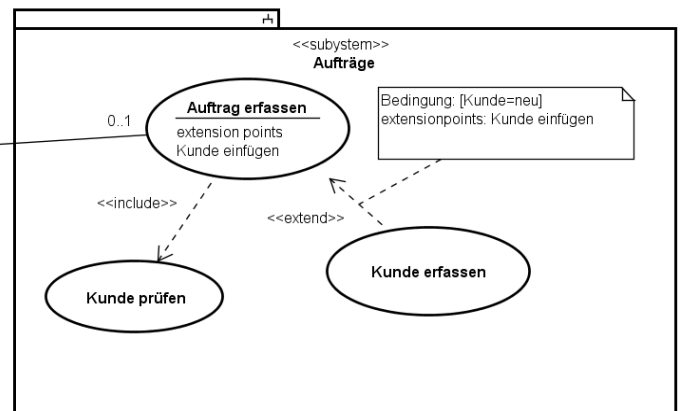
Include is a DirectedRelationship between two UseCases, indicating that the behavior of the included UseCase (the addition) is inserted into the behavior of the including UseCase (the includingCase). It is also a kind of NamedElement so that it can have a name in the context of its owning UseCase (the includingCase). The including UseCase may depend on the changes produced by executing the included UseCase. The included UseCase must be available for the behavior of the including UseCase to be completely described.

The Include relationship is intended to be used when there are common parts of the behavior of two or more UseCases. This common part is then extracted to a separate UseCase, to be included by all the base UseCases having this part in common. As the primary use of the Include relationship is for reuse of common parts, what is left in a base UseCase is usually not complete in itself but dependent on the included parts to be meaningful. This is reflected in the direction of the relationship, indicating that the base UseCase depends on the addition but not vice versa.


An Include relationship between UseCases is shown by a dashed arrow with an open arrowhead pointing from the base UseCase to the included UseCase. The arrow is labeled with the keyword «include»

## Beispiel für einen Use-Case mit include- und extend Beziehungen<sup>2</sup>

„Dieses Diagramm zeigt ein Beispiel zu einer include und extend Beziehung. Ein "Bearbeiter" erfasst einen Auftrag; er greift auf den Use Case "Auftrag erfassen" zu. Der Use Case "Auftrag erfassen" inkludiert (importiert) den Use Case "Kunden prüfen", d. h. bei jedem Erfassen eines Auftrags wird der Kunde geprüft. Inkludierte Use Cases werden beim Ausführen des inkludierenden Use Case immer ausgeführt.



Der Use Case "Auftrag erfassen" hat den extension point "Kunde einfügen". Dieser verweist über die extend-Beziehung und die in der angehängten Notiz enthaltene Information auf den Use Case "Kunden erfassen". Das bedeutet, dass der Use Case "Kunden erfassen" nur dann ausgeführt wird, wenn die Bedingung "Kunde=neu" erfüllt ist. Durch die Angabe des extension points bei der Bedingung wird deutlich gemacht an welcher Stelle des erweiterten Use Case ("Auftrag erfassen") der erweiternde Use Case ("Kunden erfassen") im Ablauf des erweiterten Use Case eingefügt wird. Die extend Beziehung drückt also eine optionale Erweiterung eines Use Case aus, die nur unter bestimmten Bedingungen ausgeführt wird.“

Arbeitsblatt Nr. 05	Q1 Technikwissenschaft: Objektorientierte Softwareentwicklung		<b>B</b> <b>S</b> <b>G</b> <b>G</b>
Datum:	Thema: Anwendungsfall-Diagramm (Use-Case-Diagramm)		
Seite 4 von 5	Name:		

## Anwendungsbereich<sup>3</sup>

„Use Case Diagramme werden zur Festlegung der Anforderungen an ein Softwaresystem (Anforderungsanalyse, Requirements Engineering) eingesetzt. Use Case Modelle sind leicht verständlich und ein gutes Kommunikationsmittel zwischen Systemanalytiker, Anwender und Entwickler. Sie legen die Grenzen des Systems fest, die Akteure, die darauf zugreifen und die Funktionalität des Systems. Die Funktionalität wird aus Sicht der zukünftigen Benutzer des geplanten Systems analysiert und definiert. Implementierungsdetails, wie z. B. die zu verwendende Programmiersprache, Systemarchitektur (Client-Server, ....), usw. sind in diesen Phasen noch nicht wichtig und deshalb zu vernachlässigen.

In der Systemanalyse wird, z. B. in Meetings zwischen Systemanalytikern und Anwendern des künftigen Systems definiert was das System aus der Sicht der Anwender leisten soll. Die einzelnen Fälle, die das System abdecken soll, werden als Use Case in die Diagramme eingezeichnet und beschrieben.

Ein Use Case Diagramm stellt also eine grobe Skizze des Systems dar, das den Zweck des geplanten Systems angibt, seine Grenzen und Schnittstellen, die auf es einwirken. Alle modellierten Elemente sind in einer Spezifikation zu beschreiben. Die Spezifikation sollte das Verhalten im Normalfall, mögliche alternativen Abläufe und das Verhalten im Fehlerfall enthalten.“

## Hinweise für die Praxis<sup>4</sup>


- **Identifikation der Akteure**  
In der Anforderungsanalyse ist es nötig die Akteure zu identifizieren und zu benennen.
- **Identifikation der Systemgrenzen**  
Die Use Cases sind Teil des Systems und enthalten die Funktionalität des zukünftigen Systems. Akteure sind Elemente, die nicht zu dem System gehören und auf das System zugreifen. Das System grenzt die Elemente, die zum System gehören von Elementen, die nicht zu dem System gehören ab. Alle Elemente innerhalb des Systems gehören zu dem System und sind zu entwickeln. Elemente außerhalb des Systems greifen auf das System zu.
- **Identifikation der Use Cases**  
Use Cases beschreiben den Zweck, für den ein System entwickelt wird auf einer hohen Abstraktionsebene. Sie beschreiben die Leistung, die ein System für den Benutzer erbringt.
- **Identifikation der Beziehungen zwischen Akteuren und Use Cases**  
Zur Identifikation der Beziehungen hilft die Beantwortung folgender Fragen:
  - Auf welche Use Cases greift ein Akteur zu?
  - Welche Antworten erhält er vom System?
  - Über welche externen Ereignisse muss ein Use Case informiert werden?

Der Text auf dieser Seite ist der angegebenen Quelle in teilweise gekürzter Form entnommen. Daher ggf. an der Quelle nachlesen!

Weitere Informationen u.a. bei: <https://de.wikipedia.org/wiki/Anwendungsfalldiagramm>

<sup>3</sup> Entnommen von <https://www.fbi.h-da.de/labore/case/uml/anwendungsfalldiagramm.html> abgerufen am 19. August 2018

<sup>4</sup> Entnommen von <https://www.fbi.h-da.de/labore/case/uml/anwendungsfalldiagramm.html> abgerufen am 19. August 2018

Arbeitsblatt Nr. 05	Q1 Technikwissenschaft: Objektorientierte Softwareentwicklung	 B S G G
Datum:	Thema: Anwendungsfall-Diagramm (Use-Case-Diagramm)	
Seite 5 von 5	Name:	

## Generalisierung von Akteuren und Anwendungsfälle

Sowohl Akteure als auch Anwendungsfälle können durch Vererbungsstrukturen beschrieben werden. Dazu aber später im Rahmen des Themas „Vererbung“ mehr.

### Übungen

Zeichnen Sie zu den folgenden Beschreibungen ein Anwendungsfalldiagramm:

#### Beschreibung 1

In einem Kino kann ein Kinobesucher Kinokarten an der Kasse kaufen. Die Karte kann auch vorbestellt sein. Außerdem ist es möglich, beim Kauf der Kinokarte Popcorn und Getränke zu bestellen. Danach bezahlt der Kunde die Rechnung. Es ist auch möglich mit Kreditkarte zu bezahlen, welche bei Bedarf einer automatischen Prüfung unterzogen werden kann.

#### Beschreibung 2

Der Benutzer einer Bildverwaltungssoftware soll mit dieser Bilder ansehen können. Weiterhin soll der Benutzer Bilder hinzufügen und suchen können. Beim Hinzufügen von Bildern kann der Benutzer, wenn gewünscht, Suchbegriffe eingeben. Auch das Bearbeiten von Bildern ist vorgesehen. Der Benutzer kann auch Bilderalben erstellen und diesen Bilderalben bereits vorhandene Bilder hinzufügen. Ein Bilderalbum wird immer mit einem Namen versehen.

#### Beschreibung 3

Für einen Wertstoffhof soll eine Verwaltungssoftware erstellt werden. Folgende Geschäftsprozesse wurden hierbei ermittelt:

Ein Mitarbeiter des Wertstoffhofes betreut die Anlieferung, Verarbeitung und Auslieferung von Wertstoffen.

Bei der Anlieferung werden die Wertstoffe immer gewogen. Je nachdem, ob die Materialien sortenrein oder gemischt angeliefert werden, sind sie ggf. zu sortieren. Anschließend werden die Wertstoffe zur Verarbeitung gepresst. Gläser und Kunststoffe werden zusätzlich geschreddert. Abschließend erfolgt die Auslieferung der Wertstoffe an die Deponie, die Verbrennungsanlage oder private Recycling-Unternehmen. Bei jeder Anlieferung wird durch den Mitarbeiter der Kontostand des Buchungskontos neu gesetzt.

Die Abrechnung einer Anlieferungsgebühr bzw. einer Gutschrift erfolgt ebenfalls mit Hilfe der Verwaltungssoftware.

Jeder Kunde wird im System durch einen Verwaltungsmitarbeiter erfasst. Ein Verwaltungsmitarbeiter kann im System den Kontostand eines Kunden ermittelt und eine Rechnung bzw. Gutschrift erstellen.

#### Beschreibung 4

Anhand der Beschreibung eines Adressbuches auf dem Arbeitsblatt „04 Objekte und Klassen (1)“ ist ein Anwendungsfalldiagramm für den Benutzer der Software zu zeichnen.