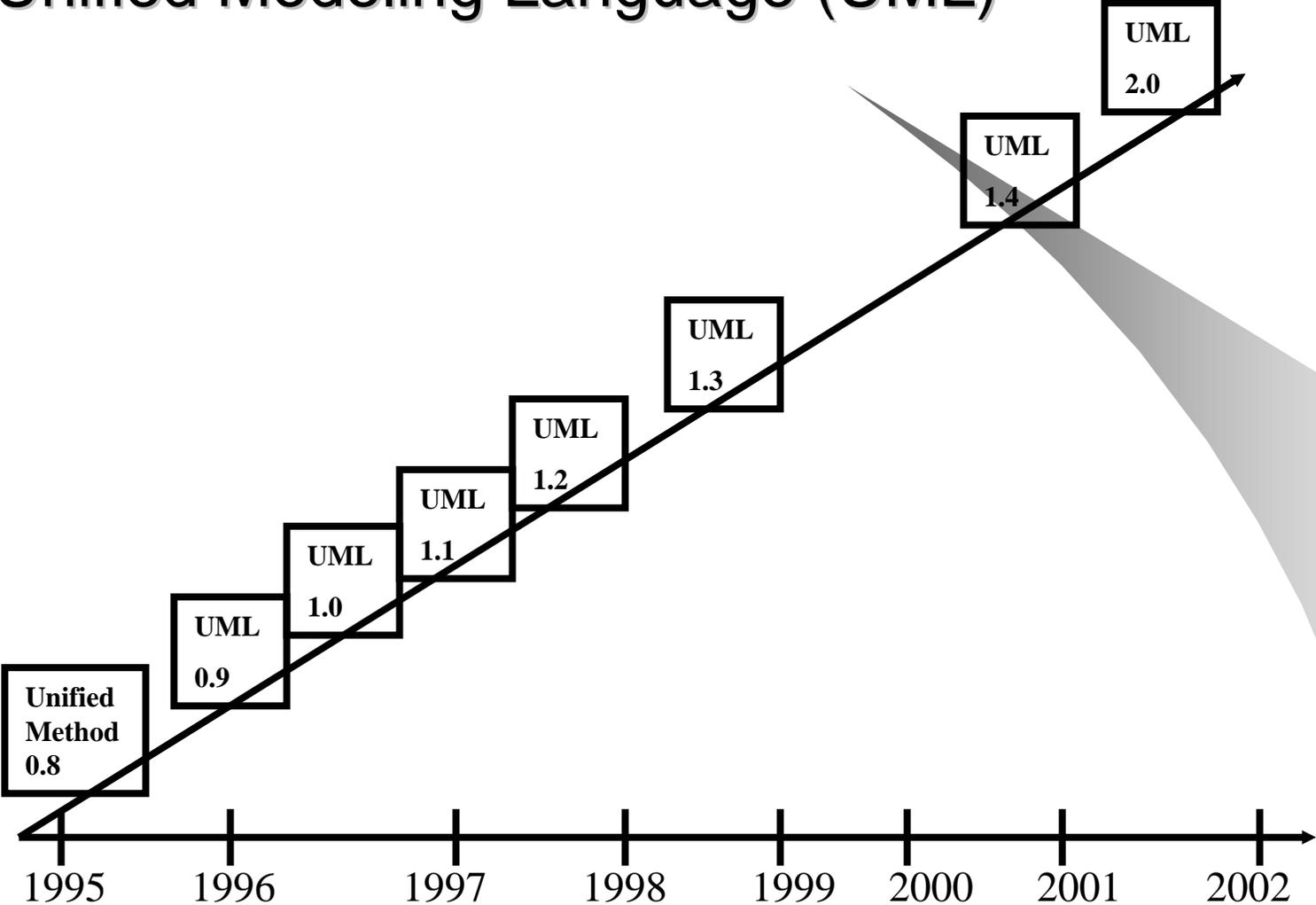


Literatur

- Martin Fowler and Kendall Scott: *UML Distilled: Applying the Standard Object Modeling Language*. Addison-Wesley 1997.
- James Rumbaugh, Ivar Jacobson, and Grady Booch: *The Unified Language Reference Manual*. Addison-Wesley 1999.
- Ivar Jacobson, Grady Booch, and James Rumbaugh: *The Unified Software Development Process*. Addison-Wesley 1999.

Unified Modeling Language (UML)

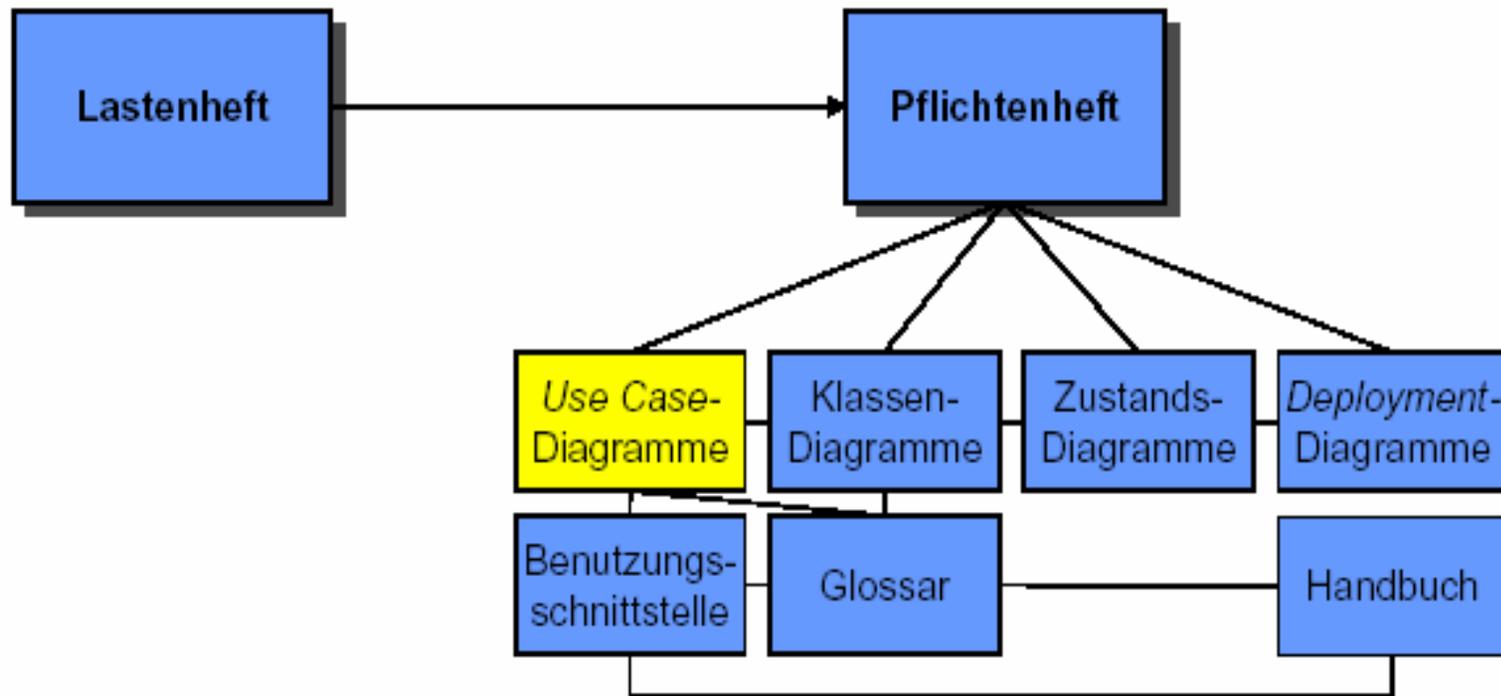


UML

Im Oktober 1994 haben sich Grady Booch und Jim Rumbaugh bei der Rational Software Corporation zusammengeschlossen, um ihre erfolgreichen Methoden zu einem einheitlichen Industriestandard weiterzuentwickeln. Es entstand zunächst der Vorgänger der *Unified Modeling Language* (UML).

Systemanalyse in UML

„Use Case Driven Requirements Engineering“



Anwendungsfall-Modelle

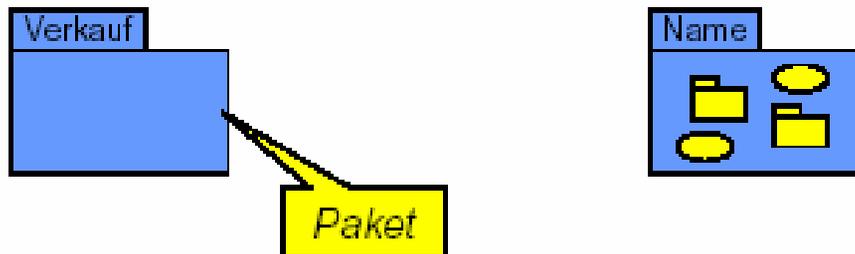
Anforderungen werden in UML durch ein **Anwendungsfallmodell** beschrieben.

Englischer Begriff: *Use Case Model*

Ein Anwendungsfallmodell besteht aus

- einzelnen Anwendungsfall-Diagrammen
- weiteren (evtl. wieder geschachtelten) **Paketen** aus Anwendungsfalldiagrammen

Das oberste Paket der geschachtelten Pakete ist das Anwendungsfallmodell.



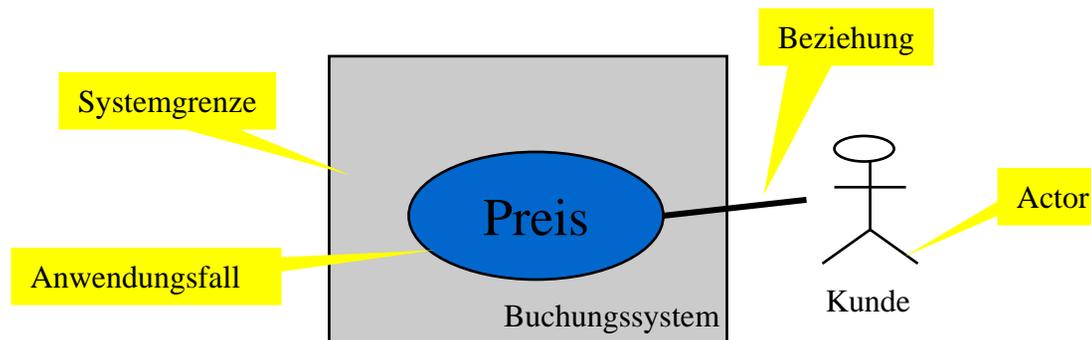
Anwendungsfall-Diagramme

Ein **Anwendungsfall-Diagramm** zeigt die Beziehungen zwischen Aktoren und Anwendungsfällen in einem System.

Englischer Begriff: *Use Case-Diagram*

Ein Anwendungsfall-Diagramm für ein System besteht aus

- der Systemgrenze
- Aktoren
- Anwendungsfällen
- Beziehungen zwischen Aktoren und Anwendungsfällen

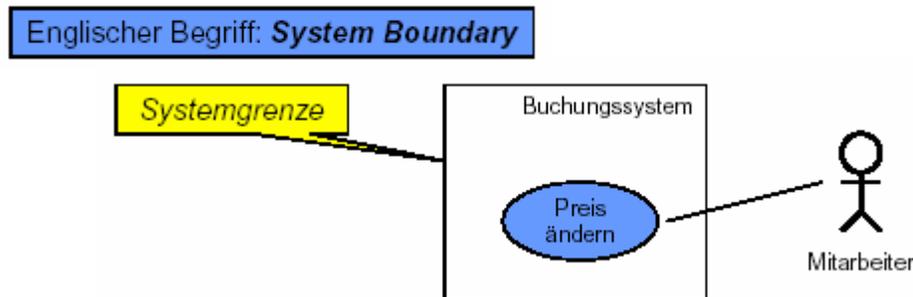


Systemgrenzen

Die Anwendungsfallsicht beschreibt die Funktionalität eines Systems aus einer Perspektive von *außerhalb*, also aus *Benutzersicht*.

Es interessieren nur die Interaktionen zwischen Aktoren (Benutzern) und System, nicht aber, was *innerhalb* des Systems geschieht.

Diese Grenze wird durch die **Systemgrenze** verdeutlicht.



Beobachtung: In weiteren Anwendungsfällen kann ein System wieder als Aktor erscheinen, z.B. ein Buchungssystem (eines Reisebüros) in Bezug auf ein anderes System (Flugreservierungssystem).

Aktoren

Ein **Aktor** ist eine Abstraktion für ein Objekt außerhalb des Systems, das mit dem System interagiert, z.B ein Benutzer.

Möglich dabei sind:

- mehrere Rollen für einen Benutzer
- mehrere Benutzer mit der gleichen Rolle
- Systeme als Aktoren

Synonym: Akteur

Englischer Begriff: **Actor**

Eigenschaften eines Aktors sind:

Name: Substantiv

Beschreibung:

- textuelle Beschreibung seiner Rolle, Tätigkeiten, Fähigkeiten, Protokolle, ...
- alle Beschreibungen zusammen bilden den **Aktoren-Katalog**



Anwendungsfälle

Ein **Anwendungsfall** ist die Spezifikation einer Folge von Interaktionen zwischen einem Akteur und dem zu untersuchenden Computersystem.

Anwendungsfälle beschreiben eine für den *Benutzer sichtbare* Funktionalität.

Eine **Instanz** eines Anwendungsfalls (Instantiierung) ist die Ausführung der beschriebenen Interaktionen.

Ein Anwendungsfall definiert:

- **Name:** Verb (o. Substantiv), Name des Vorgangs
- **Priorität:** Priorität im Entwicklungsprozeß
- **Beschreibung:**

textuelle Beschreibung der Interaktion zwischen Benutzer und System, unter Bezugnahme auf das Lastenheft

Auflistung der einzelnen Schritte unter Nutzung der Begriffe des .

Glossars

alle Beschreibungen zusammen bilden den **Anwendungsfall-Katalog**

Anwendungsfälle

Beschreibung eines Anwendungsfalls im Anwendungsfall-Katalog:

Beispiel: Preis ändern

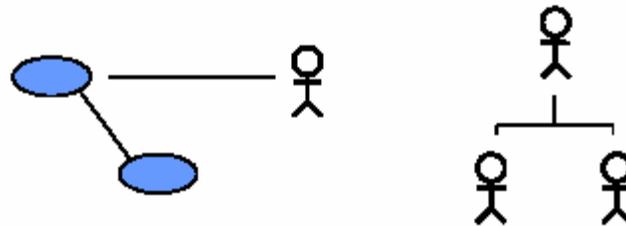
1. Der Mitarbeiter sucht das gewünschte Produkt.
2. Das System zeigt alle Produktdaten an.
3. Der Mitarbeiter trägt den neuen Gesamtpreis ein.
4. Das System prüft, ob der Preis innerhalb der zulässigen Grenzen liegt.
5. Das System prüft, ob aktuelle Aufträge betroffen sind und fragt, ob sie den alten oder neuen Preis erhalten sollen. Der Mitarbeiter kann die Änderung hier rückgängig machen.
6. Der neue Preis wird übernommen.



Beziehungen

Beziehungen zwischen Anwendungsfällen und Aktoren

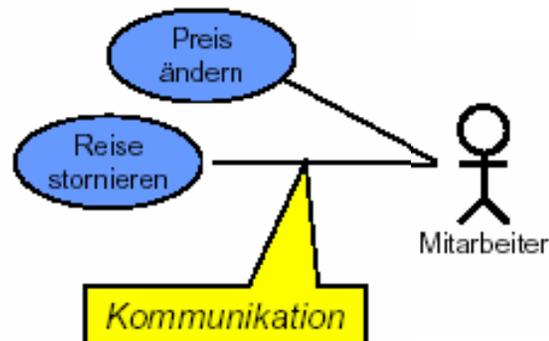
- werden durch gerichtete oder ungerichtete Pfeile mit unterschiedlichen Pfeilspitzen notiert,
- können durch **Stereotypen** annotiert werden; Schreibweise: « stereotyp »,
- können zwischen zwei Anwendungsfällen bestehen,
- können zwischen zwei Aktoren bestehen oder
- können zwischen einem Anwendungsfall und einem Aktor bestehen.



Kommunikation

Die **Kommunikation** zwischen einem Aktor und dem System wird durch eine ungerichtete Kante zwischen Aktor und Anwendungsfall dargestellt. Ein Aktor kann mit mehreren Anwendungsfällen assoziiert sein und umgekehrt.

Englischer Begriff: *Actor-Use case Communication*



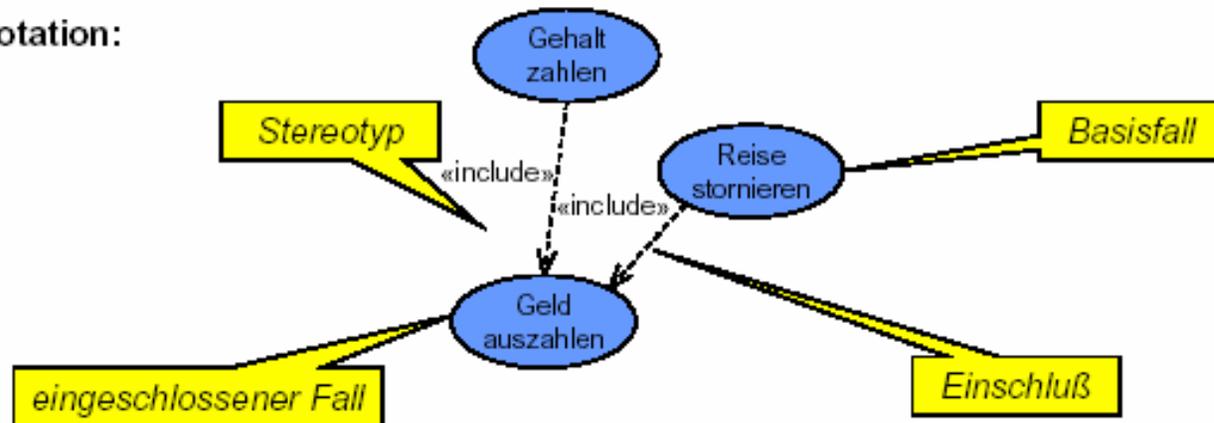
Mögliche Bedeutungen sind: „A ist beteiligt in/an B“, „A löst B aus“, „A kommuniziert mit B“, „A nutzt B“, ...

Einschluß von Anwendungsfällen

Einschluß bedeutet das Aufnehmen bestimmter Aktionsfolgen in die eines Basisfalls. Der eingeschlossene Fall kann unabhängig vom Basisfall und von verschiedenen Basisfällen genutzt werden.

Englischer Begriff: *Include*

Notation:



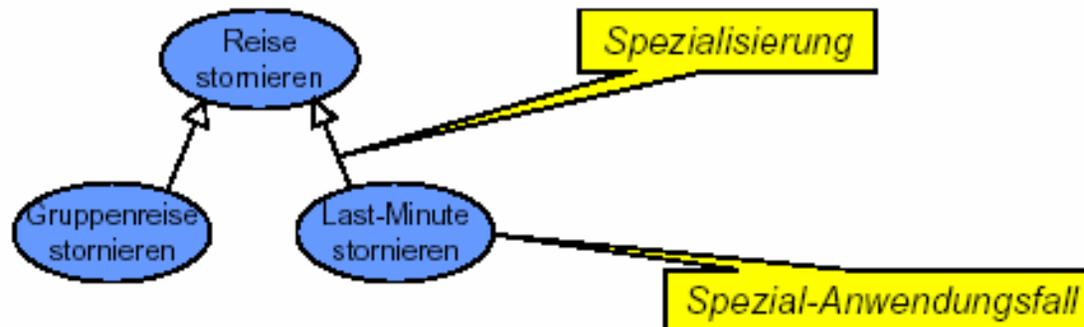
Generalisierung von Anwendungsfällen

Generalisierung setzt spezialisierte Anwendungsfälle in (taxonomische) Beziehung zu allgemeineren. Der Spezialfall erbt **alle** Eigenschaften des allgemeinen und fügt inkrementell weitere hinzu oder **ersetzt** sie.

Englischer Begriff: *Generalization*

Speziellere Anwendungsfälle können allgemeinere substituieren.

Notation:

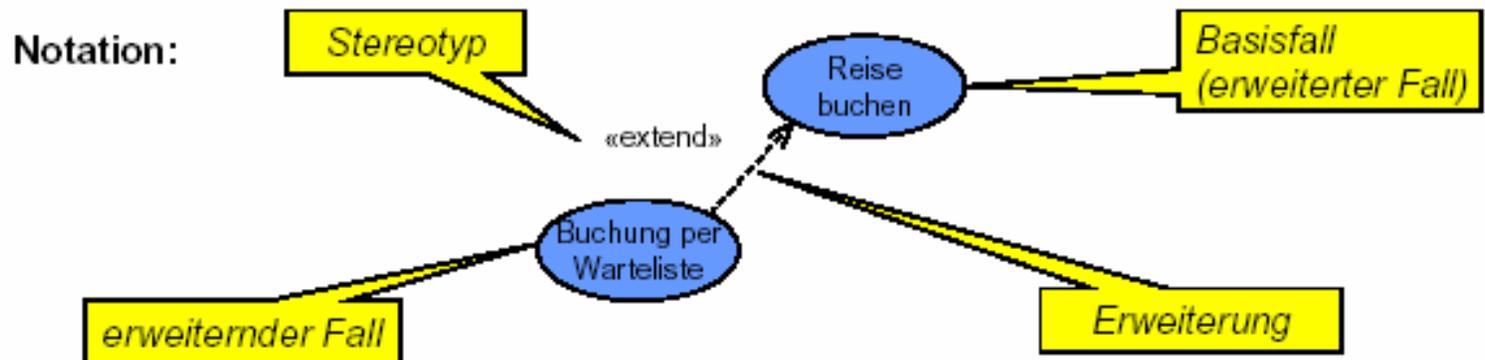


Erweiterung von Anwendungsfällen

Erweiterungen dienen zur Definition von Variationen und Sonderfällen mit der Bedeutung, daß der erweiternde Fall den Basisfall durch entsprechende Aktionen ergänzt.

Der erweiternde Fall kann nicht alleine ausgeführt werden.
Der unerweiterte Basis-Fall ist nicht mehr verfügbar.

Erweiterungen werden an sog. Erweiterungspunkten ggf. konditional eingefügt.



Erweiterung vs. Einschluß vs. Generalisierung

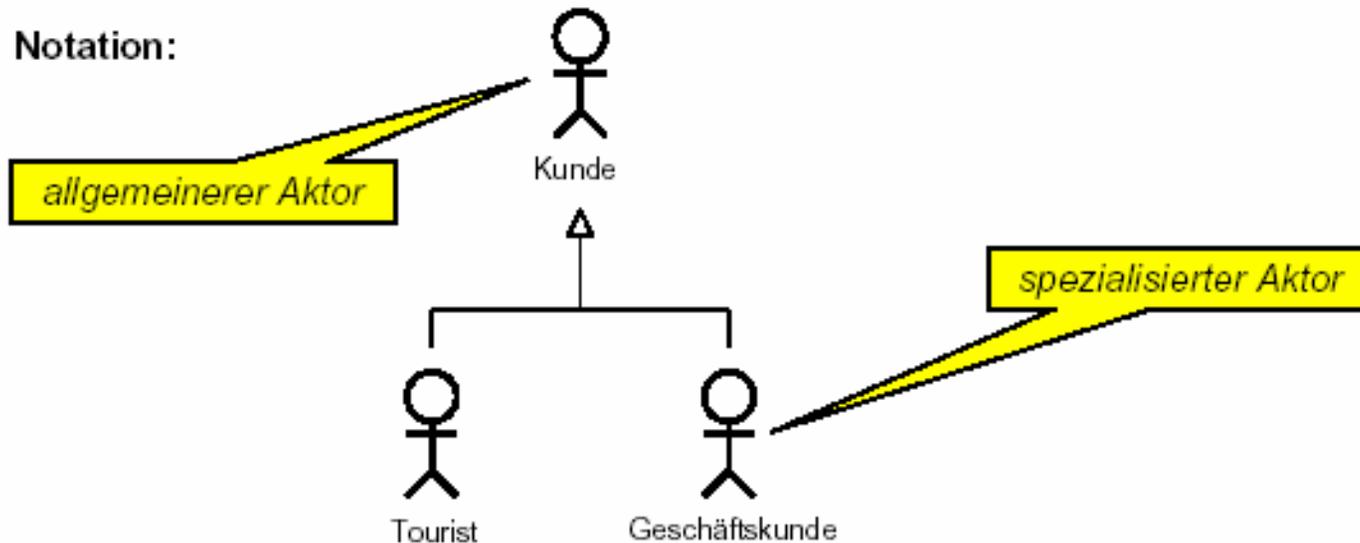
	Erweiterung <i>Extend</i>	Einschluß <i>Include</i>	Generalisierung <i>Generalization</i>
Basisfall geändert?	implizit	explizit	unverändert
Instantiierbar?	kann	kann	kann (kann abstrakt sein)
Sieht d. Basisfall die Ergänzung?	nein	ja	nein
eigene Aktoren möglich?	nein	nein	ja

Generalisierung von Aktoren

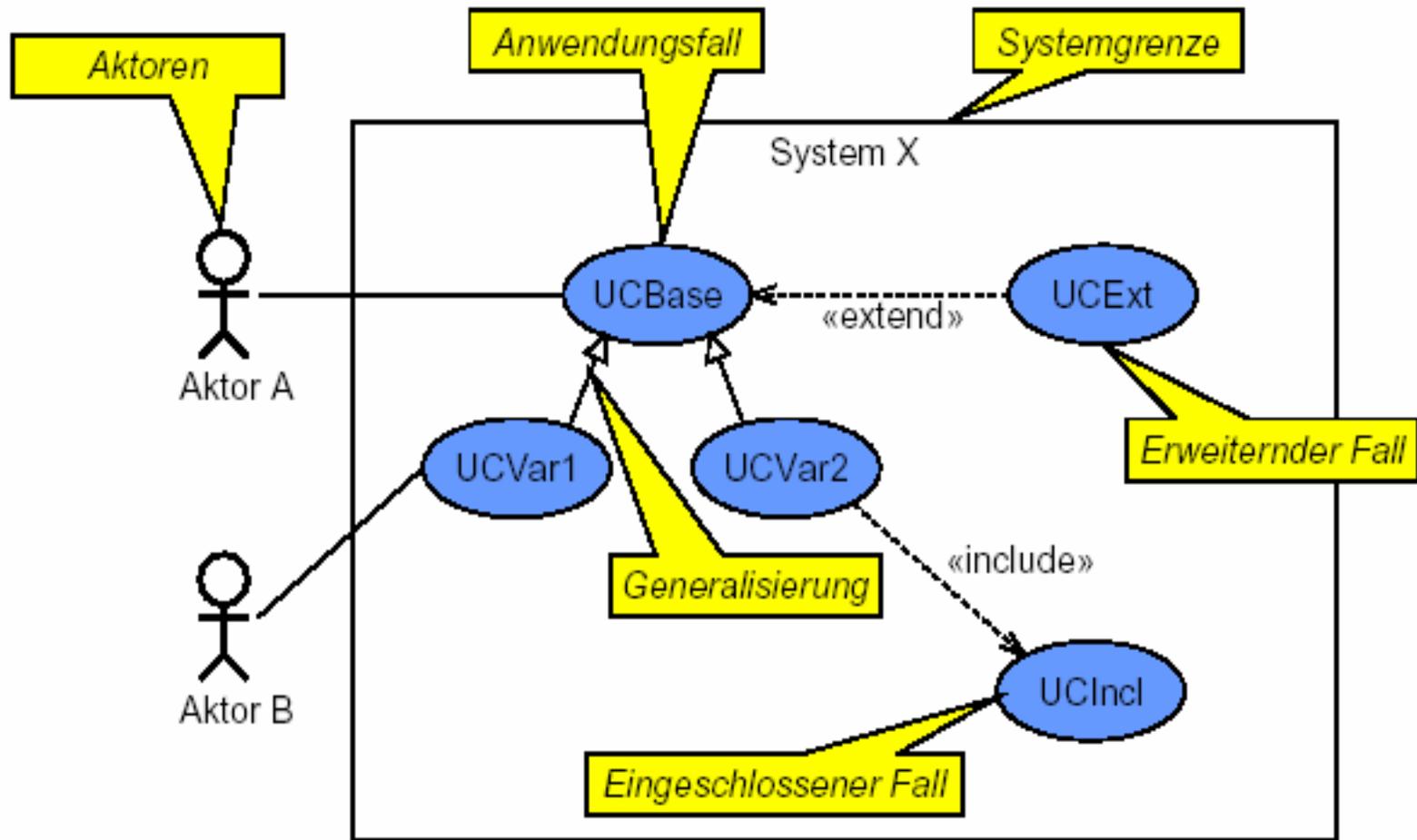
Aktoren können in einer **Taxonomie** zueinander stehen.

Englischer Begriff: *Generalization*

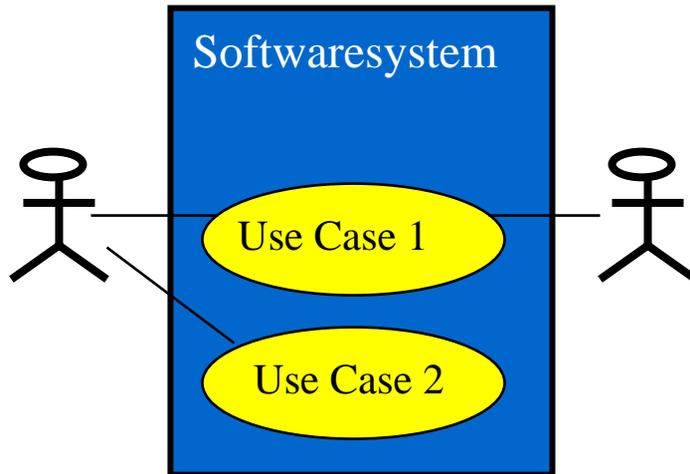
In Anwendungsfalldiagrammen üblicherweise nicht notiert, sondern in separaten Aktoren-Diagrammen.



Zusammenfassung der Konzepte



Zusammenfassung der Begriffe



Use Case-Diagramm

- Ein Use Case beschreibt die Funktionalität des Softwaresystems
- Use Cases sollen es ermöglichen, mit dem zukünftigen Benutzer über die Funktionalität des Softwaresystems zu sprechen, ohne sich gleich in Details zu verlieren

Zusammenfassung der Begriffe

Akteur (*actor*)

- Akteur ist eine Rolle, die ein Benutzer des Softwaresystems spielt
- Akteure können Menschen oder andere automatisierte Systeme sein
- Sie befinden sich stets außerhalb des Systems.

Use Case-Diagramm (*use case diagram*)

- Gibt auf hohem Abstraktionsniveau einen guten Überblick über das Softwaresystem und seine Schnittstellen
- Die Akteure werden als Strichmännchen eingetragen
- Die Use Cases als Ovale

Zusammenfassung der Begriffe

Extend-Beziehung (*extend relationship*)

Mit Hilfe der extend-Beziehung wird ein Use Case A durch einen Use Case B erweitert.

Der Use Case A beschreibt die Basisfunktionalität, der Use Case B spezifiziert Erweiterungen.

Include-Beziehung (*include relationship*)

Die include-Beziehung ermöglicht es, dass die gemeinsame Funktionalität von Use Case A und B durch einen Use Case C beschrieben wird. Der Use Case C kann niemals allein ausgeführt werden, sondern immer nur als Bestandteil von A oder B.