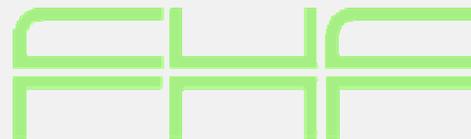


Verhaltensdiagramme

3.2 Use-Case-Diagramm

3.3 Aktivitätsdiagramm

3.4 Zustandsautomat



Prof. Mario Jeckle

Fachhochschule Furtwangen

mario@jeckle.de

<http://www.jeckle.de>

Use-Case-Diagramm

Dient

- Darstellung von externem Verhalten des Systems
- Darstellung aus Nutzersicht
- Darstellung von Reaktion des Systems auf Ereignisse in der Umwelt

Antwort auf die Frage:

Was soll das System leisten?

Use-Case-Analyse

Analysergebnis kann textuell oder graphisch dargestellt werden.

UML Use-Case-Diagramm zeigt extern sichtbares Verhalten von:

- Elementen
- Klassen
- Schnittstellen
- Knoten
- etc.

Use Case

Kapselt in sich geschlossene Sammlung von Aktionen mit spezifischer Reihenfolge:

- Auslöser/Initiator
- Schritte
- Sonderfälle
- Akteure
- Ergebnis

OHNE

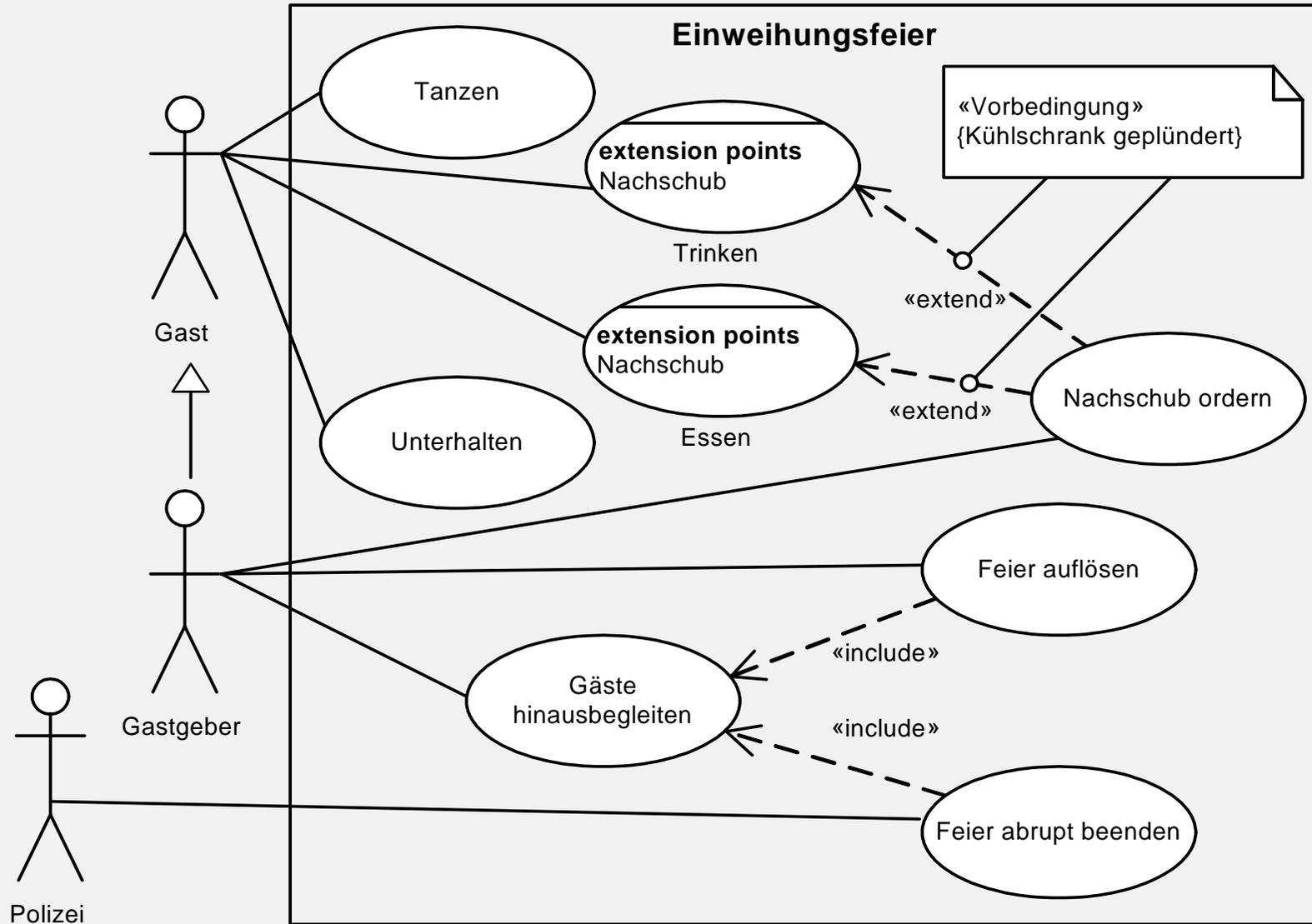
- Klassen
- Operationen

Use-Case-Diagramm

Vorteile:

- Übersichtlichkeit, da wenige Notationselemente
- Leicht verständlich auch für Laien
- Überblick über das System und Einbettung in den Kontext
- Klare Systemabgrenzung

Use-Case-Diagramm – Ein Beispiel



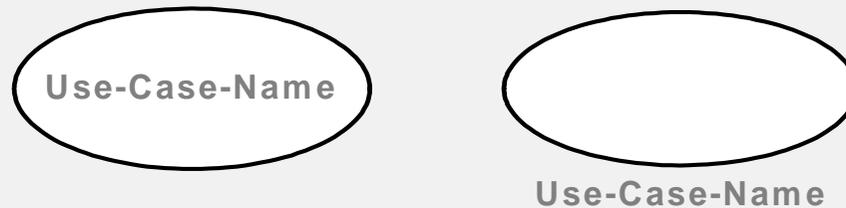
Notationselemente

Das Use-Case-Diagramm verfügt über folgende Notationselemente:

- Use Case
- System
- Akteur
- include-Beziehung
- extend-Beziehung

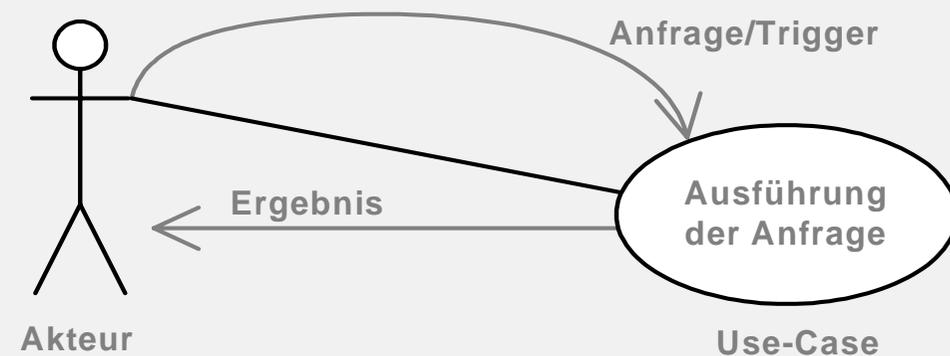
Use Case = Anwendungsfall

Darstellung durch eine Ellipse, die entweder den Use-Case-Namen enthält oder unter der der Name steht



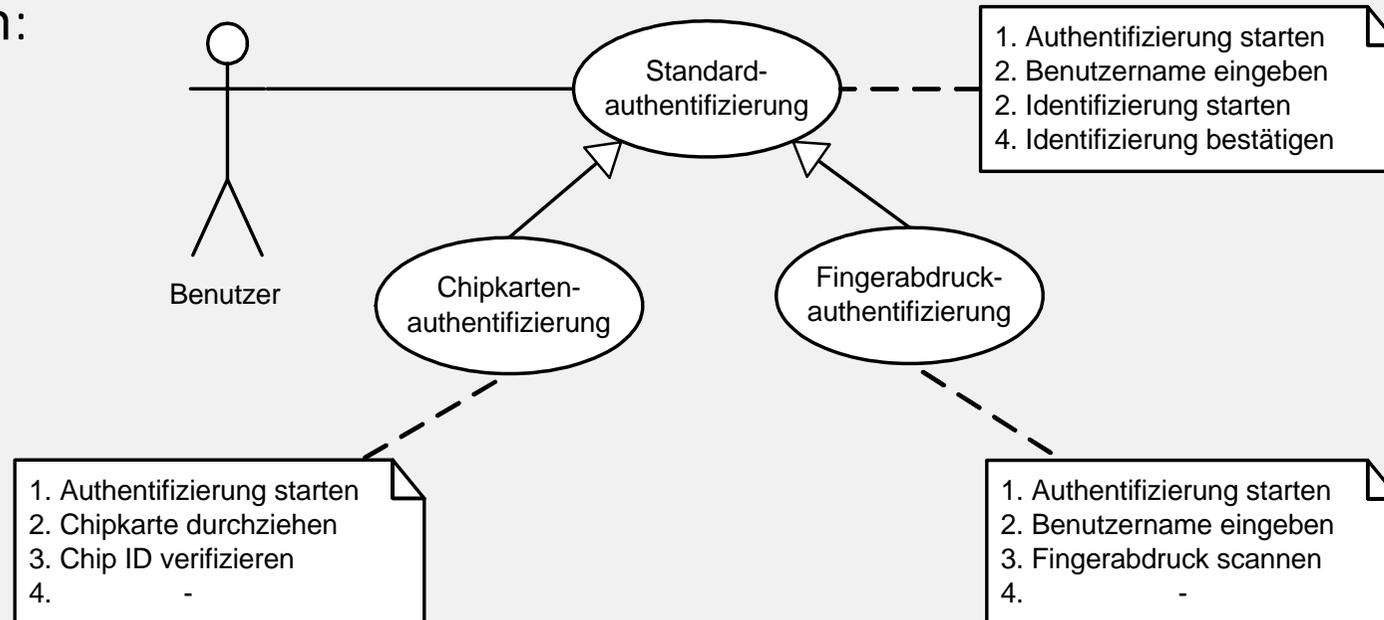
Use Case

- Beschreibt eine Reihe von Aktionen, die nacheinander ein Verhalten formen
- Wird immer von einem Akteur ausgelöst
- Hat immer ein Ergebnis
- Kann gleichzeitig mehrfach instanziiert werden
- Spiegelt funktionales Verhalten wieder – interne Abläufe sind irrelevant



Use Case

Mehrere Use Cases können auch in Beziehung gesetzt werden:



- Aus zwei Abläufen wird einer modelliert
- Generalisierung und Spezialisierung sind möglich, wenn im Basis Use Case allgemeines Verhalten definiert wird

System

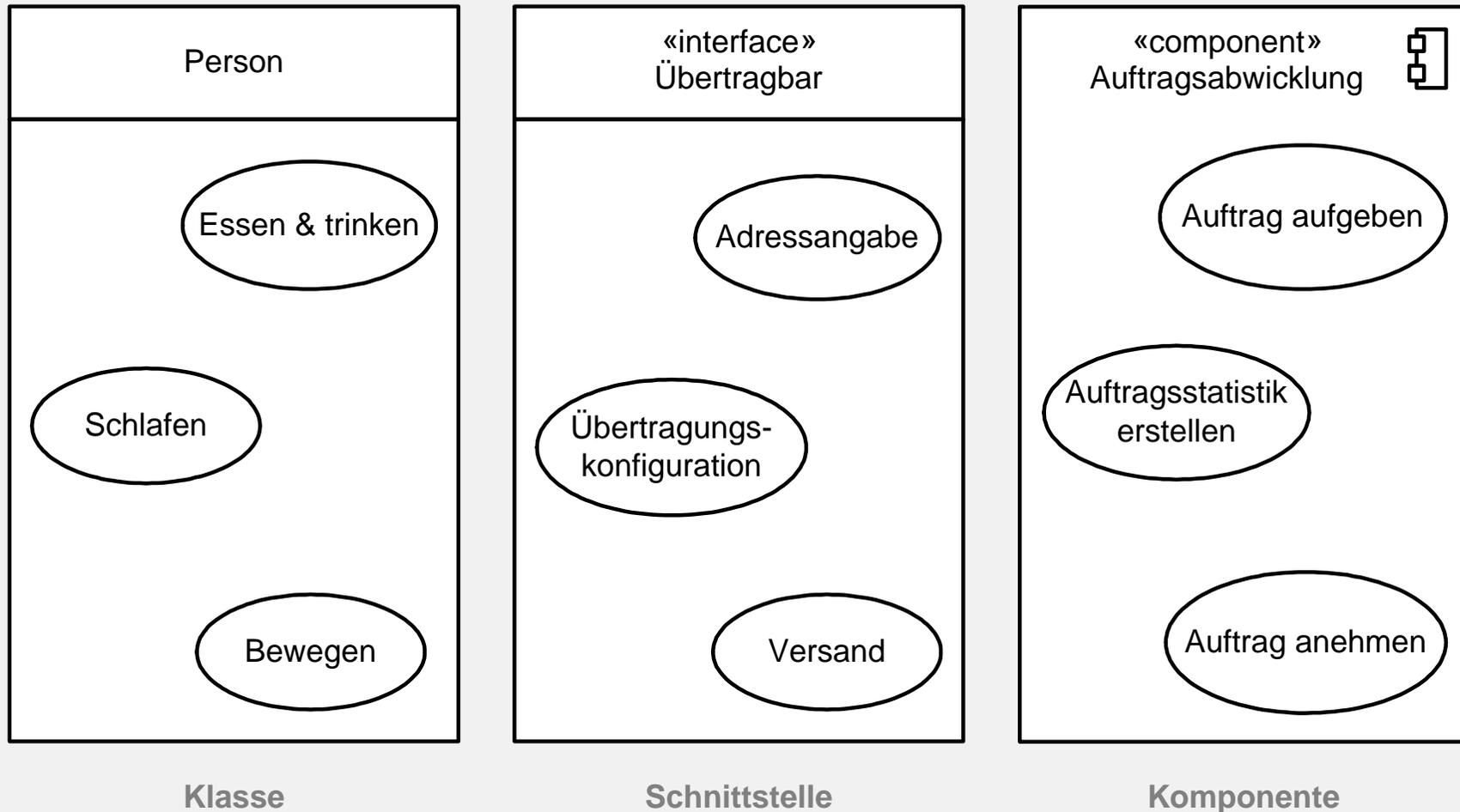
Darstellung erfolgt durch ein Rechteck, dessen Kanten die Systemgrenzen darstellen



System

- Realisiert und etabliert das im Use Case beschriebene Verhalten
- Kann Klasse, Schnittstelle, Komponente, Subsystem sein
- Ist im Use-Case-Diagramm nicht zwingend nötig

System



Akteur

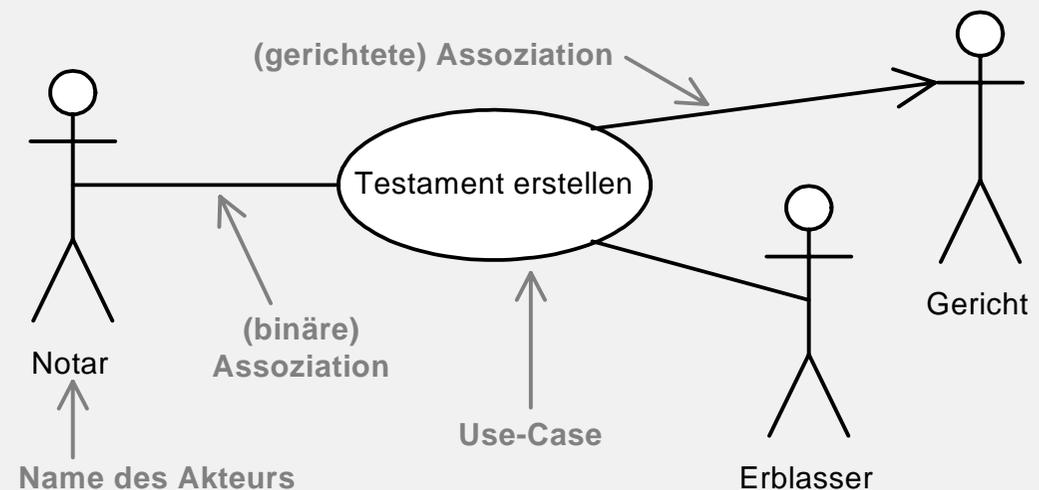
Darstellung als „Strichmännchen“



Name des Akteurs

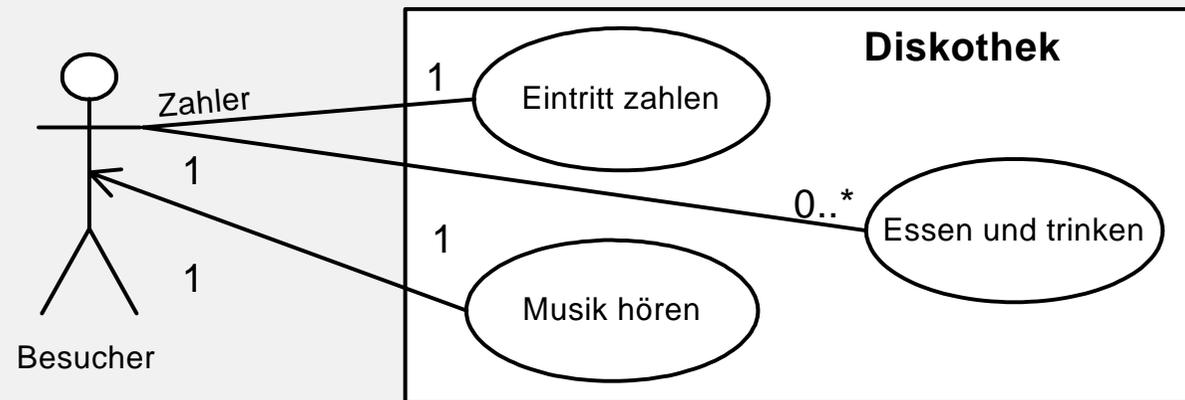
Akteur

- Repräsentiert die Rolle, die von außen mit dem System interagiert
- Steht in ständigem Signal- und Datenaustausch mit Use Case
- Beteiligung des Akteurs an Use Case wird durch binäre Assoziation dargestellt



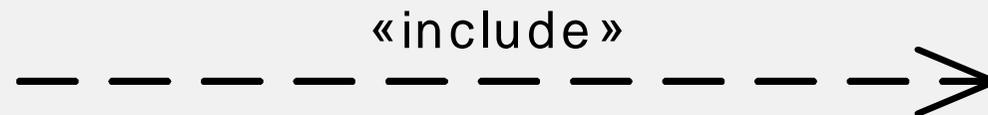
System

Richtungsfunktion der Assoziation, Rollen und Multiplizitäten sind möglich



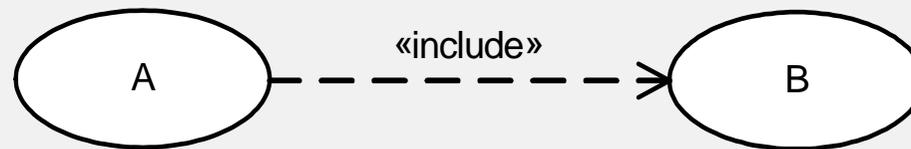
include-Beziehung

Darstellung durch Abhängigkeitsbeziehung mit Stereotyp `include` von abhängigem, importiertem Use Case zu unabhängigem, inkludiertem Use Case

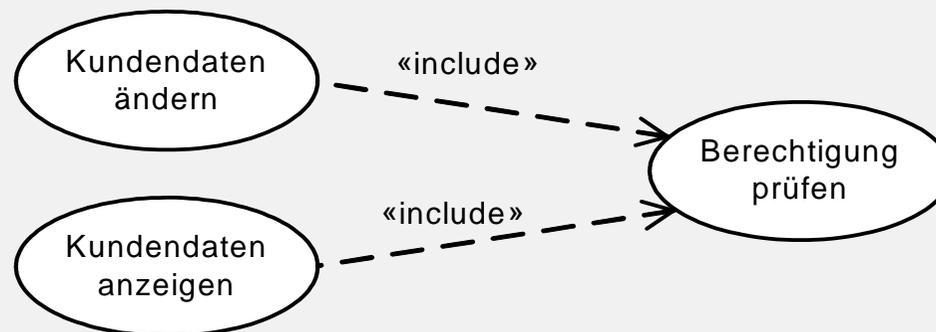


include-Beziehung

- Zeigt, daß ein Use Case das Verhalten eines anderen importiert

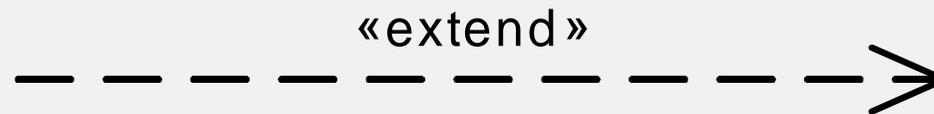


- Use Case darf durch mehrere andere Use Cases inkludiert werden
- Verhalten wird immer inkludiert – nicht optional



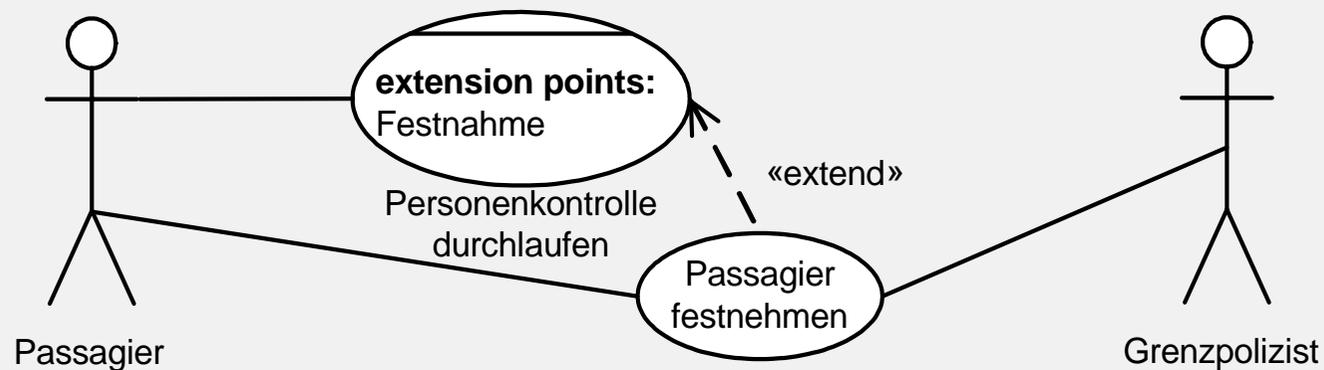
extend-Beziehung

Darstellung durch eine Abhängigkeitsbeziehung mit Stereotyp `extend` vom erweiternden Use Case zu erweitertem Use Case



extend-Beziehung

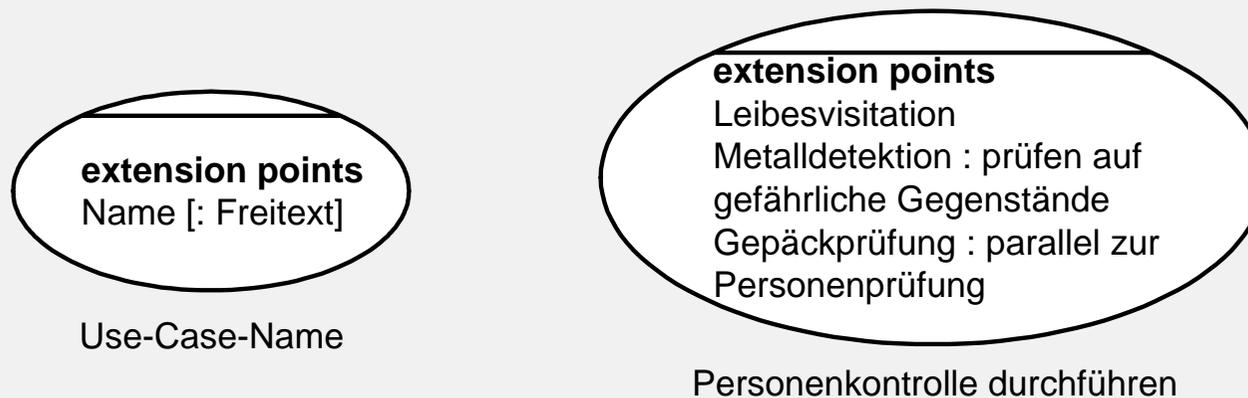
Zeigt, daß Verhalten eines Use Cases durch anderen Use Case erweitert werden kann aber nicht muss



extend-Beziehung

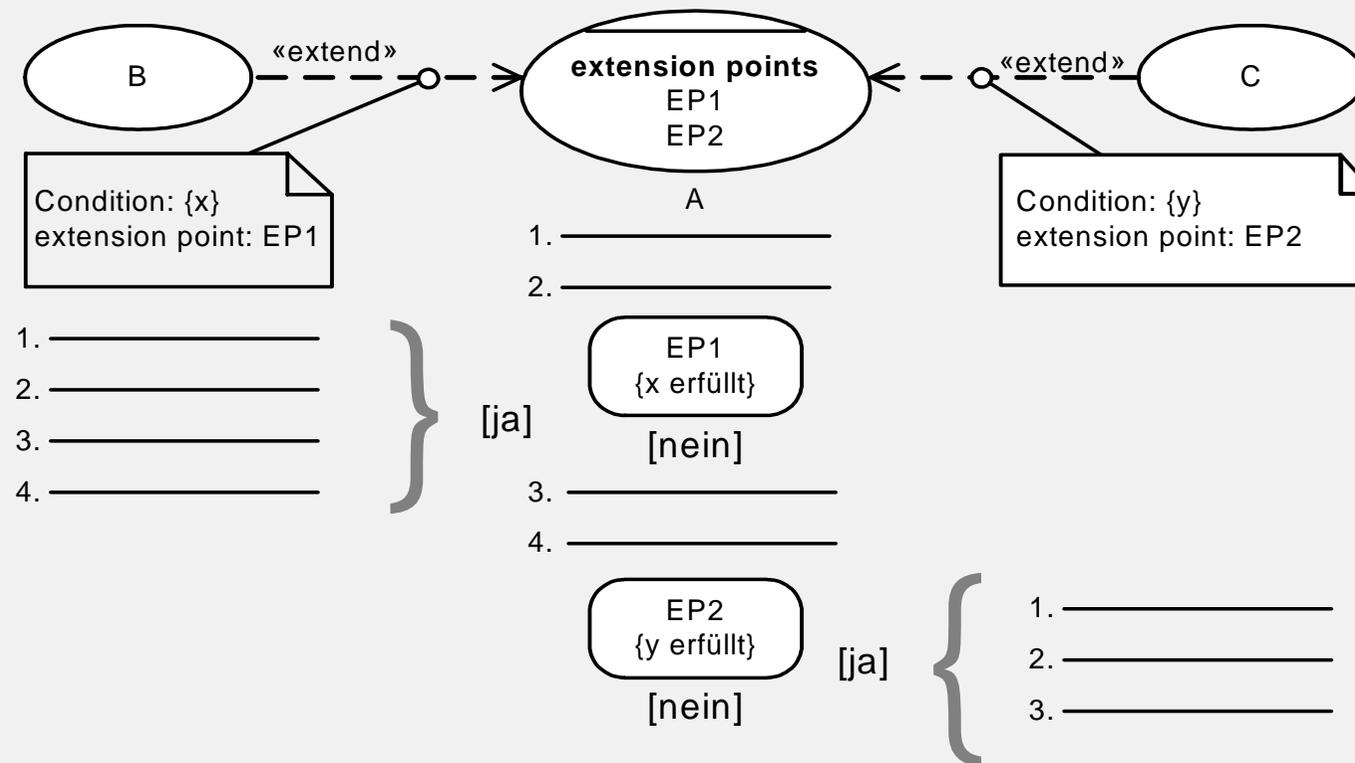
Erweiterungspunkt:

- hier kann Verhalten eines Use Case erweitert werden
- Use Case kann mehrere Erweiterungspunkte haben
- Darstellung durch eigenen, durch „extension Point“ bezeichneten Abschnitt im Use Case



extend-Beziehung

Die Auswertung einer extend-Beziehung:



3.3 Das Aktivitätsdiagramm

Aktivitätsdiagramm

Dient Darstellung von Abläufen im System inklusive Nebenläufigkeiten oder alternativen Entscheidungswegen

Antwort auf die Frage:

Wie realisiert das System bestimmtes Verhalten?

Aktivitätsdiagramm

Umfasst

- Rahmen und Regeln für Verhaltensabläufe
- Start- und Endpunkte
- Verzweigungen
- Bedingungen

Aktivitätsdiagramm

Modellieren von Aktivitäten:

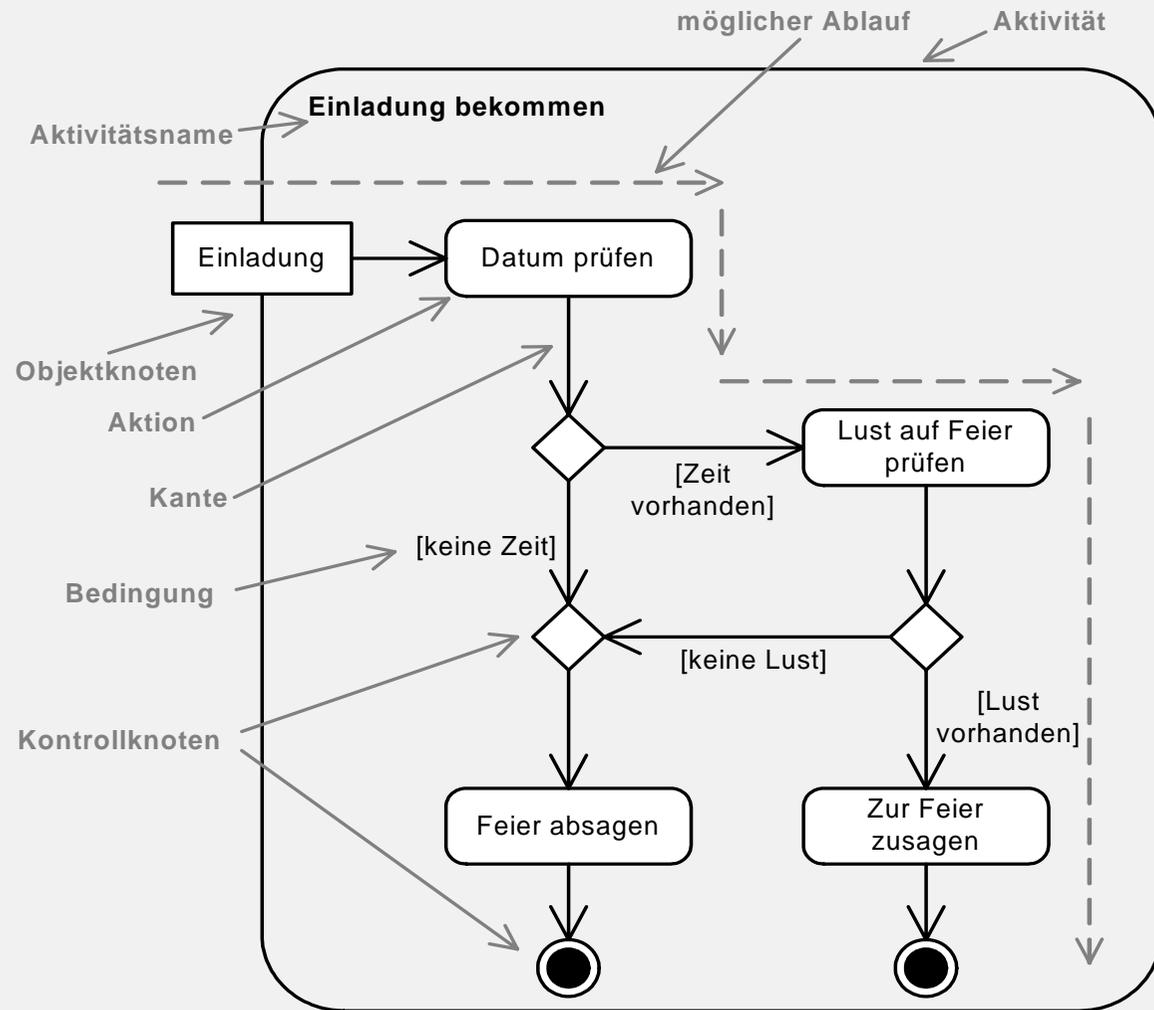
- Aktivität = Menge an potenziellen Abläufen zur Laufzeit, selten trivial
- Abläufe = z.B. Geschäftsprozesse, Methoden von Klassen, Visualisieren von Systemprozessen

Aktivitätsdiagramm - Elemente

Die Elemente sind

- Eine nichtleere Menge von Aktivitäten
- Aktionen; Einzelschritt der durchlaufen wird
- Objektknoten; repräsentiert beteiligte Daten oder Gegenstände
- Kontrollelemente zur Ablaufsteuerung zwischen Aktionen oder Objektknoten
- Verbindende Kanten

Aktivitätsdiagramm



Das Tokenkonzept

- Dient der verbesserten Darstellung nebenläufiger Abläufe
- Token =
 - „Marke“, legt Abarbeitungsposition des Ablaufs fest
 - Tokenwanderung durch Aktivität bedeutet Abarbeitung läuft
 - in einer Abarbeitung können mehrere Tokens vorkommen
 - werden im Aktivitätsdiagramm nicht graphisch dargestellt

Verwendung Token

- Abarbeiten von Aktionen
- Verzweigung und Vereinigung
- Parallelabarbeitung und Synchronisation
- Objektflüsse
- Verweildauer
- Bedingte und gewichtete Abarbeitung
- Start und Ende von Abläufen

Notationselemente

Die Notationselemente des Aktivitätsdiagramms sind zahlreich:

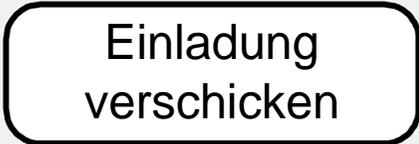
- Aktion
 - Aktivität
 - Objektknoten
 - Kanten
 - Kontrollelemente
 - Parametersatz
 - Unterbrechungsbereich
 - Exception-Handler
 - Aktivitätsbereich
- Strukturierter Knoten
 - Mengenverarbeitungsbereich
 - Schleifenknoten
 - Entscheidungsknoten

Aktion

Darstellung durch ein Rechteck mit abgerundeten Ecken mit Namen oder Text zur Operation



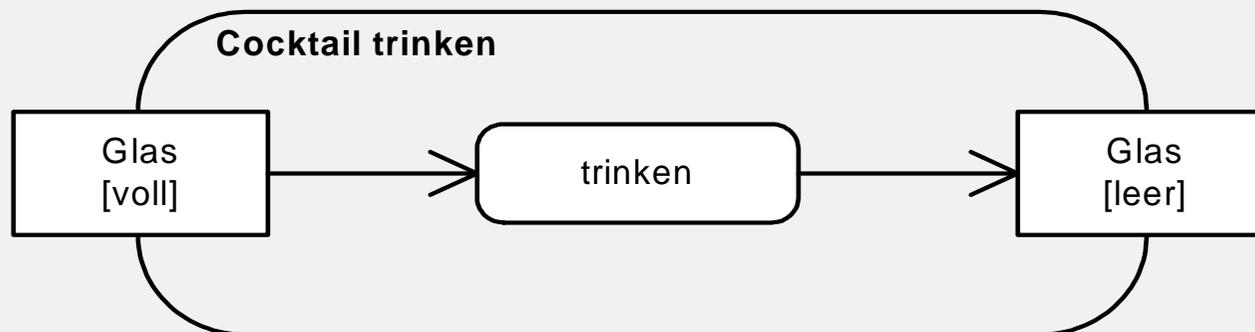
Aktionsname



Einladung
verschicken

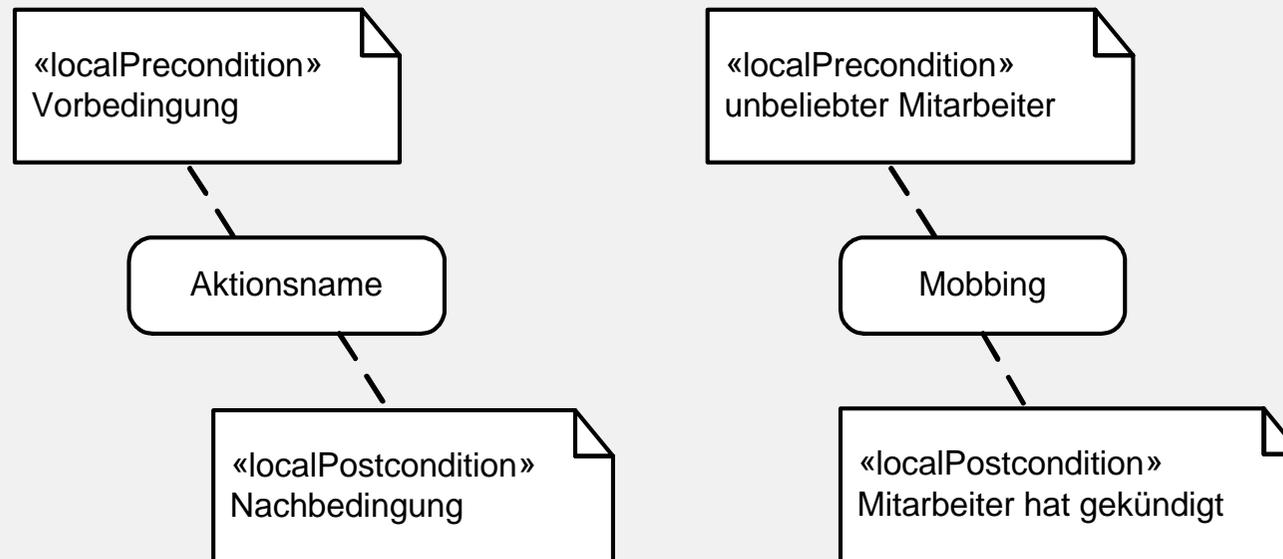
Aktion

- Aufruf von Verhalten oder Bearbeitung von Daten
- Einzelschritt der Aktivität
- Zentrales Element des Aktivitätsdiagramms
- Über Kanten mit anderen Elemente verbunden
- Um Aktion starten zu können, müssen auf allem eingehenden Kanten Token angeboten sein



Aktion

Vor- und Nachbedingung: sind bei Aktionsstart oder -ende Bedingungen zu beachten, können diese als Notizzettel notiert werden



Aktion

Signale und Ereignisse:

Sonderformen der Aktion;
`SendSignalAction` und

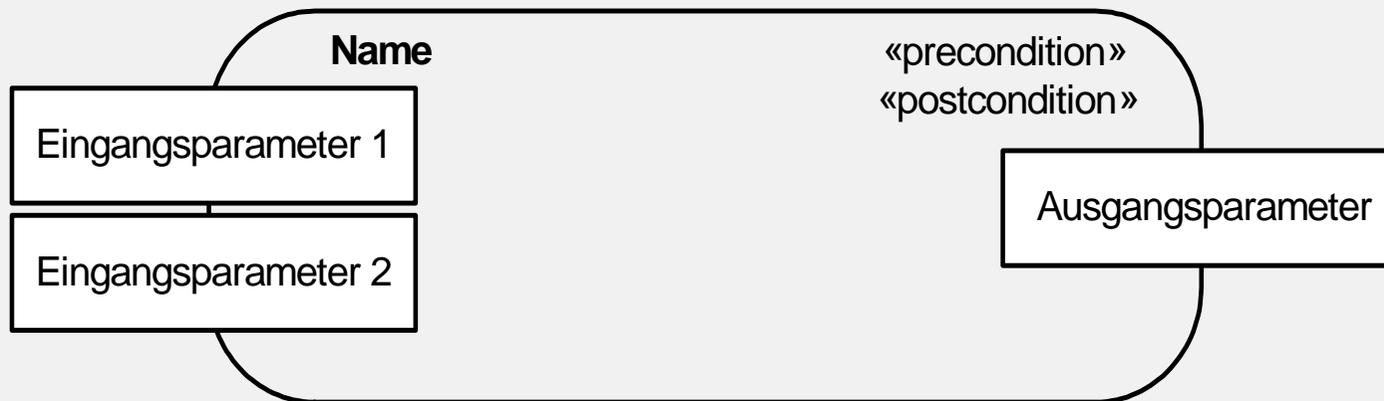


`AcceptEventAction`



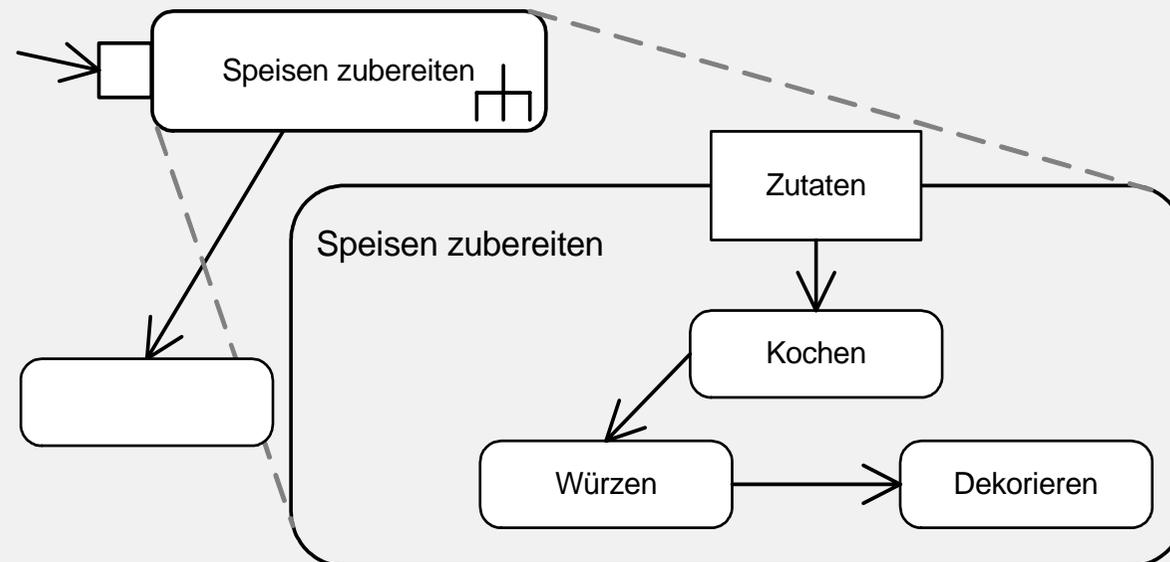
Aktivität

- Darstellung durch großes Rechteck mit abgerundeten Ecken
- Mögliche Parameter als Objektknoten auf Aktivitätsgrenze



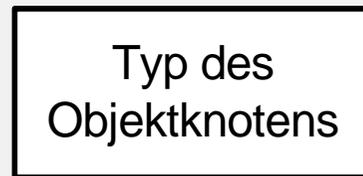
Aktivität

- Besteht aus Folge von Aktionen und Elementen
- Schachtelung möglich:
 - Vorteil für Lesbarkeit und Übersichtlichkeit
 - Kennzeichnung durch Harke rechts unten



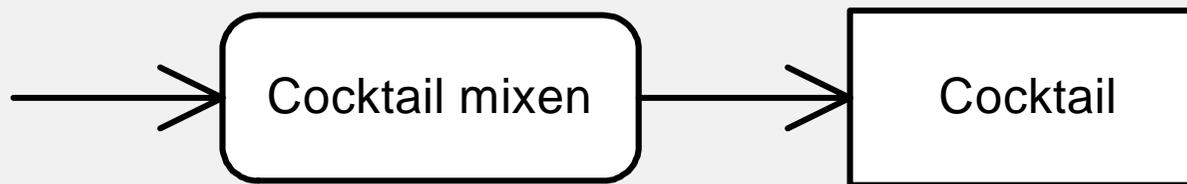
Objektknoten

Darstellung durch ein Rechteck



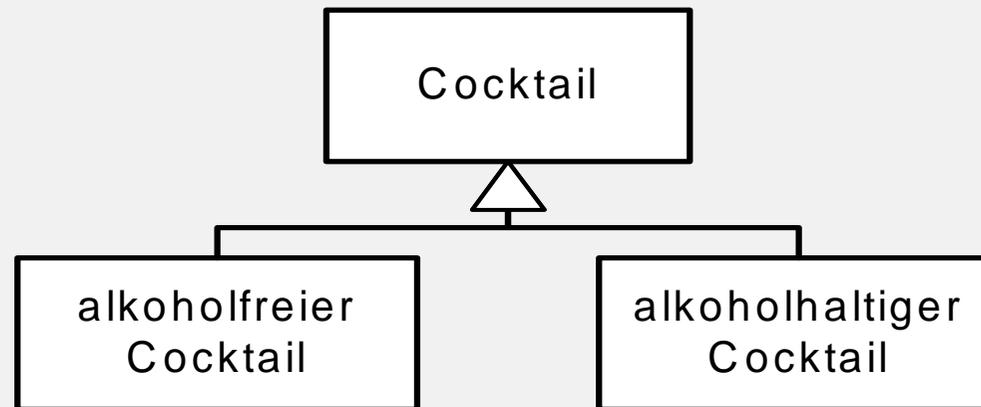
Objektknoten

- Repräsentiert Ausprägungen eines bestimmten Typs in der Aktivität
- Logisches Gerüst um Daten und Werte zu transportieren
- Ist Ergebnis von vorangegangener Aktion
- Ist Eingabe für folgende Aktion
- Führt ein Pfeil von Aktion zu Objektknoten: Ausprägung von Typ des Objektknoten ist Ergebnis der Aktion



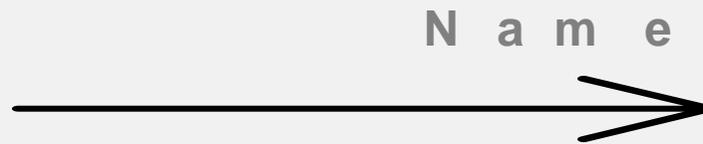
Objektknoten

Objektknoten akzeptier Daten von Token nur, wenn die Daten typkompatibel zum Objektknoten sind



Kanten

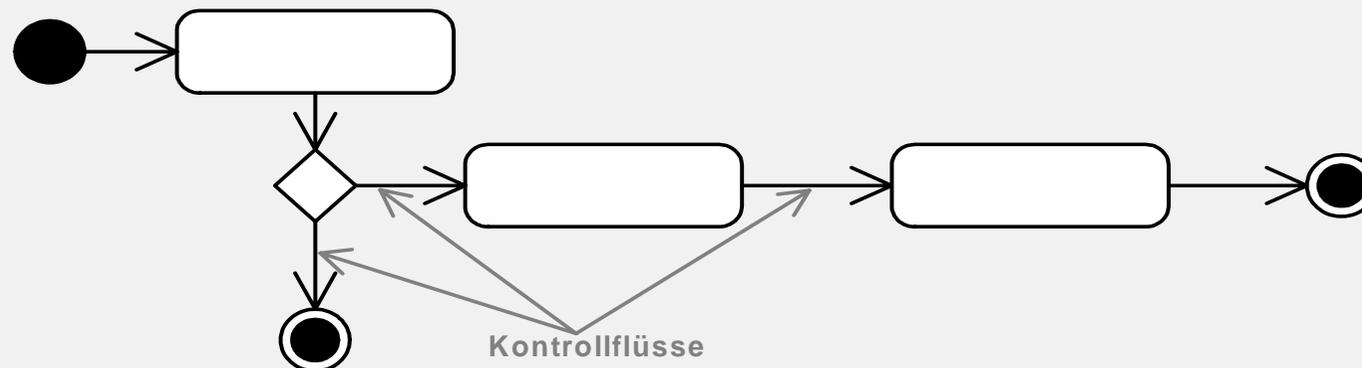
- Darstellung durch Pfeil mit Namen
- Ist gerichteter Übergang zwischen Knoten



Kanten

Es gibt zwei Sorten von Kanten

- a) Kontrollfluss: Kante zwischen zwei Aktionen oder zwischen Aktion und Kontrollelement. Token tragen hier keine Daten und dienen nur Aktionsausführung

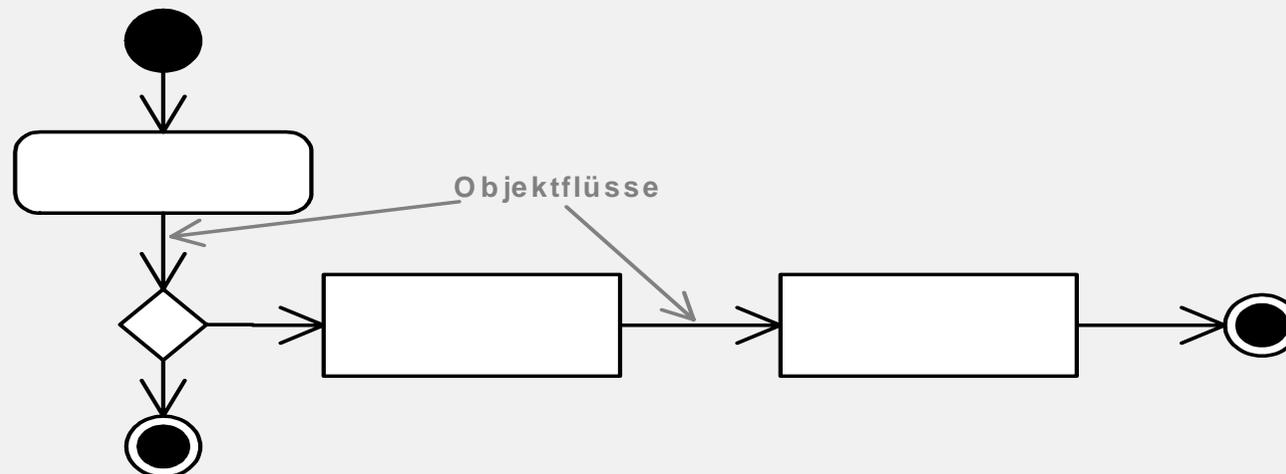


Kanten

b) Objektfluss:

Kante mit mindestens einem Objektknoten.

Kante trägt Token, der Daten zu oder von Objektknoten transportiert.



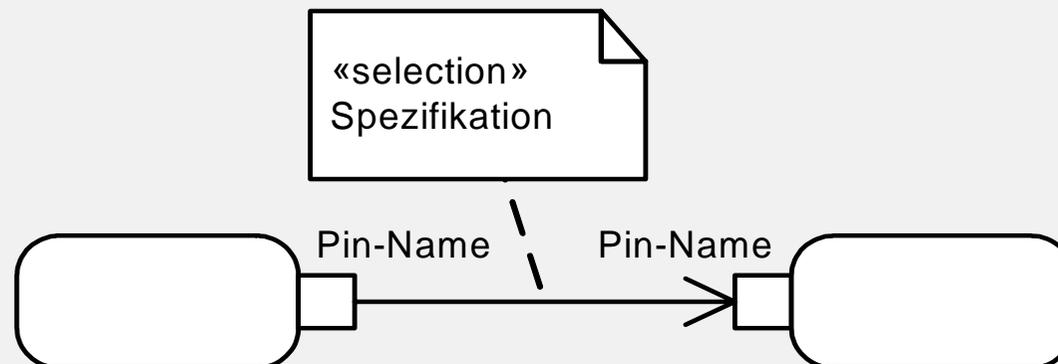
Kanten

Möglichkeiten:

- Belegung der Kante mit Bedingungen



- Sortierung von Datentoken um Reihenfolge auf Kante festzulegen

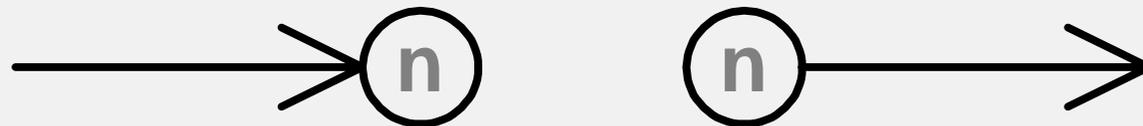


Kanten

- Gewichtung der Kanten um zu sehen, wie viele Token für einen Ablauf benötigt werden

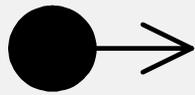


- Sprungmarken zur Unterbrechung von Kanten um Übersichtlichkeit zu wahren



Kontrollelemente

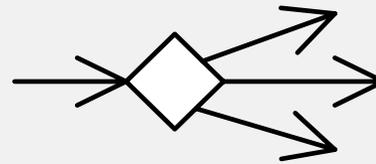
Zur Ablaufsteuerung gibt es mehrere Kontrollelemente:



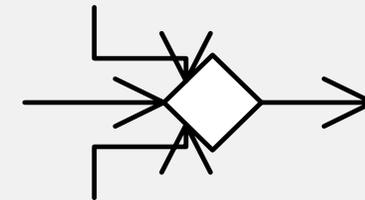
Startknoten



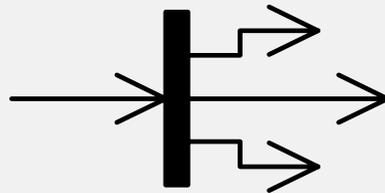
Endknoten



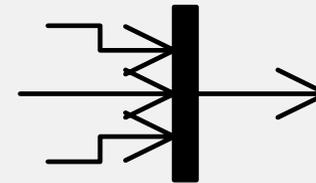
Verzweigungsknoten



Verbindungsknoten



Parallelisierungsknoten



Synchronisationsknoten

Kontrollelemente

- Startknoten: Startpunkt des Ablaufs bei Aktivierung von Aktivitäten
- Endknoten:
 - a) für Aktivitäten; beendet gesamte Aktivität
 - b) für Kontrollflüsse; markiert Ablaufende
- Verzweigungsknoten: spaltet Kante in mehrere Alternativen
- Verbindungsknoten: führt Kanten unsynchronisiert zusammen

Kontrollelemente

- Synchronisationsknoten: vereint eingehende Abläufe zu einem gemeinsamen Ablauf
- Parallelisierungsknoten: ein eingehender Ablauf wird in mehrere parallele Abläufe geteilt

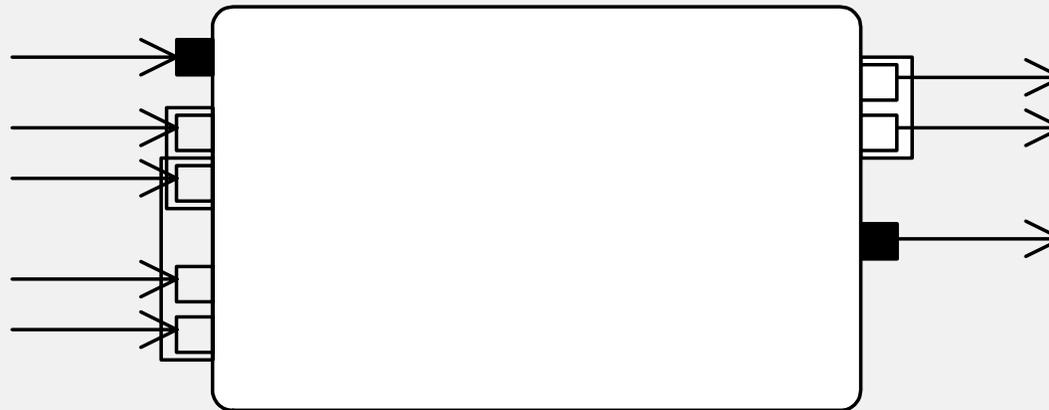
Parametersatz

Darstellung:



Parametersatz

- Dient der typhomogenen Gruppierung von Ein- und Ausgabeparametern von Objektknoten
- Kennzeichnung durch einen Rahmen um die zu gruppierenden Pins

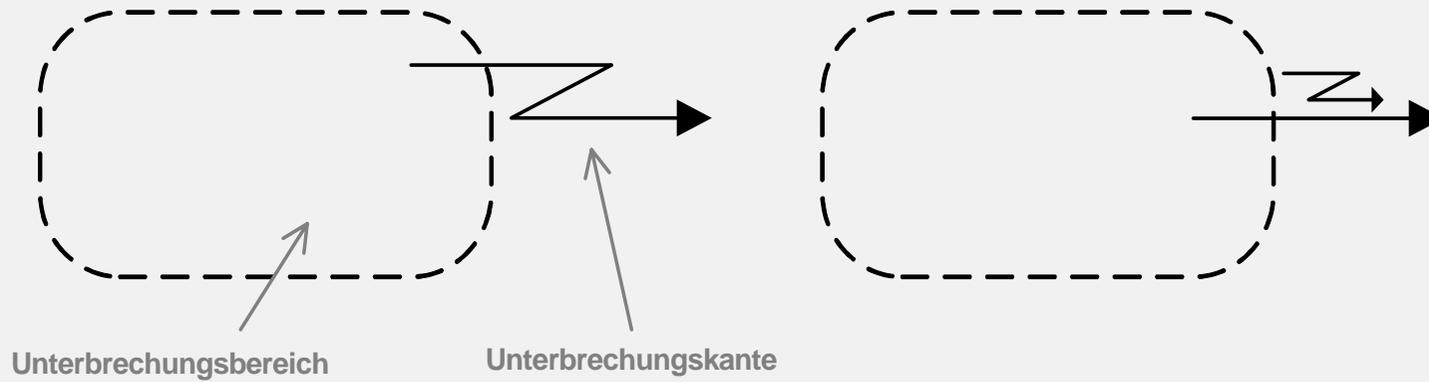


Parametersatz

- Legt fest, daß an allen Pins eines Satzes Datentoken anliegen müssen
- Ein Parametersatz bedient eine Aktion pro Aufruf
- XOR Beziehung zwischen Parametersätzen

Unterbrechungsbereich

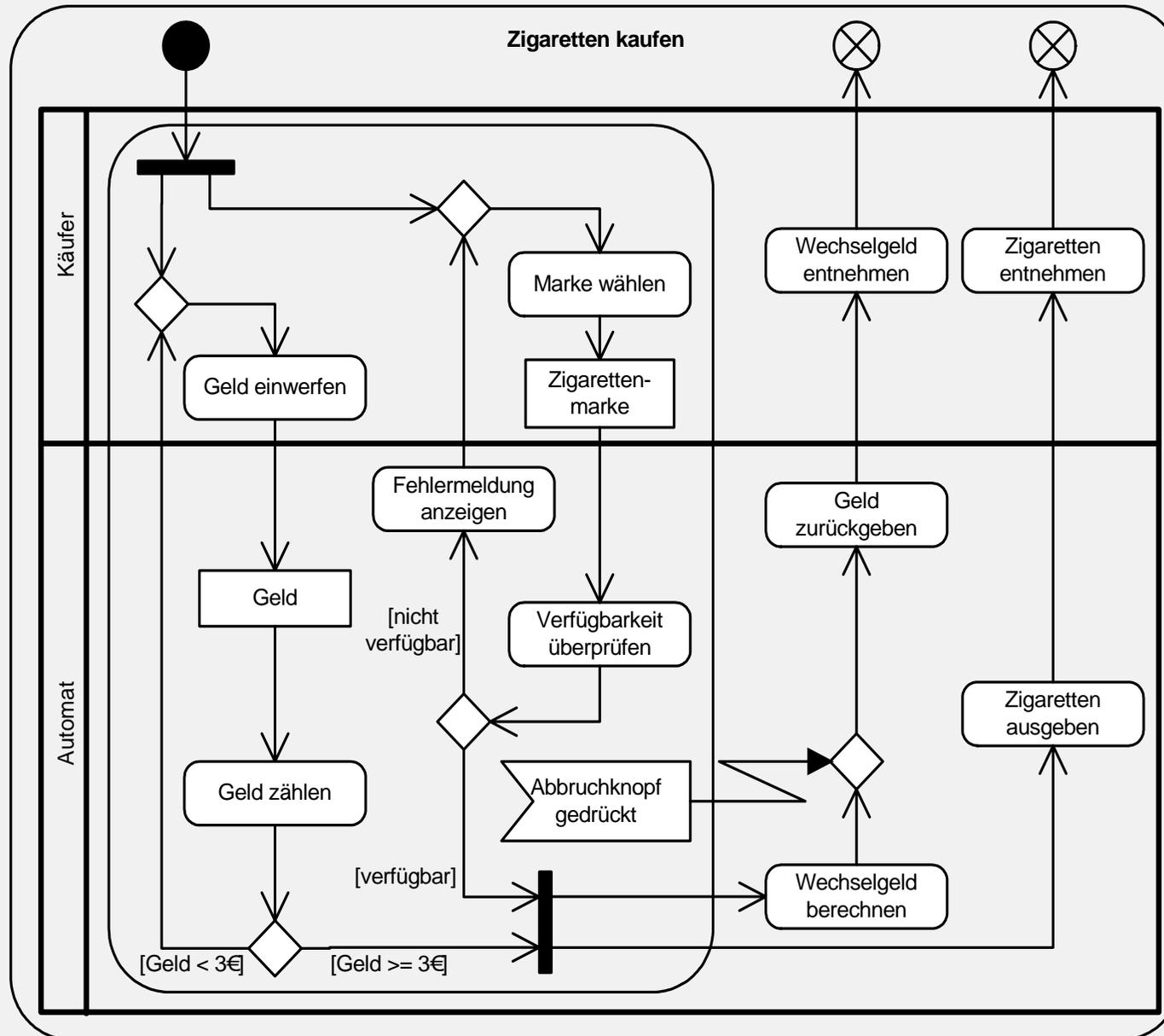
Darstellung:



Unterbrechungsbereich

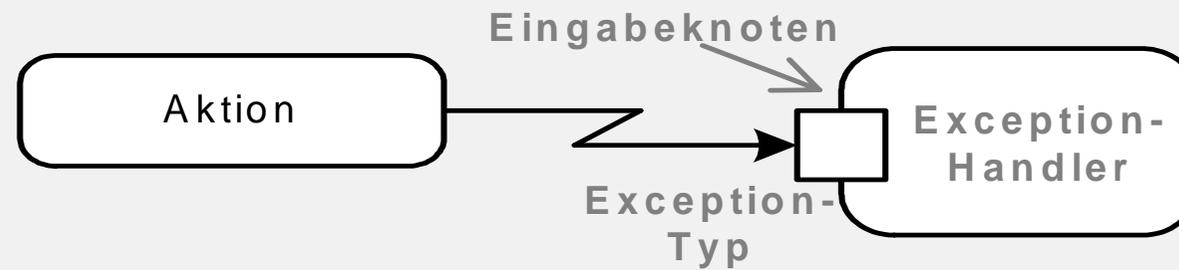
- Umschließt eine oder mehrere Aktionen
- Verlässt man Bereich über Unterbrechungskante:
 - Unterbrechen von sämtlichen Aktionen im Bereich, alle vorhandenen Token werden verworfen
 - Aktivitätsablauf wird an Zielknoten fortgesetzt

Unterbrechungsbereich



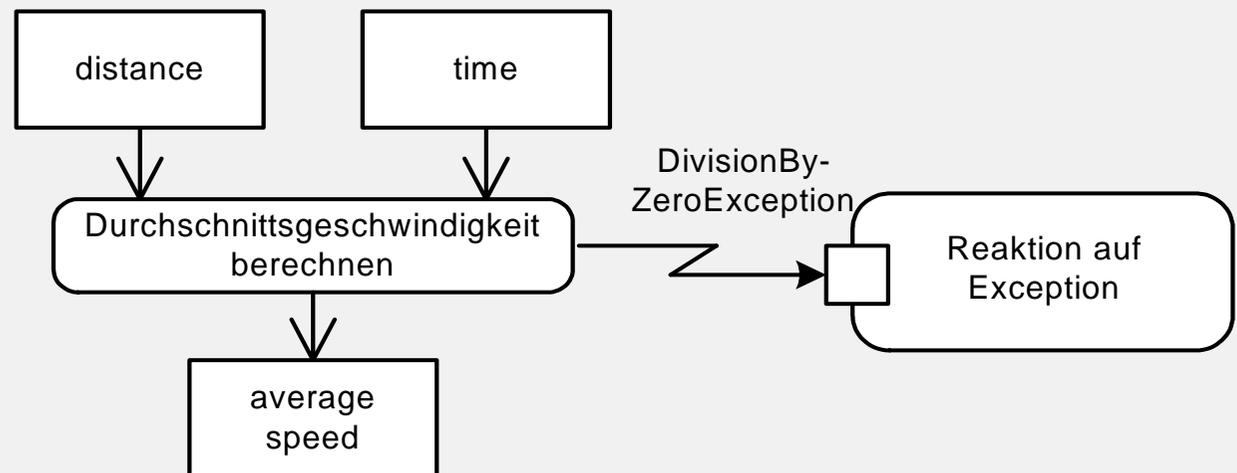
Exception-Handler

Darstellung:



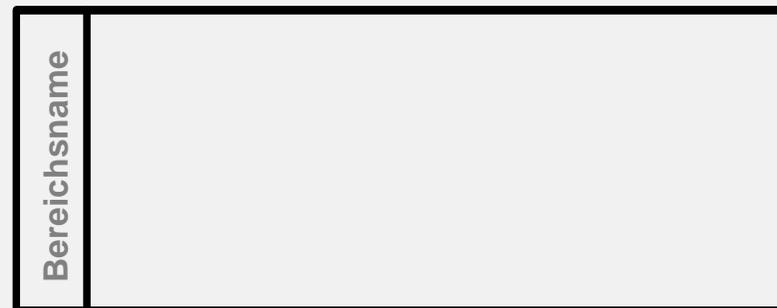
Exception-Handler

- Bearbeitungselement für Ausnahmen, die bei Aktionsausführung auftreten
- Bei Auftreten von Exception: Prüfung ob Exception-Handler vorhanden ist
- Exception-Handler ist dann für die Exception passend, wenn Datentoken von Aktion kompatibel zu Typ des Eingabeknotens des Exception-Handler ist



Aktivitätsbereich

Darstellung durch Rechteck, das in zwei Teile unterteilt wird



Aktivitätsbereich

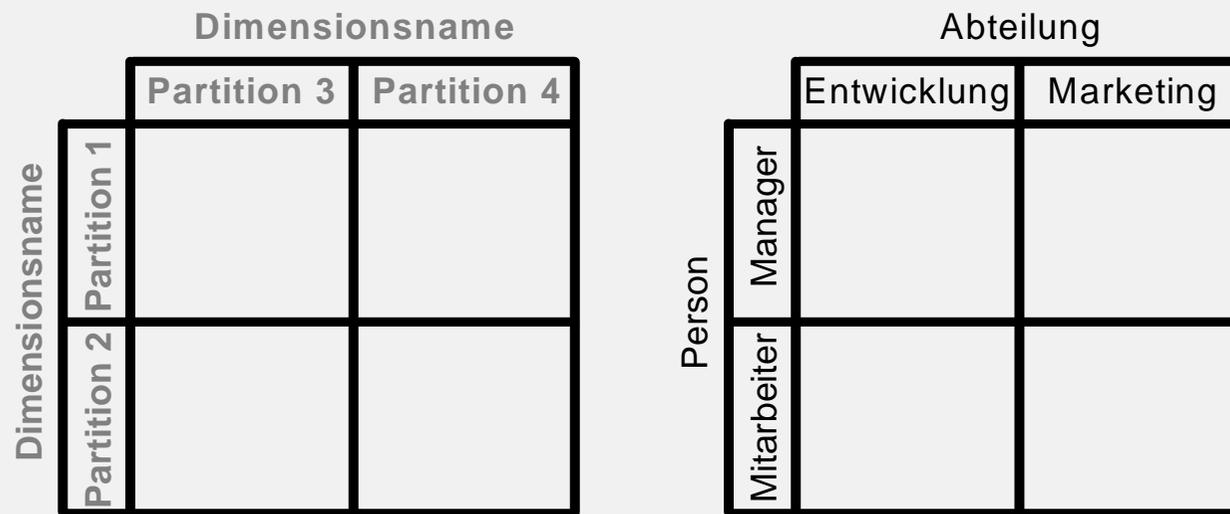
Aktivität wird in Bereiche mit gemeinsamen Eigenschaften unterteilt, wie z.B.:

- Standort
- Abteilung
- Rolle
- Verantwortlichkeit
- Subsystem

Aktivitätsbereich

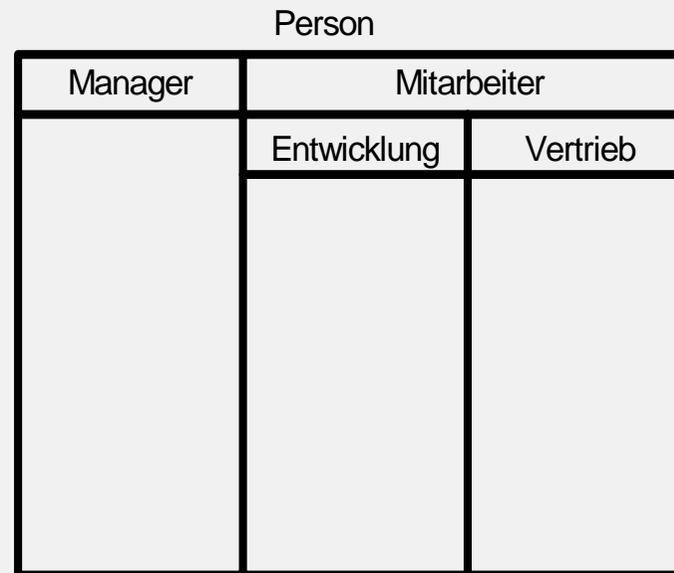
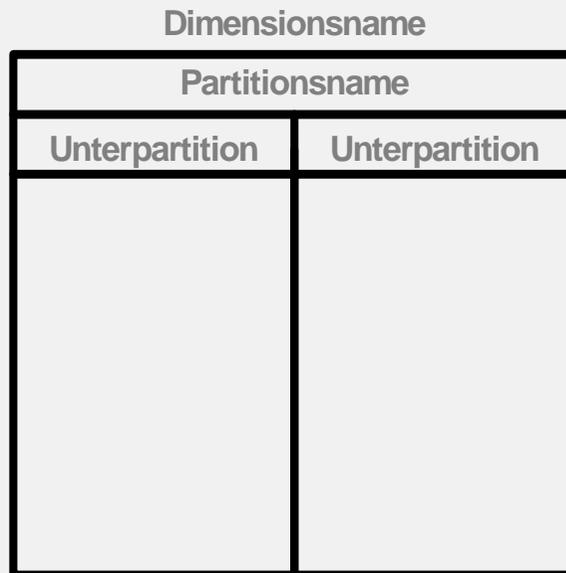
Es gibt verschiedene Aktivitätsbereiche:

a) Mehrdimensionale Aktivitätsbereiche



Aktivitätsbereiche

b) Hierarchische Aktivitätsbereiche



Aktivitätsbereiche

c) Aktivitätsbereiche für externe Aktionen



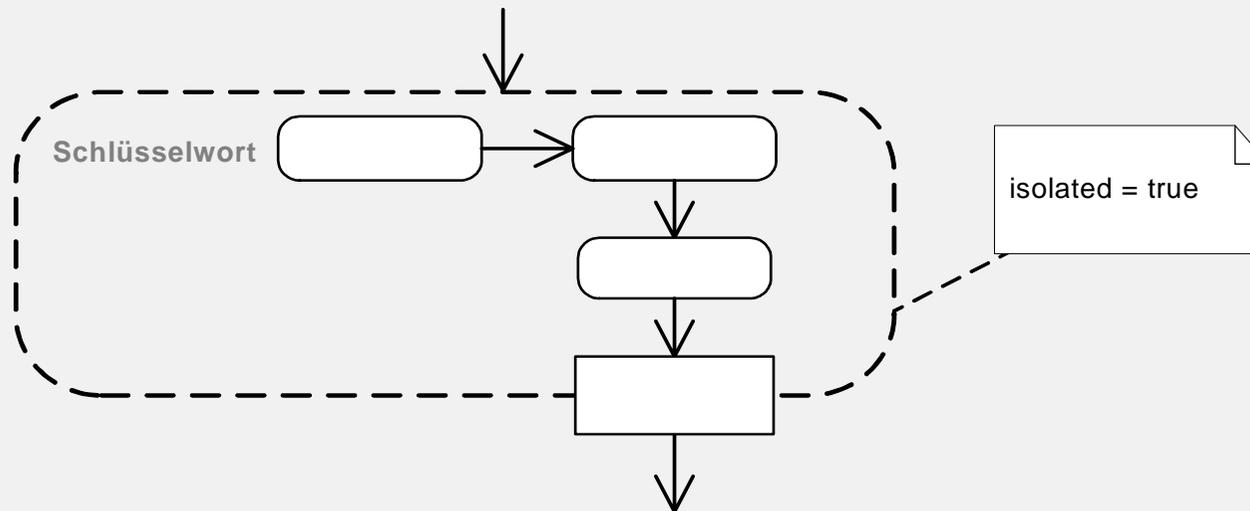
Strukturierter Knoten

Darstellung durch Rechteck mit abgerundeten Ecken und unterbrochener Linienführung



Strukturierter Knoten

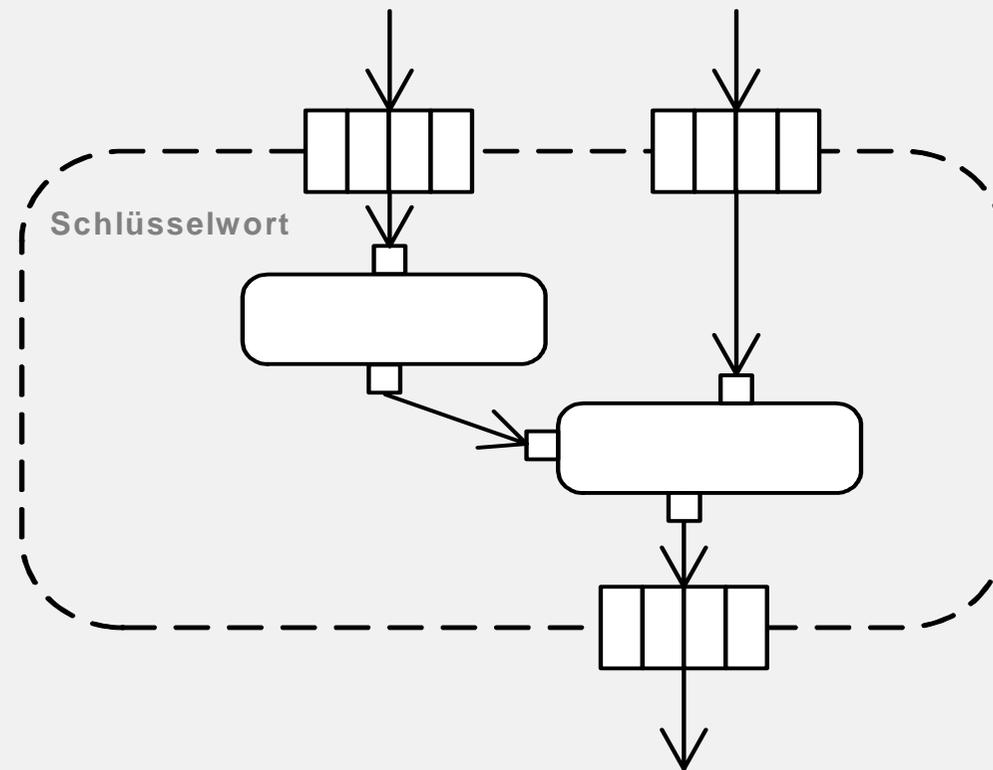
- Strukturierung durch Gruppierung der Elemente
- Darstellung als speziellen Knoten in Kontrollfluss



- Modellierung von Zugriffskontrollen auf gemeinsam genutzte Elemente bei Nebenläufigkeiten

Mengenverarbeitungsbereich

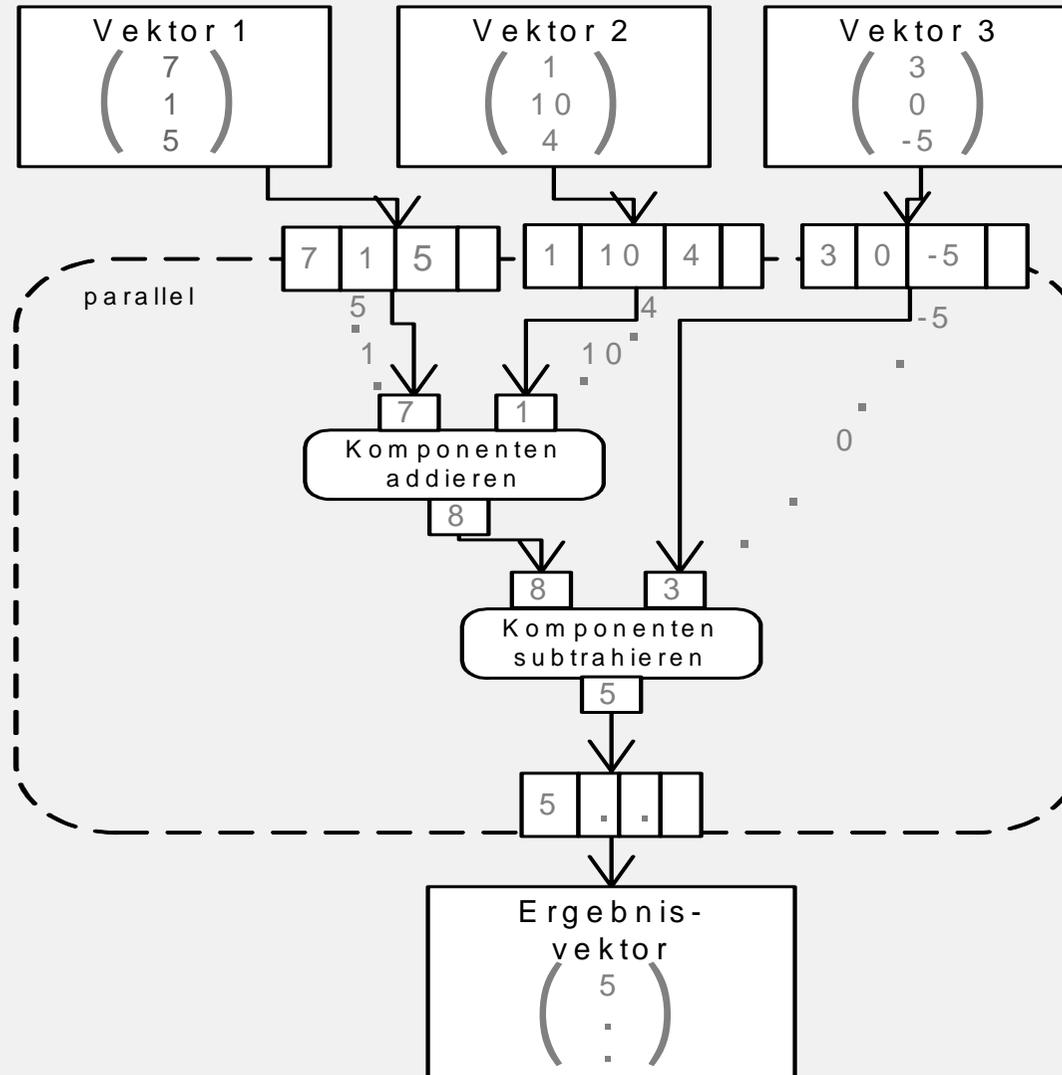
Darstellung:



Mengenverarbeitungsbereich

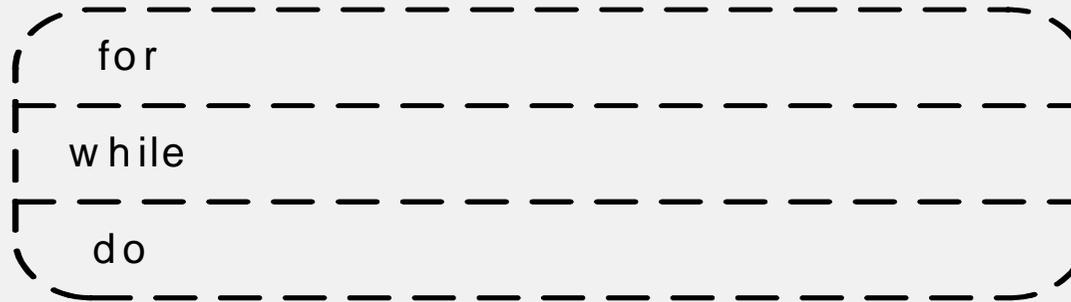
- Übergang von Aktionssammlung in Einzelelemente und wieder zurück
- Interne Weitergabe von Elementen oder Positionen der Sammlung
- Mengenverarbeitung iteriert über alle enthaltenen Elemente
- Kennzeichnung der Elemente, in welcher Reihenfolge sie abgearbeitet werden:
 - Iterativ = Sequenzielle Abarbeitung
 - Parallel = Abarbeitung findet nebenläufig statt
 - Stream = Elemente werden der Mengenverarbeitung kontinuierlich zur Verfügung gestellt

Mengenverarbeitungsbereich



Schleifenknoten

- Darstellung:



- Stellt Schleifen in Aktion dar
- Ist spezieller, strukturierter Knoten

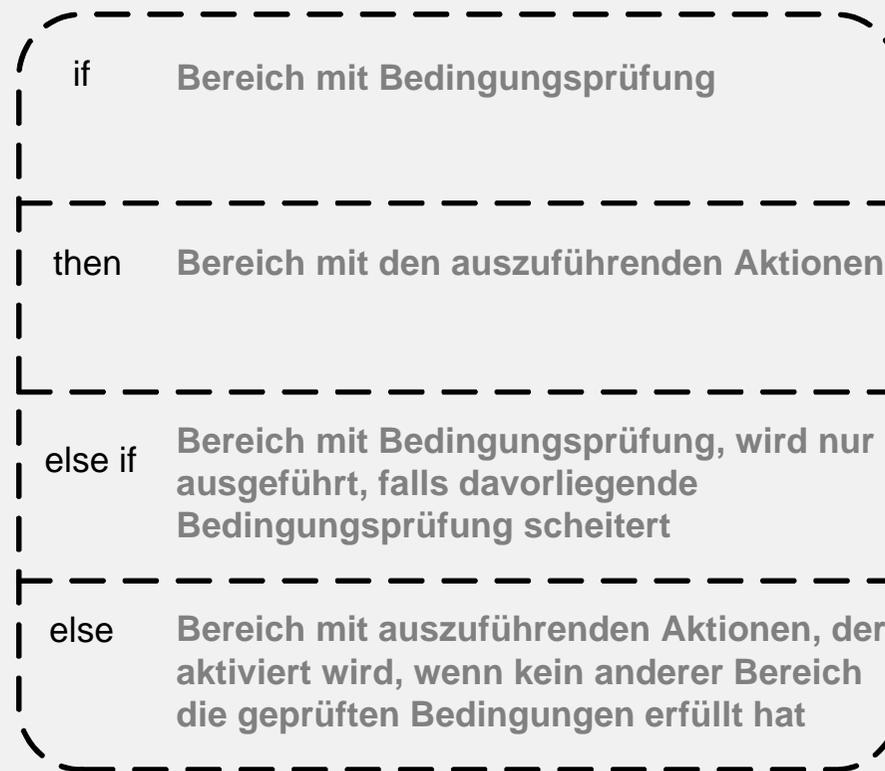
Schleifenknoten

Wird in drei Bereiche unterteilt:

- a) for-Bereich: erster Knotenbereich, wird genau einmal durchlaufen. Enthält Aktionen, Kanten, Kontrollelemente.
- b) while-Bereich: auch Testbereich; prüft ob und wie oft Schleifenrumpf durchlaufen wird. Hat Boolesche Variable als Ergebnis.
- c) do-Bereich: Schleifenrumpf, optional. Bei Ausführung müssen Tokenvorschriften beachtet werden.

Entscheidungsknoten

Darstellung:



Entscheidungsknoten

- Darstellung von Entscheidungen mit Einfach- oder Mehrfachauswahl
- Wird in vier Bereiche unterteilt:
 - a) if-Bereich: erster Knotenbereich. Enthält Aktivitätselemente für Bedingungsprüfung Ergebnis ist Boolesche Variable.
 - b) Then-Bereich: Elemente hängen von Bedingungsprüfung ab. Wenn aktiviert, endet Entscheidungsknoten und Token werden vernichtet.

Entscheidungsknoten

- c) else-if-Bereich: wie if-Bereich. Möglichkeit Prüfung in Reihenfolge zu schalten und zu schachteln.

- d) else-Bereich: wird ausgeführt, wenn if- und else-if-Bereich mit „false“ enden. Ist immer der letzte Bereich im Entscheidungsknoten.

3.4 Der Zustandsautomat

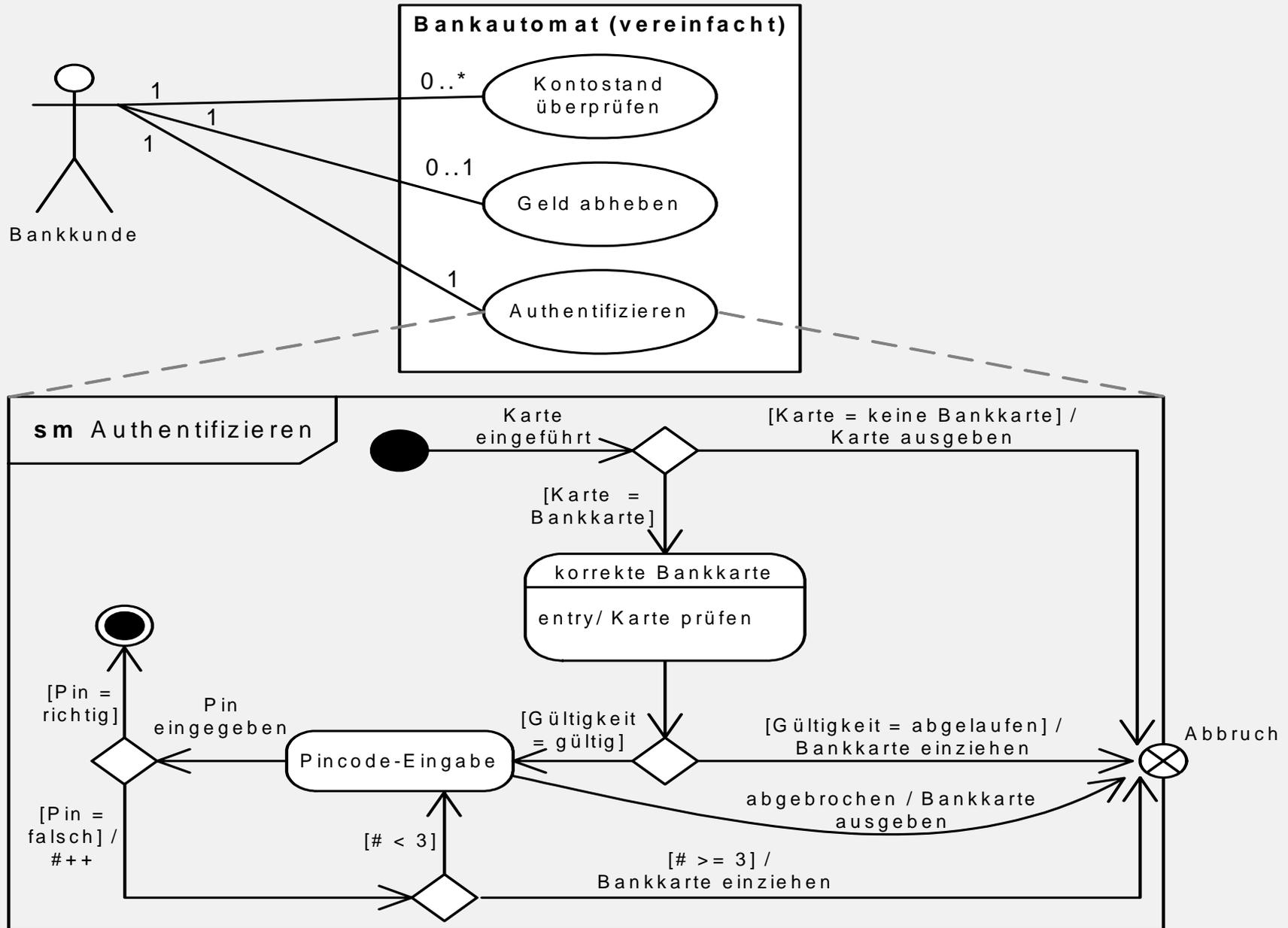
Zustandsautomat

Dient Modellierung von Verhalten von Classifiern durch Zustände und Übergängen zwischen Zuständen

Antwort auf die Frage:

„Wie verhält sich das System in einem bestimmten Zustand bei gewissen Ereignissen?“

Zustandsautomat



Zustandsautomat

- Mit dem Zustandsautomat wird die Darstellung gut les- und nachvollziehbar.
- Beschreibungsmöglichkeiten:
 - Natürlichsprachlich
 - Aktivitätsdiagramme
 - Zustandsautomaten

Notationselemente

- Einfacher Zustand
- Transition
- Start- und Endzustand
- Zustandsautomat
- Pseudozustände
- Region
- Spezialisierung
- Protokollzustandsautomat

Einfacher Zustand

Darstellung:



alternativ



Einfacher Zustand

- Bildet Situation ab, in deren Verlauf spezielle Bedingungen gelten
- Auch Situationen mit lang andauernder Aufgabe können als Zustände modelliert werden

Einfacher Zustand

Regeln für Zustände:

- Wenn Zustand Endpunkt einer durchlaufenen Transition ist, wird er betreten
- Wenn Transition durchlaufen ist, die von ihm weg führt, wird Zustand verlassen
- Wird aktiv bei Betreten (Eintrittsaktivität) und inaktiv bei Verlassen (Austrittsaktivität)
- Andauernde Aktivität folgt auf Eintrittsaktivität
- Kann Menge an verzögerbaren Triggern enthalten

Transition

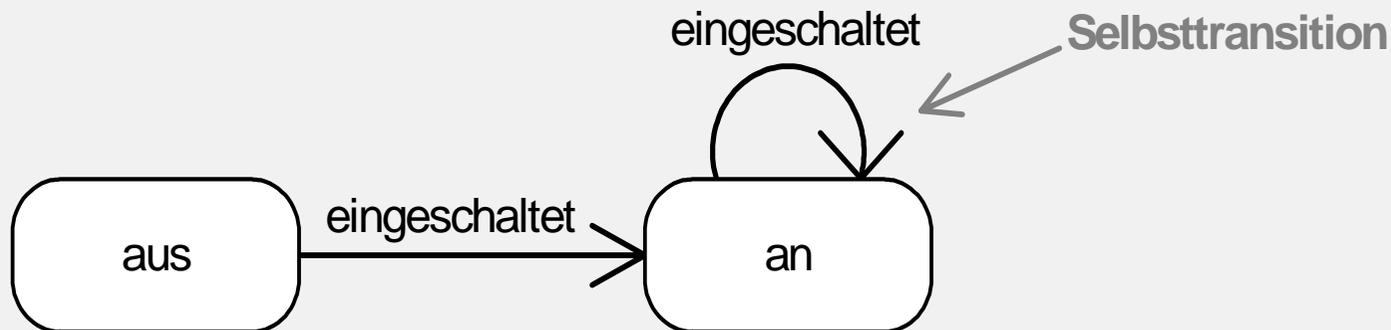
- Darstellung durch gerichtete, beschriftete Kante



- Beschriftung enthält:
 - Trigger als Auslösung der Transition
 - Guard als Durchlaufbedingung der Transition
 - Aktivität die beim Durchlauf ausgeführt wird

Transition

- Schafft Übergang von Ausgangs- zu Zielzustand
- Wenn Ausgangs- und Zielzustand gleich erfolgt die Selbsttransition



Startzustand

- Darstellung durch einen ausgefüllten Kreis



- Ist Startpunkt für Betreten des Zustandsautomats
- Von ihm aus wird der erste Zustand erreicht, da
 - Transitionstrigger nicht vorhanden oder gleich dem Trigger des Classifiers
 - Bedingung der Transition leer
 - Maximal eine abgehende Transition vorhanden sein darf

Endzustand

- Darstellung durch zwei Kreise, wobei der innere ausgefüllt ist



- Ende der Abarbeitung des Zustandsautomats
- Keine Ausführung von Aktivitäten mehr
- Keine ausgehenden Transitionen

Zustandsautomat

- Darstellung durch Rechteck, in dem ein Fünfeck in der linken oberen Ecke platziert ist



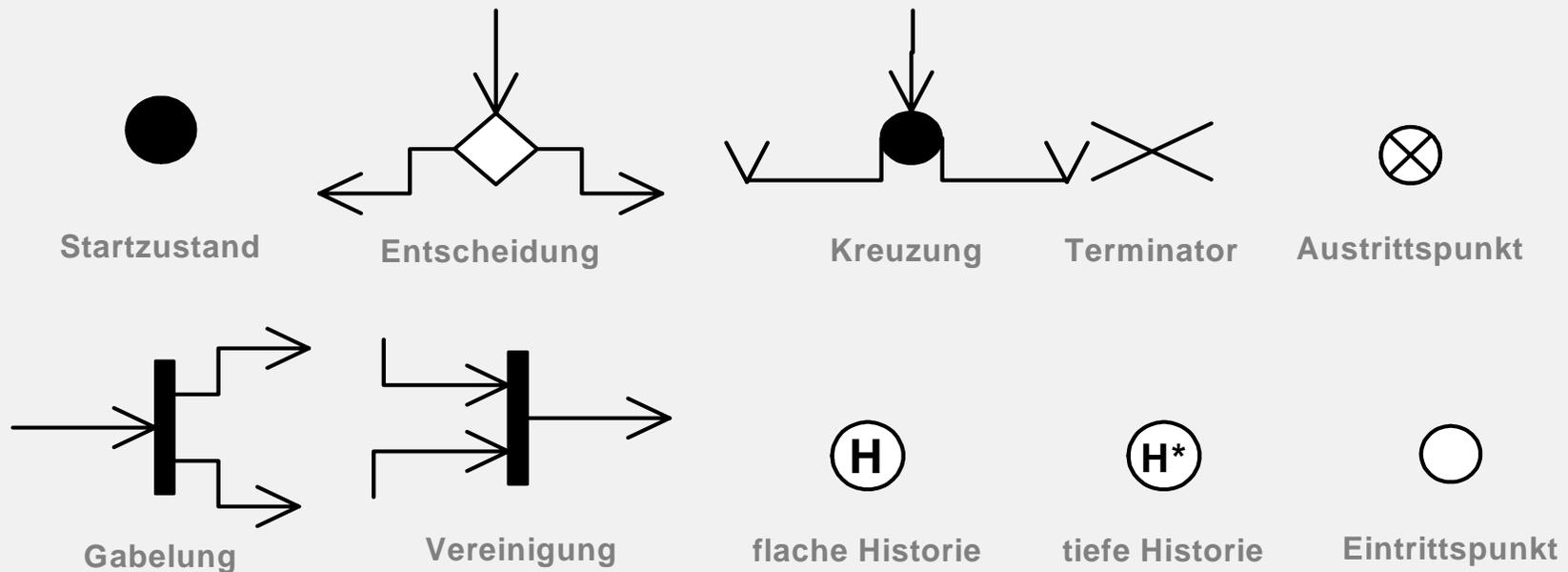
- Beschreiben das Verhalten von Classifiern
- Graph mit Knoten und Kanten

Zustandsautomat

- Knoten des Zustandsautomats:
 - Einfache Zustände, Start- und Endzustand
- Als Kantentyp kommen hier nur Transitionen vor

Pseudozustände

Hiervon gibt es zehn, die folgendermaßen dargestellt werden:



Pseudozustände

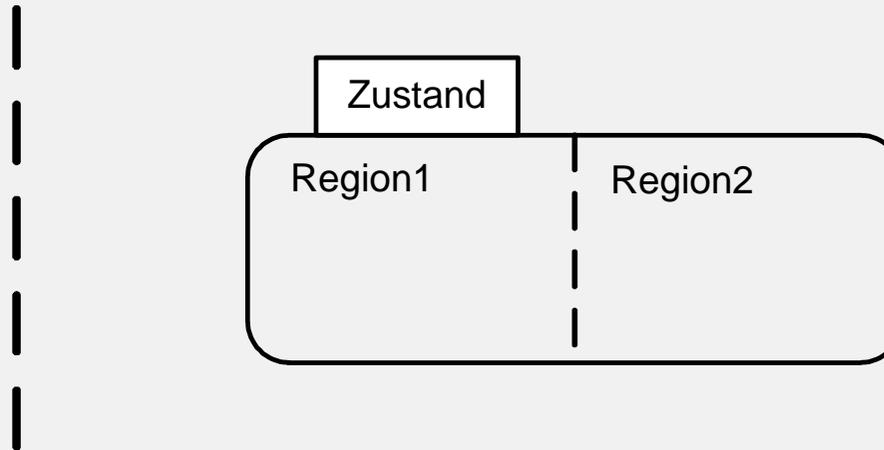
- Startzustand: Festlegung des ersten Zustands der im Zustandsautomat aktiviert werden soll
- Kreuzung: Zusammenfassung mehrerer Transitionen oder Aufteilung von einer auf mehrere Transitionen
- Entscheidung: siehe Kreuzung. Erfolgt aber dynamisch.
- Terminator: Unterscheidung ob Durchlaufen des Zustandsautomats erfolgreich oder ob Abbruch der Ausführung

Pseudozustände

- Gabelung und Vereinigung: Aufteilung einer eingehenden Transition auf mehrere, parallele Verhaltensbeschreibungen
- Ein- und Austrittspunkt: Betreten oder Verlassen von Zustandsautomat und Zustandshierarchien
- Flache und tiefe Hierarchie: Speichern des letzten aktiven Zustands eines Zustandsautomats

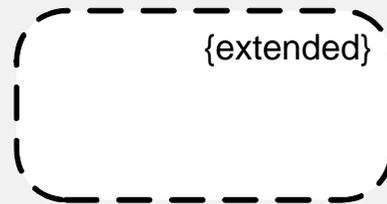
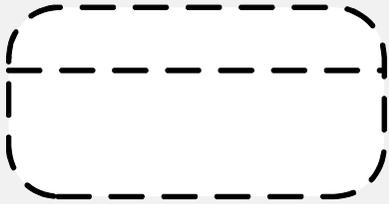
Region

- Darstellung durch eine senkrechte gestrichelte Linie
- Beschreibt parallele Abläufe



Spezialisierung

Darstellung:

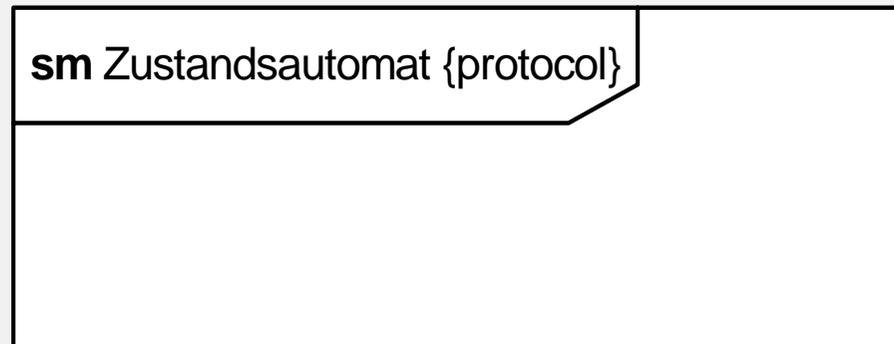


Spezialisierung

- Strukturelle Spezialisierung eines Classifiers durch Hinzufügen von Operationen oder Attributen
- Abbildung des Verhaltens des spezialisierten Classifiers durch Zustandsautomat
- Keine Modellierung neuer Zustandsautomaten; nur Darstellung der benötigten Änderungen des Verhaltens
- Änderungen bei
 - Regionen
 - Zuständen
 - Transitionen

Protokollzustandsautomat

Darstellung wie Zustandsautomat mit zusätzlichem Schlüsselwort `protocol`



Protokollzustandsautomat

- Protokolltransition ist eine Transition im Protokollzustandsautomat
- Unterscheidet sich durch Elemente von „normalem“ Zustandsautomat:
 - Vorbedingung = wie Guard bei Transition
 - Operation = Auslösen von Zustandsübergängen
 - Nachbedingung = Erfüllung des Zielzustandes

