Fachhochschule
Dortmund
University of Applied Sciences and Arts

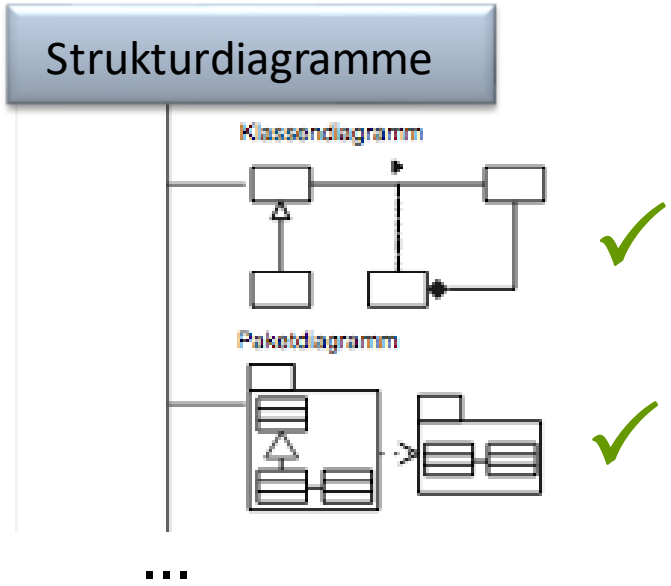
Softwaretechnik 1(A)

UML-Aktivitätsdiagramm

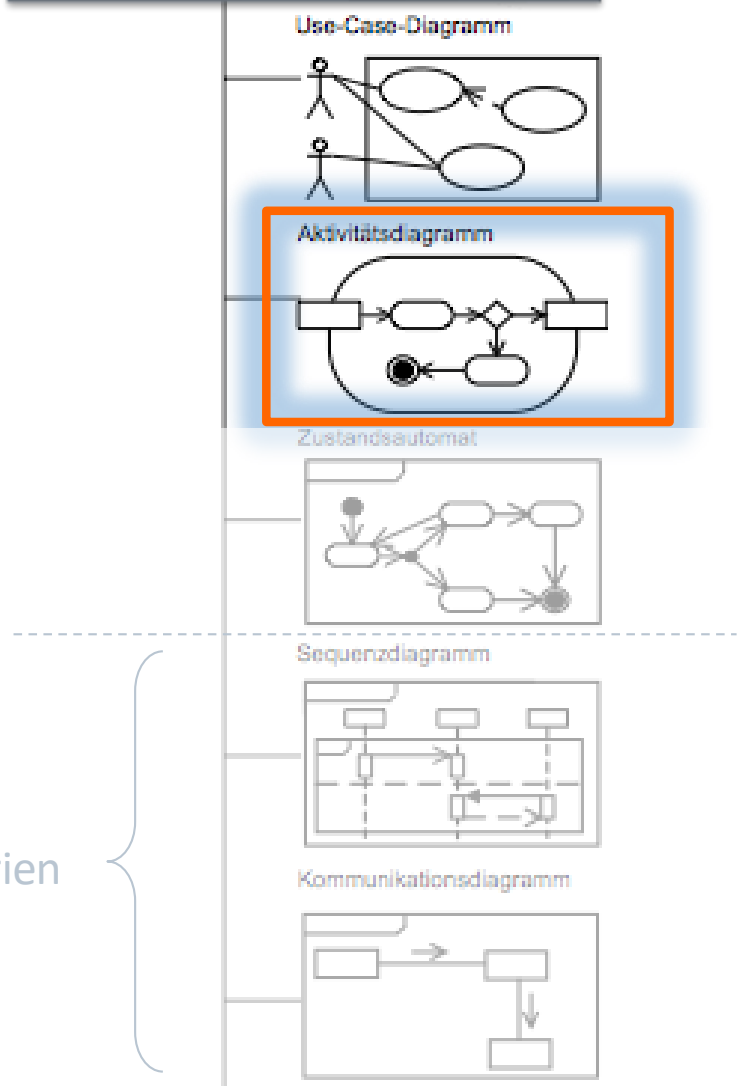
Autorin: Prof. Dr. Sabine Sachweh

UML Diagramme

Strukturdiagramme



Verhaltensmodellierung



Fachhochschule
Dortmund
University of Applied Sciences and Arts

Ein Beispiel: UNO

UNO

UNO-Spielbeschreibung (Fortsetzung)

- **Uno ist ein Kartenspiel** für 2-10 Spieler, das reihum (im Uhrzeigersinn beginnend) gespielt wird.
- **Karten:**
  Es gibt von den vier Farben (blau, grün, rot, gelb) jeweils Karten mit Werten von 0 bis 9, wobei alle Karten bis auf die Nullen doppelt vorhanden sind: ⇒ (4*10) + (4*9) = 76 Karten.

Außerdem gibt es drei farbige Aktionskarten: (Zieh Zwei 2+, Aussetzen, Richtungswechsel)
zweimal in jeder Farbe: (2*4*3) ⇒ (24 Karten):

- **Zieh Zwei (+2)**
  Der nächste Spieler muss zwei Karten aufnehmen.
- **Aussetzen**
  Der nächste Spieler wird übersprungen.
- **Richtungswechsel**
  Bei mehr als zwei Spielern wird die Spielrichtung gewechselt, bei zwei Spielern hat die Karte die gleiche Funktion wie die Aussetzen-Karte.

UNO-Spielbeschreibung (Fortsetzung)

- Zusätzlich gibt es zwei Arten schwarze Aktionskarten:
  (Farbwahl, Zieh Vier Farbwahl (4+))

  Von beiden Sorte gibt 4 Karten: (4*2) ⇒ (8 Karten).
  Schwarze Aktionskarten können nur gespielt werden, wenn der Spieler keine andere Karte ausspielen kann.

  - **Farbwahl**
    Der Spieler, der diese Karte spielt, schreibt dem nächsten Spieler vor, welche Farbe dieser legen muss.
  - **Zieh Vier Farbwahl (+4)**
    Der nächste Spieler muss vier Karten aufnehmen, außerdem kann der Spieler, der die +4 gelegt hat, die zu spielende Farbe bestimmen.

- Zusammen also 108 Karten.

UNO-Spielbeschreibung (Fortsetzung)

Zusammen gibt es also 108 Karten:

Wert

Farbe	0	1	2	3	4	5	6	7	8	9	Aussetzen	+2	Richtung wechseln	+4	Farbe wünschen
rot	1	2	2	2	2	2	2	2	2	2	2	2	2		
blau	1	2	2	2	2	2	2	2	2	2	2	2	2		
grün	1	2	2	2	2	2	2	2	2	2	2	2	2		
gelb	1	2	2	2	2	2	2	2	2	2	2	2	2		
schwarz														4	4
											10	11	12	0	1

Karte
farbe: Farbe wert: Wert

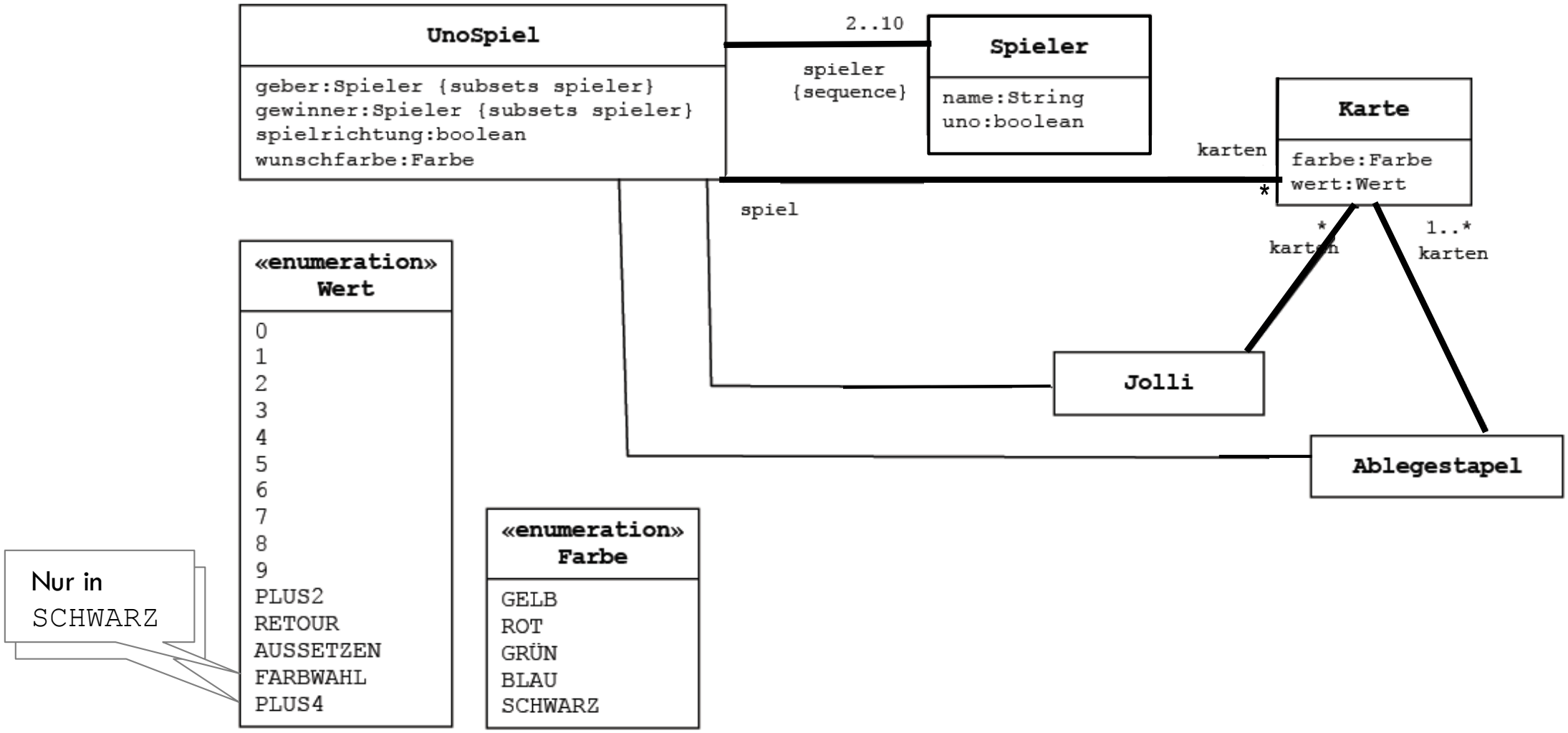
UNO- Spielbeschreibung

Grundregeln

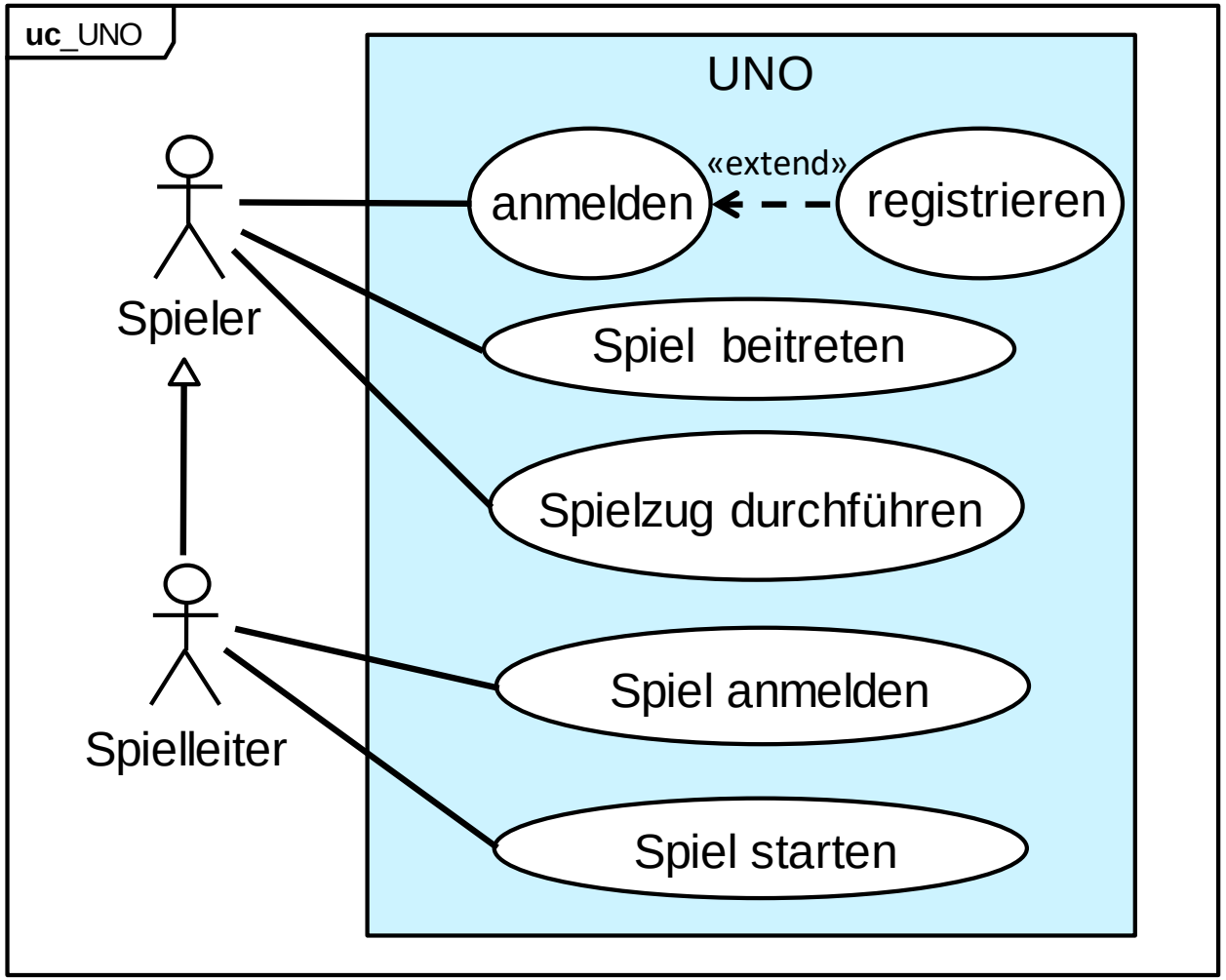
- Vorbereitung:
  Zu Beginn einer Runde werden jedem Spieler sieben Karten ausgeteilt, der Rest kommt auf einen verdeckten Stapel (Jolli) in die Mitte.
  Eine Karte wird aufgedeckt und ist damit die erste aufgedeckte Karte des Ablegestapels.

- Runden:
  - der erste Spieler beginnt, danach geht es im Uhrzeigersinn weiter
  - an eine Runde schließt sich unmittelbar die nächste an

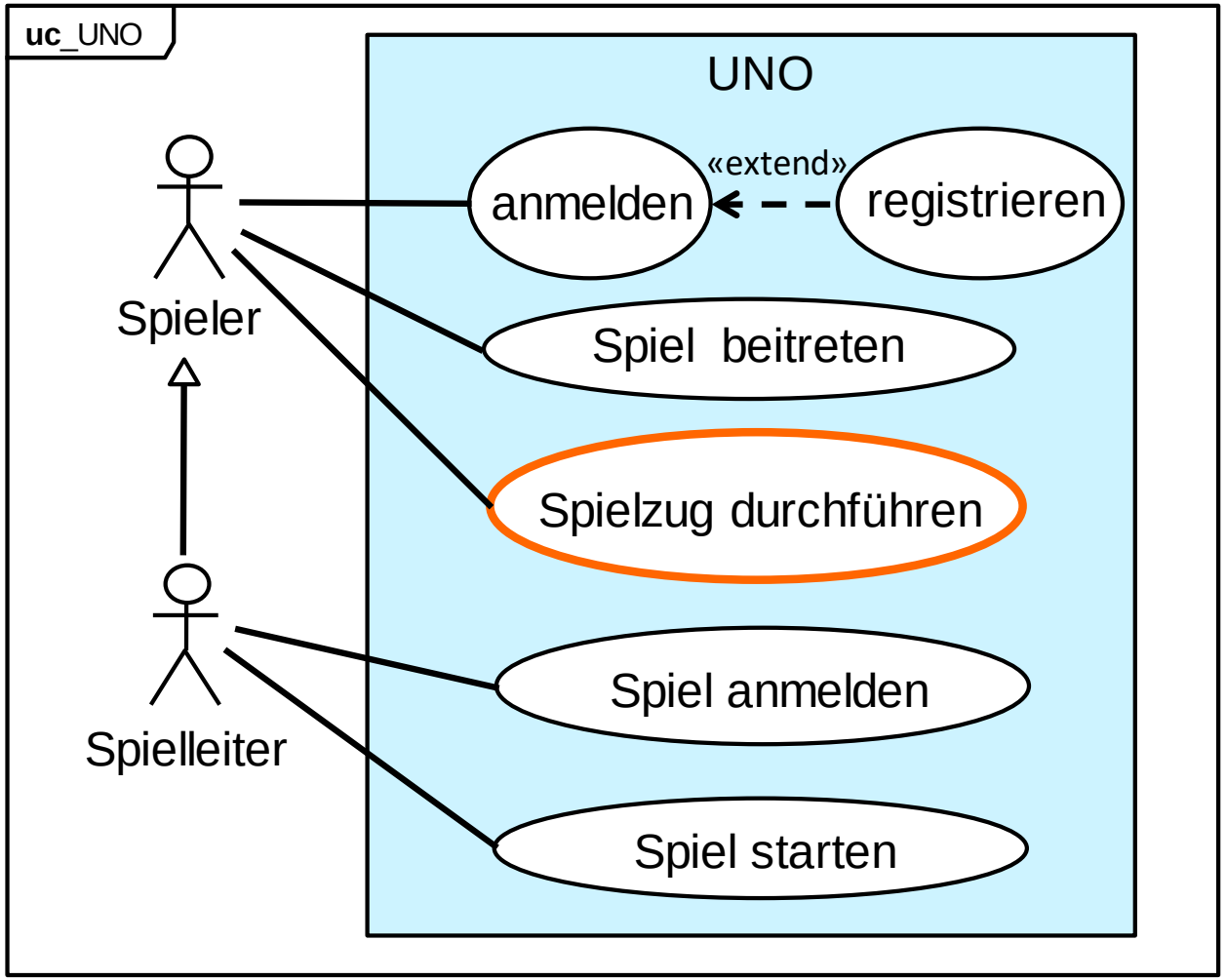
UNO


UnoSpiel
geber:Spieler {subsets spieler}
gewinner:Spieler {subsets spieler}
spielrichtung:boolean
wunschfarbe:Farbe
2..10
spieler
{sequence}
Spieler
name:String
uno:boolean
karten
*
Karte
farbe:Farbe
wert:Wert
spiel
*
karten
1..*
karten
Jolli
Ablegestapel
«enumeration»
Wert
0
1
2
3
4
5
6
7
8
9
PLUS2
RETOUR
AUSSETZEN
FARBWAHL
PLUS4
«enumeration»
Farbe
GELB
ROT
GRÜN
BLAU
SCHWARZ
Nur in
SCHWARZ

UNO – Use Case Diagramm



UNO – Use Case Diagramm



UNO- Spielbeschreibung (textuell)

- Spielzug
  - Im Normalfall muss der aktuelle Spieler eine Karte abgelegen.
    - Dazu muss die abzulegende Karte, denselben Wert oder dieselbe Farbe wie die aufgedeckte Karte bzw. die gewünschte Farbe haben. (→ Kompatibilität).
    - Wer keine passende Karte hat, muss eine vom Stapel nehmen. Diese kann er, sofern sie passt, auch gleich ausspielen.
    - Die schwarzen Aktionskarten können jederzeit abgelegt werden. Wenn eine solche Karte abgelegt wird, muss sich der aktuelle Spieler eine Farbe wünschen.
    - Richtungswechsel: Wenn eine Richtungswechsel-Karte aufgedeckt auf dem Tisch abgelegt wurde, wird die Spielrichtung umgekehrt.
    - UNO!: Wer die vorletzte Karte ablegt, muss vorher „Uno!“ rufen, wenn er gewinnen möchte, sonst muss in der nächsten Runde eine Extra-Karte gezogen werden.
    - Wenn der Spieler nach dem ablegen keine Karte mehr hat, ist er der Gewinner und die Runde ist beendet.
  - Karten ziehen: Wenn eine 2+ oder eine 4+ Karte aufgedeckt auf dem Tisch liegt, muss der aktuelle Spieler 2 bzw. 4 Karten ziehen.
  - Aussetzen: Wenn eine Aussetzen-Karte aufgedeckt auf dem Tisch liegt, muss der aktuelle Spieler aussetzen.

UNO- Spielbeschreibung - Use Case „Spielzug durchführen“

Wie sieht eine Anwendungsfallspezifikationsschablone Zum Anwendungsfall„Spielzug durchführen“ aus?

UNO- Spielbeschreibung - Use Case „Spielzug durchführen“ (Anwendungsfallspezifikationsschablone)

Anwendungsfall:	**Spielzug**	
Ziel:	Spieler führt einen Zug durch	
Kategorie:	primär	
Vorbedingung:	Spieler ist an der Reihe	
Nachbedingung Erfolg:	Karte abgelegt	
Nachbedingung Fehlschlag:	Karte(n) gezogen oder ausgesetzt	
Akteure:	Spieler	
Auslösendes Ereignis:	Spielbeginn oder Vorgänger hat Zug beendet	
Beschreibung:	**Akteur**	**Schritte des Standardfalls**
	Spieler Spieler Spieler Spieler	legbare Karten identifiziert 2 Karte ablegen 3 Zug abschließen
Erweiterung:	**Bedingung**	**Erweiterung von Aktionen des Standardfalls**
	*** Farbwahlkarte vorletzte Karte letzte Karte Richtungswechsel	1a Karte(n) ziehen 2a Farbe wünschen 2b UNO! Sagen 2c Gewonnen! sagen 2d Spielrichtung ändern
Alternativen:	**Bedingung**	**Alternativen zu Aktionen des Standardfalls**
	keine Karte ablegbar Aussetzkarte auf dem Stapel	2a aussetzen

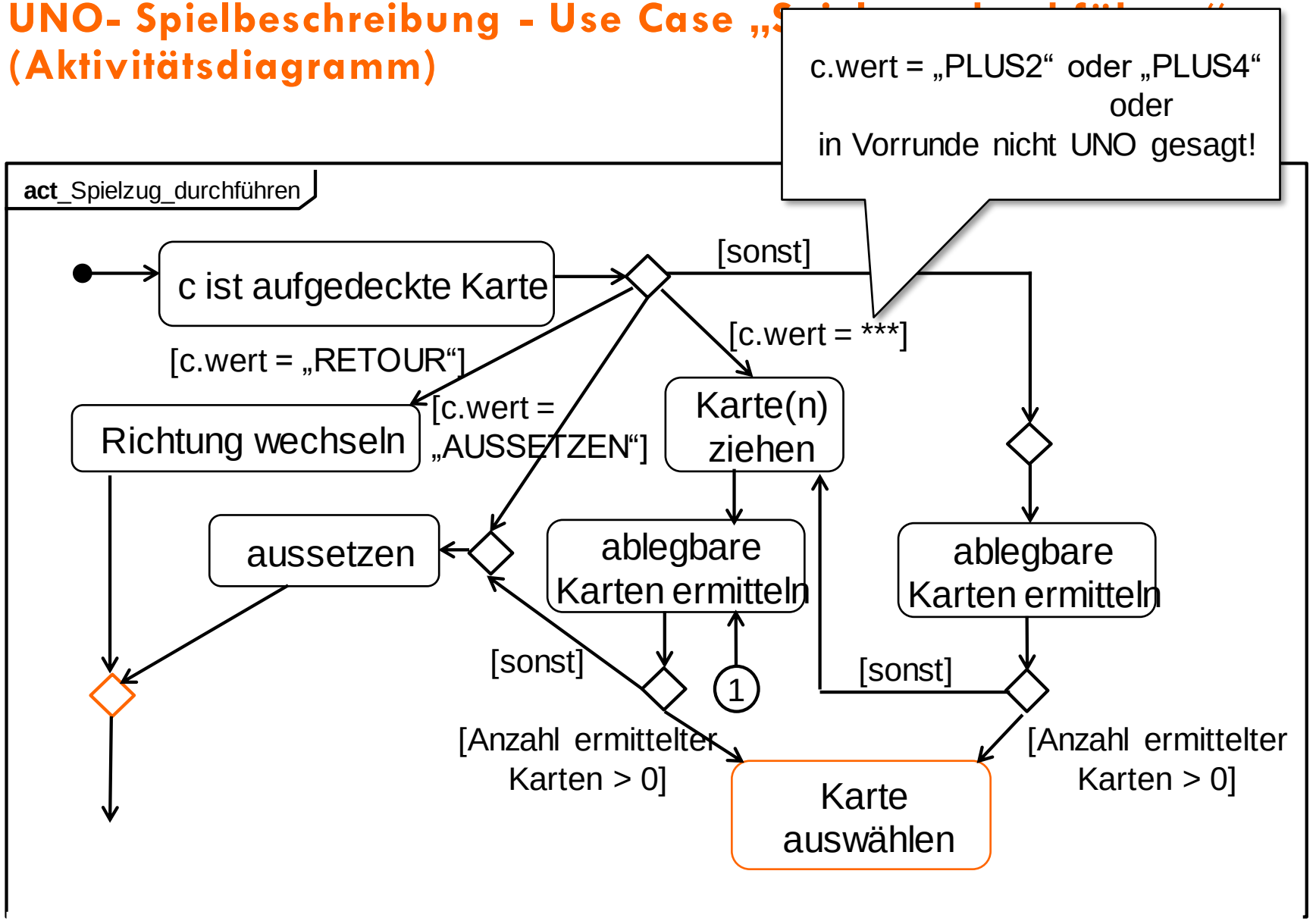
***
2+ oder 4+ Karte auf Stapel
in Vorrunde nicht UNO! gesagt
keine ablegbare Karte auf der Hand

UNO- Spielbeschreibung - Use Case „Spielzug durchführen“

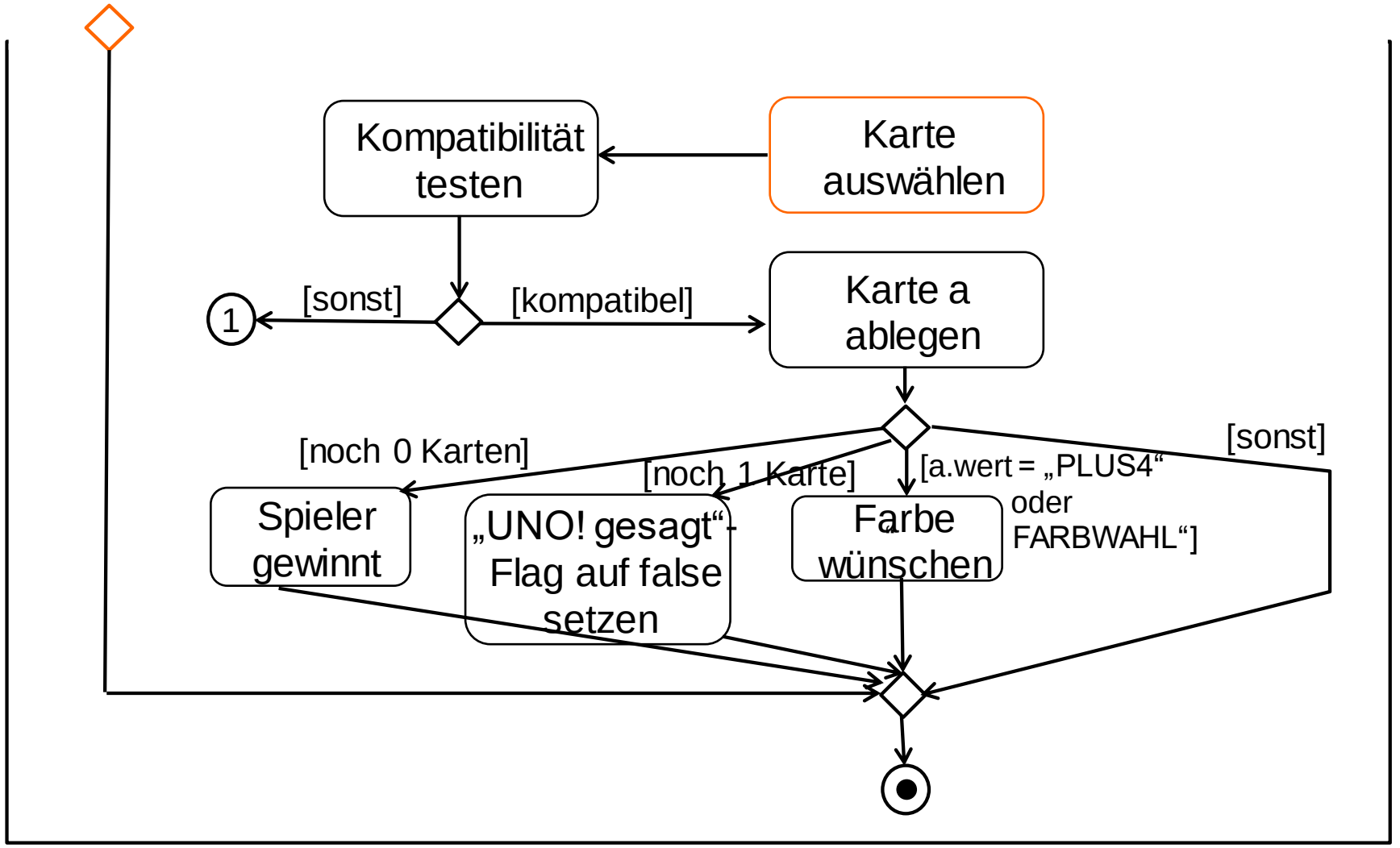
Gibt es nicht eine Darstellung, die
die Struktur des Ablaufs

deutlich zeigt?

UNO- Spielbeschreibung - Use Case „S… (Aktivitätsdiagramm)



UNO- Spielbeschreibung - Use Case „Spielzug durchführen“ (Aktivitätsdiagramm)



Fachhochschule
Dortmund
University of Applied Sciences and Arts

UML Aktivitätsdiagramm

Aktivität

Definition

Eine **Aktivität** beschreibt die Ausführung von Funktionalität bzw. Verhalten.

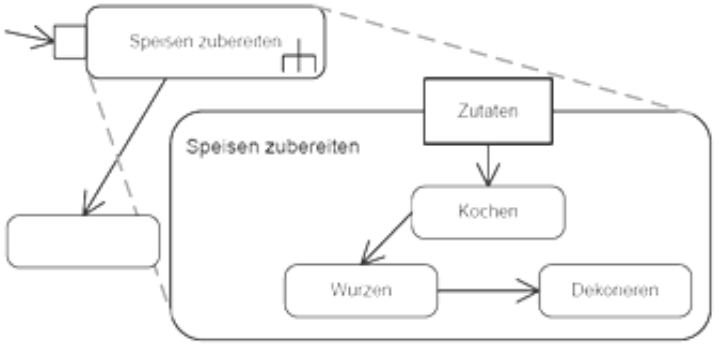
- in der Analyse werden Aktivitäten verwendet für
  - die Modellierung von Workflows oder
  - die Spezifikation von Use-Cases
- Aktivität wird durch mehrere Knoten modelliert, die durch gerichtete Kanten miteinander verbunden sind
- es lassen sich Aktionsknoten, Kontrollknoten und Objektknoten unterscheiden
- Aktivitäten besitzen sowohl
  - ein Kontrollflussmodell (→ Reihenfolgen von Aktionen) als auch
  - ein Datenflussmodell (→ Austausch der Daten zwischen den Aktionen)

Aktion

Definition

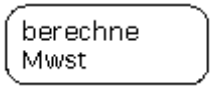
Eine **Aktion** ist die kleinste ausführbare Einheit innerhalb einer Aktivität.

- eine Aktion wird ausgeführt wenn:
  - die Vorgänger-Aktion beendet ist,
  - notwendige Daten zur Verfügung stehen oder
  - ein Ereignis auftritt

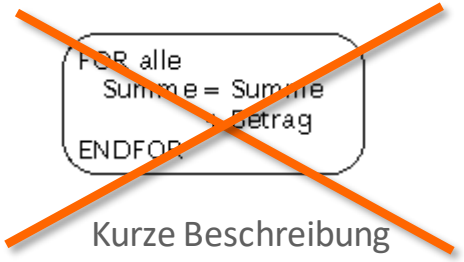


- eine Aktion ist:
  - ein elementarer Verarbeitungsschritt oder
  - ein Aktivitätsaufruf (→ Sprung zu ggf. sehr komplexer Verarbeitung)

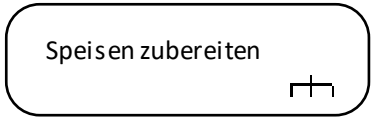
- Notation für Aktionen



Name



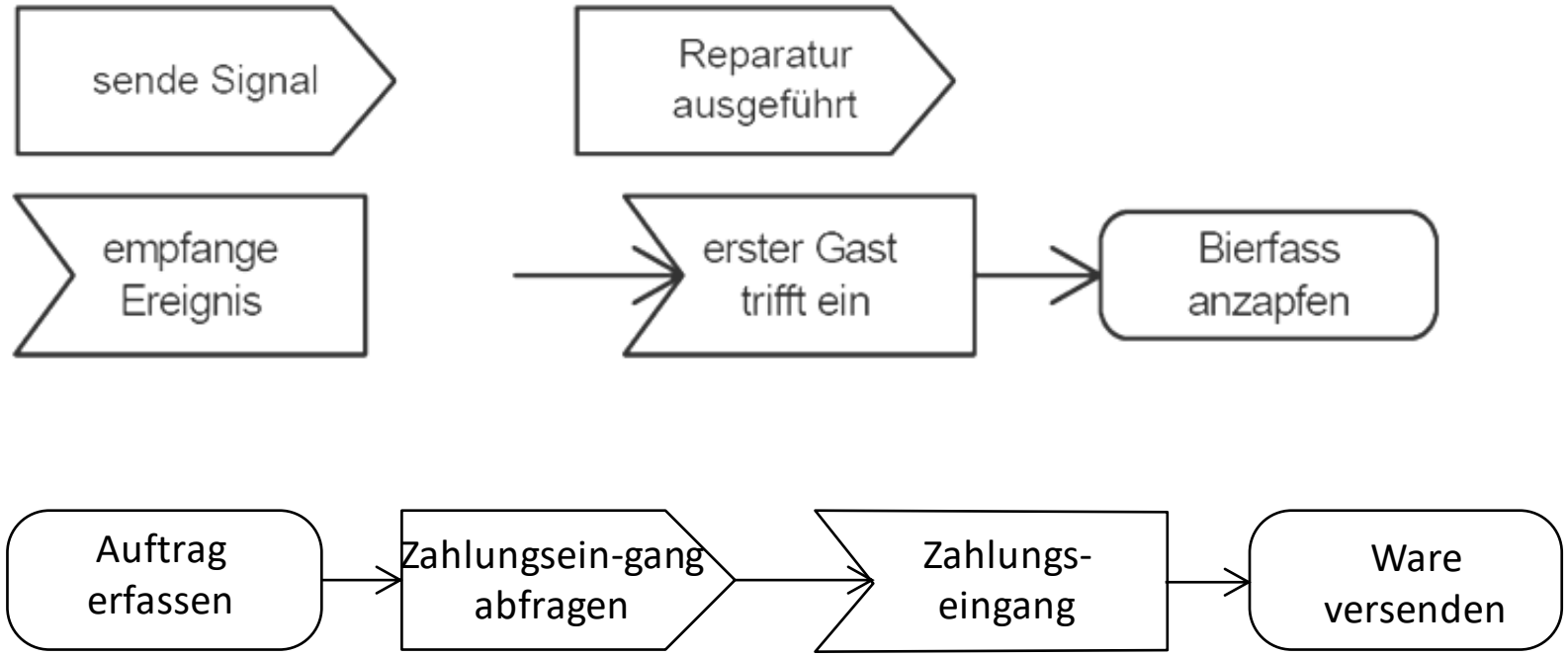
Kurze Beschreibung



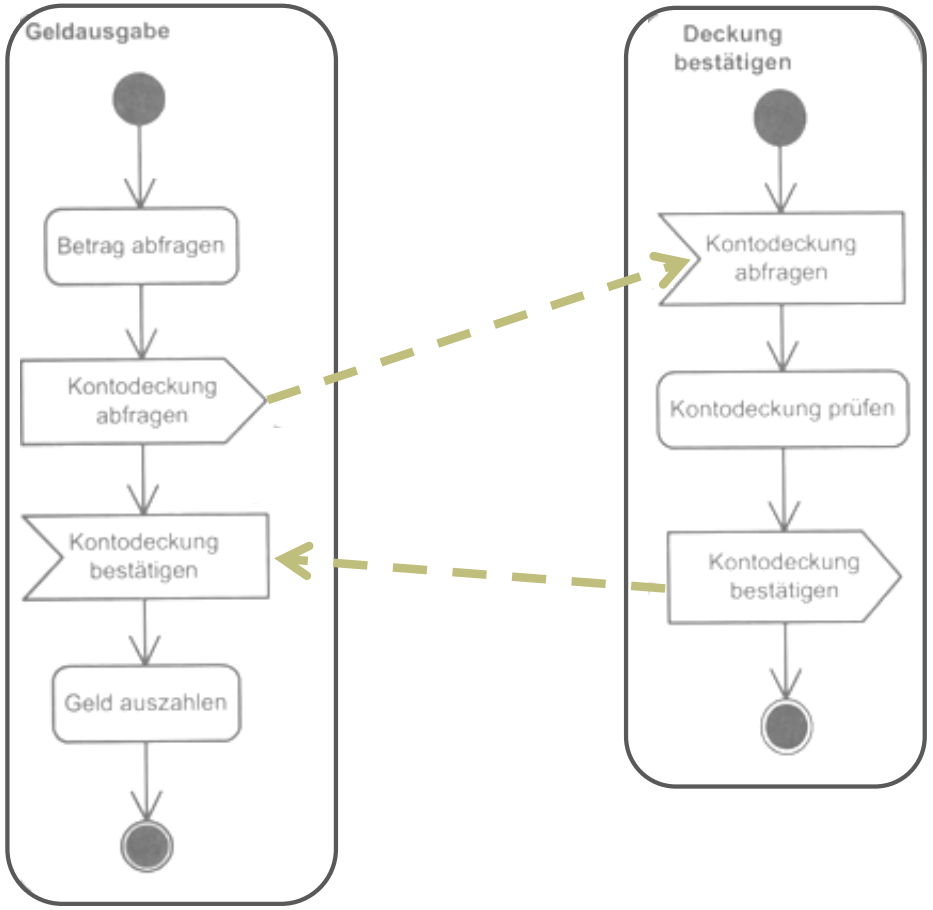
Aktivitätsaufruf

Signale und Ereignisse

- sind Sonderformen der Aktion
- es werden **Signalsender** (→SendSignalAction) und **Ereignisempfänger** (→AcceptEcentAction) unterschieden



Signale und Ereignisse - Beispiel



Signalsender

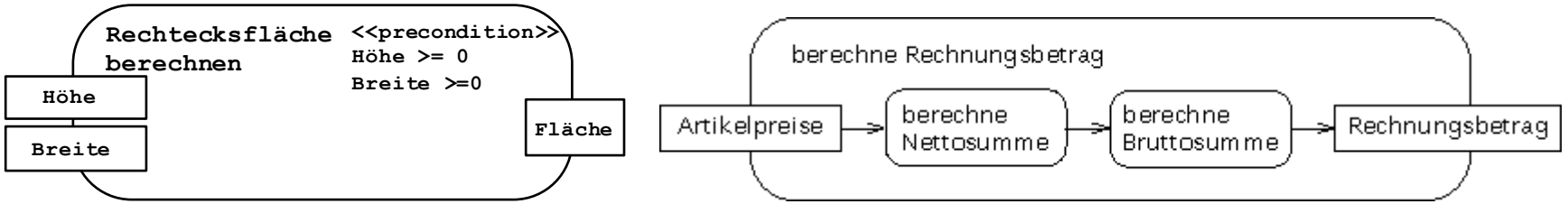
Ereignis-empfänger

Hinweis:
Zeitgetriebene Ereignisempfänger

Aktivitätsdiagramme, Parameter und Rückgabewerte

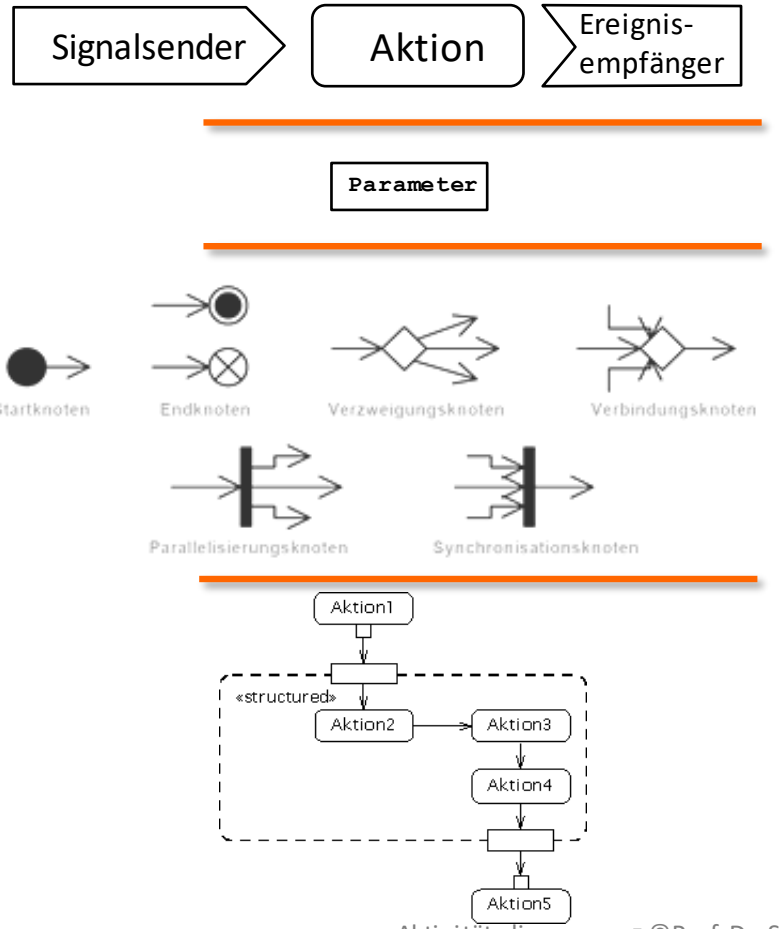
- eine Aktivität wird in einem Aktivitätsdiagramm modelliert
- mögliche Parameter werden durch Objektknoten auf der Grenze dargestellt

Beispiele

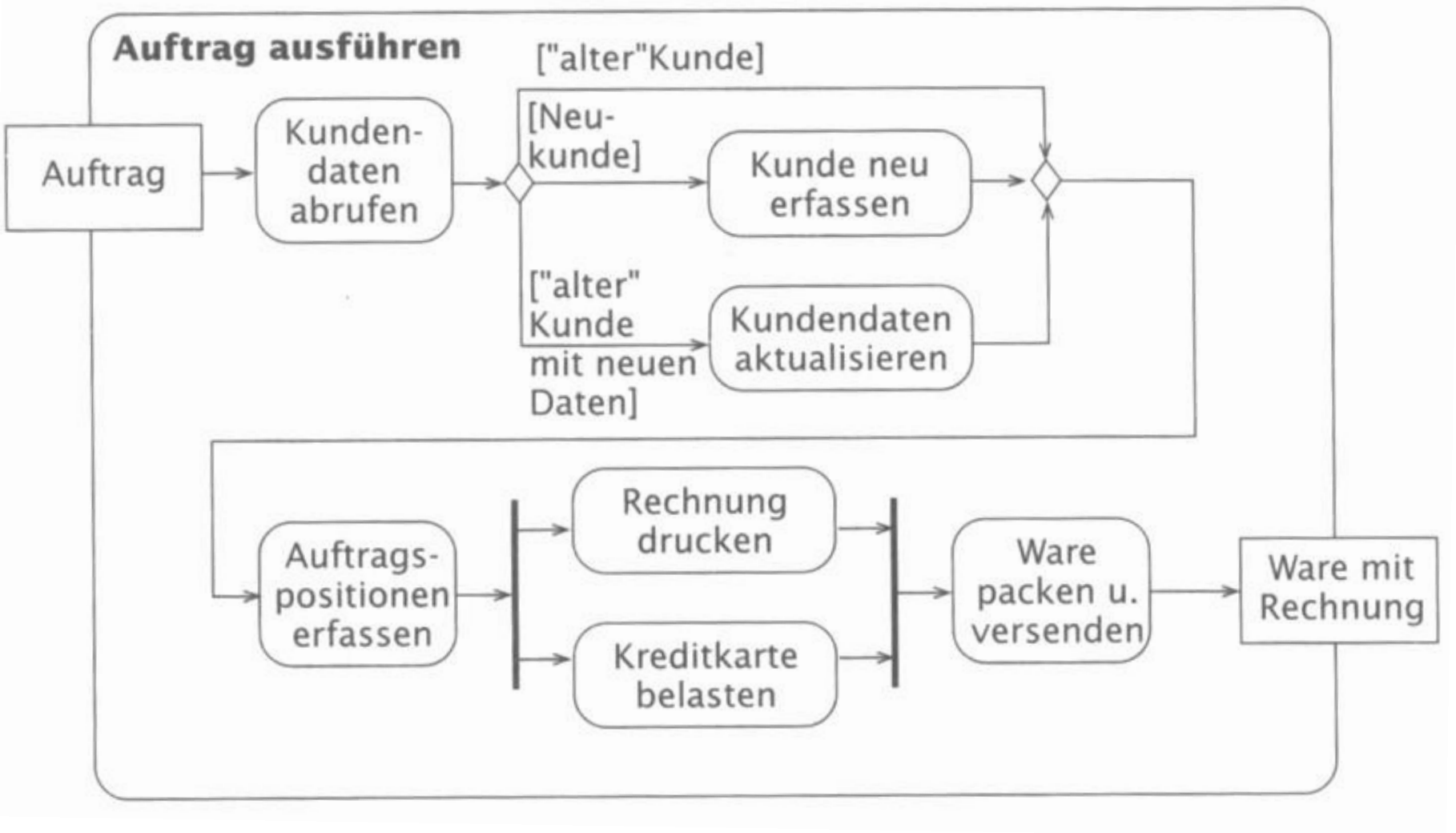


Aktivitätsdiagramm - Knoten

- Es lassen sich Aktionsknoten, Kontrollknoten, Objektknoten und Strukturknoten unterscheiden
- Aktionsknoten
  - elementare Aktion
  - Aufruf einer Aktivität
- Objektknoten
  - Weiterreichung von Daten von einer Aktion zur nächsten
  - Häufig mit dem Namen der Klasse benannt
- Kontrollknoten
  - Entscheidung und Zusammenführung
  - Splitting und Synchronisation
  - Start- und Endknoten
- Strukturknoten
  - Gruppiert Knoten und Kanten einer Aktivität
  - Darstellung:
    - gestricheltes Rechteck mit abgerundeten Ecken und
    - links oben Schlüsselwort «structured»

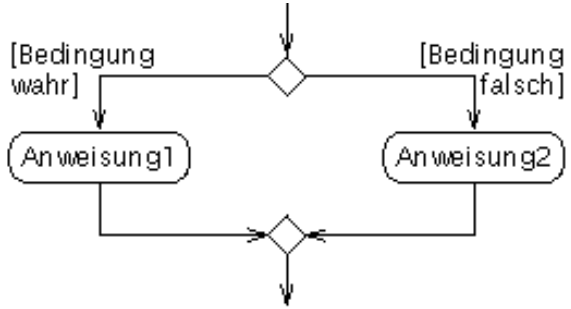


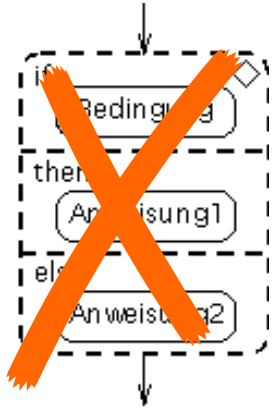
Aktivitätsdiagramm - Beispiel



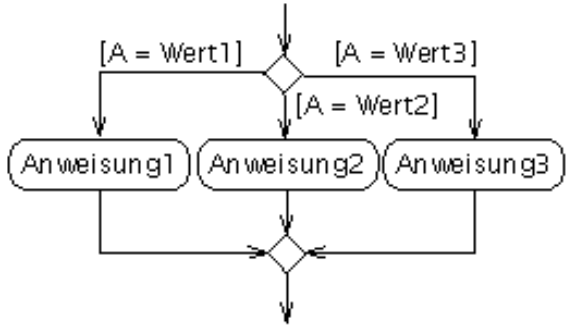
Strukturelemente: Entscheidungsknoten

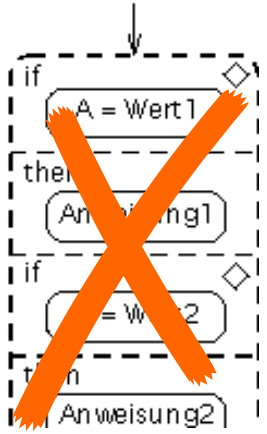
```
if (Bedingung)
   Anweisung1
else
   Anweisung2
```





```
switch (A)
Wert1: Anweisung1;
Wert2: Anweisung2;
Wert3: Anweisung3;
```



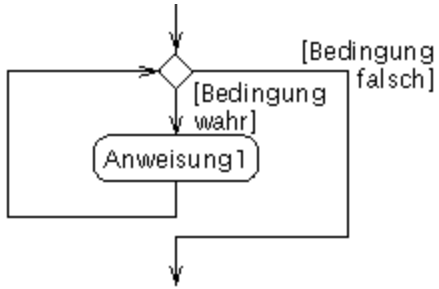


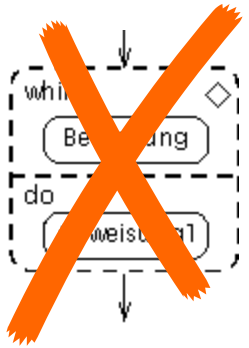
Kontrollknoten

Strukturknoten

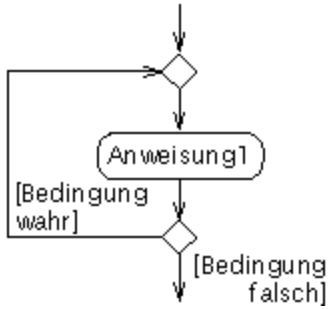
Strukturelemente: kopf- und fußgesteuerte Schleifenknoten

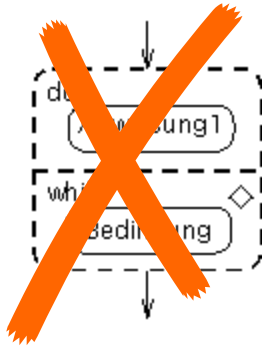
```
while (Bedingung)
   Anweisung1;
```






```
do
   Anweisung1
while (Bedingung);
```



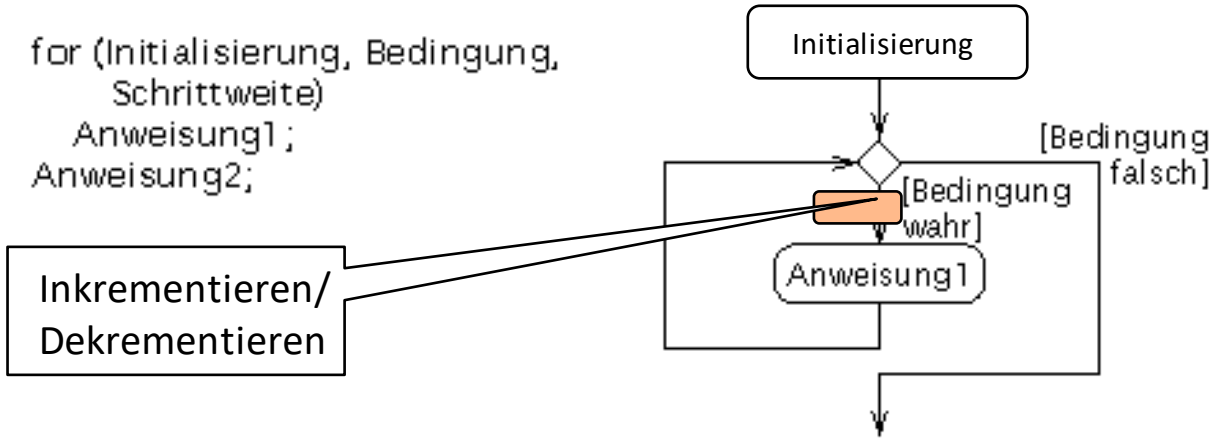


Kontrollknoten

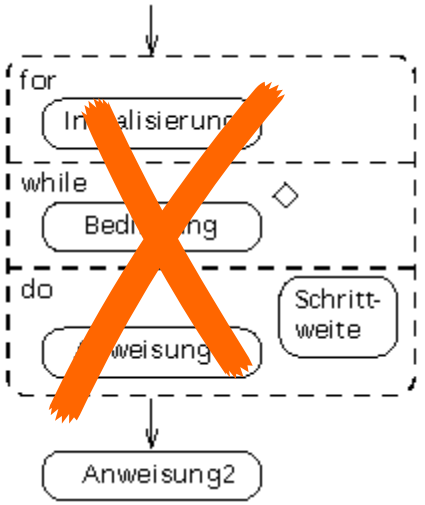
Strukturknoten

Strukturelemente: Zählscheifenknoten

```
for (Initialisierung, Bedingung,
      Schrittweite)
   Anweisung1;
Anweisung2;
```



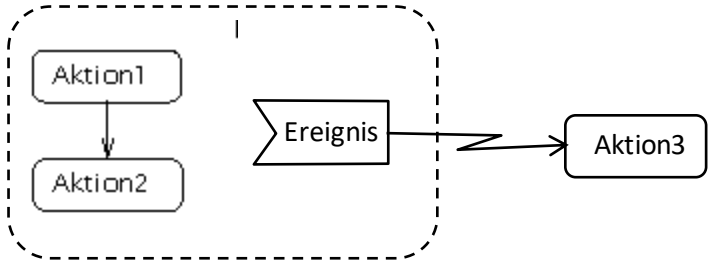
Kontrollknoten



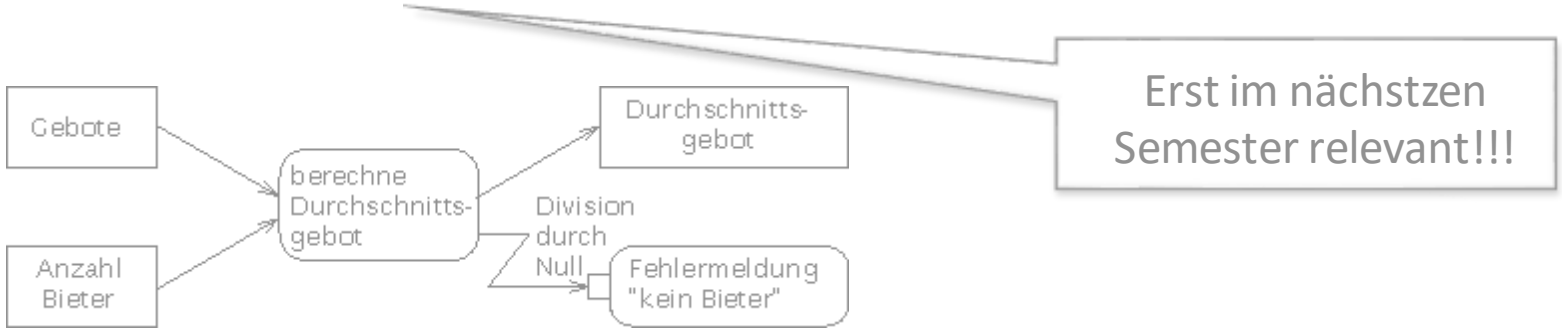
Strukturknoten

Unterbrechungsbereich und Exception Handler

Der Unterbrechungsbereich ermöglicht es, einen Bereich, der mehrere Aktionen umfasst, zu unterbrechen

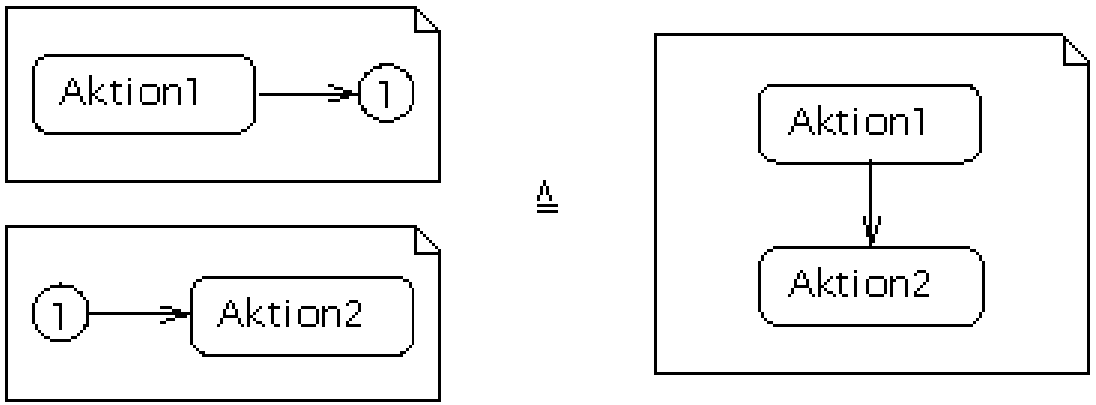


Der Exception Handler spezifiziert, welche Anweisungen auszuführen sind, falls ein bestimmter Fehler auftritt



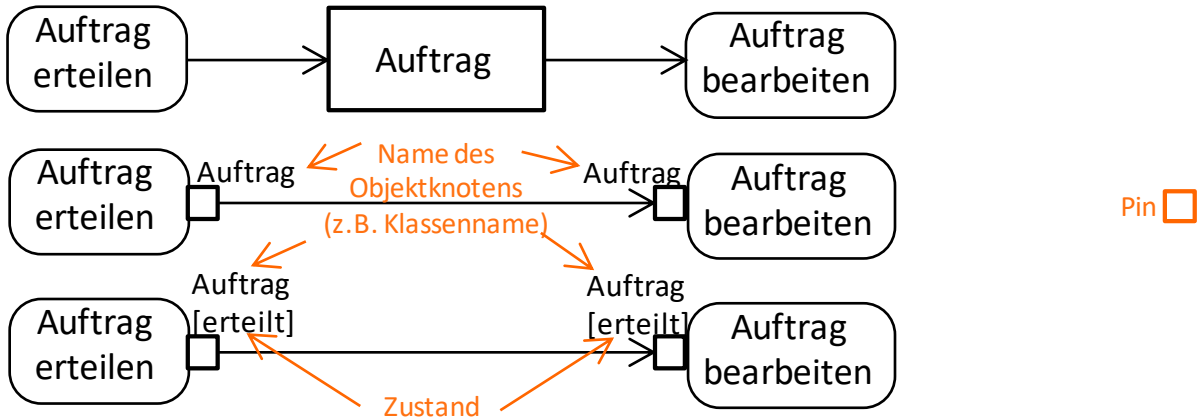
Konnektor

Ermöglicht es, eine Kante zu unterbrechen und an beliebiger Stelle fortzuführen



Objektfluss

- Mit Hilfe von Objektknoten können Daten von einer Aktion zur nächsten gereicht werden → Objektfluss → Datenmodell einer Aktivität
- Kante trägt Token, der Daten zu oder von Objektknoten transportiert
- Darstellung: Pfeile, die an mindestens einem Ende einen Objektknoten besitzen.
- Objektknoten können alternativ zur Rechteck-Notation als »Pins« modelliert werden.
- Sollten zwischen zwei Aktionen mehrere Objektflüsse stattfinden, so kann man auf beiden Seiten der Aktionen eine ganze Reihe von Pins eintragen.



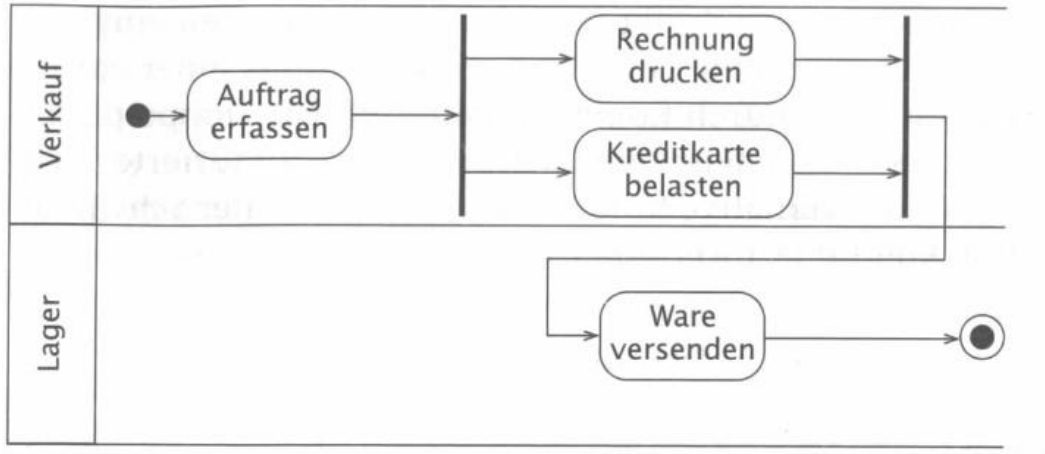
Objektfluss

- Daten, die in einem Datenspeicher liegen, also persistente Daten, können ebenfalls modelliert werden.

- Objektknoten können in einem *Streaming*-Modus verwendet werden, d.h. eine Aktion wird kontinuierlich ausgeführt und produziert laufend neue Objekte.
  Ebenso können Objekte auch laufend verbraucht werden.
- Modellierungsalternativen für den *Streaming*-Modus

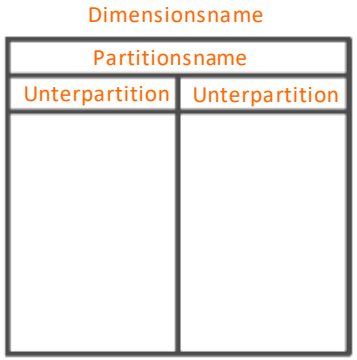
Aktivitätsbereiche

- Aktivitätsbereiche fassen Aktionen zusammen, die bestimmte Gemeinsamkeiten besitzen, z.B.
  - organisatorische Einheiten,
  - Standort,
  - Rolle oder
  - Verantwortungsbereich
- Aktivitätsbereiche werden auch als Verantwortlichkeitsbereiche oder Partitionen bezeichnet.
- Darstellung: parallele Linien und Boxen am Ende des Bereichs
  → Schwimmbahnen (engl. swimlanes)



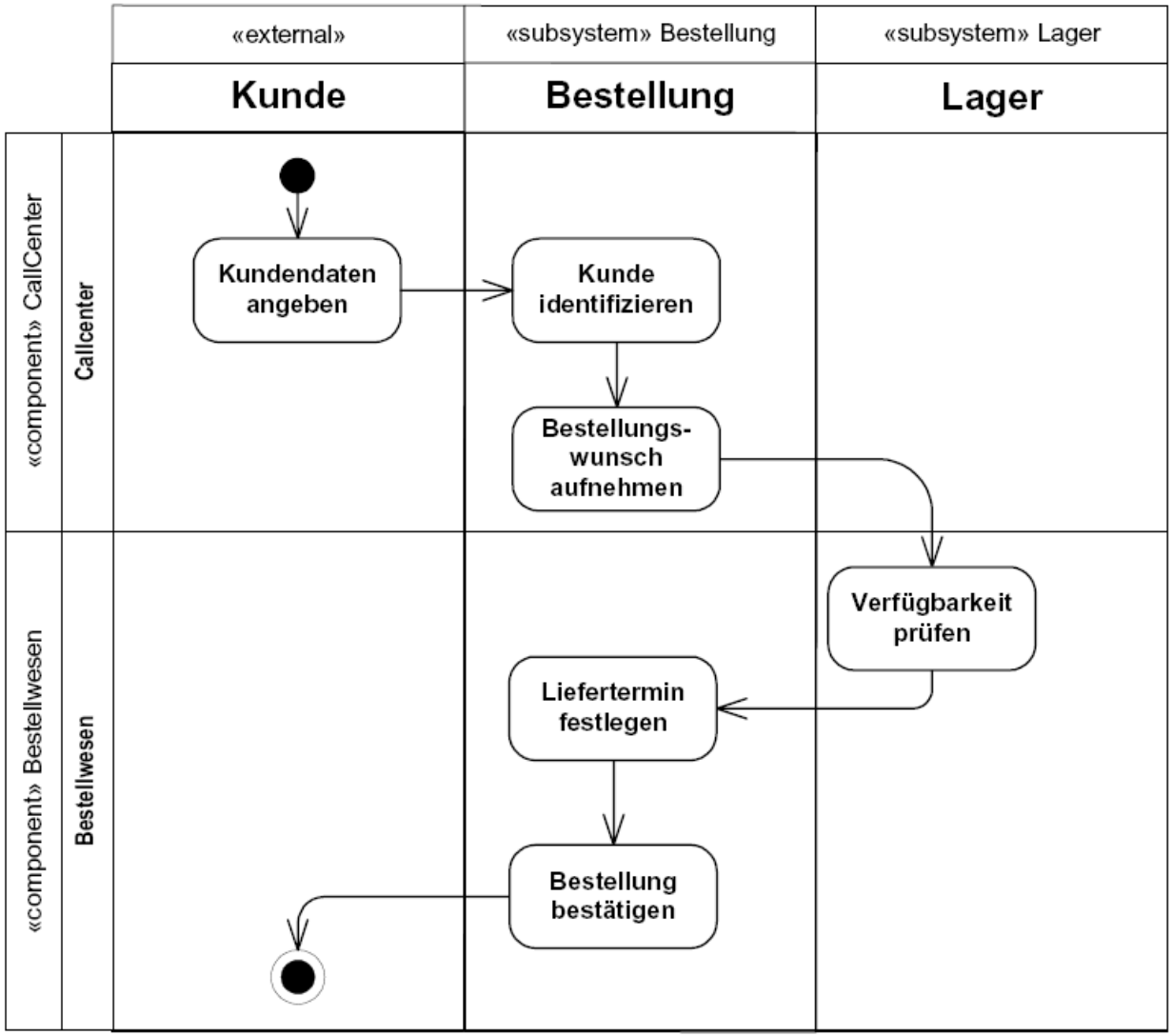
Aktivitätsbereiche

- Aktivitätsbereiche können hierarchisch strukturiert werden



- Aktivitätsbereiche für externe Aktionen ⇨ Stereotyp «external»

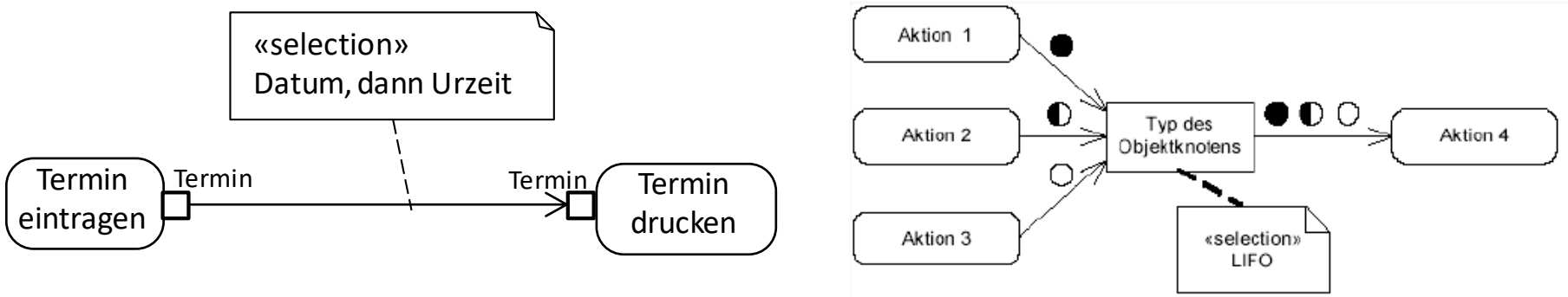
Aktivitätsbereiche



Kantenbeschriftung – Bedingung, Sortierung, Gewichtung

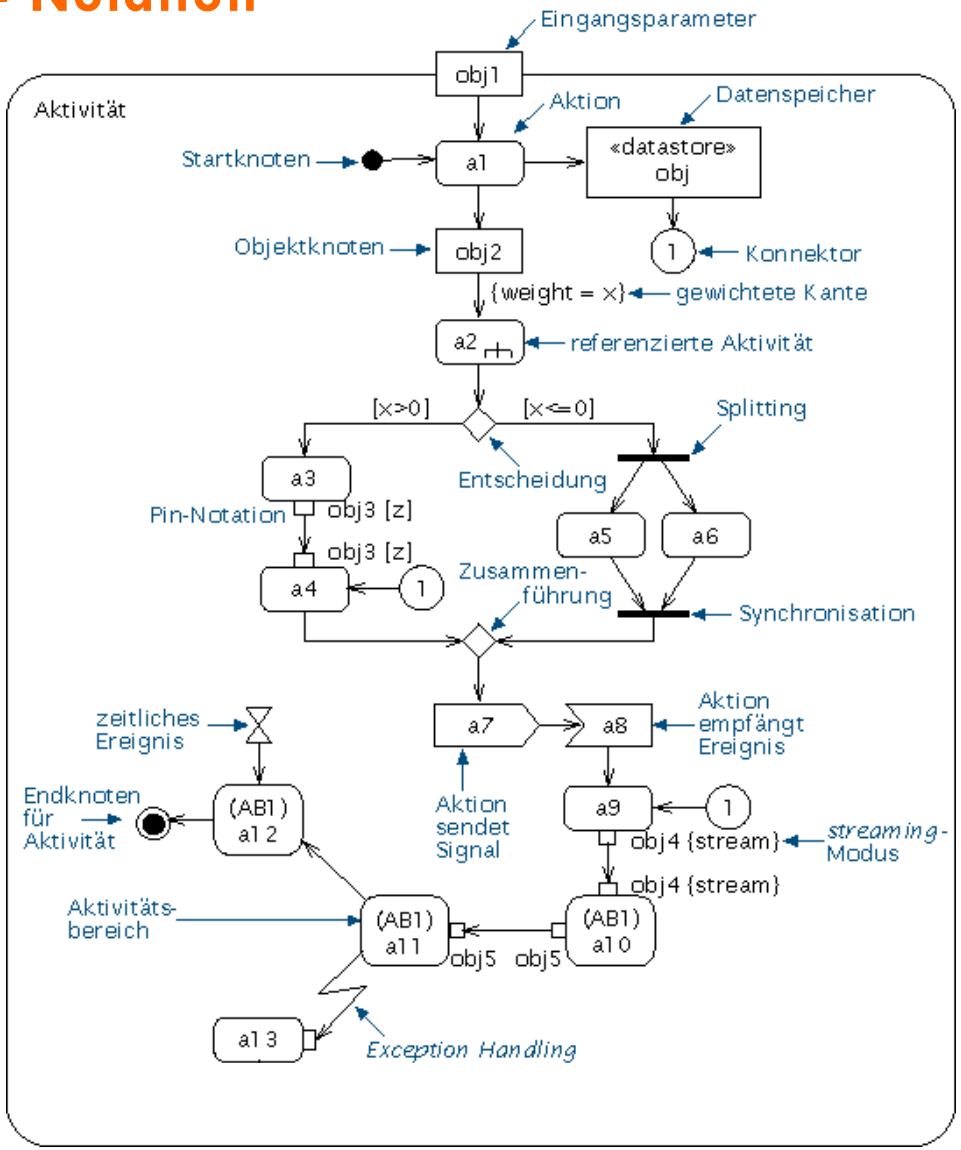
- Kanten können mit Bedingungen belegt werden

- Sortierung von Datentoken um Reihenfolge auf Kante festzulegen



- Gewichtung von Kanten um zu modellieren, wie viele Token für einen Ablauf benötigt werden → Token-Konzept

Aktivitätsdiagramm - Notation

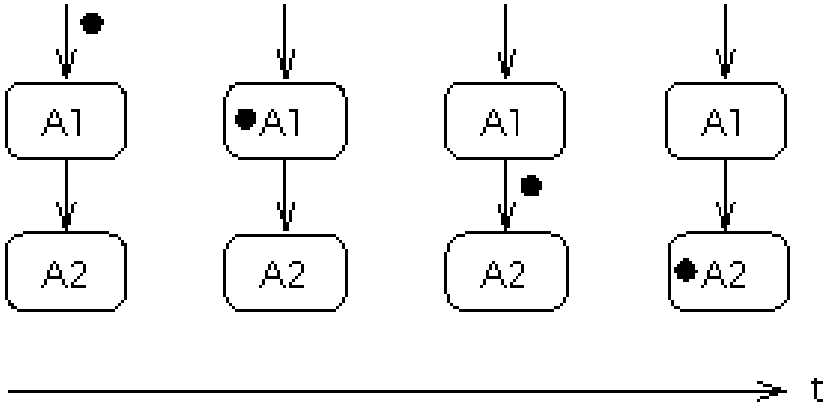


Token-Konzept

Ziel: dynamisches Verhalten bereits in der Analysephase zu simulieren

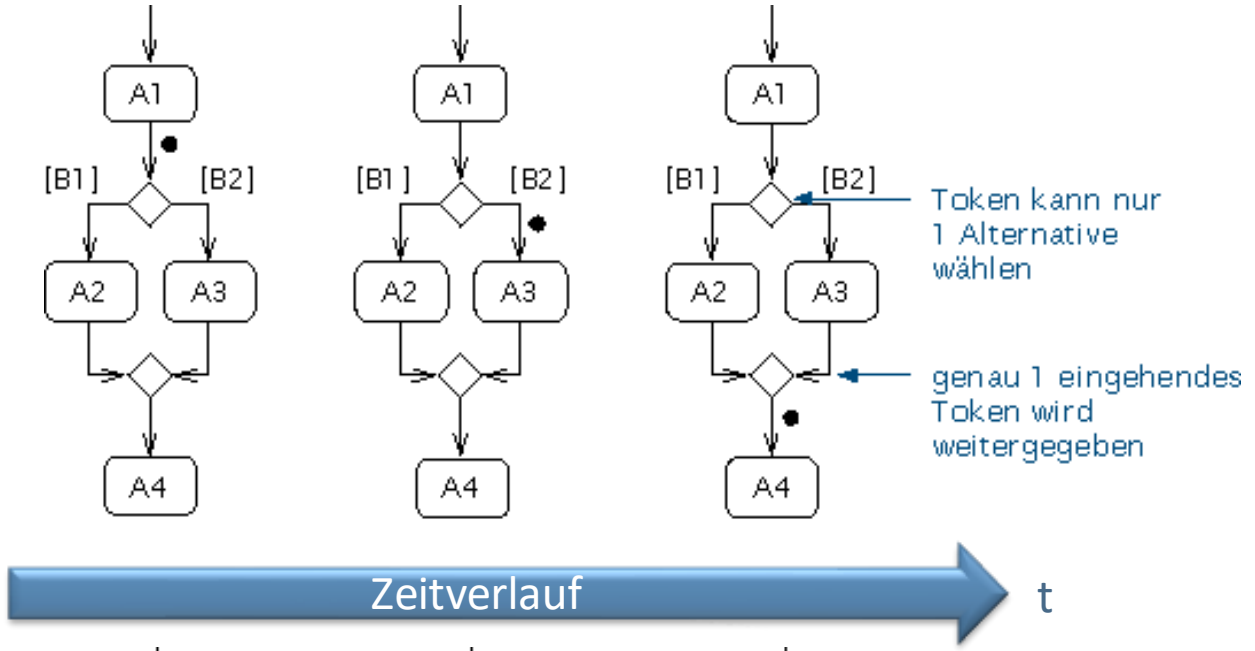
Token

- Marke, die sich nach bestimmten Regeln durch das Aktivitätsdiagramm bewegt
- Kanten und Knoten können mit Token behaftet sein
- Knoten bestimmen, wann ein Token angenommen wird und wann es den Knoten verlassen darf
- Regeln von Kanten kontrollieren, wann ein Token von einem Ausgangsknoten entfernt und einem Zielknoten übergeben wird

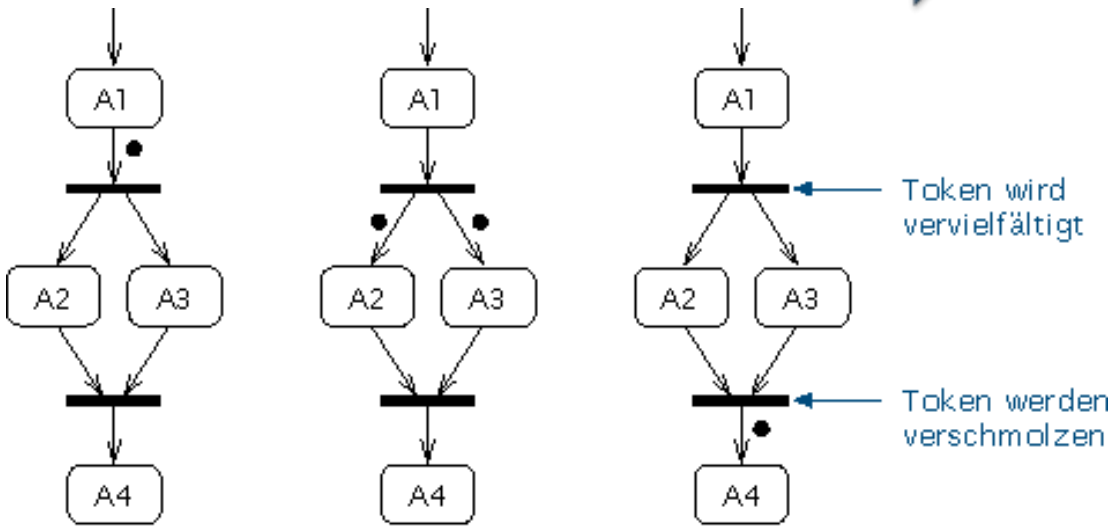


Token-Konzept

Entscheidung und Zusammenführung

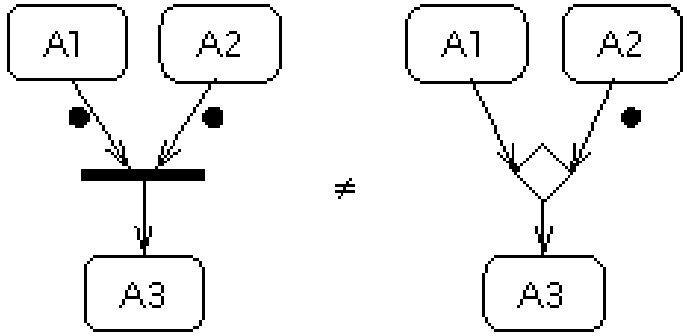


Splitting und Synchronisation



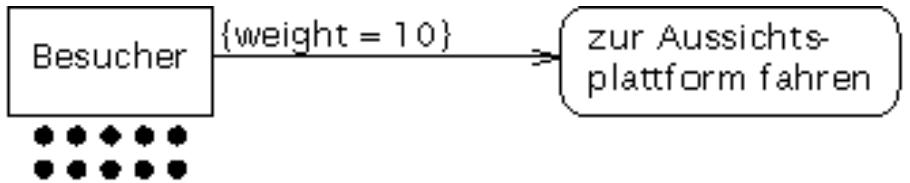
Token-Konzept

Unterschiedlicher Token-Fluss bei Zusammenführung von Kanten



Gewichtete Kanten

- Können für Objektflüsse definiert werden
- Gewicht definiert, wie viele Token vorliegen müssen, damit die nachfolgende Aktion ausgeführt wird



Exkurs: „Petri-like semantics“

Klassisches Vorgehen

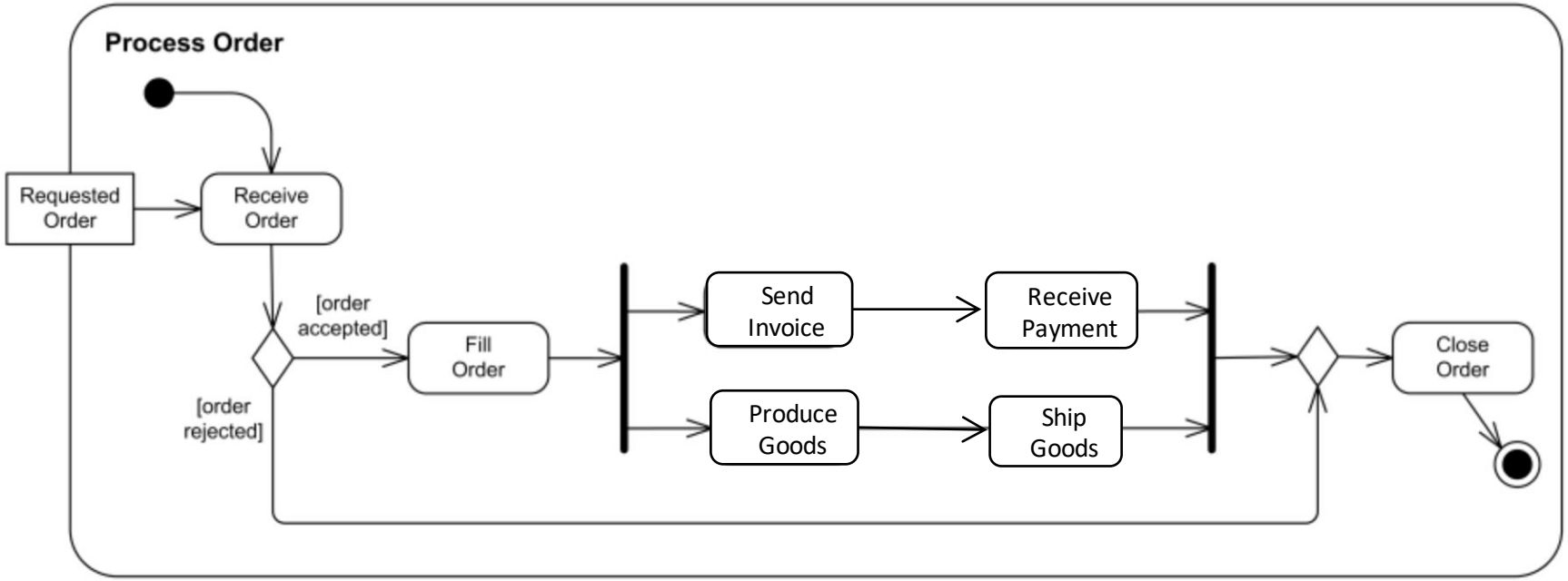
- Abbildung von Elementen der abstrakten Syntax auf eine formale Domäne, hier von Aktivitäten nach Petrinetzen, wobei:
  - ausführbare Knoten = Transitionen
  - Objektknoten = Stellen
  - Kanten = Flussrelation

Petri Netze ermöglichen die Analyse, Modellbildung und Simulation von dynamischen Systemen mit nebenläufigen und nichtdeterministischen Vorgängen. Die Petri Netze sind benannt nach dem deutschen Mathematiker Carl Adam Petri, der diese ab 1962 einführte.

Vorteile

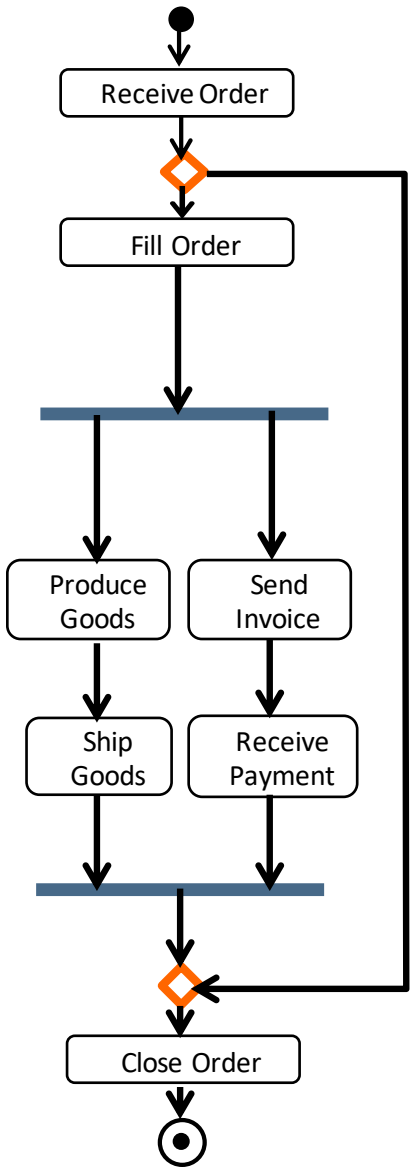
- Abbildung ist nachvollziehbar und intuitiv
- ausgereifter Formalismus mit reicher Theorie
- zahlreiche Werkzeuge verfügbar

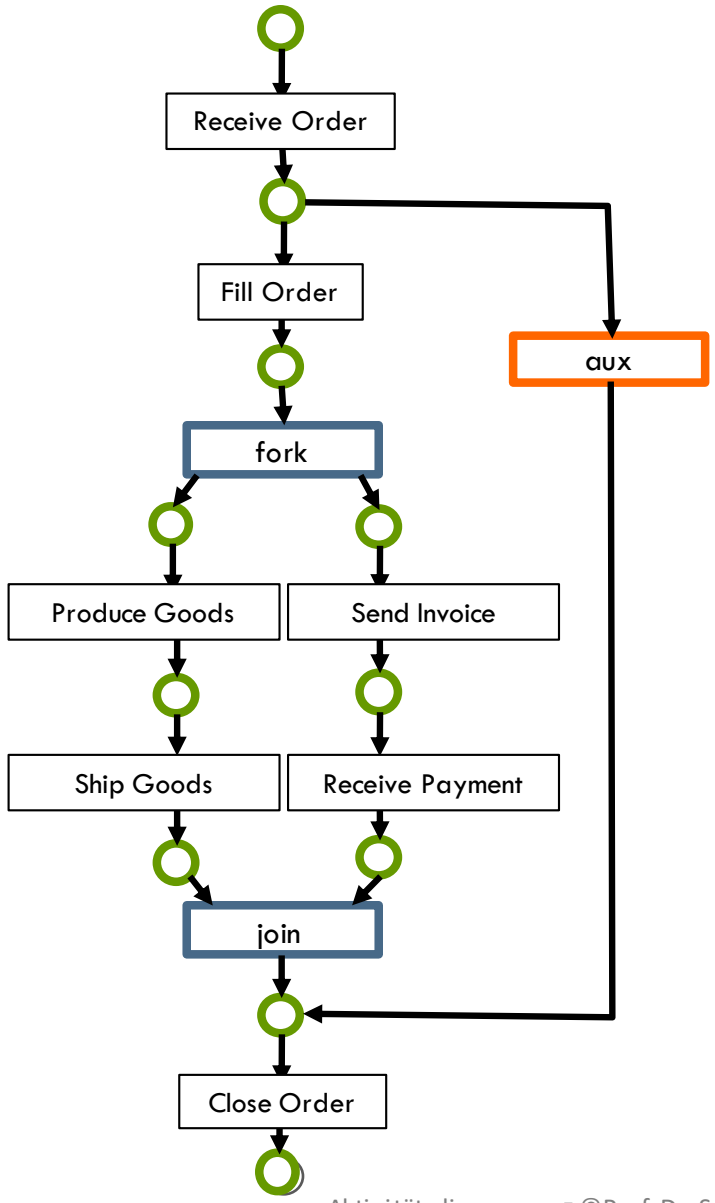
Beispiel: Process Order



Exkurs:

„Petri-like semantics“



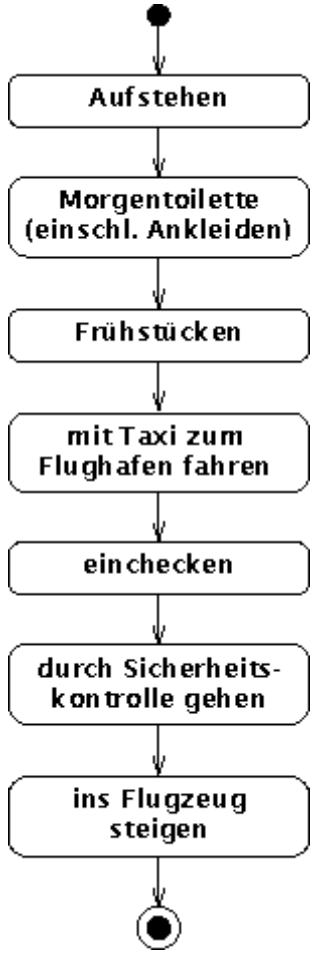


Methodische Vorgehensweise: Aktivität

Schritte zum Modellieren von Aktivitäten

- Welche Ereignisse lösen die Verarbeitung aus?
  - Startknoten links oben eintragen
  - Vorbedingungen (preconditions) ermitteln
- Wann terminiert die Verarbeitung?
  - Welches Ziel soll im Erfolgsfall erreicht werden?
  - Terminiert die gesamte Verarbeitung oder nur der aktuelle Pfad?
- Wie sieht der Standardfall aus?

Beispiel: Aktivitätsdiagramm für den optimalen Fall eines Starts in den Urlaub

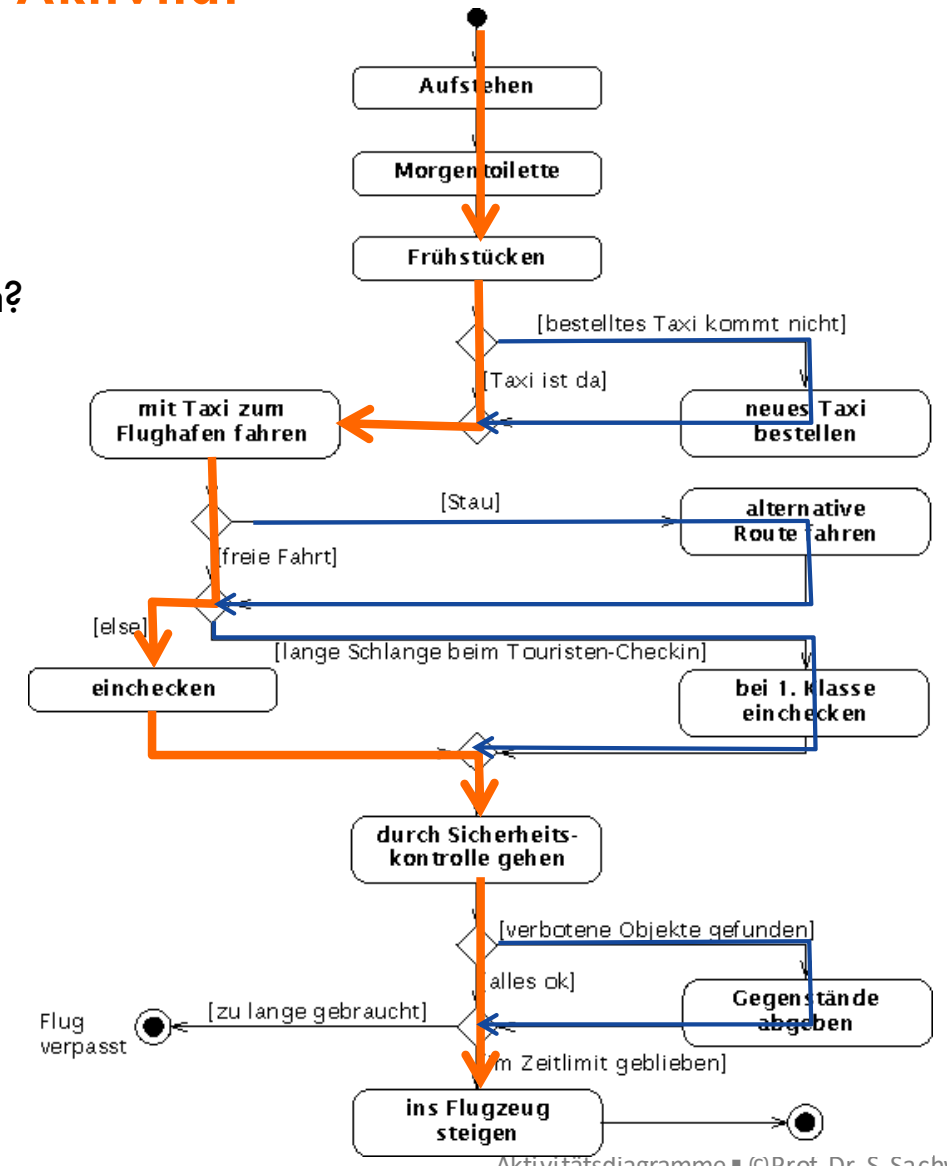


Methodische Vorgehensweise: Aktivität

Schritte zum Modellieren von Aktivitäten

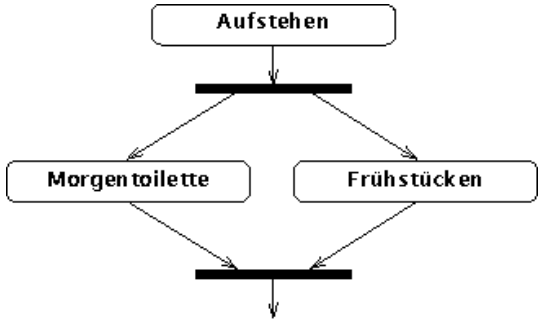
- Welche Erweiterungen zum Standardfall sind möglich?
  - Werden Aktionen nur unter bestimmten Bedingungen ausgeführt?
  - Tragen Sie Bedingungen an die Ausgangs-pfeile der Raute ein.
  - Wo wird der Kontrollfluss wieder zusammengeführt?

Beispiel: Erweiterung mit den wichtigsten Sonderfällen



Methodische Vorgehensweise: Aktivität

Beispiel: Erweiterung um parallele Verarbeitung



Schritte zum Modellieren von Aktivitäten

- Ist parallele Verarbeitung möglich?
  - Ist für bestimmte Verarbeitungsschritte die Reihenfolge irrelevant?
  - Ist echte Parallelarbeit möglich?
  - Wo wird der Kontrollfluss wieder synchronisiert?
- Objektknoten
  - Erzeugen die Aktionen Ausgabedaten oder benötigen sie Eingabedaten?
  - Tragen Sie Objektknoten ein, um die Aussagefähigkeit des Diagramms zu erhöhen
  - Bei Objekten, die in verschiedenen Verarbeitungszuständen vorkommen, ist zusätzlich der jeweilige Zustand einzutragen
- Ein-/Ausgabeparameter
  - Die Angabe von Ein- und Ausgabeparametern ist nur notwendig, wenn sie zusätzliche Informationen liefern.
- Aktivitätsbereiche
  - Verwenden Sie die Schwimmbahn-Notation, wenn die Aktivität höchstens 5 Bereiche enthält.
  - Modellieren Sie die wichtigsten Schritte im ersten Aktivitätsbereich
  - Knoten, die zwei Bereiche betreffen, werden auf der Grenzlinie dargestellt

www.fh-dortmund.de

facebook.com/fhdortmund instagram.com/fhdortmund twitter.com/fh_dortmund youtube.com/FachhochschuleDO