

# 7 Aktivitätsdiagramm

## Inhalt

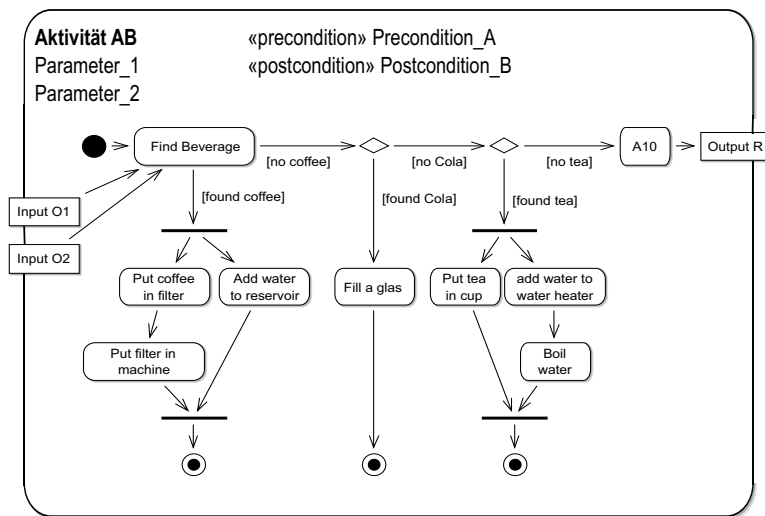
7.1	Allgemeins .....	-2
7.2	Kontrollfluss .....	-4
7.3	Nebenläufige Abläufe - Parallelisierung .....	-7
7.4	Alternative Abläufe durch Entscheidungsknoten .....	-8
7.5	Verantwortlichkeitsbereiche (Partitionen, Swimlanes).....	-12
7.6	Schachtelung von Aktivitäten .....	-14
7.7	Objektfluss .....	-15
7.8	Signale und Ereignisse .....	-17
7.9	Exception Handling - Ausnahmebehandlung .....	-17

## 7.1 Allgemeins

- ❑ Fokus liegt auf prozeduralen Verarbeitungsaspekten
- ❑ Spezifikation von **Kontroll- und/oder Datenfluss** zwischen Arbeitsschritten (Aktionen) zur Realisierung einer Aktivität
- ❑ Aktivitätsdiagramm in UML 2.x:
  - ablauforientierte Sprachkonzepte basierend u.a. auf Petri-Netzen (siehe später)
  - Sprachkonzepte und Notationsvarianten ermöglichen breite Anwendung
  - Modellierung objektorientierter und nichtobjektorientierter Systeme ist möglich
- ❑ Eine **Aktivität** ist ein **gerichteter Graph**
  - **Knoten** repräsentieren Aktionen
  - **Kanten** repräsentieren Kontroll- und Datenflüsse
  - **Token** durchlaufen nach einem bestimmten Schema die Graphen
- ❑ Kontroll-und Datenflüsse legen (potentielle) Abläufe von Arbeitsschritten (Aktionen) fest
- ❑ Vollständige **Spezifikation der Konzepte siehe [OMG 2015]**; Literaturangaben!)

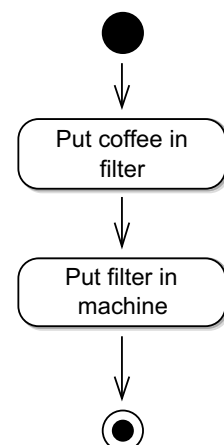
- ❑ Beispiel einer Aktivität mit Input-Parametern, pre-condition und post-condition, und einem Objekt als Resultat der Aktivität

- ❑ Parameter sind optional
- ❑ Pre-condition und Post-condition sind optional
- ❑ Resultat-Objekte sind optional

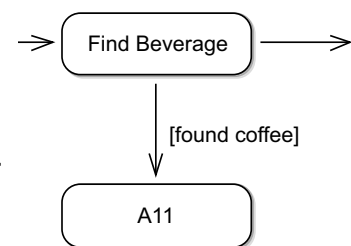


## 7.2 Kontrollfluss

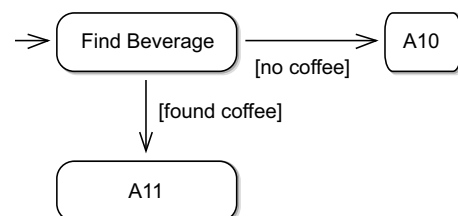
- ❑ Aktivitätsdiagramme ermöglichen die Modellierung von **Kontrollfluss**.
- ❑ Die Modellierung von Prozessen erfolgt in einem Aktivitätsdiagramm durch die Angabe einer bestimmten **Abfolge von Aktionen**.
- ❑ Eine **Aktion** wird durch ein Rechteck mit konvexen Vertikalen dargestellt, das den Namen der auszuführenden Aktion enthält.
- ❑ Beginn und Ende des Ablaufs werden durch einen eindeutigen **Startzustand** und möglicherweise mehrere **Endzustände** markiert.
- ❑ Aktionen sind untereinander durch **Kontrollflusskanten** verbunden, welche die Ausführungsreihenfolge der einzelnen Schritte festlegen.



- ❑ Ein **Kontrollfluss** von einer Aktion A zur Aktion B wird durch einen (Übergangs-) Pfeil von A nach B dargestellt. Sobald die Ausführung von A abgeschlossen ist, wird automatisch der **Übergang (Transition)** nach B ausgelöst. Somit fungiert der Abschluss von A als impliziter Trigger für die Ausführung von B.
- ❑ Üblicherweise werden in Aktivitätsdiagrammen keine Ereignisse dargestellt, die als explizite Auslöser für Übergänge dienen.
- ❑ Jeder Aktion muss **mindestens ein eingehender** und **ein ausgehender Übergang** zugeordnet sein.
- ❑ Ein eingehender Übergang legt die **Vorgängeraktivität** fest. Ein ausgehender Übergang bestimmt die **Nachfolgeraktivität**.
- ❑ Eine Aktion kann beliebig viele eingehende und ausgehende Übergänge haben.
- ❑ Übergänge können mit einer **Überwachungsbedingung** versehen werden. Je nach erfüllter Bedingung ist es möglich, eine nachfolgende Aktion auszuführen.



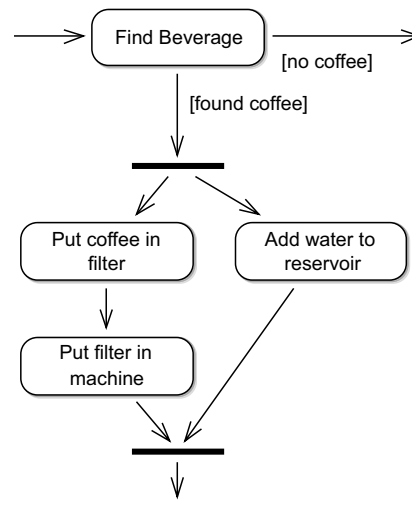
- ❑ I.d.R. ist es sinnvoll, die ausgehenden Übergänge mit einander **wechselseitig ausschließenden Bedingungen** zu überwachen, um so indeterministische Zustandsübergänge zu vermeiden. Diese bedingten Übergänge ermöglichen die Modellierung **alternativer Abläufe**.



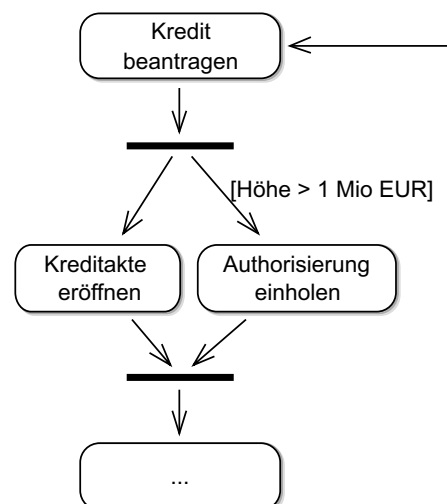
- ❑ Aktionen
  - sind elementare Bausteine für beliebige benutzerdefinierte Abläufe
  - sind atomar, können aber abgebrochen werden
  - können Eingabedaten zu Ausgabedaten verarbeiten
- ❑ Spezielle Notation für bestimmte Aktionsarten
- ❑ 44 vordefinierte Aktionen in UML:
  - Kommunikationsbezogene Aktionen (z.B. Signale und Ereignisse)
  - Objektbezogene Aktionen (z.B. Erzeugen und Löschen von Objekten)
  - Strukturmerkmals- und variablenbezogene Aktionen (z.B. Setzen und Löschen einzelner Datenwerte von Variablen)
  - Linkbezogene Aktionen (z.B. Erzeugen und Löschen von Links zwischen Objekten sowie Navigation)

### 7.3 Nebenläufige Abläufe - Parallelisierung

- Im Aktivitätsdiagrammen ist es auch möglich **nebenläufige Teilfolgen** von Aktionen zu beschreiben, die parallel abgearbeitet werden können. Dazu gibt es die Pseudozustände des **Parallelisierungsknoten** (-balken) und des **Synchronisationsknoten** (-balken).
- Der **Parallelisierungsknoten** zeigt den Beginn einer Nebenläufigkeit durch einen Balken an, von dem mehrere Übergänge wegführen.
- Die Vereinigung gibt das Ende einer Nebenläufigkeit an. Sie wird durch einen **Synchronisationsknoten** (-balken) dargestellt, der zwei oder mehrere unbedingte eingehende Übergänge besitzt.



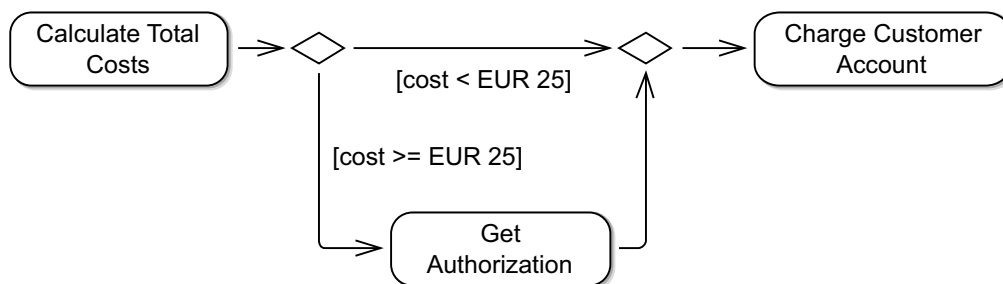
- Übergänge, die vom Parallelisierungsknoten ausgehen, können auch eine **Überwachungsbedingung** tragen. Sollte eine solche Bedingung zum Zeitpunkt des Schaltens des Übergangs nicht erfüllt sein, wird der zugehörige Zweig im Aktivitätsdiagramm nicht aktiviert.
- Sollte bei einem zugehörigen Parallelisierungsknoten auf Grund einer Überwachungsbedingung ein bestimmter Zweig nicht aktiviert worden sein, so ist dieser auch für die Vereinigung beim Synchronisationsknoten irrelevant und es wird nicht auf diesen Zweig gewartet.



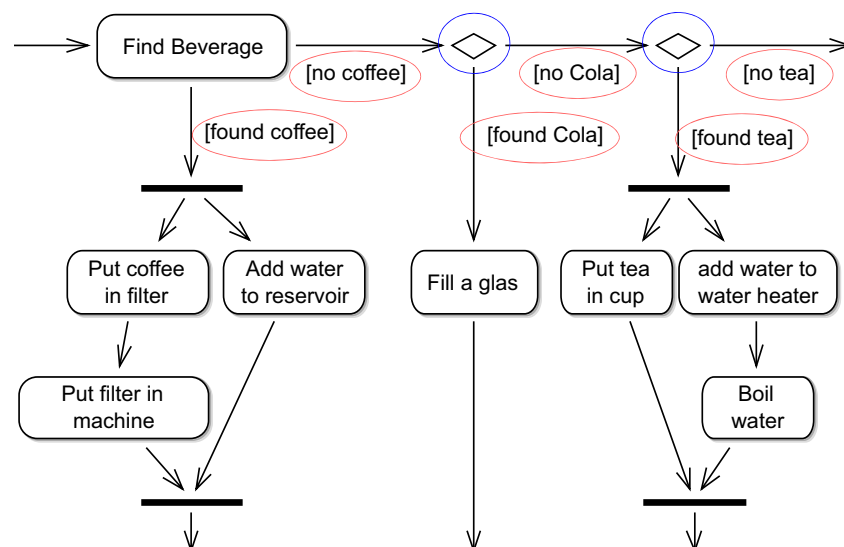
### 7.4 Alternative Abläufe durch Entscheidungsknoten

- Der **Entscheidungsknoten** ermöglicht es, alternative Abläufe explizit zu deklarieren und komplexe Entscheidungen in Form von Entscheidungsbäumen darzustellen.

- Einem Entscheidungsknoten sind eine oder mehrere eingehende Übergänge und zwei oder mehrere ausgehende Übergänge zugeordnet. (Allfällige Ereignisse dürfen nur auf eingehende Übergänge angegeben werden).
- Alle ausgehenden Übergänge müssen einander **wechselseitig ausschließende Bedingungen** tragen. Eine ausgehende Bedingung kann dabei ein Wildcard **[else]** tragen.
- Ein Kontrollfluss, der durch Entscheidungsknoten aufgespalten wurde, kann auch durch dasselbe Symbol wieder vereinigt werden. In dieser Rolle wird der Entscheidungsknoten als *Merge* bezeichnet.



- Im Falle von einfachen - also keinen geschachtelten Entscheidungen - ist es eine "Geschmacksfrage", ob unmittelbar ausgehende bedingte Übergänge oder ein dazwischengeschalteter Entscheidungsknoten mit ausgehenden bedingten Übergängen zur Spezifikation von alternativen Abläufen verwendet wird. Deutlicher erkennbar sind i.d.R. Entscheidungsknoten.



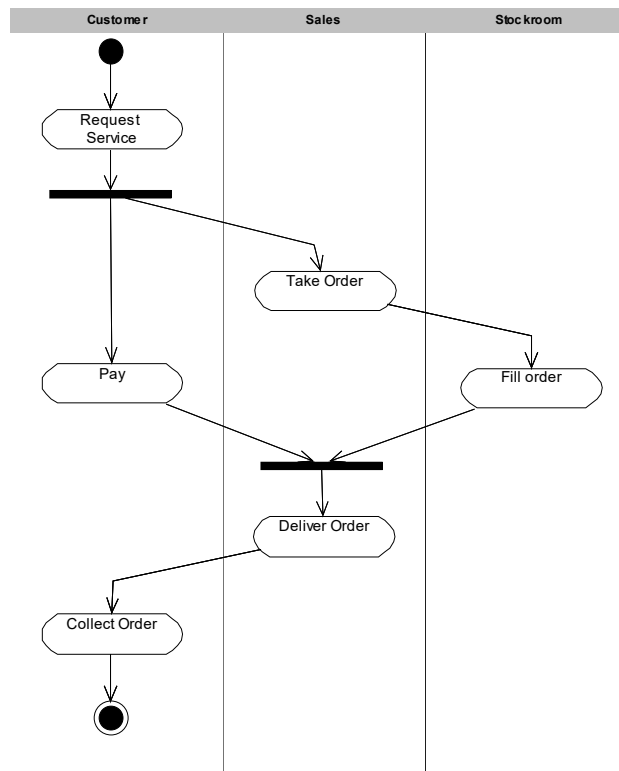
- Eine weitere Möglichkeit, den sequentiellen Kontrollfluss zu verzweigen und wieder zu vereinigen, stellen die **Verbindungsstellen** dar. Verbindungsstelle und Entscheidungsknoten sind in diesem Sinne im Wesentlichen äquivalent. Verbindungsstellen eignen sich aber besonders für die Verknüpfung von Aktionen, die aufgrund der Positionierung in einer (großen, vollen) Grafik optisch nicht einfach bzw. nicht auf direkter Linie verbunden werden können.

Verbindungsstellen werden hier nicht weiter behandelt.

(Details siehe Figure 15.19 und 15.20, Seite 382, in [OMG 2015])

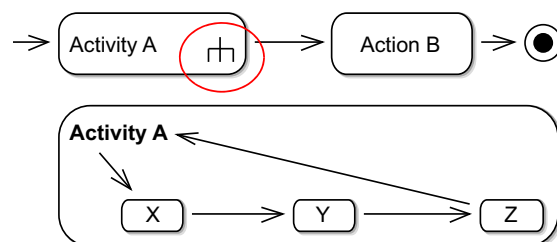
## 7.5 Verantwortlichkeitsbereiche (Partitionen, Swimlanes)

- Um die Aktionen in einem Aktivitätsdiagramm verantwortlichen Objekten zuzuordnen werden Verantwortlichkeitsbereiche (Partitionen, Swimlanes) genutzt.
- Einem Verantwortlichkeitsbereich sind einerseits bestimmte Aktionen und andererseits für die Ausführung der Aktionen verantwortliche Objekte zugeordnet. Diese Objekte sind entweder Objekte des zu realisierenden Systems, Akteure oder Organisationseinheiten eines Unternehmens, dessen Geschäftsprozesse beschrieben werden.
- Verantwortlichkeitsbereiche entsprechen Regionen im Aktivitätsdiagramm, die mit den jeweils verantwortlichen Objekten beschriftet und durch vertikale (oder horizontale) Linien voneinander getrennt werden. Ihre relative Lage zueinander ist bedeutungslos. Jede Aktion wird genau einem Verantwortlichkeitsbereich zugeordnet.



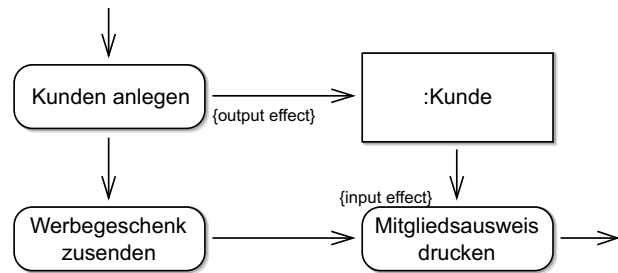
## 7.6 Schachtelung von Aktivitäten

- ❑ Im Aktivitätsdiagramm unterscheiden wir zwischen den Aktions- und den Aktivitätsknoten. Jedoch wird sowohl für Aktionen als auch Aktivitäten im Prinzip das gleiche Symbol verwendet. (Unterschiedliche Realisierung und Unterstützung durch Tools)
- ❑ Eine Aktion ist atomar. D.h. sie kann nicht weiter zerlegt und während ihrer Ausführung nicht unterbrochen werden.
- ❑ Die Aktion weist eine komplexe Struktur auf, die ein eigener Aktivitätsgraph beschreibt.
- ❑ Besitzt ein Knoten eine Verfeinerung, so handelt es sich konzeptionell um einen Aktivitätsknoten, ansonsten um einen Aktionsknoten.
- ❑ Die Verfeinerung kann graphisch in einem eigenen Diagramm oder aber auch innerhalb des Symbols des Aktivitätsknoten angegeben werden.



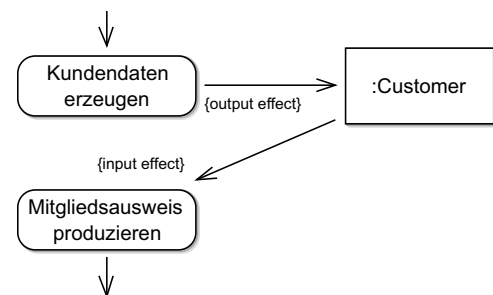
## 7.7 Objektfluss

- ❑ Aktivitätsdiagramme können auch zur Modellierung von Objektflüssen genutzt werden.



- ❑ Objekte können als Eingabe für eine Aktion verwendet werden oder sie werden von einer Aktion manipuliert und als Ausgabe erzeugt.

- ❑ Objekte werden als Rechteck mit Namen für das Objekt dargestellt. Sie werden mit den Aktionen durch eine gerichtete Kante vom/zum Objektsymbol zur jeweiligen Aktion verbunden.



- ❑ Sollte ein Objektfluss den Kontrollfluss zwischen zwei Aktionen vollständig erklären, so kann auf die graphische Darstellung des Kontrollflusses verzichtet werden.

- ❑ UML sieht **weitreichende Steuerungsmöglichkeiten** für Objektfluss vor:
  - **Reihenfolge** - wie werden Datenelemente an eine ausgehende Objektflusskante weitergegeben (FIFO, LIFO, ordered, unordered)
  - **Kapazitätsobergrenzen** und **Gewicht** - wieviele Datenelemente dürfen sich gleichzeitig in Objektknoten befinden; Anzahl der Datenelemente die vorliegen müssen, bevor eine Weitergabe erfolgt
  - **Selektionsverhalten** - wählt bestimmte Datenobjekte zur Weitergabe aus (z.B. aus Data Store auswählen, Filter anwenden, etc.)
  - **Transformationverhalten**
- ❑ Statt expliziter Objektknoten werden auch häufig **Eingabe-Pins** und **Ausgabe-Pins** an Aktionen für die Spezifikation von Objektfluss zwischen diesen Aktionen verwendet



## 7.8 Signale und Ereignisse

- ❑ Signale und Ereignisse sind Sonderformen von Aktionen
- ❑ Senden von Signalen
- ❑ Empfangen von Ereignissen
  - Spezielle Ereignisse (Zeitereignisse) vordefiniert

## 7.9 Exception Handling - Ausnahmebehandlung

- ❑ Vordefinierte Ausnahmen oder benutzerdefinierte Ausnahmen können spezifiziert werden. Nach der Ausnahmebehandlung kann die verlassene Aktion wieder weiter abgearbeitet werden.
- ❑ Ausnahmefälle können kaskadenartig von “innen” nach “außen” propagiert und behandelt werden.
- ❑ Unterbrechungsbereich, bestehend aus mehreren Aktionen, kann über eine Unterbrechungskante verlassen werden. Alle im Unterbrechungsbereich liegenden Aktionen werden abgebrochen (und alle entsprechenden Tokens gelöscht)



### Literatur und Quellen:

- ❑ [OMG 2015] OMG Unified Modeling Language™ (OMG UML), Version 2.5  
Normative Reference: <http://www.omg.org/spec/UML/2.5>  
OMG Document Number formal/2015-03-01
- ❑ Kapitel 4.2 in:  
Martin Hitz, Gerti Kappel, Elisabeth Kapsammer, Werner Retschitzegger:  
UML @ Work - Objektorientierte Modellierung mit UML2.  
dpunkt Verlag 2005 / 3. aktualis. u. überarb. Aufl. 2005.  
ISBN-13: 9783898642613 ISBN-10: 3898642615
- ❑ Tool zur Vertiefung: BEE-UP Modelling Tool:  
<http://austria.omilab.org/psm/content/bee-up/info>  
<http://www.omilab.org/web/guest/omilab-in-education/cmmc>