

SWE 1 - Spickzettel (Zusammenfassung der Inhalte)

WS 2021/22 Franjo Gießel



1. Modelle

- Modelle können...

↳ Beschreiben, entwerfen oder simulieren
↳ Abbild ↳ Vorbild

↳ abstrahiertes Abbild der Realität (deskriptiv) oder Vorbild für zu schaffende Realität (präskriptiv) sein.

→ Beispiel Warteschlange → First In, First Out (FIFO)

Papierstapel → First In, Last Out (FILO)

1.1 Drei Merkmale von Modellen (nach Stachowiak 1973)

1. Abbildungsmerkmal: Es gibt ein Original zum Modell

2. Verkürzungsmerkmal: Modell erfasst einen Ausschnitt des Originals

3. Pragmatisches M.: Ausschnitt wird abhängig des Zwecks des Modells gewählt.

1.2 Bäume

- Wachsen in der Informatik von oben nach unten

- Können hierarchische Strukturen abbilden

- Allgemein: Bäume sind Modelle für bestimmte (logische) Beziehungen zwischen Elementen

- Beispiele: Dateistruktur auf Festplatte
Berechnung eines math. Thems

2. UML-Diagramme (Unified Modeling Language)

- Sollen helfen, bei der Softwareentwicklung den Überblick zu behalten und Kommunikation zwischen den Partnern zu vereinfachen.

- UML ist objektorientiert

- Einheitliche Notation, Standardisierung

- Verschiedene Diagrammtypen ergänzen sich gegenseitig

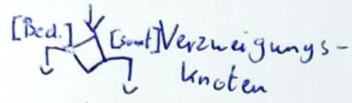
2. UML-Diagramme (Forts.)

● Startknoten

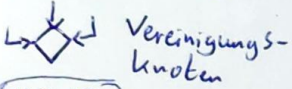
⊙ Endknoten

Aktion Aktionsknoten

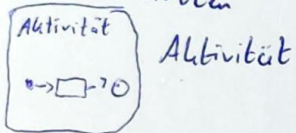
→ Kante



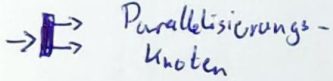
Verzweigungs-knoten



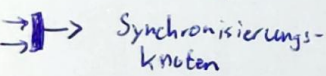
Vereinigungs-knoten



Aktivität



Parallelisierungsknoten

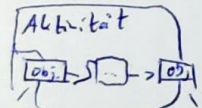


Synchronisierungsknoten

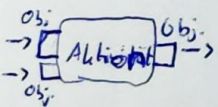


Schwimmbahnen

Bereichs-namen



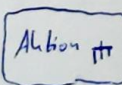
Eingabeparameter Ausgabeparameter



Objekt

Objektknoten (Zustand des Objekts)

[beobachtet] kann in eckigen Klammern dazugeschrieben werden



Aktion

Die Kante verweist auf eine weiter beschriebene Aktivität (Hierarchische Schachtelung von Aktivitäten)

2.1 Aktivitätsdiagramme

- Modellieren den Ablauf/die Logik eines Programms
- In eine Aktion kann nur eine Kante rein und nur eine Kante raus

- Ansonsten werden Verzweigungs-, Vereinigungs-, Parallelisierung- und Synchronisierungsknoten benötigt (seit UML 2)

↳ Verzweigung entspricht „logischem oder“, Parallelisierung dem „logischen und“

→ Bei der Verzweigung reicht eine Kante am Vereinigungsknoten zum Fortsetzen

→ Bei der Parallelisierung müssen alle Kanten zum Fortsetzen am Synchronisierungsknoten erfüllt sein

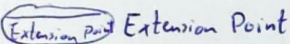
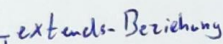
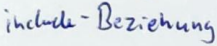
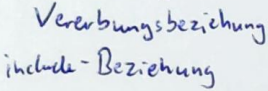
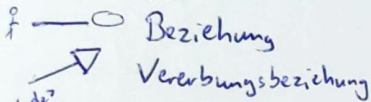
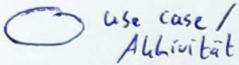
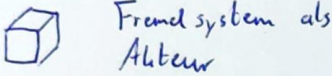
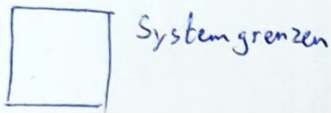
- Verzweigung muss durch Vereinigung und Parallelisierung durch Synchronisierung beendet werden (nicht mischen!)

- Eine Aktivität kann Ein- und/oder Ausgabeparameter haben (Ein- und Ausgabeparameter ersetzen Start- & Endknoten)

- Objektknoten können alternativ als Pin dargestellt werden (Eine Aktion kann mehrere Ein- und Ausgabepins haben)

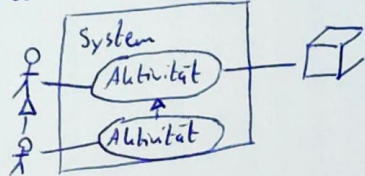
2. UML-Diagramme (Forts.)

2.2 Anwendungsfalldiagramme

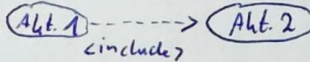


- Stellen Anforderungen verschiedener Akteure an Software dar
↳ (Anwendungsfälle / use cases)

Beispiel:

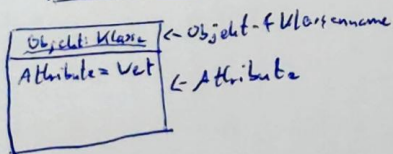
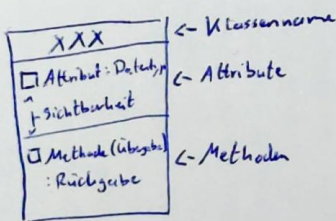
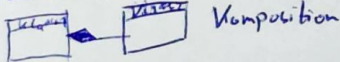
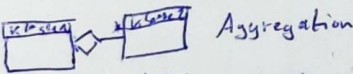
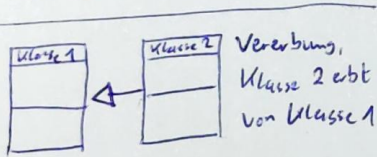
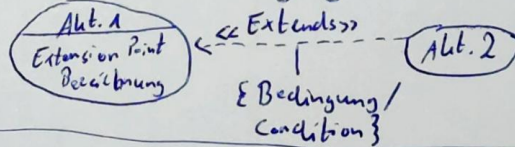


- Vererbung (von Erbenden vom Vererbenden)
→ Erbe teilt alle Eigenschaften des Vererbenden.
→ „Spezialfall“
- Include-Beziehung
→ Aktivität 1 führt immer Aktivität 2 aus
→ Aktivität 1 includes Aktivität 2

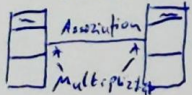


- Extends-Beziehung

→ Aktivität 1 führt Aktivität 2 aus, wenn die Condition erfüllt ist. Bedingung wird am Extension Point geprüft.



Assoziation:



2.3 UML Klassen- und Objektdiagramme

- Zur Darstellung von Klassen- und Objekten

Assoziation

- Geht von Klasse zur anderen Klasse

- Werden nicht als Attribut genannt

↳ 1 Genau 1 0..1 0 oder 1 * beliebig 3..* 3 oder beliebig

Aggregation

- Klasse besteht aus Objekten der Klasse 2

Komposition

- Strengere Form der Aggregation

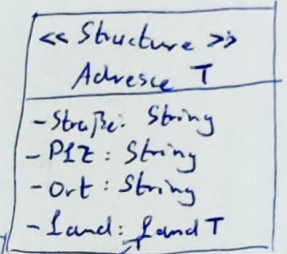
↳ Wird Objekt d. Klasse 1 gelöscht, müssen auch alle Objekte der Klasse 2 gelöscht werden

2. UML-Diagramme (Forts.)

2.3 UML Klassen- und Objektdiagramme (Forts.)

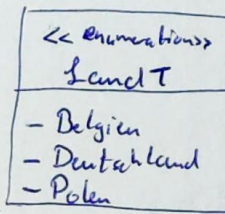
Structure Typ

- Zusammenfassung mehrerer Typen zu einem strukturierten Typ
- > Beispiel Adresse: String Straße, PLZ, Ort, Land
- Bietet den Vorteil, dass structure Typ in verschiedenen Klassen wiederverwendet werden können
- Am Ende des Namens wird ein 'T' angehängen
- ↳ Beispiel: Adresse T



Enumeration Typ

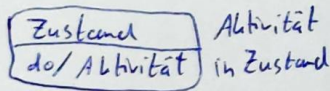
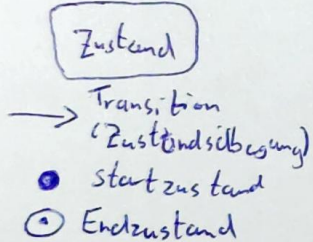
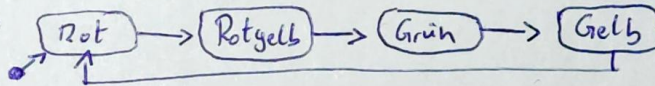
- Aufzählung bzw. endliche Wertemenge
- Ebenfalls T am Ende



2.4 UML-Zustandsdiagramme

- Objekt bekommt Lebensdauer und Zustände

↳ Beispiel Ampel:



Beschriftung einer Transition

ereignis (argument) [bedingung] / aktion (argument)

↳ Jeder Teil kann weggelassen werden

Ereignisarten

- Signal Event Empfang eines Signals (Rechtsklick / Auf OK geklickt)
- Call Event Empfang eines Operationsaufrufes (stornieren())
- Time Event Zeitabhängiges Ereignis -> after(Zeit), at (Zeitpunkt)
- Change Event Dauerhafte Überwachung -> when (Bedingung)
- Im Vergleich zu [Bed.]: Bei [Bed.] wird nur einmal geprüft
- Implizites Event Wenn vorherige Aktion beendet -> ohne Beschriftung

Aktionen (Beispiele)

- Lokale Änderung eines Attributwerts
- Versenden einer Nachricht / eines Signals an ein anderes Objekt
- Erzeugen oder Löschen eines Objekts