

Kapitel 2.2 - Weitere UML-Diagrammtypen

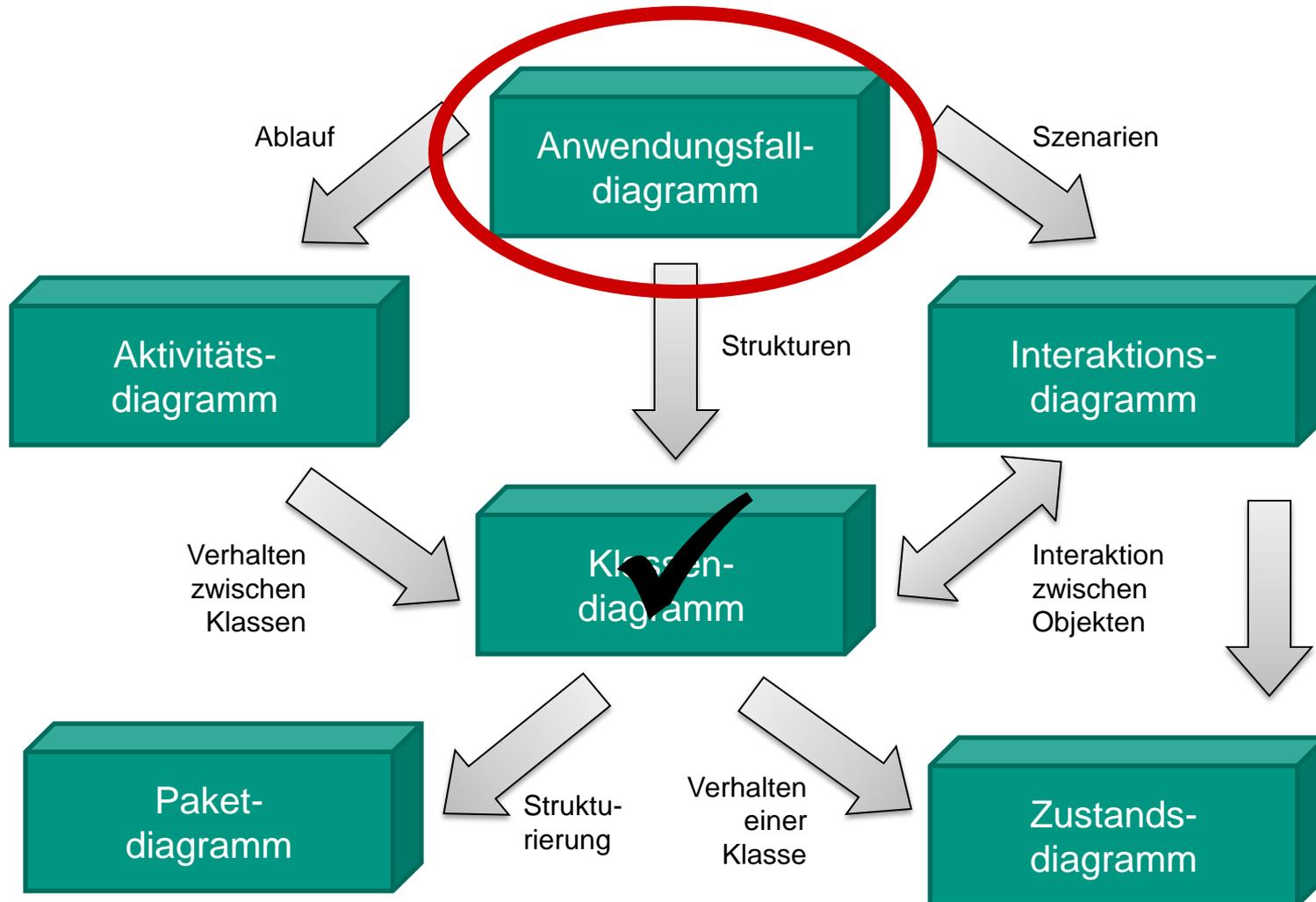
SWT I – Sommersemester 2010

Walter F. Tichy, Andreas Höfer, Korbinian Molitorisz

IPD Tichy, Fakultät für Informatik



UML-Diagramme



nach J.W. Schmidt, F. Matthes, TU Hamburg-Harburg

Anwendungsfalldiagramm (engl. *use case diagram*)

- Zur **Anforderungsspezifikation** – was will der Benutzer von seinem System?
- Modellieren typischer **Interaktionen** des Benutzers mit dem System
- Gewinnung aus
 - Dialog mit dem (zukünftigen) Benutzer
 - Dialog mit Experten } evtl. ≠ Kunde!
- Ermöglicht Kontrolle, ob das System das vom Auftraggeber gewünschte leistet (Design und Implementierung)

Anwendungsfalldiagramm (engl. *use case diagram*)

- Zur **Anforderungsspezifikation** – was will der Benutzer von seinem System?
- Modellieren typischer **Interaktionen** des Benutzers mit dem System
- Geht um die **Erkennung von Anforderungen**
 - **Wichtig: Anwendungsfalldiagramme sind ein Hilfsmittel zur Anforderungsermittlung und –verwaltung. Sie zeigen nur Zusammenhänge der an Anwendungsfällen beteiligten Modellelemente.**
- Ermitteln der **Abgrenzung** des Systems (Dokumentation)

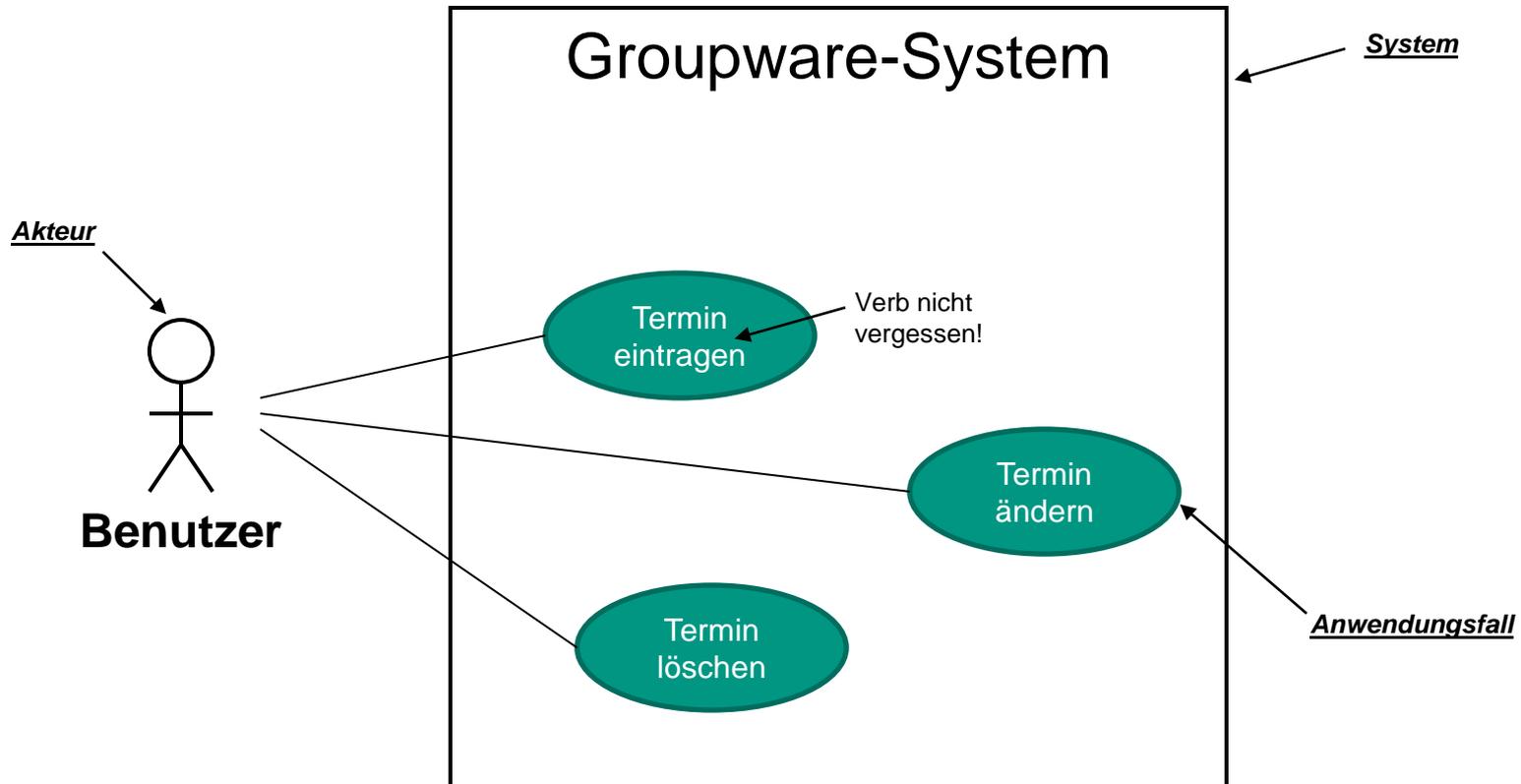
Anwendungsfalldiagramme modellieren kein Verhalten und keine Abläufe!

Definition

- Def. **Anwendungsfall**: Ein Anwendungsfall ist eine typische, gewollte **Interaktion** eines oder mehrerer **Akteure** mit einem (geschäftlichen oder technischen) **System**.
 - Ein Anwendungsfall wird stets durch einen Akteur initiiert und führt i.d.R. zu einem durch einen Akteur wahrnehmbaren Ergebnis.
 - Ein Anwendungsfall beschreibt *was* ein System leisten muss, nicht *wie* es das leisten muss – er kann insbesondere mehrere verschiedene Ablaufvarianten umfassen

könnte z.B. mit einem Aktivitätsdiagramm beschrieben werden

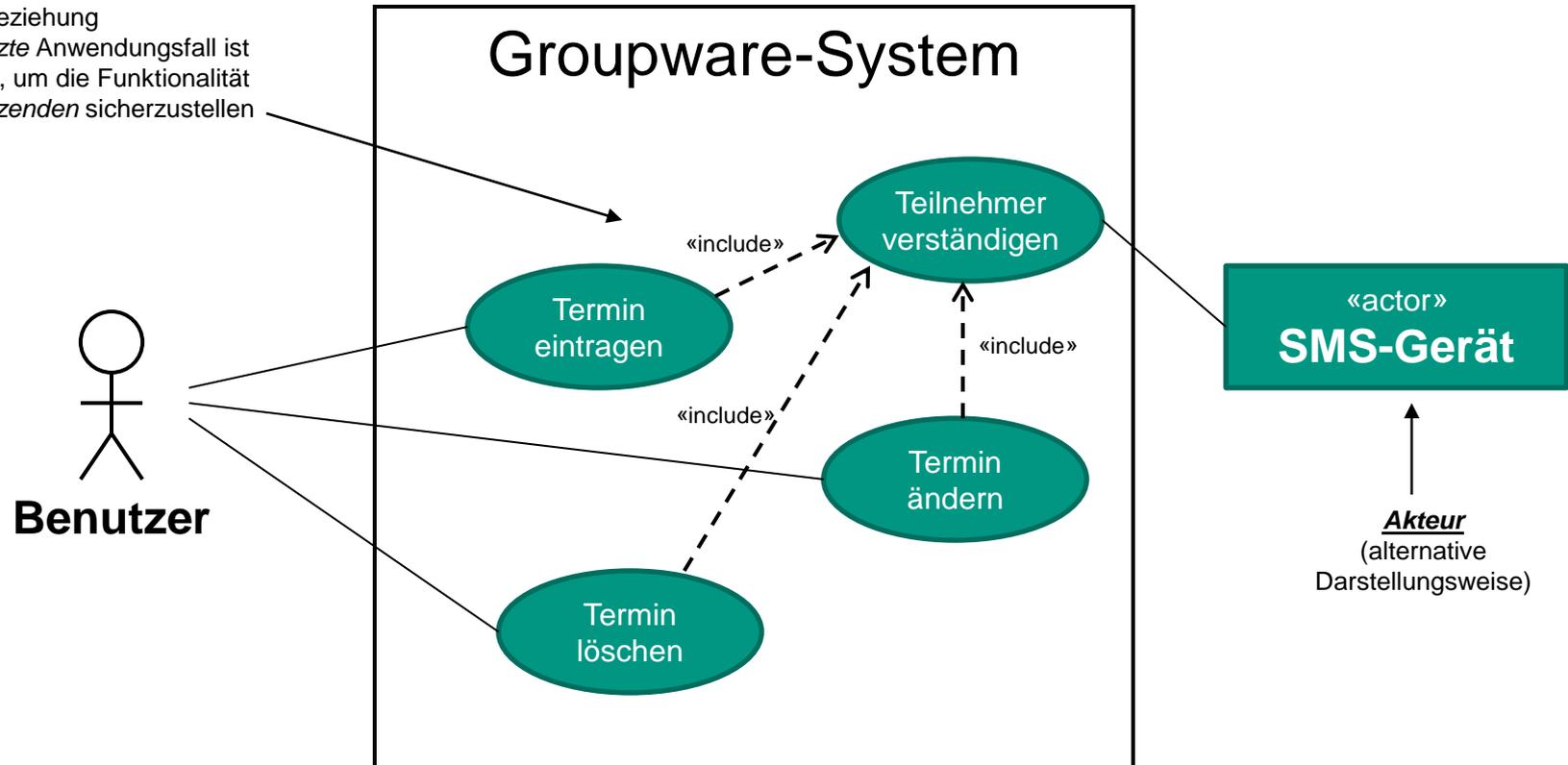
Beispiel „Groupware-System“



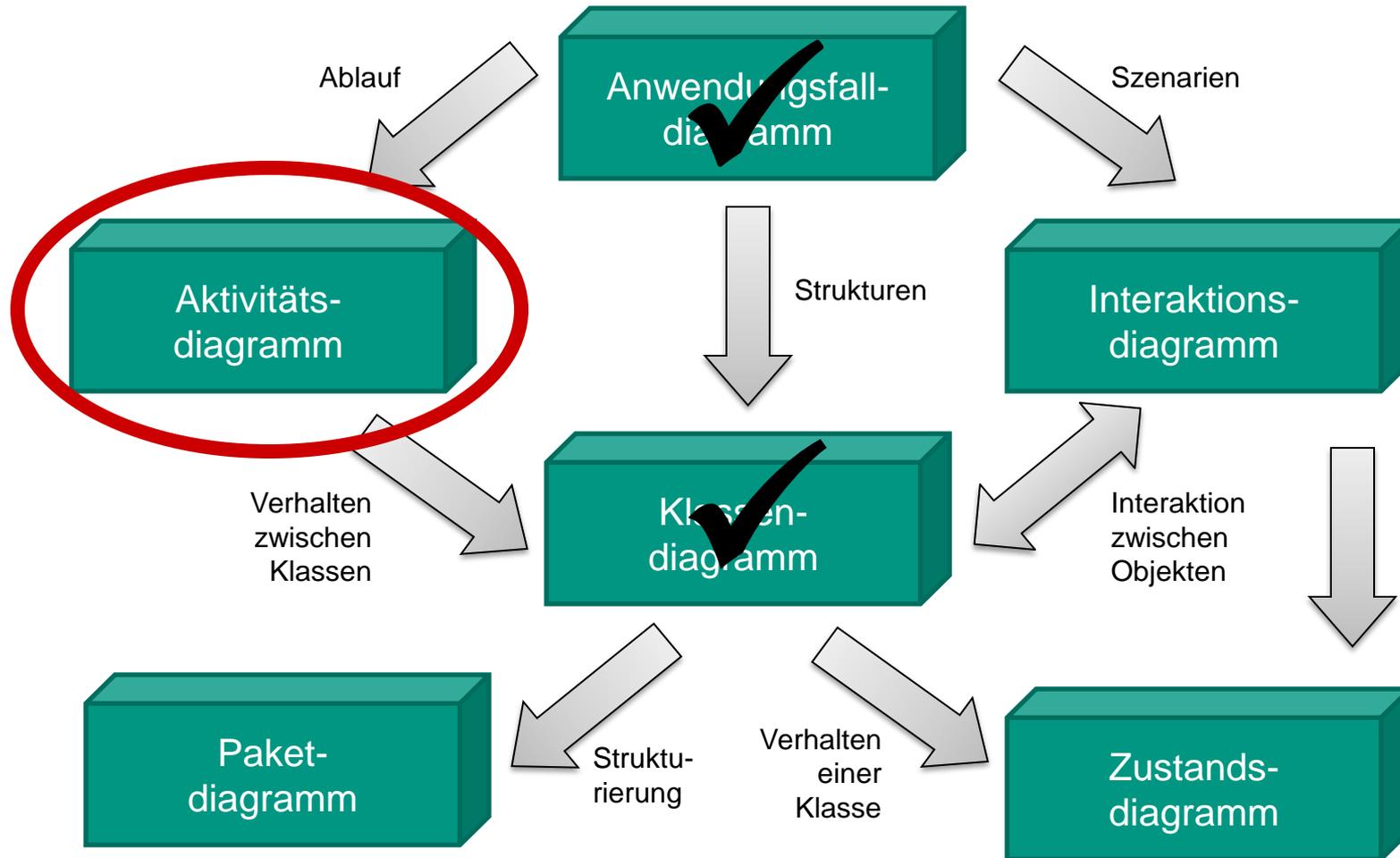
Benutzung anderer Anwendungsfälle

Enthält-Beziehung

Der *benutzte* Anwendungsfall ist notwendig, um die Funktionalität des *benutzenden* sicherzustellen



UML-Diagramme



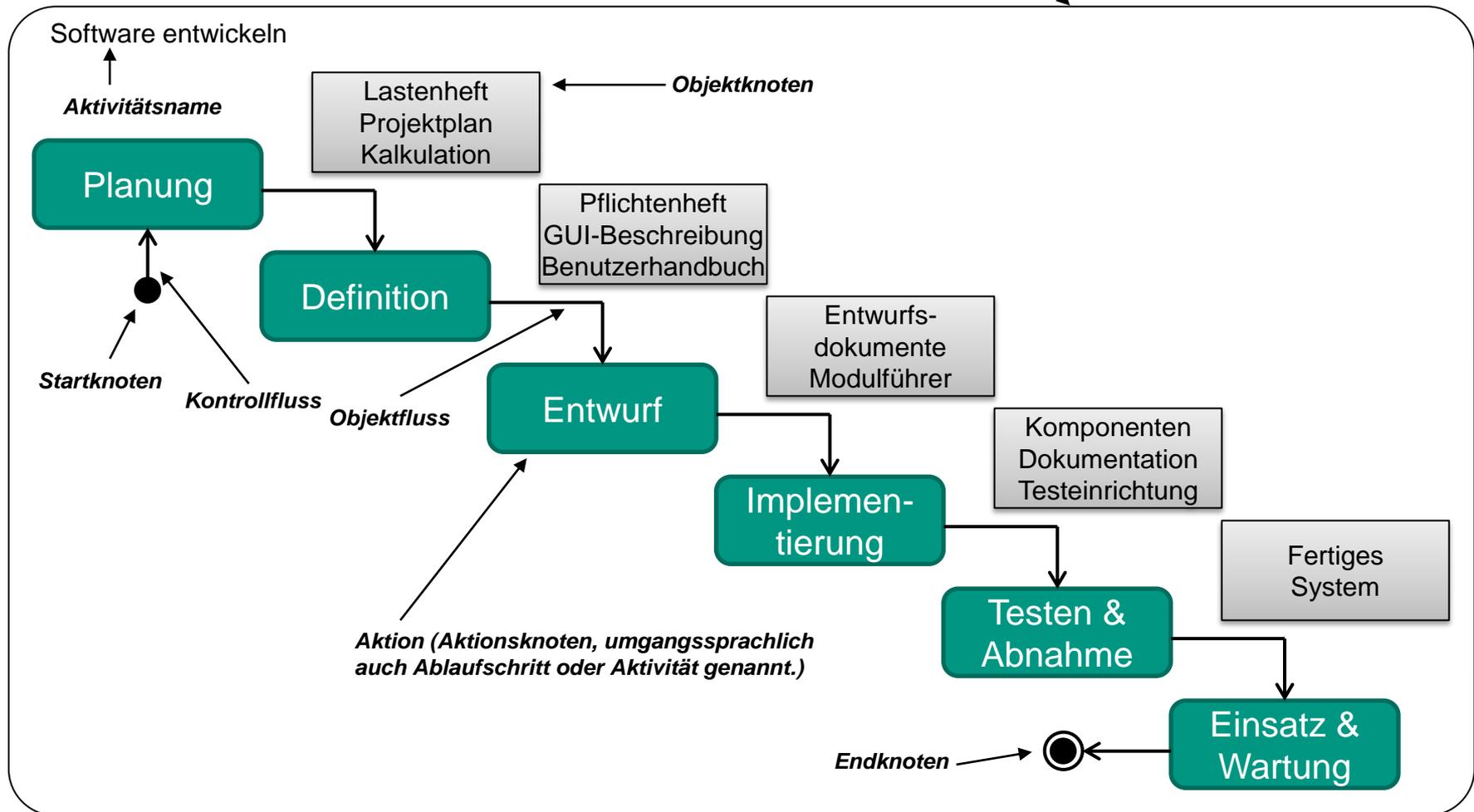
nach J.W. Schmidt, F. Matthes, TU Hamburg-Harburg

Aktivitätsdiagramm

- Ein Aktivitätsdiagramm beschreibt einen **Ablauf**
 - Betriebswirtschaftliche oder geschäftliche Prozesse
 - Technische Abläufe von Workflows und Anwendungsfällen
 - Konkrete algorithmische Abläufe in Programmen
- Aktivitätsdiagramme bestehen aus
 - Aktions-, Objekt- und **Kontrollknoten**, sowie
 - Objekt- und **Kontrollflüssen**.
- Semantik lehnt sich an Petrinetze an

Beispiel

Aktivität (auch: Aktivitätsdiagramm)



Aktion (Aktionsknoten, umgangssprachlich auch Ablaufschritt oder Aktivität genannt.)

Elemente eines Aktivitätsdiagramms

■ Aktionen

- Elementare Aktion
- Verschachtelte Aktion



■ Knoten

- Startknoten
 - Startpunkt eines Ablaufs
- Endknoten
 - Beendet alle Aktionen und Knotenrollflüsse
- Ablaufende
 - Beendet einen einzelnen Objekt- und Kontrollfluss



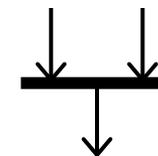
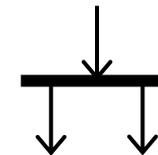
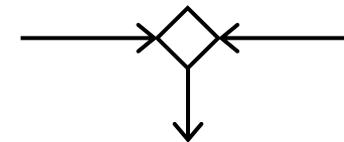
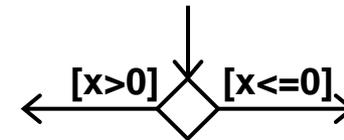
Elemente eines Aktivitätsdiagramms (2)

- Entscheidung
 - „if“-Verzweigung

- Zusammenführung
 - „oder“-Verknüpfung

- Teilung
 - Aufteilung eines Kontrollflusses

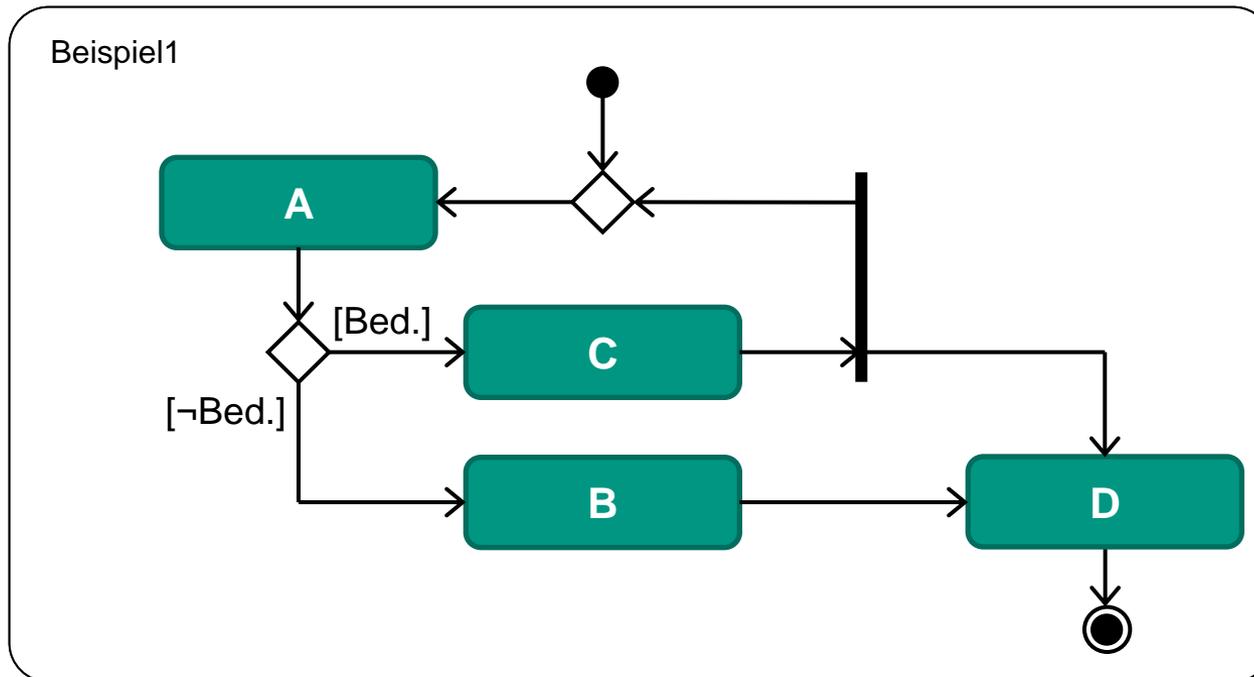
- Synchronisation
 - „und“-Verknüpfung



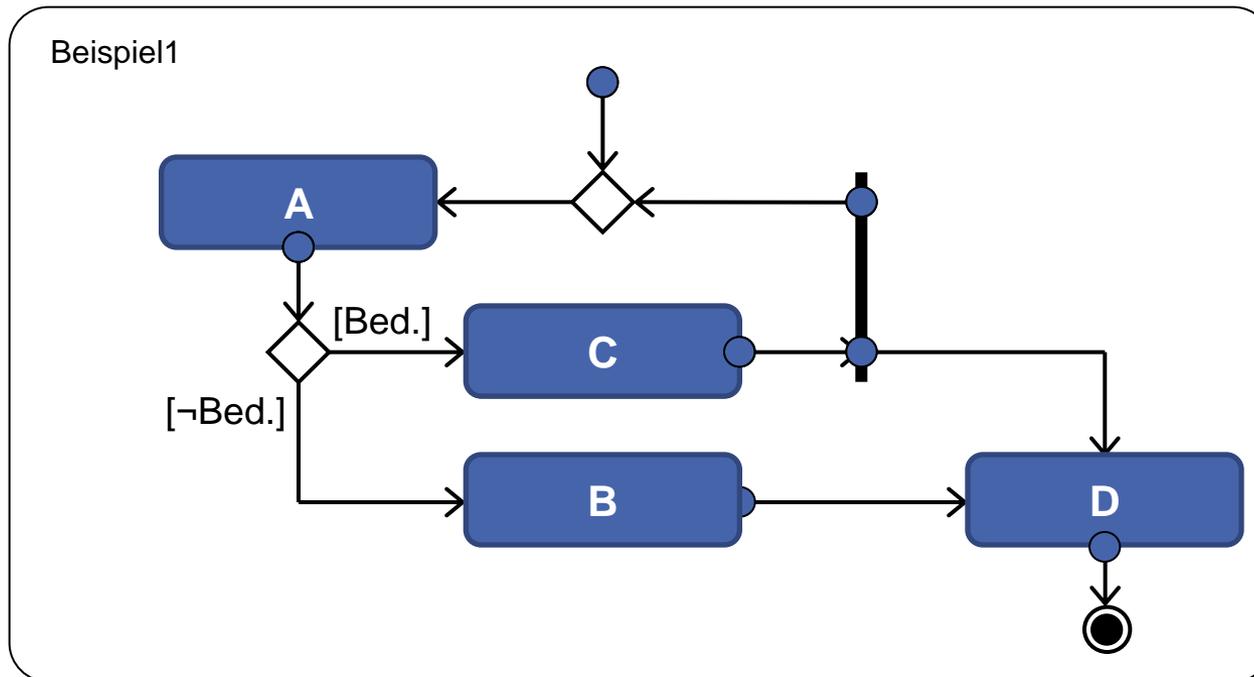
Ausführungssemantik - Aktionen

- Die Abfolge der Aktionen wird durch Kontrollfluss- und Objektflusskanten gesteuert. (Abk. KFK bzw. OFK)
 - Auf Kontrollflusskanten wandern Kontrollmarken, auf Objektflusskanten wandern Objektmarken
- Eine Aktion kann erst dann ausgeführt werden, wenn alle eingehenden KFK und OFK Marken tragen.
- Wenn die Aktion beginnt werden Marken von den eingehenden Kanten entnommen:
 - Eine Objektmarke von jeder OFK sowie
 - Alle Kontrollmarken von den KFK. (Achtung: Abweichung von Petrinetz-Semantik!)
- Nach Ende der Aktion werden auf allen ausgehenden KFK und OFK Marken angeboten, die dann wiederum anderen Aktionen zur Verfügung stehen.

Ausführungssemantik - Beispiel



Ausführungssemantik Beispiel – Animation



Elemente eines Aktivitätsdiagramms (3)

■ Objektknoten

- Eingabe- und Ausgabedaten einer Aktion
- Darstellung durch Stecker (engl. *pin*)

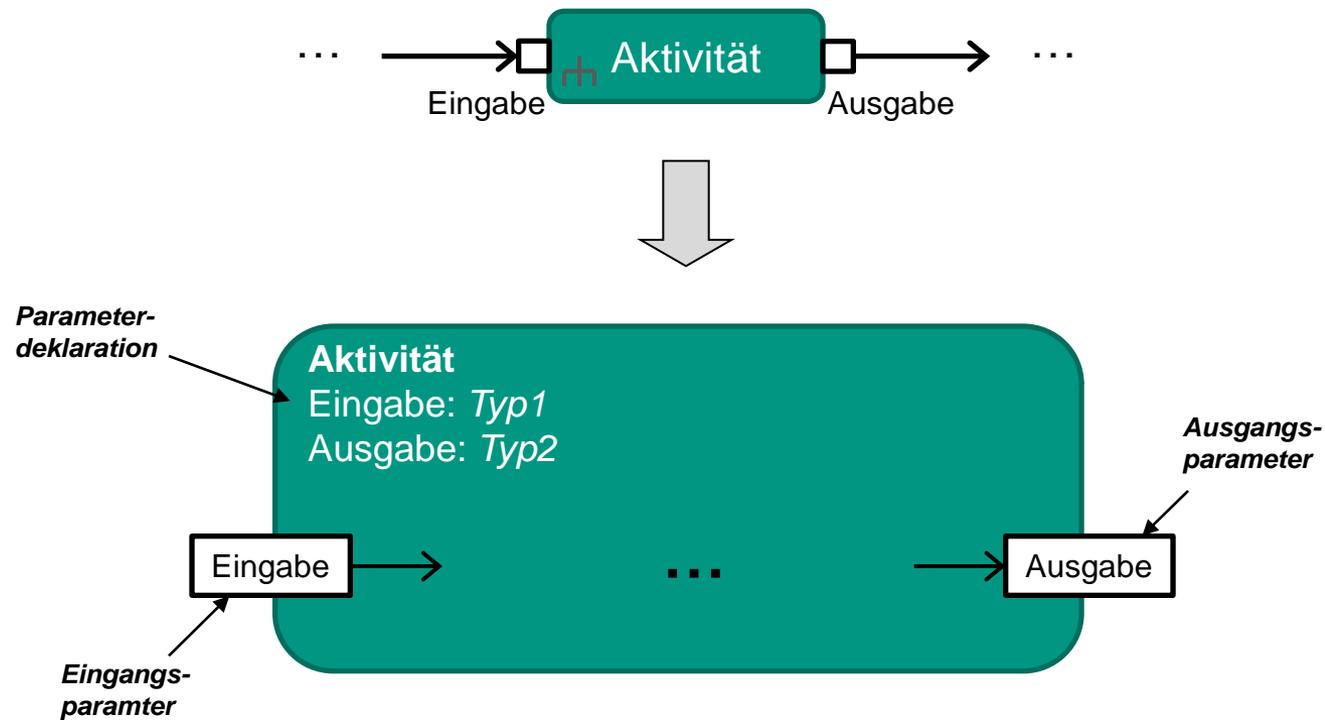


■ Alternative Darstellungen



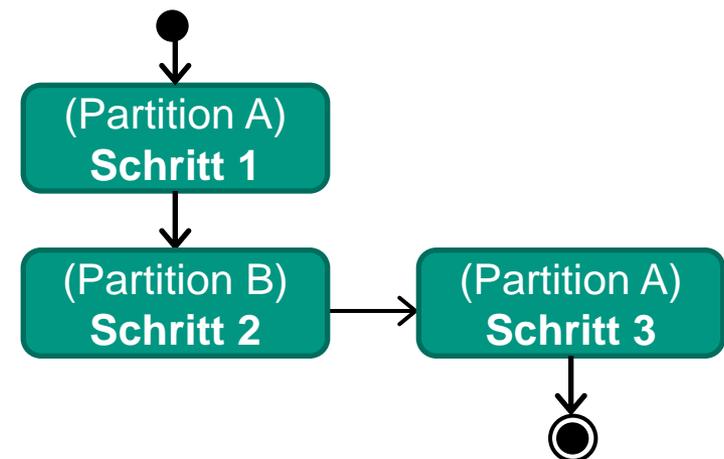
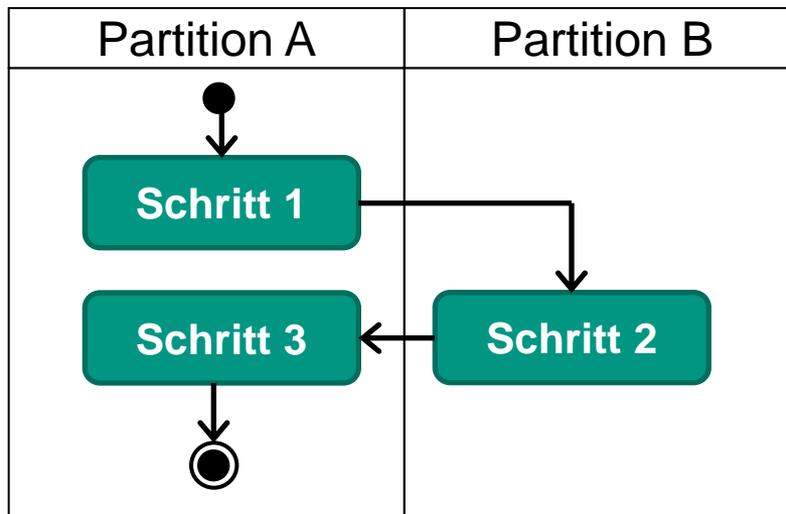
Elemente eines Aktivitätsdiagramms (4)

■ Parameter von Aktivitäten

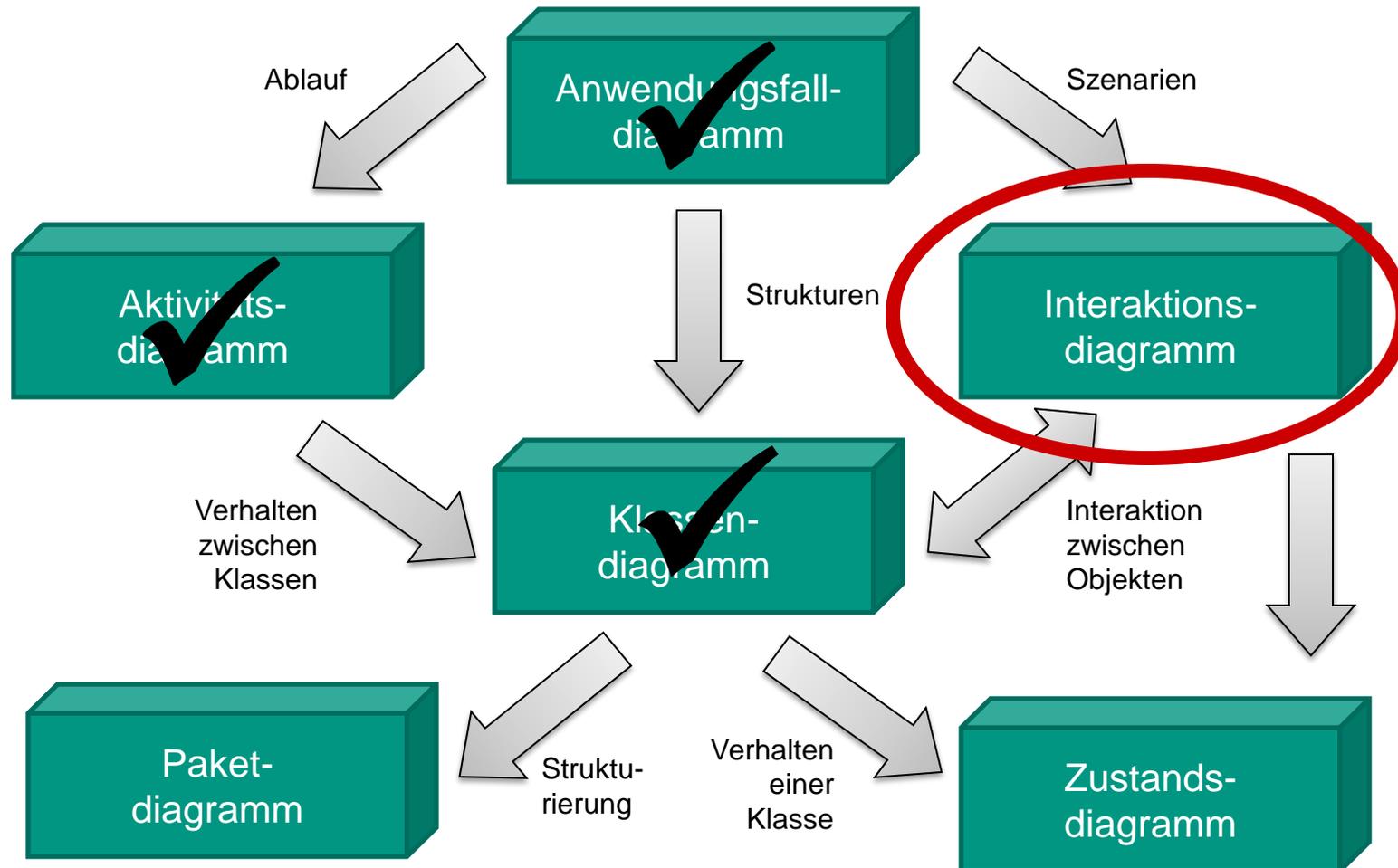


Elemente eines Aktivitätsdiagramms (5)

- Partitionen (Verantwortlichkeitsbereiche)
 - Partitionen beschreiben wer oder was für einen **Knoten** verantwortlich ist oder welche **gemeinsame Eigenschaft** sie kennzeichnet.
 - Partitionen könnten z.B. unterschiedliche Rechner sein, die zusammenwirken (z.B. Dienstgeber und Klient)



UML-Diagramme



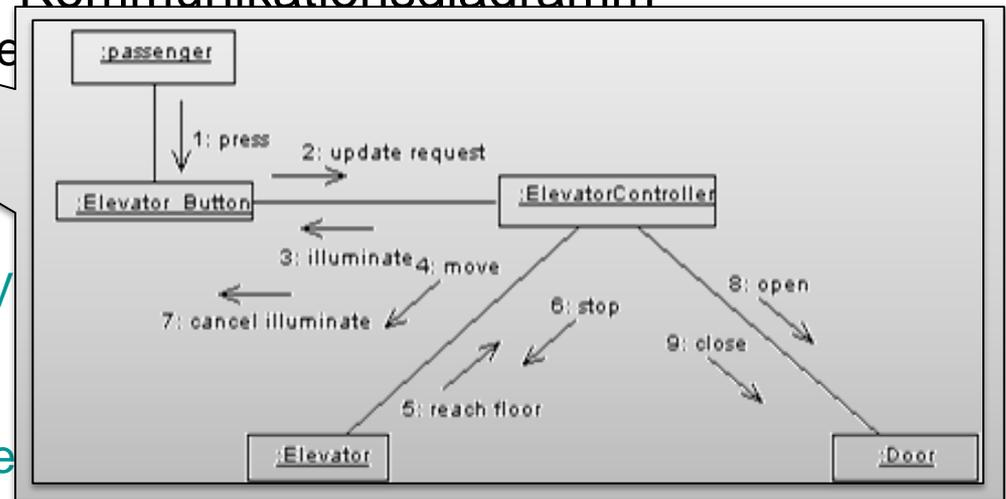
nach J.W. Schmidt, F. Matthes, TU Hamburg-Harburg

Interaktionsdiagramme

- Zeigen die für einen bestimmten Zweck notwendigen **Interaktionen** zwischen **Objekten**
- Das Klassendiagramm ist Grundlage der Interaktionsdiagramme
- Vier Typen
 - Kollaborationsdiagramm / Kommunikationsdiagramm
 - Schwerpunkt: **Struktur** der Interaktionspartner
 - Zeitdiagramm
 - Schwerpunkt: **Zeitliche Koordination**
 - Interaktionsübersicht
 - Aktivitätsdiagramm zur **Veranschaulichung** komplexer Sequenzdiagramme
 - Sequenzdiagramm
 - Schwerpunkt: **Nachrichtenaustausch**

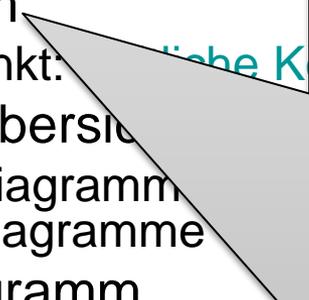
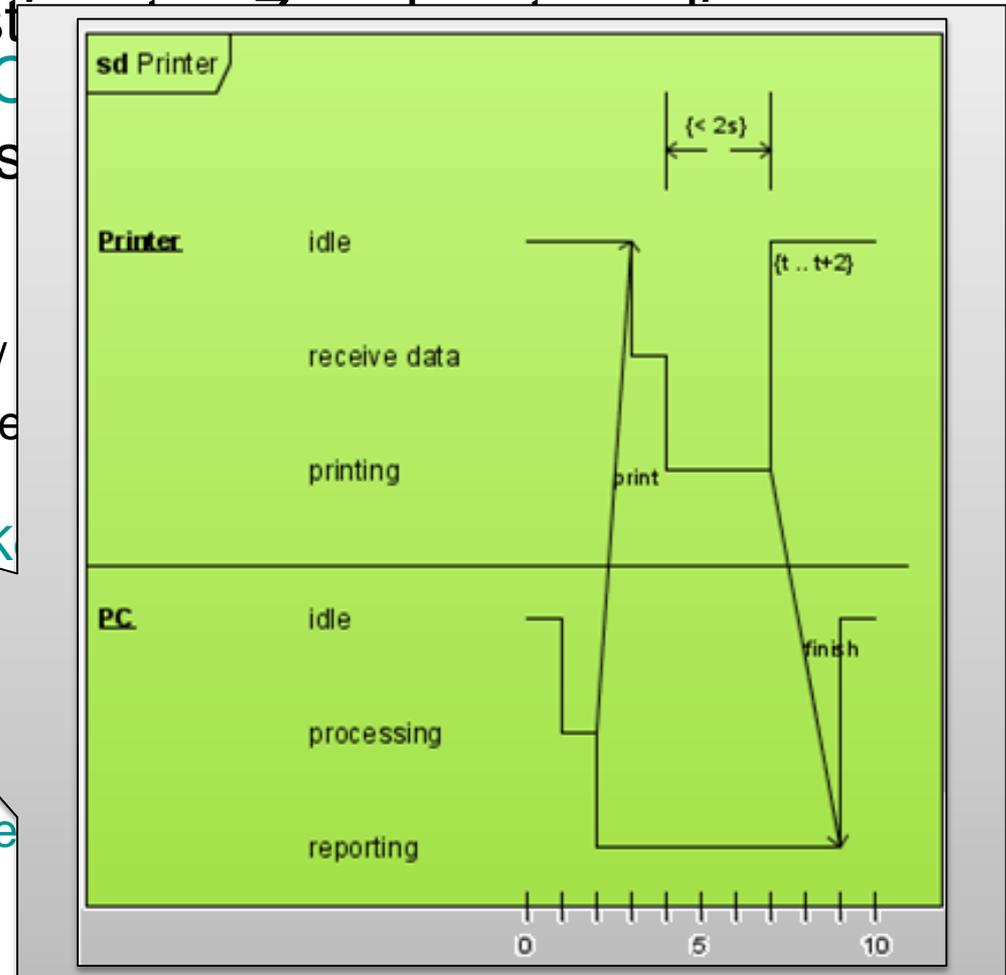
Interaktionsdiagramme

- Zeigen die für einen bestimmten Zweck notwendigen **Interaktionen** zwischen **Objekten**
- Das Klassendiagramm ist Grundlage der Interaktionsdiagramme
- Vier Typen
 - Kollaborationsdiagramm / Kommunikationsdiagramm
 - Schwerpunkt: **Struktur** der Interaktion
 - Zeitdiagramm
 - Schwerpunkt: **Zeitliche** Abfolge der Interaktion
 - Interaktionsübersicht
 - Aktivitätsdiagramm zur **Visualisierung** von Sequenzdiagrammen
 - Sequenzdiagramm
 - Schwerpunkt: **Nachrichtenaustausch**



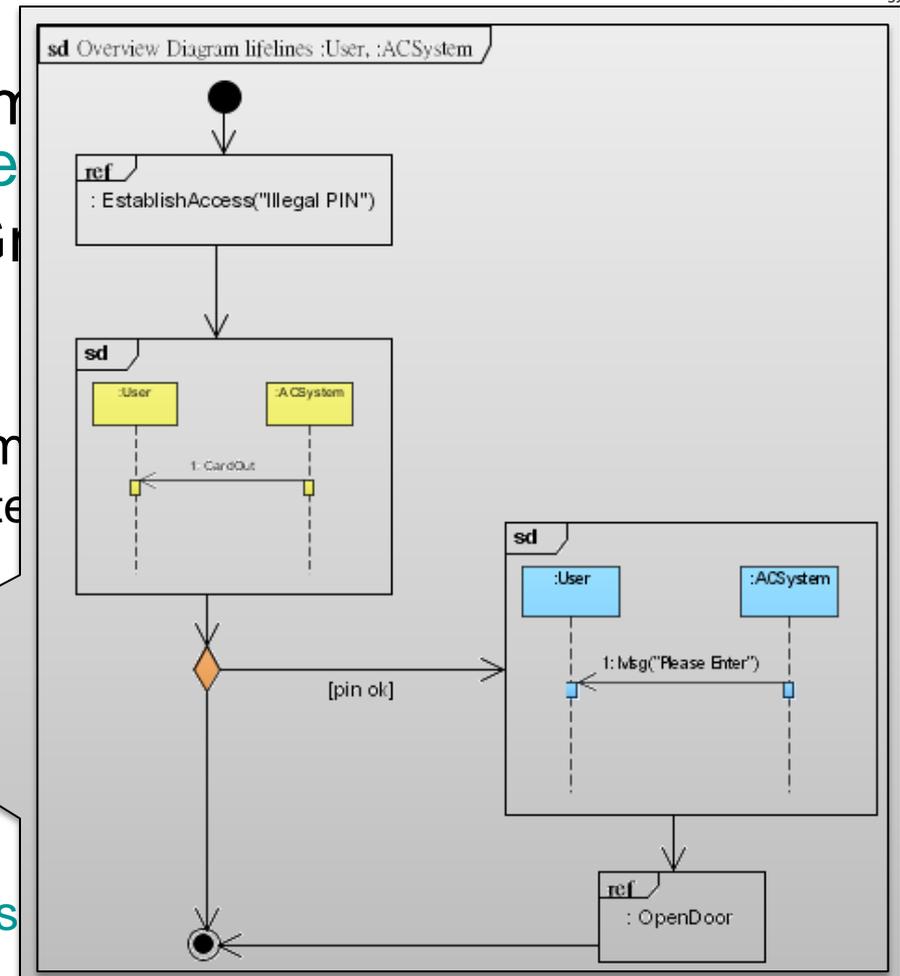
Interaktionsdiagramme

- Zeigen die für einen bestimmten Prozess relevanten **Interaktionen** zwischen Objekten
- Das Klassendiagramm ist ein **Strukturdiagramm**, die anderen sind **Interaktionsdiagramme**
- Vier Typen
 - Kollaborationsdiagramm / **Strukturdiagramm**
 - Schwerpunkt: **Struktur** der Interaktion
 - Zeitdiagramm
 - Schwerpunkt: **zeitliche Kette** der Interaktion
 - Interaktionsübersicht
 - Aktivitätsdiagramm, Sequenzdiagramme
 - Sequenzdiagramm
 - Schwerpunkt: **Nachrichtenaustausch**



Interaktionsdiagramme

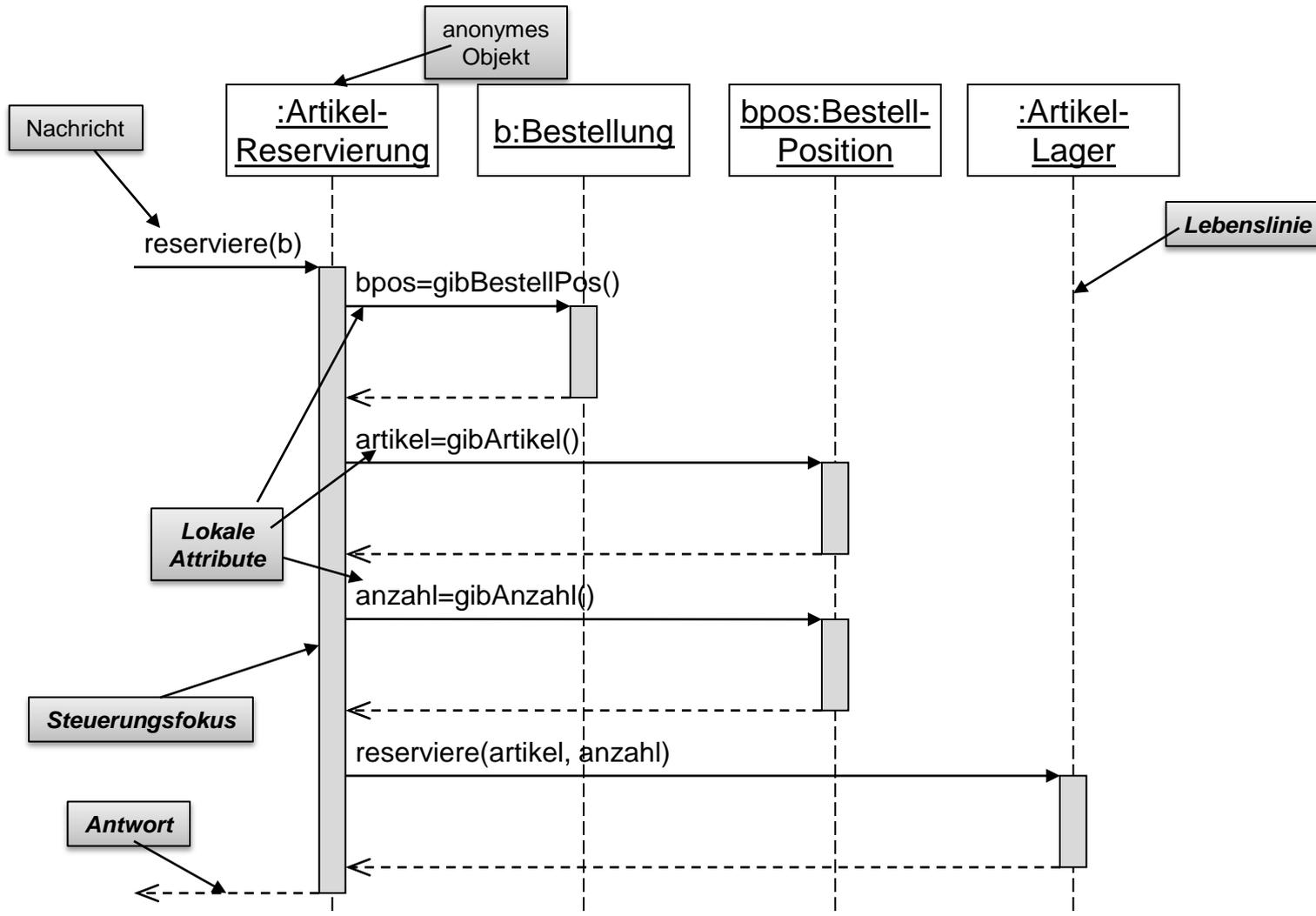
- Zeigen die für einen bestimmten Fall die Interaktionen zwischen Objekten
- Das Klassendiagramm ist Grundform aller Interaktionsdiagramme
- Vier Typen
 - Kollaborationsdiagramm / Komposition
 - Schwerpunkt: Struktur der Interaktion
 - Zeitdiagramm
 - Schwerpunkt: Zeitliche Koordination
 - Interaktionsübersicht
 - Aktivitätsdiagramm zur Veranschaulichung von Sequenzdiagrammen
 - Sequenzdiagramm
 - Schwerpunkt: Nachrichtenaustausch



Sequenzdiagramm

- Ursprünglich für die Modellierung von Telekommunikationsdiensten
- Exemplarische Darstellung eines möglichen **Ablaufs** eines **Anwendungsfalls**
 - Darstellung von Varianten sind möglich („alt“, „loop“), von dieser Möglichkeit sollte aber nur selten Gebrauch gemacht werden
- Konzentration auf den **zeitlichen Verlauf** der Nachrichten
 - Zeit verläuft von oben nach unten
 - Rollen werden durch senkrechte gestrichelte Linien dargestellt
 - Nachrichten werden durch waagerechte Pfeile zwischen Rollenlinien gezeichnet
- Intuitiv, aber benötigt viel Platz

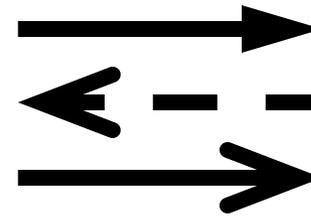
Beispiel



Elemente Sequenzdiagramm

■ Nachrichtentypen

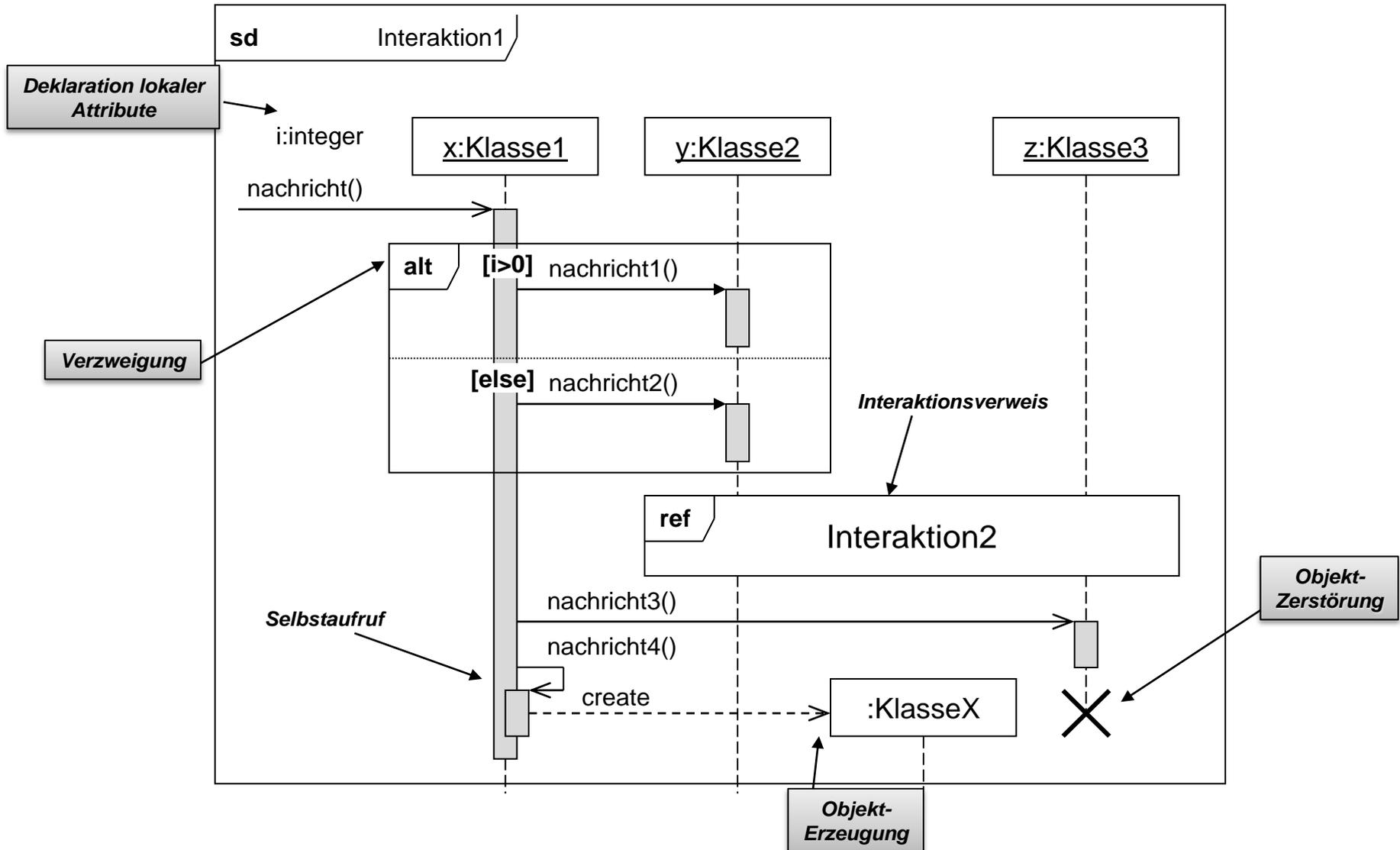
- Synchrone Nachrichten
- Antworten (optional)
- Asynchrone Nachrichten



■ Steuerungsfokus

- Überlagerung der gestrichelten Lebenslinien durch senkrechte Balken
- Gibt an, welche Rolle die Programmkontrolle besitzt
- Optional

Weitere Notationselemente

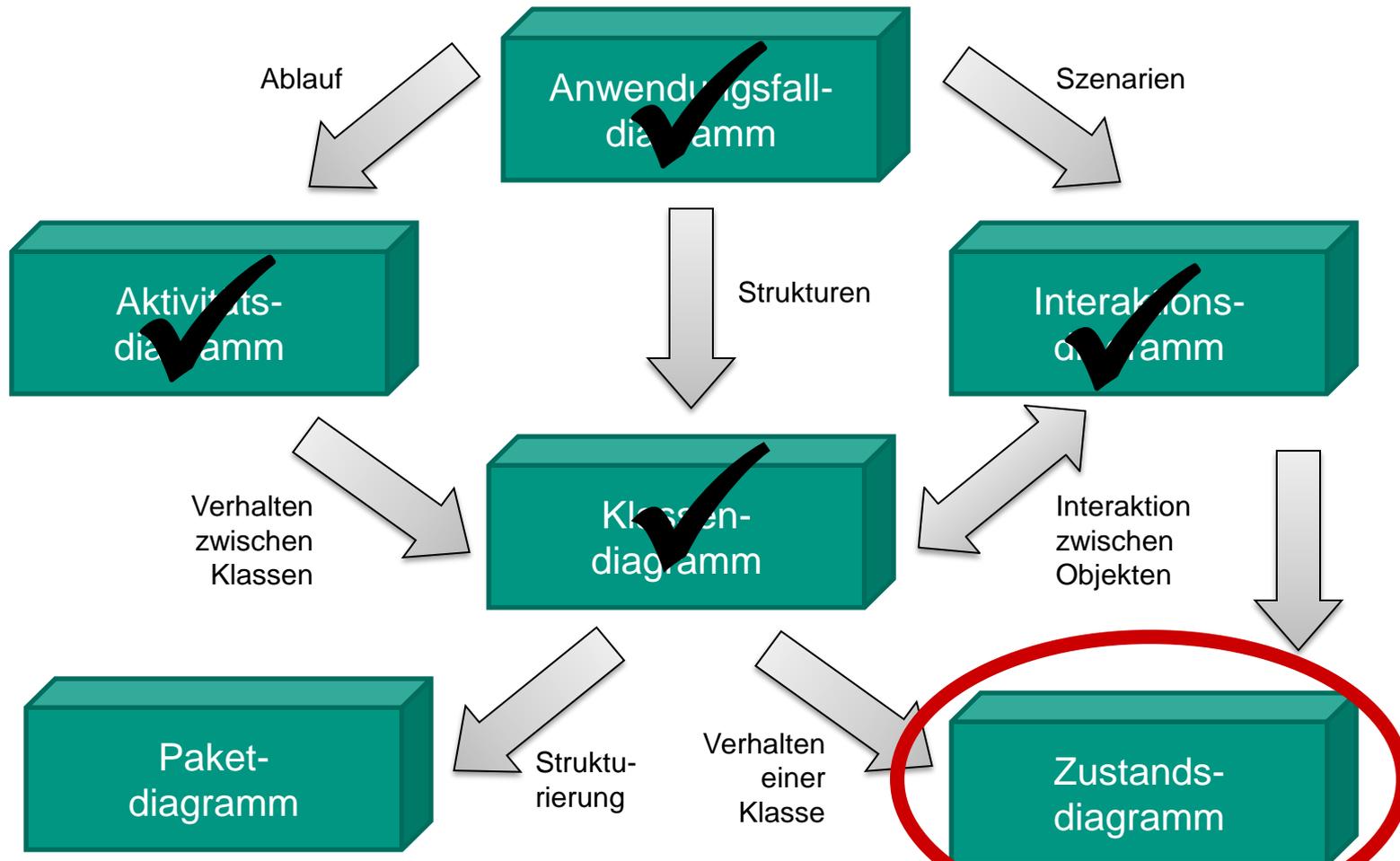


Operatoren

- Operatoren dienen dazu, alternative Abläufe und Verzweigungen auszudrücken
 - Nur mit großer Vorsicht verwenden, da dies schnell unübersichtlich wird
 - Bei einer Vielzahl von Möglichkeiten eignet sich eher ein Aktivitätsdiagramm

Operator	Bed./Parameter	Bedeutung
alt	[bed.1], [bed.1], ... [else]	Nur eine der Alternativen wird ausgeführt.
break	[bedingung]	Ist die Bedingung wahr, dann wird nur der Block ausgeführt und anschließend endet das Szenario.
opt	[bedingung]	Optionale Sequenz. Die Teilsequenz wird nur ausgeführt, wenn die Bedingung wahr ist.
par		Enthaltene Teilsequenzen werden parallel ausgeführt.

UML-Diagramme



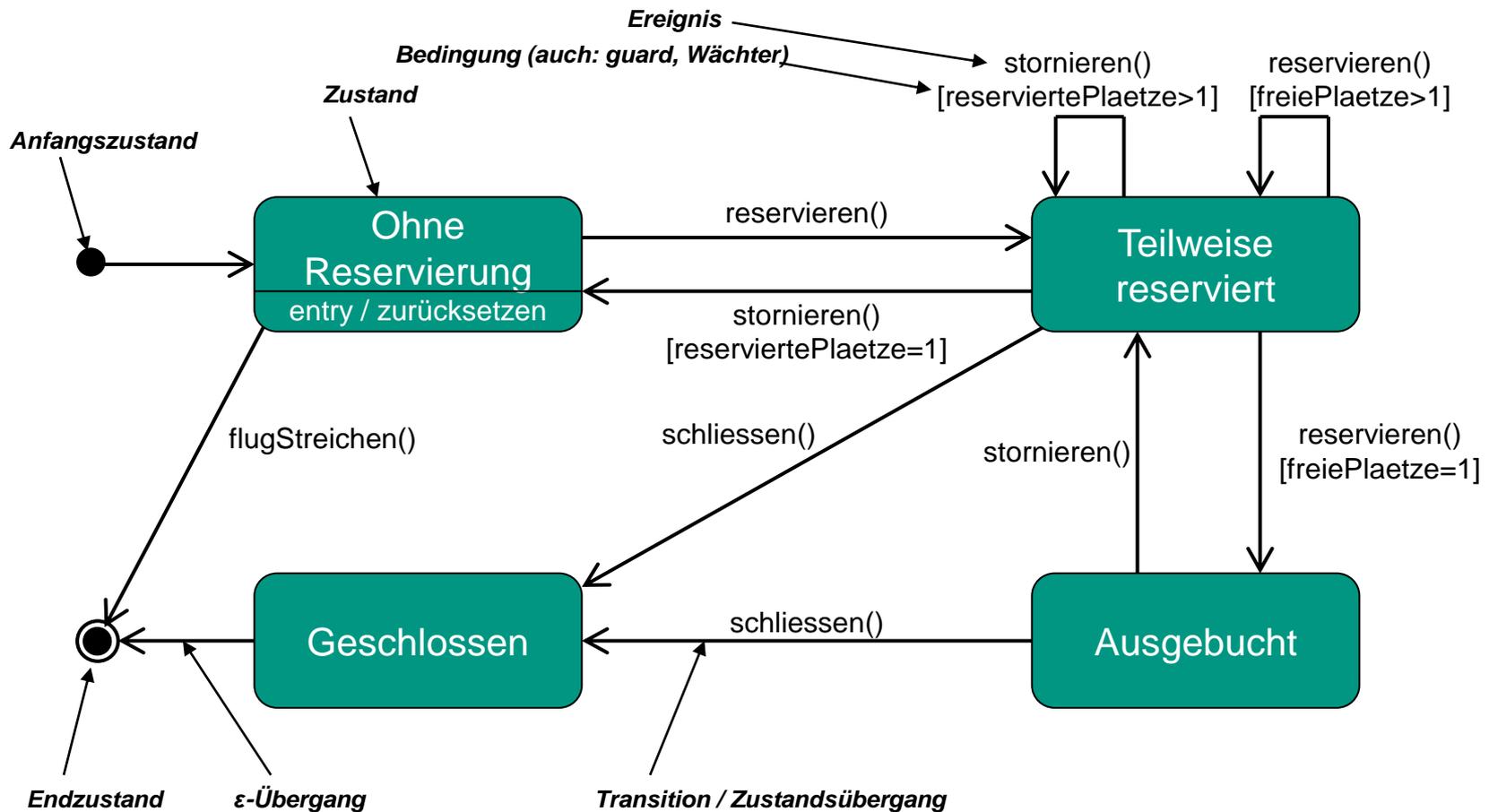
nach J.W. Schmidt, F. Matthes, TU Hamburg-Harburg

Zustandsdiagramm

- Beschreibt mögliche **Zustände** eines Objekts sowie mögliche **Zustandsübergänge** (endlicher Automat)
- Auf Basis der Interaktionsdiagramme für Klassen mit „interessantem Verhalten“, z.B.
 - Reale Dinge, die als „automatisch“ bezeichnet werden (Geldautomat, Garagenöffner, WaMa)
 - Kommunikationsprotokolle
 - Benutzerinteraktive Geräte
- Gültigkeit
 - Für den gesamten Lebenszyklus
 - Für die Ausführung einer Operation



Beispiel „Flugreservierung“



Zustandsdiagramm

- Ein Zustandsübergang wird durch ein Ereignis ausgelöst
 - An den Pfeilen werden die Transitionsbeschreibungen in folgender Form eingetragen:

ereignis(argumente)
[bedingung]
/operation(argumente)

- Ein Zustandsübergang findet nur statt, wenn zu dem Zeitpunkt, zu dem das Ereignis eintritt, auch die entsprechende Bedingung gültig ist (*guarded transition*)
- Sonderfall: ϵ -Übergang (auch „Spontanübergang“), braucht kein Ereignis, kann jederzeit erfolgen, wenn
 - Das System sich in dem Zustand befindet und
 - Die Bedingung erfüllt ist

Zustandsdiagramm

■ Spezielle Ereignisse

■ at(ausdruck)

Der Ausdruck beschreibt einen **exakten absoluten Zeitpunkt**.
Sobald der Zeitpunkt erreicht ist, feuert die Transition.

■ after(ausdruck)

Hier muss der Ausdruck einen **relativen Zeitpunkt** beschreiben.

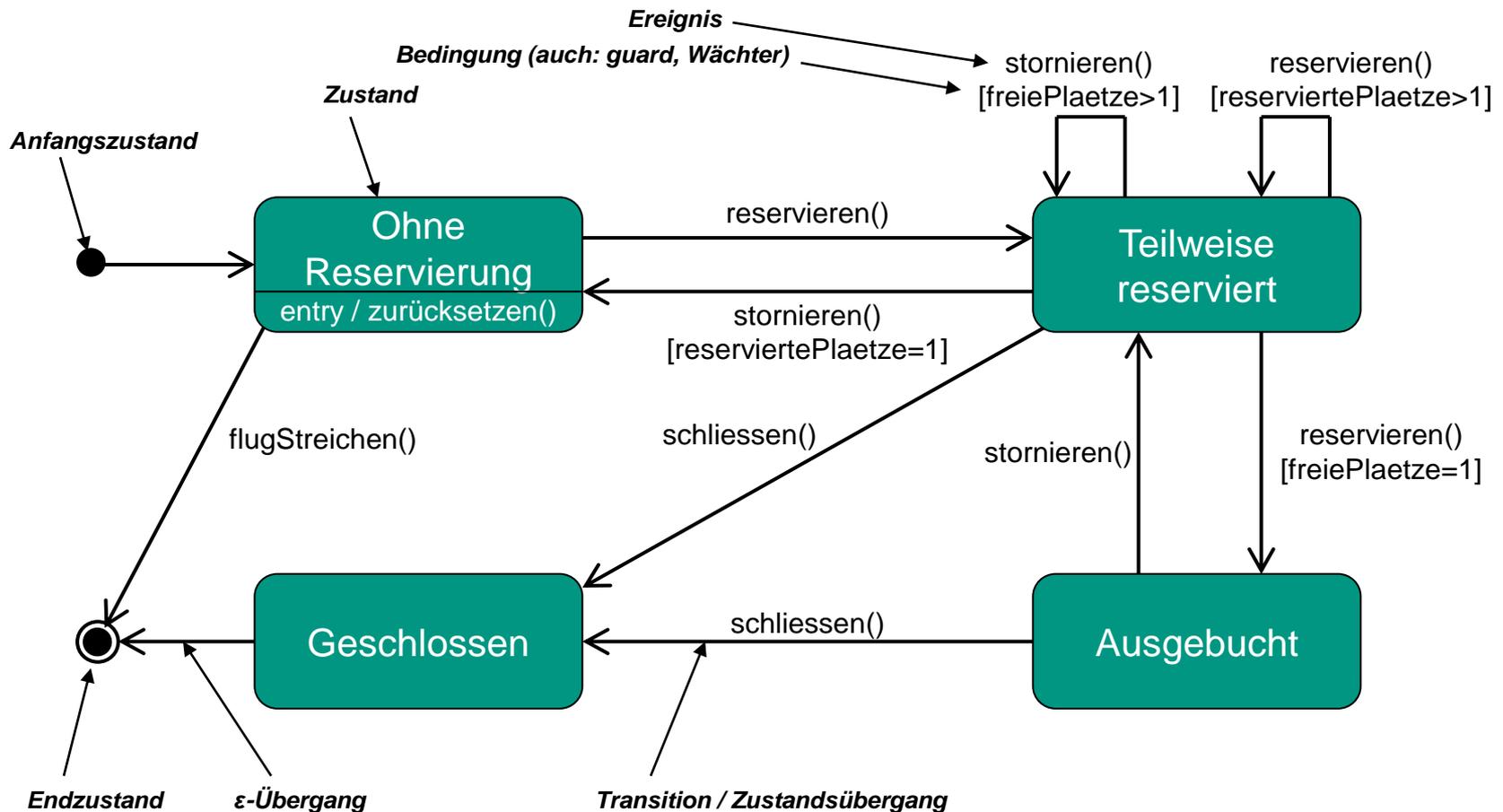
Aktionen

■ Aktionen

- Mit einem Zustandsübergang kann eine Aktion verbunden sein.
- **Eintrittsaktion** (entry action): wird beim Übergang *in* den Zustand ausgeführt.
Schreibweise: `entry / aktion()`
- **Austrittsaktion** (exit action): wird beim Übergang *aus* dem Zustand in einen anderen ausgeführt.
Schreibweise: `exit / aktion()`

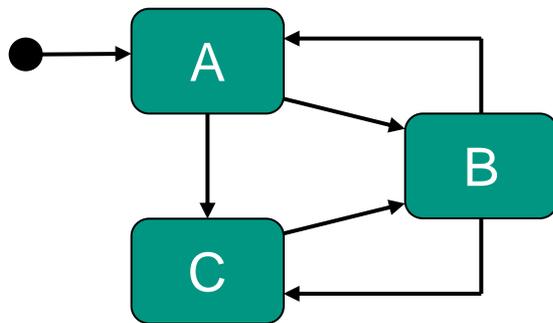
- Eine Aktion wird beim entspr. Übergang sofort ausgeführt und benötigt keine (bzw. vernachlässigbare) Zeit

Beispiel „Flugreservierung“



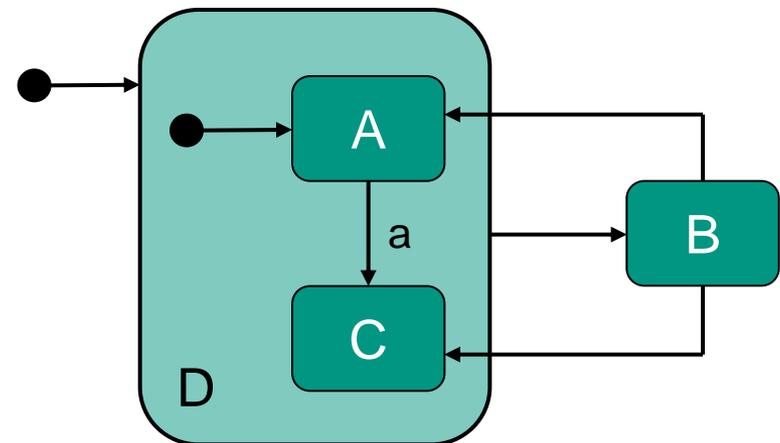
Weitere Möglichkeiten

■ Hierarchischer Zustandsautomat



Ohne Hierarchie

\cong

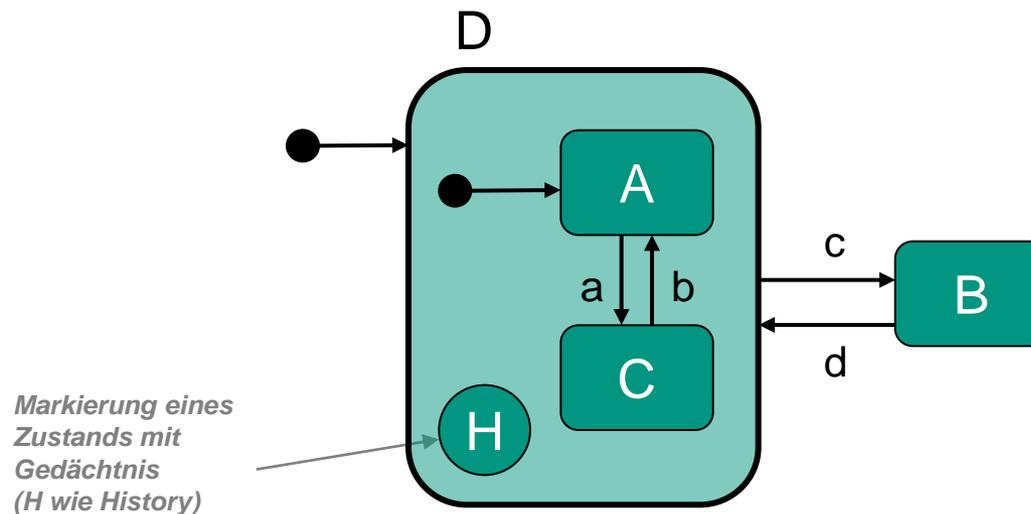


Mit Hierarchie

Weitere Möglichkeiten

■ Zustände mit Gedächtnis

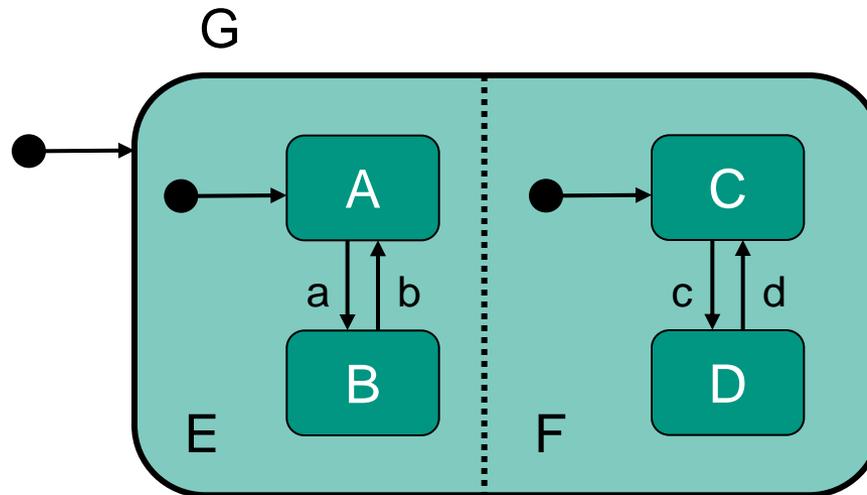
- Beim Übergang in einen Zustand mit Unterzuständen wird in den zuletzt eingenommenen Zustand zurückgekehrt



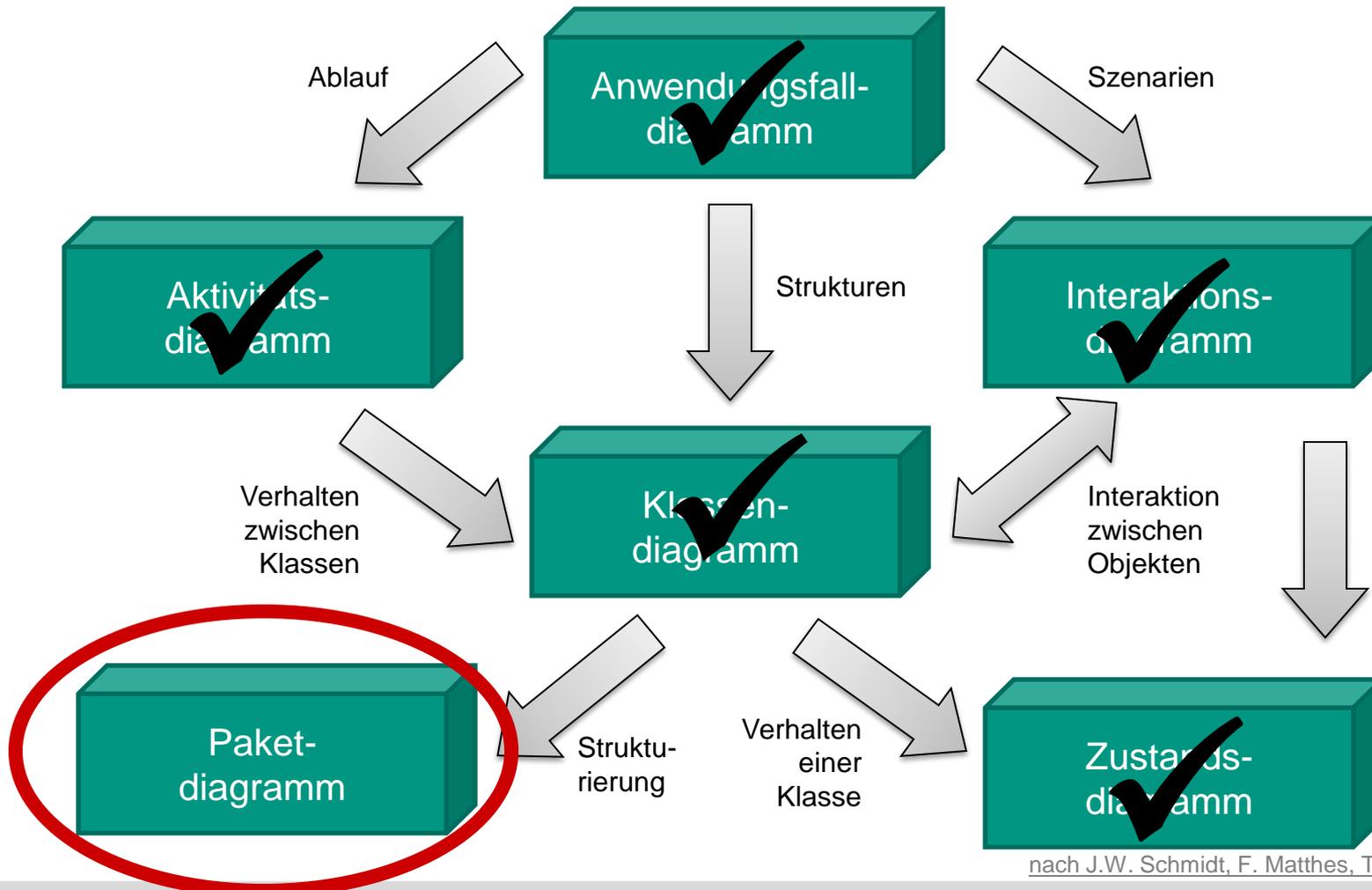
Weitere Möglichkeiten

■ Nebenläufigkeit

- Während System im Zustand G verweilt, kann es alle Zustandskombinationen aus E×F annehmen



UML-Diagramme

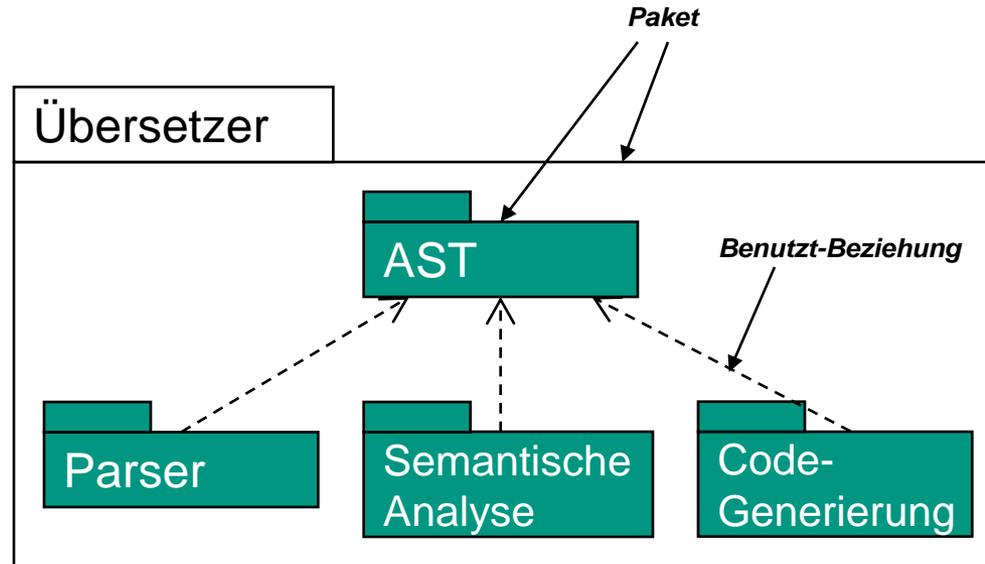


nach J.W. Schmidt, F. Matthes, TU Hamburg-Harburg

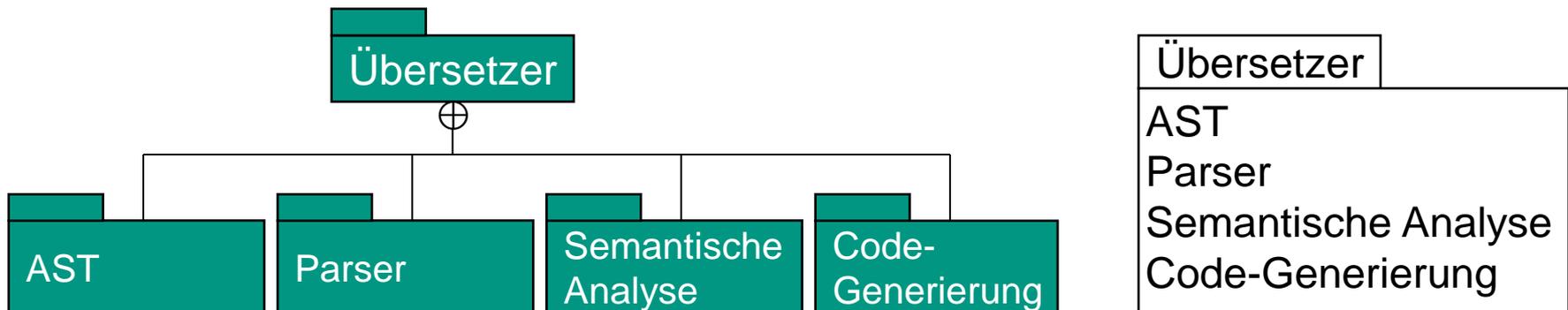
Paketdiagramm

- Pakete sind Ansammlungen von Modellelementen (ME) beliebigen Typs (z.B. Anwendungsfälle, Klassen, ...)
- Das Paketdiagramm dient der Gliederung des Gesamtmodells in überschaubare Einheiten
- Ein ME besitzt innerhalb des Pakets einen eindeutigen Namen
- Ein ME kann in anderen Paketen über seinen qualifizierten Namen zitiert werden:
 - Paketname::ME-Name
- In einem Paketdiagramm werden Abhängigkeiten zwischen Paketen mit einem gestrichelten Pfeil dargestellt.

Beispiel



Alternativ: (ohne Benutzt-Beziehungen)



Literatur

- Bernd Oestereich, „Analyse und Design mit UML 2.x – Objektorientierte Softwareentwicklung“, mit $x \geq 0$
- Harald Störrle, „UML 2 für Studenten“
- UML-Spezifikation unter <http://www.uml.org/#UML2.0>
 - Speziell „UML 2.0 Superstructure Specification“
 - und „UML 2.0 Infrastructure Specification“, insbes. Ch. 4 Terms and Definitions