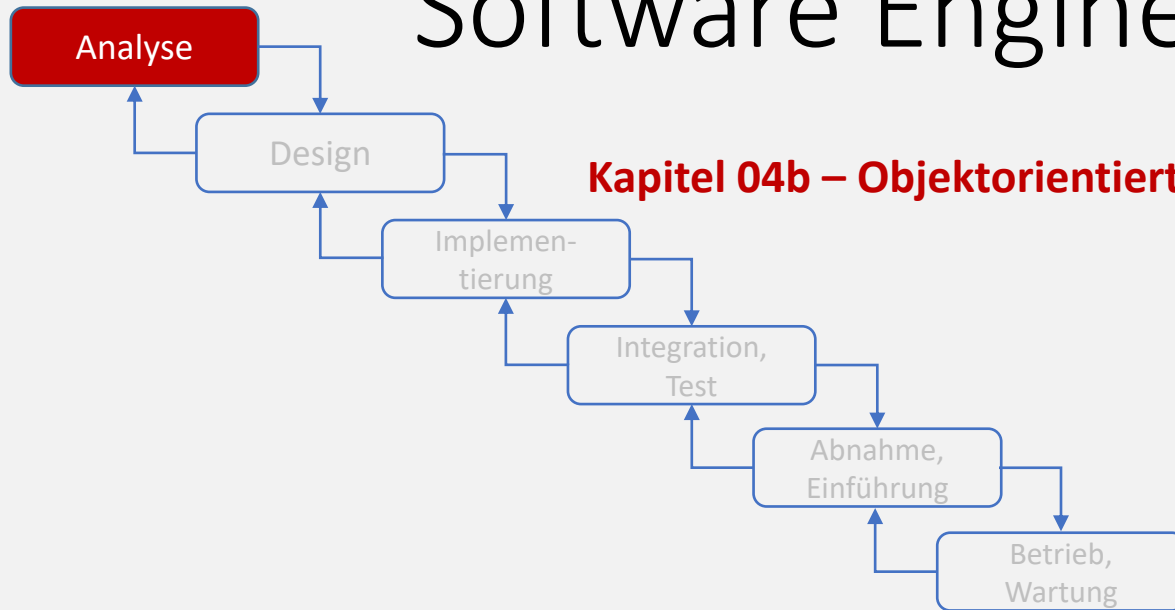




Software Engineering

Kapitel 04b – Objektorientierte Analyse (OOA)



Aktivitätsdiagramme

Wiederholung: Aufgabenbereiche des Makroprozesses

- Analyse im Großen
- 6 Schritte zum statischen Modell
- 4 Schritte zum dynamischen Modell

Erinnerung: Analyse im Großen

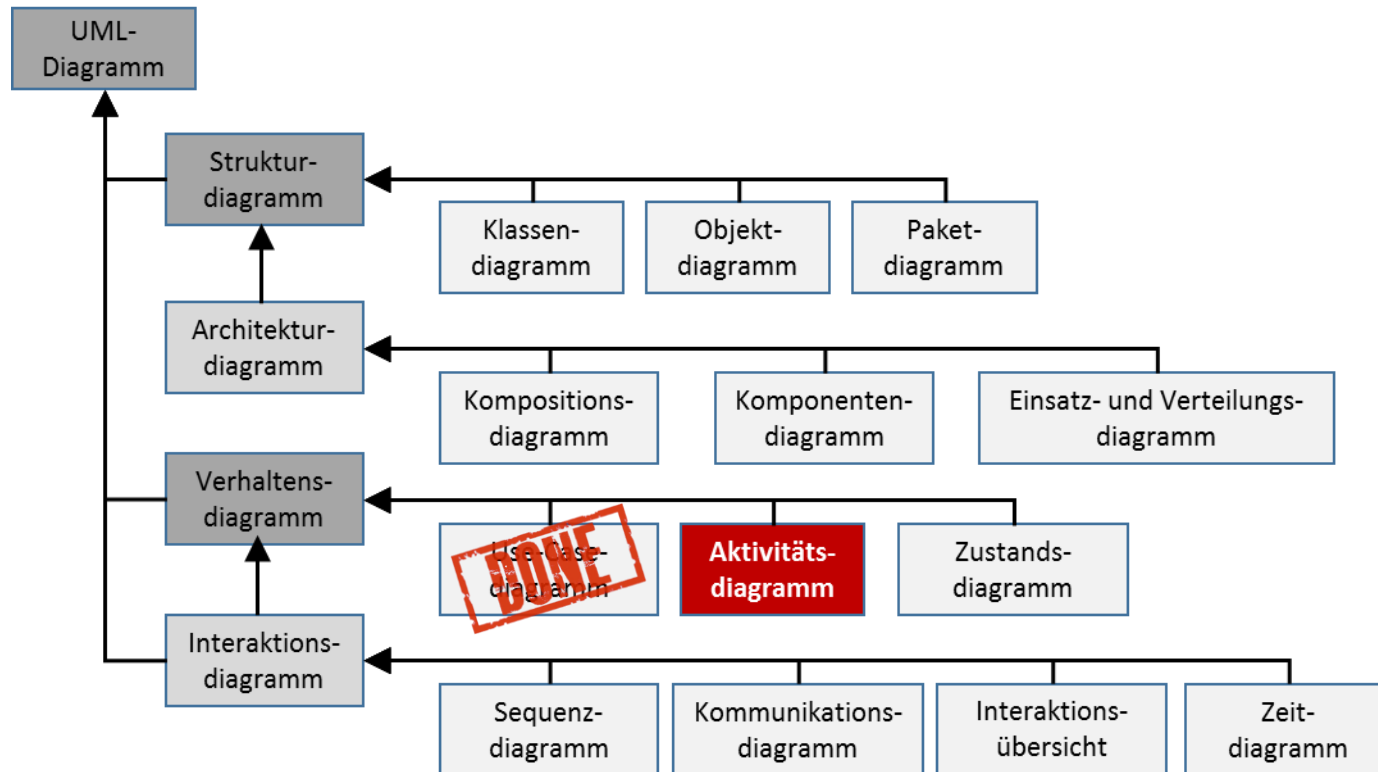
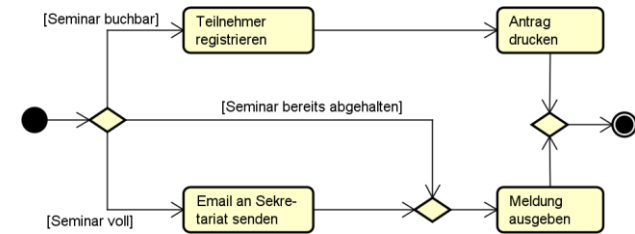
Use Case Modell aufstellen, liefert als Ergebnisse:

- Use-Case-Diagramm
- Use-Case-Beschreibung
- **Aktivitätsdiagramme**
- Zustandsdiagramm

Pakete bilden, beinhaltet:

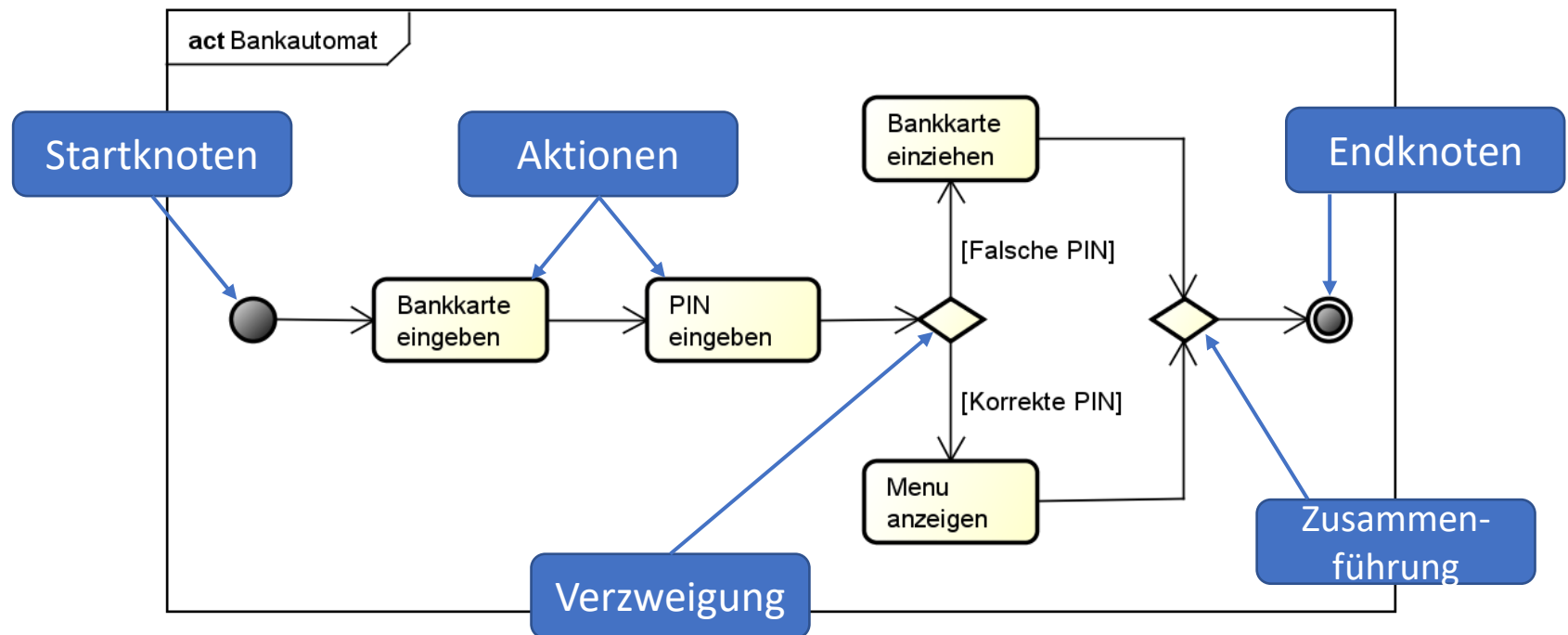
- Teilsysteme festlegen (Modellelemente zu Paketen zusammenfassen)
- bei großen Systemen erfolgt Paketbildung meist zu Anfang
- Ergebnis: Paketdiagramm

UML: Aktivitätsdiagramm



UML: Aktivitätsdiagramm

Beispiel




- Fokus: prozedurale Verarbeitungsaspekte
- Spezifikation von Kontroll- und Datenfluss zwischen Arbeitsschritten zur Realisierung einer Aktivität
- Basierend auf Petrinetzen, Grundlage für BPMN (s. Mastervorlesung "*Modeling and Verif.*")
- Zur Modellierung objektorientierter und nicht objektorientierter Systeme

UML: Aktivitätsdiagramme

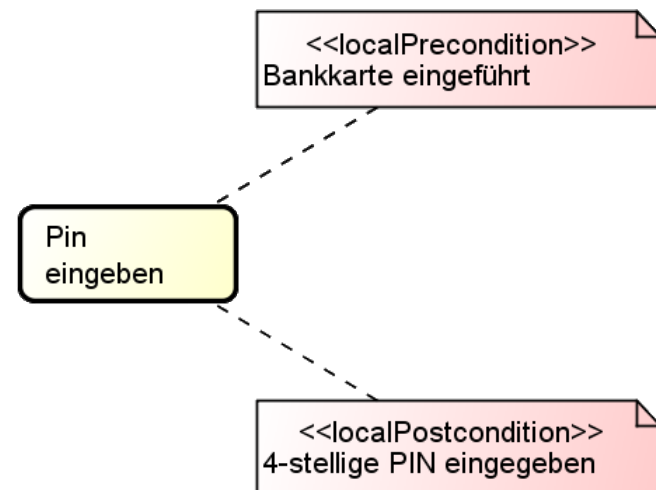
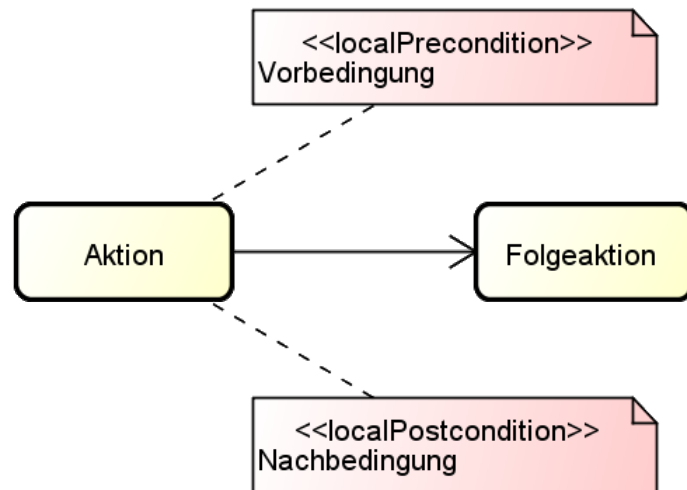
- Werden verwendet, um die **Abläufe**/Prozesse in Zusammenhang mit einem System zu **modellieren**
- Es können sowohl **sequentielle als auch parallele Abläufe** modelliert werden
- **Beispiele** für den Einsatz von Aktivitätsdiagrammen:
 - Abarbeitung eines Use Cases
 - Algorithmus einer Operation
 - Geschäftsvorfall
- Folgende Frage wird durch Aktivitätsdiagramme beantwortet:
 - „*Wie realisiert mein System ein bestimmtes Verhalten?*“

UML: Aktivitätsdiagramm-Elemente

Aktionsknoten

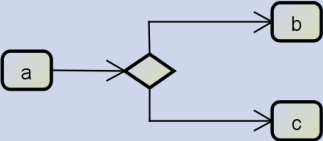
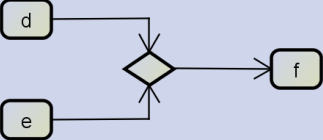
Notation	Name	Bedeutung
	Aktion	<ul style="list-style-type: none">- Zentrales Element im Aktivitätsdiagramm- Modelliert einen einzelnen Schritt einer Aktivität- Ist über Kanten mit anderen Elementen verbunden- Optional: Vor- und Nachbedingungen (bei Aktionsstart bzw. -ende)

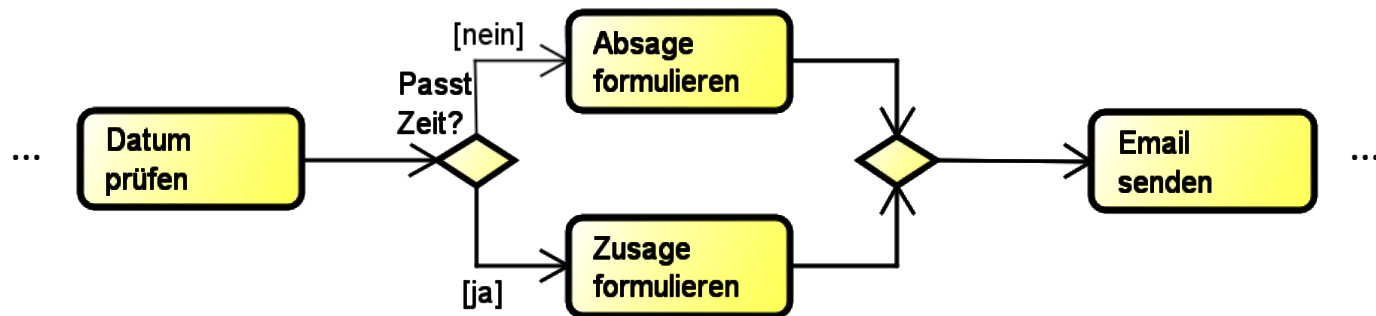
Beispiel:



UML: Aktivitätsdiagramm-Elemente

Verzweigung/Zusammenführung

Notation	Name	Bedeutung
	Verzweigung/ Entscheidung (Decision)	<ul style="list-style-type: none"> - Verzweigung des Kontrollflusses über das Rautensymbol (eine eingehende, n ausgehende Kanten) - Verzweigungsknoten muss Frage enthalten - Vor Verzweigung mittels Aktion die Informationen zur Beantwortung der Frage beschaffen - Ausgehende Kanten erhalten Guards (Bedingungen)
	Zusammenführung (Merge)	<ul style="list-style-type: none"> - Zusammenführung des Kontrollflusses über das Rautensymbol (n eingehende, 1 ausgehende Kante) - Decision und Merge treten immer paarweise auf - Pfade aus einer Decision sollten (dürfen) nicht in verschiedene Merge-Knoten führen

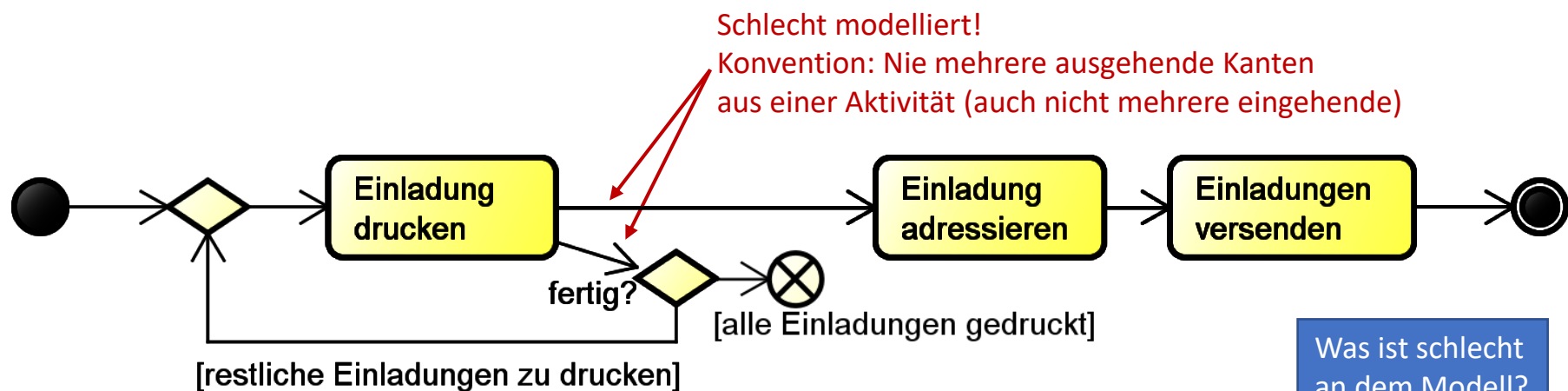


UML: Aktivitätsdiagramm-Elemente

Kontrollknoten



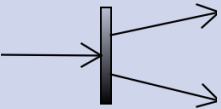
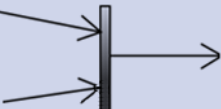
Notation	Name	Bedeutung
	Startknoten	<ul style="list-style-type: none"> - Startpunkt eines Ablaufs in einem Aktivitätsdiagramm - Pro Diagramm (mindestens) ein Startknoten
	(Aktivitäts-) Endknoten	<ul style="list-style-type: none"> - Beendet alle Abläufe einer Aktivität - Pro Diagramm mindestens ein Endknoten
	Ablauf- endknoten	<ul style="list-style-type: none"> - Beendet einen einzelnen Kontrollfluss - Pro Diagramm 0 bis beliebig viele Ablaufendknoten



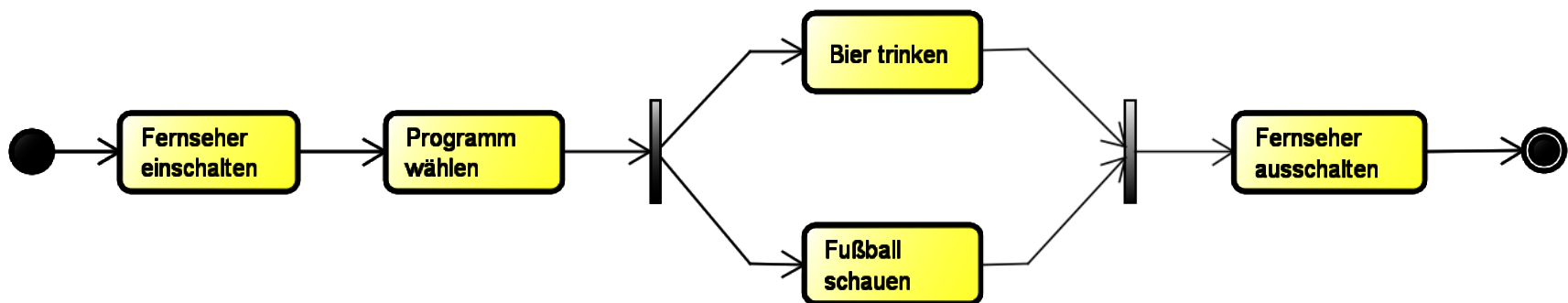
Was ist schlecht
an dem Modell?

UML: Aktivitätsdiagramm-Elemente

Fork und Join




Notation	Name	Bedeutung
	Teilung (Fork)	<ul style="list-style-type: none"> - Eingehender Kontrollfluss wird auf mehrere ausgehende, nebenläufige Kontrollflüsse aufgeteilt - Erlaubt parallele Ausführung mehrerer Kontrollfl. - Keine Bedingungen vorhanden (vgl. Decision)
	Synchronisation (Join)	<ul style="list-style-type: none"> - Wartet auf alle eingehenden Kontrollflüsse, bevor mit Ausgehendem fortgefahren wird - Synchronisiert parallele Kontrollflüsse

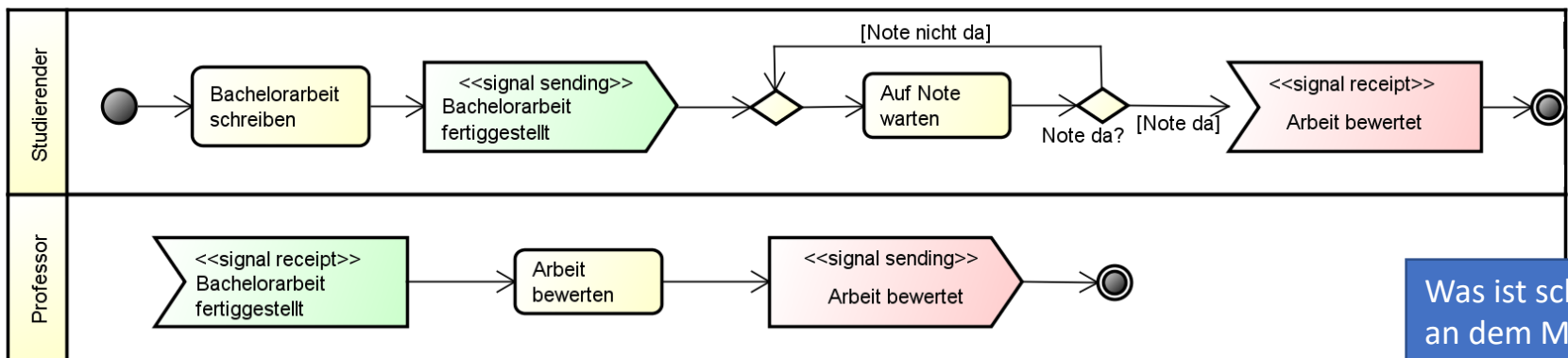
Beispiele?



UML: Aktivitätsdiagramm-Elemente

Signale und Swimlanes

Notation	Name	Bedeutung
	Signal senden	<ul style="list-style-type: none"> - Aktion, die ein Signal sendet - Signale immer wenig/vorsichtig einsetzen! (Warum?)
	Signal empfangen	<ul style="list-style-type: none"> - Aktion, die auf ein Signal wartet - Falls keine eingehende Kante existiert, können beliebig oft Signale empfangen werden
	Partition (Swimlane)	<ul style="list-style-type: none"> - Partitionierung der Aktionen in gekennzeichnete Bereiche (Verantwortlichkeitsbereiche) - Z.B.: Verschiedene Abteilungen in Geschäftsprozess - Darstellung: horizontal oder vertikal



Was ist schlecht an dem Modell?

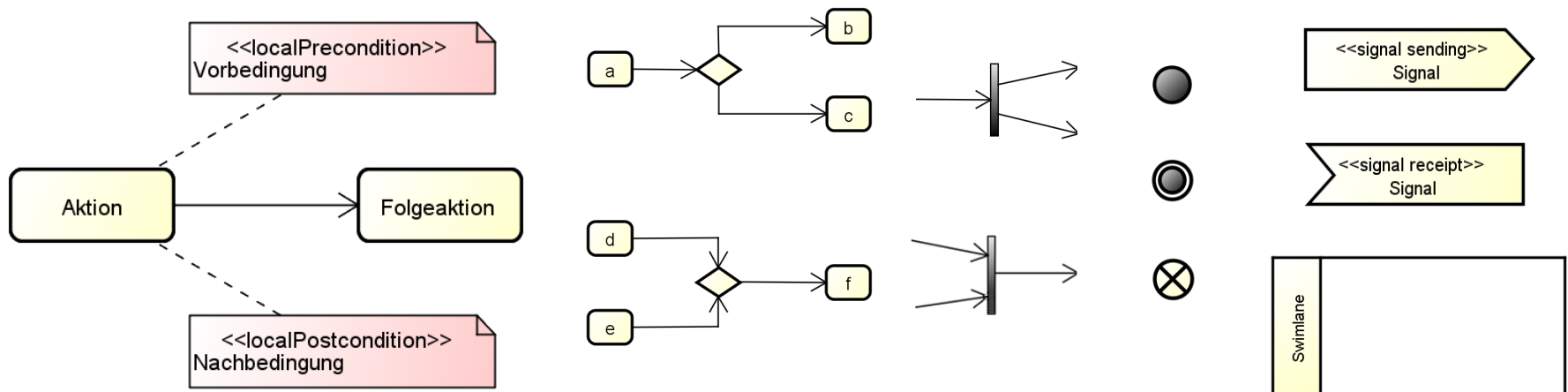
Beispiel: Partytime



1. Erstellen Sie ein UML-Aktivitätsdiagramm für folgenden Zusammenhang:

- Bernd erhält eine Einladung zu einer Party ...
- und prüft zunächst, ob das Datum passt.
- Falls ja, macht er sich Gedanken, ob er Lust verspürt zur Feier zu gehen, falls nein, sagt er umgehend ab.
- Hat er Lust auf Party, so sagt er zu, anderenfalls sagt er auch ab.

Übersicht über die bisherigen Elemente:



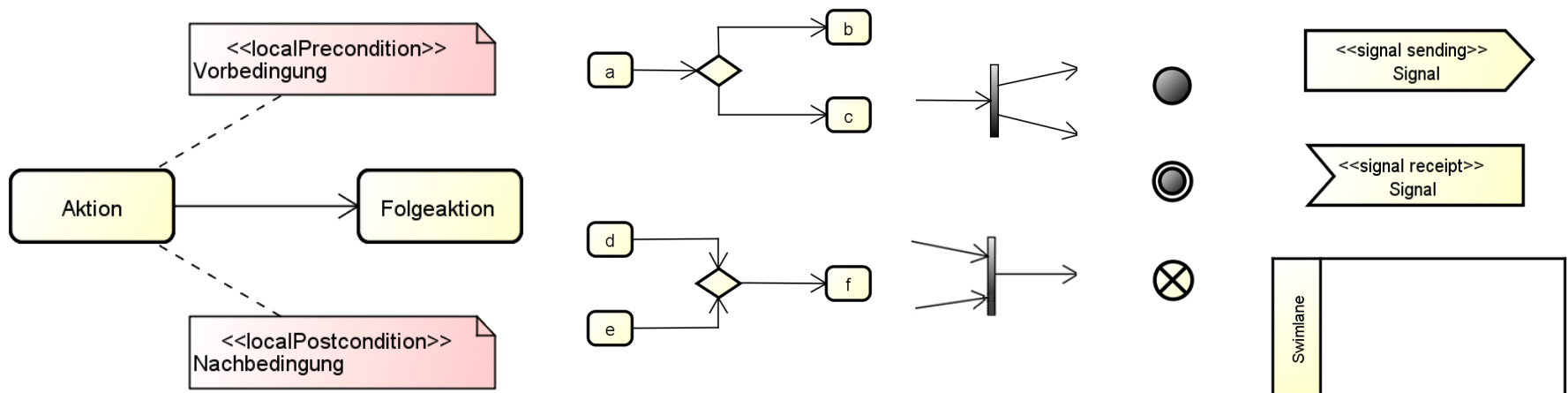
Beispiel: Partytime



2. Erstellen Sie ein weiteres Aktivitätsdiagramm für eine Partyvorbereitung:

- Zur Einstimmung auf die Party beginnt der Gastgeber Wein zu trinken.
- Gleichzeitig beginnt er die Essenzubereitung und stellt auch schon einmal die Getränke bereit.
- Irgendwann ist der Vorrat an Wein erschöpft.
- Nachdem die Getränke kaltgestellt und das Essen zubereitet wurde, kann die Party ausgelassen gefeiert werden.
- Am Ende muss die Wohnung natürlich wieder aufgeräumt werden.

Übersicht über die bisherigen Elemente:



UML: Aktivitätsdiagramme

“Token-Semantik”


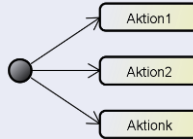


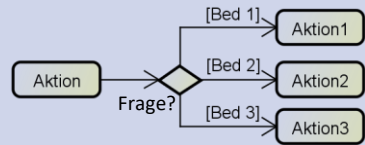
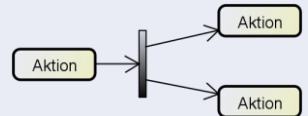
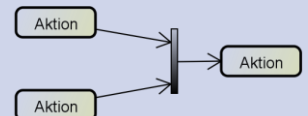
Token in Aktivitätsdiagrammen:

- Virtueller Koordinationsmechanismus
- Beschreiben Abläufe einer Aktivität zur Laufzeit
- Werden in Startknoten geboren
- Fließen entlang von Kanten vom Vorgängerknoten zu Nachfolgerknoten
- Ggf. unterteilt in Kontroll- und Datentoken:
 - Kontrolltoken: “Ausführungserlaubnis” für den Nachfolgeknoten
 - Datentoken: Transport von Datenwert (oder Referenz auf Objekt)
- Müssen an allen eingehenden Kanten einer Aktion vorliegen, um deren Ausführung zu ermöglichen
- *“Ergebnisse einer Aktivität”*
- Sterben in End- oder Ablaufendknoten



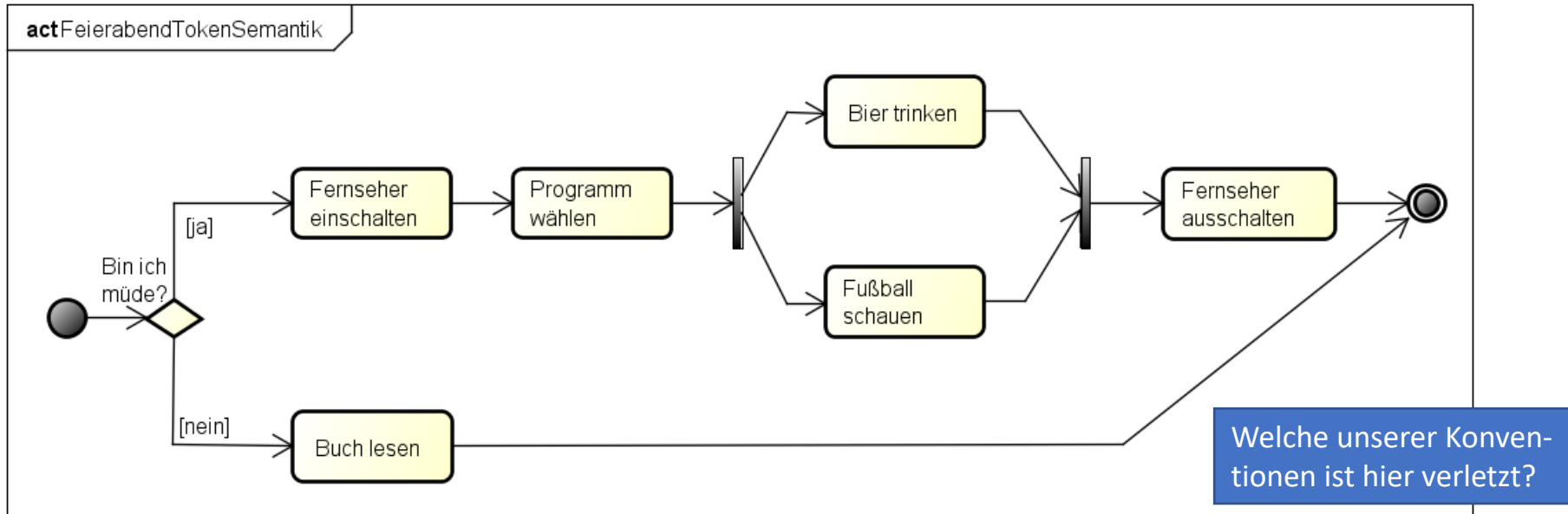
UML: Aktivitätsdiagramme

“Token-Semantik”-Überblick

Notation	Bedeutung
	<ul style="list-style-type: none"> - Token gelangt zu „Aktion“ führt diese aus - Sobald Aktion beendet, wandert es entlang der ausgehenden Kante weiter
	<ul style="list-style-type: none"> - Jede aus Startknoten führende Kante erhält ein Token (Geburt)
	<ul style="list-style-type: none"> - Token das in Endknoten läuft wird konsumiert (Tod) - Erreicht ein Token einen Endknoten wird Prozess/Instanz des Diagramms beendet
	<ul style="list-style-type: none"> - Token das in Ablaufendknoten läuft wird konsumiert (Tod) - Die Instanz des Diagramms läuft weiter, sofern noch Token vorhanden sind
	<ul style="list-style-type: none"> - Weiche für Tokenfluss (→ Schleifen realisierbar) - Bedingungen müssen sich gegenseitig ausschließen ([else] nutzen!) - Das Gegenstück der Weiche ist die Zusammenführung (auch Rautensymbol)
	<ul style="list-style-type: none"> - Modellierung der Aufspaltung von Abläufen - Eingehendes Token teilt sich, so dass alle Ausgänge ein Token erhalten
	<ul style="list-style-type: none"> - Führt nebenläufige Token zusammen - Vereinigung der Kontrolltoken, sobald an allen eingehenden Kanten ein Token - Nur ein Token wird weitergeleitet (Datentoken werden alle weitergeleitet)

UML: Aktivitätsdiagramme

“Token-Semantik”-Beispiel



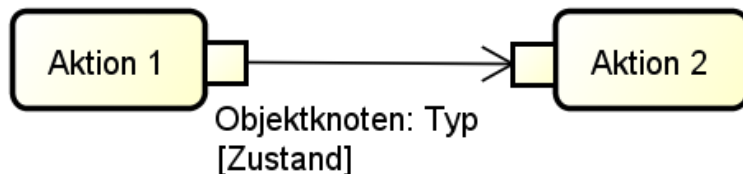
- Zu Beginn werden alle vom Initialknoten ausgehenden Kanten mit einem Token belegt
- Eine Aktion, bei der alle eingehenden Kanten mit Token belegt sind, ist aktiviert und kann durchgeführt werden
- Vor der Durchführung „nimmt“ sich die Aktion von jeder eingehenden Kante ein Token; nach der Durchführung belegt die Aktion jede ausgehende Kante mit einem Token
- Ein Entscheidungsknoten gibt den Token an **eine** ausgehende Kante weiter
- Ein Vereinigungsknoten reicht jeden Token, den er bekommt, einzeln weiter
- Ein Parallelisierungsknoten dupliziert den Token für **jede** ausgehende Kante
- Ein Synchronisierungsknoten wartet, bis alle eingehenden Kanten Token aufweisen und gibt dann eines weiter
- Endknoten beendet mit dem ersten Token, das er erhält, den gesamten Ablauf

UML: Aktivitätsdiagramme

Objektknoten

Objektknoten:

- Motivation:
 - Modellierung von Daten
 - Bindeglied zwischen statischem und dynamischem Modell
- Inhalt eines Objektknotens: Ergebnis einer Aktion und Eingabe für weitere Aktion
- Typangabe, Zustandseinschränkung und Effekte (**c**reate, **r**ead, **u**ppdate, **d**eleate) sind optional
- Logisches Gerüst, um Daten und Werte zu transportieren
- Notationsvarianten im Hinblick auf Aktionsknoten:



**Konvention: Eingabepins links oder oben
Ausgabepins rechts oder unten**

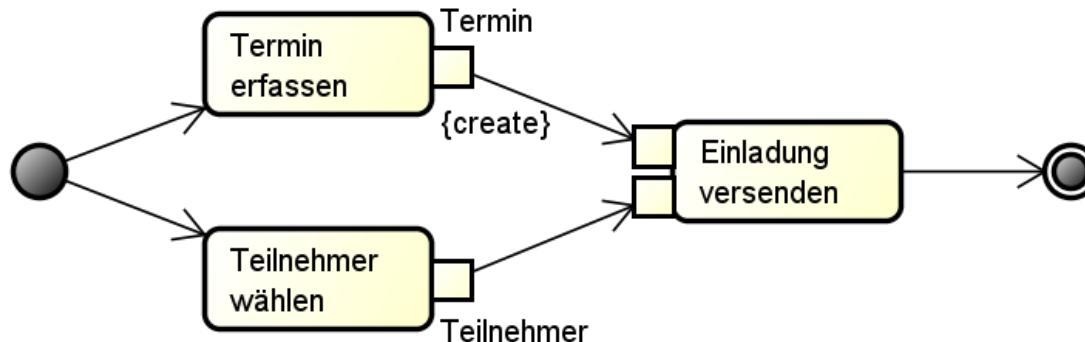
UML: Aktivitätsdiagramme

Objektknoten: Beispiel

Hinweise zu Objektknoten:

- Pfeil von Aktion zu Objektknoten: Ausprägung von Typ des Objektknotens ist **Ergebnis der Aktion**
- Pfeil von Objektknoten zu Aktion: Ausprägung von Typ des Objektknotens ist **Eingabeparameter für Aktion**
- Sind mehrere Objektknoten am Eingang, müssen alle Parameter einen Wert haben, bevor Aktion ausgeführt werden kann

Beispiel:



- Durch Aktion “Termin erfassen” wird ein Objekt vom Typ “Termin” als Ergebnis geliefert
- Dieser gelieferte Termin dient als Eingabeparameter für Folgeaktion “Einladung versenden”

UML: Aktivitätsdiagramme

Kontrollfluss vs. Objektfluss

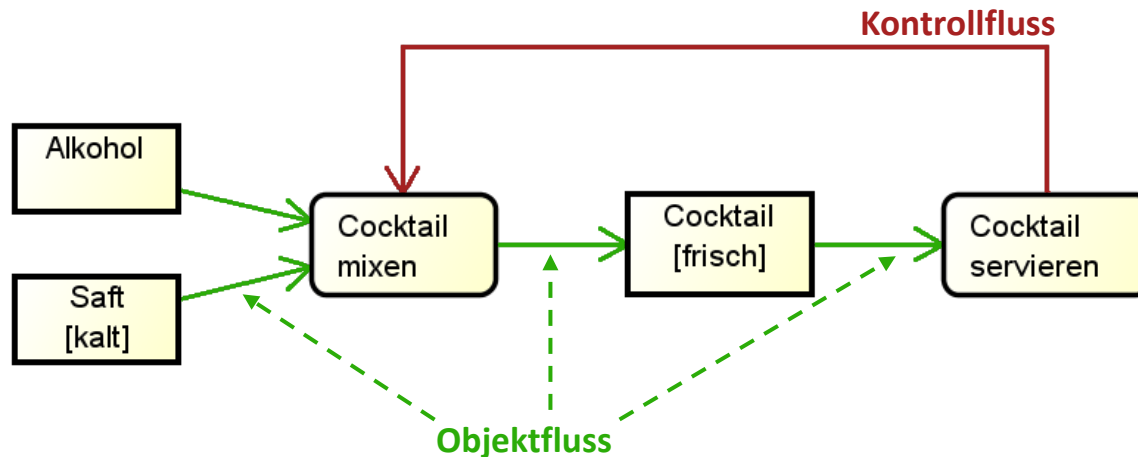
- **Kontrollfluss:**

- Kante zwischen zwei Aktionen oder zwischen Aktion und Kontrollelement
- Token tragen hier keine Daten und dienen nur der Aktionsausführung

- **Objektfluss:**

- Kante mit mindestens einem Objektknoten
- Kante trägt Token, das Daten zu oder von Objektknoten transportiert

- **Beispiel:** (Welche Kanten sind Kontroll-, welche Objektflüsse)



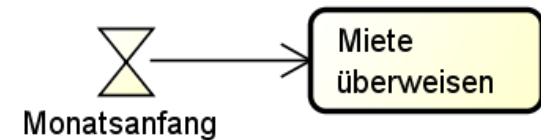
Welche unserer Konventionen ist hier verletzt?

UML: Aktivitätsdiagramme weitere Elemente (1)

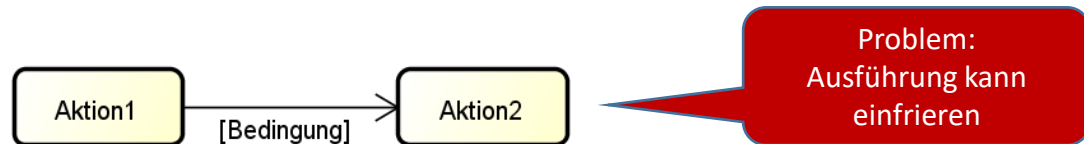
- **Asynchrones Zeitereignis:**



Beispiel:



- **Bedingte Kanten:**

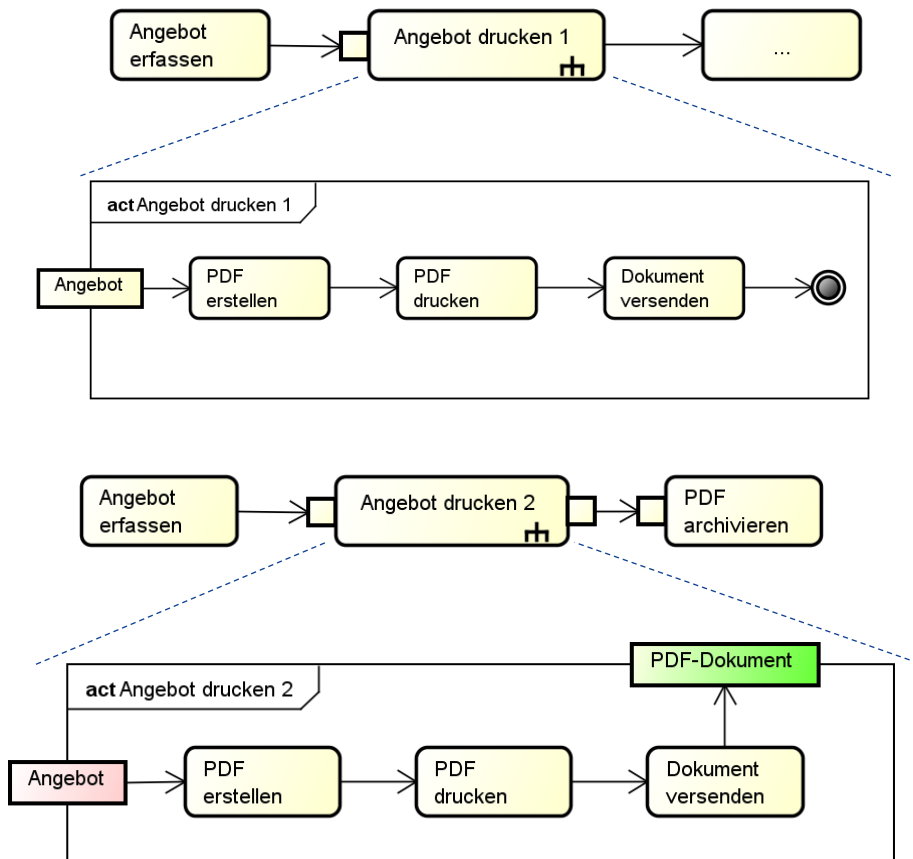


- **Sprungmarken:**



UML: Aktivitätsdiagramme weitere Elemente (2)

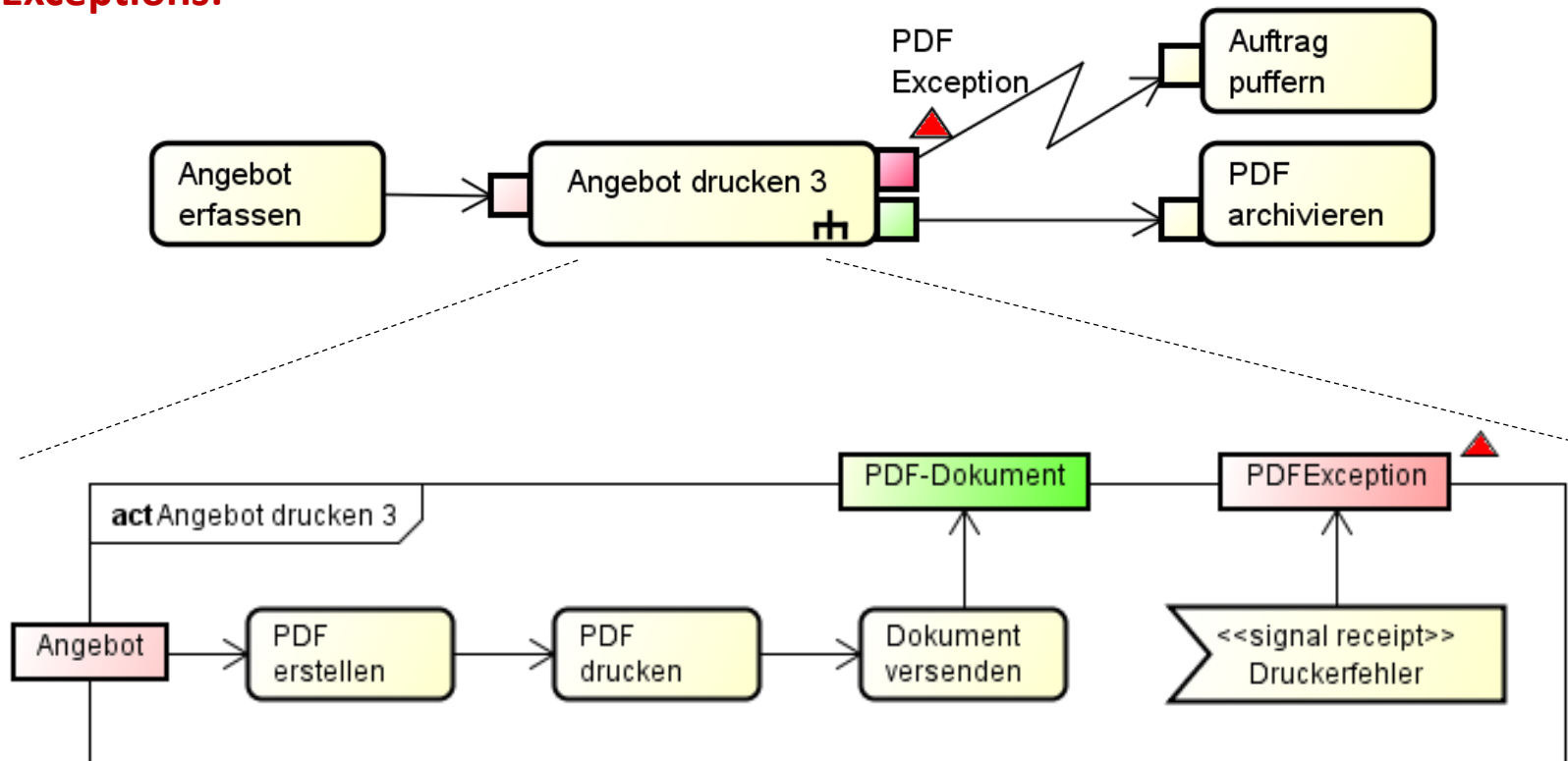
- **Aufruf von Aktivitäten und Aktivitäten mit Parametern:**



- Aktivitäten können wieder Aktivitäten aufrufen (Hierarchischer Aufbau)
- Details können in tiefere Ebenen verlagert werden
- Vorteile:
 - Übersichtlichkeit/bessere Lesbarkeit
 - Wiederverwendbarkeit
- Dabei können Parameter (Eingabe/Ausgabe) genutzt werden

UML: Aktivitätsdiagramme weitere Elemente (3)

- **Exceptions:**



UML: Aktivitätsdiagramme

weitere Elemente (4)

- **Unterbrechungsbereich mit Unterbrechungskante:**



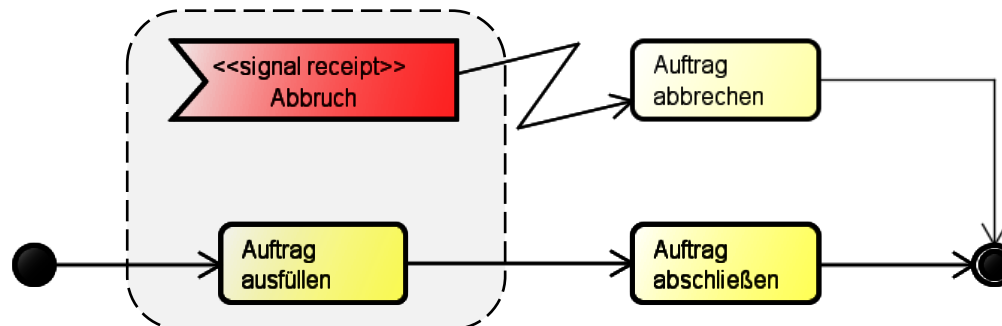
oder:



- Umschließt eine Menge von Aktionen
- Wird Unterbrechungsbereich durch Unterbrechungskante verlassen, so werden alle darin befindlichen Token gelöscht
- Aktivitätsablauf wird an Zielknoten fortgesetzt

- Ziel: Unterbrechungen modellieren (z.B. Exceptions)
Verhalten bei Fehler beschreiben

- **Beispiel:**



Beispiel: Klausur



Modellieren Sie folgenden Ablauf als UML-Aktivitätsdiagramm:

Kurz vor Beginn der Klausur nehmen wir den Platz ein und anschließend werden die Klausuren ausgeteilt. Während Sie anfangen, die Klausur zu lösen, legen Sie Ihren Studierendenausweis sichtbar vor sich. Am Ende packen Sie Ihre Sachen, verlassen den Hörsaal und gehen nach Hause. Nun kann es während der Klausurzeit aber auch passieren, dass ein Feueralarm eintritt und alle den Hörsaal schnell verlassen müssen. Auch in diesem Fall gehen Sie nach Hause.

Beispiel: Essen zahlen in der Mensa

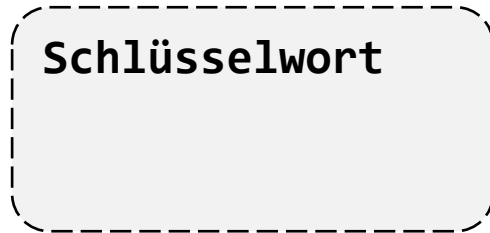


Modellieren Sie den Zahlungsvorgang in der Mensa als Aktivitätsdiagramm:

- Nutzen Sie dabei zur übersichtlicheren Darstellung Swimlanes (Kasse, Displaykasse, Kartenleser)
- Bedenken Sie auch den Sonderfall, dass die Karte nicht genug Geld aufweist
- Verwenden Sie insgesamt ca. 10 Aktivitäten

Der Vollständigkeit halber: Strukturierte Knoten

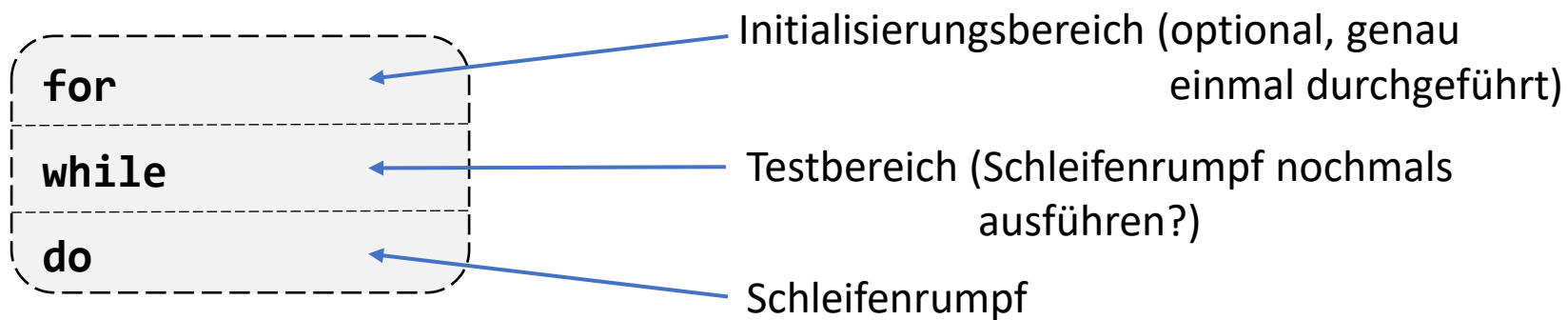
- **Strukturierte Knoten:**



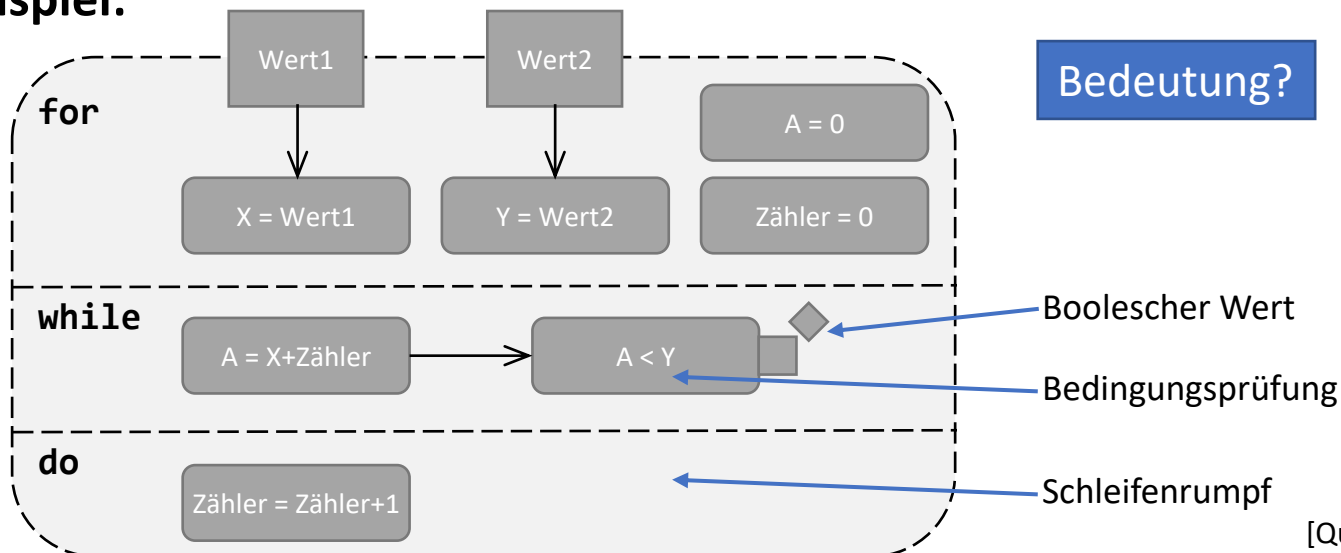
- Darstellung durch Rechteck mit abgerundeten Ecken und unterbrochener Linienführung
- Grundidee: Gruppierung mehrere Elemente zu einer Ausführungseinheit
- Darstellung als spezieller Knoten im Kontrollfluss
- Das „**Schlüsselwort**“ gibt Art des Knotens an (z.B.: for, if, ...)

Der Vollständigkeit halber: Strukturierte Knoten

- **Strukturierte Knoten: Schleifen**



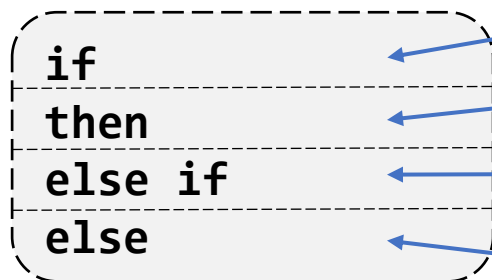
- **Beispiel:**



[Quelle: UML 2 glasklar]

Der Vollständigkeit halber: Strukturierte Knoten

- **Strukturierte Knoten: Entscheidungen**



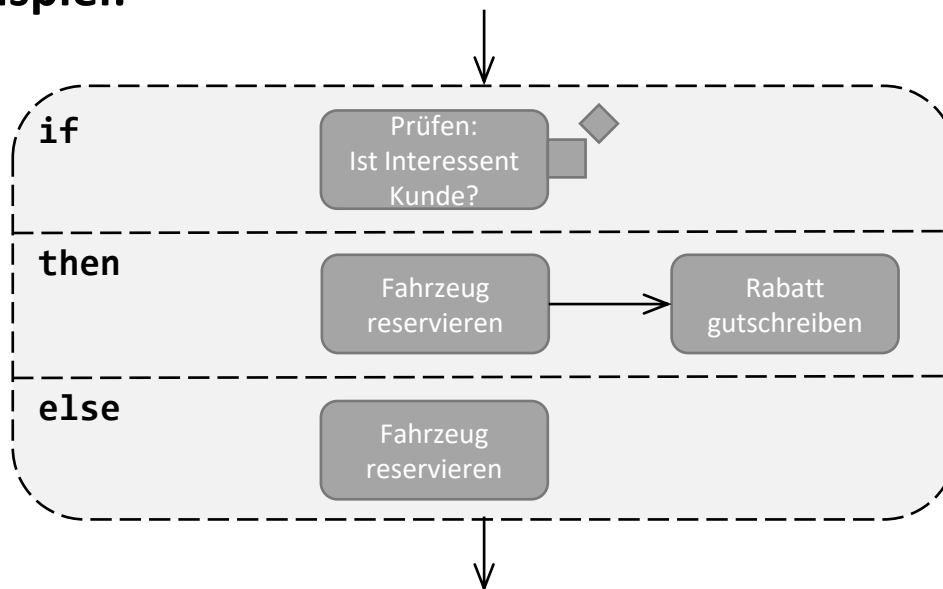
Bereich der Bedingungsprüfung

Bereich mit auszuführenden Aktionen

Wird nur ausgeführt, wenn davorliegende Bedingung scheitert

Wird nur ausgeführt, wenn keine der davorliegenden Bedingungen erfüllt war

- **Beispiel:**



[Quelle: UML 2 glasklar]

UML: Aktivitätsdiagramme

Konstruktion (1)

Idee:

- Interaktion eines Use Cases bzw. Zusammenspiel zwischen Use Cases wird über Aktivitätsdiagramm modelliert

Notwendige Schritte (Standardelemente):

1. Auslösendes Ereignis bestimmen:

- Startknoten eintragen und Vorbedingungen ermitteln

2. Terminierende Ereignisse bestimmen:

- Ziel bestimmen, das im Erfolgsfall erreicht werden soll
- Terminierung festlegen (terminiert gesamte Verarbeitung oder nur aktueller Pfad)

3. Standardfall modellieren

4. Erweiterungen zum Standardfall bestimmen

- Aktionen bestimmen, die nur unter bestimmten Bedingungen ausgeführt werden
- Stellen bestimmen, an denen Kontrollfluss wieder zusammenführt

5. Möglichkeiten zur parallelen Verarbeitung finden:

- Stellen finden, für die Verarbeitungsreihenfolge irrelevant ist (echte Parallel. möglich?)
- Stellen bestimmen, an denen parallele Zweige synchronisiert werden

UML: Aktivitätsdiagramme

Konstruktion (2)

Idee:

- Interaktion eines Use Cases bzw. Zusammenspiel zwischen Use Cases wird über Aktivitätsdiagramm modelliert

Notwendige Schritte (Fortgeschrittene Elemente):

6. Verantwortungsbereiche festlegen:

- Swimlanes verwenden, um Verantwortlichkeitsbereiche zu verdeutlichen

7. Objektknoten bestimmen:

- Prüfen, ob Aktionen Eingabedaten benötigen oder Ausgabedaten erzeugen
- Objektknoten nutzen, um Aussagekraft des Diagramms zu erhöhen

8. Ein-/Ausgabeparameter bestimmen:

- Ein-/Ausgabeparameter eintragen, falls so Aussagekraft des Diagramms erhöht wird

9. Exceptions und Unterbrechungsbereiche ermitteln:

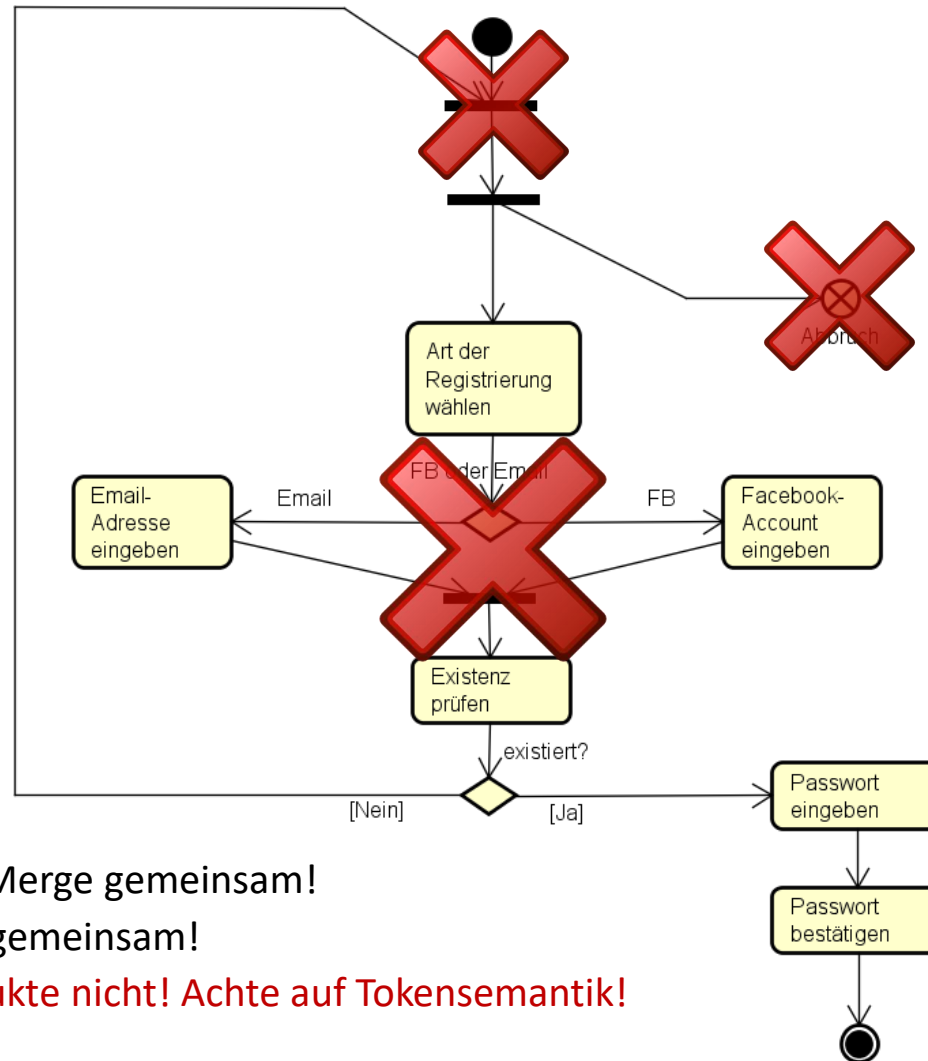
- Exceptions und Unterbrechungsbereiche nutzen, um Fehlerbehandlung zu verdeutlichen

UML: Aktivitätsdiagramme

Zusätzliche Konventionen

- **Sprechende Aktivitätsbezeichner:**
 - Aussagekräftiges Substantiv + Verb
 - Im Kontext der Aktivität eindeutig und verständlich
- **Vermeidung von Black-Holes und Miracles:**
 - Black-Holes: Aktionen ohne ausgehende Kante
 - Miracles: Aktionen ohne eingehende Kante
- **Korrekte Verzweigungen:**
 - Bedingungen müssen disjunkt sein
 - Standardfall (“[else]”) einführen

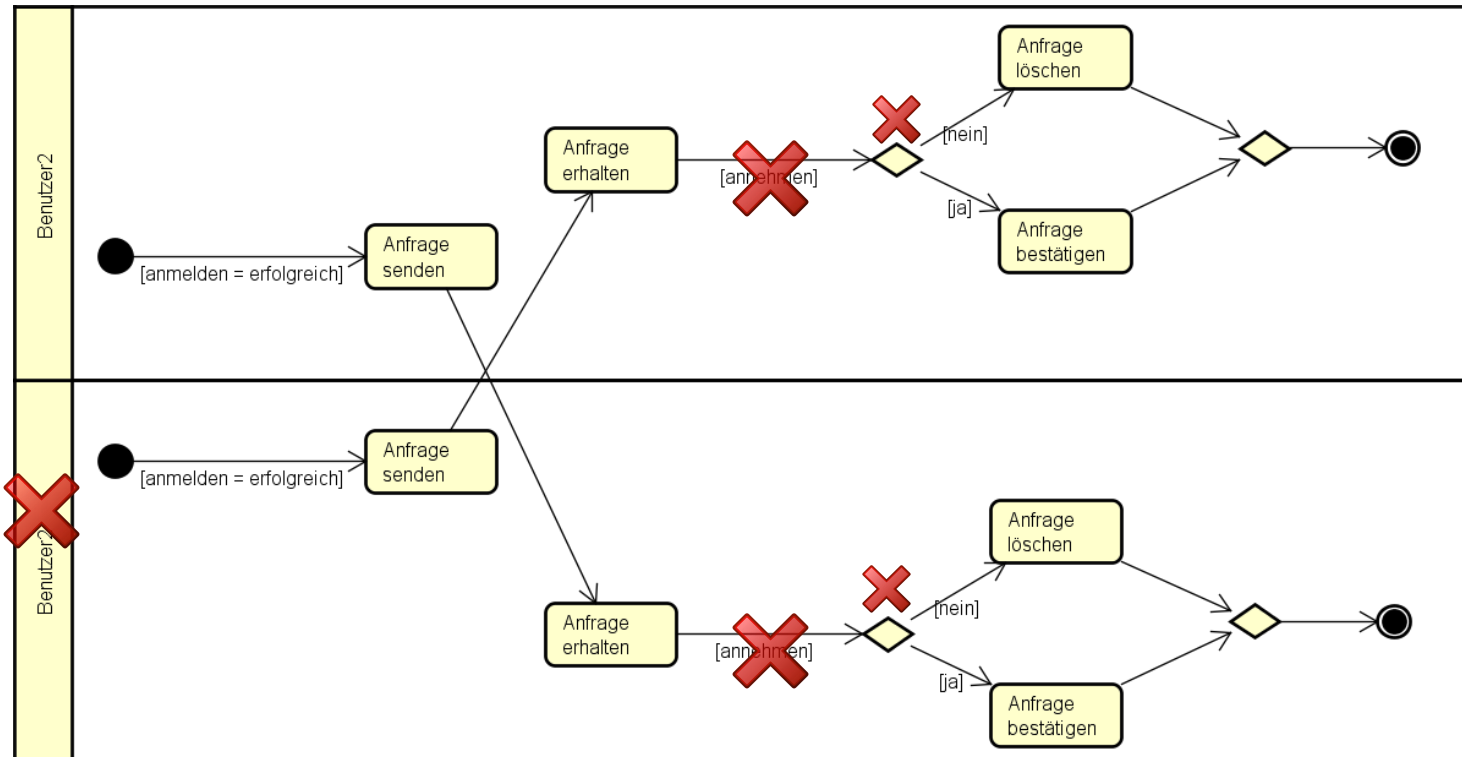
Aktivitätsdiagramme: Typische Modellierungsfehler (1)



Einige Verbesserungsvorschläge:

- Nutze Decision und Merge gemeinsam!
- Nutze Fork and Join gemeinsam!
- **Mische diese Konstrukte nicht! Achte auf Tokensemantik!**

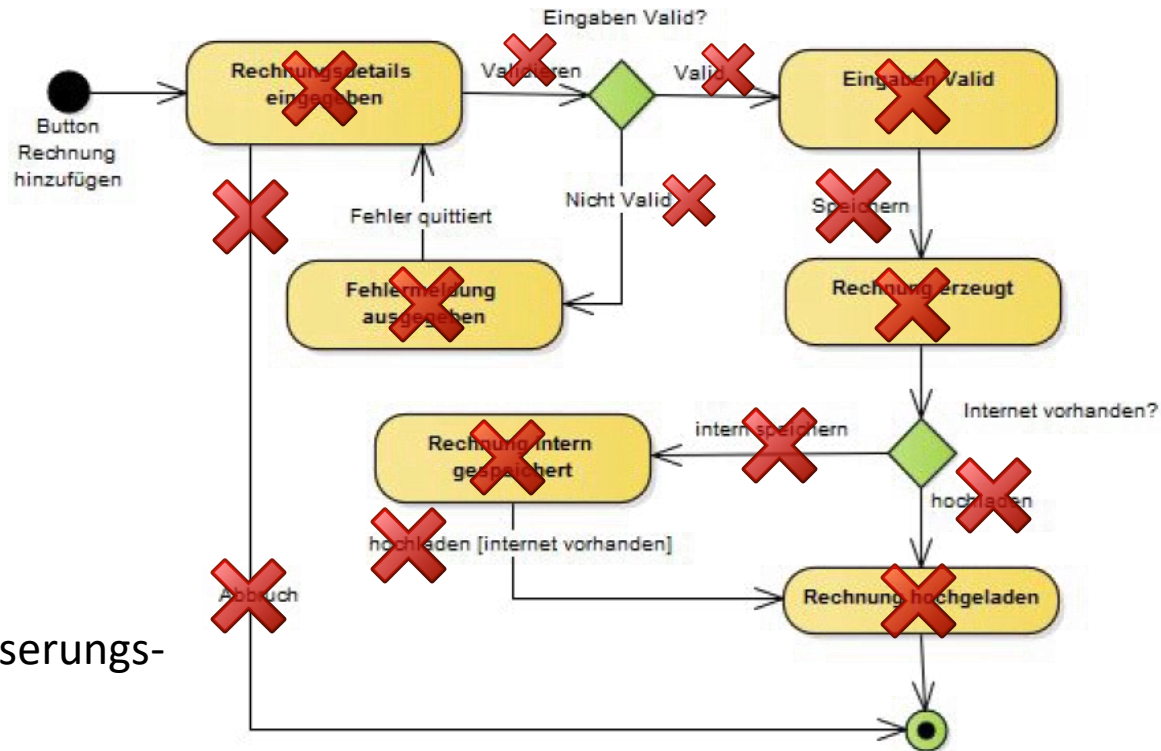
Aktivitätsdiagramme: Typische Modellierungsfehler (2)



Einige Verbesserungsvorschläge:

- Vermeide Kanten mit Bedingungen! (Gefahr von Deadlocks)
- Beschrifte Decisions mit Fragen! (Fehlt im Beispiel)
- Beschaffe Information für Beantwortung der Frage in Vorgängeraktivität
- Benenne Lanes eindeutig!

Aktivitätsdiagramme: Typische Modellierungsfehler (3)



Einige Verbesserungsvorschläge:

- Aktivitäten dürfen nur eine ausgehende Kante haben!
- Bedingungen als Guards markieren!
- Aktivitäten mit Substantiv + Verb (Infinitiv) beschr.
- Nicht jede Kante beschriften!
- Diagramm und Fluss strukturieren

(Semantik sonst nicht klar)

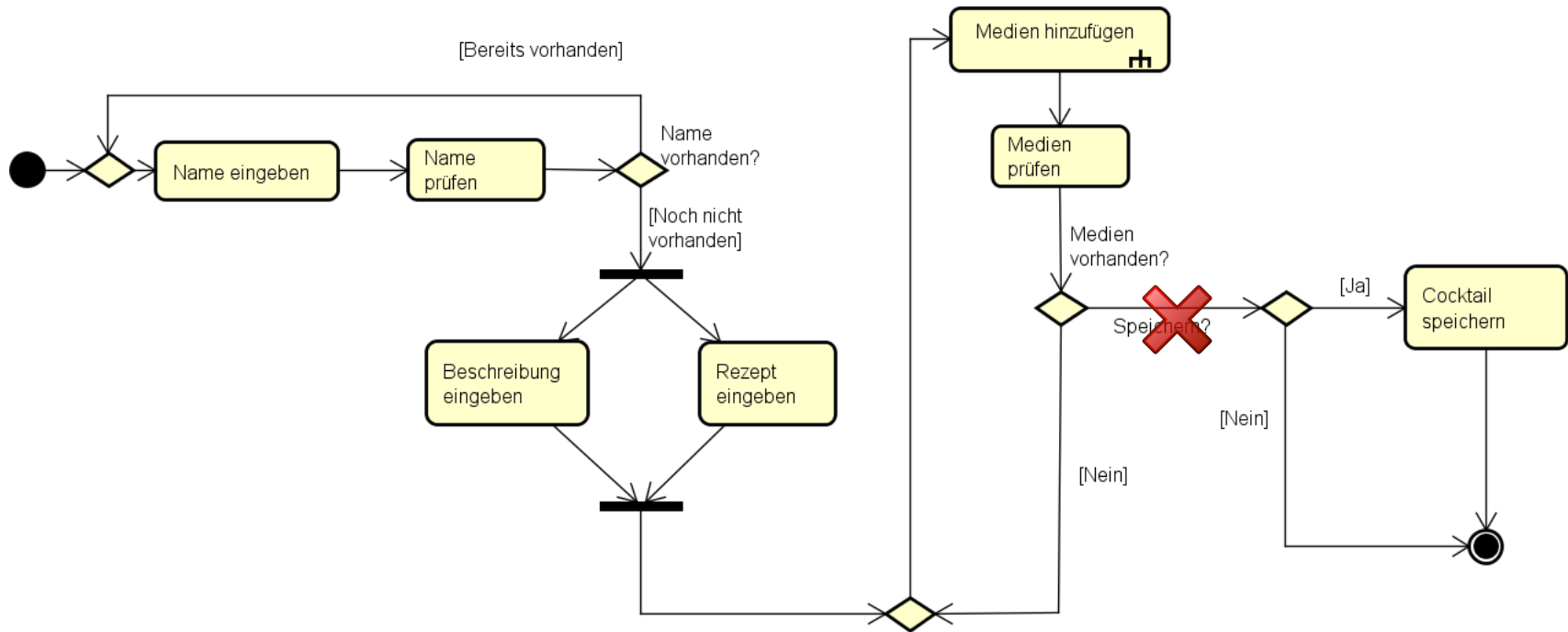
(Semantik sonst nicht klar)

(Sonst keine Aktivität)

(Sonst unübersichtlich)

(Sonst unübersichtlich)

Aktivitätsdiagramme: Typische Modellierungsfehler (4)



Literatur

- *UML 2 glasklar*, Chris Rupp et al., Hanser, 2012
- *Lehrbuch der Objektmodellierung*, Heide Balzert, Spektrum Akademischer Verlag, 2011