

# VORLESUNG

## Automatisierung industrieller Workflows

### Teil B: Die Sprache UML

Joachim Fischer

# Inhalt

- ⊙ **Teil A**  
Aspekte von Modellierung und Simulation dynamischer Systeme

- ⊙ **Teil B**  
Die Modellierungssprache UML

- ⊙ **Teil C**  
Die ausführbare Modellierungssprache SLX

- ⊙ **Teil D**  
Modellierung von Lieferketten

- ⊙ **B.1**  
Wozu UML im Kontext der Computersimulation?
- ⊙ **B.2**  
UML-Teilsprachen, Sprachkonzepte
- ⊙ **B.3**  
Klassendiagramme
  - ⊙ Klassen und Assoziationen
  - ⊙ Templates
- ⊙ **B.4**  
Verhaltensbeschreibung mit Zustandsautomaten
- ⊙ **B.5**  
OCL

# Zwischenfazit

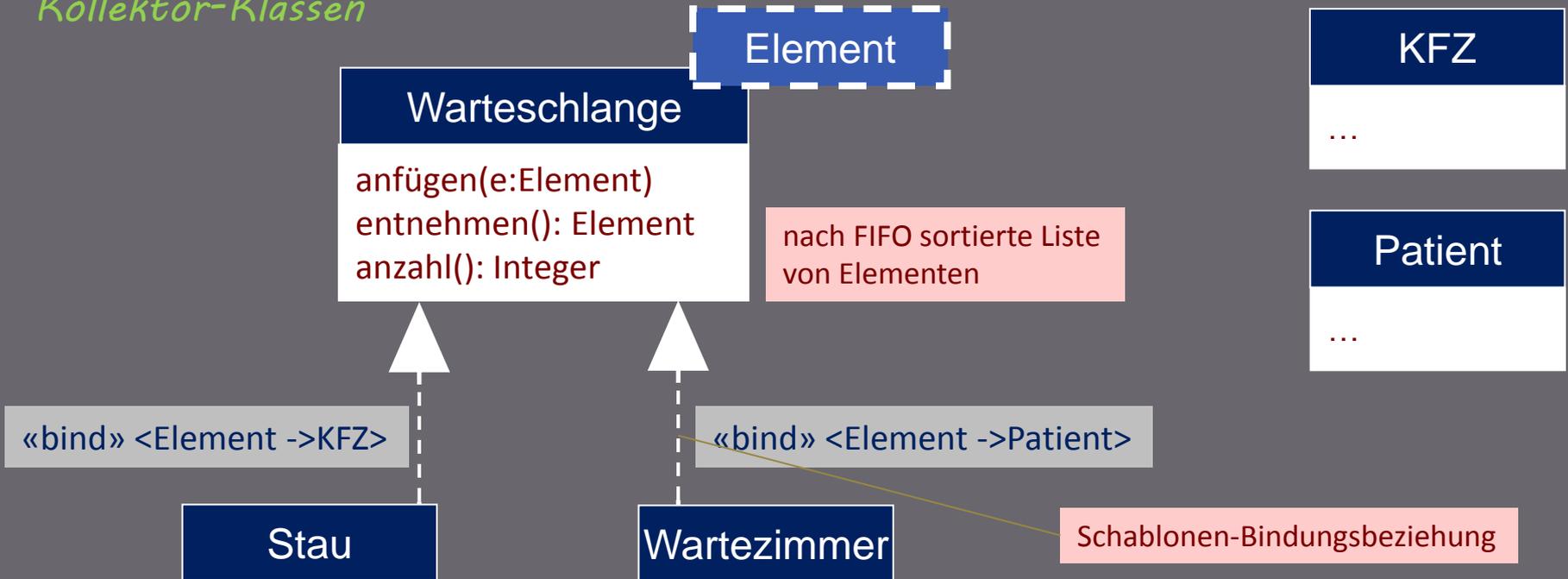
## UML

- Universelle Modellierungssprache mit
  - informal definierter dynamischer Semantik
  - semi-formal definierter statischer Semantik (UML und OCL)→ UML bleibt damit nur gedanklich ausführbar
  
- Dynamische Semantik
  - enthält sog. semantische Variationspunkte (z.T. alternative semantische oder offene semantische Festlegungen)→ Problem für Compiler (Zielcode, Simulatorcode)
  
- Action-Sprache
  - befindet sich in Entwicklung
  - hat Einfluss auf Datentypen in UML (Definition von Literalen, Operatoren offen)
  - bisher DataTypes in UML mit Magie
  
- Pragmatischer Ansatz
  - SDL-ASN.1-UML-Compiler (HU Berlin) benutzt C als Action-Sprache
  - Workflow-Vorlesung benutzt SLX als Actionsprache (notiert in Form von Kommentaren)

# Parametrisierte Klassen / Templates (Wdh.)

typischer Anwendungsfall sind  
Kollektor-Klassen

bereits existierende Klassen



Resultat der Bindung: homogene Listen von KFZ-Objekten bzw. Patient-Objekten

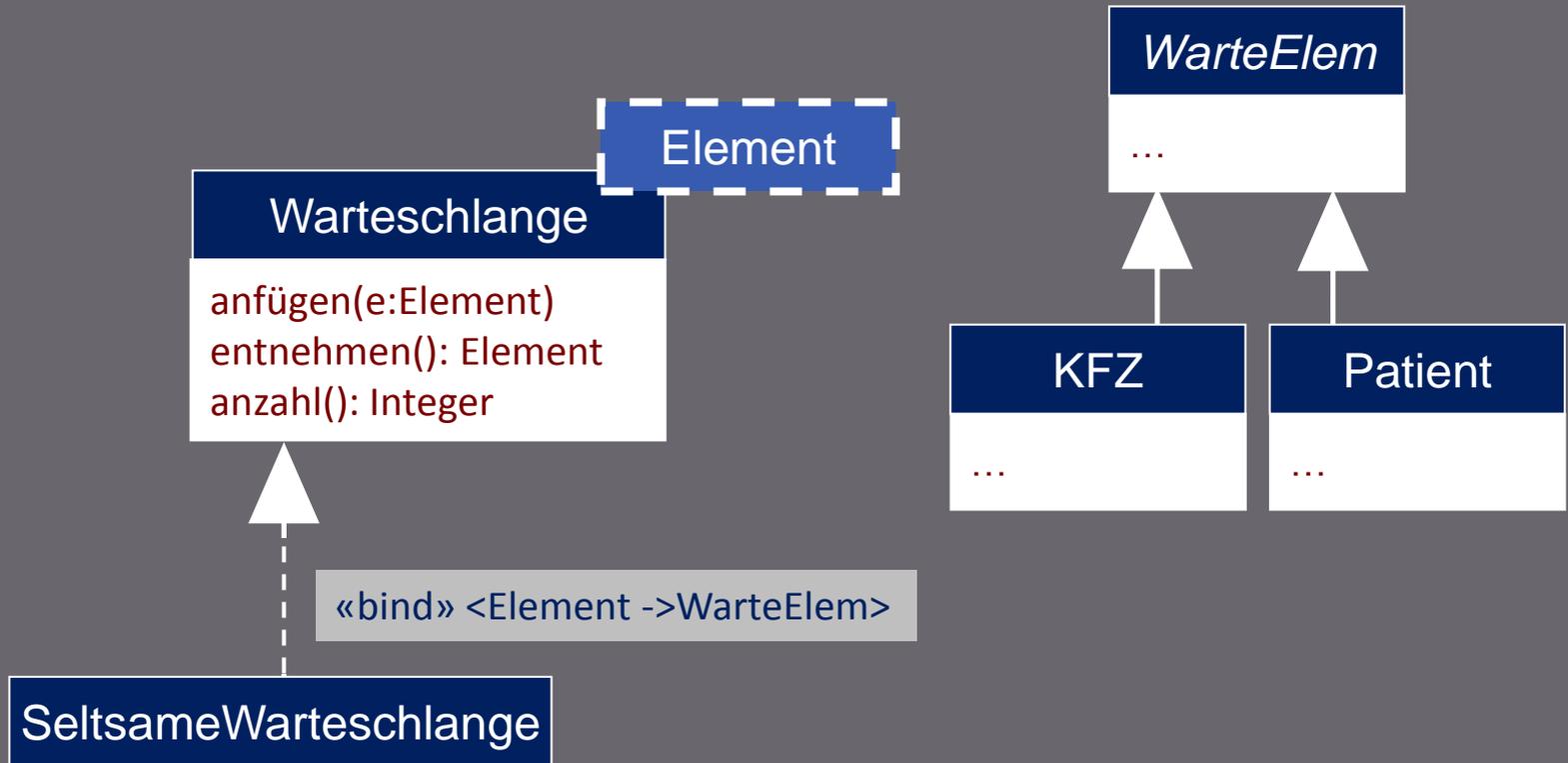
st: Stau

wz: Wartezimmer

**FRAGE:** Wie lässt sich eine heterogene Liste definieren ?

# Parametrisierte Klassen / Templates (Wdh.)

Lösung



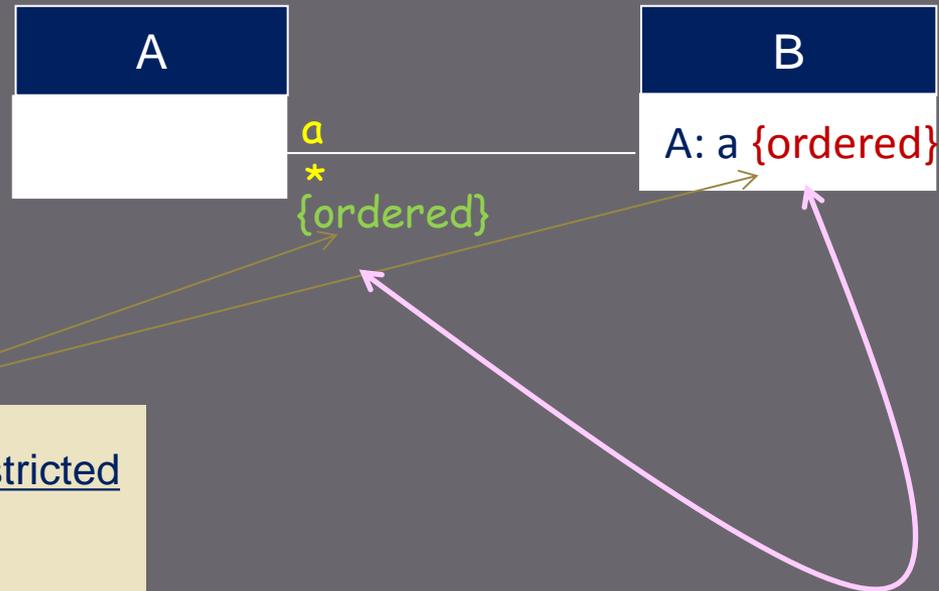
Resultat der Bindung: heterogene Listen von KFZ-Objekten oder/und Patient-Objekten

swz: SeltsameWarteschlange

swz.anfügen (**new** KFZ);  
swz.anfügen (**new** Patient);

**SLX-Aktionen als Kommentar**

# Eigenschaften von Assoziationsenden bzw. Attributen



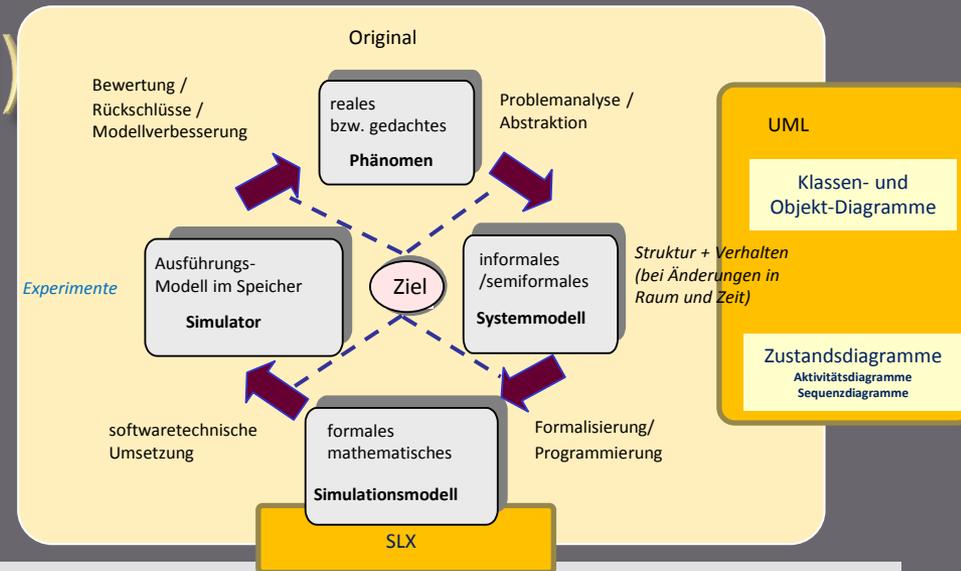
/  
readonly  $\leftarrow \rightarrow$  unrestricted  
composite  
redefines *Attr*  
subsets *Attr*  
union  
unique  $\leftarrow \rightarrow$  nonunique  
ordered  $\leftarrow \rightarrow$  unordered  
seq (Vor.: Multiplizität > 1)  
bag (Vor.: Multiplizität > 1)

entweder oder

UML empfiehlt  
als Assoziationsende

# Zusammenfassung (1)

... formuliert als Fragensammlung



- Charakterisieren Sie die Sprache UML und deren Bedeutung für die Workflow-Modellierung und deren Simulation.
- Wie hängen die UML-Diagrammarten mit dem Viewpoint-Konzept der Modellierung zusammen?
- Welche Modifikationen einer UML-Klasse kennen Sie? Welche Modifikationen von Klassen- und Objektbeziehungen gibt es?
- Welcher Unterschied besteht zwischen einer normalen und einer Assoziationsklasse?
- Erläutern Sie die Dualität der Konzepte Rollenende einer Assoziation und Klassenattribut?
- Zeigen Sie die Wirkungsweise des Template-Konzeptes beispielhaft für die Definition homogener und heterogener Warteschlangen.
- Diskutieren Sie die praktische Bedeutung des Zusammenhangs von aktiver Klasse und dem Zustandsautomatenkonzept von UML.

# Inhalt

- ④ **Teil A**  
Aspekte von Modellierung und Simulation dynamischer Systeme
- ④ **Teil B**  
Die Modellierungssprache UML
- ④ **Teil C**  
Die ausführbare Modellierungssprache SLX
- ④ **Teil D**  
Modellierung von Lieferketten

- ④ **B.1**  
Wozu UML im Kontext der Computersimulation?
- ④ **B.2**  
UML-Teilsprachen, Sprachkonzepte
- ④ **B.3**  
Klassendiagramme
- ④ **B.4**  
Verhaltensbeschreibung
  - ④ Aktive Klasse und Zustandsautomat
  - ④ Automaten zur Laufzeit (außerhalb von UML)
  - ④ Beispiel-1 „Flugreservierung“
  - ④ Trigger und verschachtelte Zustände
  - ④ Beispiel-2 „Heizungssteuerung“
  - ④ Beispiel-3 „Fahrkartenautomat“
  - ④ Beispiel-4 „Maschinenbelegung“
- ④ **B.5**  
OCL

# Zustandsdiagramm

**...zeigt eine Folge von Zuständen eines endlichen Automaten, die ein Objekt (einer aktiven Klasse) im Laufe seines Lebens einnehmen kann**

- endliche Anzahl von Grundzuständen
- ein Startzustand, eventuelle Stopp-Zustände
- Stimuli/Trigger als Auslöser von Zustandsübergängen bei der eventuell Aktionen ausgeführt werden können

Im Kontext einer **aktiven Klasse**

## A) Erweiterung eines Endlichen Zustandsautomaten

- lokale Variablen (Attribute des Objekts)
- lokaler Timer
- Empfangs-Pool von Nachrichten, Remote-Operationsrufen (asynchrone Kommunikation)
- Trigger
- Potentielle Empfangsmenge (Nachrichten/Remote-Op-Rufe)
- Guards (als Wächterbedingungen über lokale/globale Zustandsgrößen)

## B) Zustandserweiterung

- Eintrittsaktionen
- Do-Aktivität
- Austrittsaktionen

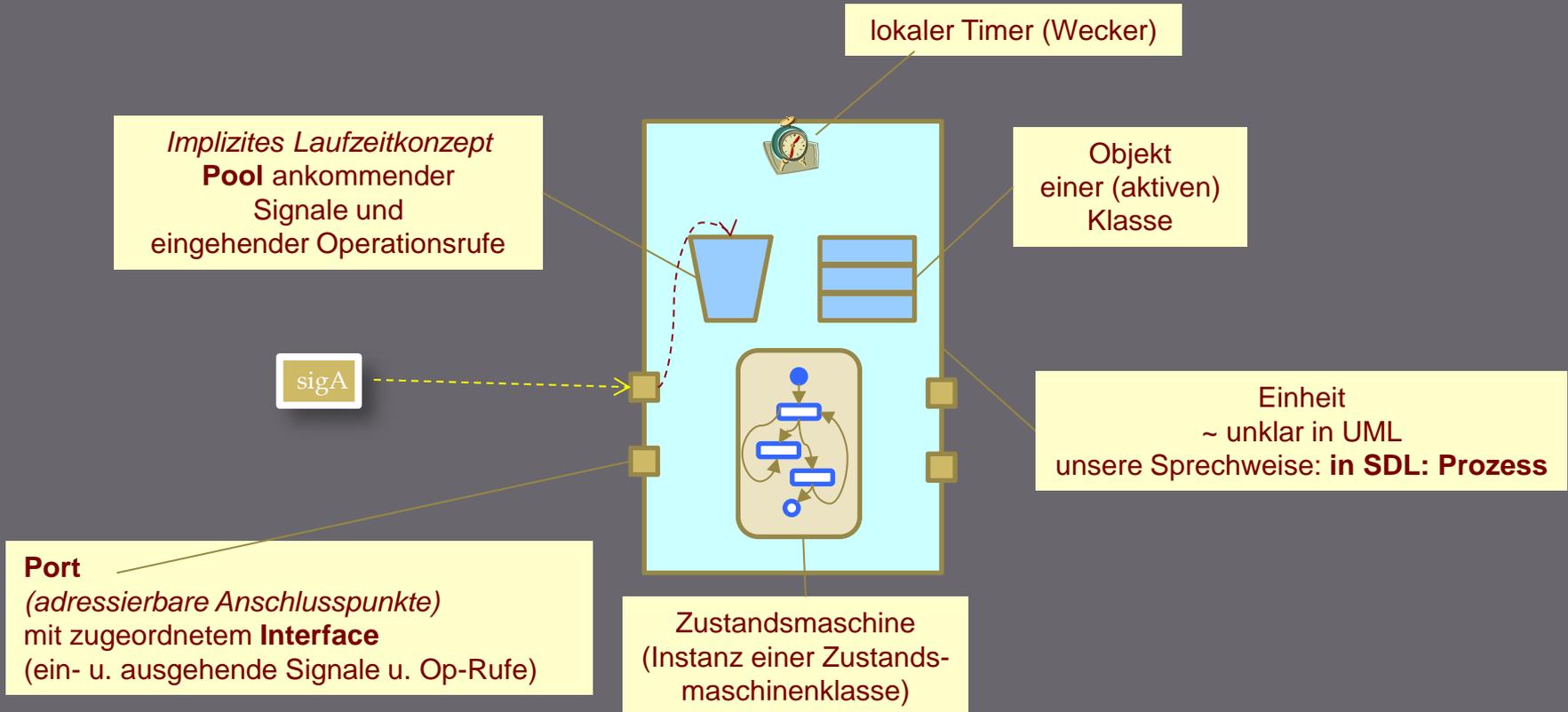
**UML-Automat**

# Inhalt

- ⊙ **Teil A**  
Aspekte von Modellierung und Simulation dynamischer Systeme
- ⊙ **Teil B**  
Die Modellierungssprache UML
- ⊙ **Teil C**  
Die ausführbare Modellierungssprache SLX
- ⊙ **Teil D**  
Modellierung von Lieferketten

- ⊙ **B.1**  
Wozu UML im Kontext der Computersimulation?
- ⊙ **B.2**  
UML-Teilsprachen, Sprachkonzepte
- ⊙ **B.3**  
Klassendiagramme
- ⊙ **B.4**  
Verhaltensbeschreibung mit Zustandsautomaten
  - ⊙ Aktive Klasse und Zustandsautomat
  - ⊙ Automaten zur Laufzeit (außerhalb von UML)
  - ⊙ Beispiel-1 „Flugreservierung“
  - ⊙ Trigger und verschachtelte Zustände
  - ⊙ Beispiel-2 „Heizungssteuerung“
  - ⊙ Beispiel-3 „Fahrkartenautomat“
  - ⊙ Beispiel-4 „Maschinenbelegung“
- ⊙ **B.5**  
OCL

# Automaten zur Laufzeit



## Zustand eines Agenten zu einem Zeitpunkt:

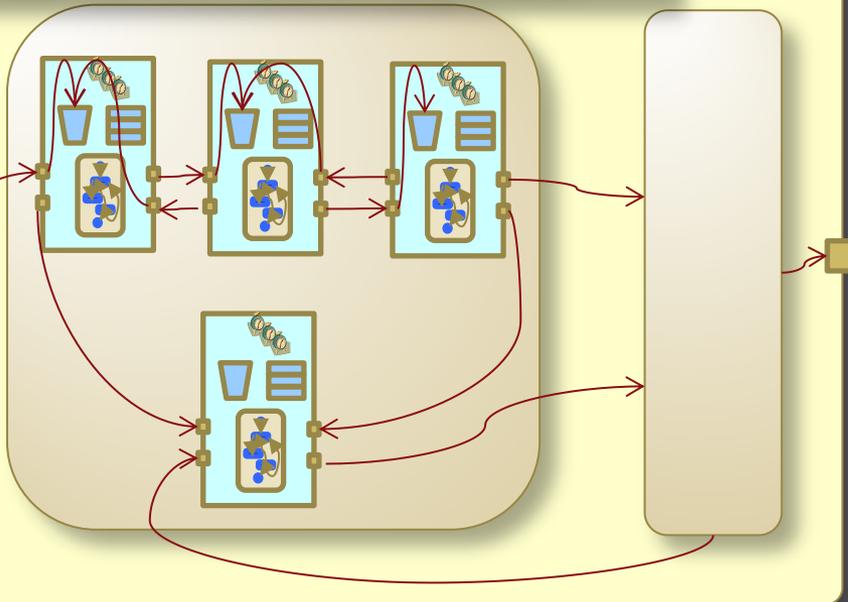
- Stand des gestarteten Timers
- Belegung des Empfang-Pools (inklusive aktueller Parameter der Signale u. Op-Rufe)
- Wertebelegung der Attribute des zugeordneten Objektes
- Aktueller Zustand der Zustandsmaschine
- Befehlsregister (falls im Zustandsübergang)

## Achtung (noch nicht behandelt):

- Spezialisierung (Vererbung) von Zustandsautomaten)
- hierarchische Zustände
- orthogonale Zerlegung (in parallel) agierende Untermaschinen

# System als Agent-Ensemble

- Agenten kommunizieren asynchron



## UML-Probleme

- **Pool-bearbeitung:**  
sog. semantischer Variationspunkt  
(Reihenfolge, ...)
- **Adressierung** von Signalen  
(nicht komplett gelöst,  
Bekanntmachung der Agenten/Ports)
- **Übertragung** von Signalen  
(ungelöst: Zeitverbrauch, Sicherheit)
- semantischer Variationspunkt  
Offenheit bzgl. **Action-Sprache**  
Datentypen, Anweisungen

Es gibt kein UML-Werkzeug, das das Agent-Konzept allgemein unterstützt.

**ABER:** es gibt praktikable Teillösungen

**SDL (Specification and Description Language)**

# Inhalt

- ⊙ **Teil A**  
Aspekte von Modellierung und Simulation dynamischer Systeme

- ⊙ **Teil B**  
Die Modellierungssprache UML

- ⊙ **Teil C**  
Die ausführbare Modellierungssprache SLX

- ⊙ **Teil D**  
Modellierung von Lieferketten

- ⊙ **B.1**  
Wozu UML im Kontext der Computersimulation?
- ⊙ **B.2**  
UML-Teilsprachen, Sprachkonzepte
- ⊙ **B.3**  
Klassendiagramme
- ⊙ **B.4**  
Verhaltensbeschreibung mit Zustandsautomaten
  - ⊙ Aktive Klasse und Zustandsautomat
  - ⊙ Automaten zur Laufzeit (außerhalb von UML)
  - ⊙ **Beispiel-1 „Flugreservierung“**
  - ⊙ Trigger und verschachtelte Zustände
  - ⊙ Beispiel-2 „Heizungssteuerung“
  - ⊙ Beispiel-3 „Fahrkartenautomat“
  - ⊙ Beispiel-4 „Maschinenbelegung“
- ⊙ **B.5**  
OCL

# Beispiel-1: Flugreservierung

## Aktive Klasse

- Attribute
- Operationen
- Signale (potentieller Empfang)  
.....
- Methoden als Implementierungen der Operationen (z.B. zugeordnete Aktivitätsdiagramme)
- zugeordnete Zustandsmaschine (eigenständiger Lebenslauf)

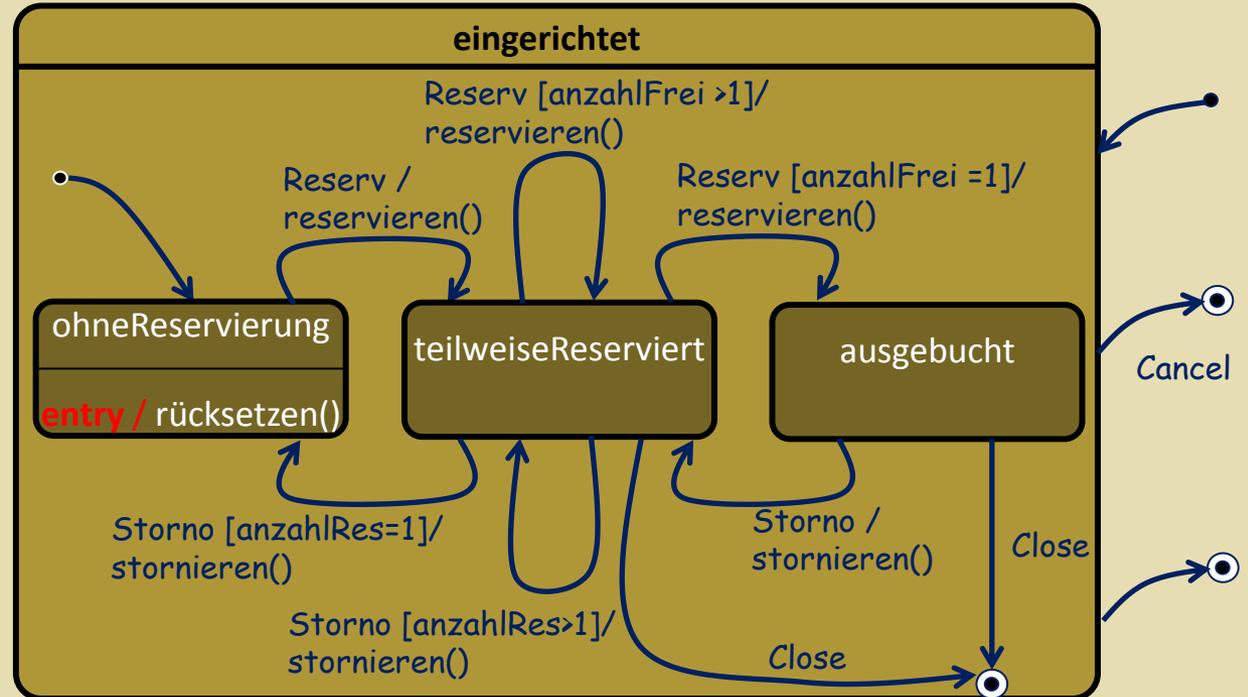
## Objekt

ist mit einer Instanz der Zustandsmaschine verbunden, die über das Objekt operiert

Zustandsmaschine

- bearbeitet ankommende Signale u. (Remote-) Operationsrufe
- führt lokale Operationsrufe durch
- manipuliert Objekt-Attribute

## Flugreservierung Verhalten





# Inhalt

- ④ **Teil A**  
Aspekte von Modellierung und Simulation dynamischer Systeme

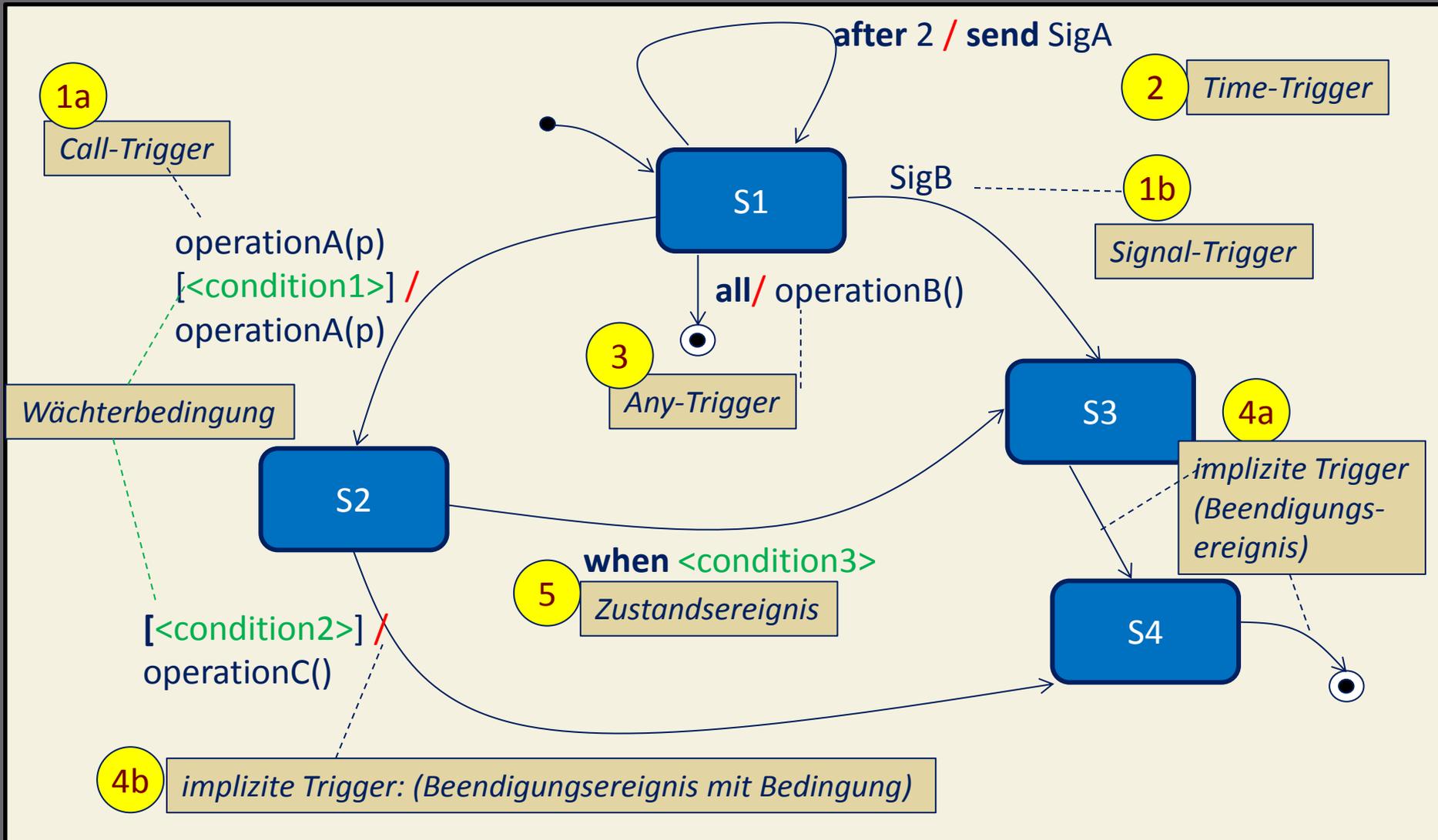
- ④ **Teil B**  
Die Modellierungssprache UML

- ④ **Teil C**  
Die ausführbare Modellierungssprache SLX

- ④ **Teil D**  
Modellierung von Lieferketten

- ④ **B.1**  
Wozu UML im Kontext der Computersimulation?
- ④ **B.2**  
UML-Teilsprachen, Sprachkonzepte
- ④ **B.3**  
Klassendiagramme
- ④ **B.4**  
Verhaltensbeschreibung mit Zustandsautomaten
  - ④ Aktive Klasse und Zustandsautomat
  - ④ Automaten zur Laufzeit (außerhalb von UML)
  - ④ Beispiel-1 „Flugreservierung“
  - ④ Trigger und verschachtelte Zustände
  - ④ Beispiel-2 „Heizungssteuerung“
  - ④ Beispiel-3 „Fahrkartenautomat“
  - ④ Beispiel-4 „Maschinenbelegung“
- ④ **B.5**  
OCL

# Verschiedene Trigger-Arten



# Zustände und Interne Aktivitäten

## Compartments eines Zustands

- Namensabteil

*spezifiziert den Namen des Zustandes*

- Interne Aktivitäten

*Aktivitäten, die ausgeführt werden, wenn sich eine Instanz des beschriebenen Classifiers in einem Zustand befindet*

Ausführung erfolgt **ohne** Zustandswechsel

### ReadCardNumber

**entry** / ShowReadCardDialog

**do** / ReadCardNumbers

**defer** <SignalName-(Liste)>

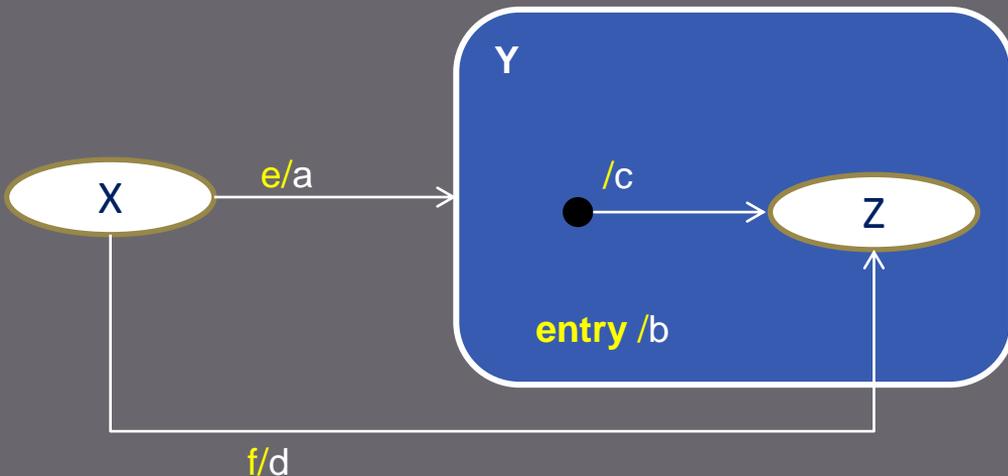
**exit** / HideReadCardDialog

- **entry** – Ausführung bei Betreten des Zustandes, vor allem anderen
- **exit** – Ausführung bei Verlassen des Zustandes, nach allem anderen
- **do** – Ausführung bis entweder fertig (implizites Endeereignis) oder Zustand per expliziten Trigger verlassen wird
- weitere nutzerspezifische interne Aktivitäten  
z.B. **defer** Rückstellung von Signalen in der Betrachtung für den aktuellen Zustand

# Besonderheiten interner Zustandsaktivitäten

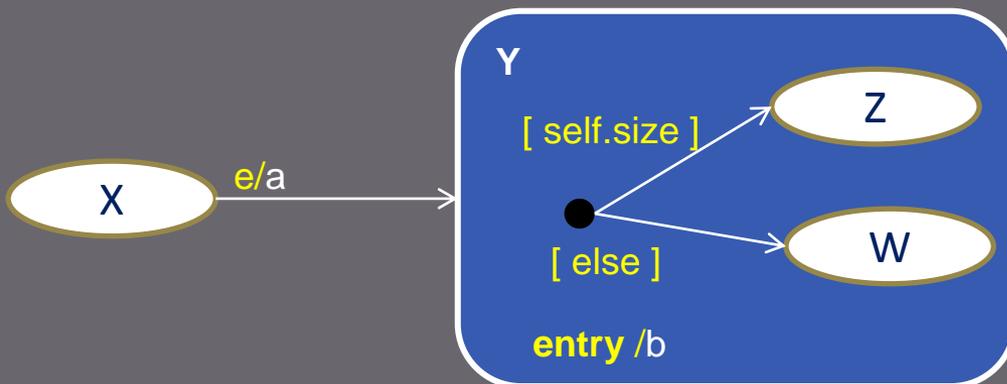
- entry- und exit-Aktivitäten dürfen
  - keine Parameter besitzen
  - Keine Randbedingungen besitzen
- interne Transitionen können
  - über Ereignisse mit Parametern und Wächterbedingungen verfügen
  - Unterschied zu einer entsprechenden externen Transition:  
keine Ausführung der entry- und exit-Aktivitäten
- befindet sich ein Objekt in einem Zustand,
  - dann befindet sich das Objekt i.allg. im „Leerlauf“  
(wartet auf Ereignis)
  - es kann aber auch eine (Dauer-)Aktivität laufen  
Bei Beendigung wird ein implizites Ereignis erzeugt, dass das Verlassen des Zustandes überprüft und evtl. realisiert

# Zustandsübergänge, Aktionsreihenfolge



## Verharren in X

- (1) bei Eintritt von **e** feuert die Transition
  - Aktion **a** wird ausgeführt
  - Übergang nach **Y**
- Eintrittsaktion **b** wird ausgeführt  
Startzustand von **Y** wird aktiv
- Übergang nach **Z** wird unmittelbar ausgeführt
  - Ausführung der Aktion **c**
  - Übergang in **Z** (bleiben auch **Y**)
- (2) Bei Eintritt von **f** feuert die andere Transition bei Ausführung von **d** und **b** (ohne c)



## Verharren in X

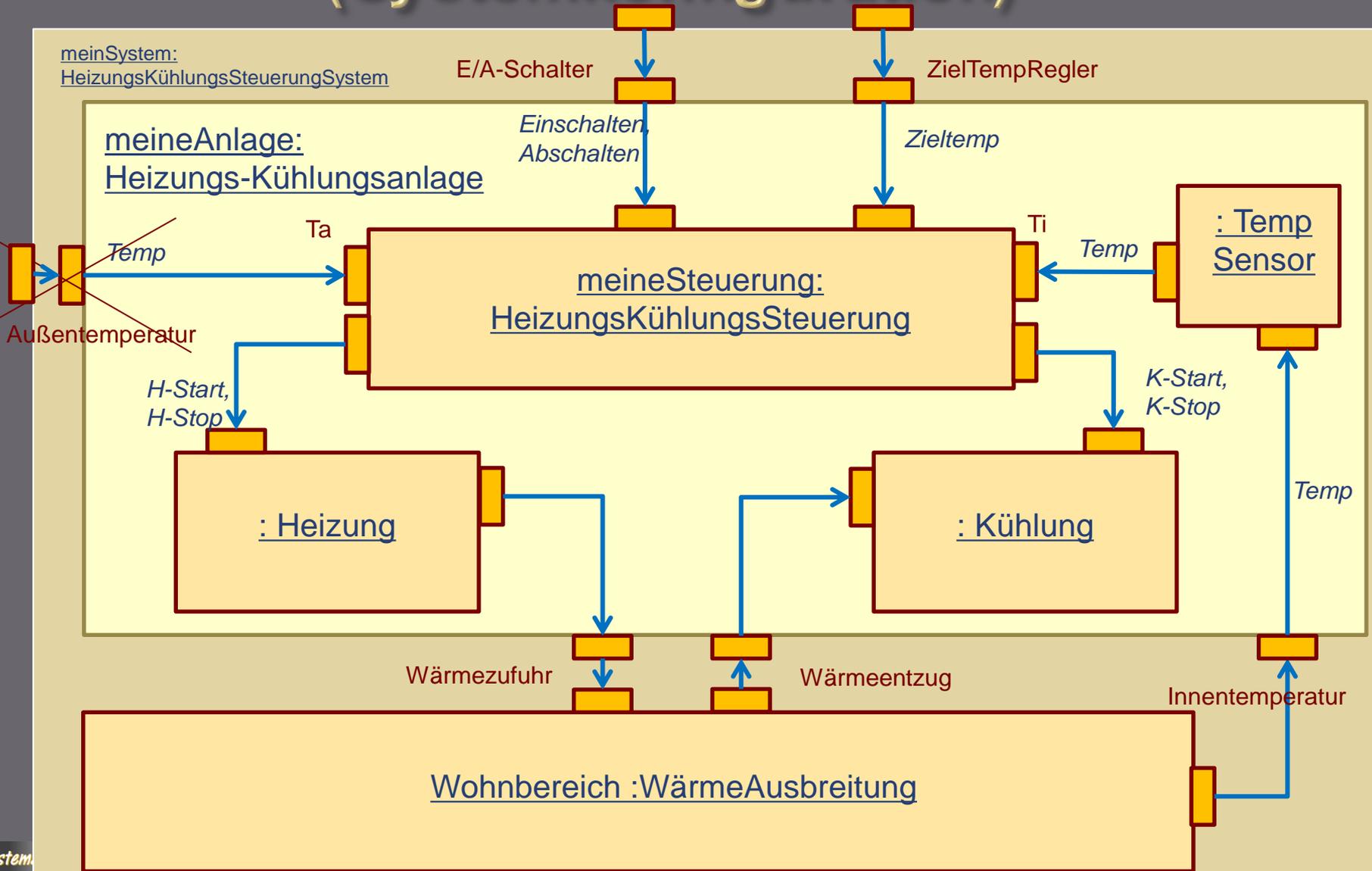
- bei Eintritt von **e** feuert die Transition
- Aktion **a** wird ausgeführt
  - Übergang nach **Y**
- Eintrittsaktion **b** wird ausgeführt  
Startzustand von **Y** wird aktiv
- Übergang nach **Z** oder **W** in Abh. Vom Wert von **size** als Attribut des zugehörigen Objekts

# Inhalt

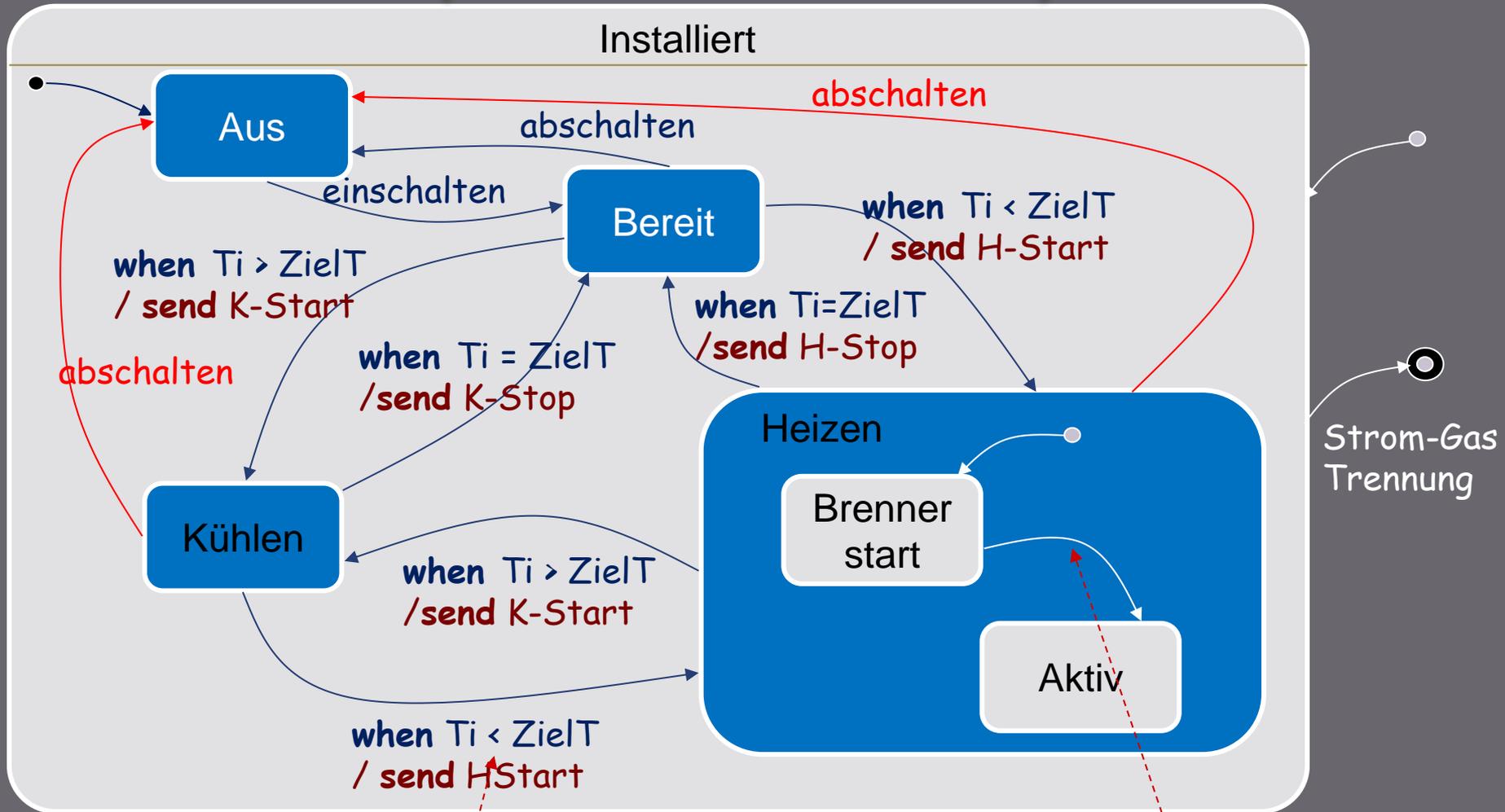
- ④ **Teil A**  
Aspekte von Modellierung und Simulation dynamischer Systeme
- ④ **Teil B**  
Die Modellierungssprache UML
- ④ **Teil C**  
Die ausführbare Modellierungssprache SLX
- ④ **Teil D**  
Modellierung von Lieferketten

- ④ **B.1**  
Wozu UML im Kontext der Computersimulation?
- ④ **B.2**  
UML-Teilsprachen, Sprachkonzepte
- ④ **B.3**  
Klassendiagramme
- ④ **B.4**  
Verhaltensbeschreibung mit Zustandsautomaten
  - ④ Aktive Klasse und Zustandsautomat
  - ④ Automaten zur Laufzeit (außerhalb von UML)
  - ④ Beispiel-1 „Flugreservierung“
  - ④ Trigger und verschachtelte Zustände
  - ④ Beispiel-2 „Heizungssteuerung“
  - ④ Beispiel-3 „Fahrkartenautomat“
  - ④ Beispiel-4 „Maschinenbelegung“
- ④ **B.5**  
OCL

# Beispiel-2: Heizungs-Kühlungsanlage (Systemkonfiguration)



# Beispiel-2: HeizungsKühlungsSteuerung (Zustandsautomat)



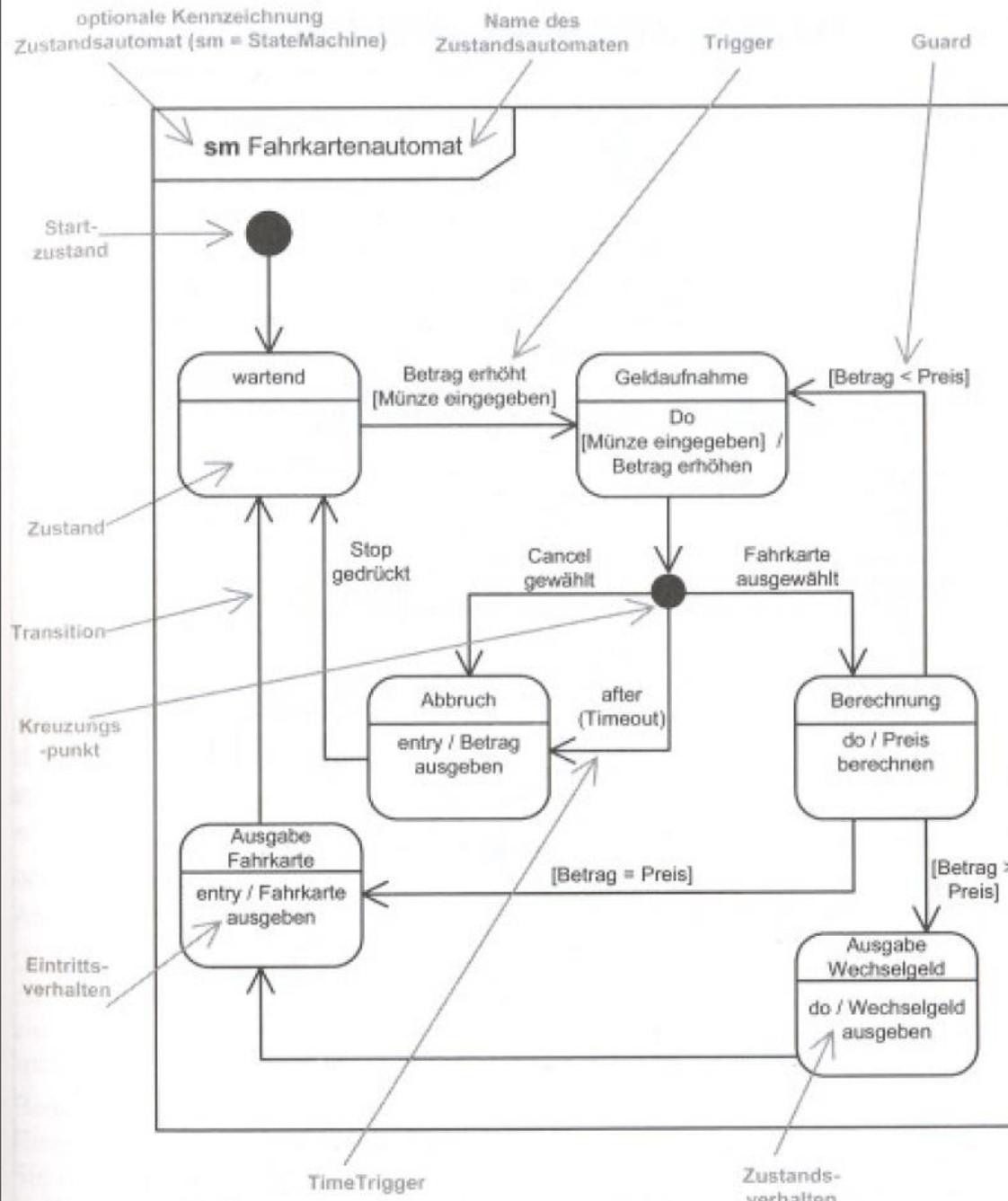
- Ereignisse lösen Transitionen aus, wenn dafür Trigger vorhanden sind
- Trigger können **explizit** angegeben sein, oder sie ergeben sich **implizit**

# Inhalt

- ④ **Teil A**  
Aspekte von Modellierung und Simulation dynamischer Systeme
- ④ **Teil B**  
Die Modellierungssprache UML
- ④ **Teil C**  
Die ausführbare Modellierungssprache SLX
- ④ **Teil D**  
Modellierung von Lieferketten

- ④ **B.1**  
Wozu UML im Kontext der Computersimulation?
- ④ **B.2**  
UML-Teilsprachen, Sprachkonzepte
- ④ **B.3**  
Klassendiagramme
- ④ **B.4**  
Verhaltensbeschreibung mit Zustandsautomaten
  - ④ Aktive Klasse und Zustandsautomat
  - ④ Automaten zur Laufzeit (außerhalb von UML)
  - ④ Beispiel-1 „Flugreservierung“
  - ④ Trigger und verschachtelte Zustände
  - ④ Beispiel-2 „Heizungssteuerung“
  - ④ Beispiel-3 „Fahrkartenautomat“
  - ④ Beispiel-4 „Maschinenbelegung“
- ④ **B.5**  
OCL

# Beispiel-3



- Zustand
- Trigger
- Time-Trigger
- Guard
- Entscheidung
- Eintrittsverhalten
- Zustandsverhalten
- Austrittsverhalten

Quelle: „UML-Glasklar“

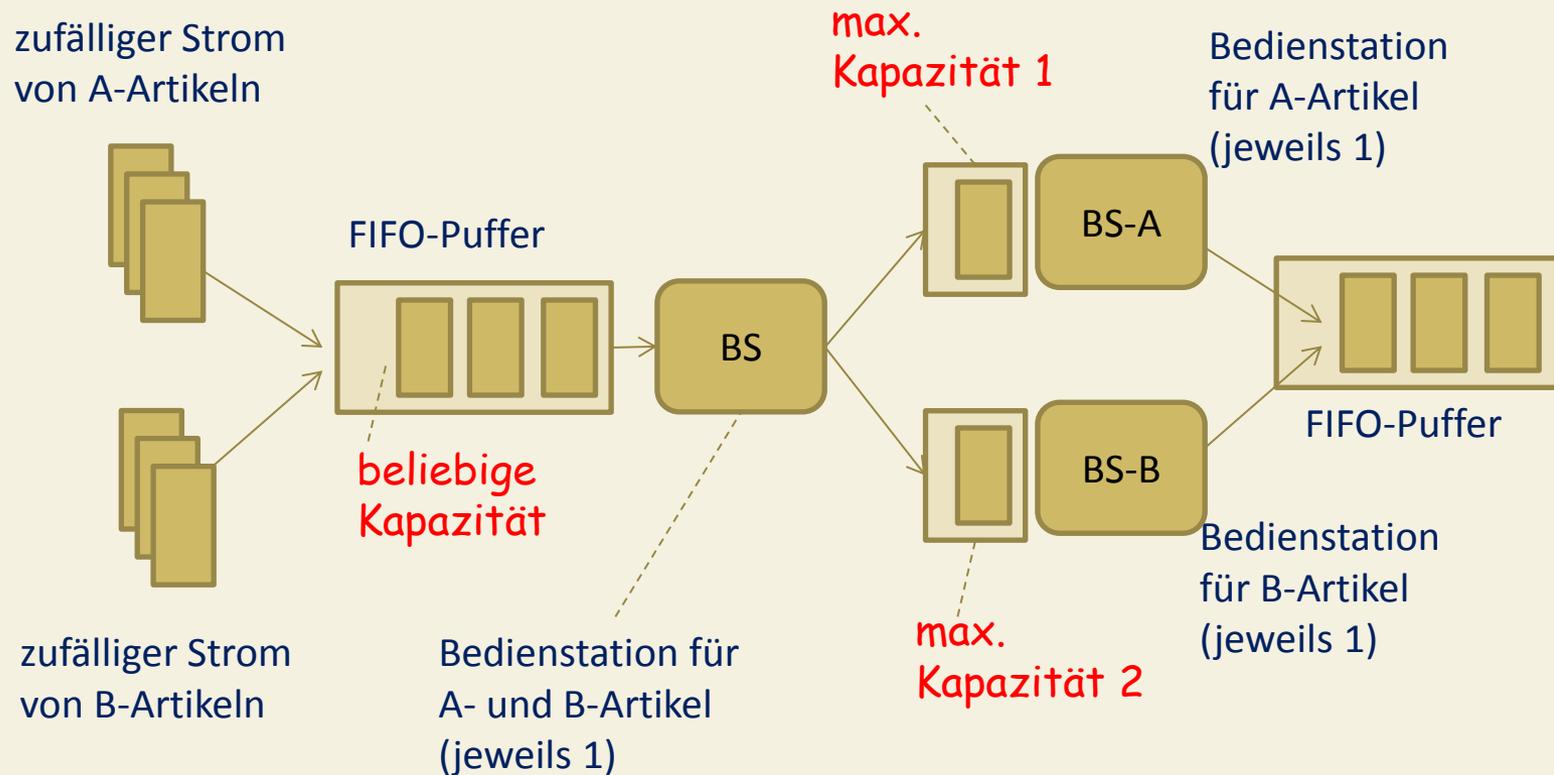
# Inhalt

- ④ **Teil A**  
Aspekte von Modellierung und Simulation dynamischer Systeme
- ④ **Teil B**  
Die Modellierungssprache UML
- ④ **Teil C**  
Die ausführbare Modellierungssprache SLX
- ④ **Teil D**  
Modellierung von Lieferketten

- ④ **B.1**  
Wozu UML im Kontext der Computersimulation?
- ④ **B.2**  
UML-Teilsprachen, Sprachkonzepte
- ④ **B.3**  
Klassendiagramme
- ④ **B.4**  
Verhaltensbeschreibung mit Zustandsautomaten
  - ④ Aktive Klasse und Zustandsautomat
  - ④ Automaten zur Laufzeit (außerhalb von UML)
  - ④ Beispiel-1 „Flugreservierung“
  - ④ Trigger und verschachtelte Zustände
  - ④ Beispiel-2 „Heizungssteuerung“
  - ④ Beispiel-3 „Fahrkartenautomat“
  - ④ Beispiel-4 „Maschinenbelegung“
- ④ **B.5**  
OCL

# Beispiel-4: Maschinenbelegung

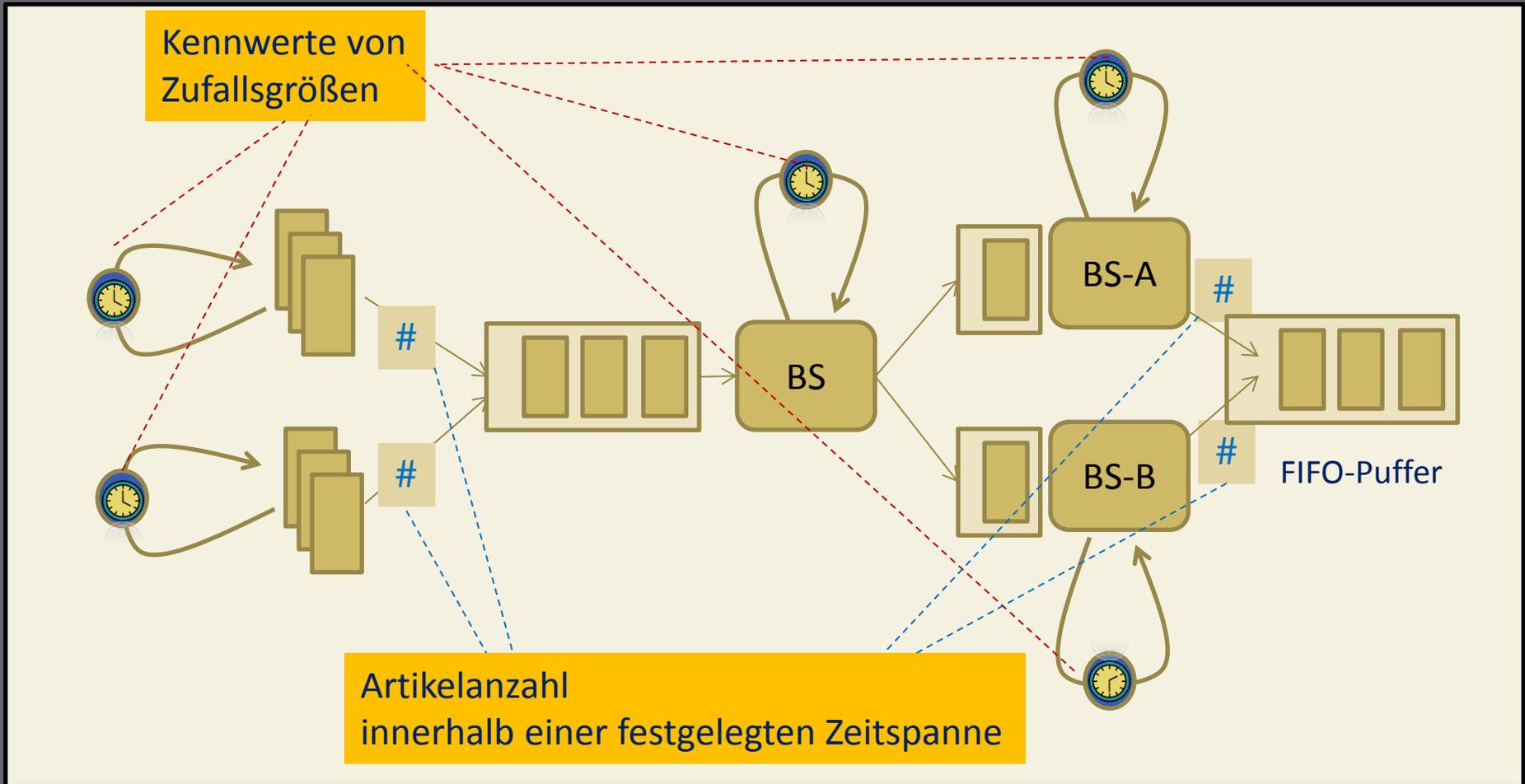
Informales Modell ~ Ziel: Durchsatzermittlung i-Abh. stochastischer Parameter



Bearbeitungsstation mit zufällig schwankenden Fertigungszeiten

# Beispiel-4: Maschinenbelegung

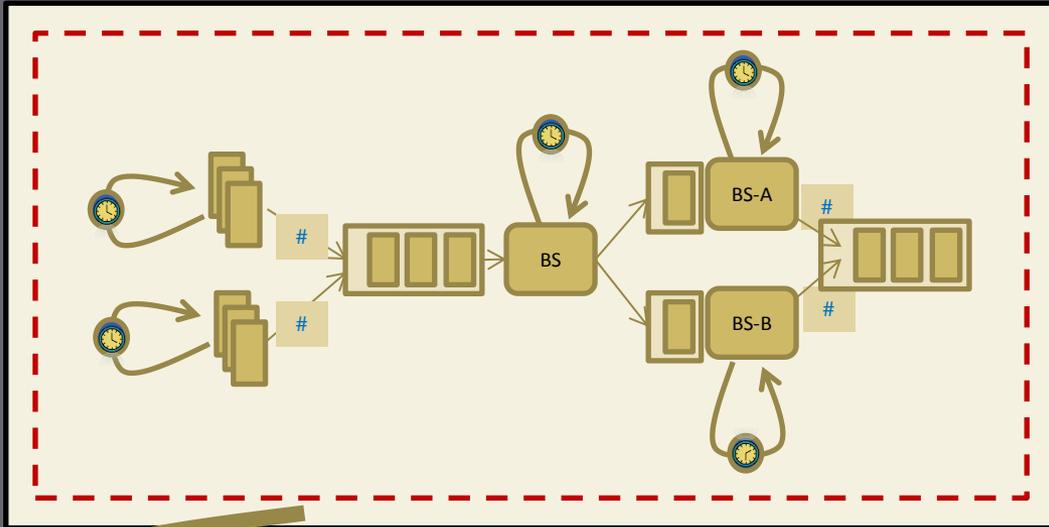
*ExperimentierParameter*



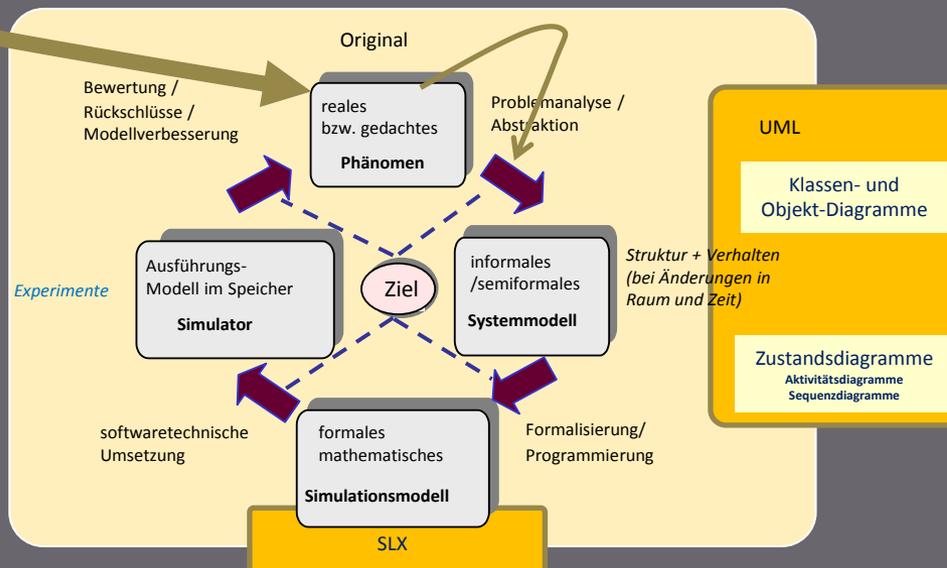
*Beobachtungsgrößen*

# Beispiel-4: Maschinenbelegung

Präzisierung des Scientific Workflows



Phänomen  
als  
hypothetisches Original-System  
bei  
formulierter  
Zielstellung

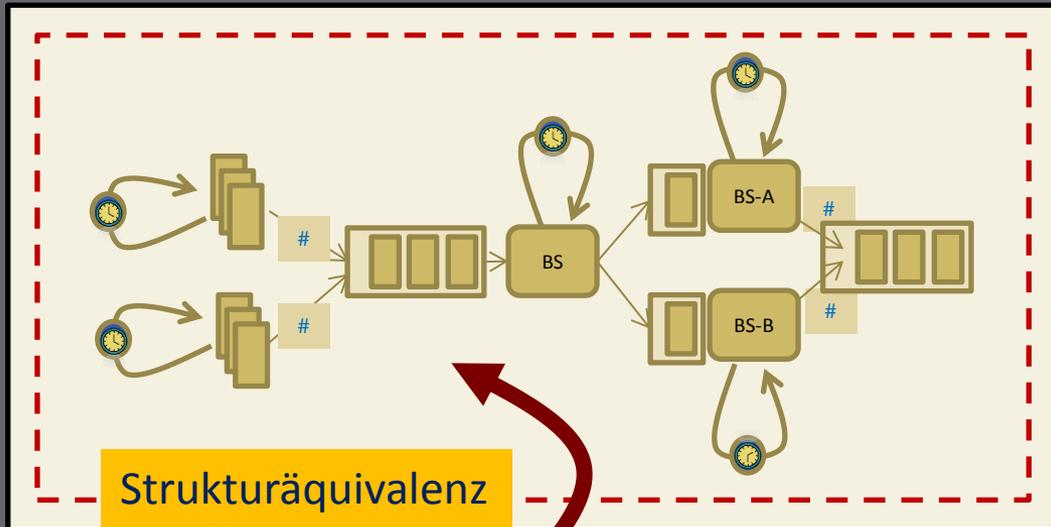


**Annahme:**  
verfügen über Paket mit

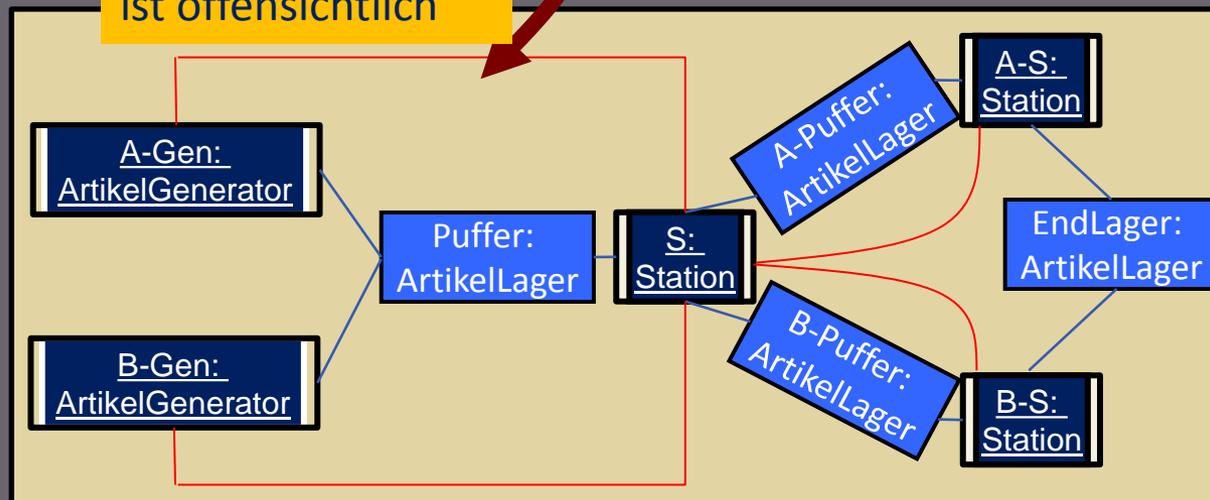
- generischer Factory (Template für aktive Klasse)
- Generischer FIFO-Warteschlange (Template für passive Klasse)

# Beispiel-4: Maschinenbelegung

*UML-Systemkonfiguration als Objektdiagramm bei Festlegung von aktiver und passiver Klasseigenschaft*



Artikel ~ passive Klasse  
 Puffer ~ passive Klasse  
 Factory ~ aktive Klasse  
 Bedienstation ~ aktive Klasse



Link der Assoziation  
**AuftragsZustellung:**  
 Automat → Automat  
 (~ per Signalversand)

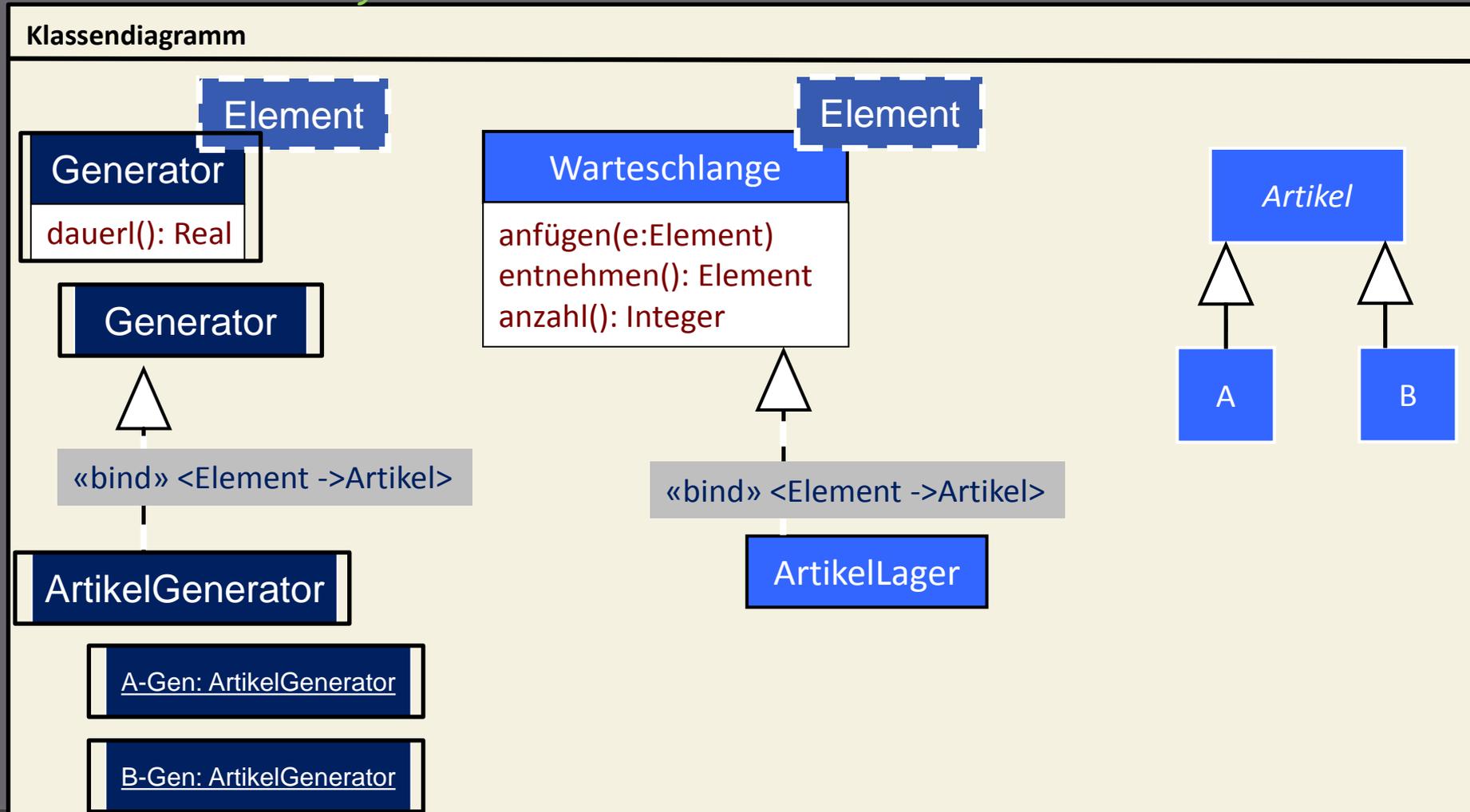
Link der Assoziation  
**ArtikelZustellung:**  
 Automat → ArtikelLager  
 (~ per Artikelversand)

Jeder UML-Zustandsautomat hat einen lokalen Timer (Uhr)

# Beispiel-4: Maschinenbelegung

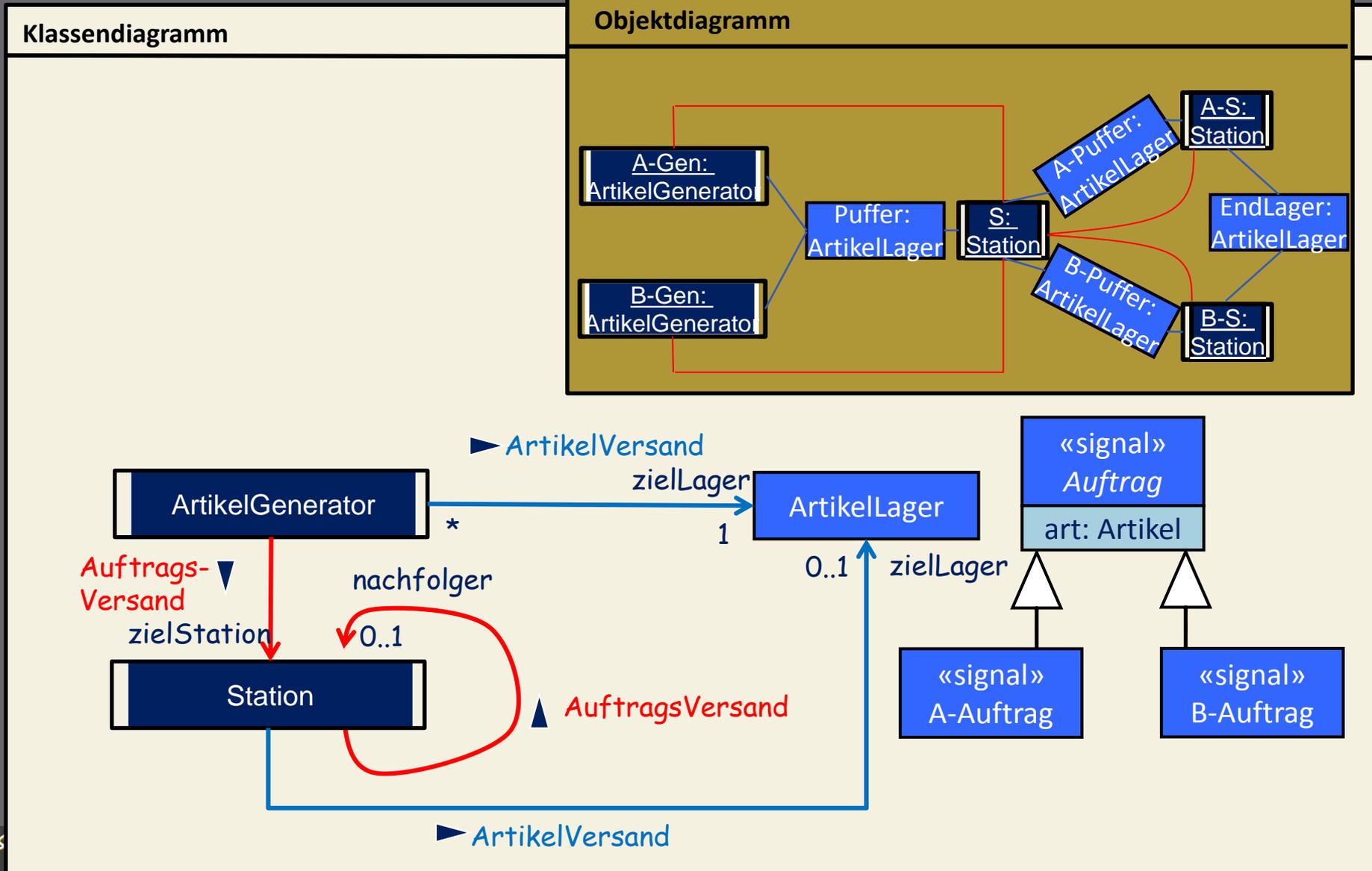
Anwendung von Typschablone zur Definition von

- ArtikelGenerator
- ArtikelLager



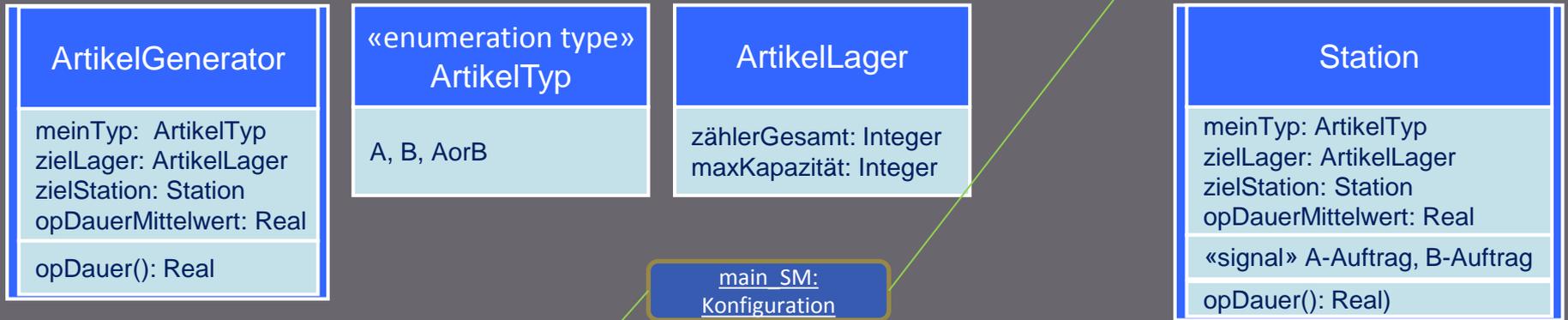
# Beispiel-4: Maschinenbelegung

Klassendiagramm: Assoziationen zur Definition der Links im Objektdiagramm



# Beispiel-4: Maschinenbelegung

Erzeugung des UML-Systems: Unklar, wer die Objektconstructoren ruft



## Initiale Objektkonfiguration

