Kurs OMSI
im WiSe 2014/15

Objektorientierte Simulation mit ODEMx

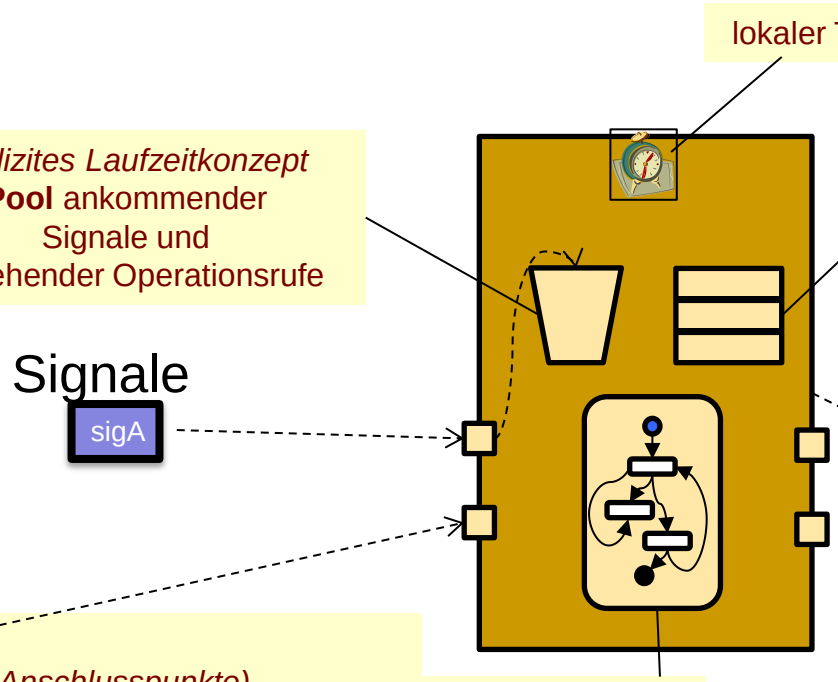
Prof. Dr. Joachim Fischer
Dr. Klaus Ahrens
Dr. Markus Scheidgen
Dipl.-Inf. Ingmar Eveslage

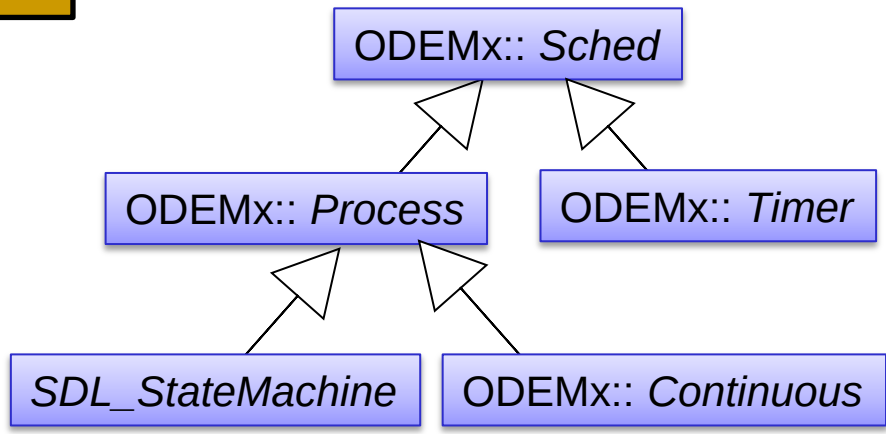
fischer|ahrens|eveslage@informatik.hu-berlin.de

Letzte Vorlesung

1. Systemsimulation – was ist das?
2. Ein Blick zurück in die Anfänge
3. Modelle und Originale
4. Modellierungssprachen, Simulationsumgebungen
5. Bespiele aus der aktuellen Forschung
6. Paradigma der objektorientierten Modellierung
7. Einordnung von UML
8. Klassifikation dynamischer Systeme
9. M&S eines Niedertemperaturofens

Automaten zur Laufzeit





Zustand eines SDL-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

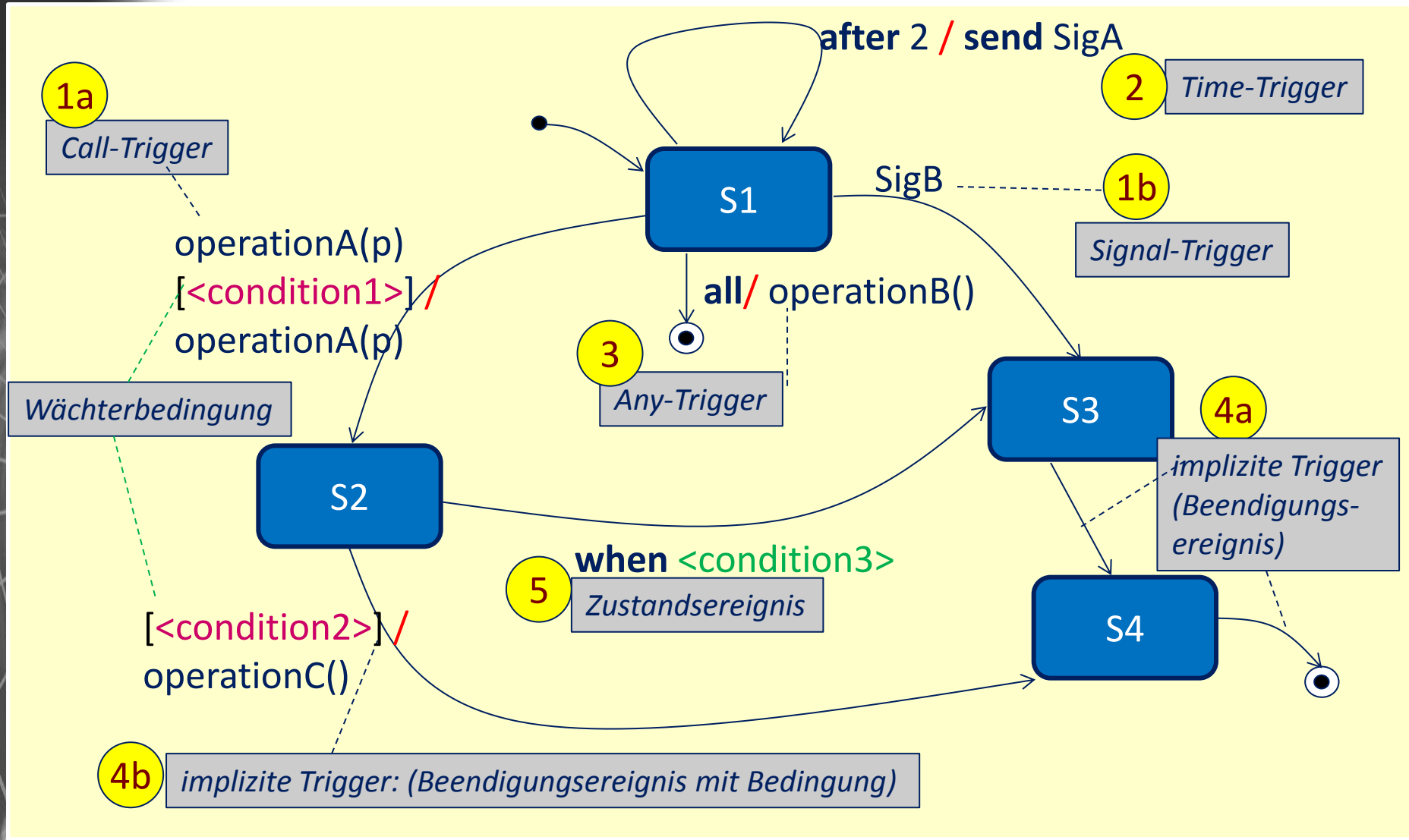
Allgemeine Arbeitsweise (UML-Automat)

- Automat **verharrt** in seinem Grundzustand bis
  - ein **Signal** oder **RemoteCall** eintrifft (im Pool)
    → Behandlung für den Zustand definiert?
    dann: Behandlung bei Signalkonsumtion
    bzw. Retten (**defer**)
    sonst: Verwerfen

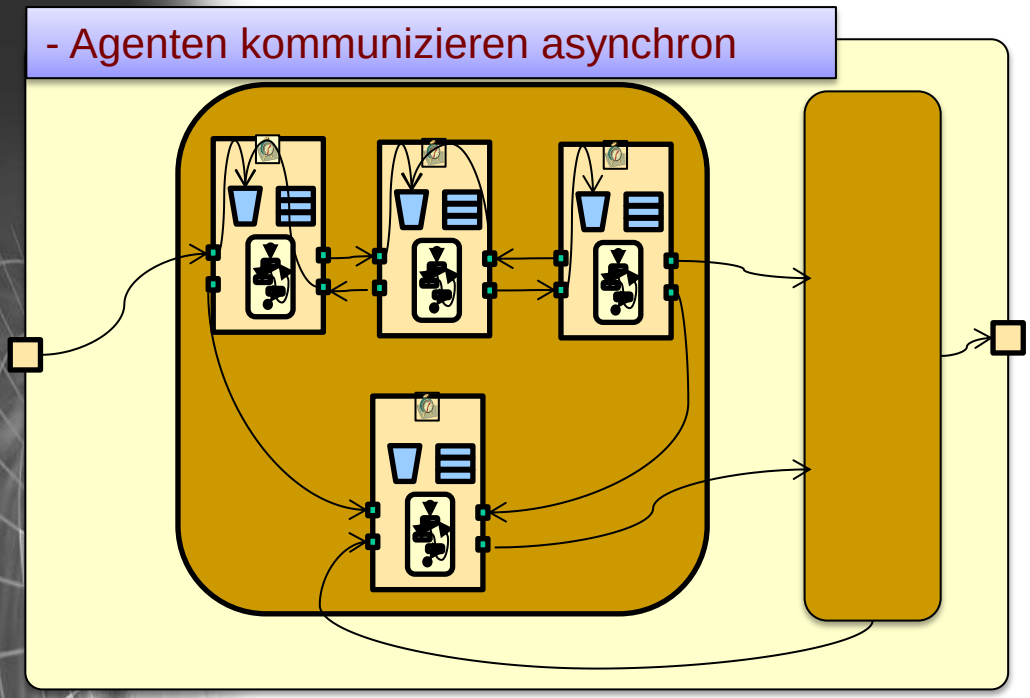
    zusätzl.
    optionale Wächterbedingung
  - der evtl. gestartete **Timer** läuft ab
    dann: vorgesehene Behandlung
  - die evtl. gestartete **Do-Aktivität** wird beendet
    → Auslösung eines impliziten Finalisierungs-Ereignisses
    dann: vorgesehene Behandlung
    (Transition ohne TriggerAngabe)

    zusätzl.
    optionale Wächterbedingung
  - für den Zustand ist ein **Zustandsereignis** definiert (**when** …)
    und dieses tritt ein
    → Auslösung eines impliziten Finalisierungs-Ereignisses
    dann: vorgesehene Behandlung
- und führt dann i.allg einen **Zustandsübergang** durch (nicht bei **defer**)

Verschiedene Trigger-Arten (Überblick)



System als Agent-Ensemble

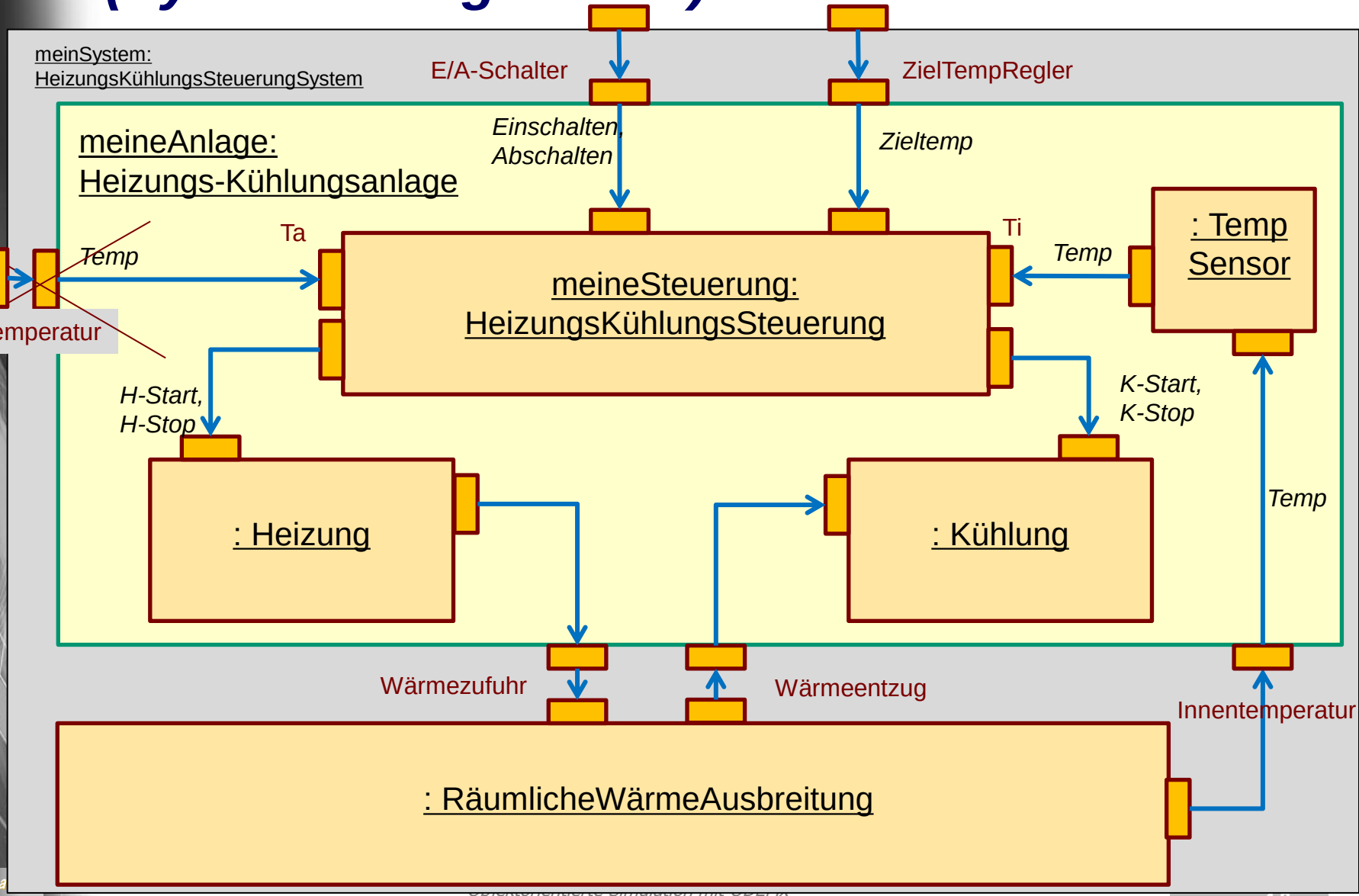


UML-Probleme

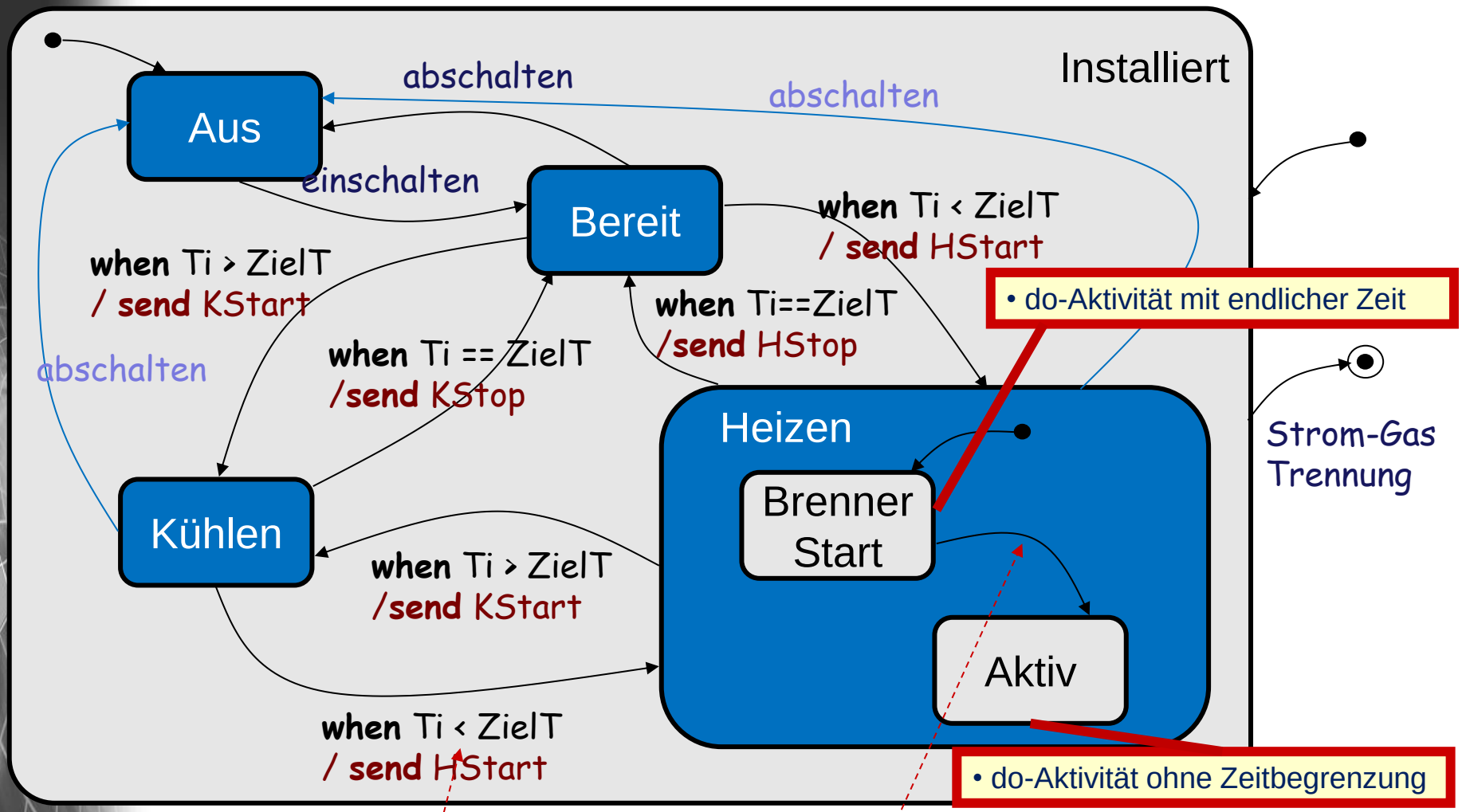
- Pool-bearbeitung:
  sog. semantischer Variationspunkt
  (Reihenfolge, …)
- Adressierung von Signalen
  (nicht komplett gelöst,
  Bekanntmachung der Agenten/Ports)
- Übertragung von Signalen
  (ungelöst: Zeitverbrauch, Sicherheit)
- semantischer Variationspunkt
  Offenheit bzgl. Action-Sprache
  Datentypen,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

Es gibt kein UML-Werkzeug, welches das Agent-Konzept allgemein unterstützt.
ABER: es gibt praktikable Teillösungen
SDL (Specification and Description Language)

Installierte Heizungs-Kühlungsanlage (Systemkonfiguration)



HeizungsKühlungsSteuerung (Zustandsautomat)



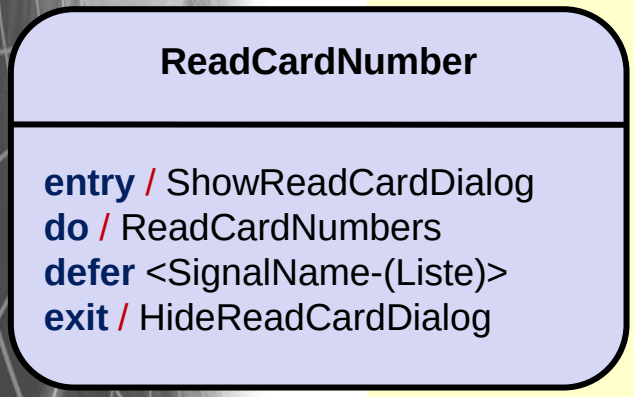
- Ereignisse lösen Transitionen aus, wenn dafür Trigger vorhanden sind
- Trigger können explizit angegeben sein, oder sie ergeben sich implizit
  Annahme:BrennerStart und Aktiv sind mit einer do-Aktivität ausgestattet

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

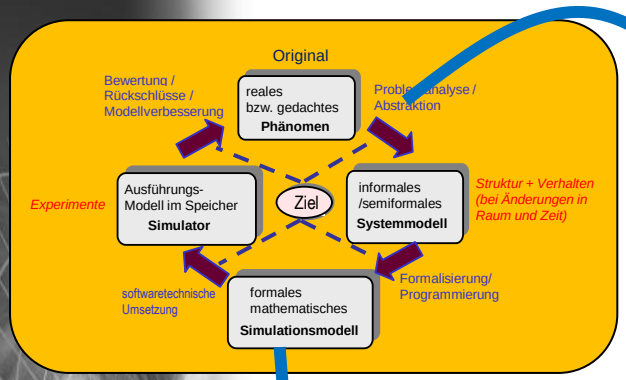
  Ausführung erfolgt **ohne** Zustandswechsel
  - entry – Ausführung bei Betreten des Zustandes, vor allem anderen
  - exit – Ausführung bei Verlassen des Zustandes, nach allem anderen
  - do – Ausführung bis entweder fertig (implizites Endeereignis) oder Zustand per expliziten Trigger verlassen wird
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



1. Einführung

1. Systemsimulation – was ist das?
2. Ein Blick zurück in die Anfänge
3. Modelle und Originale
4. Modellierungssprachen, Simulationsumgebungen
5. Bespiele aus der aktuellen Forschung
6. Paradigma der objektorientierten Modellierung
7. Einordnung von UML
8. Klassifikation dynamischer Systeme
9. M&S eines Niedertemperaturofens

Verhaltens- und Zustandsgrößen



Modellierungsaspekte realer oder gedachter Phänomene

- Existenz/Substanz (Ausdehnung in Raum und Zeit) repräsentiert durch statische und dynamische Objekt-Strukturen
- Verhaltensgrößen (messbare Eigenschaften) repräsentiert durch Werte der Objektattribute
- Veränderung der Substanz (dynamisches Verhalten) repräsentiert durch interagierende Objekte aktiver Klassen in Abhängigkeit einer Modellzeit bei Nutzung von Objekten passiven Klassen

essentielle Verhaltensgrößen

- nicht immer beobachtbar
- **Zustandsgrößen** als ausgezeichnete Verhaltensgrößen (spielen eine zentrale Rolle bei der Modellierung)

Zustandsgrößen

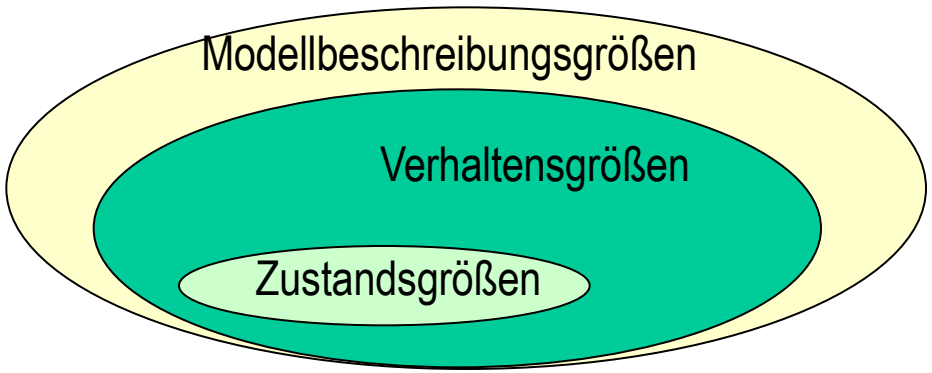
... sind

- Modellbeschreibungsgrößen, aus denen sich der Zustand eines Systems **vollständig** ergibt (Gedächtnis eines Systems)
  → Basis der Verhaltensbeschreibung

- Zustandsgrößen sind voneinander **unabhängig**
  → eine Zustandsgröße kann nicht als Kombination anderer Zustandsgrößen dargestellt werden

- sind **nicht immer eindeutig** definierbar

- sind i. Allg. strukturierte Größen



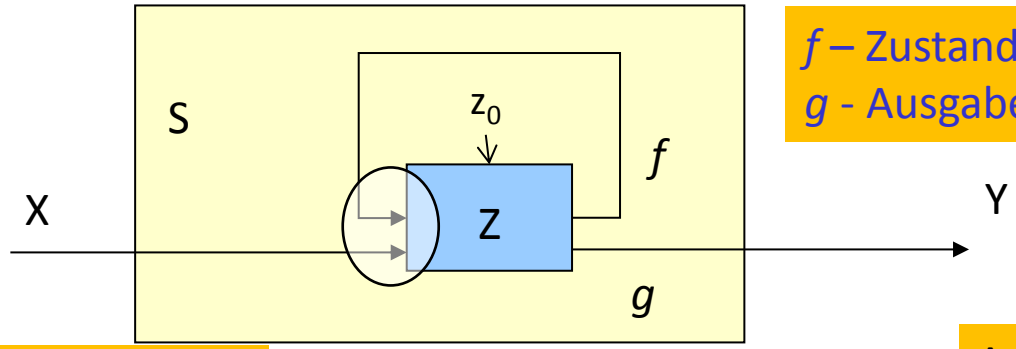
Zustandsänderungen (Prinzip)

- Sei Z n-dimensionaler Zustandsvektor (Zustand z = Belegung von Z), der Zustandsgrößen eines (Teil-)Systems S zu diesem Zeitpunkt beschreibt
- der (neue) Zustand ergibt sich aus dem bisherigen (aktuellen) Zustand bei Berücksichtigung von "Zuwachs" und "Reduktion (negativer Zuwachs)" für die Zustandsgrößen im betrachteten Zeitraum des Zustandswechsels
- ausgehend von einem ausgezeichneten Anfangszustand z_0

Zeit t_0 t_1 t_2 t_3

$z_0 \rightarrow z_1 \rightarrow z_2 \rightarrow z_3 \rightarrow \ldots$

Zustandsentwicklung in Abhängigkeit der Zeit



f – Zustandsübergangsfunktion,
g - Ausgabefunktion

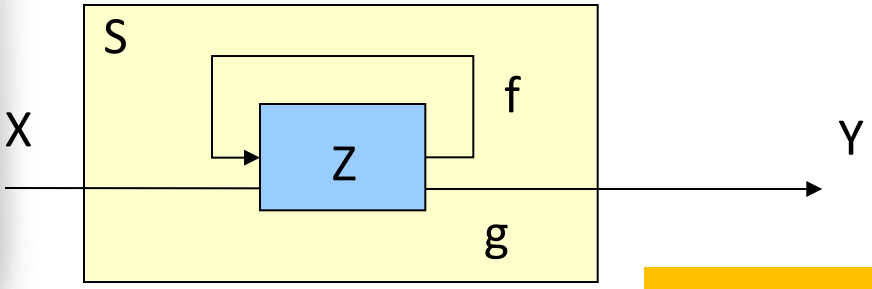
Eingabe X
Zuwachs
im Zeitintervall $(t_k, t_{k+1}]$

Änderung des
Zustandsvektors Z
im Zeitintervall $[t_k, t_{k+1}]$
in Abhängigkeit von $x(t_{k+1}), z(t_k)$

Ausgabe Y
äußere Reaktion
des Systems auf die Eingabe
im Zeitintervall $(t_k, t_{k+1}]$

Allgemeine (Teil-)Systemdefinition

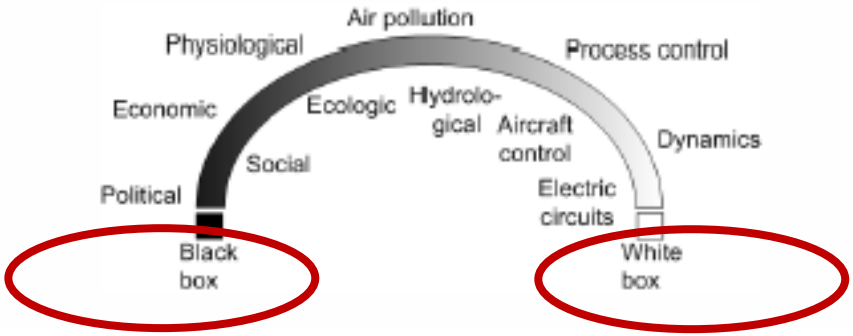
dient mehr der Klassifikation von Verhaltensmodellen



$S = (Z, z_0, X, Y, f, g, time)$

- Z Menge der möglichen Zustände
- $z_0 \in Z$ Anfangszustand
- X Menge der möglichen Eingaben
- Y Menge der möglichen Ausgaben
- time Zeitbasis als (T, <=, t_0) mit
  - Menge möglicher Zeitpunkte T,
  - einer Ordnungsrelation <= und
  - einem minimalen Element t_0
- f $Z \times X \times T \rightarrow Z$ als Zustandsübergangsfunktion
- g $Z \times X \times T \rightarrow Y$ als Ausgabefunktion

Modellklassifikation: Black-Box - White-Box



nach Karplus, 1977

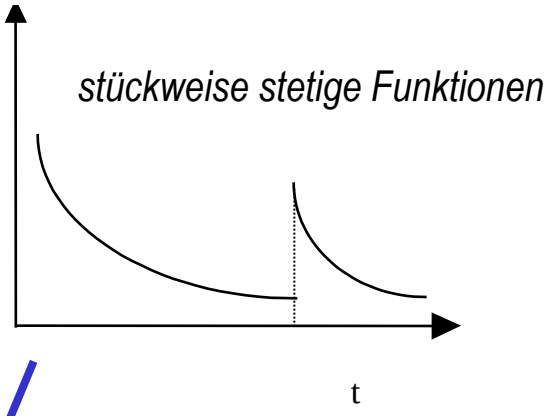
Systeme,
für die nur **Traces** von Zustandsänderungen bekannt sind

Systeme,
für die **Zustandsübergangsfunktionen** bekannt oder entwickelt werden können (Kausalmodelle)

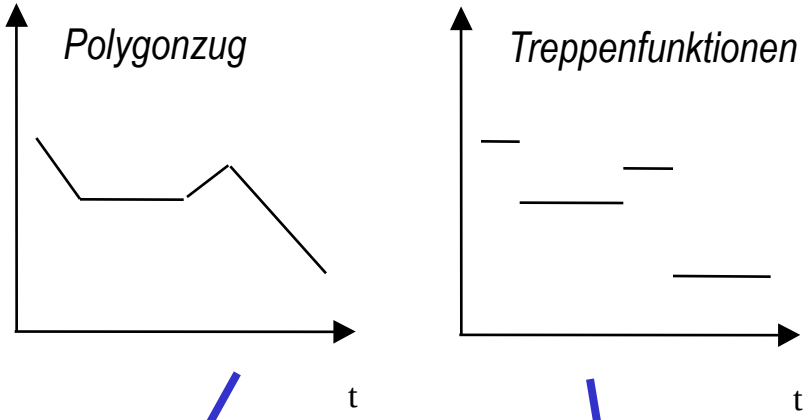
unser Fokus in OMSI

Arten von Zustandsänderungen

zeitkontinuierliche Zustandsänderung



zeitdiskrete Zustandsänderung



$z'(t) = f(z(t), x(t), t)$
mit $z(t) \in Z,\ x(t) \in X,\ t \in T$

$z(t_{n+1}) = f(z(t_n), x(t_{n+1}), t_{n+1})$
mit $z(t_n) \in Z,\ x(t_{n+1}) \in X,\ t_{n+1} \in T$

Differentialgleichungen

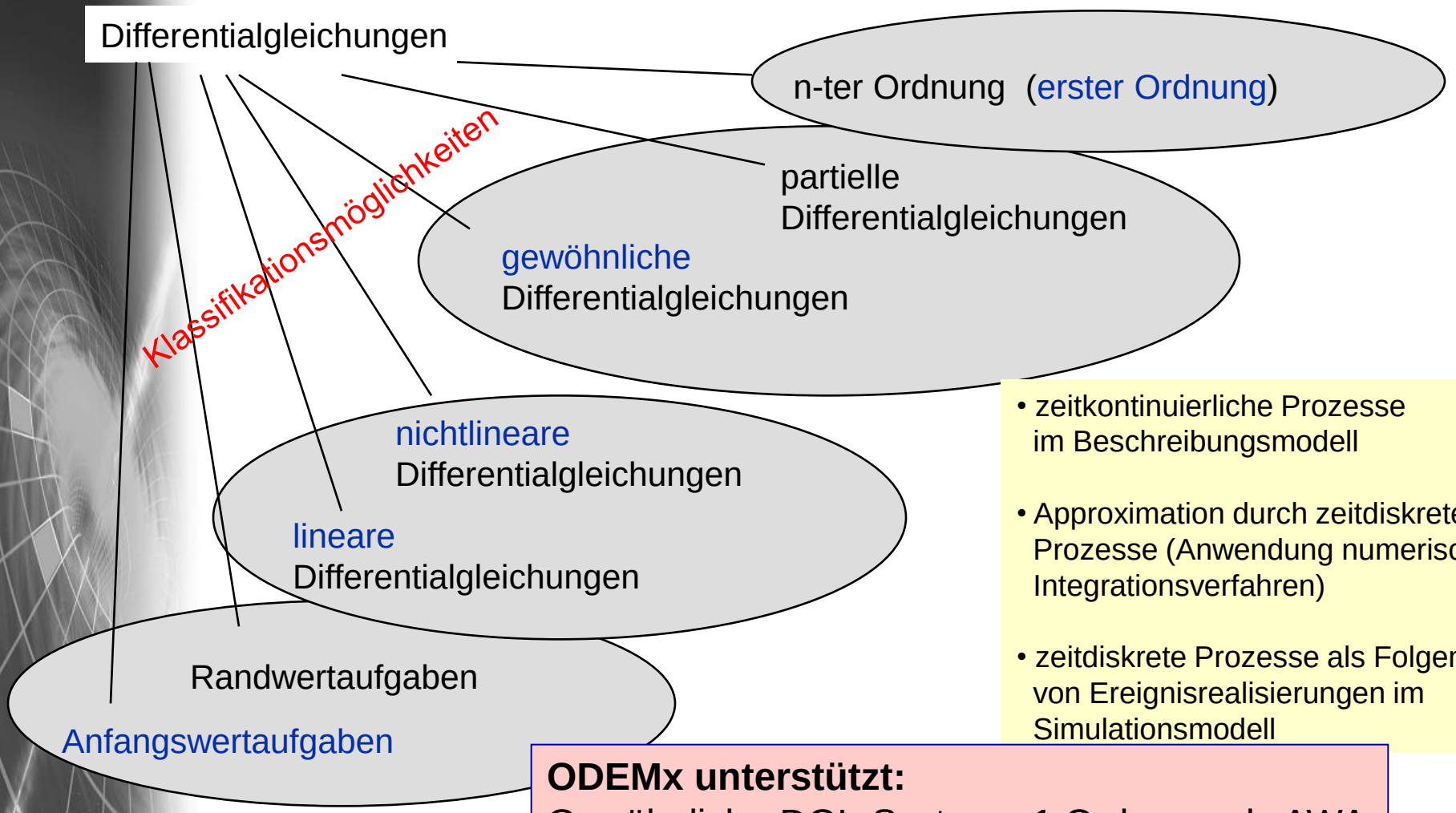
numerische Lösungsverfahren

Differenzengleichungen
zelluläre Automaten

Ereignissimulationen

Prozesssimulation

Verhaltensklassen zeitkontinuierlicher Art



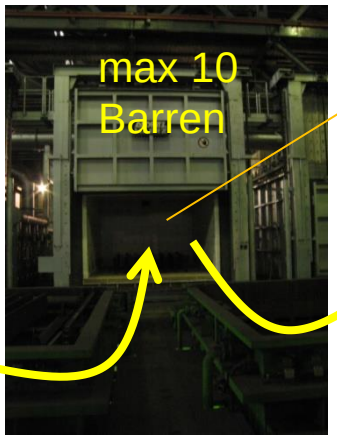
- zeitkontinuierliche Prozesse im Beschreibungsmodell
- Approximation durch zeitdiskrete Prozesse (Anwendung numerischer Integrationsverfahren)
- zeitdiskrete Prozesse als Folgen von Ereignisrealisierungen im Simulationsmodell

ODEMx unterstützt:
Gewöhnliche DGL-Systeme 1.Ordnung als AWA (linearer und nichtlinearer Art)

1. Einführung

1. Systemsimulation – was ist das?
2. Ein Blick zurück in die Anfänge
3. Modelle und Originale
4. Modellierungssprachen, Simulationsumgebungen
5. Bespiele aus der aktuellen Forschung
6. Paradigma der objektorientierten Modellierung
7. Einordnung von UML
8. Klassifikation dynamischer Systeme
9. M&S eines Niedertemperaturofens

Beschickung eines Niedrigtemperaturofens



25°C

~200°C

max.
500°C

380°C

Ziel: Bestimmung der Frequenz ankommender Eisenbarren für optimalen Ofendurchsatz (bei begrenztem Puffer)

Original

Bewertung /
Rückschlüsse /
Modellverbesserung

reales
bzw. gedachtes
Phänomen

Problemanalyse /
Abstraktion

UML

Klassen- und
Objekt-Diagramme

Experimente

Ausführungs-
Modell im Speicher
Simulator

Ziel

informales
/semiformales
Systemmodell

*Struktur + Verhalten
(bei Änderungen in
Raum und Zeit)*

Zustandsdiagramme

bei Animation
der Abläufe

softwaretechnische
Umsetzung

formales
mathematisches
Simulationsmodell

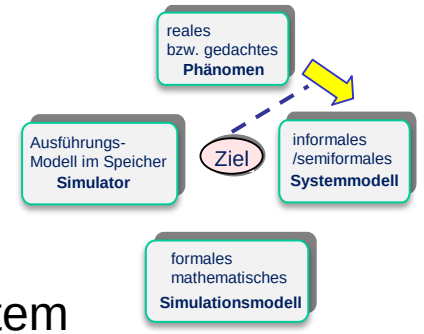
Formalisierung/
Programmierung

C++

ODEMx

1. Schritt: Problemanalyse

- allgemein -



mit informaler Darstellung des Phänomens als System

(aus systemtheoretischer Sicht)

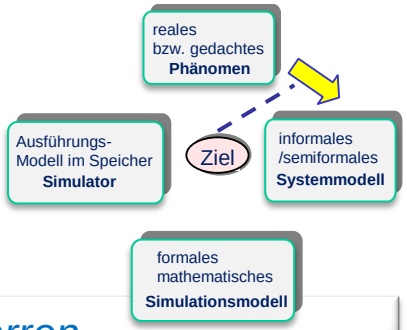
bei Identifikation von

- Systemelementen und Systemumgebung
- Relationen (Wechselwirkungen) zwischen Systemelementen untereinander und zur Umgebung

zur Erbringung des bereits bestimmten

- Systemzwecks / Untersuchungsziels

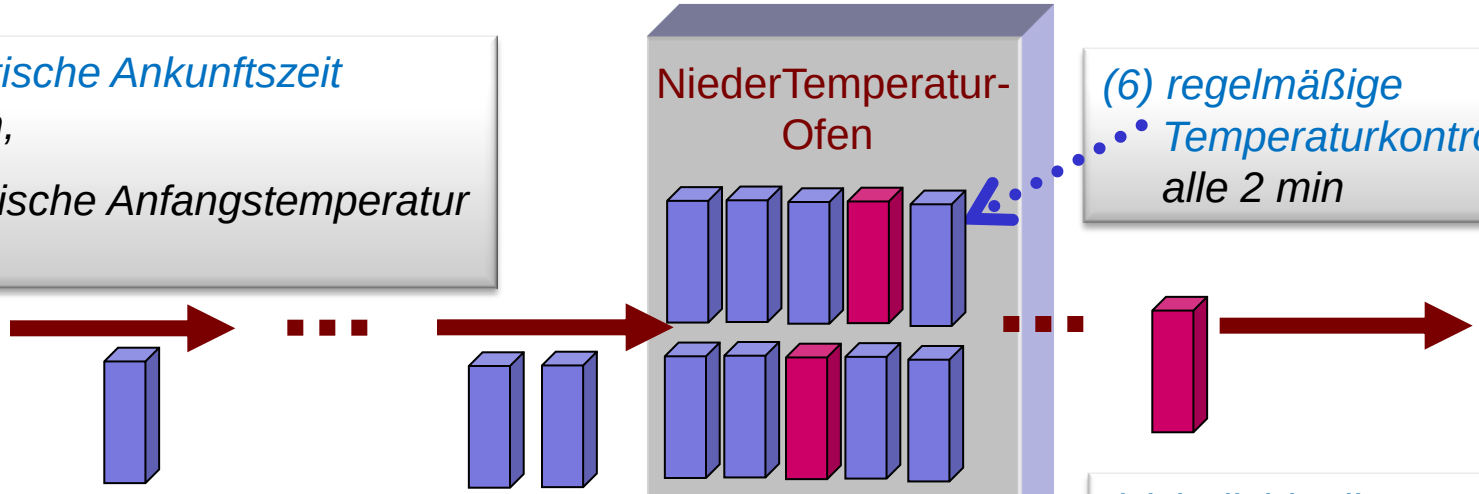
2. Schritt: Problemanalyse
- Informale Darstellung -



(5) Änderung der Ofentemperatur
in Abhängigkeit der Belegung
(maximal 500°C)

(4) Erwärmung der Barren
auf Zieltemperatur von mind. 380 °C

(1) stochastische Ankunftszeit
2-10 min,
stochastische Anfangstemperatur
~200 °C



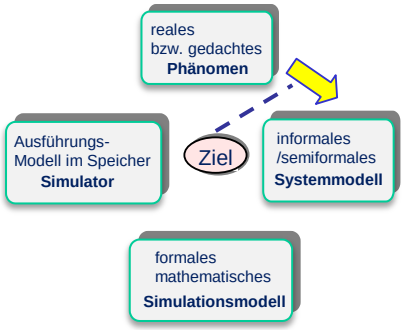
(6) regelmäßige
Temperaturkontrolle
alle 2 min

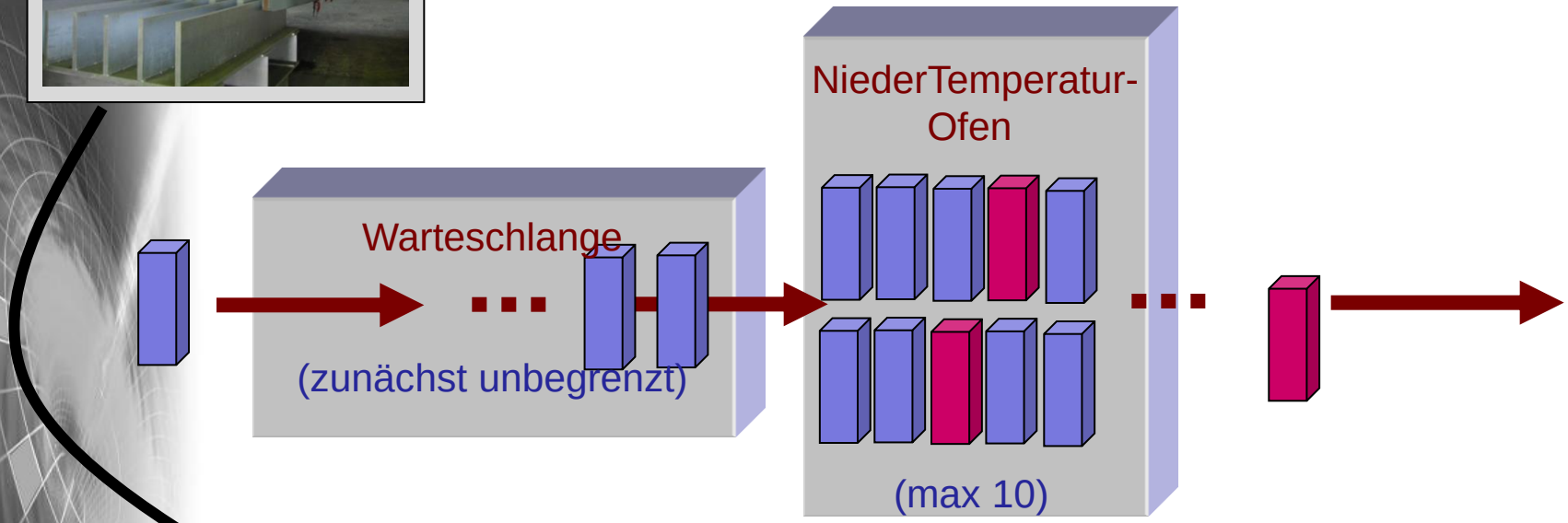
(2) evtl. Abkühlung
auf Umgebungstemperatur
25 °C möglich

(3) Beschickung
Ofen hat begrenzte
Aufnahmekapazität
von 10 Barren

(7) individuelle Entnahme
und
(8) Nachfüllung

2. Schritt: Problemanalyse
- Informale Darstellung -

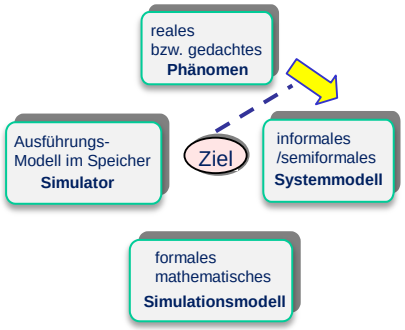




Vernachlässigung von Transportressourcen:
Verfügbarkeit, Zeitverbrauch
→ Befüllung und Entnahme von Ofen oder Warteschlange sind zunächst zeitlos

3. Schritt: partielle Modellformalisierung

- UML-Notation -



informal beschriebenes Systemmodell

- **Struktur** (Objekt- und Klassendiagramm) und
- **Verhalten** als Ensemble asynchron kommunizierender Automaten

  Zustandsdiagramm

bei Identifikation

- verwendeter **Modellklassen**
- und von **Zustands-** und **Bewertungsgrößen**

bei (zunächst noch) verbaler **Verhaltensbeschreibung**

passive Klassen

Assoziationen

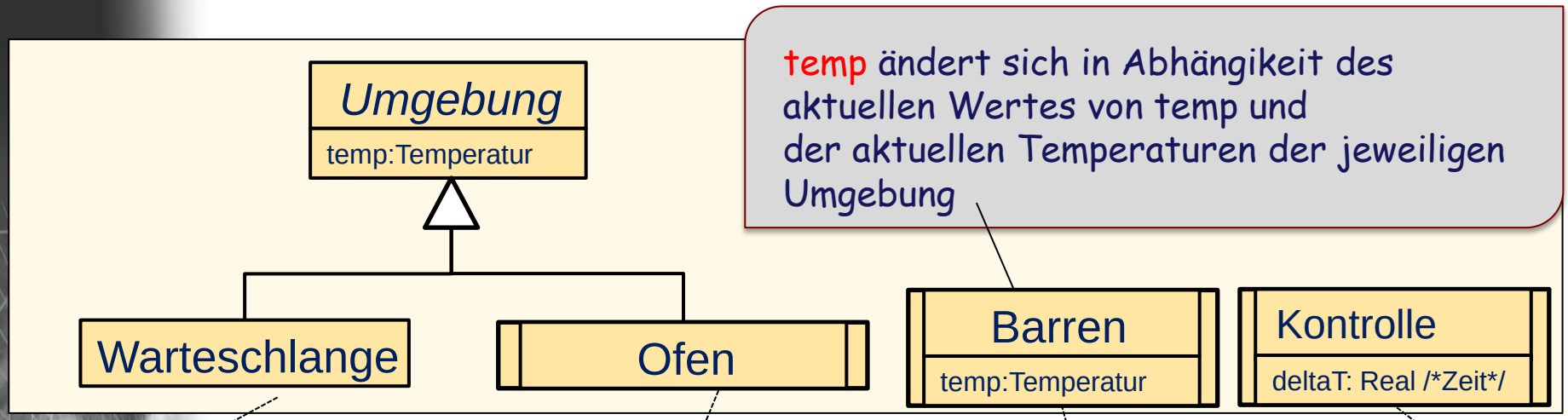
aktive Klassen

lifeCycle

zeitdiskrete Zustandsänderungen

zeitkontinuierliche Zustandsänderungen

Klassendiagrammvarianten



Objekte von konkreten Umgebungs-Ableitungsklassen stehen in Wechselwirkung mit eingefügten (zugeordneten) Barren-Objekten

System (Umgebung, Grenze, Systemkomponenten)

Was ist zu prüfen?
- können die Systemelemente in ihrer Wechselwirkung die Systemfunktionalität erbringen?
- Sind die Wechselwirkungen mit der Umgebung rückkopplungsfrei?

Achtung: noch kein sauberes UML