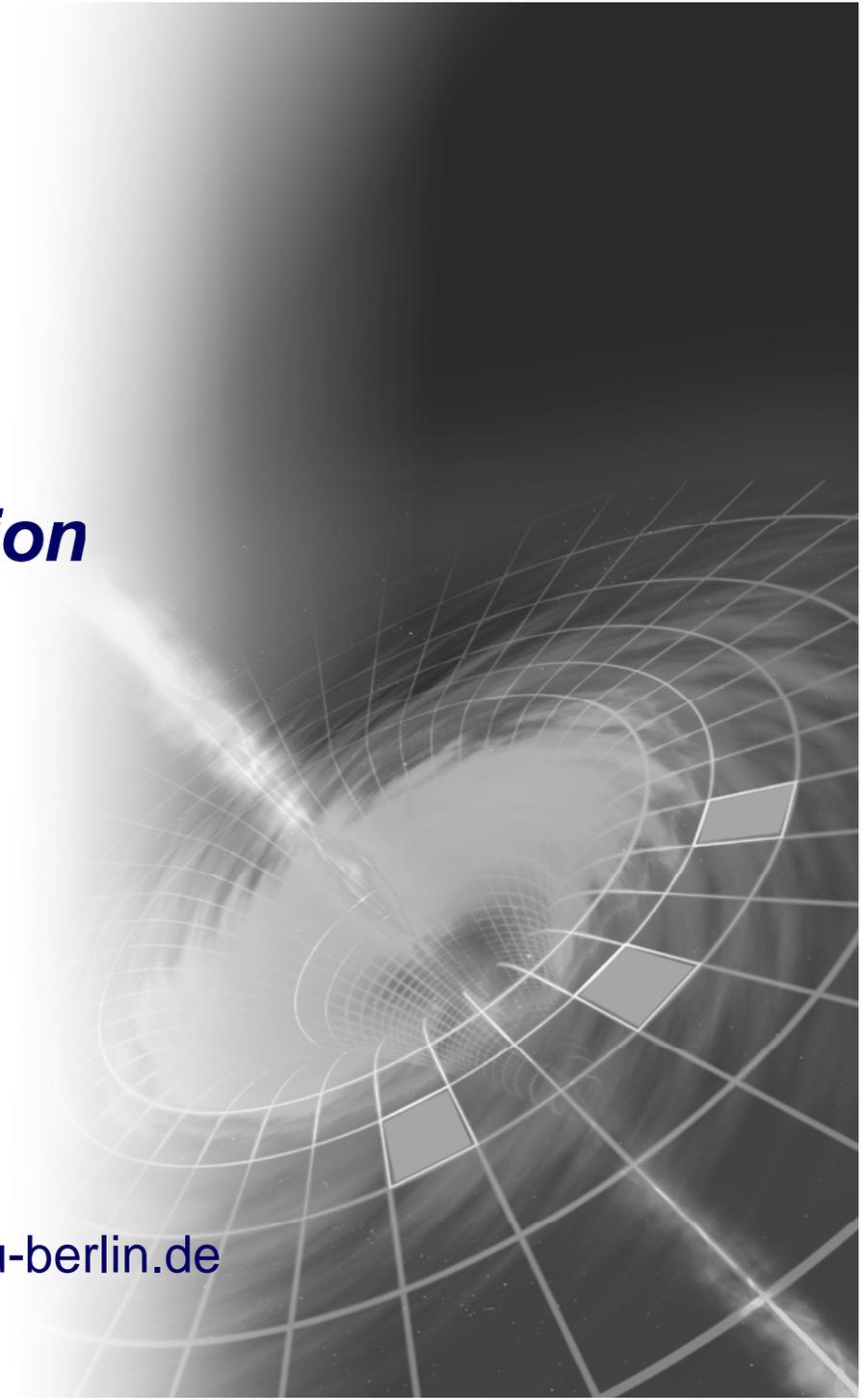


Kurs OMSI im WiSe 2010/11

Objektorientierte Simulation mit ODEMx

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

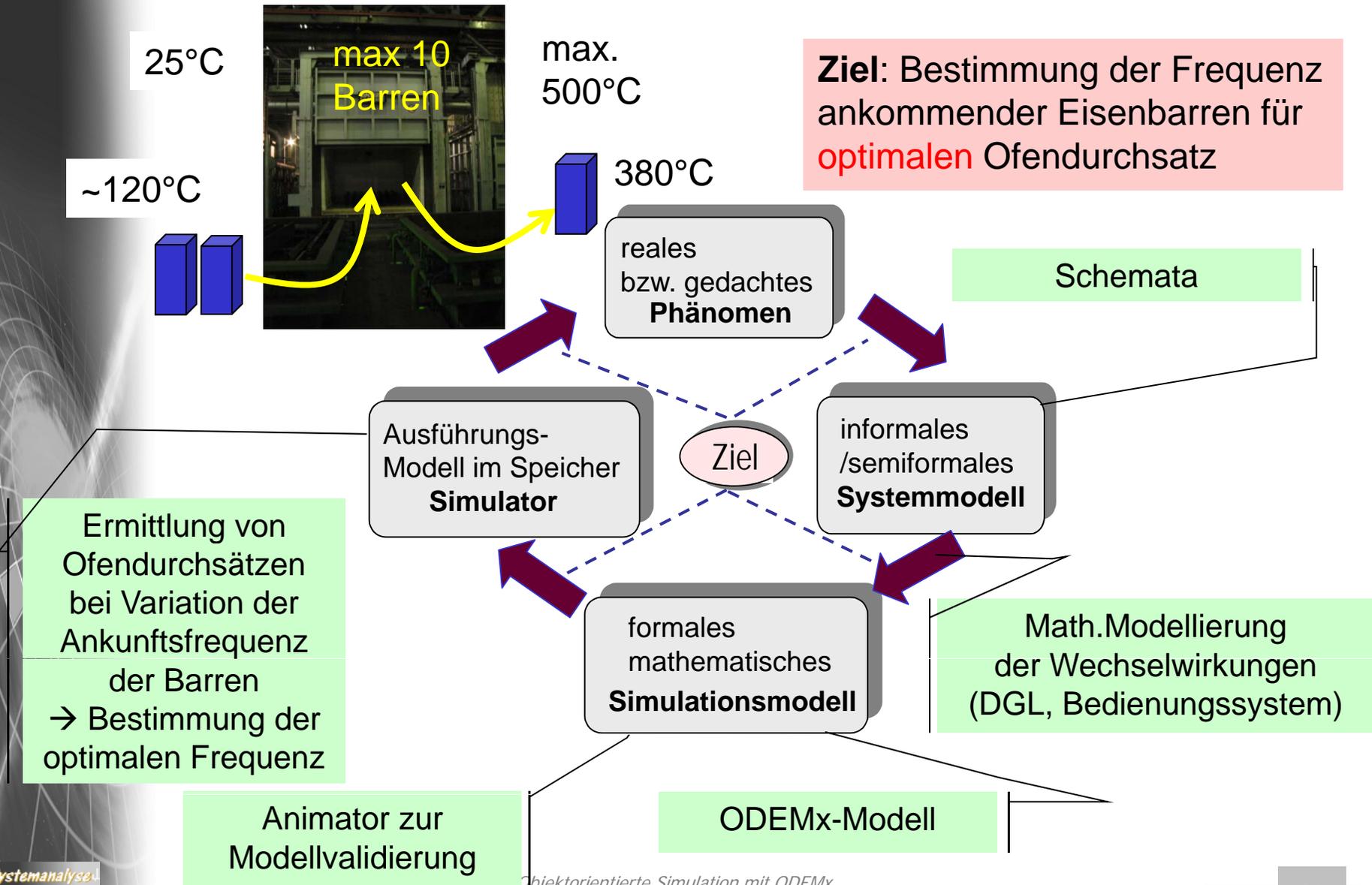
fischer|ahrens|eveslage@informatik.hu-berlin.de



Einführung

1. Systemsimulation – was ist das?
2. Ein Blick zurück in die Anfänge
3. Modelle und Originale
4. Modellierungssprachen, Simulationsumgebungen
5. Beispiele aus der aktuellen Forschung
6. Paradigma der objektorientierten Modellierung
7. Klassifikation dynamischer Systeme
8. Konzept der Next-Event-Simulation
9. Modellierung eines Niedertemperaturofens

Beschickung eines Niedrigtemperaturofens



1. Schritt: Problemanalyse

- allgemein -

mit informaler Darstellung des Phänomens als System
(aus systemtheoretischer Sicht)

bei Identifikation von

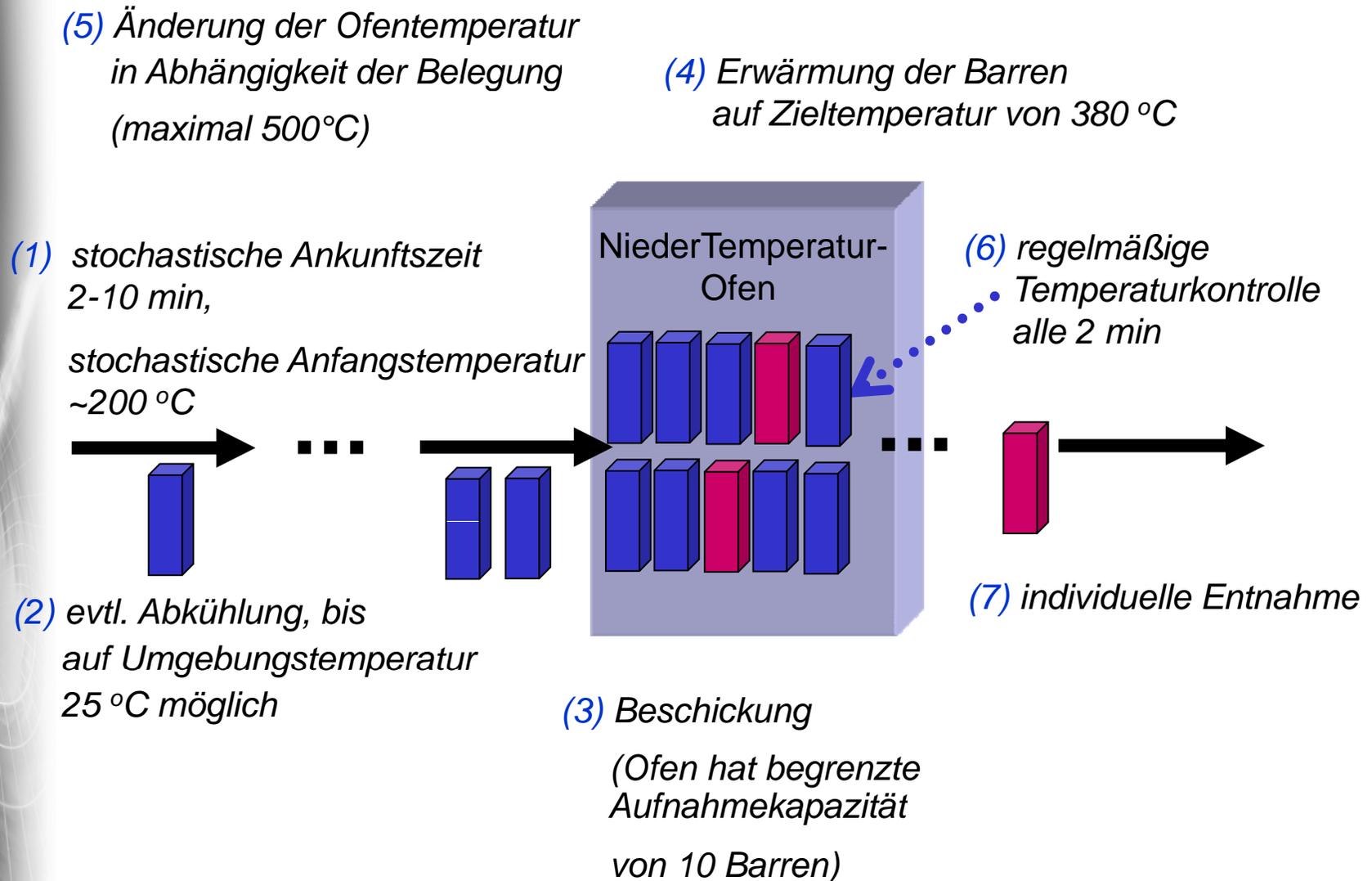
- Systemelementen und Systemumgebung
- Relationen (Wechselwirkungen) zwischen Systemelementen untereinander und zur Umgebung

zur Erbringung des bereits bestimmten

- Systemzwecks / Untersuchungsziels

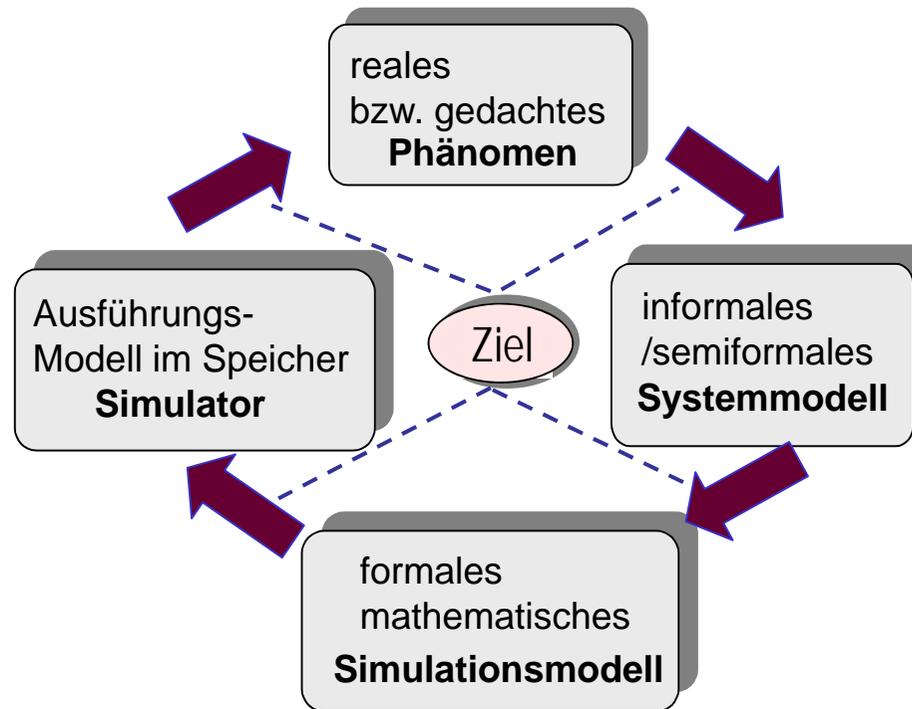
1. Schritt: Problemanalyse

- Informale Darstellung am Beispiel -



1. Schritt: Problemanalyse

- Identifikation des prinzipiellen Herangehensweise -



Untersuchungsziel:

Wie beeinflusst die Intensität des Ankunftsstroms den Ausgangsstrom von Barren?

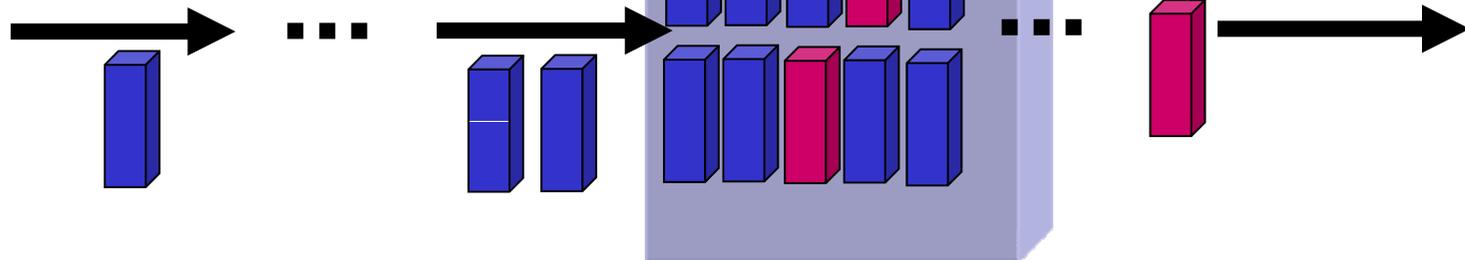
Methodik: Verhaltensmodellierung (Erfassung von Belegungszeiten), Wechselwirkung der Ofentemperatur und der jeweiligen Barrentemperaturen
Simulation unterschiedlicher Varianten, Vergleich

1. Schritt: Problemanalyse

- Identifikation von Modellbeschreibungsgrößen -

Warteschlangenstatistik

- Belegung
- max. Länge
- mittlere Wartezeit



Experiment/Beobachtung
bei Veränderung der
Kontrollabstände

Durchsatz

(Anzahl fertiger Barren
nach 8 h)

Experiment/Beobachtung
bei Veränderung der
Beschickungsintensität

Ofenstatistik

- Temperaturen von Ofen
und Barren
- Belegung des Ofens
- Auslastung des Ofens

2. Schritt: Modellformalisierung **- allgemein -**

von

- Struktur (Objekt- und Klassenstruktur) und
- Verhalten

bei Identifikation

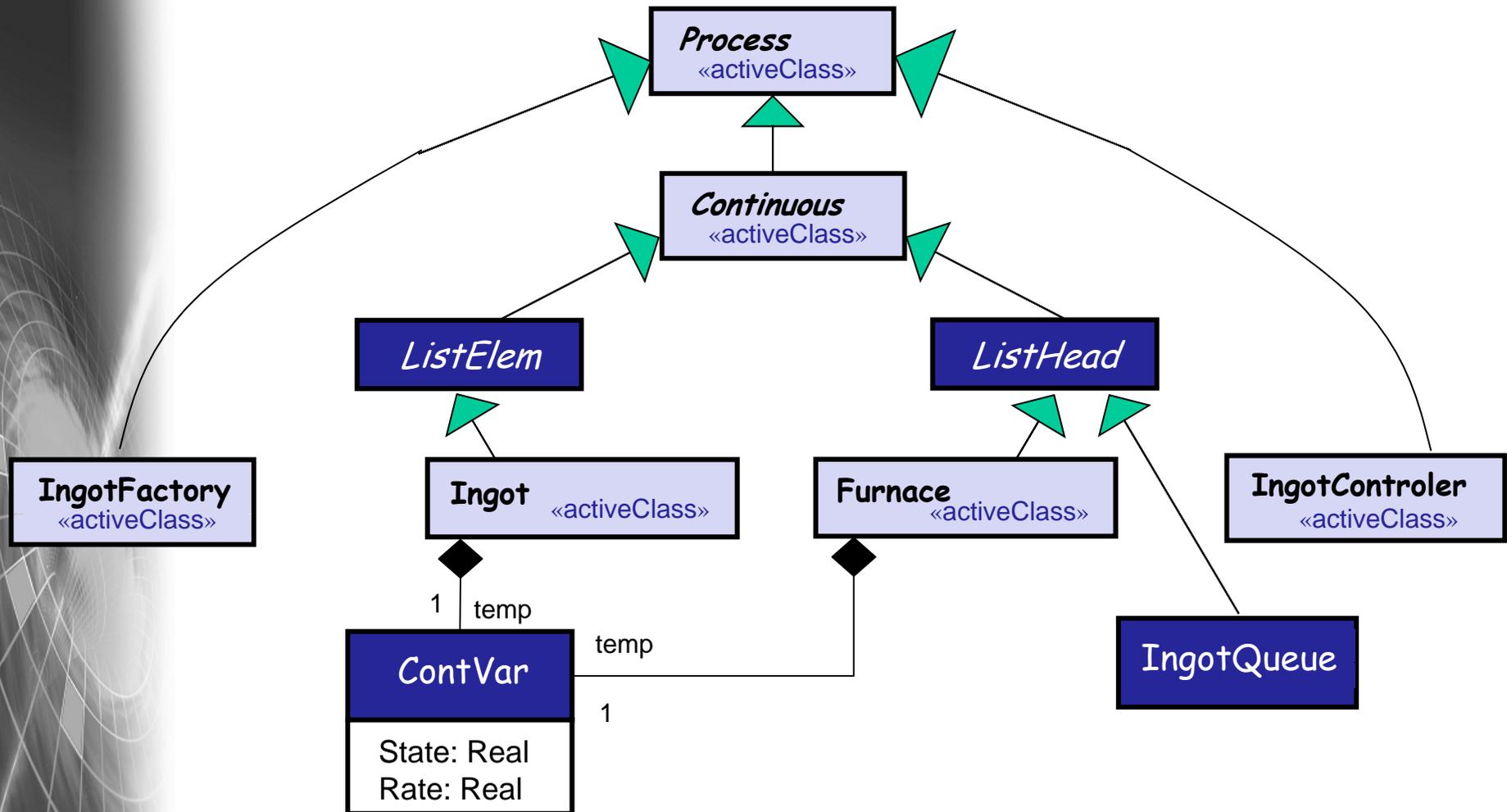
- verwendeter Modellklassen
- und von Zustands- und Bewertungsgrößen

bei Bestimmung

- der Verhaltensfunktionen/-Gleichungen

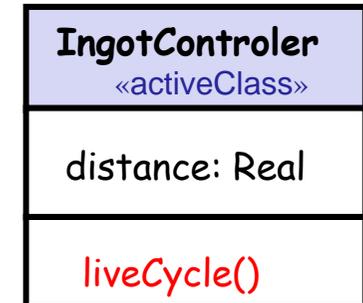
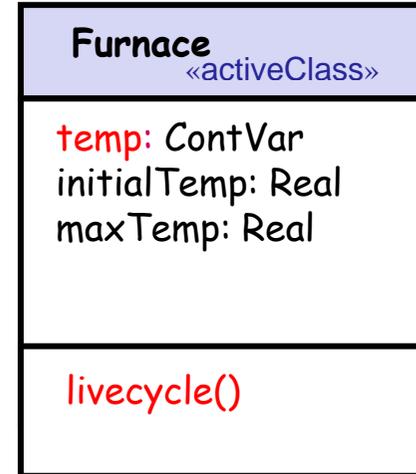
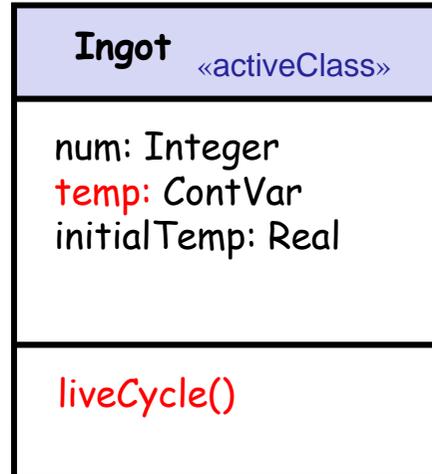
2. Schritt: Modellformalisierung

- Identifikation von Strukturelementen (d.h. ihrer Klassen)



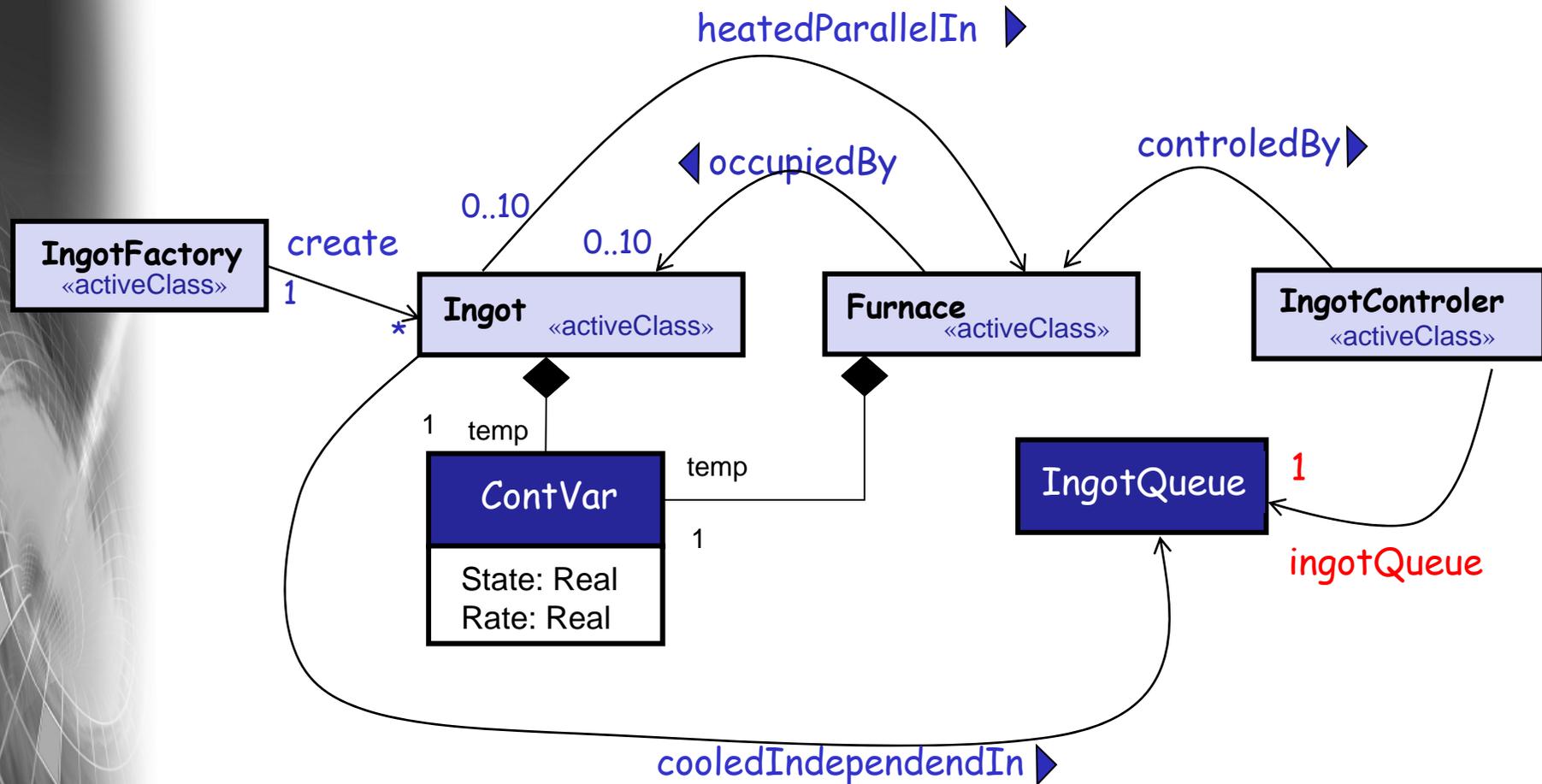
2. Schritt: Modellformalisierung

- Identifikation von Modellgrößen der Strukturelemente



2. Schritt: Modellformalisierung

- Identifikation von Strukturelementen (d.h. ihrer Klassen)



Achtung

jedes navigierbare Assoziationsende hat implizit einen Namen (kleingeschriebener Klassenname)

2. Schritt: Modellformalisierung

- Festlegung der Art der Zustandsänderung → Verhaltensklassen

von

- Struktur (Objekt- und Klassenstruktur) und
- Verhalten

bei Identifikation

- verwendeter Modellklassen
- und von Zustands- und Bewertungsgrößen

bei Bestimmung

- der Verhaltensfunktionen/-Gleichungen

passive Klassen

aktive Klassen

UML bietet
Zustandsmaschine
zur Beschreibung

lifeCycle

zeitdiskrete
Zustandsänderungen

zeitkontinuierliche
Zustandsänderungen

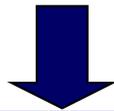
2. Schritt: Modellformalisierung

- Identifikation von Verhaltensarten -

verwendete Modelltypen

zeitkontinuierliche Vorgänge:

- Veränderung der Ofen- und
- Eisenbarrentemperaturen



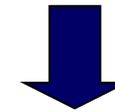
Beschreibung als

**Gewöhnliche
Differentialgleichungen** (AWA)

- Stoffkonzentrationen,
- Gasdrücke,
- Temperaturen, ...

zeitdiskrete Vorgänge:

- Ankunft/Anordnung der Barren in Warteschlange und Ofen,
- Ablesen der Temperaturen
- Entfernung der Barren



Beschreibung als

überlagerte Folgen von Ereignissen

- Bedienungsvorgänge,
- Steuerungseingriffe
- Kommunikationen

2. Schritt: Modellformalisierung

- Identifikation zeitkontinuierlicher Verhaltensklassen -

Differentialgleichungen

Klassifikationsmöglichkeiten

n-ter Ordnung (erster Ordnung)

partielle
Differentialgleichungen

gewöhnliche
Differentialgleichungen

nichtlineare
Differentialgleichungen

lineare
Differentialgleichungen

Randwertaufgaben

Anfangswertaufgaben

- 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)

2. Schritt: Modellformalisierung - Beschreibung des Verhaltens bei gewählter Modellklasse -

Modellierung der Abkühlungsvorgänge (Barren)

Ingot.deriveTemp

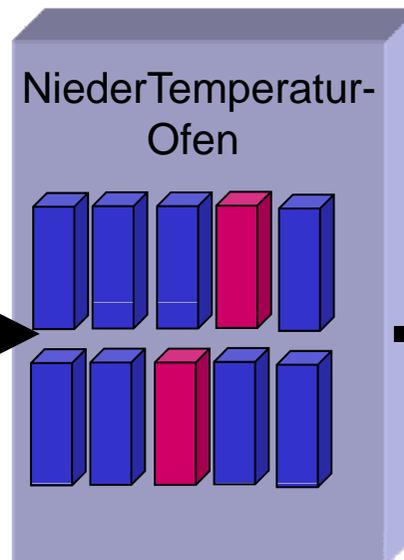
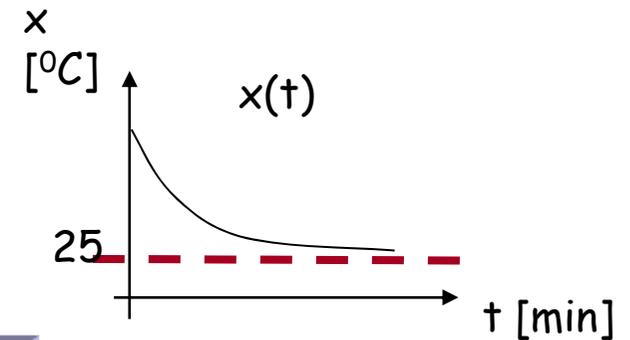
Ingot.temp

Barrentemperatur

$$x'(t) = \{ u(t) - x(t) \} / 7$$

Anfangstemperatur: ca. 200 °C

Ingot.envTemp()



Umgebungstemperatur

$$u(t) = 25$$

$$x'(t) = (25 - 200) / 7$$

negativer Wert für $x'(t)$ → Reduktion von $x(t)$

Änderung von $x(t)$:

- zunächst stark,
- dann schwach,
- asymptotische Annäherung an den Wert 25

2. Schritt: Modellformalisierung - Beschreibung des Verhaltens bei gewählter Modellklasse -

Modellierung der Erhitzungsvorgänge (Barren)

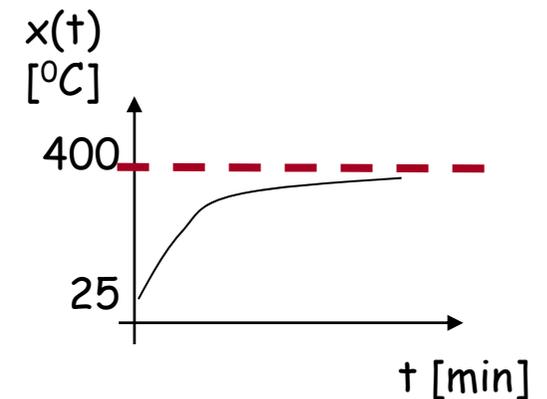
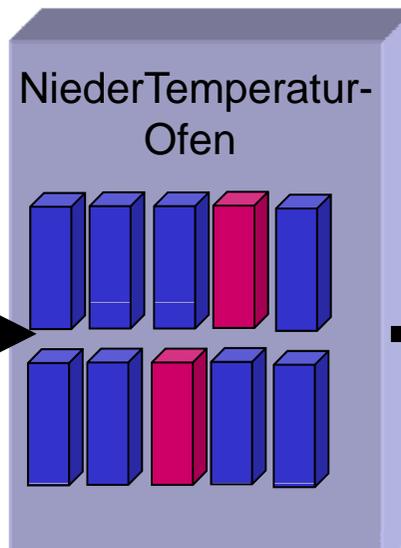
Umgebungstemperatur $u(t) = y(t)$

Barrentemperatur

$$x'(t) = \{ u(t) - x(t) \} / 7$$

Eintrittstemperatur:
200 °C .. 25 °C

Ofentemperatur $y(t)$
~400 °C



$$x'(t) = (400 - 25) / 7$$

positiver Wert für $x'(t)$, d.h. Zunahme von $x(t)$

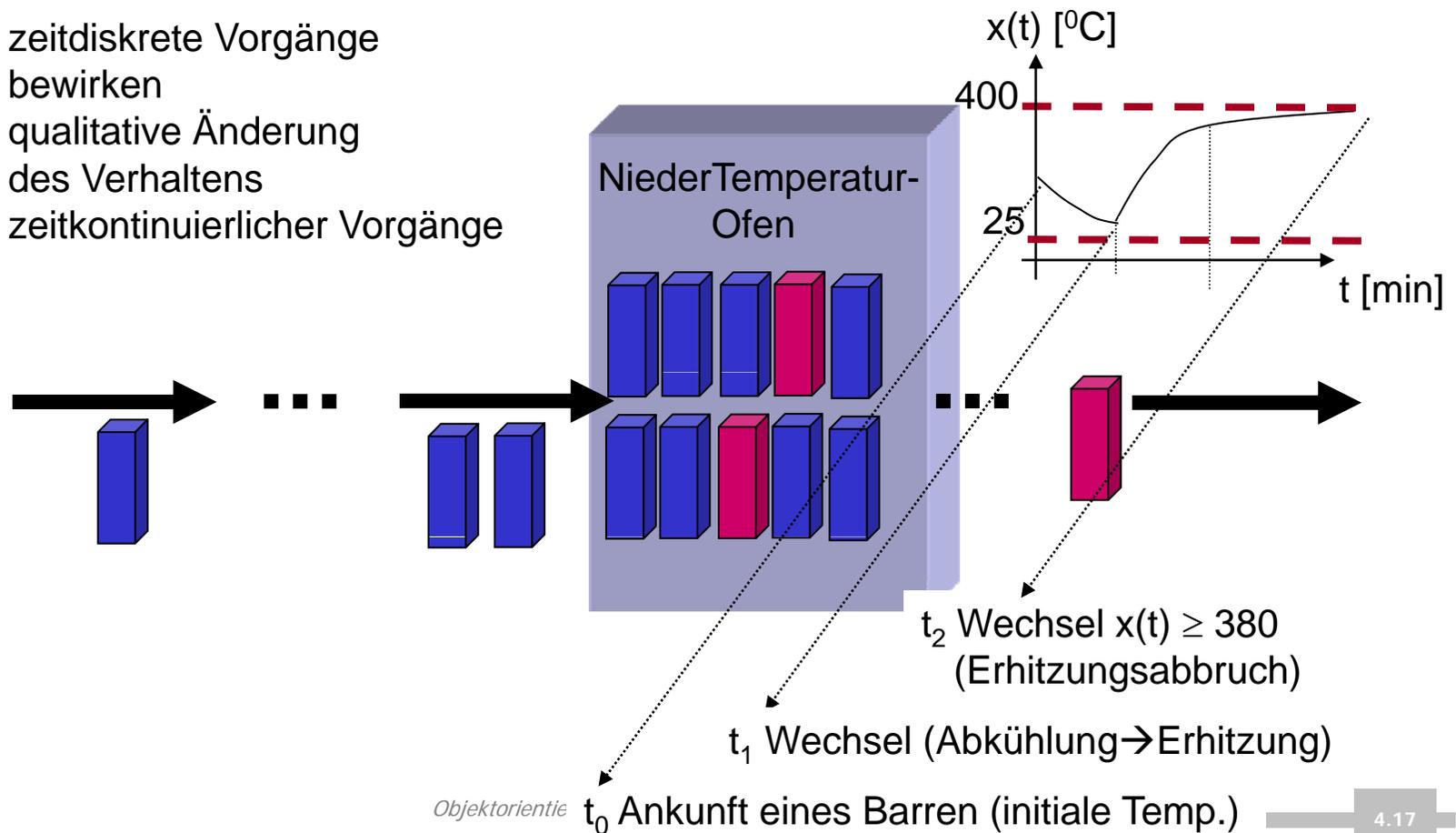
Änderung von $x(t)$:

- zunächst stark, dann schwach und schwächer,
- asymptotische Annäherung an den Wert 400 (wenn Ofentemperatur konstant bliebe !!!)

2. Schritt: Modellformalisierung - Beschreibung des Verhaltens bei gewählter Modellklasse -

Modellierung von (zeitdiskreten) Ereignissen

zeitdiskrete Vorgänge
bewirken
qualitative Änderung
des Verhaltens
zeitkontinuierlicher Vorgänge



2. Schritt: Modellformalisierung

- Identifikation des Verhaltens bei gewählter Modellklasse -

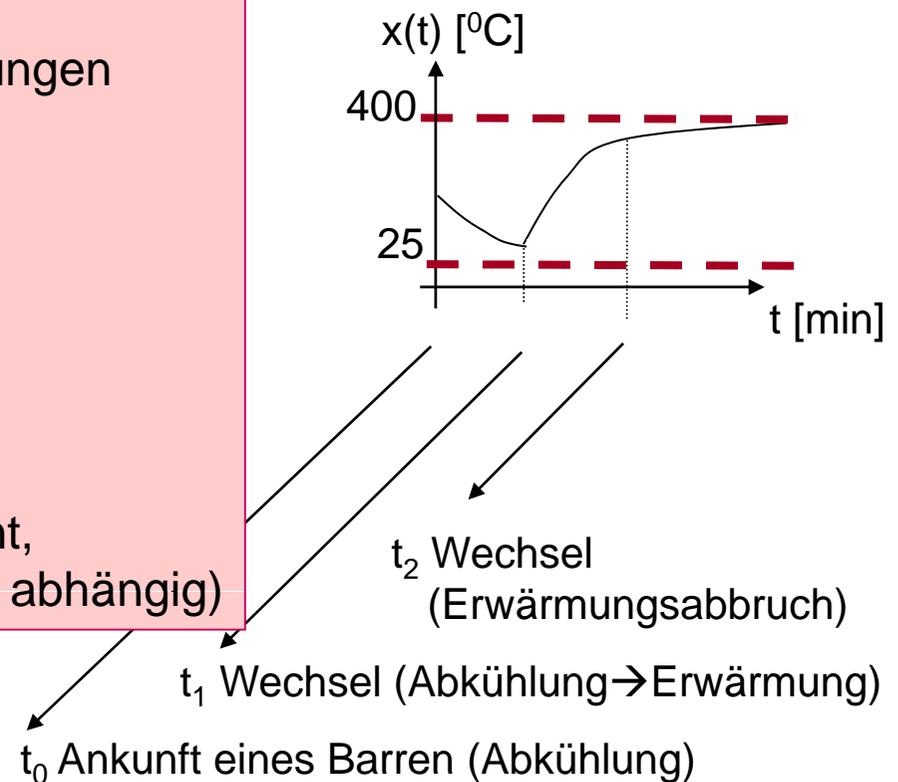
Modellierung von (zeitdiskreten) Ereignissen

Ereignismodellierung

Erfassung von Modellgrößenänderungen zu einem festen Zeitpunkt

Ereignisklassen

- Zeitereignisse
(Zeitpunkt bekannt)
- Zustandsereignisse
(Zeitpunkt nicht a priori bekannt,
vom Erreichen eines Zustands abhängig)



2. Schritt: Modellformalisierung - Identifikation des Verhaltens bei gewählter Modellklasse -

Modellierung der Änderung der Ofentemperatur

Ofentemperatur y als $y(t)$

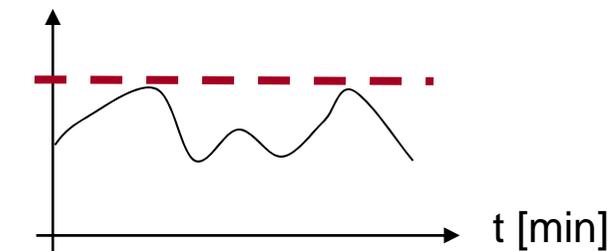
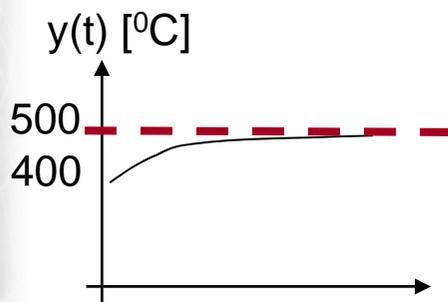
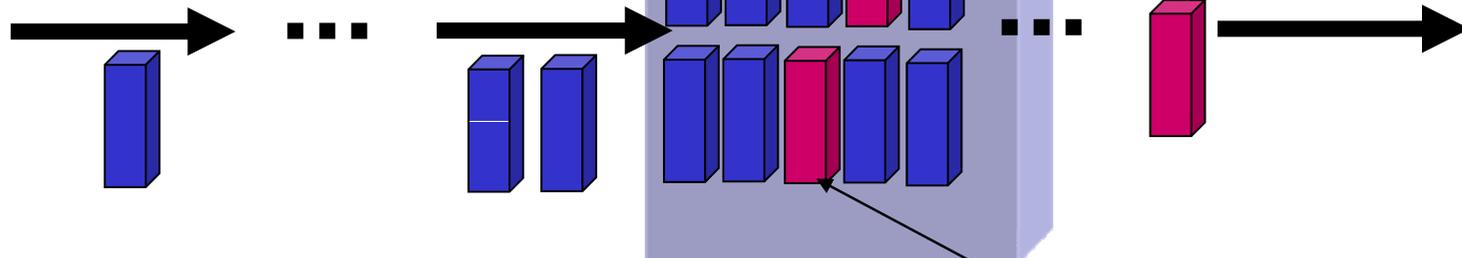
$$y'(t) = 500 - y(t) - n(t) \cdot \{y(t) - m(t)\}$$

Furnace.deriveTemp

Furnace.temp

NiederTemperatur-Ofen

- Energiezufuhr: max. 500°C
- $n(t)$: Anzahl von Barren im Ofen
- $m(t)$: mittlere Temperatur der Barren im Ofen
- Anfangstemperatur: 400°C



Zieltemperatur

$$x(t) = 380 \text{ } ^\circ\text{C}$$

per Beobachtung