

# ***Proseminar***

*(Software-Spezifikation mit UML 2)*

***Systemspezifikation mit SysML  
im SoSe 2010***

## ***Einführung***

Prof. Dr. Joachim Fischer  
Dipl.-Inf. Andreas Blunk

fischer@informatik.hu-berlin.de

# Themen, 1. Runde

#	Thema	Literatur	Name
1	UML: Klassendiagramm mit Objekt, Klasse, Attribut, Operation	Balzert, LE2, S.19-37	Martin Fobian
2	UML: Klassendiagramm mit Assoziation, Generalisierung, Paket	Balzert, LE3, S.41-57	Florian Hönicke
3	UML: Kompositionsstrukturdiagramm mit Strukturierte Klasse, Rolle, Konnektor, Port, Schnittstelle, Signal	Weilkiens S.225-230 und S. 220-223, Born/Holz/Kath S.131-149	
4	UML: Verwendung von OCL in UML-Modellen	The Object Constraint Language Kap. 1 und 2	
5	UML: Use-Case-Diagramm mit Use-Case, Akteur, Include-Beziehung, Extends-Beziehung	Weilkiens S.231-237, Balzert S.62-69	
6	UML: Aktivitätsdiagramm mit Aktivität, Aktion und Pin, Aktivitätskante, Start- und Endknoten, Entscheidung und Zusammenführung, Splitting und Synchronisation	Weilkiens S.238-247 und S.249-256	
7	UML: Aktivitätsdiagramm mit Parametermenge, Unterbrechbarer Aktivitätsbereich, Mengenverarbeitung, Aktivitätspartition	Weilkiens S.248 und S.257-260	
8	UML: Zustandsdiagramm	Weilkiens S.261-274	
9	UML: Interaktionsdiagramm in Form des Sequenzdiagramms	Weilkiens S.275-287	
10	UML: sonstige Modellelemente - Diagrammrahmen, <i>Stereotypen</i> , Informationsfluss, Kommentar	Weilkiens S.289-298, Born/Holz/Kath S.245-255	

# Themen, 2. Runde

#	Thema	Literatur	Name
11	SysML: Anforderungsdiagramm	Weilkiens S.304-318	
12	SysML: Blockdiagramm	Weilkiens S.324-336	
13	SysML: Zusicherungsdiagramm	Weilkiens S.337-341	
14	SysML: Aktivitätsdiagramm	Weilkiens S.342-351	
15	SysML: Zustandsdiagramm, Interaktionsdiagramm, Zuteilungsbeziehung, sonstige Modellierungselemente	Weilkiens S.352-357	
16	SYSMOD: Fallbeispiel, Anforderungen ermitteln	Weilkiens S.37-63 + Kap.5	
17	SYSMOD: Systemkontext modellieren	Weilkiens S.64-87 + Kap.5	
18	SYSMOD: Anwendungsfälle modellieren 1	Weilkiens S.88-108 + Kap.5	
19	SYSMOD: Anwendungsfälle modellieren 2	Weilkiens S.109-139 + Kap.5	
20	SYSMOD: Anwendungsfälle realisieren	Weilkiens S.149-175 + Kap.5	

# Die UML

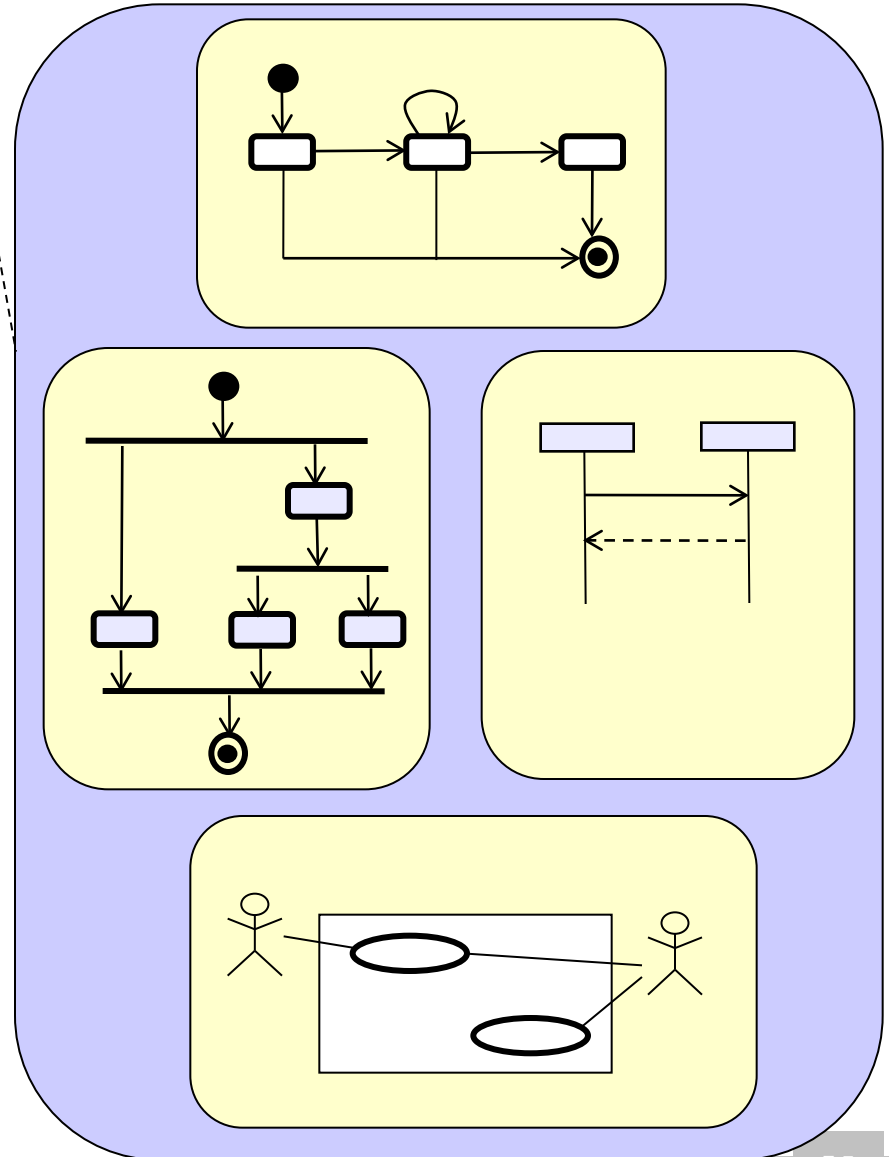
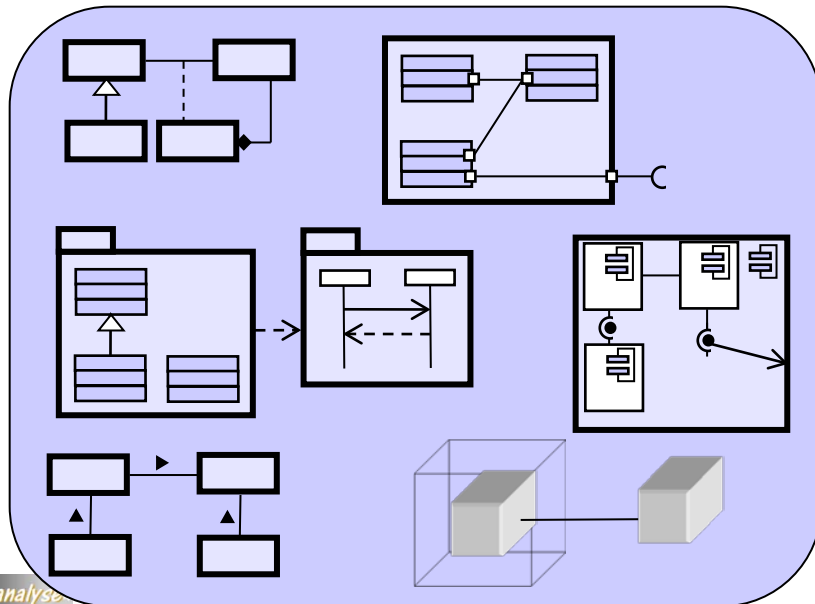


„Wenn die Sprache nicht stimmt,  
ist das was gesagt wird, nicht das, was gemeint ist.“  
(Konfuzius)

- UML = Unified Modeling Language
- ... ist zunächst Standardsprache (der OMG) zur Visualisierung, Spezifikation, Konstruktion und Dokumentation komplexer Softwaresysteme
- ... kombiniert Konzepte der
  - Objektorientierten Modellierung
  - Datenmodellierung (Entity-Relationship-Diagramme)
  - Business-Modellierung (Work Flows)
  - Komponentenmodellierung
  - Verhaltensmodellierung (Erweiterte Zustandsautomaten)
  - ...
- UML-Modelle sind in erster Linie graphische Repräsentationen in Form von Diagrammen

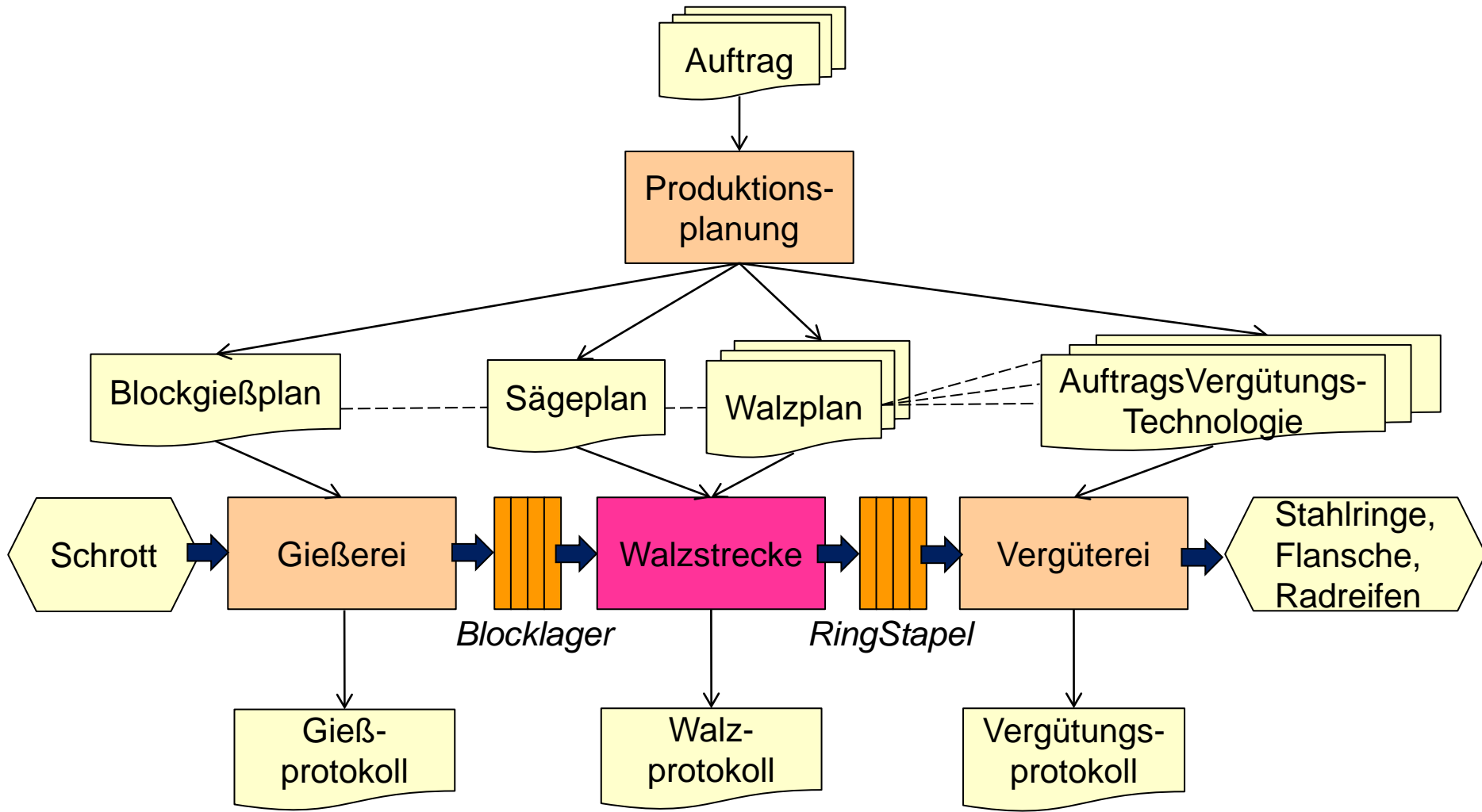
551 v. Chr. bis 479 v. Chr.

# Diagrammarten zur Anordnung von Entitäten: Struktur und Verhalten



# *Einführung*

1. Organisatorisches
  2. Trends in der Software-Entwicklung
  3. Modelle in UML
  4. Grundsätzliches zu Systemen und Modellen
- Beispiel eines komplexen dynamischen Systems
  - Objektorientiertes Modellierungsparadigma
  - Klassifikation dynamischer Systeme
  - Beispiel: Niedrigtemperaturofen

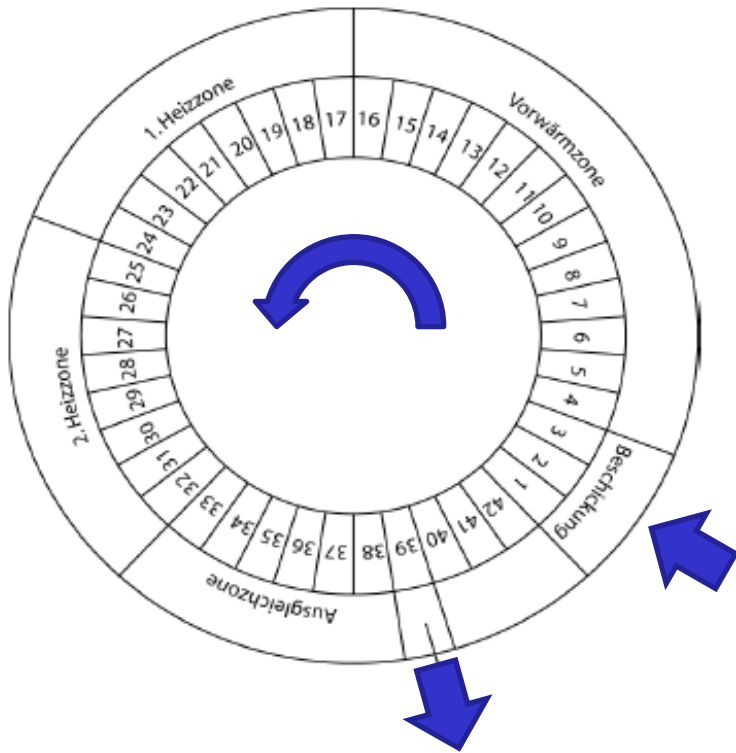


# Blöcke → Säge → Blöckchen





# Drehherdofen



# Blöckchen → Stauch- und Lochpresse → Ring



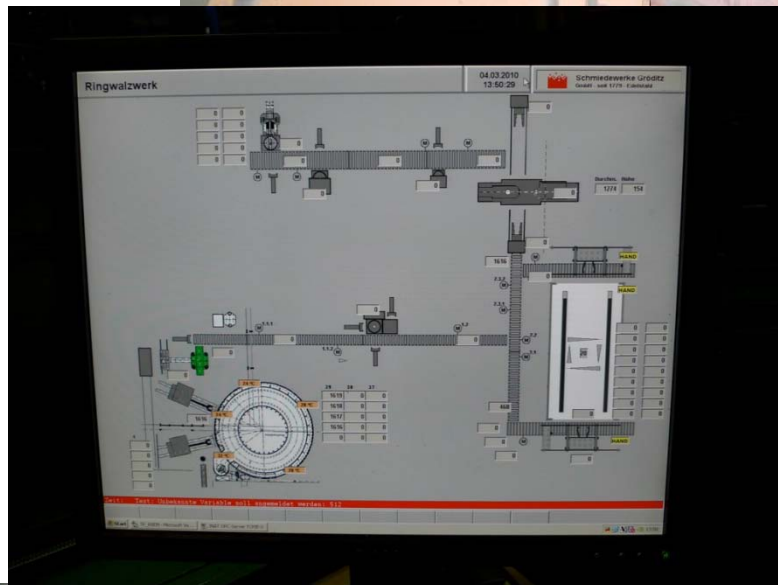
# Aufweitpresse

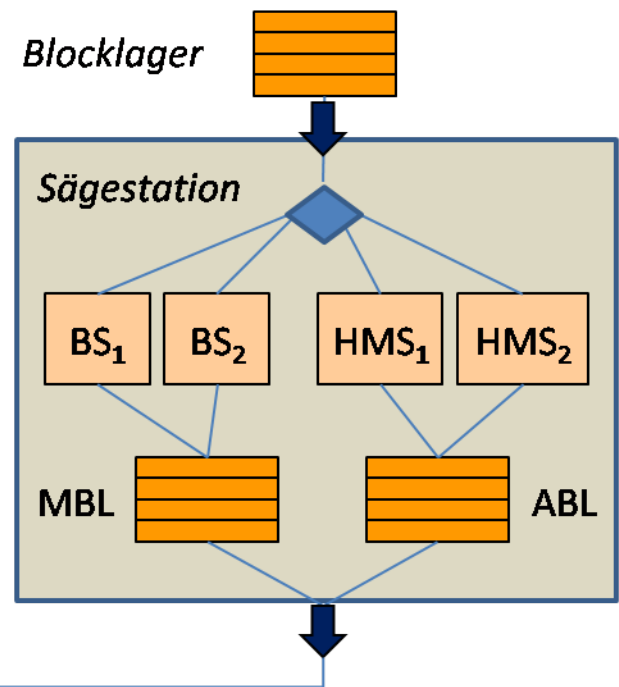
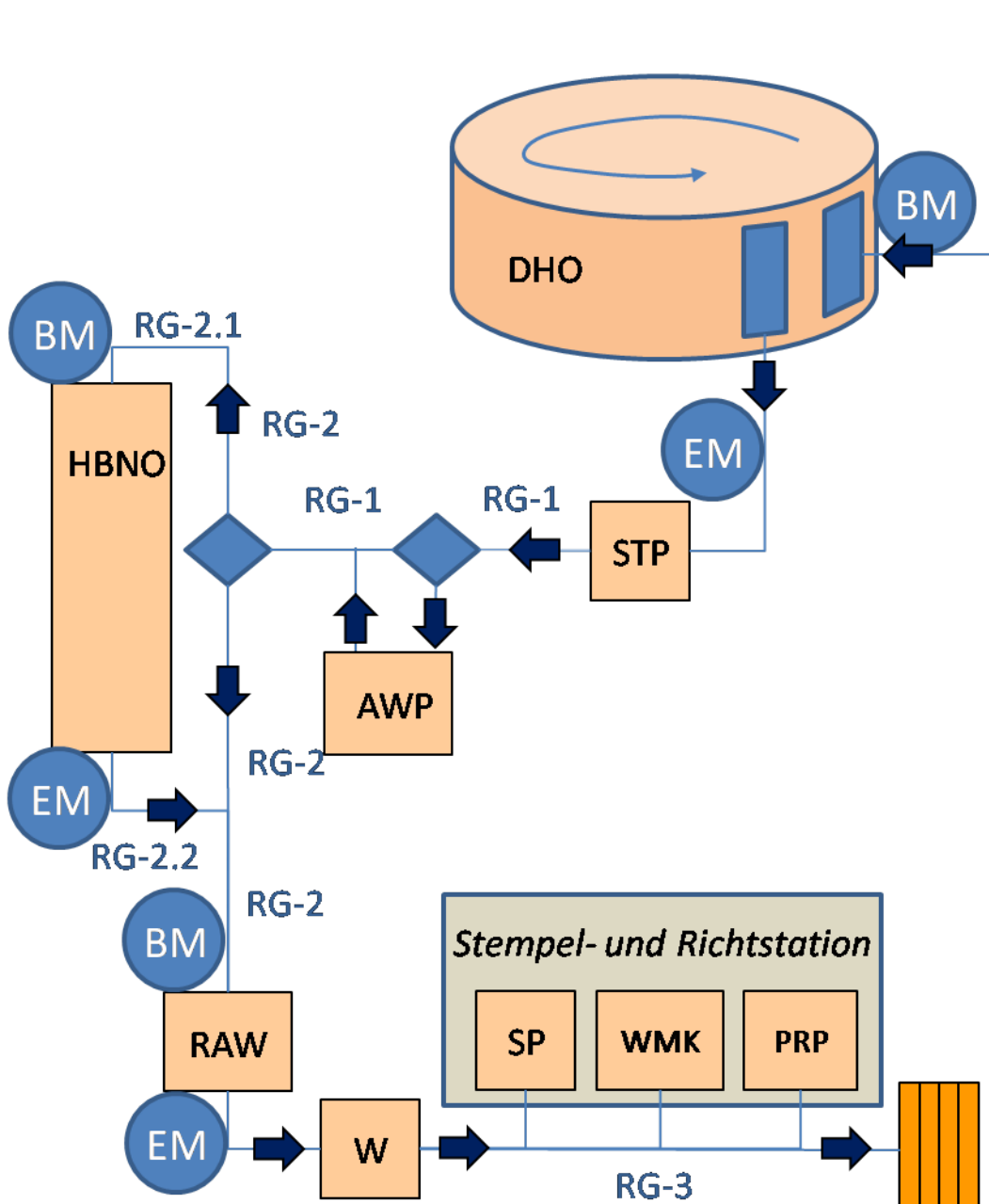


# Hubbalken-Nachwärmofen

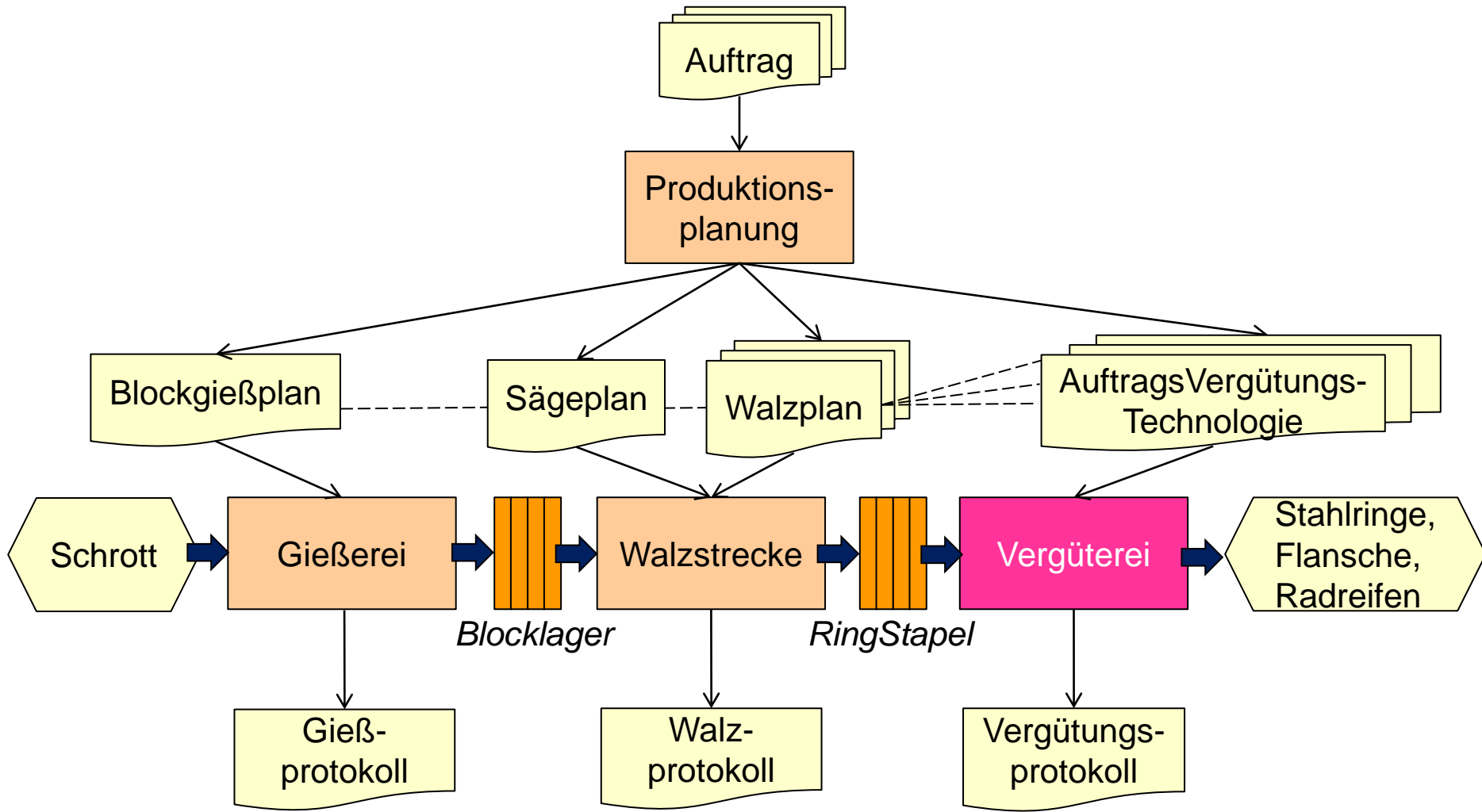


# Radialaxial-Walze

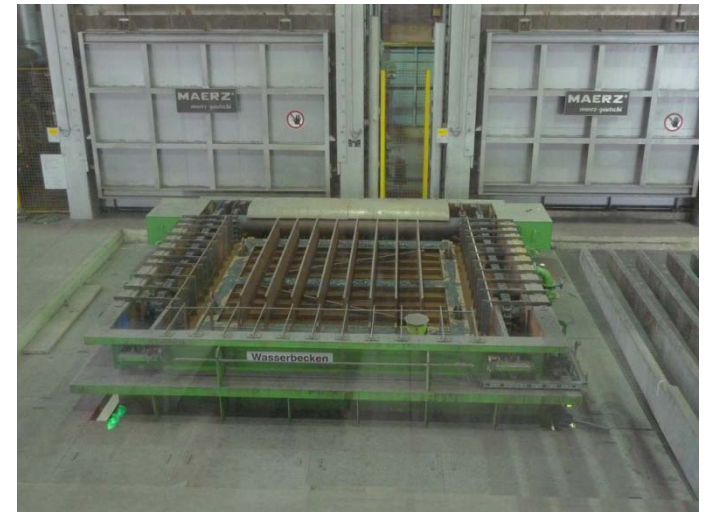
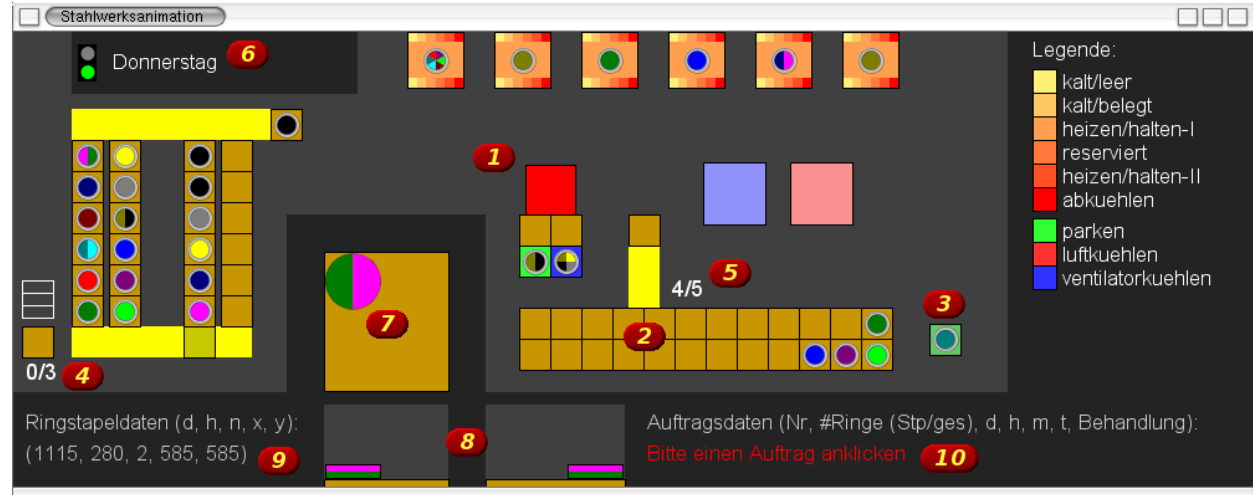




HMS <sub>i</sub>	Hartmetallsäge i (i=1, 2)
BS <sub>i</sub>	Bandsäge i (i= 1, 2)
ABL	automatisiertes Blöckchenlager
MBL	manuelles Blöckchenlager
BM	Belademaschine
EM	Entnahmemaschine
DHO	Dreh-Herdofen
STP	Stauchpresse
HBNO	Hubbalken-Nachwärmofen
AWP	Aufweitpresse
RAW	Radial-Axial-Walze
W	Waage
SP	Stempelpresse
WMK	Wärmmaßkontrolle
PRP	Planrichtpresse
RG-i	Rollgang i

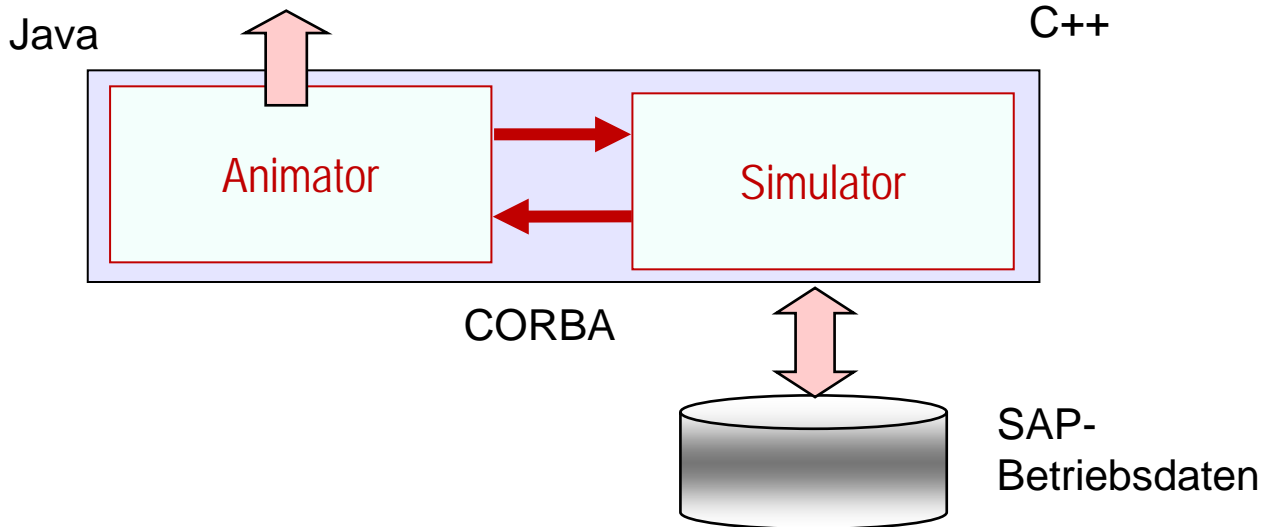
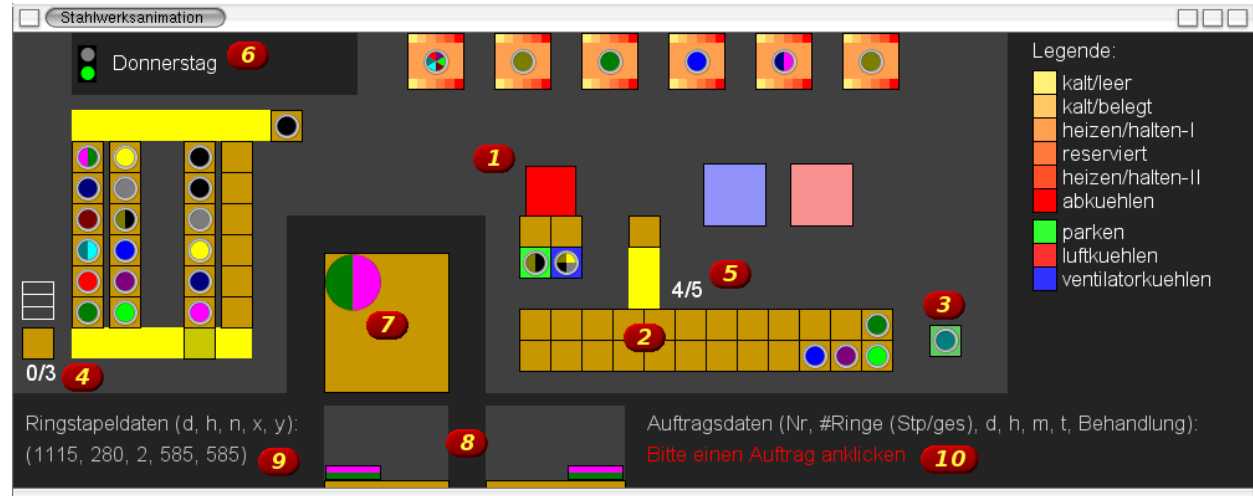


# Die neue Vergüterei

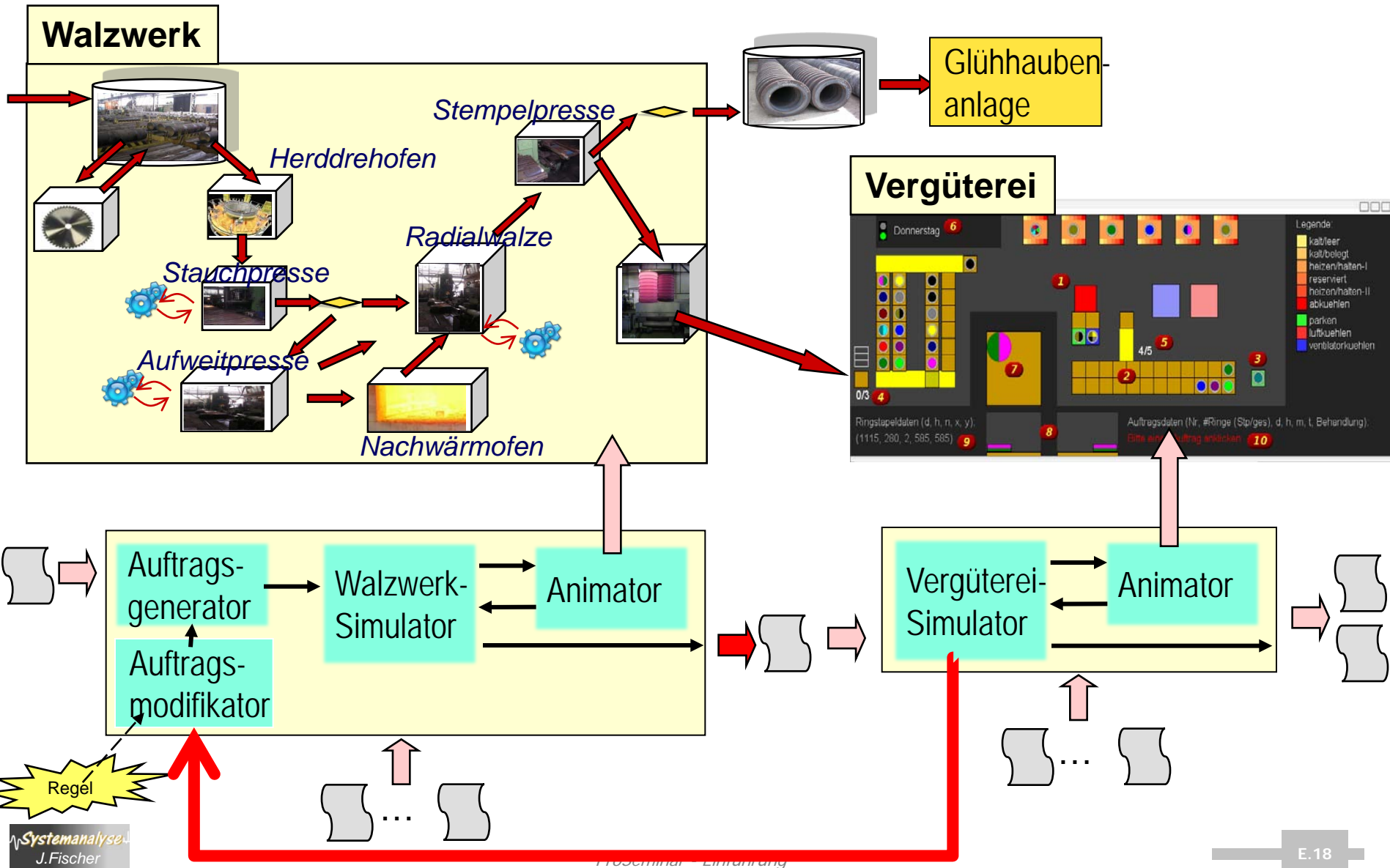




# SimRing- eine spezifische Simulationsumgebung



# Geplante Erweiterung von SimRing



# Was sind Systeme ?

Sind reale oder gedachte Phänomene, deren Zweck / Funktionalität klar erkennbar ist.

Die Bestandteile eines Systems erbringen in ihrer Wechselwirkung untereinander und mit der Umgebung den Zweck des Systems.

Es gehören nur die Elemente zu einem System, die zur Erbringung des identifizierten Zwecks notwendig sind.

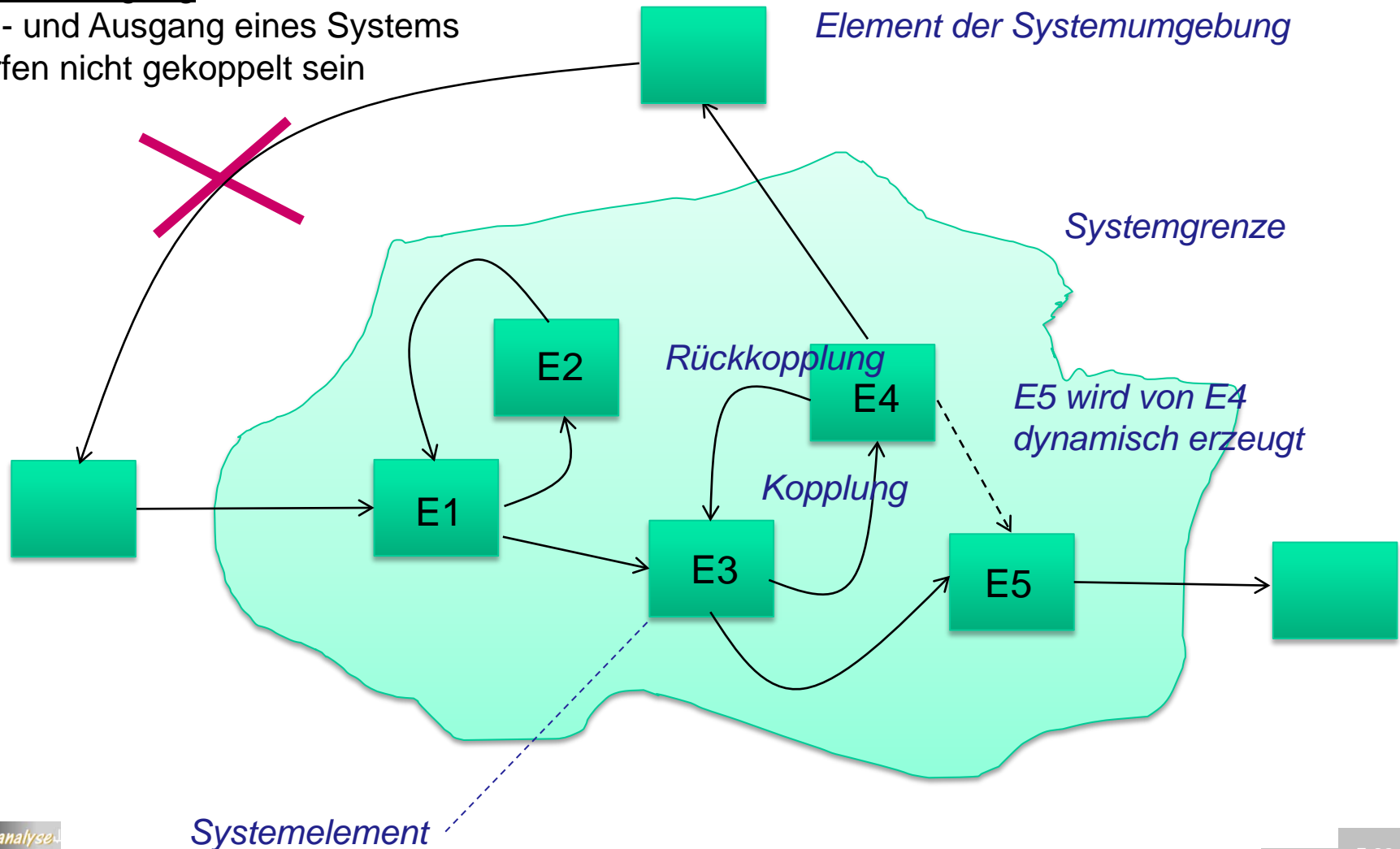
Dynamische Systeme unterliegen zeitlichen Veränderungen, zeigen ein zeitabhängiges Verhalten, reagieren auf Einflüsse der Umgebung

# System-Struktur und Verhalten

Identifikation von Elementen und ihren Abhängigkeiten/Wirkungsbeziehungen

## Randbedingung

Ein- und Ausgang eines Systems dürfen nicht gekoppelt sein



# *Einführung*

1. Organisatorisches
2. Trends in der Software-Entwicklung
3. Modelle in UML
4. Grundsätzliches zu Systemen und Modellen
  - Beispiel eines komplexen dynamischen Systems
  - Objektorientiertes Modellierungsparadigma
  - Klassifikation dynamischer Systeme
  - Beispiel: Niedrigtemperaturofen

# Modellierungsparadigmen

... Objektorientiertes Modellierungsparadigma:  
bestimmt durch spezifische  
Abstraktionsprinzipien

1. Klassifikation /Exemplifikation
2. Instanz: Identität (Referenz),  
Struktur, Verhalten
3. **Unterscheidung:**  
aktive und passive Klassen
4. Beziehungen zwischen Instanzen /  
Instanzmengen
5. Beziehung zwischen Klassen
  - Spezialisierung / Generalisierung
  - abstrakte und konkrete Klassifizierer
6. Polymorphie von (getypten) Referenzen

**zusätzlich benötigte  
Konzepte:**

Gruppierung von  
Modellelementen

Komposition/  
Dekomposition

Nebenläufigkeit/  
Parallelität/  
Synchronisation

zeitdiskretes/  
zeitkontinuierliches  
Verhalten

stochastisches  
Verhalten

# Modellierungsparadigmen von UML/SysML

... Objektorientiertes Modellierungsparadigma:  
bestimmt durch spezifische  
Abstraktionsprinzipien

1. Klassifikation /Exemplifikation
2. Instanz: Identität (Referenz),  
Struktur, Verhalten
3. **Unterscheidung:**  
aktive und passive Klassen
4. Beziehungen zwischen Instanzen /  
Instanzmengen
5. Beziehung zwischen Klassen
  - Spezialisierung / Generalisierung
  - abstrakte und konkrete Klassifizierer
6. Polymorphie von (getypten) Referenzen

zusätzlich benötigte  
Konzepte:

Gruppierung von  
Modellelementen

Komposition/  
Dekomposition

Nebenläufigkeit/  
Parallelität/  
Synchronisation

zeitdiskretes/  
zeitkontinuierliches  
Verhalten

stochastisches  
Verhalten

erst mit SysML

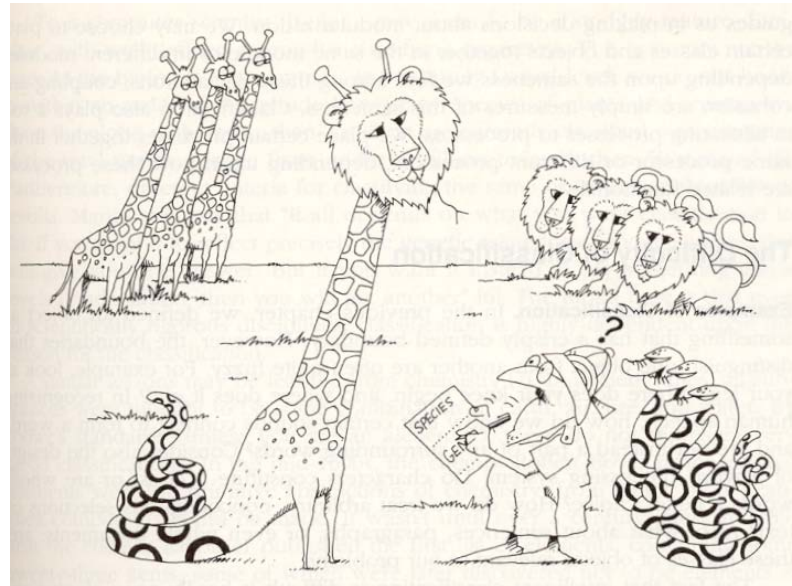
# Klassifikation / Exemplifikation

- in UML:

Classifier

Instanz/Objekt

nach Grady Booch



*Klassifikation von Objekten*

Klassifikation = herausfinden und ordnen von Gemeinsamkeiten

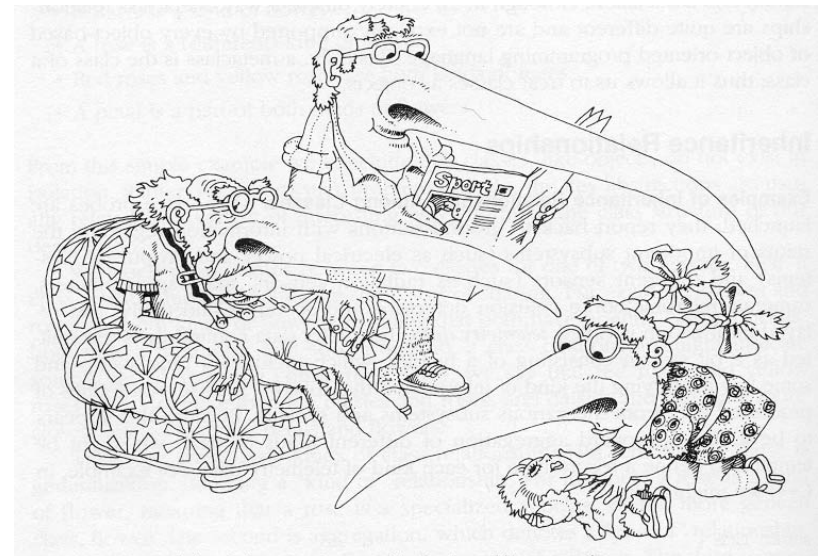


# Beziehungen zwischen Classifiern (z.B. Klassen)

In UML:

Spezialisierung / Verallgemeinerung (Generalisierung)

- viele Objekte haben Gemeinsamkeiten, aber auch Unterschiede bezüglich ihres Verhaltens und ihrer Attribute
- Spezialisierung bedeutet Wiederverwendung der allgemeineren Konzepte
- Generalisierung durch Klassifikation

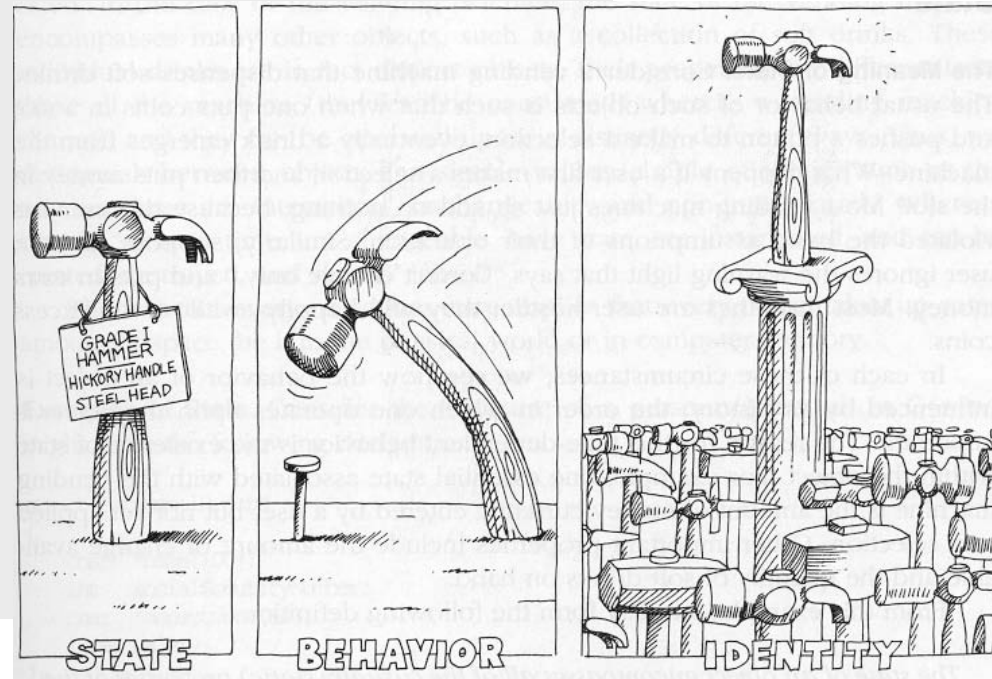


**Grady Booch**

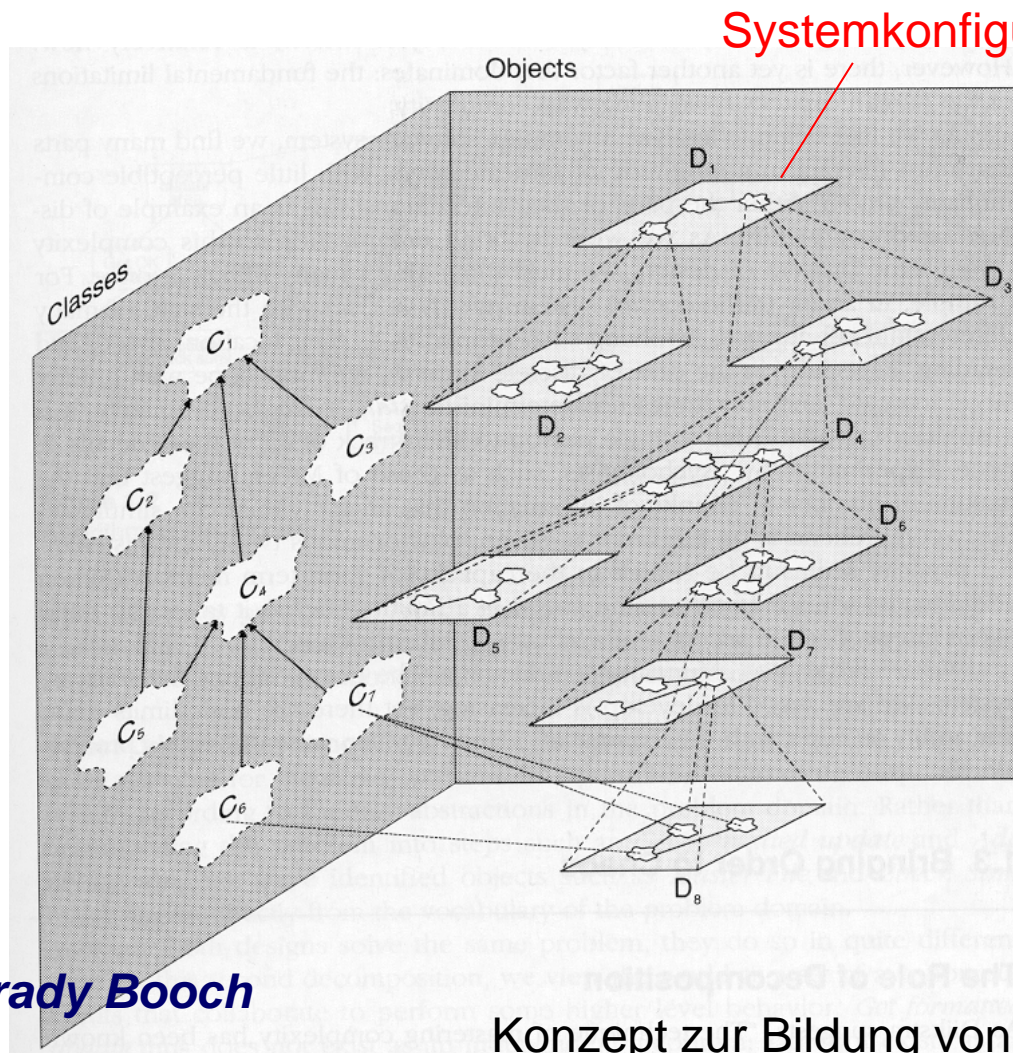
# Instanzen

- jede Instanz einer Klasse ist ein Objekt, das individuell angesprochen und manipuliert werden kann
- jede Instanz hat
  - Identität
  - Zustand  
(Menge der Attributwerte zu einem Zeitpunkt)
  - Verhalten  
in Wechselwirkung mit anderen Instanzen

nach Grady Booch



# Komposition / Dekomposition



Systemkonfiguration

**Klassifikation**

**Spezialisierung**

**Exemplifikation**

**Komposition**

**Wiederverwendung**

**Grady Booch**

Konzept zur Bildung von Instanz/Objekt-Hierarchien

# Instanzen und Verhalten

- **Verhaltensbeschreibung**

- eine Klassifikation von Verhalten basiert auf **drei Grundprinzipien**:

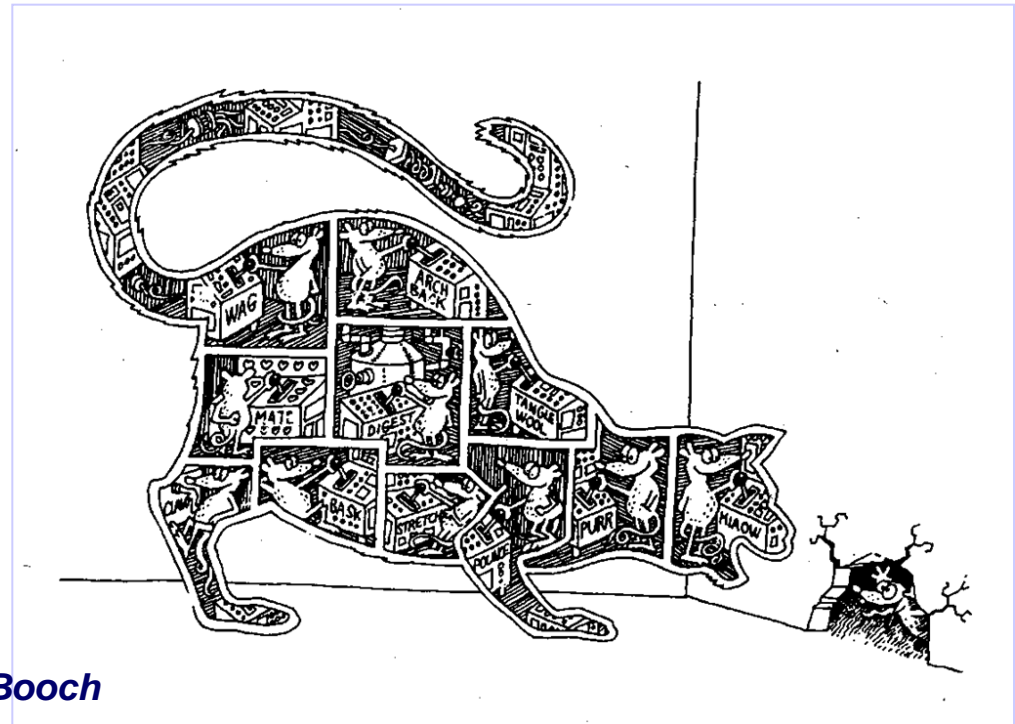
- Ursache – Wirkung
- zeitlicher Ablauf (Änderung von Größen in Abhängigkeit der Zeit)
- Funktionale Ähnlichkeit (Verhaltensanalogie)

- zu berücksichtigende Aspekte:

- Parallelität
- Nebenläufigkeit
- Kommunikation

- Unterscheidung

- aktive
- passive Klassen



*nach Grady Booch*

# *Einführung*

1. Organisatorisches
2. Trends in der Software-Entwicklung
3. Modelle in UML
4. Grundsätzliches zu Systemen und Modellen
  - Beispiel eines komplexen dynamischen Systems
  - Objektorientiertes Modellierungsparadigma
  - **Klassifikation dynamischer Systeme**
  - Beispiel: Niedrigtemperaturofen

# 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

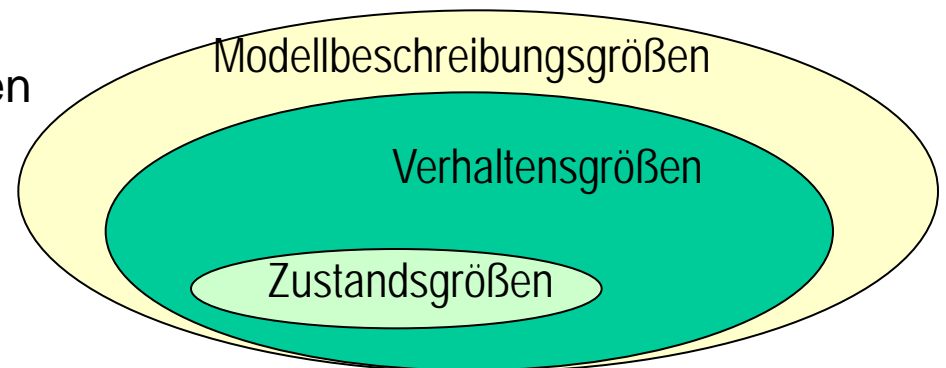
## essentielle Verhaltensgrößen

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

# Zustandsgrößen

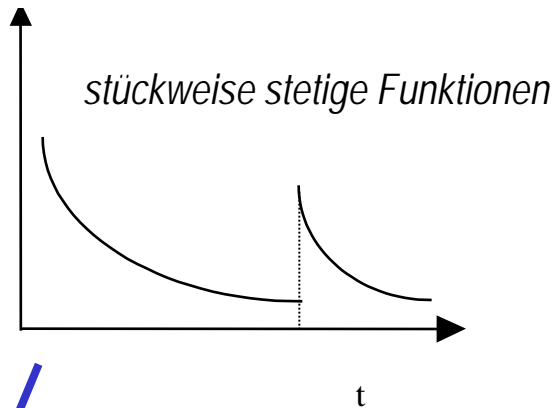
... sind

- 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



# Arten von Zustandsänderungen

zeitkontinuierliche  
Zustandsänderung



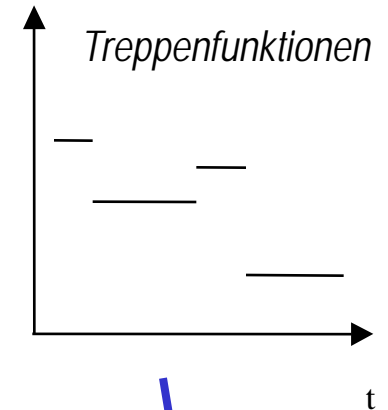
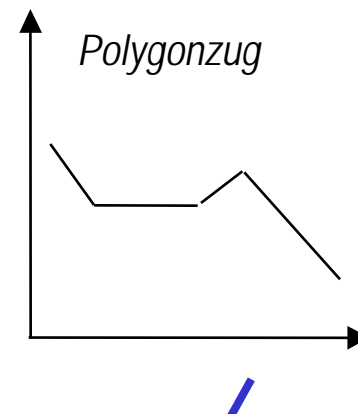
$$z'(t) = f(z(t), x(t), t)$$

mit  $z(t) \in Z, x(t) \in X, t \in T$

Differentialgleichungen

numerische  
Lösungsverfahren

zeitdiskrete  
Zustandsänderung



$$z(t_{n+1}) = f(z(t_n), x(t_{n+1}), t_{n+1})$$

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

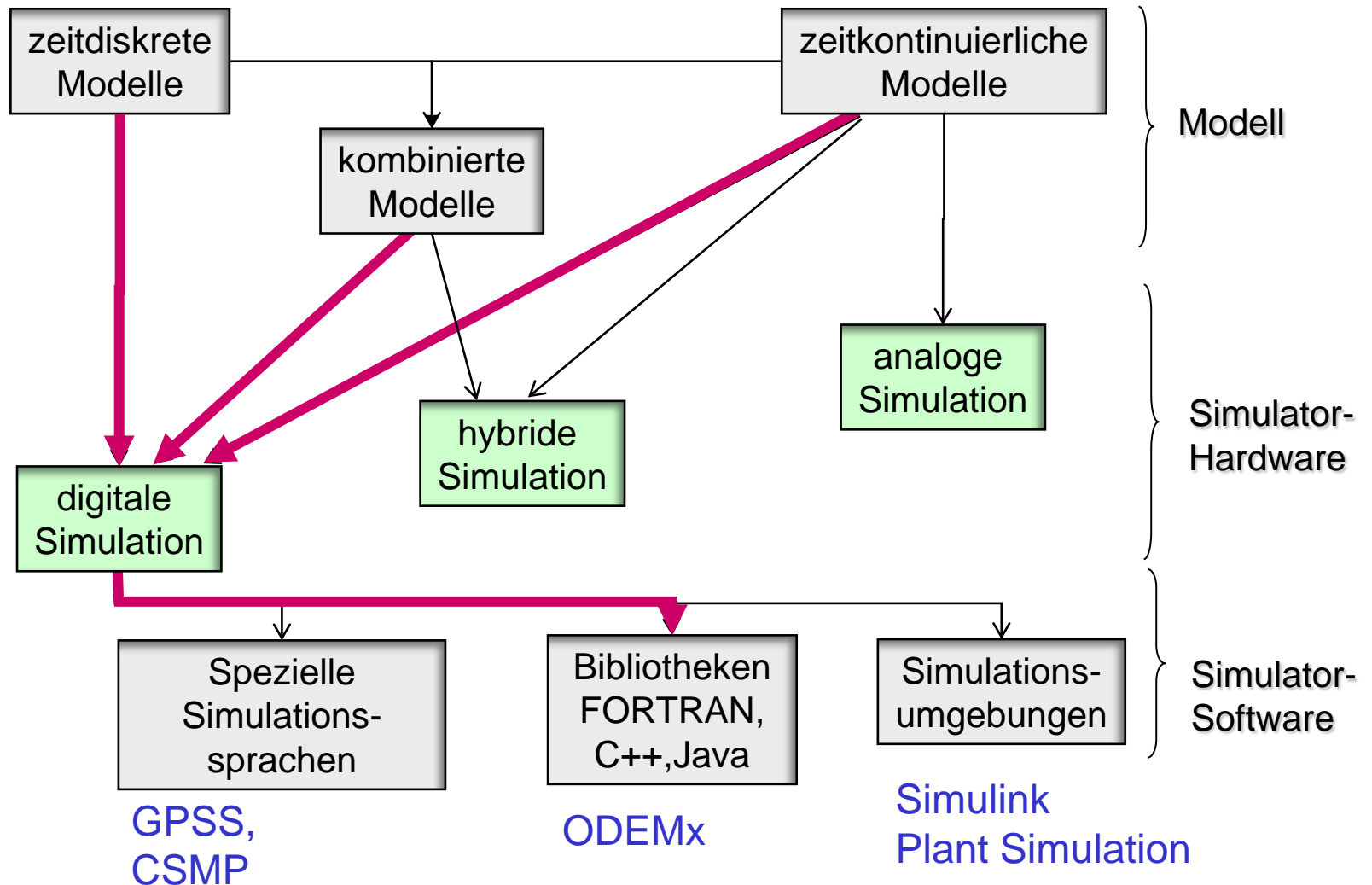
Differenzgleichungen  
zelluläre Automaten

Ereignissimulationen

Prozesssimulation



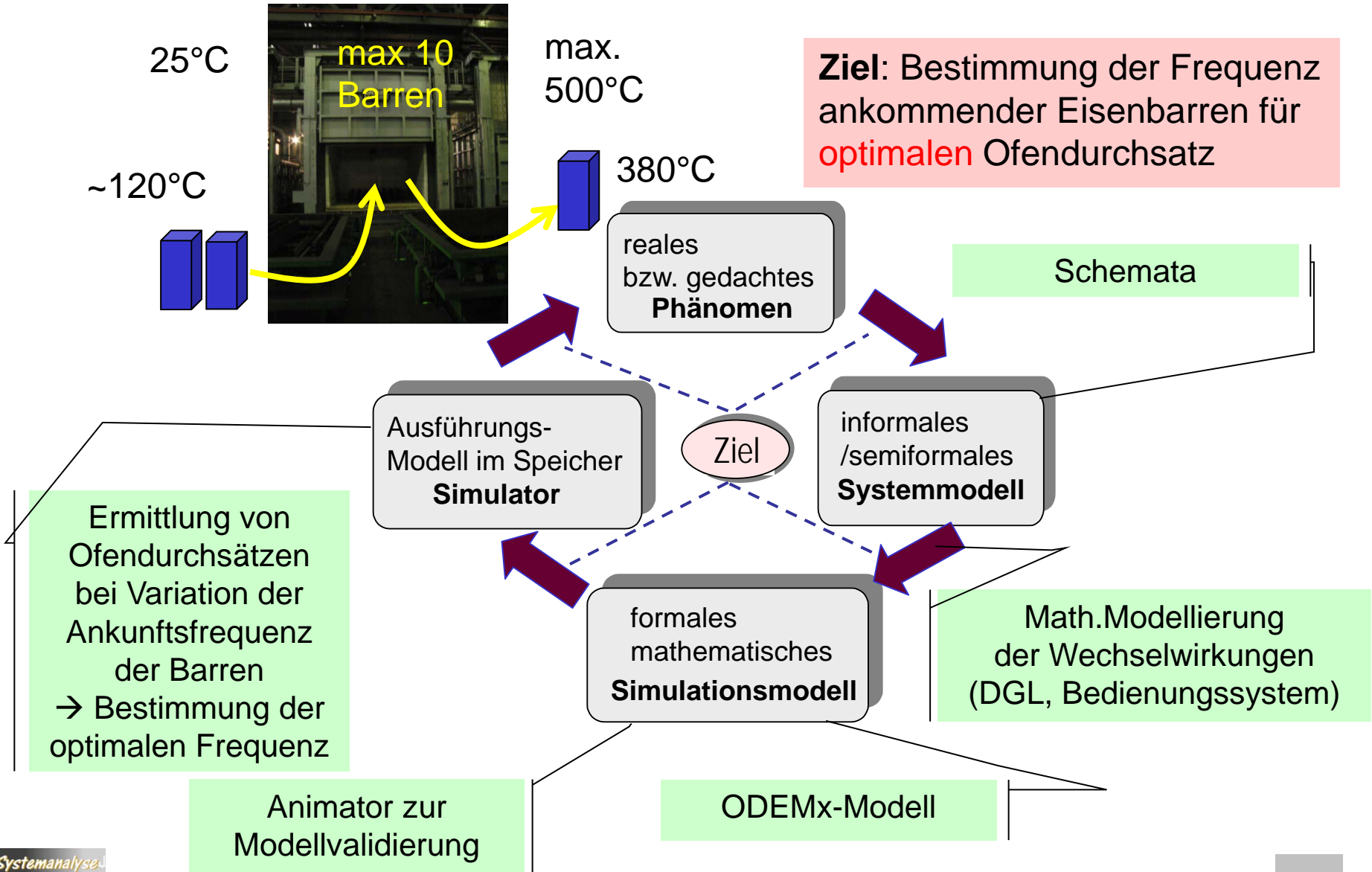
# Klassifizierung von Modellen und Simulationsverfahren



# *Einführung*

1. Organisatorisches
  2. Trends in der Software-Entwicklung
  3. Modelle in UML
  4. Grundsätzliches zu Systemen und Modellen
    - Beispiel eines komplexen dynamischen Systems
    - Objektorientiertes Modellierungsparadigma
    - Klassifikation dynamischer Systeme
- Beispiel: Niedrigtemperaturofen

# Beschickung eines Niedrigtemperaturofens



# 1. Schritt: Problemanalyse

## - allgemein -

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

### bei Identifikation von

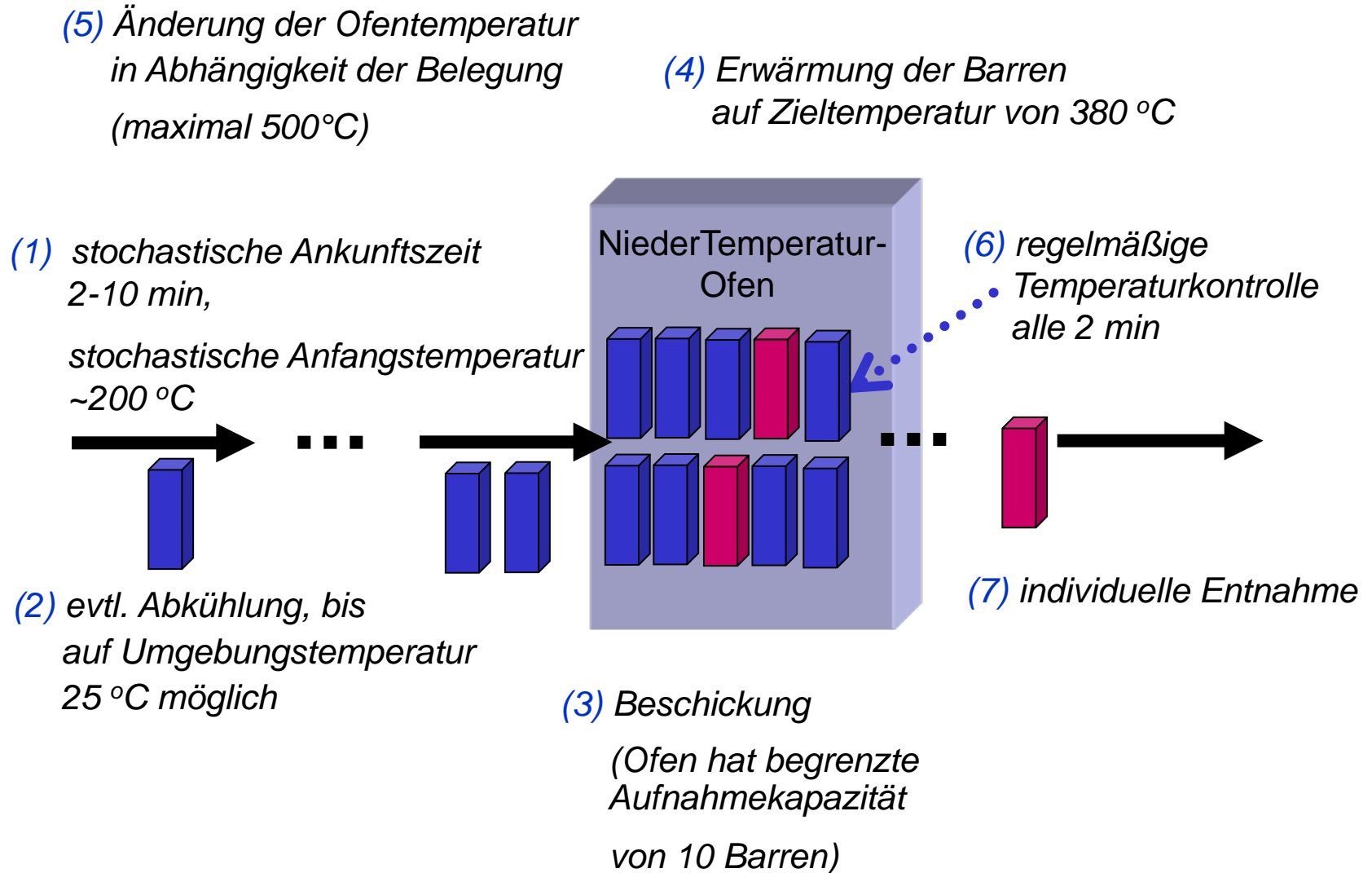
- Systemelementen und Systemumgebung
- Relationen (Wechselwirkungen) zwischen Systemelementen untereinander und zur Umgebung

zur Erbringung des bereits bestimmten

- Systemzwecks / Untersuchungsziels

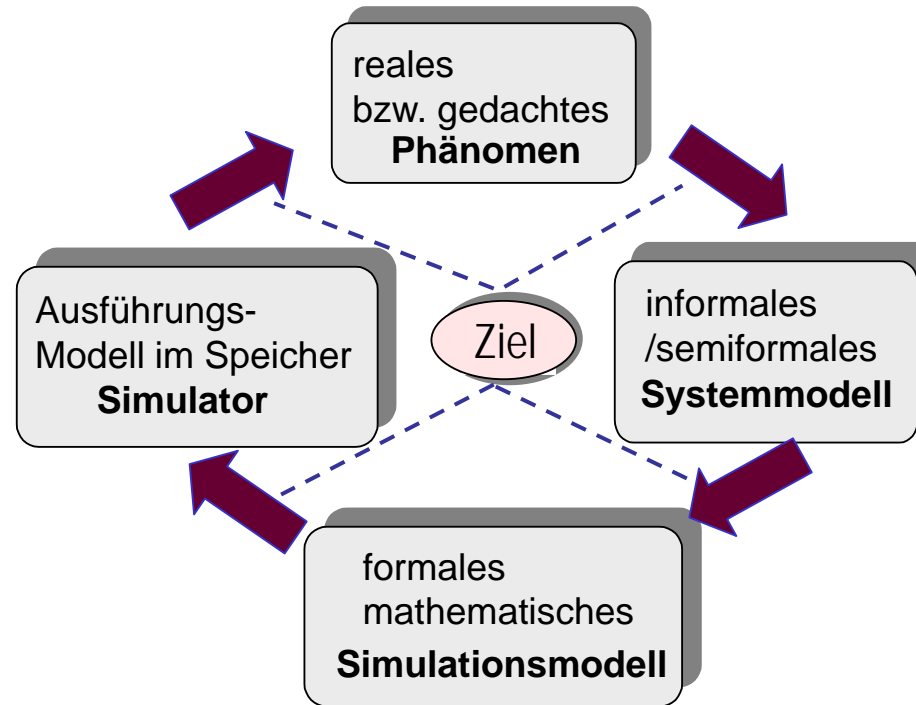
# 1. Schritt: Problemanalyse

## - 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

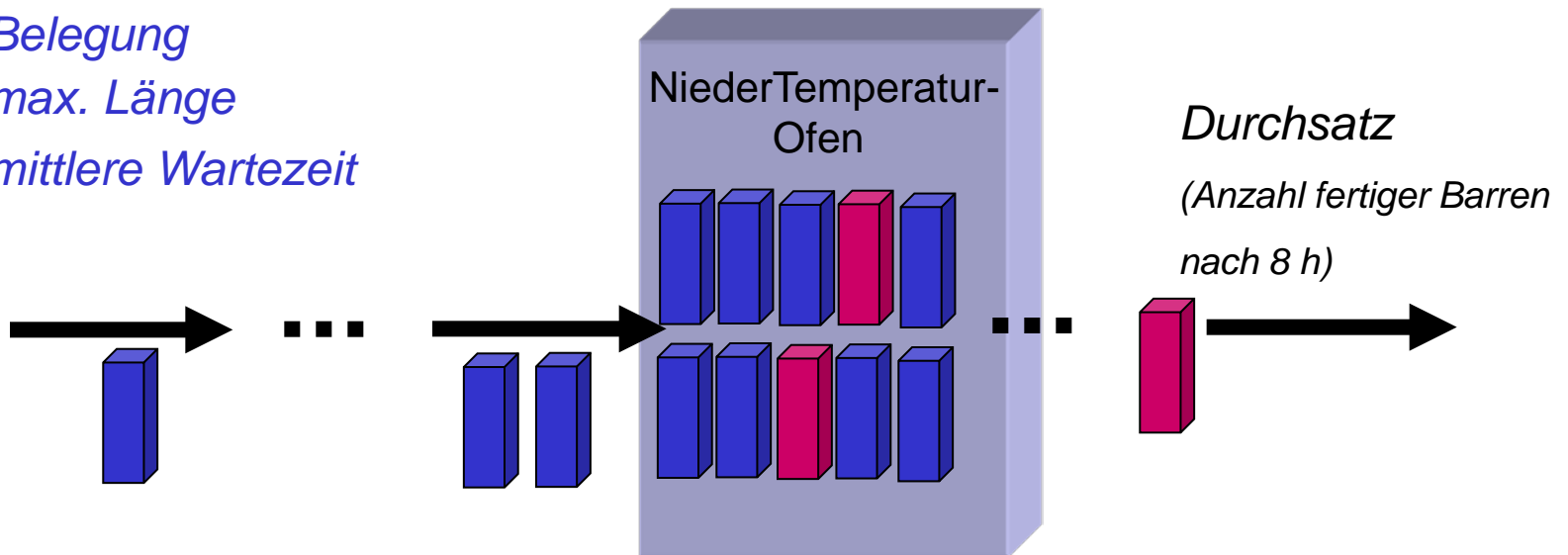
# 1. Schritt: Problemanalyse

## - 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

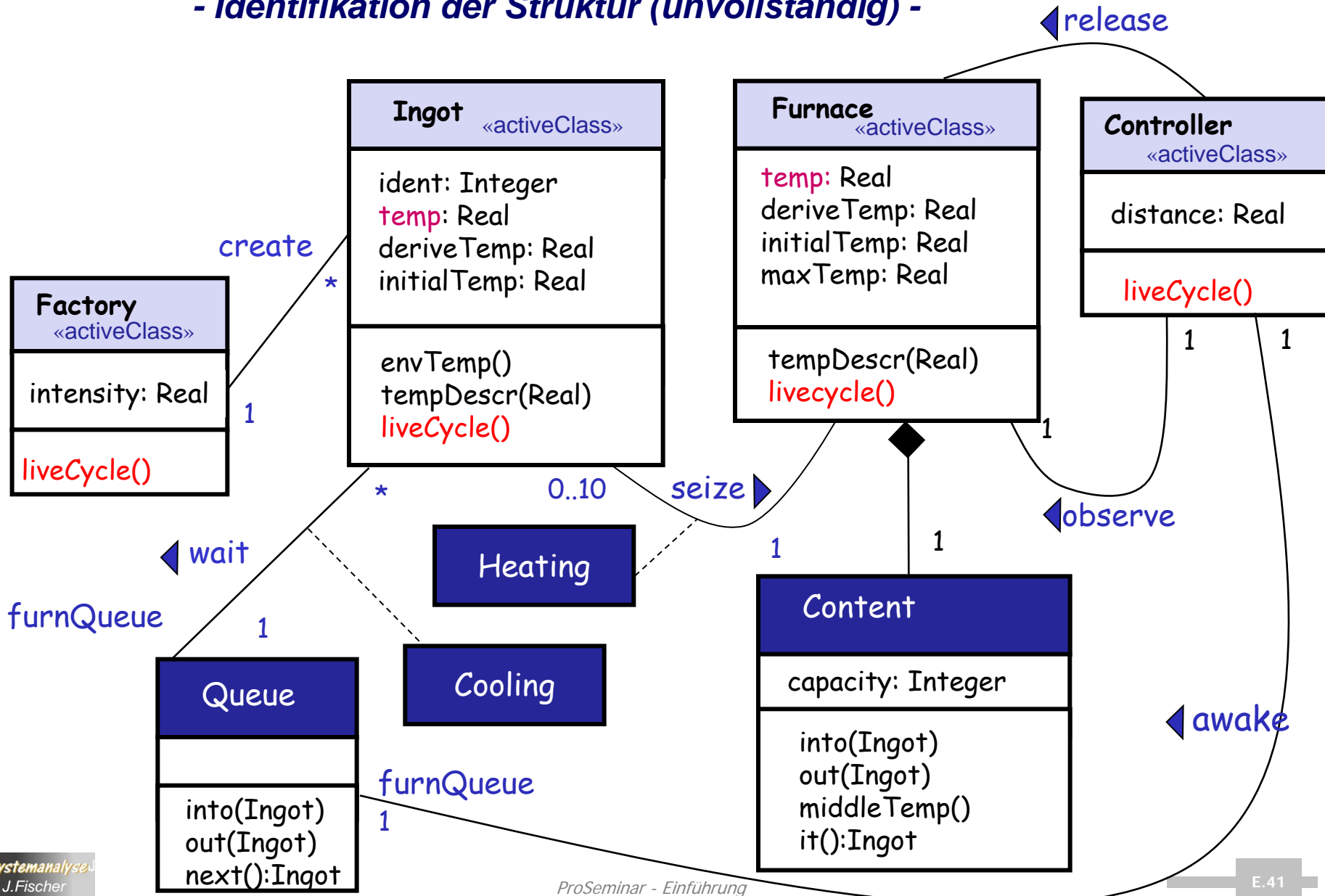
bei Bestimmung

- der Verhaltensfunktionen/-Gleichungen



## 2. Schritt: Modellformalisierung

### - Identifikation der Struktur (unvollständig) -



## 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



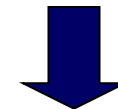
Beschreibung als

**Gewöhnliche  
Differentialgleichungen** (AWA)

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

**zeitdiskrete** Vorgänge:

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



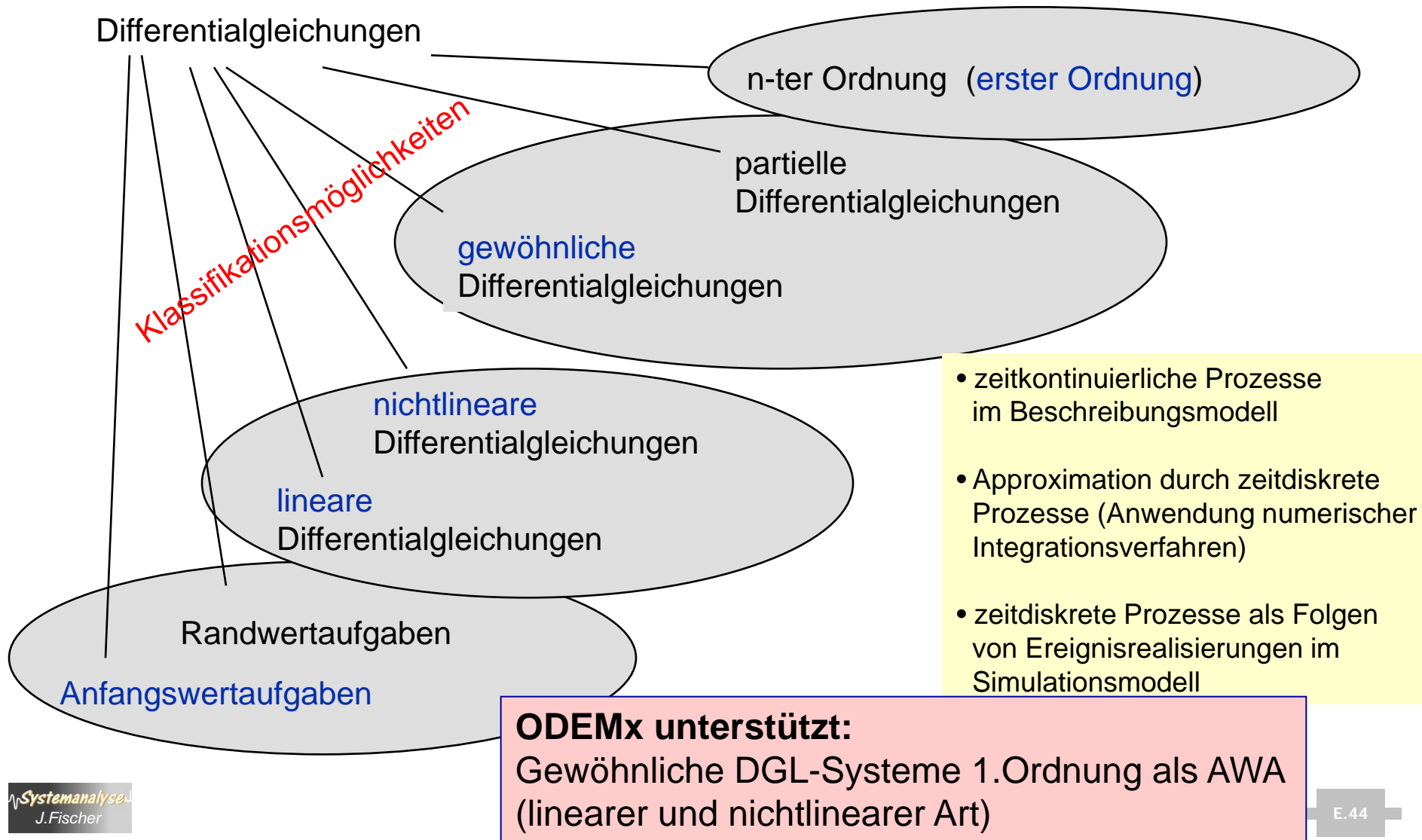
Beschreibung als

überlagerte Folgen von Ereignissen

- Bedienungsvorgänge,
- Steuerungseingriffe
- Kommunikationen

## 2. Schritt: Modellformalisierung

### - Identifikation zeitkontinuierlicher Verhaltensklassen -



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

Modellierung der Abkühlungsvorgänge (Barren)

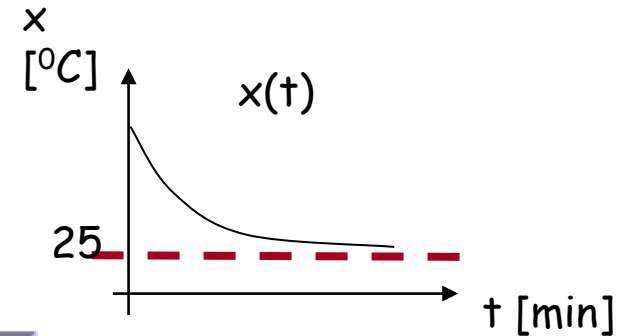
Ingot.deriveTemp

Ingot.temp

Barrentemperatur

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

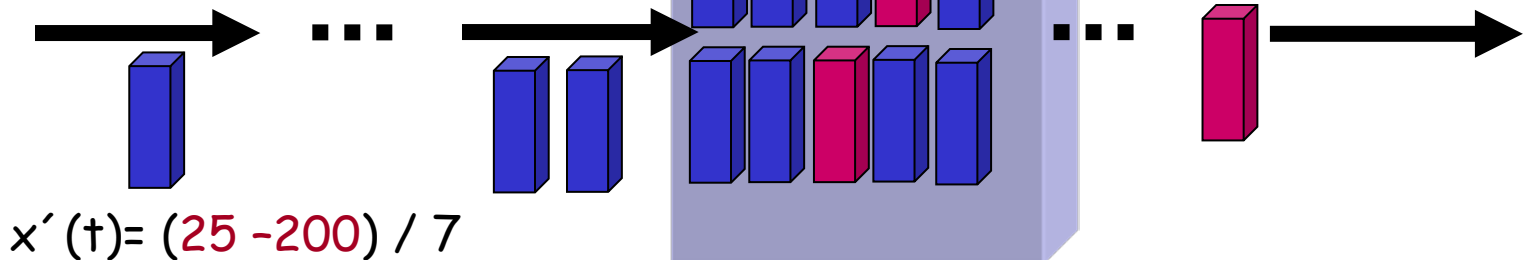
Anfangstemperatur: ca. 200 °C



Umgebungstemperatur

$$u(t) = 25$$

Ingot.envTemp()



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

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

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

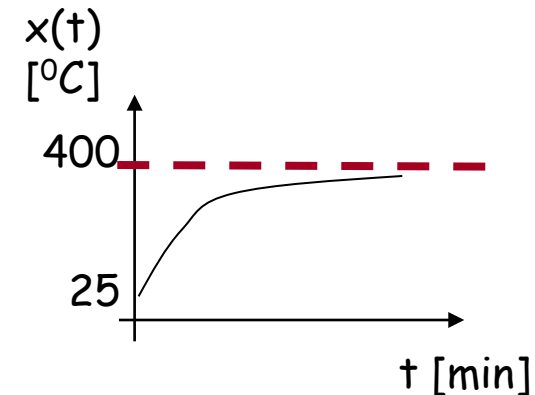
Barrentemperatur

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

Eintrittstemperatur:  
200 °C .. 25 °C

Ofentemperatur  $y(t)$   
~400 °C

NiederTemperatur-Ofen



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

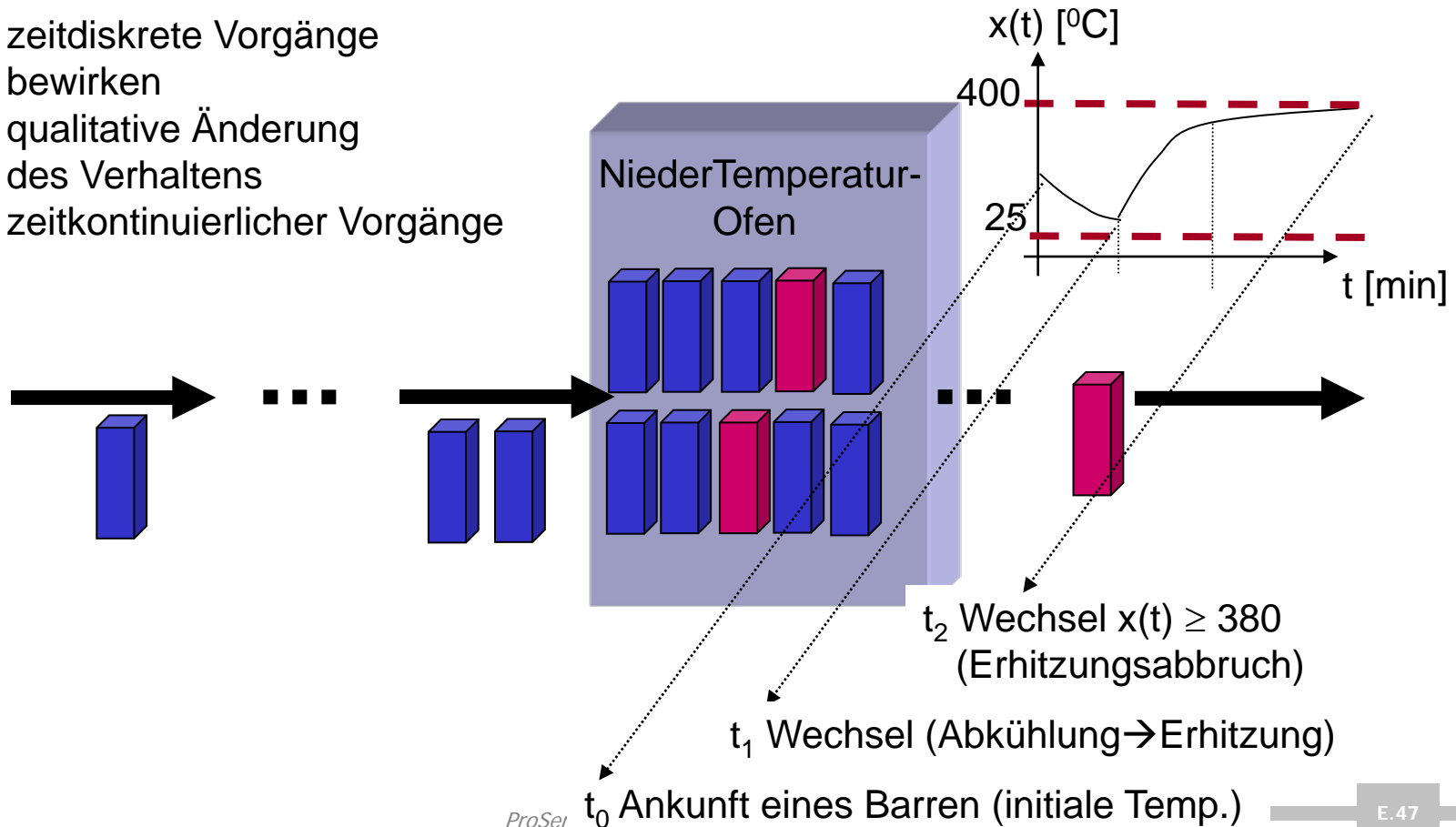
positiver Wert für  $x'(t)$ , d.h. Zunahme von  $x(t)$   
Änderung von  $x(t)$ :

- 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 -

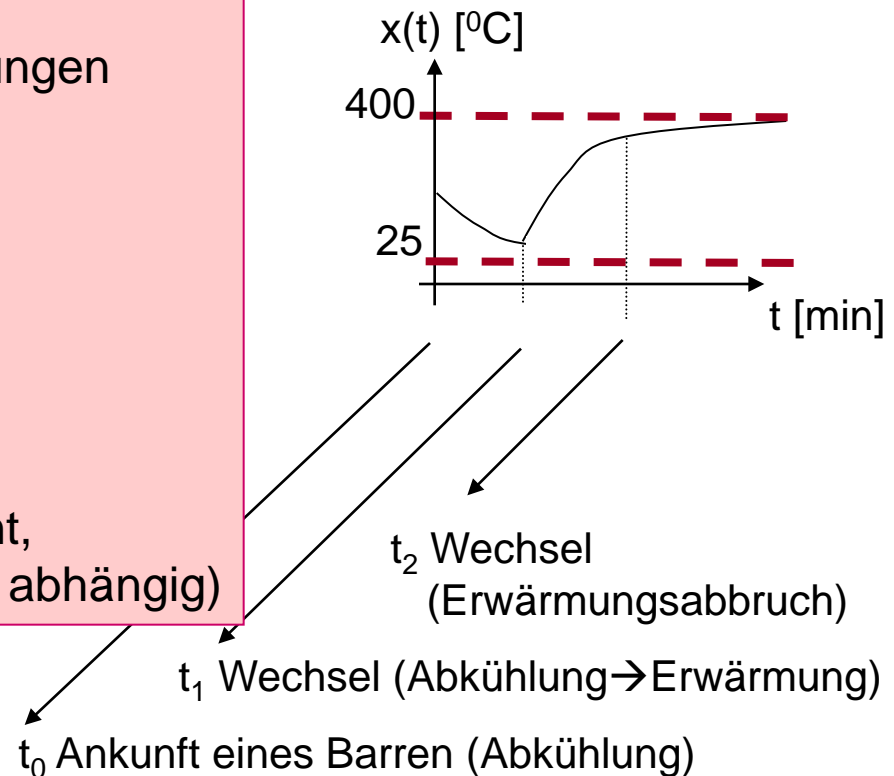
#### Modellierung von (zeitdiskreten) Ereignissen

##### Ereignismodellierung

Erfassung von Modellgrößenänderungen zu einem festen Zeitpunkt

##### Ereignisklassen

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





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

### Modellierung der Änderung der Ofentemperatur

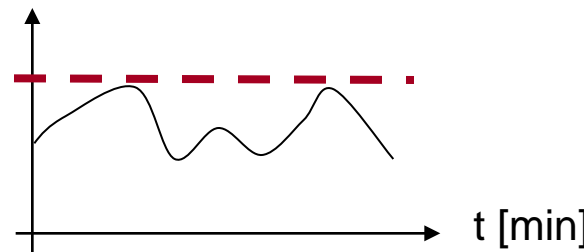
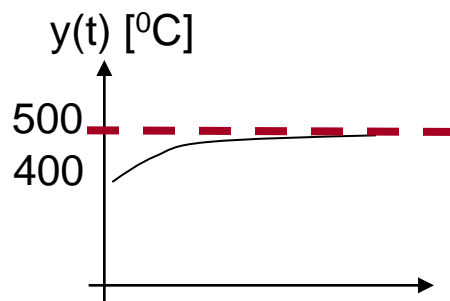
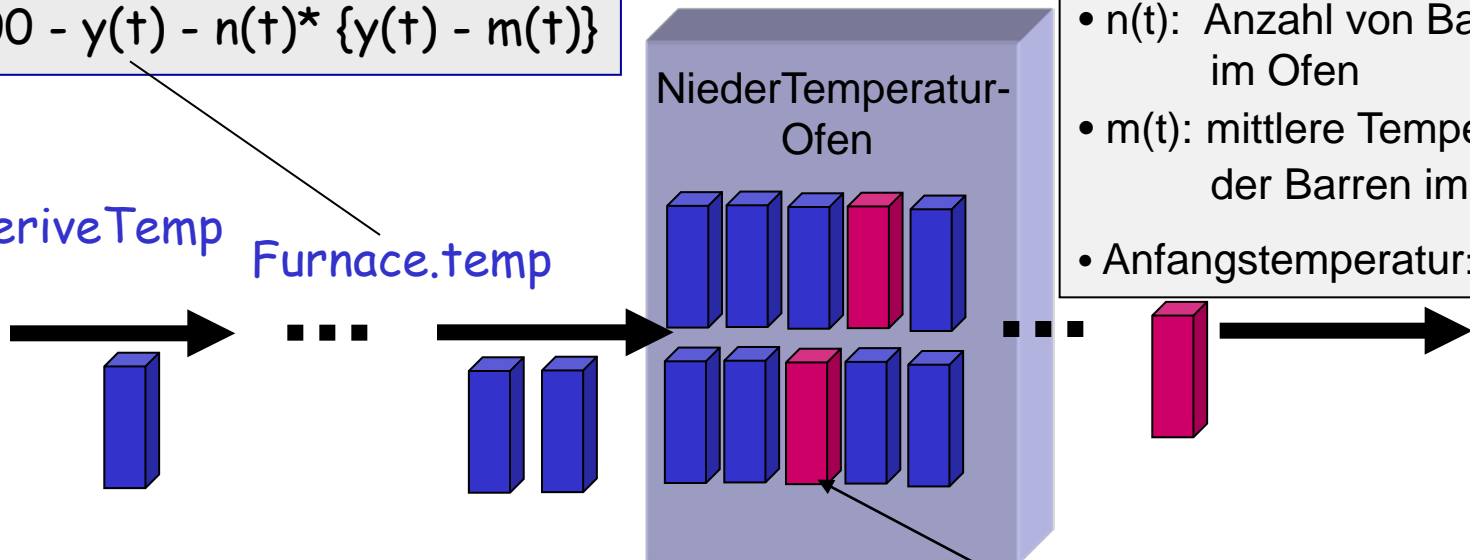
Ofentemperatur  $y$  als  $y(t)$

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

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

Furnace.deriveTemp

Furnace.temp



Zieltemperatur

$x(t) = 380$  °C  
per Beobachtung

### **3. Schritt: Softwaretechnische Umsetzung** **- allgemein -**

#### **Ziel: ausführbares Simulationsmodell**

als Umsetzung der formalen Struktur- (Objekt- und Klassenstruktur) und der Verhaltensmodellgleichungen

bei Anwendung der Simulationsbibliothek ODEMX:  
vordefinierte Bausteine zur Modellierung

- zeitdiskreter und
- zeitkontinuierlicher Prozesse

# 3. Schritt: Softwaretechnische Umsetzung

- bei Nutzung von ODEMx -

