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. Bespiele aus der aktuellen Forschung
6. Paradigma der objektorientierten Modellierung
7. Klassifikation dynamischer Systeme
8. Konzept der diskreten Ereignissimulation
9. M&S eines Niedertemperaturofens

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

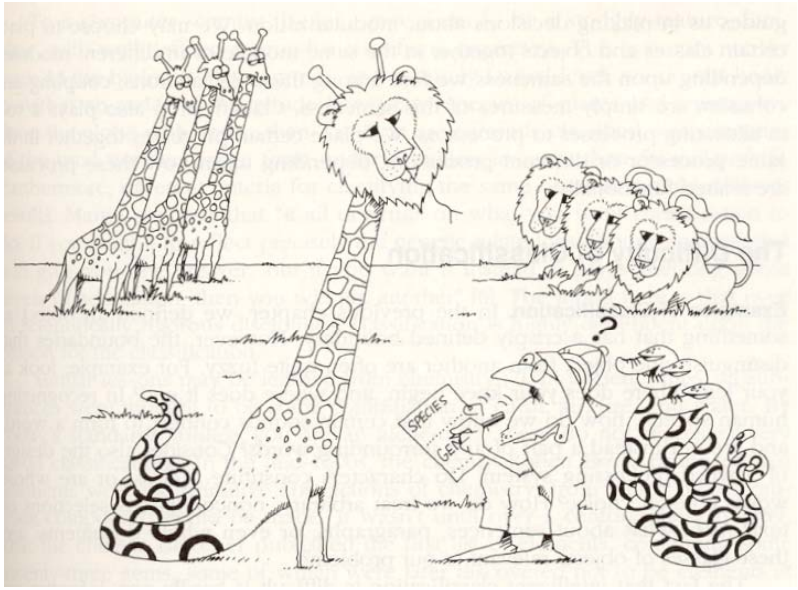
Klassifikation / Exemplifikation

- in UML:

Classifier

Instanz/Objekt

nach Grady Booch



Klassifikation von Objekten

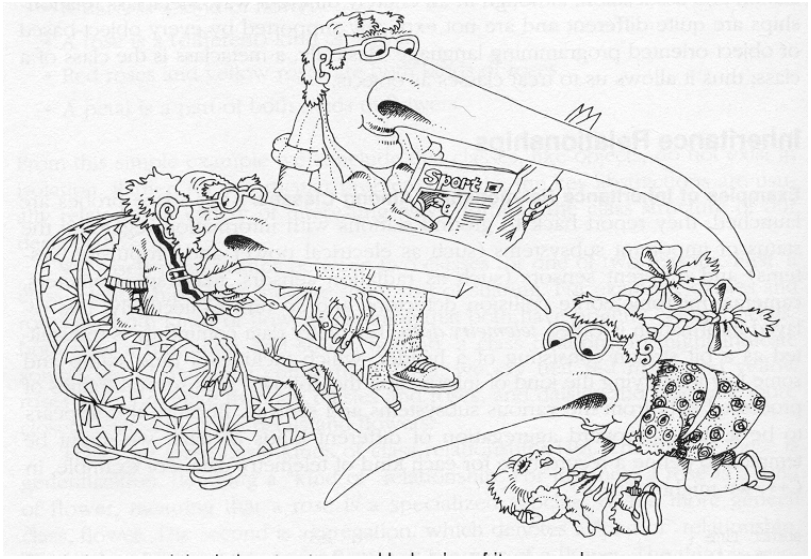
Klassifikation = herausfinden und ordnen von Gemeinsamkeiten

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



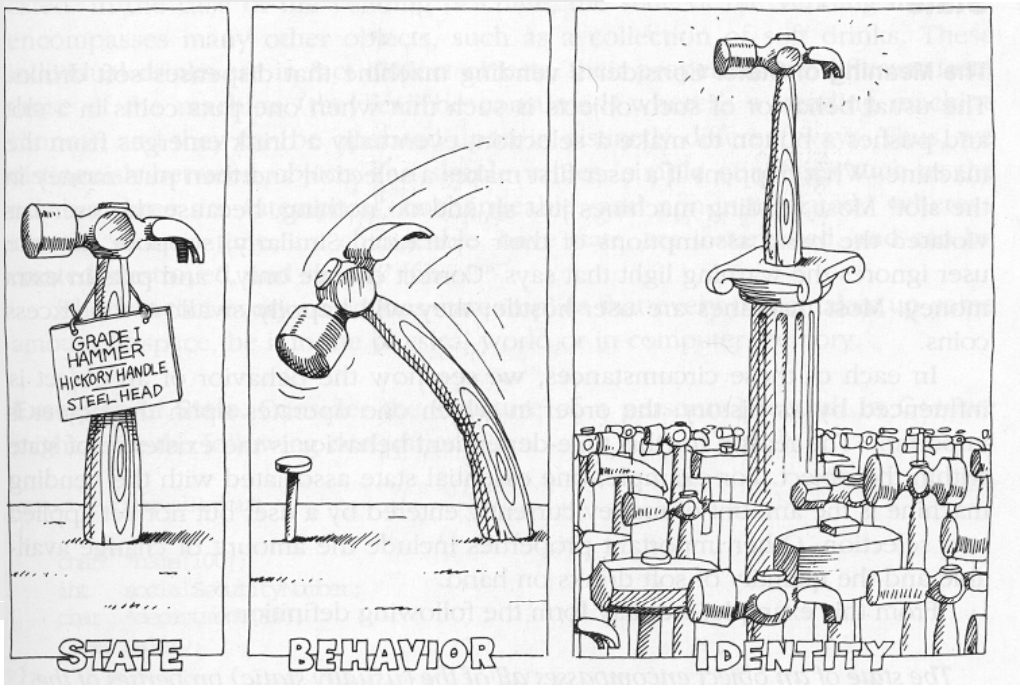
Grady Booch

Instanzen

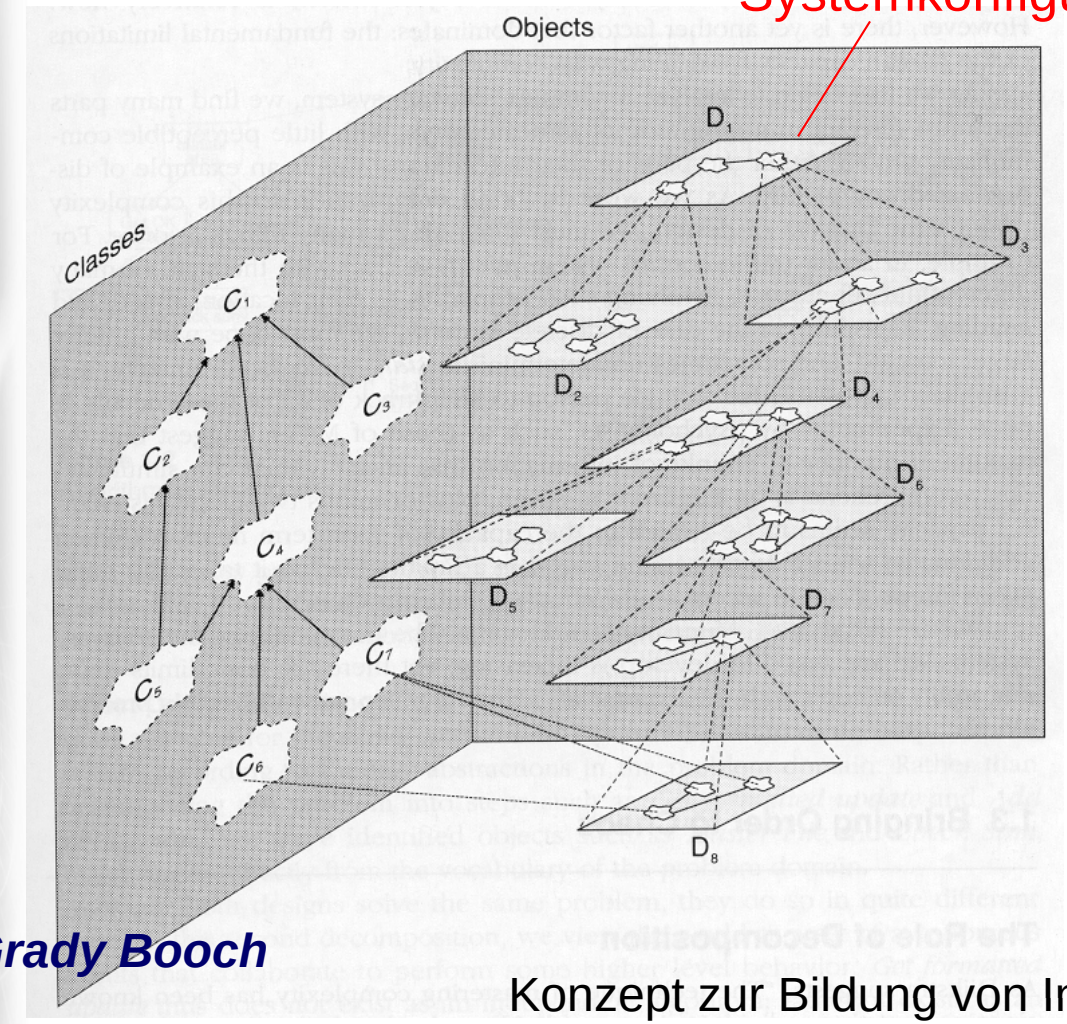
- jede Instanz einer Klasse ist ein Objekt, das individuell angesprochen und manipuliert werden kann

- jede Instanz hat
  - Identität
  - Zustand
    (Menge der Attributwerte zu einem Zeitpunkt)
  - Verhalten
    in Wechselwirkung mit anderen Instanzen

nach Grady Booch



Komposition / Dekomposition



Klassifikation

Spezialisierung

Exemplifikation

Komposition

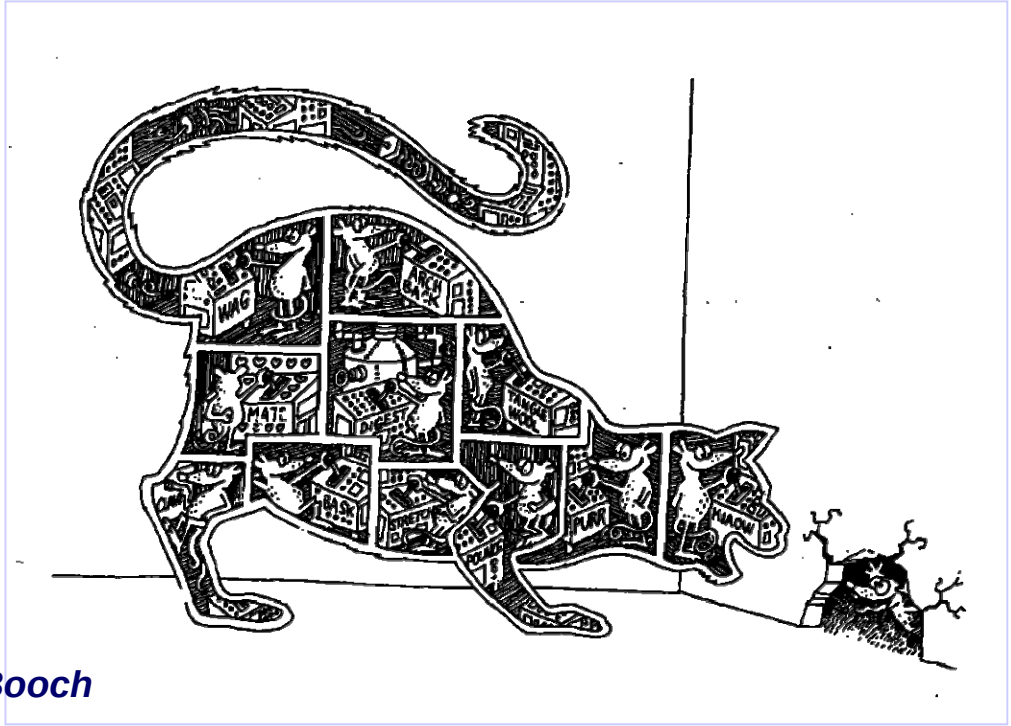
Wiederverwendung

Grady Booch

Konzept zur Bildung von Instanz/Objekt-Hierarchien

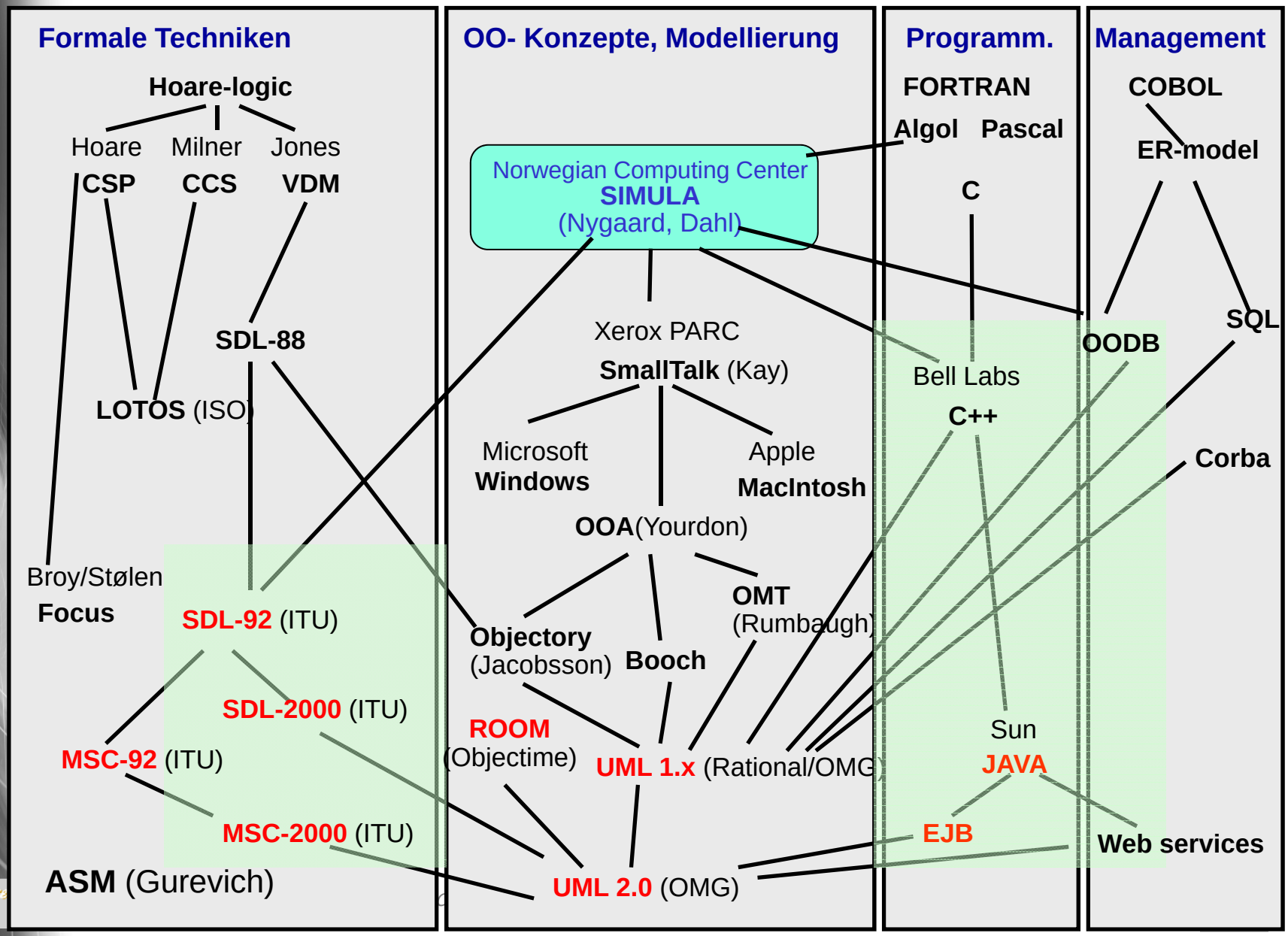
Instanzen und Verhalten

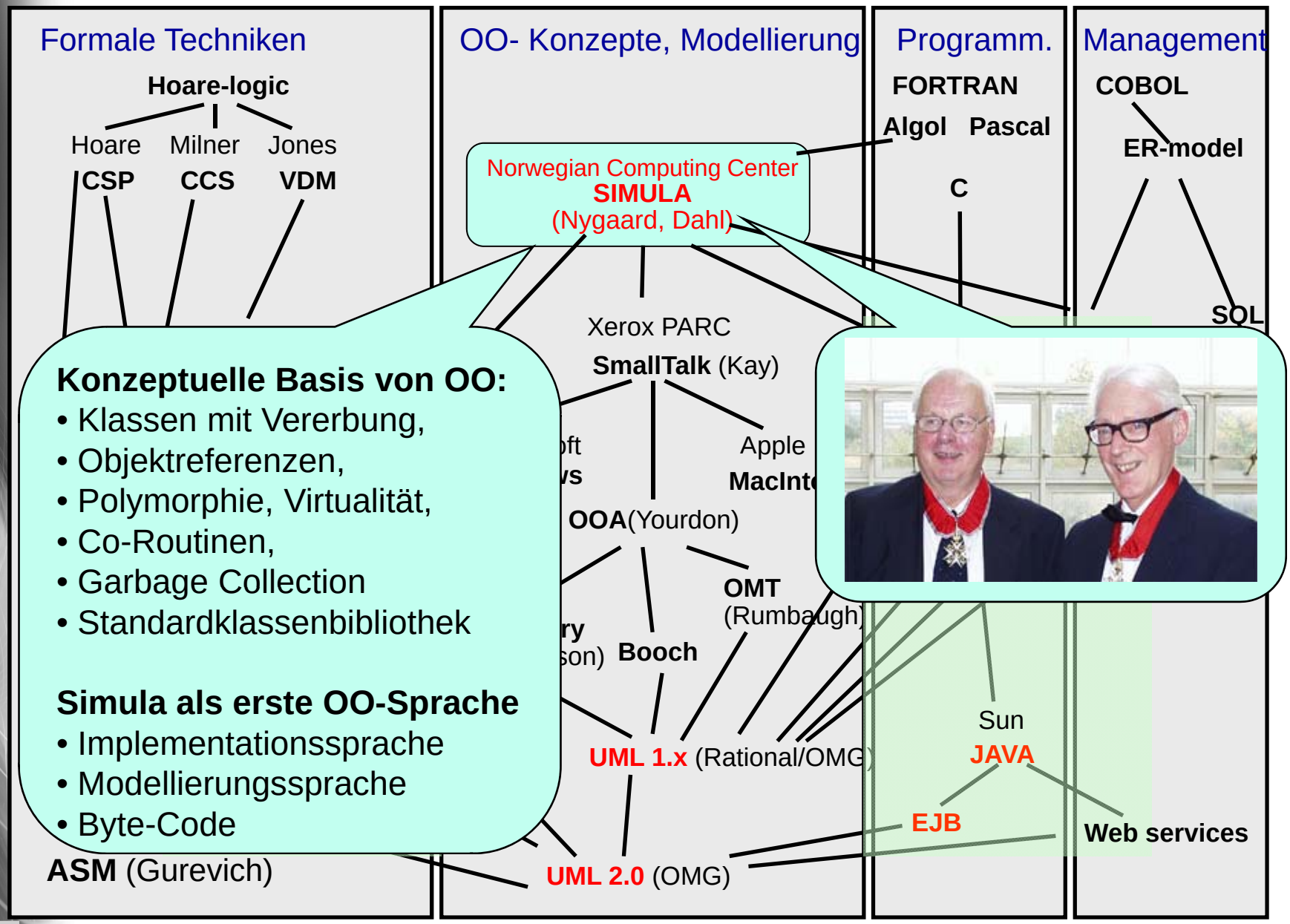
- **Verhaltensbeschreibung**
  - eine Klassifikation von Verhalten basiert auf drei Grundprinzipien:
    - Ursache – Wirkung
    - zeitlicher Ablauf (Änderung von Größen in Abhängigkeit der Zeit)
    - Funktionale Ähnlichkeit (Verhaltensanalogie)
  - zu berücksichtigende Aspekte:
    - Parallelität
    - Nebenläufigkeit
    - Kommunikation
  - Unterscheidung
    - aktive
    - passive Klassen

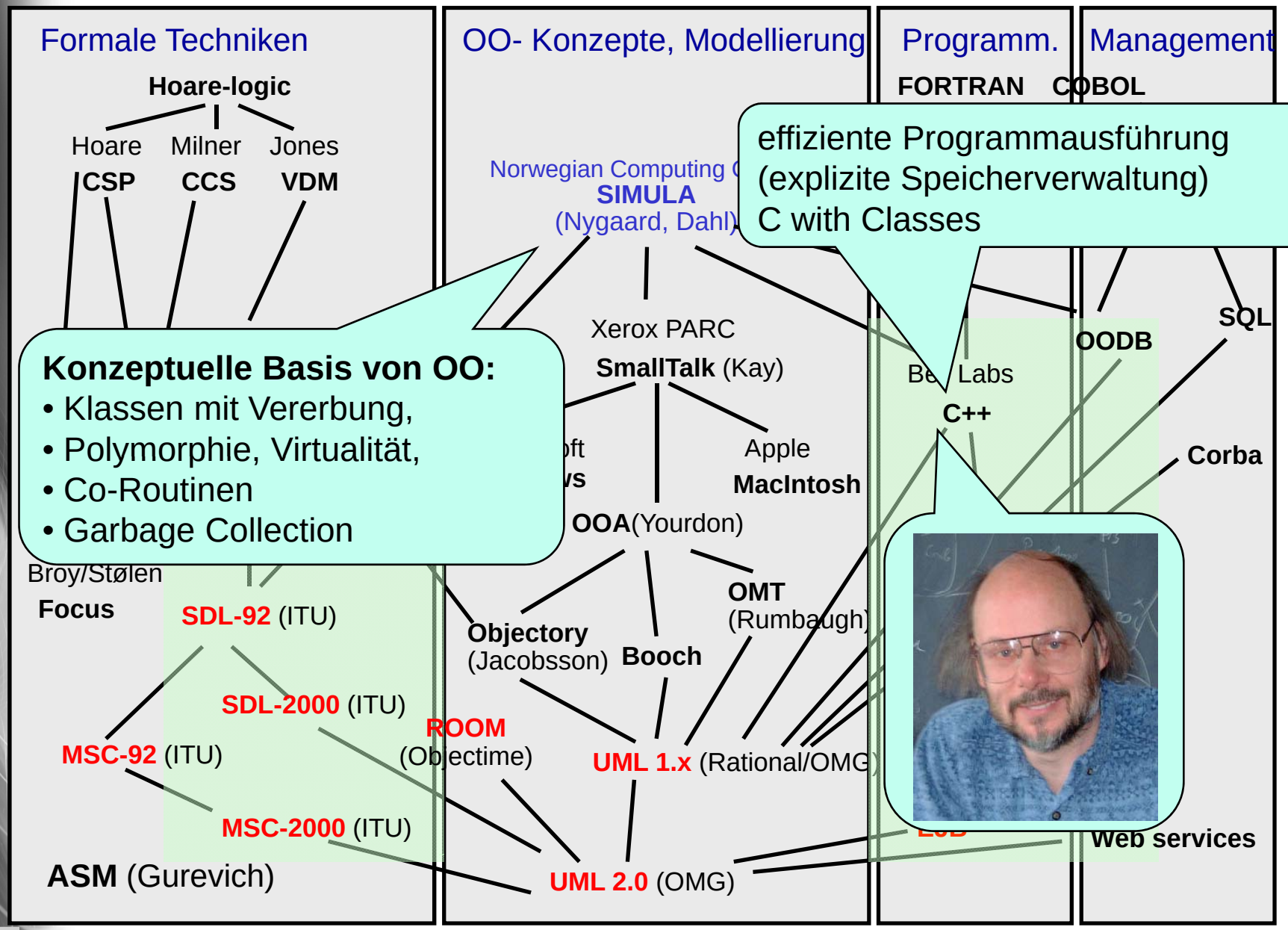


nach Grady Booch

Wurzeln der Objektorientierten Modellierung/ Programmierung




Formale Techniken
OO- Konzepte, Modellierung
Programm.
Management
Hoare-logic
Hoare
Milner
Jones
CSP
CCS
VDM
FORTRAN
Algol
Pascal
C
COBOL
ER-model
SQL
Norwegian Computing Center
SIMULA
(Nygaard, Dahl)
Xerox PARC
SmallTalk (Kay)
Apple
OOA(Yourdon)
OMT
(Rumbaugh)
Booch
UML 1.x (Rational/OMG)
Sun
JAVA
EJB
Web services
UML 2.0 (OMG)
ASM (Gurevich)
Konzeptuelle Basis von OO:
• Klassen mit Vererbung,
• Objektreferenzen,
• Polymorphie, Virtualität,
• Co-Routinen,
• Garbage Collection
• Standardklassenbibliothek
Simula als erste OO-Sprache
• Implementationssprache
• Modellierungssprache
• Byte-Code


Formale Techniken
OO- Konzepte, Modellierung
Programm.
Management
Hoare-logic
Hoare
Milner
Jones
CSP
CCS
VDM
FORTRAN
COBOL
Norwegian Computing C
SIMULA
(Nygaard, Dahl)
effiziente Programmausführung
(explizite Speicherverwaltung)
C with Classes
Konzeptuelle Basis von OO:
• Klassen mit Vererbung,
• Polymorphie, Virtualität,
• Co-Routinen
• Garbage Collection
Xerox PARC
SmallTalk (Kay)
Be Labs
C++
OODB
SQL
Corba
Apple
MacIntosh
OOA(Yourdon)
Broy/Stølen
Focus
SDL-92 (ITU)
Objectory
(Jacobsson)
Booth
OMT
(Rumbaugh)
SDL-2000 (ITU)
ROOM
(Objectime)
MSC-92 (ITU)
UML 1.x (Rational/OMG)
MSC-2000 (ITU)
Web services
ASM (Gurevich)
UML 2.0 (OMG)

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. Klassifikation dynamischer Systeme
8. Konzept der diskreten Ereignissimulation
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

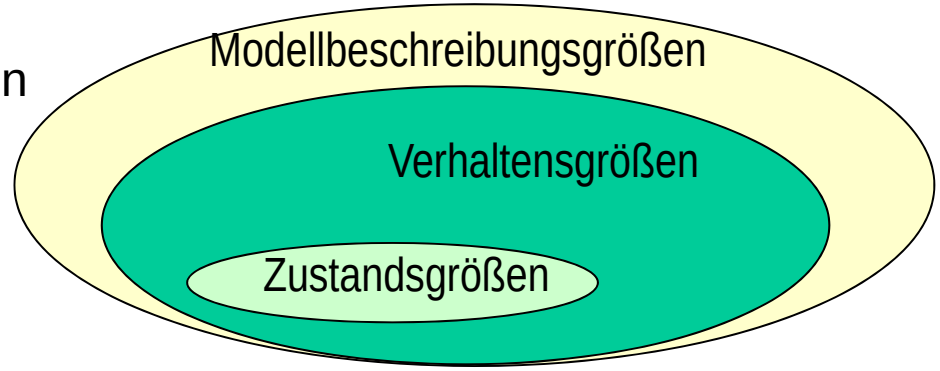
essentielle Verhaltensgrößen

- 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

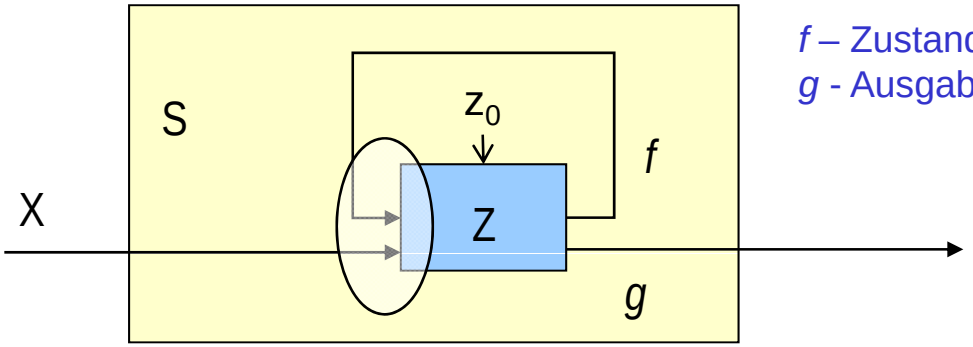


Zustandsänderungen (Prinzip)

- 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

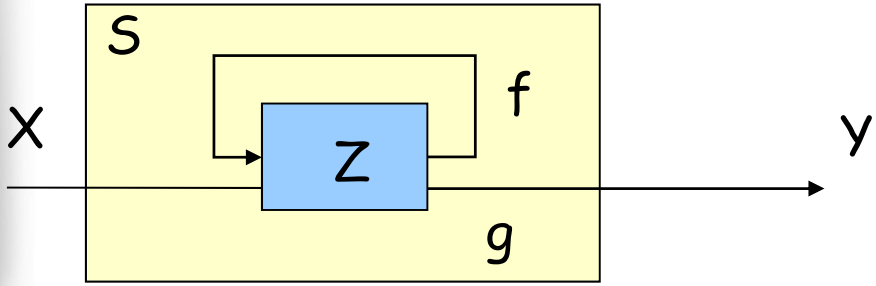
<u>Eingabe X</u>
Zuwachs
im Zeitintervall $(t_k, t_{k+1}]$

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

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

Allgemeine (Teil-)Systemdefinition

dient mehr der Klassifikation von Verhaltensmodellen

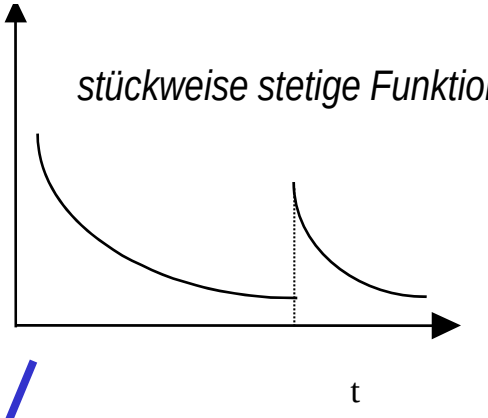


$S = (Z, z_0, X, Y, f, g, time)$

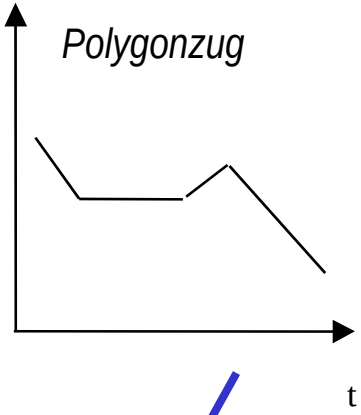
- Z — Menge der möglichen Zustände
- $z_0 \in Z$ — Anfangszustand
- X — Menge der möglichen Eingaben
- Y — Menge der möglichen Ausgaben
- time — Zeitbasis als $(T, <=, t_0)$ mit
  - Menge möglicher Zeitpunkte T,
  - einer Ordnungsrelation <= und
  - einem minimalen Element t_0
- f — $Z \times X \times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

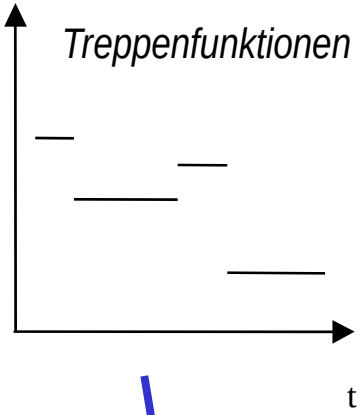
Arten von Zustandsänderungen

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

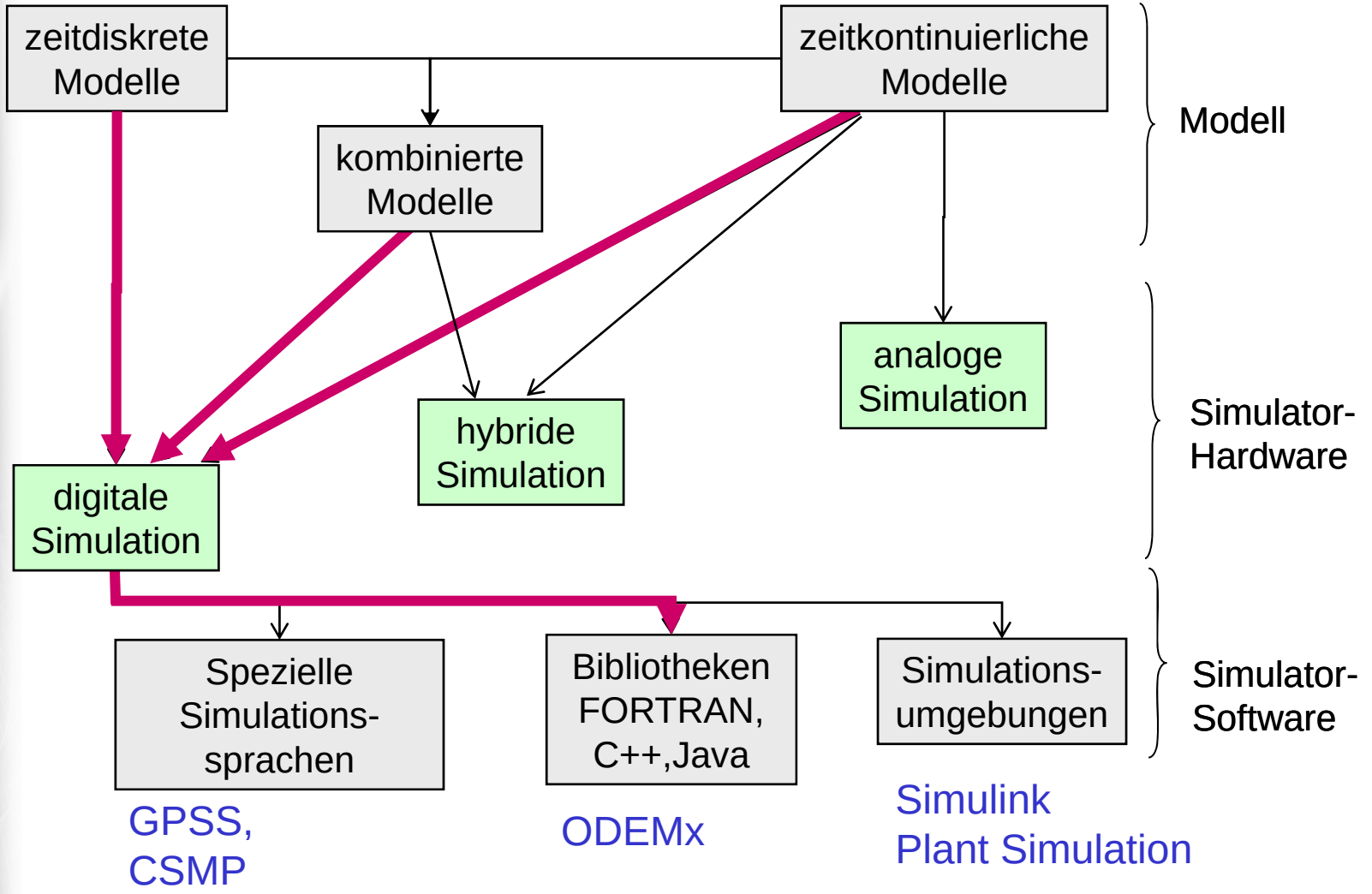
numerische Lösungsverfahren

Differenzengleichungen
zelluläre Automaten

Ereignissimulationen

Prozesssimulation

Klassifizierung von Modellen und Simulationsverfahren



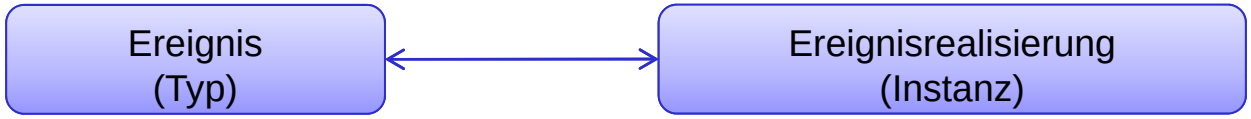
Einführung

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. Klassifikation dynamischer Systeme
8. Konzept der diskreten Ereignissimulation
9. M&S eines Niedertemperaturofens

Diskrete Ereignissimulation

- Verhalten eines Systems wird als chronologische Reihenfolge von Ereignissen verstanden

- Ereignis: Menge von Aktionen,
  die Systemveränderungen zu einem diskreten Ereigniszeitpunkt bewirken,
  wenn denn das Ereignis tatsächlich eintritt



- Sichere und unsichere Ereignisse → Bedingungen für das Eintreten von Ereignissen
  - Zeitbedingungen (→ Zeitereignisse)
  - Zustandsbedingungen (→ Zustandsereignisse)
  - Wahrscheinlichkeit (→ stochastisches Ereignis)

- Bedienungssysteme sind klassische Vertreter von Systemen, deren Verhalten sich durch sequentielle Ereignissimulationen beschreiben lassen
  (Ereignisse: Ankunft von Kunden, Beginn und Ende einer Bedienung)

- klassische Methode: NEXT-EVENT-Simulation (Sprung von Ereignis zu Ereignis)
  verschiedene Realisierungsverfahren:
  - Ereignis-basiert,
  - Aktivitäts-basiert
  - Prozess-basiert

Konzepte der diskreten Ereignissimulation

- SystemStruktur (Zustand)
  eine Menge von Variablen
  (ausgezeichnete Menge von Modellbeschreibungsgrößen)
  beschreibt in ihrer Belegung den Systemzustand zum aktuellen Zeitpunkt

- Systembewertungsgrößen
  (z.B. statistische Kenngrößen)

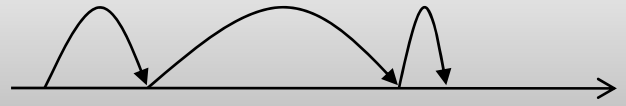
- Uhr
  (Modellzeit, dimensionslos, monoton wachsend)

- Ereigniskalender (Future Event List)

- Umgang mit Gleichzeitigkeit von Ereignissen (Current Event List)

- Zeitfortschritt:
  Bestimmung des nächsten Ereignisses aus der Future Event List,
  Setzen der Uhr (aktuelle Modellzeit:= Ereigniszeit)

Zustandsänderungen finden immer nur zu diskreten Zeitpunkten statt



Ereignis-Chakteristik

Ereigniszeit

Ereignisroutine (Parameter: Referenzen zu Systemstruktur komponenten {seq. Code}

(Event) Scheduler

Zeitfortschritts-realisierung

Realisierung des aktuellen Ereignisses

Event-Scheduling-Algorithmus