

# ***Kurs OMSI*** ***im WiSe 2011/12***

## ***Objektorientierte Simulation*** ***mit ODEMx***

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

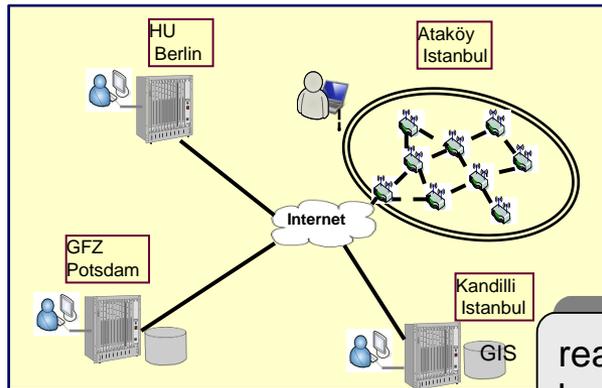
[fischer|ahrens|eveslage@informatik.hu-berlin.de](mailto:fischer|ahrens|eveslage@informatik.hu-berlin.de)

# 1. *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. Scheduler für zeitdiskrete und zeitkontinuierliche Systemmodelle
9. M&S eines Niedertemperaturofens

# 1. Projekt: Erdbebenfrühwarnsystem

Sonderfall:  
Modell und Original-  
Teile  
sollen als Software  
vorliegen



reales  
bzw. gedachtes  
**Phänomen**

Ausführungs-  
Modell im Speicher  
**Simulator**

Experiment-  
Management-  
System

**Ziel**

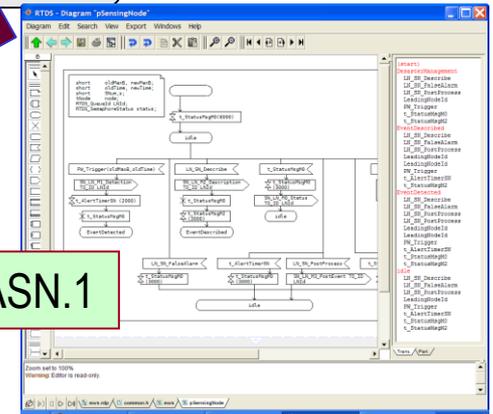
informales  
/semiformales  
**Systemmodell**

unvollständige  
Darstellung:  
UML/SDL

formales  
mathematisches  
**Simulationsmodell**

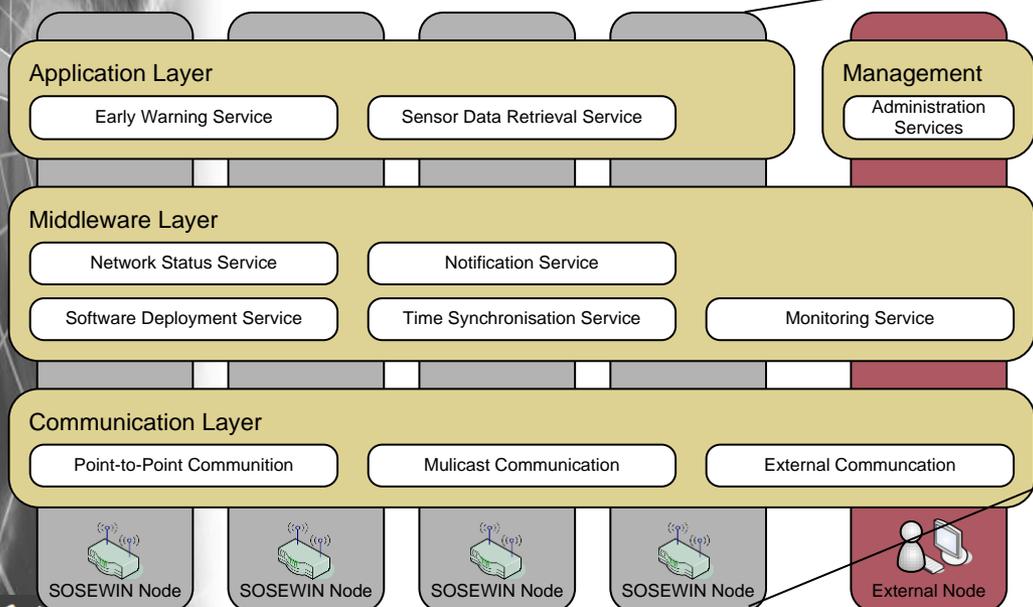
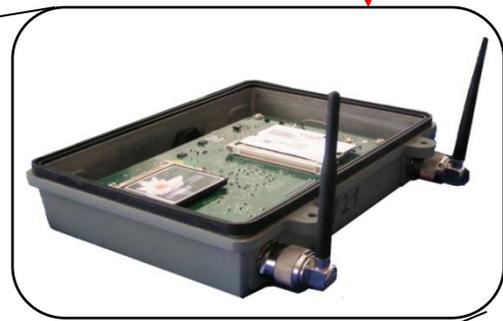
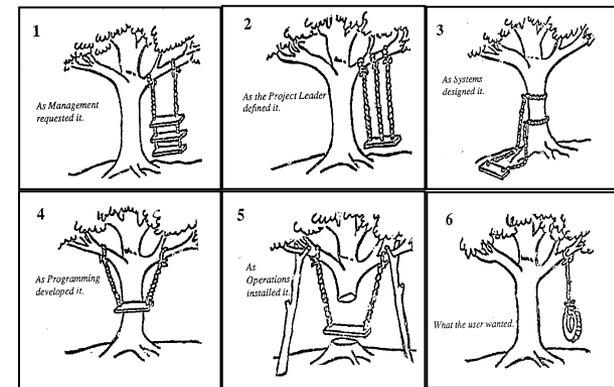
**PIM: SDL,UML,ASN.1**

**PSM (Zielcode): C++, OpenWRT, Boost**



*Objektorientierte Simulation mit ODEMX*

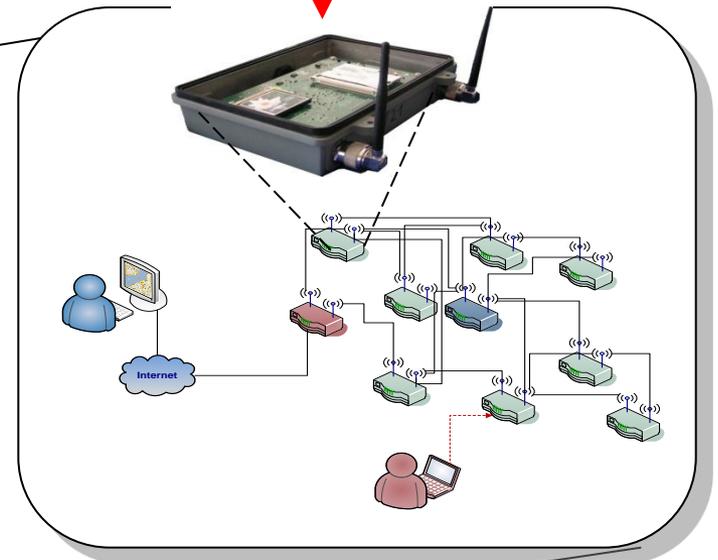
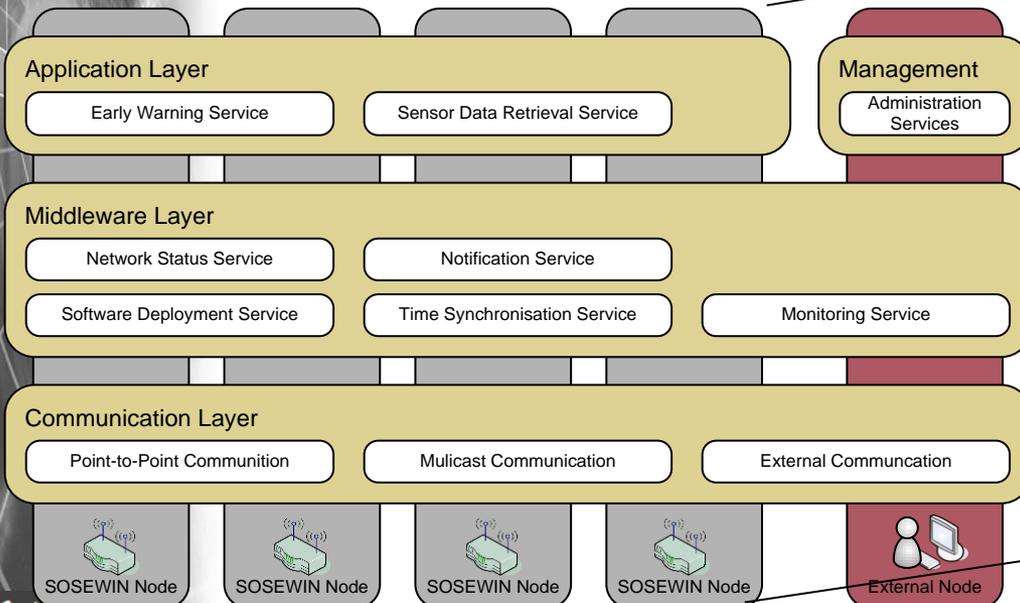
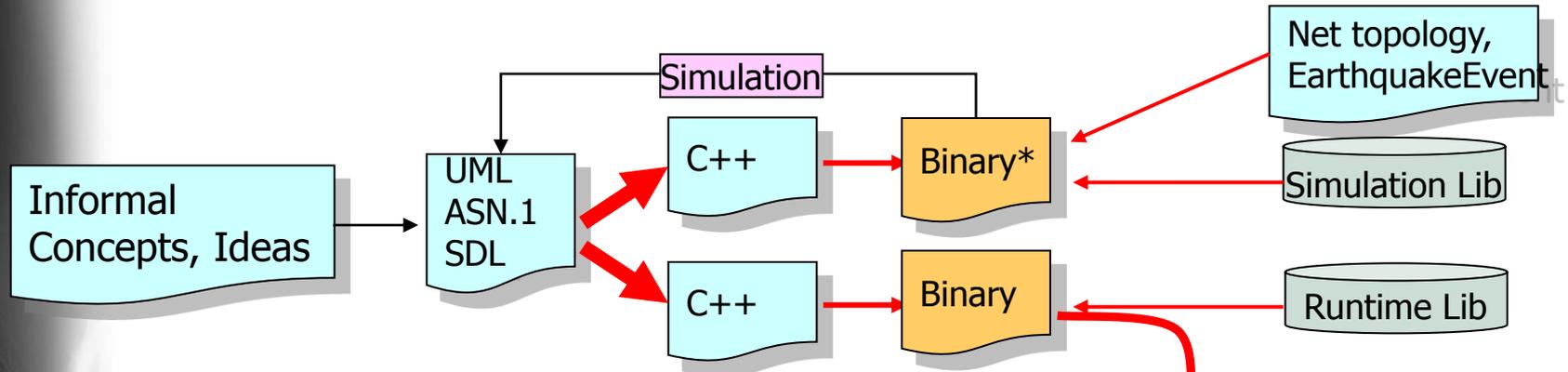
# Traditionelle Software-Entwicklung



Test eines einzelnen Knotens reicht nicht

Test eines kompletten Systems ist problematisch

# Ein modellgetriebener Ansatz



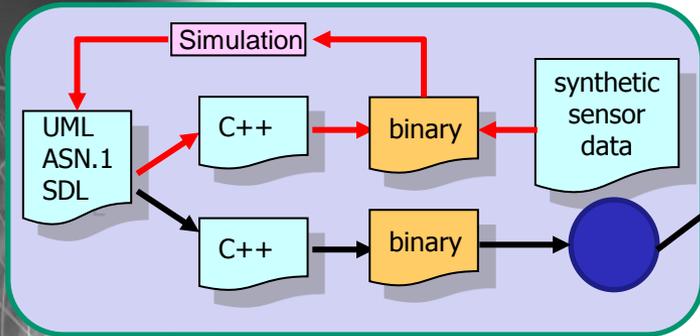
Simulation des kompletten Systems vereinfacht das Testen enorm

# SOSEWIN-Überblick

Self-organized Seismic Early Warning Information Network

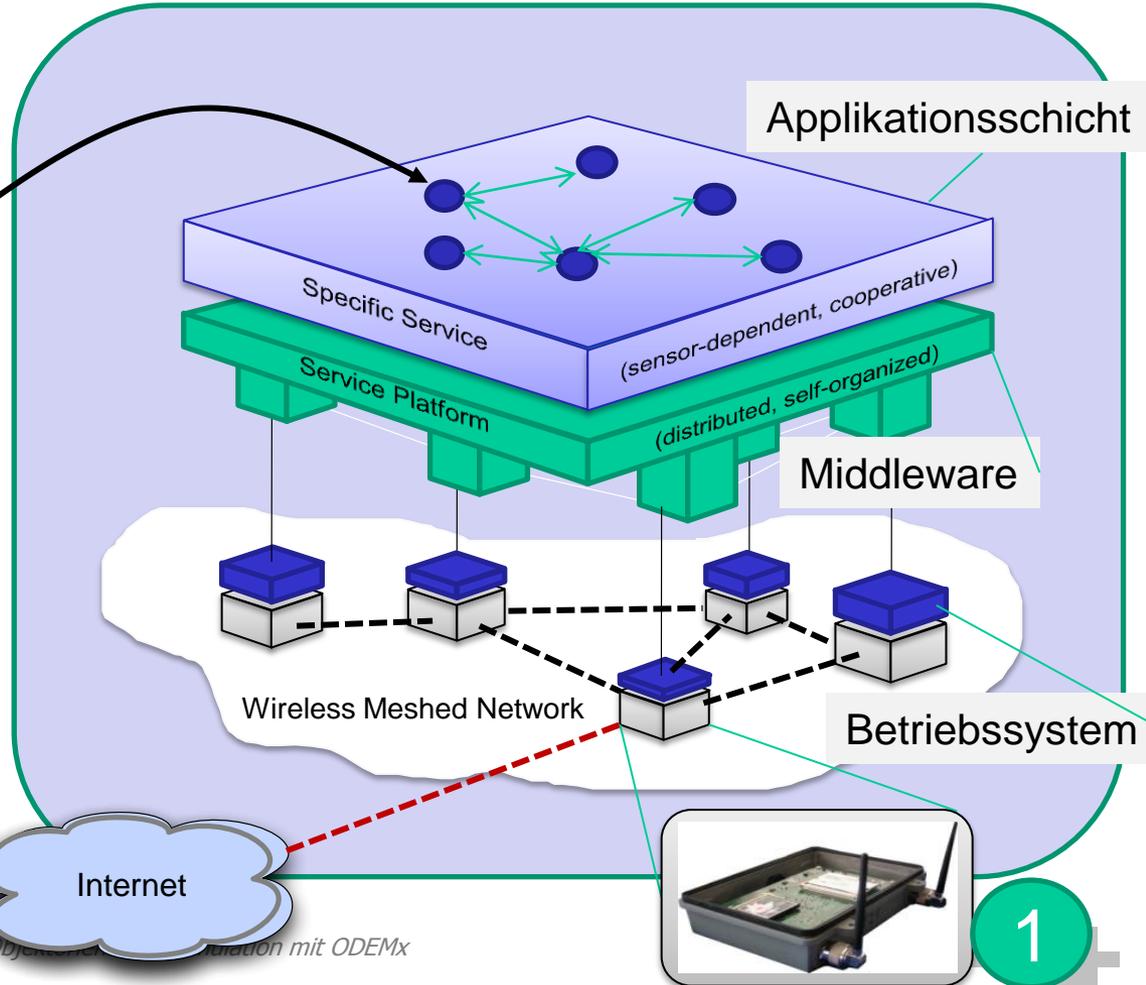
3

SW- Entwicklungstechnologie  
(Modelleditor, Simulator, Code -Generator, ...)



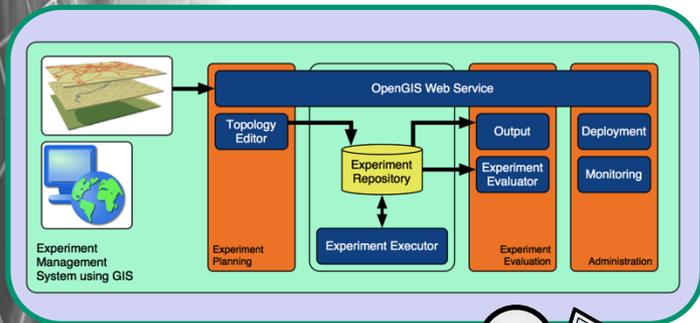
2

SOSEWIN-HW/SW Architektur



4

GIS-basierte Netzmanagement-  
und Experiment-Unterstützung



# Netz-Prototyp

Self-organized Seismic Early Warning Information Network



HU  
Berlin



5

Ataköy  
Istanbul

GIS



Internet



Kandilli  
Istanbul

GFZ  
Potsdam



GIS



Istanbul-  
Infrastruktur-  
daten



Objektorientierte Simulation mit ODEMX

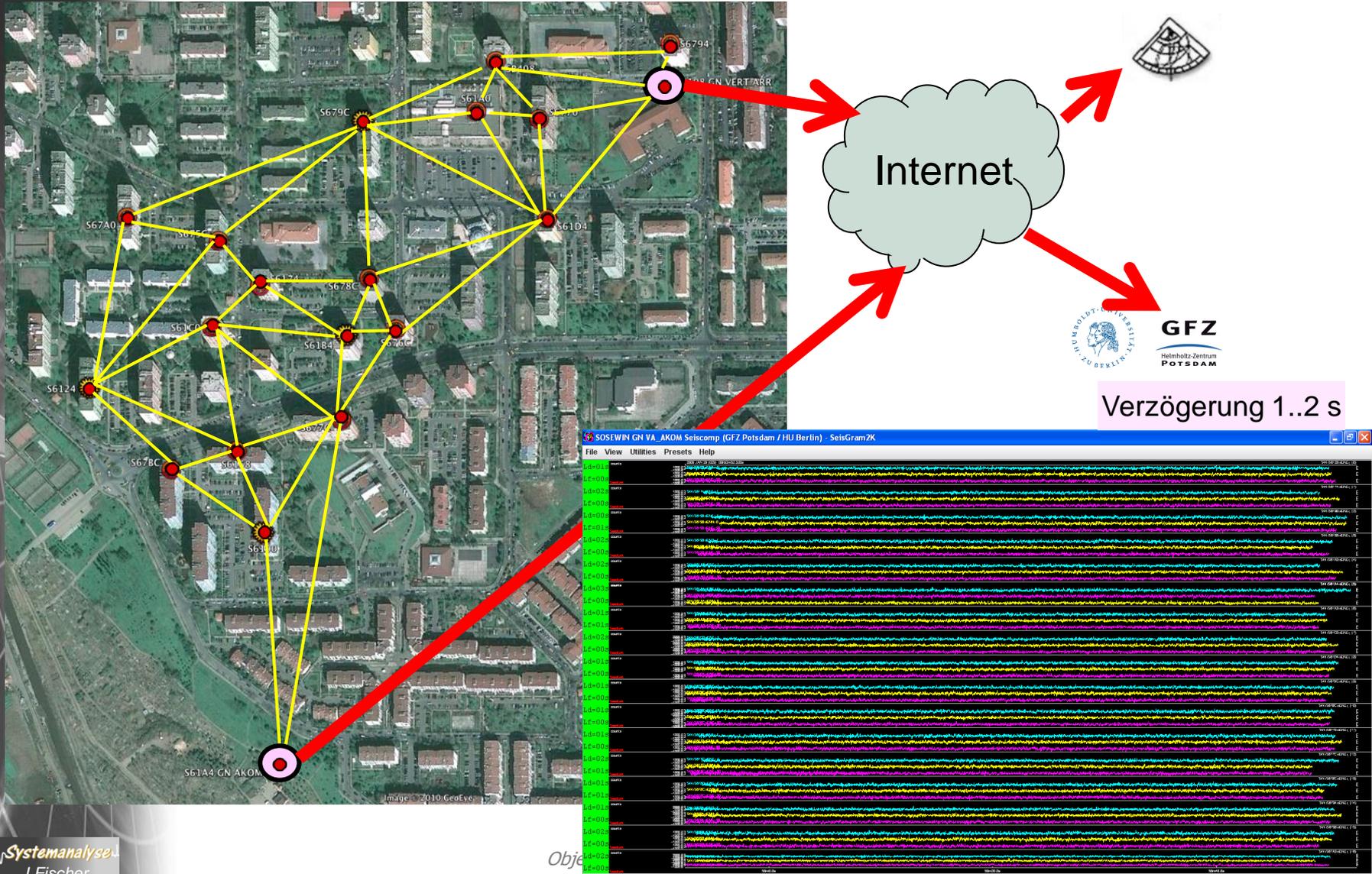
# Erstinstallation von SOSEWIN



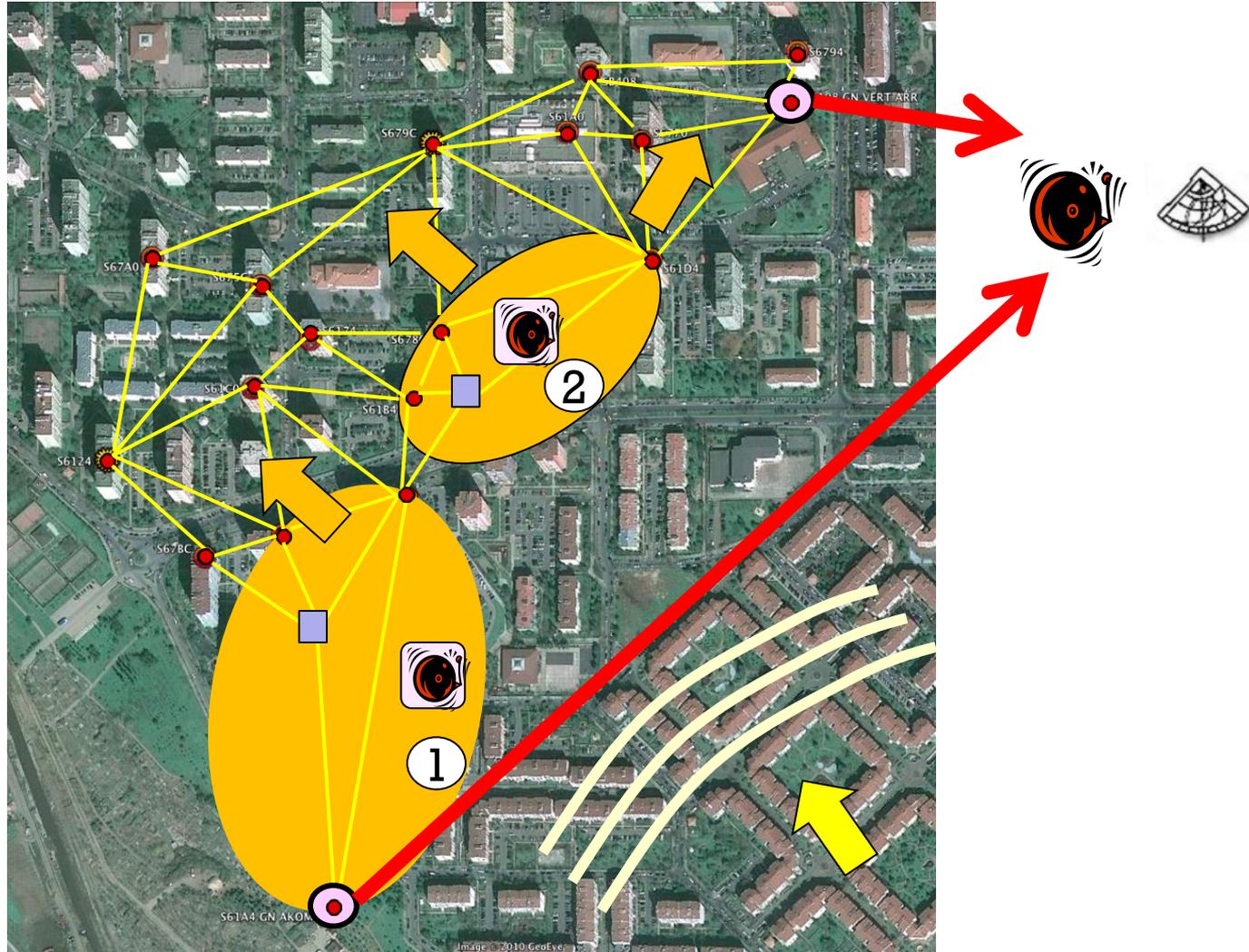
Istanbul, Mai 2008

*mit Sensorik, aber zunächst noch ohne Alarmierungssoftware*

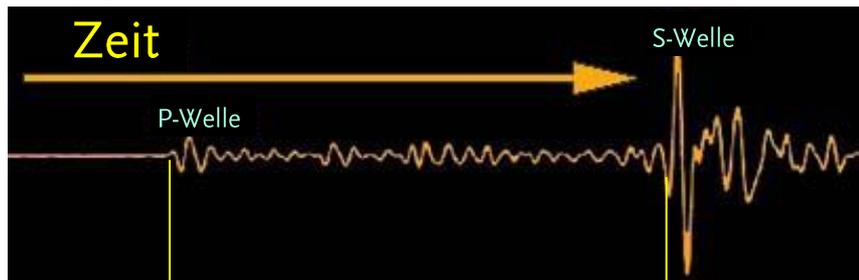
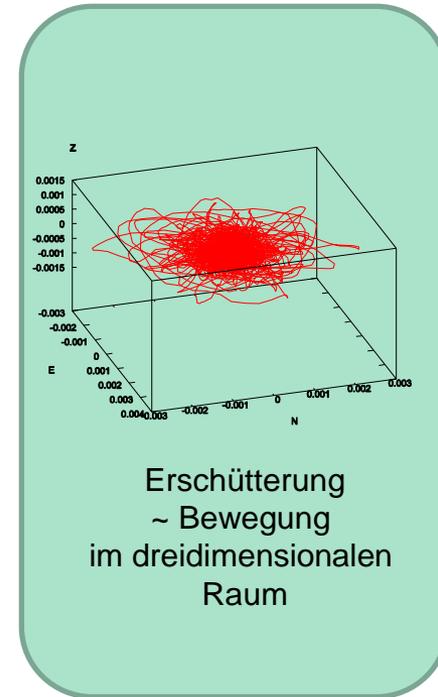
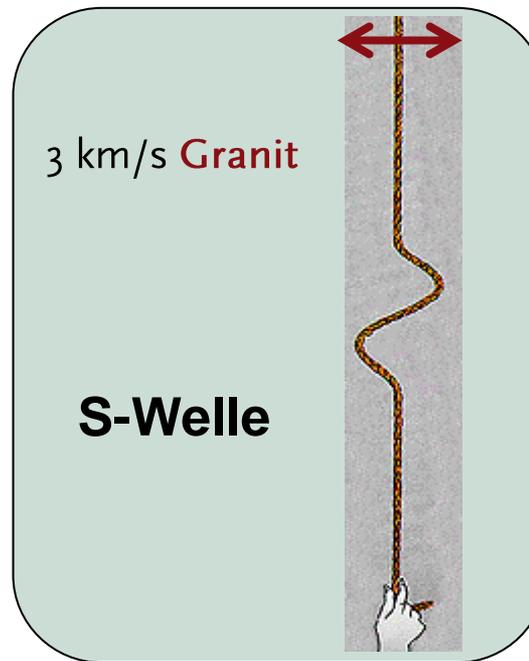
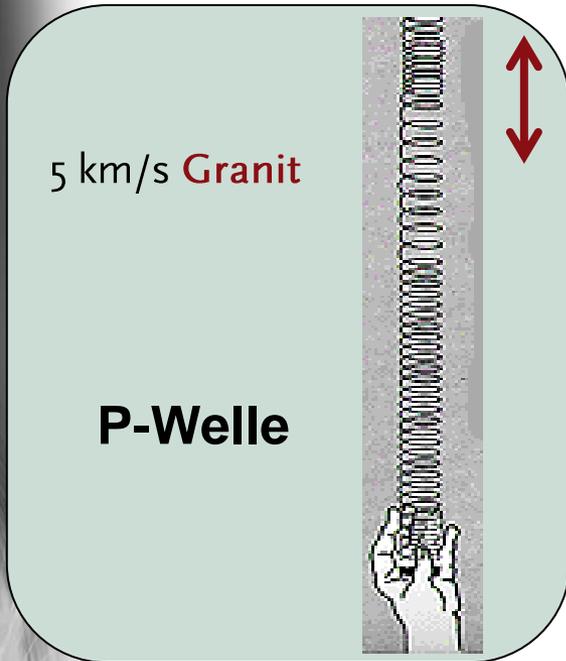
# Alle Seismometer arbeiten zusammen



# Alarm bei Eintreffen der P-Welle



# Wellenarten

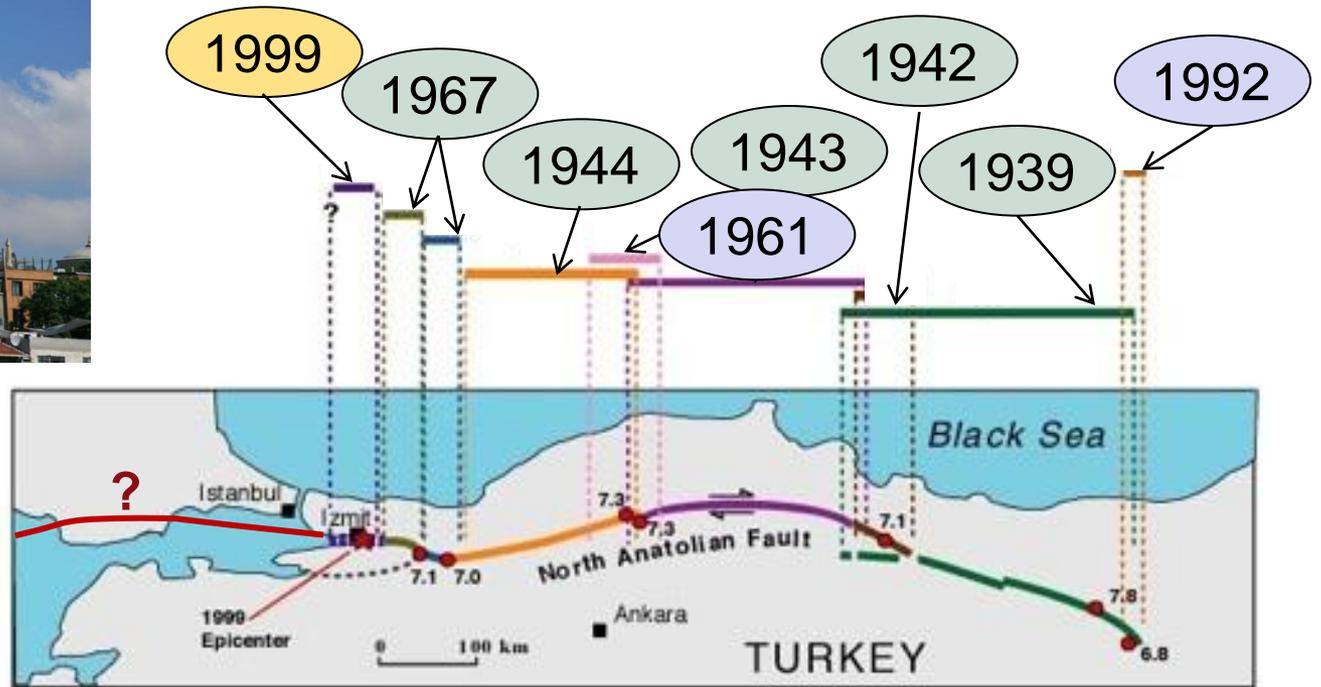


**Vorwarnzeit**



*Objektorientierte Simulation mit ODEIMx*

# Ernsteste Bedrohung von Istanbul



Quelle:  
<http://pubs.usgs.gov/circ/2000/c1193/c1193.pdf>

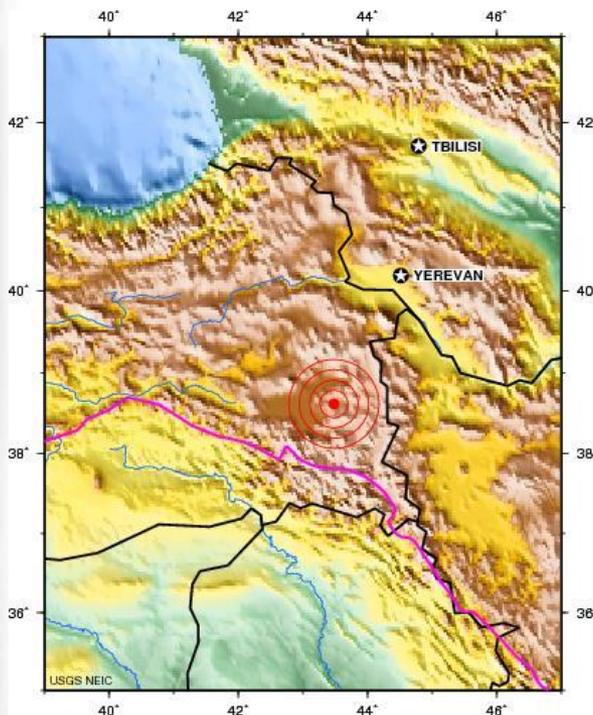
**Izmet-Beben: M 7,4 ~ 125-fache Energie der Hiroshima-Bombe**

- 20.000 zerstörte Häuser
- 40.000 Verletzte
- 24.000 Tote

# Jüngstes Beben

Ursache des Bebens ist die Kollision der relativ kleinen Arabischen und der Eurasische Platte, die von Island bis Japan reicht.

Beide bewegen sich mit etwa 2 bis 2,5 Zentimetern pro Jahr aufeinander zu. Dadurch sei auch der steil ansteigende Kaukasus entstanden, (wächst immer noch ~ähnlich baut die indische Platte das Himalaya-Gebirge auf



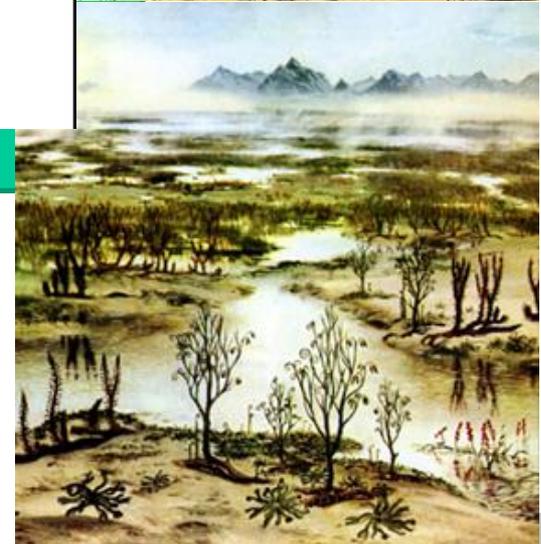
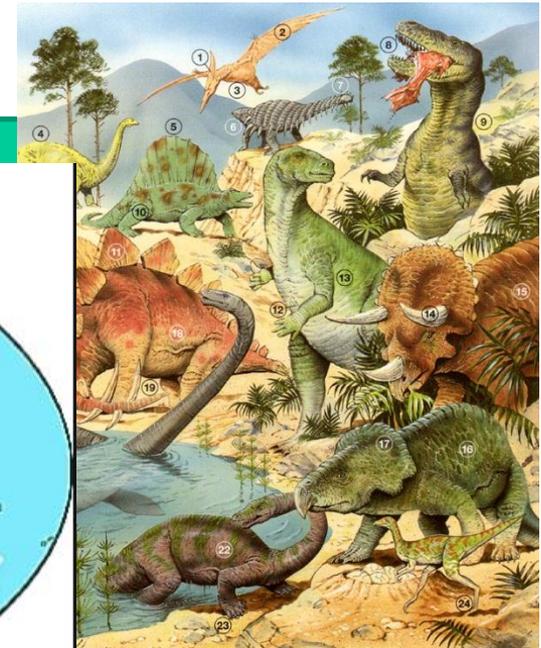
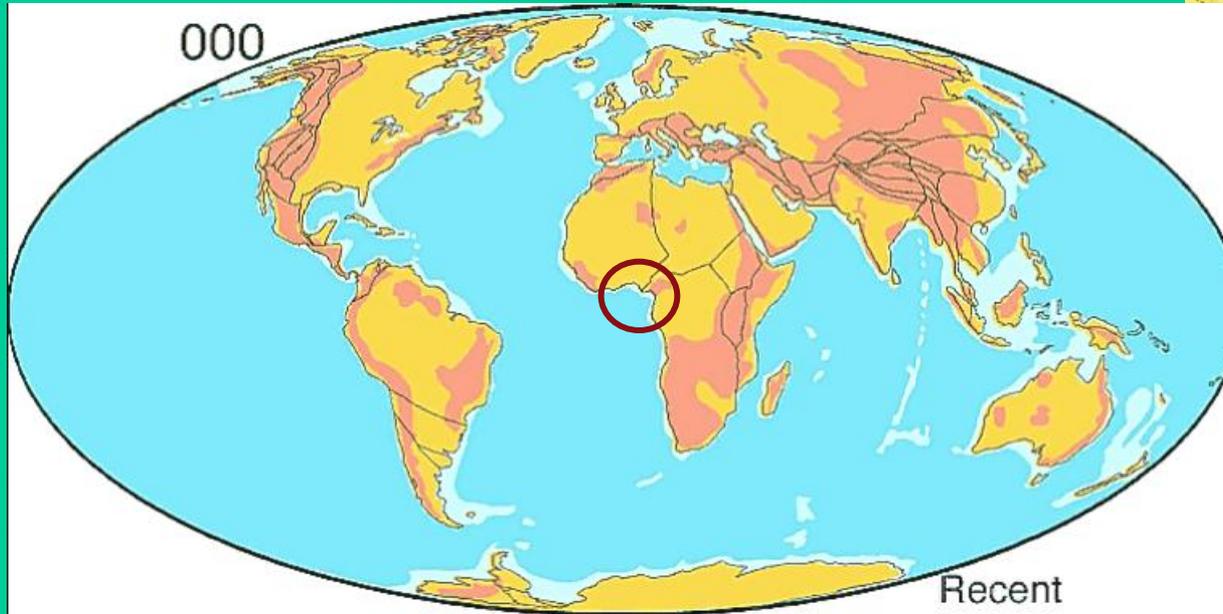
EASTERN TURKEY

2011 10 23 10:42:21 UTC 38.62N 43.48E Depth: 20.0 km

Earthquake Location

|                      |  |
|----------------------|--|
| Magnitude            | <b>7.2</b>   |
| Date-Time            | <b>•Sunday, October 23, 2011 at 10:41:21 UTC</b><br>•Sunday, October 23, 2011 at 01:41:21 PM at epicenter  |
| Location             | 38.628°N, 43.486°E   |
| Depth                | 20 km (12.4 miles) set by location program   |
| Region               | EASTERN TURKEY   |
| Distances            | 16 km (9 miles) NNE of <b>Van, Turkey</b><br>118 km (73 miles) N of <b>Hakkari, Turkey</b><br>127 km (78 miles) SSE of <b>Agri (Karakose), Turkey</b><br>929 km (577 miles) E of <b>ANKARA, Turkey</b> |
| Location Uncertainty | Error estimate not available   |
| Parameters           | Nph=0, Dmin=0 km, Rmss=0 sec, Gp= 0,<br>M-type=centroid moment magnitude (Mw), Version=1   |
| Source               | •USGS NEIC (WDCS-D)  |
| Event ID             | usb0006bqc   |

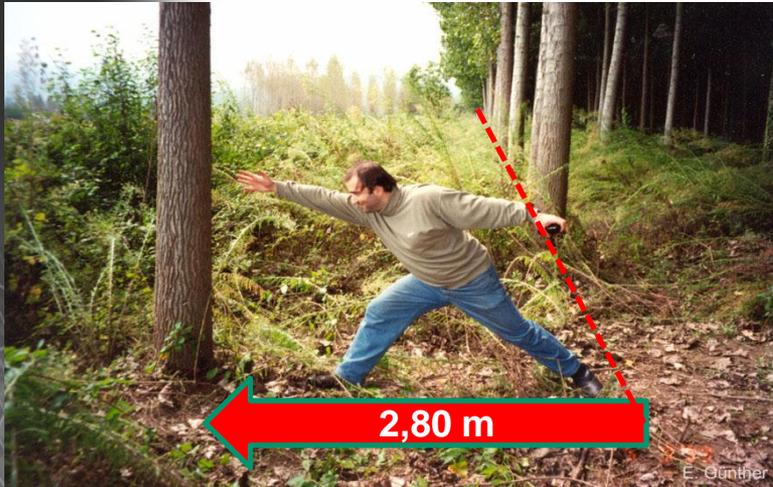
# Animierte Zeitreise: Wir drehen die Zeit zurück



Wie sah die Erde vor 300 Mill Jahre aus ?

wo hätte sich Berlin befunden?

# Letzte Warnung 1999: Izmit-Beben



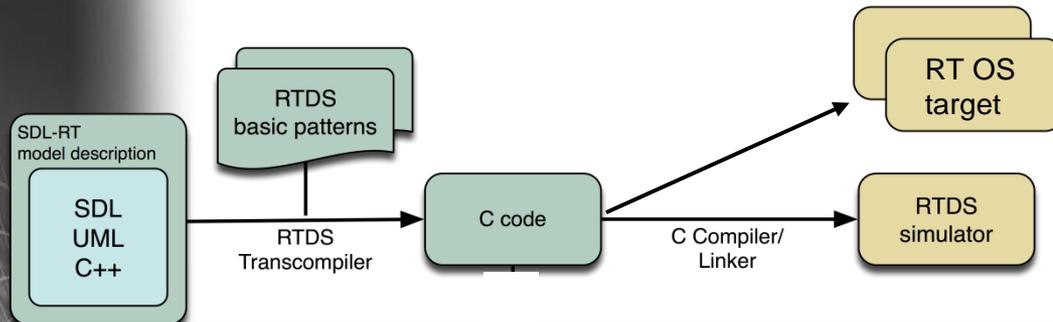
A right-lateral displacement of 2.8 m was observed at the Sapanca segment near Caybasi (40.703° N, 30.451° E)



erte Simulation mit ODEMx

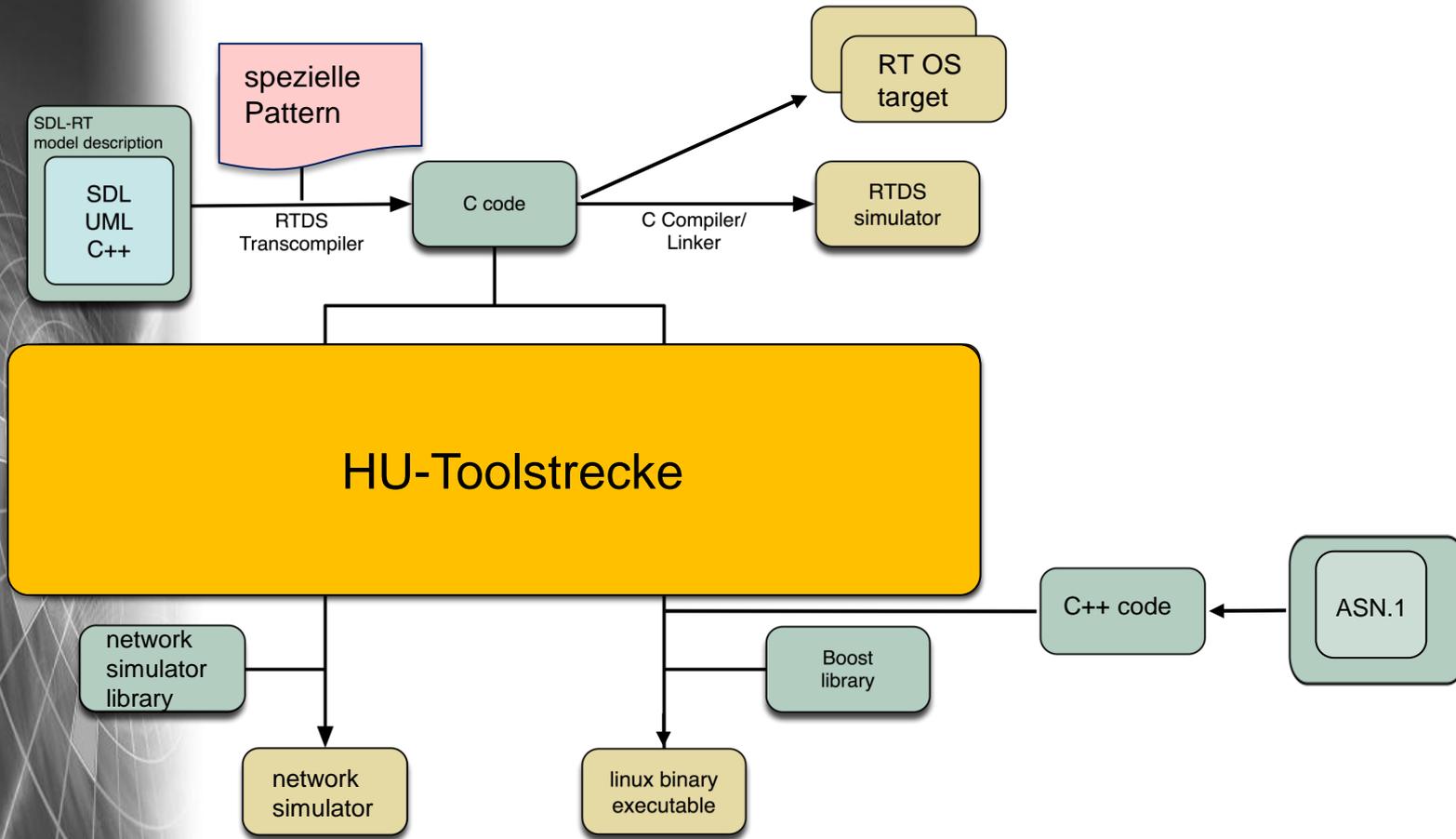
# Erweiterung des PragmaDevs SDL/RT-Compilers

## ODEMx/Boost Pattern

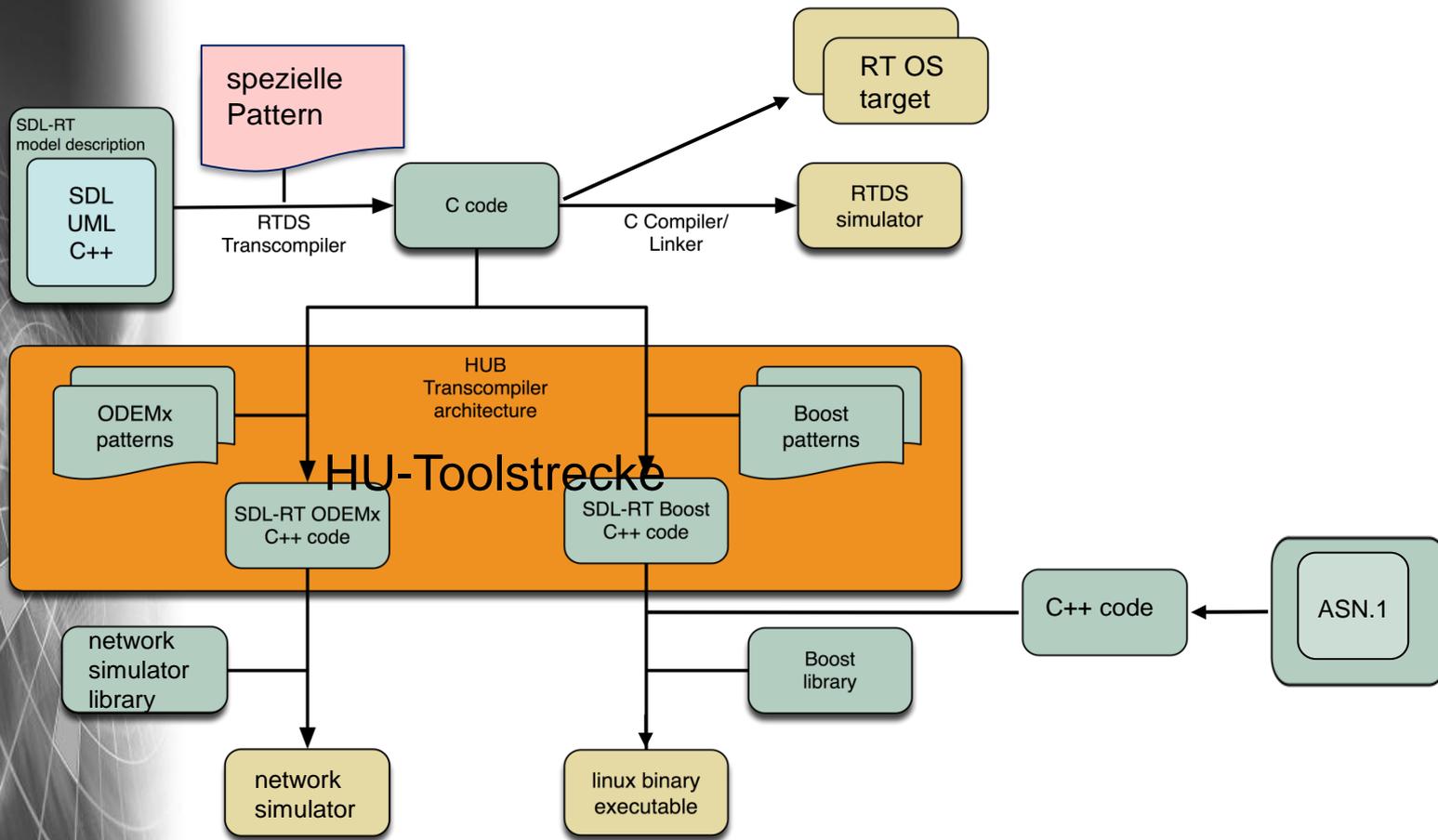


C++ Artefakte  
(nach Transformation  
von markiertem  
C-code)

# Erweiterung des PragmaDevs SDL/RT-Compilers



# Erweiterung des PragmaDevs SDL/RT-Compilers



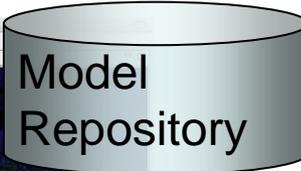
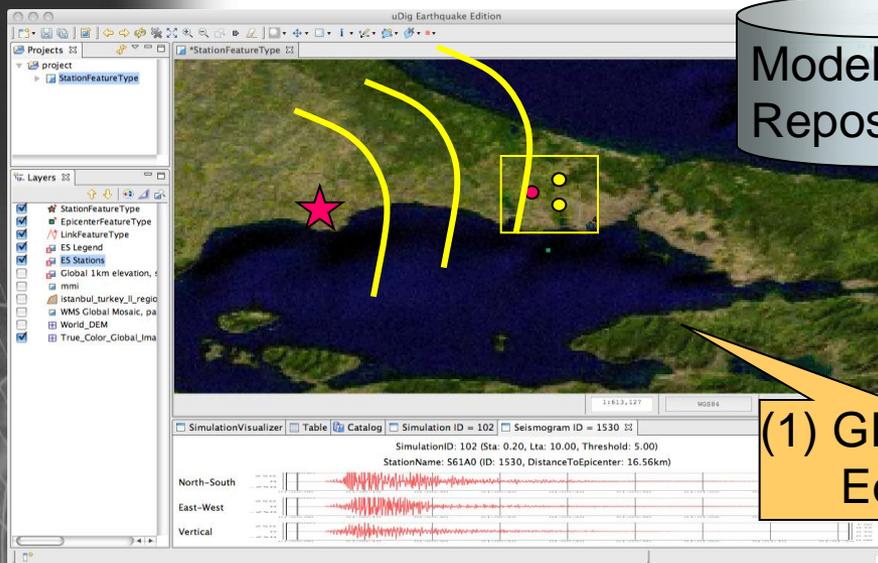
# Benötigte Simulatoren

zur Nachbildung von

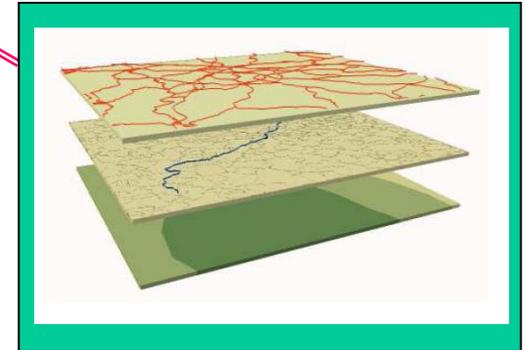
- Erdbebenwellenausbreitung
  - Epizentrum, Tiefe, Stärke, Bruchzonenbeschreibung
- Knotenfunktionalität
  - unterschiedlicher Netzwerkschichten bei Variation von Routing-Protokollen
  - unterschiedlicher Topologien
  - unterschiedlicher Störungseinflüsse

bei Einbindung in ein Geo-Informationssystem

# Models, Target Components, and Tools

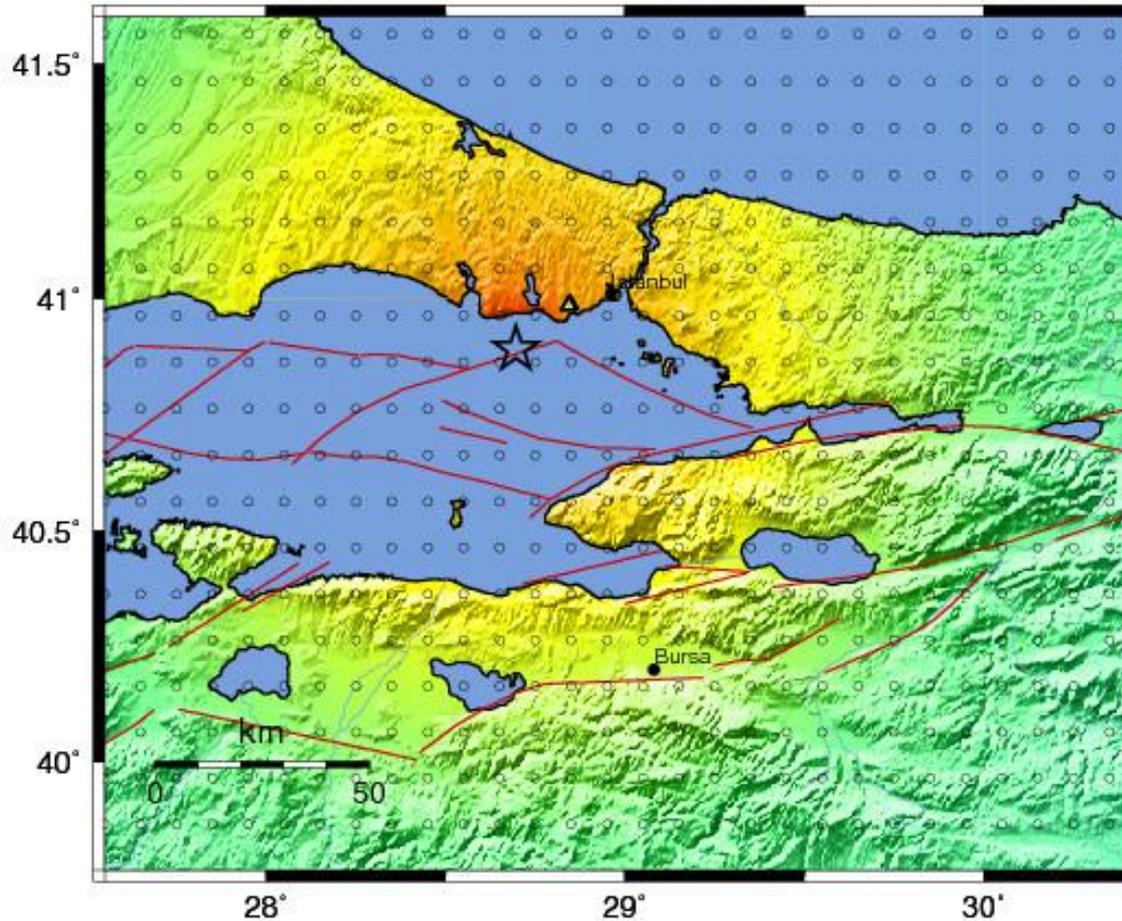


(1) GIS-based Editor



GFZ ShakeMap : Location description default. Created by Network Editor.

Thu Feb 5, 2009 05:16:06 PM SST M 7.4 N40.89 E28.70 Depth: 10.0km ID:60



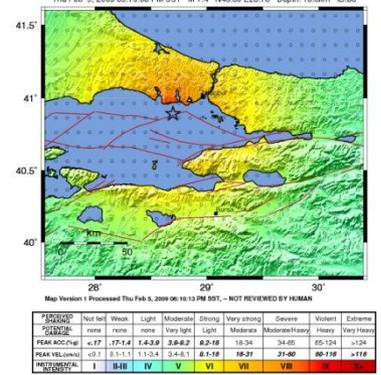
Map Version 1 Processed Thu Feb 5, 2009 06:10:13 PM SST, -- NOT REVIEWED BY HUMAN

| PERCEIVED SHAKING      | Not felt | Weak    | Light   | Moderate   | Strong | Very strong | Severe         | Violent | Extreme    |
|------------------------|----------|---------|---------|------------|--------|-------------|----------------|---------|------------|
| POTENTIAL DAMAGE       | none     | none    | none    | Very light | Light  | Moderate    | Moderate/Heavy | Heavy   | Very Heavy |
| PEAK ACC.(%g)          | <.17     | .17-1.4 | 1.4-3.9 | 3.9-9.2    | 9.2-18 | 18-34       | 34-65          | 65-124  | >124       |
| PEAK VEL.(cm/s)        | <0.1     | 0.1-1.1 | 1.1-3.4 | 3.4-8.1    | 8.1-16 | 16-31       | 31-60          | 60-116  | >116       |
| INSTRUMENTAL INTENSITY | I        | II-III  | IV      | V          | VI     | VII         | VIII           | IX      | X+         |

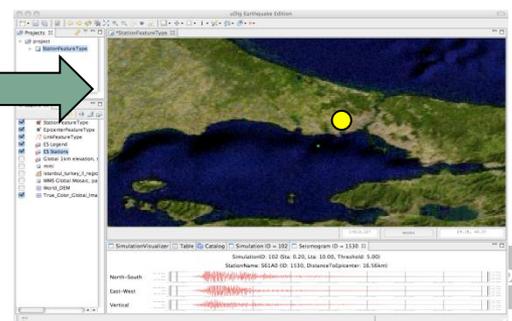
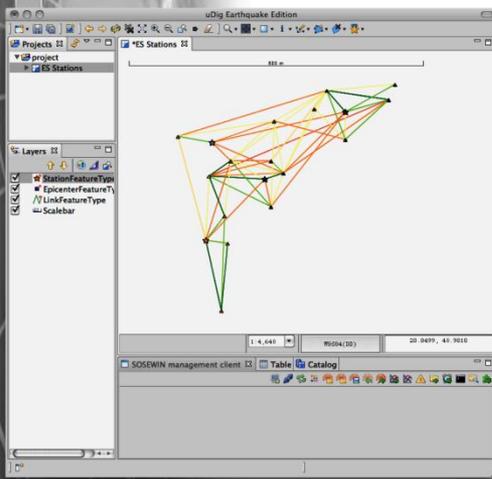
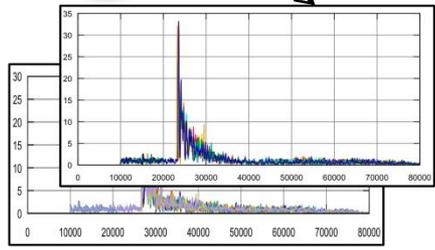
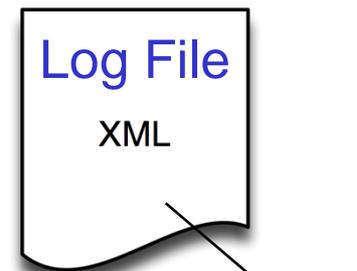
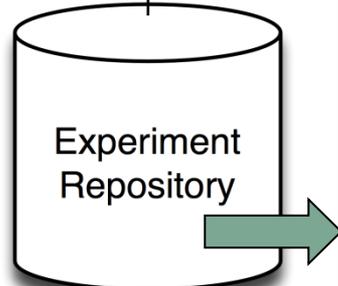
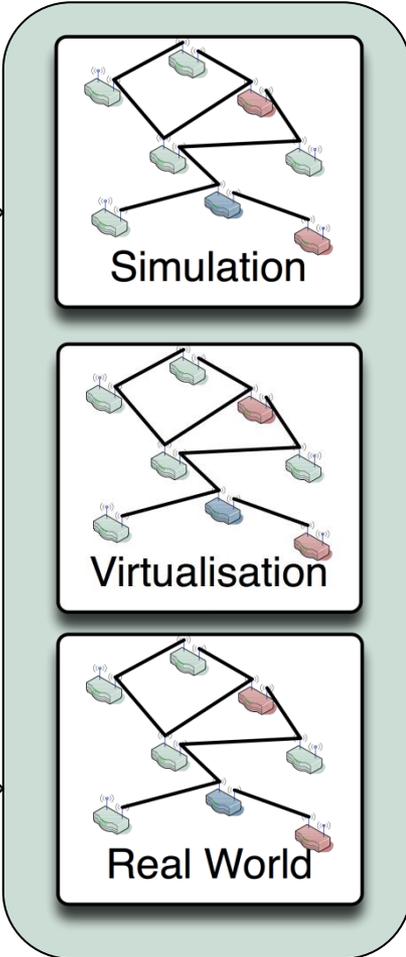
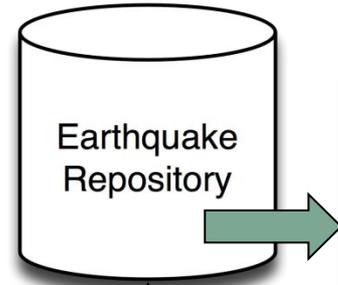
# Experiment Management System

## Executables

GFZ ShakeMap : Location description default. Created by Network Editor.



| ID            | Name                                       | Status    | Date                |
|---------------|--|-----------|---------------------|
| Completed 208 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 209 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 210 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 211 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 212 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 213 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 214 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 215 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 216 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 217 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 218 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 219 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 220 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 221 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 222 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 223 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 224 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 225 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 226 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 227 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 228 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 229 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 230 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 231 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 232 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 233 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 234 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 235 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 236 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 237 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 238 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 239 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 240 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 241 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 242 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 243 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 244 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 245 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 246 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 247 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 248 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 249 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |
| Completed 250 | Intelligence/Operations/Windows SP 1.1.480 | Completed | 2009-08-29 22:24:59 |

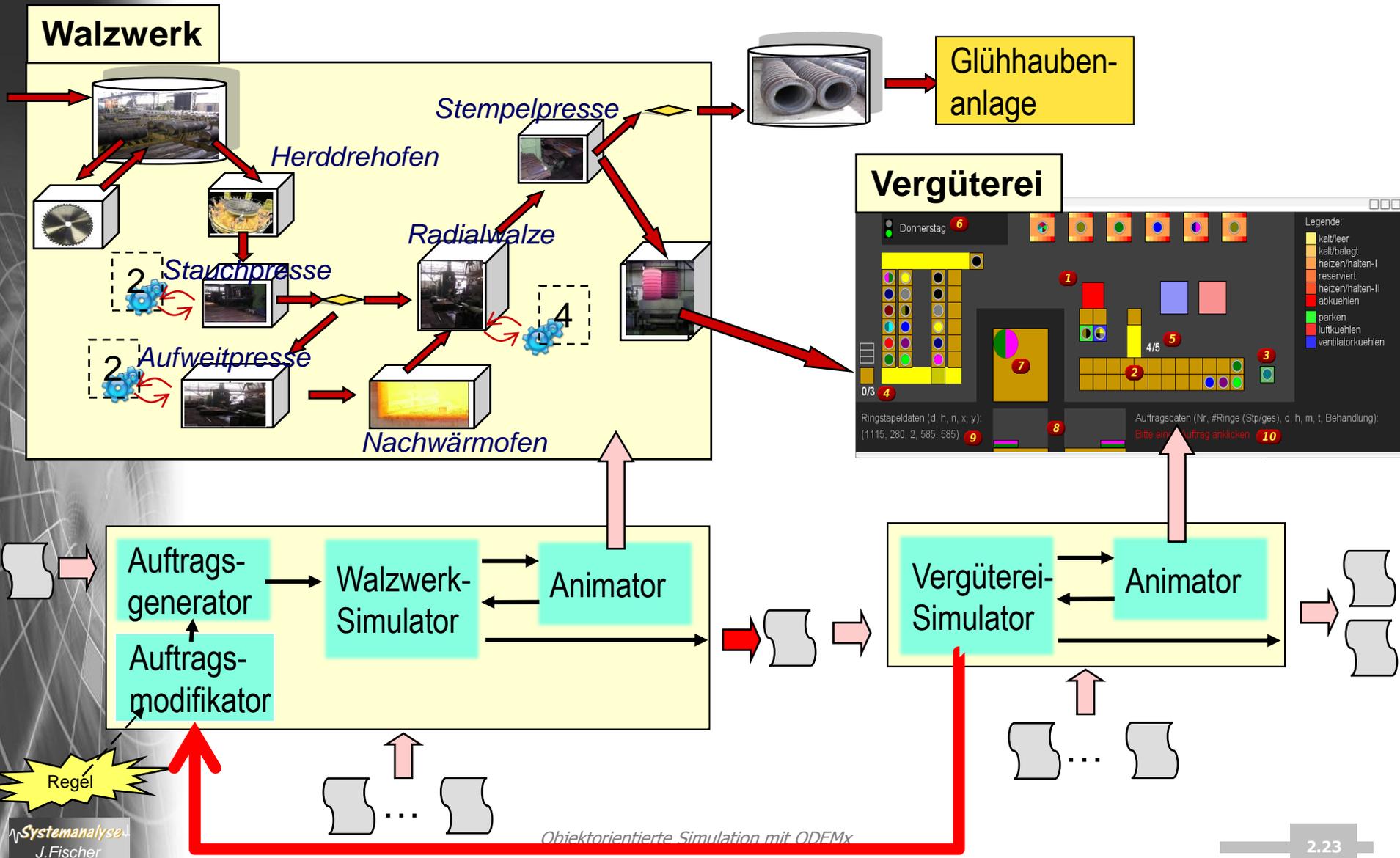


GIS-based editor

J.Fischer

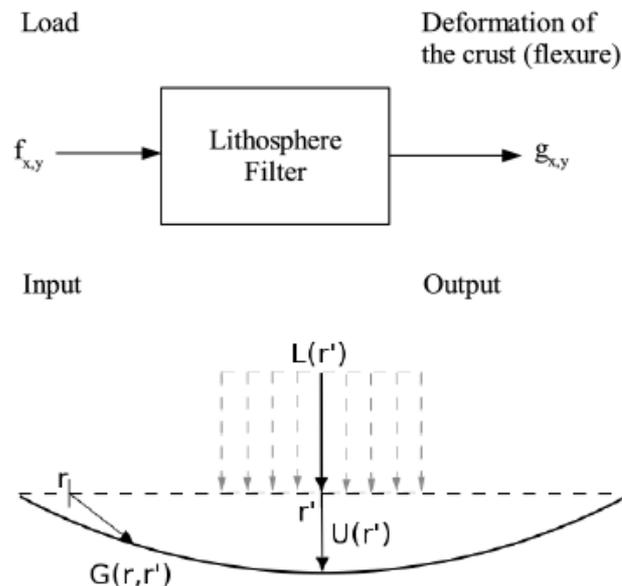
orientierte Simulation mit ODEMx

# 2. Projekt: Workflow-Modell von Ring-Walzprozessen

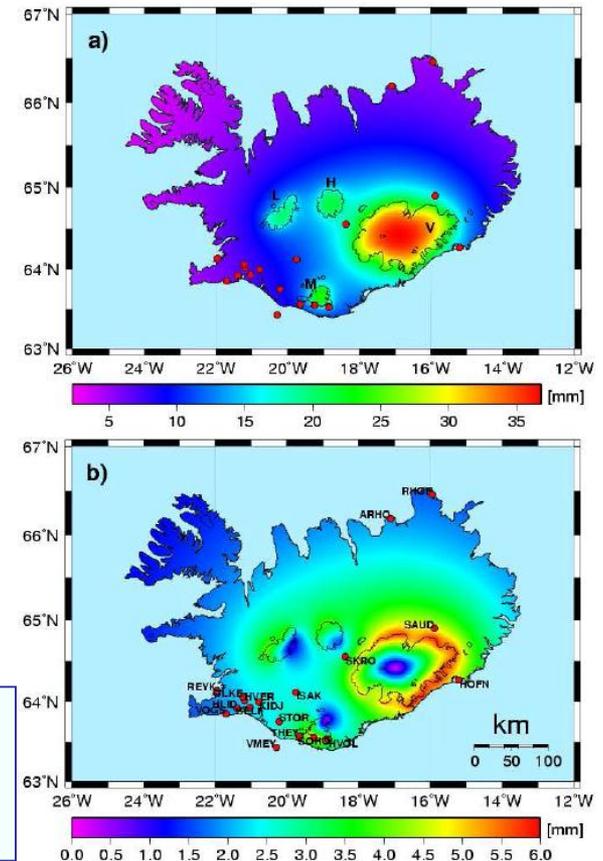


# 3. Projekt: Gletschereis-Last-Simulator

**Hintergrund:** FO-Arbeiten am Nordic Volcanological Centre (Reykjavik)  
*Einfluss des Abschmelzens isländischer Gletscher auf die Deformation der Erdoberfläche*



**Ziel:** Aufbau einer komponentenorientierten Simulationsumgebung zur Modellvalidierung im Abgleich mit historischen Messungen

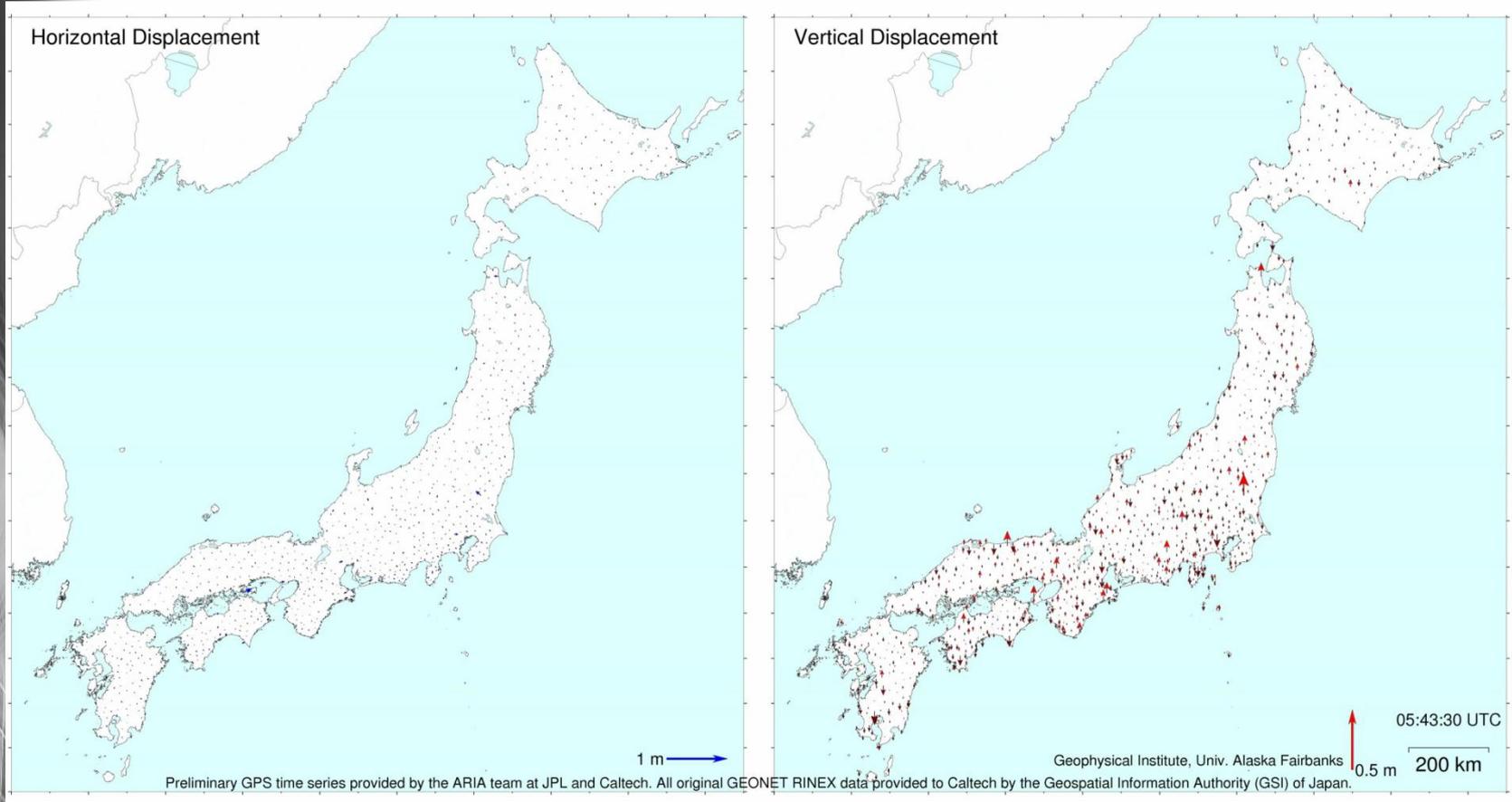








# Bedeutung von Animationen: *Visualisierung von Erdbewegungen*



# 1. *Einführung*

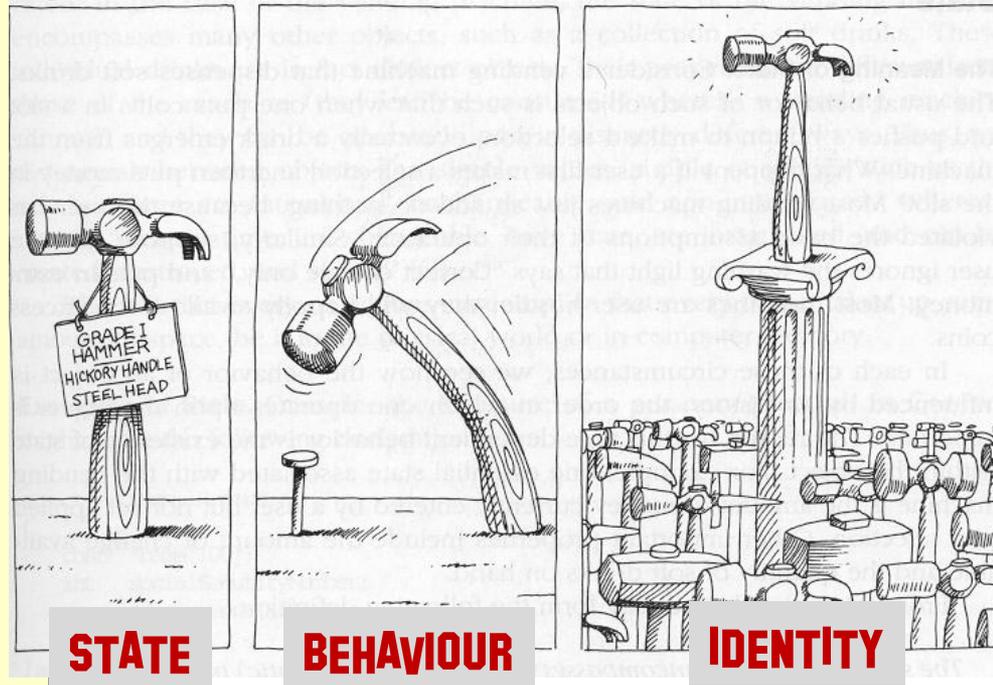
1. Systemsimulation – was ist das?
2. Ein Blick zurück in die Anfänge
3. Modelle und Originale
4. Modellierungssprachen, Simulationsumgebungen
5. Beispiele aus der aktuellen Forschung
6. Paradigma der objektorientierten Modellierung
7. Klassifikation dynamischer Systeme
8. Scheduler für zeitdiskrete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

# Modellierungsparadigmen

## Objektorientiertes Abstraktionskonzepte

1. eigenverantwortlich handelnde, interagierende Dinge (Objekte)
  - Zustand (Attribute)
  - individuelles Verhalten (Methoden, Dienste)
  - Identität (Referenz)
2. Klassen (Definition von Objekten)
3. Unterscheidung zw.
  - aktiven und
  - passive Klassen / allg. Classifier
4. Identifikation verschiedenster Beziehungen zwischen Instanzen (bzw. Instanzmengen)
  - Navigierbarkeit
  - Abhängigkeit
  - ...
5. Beziehung zwischen Klassen
  - Spezialisierung / Generalisierung
  - abstrakte und konkrete Klassifizierer
6. Polymorphie von (getypten) Referenzen

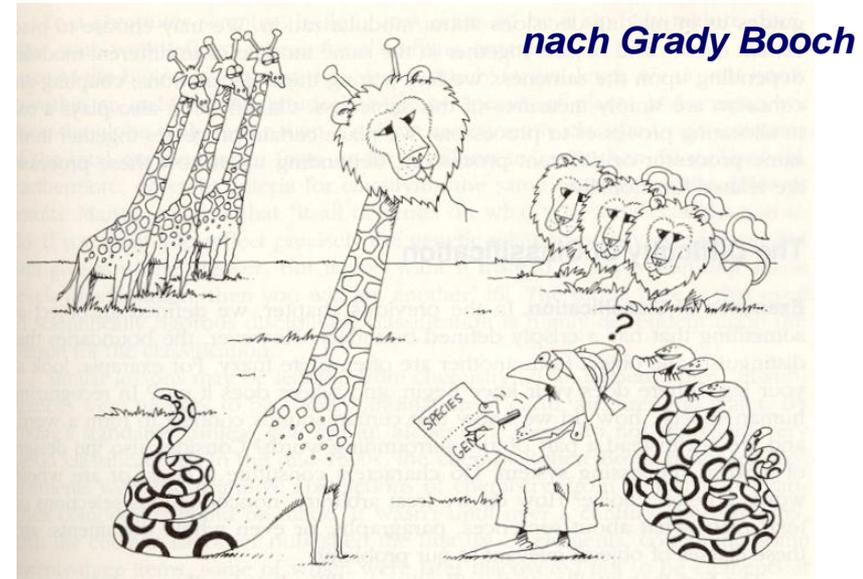
nach Grady Booch



# Modellierungsparadigmen

## Objektorientiertes Abstraktionskonzepte

1. eigenverantwortlich handelnde, interagierende Dinge (Objekte)
  - Zustand (Attribute)
  - individuelles Verhalten (Methoden, Dienste)
  - Identität (Referenz)
2. **Klassifikation (Definition von Objekten)**
3. Unterscheidung zw.
  - aktiven und
  - passive Klassen
4. Identifikation verschiedenster Beziehungen zw. Instanzen (bzw. Instanzmengen)
  - Navigierbarkeit
  - Abhängigkeit
  - ...
5. Beziehung zwischen Klassen
  - Spezialisierung / Generalisierung
  - abstrakte und konkrete Klassifizierer
6. Polymorphie von (getypten) Referenzen



## **KLASSIFIKATION VON OBJEKTEN**

Definition benötigter Objektklassen  
statt  
Definition einzelner Objekte

## **PRINZIP DER DATENKAPSLUNG**

# Modellierungsparadigmen

## Objektorientiertes Abstraktionskonzepte

1. eigenverantwortlich handelnde, interagierende Dinge (Objekte)
  - Zustand (Attribute)
  - individuelles Verhalten (Methoden, Dienste)
  - Identität (Referenz)
2. Klassifikation (Definition von Objekten)
3. Unterscheidung zw.
  - aktiven und
  - passive Klassen
4. Identifikation verschiedenster Beziehungen zw. Instanzen (bzw. Instanzmengen)
  - Navigierbarkeit
  - Abhängigkeit
  - ...
5. Beziehung zwischen Klassen
  - Spezialisierung / Generalisierung
  - abstrakte und konkrete Klassifizierer
6. Polymorphie von (getypten) Referenzen

## OBJEKTE AKTIVER KLASSEN

### ~ VORGÄNGE IN RAUM UND ZEIT

... agieren eigenständig und in Kooperation miteinander bei Austausch von Informationen /Daten (synchron /asynchron)  
statische oder dynamische Existenz

## OBJEKTE PASSIVER KLASSEN

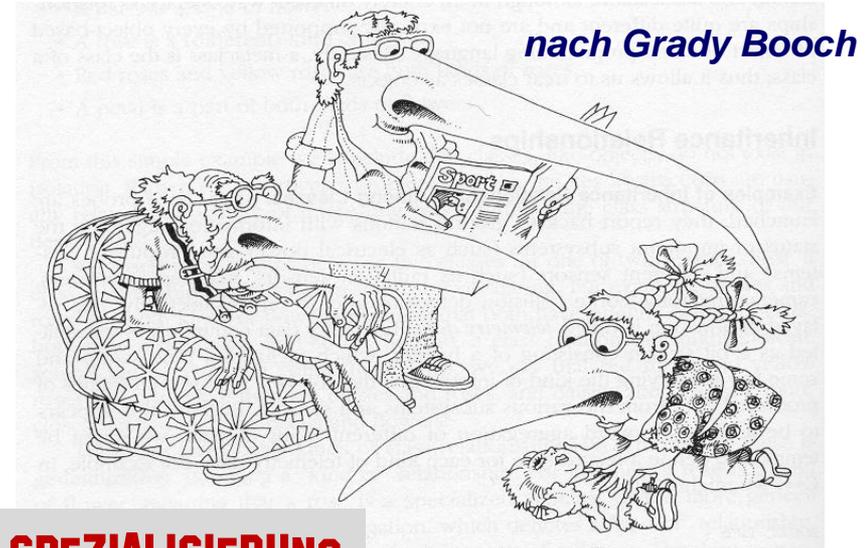
... werden von Objekten aktiver Klassen benutzt

*Bsp: ein Thread führt Operationen über Objekte von Java-Klassen aus*

# Modellierungsparadigmen

## Objektorientiertes Abstraktionskonzepte

1. eigenverantwortlich handelnde, interagierende Dinge (Objekte)
  - Zustand (Attribute)
  - individuelles Verhalten (Methoden, Dienste)
  - Identität (Referenz)
2. Klassifikation (Definition von Objekten)
3. Unterscheidung zw.
  - aktiven und
  - passive Klassen
4. Identifikation verschiedenster Beziehungen zw. Instanzen (bzw. Instanzmengen)
  - Navigierbarkeit
  - Abhängigkeit
  - ...
5. **Beziehung zwischen Klassen**
  - **Spezialisierung / Generalisierung**
  - **abstrakte und konkrete Klassifizierer**
6. Polymorphie von (getypten) Referenzen



## **SPEZIALISIERUNG**

bedeutet  
Wiederverwendung der  
gemeinsamen Basiskonzepte

- Attribute
- Verhalten / Methoden

# Modellierungsparadigmen

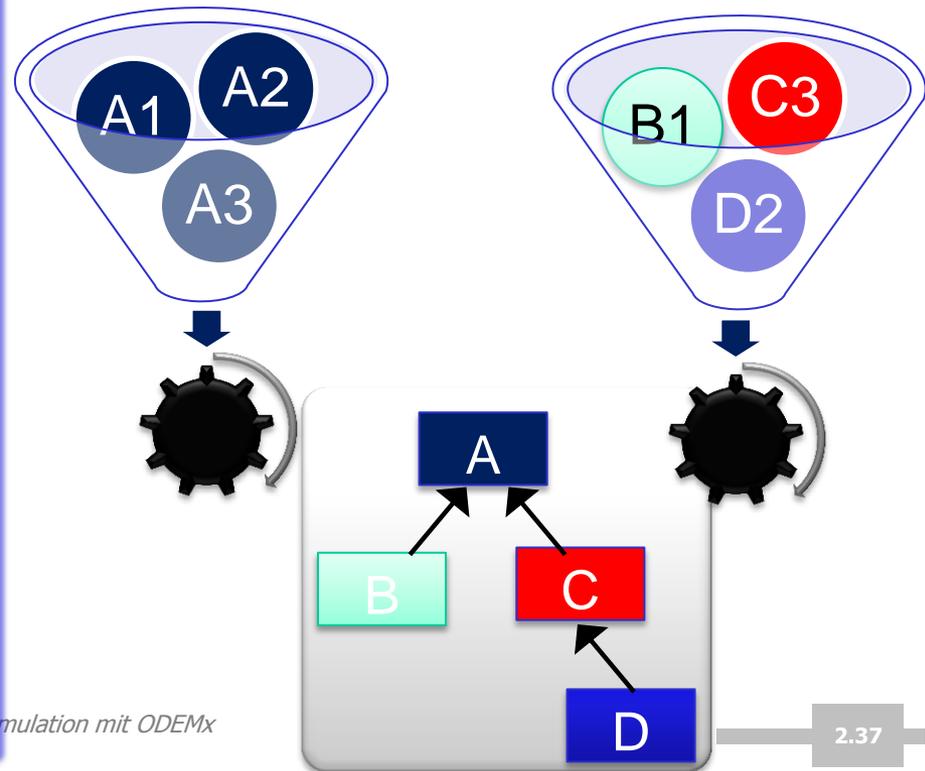
## Objektorientiertes Abstraktionskonzepte

1. eigenverantwortlich handelnde, interagierende Dinge (Objekte)
  - Zustand (Attribute)
  - individuelles Verhalten (Methoden, Dienste)
  - Identität (Referenz)
2. Klassifikation (Definition von Objekten)
3. Unterscheidung zw.
  - aktiven und
  - passive Klassen
4. Identifikation verschiedenster Beziehungen zw. Instanzen (bzw. Instanzmengen)
  - Navigierbarkeit
  - Abhängigkeit
  - ...
5. Beziehung zwischen Klassen
  - Spezialisierung / Generalisierung
  - abstrakte und konkrete Klassifizierer
6. **Polymorphie von (getypten) Referenzen**

## VIELGESTALTIGKEIT

überall dort, wo die Verarbeitung von Objekten einer (Basis-) Klasse definiert ist,

lässt sich auch eine Verarbeitung von Spezialisierungen organisieren



# Modellierungsparadigmen

## Objektorientiertes Abstraktionskonzepte

1. eigenverantwortlich handelnde, interagierende Dinge (Objekte)
  - Zustand (Attribute)
  - individuelles Verhalten (Methoden, Dienste)
  - Identität (Referenz)
2. Klassifikation (Definition von Objekten)
3. Unterscheidung zw.
  - aktiven und
  - passive Klassen
4. Identifikation verschiedenster Beziehungen zw. Instanzen (bzw. Instanzmengen)
  - Navigierbarkeit
  - Abhängigkeit
  - ...
5. Beziehung zwischen Klassen
  - Spezialisierung / Generalisierung
  - abstrakte und konkrete Klassifizierung
6. **Polymorphie von (getypten) Referenzen**

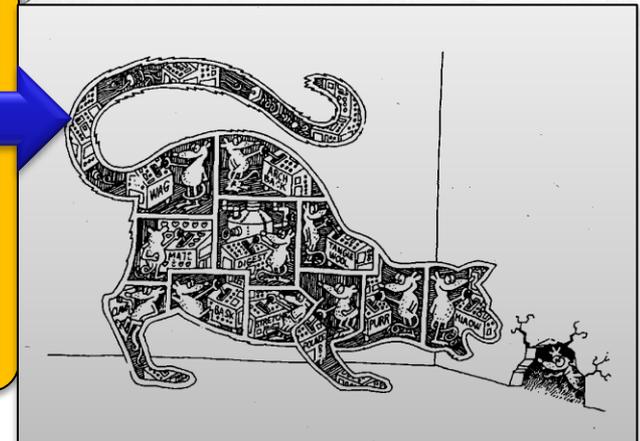
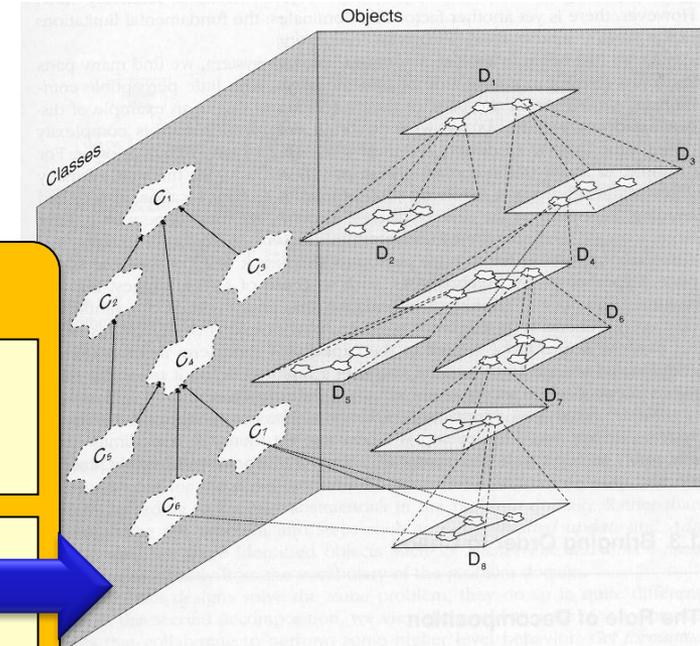
zusätzlich ...

Gruppierung von Modellelementen

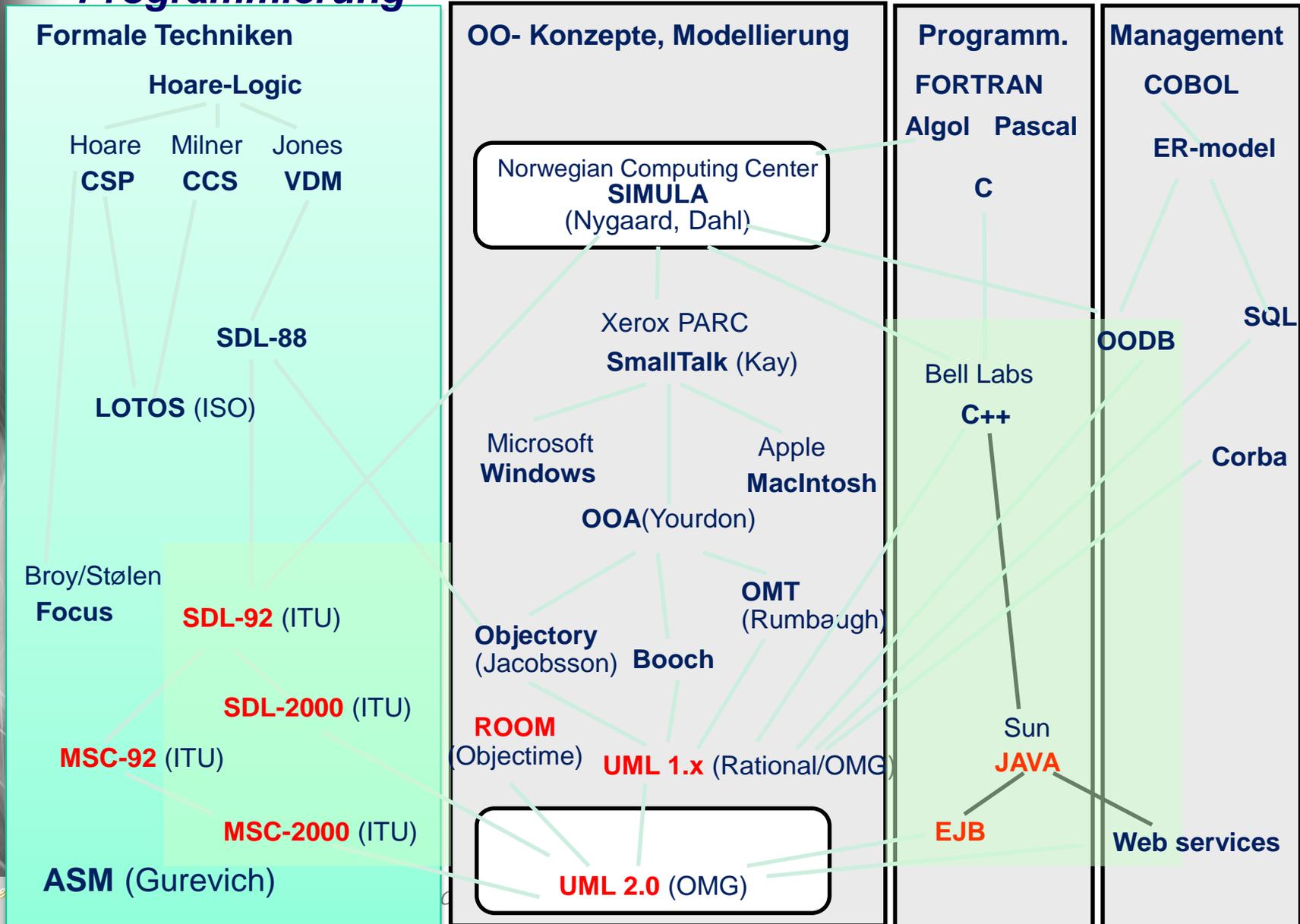
Komposition/  
Dekomposition

Nebenläufigkeit/  
Parallelität/  
Synchronisation

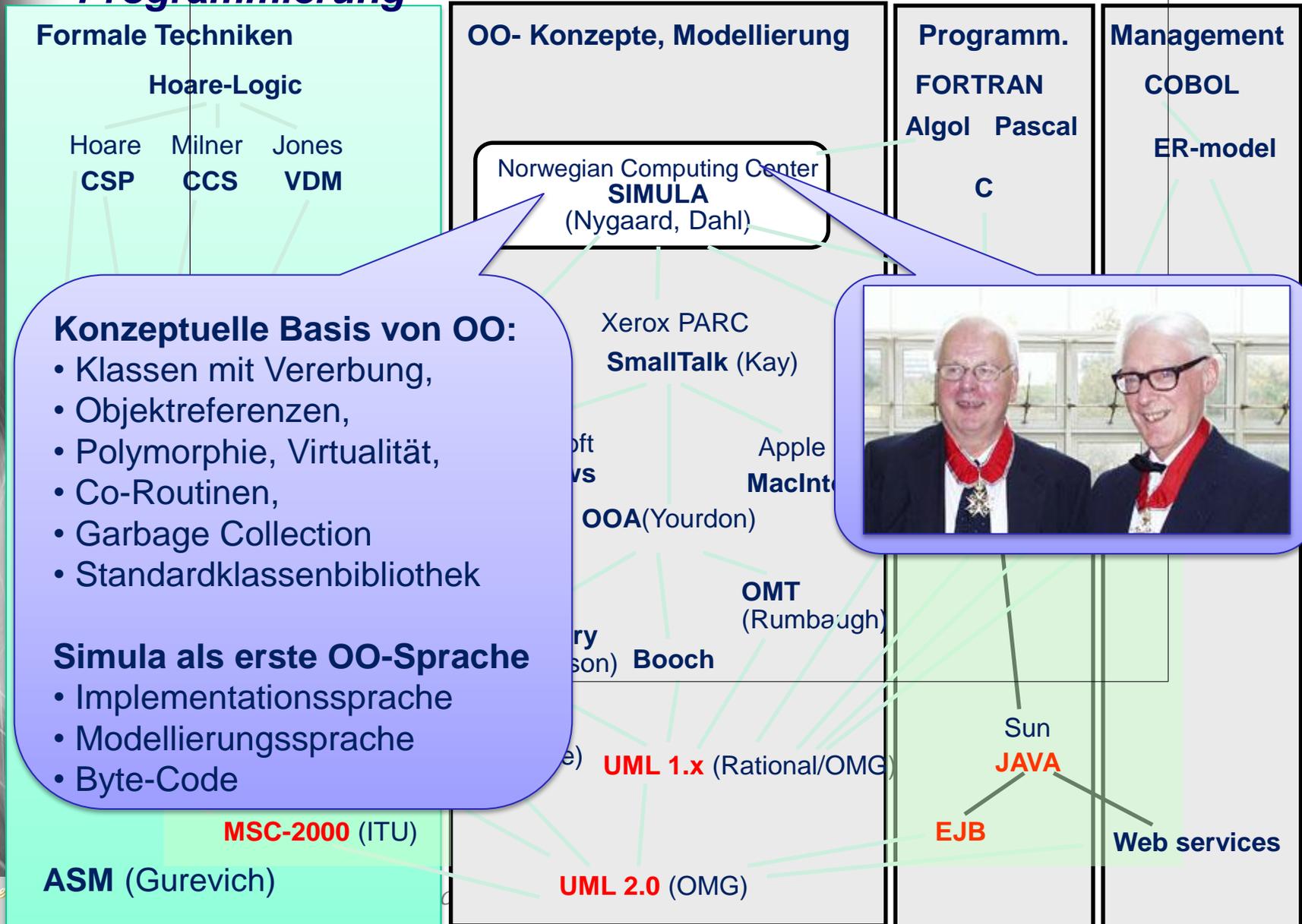
zeitdiskretes/  
zeitkontinuierliches  
Verhalten



# Wurzeln der Objektorientierten Modellierung/ Programmierung



# Wurzeln der Objektorientierten Modellierung/ Programmierung



## Konzeptuelle Basis von OO:

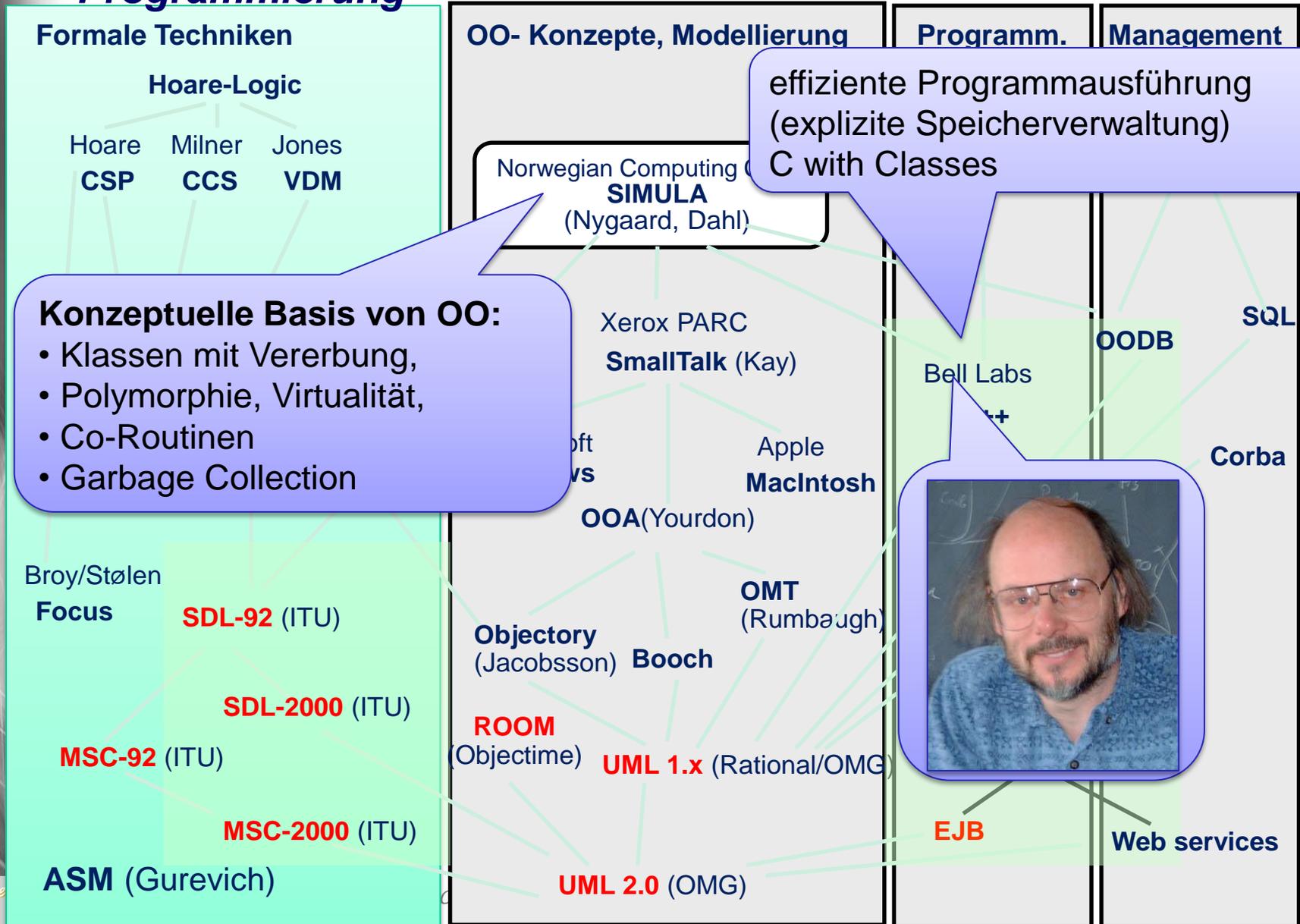
- Klassen mit Vererbung,
- Objektreferenzen,
- Polymorphie, Virtualität,
- Co-Routinen,
- Garbage Collection
- Standardklassenbibliothek

## Simula als erste OO-Sprache

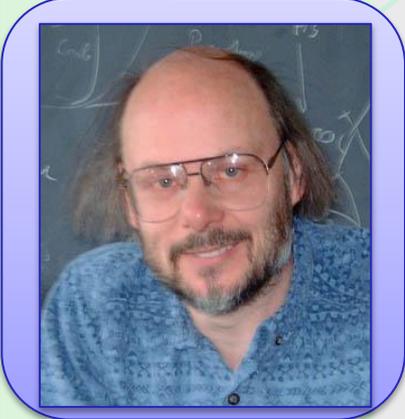
- Implementationssprache
- Modellierungssprache
- Byte-Code



# Wurzeln der Objektorientierten Modellierung/ Programmierung



effiziente Programmausführung  
(explizite Speicherverwaltung)



# 1. 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. Scheduler für zeitdiskrete und zeitkontinuierliche Systemmodelle
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

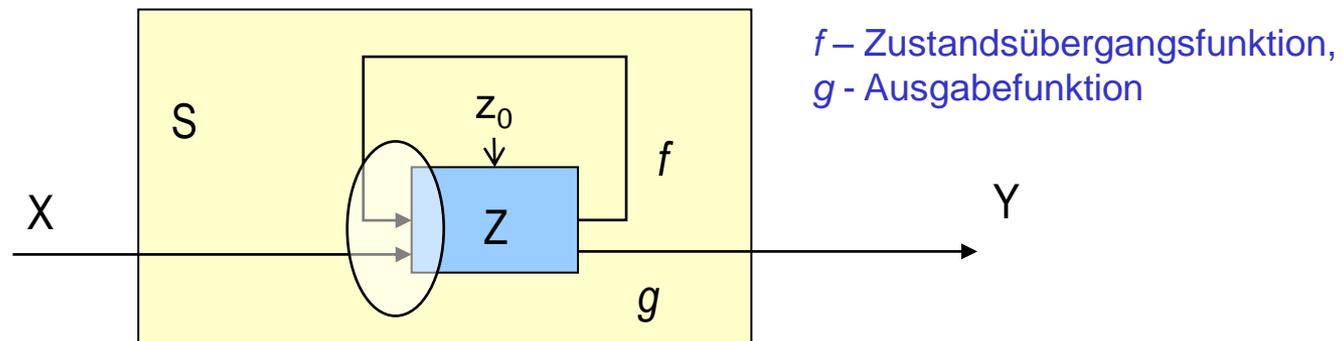
- Modellbeschreibungsgrößen, aus denen sich der Zustand eines Systems **vollständig** ergibt (Gedächtnis eines Systems)  
→ Basis der Verhaltensbeschreibung
- Zustandsgrößen sind voneinander **unabhängig**  
→ eine Zustandsgröße kann nicht als Kombination anderer Zustandsgrößen dargestellt werden
- sind **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  $t_0 \quad t_1 \quad t_2 \quad t_3$   
 $z_0 \rightarrow z_1 \rightarrow z_2 \rightarrow z_3 \rightarrow \dots$  Zustandsentwicklung in Abhängigkeit der Zeit



$f$  – Zustandsübergangsfunktion,  
 $g$  – Ausgabefunktion

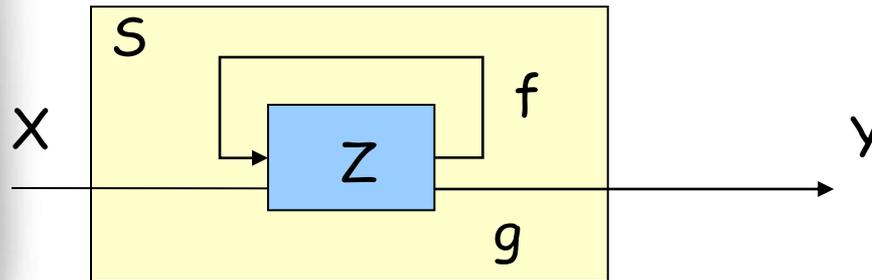
Eingabe X  
 Zuwachs  
 im Zeitintervall  $(t_k, t_{k+1}]$

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

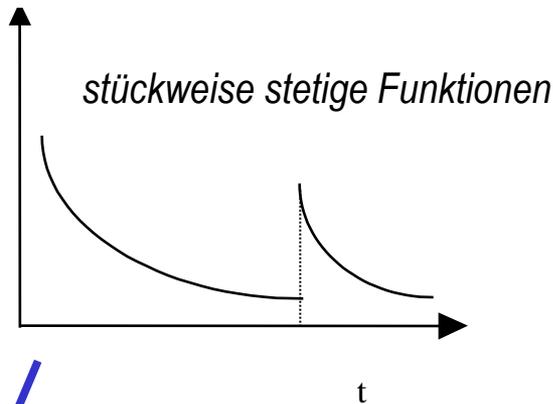


$$S = (Z, z_0, X, Y, f, g, \text{time})$$

- $Z$  Menge der möglichen Zustände
- $z_0 \in Z$  Anfangszustand
- $X$  Menge der möglichen Eingaben
- $Y$  Menge der möglichen Ausgaben
- **time** Zeitbasis als  $(T, \leq, t_0)$  mit
  - Menge möglicher Zeitpunkte  $T$ ,
  - einer Ordnungsrelation  $\leq$  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



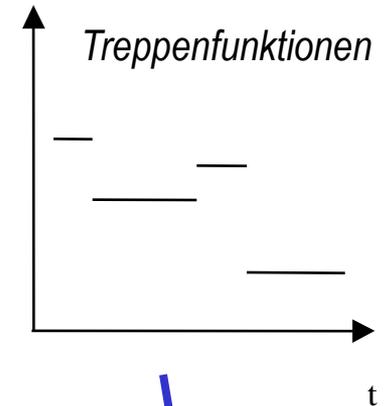
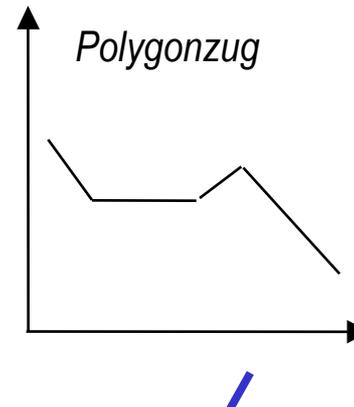
$$z'(t) = f(z(t), x(t), t)$$

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

# Verhaltensklassen zeitkontinuierlicher Art

Differentialgleichungen

Klassifikationsmöglichkeiten

n-ter Ordnung (erster Ordnung)

partielle  
Differentialgleichungen

gewöhnliche  
Differentialgleichungen

nichtlineare  
Differentialgleichungen

lineare  
Differentialgleichungen

Randwertaufgaben

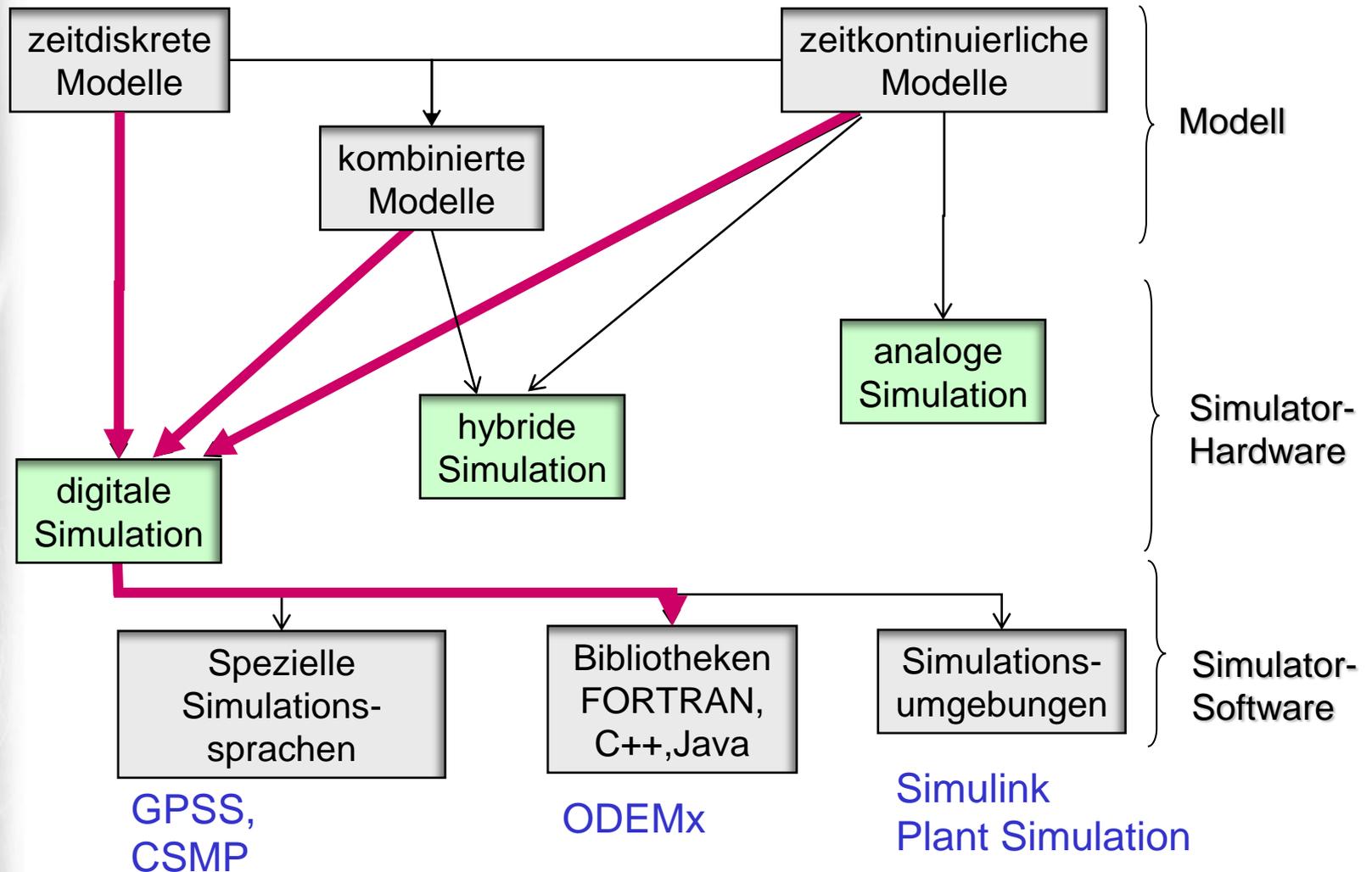
Anfangswertaufgaben

- 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)

# Klassifizierung von Modellen und Simulationsverfahren

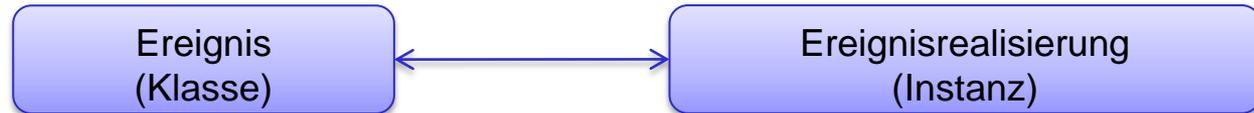


# 1. *Einführung*

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. Scheduler für zeitdiskrete und zeitkontinuierliche Systemmodelle (Grundprinzip)
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 (→ Zeitergebnisse)
  - Zustandsbedingungen (→ Zustandsergebnisse)
  - 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 Modellbeschreibungsgroßen*)  
beschreibt in ihrer Belegung den Systemzustand zum aktuellen Zeitpunkt
- **Systembewertungsgrößen**  
(z.B. *statistische Kenngrößen*)
- **Uhr**  
(*Modellzeit, dimensionslos, monoton wachsend*)
- **Ereigniskalender**
  - Umgang mit **Gleichzeitigkeit** von Ereignissen
  - **Zeitfortschritt:**  
Bestimmung des nächsten Ereignisses im Kalender,  
Setzen der Uhr  
(aktuelle Modellzeit:= Ereigniszeit)

Zustandsänderungen finden immer nur zu diskreten Zeitpunkten statt



Ereignis-Charakteristik

Ereigniszeit

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

(Event) Scheduler

Zeitfortschritts-  
realisierung

Realisierung des  
aktuellen  
Ereignisses