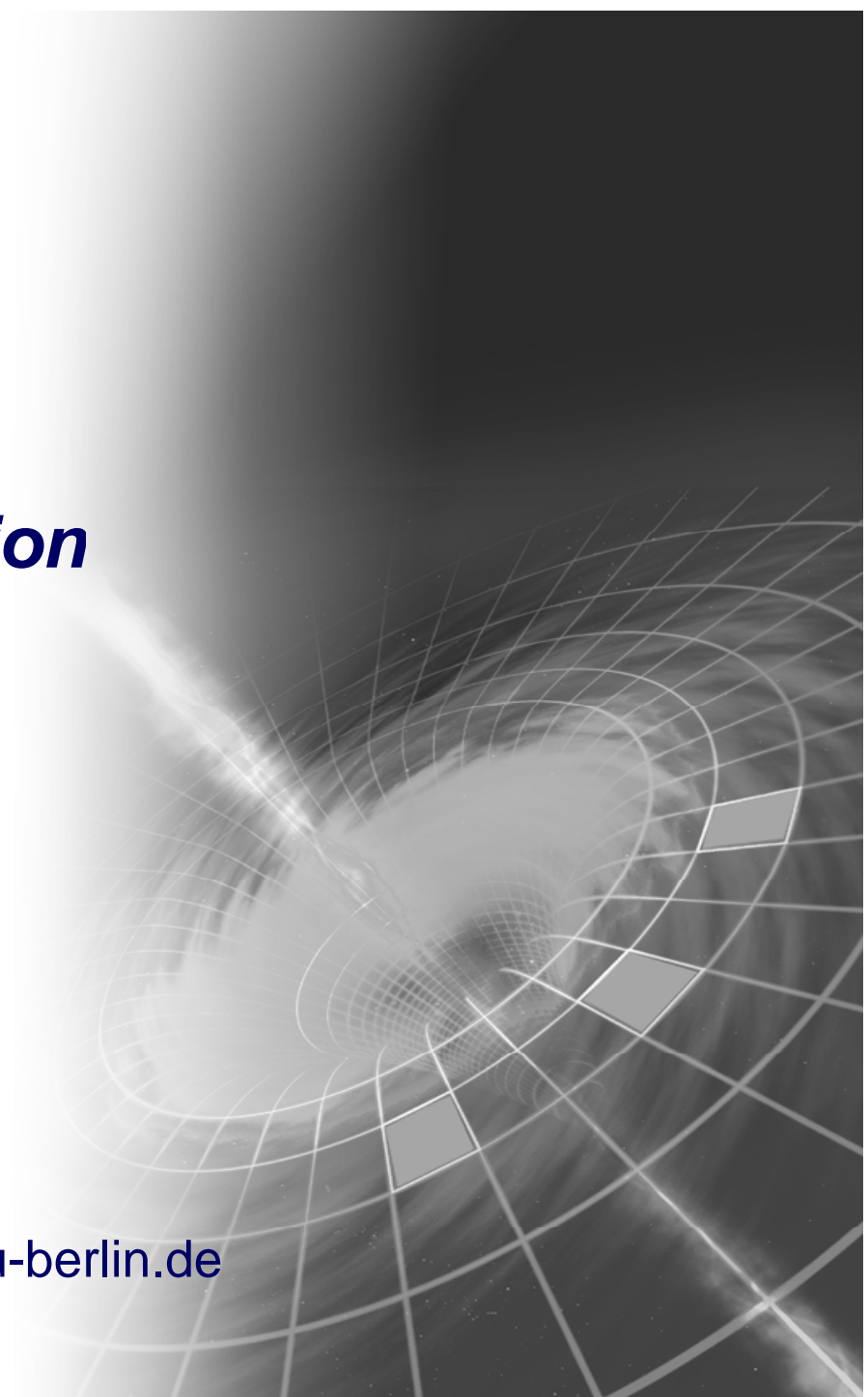


Kurs OMSI im WiSe 2010/11

Objektorientierte Simulation mit ODEMx

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

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

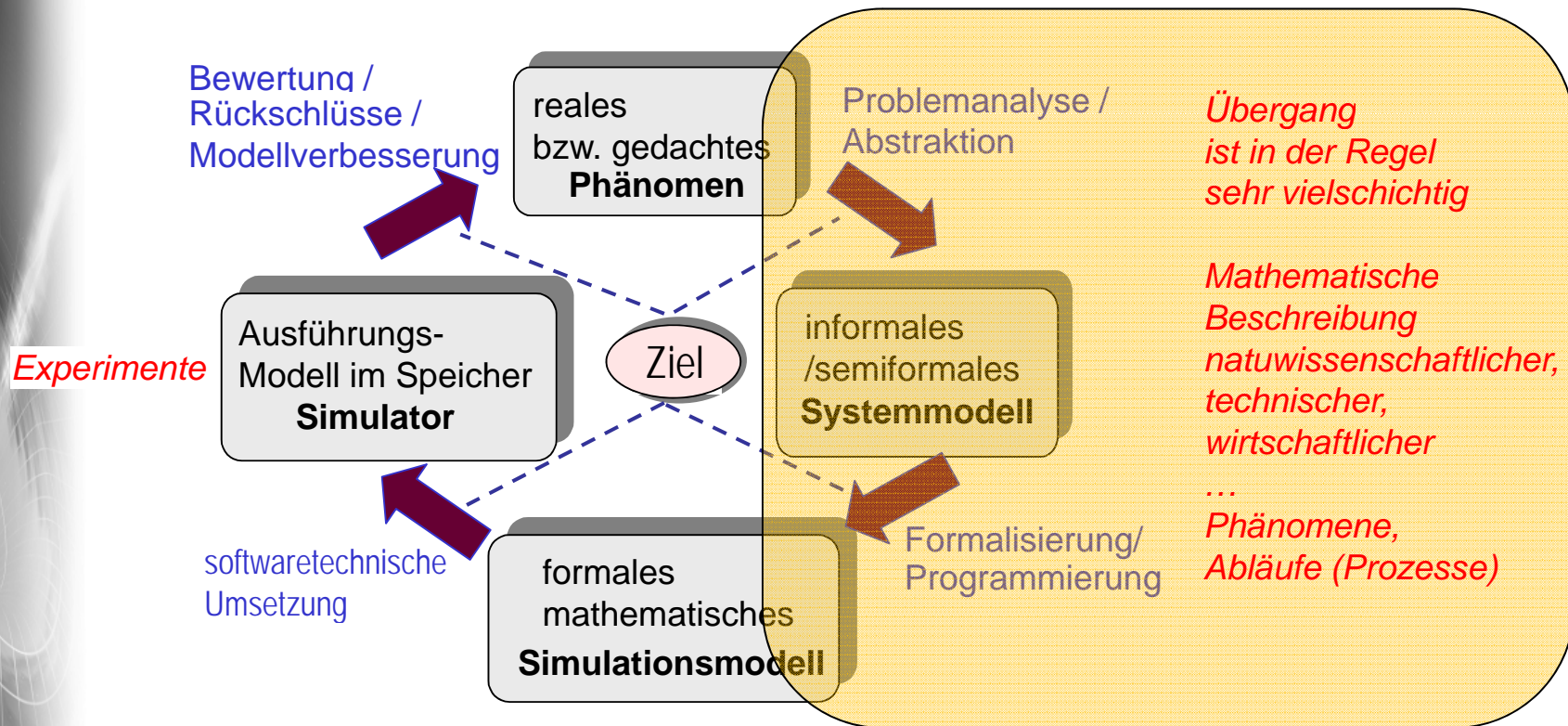


1. Einführung

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. M&S eines Niedertemperaturofens

Vorgehensweise bei der Systemsimulation

Experimentieren mit ausführbaren Modellen auf dem Computer
- anstatt mit Originalen -



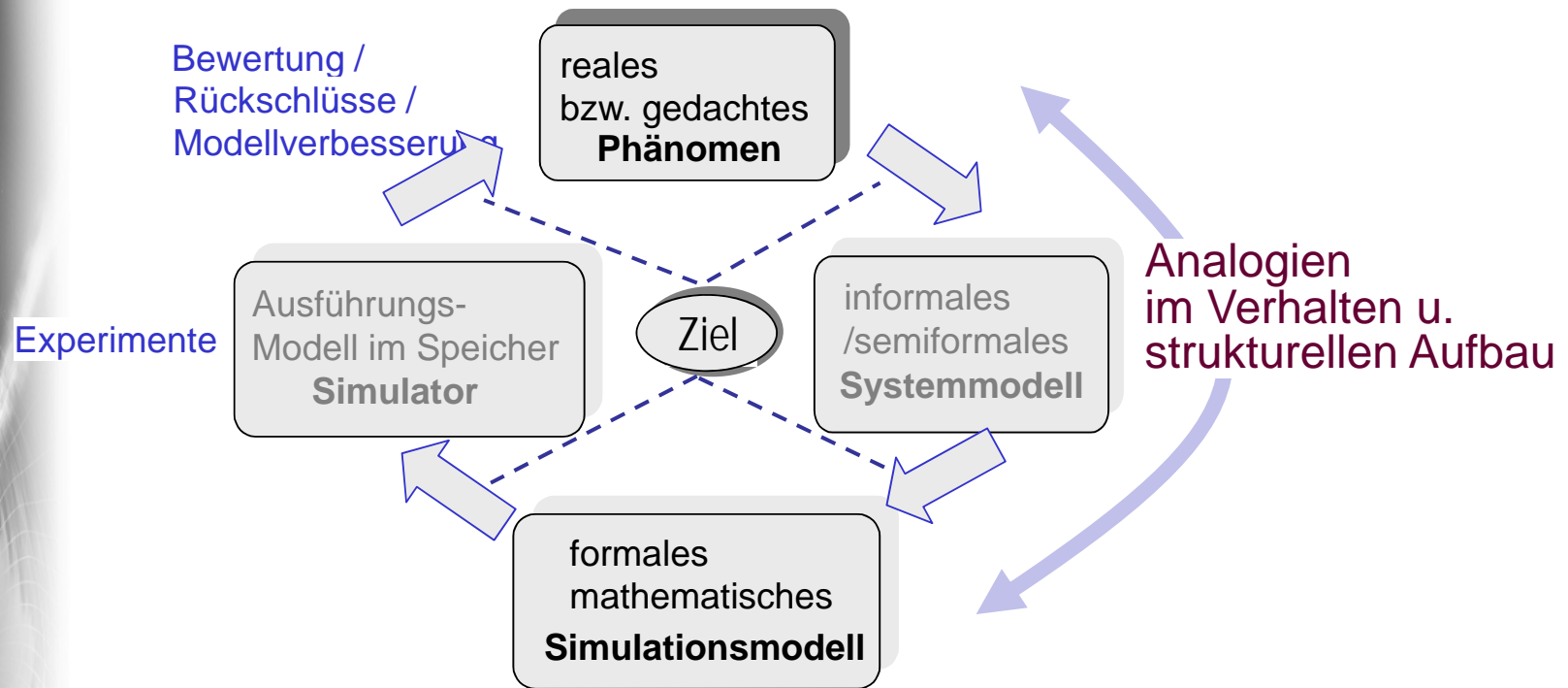
Besonderheit:

Zustandsgrößen ändern sich zeitabhängig
(kontinuierlich, diskret / ereignishaft)



dynamische Systeme

Bedeutung von Analogien



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. M&S eines Niedertemperaturofens

Präzisere Begriffsbestimmung

Original

- Ausschnitt einer gedachten oder real existierenden Welt **als System**
(Systemzweck, Abgrenzung zur Systemumgebung, Systemstruktur, Systemverhalten)
- Originale als statische oder dynamische **Systeme**

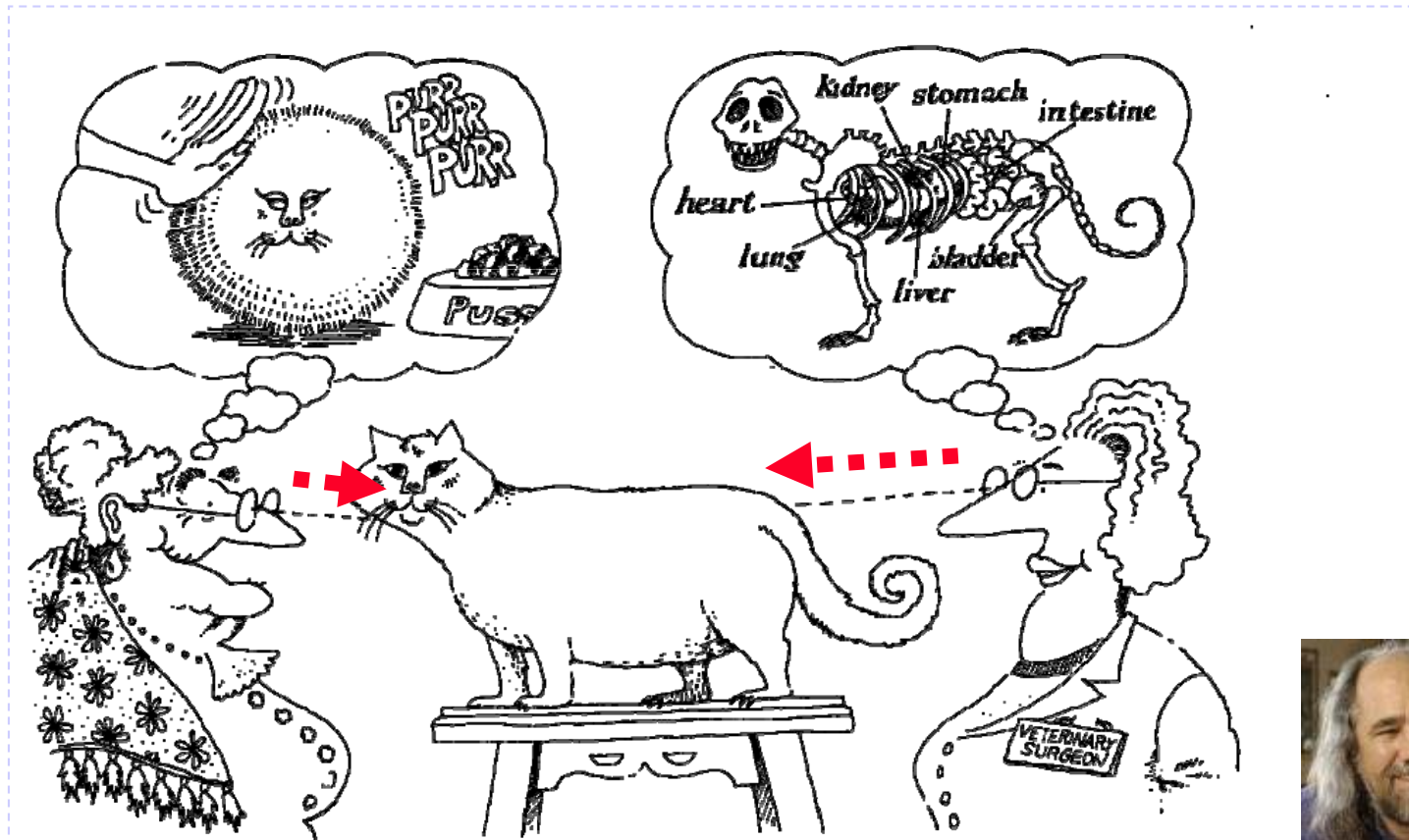
Modelle

- sind als Abstraktionen von Originalen
Abstraktionen= Vereinfachungen aus einer bestimmten Sicht mit einer bestimmten Zielstellung
- Modelle sind Abstraktionen kompletter Systeme oder einzelner Systemelemente
- Systemmodelle= Struktur- und Verhaltensmodelle des Systems

Gibt es perfekte Modelle?

- Modelle werden aus einer bestimmten Sicht bei Verfolgung eines bestimmten Untersuchungsziels abgeleitet
- kein einziges Modell, keine einzige Sicht ist ausreichend um ein komplexes System zu erfassen
→ es gibt kein Modell an sich
- Entscheidung, welche Modelle erzeugt werden, hat großen Einfluss auf die Modelluntersuchung
- jedes Modell kann in unterschiedlichen Abstraktionsniveaus und aus unterschiedlichen Blickwinkeln dargestellt werden
- die besten Modelle sind realitätsnah
- **Gefahr:** bereits bewährte Modelle werden für Untersuchungen mit anderem Untersuchungsziel eingesetzt

Modelle in unterschiedlichen Sichten auf einen Realitätsausschnitt



© Booch: Object-Oriented Design with applications
Benjamin/Cummings Publishing Company Inc.

Das jeweilige Untersuchungsziel bestimmt Sicht und Abstraktionsgrad

Begrenztheit von Modellen

Norbert Wiener (1894-1964)

Begründer der Kybernetik, Kommunikation, Steuerung
und Regelung



Was ist das beste Modell einer Katze ?

scherzhaft: „Das beste Modell einer Katze ist ...

eine Katze.

Am besten dieselbe Katze.“



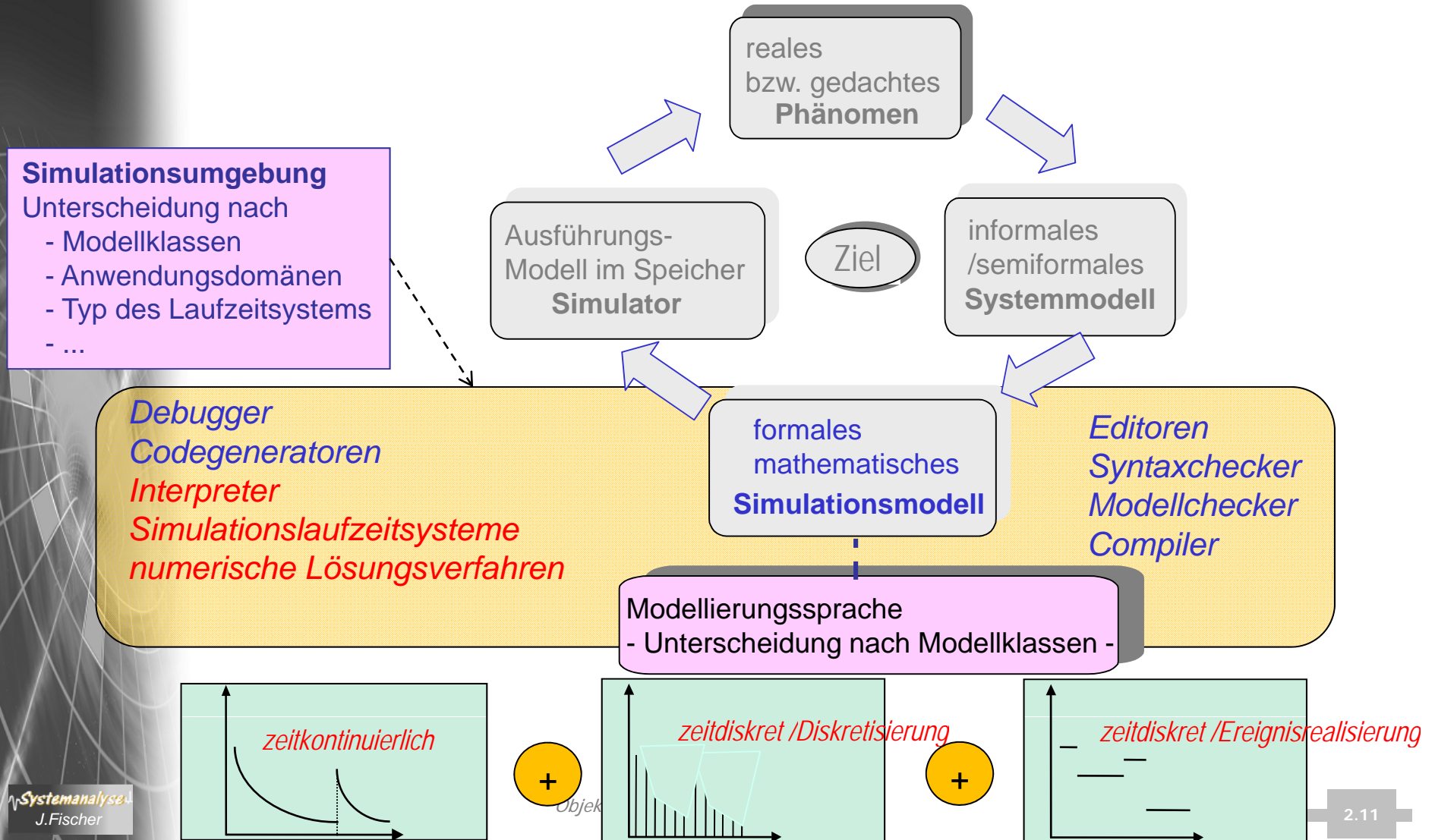
Modelle ersetzen ihre Originale nur bedingt

1. Einführung

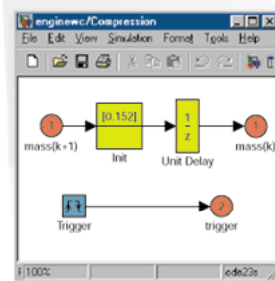
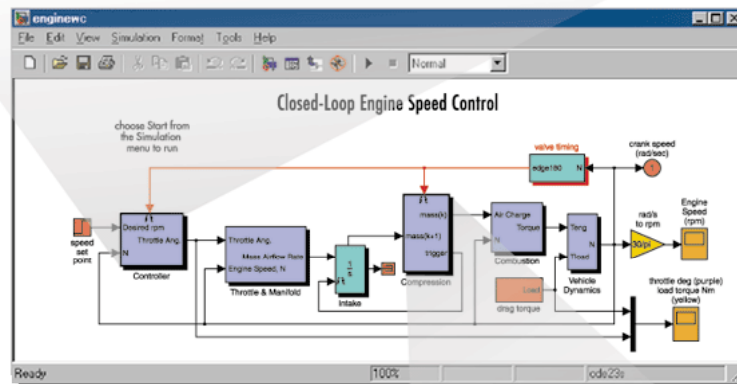
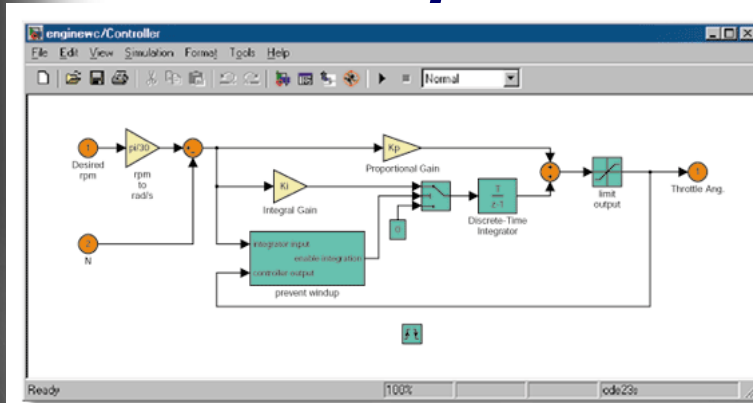
1. Systemsimulation – was ist das?
2. Ein Blick zurück in die Anfänge
3. Modelle und Originale
4. Modellierungssprachen, Simulationsumgebungen
5. Beispiele aus der aktuellen Forschung
6. Paradigma der objektorientierten Modellierung
7. Klassifikation dynamischer Systeme
8. M&S eines Niedertemperaturofens

Modellierungssprachen und Simulationsumgebung

Zustandsänderungen kontinuierlich oder/und diskret in Raum und Zeit

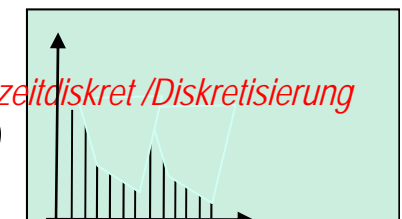
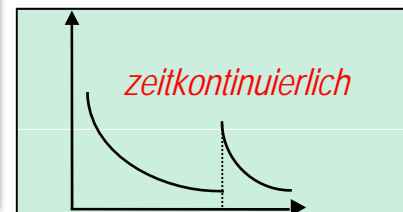


Beispiele domänenspezifische Modellierungssprachen

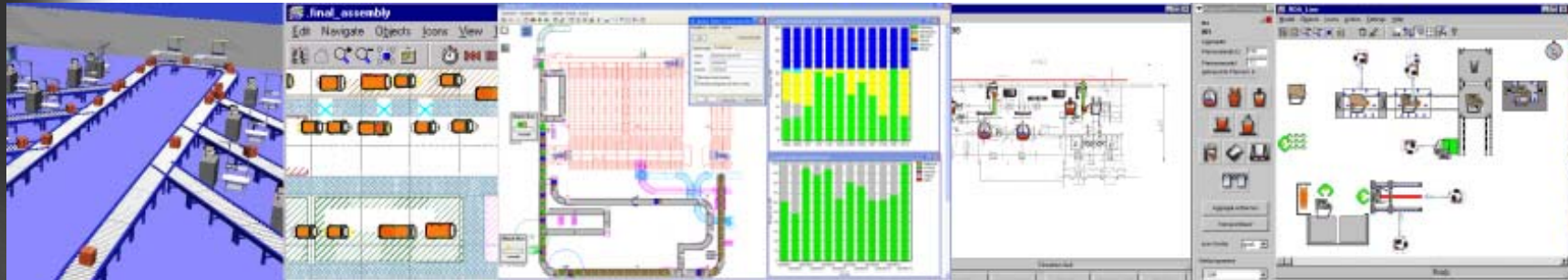


Simulink

- hierarchische graphische Modellierung
- kontinuierliche u. diskreter Schaltblöcke
- S-Functions: eigener Code u. MATLAB
- für einzelne Domänen (wie **mechanische**, **elektrische** oder **hydraulische** Systeme) stehen spezielle Zusätze zur Verfügung, welche die Modellierung von physikalischen Systemen zusätzlich vereinfachen

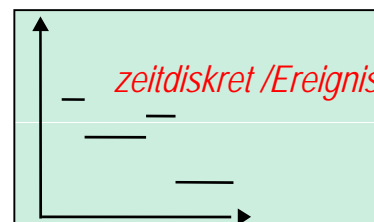
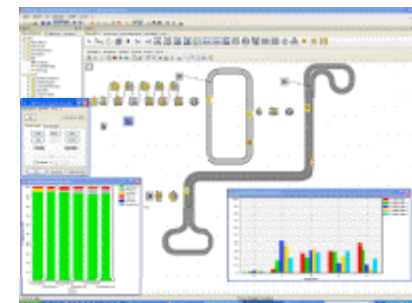


Beispiele domänenspezifische Modellierungssprachen



Plant Simulation

- graphische Modellierung, Simulation, Visualisierung
- Optimierung von Logistik- und Geschäftsprozessen



zeitdiskret / Ereignisrealisierung

Solution Partner

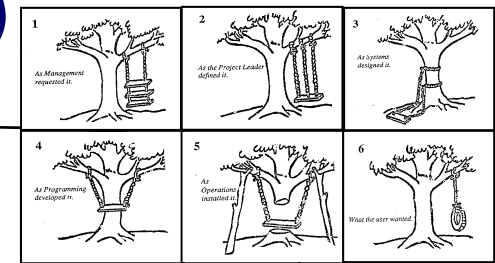
PLM

SIEMENS

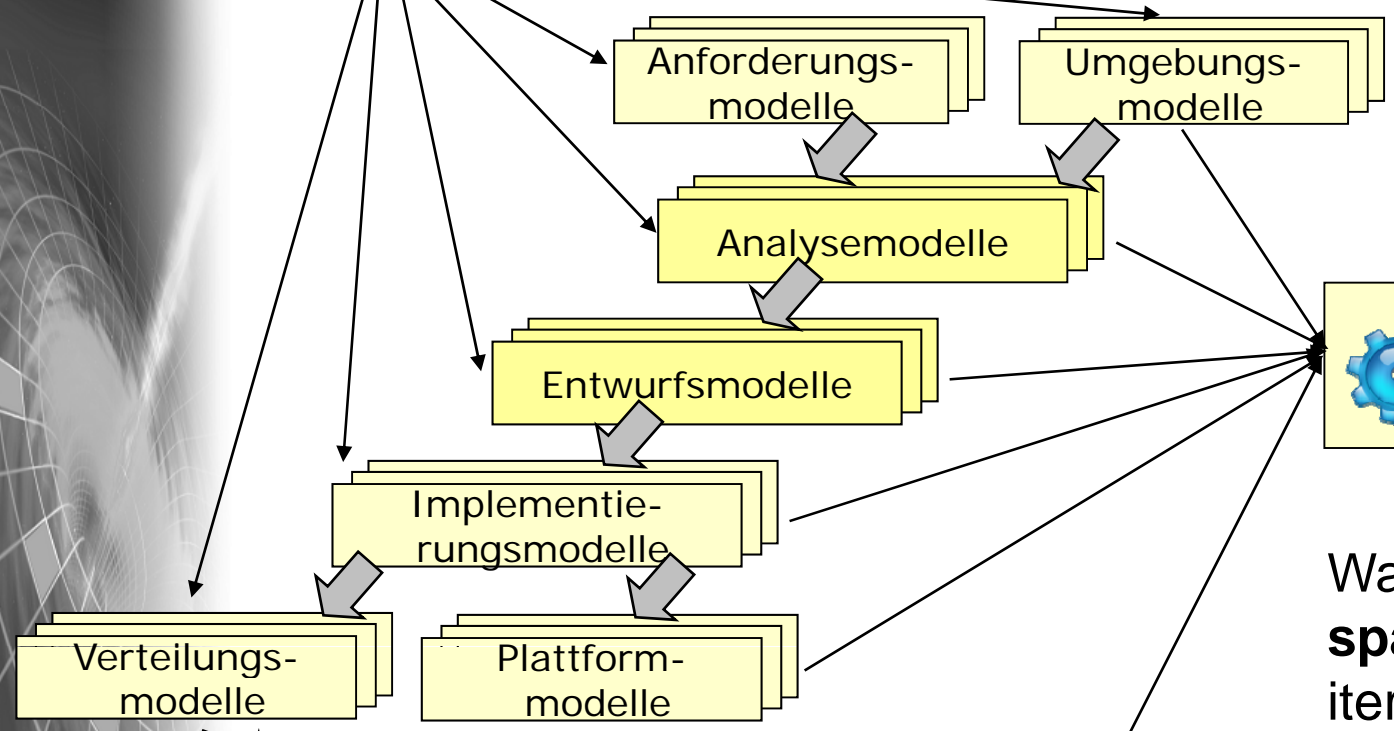
Frage:

Wie ist der Stand des Modelleinsatzes bei der Software-Produktion?

Modellbasierte Software-Entwicklung verteilter Systeme (vereinfacht)

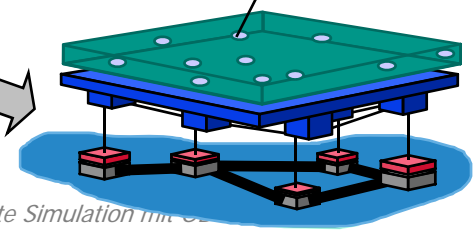


Anforderungen

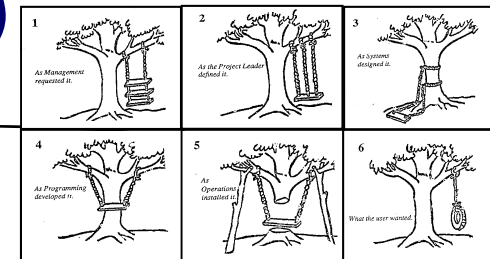


Modell-Checker
Simulator
 Test, Wartung

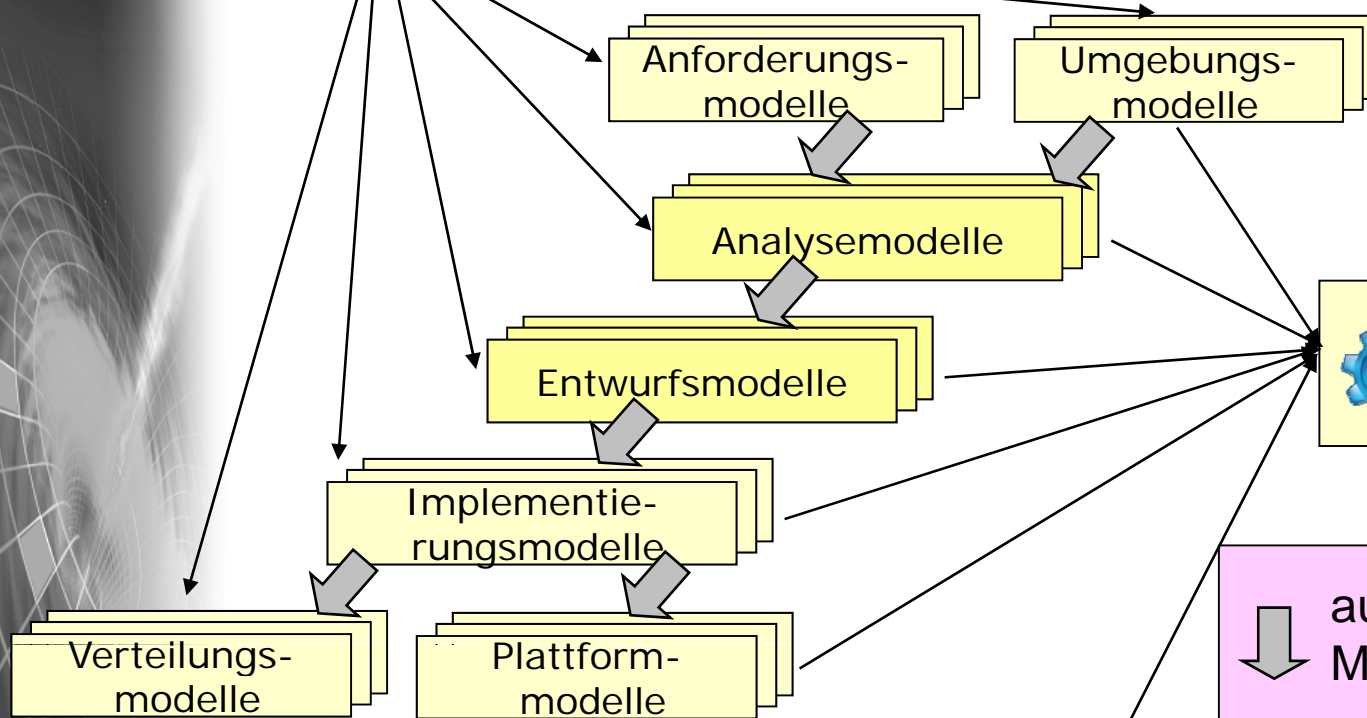
Wasserfallmethode
später verbessert:
 iterativ, inkrementell



Modellgetriebene Software-Entwicklung verteilter Systeme (vereinfacht)



Anforderungen

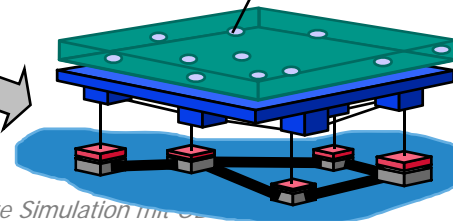


Modell-Checker
Simulator
 Test, Wartung

↓ automatisierte
 Modelltransformation

Zielcode

Binärcode-Komponenten



Modellgetriebene Software-Entwicklung

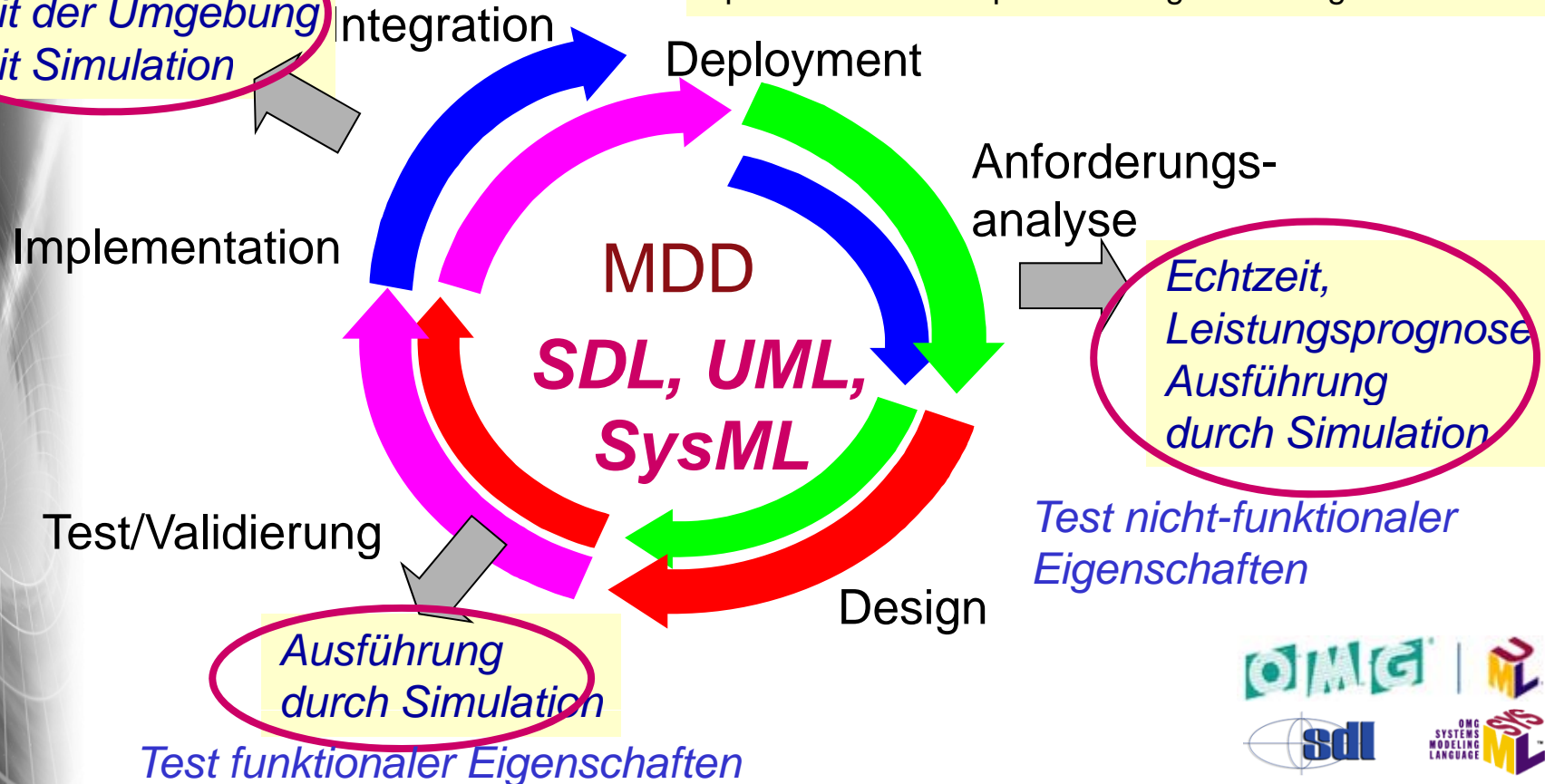
spiralförmig, inkrementell & iterativ

Test funktionaler und
nicht-funktionaler
Rückkopplungen

MDD:= Model Driven Development

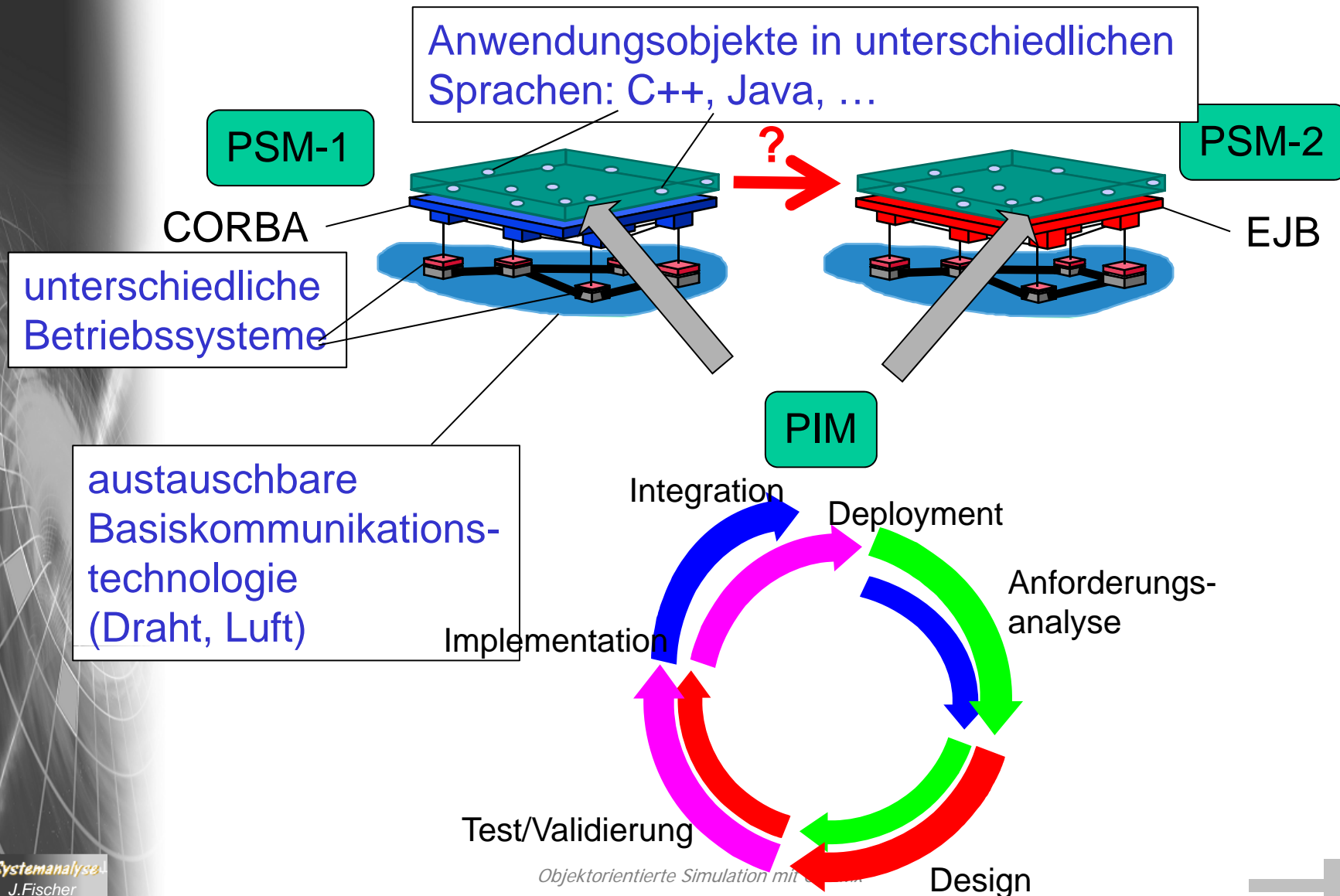
- SW-Entwicklung ist modellzentriert (Modelle begleiten ges. SW-Lebenszyklus)
- automatische Transformationen für Modellübergänge
- spezifische Analysen (Checker, Simulatoren, ...)
- partielle oder komplette Codegenerierung

Wechselwirkung
mit der Umgebung
mit Simulation



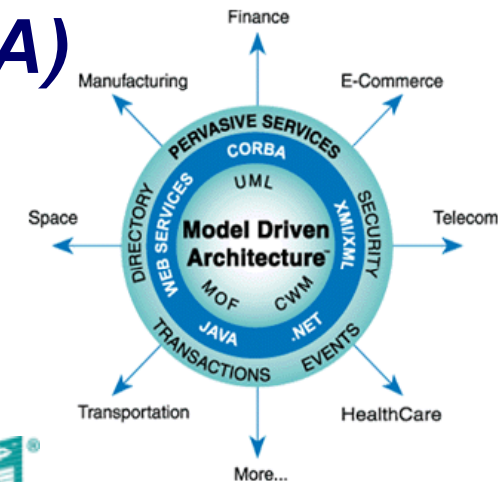
Modellgetriebene Software-Entwicklung

spiralförmig, inkrementell & iterativ



Model-Driven Architecture (MDA)

- ... fasst die gesammelten Erkenntnisse über
 - SW-Modelle, Modellierung und Transformation,
 - angereichert mit einer Reihe weiterer Standardszu einer offiziell anerkannten **Spezifikation** zur modellgetriebenen Softwareentwicklung zusammen



- **Ziel:**
Abbildung des gesamter Softwareentwicklungsprozesses
 - von der Fachdomäne des späteren Anwenders,
 - über die Anforderungsanalyse
 - bis hin zur Implementierung des Zielsystems mit allen seinen Schichten)in Modellen,
so dass das System selbst über Modelltransformation, erzeugt werden kann
- Sind alle Transformatoren geschrieben, so erreicht man auf diesem Weg eine hohe Wiederverwendbarkeit und Wartbarkeit
- Darüber hinaus gilt die MDA als ein **möglicher Schlüssel** zur anforderunggetriebenen Softwareentwicklung, da die technischen Aspekte weitestgehend vollständig von den inhaltlichen (semantischen) Aspekten getrennt werden.

Spezielle Modelle der MDA

- Plattform Independent Models (PIM):
die Modellierung der **Fachdomäne** (also der Zielwelt) ist **vollständig plattformunabhängig** zu gestalten,
es sind also ausschließlich rein fachliche Aspekte zu betrachten und zu modellieren.
- **Plattform Description Models (PDMs)** sind (Meta-)Modelle,
die die **Zielplattform** des Systems beschreiben.
Über die Kombination von einem PIM, also einer formalen semantischen Beschreibung der Zusammenhänge und Abläufe mit einem PDM kann letztendlich über Modelltransformation das Zielsystem
(welches im Sinne der MDA auch wieder nur ein Modell ist) generiert werden
- **Plattform Specific Model (PSM)**
ist das Ergebnis der Modelltransformation

Model-Driven Architecture (Leitsätze)

- **Formalisierung** ist ein wichtiger Baustein für ein erfolgreiches **Qualitätsmanagement** in Softwareprojekten. Speziell in den Bereichen der Anforderungs- und Systemanalyse besteht häufig noch ein hohes Optimierungspotential.
- Ein möglicher Weg, um den **Formalisierungsgrad** von Projektinformationen zu erhöhen, ist die Verwendung von **formal eindeutigen Modellen**. Für den erfolgreichen Einsatz von Modellen ist es jedoch unabdingbar, die **Syntax und die Semantik der Modelle über Metamodelle** exakt festzulegen. Ist dies einmal geschehen, ergibt sich meist eine deutliche Steigerung der Qualität wie auch der Effizienz in der Projektarbeit.
- Über den gezielten **Einsatz von Metamodellen** in der Softwareentwicklung können große Teile der Prozessaktivitäten automatisiert werden. Dennoch muss berücksichtigt werden, dass die Formalisierung eines Softwareentwicklungsprozesses **nicht in einem Schritt** erfolgen kann. Sie sollte vielmehr als ein **iterativer Prozess** verstanden werden, in dem die entstehenden Metamodelle von Projekt zu Projekt immer weiter verfeinert werden müssen.

Fazit

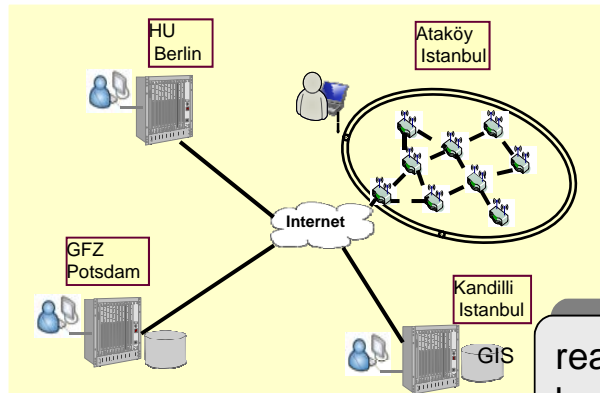
- Modellierung in allen Wissenschaftsdisziplinen das zentrale Paradigma zum Verständnis komplexer realer oder hypothetischer Systeme
(auch in bestimmten Bereichen der Informatik)
- In der SW-Entwicklung lange Zeit nicht hoffähig :
Alternative: von der Idee direkt zum gut dokumentierten Quellcode
aber: Komplexität der Systeme bereiten praktische Probleme
- **Achtung**: MDD verlangt nicht nur Konzepte,
sondern integrierte Werkzeugunterstützung

→OMSI: Technologien zum Bau effizienter Simulatoren dynamischer Systeme

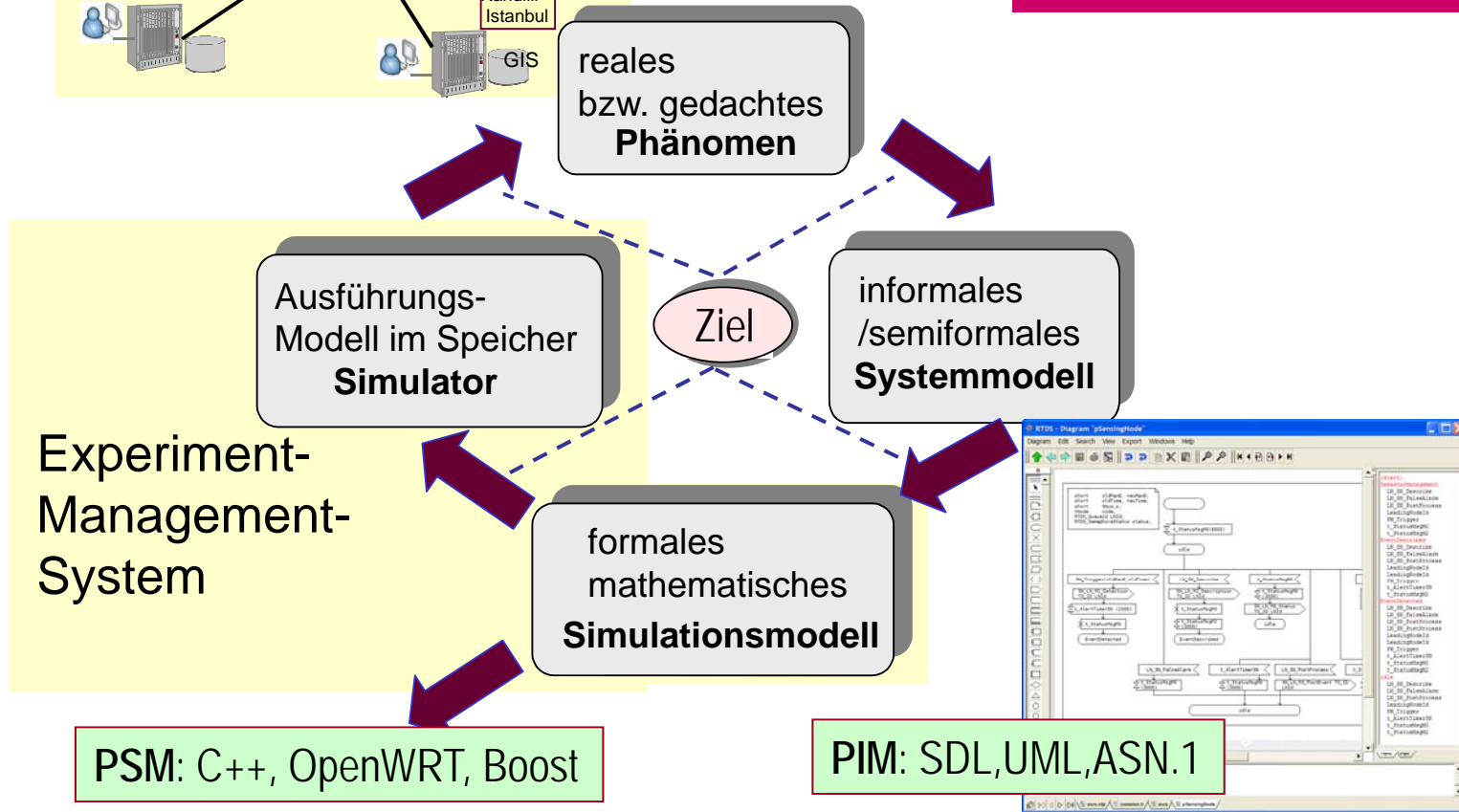
1. *Einführung*

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. M&S eines Niedertemperaturofens

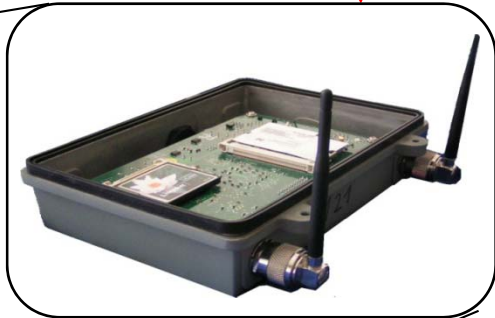
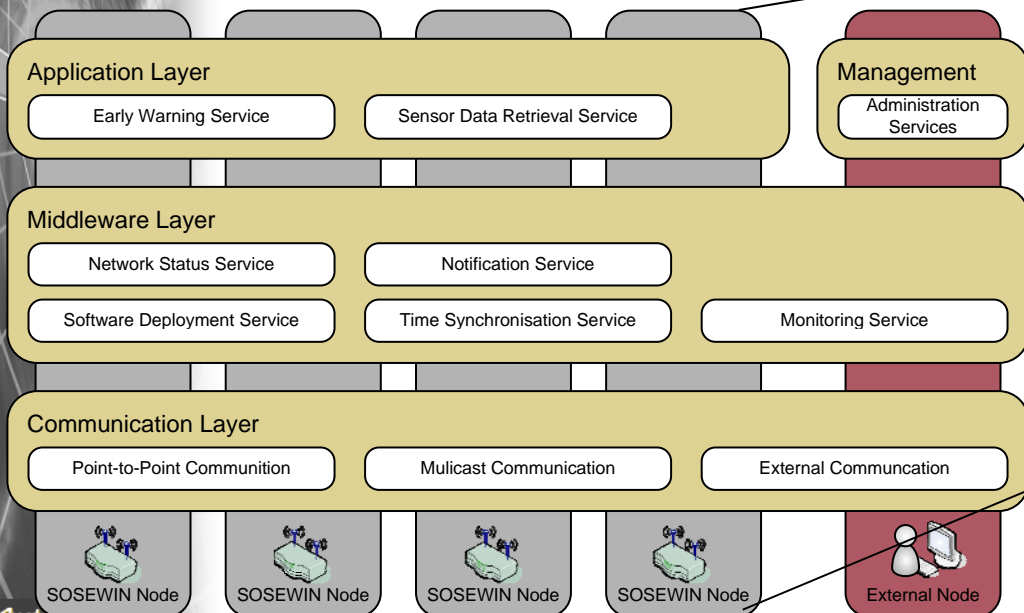
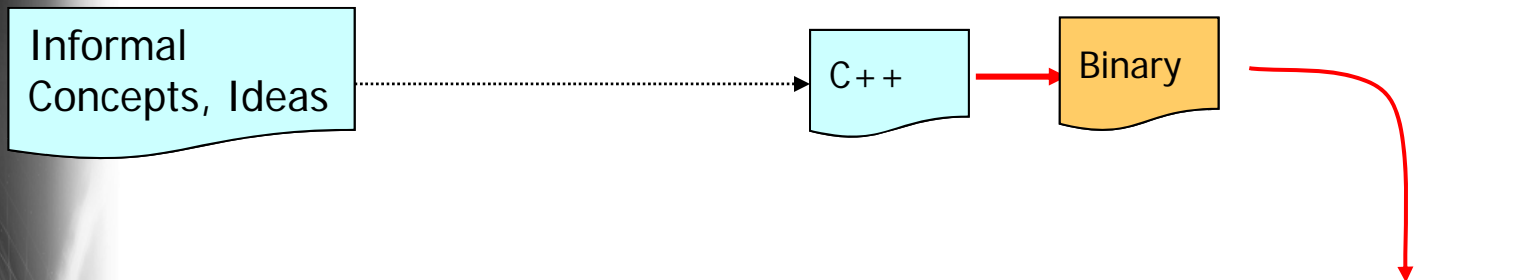
1. Erdbebenfrühwarnsystem



Sonderfall:
Modell und Original-
Teile
liegen als Software
vor

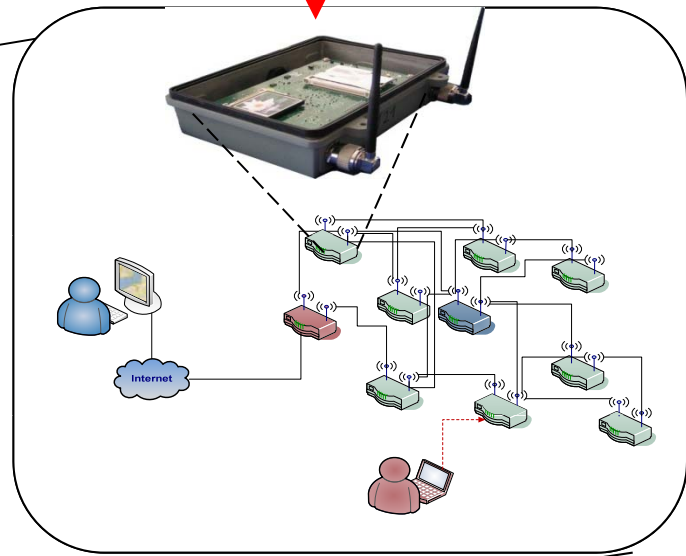
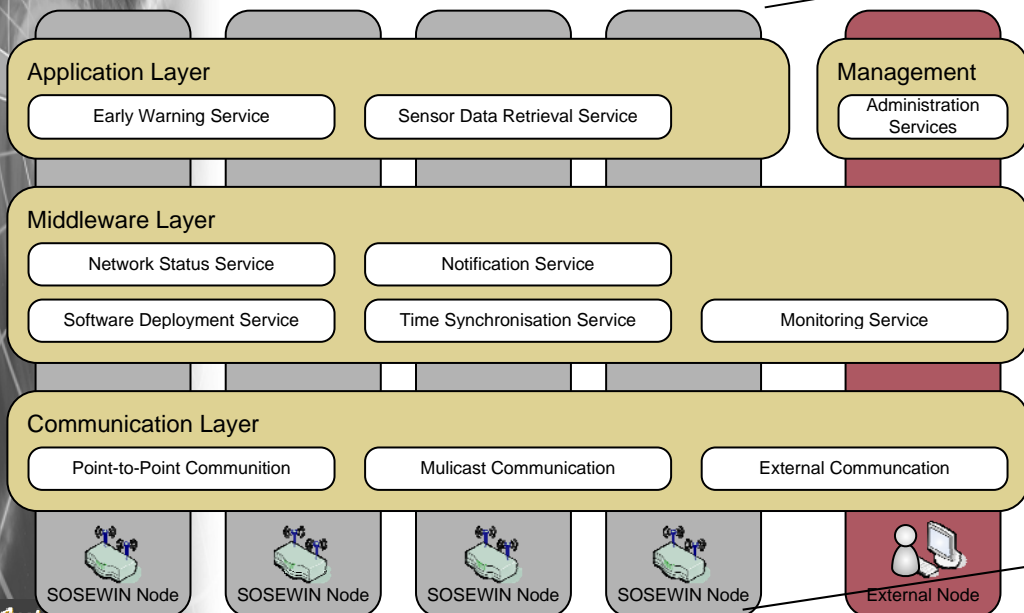
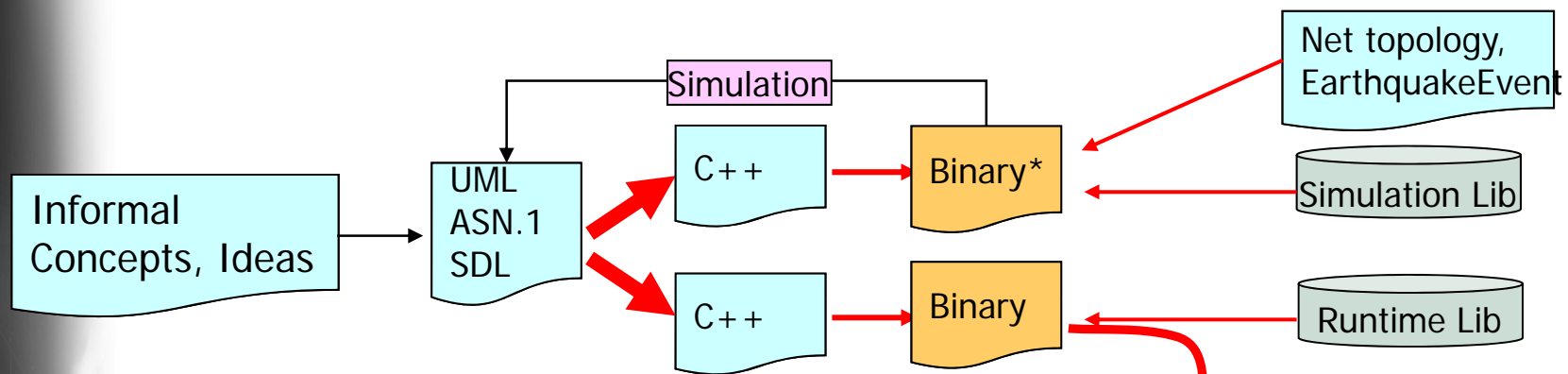


Traditional Software Development



test of a single node is not sufficient and of a complete system problematic

Our Model-Driven Approach

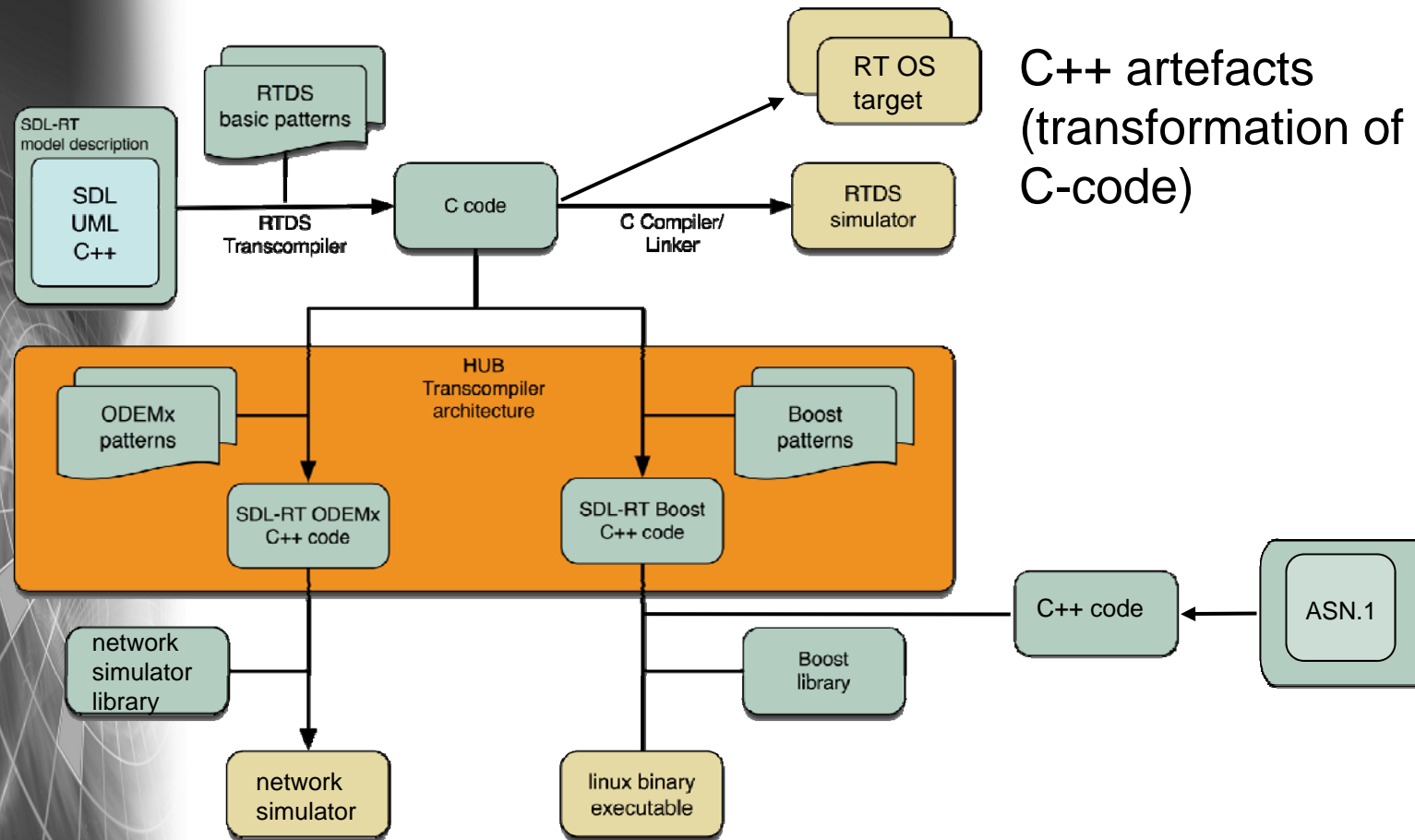


simulation of a complete system simplifies testing

Extension of PragmaDevs SDL/RT Compiler

ODEMx/Boost Pattern

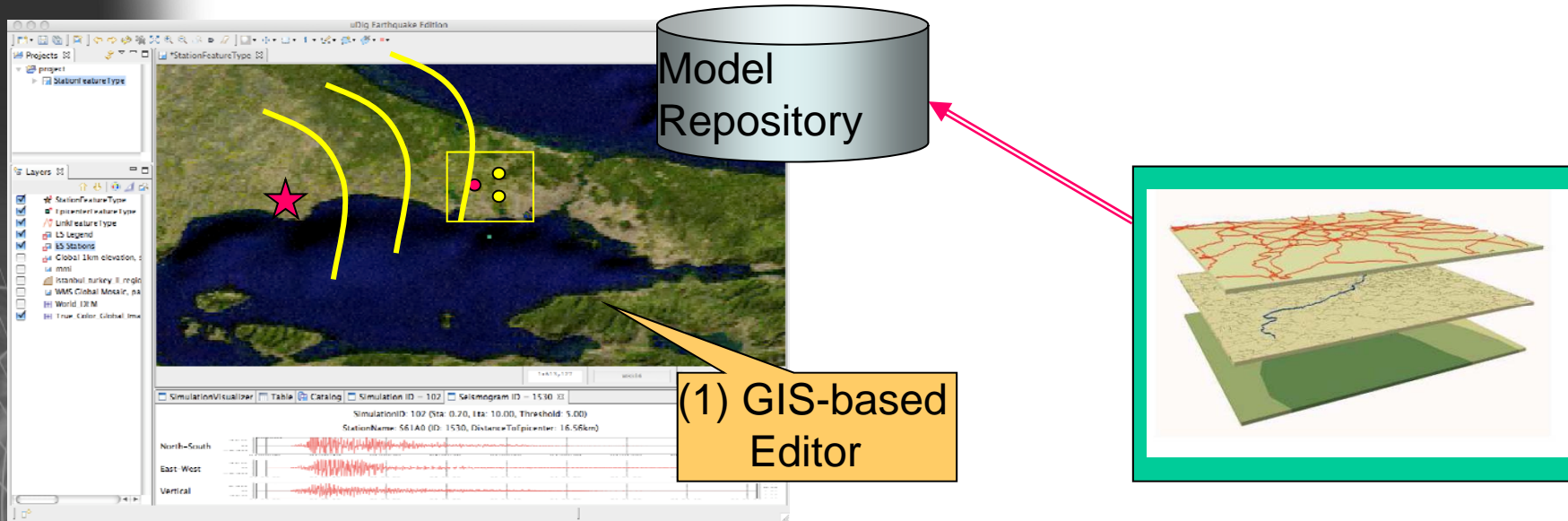
C++ artefacts
(transformation of labelled
C-code)



Benötigte Simulatoren

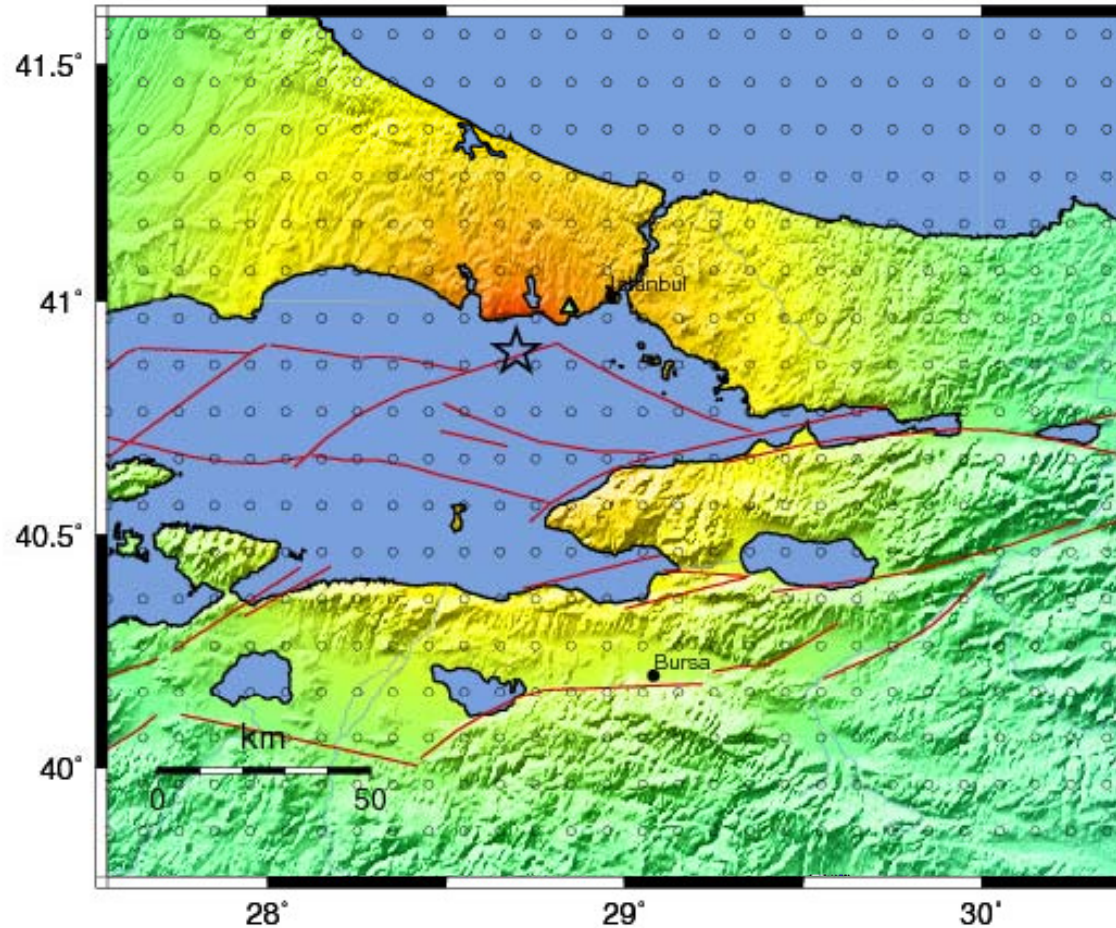
- Erdbebenwellenausbreitung
 - Epizentrum, Tiefe, Stärke, Bruchzonenbeschreibung
- Knotenfunktionalität
 - Unterschiedlicher Netzwerkschichten bei Variation von Routing-Protokollen
 - Unterschiedlicher Topologien
 - Unterschiedlicher Störungseinflüsse
- bei Einbindung in ein Geo-Informationssystem

Models, Target Components, and Tools



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

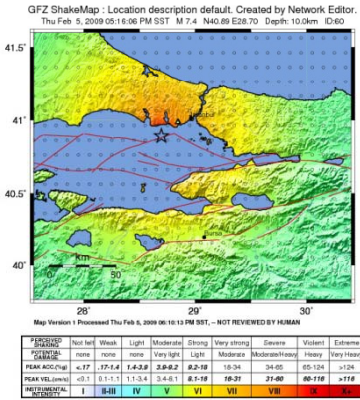
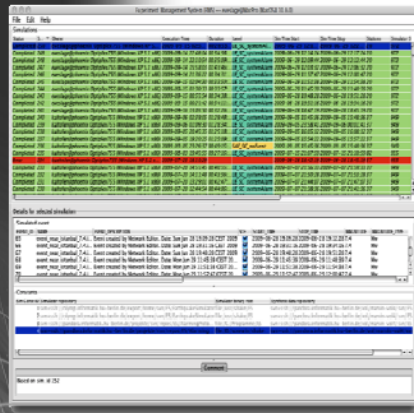


Map Version 1 Processed Thu Feb 5, 2009 06:10:13 PM SST, -- NOT REVIEWED BY HUMAN

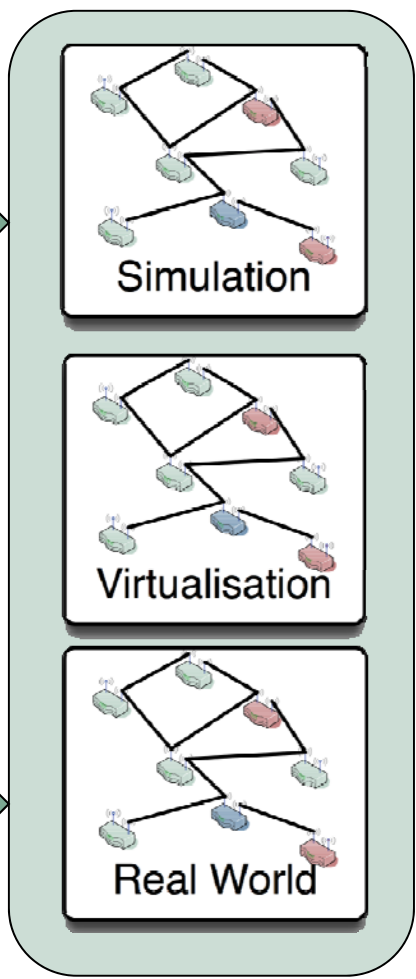
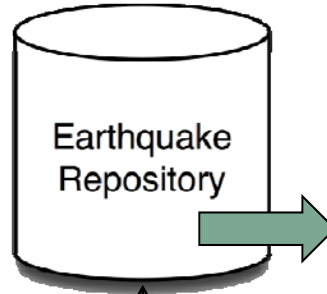
PERCEIVED SHAKING	Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
POTENTIAL DAMAGE	none	none	none	Very light	Light	Moderate	Moderate/Heavy	Heavy	Very Heavy
PEAK ACC.(%g)	<.17	.17-1.4	1.4-3.9	3.9-9.2	9.2-18	18-34	34-65	65-124	>124
PEAK VEL.(cm/s)	<0.1	0.1-1.1	1.1-3.4	3.4-8.1	8.1-16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



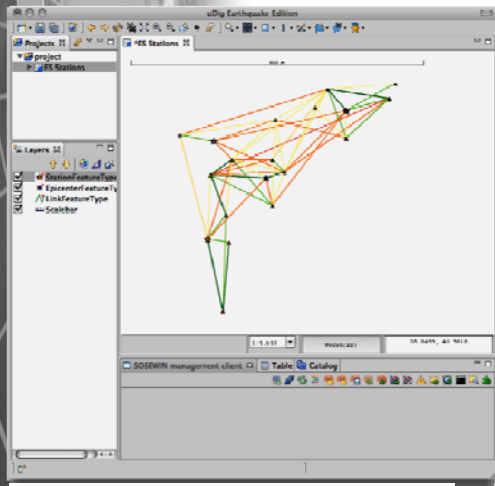
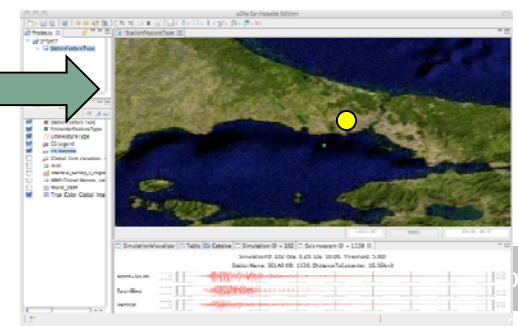
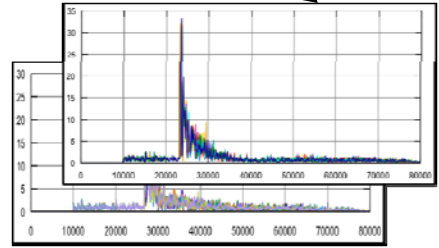
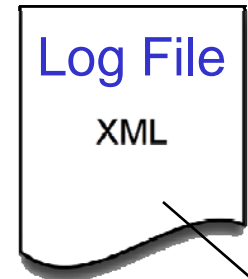
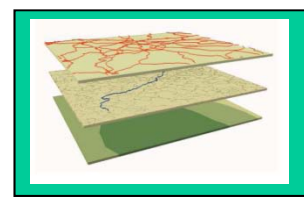
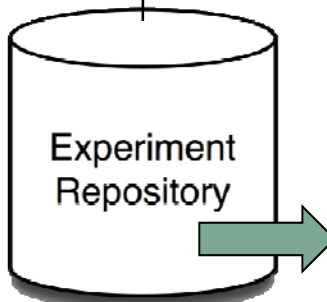
Experiment Management System



Executables



Earthquake Synthesizer

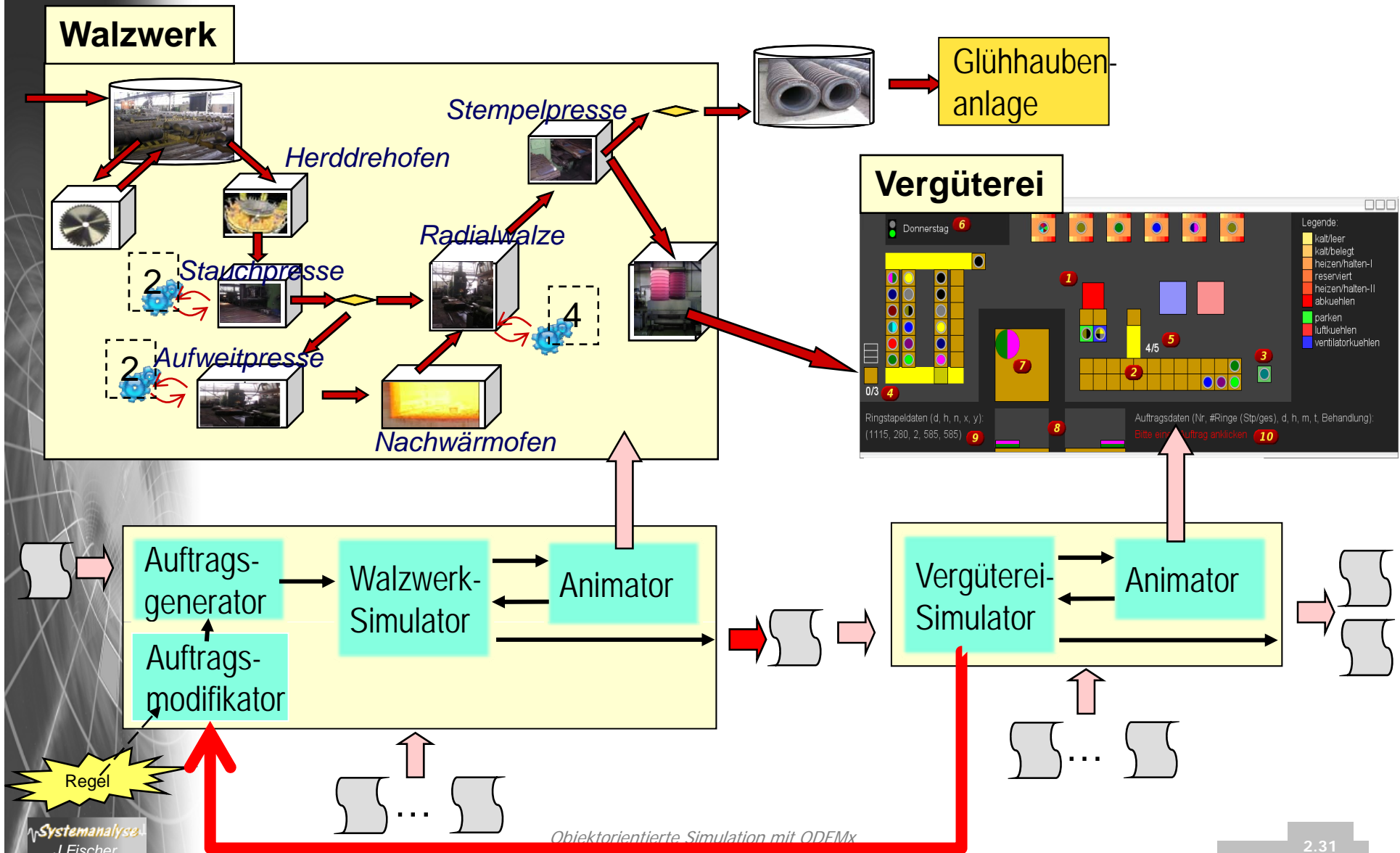


GIS-based editor

J.Fischer

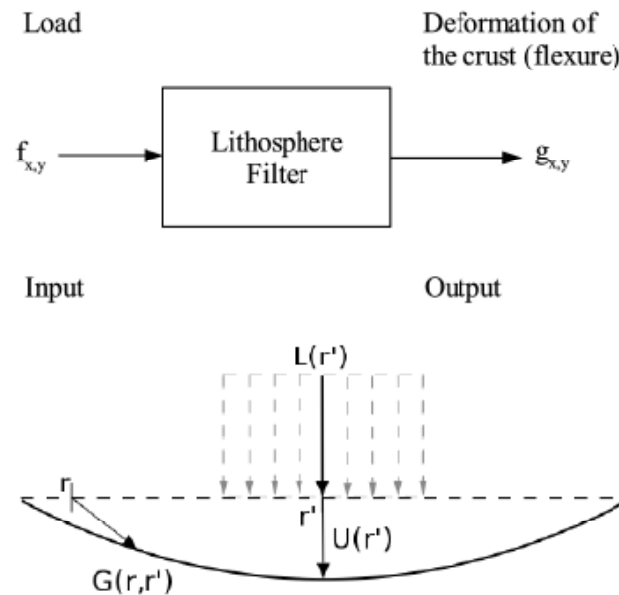
orientierte Simulation mit ODEMx

2. Workflow-Modell von Ring-Walzprozessen

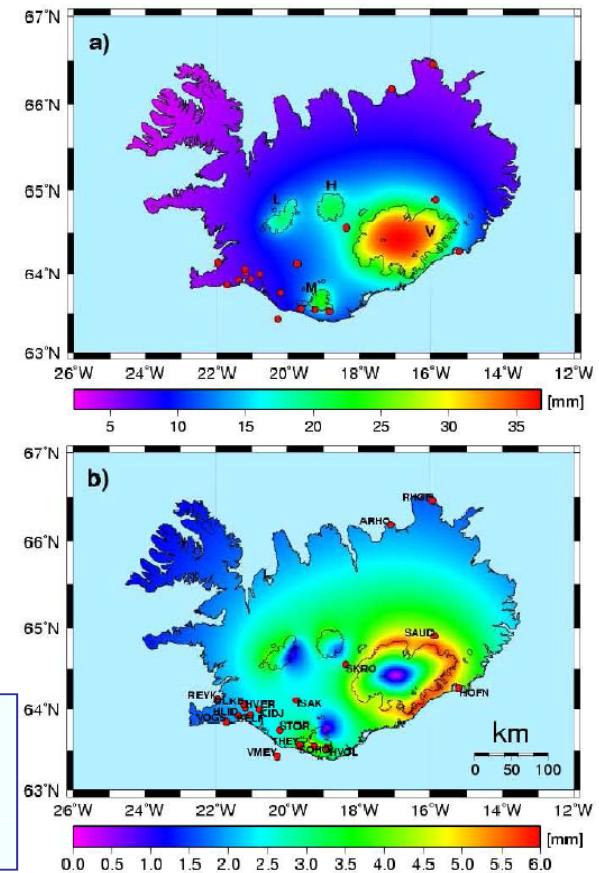


3. Gletschereis-Last-Simulator

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



SimGlazier: eine spezifische Simulationsumgebung

