



# Projekt Erdbebenfrühwarnung im SoSe 2011



## Entwicklung verteilter echtzeitfähiger Sensorsysteme



Joachim Fischer  
Klaus Ahrens  
Ingmar Eveslage

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

EDIM  
SOSEWIN-extended



# Ausbildungsinhalte

Dienstag, 11.15 Uhr  
R 4.112

Erdbeben

Eingebettete Echtzeitsysteme  
Maschennetzwerke

Modellierung von  
Echtzeitsystemen

Simulation/Codegenerierung/  
Test

Donnerstag, 11.15 Uhr  
R 4.112

Objective-C erlernen

- ▶ Memory Management
- ▶ Multithreading, Blocks & GCD
  - mit xcode arbeiten
  - Frameworks kennenlernen:
- ▶ Foundation
- ▶ Protocols, Views und View Controllers
- ▶ Gestures, UI Kit
- ▶ Core Data, Core Location, Core Motion, Map Kit, Media

Dienstag, 13.00 Uhr  
R 4.112

**Praktikum:** Service-Entwicklung  
Applikationen für iPhone (und iPad) bauen

# 1. Kleine Erdbebenkunde

- Plattentektonik
- Klassifikation von Beben
- Magnitudenbestimmung
- Messgeräte
- Erdbebenwellenklassifikation
- Beben in Norditalien

# GFZ Task Force

## Earthquake in Central Italy



**D. Di Giacomo, R. Ditommaso, M. Picozzi, M. Pilz**

## Mw 6.3 Central Italy Earthquake of 6 April 2009 (01:32 UTC)



Date 04/06/09 Region CENTRAL ITALY MI 6.2 Mw 6.3  
Centroid Location:  
Or. Time 1:32:47.2 Lat. 42.32 N Long. 13.32 E Dep 12. fixed

Best Double Couple  $M_0: 3.7 \cdot 10^{25}$   
P1 str: 147 dip: 43 slip: -88  
P2 324 47 -92

040609A

Moment Tensor ( $10^{25}$  dyn-cm)  
mrr: -3.75 mtt: 1.22 mff: 2.53  
mrt: 0.23 mrf: -0.16 mtf: -1.71

### Principal Axes

T val: 3.72 plg: 2 az: 55  
N 0.05 1 325  
P -3.76 88 200

DATA USED: MEDNET

AUTHOR: INGVBO



According to the INGV, the mainshock occurred between 10-12 km depth along an extensional fault system in direction NO-SE.

### Aim of the mission

Provide scientific support to the Italian colleagues in the investigation of site effects, and monitoring of buildings



L'Aquila

72.948 inhabitants

# M 6.3, CENTRAL ITALY

Origin Time: Mon 2009-04-06 01:32:39 UTC

Location: 42.33°N 13.33°E Depth: 8 km

**PAGER**  
**Version 7**

Created: 3 days, 22 hrs after earthquake

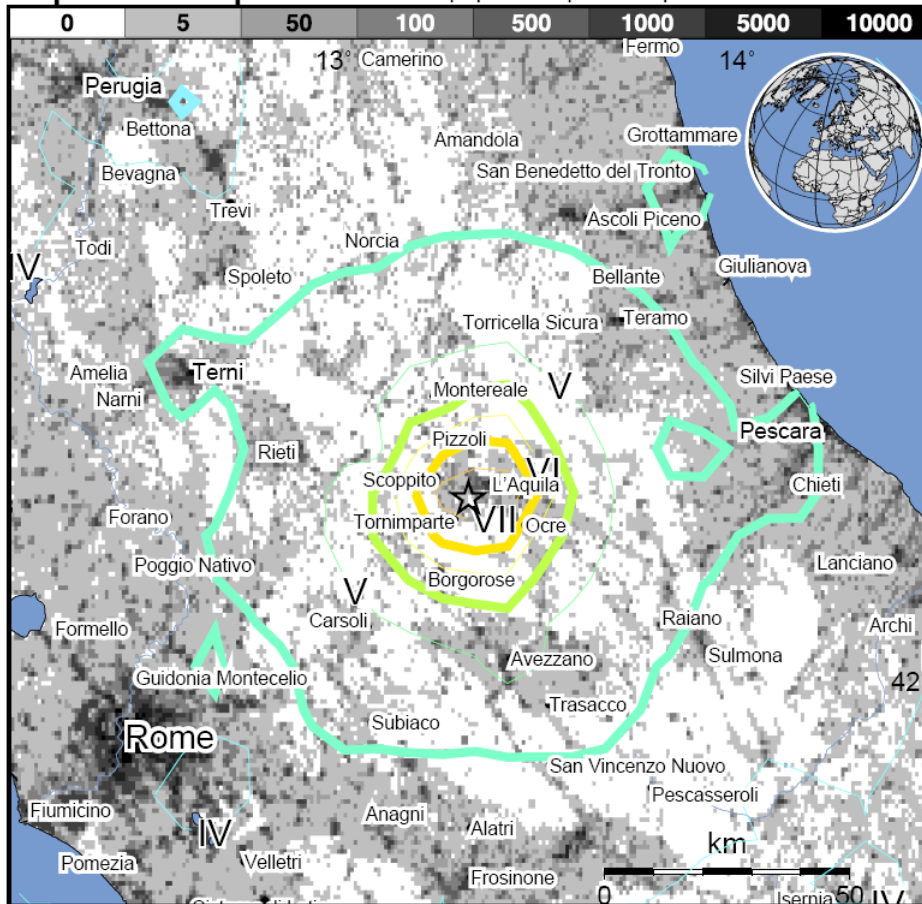
## Estimated Population Exposed to Earthquake Shaking

ESTIMATED POPULATION EXPOSURE (k = x1000)		--*	3k*	5,310k*	989k	28k	79k	4k	0	0
ESTIMATED MODIFIED MERCALLI INTENSITY		I	II-III	IV	V	VI	VII	VIII	IX	X+
PERCEIVED SHAKING		Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
POTENTIAL DAMAGE	Resistant Structures	none	none	none	V. Light	Light	Moderate	Moderate/Heavy	Heavy	V. Heavy
	Vulnerable Structures	none	none	none	Light	Moderate	Moderate/Heavy	Heavy	V. Heavy	V. Heavy

\*Estimated exposure only includes population within the map area.

### Population Exposure

population per ~1 sq. km from Landsat 2006

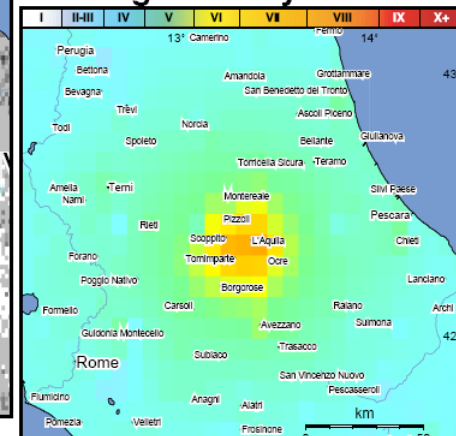


### Selected City Exposure

MMI City	Population
<b>VII L'Aquila</b>	<b>68K</b>
<b>VII Tornimparte</b>	<b>2k</b>
<b>VII Scoppito</b>	<b>2k</b>
<b>VII Pizzoli</b>	<b>3k</b>
<b>VI Ocre</b>	<b>1k</b>
<b>VI Poggio Picenze</b>	<b>1k</b>
<b>V Terni</b>	<b>105K</b>
<b>IV Pescara</b>	<b>116K</b>
<b>IV Guidonia</b>	<b>75k</b>
<b>IV Rome</b>	<b>2,563K</b>
<b>IV Perugia</b>	<b>149K</b>

bold cities appear on map (k = x1000)

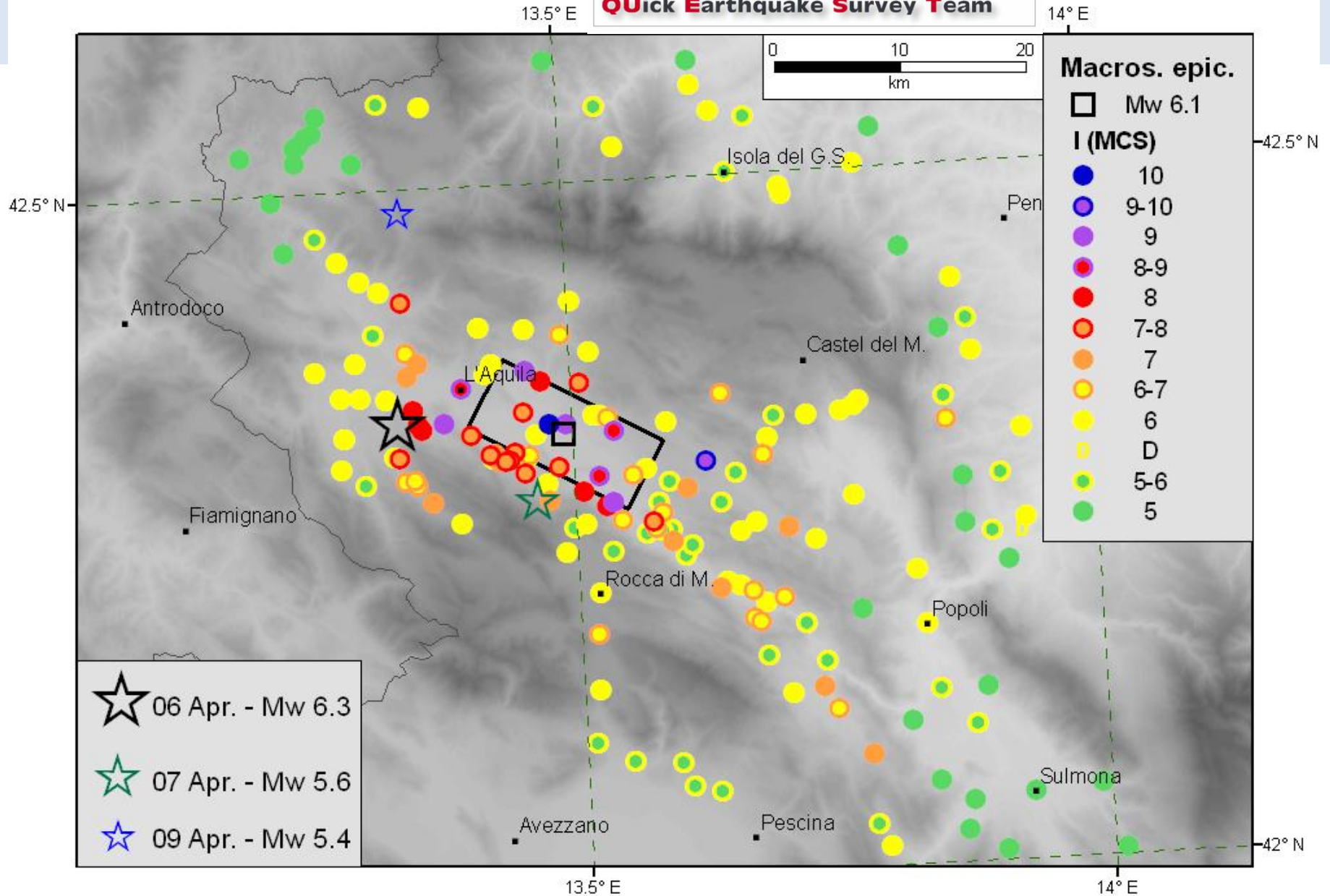
### Shaking Intensity



# Macroseismic survey by

# QUEST

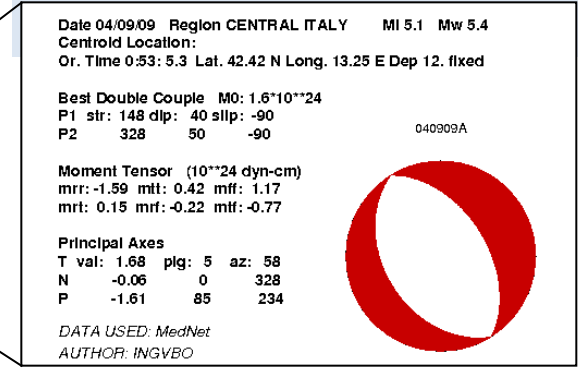
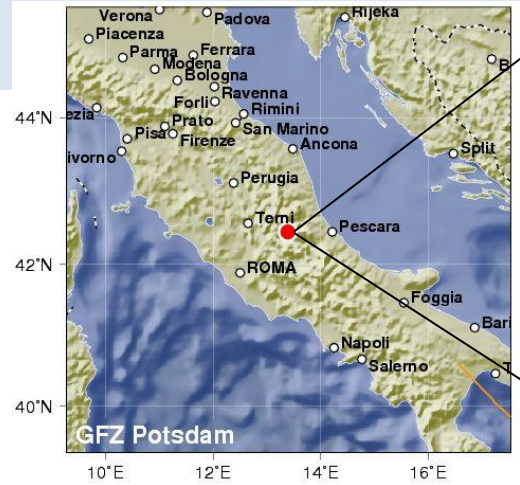
QUICK Earthquake Survey Team



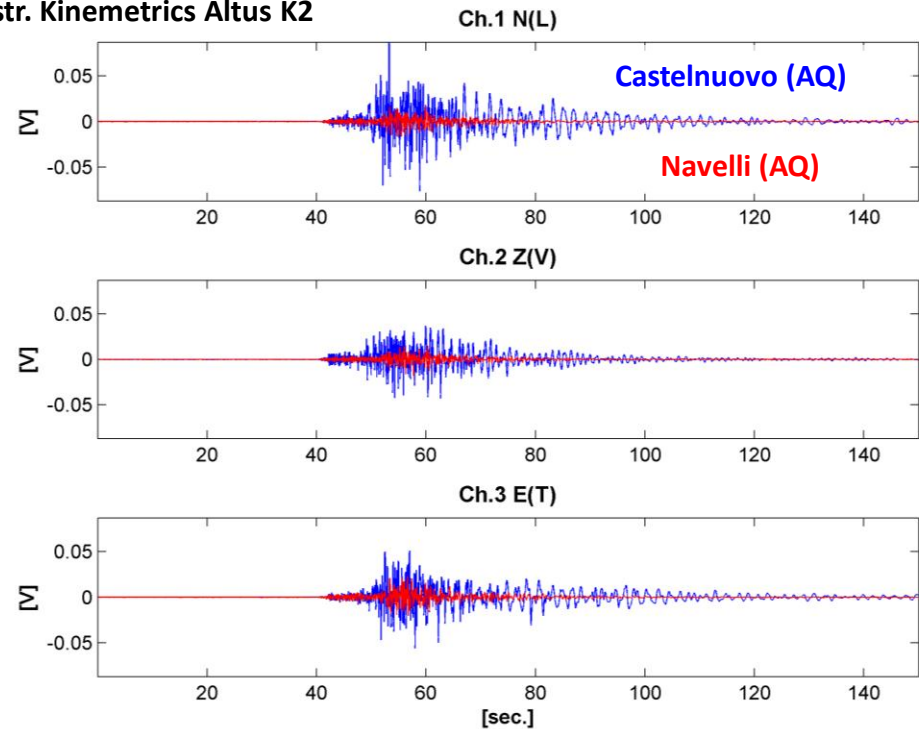




# Mw 5.4 aftershock of 9 April 2009 (00:53 UTC)

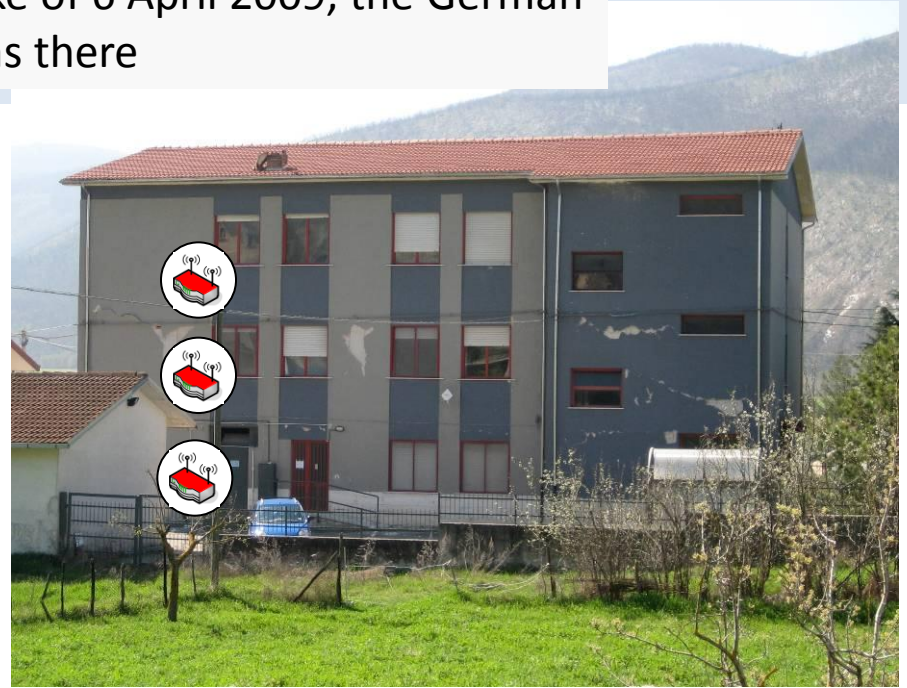


Instr. Kinematics Altus K2

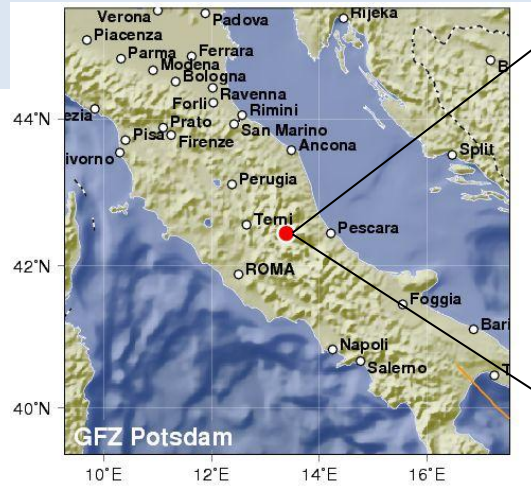


Soon after the Mw 6.3 Central Italy Earthquake of 6 April 2009, the German Earthquake Task Force supported Italian teams there

Navelli municipality's center



# Mw 5.4 aftershock of 9 April 2009 (00:53 UTC)



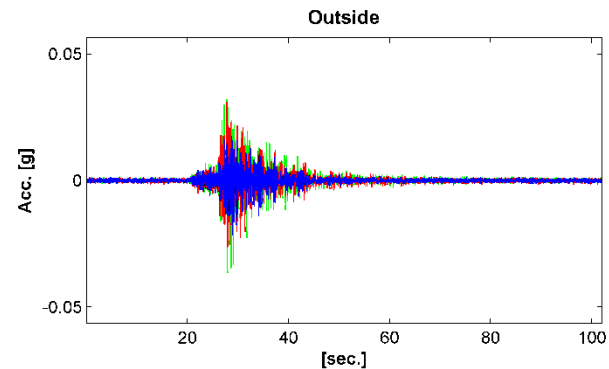
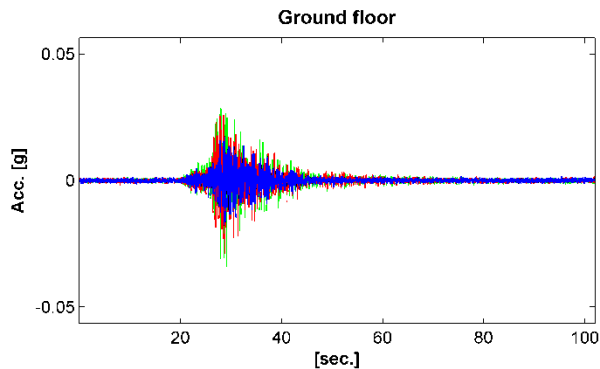
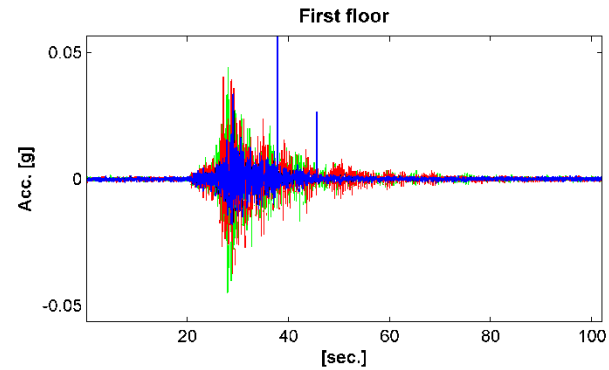
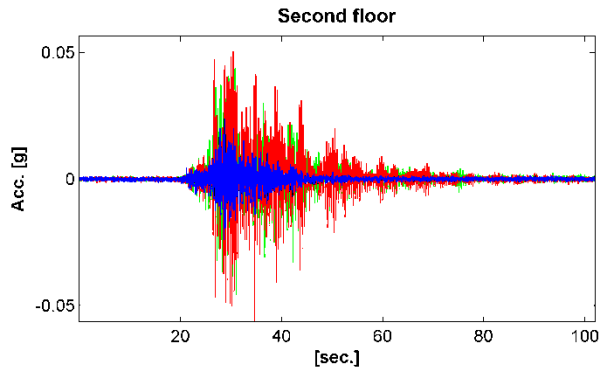
Date 04/09/09 Region CENTRAL ITALY MI 5.1 Mw 5.4  
 Centroid Location:  
 Or. Time 0:53: 5.3 Lat. 42.42 N Long. 13.25 E Dep 12. fixed

Best Double Couple M0:  $1.6 \cdot 10^{24}$   
 P1 str: 148 dlp: 40 slip: -90  
 P2 328 50 -90 040909A

Moment Tensor ( $10^{24}$  dyn-cm)  
 mrr: -1.59 mlt: 0.42 mlf: 1.17  
 mrt: 0.15 mrl: -0.22 mtf: -0.77

Principal Axes  
 T val: 1.68 plg: 5 az: 58  
 N -0.06 0 328  
 P -1.61 85 234

DATA USED: MedNet  
 AUTHOR: INGVBO

























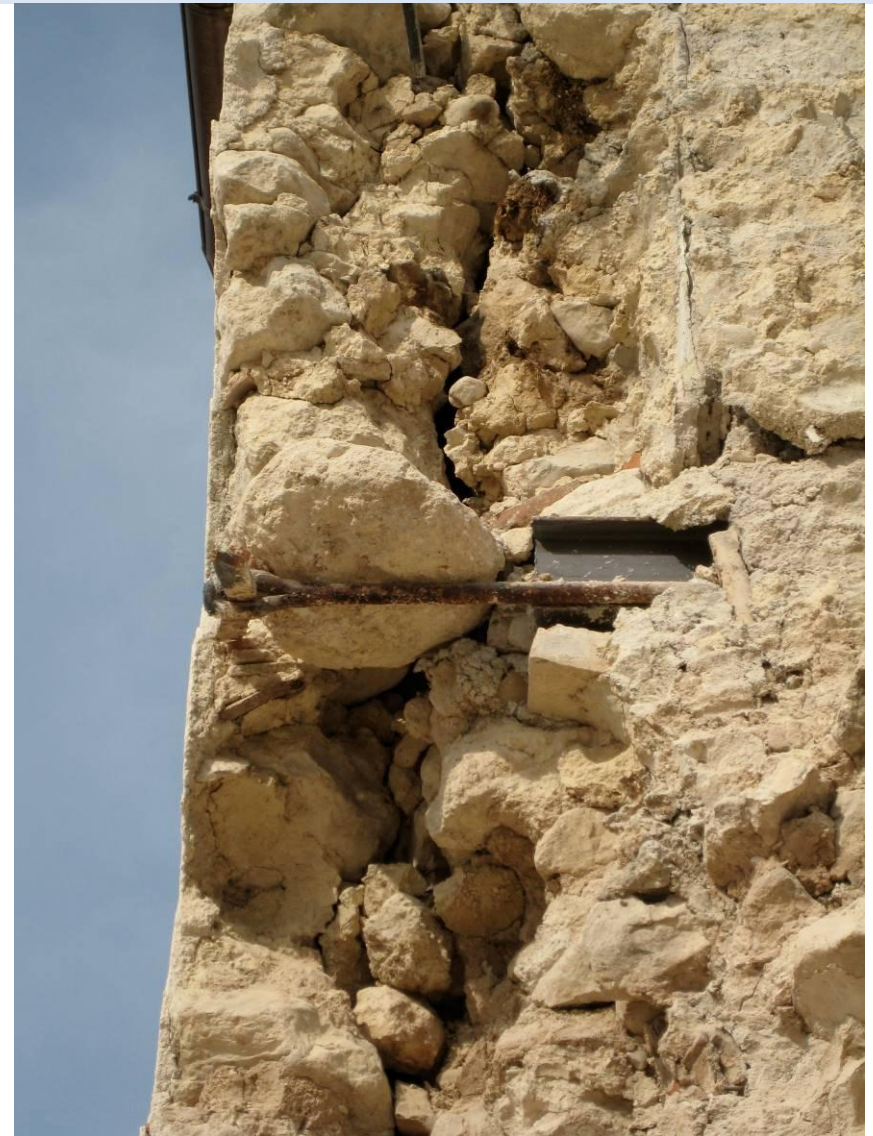






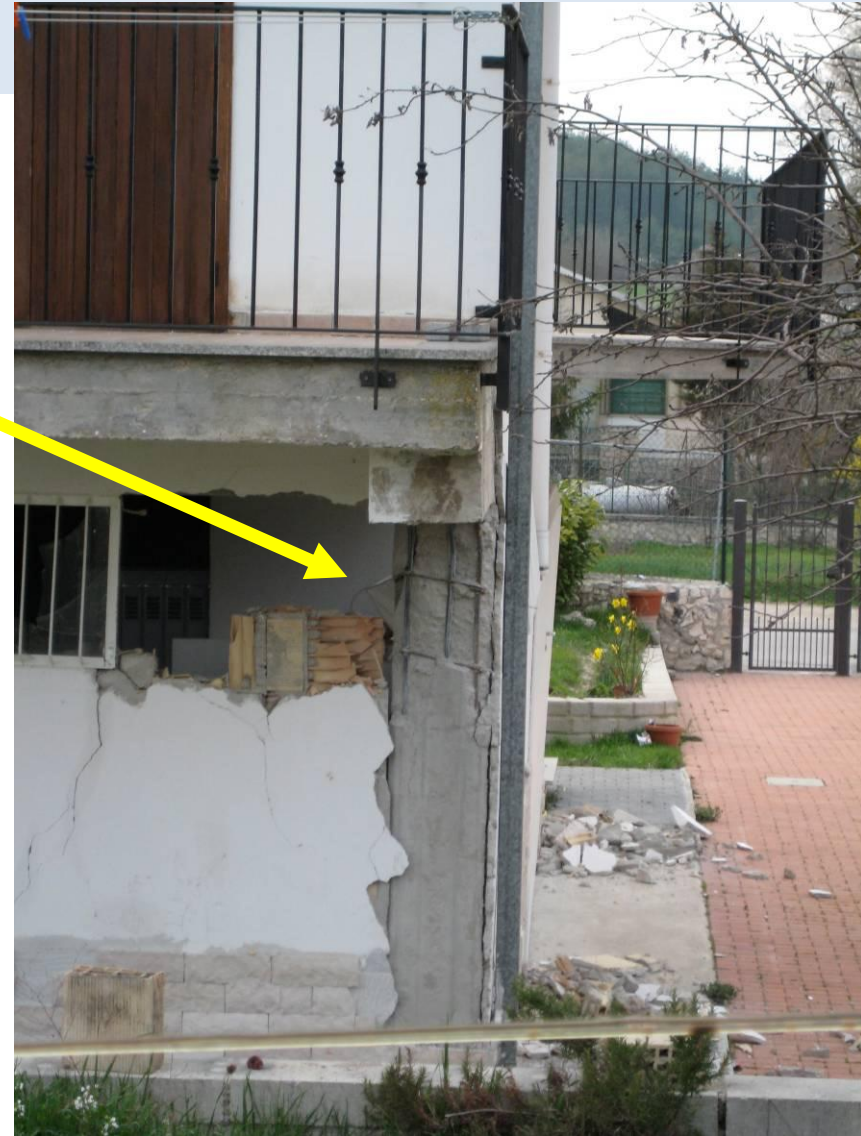


# Castelnuovo



# Fossa





Castelnuovo

# Paganica



# Paganica





San Gregorio

Directional effects ???







# Pianola MCS= VII-VIII



## QUEST

QUICK Earthquake SURVEY TEAM





# Überwachung von Brückenkonstruktionen

## STRUCTURAL MONITORING OF THE ADOLPHE BRIDGE, LUXEMBOURG CITY

Adrien Oth and Matteo Picozzi

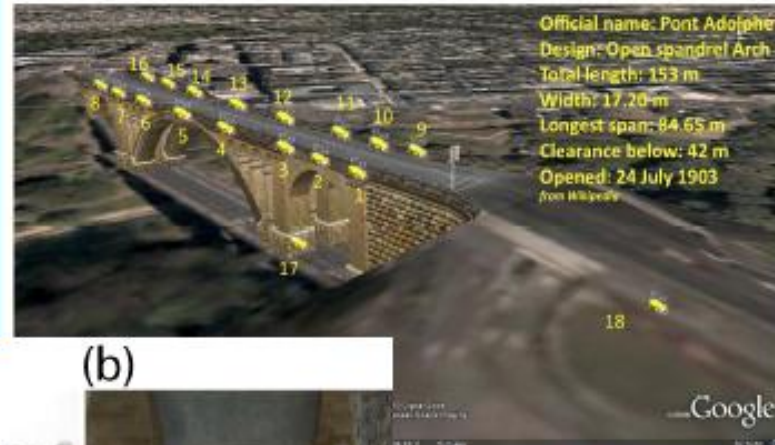


European Center for  
Geodynamics and Seismology

LUXEMBOURG – 11/05/2010

18 sosewin + 4.5 geophones  
Recording time: 9:00 UTC – 17:00 UTC

(a)



Official name: Pont Adolphe  
Design: Open spandrel Arch  
Total length: 153 m  
Width: 17.20 m  
Longest span: 84.65 m  
Clearance below: 42 m  
Opened: 24 July 1903  
from Wikipedia

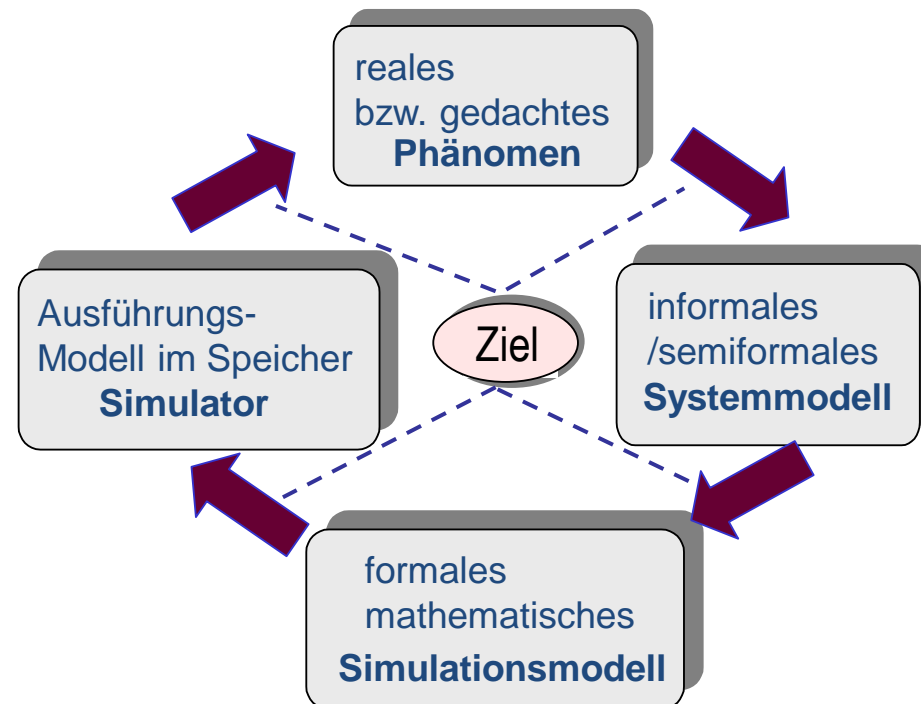


## *2. Konzepte und Sprachen zur Modellierung verteilter Systeme*

### 1. MDD - Allgemeine Ziele

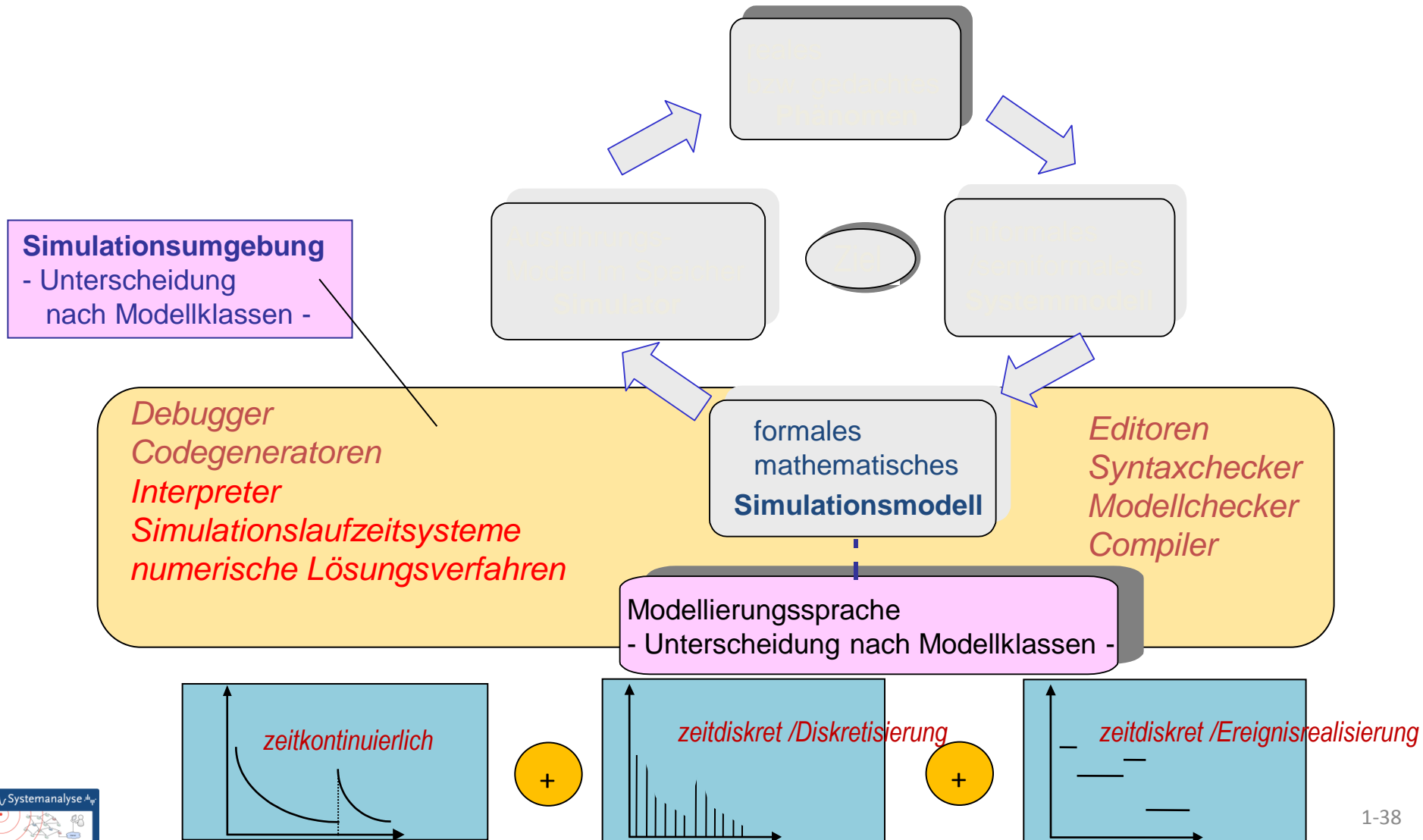
2. Charakterisierung von UML
3. Modellierungselemente von UML im Überblick
4. Diagrammarten
5. Struktur des UML-Standards

# Modellierung und simulative Untersuchung von Systemen



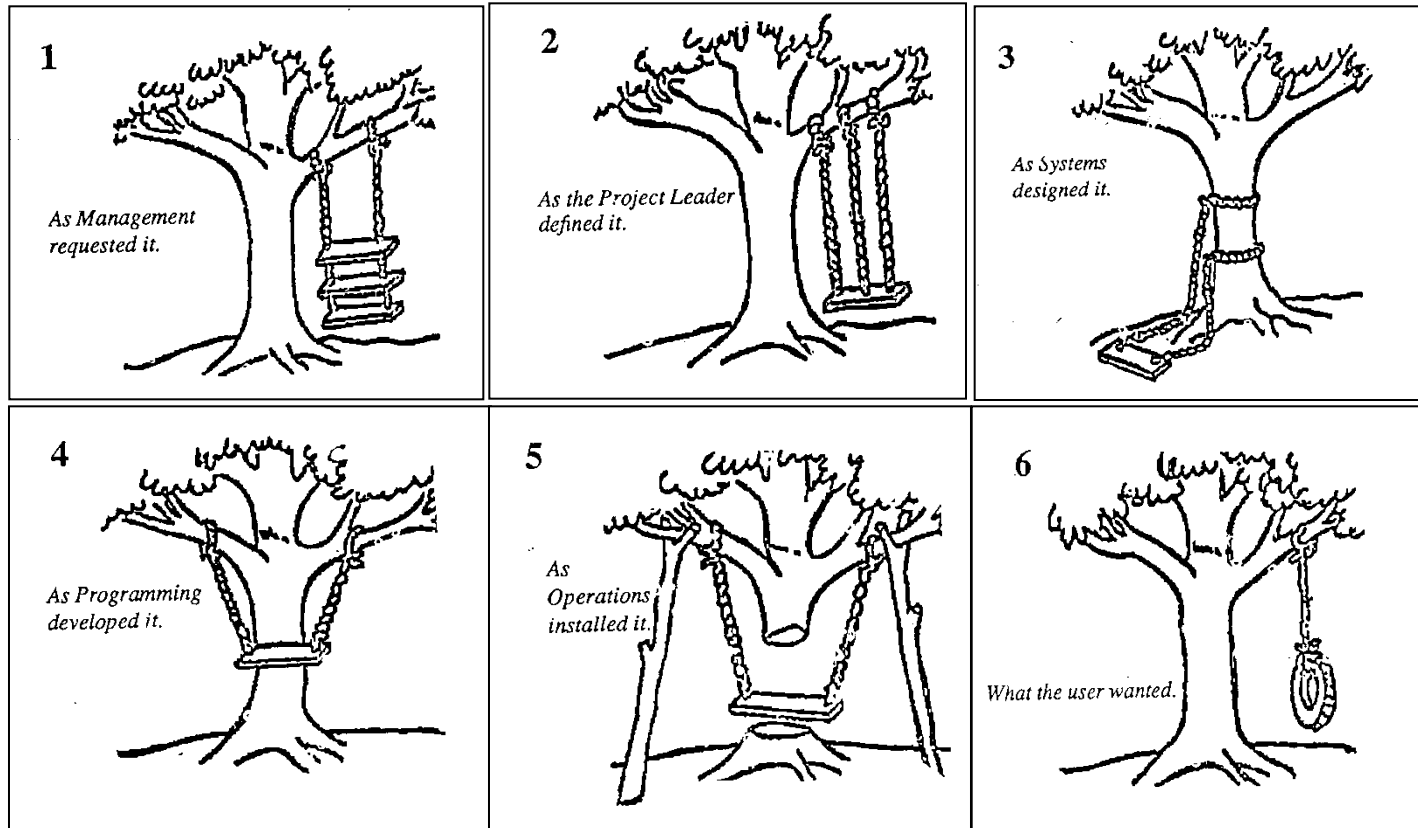
# Modellierungssprachen und Simulationsumgebung

Zustandsänderungen kontinuierlich oder/und diskret in Raum und Zeit



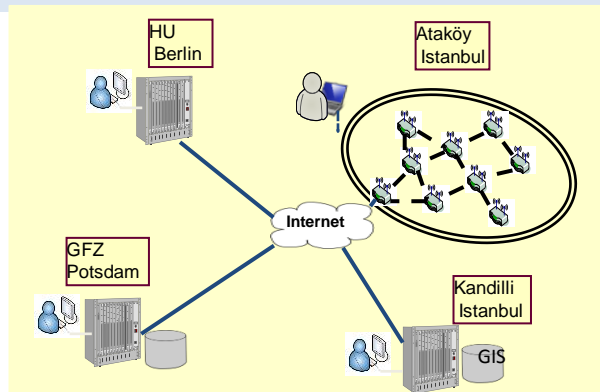
# Probleme der traditionellen SW-Entwicklung

Wozu Modellierung ?



# Modellierung und simulative Untersuchung von Systemen

Sonderfall:  
Modell und Original-  
Teile  
liegen als Software  
vor



reales  
bzw. gedachtes  
Phänomen

Ausführungs-  
Modell im Speicher  
**Simulator**

informales  
/semiformales  
**Systemmodell**

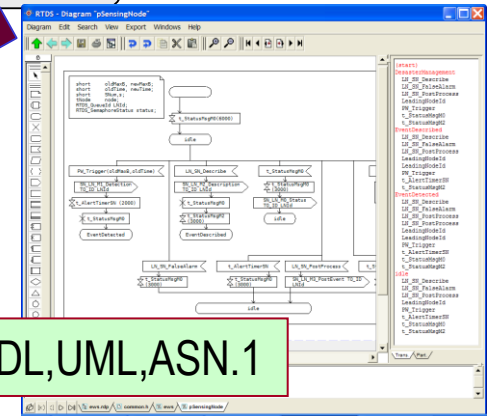
Ziel

formales  
mathematisches  
**Simulationsmodell**

Experiment-  
Management-  
System

**Zielcode: C++, OpenWRT, Boost**

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





# Modellierung in der SW-Entwicklung

- 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**
- ➔ UML-Technologien  
sind stark im Kommen –  
aber immer noch mit Lücken

# SW-Entwicklungsprozess: spiralförmig, inkrementell & iterativ

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

*Test funktionaler und nicht-funktionaler Rückkopplungen*

*Wechselwirkung mit der Umgebung mit Simulation*

Integration

Deployment

Anforderungsanalyse

Implementation

MDD

*SDL, UML, SysML*

*Echtzeit, Leistungsprognose Ausführung durch Simulation*

Test/Validierung

*Test nicht-funktionaler Eigenschaften*

Design

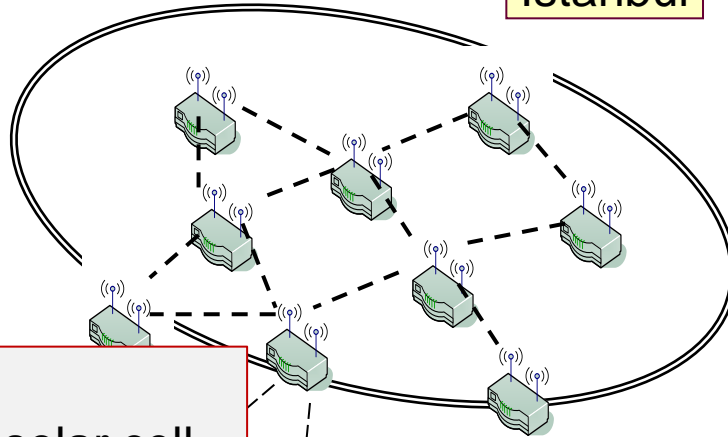
*Ausführung durch Simulation*

*Test funktionaler Eigenschaften*



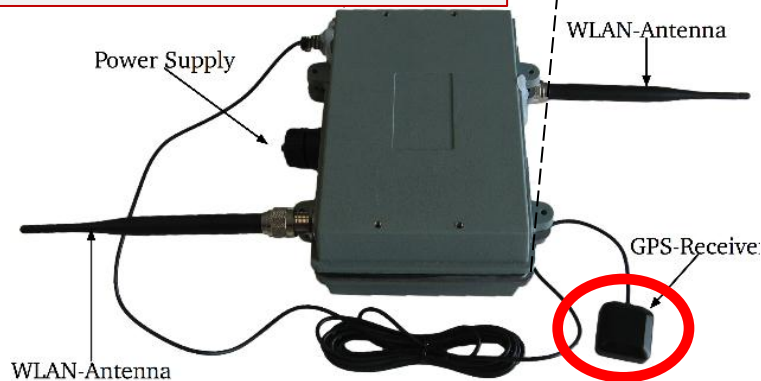
# SOSEWIN – Hardware, Software

Ataköy  
Istanbul



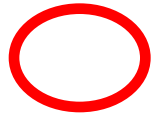
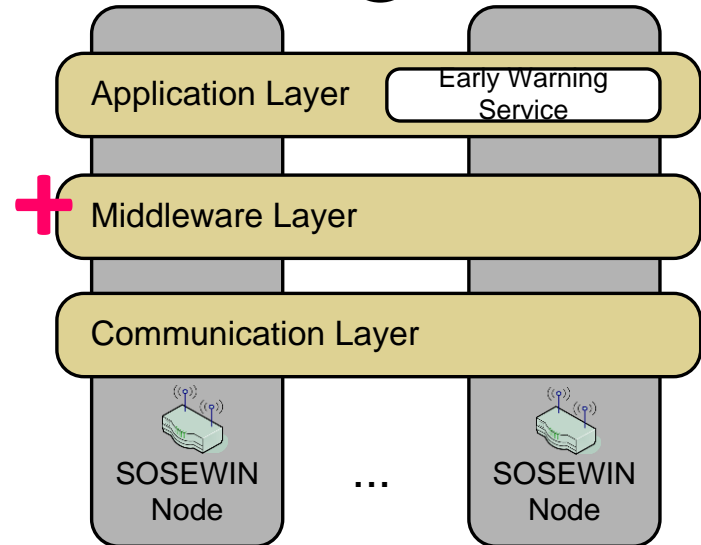
## source

- battery with solar cell
- grid

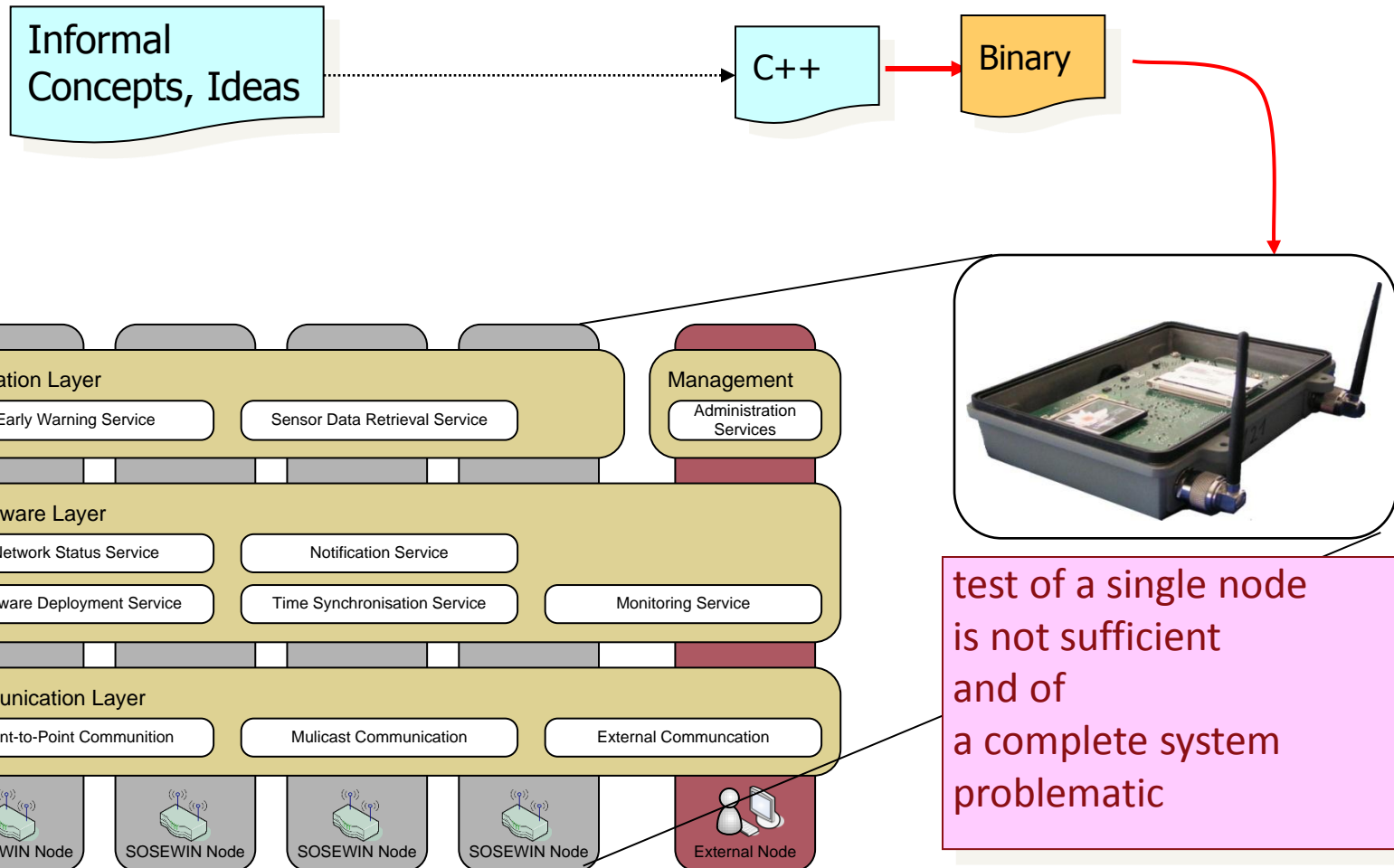


## Wrap-Board

- 32-bit Processor (AMD Geode)
- 233 MHz
- 128 MB RAM
- 1024 MB Flash
- 2 x 54 Mb/s Transceiver
- **Linux OS**  
(sensor drivers)

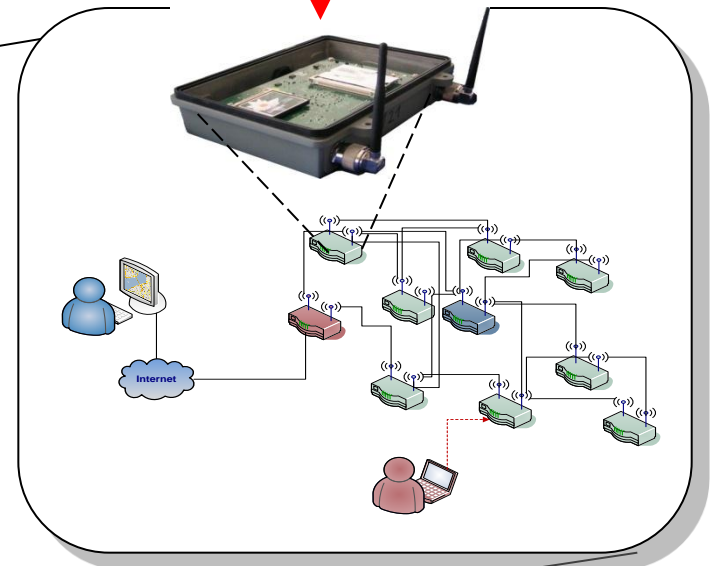
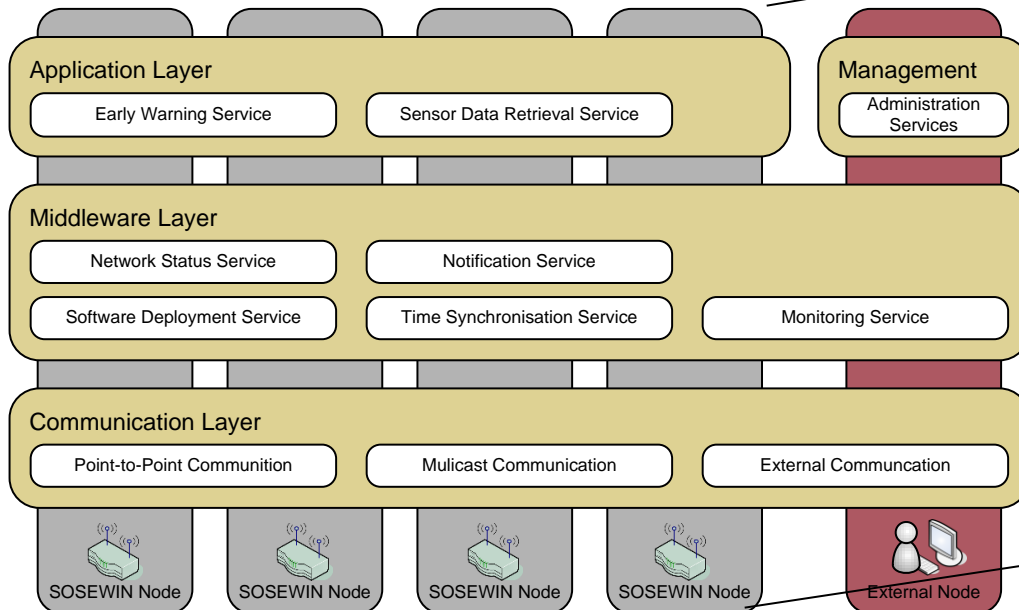
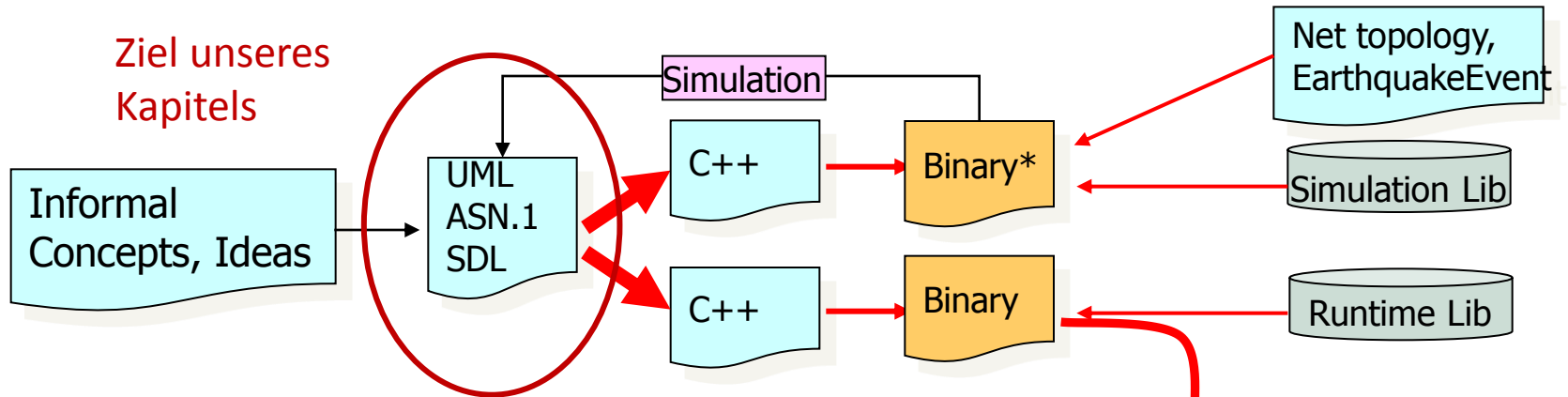


# Traditionelle Software-Entwicklung



# Unser Ansatz einer modellgetriebenen SW-Entwicklung

Ziel unseres Kapitels



simulation of a complete system simplifies testing

## *2. Konzepte und Sprachen zur Modellierung verteilter Systeme*

1. MDD - Allgemeine Ziele
2. Charakterisierung von UML
3. Modellierungselemente von UML im Überblick
4. Diagrammarten
5. Struktur des UML-Standards

# UML-Charakterisierung (1)



„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

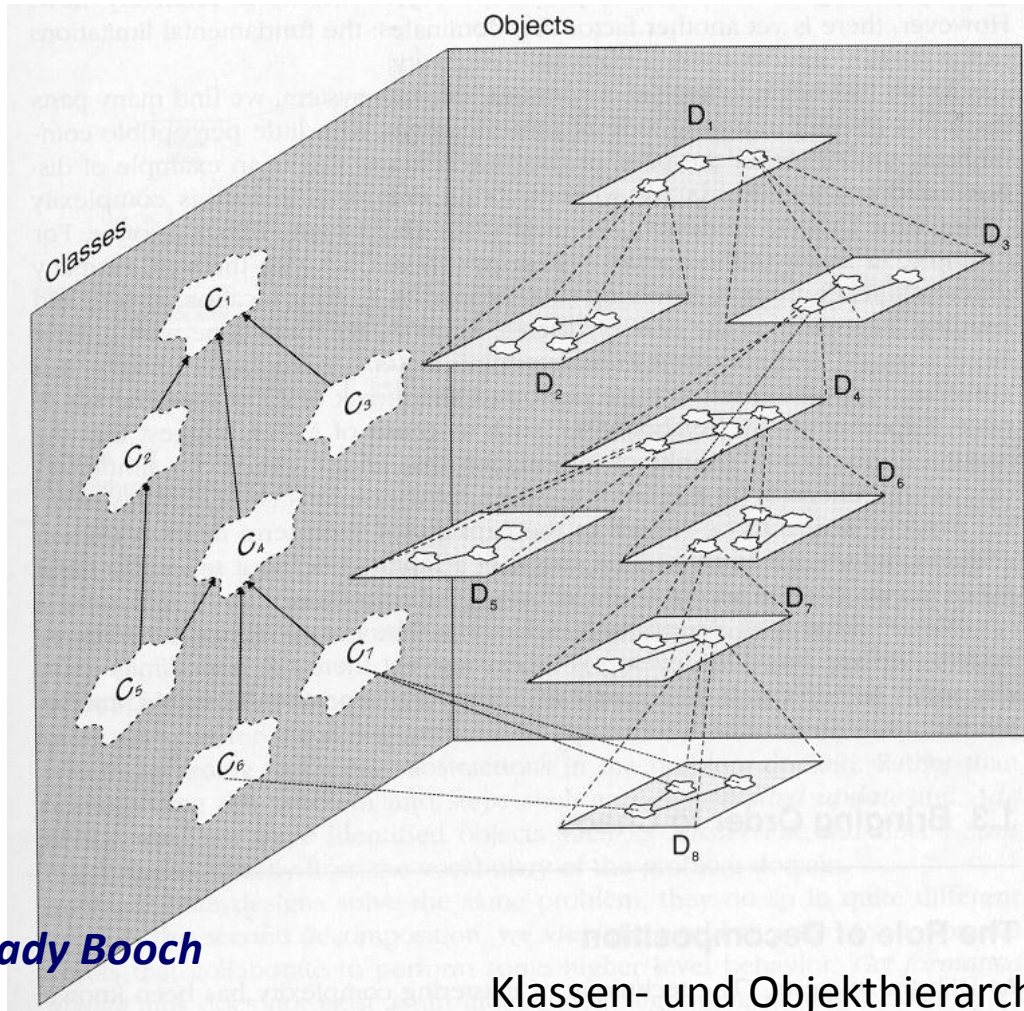
# Objektorientierte Modellierung

- als bestimmendes u. durchgehendes Modellierungsparadigma von UML
  - Klassifikation /Exemplifikation
  - Instanz und Klassifizierer
  - Beziehungen zwischen Klassifizierern
    - Spezialisierung / Generalisierung (Wiederverwendung von Modellbausteinen)
    - abstrakte und konkrete Klassifizierer
    - aktive und passive Klassifizierer
  - Instanz als Dreiklang von Identität (Referenz), Struktur und Verhalten
  - Verhalten
    - Ursache-Wirkungsprinzip,
    - zeitliche Abhängigkeit,
    - Parallelität/Nebenläufigkeit,
    - Kommunikation
  - Beziehungen zwischen Instanzen / Instanzmengen
    - Kommunikation
    - Generierung/Terminierung
  - Vielgestaltigkeit von Referenzen (Polymorphie)

Abstraktionsmechanismen



# Komposition / Dekomposition



**Grady Booch**

Klassen- und Objekthierarchien

**Klassifikation**

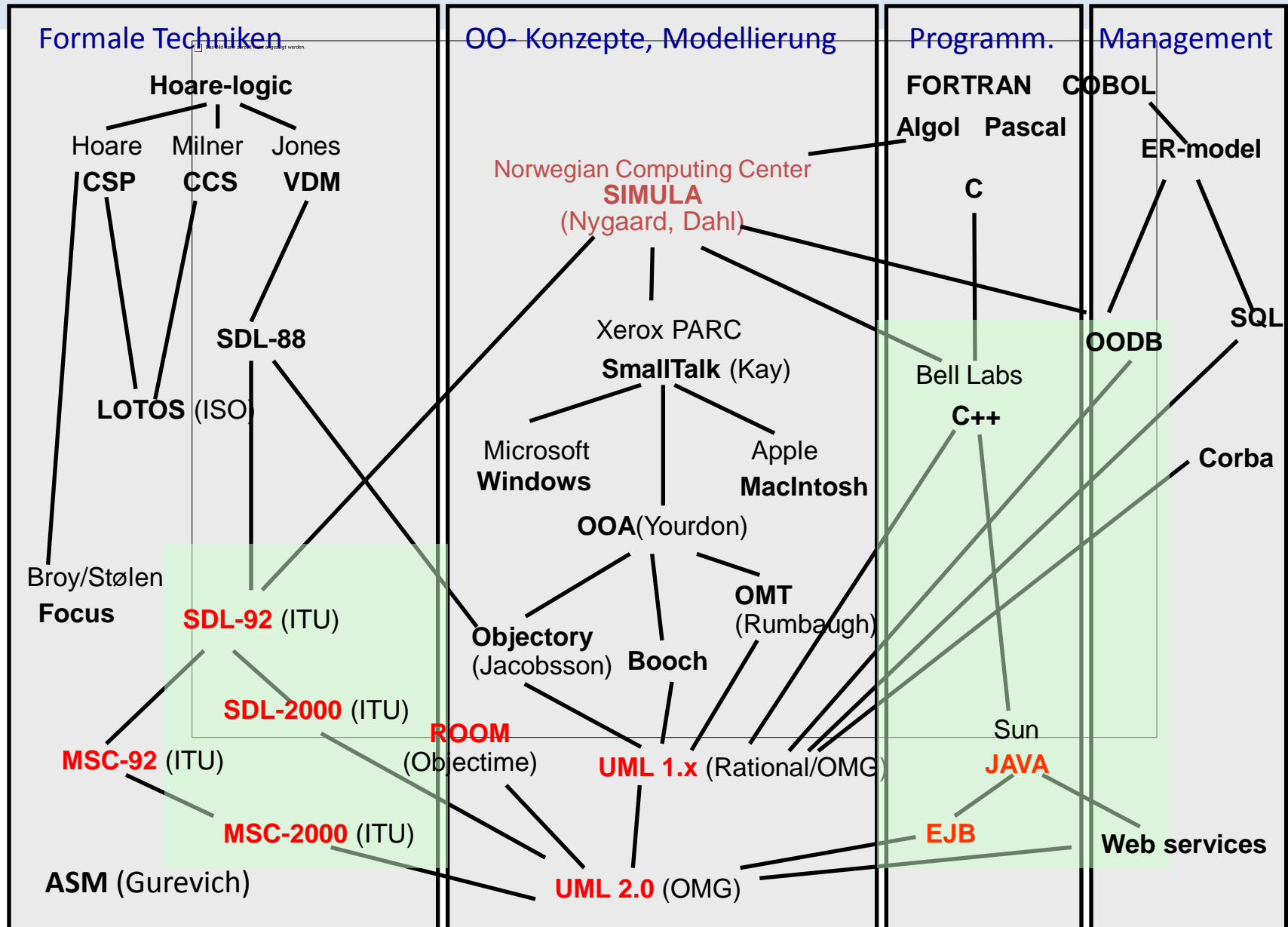
**Spezialisierung**

**Exemplifikation**

**Komposition**

**Wiederverwendung**

# Die Vorläuferkonzepte von UML



# UML-Charakterisierung (2)

- ist eine Notation/Sprache, keine Methode, Framework für UML-ähnliche Modellierungssprachen
- abstrahiert von
  - architekturellen Vorgaben,
  - Design- und Implementierungs-Styles,
  - Technologien (Software, Hardware, Infrastrukturen, ...),
  - Entwicklungsprozessen
- standardisiert
  - Begriffswelt (Modellierungskonzepte),
  - Semantik (Bedeutung der Modellierungskonzepte),
  - visuelle Darstellung (Notation der Modellierungskonzepte)
- führt Ideen verschiedener Techniken zusammen
  - Booch, OMT, Jacobsson, ROOM, SDL, EDOC, MSC, Component Based Modeling, ...

# UML-Modell

..., Definition aus dem Standard

A **model** captures a view of a physical system.

It is an abstraction of the physical system, with a certain purpose.

This purpose determines what is to be included in the model and what is irrelevant.

Thus the model completely describes those aspects of the physical system that are relevant to the purpose of the model, at the appropriate level of detail.

Modelliere nicht so detailliert wie möglich,  
sondern  
so detailliert wie nötig !

# Modellierungskultur in der Softwareentwicklung: Sichtweisen

am Beispiel von ODP (Open Distributed Processing)  
ISO- und ITU-Standard zur Modellierung verteilter Software-Systeme

## Enterprise

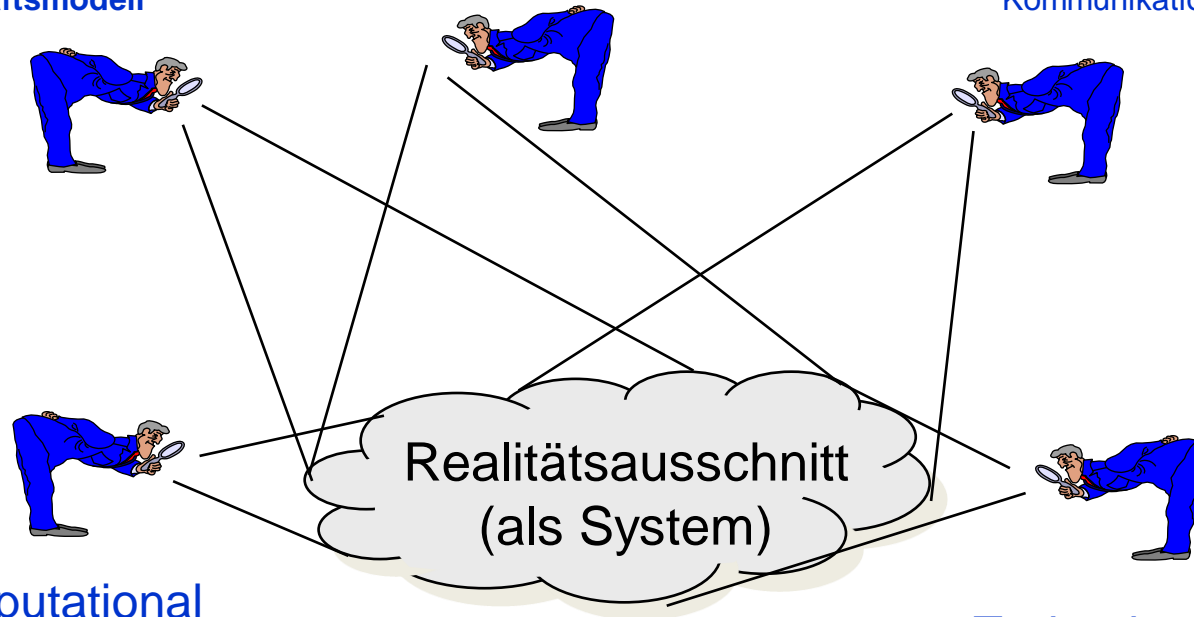
Unternehmens  
Anwendungsaspekte  
**Geschäftsmodell**

## Information

Informationsdarstellung,  
-semantik,-verarbeitung

## Engineering

**Verteilungsaspekte**  
Kommunikationsstruktur



## Computational

Dekomposition in verteilbare **Komponenten**  
Definition von **Schnittstellen**

## Technology

Implementationsprache  
Werkzeuge  
Plattformen

5 Modellklassen mit spezifischen Darstellungskonzepten

## *2. Konzepte und Sprachen zur Modellierung verteilter Systeme*

1. MDD - Allgemeine Ziele
2. Charakterisierung von UML
3. Modellierungselemente von UML im Überblick
4. Diagrammarten
5. Struktur des UML-Standards

# UML-Grundkonzepte

## 1. UML-Modellbausteine (gruppiert nach Sichten)

- Dinge/Entitäten (im Sinne von Abstraktionen)
- Beziehungen zwischen den Entitäten
- Diagramme zur Anordnung von Entitäten

## 2. Regeln zur Komposition der Modellbausteine

- diagrammspezifisch (später)

## 3. universelle Mechanismen,

- gültig für gesamte Sprache (später)

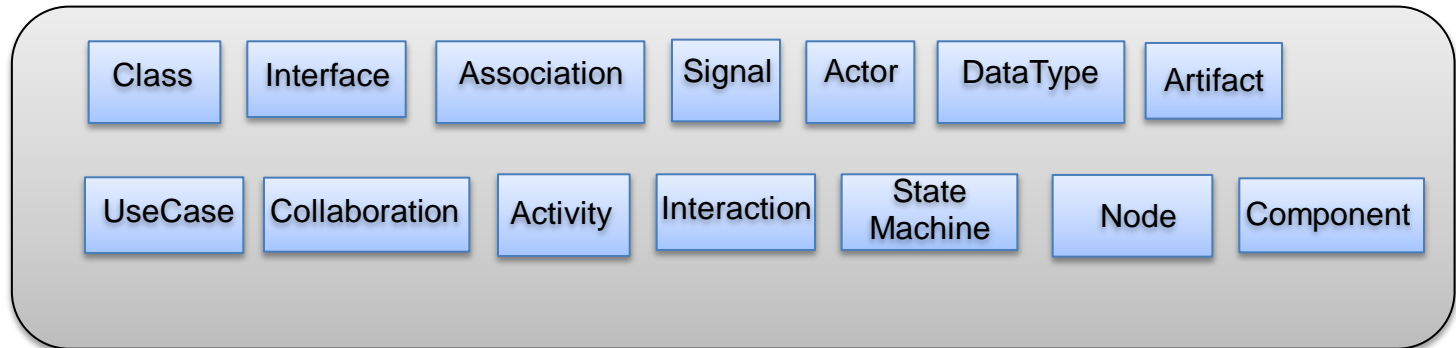
- Bildung von Strukturen
- Beschreibung von Verhalten
- Bildung von Gruppierungen von Entitäten
- Formulierung von Anmerkungen

➔ damit lassen sich UML-Modelle komplett beschreiben

# Überblick zu Strukturellen UML-Entitäten

- Substantive in Modellen (formuliert in UML)
- bilden die Struktur eines Modells, meist statisch- aber auch dynamisch
- Zusammenfassung als *Classifier*

als abstraktes Konzept (mit Vererbung-/Spezialisierung, Virtualität)



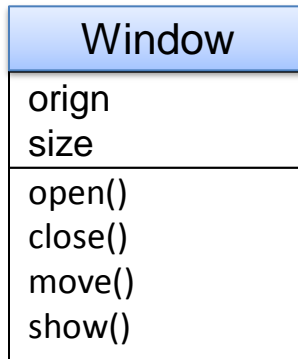
OO als bestimmendes und durchgehendes Modellierungsparadigma

z.T. mit weiteren Spezialisierungen

dafür einige Beispiele (Syntax, Semantik) ...



# Strukturelle Entitäten: Klasse, Interface

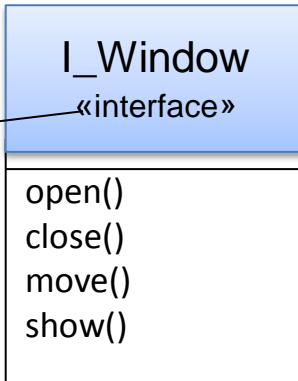


## Klasse (Class)

- beschreibt Abstraktion einer Menge von Objekten
- mit gleichen Attributen, Operationen, Beziehungen, Semantik
- dargestellt als Rechteck mit Unterteilungen (Compartments)
- besitzt mindestens ein Interface

graph. Symbol  
(hier das einer  
Klasse)  
erhält neue  
Semantik

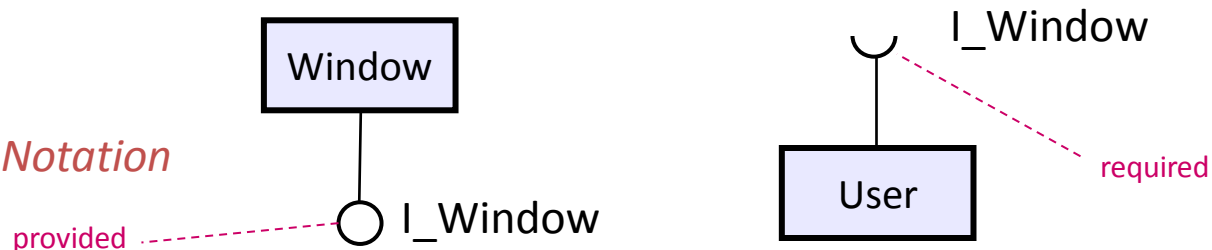
(z.B. **nicht**  
instanzierbar)



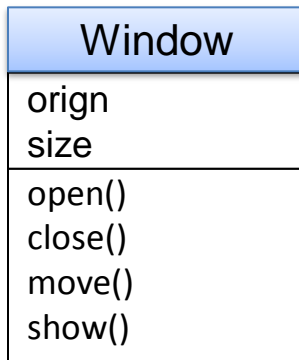
## Interface (Interface)

- beschreibt Sammlung von Operationen, die durch Dienste einer Klasse, Komponente, oder eines Ports festgelegt sind
- beschreibt mit Signaturen potentiell äußeres Verhalten einer Klasse/Komponente komplett oder partiell, bzw. Signatur eines Ports
- **keine Implementation !!!**
- Bezug zur Klasse/Komponente/Port muss explizit dargestellt werden

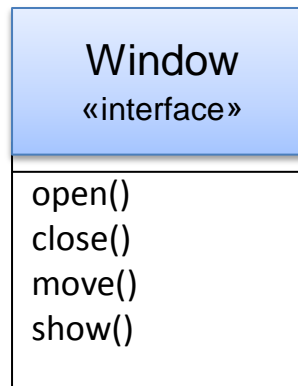
*Lolli-Pop-Notation*



# Strukturelle Entitäten: Klasse, Interface



Klasse



Interface

zulässig:  
identische Namen für  
unterschiedliche Konzepte

