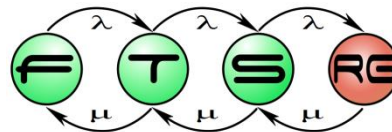


Systemmodellierung

Dr. PATARICZA András
HUSZERL Gábor

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group



Die Lehrveranstaltung

- Lehrveranstaltung (VIMIAA00)
 - 3 in 1 (auf ungarisch/deutsch/englisch)
- Dr. PATARICZA András
 - Verantwortlicher Professor
- GÖNCZY László, BERGMANN Gábor
 - Operative Leitung, Organisatorische Fragen
- HUSZERL Gábor (huszerl@mit.bme.hu)
 - deutschsprachige Vorlesung, ...

Die Deutschsprachige Lehrveranstaltung

■ Vorlesungen und Leitung

- HUSZERL Gábor

■ Übungen

- ERDŐS Szilvia
- TEMESVÁRI Fanni



■ Webseite

- <https://inf.mit.bme.hu/edu/courses/remo-de>
- Infos, Nachrichten, Folien, ...

Die Lehrveranstaltung

- 13 Vorlesungen
 - Mittwochs 10-12 Uhr, I.L405 (15. März fällt weg)
- 6 Übungen
 - Freitags 10-12 Uhr (in ungeraden Wochen), I.B144
 - Zu erst am 24. Februar
 - Tests (keine Eingangstests!), Anwesenheitspflicht
- 1 Hausaufgabe
 - mit mündlicher Verteidigung
- 2 Klausuren
 - mit Eingangstest

Bewertung

- Klausuren (mit Eingangstest): 35%+35% der Endnote
- Hausaufgabe (mit Verteidigung): 30% der Endnote
 - alle drei Teile mit mindestens 40% der Punkte
 - eine Klausur und die Hausaufgabe kann nachgeholt werden
- Optionale Zusatzpunkte:
 - Übungsvorbereitungen (mind. 5 Tests aus den 6): +5%
 - optionale Zusatzaufgaben für zusätzliche Punkte

Hausaufgabe

- Abgabepflicht, Verteidigungspflicht
- Persönliche Aufgabe
- Wichtige Termine
 - Ausgabe der Aufgaben: 3. Semesterwoche
 - Erste Abgabe (optional): 5. Semesterwoche
 - Abgabe: 12. Semesterwoche
- Elektronische Abgabe
- Mündliche Verteidigung

Thematik

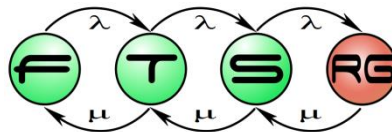
- Visuelle Datenanalyse
- Strukturmodelle
- Verhaltensmodelle
 - Zustandsmodelle, Prozessmodelle
- Entwicklung von Modellen
- Überprüfung von Modellen
- Leistungsmodellierung
- Simulation
- Benchmarking, Codegenerierung

Grundlagen der Modellierung

Dr. PATARICZA András

HUSZERL Gábor

**Budapest University of Technology and Economics
Fault Tolerant Systems Research Group**



Inhalt

Modelle und
Modellierung

Wofür werden Modelle
benutzt?

Grundbegriffe

Illustrative
Beispiele

Inhalt

Modelle und
Modellierung

Wofür werden Modelle
benutzt?

Grundbegriffe

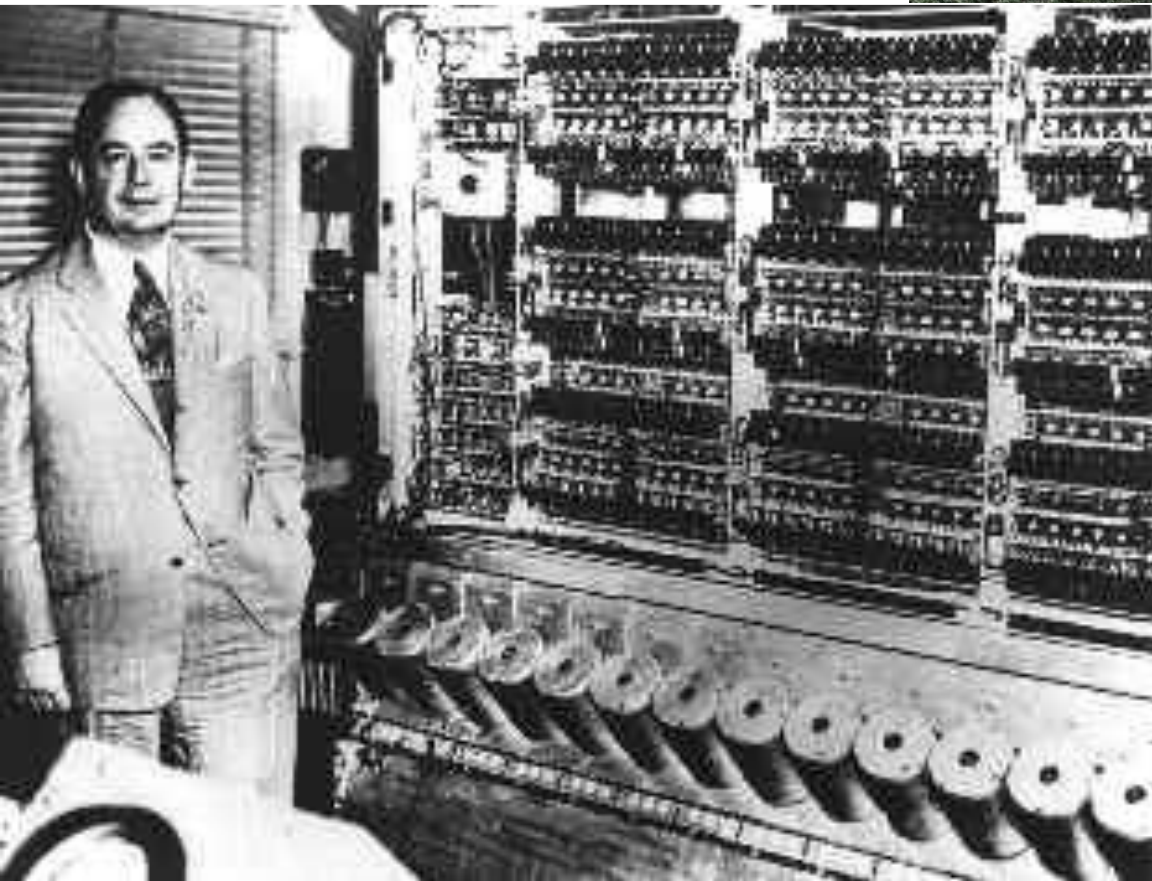
Illustrative
Beispiele

Modelle und Modellierung

Was ist ein Modell?

- "The sciences
 - do not try to explain,
 - they hardly even try to interpret,
 - they mainly make models.
- By a model is meant
 - a mathematical construct which,
 - with the addition of certain verbal interpretations,
 - describes observed phenomena.
- The justification of such a mathematical construct is solely and precisely that it is expected to work.,,

János von Neumann



E HÁZBAN SZÜLETETT
ÉS ÉLT 18 ÉVES KORÁIG
NEUMANN JÁNOS
1903 — 1957

A XX. SZÁZAD EGYIK LEGKIVÁLÓBB
MATEMATIKUSA,
AKI 1951 — 1952 — BEN
AZ AMERIKAI MATEMATIKAI
TÁRSULAT ELNÖKE VOLT.
AZ EMLÉKTÁBLÁT SZÜLETÉSÉNEK
100. ÉVFORDULÓJÁRA
A BOLYAI JÁNOS MATEMATIKAI
TÁRSULAT ÉS
AZ AMERIKAI MATEMATIKAI
TÁRSULAT KÖZÖSEN ÁLLÍTOTTA.



IN THIS HOUSE WAS BORN
AND LIVED UNTIL HE WAS 18
JOHN VON NEUMANN
1903 — 1957

ONE OF THE MOST OUTSTANDING
MATEMATICIANS OF THE 20TH
CENTURY. PRESIDENT OF THE
AMERICAN MATHEMATICAL
SOCIETY IN 1951 — 1952.

THIS MEMORIAL PLAQUE WAS
ERECTED JOINTLY BY THE
JÁNOS BOLYAI MATHEMATICAL
SOCIETY AND THE AMERICAN
MATHEMATICAL SOCIETY ON THE
100TH ANNIVERSARY OF HIS BIRTH.

Was ist ein Modell?

- **Vereinfachtes Bild** eines **Teiles** einer realen oder hypothetischen Welt („des Systems“), das das System in bestimmten Überlegungen **ersetzen kann**
- Entscheidungen:
 - **Welches Teil der Welt?**
 - **Was wird vernachlässigen?**
 - **Wie kann es der Welt entsprechend gemacht werden?**
- Vorteile
 - kleiner (endlich)
 - übersichtlicher

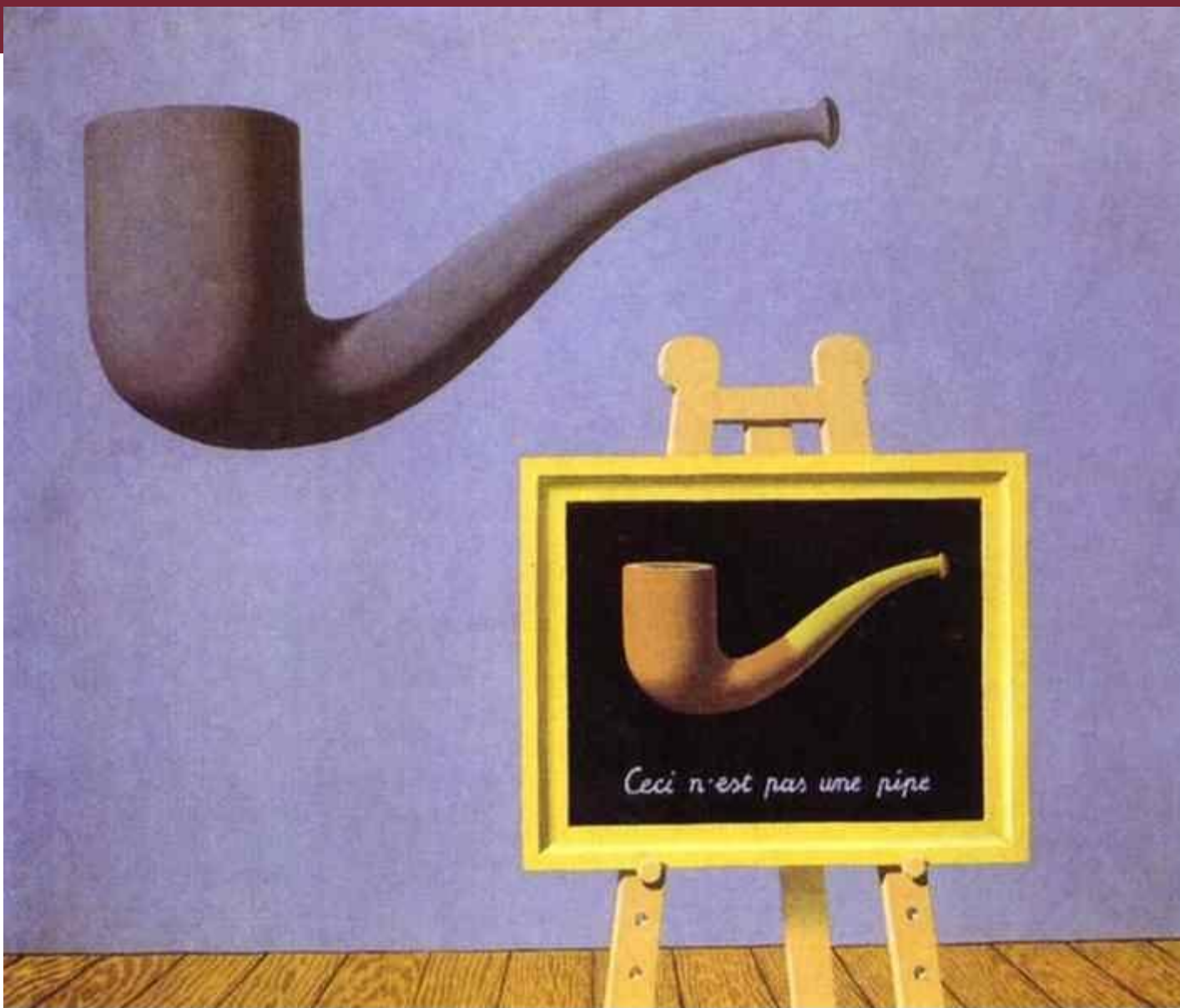
Wann kann es verwendet werden und wann lohnt es sich?

Was ist KEIN Modell?

- Das Modell ist nicht die Wirklichkeit!



- Das Modell ist nicht das Diagramm
 - nur eine Sicht



abcgallery.com - Internet's biggest art collection

Modell vs. Wirklichkeit



Mathematisches Modell vs. Wirklichkeit

- Jedes Modell:
eine geschlossene Welt
 - Effekte, Faktoren
 - Parameter
 - Gültigkeit
- Das Modell funktioniert außer dieser Welt unsicher
- Nicht alles kann im Voraus ausgedrückt werden
 - *menschliche Entscheidung*
 - *generierte Modelle*
- *Validation der Lösung*
- *Für Beantwortung von Fragen gebaut*

- Normale Funktion
 - Randbedingungen
 - Genug Material steht zur Verfügung
 - **Jede** Bestellung termingerecht
 - Zielfunktion:
 - Kosteneffizienz
- Außerordentlicher Fall
 - Randbedingung
 - Materialmangel
 - Zielfunktion:
 1. **Möglichst viele** Bestellung termingerecht
 2. Kosteneffizienz

Beispiel: Sicherheitskritische SW

- Bremsen von Flugzeugen: Radbremse + Schub



1993 Warschau: Lufthansa 2904

- (SW) Sicherung:

(load on both weels) ✘

OR

(one wheel is rotating fast) ✘

→ *(plane on the ground)* ✘

→ **(PILOT CAN BRAKE)** ✘

Wheel sliding

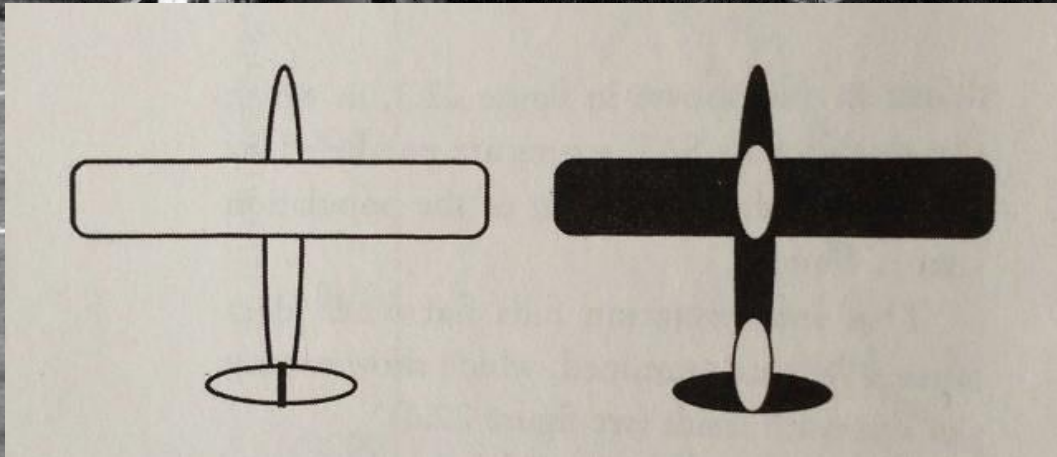
Wheel in the air



Qualität der Modelle

- Realität: eine **offene Welt** \leftrightarrow
Modell: eine **geschlossene Welt**
- **“Treue” eines Modells:**
 - für wahrscheinliche Fälle
 - für kritische Fälle
- Die Umsetzung eines schlechten Modells kann tödlich sein ...

Wald Ábrahám

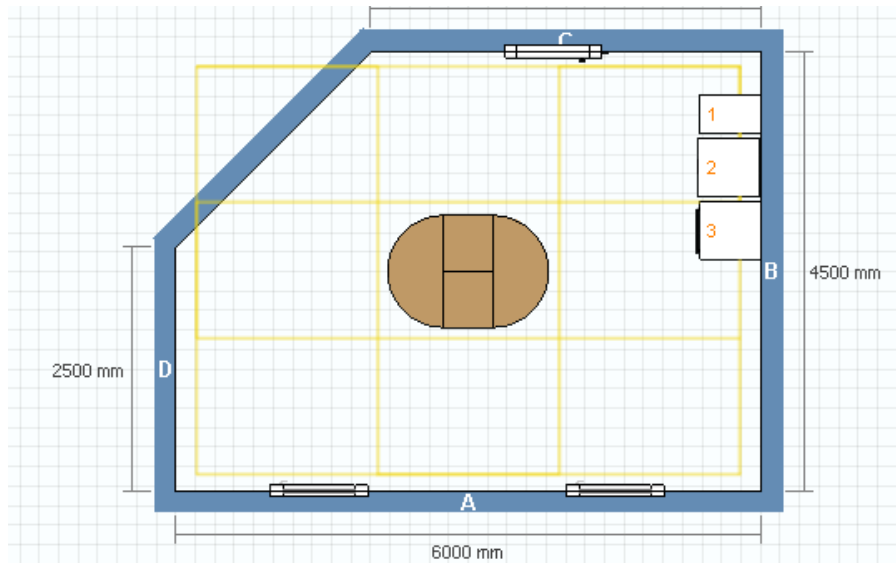


Welchen Sinn hat die Modellierung?

- Ich schreibe Software. Soll ich auch modellieren?
 - Du machst es schon!
 - (Der Quellencode der Software ist auch ein Modell)
 - Was wichtiger sind: **mentale Modelle**
 - Wann müssen diese Modelle ausdrücklich *dokumentiert* werden?
 - Hauptfunktion der Modelle: **Kommunikation**
 - Mensch → Mensch
 - Mensch → Maschine
 - Maschine → Maschine
 - Mensch → er selbst, wenig später
 - z. B. sollte man sich jahrelang auf die Gründe der Entwurfsentscheidungen erinnern

Modellierung in der praktischen Welt?

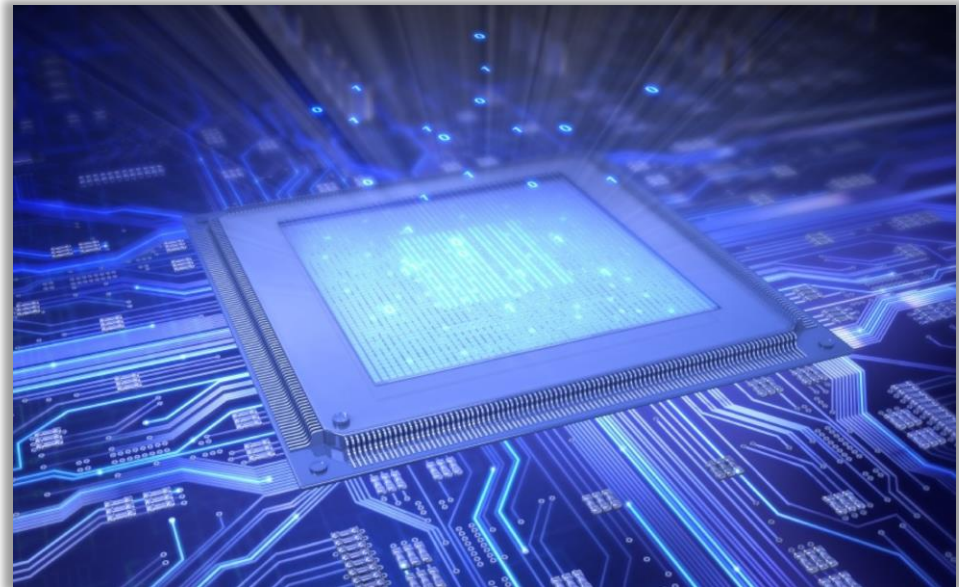
z.B.: webbasierter Küchenplaner
[einer schwedischen Firma]



Auch das ist eine Modellierungssprache! 😊

- VHDL, Verilog – Fachgebietsspezifische Hardwarebeschreibungssprachen

```
1
2 ...
3 ARCHITECTURE Struct OF MyLogic IS
4
5     COMPONENT And2 IS
6         PORT (x, y: IN std_logic;
7             f: OUT std_logic);
8     END COMPONENT;
9
10    COMPONENT CustomHW IS
11        PORT (x: IN std_logic;
12            f: OUT std_logic);
13    END COMPONENT;
14
15    SIGNAL n1, n2: std_logic;
16
17 BEGIN
18     And2_1: And2 PORT MAP ( , , );
19     And2_2: And2 PORT MAP ( , , );
20     CustHW: CustomHW PORT MAP ( , , );
21 END Struct;
22
```



Modellierungssprachen

- Das Ziel ist die Kommunikation
 - Verständnis des Modells ist nötig
 - **Modellierungssprachen** (Wann brauchen wir sie?)
- Syntax
 - „mathematische Struktur“: abstrakte Syntax
 - Darstellung: konkrete Syntax
 - graphische Symbole / Textformat
- Semantik
- Randbedingungen, Einschränkungen
 - Syntaktische Korrektheit, Wohlgeformtheit
 - Entwurfskonventionen (jede Gruppe hat ihre eigene)

Inhalt

Modelle und
Modellierung

Wofür werden Modelle
benutzt?

Grundbegriffe

Illustrative
Beispiele

**WOFÜR WERDEN MODELLE
BENUTZT?**

Modelle in der Ingenieurarbeit

- **Ingenieurarbeit**

(Architekten, Maschinenbauingenieure, Elektroingenieure, ..., Landschaftsarchitekten, ...)

Es ist die altbewährte Ingenieurmethode: **Planung**

Systemplanung als Prozess

Management

- Besorgung, Versorgung
- Planung, Führung, Bewertung

ANSI/EIA 632
Standard

System-
planung

- Definition der Anforderungen
- Definition der Lösung

Ingenieuraufgaben

Produkt
Herstellung

- Implementation
- Benutzbarkeit

Auswertung

- Systemanalyse
- Anforderungsvvalidation
- Systemsverifikation
- Endproduktvalidation

Diese werden
typischerweise
mit Modellen
unterstützt

Systemplanung als Prozess - Analogie



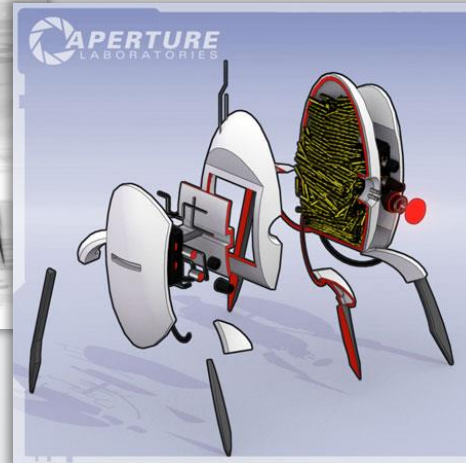
Management

System-
planung

Herstellung
der Produkt

Auswertung

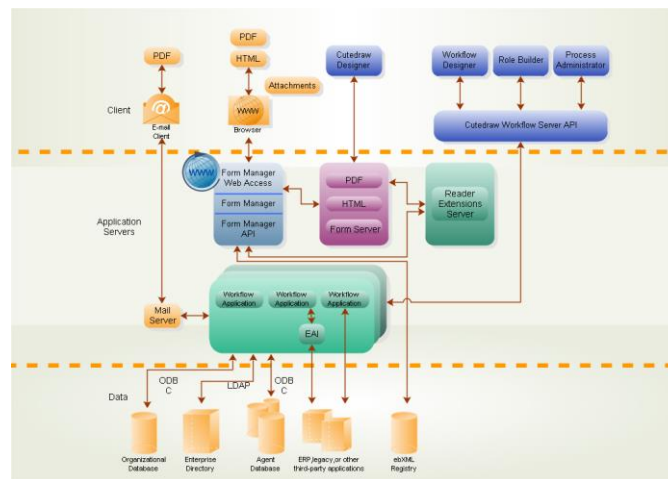
Validation: Bauen wir das richtige Produkt?
Verifikation: Bauen wir das Produkt richtig?



ingenieuraufgaben

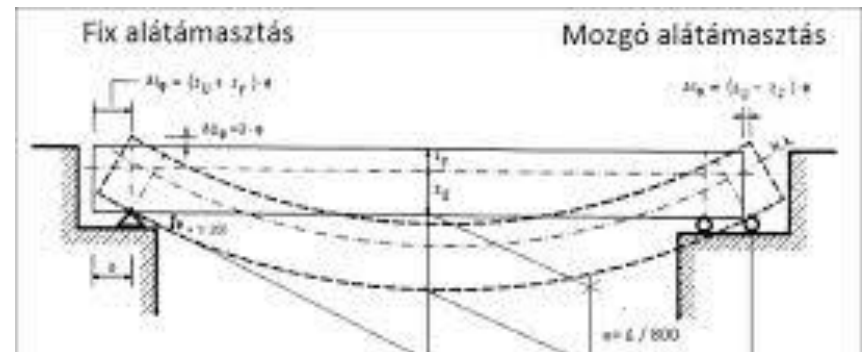
Anwendung – Dokumentation

- Das Modell ist einfacher
 - leichter kommunizierbar, als die ganze Wirklichkeit
 - ständig verfeinerbar (siehe später ...)
- Kommunikation, Veranschaulichung
 - Demonstration (siehe später ...)
 - verständliche textuelle Sprache
 - anschauliches Diagramm
- Unterstützung für Konzept- und Produktentwicklung
 - die Aspekte sind ähnlich
 - „Selbstkommunikation“



Anwendung – Analyse

- Manuell oder (teilweise) automatisiert
- Methode
 - Oberflächliche, statische Analyse
 - Mit dynamischer Zustandsraumexploration – Modellprüfung (model checking)
 - Mit Beweis von formalen Aussagen
- Ziel
 - Untersuchung, Fehlersuche (best effort)
 - Bestätigung von Dienstleistungszuverlässigkeitskriterien (stärker!)
 - Charakteristiken rechnen/planen (z.B. Zeitplanerstellung)



Anwendung – Ableitung

- Manuell oder (teilweise) automatisiert
- Ergebnis
 - Generierung von Programmcode, analysierbarer Sprache, usw.
 - Anderes Modell
 - Verfeinerung, nächste Entwurfsphase
 - Teilaspekte
 - Integration der Modelle
- Es kann eigenschaftserhaltend sein

Anwendung – Simulation

- Demonstration
 - Als Mittel der Kommunikation
- Validation
 - „Ich habe es richtig erbaut ..., aber habe ich das Richtige gebaut?“
- Experimente
 - Überprüfung von bestimmten erwünschten Eigenschaften
 - Messung von quantitativen Eigenschaften
 - Ersatz für in der Wirklichkeit kostspielige Versuche
 - für Eigenschaften, die auf dem theoretischen Weg nicht zu bestimmen sind



Inhalt

Modelle und
Modellierung

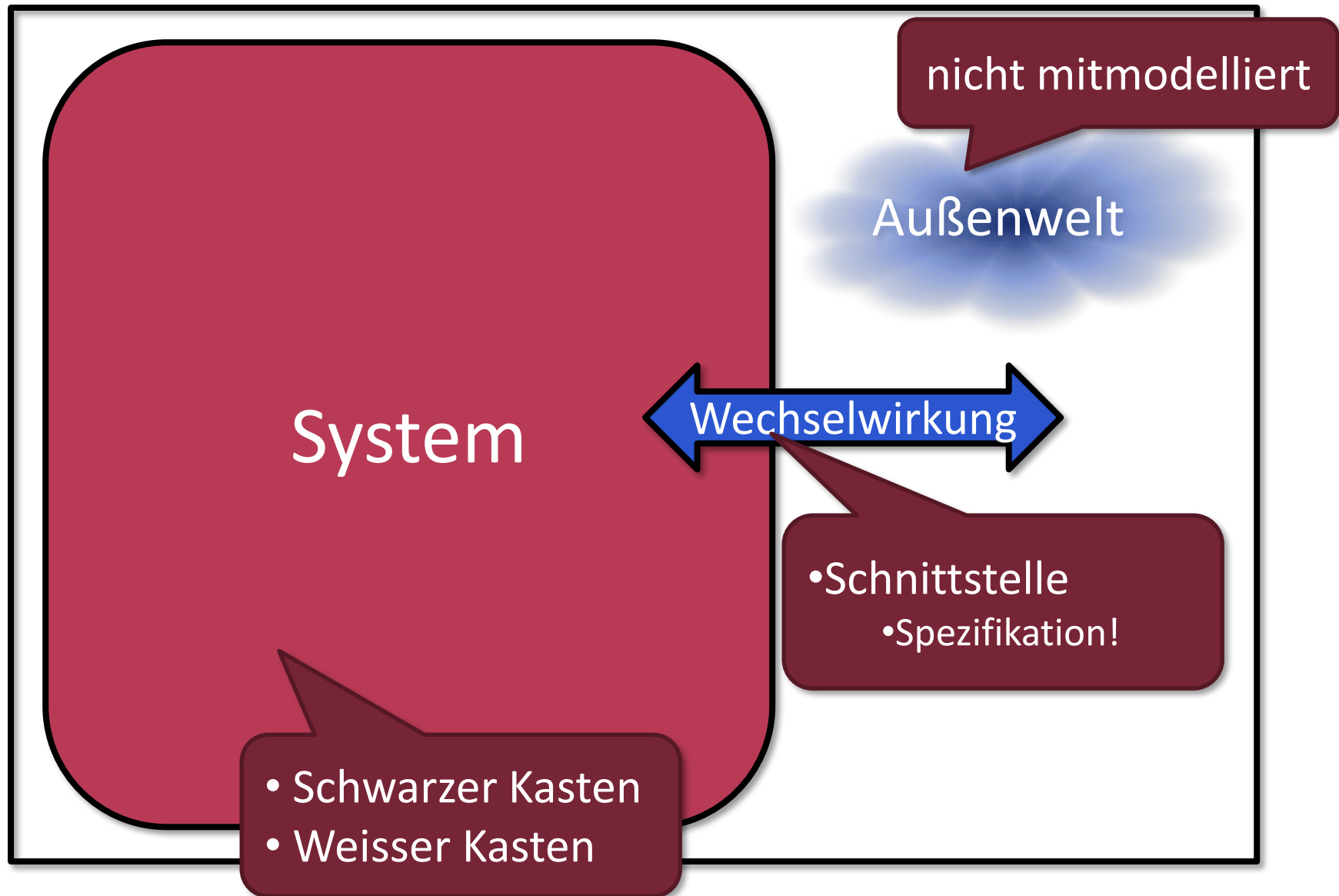
Wofür werden Modelle
benutzt?

Grundbegriffe

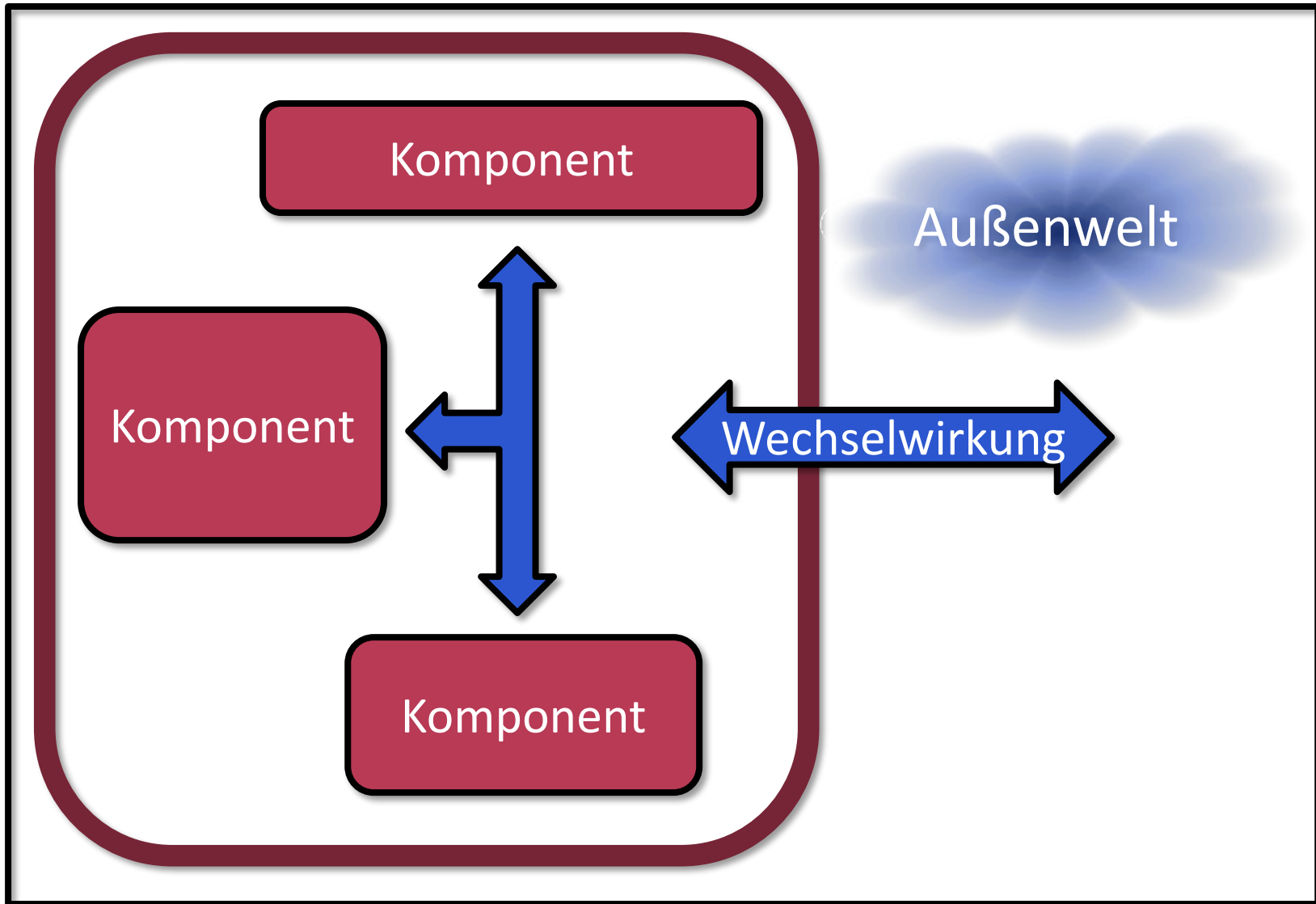
Illustrative
Beispiele

Grundbegriffe der Modellierung

Grundbegriffe – System und Außenwelt



Grundbegriffe – System und Außenwelt



Grundbegriffe – Verfeinerung/Abstraktion

■ *Verfeinerung:*

Bereicherung des Modells mit Einzelheiten

- ... so dass die originale Modellabstraktion erhalten bleibt
- Auf die vorigen Folie wurde eine *hierarchische Verfeinerung*
 - „Kasten auspacken“

■ *(vertikale) Abstraktion:* Inverse der Verfeinerung

■ Nicht nur die Struktur kann verfeinert werden ...

- z.B. Mengenverfeinerung: Wertemengen von Variablen
 - anstatt **gut** / **schlecht**
 - **schnell** / **durchschnittlich** / **langsam** / **mangelhaft** / **gefährlich**

Grundbegriffe – Verfeinerung

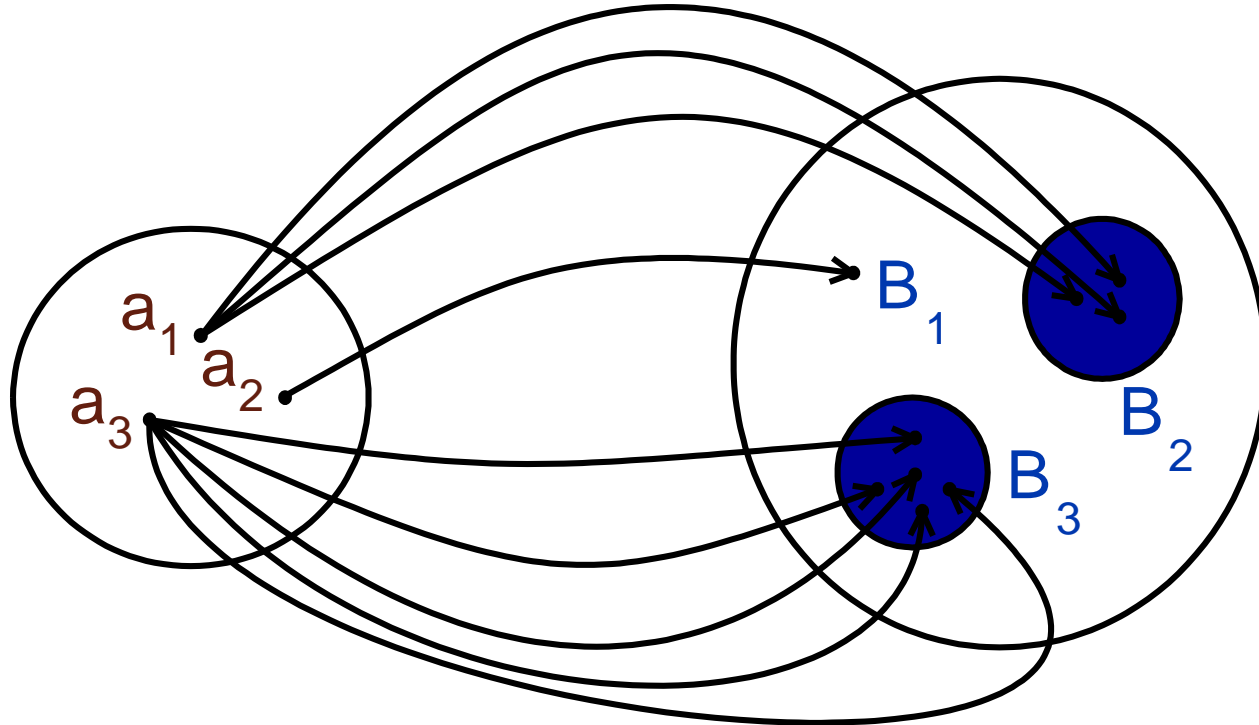


Grundbegriffe – Verfeinerung



Mengenverfeinerung

Zuordnung disjunkter Teilmengen zu den Elementen



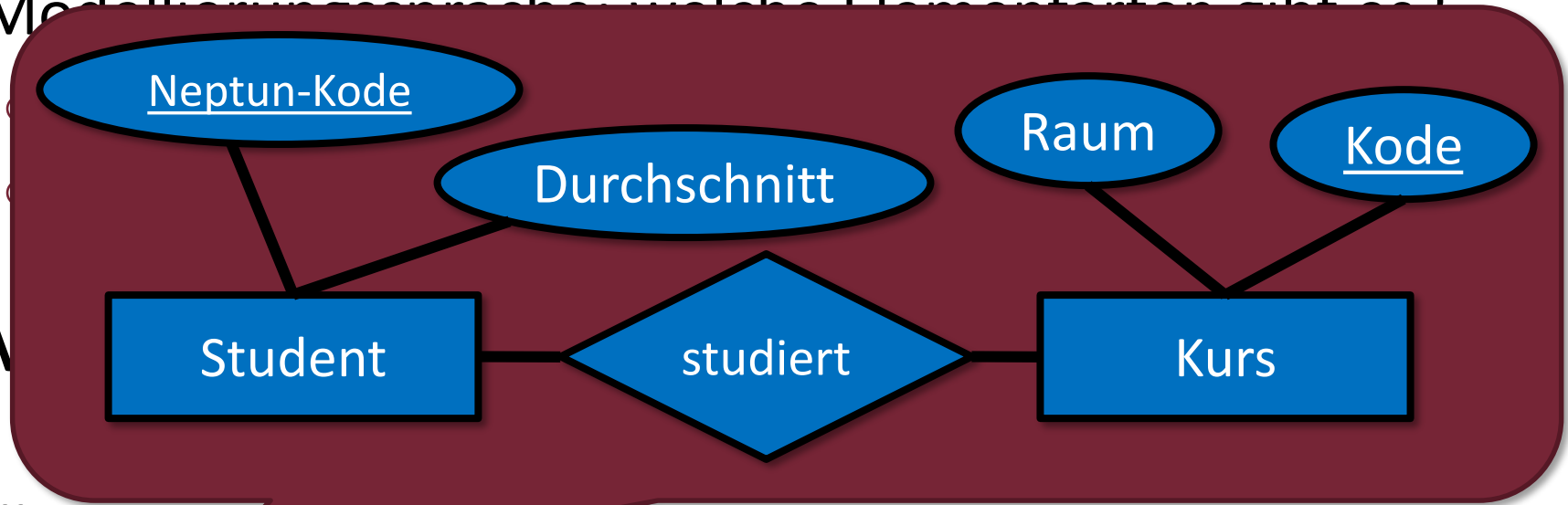
$\forall a_i \in A, R(a_i) \subset B$ so, dass $R(a_i) \cap R(a_j) = \emptyset \quad \forall i, j$

Grundbegriffe - Metamodellierung

- Modellierungssprache: welche Elementarten gibt es?
 - ... was für Relationen kann es unter diesen Elementen geben?
 - ... wie sind die Relationen zwischen diesen Elementarten?
- **Metamodell** = das Modell einer Modellierungssprache

Grundbegriffe - Metamodellierung

- Modellierungsmetamodelle, welche Elementarten existieren?



- M

- Illustration, die jeder kennt

- Individuum-Verbindung (ER) Modell
- UML Class Diagramm → Klassendiagramm
- Datenbankplatte → Datenbankschema mit Relationen
- XML Dokument → XML Schema (oder DTD)
- ...

Inhalt

Modelle und
Modellierung

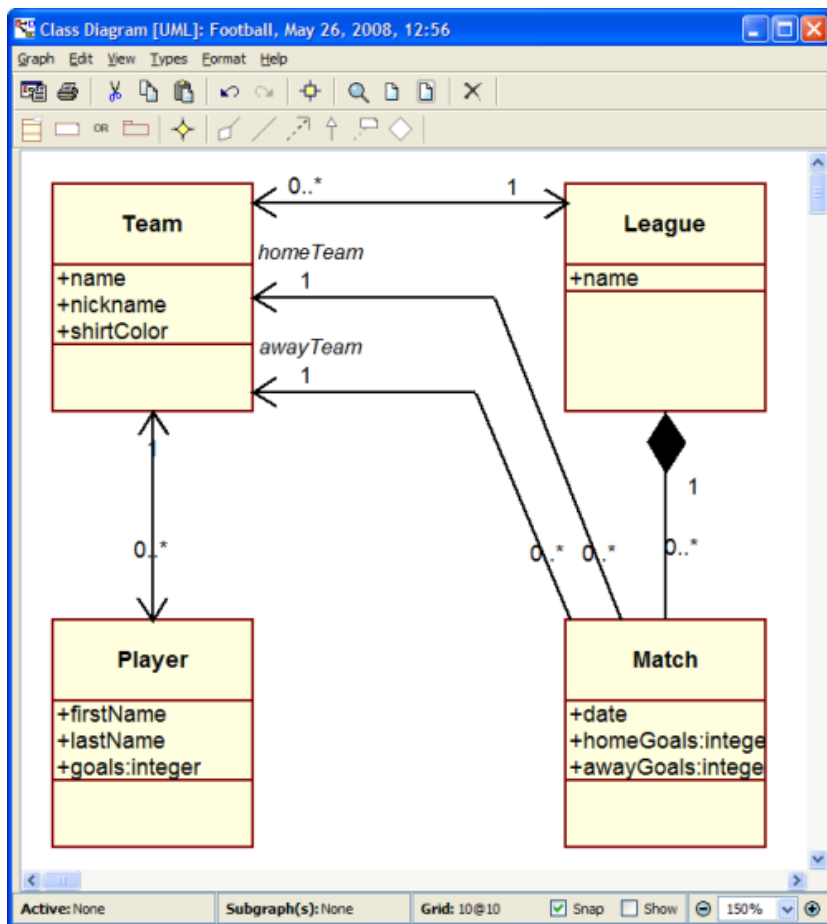
Wofür werden Modelle
benutzt?

Grundbegriffe

Illustrative Beispiele

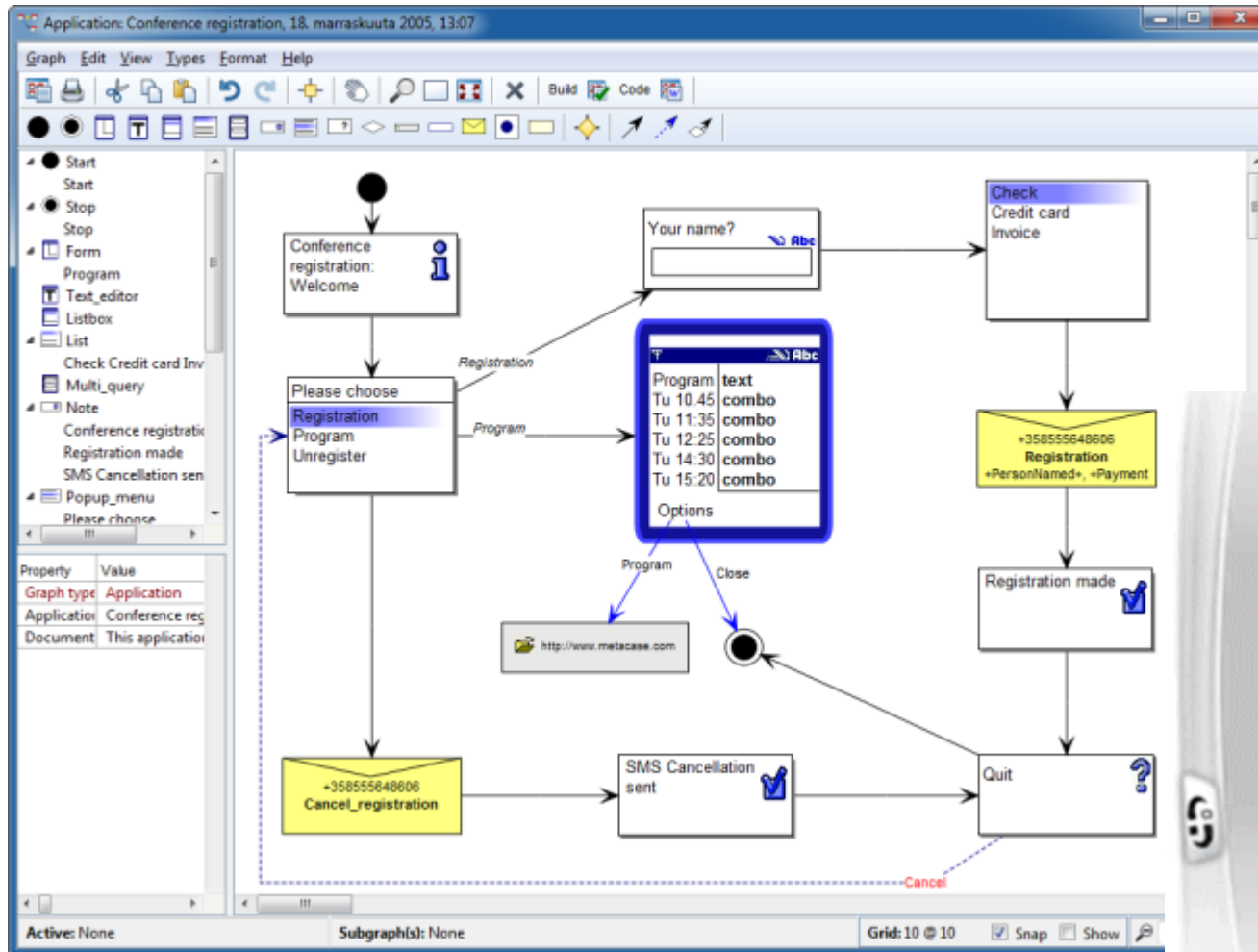
Illustration

Entwicklung einer Webanwendung

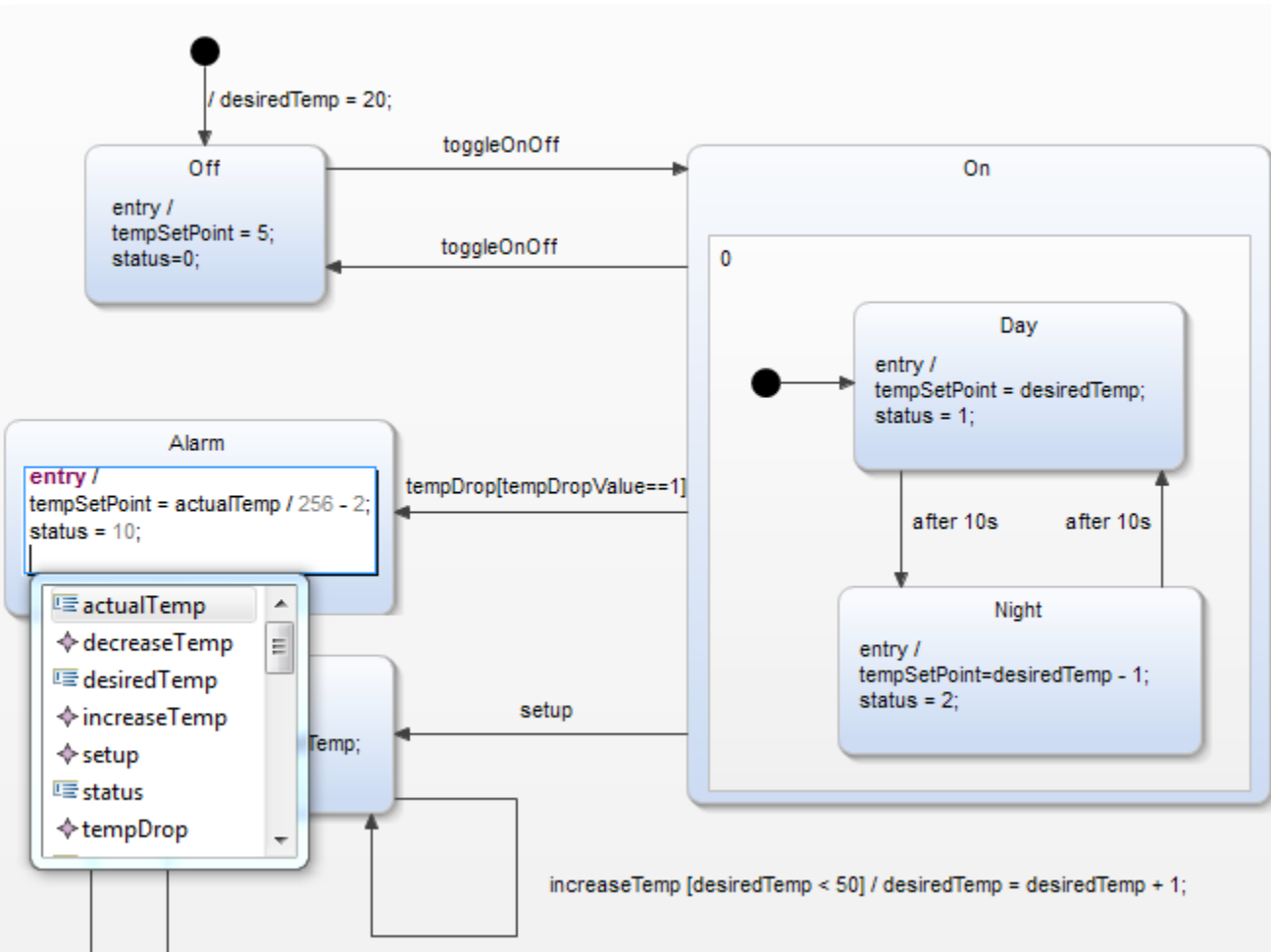


The screenshot shows a web browser window titled "League [Gears] - Mozilla Firefox". The page has a yellow header with the title "League [Gears]". Below the header, there is a form to "Enter a League to store in the database:" with a text input for "name" and an "OK" button. Underneath, there are two sections: "Teams" with "New" and "View All" buttons, and "Matches" with "New" and "View All" buttons. A section titled "4 most recently edited League entries:" shows a table with one entry: "Premiership" with "Edit" and "Delete" buttons. At the bottom, there are "New", "View All", and "Delete All" buttons. A link "Back to top page of Football application." is present. A note at the bottom states: "This page uses Gears to record your entries on the local disk. If you navigate away and revisit this page, all your data will still be here. Try it!"

Entwicklung einer Smartphone-Anwendung



Yakindu - Zustandsdiagramm



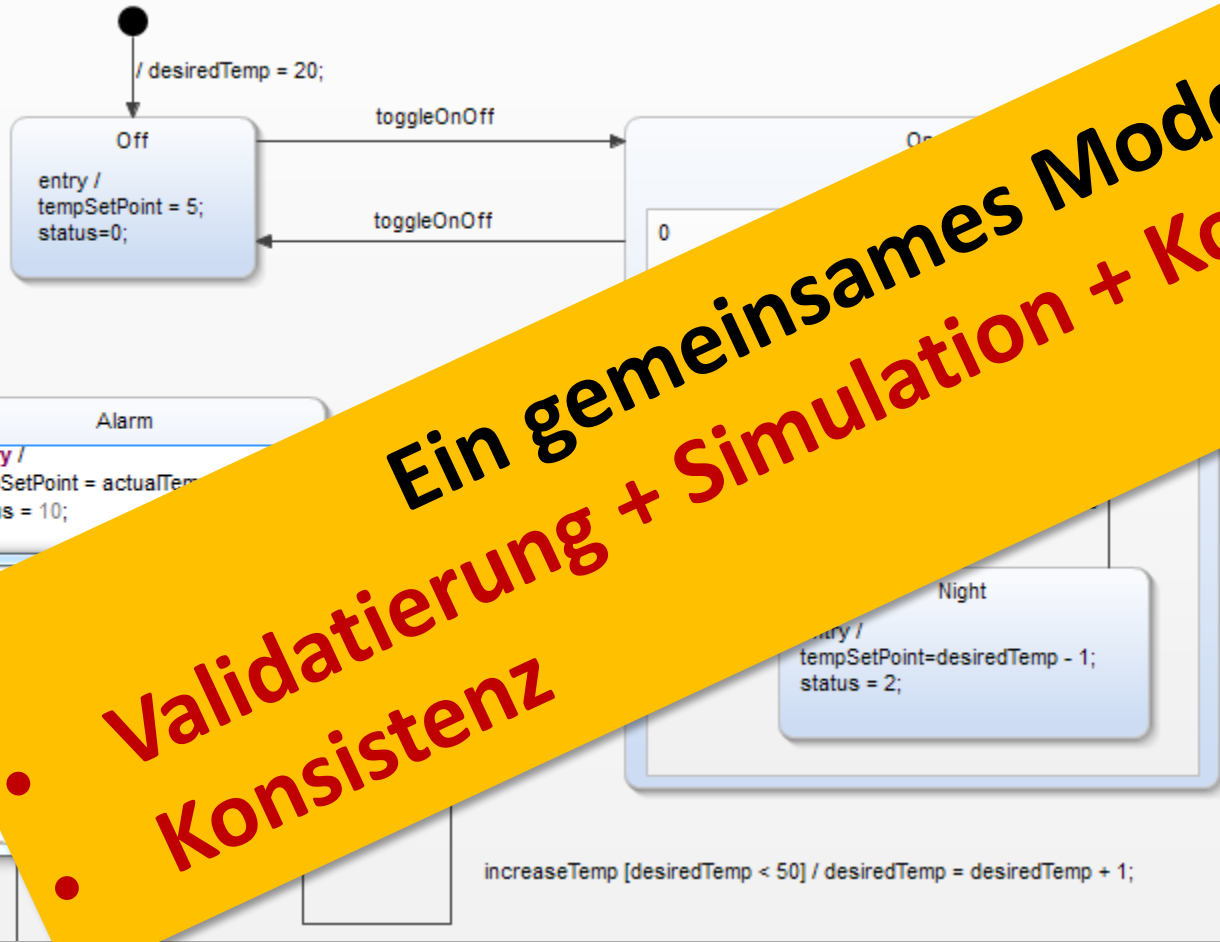
```

import java.io.*;
import java.net.*;
import java.security.*;
import protection;

public class Client {
    public void sendAuthentication(String user,
    OutputStream outStream) throws IOException {
        DataOutputStream out = new DataOutputStream(
        outStream);
        long t1 = (new Date()).getTime();
        byte[] protected1 = ProtectionUtil.encrypt(
        user.getBytes());
        long t2 = (new Date()).getTime();
        double q1 = Math.random();
        double q2 = Math.random();
        byte[] protected2 = ProtectionUtil.encrypt(
        q1 + "" + q2);
        out.writeUTF(user);
        out.writeInt(protected1.length);
        out.write(protected2);
        out.flush();
    }
}

public static void main(String[] args) {
    String host = args[0];
    int port = 7999;
    String user = "John";
    String password = "shh";
    Socket s = new Socket(host, port);
    Client client = new Client();
    client.sendAuthentication(user, s.getOutputStream());
}
    
```

Yakindu - Zustandsdiagramm



Ein gemeinsames Modell

- **Validatierung + Simulation + Kodengenerierung**
- **Konsistenz**

```
public static void main(String[] args) {
    String host = "localhost";
    int port = 7999;
    String user = "John";
    String password = "secret";
    Socket s = new Socket(host, port);

    Client client = new Client(s);
    client.sendAuthenticat
```

KOMPLEXE FALLSTUDIEN

Ausblick

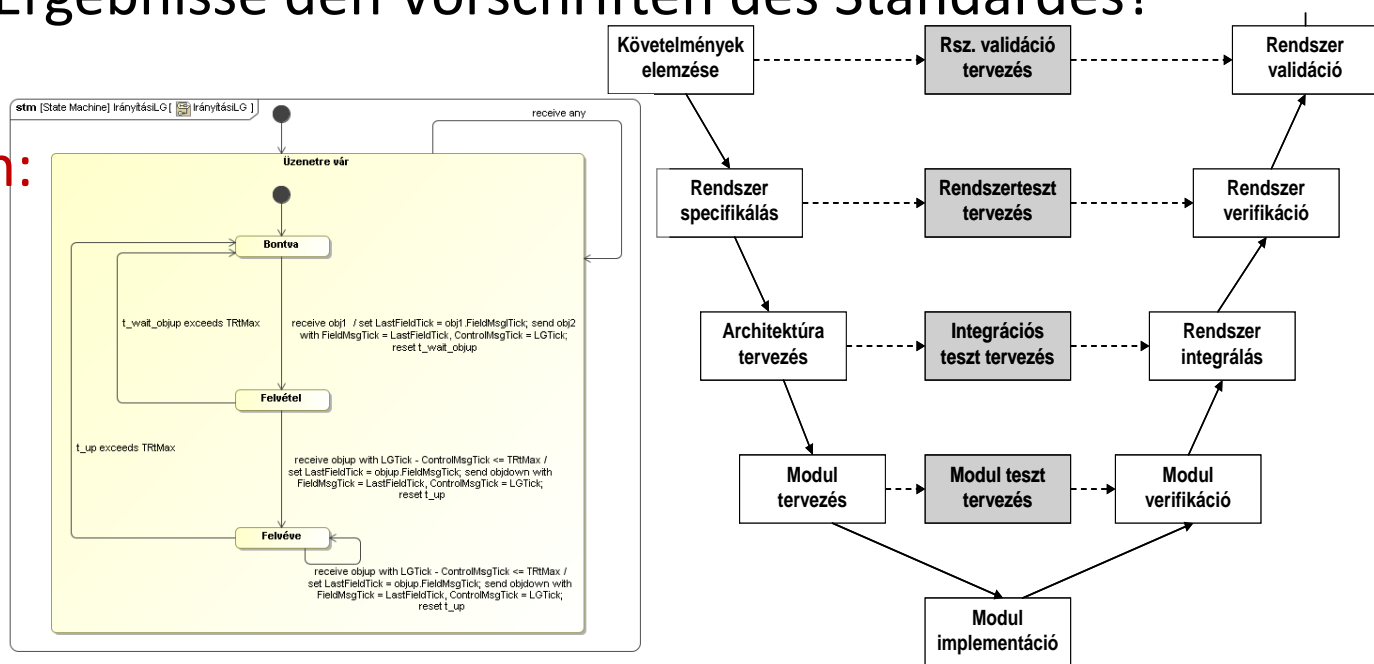
- Entwicklung von Bahnsicherheitssysteme
- Entwicklung von Flugzeugsysteme
- Robotik
- „Wolken“ Rechentechnik

Entwicklung von Bahnsoftware

- **Zu lösendes Problem:**
Überprüfung der Entwicklungsphasen einer SIL-4 Bahnsignalvermittlungsanwendung und dessen Ergebnissen aufgrund des für die Software in Bahnwesen geltende Standardes (EN50128)
- Entsprechen die Ergebnisse den Vorschriften des Standardes?

Herausforderungen:

- systematische Überprüfung
- Modellierung und formale Verifikation



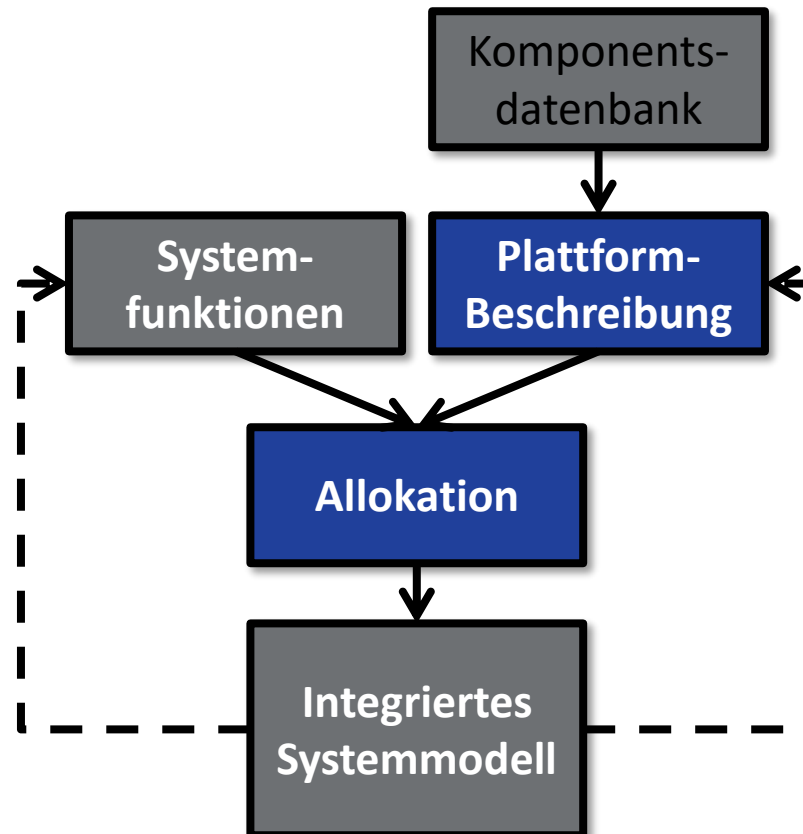
Prozess → Modell → Qualitätssicherung

Planung von Flugzeug

- **Zu lösendes Problem:** Allokation von Softwarekomponenten auf eine individuell geplante integrierte modulare Flugzeugplattform (Kompatibilität mit dem Standard ARINC 653)

Herausforderungen:

- Unterstützung von existierenden Komponentendatenbanken
- Generierung der Kommunikationsarchitektur
- Nachweisbarkeit
- MATLAB Simulink – Eclipse Integration

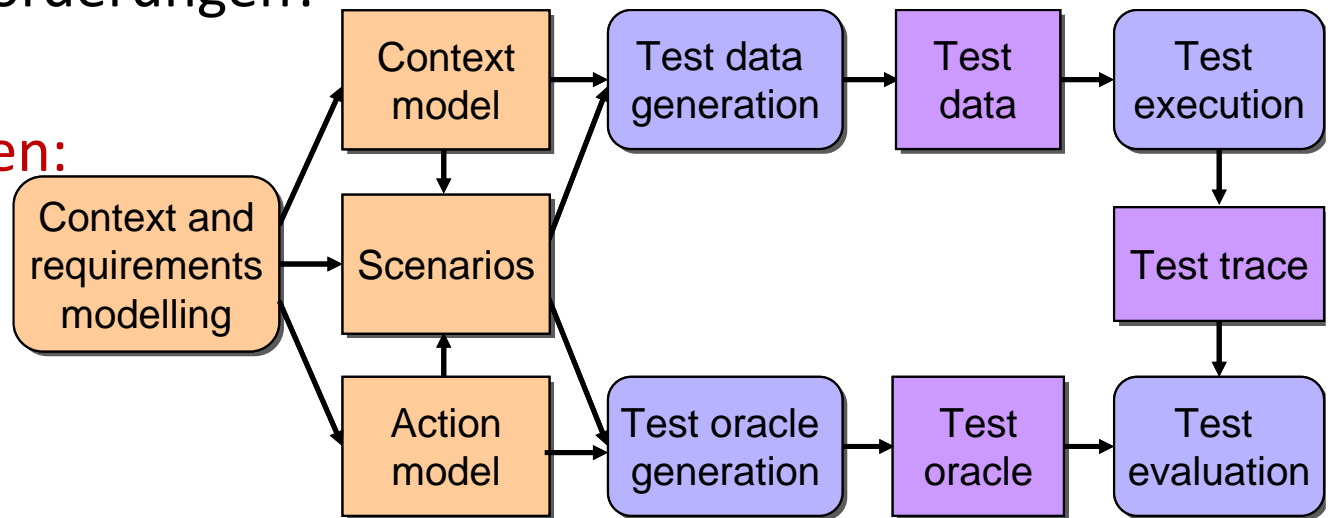


Entwicklung von autonomen Robotern

- **Zu lösendes Problem:** Testen der robusten und sicheren Funktion von umgebungabhängigen, adaptiven, autonomen Robotern
- In welcher Umgebung verletzt das Verhalten die Sicherheitsanforderungen?

Herausforderungen:

- Präzise Bestimmung der Anforderungen (Szenarien)
- Systematische Generierung der Testkonfigurationen (Umgebungen)
- Automatisierung der Testauswertung



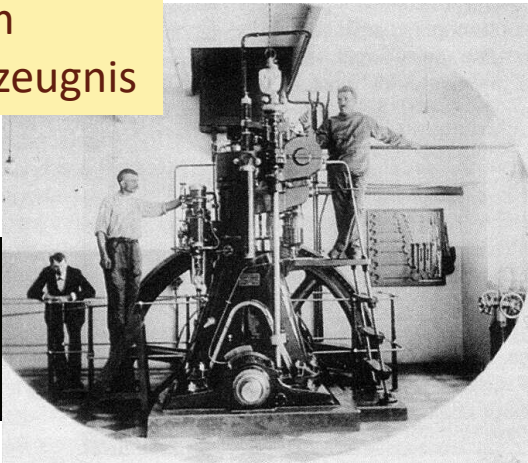
Modell=Spezifikation → Test



„Wolken“ Rechentechnik

Anfang: Verbraucher produziert Energie selbst

Individuelle Infrastruktur zum Spitzenerzeugnis



Die Wolken Rechentechnik ist das selbe:

- Energie = Rechenleistung
- Kraftwerk = Server
- Fernleitung = Internet
- Verteilung, Schutz, Messung, Regelung



Energieversorgung

Effiziente Produktion



Schwellwerk



Konfiguration, Schutz, Redundanz



Lastverteilung, Monitoring, Regelung



Effiziente Lieferung



Verbrauch SLA

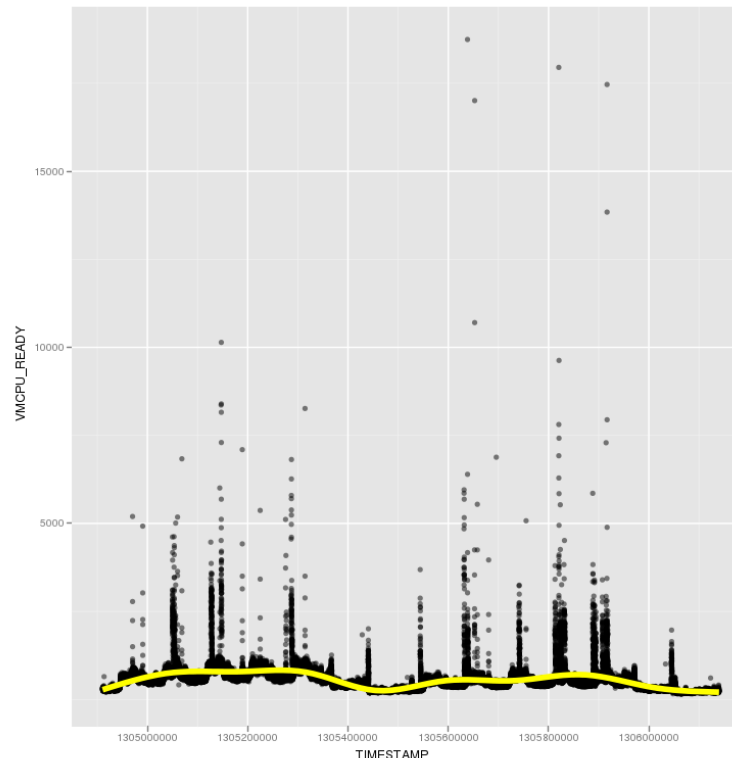


Eine Weltfirma

- **Zu lösendes Problem:** In einem auf privater Wolke basierten Mehrbenutzersystem „schlucken“ manchmal die Rechner der Benutzer
- Ist die Kapazität auch statisch zu wenig, oder ist nur die Dynamik der Verteilung falsch?

Herausforderungen

- Datensatz: 180 Million x 20 000
BIG DATA
- 6000 davon verweist auf Fehler
ANALYSIS SELTENER EREIGNISSE
- Messungsfehler
DATENREINIGUNG



Einige repräsentative Projekte



R3COP (EU ARTEMIS)

- Automated testing of robots
- Robustness and security analysis
- ARTEMIS Innovation Award 2012
- Altogether: 15 EU projects

TRANS-IMA (Embraer)

- Eclipse based development tooling
- HW-SW allocation: avionics architecture
- Integration to the distributed Embraer simulator
- (1st time in Europe)

Data Storage Systems (IBM)

- Supply-chain simulation and optimization
- Prediction of order data
- IBM Vác: Data Storage Systems
- Supply Chain Technology Award 2012

VCL: Virtual Computing Lab

- Open source cloud infrastructure
- Apache project
- Education: lab courses" (BYOD)
- First time in Hungary
- Tempus Award