

Formale Modellierung
Vorlesung 1 vom 16.04.15: Einführung

Christoph Lüth

Universität Bremen

Sommersemester 2015

Organisatorisches

- ▶ Veranstalter:

Christoph Lüth

`christoph.lueth@dfki.de`

MZH 3110, Tel. 59830

- ▶ Termine:

- ▶ Vorlesung: Montag, 16 – 18, MZH 1470

- ▶ Übung: Donnerstag, 14 – 16, MZH 5210

- ▶ Webseite:

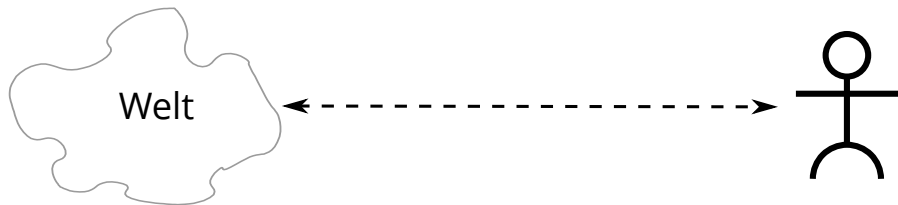
`http://www.informatik.uni-bremen.de/~cxl/lehre/fm.ss15`

Warum formale Modellierung?



Die Vasa, 10. August 1628

Modellierung — Das Prinzip



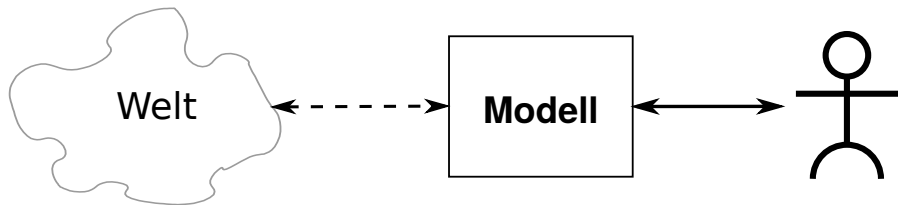
- ▶ Grundlegendes Prinzip der Naturwissenschaften

Modellierung — Das Prinzip



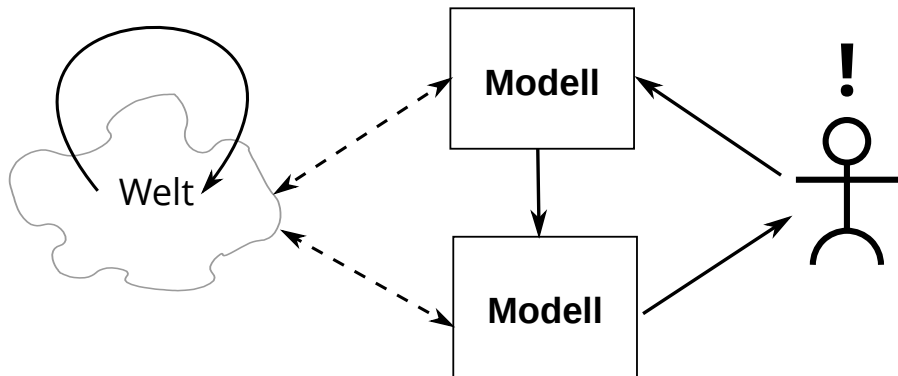
- ▶ Grundlegendes Prinzip der Naturwissenschaften

Modellierung — Das Prinzip



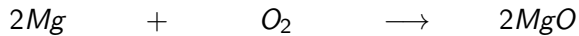
- ▶ **Grundlegendes** Prinzip der Naturwissenschaften

Modellierung — Das Prinzip

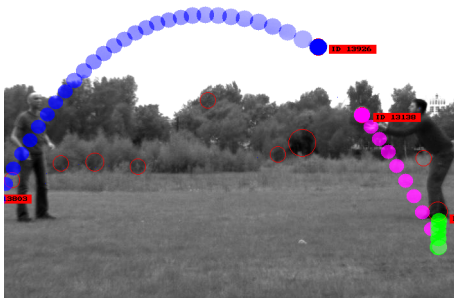


- ▶ **Grundlegendes** Prinzip der Naturwissenschaften

Modellierung — Beispiele

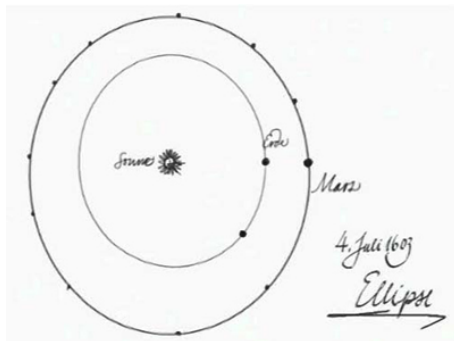


Modellierung — Beispiele



$$x(t) = v_0 t \cos(\beta)$$
$$y(t) = v_0 t \sin(\beta) - \frac{g}{2} t^2$$

Modellierung — Beispiele



$$\left(\frac{T_1}{T_2}\right)^2 = \left(\frac{a_1}{a_2}\right)^3$$

Arten der Modellierung

- ▶ **Physikalischen** Systeme:
 - ▶ Modellierung durch **kontinuierliche** Mathematik (Analysis, DGL)
- ▶ Frage: wie modellieren wir **Programme** und ihr **Verhalten**?

Arten der Modellierung

- ▶ **Physikalischen** Systeme:
 - ▶ Modellierung durch **kontinuierliche** Mathematik (Analysis, DGL)
- ▶ Frage: wie modellieren wir **Programme** und ihr **Verhalten**?
- ▶ Modellierung von **Programmen**: **Berechenbarkeitsbegriff**
 - ▶ Turing-Maschinen, rekursive Funktionen, ...
- ▶ Modellierung der **Eigenschaften**: **formale Logik**
- ▶ Formale Logik ist die **Grundlage** der Modellierung in der Informatik

Was ist mit der UML?

- ▶ **Allgemeine** Modellierungssprache für **problemorientierte** Spezifikationen
- ▶ Ziel ist nicht der **Beweis** von Eigenschaften
- ▶ Nur **bestimmte** Aspekte sind formal
- ▶ Als **Grundlage** nicht geeignet

Lernziele

1. Modellierung — Formulierung von Eigenschaften

Lernziele

1. **Modellierung** — Formulierung von Eigenschaften
2. **Beweis** — Formaler Beweis der Eigenschaften

Lernziele

1. **Modellierung** — Formulierung von Eigenschaften
2. **Beweis** — Formaler Beweis der Eigenschaften
3. **Spezifikation** und **Verifikation** — Eigenschaften von Programmen

Themen

▶ Formale Logik:

- ▶ Aussagenlogik ($A \wedge B$, $A \longrightarrow B$), Prädikatenlogik ($\forall x.P$)
- ▶ Formales Beweisen: natürliches Schließen
- ▶ Induktion, induktive Datentypen, Rekursion
- ▶ Berechenbarkeitsmodelle
- ▶ Die Gödel-Theoreme

▶ Spezifikation und Verifikation:

- ▶ Formale Modellierung von Programmen
- ▶ Temporale Logik
- ▶ Modellprüfung

Der Theorembeweiser Isabelle

- ▶ **Interaktiver** Theorembeweiser
- ▶ Entwickelt in **Cambridge** und **München**
- ▶ Est. 1993 (?), ca. 500 Benutzer
- ▶ Andere: PVS, Coq, ACL-2
- ▶ Vielfältig benutzt:
 - ▶ VeriSoft (D) — <http://www.verisoft.de>
 - ▶ L4.verified (AUS) — <http://ertos.nicta.com.au/research/l4.verified/>
 - ▶ SAMS (Bremen) — <http://www.projekt-sams.de>

Formale Logik

- ▶ Formale (symbolische) Logik: Rechnen mit Symbolen
- ▶ Programme: Symbolmanipulation
- ▶ Auswertung: Beweis
- ▶ Curry-Howard-Isomorphie:
funktionale Programme \cong konstruktiver Beweis

Geschichte

- ▶ Gottlob Frege (1848– 1942)
 - ▶ 'Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens' (1879)
- ▶ Georg Cantor (1845– 1918), Bertrand Russel (1872– 1970), Ernst Zermelo (1871– 1953)
 - ▶ Einfache Mengenlehre: inkonsistent (Russel's Paradox)
 - ▶ Axiomatische Mengenlehre: Zermelo-Fränkel
- ▶ David Hilbert (1862– 1943)
 - ▶ Hilbert's Programm: 'mechanisierte' Beweistheorie
- ▶ Kurt Gödel (1906– 1978)
 - ▶ Vollständigkeitssatz, Unvollständigkeitssätze

Grundbegriffe der formalen Logik

- ▶ **Ableitbarkeit** $\mathcal{Th} \vdash P$
 - ▶ Syntaktische Folgerung
- ▶ **Gültigkeit** $\mathcal{Th} \models P$
 - ▶ Semantische Folgerung
- ▶ **Klassische Logik**: $P \vee \neg P$
- ▶ **Entscheidbarkeit**
 - ▶ Aussagenlogik
- ▶ **Konsistenz**: $\mathcal{Th} \not\vdash \perp$
 - ▶ Nicht alles ableitbar
- ▶ **Vollständigkeit**: jede gültige Aussage ableitbar
 - ▶ **Prädikatenlogik** erster Stufe

Unvollständigkeit

- ▶ Gödels 1. Unvollständigkeitssatz:
 - ▶ Jede Logik, die Peano-Arithmetik formalisiert, ist entweder inkonsistent oder unvollständig.

Unvollständigkeit

- ▶ Gödels 1. **Unvollständigkeitssatz**:
 - ▶ Jede **Logik**, die **Peano-Arithmetik** formalisiert, ist entweder **inkonsistent** oder **unvollständig**.
- ▶ Gödels 2. **Unvollständigkeitssatz**:
 - ▶ Jede **Logik**, die ihre eigene **Konsistenz** beweist, ist **inkonsistent**.

Unvollständigkeit

- ▶ Gödels 1. **Unvollständigkeitssatz**:
 - ▶ Jede **Logik**, die **Peano-Arithmetik** formalisiert, ist entweder **inkonsistent** oder **unvollständig**.
- ▶ Gödels 2. **Unvollständigkeitssatz**:
 - ▶ Jede **Logik**, die ihre eigene **Konsistenz** beweist, ist **inkonsistent**.
- ▶ Auswirkungen:
 - ▶ **Hilbert's Programm** terminiert nicht.
 - ▶ **Programme** nicht vollständig spezifizierbar.
 - ▶ **Spezifikationsprachen** immer **unvollständig** (oder uninteressant).

Unvollständigkeit

- ▶ Gödels 1. **Unvollständigkeitssatz**:
 - ▶ Jede **Logik**, die **Peano-Arithmetik** formalisiert, ist entweder **inkonsistent** oder **unvollständig**.
- ▶ Gödels 2. **Unvollständigkeitssatz**:
 - ▶ Jede **Logik**, die ihre eigene **Konsistenz** beweist, ist **inkonsistent**.
- ▶ Auswirkungen:
 - ▶ **Hilbert's Programm** terminiert nicht.
 - ▶ **Programme** nicht vollständig spezifizierbar.
 - ▶ **Spezifikationssprachen** immer **unvollständig** (oder uninteressant).
 - ▶ Mit anderen Worten: **Es bleibt spannend**.

Nächste Woche

- ▶ Aussagenlogik
- ▶ Erstes Übungsblatt