Formale Modellierung
Vorlesung vom 16.04.12: Einführung

Till Mossakowski & Christoph Lüth

Universität Bremen

Sommersemester 2012

Organisatorisches

- Veranstalter:

Till Mosskawski	Christoph Lüth
till.mossakowski@dfki.de	christoph.lueth@dfki.de
Cartesium 2.51, Tel. 64216	MZH 3110, Tel. 59830

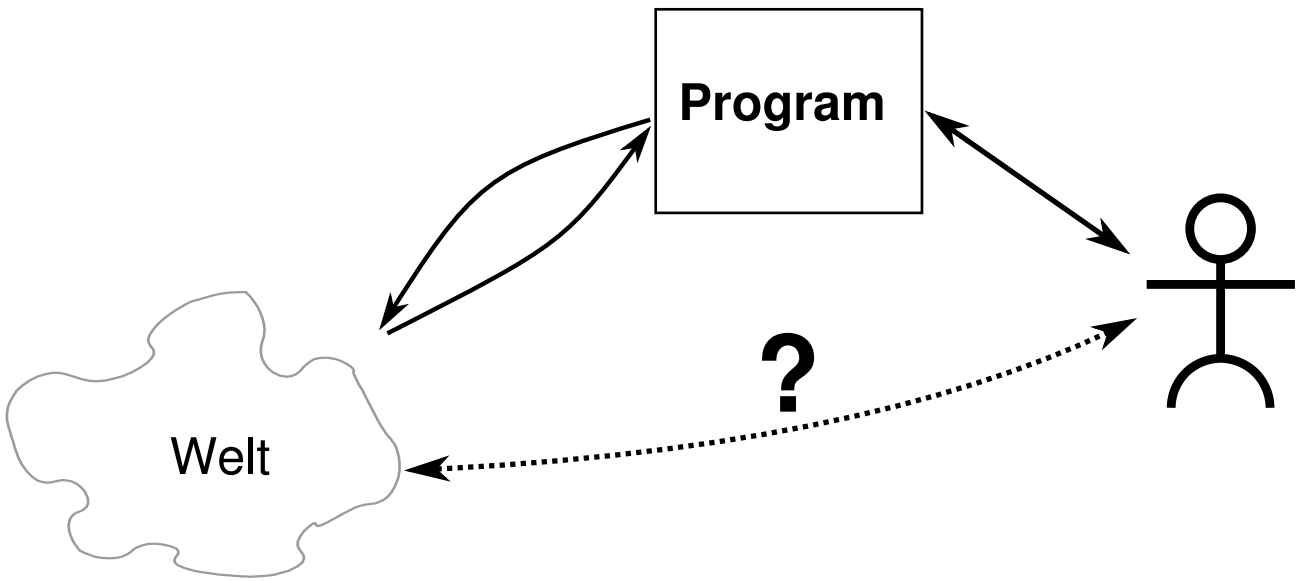
- Termine: Vorlesung: Montag, 14 – 16, Cartesium 2.43
  Übung: Donnerstag, 8 – 10, GW1 C1070

Therac-25

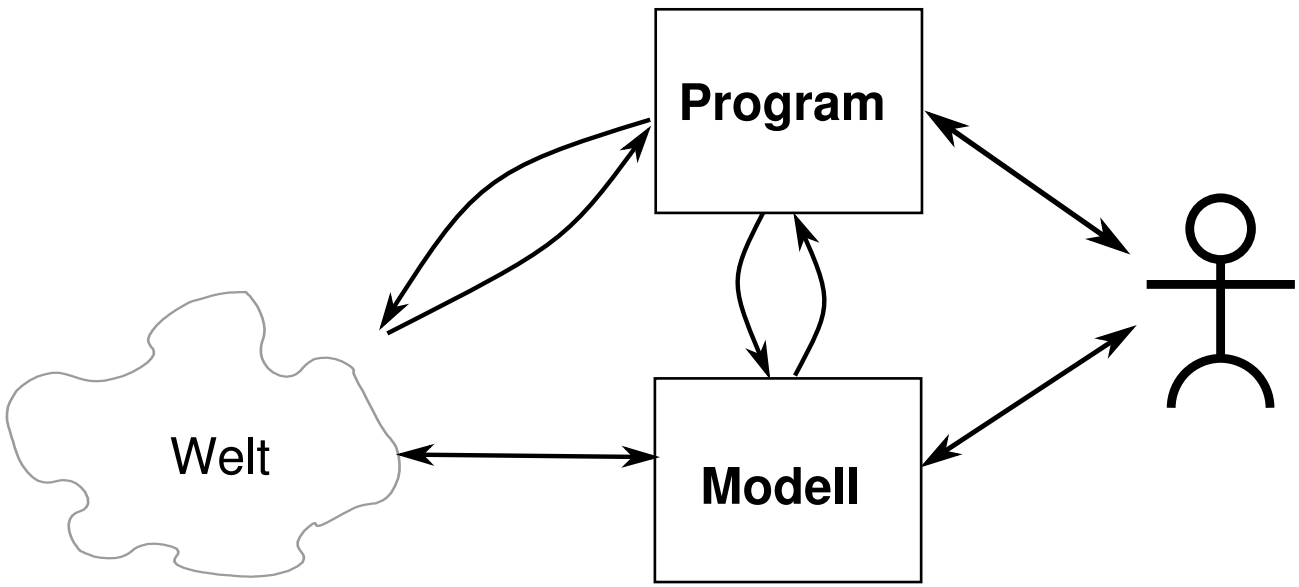
- Neuartiger Linearbeschleuniger in der Strahlentherapie.
  - Computergesteuert (PDP-11, Assembler)
- Fünf Unfälle mit Todesfolge (1985– 1987)
  - Zu hohe Strahlendosis (4000 – 20000 rad, letal 1000 rad)
- Problem: Softwarefehler
  - Ein einzelner Programmierer (fünf Jahre)
  - Alles in Assembler, kein Betriebssystem
  - Programmierer auch Tester (Qualitätskontrolle)

Ariane-5

Das Problem



Das Problem



Lernziele

1. Modellierung — Formulierung von Eigenschaften

Lernziele

1. Modellierung — Formulierung von Eigenschaften

2. Spezifikation — Eigenschaften von Programmen

Lernziele

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

Lernziele

1. Modellierung — Formulierung von Eigenschaften
2. Spezifikation — Eigenschaften von Programmen
3. Verifikation — Beweis der Eigenschaften
4. Vertrautheit mit aktuellen Techniken: JML, UML, Temporallogik

Techniken der Modellierung und Spezifikation

- Annotationen an den Code
  - JML für Java: Abstraktion von konkreter Implementation
- Abstraktion vom Quellcode:
  - Abstrakte Programmmodelle: Zustandsautomaten, Klassendiagrame
- Abstraktion vom Ausführungsmodell:
  - Hier: Nebenläufigkeit (Temporallogik)

Java Modeling Language (JML)

- Zentral: funktionale Korrektheit
- Design by contract
- Spezifikation nahe am Code
- Hoare-Logik (pre/post conditions)
- Werkzeuge: ESC/Java2, Mobius

Unified Modeling Language (UML)

- allgemeine Modellierungssprache
- Spezifikation problemorientierter
- Übersetzung in verschiedene Programmiersprachen möglich
- Nur bestimme Aspekte sind formal
- hier: State Machines und Object Constraint Language (OCL)
- Werkzeuge: Hugo/RT und argouml

Temporallogik

- Spezifikation nebenläufiger Programme
- Sicherheits- und Fairness-Eigenschaften
- Werkzeug: nuSMV Modelchecker

Aufbau der Vorlesung

- Plan:
  - Woche 01 – 05 : Codeannotation mit JML
  - Woche 06 – 11 : Modellierung mit UML
  - Woche 11 – 14 : Modellchecking mit NuSMV
- Vorlesung und Übung dynamisch im Wechsel
- ca. 5 Übungsblätter, Gruppengröße max. 3
- Übungen werden vorgestellt und besprochen
- Scheinkriterien:
  - Übungsblätter bearbeiten und Fachgespräch
  - oder mündliche Prüfung