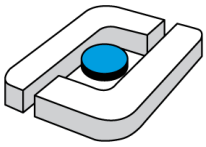
Integration von Model-Driven Development und formaler Verfikation in den Softwareentwicklungsprozess

Eine Fallstudie mit einem 3D-Tracking-System

Dipl.-Inform. Christian Ammann

Fachhochschule Osnabrück
Fakultät für Ingenieurwissenschaften und Informatik

17.6.2010



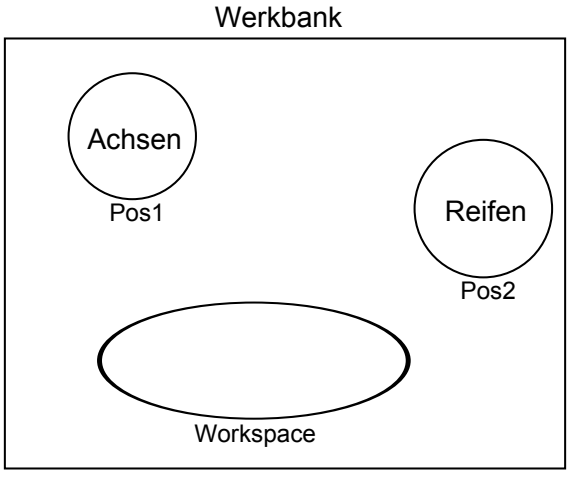
Inhalt

Einleitung

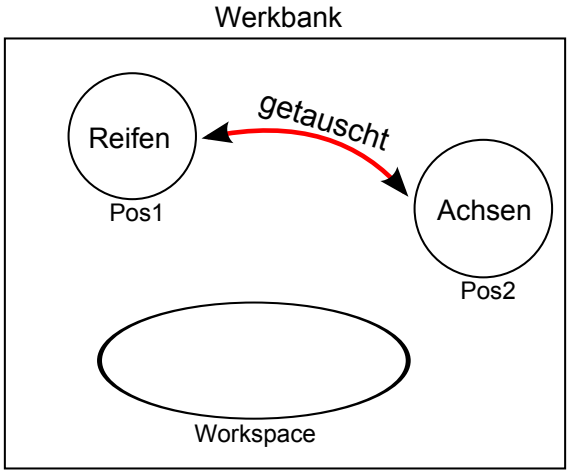
- 3D-Tracking System entwickelt von der *soft2tec GmbH* für die *Otto Kind AG*.
- Überwacht Handmontage und meldet Fehler.



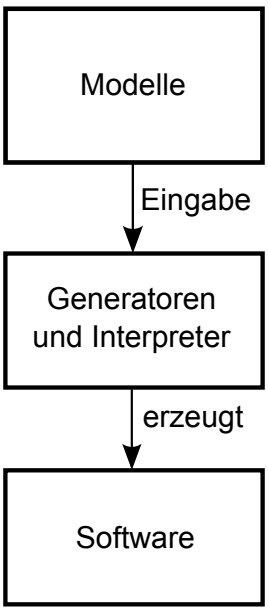
Funktionsweise



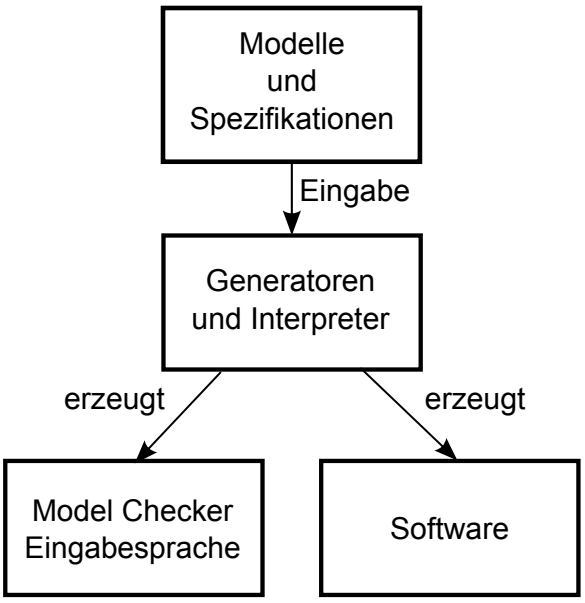
Problem - Schichtwechsel verändert Arbeitsplatz



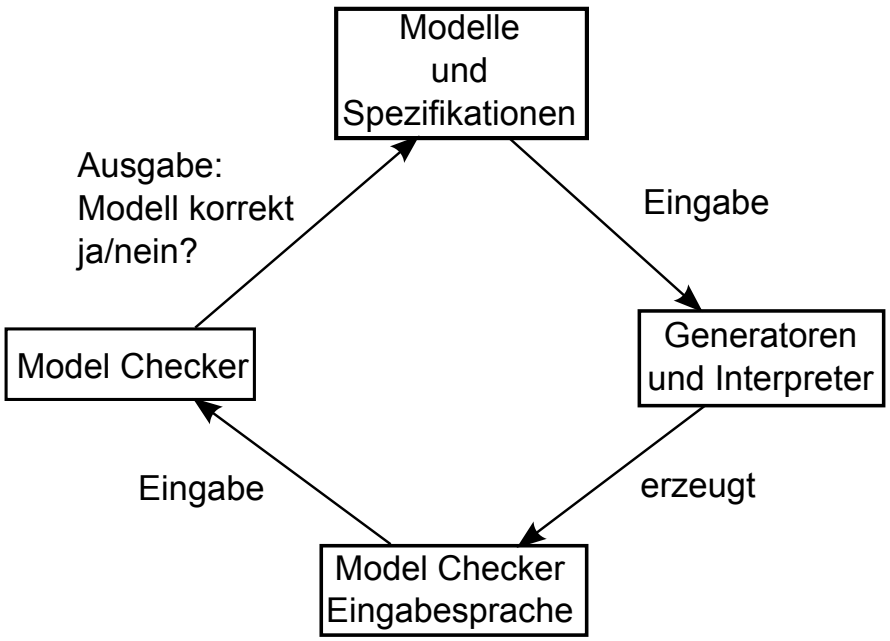
MDSD Ansatz



MDSD mit Model Checker



Vorteil von MDSD mit Model Checker



Modellierung von assyControl als DSL

```
component{name=axis}
component{name=tire}
recipe{
    axis, tire, tire
}

position{name=Box1}
position{name=Work1 start=1}
position{name=Box2}
```


assyControl MDSD Verifikation von UML-Statecharts Fallstudie Fazit

Modellierung von assyControl als DSL

```
component{name=axis}
component{name=tire}
recipe{
    axis, tire, tire
}

position{name=Box1}
position{name=Work1 start=1}
position{name=Box2}
```

Christian Ammann - FH Osnabrück 9 / 23

Modellierung von assyControl als DSL

```
component{name=axis}
component{name=tire}
recipe{
    axis, tire, tire
}

position{name=Box1}
position{name=Work1 start=1}
position{name=Box2}
```


Vorgehen

- DSL wird in ein Klassendiagramm überführt.
- DSL wird in eine Promela-Beschreibung überführt.
- DSL bildet nur strukturelle Aspekte ab.
- Fehler auf Verhaltensebene können nicht durch anpassen des Modells repariert werden.

Java Pathfinder

- Verifikation von Java-Anwendungen.
- Früher: Übersetzung von Java-Sourcecode nach Promela.
- Heute: Verifikation von Java-Bytecode.
- Verifiziert unveränderte Java-Applikationen und speziell für den JPF angepasste Modelle.
- Besitzt UML-Statechart-Extension.

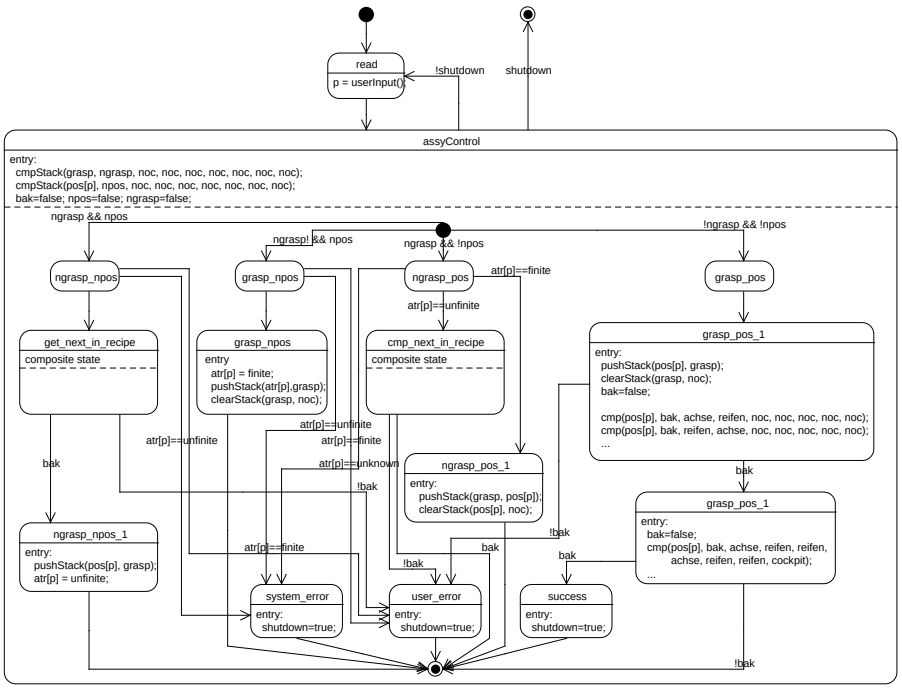
Hugo

- Übersetzt UML-Statecharts nach Java.
- Übersetzt UML-Statecharts nach Promela.
- Vorteil: Übersetzung von UML-Statecharts gut erforscht und Forschungsergebnisse werden von Hugo umgesetzt.
- Verschiede Eingabeformate.

Open ArchitectureWare

- Framework für modellgetriebene Software Entwicklung.
- Erlaubt das Erzeugen von Codegeneratoren.
- Früher eigenständig, jetzt in das *Eclipse Modeling Framework* integriert.
- Xtext: Erlaubt das Erstellen von Grammatiken für eine DSL.
- Xpand: Übersetzt eine DSL in eine andere Sprache.

Fallstudie - assyControl Erweiterung



Probleme: JPF

- Kompletter Zustandsraum konnte nicht abgesucht werden.
- Verglichen mit SPIN sehr hohe Laufzeit.
- UML-Statechart-Extension nicht ausreichend dokumentiert.
- Gebunden an die Java-Platform.

Probleme: Hugo und oAW

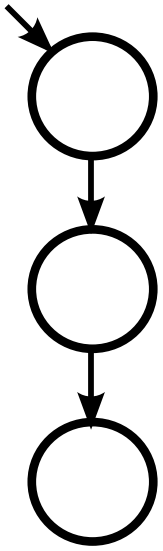
- Kompletter Zustandsraum konnte nicht abgesucht werden.
- Sehr viele Artefakte im automatisch generierten Promela-Code.
- Nicht optimiert durch einfügen von beispielsweise *d_step* Instruktionen.
- Spin unterstützt keine Symmetrie.
- Spin unterstützt keine abstrakten Datentypen wie zum Beispiel *Stack*.

Optimierung 1: Zusammenfassen von Zuständen

Promela

```
example:
  a=7*x;
  b=4/y;
  c=true;
```

SPIN Verifier

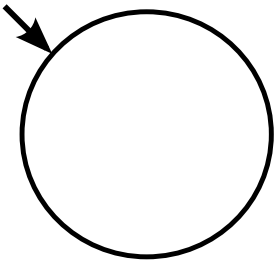


Optimierung 1: Zusammenfassen von Zuständen

Promela

```
example:
  d_step{
    a=7*x;
    b=4/y;
    c=true;
  }
```

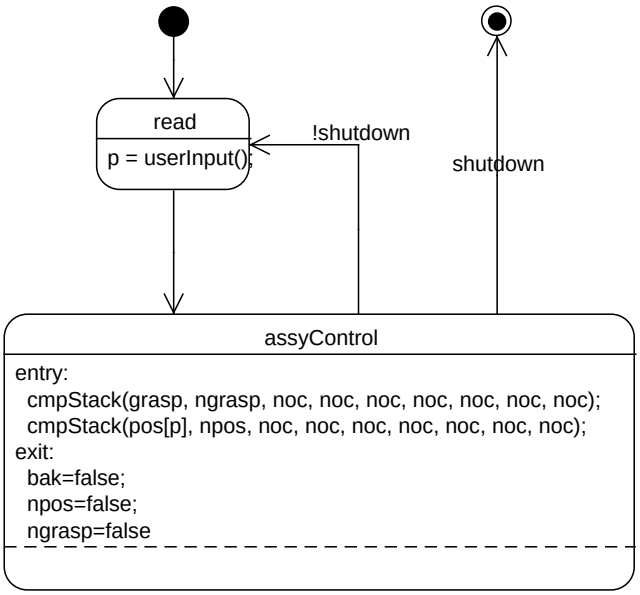
SPIN Verifier



Optimierung 2: Hilfsvariablen

- Hilfsvariablen nicht relevant für den Zustandsvektor.
- Möglichkeit 1: Deklarieren als *hidden* in einem *d_step* Block.
- Möglichkeit 2: Zurücksetzen nach Verlassen eines *d_step* Blocks.

von Hand optimierte assyControl Erweiterung



Zusammenfassung

- Verknüpfung von MDSD und formaler Verifikation.
- Modelle: UML-Statecharts
- Model Checker: SPIN
- Bestehende Tools: JPF, oAW und Hugo
- Fallstudie: 3D-Tracking System

Fazit und Ausblick

- Keine der bestehenden Lösungen konnte Zustandsraum komplett absuchen.
- Optimierungen von Hand notwendig.
- Untersuchung: SPIN erweitern für Symmetrie (Scalarset variables).
- Automatische Optimierung von UML-Statecharts im Hinblick auf die Erkennung von Hilfsvariablen und *d_dstep* Instruktionen .

Ende

Fragen?