Formale Methoden für eingebettete Systeme
Teil 1: Einführung

Prof. Dr. Stefan Kowalewski

Lehrstuhl Informatik XI
RWTH Aachen

Sommersemester 2005

Organisatorisches (1)

- Sprache ist Deutsch.
- Adressatenkreis
  - Diplom Informatik / Praktische Informatik
  - Diplom Informatik / Vertiefungsgebiet Software für eingebettete Systeme
  - MSc Software Systems Engineering
  - MSc Software Systems Engineering / Specialization Area Embedded Systems
  - MSc Media Informatics
  - ERASMUS

You should be able to follow a **German** lecture

Organisatorisches (2)

- **Website:**

  www-i11.informatik.rwth-aachen.de,
  Lehre/Sommersemester 2005,
  Formale Methoden für eingebettete Systeme

  - Ankündigungen
  - Folien / ggfls. handschriftliche Manuskripte
  - Links
  - Forum

RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

Organisatorisches (3): Termine

- **Vorlesung:**
  - **Donnerstag, 10:00 – 11:30, AH I**
- **Übung:**
  - **Mittwochs, 15:15 – 16:45, AH III, 14-tägig, Start: 27.04.05**
- **Keine Vorlesung:**
  - **5. Mai 2005 (Christi Himmelfahrt)**
  - **19. Mai 2005 (Exkursionswoche)**
  - **26. Mai 2005 (Fronleichnam)**
- **Schriftliche Klausur (Übungsschein, MSc/ERASMUS 4 ECTS):**
  - **21. Juli 2005, 10:00 – 11:00**

RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

Betreuer

Bastian Schlich

Raum 2U07 (Altbau),

Email: schlich at informatik.rwth-aachen.de

RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

Professor

- **Prof. Dr. Stefan Kowalewski**
  - 1990 Diplom Elektrotechnik, Universität Karlsruhe
  - 1995 Promotion, Fachbereich Chemietechnik, Universität Dortmund
  - 2000 – 2003 Robert Bosch GmbH, Forschung und Vorausentwicklung, Software-Technologie, Frankfurt am Main
  - 2003 Habilitation in Automatisierungs- und Sicherheitstechnik, Universität Dortmund
  - Seit 11/2003 Lehrstuhl Informatik XI, RWTH Aachen

RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

Verwandte Veranstaltungen

Von Kollegen:

- Modelchecking (Prof. Thomas, als Aufzeichnung auf i7-Website)

In diesem Semster von Informatik XI:

- Design of Embedded Software, Dienstag, 10:00, AH II
- Safety and Reliability Engineering, Donnerstag, 14:30, B-IT, Bonn, und 2317, Aachen, sowie Aufzeichnung auf iXI-Website

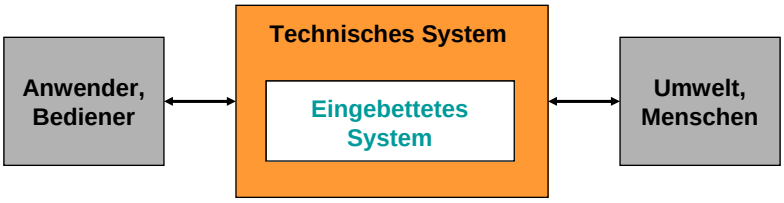
Wintersemester:

- Introduction to Embedded Systems (als Vorkenntnis günstig)
- Dynamische Systeme für Informatiker
- Automotive SW Engineering

Literatur

- Die Vorlesung folgt nicht einem bestimmten Buch.
- Die folgenden Bücher behandeln Themen aus der Vorlesung (aber auch noch viel mehr):
  - D. Peled: Software Reliability Methods. Springer, 2001.
  - E. Clarke, O. Grumberg, D. Peled: Modelchecking. MIT Press, 2001.
  - B. Berard, M. Bidoit, A. Finkel: Systems and Software Verification. Springer, 2001
  - W. Ehrenberger: Software-Verifikation. Hanser, 2002.
- Mehr wird im Laufe der Vorlesung auf die Website gestellt.
- Zum Bestehen der Prüfung ist der Stoff aus Vorlesung und Übung relevant.

Formale Methoden für **eingebettete Systeme**



Produktions-Automatisierung
z.B.
Leitsysteme in
- Fertigungstechnik
- Verfahrenstechnik

Produkt-Automatisierung
z.B.
- Fahrzeugelektronik
- Avionik,
- Medizintechnik

RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN
LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

Formale Methoden für eingebettete Systeme

Formale Methoden =
Auf mathematisch/logisch strengen Darstellungsmitteln und Verfahren beruhende Vorgehensweisen zur Beschreibung, Analyse und Realisierung von Entwürfen.

Beispiele für Formale Methoden:
- Spezifikation
- Verifikation
- Synthese

Keine formalen Darstellungen:
z.B.:
- Programmiersprachen
- Simulationsmodelle

Im Sprachgebrauch der Informatik:
formale "Semantik" **vs. ausführbare "Semantik"**

RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN
LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

Formale vs. ausführbare Semantik

Beispiele für formale Modelle:

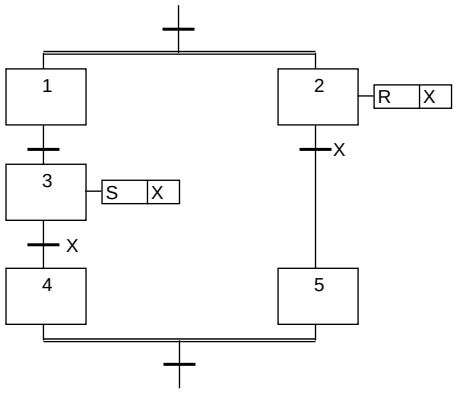
- Zustandsübergangssysteme
- Petrinetze
- Logiken
- Gleichungen, DGLn

Zwei Kennzeichen:

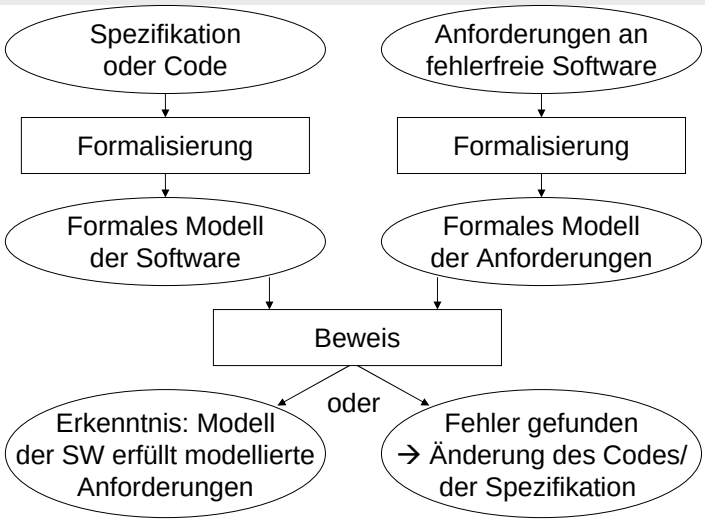
- Formale **Syntax**
- Vollständige, eindeutige und widerspruchsfreie **Semantik**

Nur ausführbar, nicht formal:

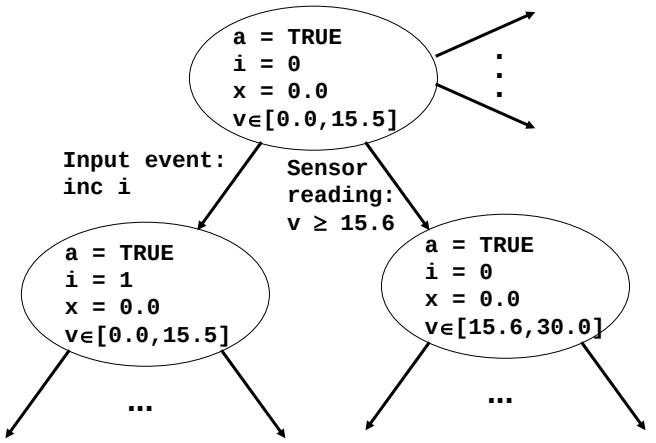
Bsp. SFC



Formale Verifikation (Beispiel Software)



Formales Modell der Software – Beispiel



Semantik?

Formales Modell der Anforderungen

Generische Anforderungen:

z.B.: Software verhält sich deterministisch.

Spezifische Anforderungen:

z.B.:

Nicht formal: **i** liegt immer zwischen 0 und 99, bis zu einem bestimmten Ereignis, ab dem es 100 bleibt. In jedem Fall wird es nicht negativ.

Formal: $(0 \leq \mathbf{i} \leq 99)\ \mathbf{U}\ (\mathbf{i} = 100) \wedge \mathbf{[]}\ \mathbf{i} < 0$

Oder: Zustand z ist nicht erreichbar

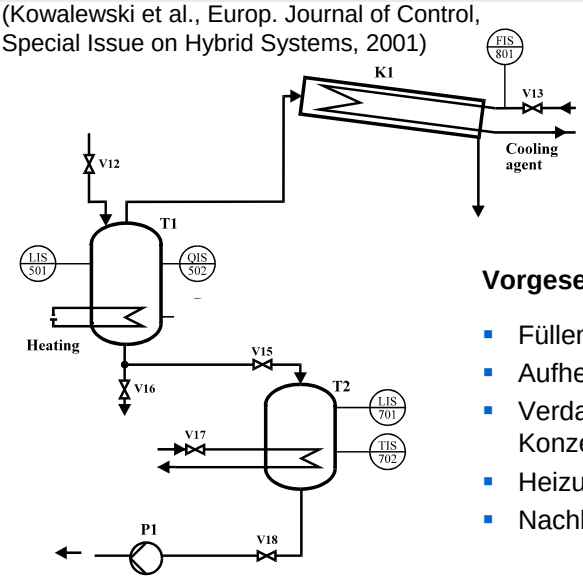
(Bsp.: Die Steuerung versucht nie die Pumpe zu starten, wenn das dahinterliegende Ventil geschlossen ist.)

Zwei Beispiele für Anwendungen

- Verfahrenstechnik
- Automobilelektronik

Beispiel 1: Verfahrenstechnik

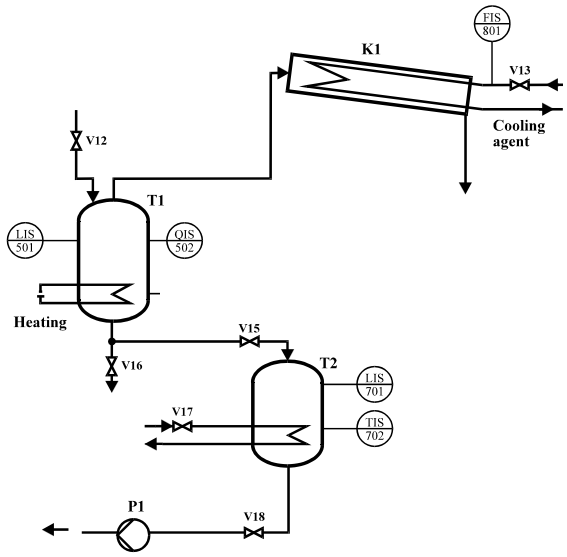
(Kowalewski et al., Europ. Journal of Control, Special Issue on Hybrid Systems, 2001)



Vorgesehener Produktionsablauf:

- Füllen von Behälter T1
- Aufheizen
- Verdampfen bis gewünschte Konzentration erreicht ist
- Heizung ausschalten und T1 leeren
- Nachbearbeitungsschritt in T2

Notabschaltung bei Kühlungsausfall



Randbedingungen:

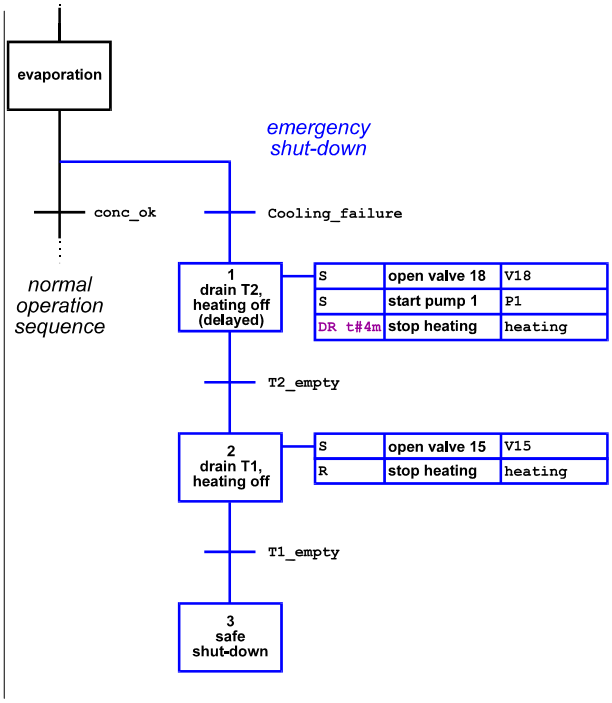
1. Bei fortgesetzter Wärmezufuhr steigen Temperatur und Druck (geschlossenes System).
2. Bei Abkühlen unter einen Grenzwert beginnt das Material im Verdampfer zu kristallisieren.

→ **Gegenläufige Anforderungen:**

- Heizung muss **früh genug** abgeschaltet werden, um gefährlichen Druckanstieg zu vermeiden.
- Heizung muss **lange genug** eingeschaltet bleiben, um Kristallisierung zu vermeiden.

Vorschlag für Steuerungs-programm

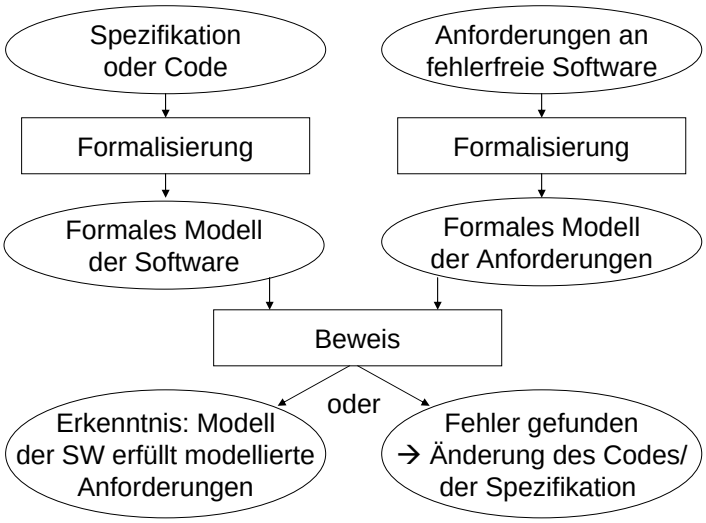
Sequential Function Chart (SFC) nach IEC 1131-3



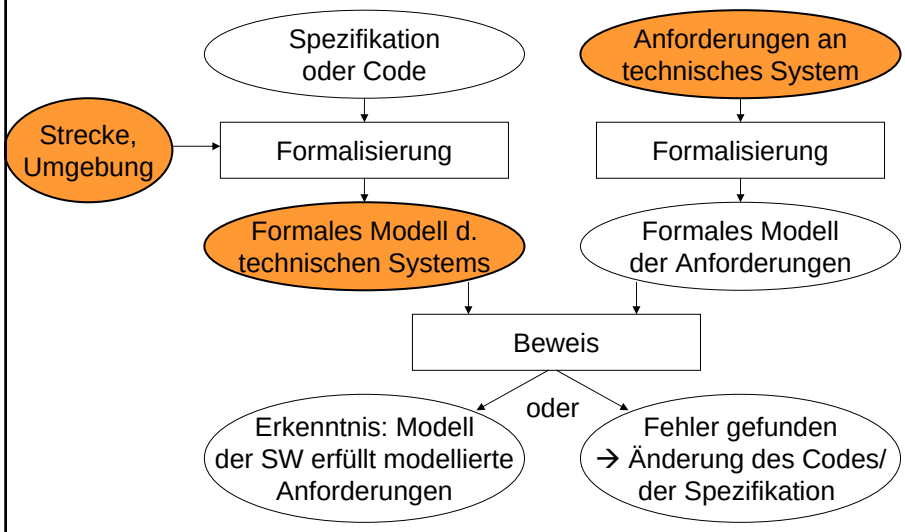
Problemstellung

- Erfüllt das Steuerungsprogramm die folgenden Anforderungen?
  - Der Druck darf oberen Grenzwert nicht überschreiten.
  - Die Temperatur darf Kristallisationspunkt nicht unterschreiten.
- Zu überprüfen ist:
  - logischer Entwurf des SFCs
  - Wahl des Wertes für den Parameter Wartezeit (t#4m)
- Anforderungen sind **für den gesteuerten Prozess** ("geschlossenen Kreis", "technisches System") und nicht für das Steuerungsprogramm formuliert.

Formale Verifikation von Automatisierungssystemen



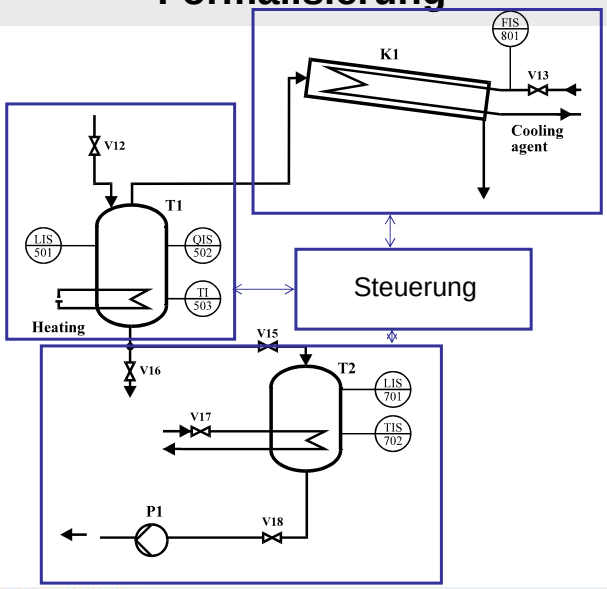
Formale Verifikation von eingebetteten Systemen



RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

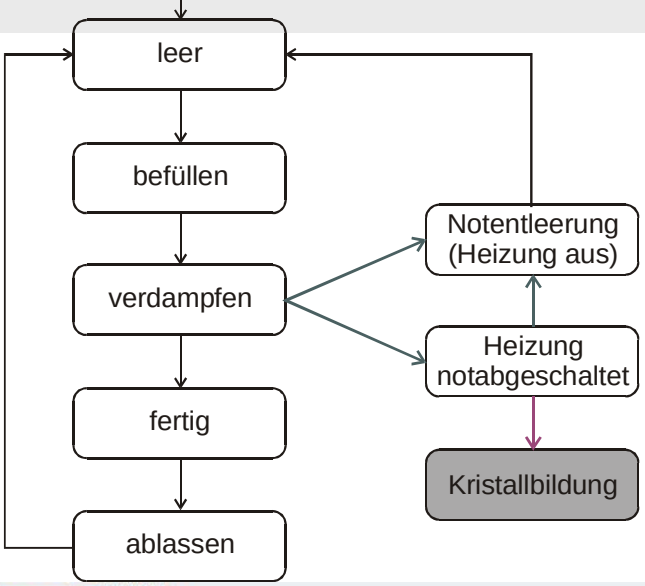
Formalisierung



RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

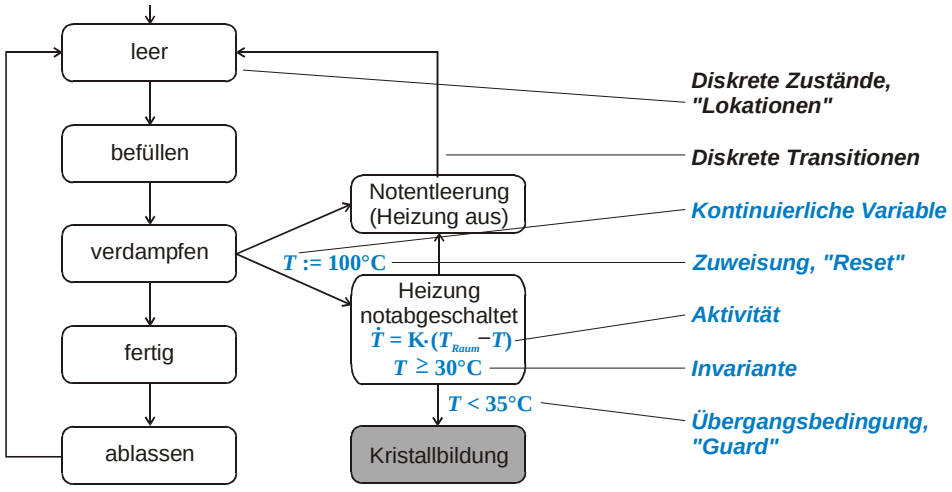
Diskretes Modell für Tank 1



RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

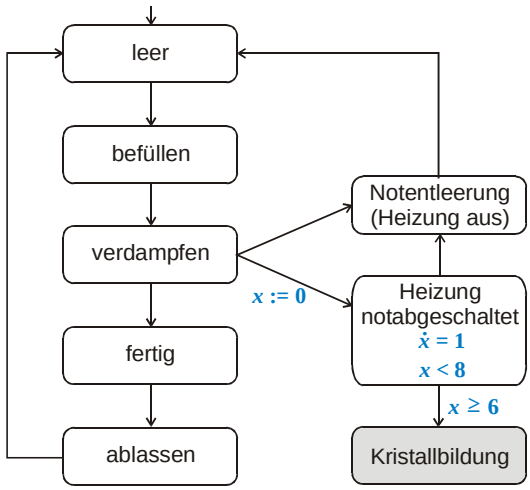
Hybrider Automat (Henzinger, Alur, Sifakis, 1992)



RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

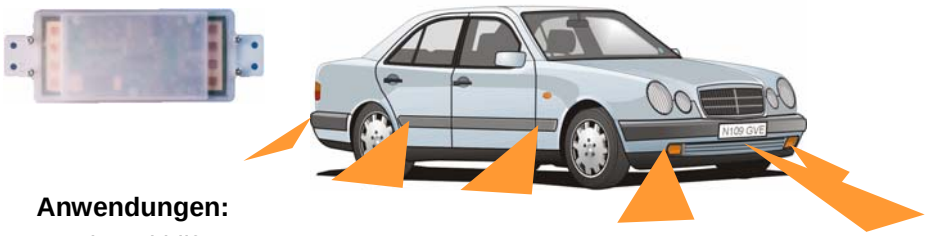
LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

Echtzeitautomat (Alur, Dill, 1990)



Beispiel 2: Fahrzeug-Umfeld-Sensorik:

Nahbereichsradar-Technologie (24 GHz)



Anwendungen:

- Einparkhilfe
- **Automatisches Einparken**
- ACC Stop & Go, Staufolgefahren
- Pre-Crash-Detection
- **Kollisionsvermeidung**
- Toter-Winkel-Überwachung
- **Abbiege- und Spurwechselassistent**

Problemstellung

Bisher:

- Anwendungen unabhängig voneinander entwickelt und realisiert.
- Jede Anwendung: eigene Sensoren, Steuergerät, SW

Jetzt:

- Bisheriger Ansatz wegen Kosten- und Bauraum nicht mehr möglich.
- Einführung neuer Anwendungen erfordert gemeinsame Nutzung von Sensoren und Integration von Diensten auf wenigen Steuergeräten.
- Scheduling für Ressourcenzuteilung notwendig

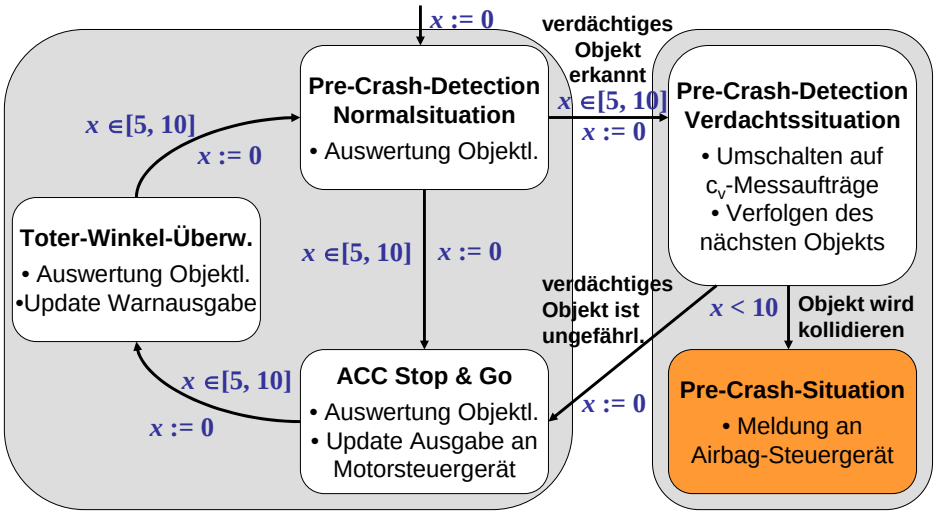
Problem:

- **Garantiert das Scheduling, dass in jeder Situation die Echtzeitanforderungen der einzelnen Anwendungen eingehalten werden?**

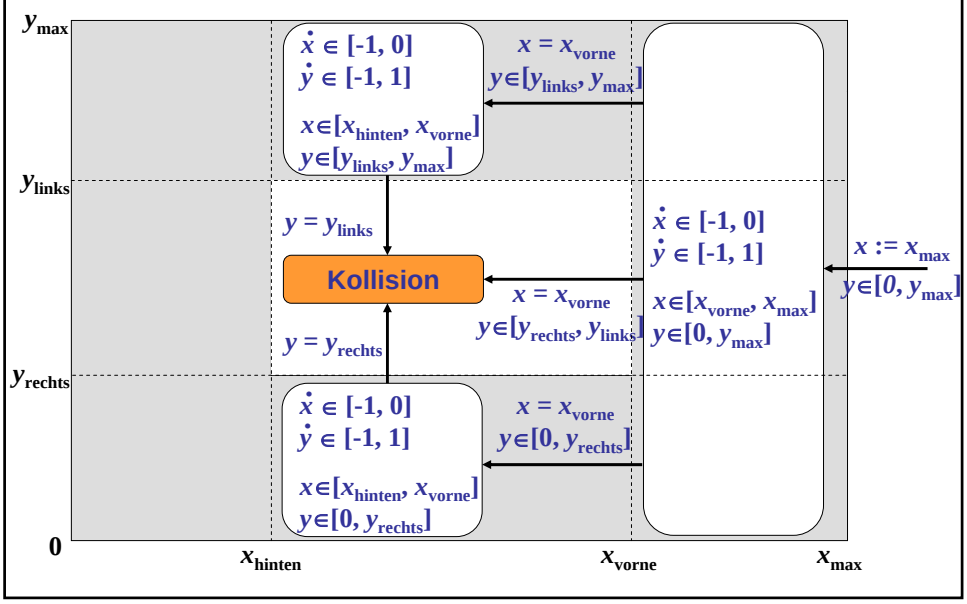
RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

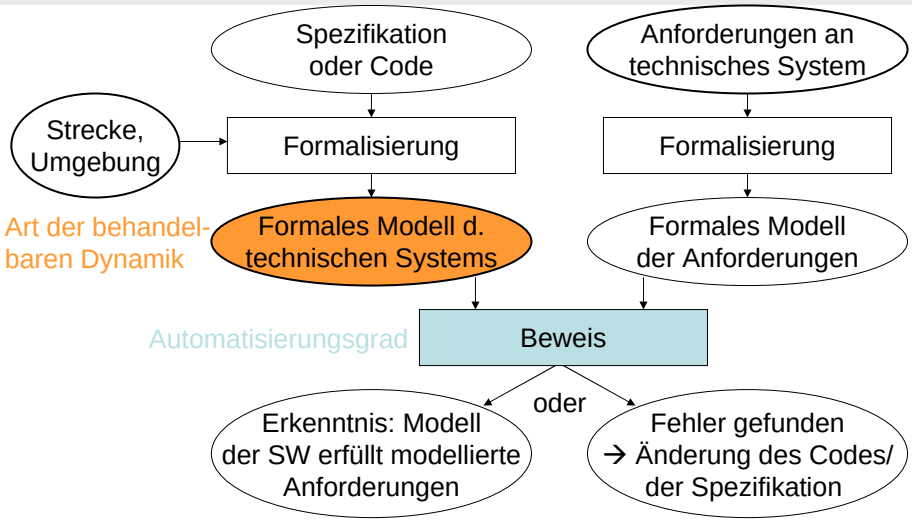
Echtzeitautomat für Ressourcen-Zuteilung



Modell das Auto-Umfeldes als hybrider Automat
Beispiel: ein Objekt von vorne



Stand der Technik



RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

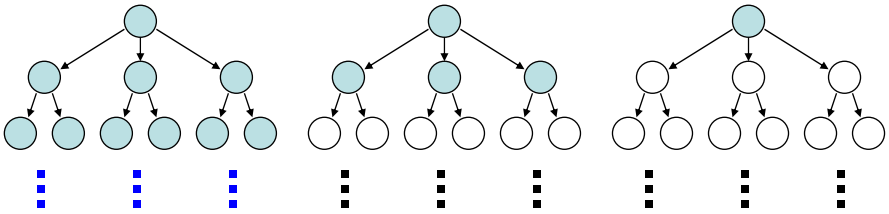
Automatisierungsgrad

Deduktive Analyse (Theorem Proving)

Algorithmische Analyse

- Vollständige Suche im Zustandsraum (Model Checking)
- Begrenzte Suche im Zustandsraum (Bounded Model Checking)
- Statische Analyse (Abstrakte Interpretation)

Model Checking: Bounded Model Checking: Statische Analyse:



Prinzip der abstrakten Interpretation

Beispiel:

```
...
FOR i=1..100 DO
  ...
  a=b/(i-90)
  ...
END_FOR
...
```

Kein Durchlaufen der Schleife, sondern Rechnen mit $i \in [1,100]$.

Art der behandelbaren Dynamik

Diskret:

- Kommerzielle Werkzeuge
- Behandlung von Zeitanforderungen durch "Zählen von Zustandsübergängen"

Diskret + Zeit:

- Model Checking: belastbare akademische Werkzeuge (z.B. UPPAAL)
- Konservative Abschätzung von Ausführungszeiten: kommerzielle Werkzeuge verfügbar (AbsInt, Saarbrücken)

Hybrid:

- Forschungsthema, akademische Werkzeuge nicht belastbar
- Lösungsansatz: Abstraktion durch einfachere Dynamik, gegebenenfalls Konkretisierung, wo nötig

Kommerzielle Werkzeuge und industrielle Akzeptanz

Model Checker:

- *SCADE* – Esterel Technologies
- *SILDEX* – TNI Valiosys
- *OSC-Verifier* – OSC Embedded Systems, Oldenburg
- *Autofocus* – Validas (TU München)

Abstrakte Interpretation:

- *Polyspace C-Verifier* – Polyspace
- *aiT* – AbsInt, Saarbrücken

Kommerzielle Werkzeuge und industrielle Akzeptanz

Model Checker:

▪ *SCADE* – Esterel Technologies	Avionik
▪ *SILDEX* – TNI Valiosys	Avionik
▪ ***OSC-Verifier*** – OSC Embedded Systems, Oldenburg	Automobil: v.a. **Testfallgenerierung**
▪ *Autofocus* – Validas (TU München)	

Abstrakte Interpretation:

▪ ***Polyspace*** *C-Verifier* – Polyspace	Avionik, Automobil: **Code-Analyse**
▪ *aiT* – AbsInt, Saarbrücken	Avionik, Automobil: WCET

RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME

Ausblick

Inhalt der Vorlesung:

- Formale Beschreibungsmittel (vor allem für Verhalten von Systemen)
  - deklarativ/operationell
  - diskret/mit Zeit/hybrid
- Formale Verifikation
- Formale Testunterstützung
- Formale Synthese

RWTH RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

LEHRSTUHL INFORMATIK XI SOFTWARE FÜR EINGEBETTETE SYSTEME