Modelica und Metamodellbildung: Ein Vergleich und Kombination

Pieter J. Mosterman
Institut für Robotik und Mechatronik
DLR Oberpfaffenhofen

Manuel A. Pereira Remelhe
Lehrstuhl für Anlagensteuerungstechnik
Universität Dortmund

DFG KONDISK Projekt: Objektorientierte Modellierungs- und Simulationsumgebung für kontinuierlich/diskrete Systeme

Antragsteller: Prof. Sebastian Engell und Dr. Martin Otter

Modelle im Systementwurf

Mächtiges Mittel zur

- **Analyse**
- **Synthese**
- **Einsicht**
- **Kommunikation**
- **Formalisierung intuitiver Ideen**

Erlaubt die einheitliche Behandlung und Integration verschiedener Aktivitäten und Aspekte

- **z.B. des Softwareteils und des physikalischen Teils**

Domänenspezifische Formalismen

Komplexe Probleme erfordern die Verwendung mehrerer Modellierungssprachen die auf jeweilige Problem- und Anwenderkreise zugeschnitten sind

Je spezifischer die Formalismen, desto

- **schneller können Modelle erstellt werden**
- **weniger Modellierfehler treten auf**
- **besser analysierbar sind die Modelle**

Gesamtaufgabe umfaßt unterschiedliche Teilprobleme

- **die Integration von und Beziehung zwischen verschiedener Formalismen**
  - **Kombination, Verfeinerung, mehrere Sichten**

Erfordert den Umgang mit einer Vielzahl von Formalismen

Metamodellierung

Modellierung der Modellierungsformalismen

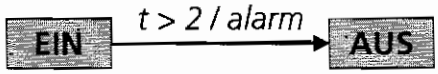
Bestandteile der Definition eines Formalismus:

- **Syntax**
  - **konkret** 
  - **abstrakt** *Resistor*
- **Semantik**
  - **statisch** `positive and negative pin must be connected`
  - **dynamisch** $v = R * i$

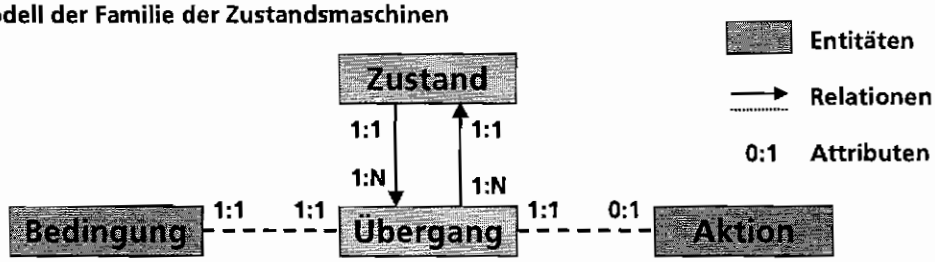
DLR

Zustandsmaschine

Eine Instanz



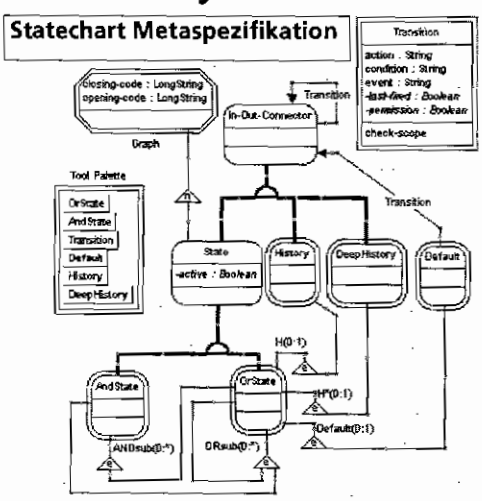
Modell der Familie der Zustandsmaschinen



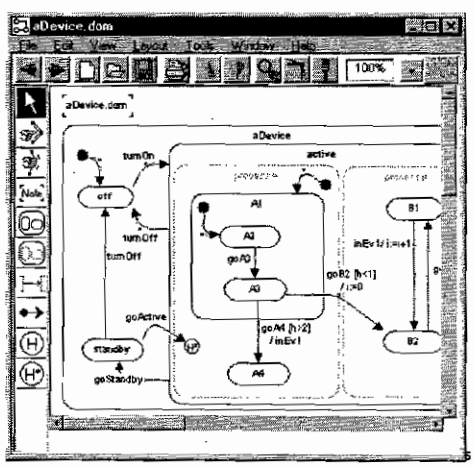
DLR

DoME für Syntax und statische Semantik (Honeywell)

Statechart Metaspezifikation



Automatisch generierter Editor



Dynamische Semantik

Ausführung der Modelle muß extra kodiert werden

- z.B. Markenfluß im Modellgraphen

Dynamik von physikalischen Modellen

- differentielle und algebraische Gleichungen
- nicht kausal
  - Markenfluß erfordert aber explizite Differentialgleichungen

Globaler Modellinterpreter (Gleichungslöser) wird benötigt, z.B. *Dymola*

- benutzt *Modelica* als Speicherformat

Modelica

Objektorientierte Modellierungssprache für physikalische Systeme

- basiert auf nicht kausalen Gleichungen
- hinreichend für viele physikalische Domänen

Basisspeicherformat für physikalische Systeme

- Modellaustausch
- lesbar
- grafisch und textuell

Umfasst Syntax und Semantik

Ausführung

Kontinuierlicher Teil

- **nicht kausale Gleichungen**
  - **algebraische**
  - **differentielle (*der* Operator)**
- **globaler Interpreter zum Sortieren und Lösen**

Diskreter Teil

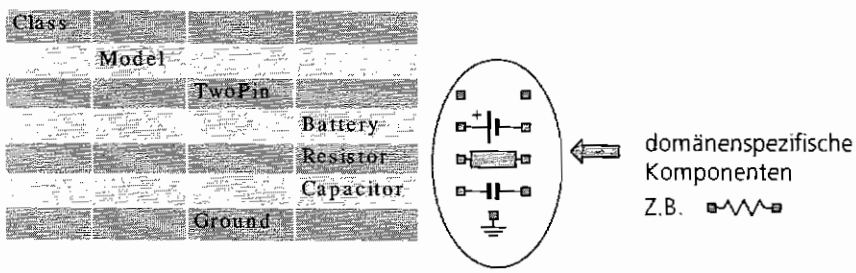
- **Zustandsverhalten (*pre* Operator)**
  - *ON = pre(ON) or (x > 2)*
- **globale Zustandsmaschine**
  - **Kreuzprodukt lokaler Zustandsmaschinen**

Modelica als Metasprache auf der Komponentebene

Modellklassen bilden Grundeinheiten der Domäne

- **Bibliotheken**

z.B., elektrische Schaltkreise



Modelica als Metasprache

Feste Grammatik

- port basiert
- ungerichtete Verknüpfungen mit fester (`=´) Semantik

Keine explizite Domänenbeschränkungen möglich

Ausführungssemantik auf Objektebene

- das Verhalten einer Zustandsmaschine muss z.B. lokal für Zustände und Übergänge beschrieben werden

Kombination gleichungsbasierter und meta Prinzipien

DoME

- domänenspezifische Formalismen
  - Syntax
  - statische Semantik

Modelica

- dynamische Semantik
  - differentielle und algebraische Gleichungen (CT)
  - Zustandsmaschinen (FSM)

Abbildung von DoME auf Modelica Objekte

DLR

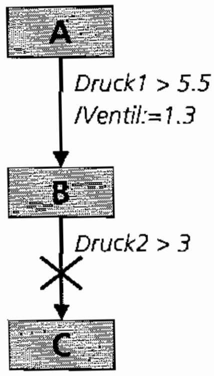
Semantische Einschränkungen von Modelica

Globale Iterationsschleife

- **immer synchrones Verhalten aller Modellteile**
- **Statecharts schwierig in Modelica auszudrücken**
  - **Ausiterieren nach externer Anregung**

Weitere Probleme

- **Formulierung als simultane Gleichungen**
- **Zustandsmaschine ist *ein* semantisches Objekt**
  - **Modelica erfordert aber lokale Zustands- und Übergangs-Objekte**



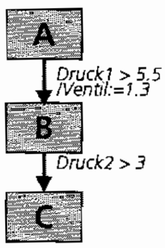
DLR

Neues Rechenmodell erforderlich

Verwendung von *'algorithms'*

- **werden behandelt als Gleichungen mit festen Ein- und Ausgangsvariablen**
- **enthalten Sequenzen von Zuweisungen**

Ein ganzes Statechart wird in ein Algorithmus übersetzt




```
class SimpleAutomaton
  PLCPort port;
  Boolean T1, T2, A(start=true), B, C;
  Boolean pA, pB, pC;
algorithm
  pA:=pre(A); pB:=pre(B); pC:=pre(C);
  T1 := pA and (port.druck1 > 5.5);
  T2 := pB and (port.druck2 > 3.0);
  while T1 or T2 loop
    A := pA and not(T1);
    B := pB and not(T2) or T1;
    C := pC or T2;
    if T1 then port.ventil := 1.3; end if;
    pA:=A; pB:=B; pC:=C;
    T1 := pA and (port.druck1 > 5.5);
    T2 := pB and (port.druck2 > 3.0);
  end while;
end SimpleAutomaton;
```

Zusammenfassung

Bedarf für

- standardisierte Sprache für Modellaustausch
- (nicht kausal) gleichungsbasiert Modellieren, z.B. Modelica
  - feste Grammatik, keine Domänenbeschränkungen, ...
- Metamodellieren, z.B. DoME
  - nur einfache Dynamik möglich, keine Gleichungslöser

Kombination

- DoME für domänenspezifische Formalismen
- Modelica für Modellaustausch und Ausführungsmodelle der Modellobjekte
- zugeschnittene Rechenmodelle mit Hilfe von Algorithmen

Modelica und Metamodellbildung: Ein Vergleich und Kombination

Pieter J. Mosterman, *DLR Oberpfaffenhofen*, http://www.op.dlr.de/~pjm
Manuel A. Pereira Remelhe, *Universität Dortmund*

KONDISK-Projekt: Objektorientierte Modellierungs- und Simulationsumgebung für kontinuierlich/diskrete Systeme, http://www-er.df.op.dlr.de/kondisk/index.html
Antragsteller: Prof. Sebastian Engell, *Universität Dortmund*, und Dr. Martin Otter, *DLR Oberpfaffenhofen*.

Die Bedeutung des modellbasierten Systementwurfs hat in den letzten Jahren aufgrund der Möglichkeiten zur frühen Analyse und zur automatischen Synthese von Software-Code bzw. Hardwarebeschreibungen enorm zugenommen. Eine wichtige Voraussetzung für eine weitergehende Verbreitung ist eine zunehmende Werkzeugunterstützung von Modellierungssprachen, die auf bestimmte Problem- und Anwenderkreise zugeschnitten sind. Dazu gehört auch die Entwicklung neuer Modellierungssprachen sowie die Unterstützung der Übersetzung zwischen den verschiedenen Sprachen. Dies betrifft sowohl die komponentenorientierte Modellierung physikalischer Systeme mit vordefinierten domänenspezifischen Bauteilen (z.B. verschiedene Hydraulikpumpen, Leitungen, Zylinder, etc.), als auch die Modellierung logischer Systeme, wie z.B. Steuerungen, Kommunikationssysteme, etc., mit Hilfe verschiedenster ereignis-diskreter Formalismen. Insbesondere sollte die Verknüpfung (Kombination, heterogene Verfeinerung, Darstellung mehrerer Sichten und Bildung von Abstraktionsschichten) dieser unterschiedlichen Modellformen unterstützt werden. Um dieses hochgesteckte Ziel ohne übermäßigem Aufwand realisieren zu können, werden in Zukunft Meta-Modellierungs-Techniken eine immer größere Rolle spielen.

Zu beachten ist weiterhin, dass es bereits viele modellbasierte Entwurfswerkzeuge gibt, die für bestimmte Aufgaben bestens geeignet sind und sich daher in den entsprechenden Anwendungsfeldern durchgesetzt haben. Um eine kohärente Entwurfsphase zu gewährleisten, müssen die verwendeten Werkzeuge miteinander verkoppelt werden, was auf drei Ebenen möglich ist: (i) auf der numerische Ebene, z.B. Co-Simulation, (ii) auf der Daten-Ebene, wie z.B. bei Ptolemy, und (iii) auf der Modellebene. Für die beiden letzten Fälle ist es sinnvoll, eine hinreichend mächtige Basisbeschreibungssprache zu etablieren, auf die die Daten bzw. die Modelle transformiert werden können.

Diese Zielsetzung erfordert Methoden und Werkzeuge für den schnellen Entwurf neuer Formalismen sowie für die Unterstützung des Modellaustauschs. Im Rahmen von Modelica wird versucht eine hinreichend mächtige Sprache zu definieren, die als Basisspeicherformat für physikalische Systeme verwendet werden kann und zugleich gut lesbar ist. In dem Beitrag wird gezeigt, inwiefern Modelica als Meta-Sprache angesehen werden kann und worin die Beschränkungen bestehen im Vergleich zu expliziten Meta-Modellierungsansätzen. Es wird schließlich skizziert, wie mittels Kopplung auf der Modellebene auch komplexe diskrete Formalismen unter Erhalt ihres spezifischen Rechenmodells in das Modelica-Rechenmodell integriert werden können. Dazu wurde mit Hilfe des Meta-Modellierungs-Werkzeugs DoME (entwickelt bei Honeywell) ein Statechart-Editor sowie eine Modell-Export-Funktion nach Modelica implementiert.