

# Überblick und Aktuelles zu Dymola



# Inhalt

- Dymola
- Modelica
- Modellieren mit Dymola
- Modellbibliotheken
- Simulieren mit Dymola
- Dymola-Simulink-Interface
- Modelica 2.1
- Dymola 5.4

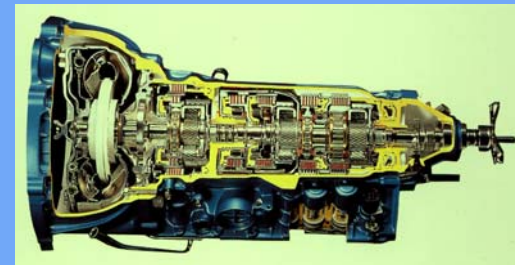
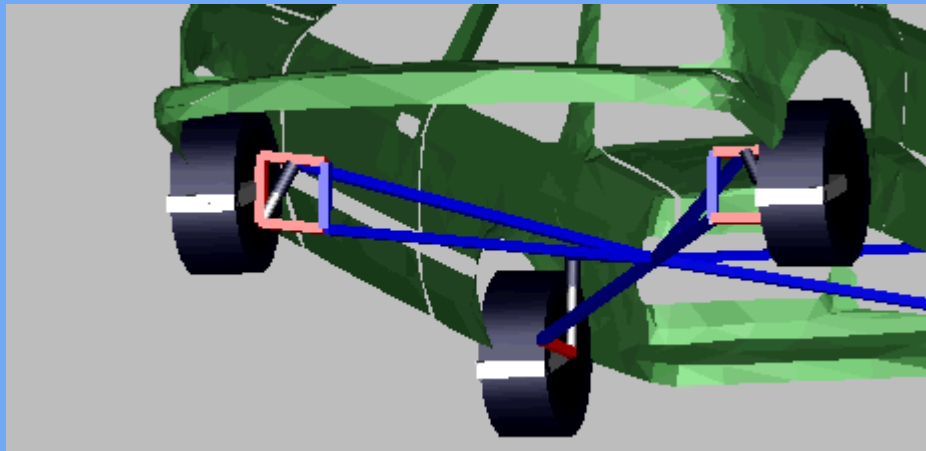
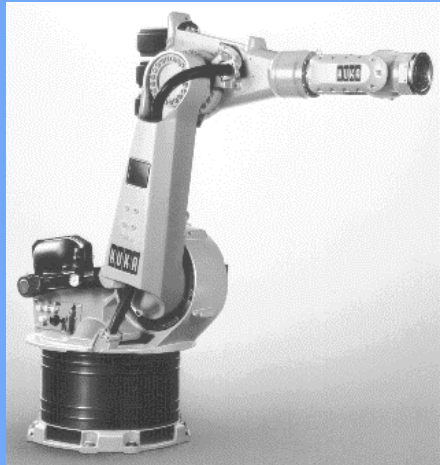
# Dymola

- Dymola ist eine Software zur Modellierung und Simulation
- entwickelt von Dynasim AB in Schweden
- Ziel: einheitliche Modellierung verschiedener Ingenieurdisziplinen
- einige Stichworte
  - “Multi-domain modelling”
  - Nichtkausale Modellierung
  - Graphische Modellierungsoberfläche
  - Schnittstelle zu anderen Programmen

# Modelica

- neue Methode
  - objektorientiertes Modellieren
  - gleichungsorientiertes Modellieren
- austauschbare, standardisierte formale Sprache
  - zum Modellaustausch
  - Weiterverwendung von Modellierungswissen
- Modelica --- Sprachdefinition  
[www.modelica.org](http://www.modelica.org)
- Dymola --- Software zur Modellierung und Simulation mit Modelica  
[www.dynasim.se](http://www.dynasim.se)

# Anwendungsbeispiele



# Modelica Design Group

- Zusammenschluß von Simulationsexperten
- ehrenamtlich
- mehrere Treffen jährlich
- Erstellung und Veröffentlichung der Sprachdefinition
- Pflege der Homepage

# Ziele der Modelica Design Group

- offene Sprachdefinition
- Modelle unterschiedlicher Herkunft
- objektorientierte Sprachen vereinheitlichen
- Wiederverwendbarkeit der Modelle
- deklarative statt prozedurale Sprache
- Sprache soll effiziente Simulation unterstützen

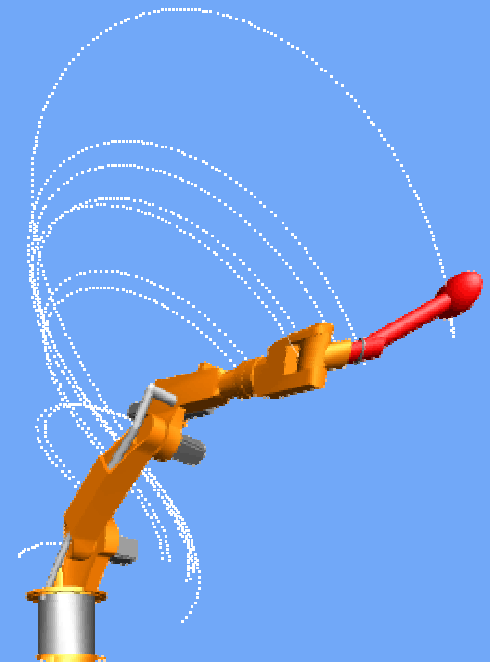
# Modelica - Entwicklungsstand

- Entwurf startete im September 1996
- Version 1.0 im September 1997
- Heute: Version 2.1
- Werkzeuge und Bibliotheken stehen zur Verfügung
- Homepage: [www.modelica.org](http://www.modelica.org)
- regelmäßige Tagungen
- nächste Tagung: März 2005 an der TU HH



# Die Firma Dynasim

- Entwickler von Dymola
- gegründet 1992 von Hilding Elmqvist
- Jetzt: 7 Angestellte, davon 5 technisch/wissenschaftlich
- Zusammenarbeit mit Wissenschaftlern
- Kunden weltweit, z.B.: Toyota, Volvo, Rover, Ford, GM
- Deutschland z.B.: BMW, DaimlerChrysler, ZFF, Volkswagen, viele Universitäten und Forschungsinstitute
- Österreich, z.B.: AVL, Elin, MAGNA STEYR, TU Wien

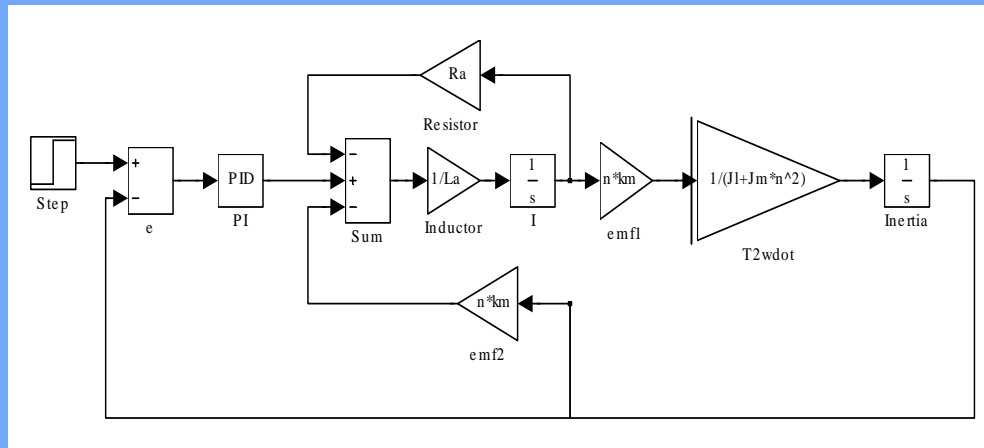
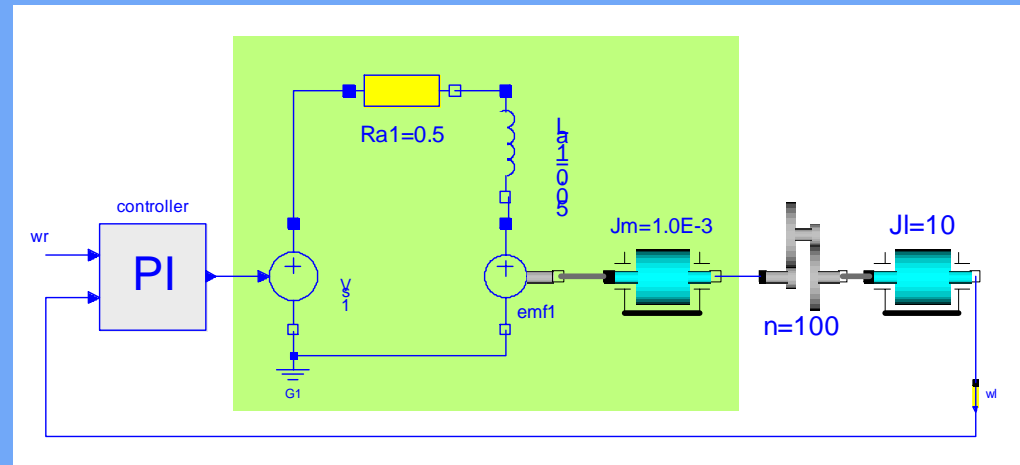


# Entwicklungsziele

- Software zur Modellierung komplexer Systeme
- Modelle verschiedener Ingenieurdisziplinen
- Verfügbarkeit der Modelle als Quellcode
- homogenes Simulationsmodell
- effiziente Simulation
- Verkürzung der Entwicklungszeit

# Energieflußorientierte Modellierung

Dymola

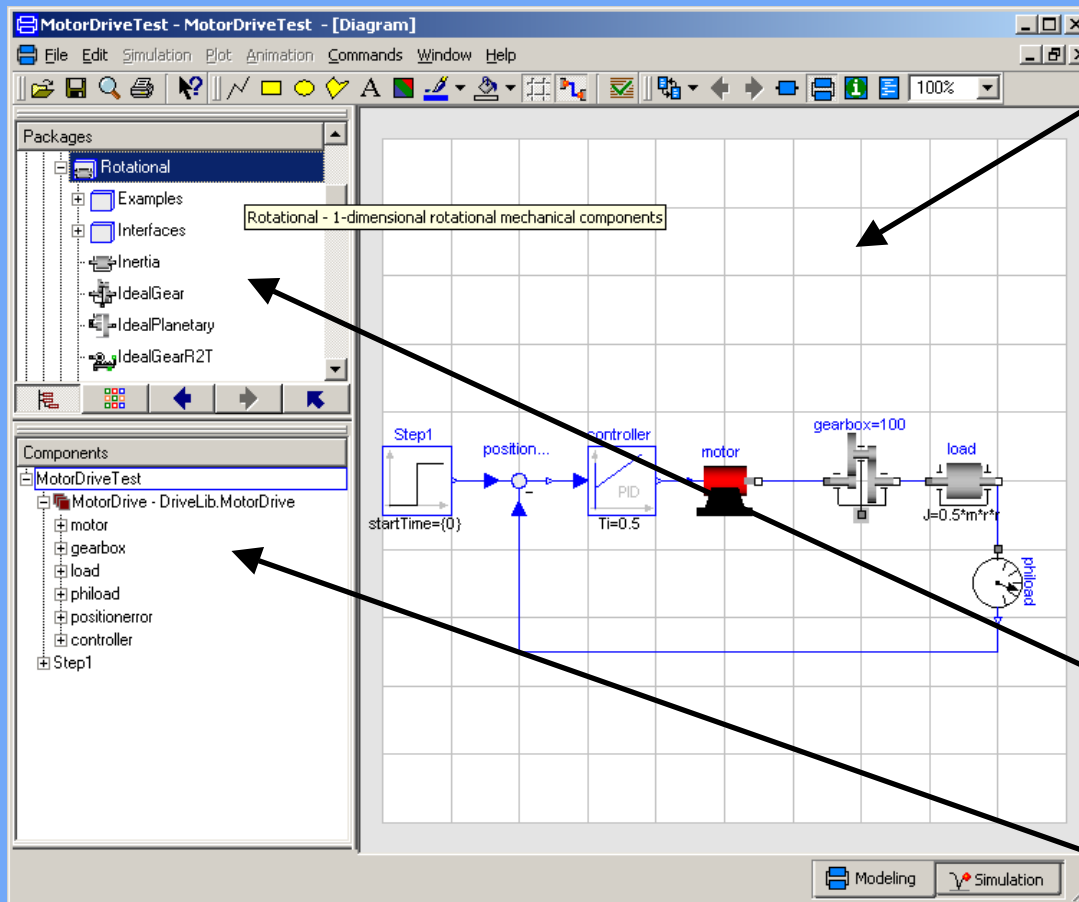


signalflußorientierte  
Modellierung

# Modellierung mit Dymola

- versteht Modelica
- Gleichungen als Modellbasis
- Variablen, Typen, Einheiten (units)
- objektorientierte Struktur der Modelle  
Klassen, Konnektoren, Blöcke
- Modellbibliotheken, sog. Packages
- Felder und Matrizen, einschl. Operatoren
- Funktionen und Algorithmen
- symbolische Manipulation der Gleichungen

# Dymola-Modellierungsfenster



Modellierungsfenster

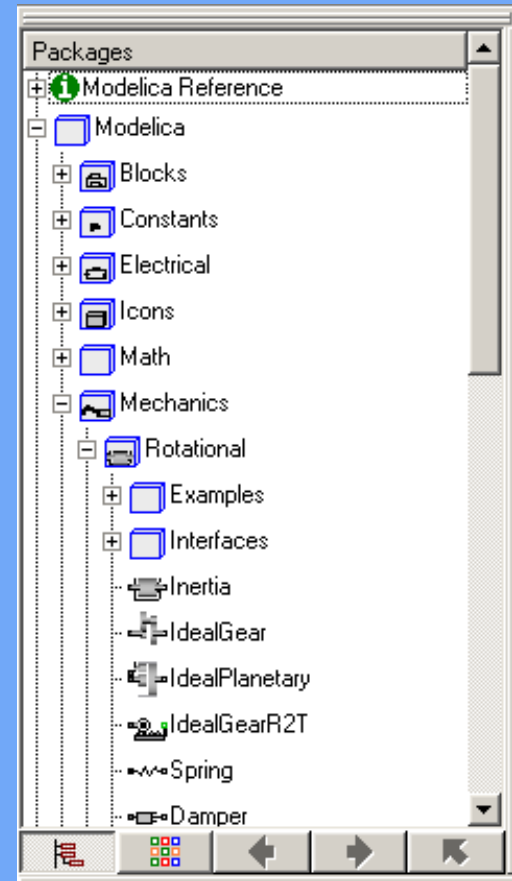
- Icon Layer: Definition der Graphik
- Diagram Layer: Zusammenstellen des Modells
- Documentation Layer  
HTML-Dokumentation einfügen
- Equation Layer:  
Gleichungen als Modelica-Kode

Model Library Browser

Model Browser

# Modelica-Bibliotheken

- Hierarchisch
- Komponenten lassen sich sinnvoll anordnen
- grosse Komponentenvielfalt
- freie Bibliotheken enthalten z.B.:
  - einfache Regelungstechnik (Blöcke)
  - 1D-Mechanik (rotatorisch, translatorisch)
  - analoge und digitale Elektrik
  - Mehrkörpersysteme
  - einfache Pneumatik und Hydraulik
  - Fahrdynamik (VehicleDynamics)



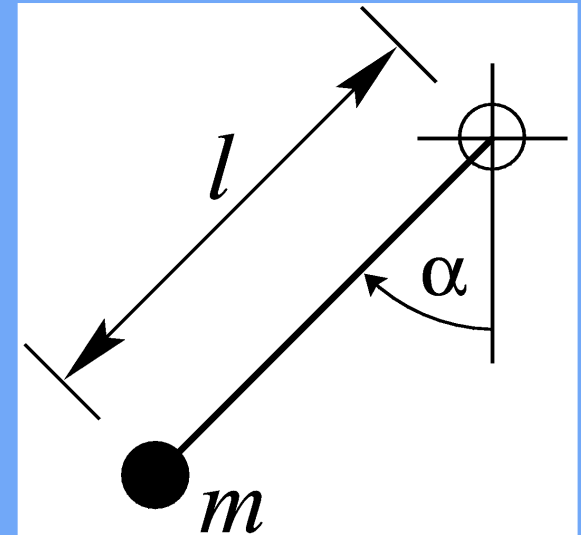
# Modelica-Bibliotheken

- kommerzielle Bibliotheken für
  - Antriebsstränge (Powertrain)
  - Pneumatik
  - Hydraulik
  - Klimaanlage (AirCondition)
- weitere freie Bibliotheken

[www.modelica.org](http://www.modelica.org)

# Grundlage der Modellbildung in Dymola

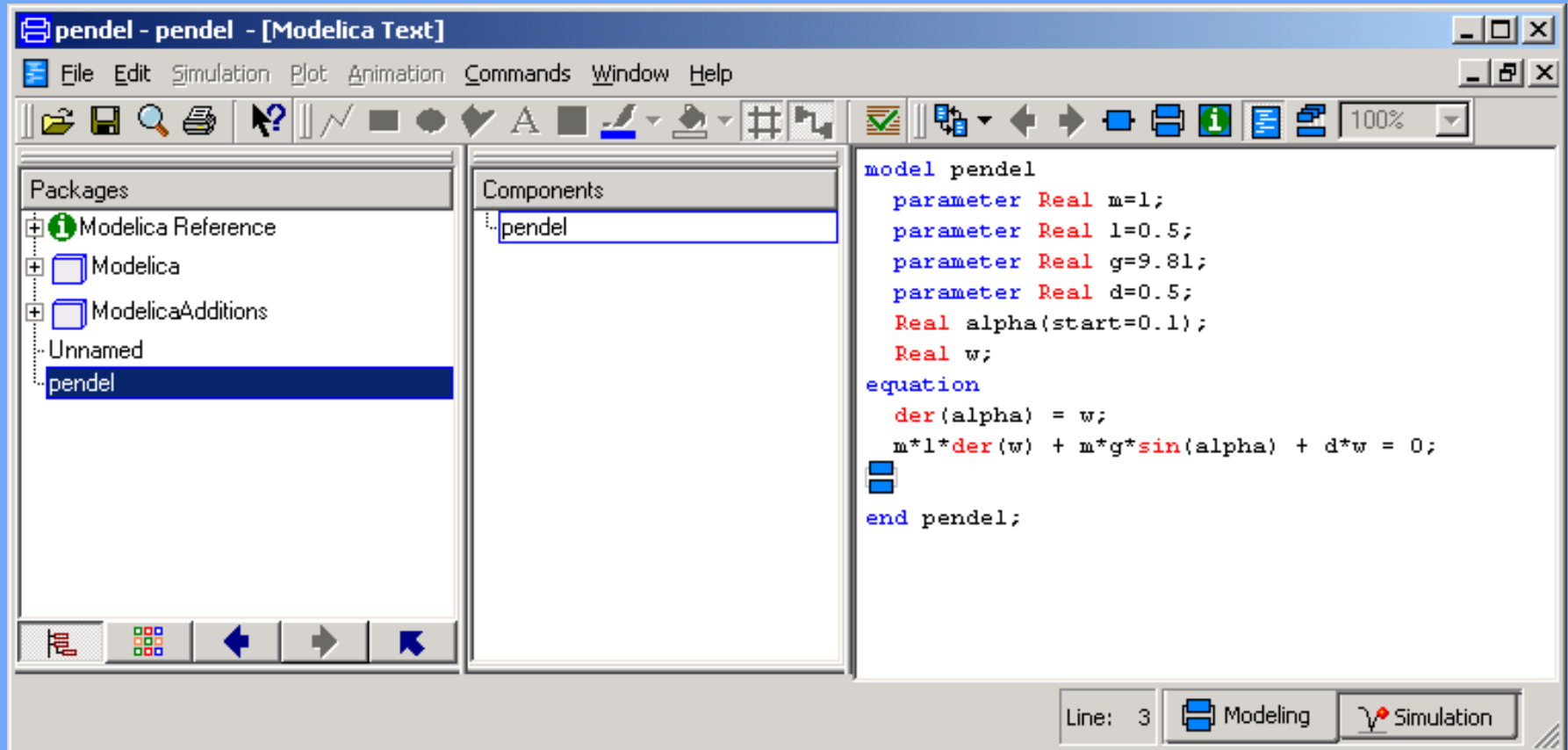
- Modellierungsgrundlage: Gleichungssysteme
- Einfacher Fall: Pendel



$$ml\ddot{\alpha} + mg \sin(\alpha) + dl\dot{\alpha} = 0$$



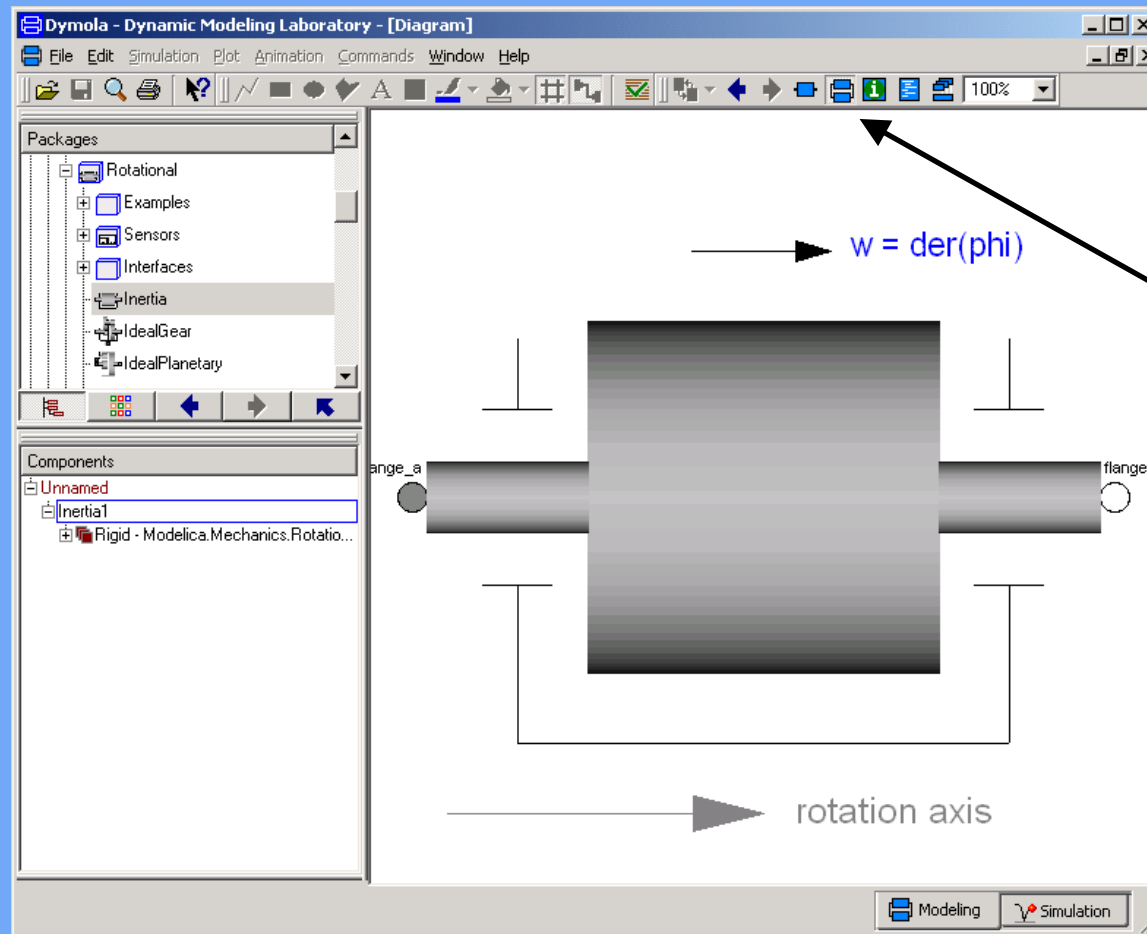
# Dymola-Modell



# Modellierung

- jedes Bauteil als abgeschlossene Komponente
- physikalische Konnektoren
- sinnvolle Graphik
- HTML-Dokumentation
- Beispiel: Rotational Inertia aus der Modelica Library

# Modelica - Rotational - Inertia



Diagram

# Analogien für Konnektoren

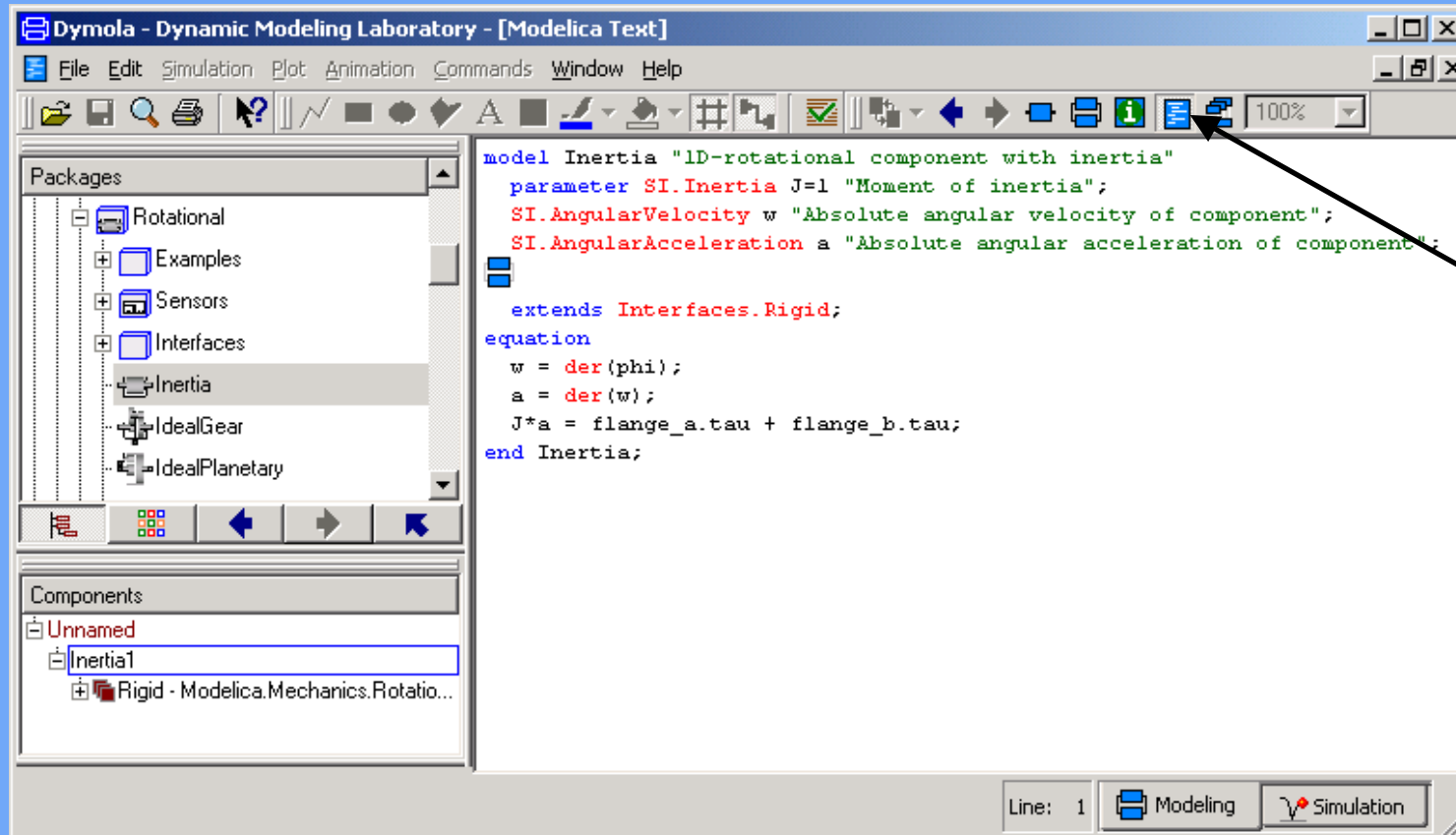
<u>Fachgebiet</u>	<u>Potential</u>	<u>Fluss</u>
Elektrotechnik	Spannung	Strom
Mechanik, transl.	Position	Kraft
Mechanik, rotat.	Winkel	Drehmoment
Hydraulik	Druck	Volumenstrom
Thermodynamik	Temperatur	Wärmefluß

# Konnektoren

```
connector Modelica.Mechanics.Rotational.Interfaces.Flange_a  
  "1D rotational flange (filled square icon)"  
  SI.Angle phi "Absolute rotation angle of flange";  
  flow SI.Torque tau "Cut torque in the flange";  
end Flange_a;
```

```
connector Flange_b "1D rotational flange (non-filled square icon)"  
  SI.Angle phi "Absolute rotation angle of flange";  
  flow SI.Torque tau "Cut torque in the flange";  
end Flange_b;
```

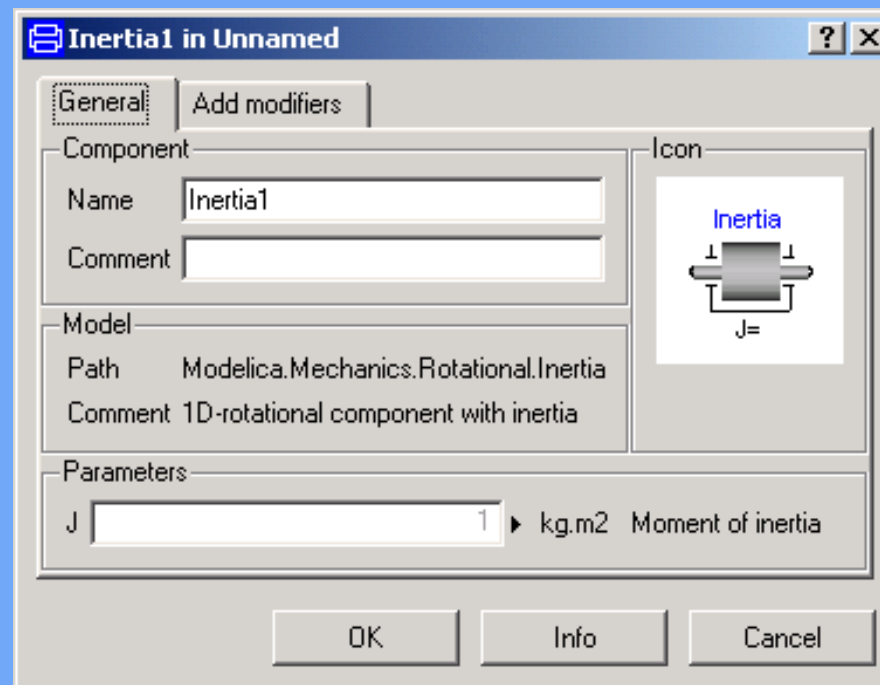
# Modelica - Rotational - Inertia



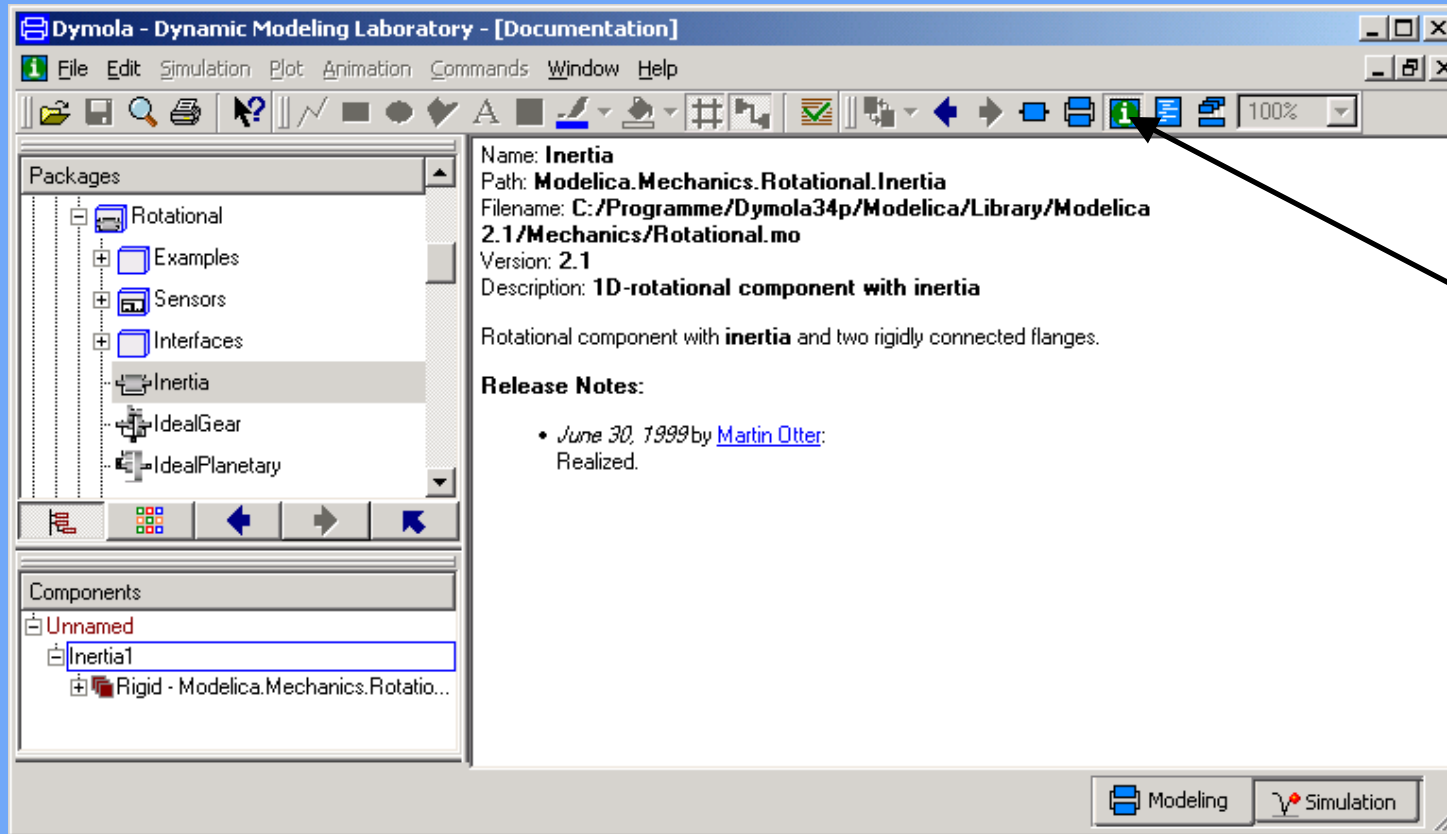
Text

# Modelica - Rotational - Inertia

Parameterfenster wird automatisch erstellt



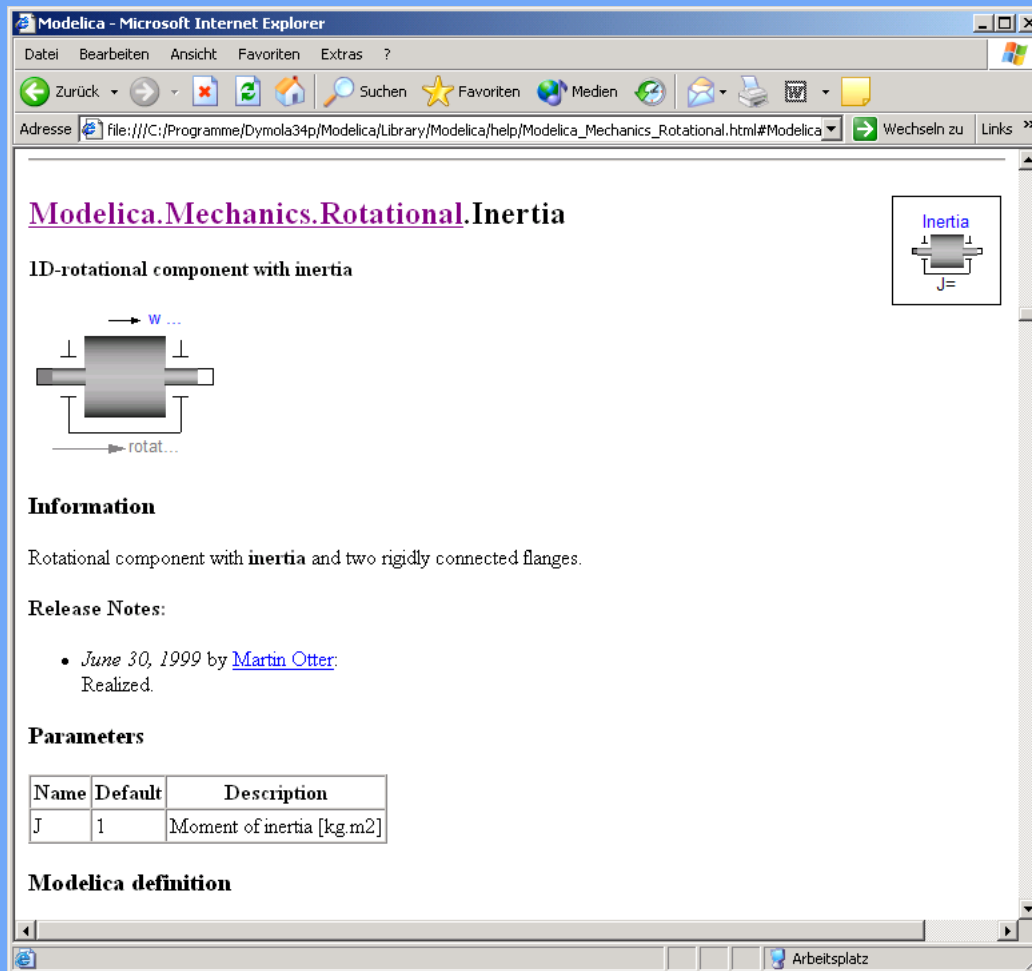
# Modelica – Rotational - Inertia



Beschreibung  
als HTML



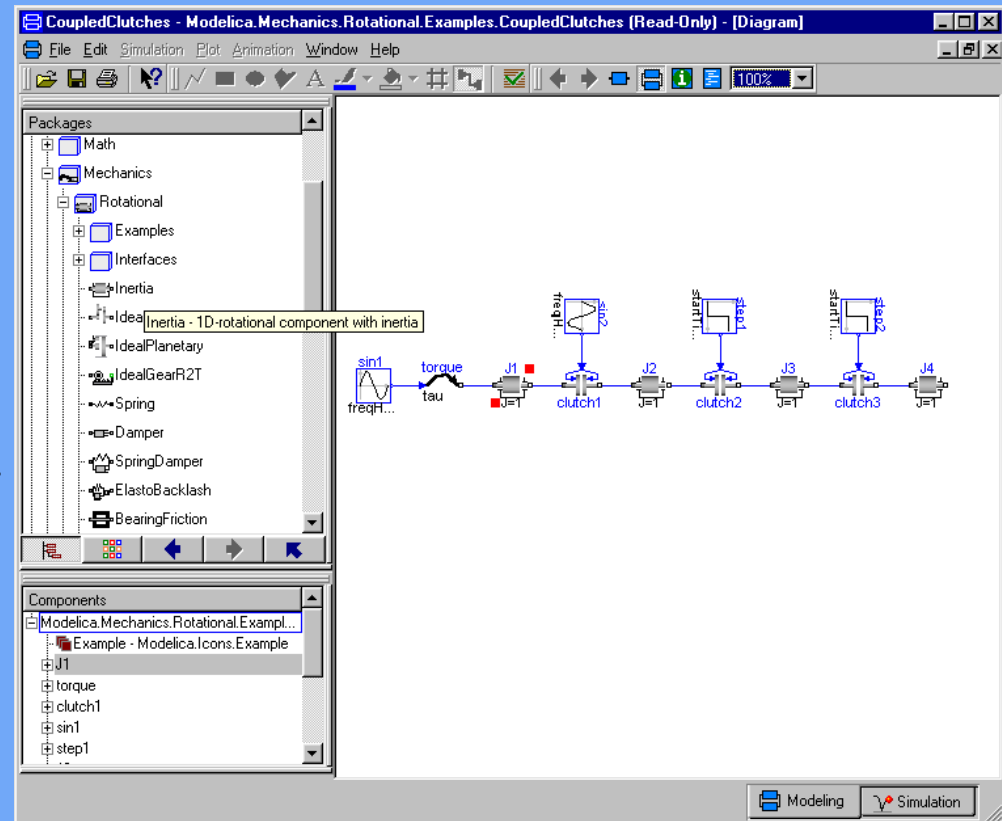
# Modelica – Rotational - Inertia



automatisch  
erstellte  
Dokumentation

# Modellierung aus Komponenten

- Komponenten in Bibliotheken
- Modell aus Grundkomponenten zusammengesetzt
- Diagramm erscheint im Modellierungsfenster
- Komponenten parametrieren
- Hierarchie für grosse Modelle



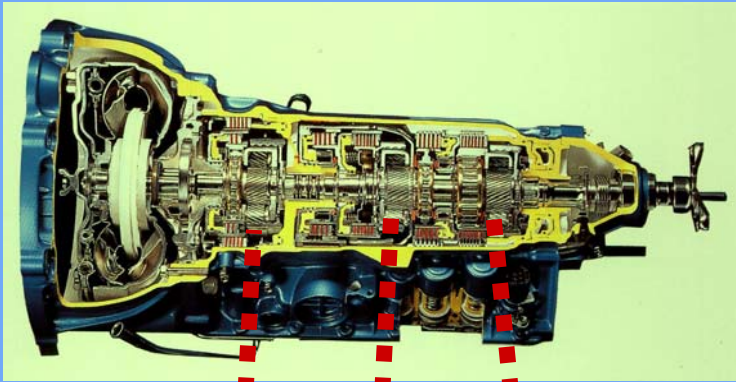
# Modellierung von Ereignissen (event)

- z. B. Schalter, Reibung, Schlupf
- Strategie
  - Integration bis zum Ereignis
  - unstetige Änderung durchführen
  - Integration neu starten
  - Werte werden 2x gespeichert, vor und nach dem Ereignis

# Modellierung von Ereignissen

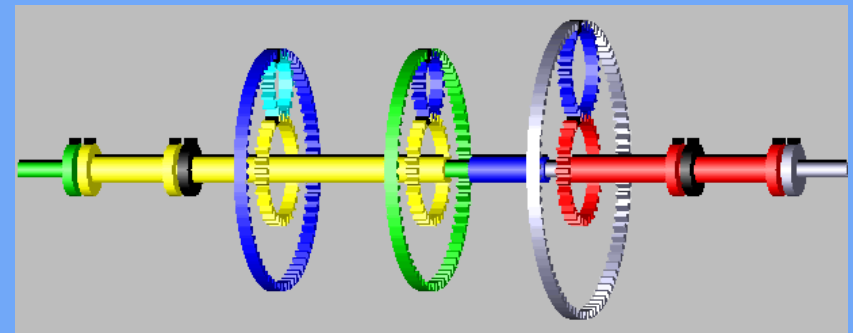
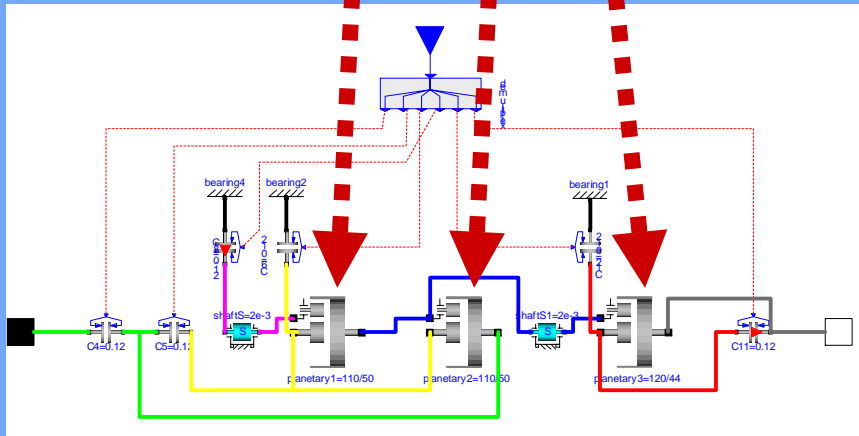
- Durch logische Ausdrücke, z.B.  
 $y = \text{if } u > 0 \text{ then } 1 \text{ else } -1$
- Glätten (stetig)  
 $y = \text{smooth}(1, \text{if } u > 0 \text{ then } 1 \text{ else } -1)$
- Event-Suche abschalten  
 $y = \text{if noEvent}(u > 0) \text{ then } 1 \text{ else } -1$

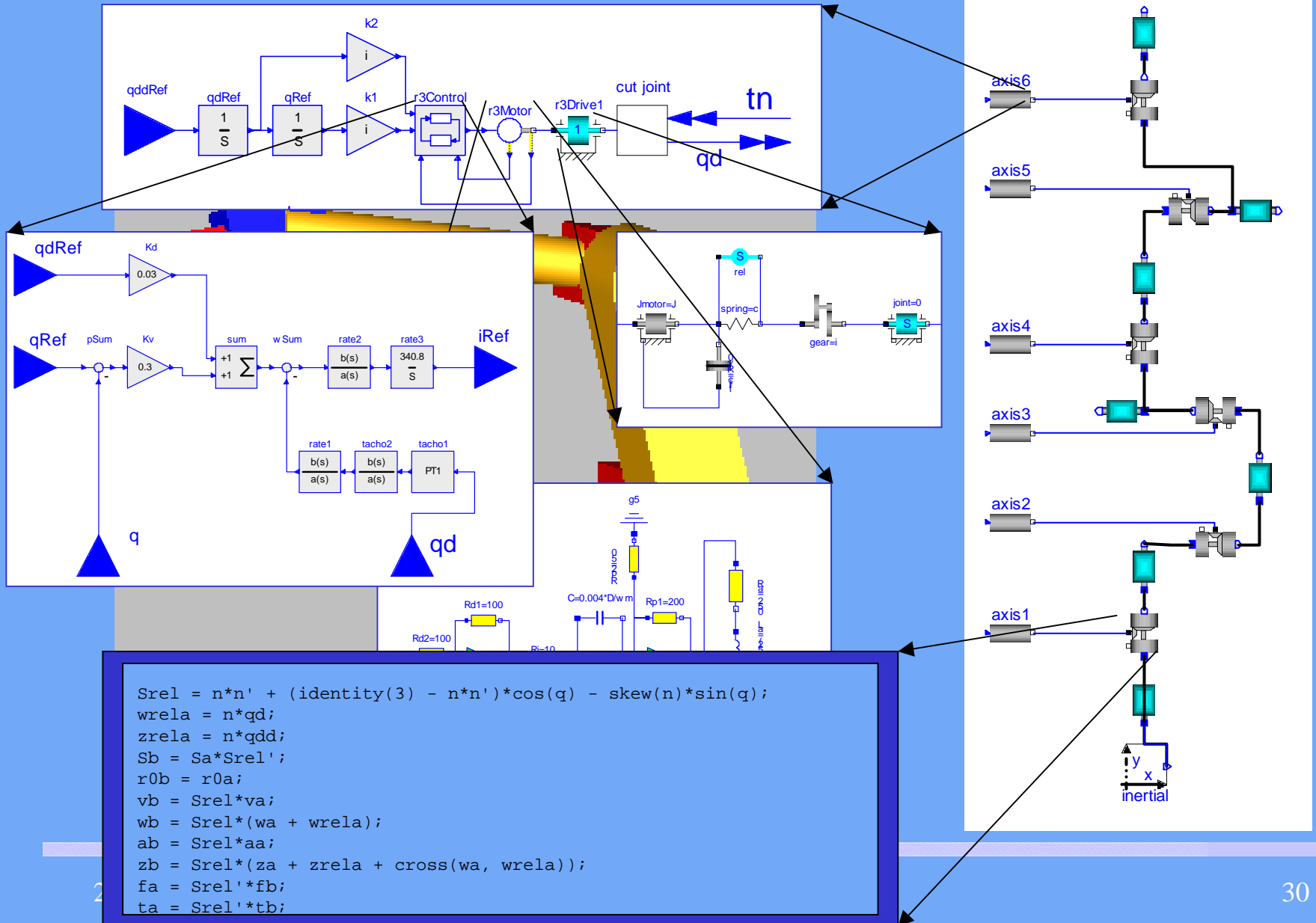
# Getriebe-Modell



Courtesy Toyota Techno-Service

## Powertrain Library





# Dymola

- versteht und simuliert Modelica
- Modellierungs- und Simulationsfenster
- übersetzt in C-Code
- Modelldokumentation in HTML, in gleicher Datei
- automatische Extraktion der Modelldokumentation

# Simulation

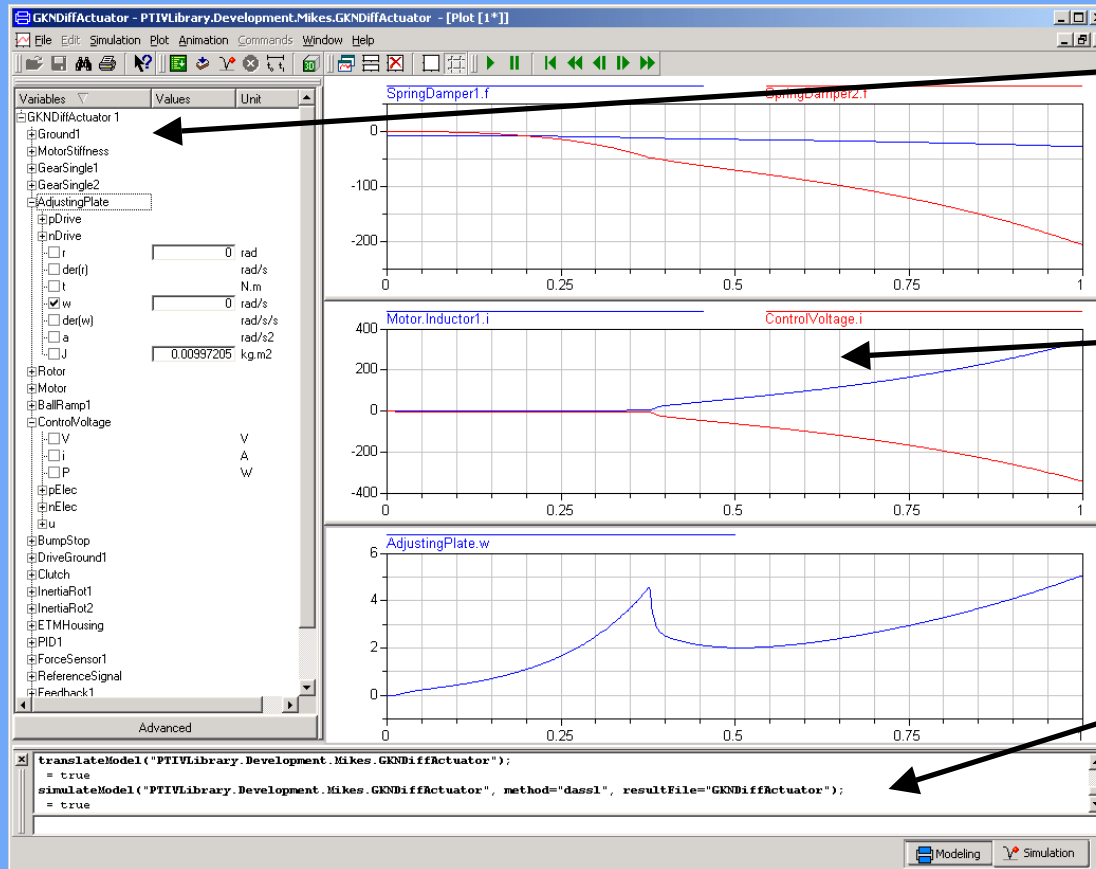
- Simulationsfenster
- automatische Initialisierung
- interaktive Studien
- viele effiziente und aktuelle Integrationsverfahren
- Plotmöglichkeit
- Animation



# Simulation

- Skripts für vorbereitete Experimente
- Linearisierung der Modelle
- Real-time Hardware-in-the-loop Simulation
- Schnittstelle zu MATLAB zur Datenauswertung
- Modell-Schnittstelle zu Simulink

# Simulationsfenster in Dymola



## Signal Browser

Alle Parameter und Ergebnisse werden hier angezeigt

## Plots

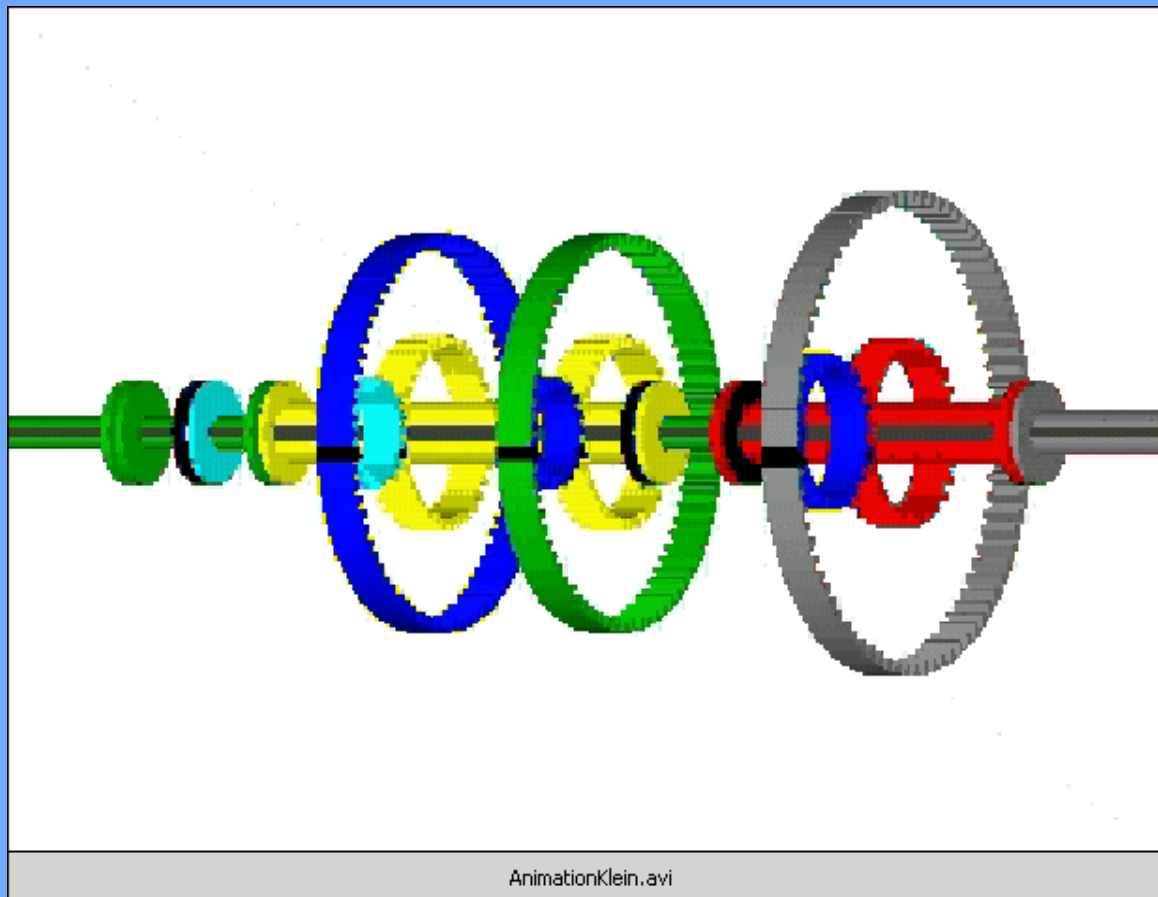
- mehrere Signale in einem Diagramm
- mehrere Plots in einem Fenster

## Command window

Befehlssprache und Aufruf von Skripten möglich



# Animation



# Dymola Experiment Skriptfile

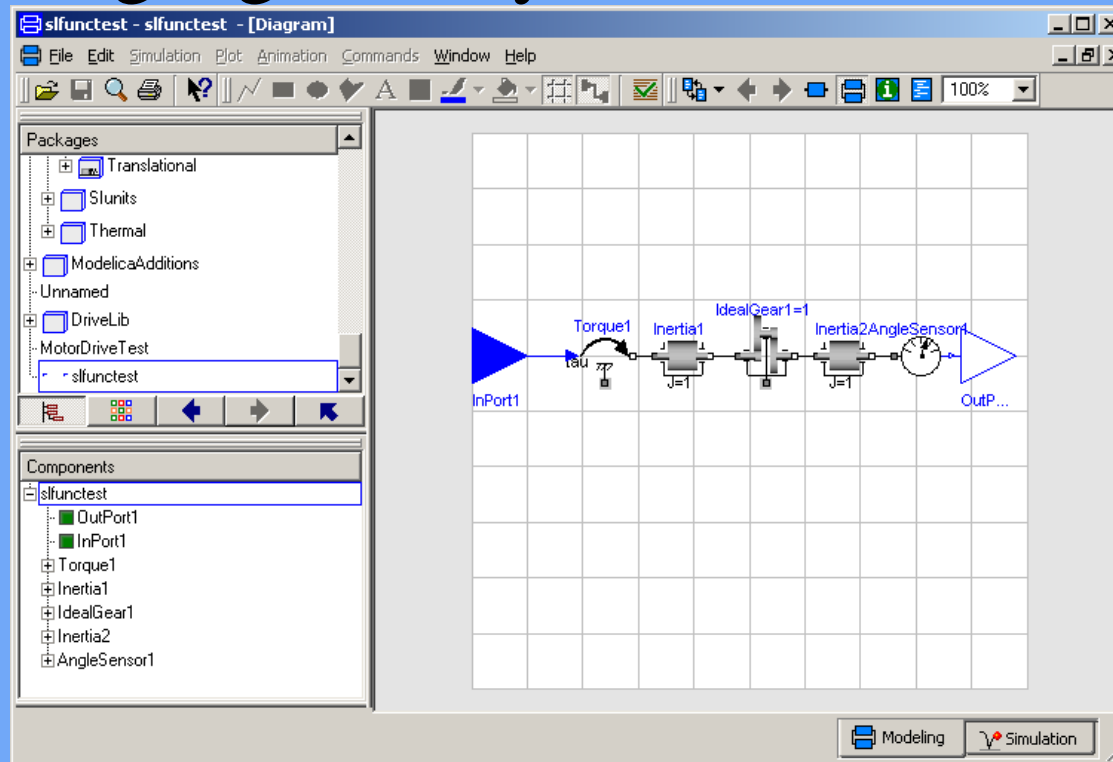
- Ablaufsteuerung
- Parametervariationen
- Plotmöglichkeiten
- Modelica Syntax
- benutzerdefinierte Funktionen

# Skript – Beispiel: Parameterstudie

```
openModel("controllerTest.mo");  
omega = 1;      // Declare omega.  
k = 1;          // Declare gain.  
for D in {0.1, 0.2, 0.4, 0.7} loop  
    // Parameter sweep over damping coefficient.  
    tr.a = {1, 2*D*omega, omega**2};  
    tr.b = {k*omega**2};  
    simulateModel("controllerTest", 0, 10);  
    plot({"u", "y"});  
end for;
```

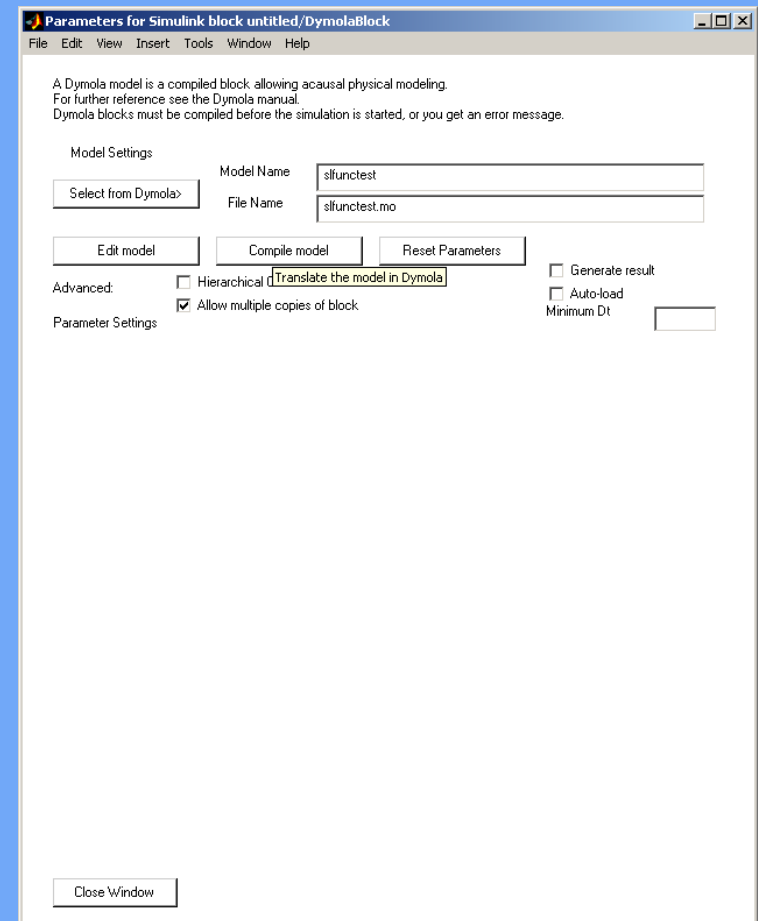
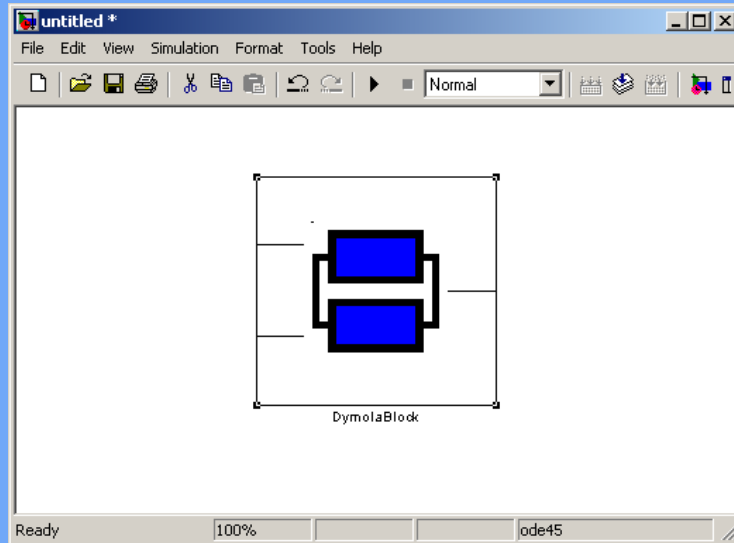
# Dymola-Simulink-Interface

## Ein-/Ausgänge im Dymola-Modell festlegen



# Dymola-Simulink-Interface

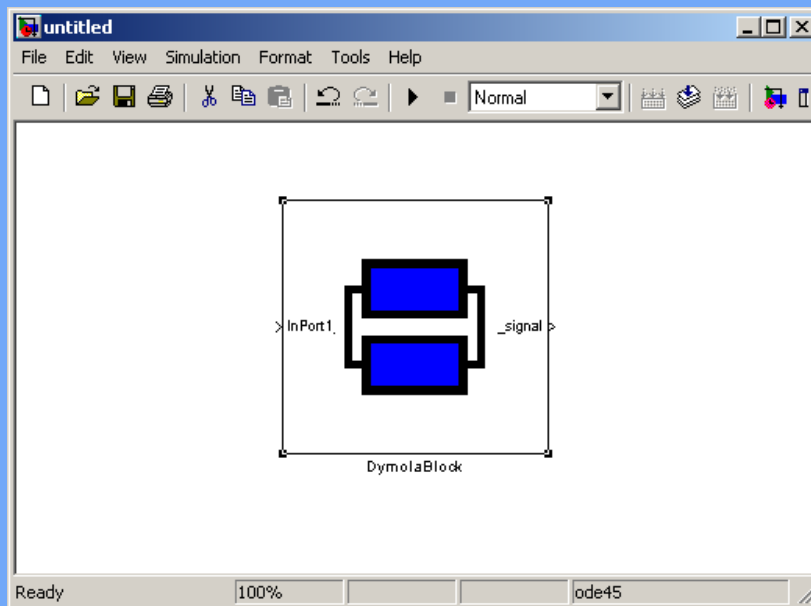
Dymola-Block in  
Simulink-Modell einfügen  
Modell übersetzen





# Dymola-Simulink-Interface

ergibt Block mit Ein- und Ausgängen



The screenshot shows the 'Parameters for Simulink block untitled/DymolaBlock' dialog box. It has a menu bar with 'File', 'Edit', 'View', 'Insert', 'Tools', 'Window', and 'Help'. The main text area contains the following information:

A Dymola model is a compiled block allowing acausal physical modeling.  
For further reference see the Dymola manual.  
Dymola blocks must be compiled before the simulation is started, or you get an error message.

**Model Settings**

Model Name:   
File Name:   
Buttons:

**Advanced:** ☐ Hierarchical Connector as Bus ☐ Generate result  
☒ Allow multiple copies of block ☐ Auto-load  
Minimum Dt:

**Parameter Settings**

Inertia1.J:   
IdealGear1.ratio:   
Inertia2.J:   
Inertia2.phi:   
Inertia2.w:

**Start values**

# Einige Neuerungen in Modelica 2.1

- Felder und deren Indizierung
- Konnektoren verbessert
  - Verbindung hierarchischer Konnektoren
  - überbestimmte Konnektoren
- Break- und Return-Anweisungen
- Builtin-Funktionen: String, Integer, Semilinear
- Neue Annotationen
- und mehr, sh.

[www.modelica.org/news\\_items/modelica\\_2\\_1](http://www.modelica.org/news_items/modelica_2_1)

# Weiterentwicklung: Version 5.3 nach 5.4

Beispiele (sh. Dymola Release Notes 5.3++++)

- bessere automatische Initialisierung
- Modelica 2.1 mit automatischer Modellkonversion
- bessere Tabelleneditoren

# Zusammenfassung

- Modelica soll (de-facto) **Standard** werden
- Dymola unterstützt Modelica voll
- viele Experten arbeiten an Modelica mit
- freie Bibliotheken stehen auf der Modelica-Homepage zur Verfügung
- einige werden mit Dymola geliefert
- zusätzliche Bibliotheken können erworben werden
- Homepages:  
    [www.dynasim.se](http://www.dynasim.se)  
    [www.modelica.org](http://www.modelica.org)