

# Überblick zu Dymola und Modelica



# Inhalt

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

# Dymola

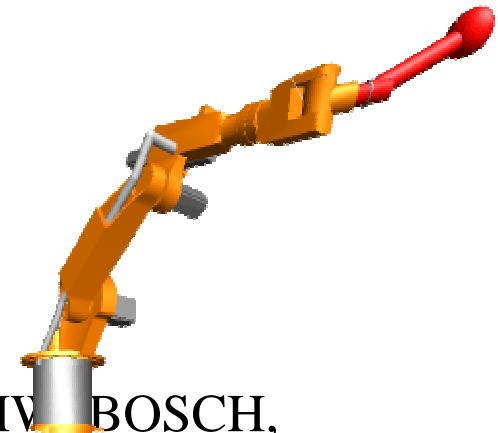
- Dymola ist eine Software zur Modellierung und Simulation aus verschiedenen Ingenieurdisziplinen
- entwickelt von Dynasim AB in Schweden
- Vertrieb im deutschsprachigen Raum:  
Bausch-Gall GmbH
- Dymola versteht den offenen Sprachstandard  
Modelica

# BAUSCH-GALL GmbH

- Seit 1987 GmbH
- Hans Gall und Dr. Ingrid Bausch-Gall seit 1981 selbständig
- Vertrieb von Simulationssoftware
- Beratung im Simulationsumfeld
- Projektarbeit
- Schulungen
  - Dymola, Spice, Simulink, Matlab

# Die Firma Dynasim

- Entwickler von Dymola
- gegründet 1992 von Dr. Hilding Elmqvist
- Jetzt: 9 Angestellte, davon 7 technisch/wissenschaftlich
- Zusammenarbeit mit Wissenschaftlern
- Einige Kunden
  - weltweit, z.B.: Toyota, Volvo, Ford, GM, EDF
  - Deutschland, z.B.: ABB, Airbus, BASF, Behr, BMW, BOSCH, DaimlerChrysler, Hella, Siemens, ZF, Volkswagen, Universitäten und Forschungsinstitute
  - Österreich, z.B.: AVL, Elin, MAGNA STEYR, Arsenal Research, TU Wien



# Dymola

- Ziel: einheitliche Modellierung verschiedener Ingenieurdisziplinen
- einige Stichworte
  - “Multi-Engineering Modeling and Simulation”
  - 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.com](http://www.dynasim.com)

## Modelica Association

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



## Ziele der Modelica Assosiation

- offene Sprachdefinition
- objektorientierte Sprachen vereinheitlichen
- Wiederverwendbarkeit der Modelle fördern
- deklarative, d.h. mathematische Gleichungen statt prozedurale Zuweisungen
- Sprache soll effiziente Simulation unterstützen

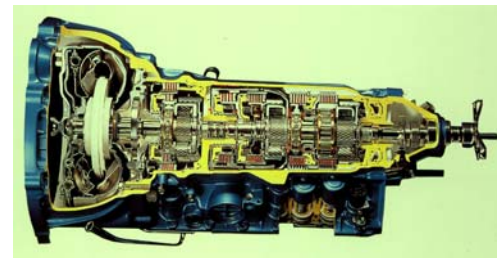
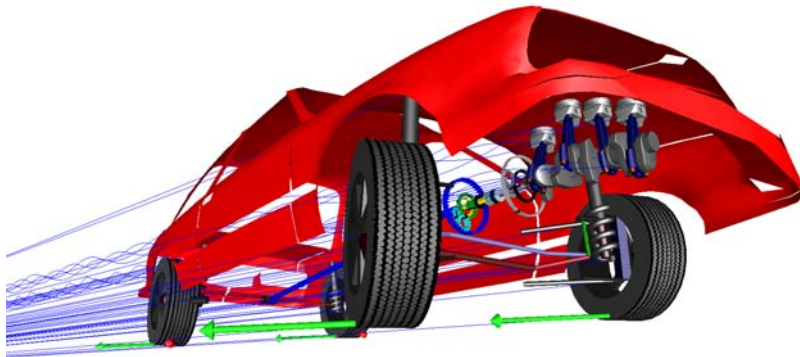
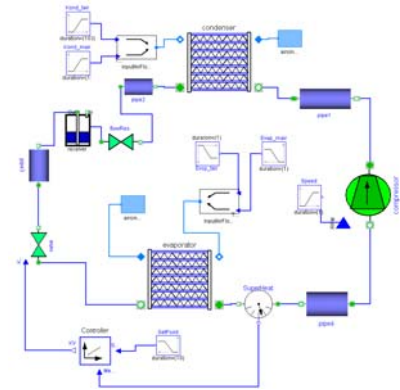
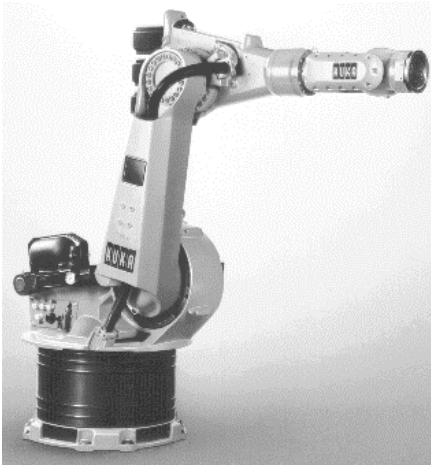
## Modelica - Entwicklungsstand

- Entwurf startete im September 1996
- Version 1.0 im September 1997
- Heute: Version 2.2
- Werkzeuge und Bibliotheken stehen zur Verfügung
- Homepage: [www.modelica.org](http://www.modelica.org)
- regelmäßige Tagungen
- letzte Tagung: März 2005 an der TU HH
- nächste Tagung: 5.-7. Sept. 2006 in Wien

# Dymola-Entwicklungsziele

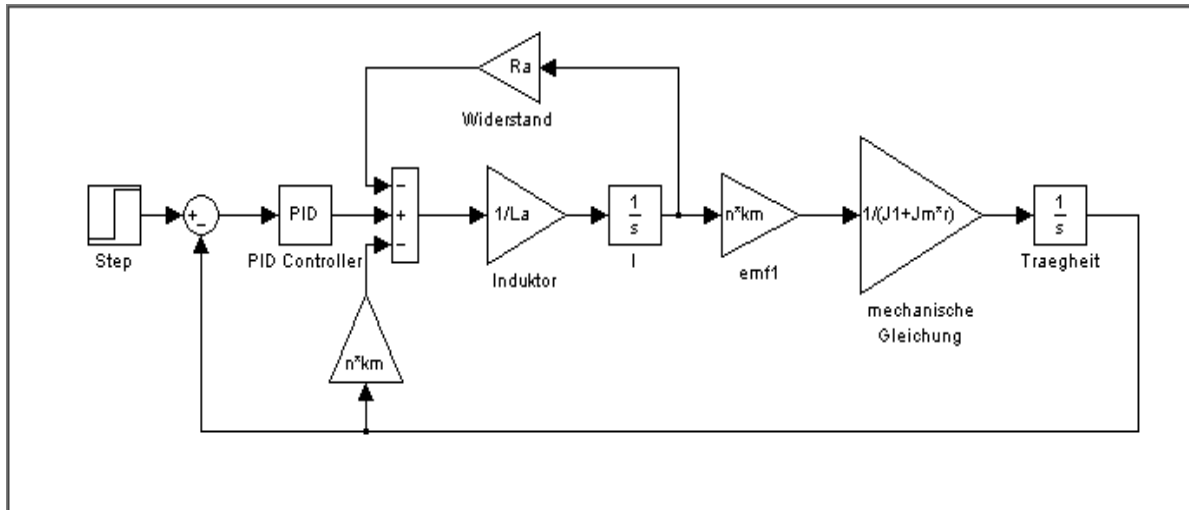
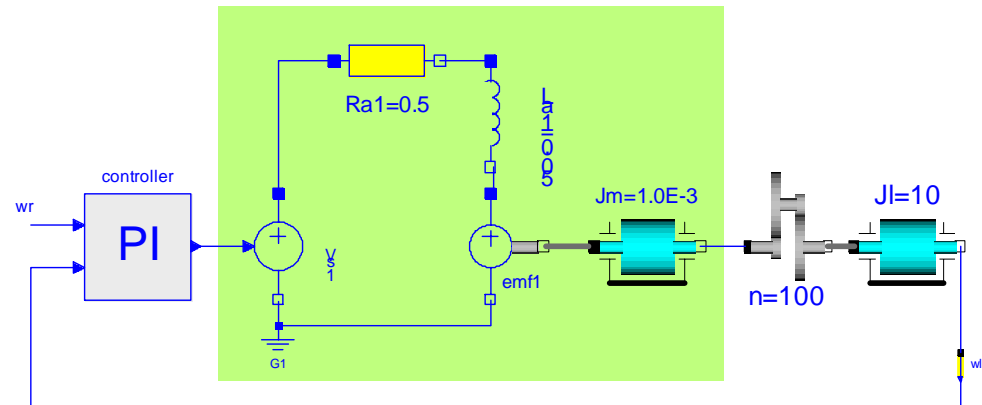
- Idee, wie Modelica
- Jedoch: Software zur Modellierung komplexer Systeme
- Modelle verschiedener Ingenieurdisziplinen
- Verfügbarkeit der Modelle als Quellcode
- homogenes Simulationsmodell
- effiziente Simulation
- Verkürzung der Entwicklungszeit

# Anwendungsbeispiele



# Energieflußorientierte Modellierung

Dymola

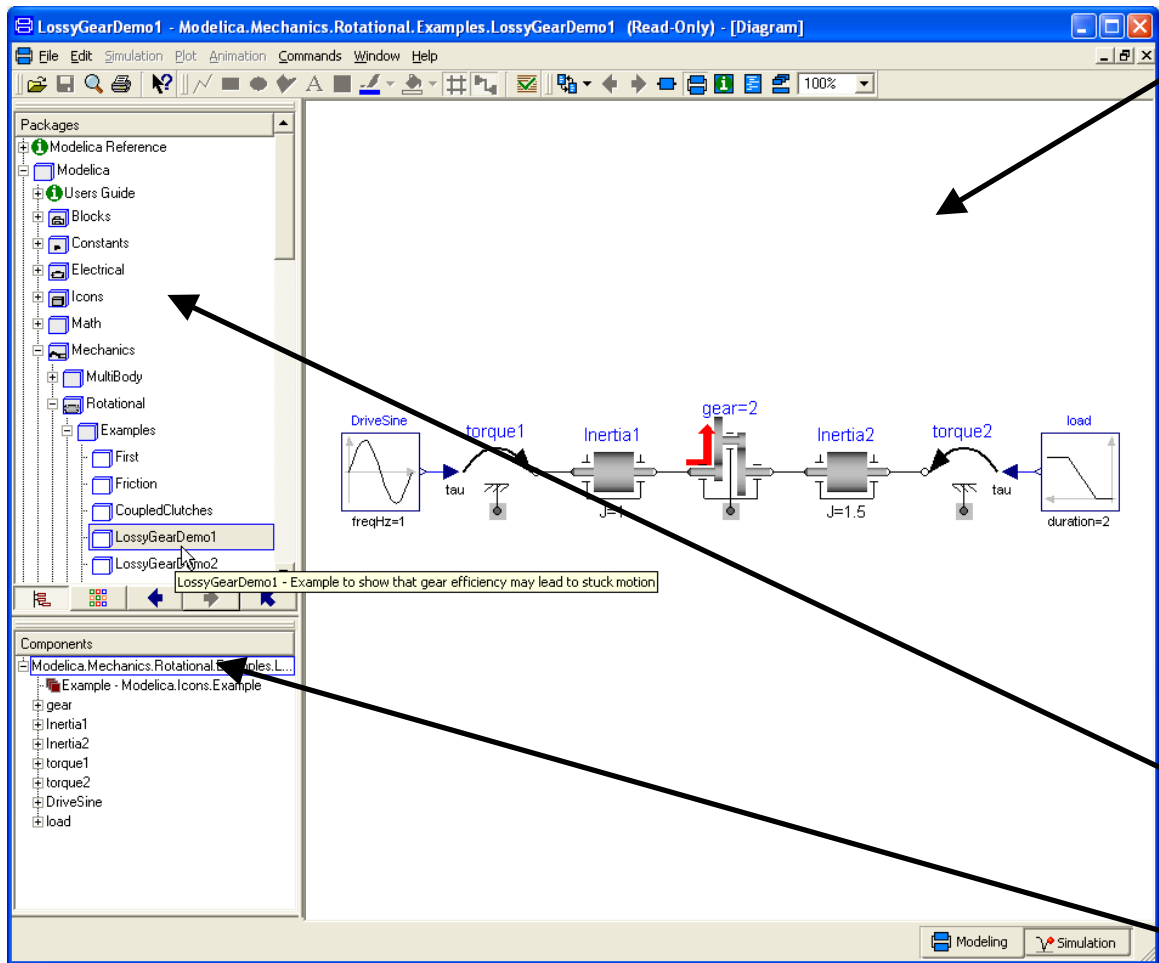


signalflußorientierte  
Modellierung

# Modellierung mit Dymola

- versteht Modelica weitgehend
- Gleichungen als Modellbasis
- objektorientierte Struktur der Modelle  
Klassen, Konnektoren, Blöcke
- übliche Datenstrukturen und Programmierkonzepte
- Units, Kommentare
- Felder und Matrizen, Operatoren
- Funktionen und Algorithmen
- symbolische Manipulation der Gleichungen
- Modellbibliotheken

# Dymola-Modellierungsfenster



Modellierungsfenster

Icon Layer: Definition der Graphik

Diagram Layer: Zusammenstellen des Modells

Documentation Layer

HTML-Dokumentation einfügen

Equation Layer:

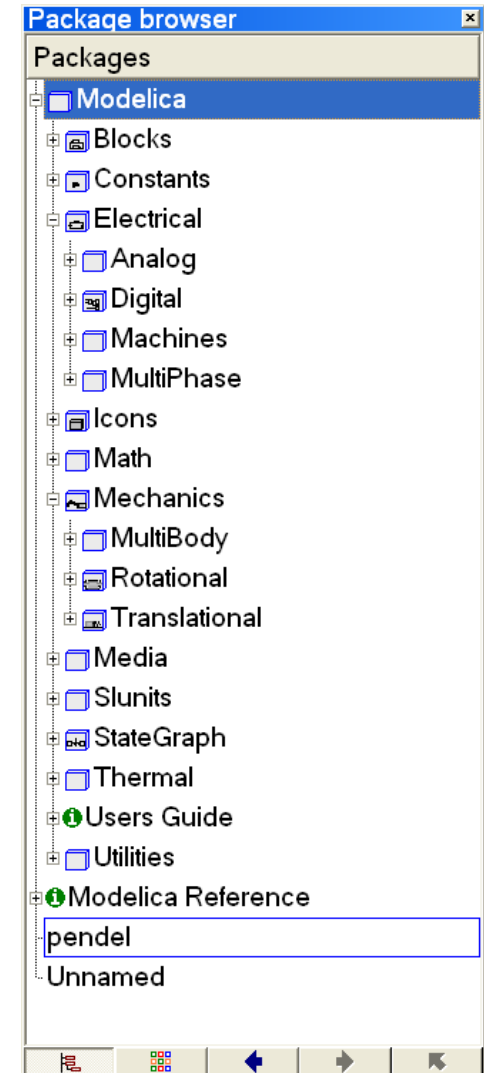
Gleichungen als Modelica-Kode

Model Library Browser

Model Browser

# Modelica-Bibliotheken

- hierarchisch
- Komponenten lassen sich sinnvoll anordnen
- grosse Komponentenvielfalt
- Modelica-Bibliotheken enthalten z.B.:
  - einfache Regelungstechnik (Blöcke)
  - 1D-Mechanik (rotatorisch, translatorisch)
  - Mehrkörpersysteme
  - analoge und digitale Elektrik
  - elektrische Maschinen
  - einfache Pneumatik und Hydraulik
  - Zustandsautomaten (State Graph)



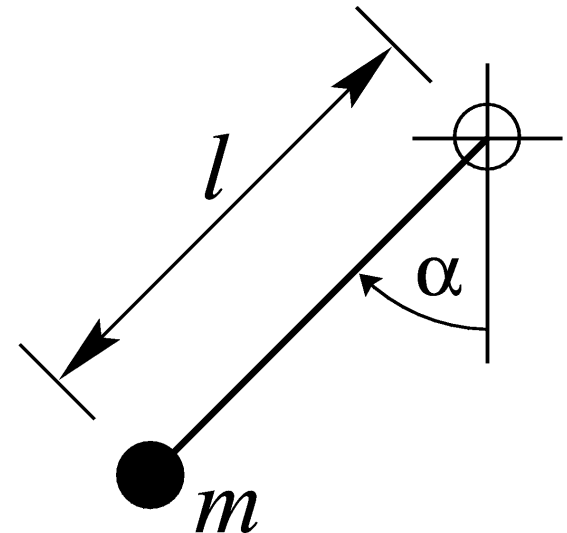


# Modelica-Bibliotheken

- kommerzielle Bibliotheken für
  - Antriebsstränge (Powertrain)
  - Pneumatik
  - Hydraulik
- von der Firma Modelon ([www.modelon.se](http://www.modelon.se))
  - Klimaanlage (AirConditioning)
  - Bald: Fahrdynamik (VehicleDynamics)
- weitere Bibliotheken sind in der Entwicklung
- weitere freie Bibliotheken  
[www.modelica.org](http://www.modelica.org)

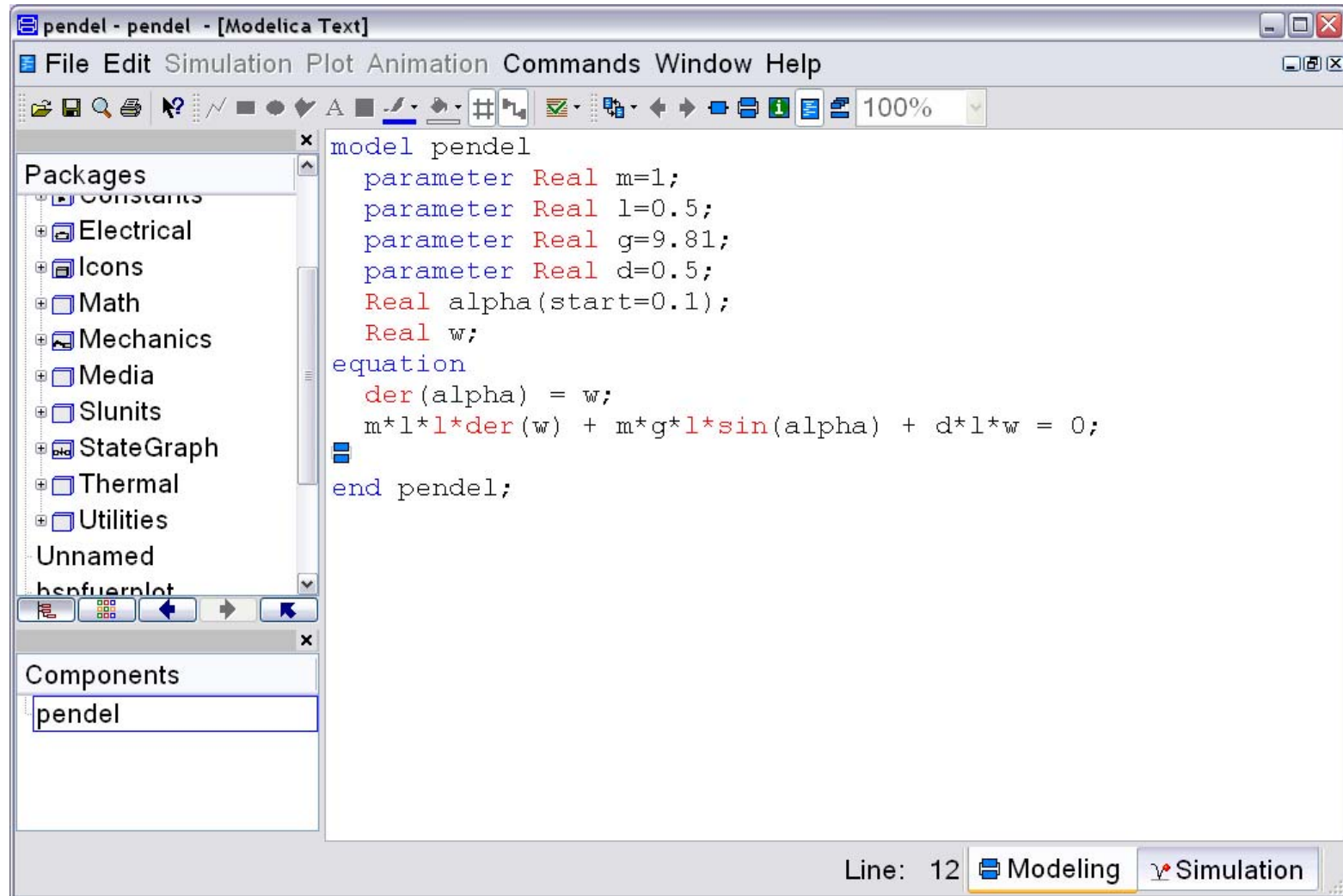
# Grundlage der Modellbildung in Dymola

- Modellierungsgrundlage:
  - Differentialgleichungssysteme
  - Differenzengleichungen
  - algebraische Gleichungen
- Einfacher Fall: Pendel



$$m l^2 \ddot{\alpha} + m g l \sin(\alpha) + d l^2 \dot{\alpha} = 0$$

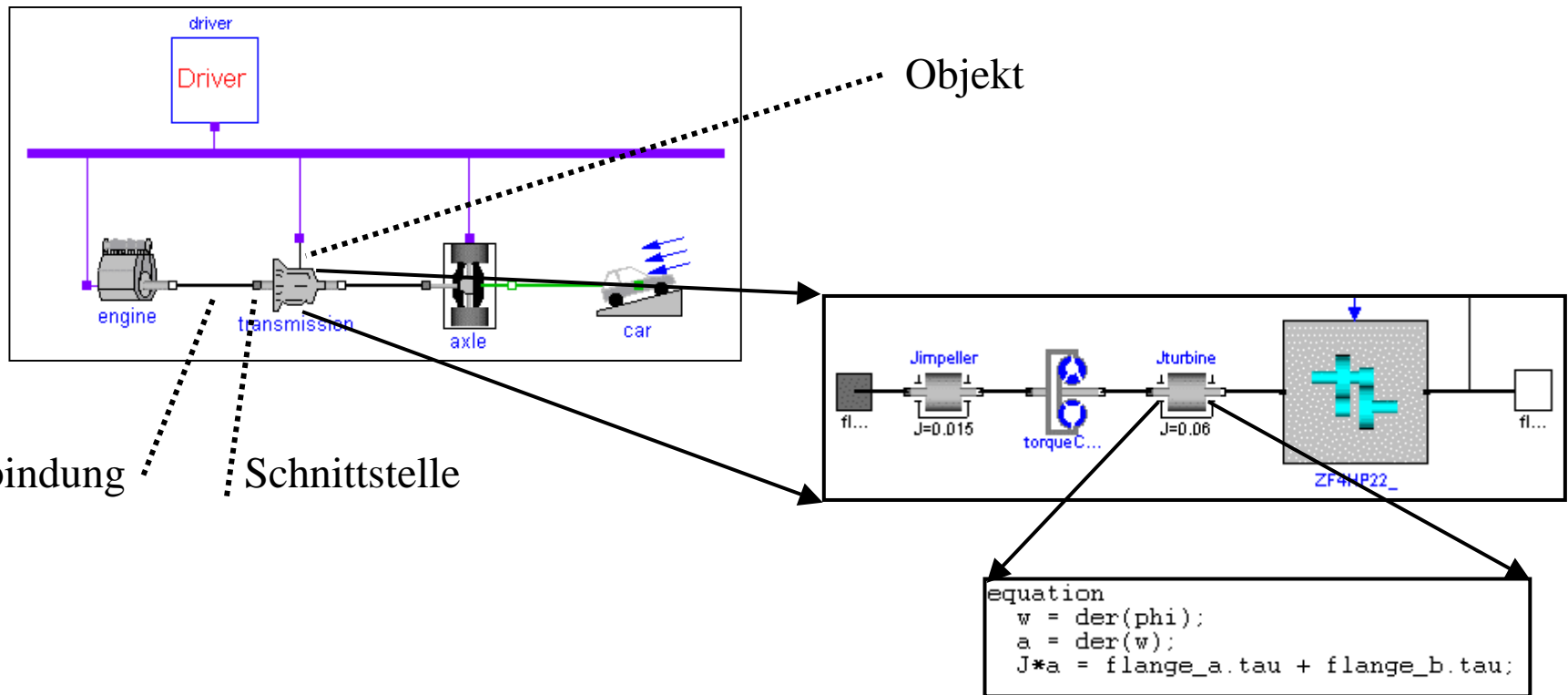
# Dymola-Modell



# Modellierung

- sog. Objektdiagramme
- jedes Bauteil oder Modellteil als abgeschlossenes Objekt
- physikalische Konnektoren
- sinnvolle Graphik
- HTML-Dokumentation
- Beispiele:
  - allgemeines Objektdiagramm
  - Objekt: Rotational Inertia aus der Modelica Library

# Objektdiagramme



# Objektdiagramme

- jedes Icon repräsentiert ein physikalisches oder logisches Objekt
- Verbindungen stellen tatsächliche physikalische Verbindungen dar
- jedes Objekt wird hierarchisch erstellt oder durch Gleichungen beschrieben
- mit **symbolischen** Algorithmen, wird die Modelica Beschreibung in die Zustandsform transformiert

$$dx/dt = f(t, x, u)$$

$$y = g(t, x, u)$$

# Modelica - Rotational - Inertia

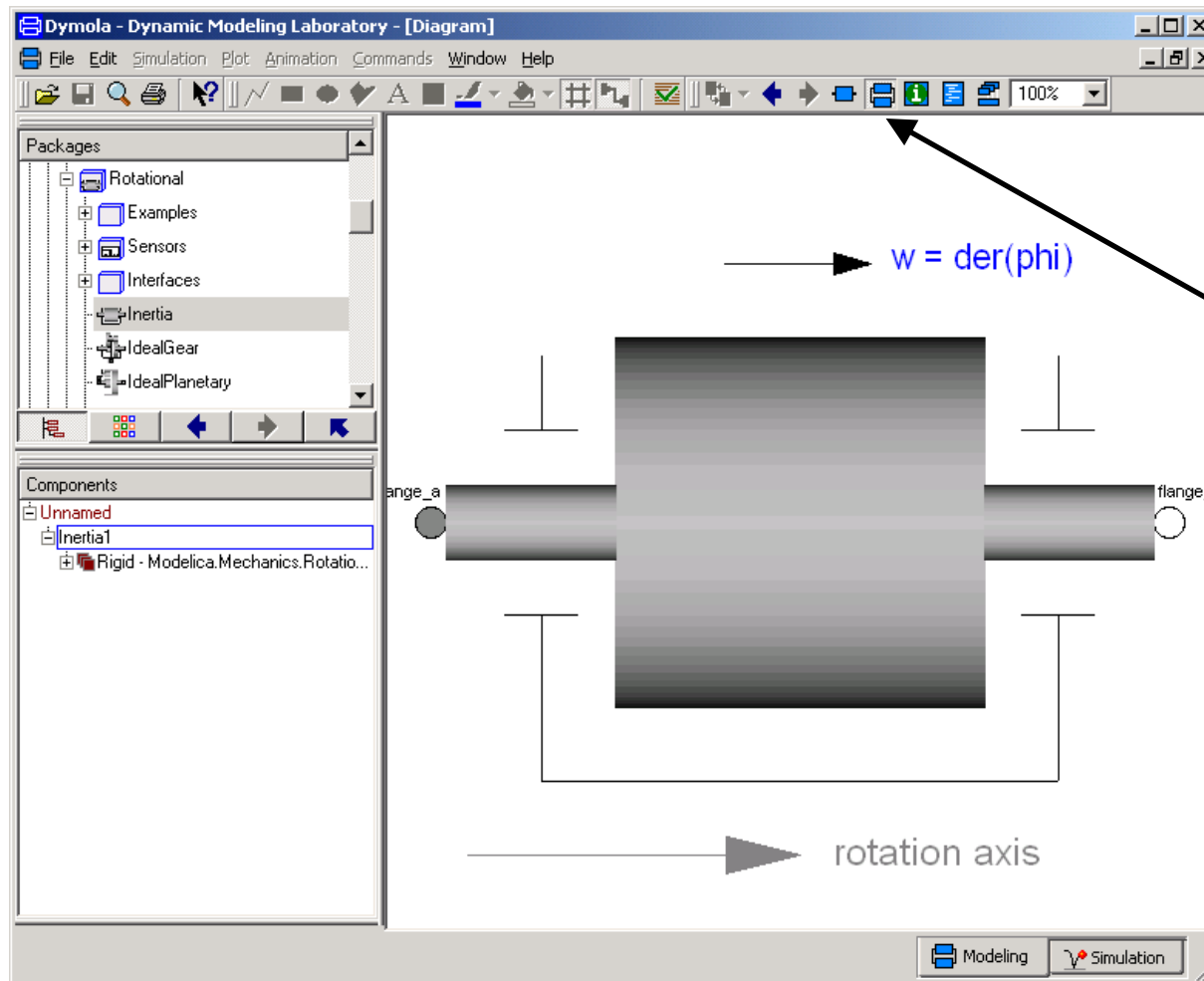
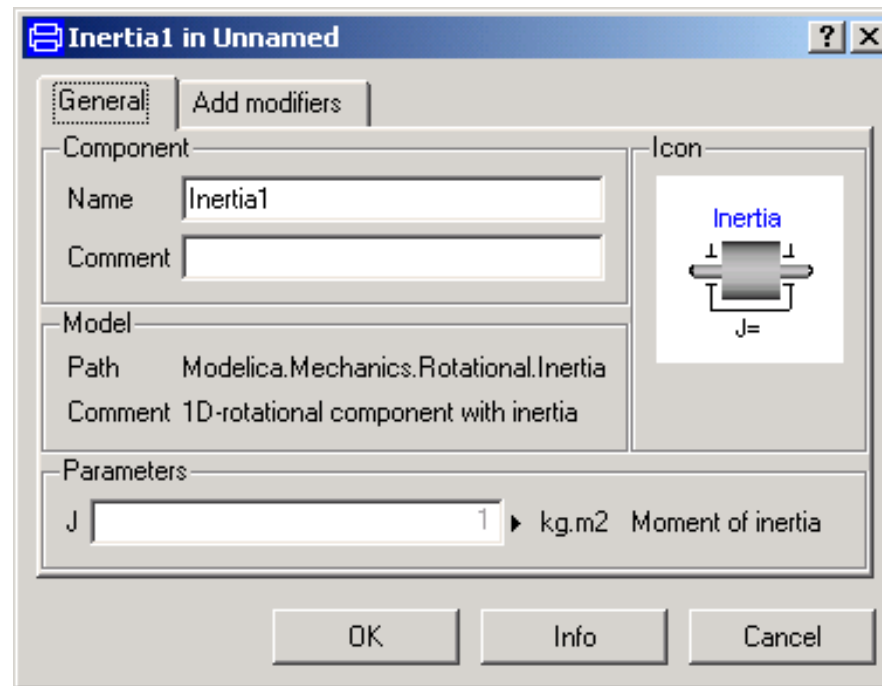


Diagram-  
Layer

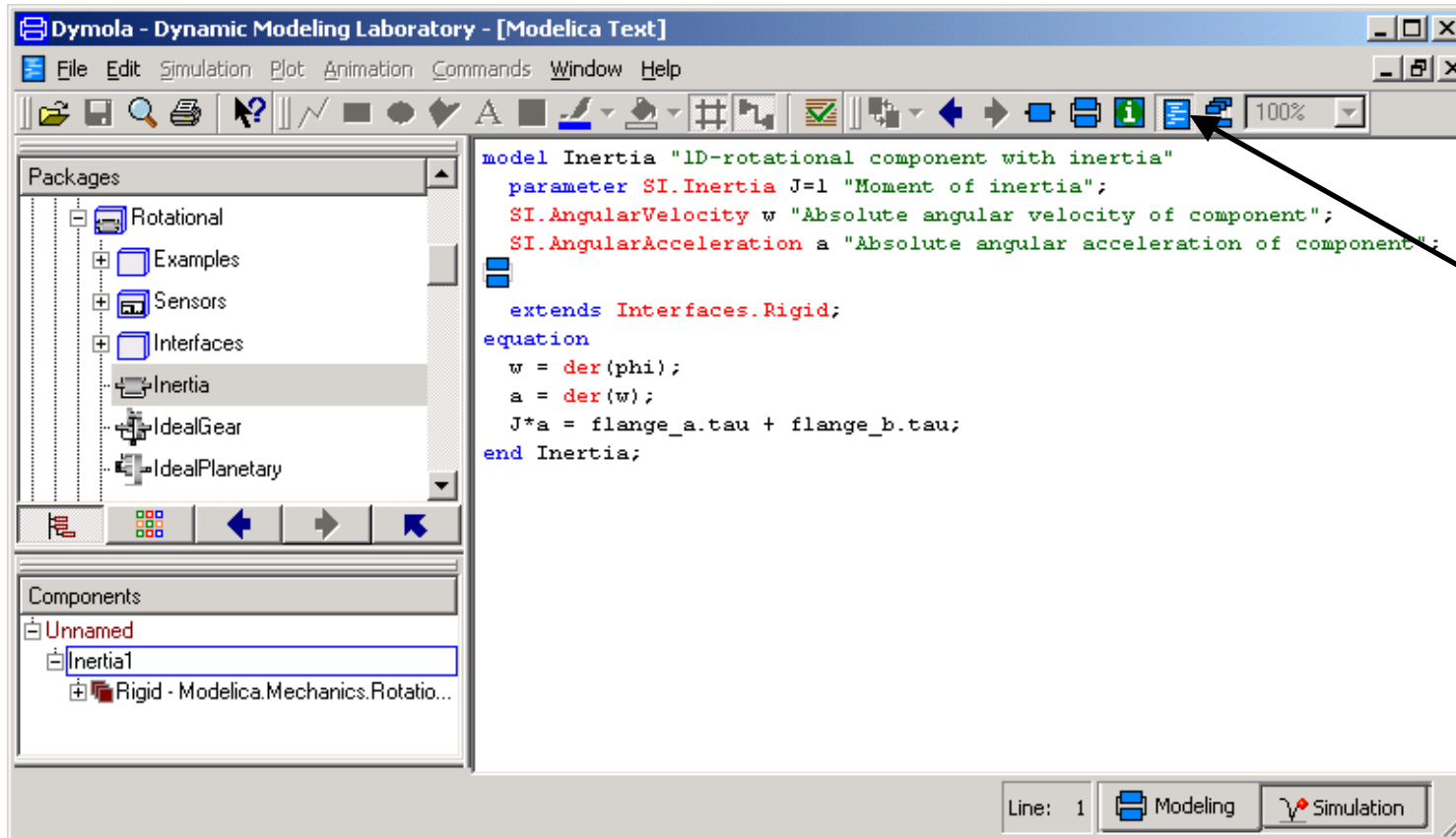
# Modelica - Rotational - Inertia

Parameterfenster wird automatisch erstellt

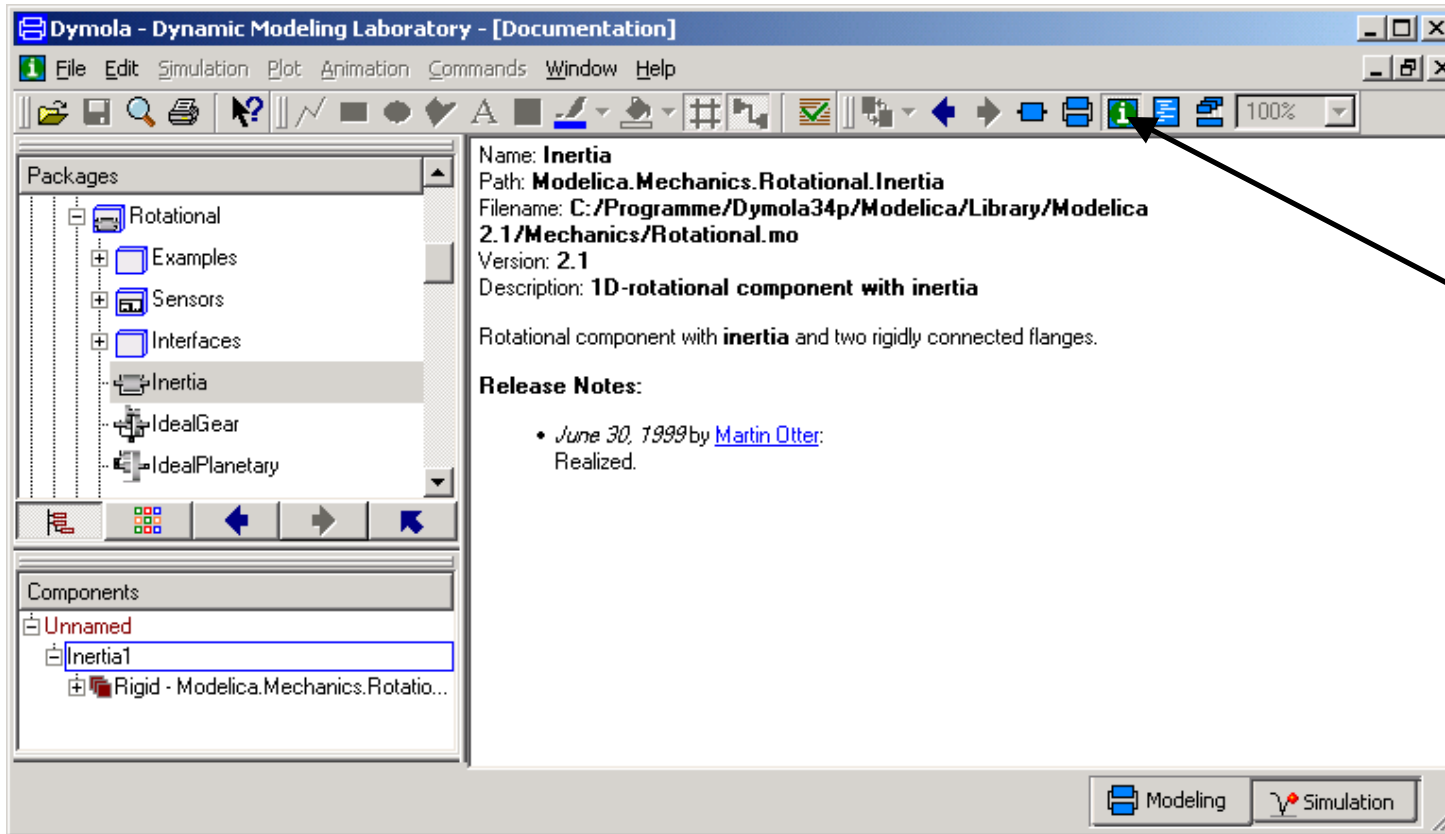




# Modelica - Rotational - Inertia

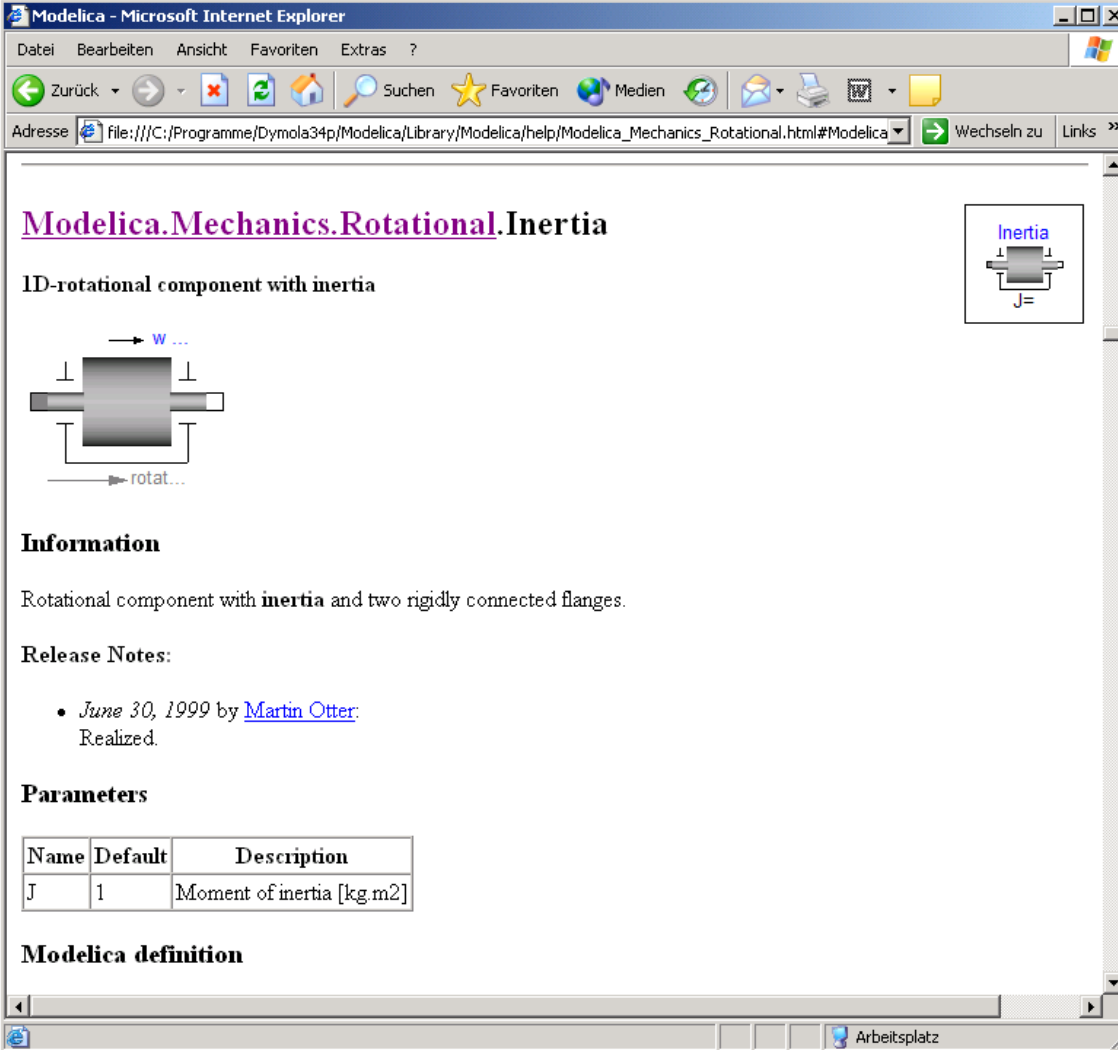


# Modelica – Rotational - Inertia



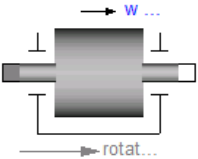
Beschreibung  
als HTML

# Modelica – Rotational - Inertia



**Modelica.Mechanics.Rotational.Inertia**

ID-rotational component with inertia



**Information**

Rotational component with **inertia** and two rigidly connected flanges.

**Release Notes:**

- *June 30, 1999* by [Martin Otter](#):  
Realized.

**Parameters**

Name	Default	Description
J	1	Moment of inertia [kg.m2]

**Modelica definition**

automatisch  
erstellte  
Dokumentation

# 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

Grundregeln an Knoten:

Flüsse summieren sich zu Null (Energieerhaltung)

Potentiale haben den gleichen Wert

# Konnektoren

```
connector Flange_a
```

```
  SI.Angle phi "Absolute rotation angle of flange";
```

```
  flow SI.Torque tau "Cut torque in the flange";
```

```
end Flange_a;
```

```
connector Flange_b
```

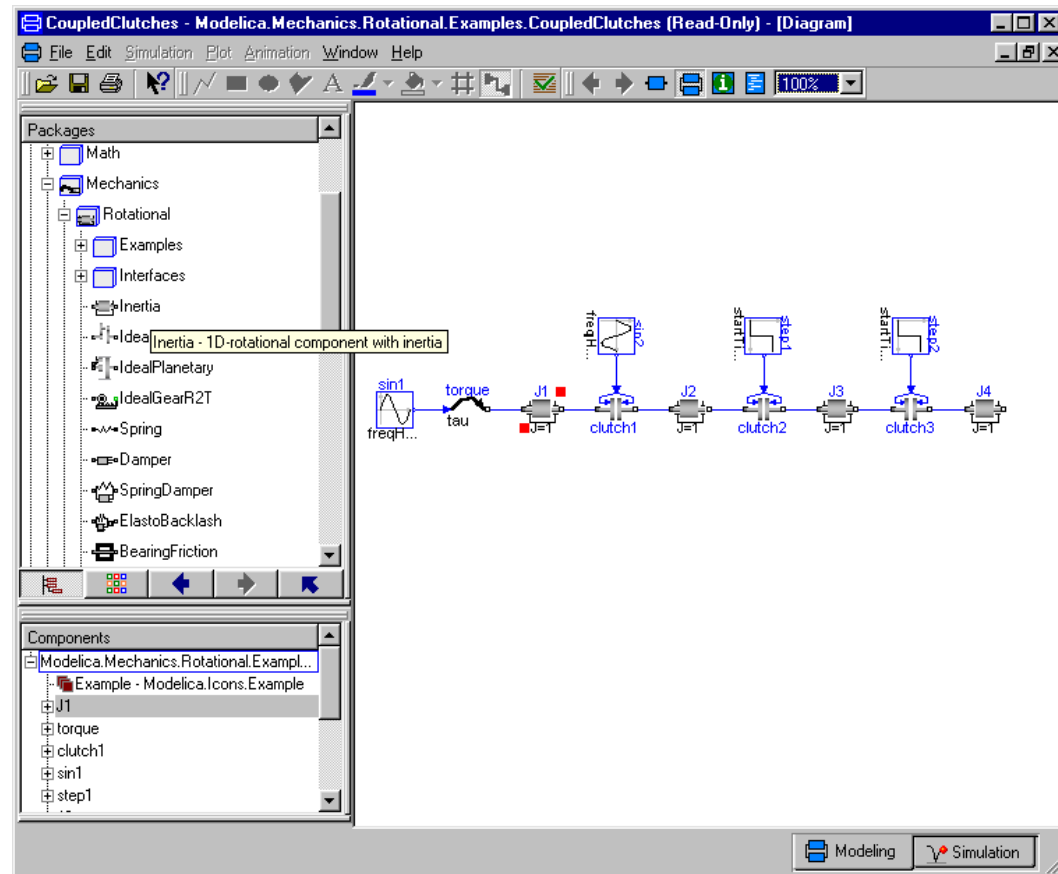
```
  SI.Angle phi "Absolute rotation angle of flange";
```

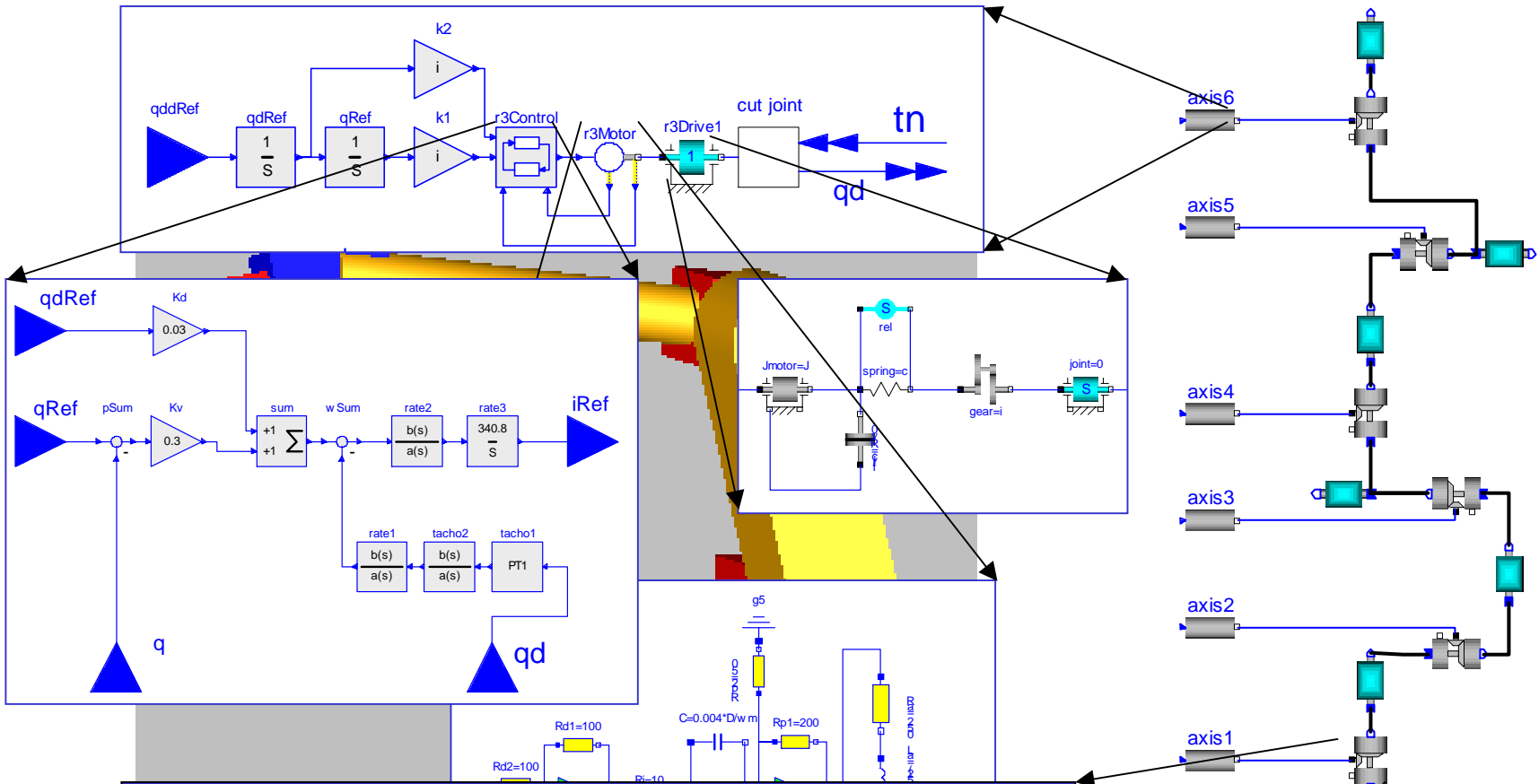
```
  flow SI.Torque tau "Cut torque in the flange";
```

```
end Flange_b;
```

# Modellierung aus Objekten

- Objekte sind in Bibliotheken
- Modell aus Grundobjekten zusammengesetzt
- Diagramm erscheint im Modellierungsfenster
- Objekte parametrieren
- Hierarchie für große Modelle

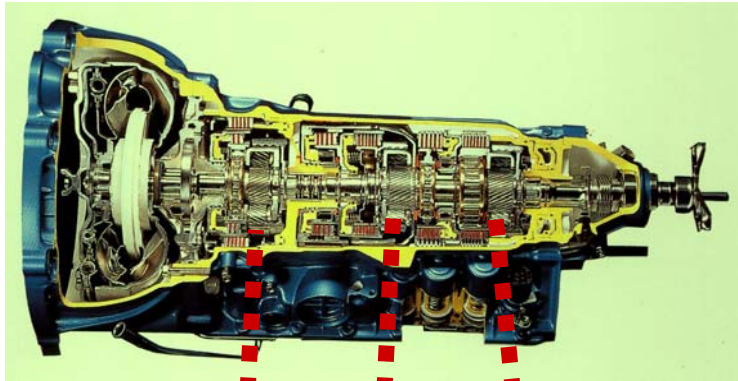




```

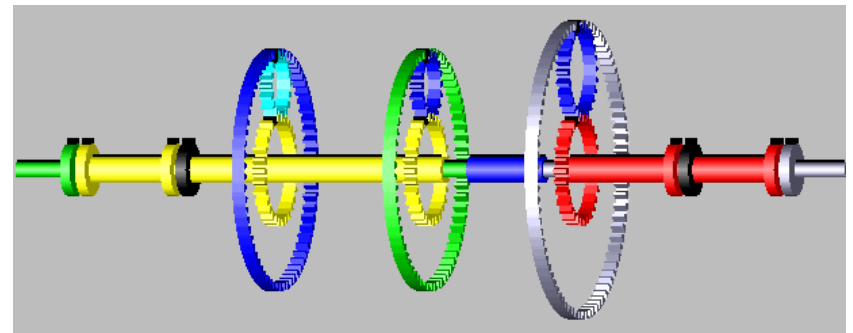
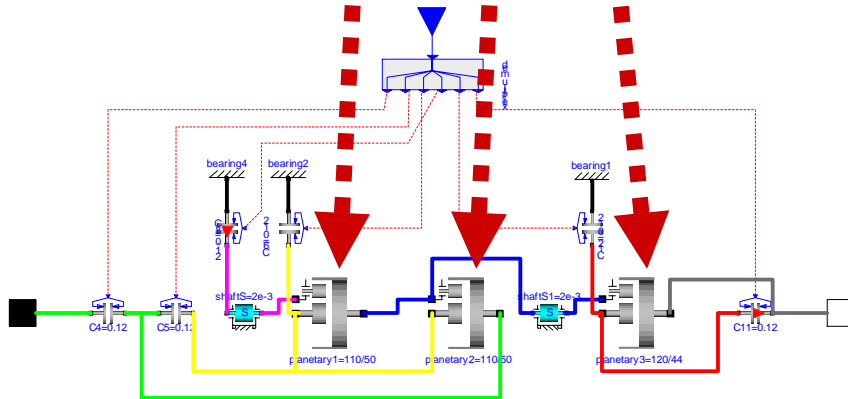
Srel = n*n' + (identity(3) - n*n')*cos(q) - skew(n)*sin(q);
wrela = n*qd;
zrela = n*qdd;
Sb = Sa*Srel';
r0b = r0a;
vb = Srel*va;
wb = Srel*(wa + wrela);
ab = Srel*aa;
zb = Srel*(za + zrela + cross(wa, wrela));
fa = Srel'*fb;
ta = Srel'*tb;
    
```

# Getriebe-Modell



Courtesy Toyota Techno-Service

## Powertrain Library





# 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  
 $y = \text{if noEvent}(u > 0) \text{ then } 1 \text{ else } -1$

# Dymola

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

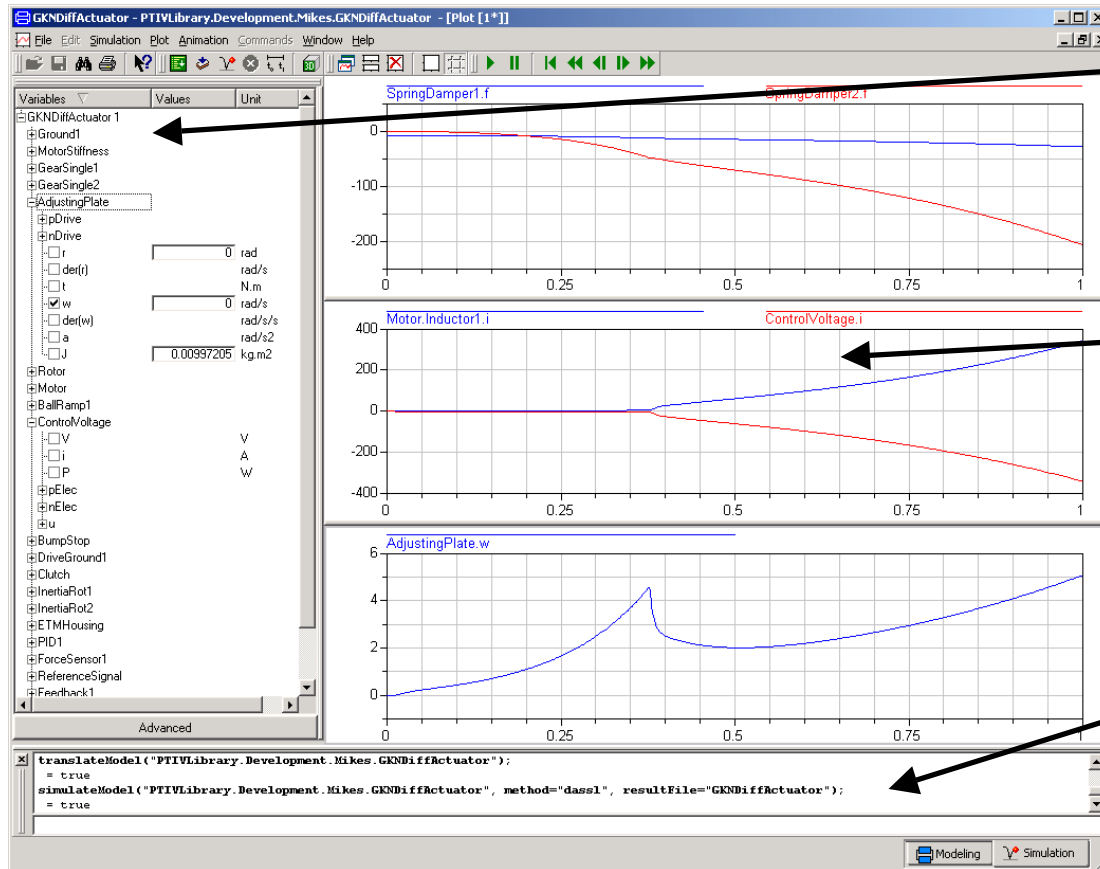
# Simulation

- 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

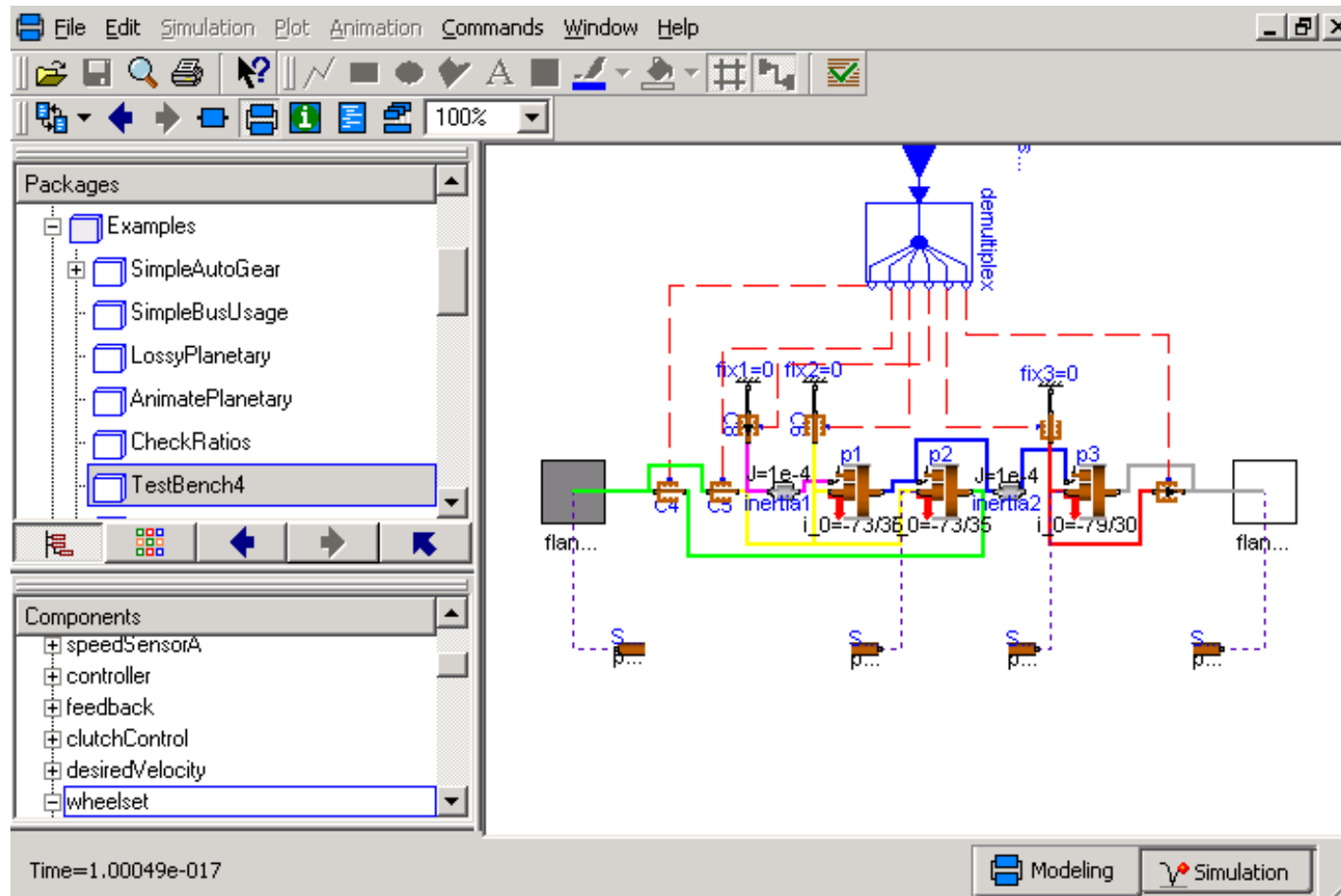
## Plots

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

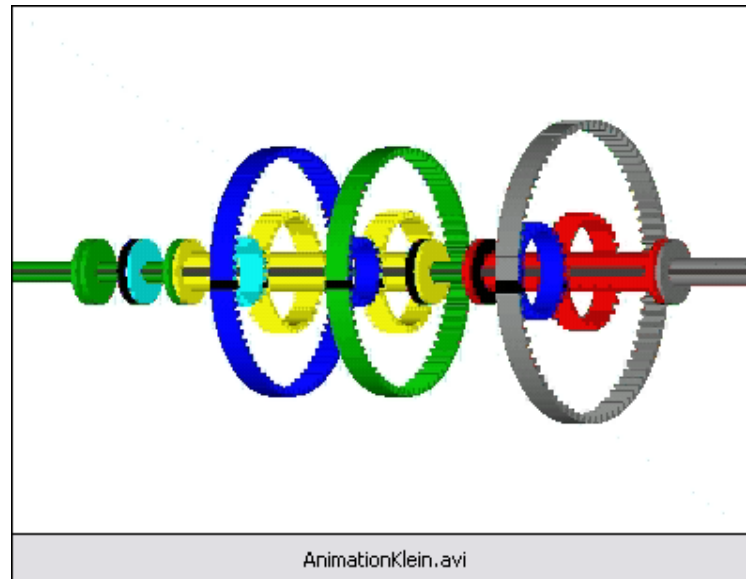
## Command window

Befehlssprache und Aufruf von Skripten möglich

# Animationsbeispiel: Getriebe



# Animation





# Dymola Experiment Skriptfile

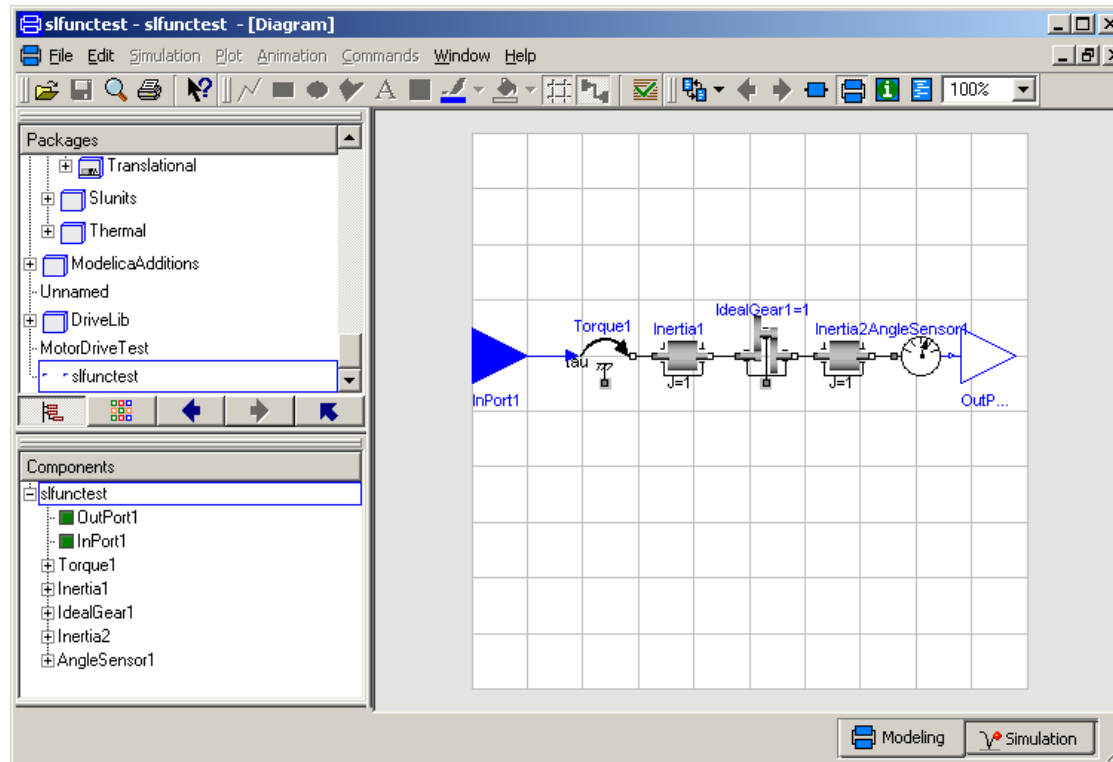
- Ablaufsteuerung
- Parametervariationen
- Plotmöglichkeiten
- Modelica Syntax
- benutzerdefinierte Funktionen

# 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;
```

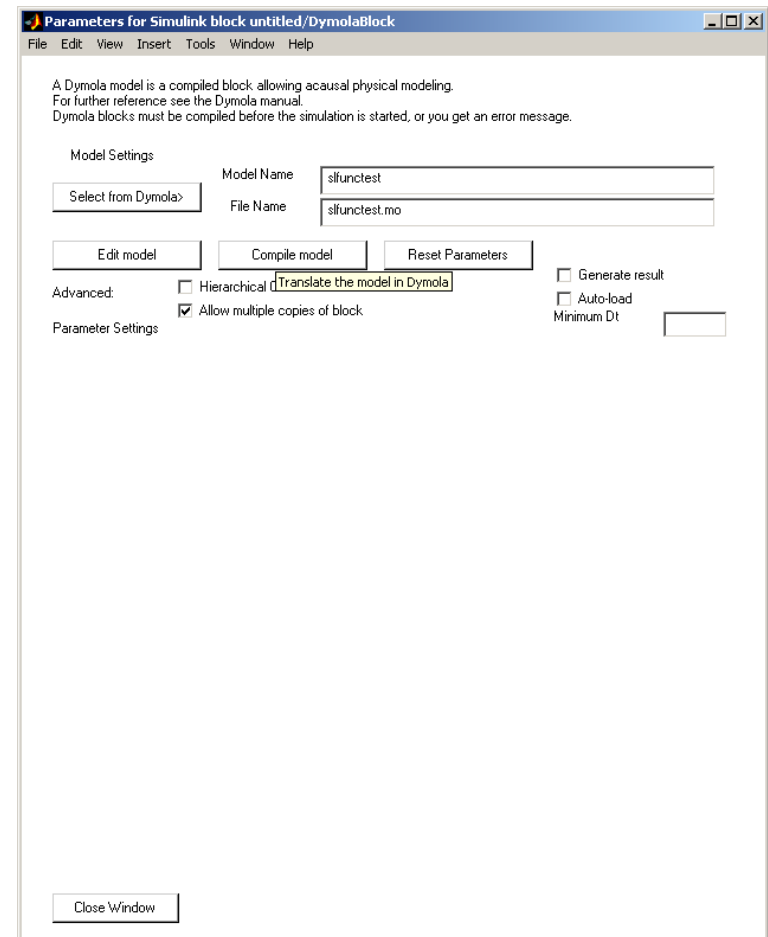
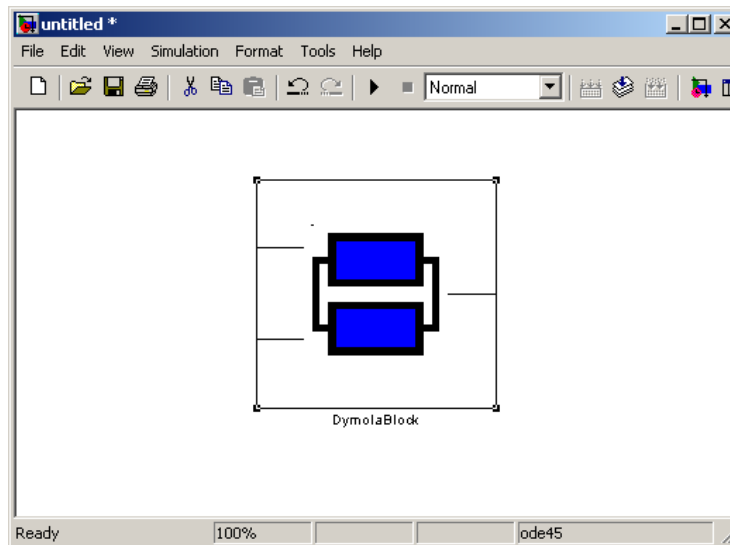
# Dymola-Simulink-Interface

- Ein-/Ausgänge im Dymola-Modell festlegen



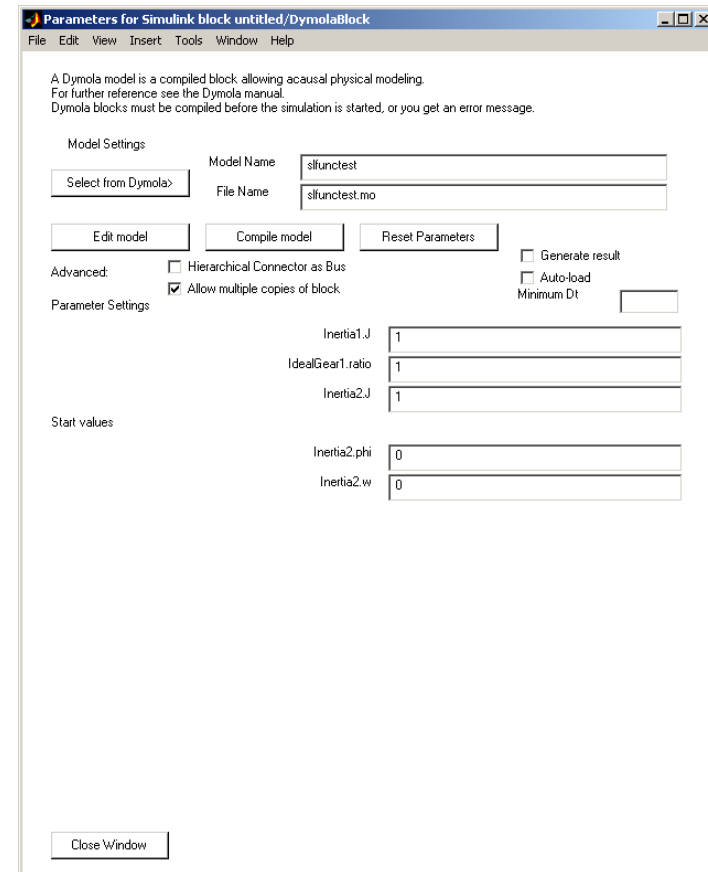
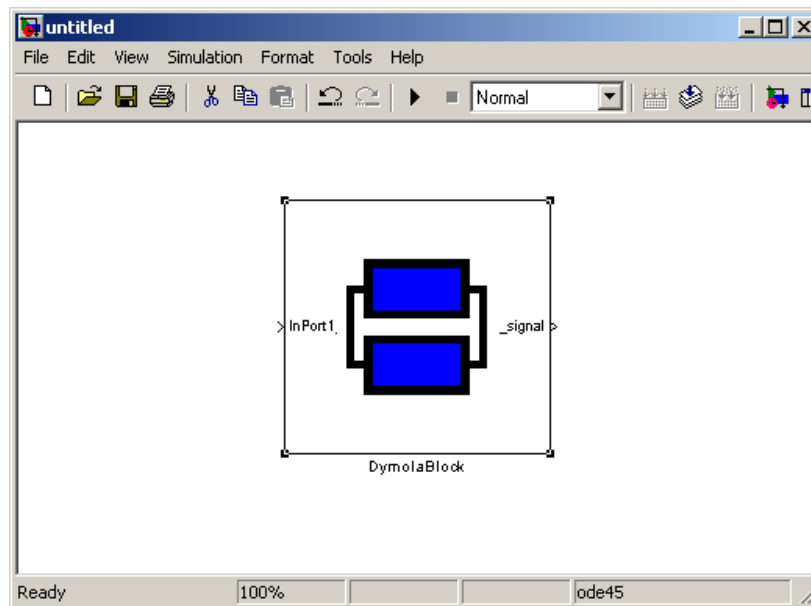
# Dymola-Simulink-Interface

- Dymola-Block in Simulink-Modell einfügen
- Modell übersetzen



# Dymola-Simulink-Interface

- ergibt Block mit Ein- und Ausgängen



## Ausblick auf Dymola 6

- Optimierung
- Kalibrierung von Parametern
- Verbesserte Experimentierumgebung
- Verbesserte Visualisierung
- Benutzereigene Dialoge für Experimente
- Deutlich verbesserte Skriptsprache

Nächste Modelica-Tagung: September 2006 in Wien

# Zusammenfassung

- Modelica soll (de-facto) Sprach-**Standard** werden
- Dymola (Software) 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.com](http://www.dynasim.com)

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