Supporting your vision

Model Based Design and System Simulation with SimulationX and Tool Integration with optiSLang

Dr. Andreas Uhlig
Uwe Grätz

ITI GmbH

Supporting your vision

Who we are · Where we are

- Multi-faceted high-technology
- Leader in virtual system engineering
- Headquarters located in Dresden downtown

Core Business

- Offering complete solutions for system modeling, simulation, analysis and testing
  - development of simulation software
  - engineering services
  - product distribution
  - product integration

SIMULATION X®
Powered by ITI

Simulation Software

Engineering

Customizing

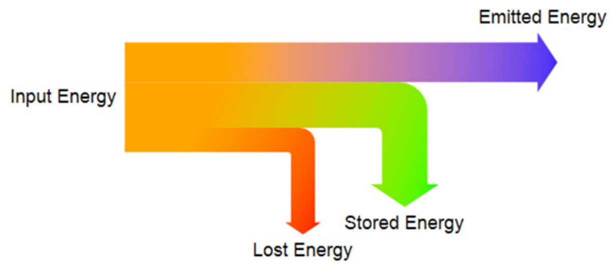
Training & Support

Content

- **Model Based Design**
- Equation Based Modeling
- Design of a Membrane Cylinder – an Example for Sensitivity Analysis
- Interface to optiSLang
- Summary and Outlook

System Simulation / Model Based Design

- New challenges in systems engineering
  - Time, cost,
  - Quality, safety,
  - Energy efficiency
- Dependency of subsystems
- Early assessment of designs
- Virtual prototyping - of system!
- Use models and simulations
- Re-use models

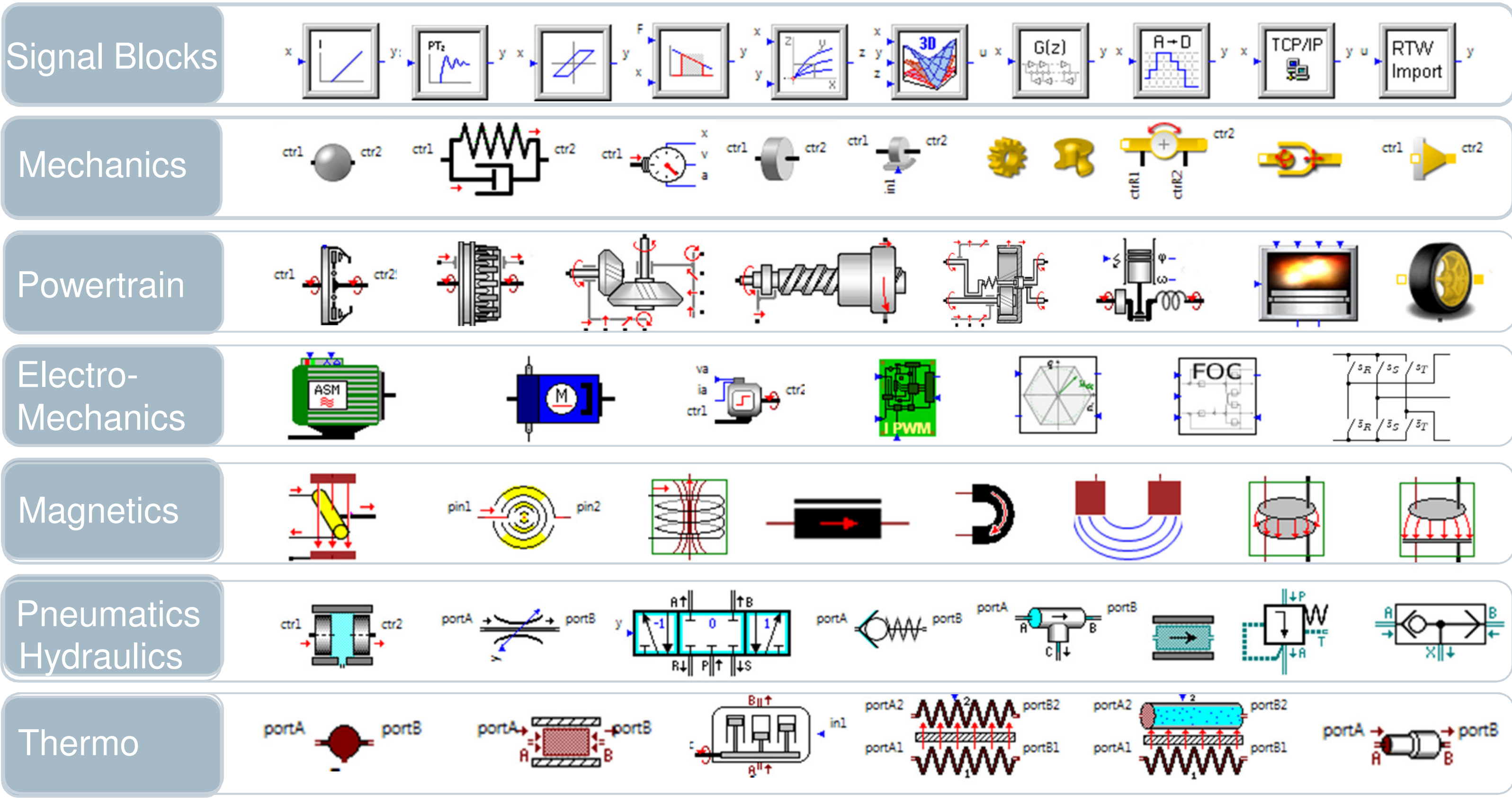


Requirements for Model Based System Design

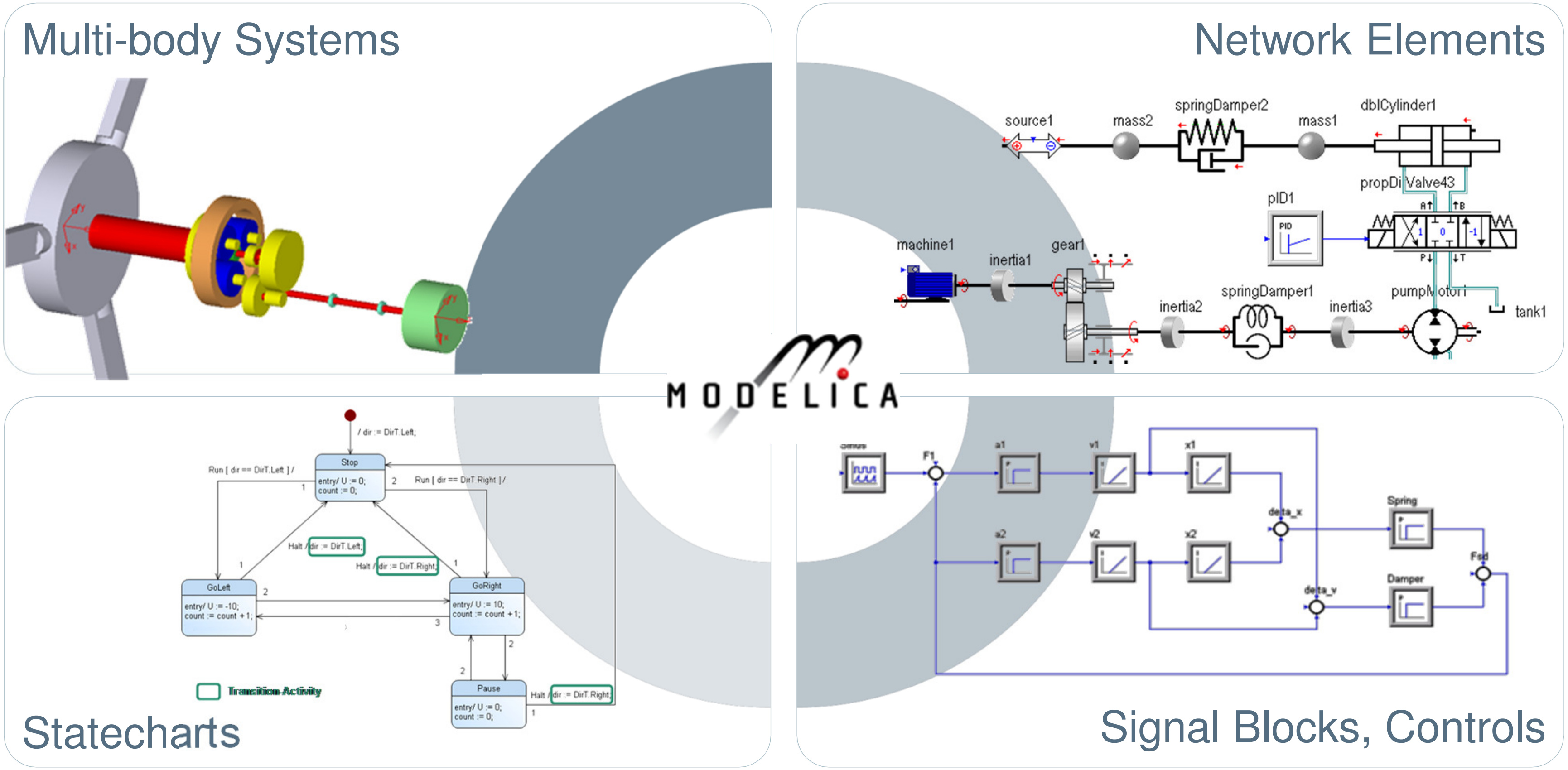
- **Multidomain modeling**
- Hierarchical modeling
- Replaceable models
- One model for multiple analyses

Supporting your vision

Multidomain Modeling (SimulationX Libraries)

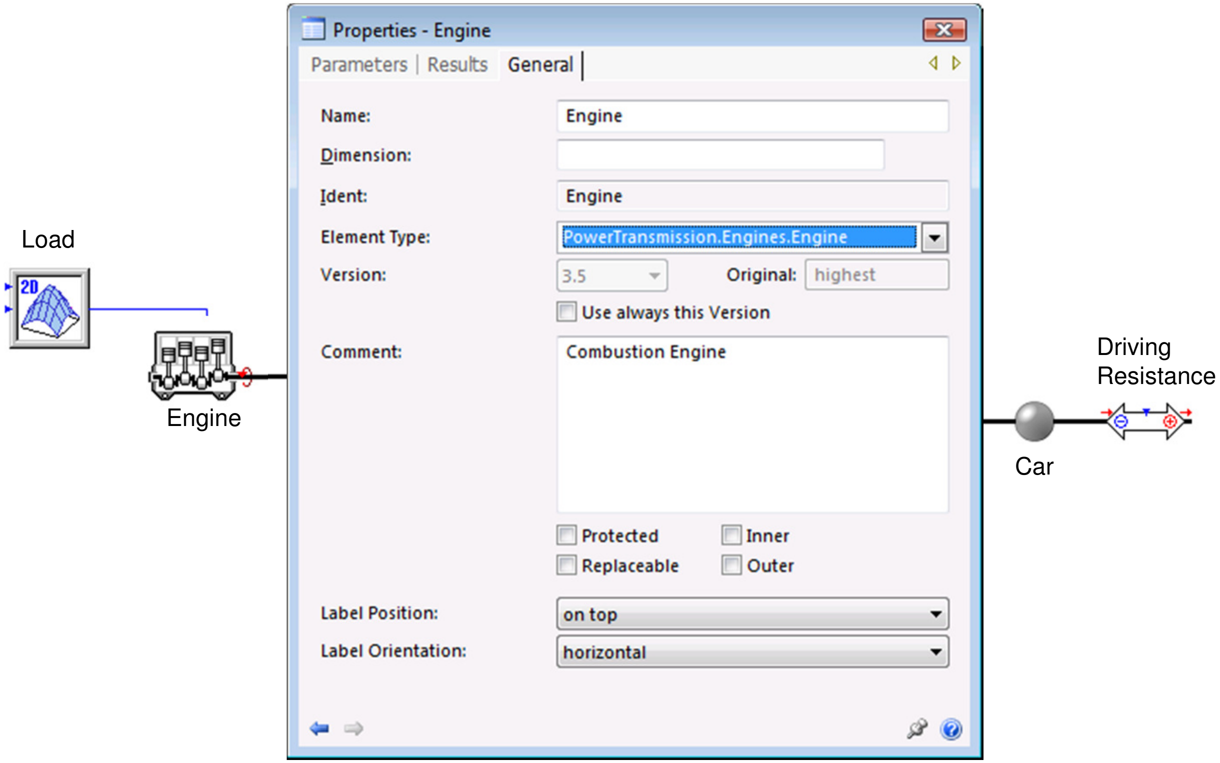


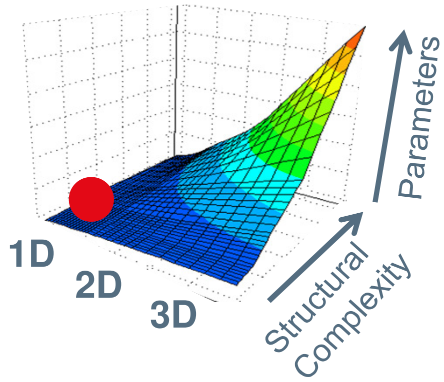
Domain Specific Workspaces for the User



Level 1 – Basic Model

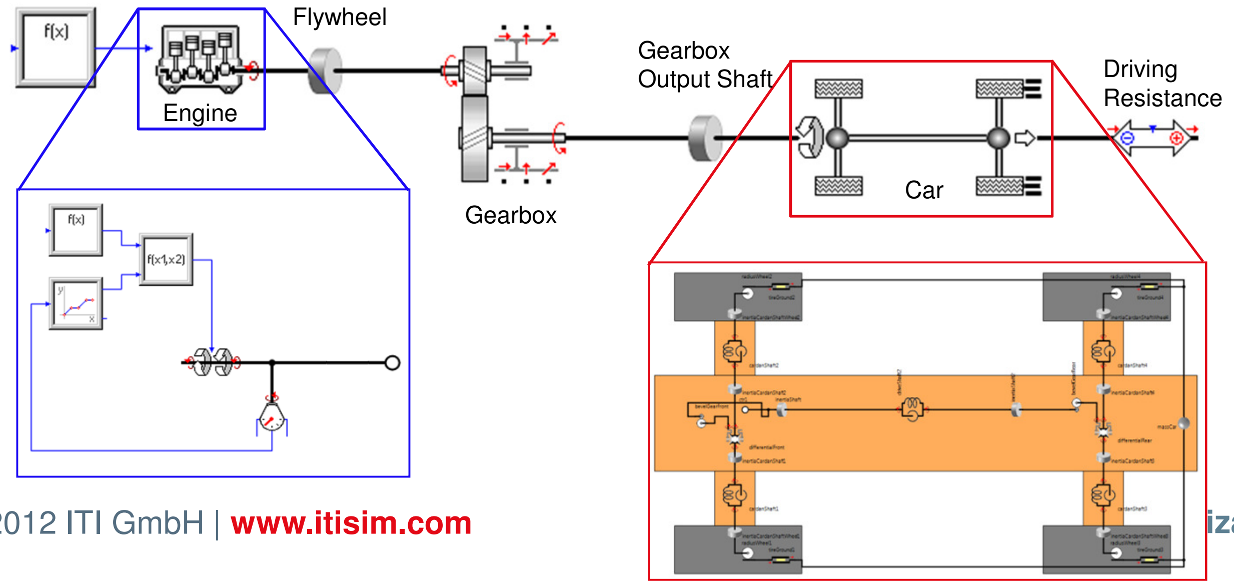
- Simple rigid powertrain model including main components
- Analysis of feasibility and general physical relations

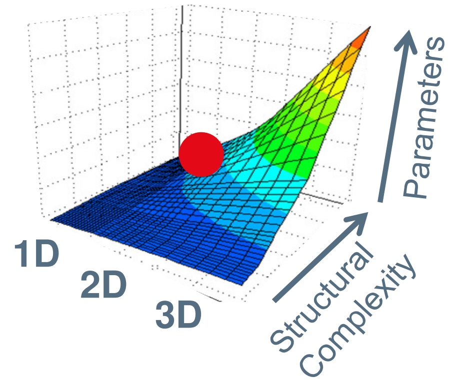




Level 2 – Detailed Model

- Detailing of interesting elements
- Regard of elastic forces
- Simple engine model
- Merge elements in compounds

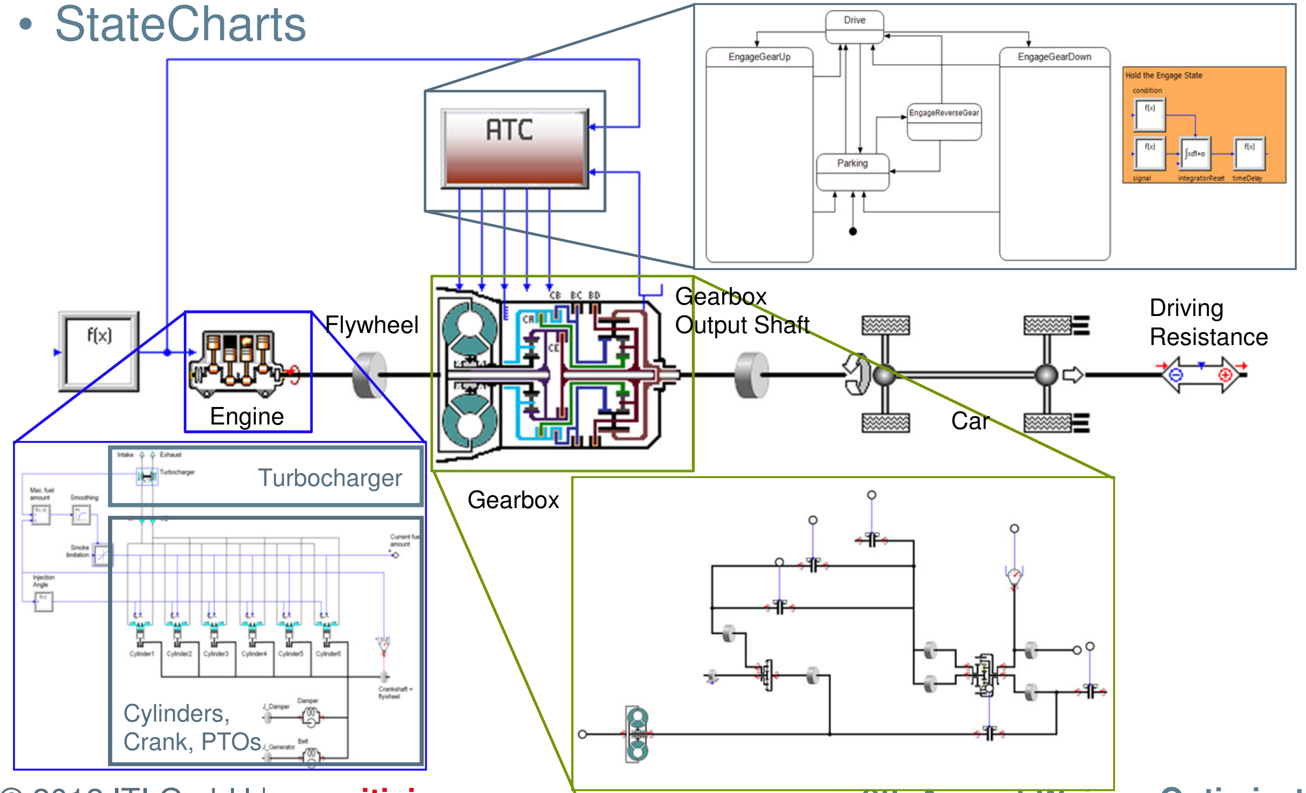


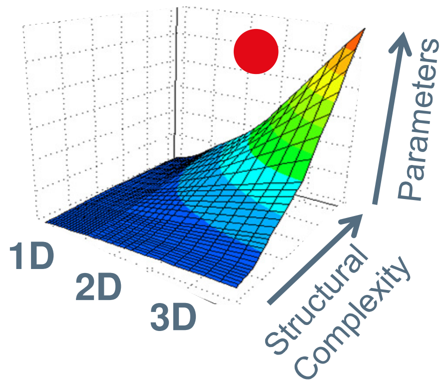


- DoF: < 10
- Parameter: 20…50
- Frequency: < 100 Hz

Level 3 – Complex Model

- Multi-domain compounds
- Detailed engine & gearbox model
- StateCharts



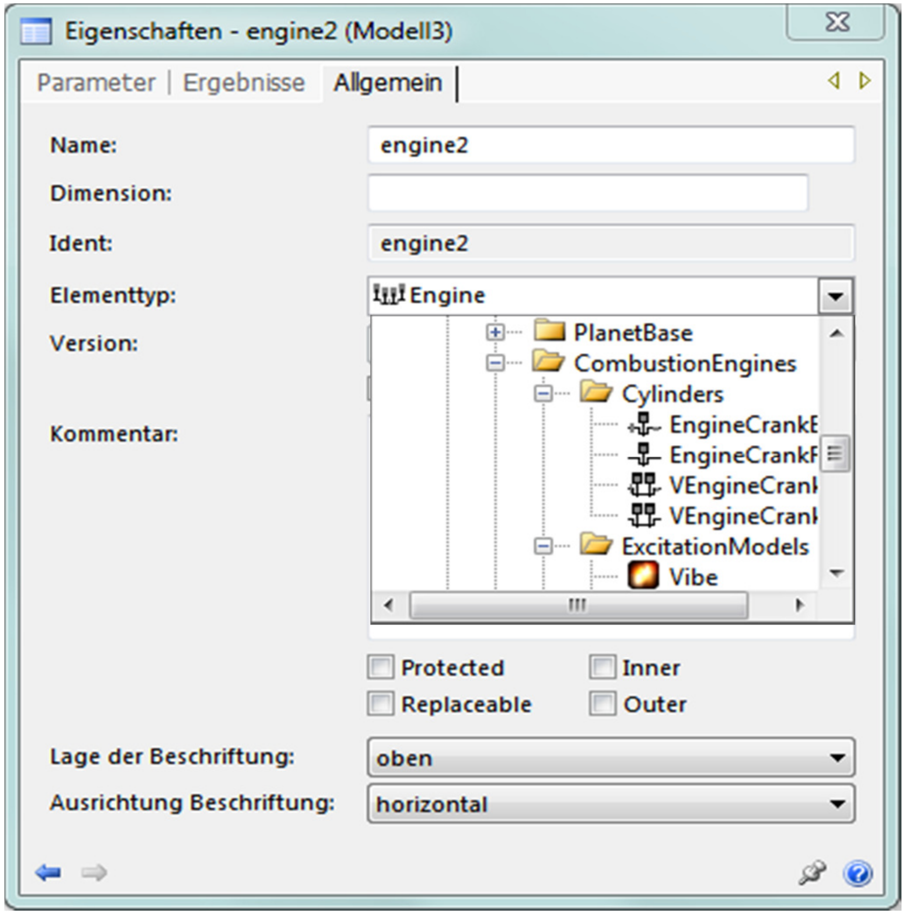


- DoF: > 10
- Parameter: > 50
- Frequency: > 100 Hz

Replaceable Models

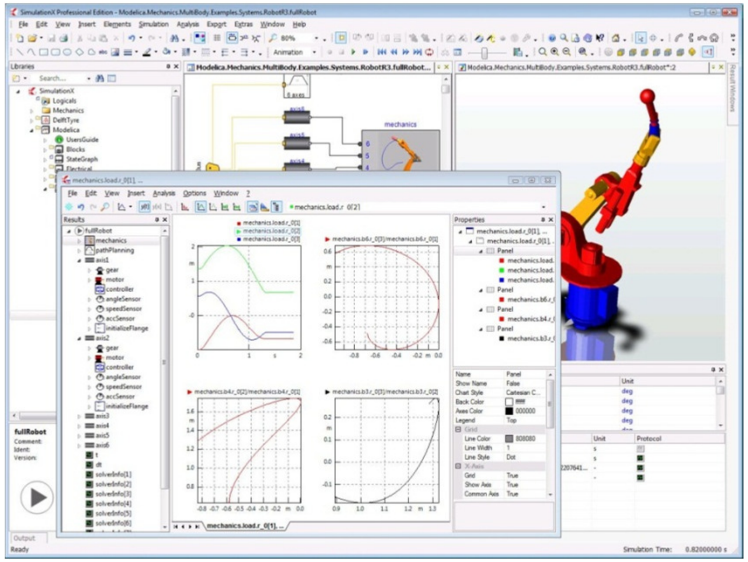
Easily switch
to compatible type

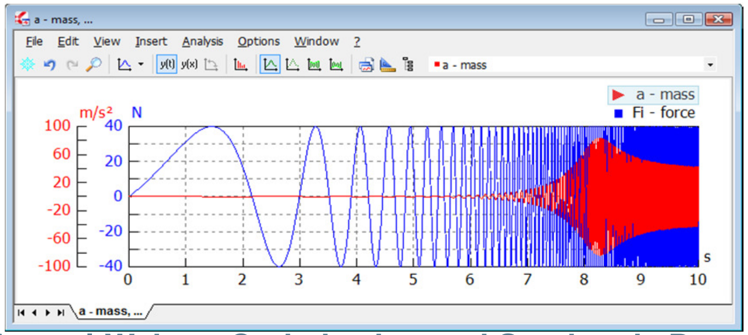
→ Efficient modeling



One model for multiple analyses

- **Transient** in time domain
- Static equilibrium (DC analyses)
- Stationary simulation (non-linear, frequency domain)
- **Linear system analysis** of the entire system:
    - Eigenfrequencies, Eigenmodes
    - Energy analyses
    - Frequency response
    - Poles/Zeros





Equation Based Modeling

- Physical laws are described in text books in terms of formulas like
- In programs this is implemented dependent on the goal of the simulation

$$F = m\,a$$

Equation

$$F := m\,a$$

or

$$a := F / m$$

or

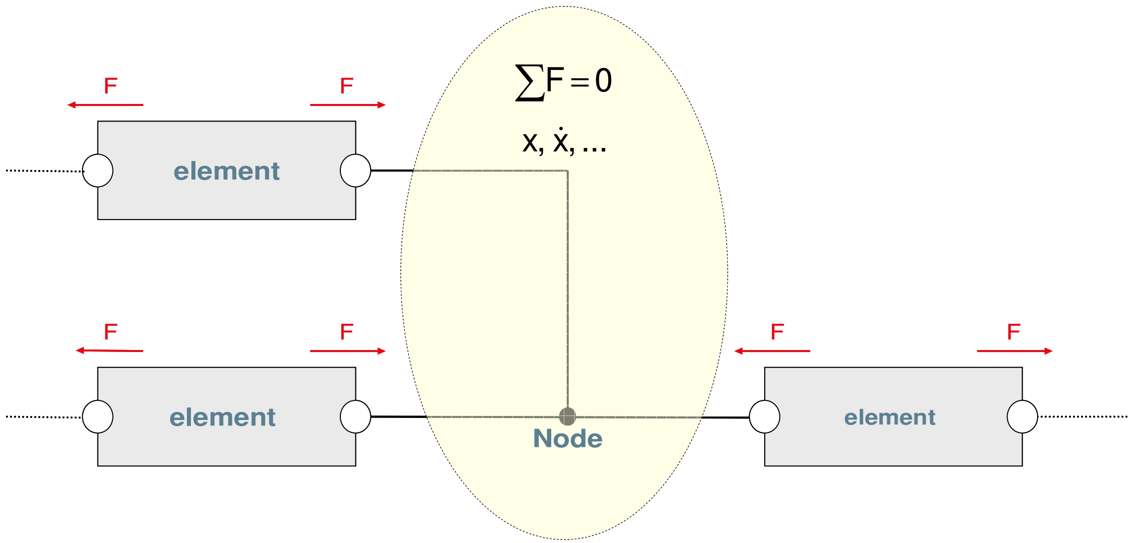
$$m := F / a$$

Assigments

Modeling Concept – Lumped (Network) Elements

- Definition of **potential** and **flow** quantities for each physical domain
- Models consist of ***Elements*** and ***Connections***

- ***Connections***
  calculate potential quantities and define conservation equations
  (e.g. ΣF = 0
  for mechanical nodes)



- ***Elements***
  define relations between
  flow- und potential variables within elements
  (e.g. F = k * Δx in a mechanical spring model)

Supporting your vision

Analogies in Physical Domains

	SOURCES		BASIC ELEMENT TYPES		
	Potential Quantity	Flow Quantity	Capacitive Element	Inductive Element	Dissipative Element
Electronics	Voltage V	Current I	Capacitor $I = C \cdot \dot{U}$	Inductor $I = \frac{1}{L} \cdot \int U\,dt$	Resistor $I = \frac{U}{R}$
Mechanics translational	Velocity v	Force F	Mass $F = m \cdot \dot{v}$	Spring $F = c \cdot \int v\,dt$	Damper $F = d \cdot v$
Mechanics rotational	Angular Velocity ω	Torque T	Inertia $T = J \cdot \dot{\omega}$	Rotational Spring $T = c \cdot \int \omega\,dt$	Rotational Damper $T = d \cdot \omega$
Hydraulics	Pressure p	Volume Flow Q	Volume $Q = V \cdot \dot{p}$	Line (without loss) $Q = \frac{1}{L_H} \cdot \int p\,dt$	Throttle $Q = \frac{p}{R_H}$
Thermics	Temperature T	Heat Flow P	Heat Capacity $P = C \cdot \dot{T}$	-	Heat Resistance $P = \frac{R_{th}}{T}$

Modelica Language

- Allows **Acausal** and causal modeling
- Distinguishes **Through** and **Across** variables
- **Multi-domain** modeling
- Variables carry **Attributes** like units, documentation, min value, max value and nominal value
- Modeling based on a modeling language **Standard**

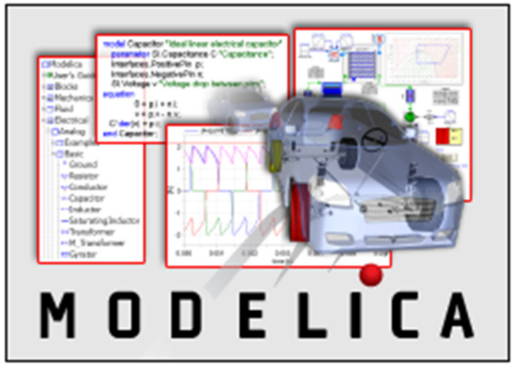
Modelica is a registered trademark of Modelica Association

Who is Modelica?

- Standardized by Modelica Association
- Formed in September 1996
- An independent, international, non-profit modeling standards organization
- A clearing house for public domain Modelica libraries and documentation
- Organizer of Modelica design meetings and conferences
- www.modelica.org
- SimulationX is based on Modelica

MODELICA

Modelica and the Modelica Association



Modelica® is a non-proprietary, object -oriented, equation based language to conveniently model complex physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents. See also,

Modelica is a registered trademark of Modelica Association

Mathematical Models represented by Modelica

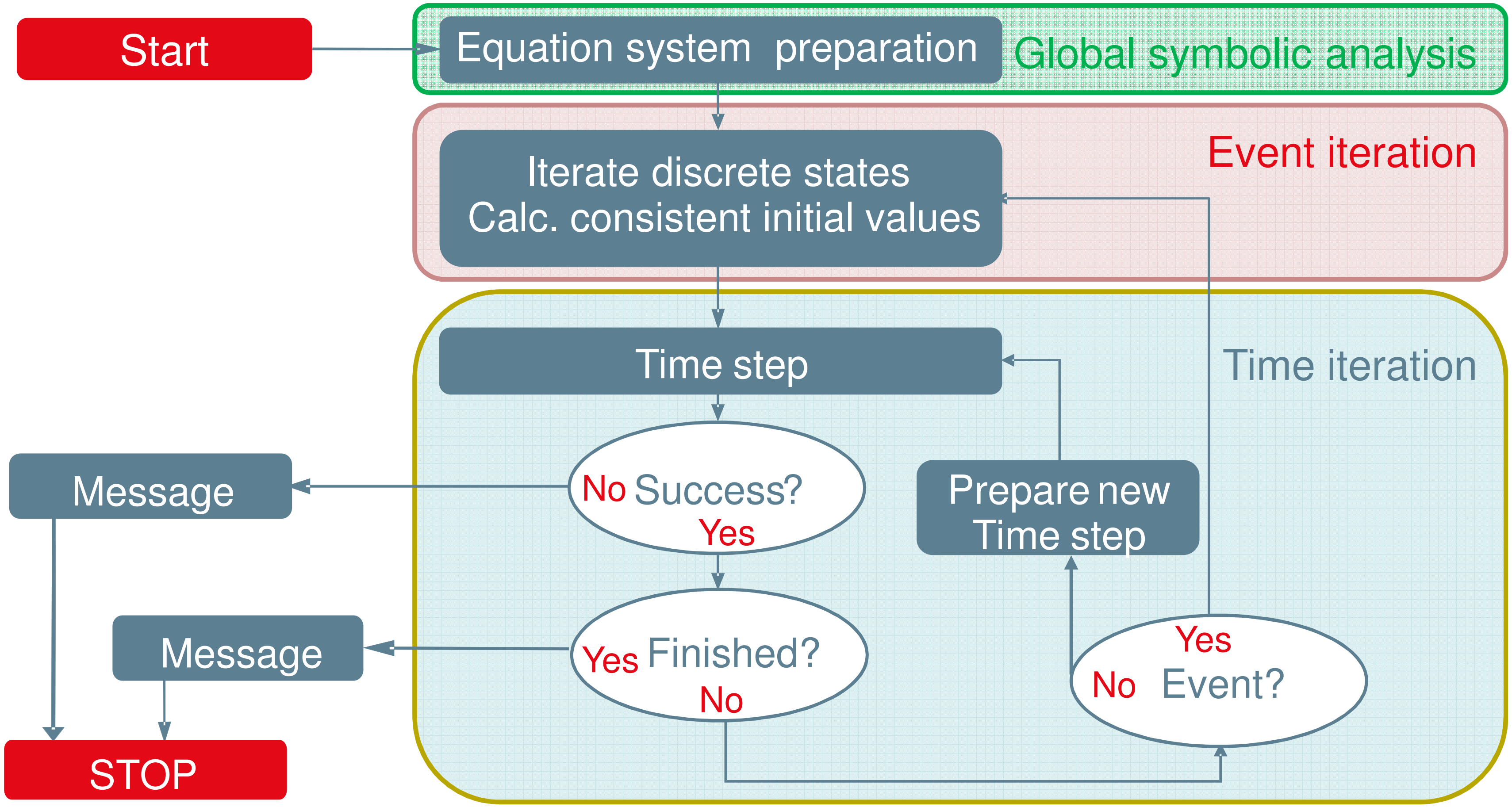
- Differential Algebraic Equation (DAE)

$$0 = f(x, \dot{x}, p, d, r, t)$$

- DAE (semi-explizit)

$$\dot{x} = f(x, y, p, d, r, t)$$
$$0 = f(x, y, p, d, r, t)$$

x… continuous states
y… algebraic variables
p… parameters
d… discrete variable
R…root functions
t… time


Start
Equation system preparation
Global symbolic analysis
Event iteration
Iterate discrete states
Calc. consistent initial values
Time iteration
Time step
Message
No Success?
Yes
Prepare new
Time step
Message
Yes Finished?
No
Yes
No Event?
STOP

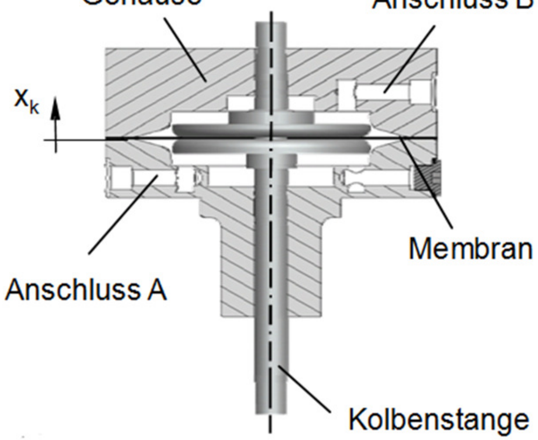
Symbolic Equation Handling

SIMULATION X®
Powered by ITI

- No need for the user to decide which variables to solve for
- Reduces the number of different components in a library
- Redundant states can be eliminated in models
- Parts of models can be solved symbolically rather than numerically
- Symbolic initialization of model variables
- Increased performance of the simulation is gained at translation time

Pneumatic actuation of testing machines

- Steady state tests and oscillating loads with higher frequency
- Membrane cylinder:
  - Low friction losses
  - Low sticking forces
  - Low tolerance required -> low cost
  - Small stroke
- Application with membrane cylinders
  - Fatigue tests up to 50Hz and ±0,5 mm
  - Testing machines (e.g. dental implants)



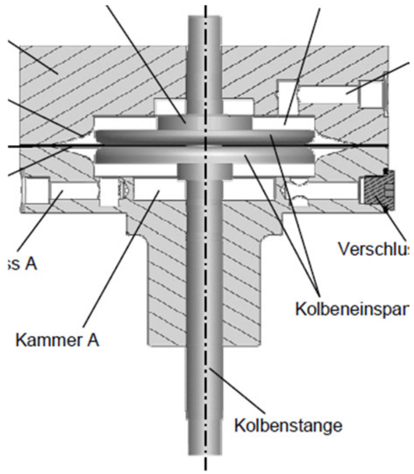
Zahnimplantat

Quelle: QZM e.V.

Fiedler, M.: Modellbildung und numerische Optimierung am Beispiel eines servopneumatischen Membranzylinderantriebs, TU Dresden, Dissertation 2010

Supporting your vision

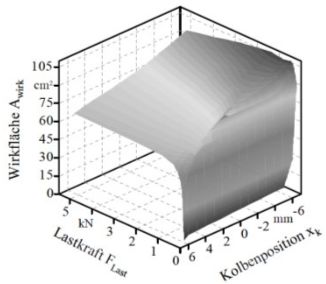
Development Objectives

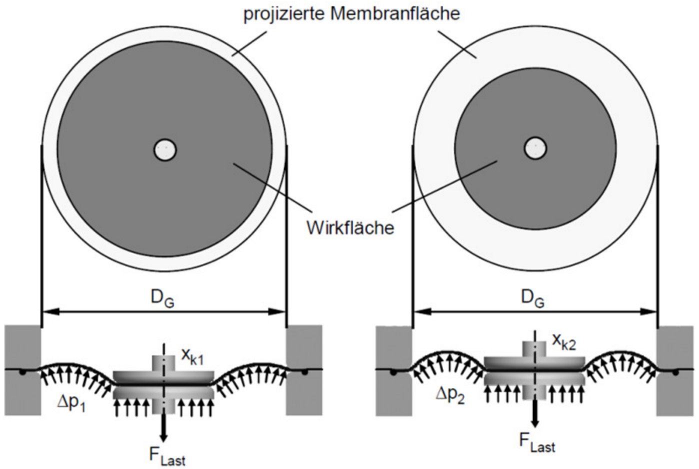


- Enlargement of amplitude of the membrane cylinder wrt. the whole drive concept
- Modification of the cylinder design – especially of the inner geometry
- Decreasing package dimensions, too
- Complex relation between stroke *x*, pressure *p*, chamber volume *V*, force *F*

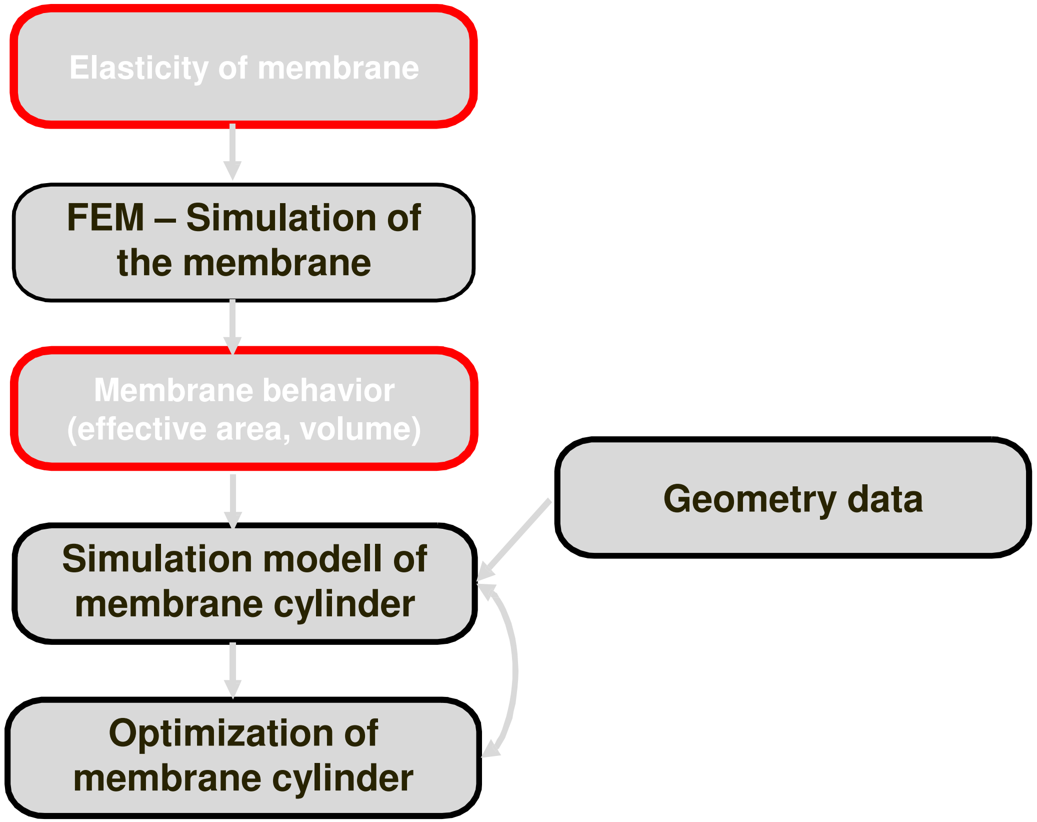
$F = \mathrm{f}(p, A)$

$A = \mathrm{f}(p, x)$





Workflow of optimization prozess



Supporting your vision

Simulation Model

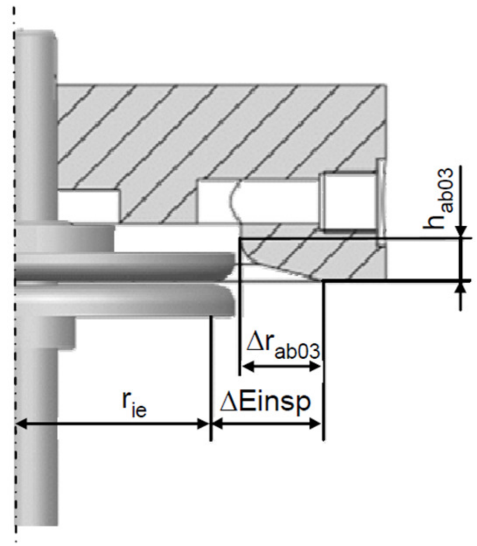
- Definition of **limits** of considered **design parameters**
- Control of the cylinder by a **pneumatic valve**
- Calculation of the influence of **design parameters** to the steady state membrane behavior

$33{,}5 \leq r_{ie} \leq 44{,}5$

$7{,}7 \leq h_{ab03} \leq 10{,}7$

$15{,}5 \leq \Delta Einsp \leq 35{,}5$

$2{,}5 \leq \Delta r_{ab03} \leq 20$

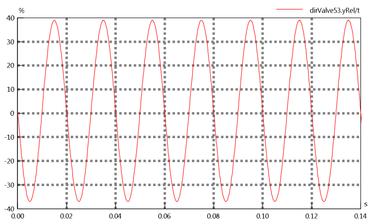


Simulation Model

Membrane Cylinder - Membrane behavoir
- no friction and damping
- Implementation of maps for effective area and chamber volumes

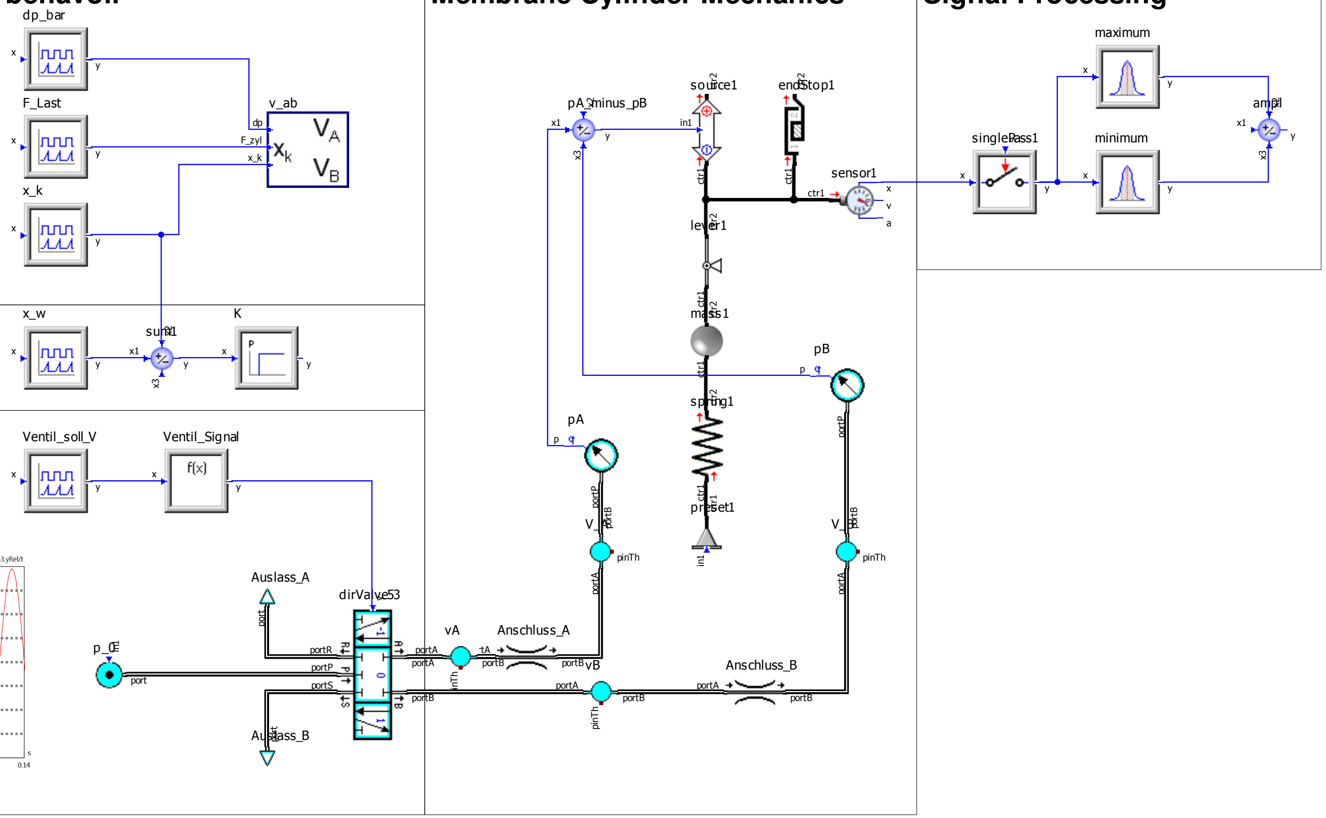
P-Controler
(only option, for valve control)

Ideal Valve Behavior
- no dynamics
- flow like VP60 1200 l/min



Membrane Cylinder Mechanics

Signal Processing



Simulation Model for Optimization

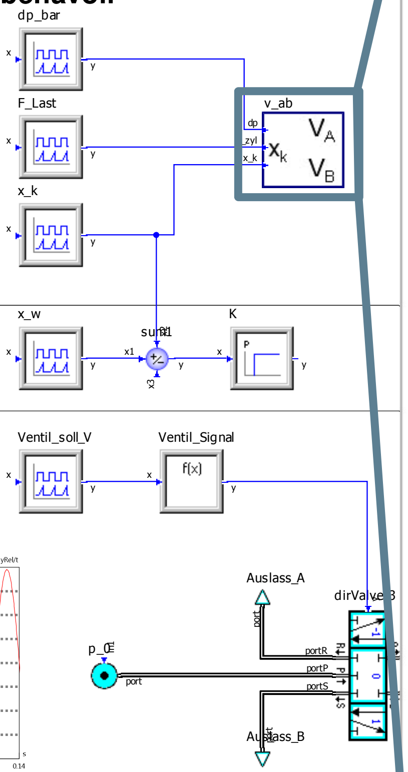
Membrane Cylinder - Membrane behavoir

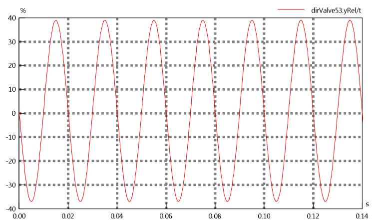
- no friction and damping
- Implementation of maps for effective area and chamber volumes

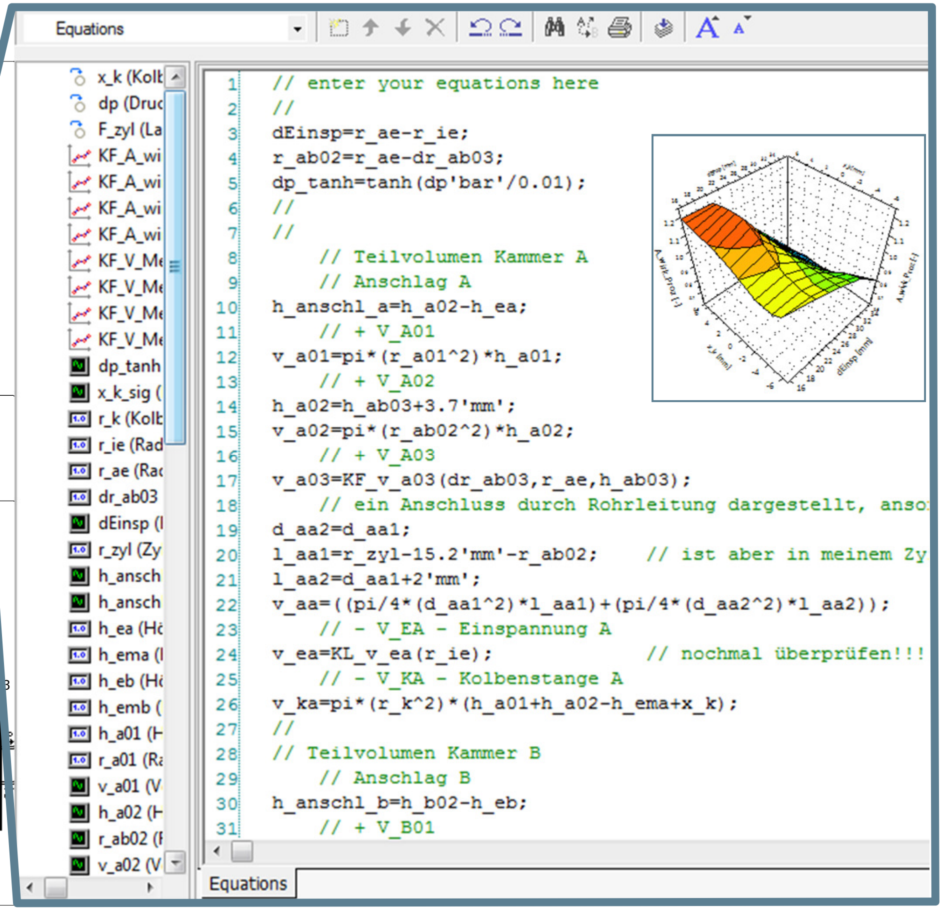
P-Controler
(only option, for valve control)

Ideal Valve Behavior

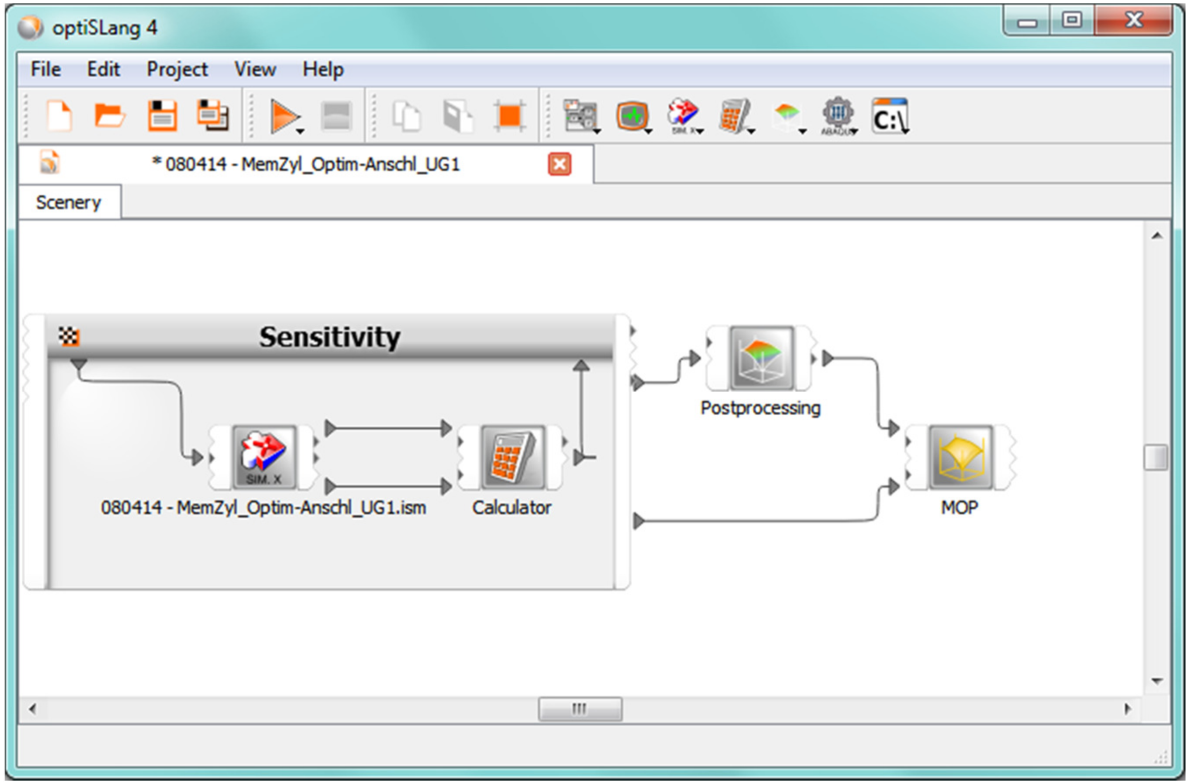
- no dynamics
- flow like VP60 1200 l/min



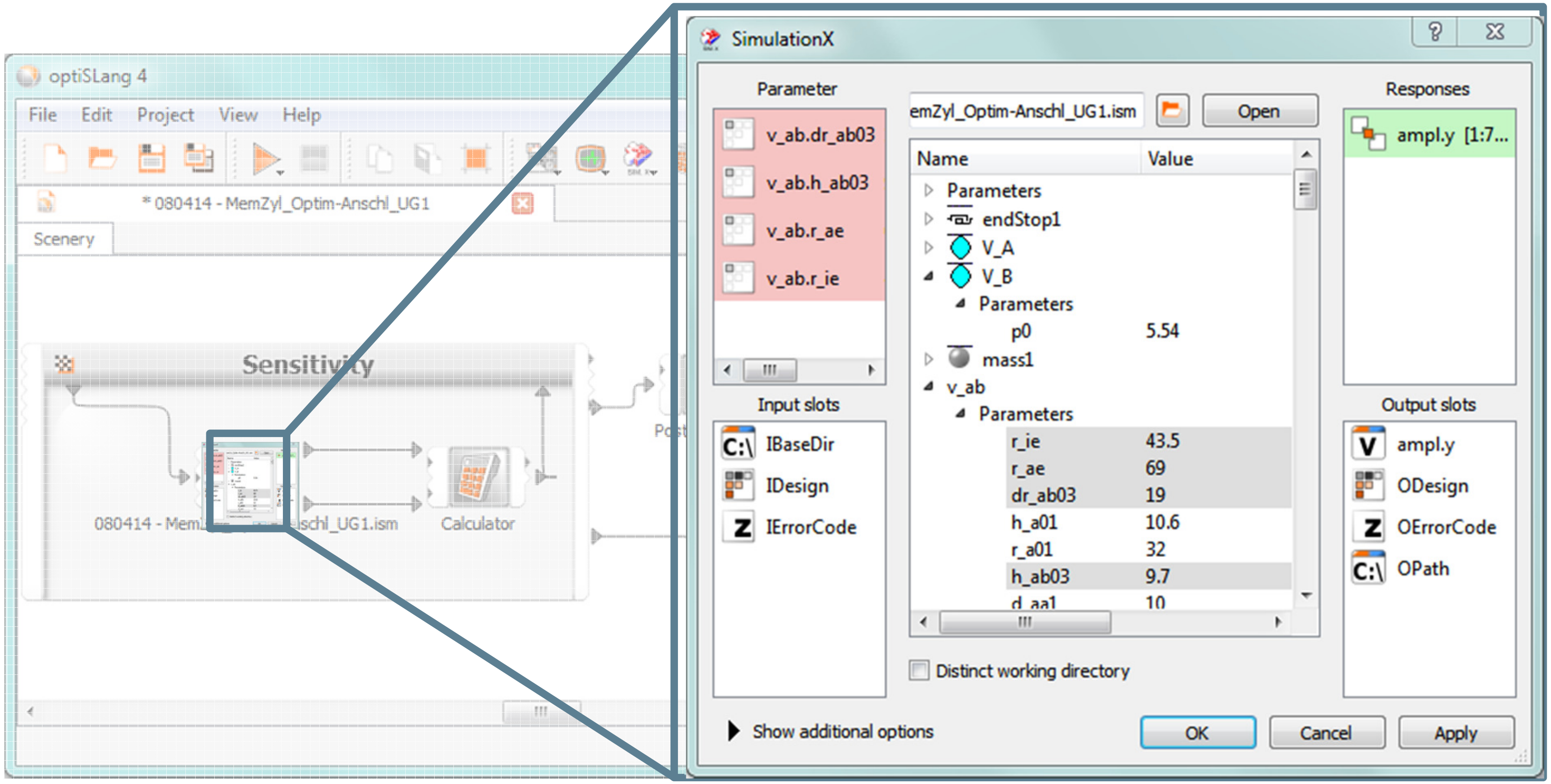




Optimization with optiSLang



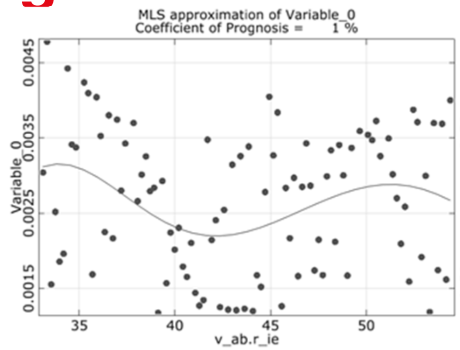
Optimization with optiSLang

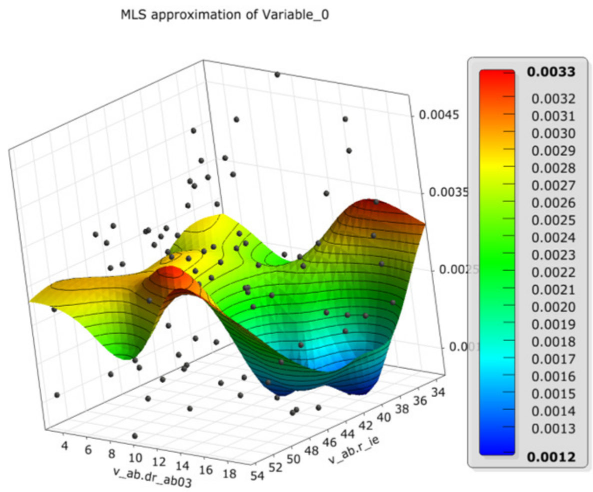


Sensitivity Analysis with optiSLang

First Attempt:

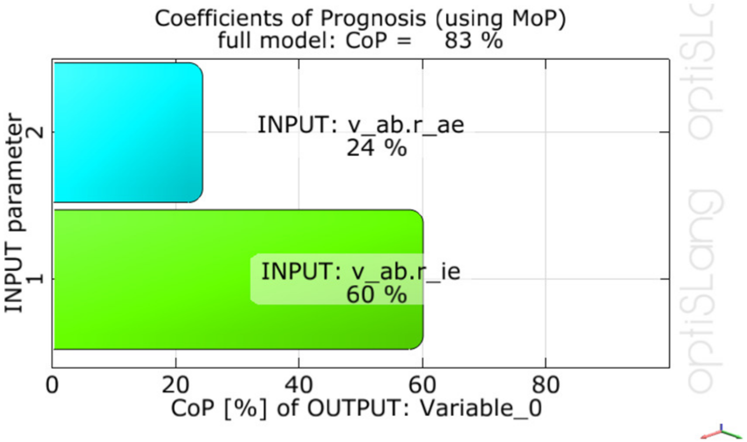
- 4 Parameters included
- Sensitivity Analyses executed
- CoP 1% → results not usable
- …
- Check model again
- Include more parameters
- …

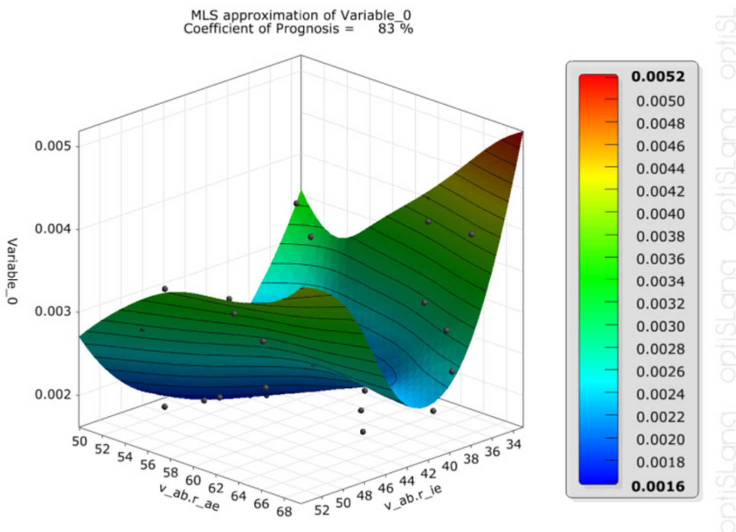




Sensitivity Analysis with optiSLang

- 6 Parameters included
- Sensitivity Analyses executed
- CoP 83%





- Continue with optimization….

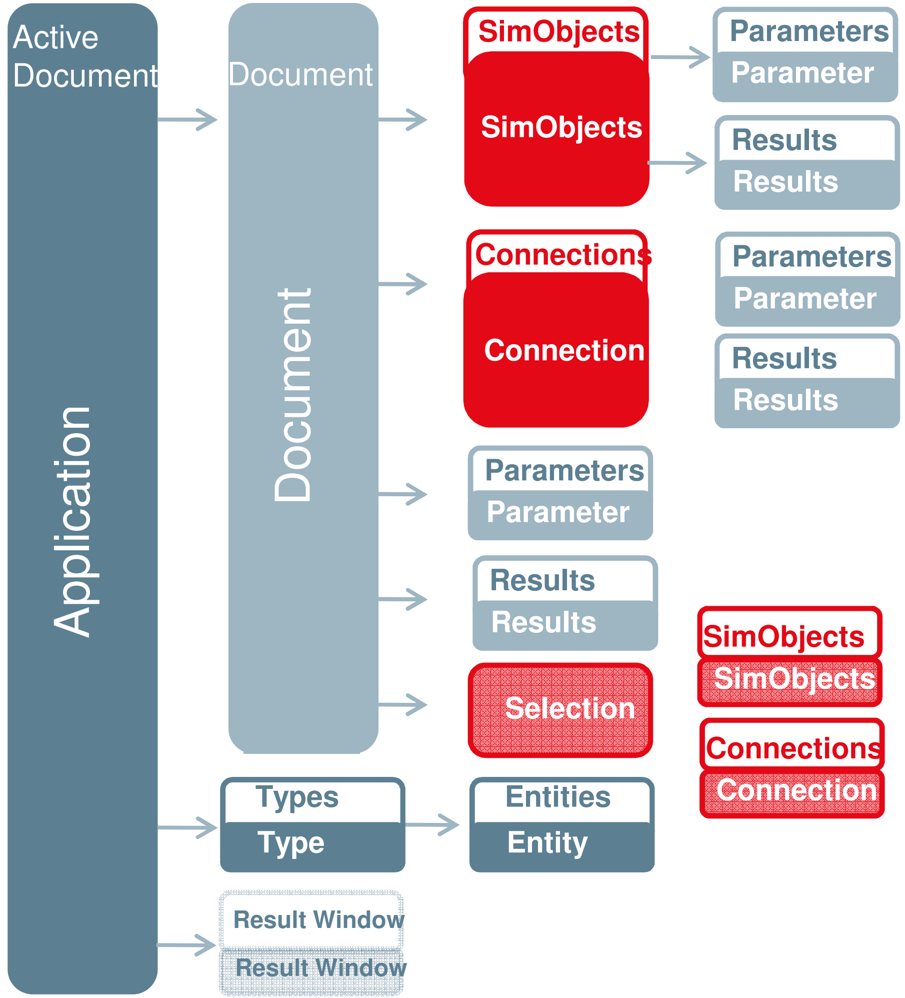
Benefit of Integration

- **optiSLang 4** offers a range of **direct integration nodes**.
- In the case of SimulationX, the direct interface allows an **easy and user-friendly parameter and response definition**.
- In the **optimization or calibration analysis**, the specified properties are modified directly in the SimulationX model according to the defined ranges and the response values calculated for each design.
- Using the **SimulationX API** (COM based), the SimulationX model components, including their properties, can *be directly accessed* in the parametrization process of optiSLang.

(Component) Object Model

Objects & Collections

- Application
- Simulation models
- Model components
- Parameters
- Result variables
- Element Types
- Result windows
- Selection



Summary

- Model based design is core of future systems engineering
- Use noncausal models for different tasks (Modelica)
- SimulationX as powerful platform for systems engineering
- A pneumatic application used as test case for sensitivity analysis
- Productiv optimization workflow with optiSLang 4

Next Steps

→ Further increase efficiency

→ Jump start guide (SimulationX + optiSLang)

Supporting your vision

ITI GmbH

Headquarters
Webergasse 1 · Haus C2
01067 Dresden · Germany

T + 49 (0) 351.260 50 - 0

www.itisim.com