

## Fact Sheet

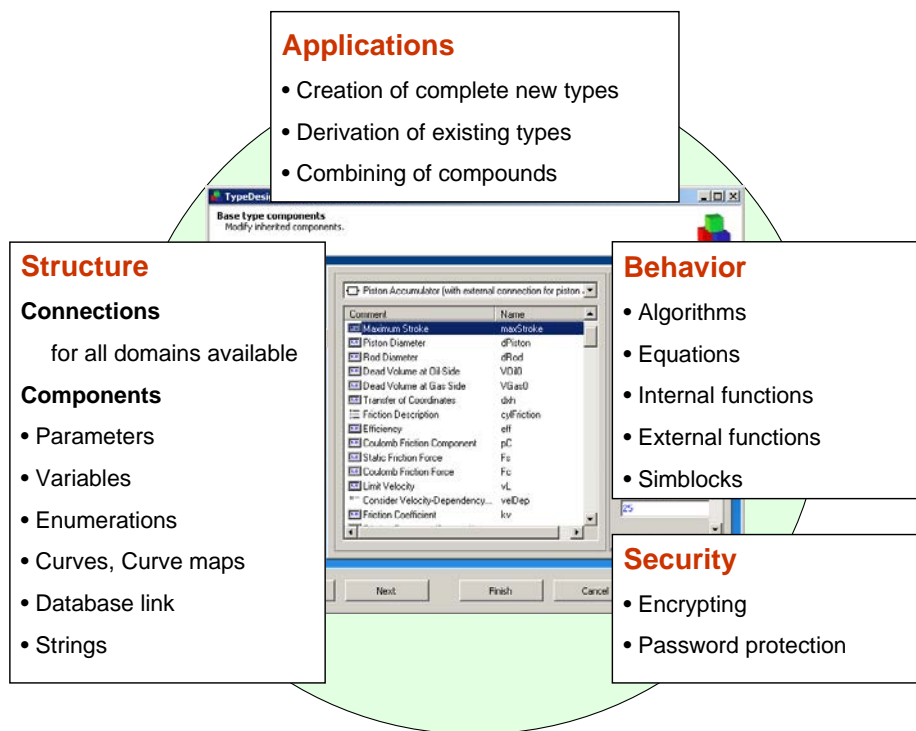
### TypeDesigner

The object oriented modeling concept of SimulationX uses element types for description of structure and behavior of model elements. An element is an instance of the element type and defines the values of the components in the structure.

The TypeDesigner is a tool for creation and modification of element types. A wizard guides you through the complete process of type development.

You can:

- design completely new types
- extend existing types by inheritance
- create hierarchical compound types (submodels) by merging elements of a model



- Wizard guides through creation and modification of types.

- All types and kinds of components are useable in user defined types.

- Extension and customization available by inheritance and building of compounds.

- Functions and SimBlocks define reusable calculation modules.

- Replaceable base types for several implementations of one type.

- Activity groups allow creating dynamic user interfaces.

#### Structure

All kinds of connections and components of SimulationX are available for the development of your element type:

- Connectors of all available domains
- Components (parameters, variables, curves, enumerations ...)
- Calculation modules (functions, SimBlocks)

#### Behavior

The behavior is described by algorithm-and/or equation sections. They describe the relationships between the components and connections. Algorithms and equations are noted in the object oriented model description language Modelica®. Array- and matrix operations are supported.

- Password protection for editing and usage.

- Integrated HTML-documentation page.

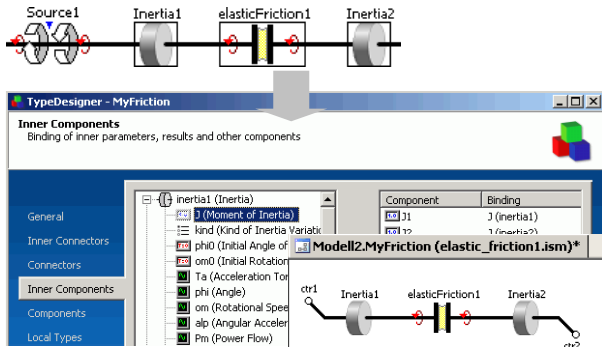
The TypeDesigner allows creating reusable calculation modules for usage the behavior description. Calculation modules have no user interface, but only input- and output parameters. They can be reused in any element type. The following kinds of calculation module are available.

- *Internal functions* – Calculation of output parameters is done by one algorithm section
- *External functions* – Calculation of output parameters is done by a function call in a DLL
- *SimBlocks* – Calculation of output parameters is done by algorithm- and/or equation sections

## Inheritance

The functionality of existing types can be extended and customized by inheritance. The derived element type can add own structure elements and behavior description.

## Compound types



Compound-types are hierarchical element types and are created by merging model parts.

You define which contained components are available in the properties dialog of the compound type and you can add own components and behavior description.

The components of the compound type remains editable in the compound type.

## Modelica Editor

```

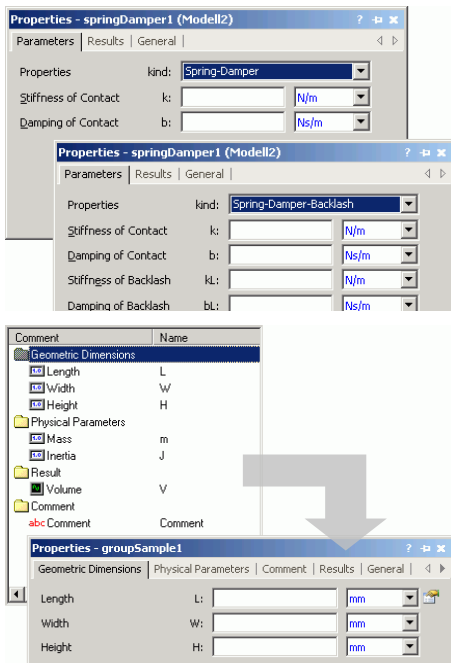
connect(connection2, constThrottle1.portB);
connect(connection2, portA);
algorithm
// Hysteresis-Parameter
eps:=1e-5;
T:=1/(2*pi);
if noEvent(dynamicsON) then
// druckkennlinie als eingang des PT2-gliedes
yRelin:= if ctrl>=1 then 1.0 elseif ctrl<=0 then 0.0 else ctrl;
if (pre(os)==0) ...
elseif (pre(os)==1) then
if ((yRelin)>(1-eps)) then // G*x, ymax=1
os:=1;
else
os:=0;
end if;

```

The Modelica source of a type can be edited directly. The editor supports code folding and syntax highlighting.

All other pages of the TypeDesigner will be automatically updated corresponding to the changes of the Modelica source code.

## Design of user interface



The visibility of components in the property dialog can be controlled by Boolean- and Enumeration components.

This makes the dynamic view of the relevant properties possible.

The components of the type can be arranged on predefined or on user defined property pages.