INTRODUCTION

PROOSIS

– Simple and complex physical processes modeling software.

– EcosimPro Language **EL** to model continuous and discrete processes.

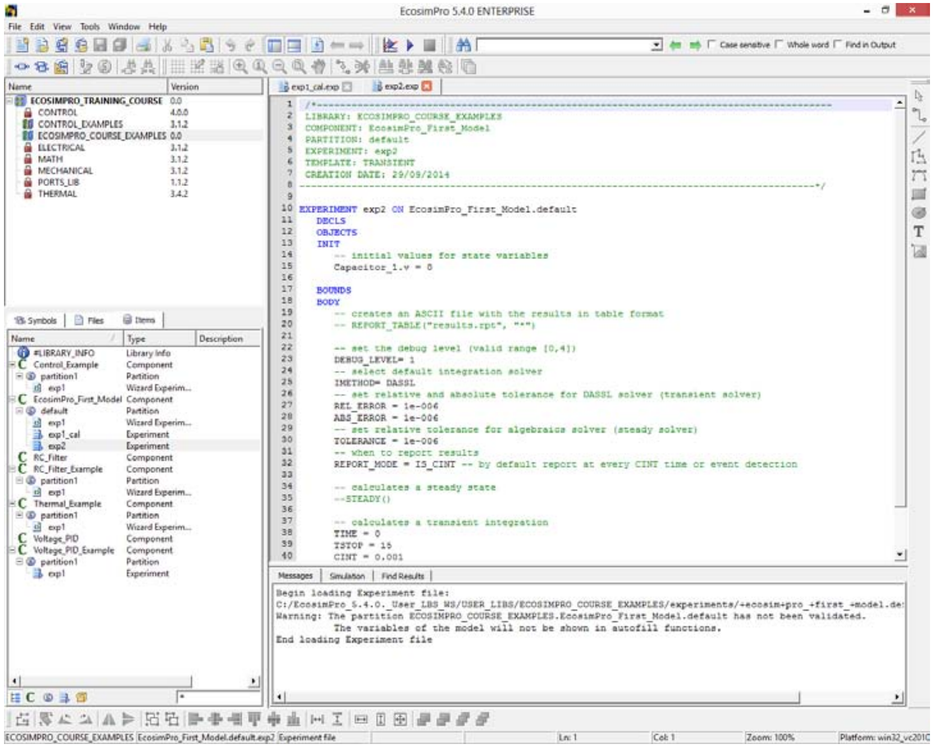
– Differential Algebraic Equations (**DAE**s), Ordinary Differential Equations (**ODE**s) and Discrete Events solver.

– EcosimPro provides a set of reusable **Libraries**: Mechanical, Electrical, Thermal, Mathematical, Control,…

– **Schematic** Modeling: user-friendly environment.

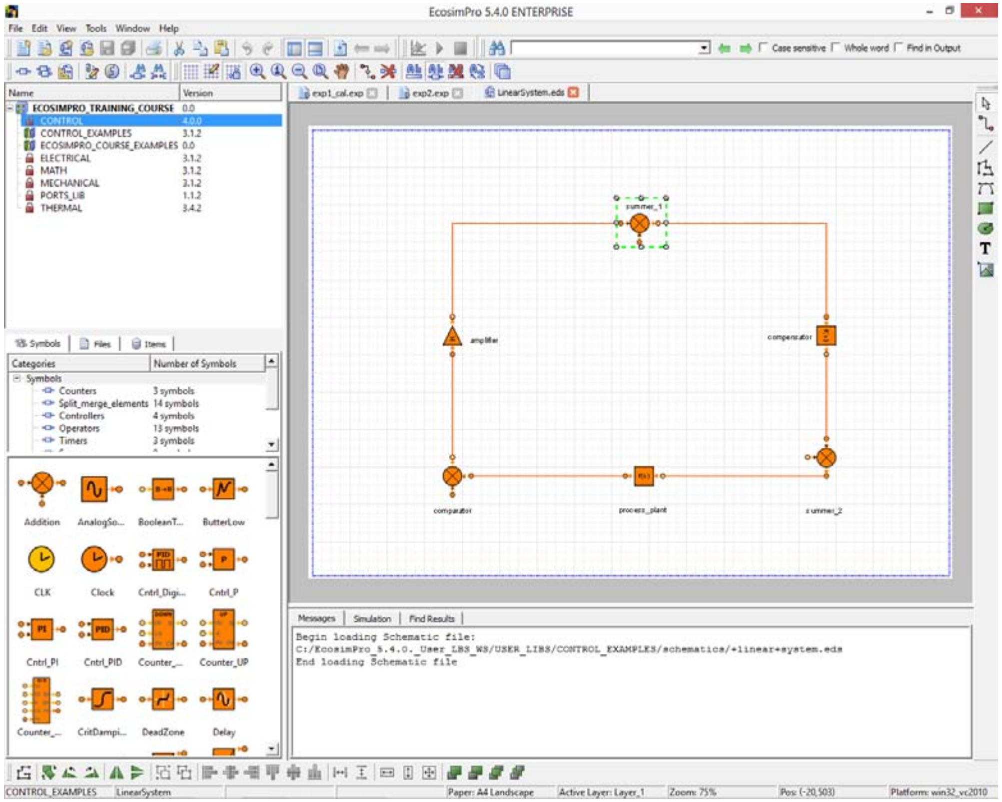
Users

- **Level 1: Users who develop libraries of components.**
  - Model physics knowledge.
  - Model math knowledge.
  - EL code knowledge.



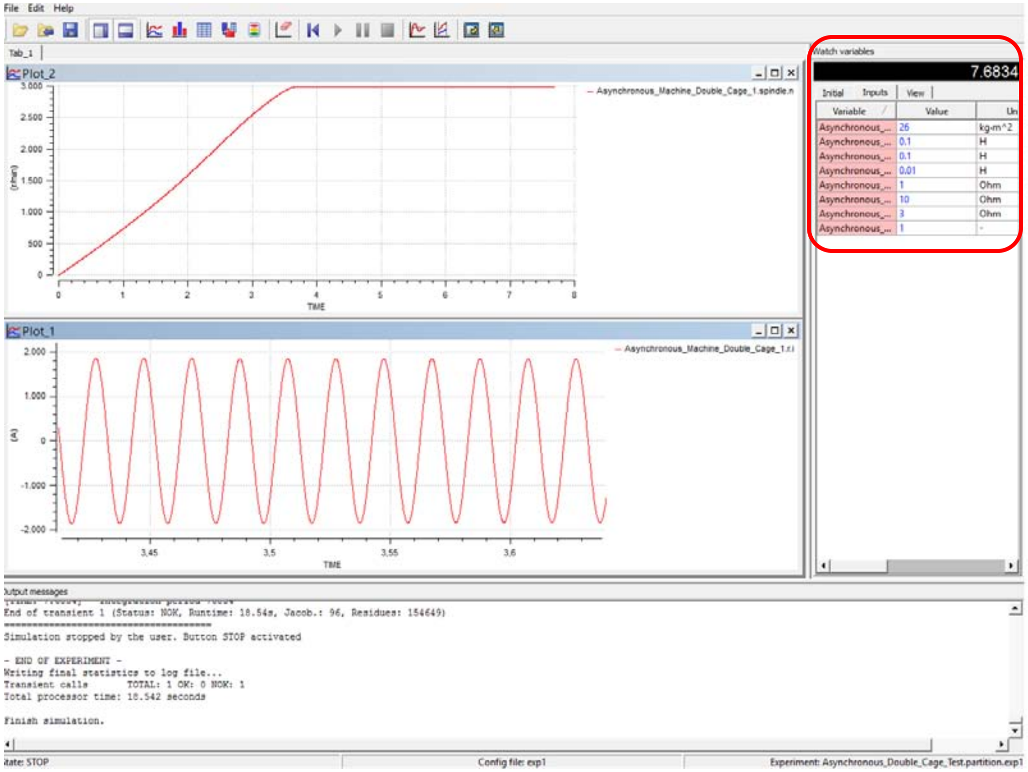
Users

- **Level 2: Users who create models based on existing libraries.**
  - Components deep knowledge.
  - Schematic modelling.



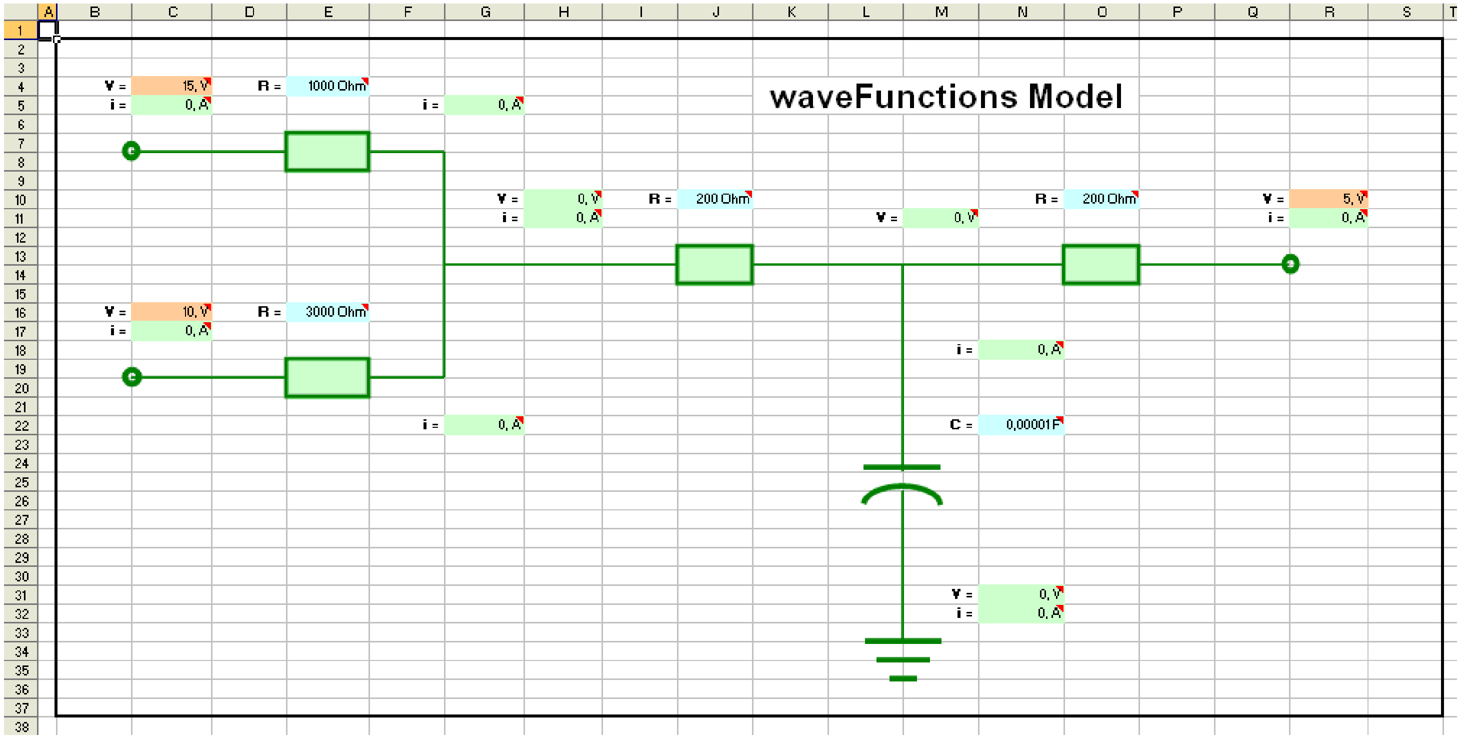
Users

- **Level 3: Users who run simulations from existing models.**
  - Running the simulation changing the input data.
  - Background knowledge of the final application.
  - Mainly by means of the monitor.



Users

- **Level 4: Users who use exported models.**
  - From EcosimPro to other tools (Excel, Matlab, Simulink) .
  - From EcosimPro as black boxes (Decks).

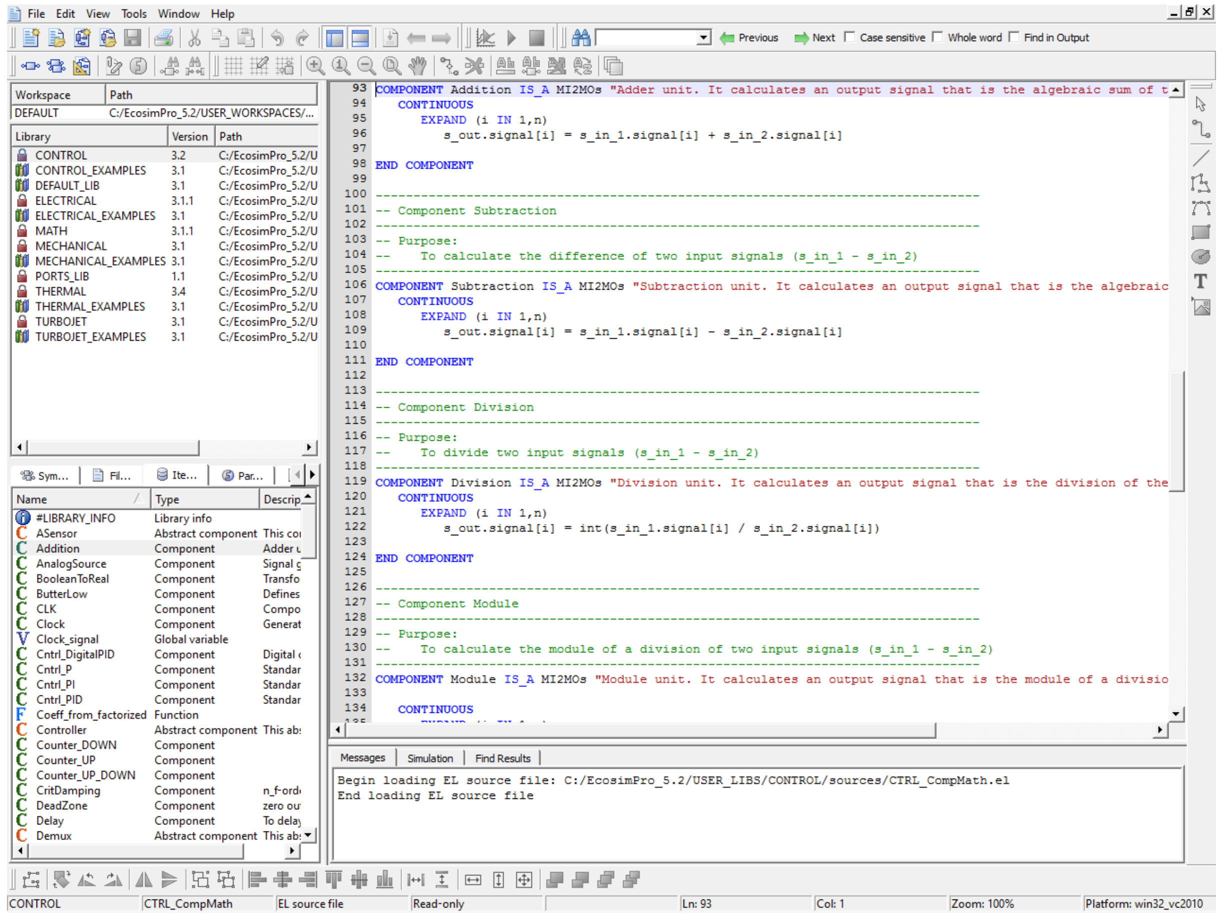


MAIN CONCEPTS

Main Concepts

Global View

- Workspace.
- Library.
- Component.
- Schematic.
- Partition.
- Experiment.
- Monitor.
- Deck.



Main Concepts

WorkSpace

– Set of libraries related to a specific simulation environment and configuration setup.

– Each user may create different workspaces according to the different developing areas or projects.

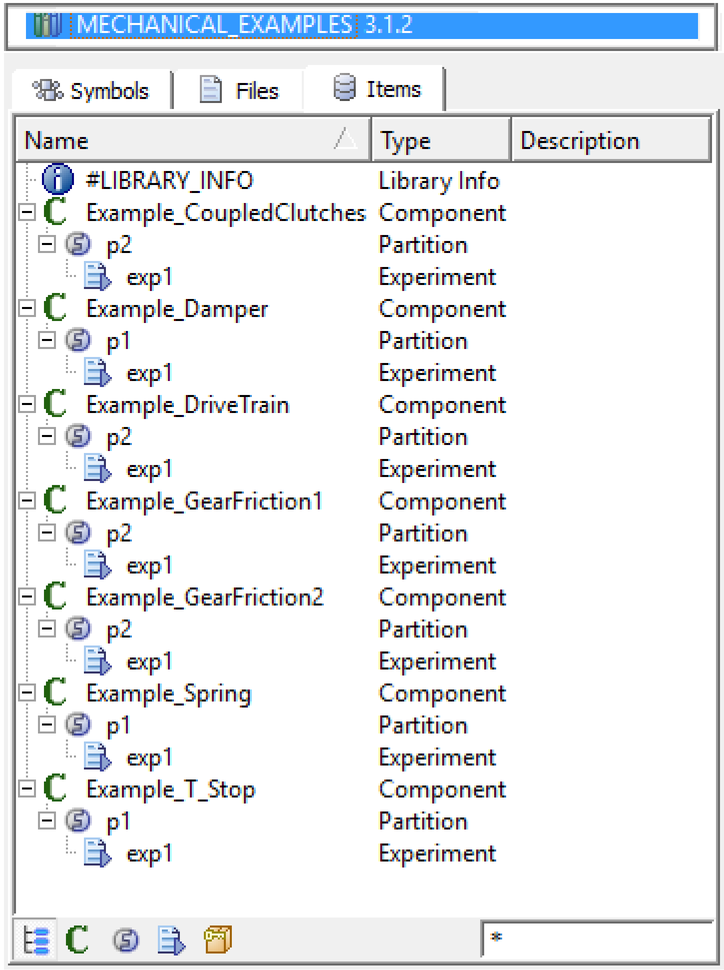
Name	Version
DEFAULT	1.3
CONTROL	4.0.0
CONTROL_EXAMPLES	3.1.2
DEFAULT_LIB	3.1.2
ELECTRICAL	3.1.2
ELECTRICAL_EXAMPLES	3.1.2
MATH	3.1.2
MECHANICAL	3.1.2
MECHANICAL_EXAMPLES	3.1.2
PORTS_LIB	1.1.2
THERMAL	3.4.2
THERMAL_EXAMPLES	3.1.2
TURBOJET	3.1.2
TURBOJET_EXAMPLES	3.1.2

Main Concepts

Library

- Collection of elements related to a specific simulation discipline (Electrical, Mechanical, Thermal, etc).

- Library area: Files, Items, Symbols, Partitions and Experiments related to the library.



Main Concepts

Component

- The Components are the basic elements in EcosimPro.
- They are usually related to a physical element such as compressors, valves, pipes, pumps, ...
- Their formulation describes the behaviour of the related physical element.

Symbols | Files | Items

Name	Type	Description	Pa
C MIMOs	Abstract Co...	This abstrac...	
C MO	Abstract Co...	This abstrac...	
C MathFunction	Component	To perform ...	
E MathOption	Enumerative		
C Maximum	Component	Outputs the ...	
C Minimum	Component	Outputs the ...	
C Module	Component	Module unit...	
C Move	Component	Move stand...	
C Mux	Abstract Co...	This abstrac...	
C Mux2	Component	Multiplexer ...	

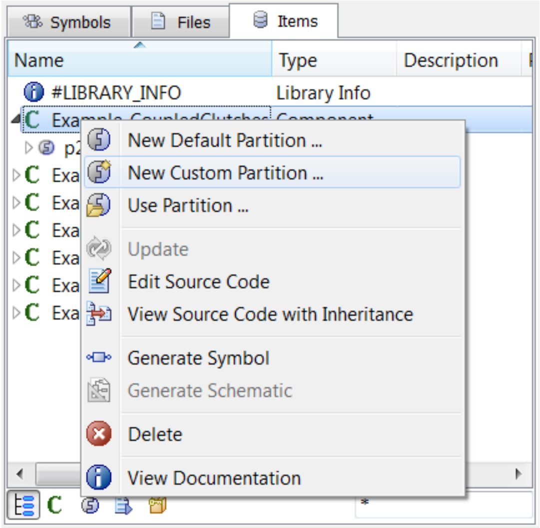
```
---------------------------------------------------------------------------
-- Component Minimum
---------------------------------------------------------------------------
-- Purpose:
--    Outputs the minimum input value.
---------------------------------------------------------------------------
COMPONENT Minimum IS_A MI2MOs "Outputs the minimum input value"
   CONTINUOUS
      EXPAND (i IN 1, n)
         s_out.signal[i] = ZONE (s_in_1.signal[i] < s_in_2.signal[i]) s_in_1.signal[i]
                           OTHERS                                     s_in_2.signal[i]

END COMPONENT

```

Partition

– Mathematical model associated to a component.

– Defines the inputs and outputs of the simulation.

– Generates a definite solution for the complete equation system.



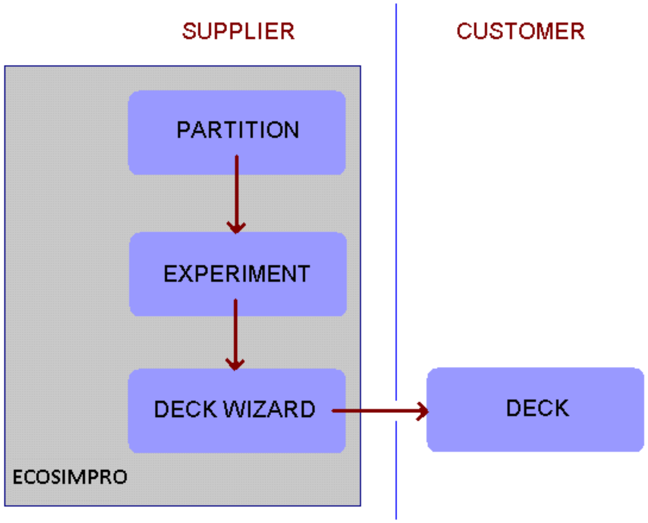
Experiment

- Configures the simulation on a defined partition of a component.
- Customizes the reports of the simulations
- Generates a sequence of studies of the behaviour of the component

```
EXPERIMENT exp1 ON Component.Partition
DECLS
--
INIT     -- set initial values for variables
--       -- Dynamic variables
--       -- Algebraic variables
BOUNDS   -- set expressions for boundary variables: v = f(t,...)
--
BODY
   REPORT_MODE = IS_STEP
   TIME = 0
   TSTOP = 2
   CINT = 0.005 --0.01
   INTEG()
END EXPERIMENT
```

Deck

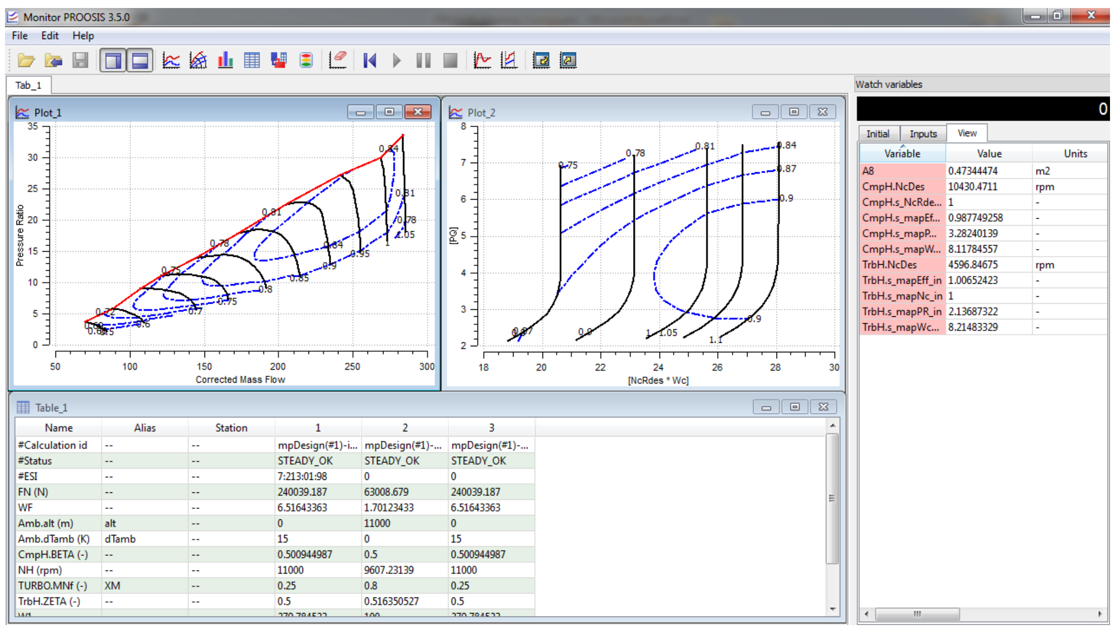
- Standalone Executable file that encapsulates models and experiments
- From the user point of view, a Deck is like a black – box
- Allows the connection with other software



Monitor

– Graphical Interface to analyze the results of simulations in detail

– Representation of variables and performance maps.



GRAPHICAL USER INTERFACE

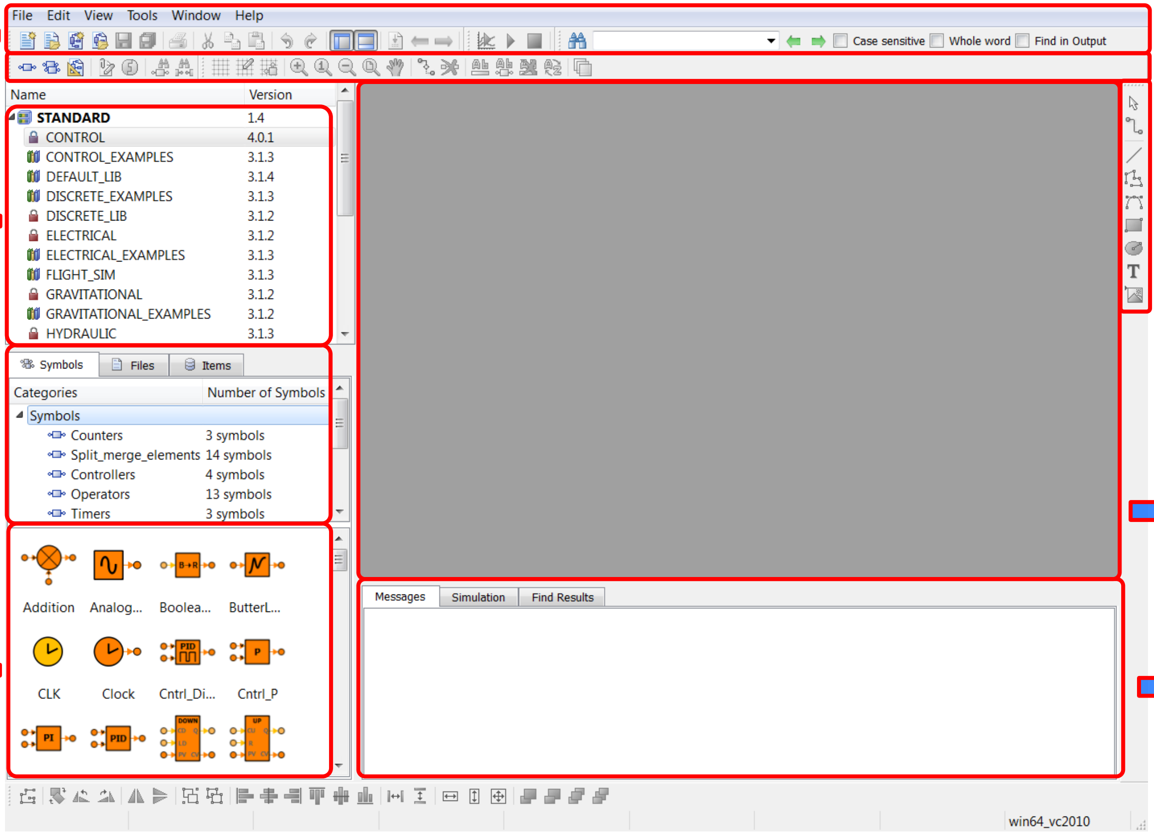
Graphical User Interface

General Toolbar

Work Space Libraries

Library Elements

Library Palette



Schematic Toolbar

Graphical Toolbar

Editing Area

Messages Area

Graphical User Interface

Library View

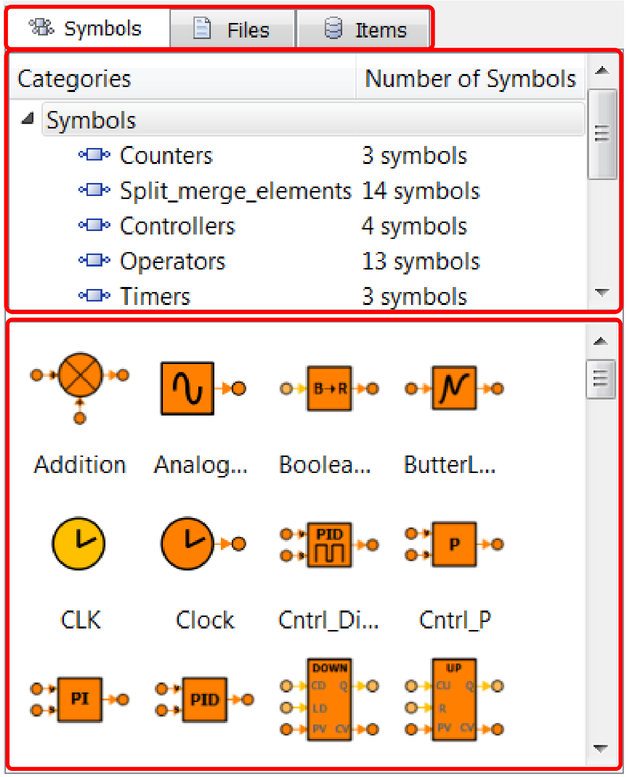
Library Tabs:

- Symbols
- Files
- Items

Pallette Subcategories.

Palette:

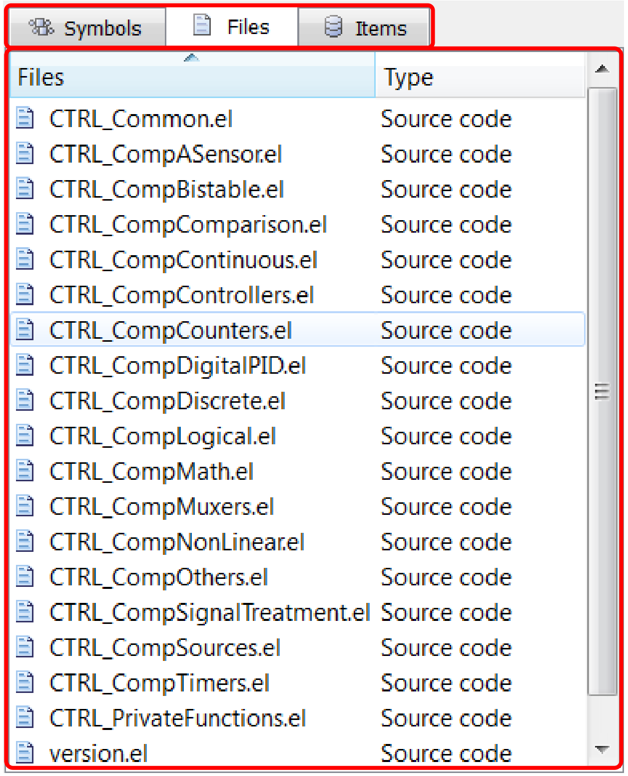
Library Symbols



Library View

Library Tabs:

- Symbols
- Files
- Items



Files:

- Component Code
- Schematics

Library View

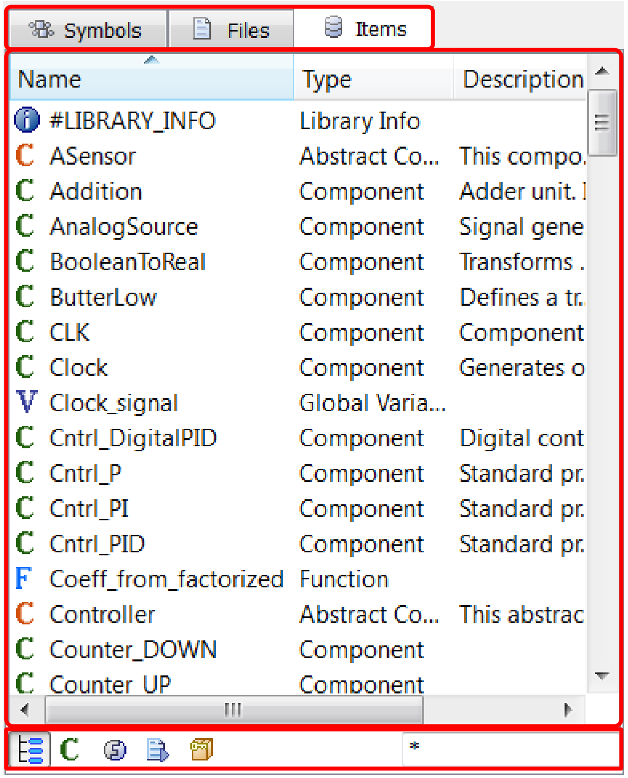
Library Tabs:

- Symbols
- Files
- Items

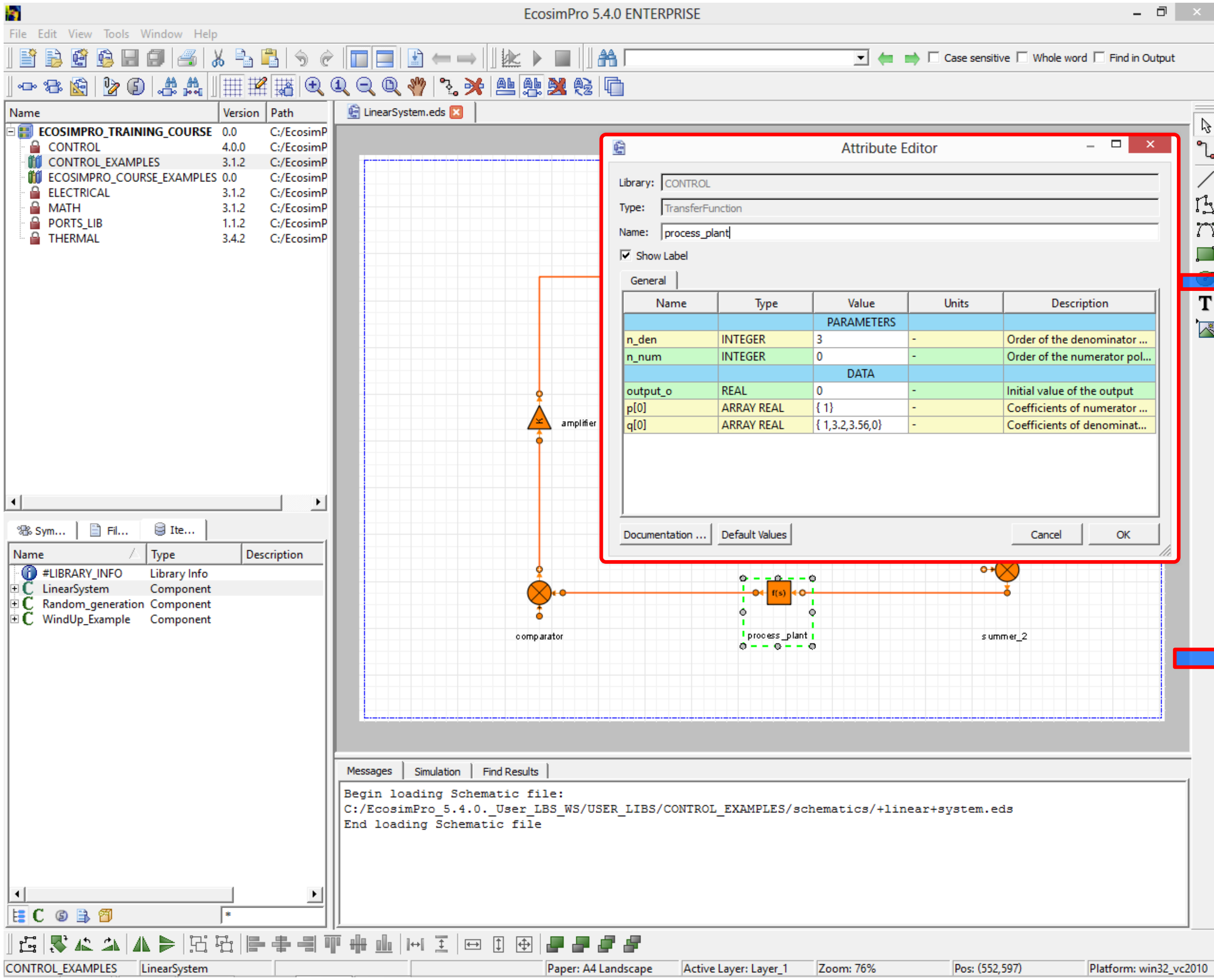
Items:

- Components
- Variables
- Functions
- Partitions
- Experiments

Filters & View



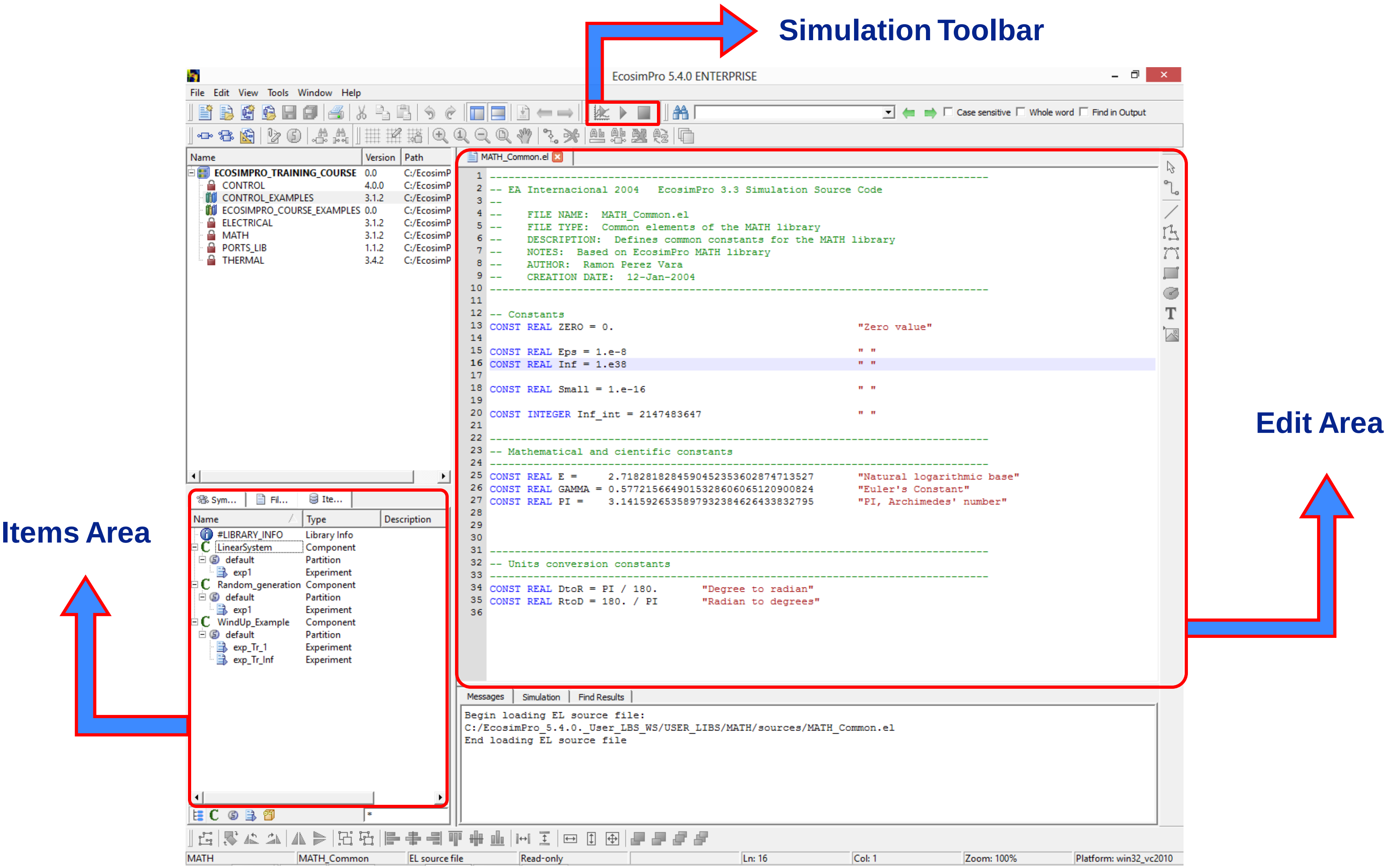
Graphical User Interface



Attribute Editor

Drag and Drop Modelling

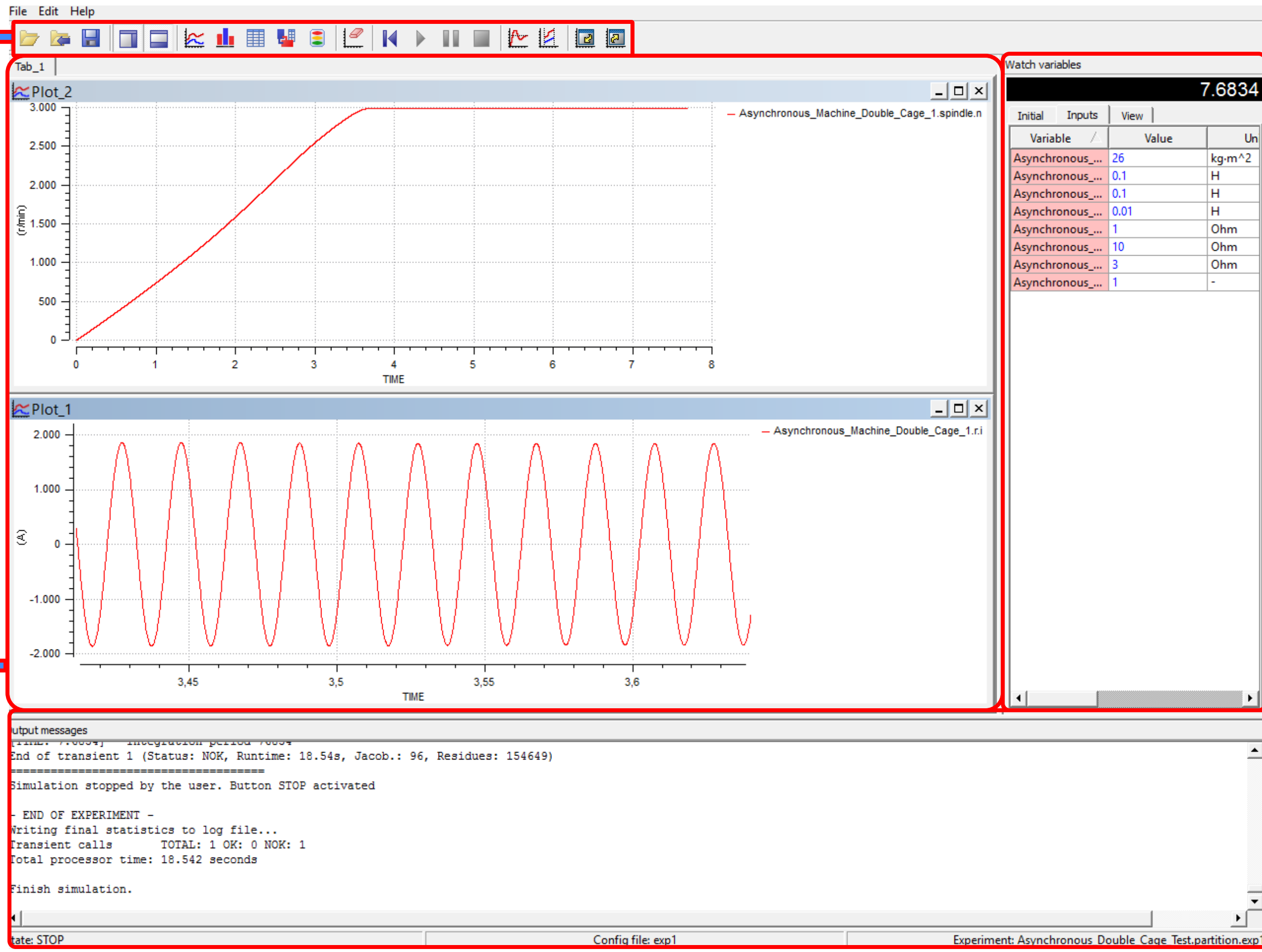
Graphical User Interface



Graphical User Interface

Simulation Tools

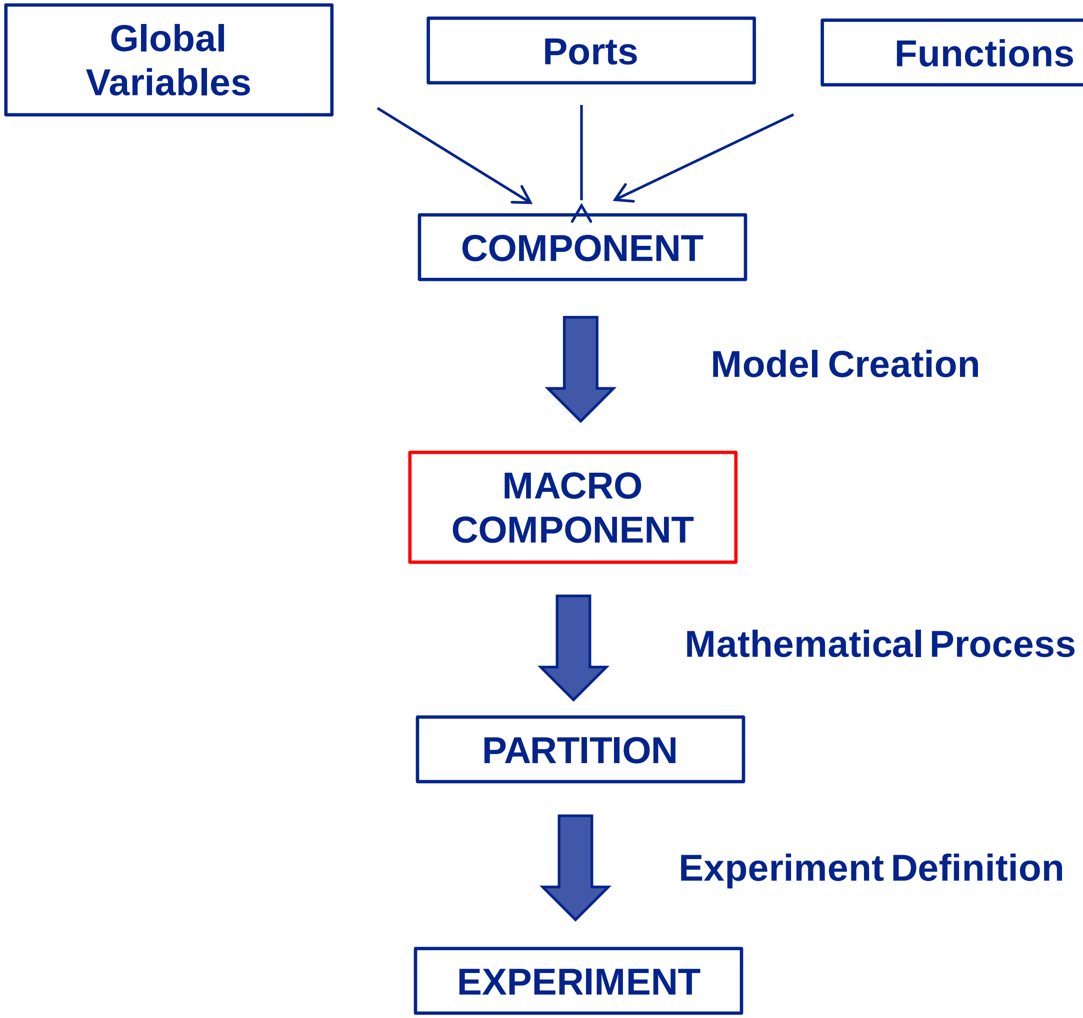
Plots Area



Initial, Inputs and View Tabs

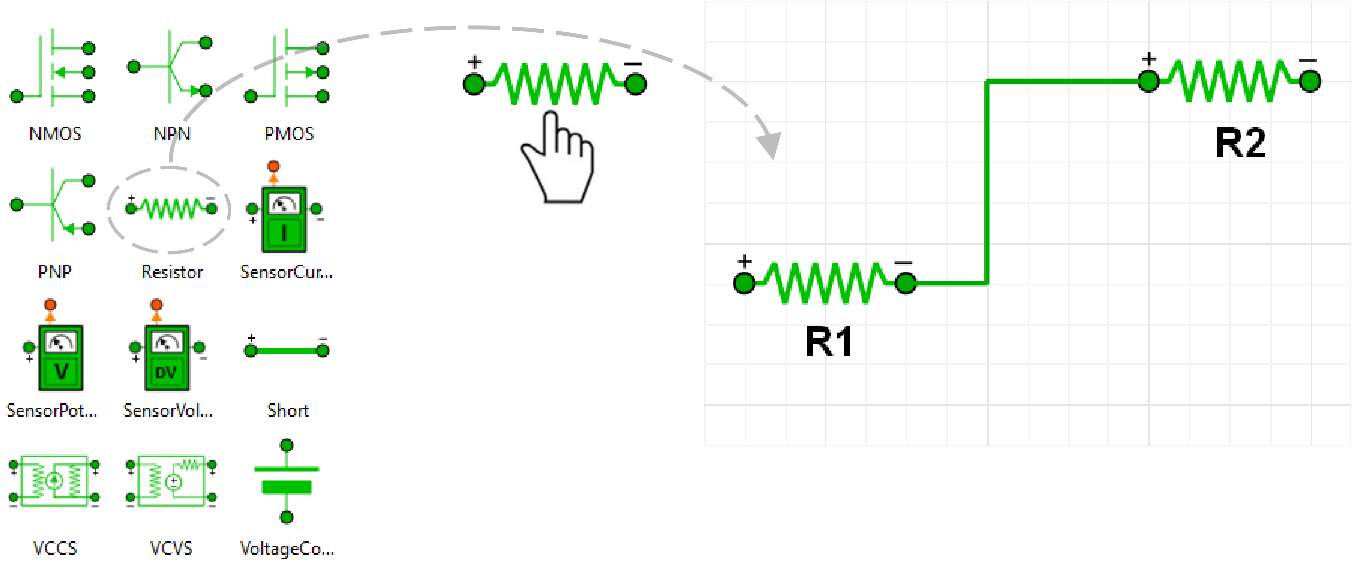
Messages Area

Schematics


Global Variables
Ports
Functions
COMPONENT
Model Creation
MACRO COMPONENT
Mathematical Process
PARTITION
Experiment Definition
EXPERIMENT

Schematics

- Macro-components can be created graphically by means of Schematic diagrams
- Components from different libraries can be instanciated by **drag & drop**
- **Ports** and **Connections** define the interface between components

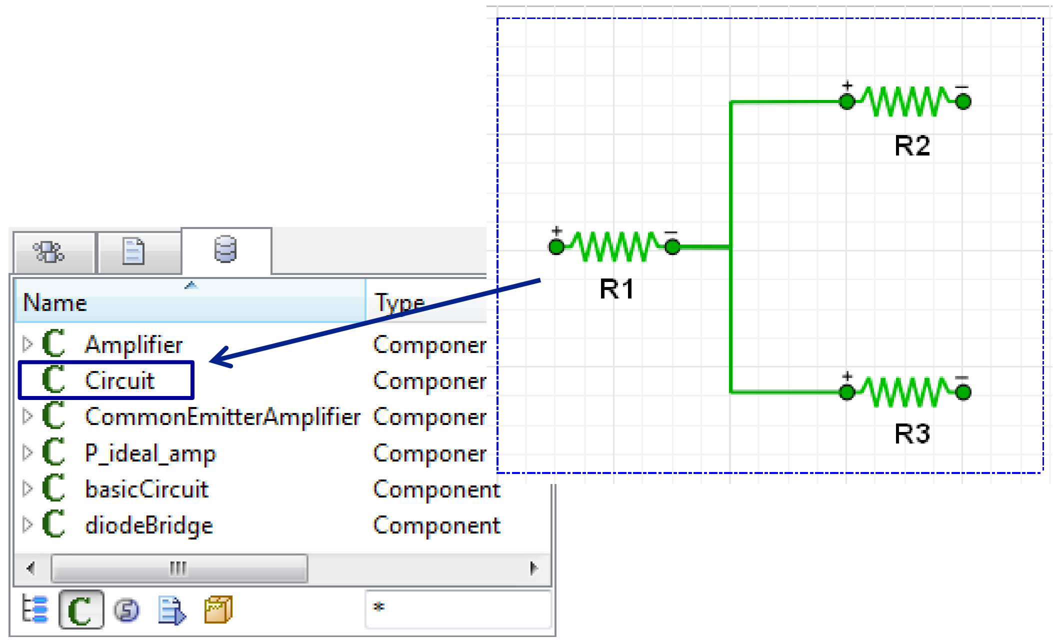


Schematics

- When Schematics are compiled, the component is added to the library

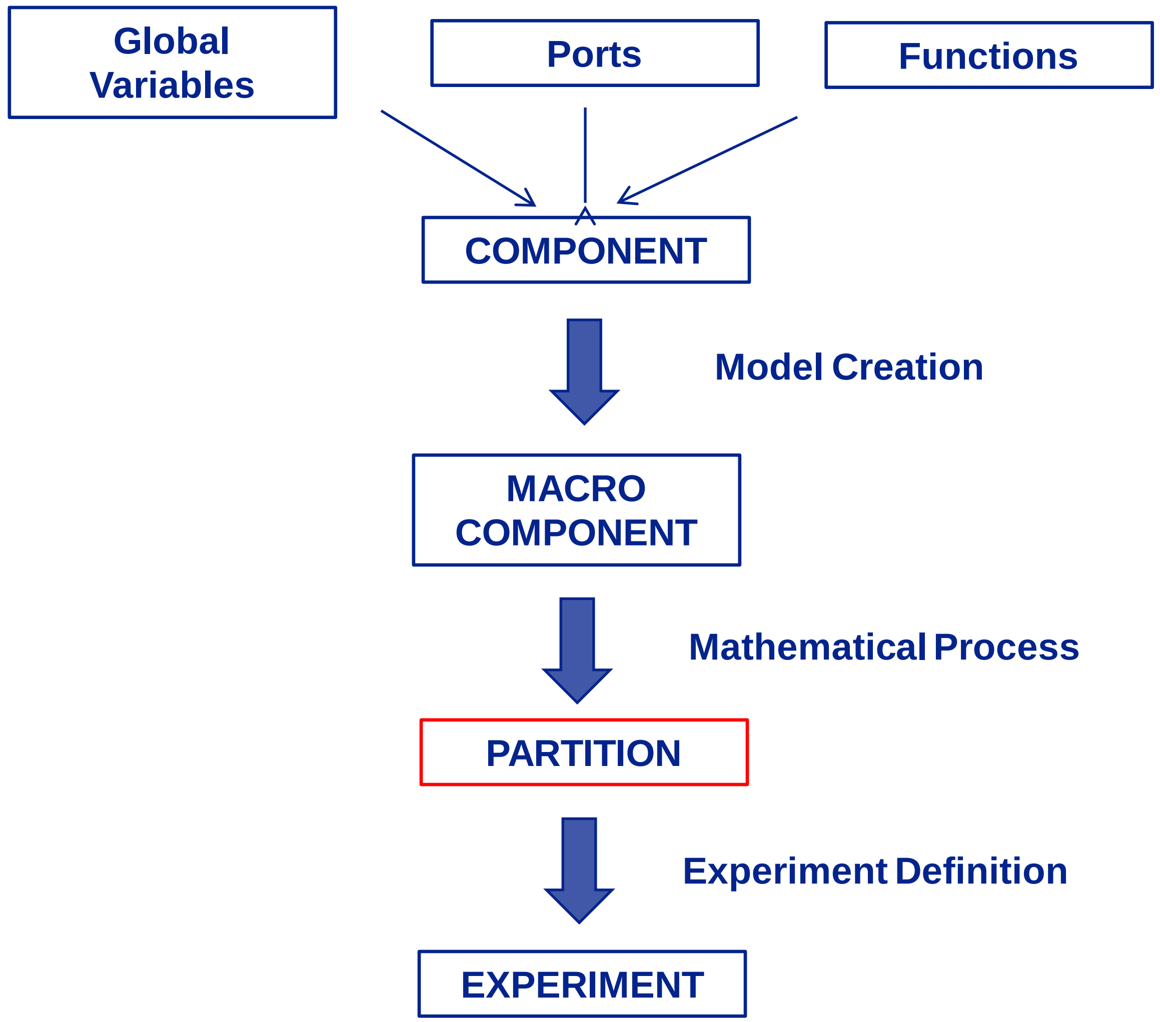
- This new item contains the complete formulation of the instances, and their port variables relationships

```
COMPONENT Resistor IS_A OnePort
    "Resistor"
    DATA
        REAL R=1 UNITS u_Ohm  "Resistance"
    CONTINUOUS
        R*i = v
END COMPONENT

COMPONENT Circuit
TOPOLOGY

      ---------------------------
      -- Coded in Circuit
      ---------------------------
   ELECTRICAL.Resistor R1  (R = 1)
   ELECTRICAL.Resistor R2  (R = 1)
   ELECTRICAL.Resistor R3  (R = 1)

      ---------------------------
      -- Coded in Circuit
      ---------------------------
   CONNECT R1.e_n TO R2.e_p
   CONNECT R1.e_n TO R3.e_p
END COMPONENT
```



PARTITION

Partitions


Global Variables
Ports
Functions
COMPONENT
Model Creation
MACRO COMPONENT
Mathematical Process
PARTITION
Experiment Definition
EXPERIMENT

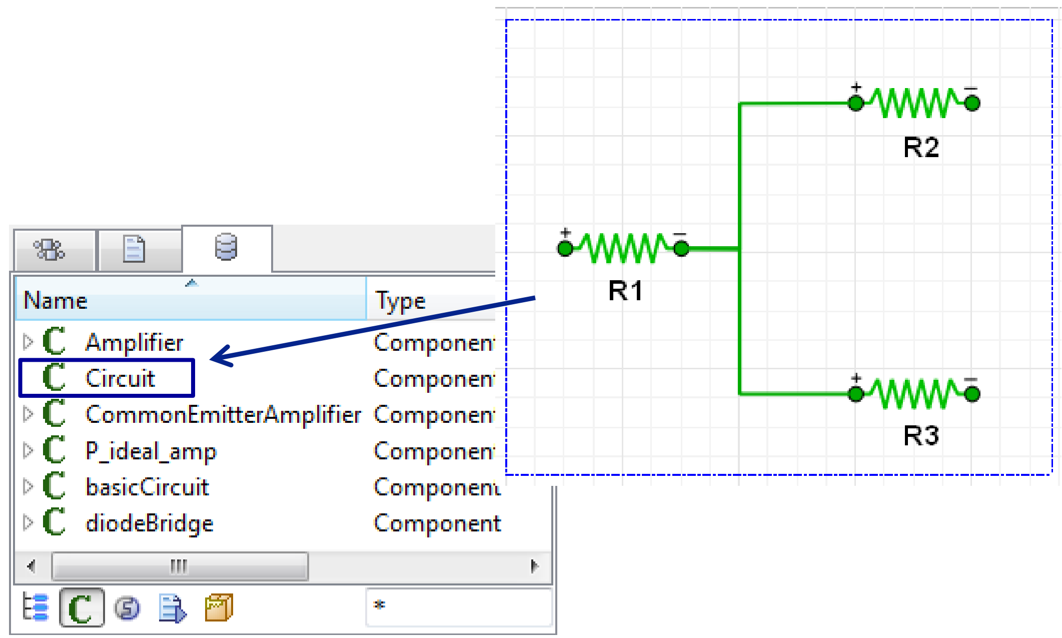
Partitions

System of Equations

- When a Component source code is compiled, all its variables and equations are encapsulated to be reused in the future

- This component contains the equations that must be satisfied during the simulation

```
COMPONENT Resistor IS_A OnePort
    "Resistor"
    DATA
        REAL R=1 UNITS u_Ohm  "Resistance"
    CONTINUOUS
        R*i = v
END COMPONENT

COMPONENT Circuit
TOPOLOGY

      ----------------------------
      -- Coded in Circuit
      ----------------------------
   ELECTRICAL.Resistor R1 (R = 1)
   ELECTRICAL.Resistor R2 (R = 1)
   ELECTRICAL.Resistor R3 (R = 1)

      ----------------------------
      -- Coded in Circuit
      ----------------------------
   CONNECT R1.e_n TO R2.e_p
   CONNECT R1.e_n TO R3.e_p
END COMPONENT
```



Partitions

System of Equations

- These equations are not organized. The system cannot be solved

- The partition organize all the equations and check the number of **boundaries** that must be set by the user:

N_bounds = N_variables – N_equations

- **Dynamic** variables and **Algebraic** loops are detected

Then, the **Mathematical Model** is ready to be solved if the initial values and Boundaries are given

Name	Type
Amplifier	Component
Circuit	Component
partition	Partition
CommonEmitterAmplifier	Component
P_ideal_amp	Component
basicCircuit	Component
diodeBridge	Component

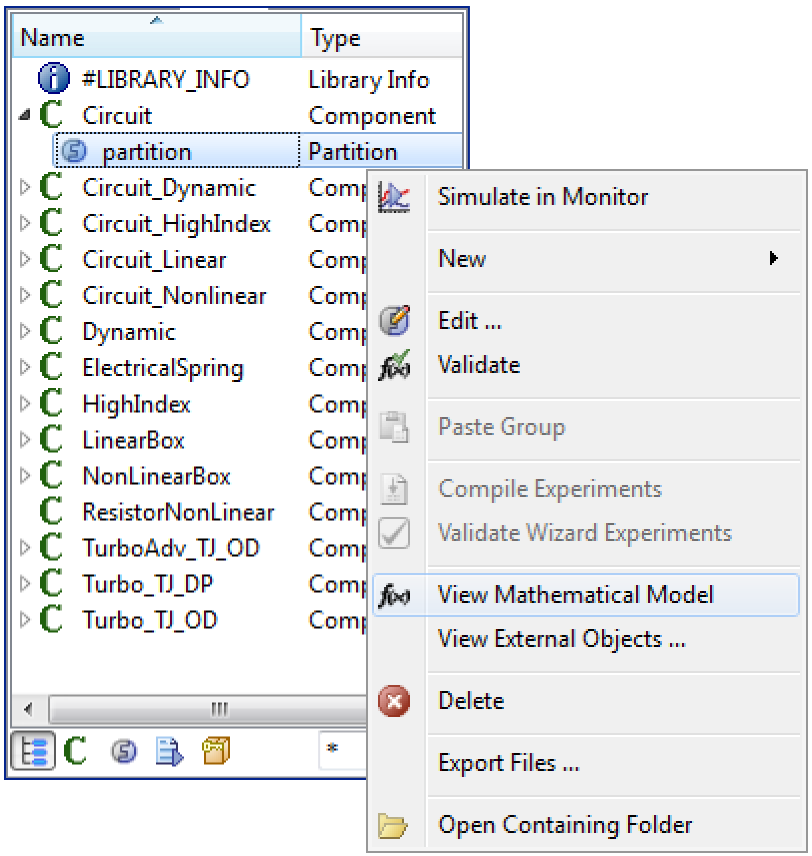
→

Sort of **Equations**
Boundaries information
Variables **Initialization**

Partitions

System of Equations

- The **Mathematical Model** of any partition can be analyzed to check the sequential solving process



GENERAL STATISTICS

INFO	#
Number of equations:	10
Number of boxes (coupled subsystems of equations):	0
Number of linear boxes:	0
Number of nonlinear boxes:	0
Number of input DATA:	3
Number of input BOUNDARY:	3
Number of output EXPLICIT :	10
Number of output DYNAMIC/DERIVATIVE:	0
Number of output ALGEBRAIC:	0
Size of Jacobian matrix (DYNAMIC+ALGEBRAIC):	0x0
Default integration method:	DASSL

TYPE OF VARIABLES

TYPE	VARIABLE	DATA	CONSTANT
REAL	13	3	0
INTEGER	0	0	0
STRING	0	0	3
TABLE	0	0	0

GLOBAL FLAGS:

FLAG	VALUE
Remove derivatives	FALSE
Inhibit automatic reduction of equations	FALSE
Remove not used variables	FALSE
Generate code to check mathematical functions	TRUE
Generate code to analyse the performance of functions	FALSE
Obfuscate partition and experiment C++ generated code	FALSE

BOUNDARIES:

#	NAME	ALIAS	UNITS	DESCRIPTION	INITIAL
1	R1.e_p.v	v1	V	Potential at pin	
2	R2.e_p.i	i2	A	Current flowing into the pin	
3	R3.e_p.i	i3	A	Current flowing into the pin	

JACOBIAN INDEPENDENT VARIABLES:

POS	VARIABLE	ALIAS	CATEGORY	UNITS	DESCRIPTION	INITIAL	RESIDUE EQUATION

SORTED EQUATIONS:

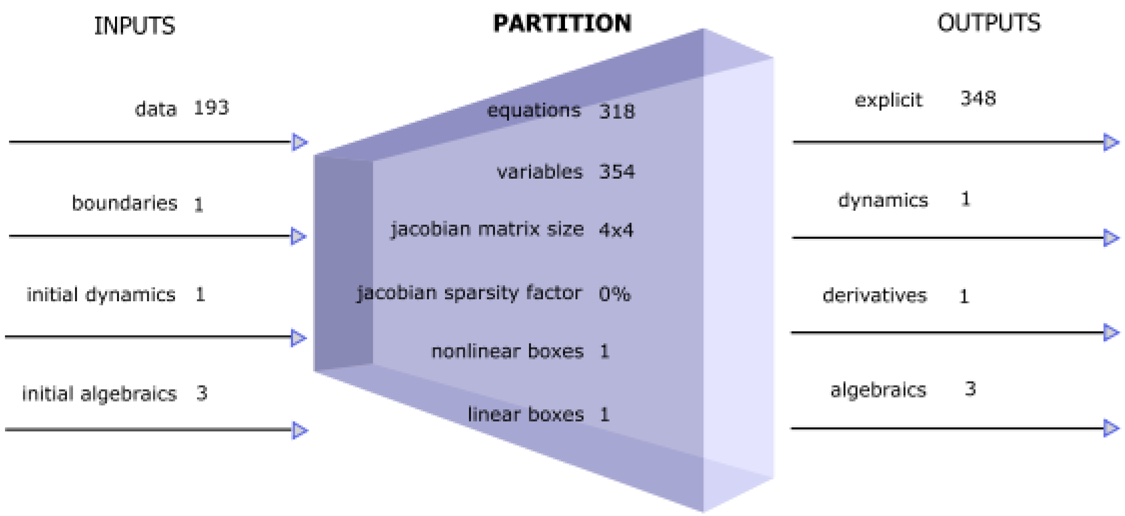
```
###eqts 10
[E-1] i1 = i2 + i3
[E-2] R1.v = R1.R * i1
[E-3] R3.v = R3.R * i3
[E-4] v3 = v1 - R3.v - R1.v
[E-5] vc = v3 + R3.v
[E-6] R1.e_n.i = -i1
[E-7] R2.e_n.i = -i2
[E-8] R2.v = R2.R * i2
[E-9] v2 = v3 + R3.v - R2.v
[E-10] R3.e_n.i = -i3
```

Partitions

Component vs. Partition

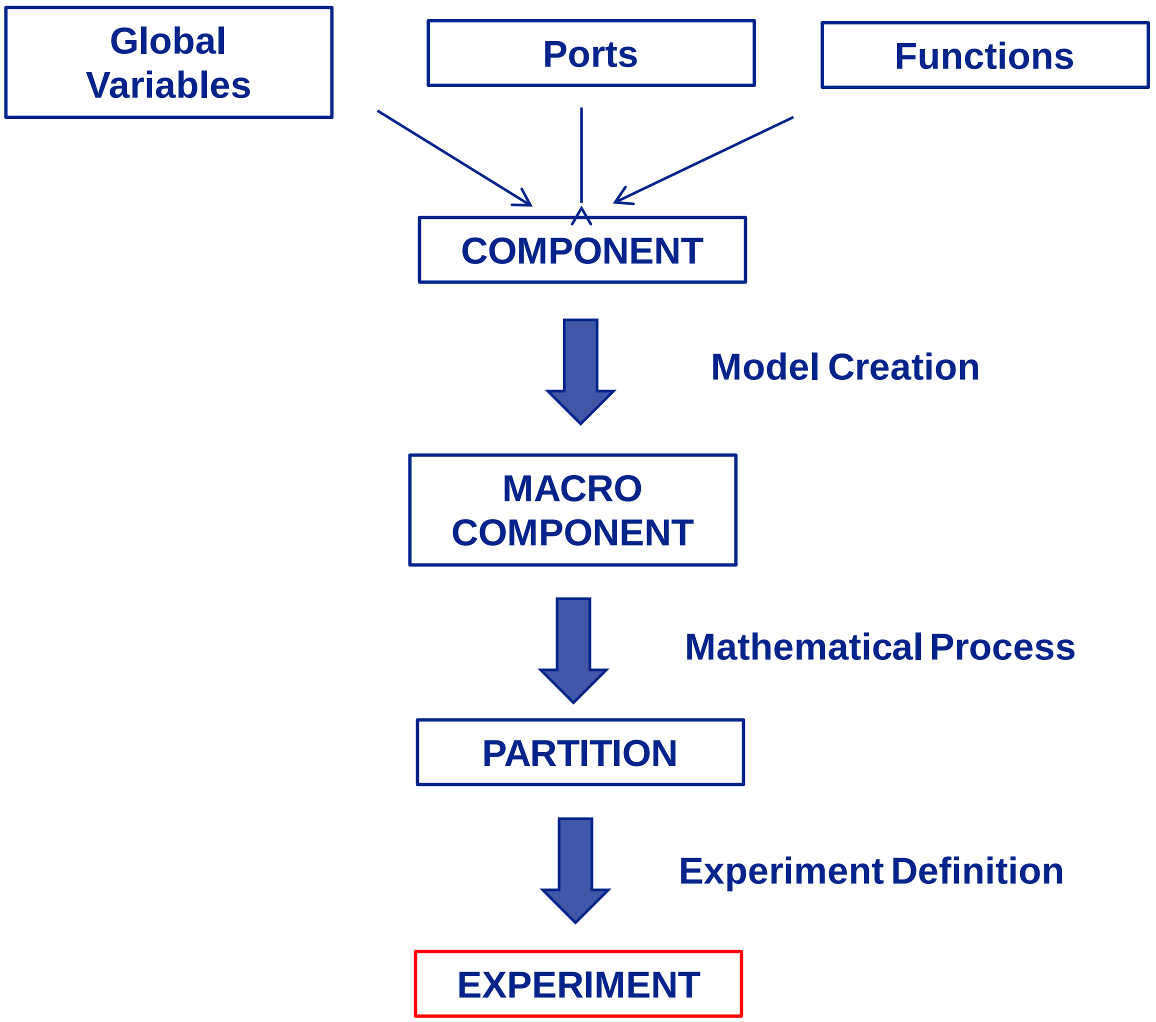
- The **Component** provides the set of equations to be solved. The **Partition** organizes them to obtain a consistent system of equations

- The Partition defines the **INPUTS** and **OUTPUTS** of the model.



EXPERIMENT

Experiments



Experiments

Simulation Definition

- The Partition defines the **Inputs** and **Outputs**. Different calculations can be defined for it in a Experiment

- The Experiment is a sequence of statements which allows the user to:
  - **Initialize data**
  - **Solve** the **system of equations** defined in the partition
  - Create **extra functions** to manage the model variables
  - Define **extra equations** to extend the model equations
  - Prepare the model for **external connections**

Experiments

Simulation Definition

- Typical calculations performed in experiments:
  - **Steady**
  - **Transient**
  - Design
  - Parametric
  - Sensitivity Analysis

- The Experiment can be used to define complex sets of calculations to perform complete analysis of models

- The **Experiment Wizard** provides an intuitive interface to create automatically the experiment code

Experiment Wizard

Create experiment code requires a **deep knowledge** of PROOSIS functions

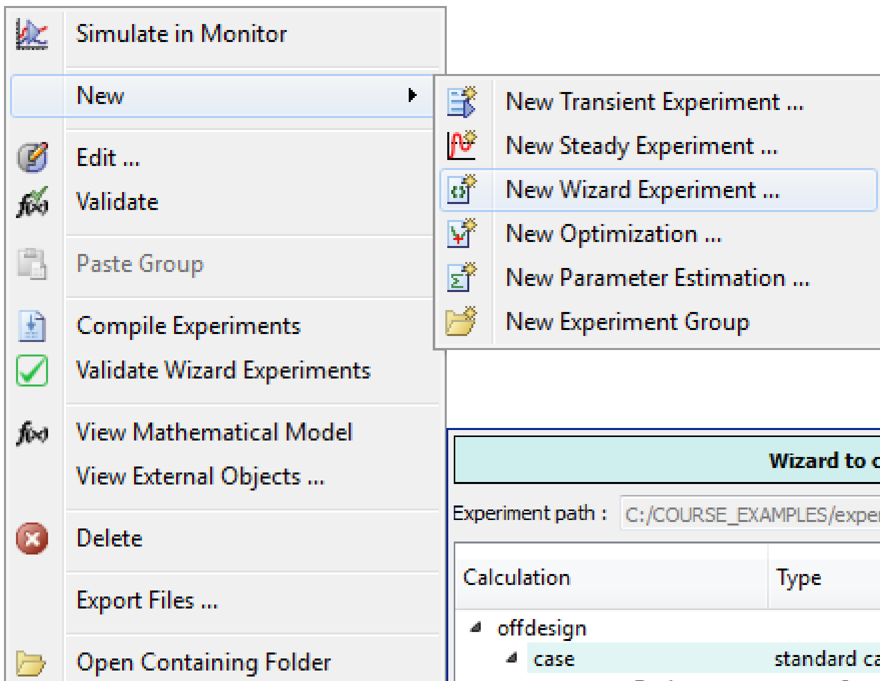
The Experiment Wizard helps the user to generate experiments **graphically**

Wizard experiments are **easier to maintain** when models/partitions change

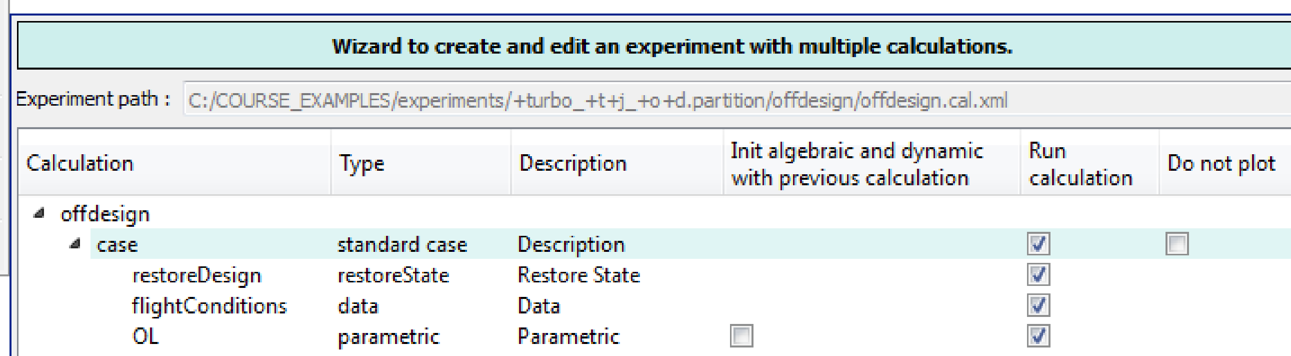
Complex experiments can be generated

Experiment Wizard

The Wizard experiments have the following structure:



- **Case1**
  - Calc1
  - Calc2
  - ...
- **Case2**
  - Calc3
  - Calc4



Experiments

Experiment Wizard

CASE: Group of calculations

- **Standard**: Executes the calculations sequentially
- **Parametric**: Create a loop to execute a set of calculations iteratively.

CALCULATION: Series of statements to set the Inputs/Outputs and solve the Mathematical Model.

- **Simulations:** transient, steady, sensitivity, parametric, ...
- **Functions:** function call, data, data cardpack
- **Reset:** reinitialize variables, execute init blocks, ...
- **Report management:** save state, restore state

Experiment Wizard

TYPE OF CALCULATIONS

- **Steady:** Calculate steady state point. Dynamics are calculated iteratively (as Algebraics)
- **Extended Steady:** Calculate an steady point. Design data are calculated iteratively to fulfil extra closure equations
- **Transient:** Simulate time evolutions. Dynamics variables are initialized by the user
- **Extended Transient:** Transient simulation. Boundaries are calculated to fulfil extra closure equations
- **Parametric:** Steady / Transien loops
- **Sensitivity:** Calculate input (Data or Bounds) sensitivity in Steady calculation results

Electrical Example

DATA

C1 = 1e-3 F

L1 = 10 H

R1 = 2e3 Ohm

V = 10 V

BOUNDARY

V3 = 100 V

DYNAMIC

C1.v = 0 V

i2 = 0 A

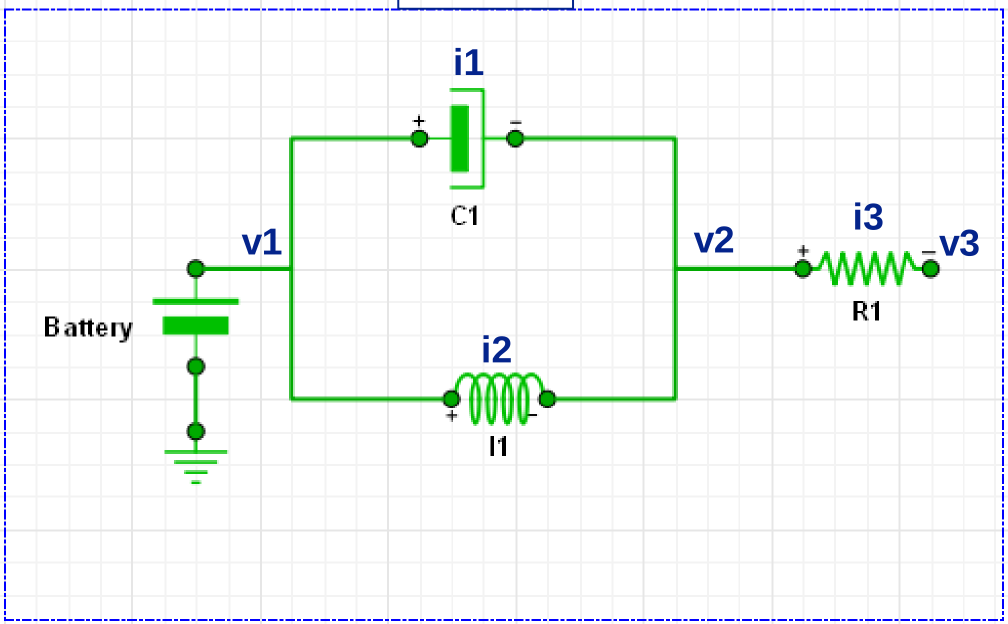
CONFIG

CINT = 1E-3

TIME = 0

TSTOP = 10

MODEL



INPUTS

OUTPUTS

EXPLICIT

v1, v2

i3

…

DYNAMIC

C1.v

i2

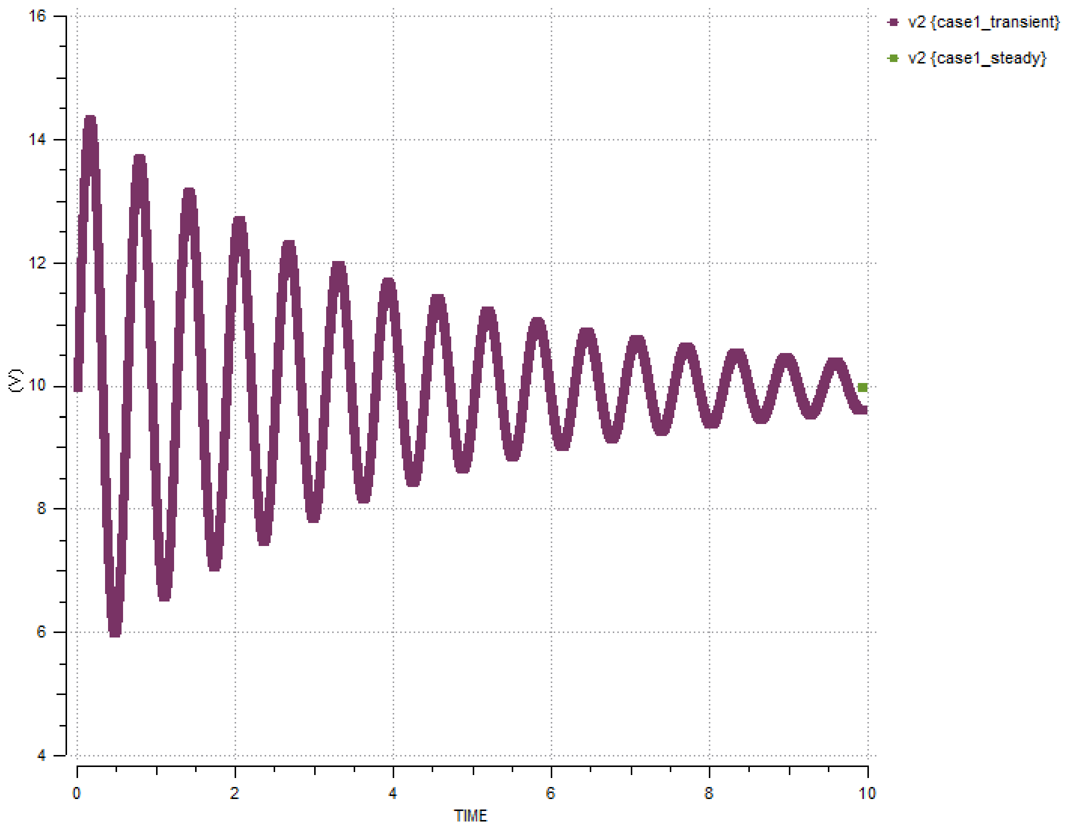
DERIVATIVE

C1.v

i2’

Experiments

Transient vs. Steady

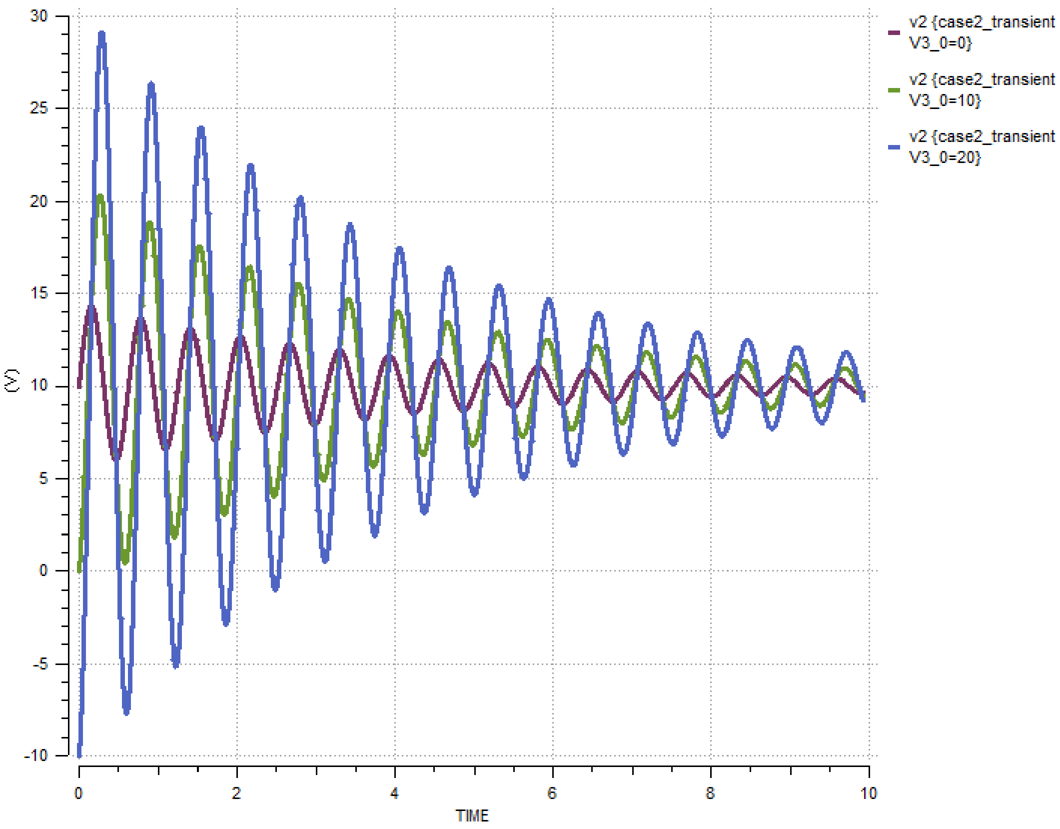


TRANSIENT: the violet line shows the evolution of the v2 variable from t= 0s to t=10s

STEADY: the green point represents the value of v2 when the system becomes stable and does not changes

Experiments

Parametric Study



PARAMETRIC STUDY:

Loops of Transient or Steady calculations carried out to analyze the influence of:

- **Initial states**
- **Component characteristics**
- **Boundary conditions**
- **...**