OMEdit

OpenModelica Connection Editor

User Manual
Version 1.7.0

August 2011

This Description is extracted
from Chapter 2 of the OpenModelica 1.7 User's Guide

Table of Contents

Table of Contents 2

Chapter 1 : Getting Started 3

1.1 About OMEdit 4
1.2 How to Start OMEdit? 4
1.2.1 Windows 4
1.2.2 Linux 4
1.2.3 Mac OS X 5
1.3 Introductory Model in OMEdit 5
1.3.1 Creating a New File 5
1.3.2 Adding Component Models 6
1.3.3 Making Connections 6
1.3.4 Simulating the Model 8
1.3.5 Plotting Variables from Simulated Models 9
1.4 How to Create User Defined Shapes – Icons 10

Chapter 2 : OMEdit Views and Windows/Tabs 12

2.1 Views 13
2.1.1 Modeling View 13
2.1.2 Plotting View 13
2.1.3 Interactive Simulation View 13
2.2 Windows 13
2.2.1 Library Window 13
2.2.1.1 Viewing Models Description 14
2.2.1.2 Viewing Models Documentation 14
2.2.1.3 How to Open an Existing Model? 14
2.2.1.4 How to create a Copy of an Existing Model? 14
2.2.1.5 How to Check a Model? 14
2.2.1.6 How to Instantiate a Model? 14
2.2.1.7 How to Rename a Model? 14
2.2.1.8 How to Delete a Model? 14
2.2.2 Designer Window 15
2.2.3 Plot Variables Window 15
2.2.4 Messages Window 15
2.2.5 Documentation Window 15
2.2.6 Model Browser Window 16
2.3 Dialogs 17
2.3.1 New Dialog 17
2.3.2 Simulation Dialog 17
2.3.3 Model Properties Dialog 17
2.3.4 Model Attributes Dialog 18

Chapter 3 : Interactive Simulation in OMEdit 19

3.1 OpenModelica Interactive 20
3.2 Invoking Interactive Simulation 20
3.3 Interactive Simulation View 20

Chapter 1 : Getting Started

- ✓ A brief introduction of OMEdit
- ✓ How to start OMEdit?
- ✓ How to create a DCmotor model in OMEdit?
- ✓ How to create user defined shapes in OMEdit?

1.1 About OMEdit

OMEdit - OpenModelica Connection Editor is the new Graphical User Interface for graphical model editing in OpenModelica. It is implemented in C++ using the Qt 4.7 graphical user interface library and supports the Modelica Standard Library version 3.1 that is included in the latest OpenModelica (version 1.6.0) installation. This chapter gives a brief introduction to OMEdit and also demonstrates how to create a `DCmotor` model in it.

OMEdit provides user friendly features like;

- *Modeling* – Easy model creation for Modelica models.
- *Pre-defined models* – Browsing the Modelica Standard library to access the provided models.
- *User defined models* – Users can create their own models for immediate usage and later reuse.
- *Component interfaces* – Smart connection editing for drawing and editing connections between model interfaces.
- *Simulation* – Subsystem for running simulations and specifying simulation parameters start and stop time, etc.
- *Plotting* – Interface to plot variables from simulated models.

1.2 How to Start OMEdit?

1.2.1 Windows

OMEdit can be launched using the executable placed in `OpenModelicaInstallationDirectory/bin/OMEdit/OMEdit.exe`. Alternately, choose `OpenModelica > Open Modelica Connection Editor` from the start menu in Windows. A splash screen similar to the one shown in Figure 1-1 will appear indicating that it is starting OMEdit. After the splash screen the main OMEdit window will appear; see Figure 1-2.

Figure 1-1: OMEdit Splash Screen.

1.2.2 Linux

?? fill in

1.2.3 Mac OS X

?? fill in

1.3 Introductory Model in OMEdit

In this section we will demonstrate how one can create Modelica models in OMEdit, e.g. a `DCmotor`.

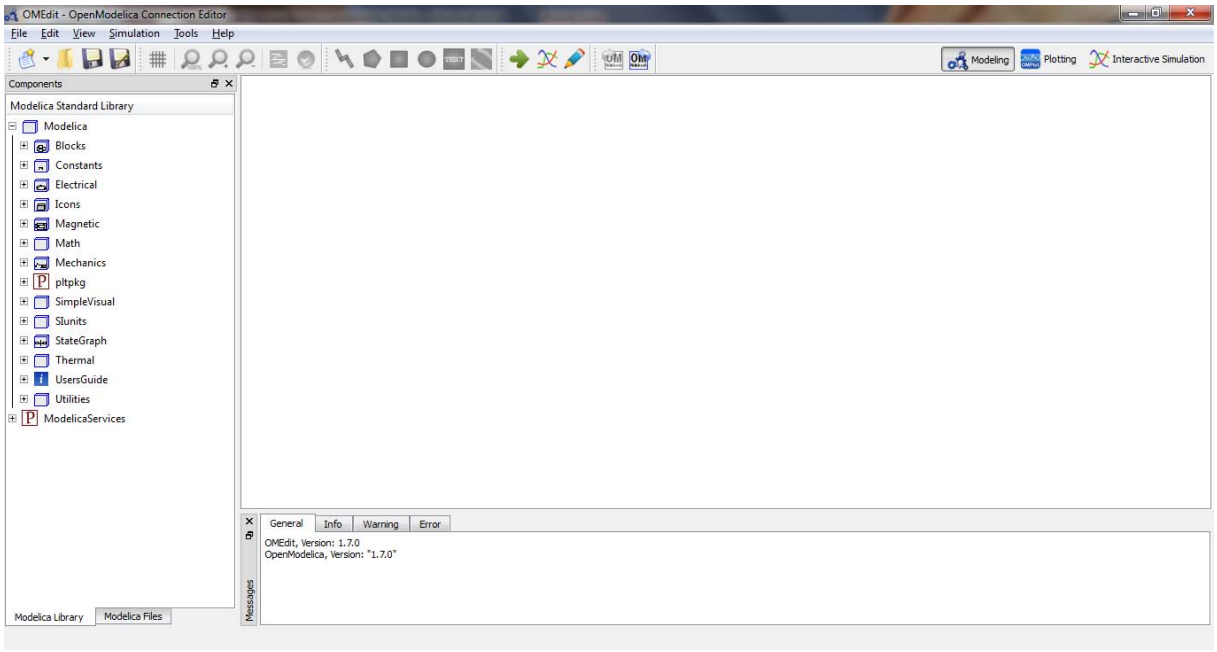

Figure 1-2: OMEdit Main Window.

1.3.1 Creating a New File

Creating a new file/model in OMEdit is rather straightforward. In OMEdit the new file can be of type `model`, `class`, `connector`, `record`, `block`, `function` and `package`. User can create any of the file types mentioned above by selecting `File > New` from the menu. Alternatively, you can also click on the drop down button beside `new icon` shown in toolbar right below the File menu. See Figure 1-4.

For this introductory example we will create a new model named `DCmotor`. By default the newly created model will open up in the tabbed view of OMEdit, also called `Designer Window` (see Chapter 2 section 2.2.2), and become visible. The models are created in the OMC global scope unless you specify the parent package for it.

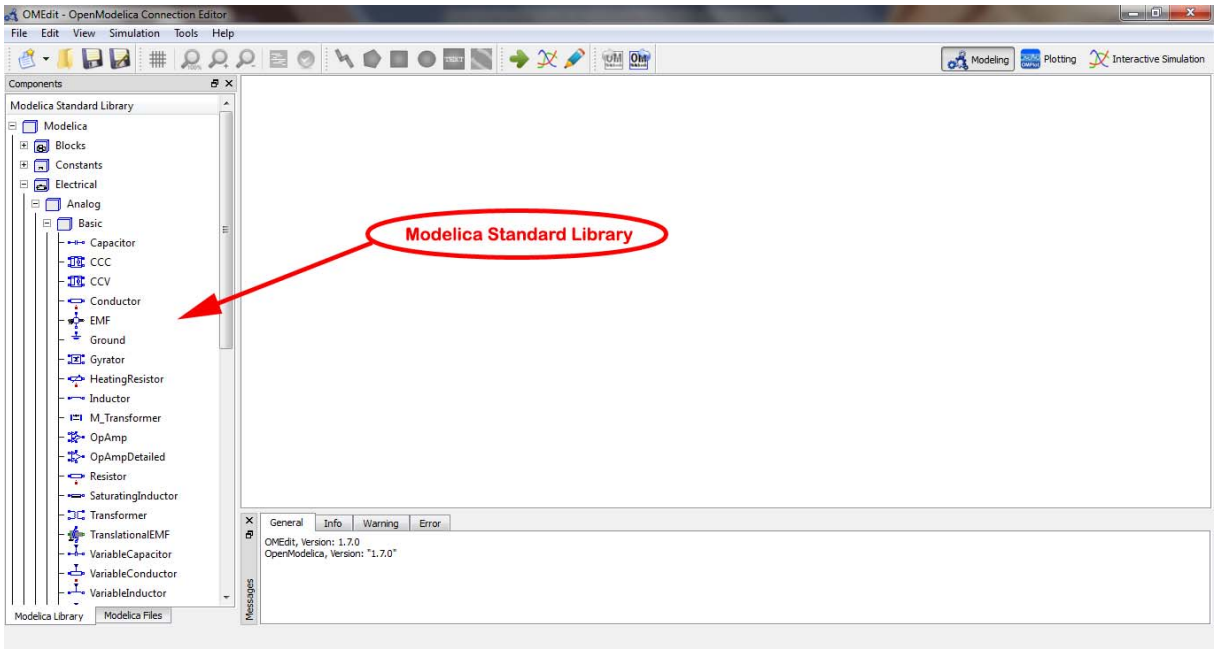

Figure 1-3: Modelica Standard Library.

1.3.2 Adding Component Models

Modelica standard library is loaded automatically and is available in the left dock window. The library is retrieved through the `loadModel(Modelica)` API call and is loaded in the OMC symbol table and workspace after the command execution is completed. Component models available in the Modelica standard library are added to the models by doing a drag and drop from the `Library Window` (see Chapter 2 section 2.2.1). Navigate to the component model in the library tree, click on it, drag it to the model you are building while pressing the mouse left button, and drop the component where you want to place it in the model.

Similarly, the component models present in the Modelica Tree View, i.e. the Custom Modelica Models also can be added into some other custom modelica models by a similar drag and drop. The dropped component keeps getting updated as soon as we make any changes in the original model.

For this example we will add four components as instances of the models `Ground`, `Resistor`, `Inductor` and `EMF` from the `Modelica.Electrical.Analog.Basic` package, an instance of the model `SignalVoltage` from the `Modelica.Electrical.Analog.Sources` package, one instance of the model `Inertia` from the `Modelica.Mechanics.Rotational.Components` package and one last instance of the model `Step` from the `Modelica.Blocks.Sources` package.

1.3.3 Making Connections

In order to connect one component model to another the user simply clicks on any of the ports. Then it will start displaying a connection line. Then move the mouse to the target component where you want to finish the connection and click on the component port where the connection should end. You do not need to hold the mouse left button down for drawing connections.

In order to have a functioning `DCmotor` model, connect the `Resistor` to the `Inductor` and the `SignalVoltage`, `EMF` to `Inductor` and `Inertia`, `Ground` to `SignalVoltage` and `EMF`, and finally `Step` to `SignalVoltage`. Check Figure 1-7 to see how the `DCmotor` model looks like after connections.

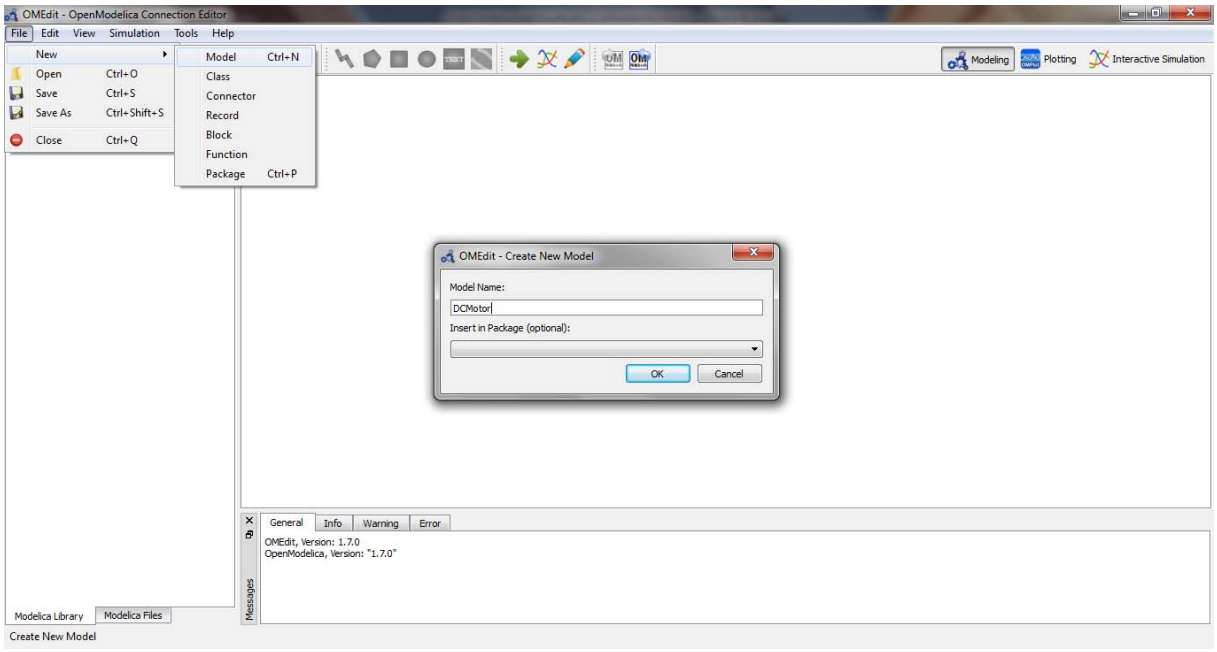

Figure 1-4: Creating a new model.

In order to connect one component model to another the user first enables the connect mode from the toolbar to make the connect mode active. See Figure 1-5.

Figure 1-5: Connect mode button.

The different kinds of connections are:

- *Connections for Models* – If the connect mode is active then user then simply clicks on any of the ports.
- *Connection for Connector Types* – If the connect mode is active then we can also connect two components of connector types to each other if they are of the same type. Since, it is a connector type, to start or end a connection, you can just click anywhere on the component icon and it will start.
- *Connection for Connector Array Type* – If any of the start port or the end port of the connection is of an array type, the user also needs to add indices at which the connection is to be made. For this a dialog box will pop up, as soon as the user clicks on the end port of the connection asking the user for indices of the array for whichever port it is necessary. For example, let's say, the user wants to connect, the `x` component instance of `Modelica.Electrical.Digital.Converters.BooleanToLogic` to the `y` component instance of `Modelica.Blocks.Sources.BooleanConstant`. Now since, `Modelica.Electrical.Digital.Converters.BooleanToLogic.x` is a connector array, as soon as the user clicks on the end port for this connection, a dialog box appears asking user the index of the start instance. See Figure 1-6.

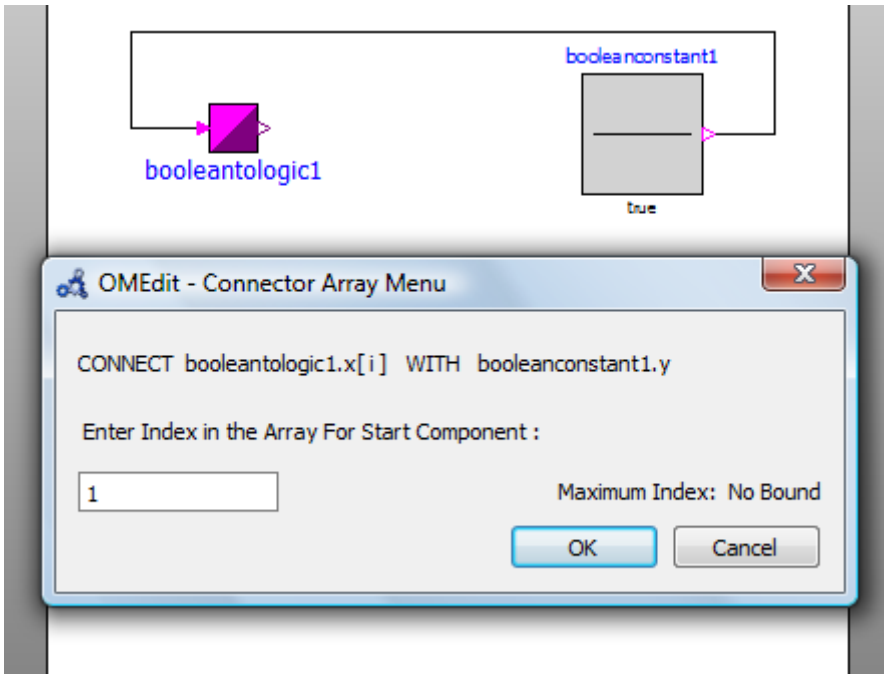

Figure 1-6: Connector Array Menu

1.3.4 Simulating the Model

The OMEdit Simulation dialog (see Chapter 2 section 2.3.2) can be launched either from `Simulation > Simulate` or by clicking the `simulate icon` from the toolbar. Once the user clicks on `Simulate!` button, OMEdit starts the simulation process, at the end of the simulation process the Plot Variables Window (see Chapter 2 section 2.2.3) useful for plotting will appear at the right side. Figure 1-8 shows the simulation dialog.

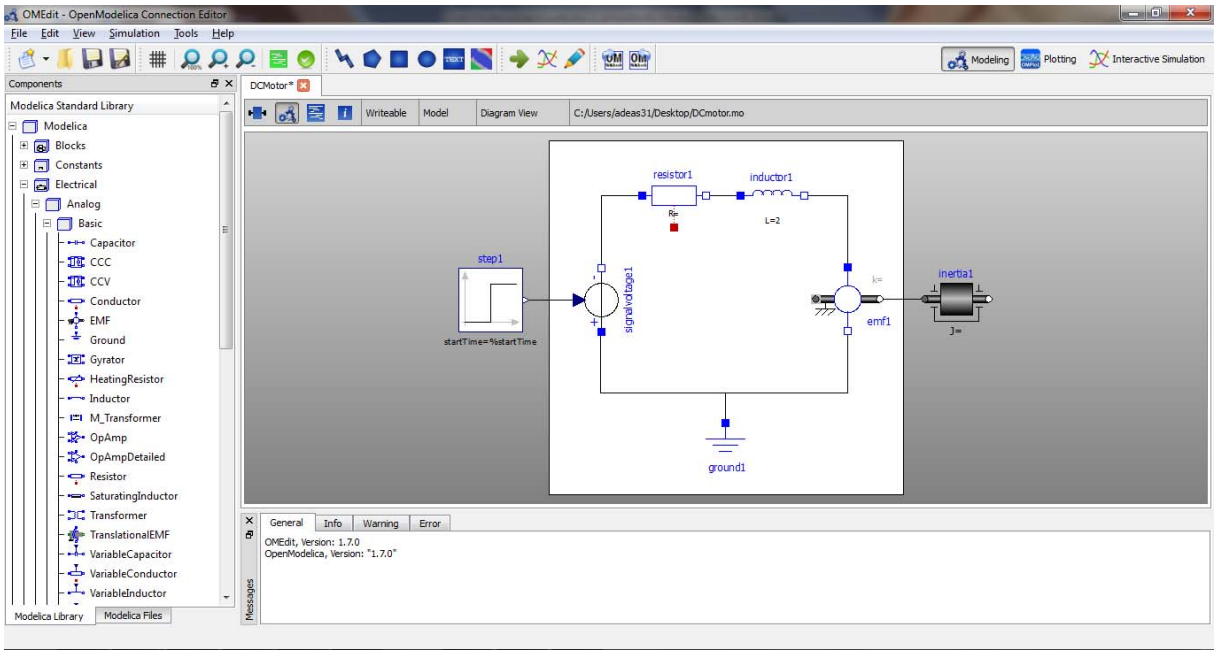

Figure 1-7: DCmotor model after connections.

1.3.5 Plotting Variables from Simulated Models

The instance variables that are candidate for plotting are shown in the right dock window. This window is automatically launched once the user simulates the model; the user can also launch this window manually either from `Simulation > Plot Variables` or by clicking on the `plot icon` from toolbar. It contains the list of variables that are possible to use in an OpenModelica plot. The plot variables window contains a tree structure of variables; there is a checkbox beside each variable. The user can launch the plotted graph window by clicking the checkbox.

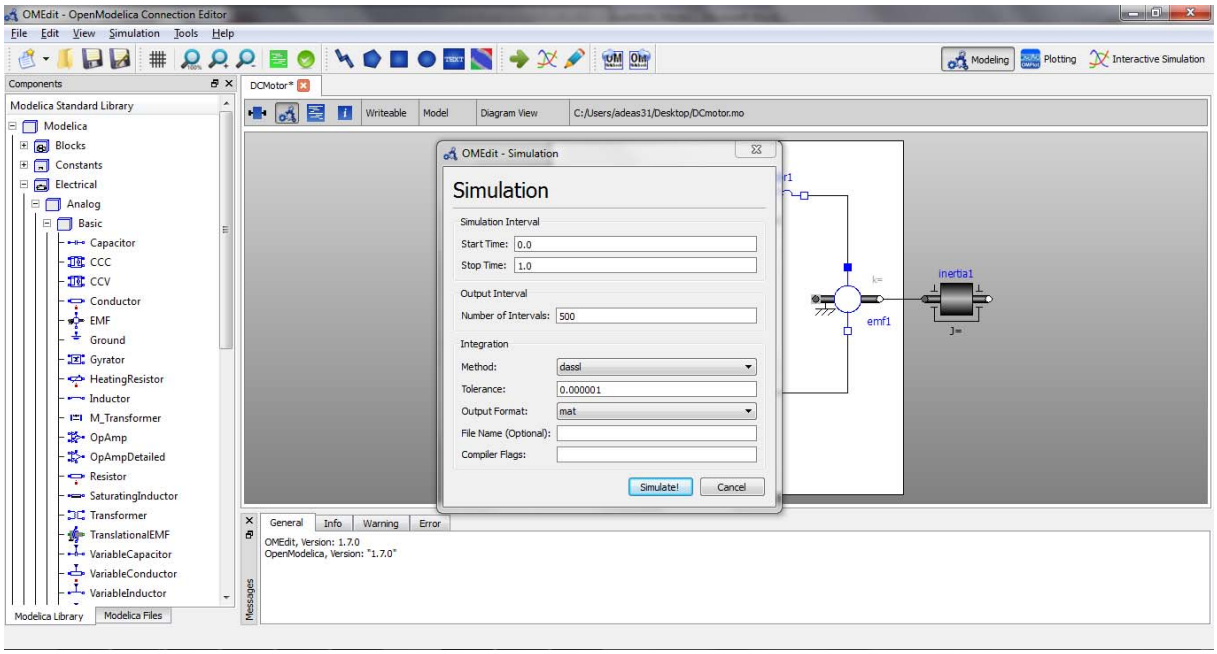

Figure 1-8: Simulation Dialog.

Figure 1-9 shows the complete `DCmotor` model along with the list of plot variables and an example plot window.

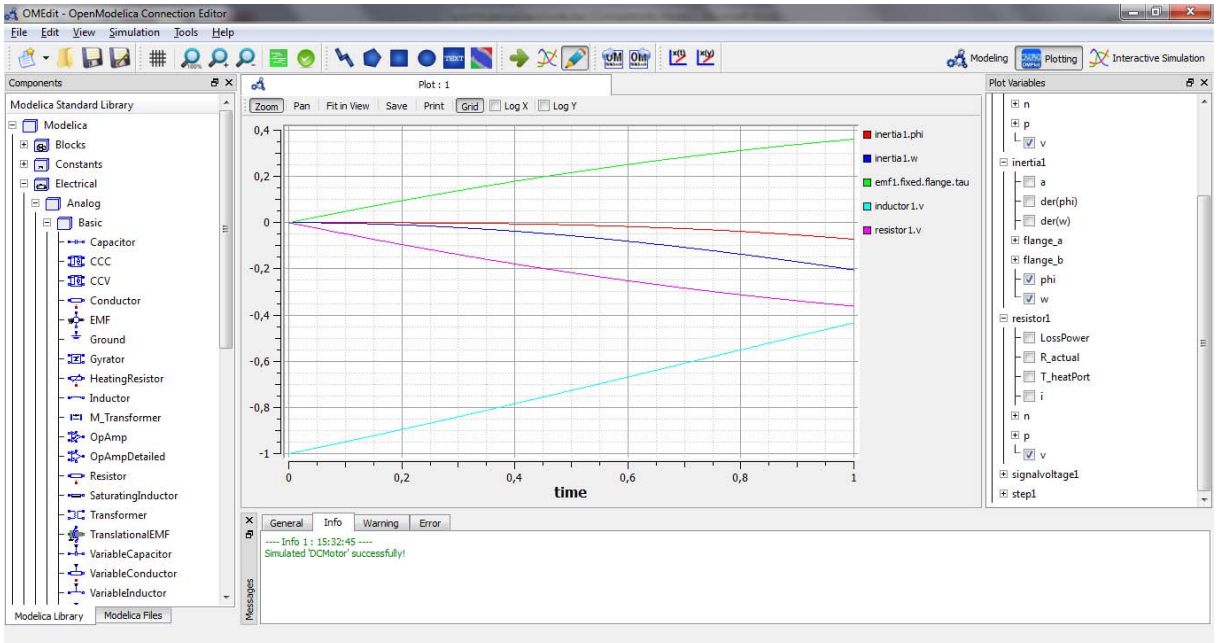

Figure 1-9: Plotted variables.

1.4 How to Create User Defined Shapes – Icons

User can create shapes of their own by using the 6 types of shape tools available in OMEdit.

- *Line Tool* – Draws a line. A line is created with a minimum of two points. In order to create a line, the user first selects the line tool from the toolbar and then click on the `Designer Window`; this will start creating a line. If a user clicks again on the `Designer Window` a new line point is created. In order to finish the line creation, user has to double click on the `Designer Window`.
- *Polygon Tool* – Draws a polygon. A polygon is created in a similar fashion as a line is created. The only difference between a line and a polygon is that, if a polygon contains two points it will look like a line and if a polygon contains more than two points it will become a closed polygon shape.
- *Rectangle Tool* – Draws a rectangle. The rectangle only contains two points where first point indicates the starting point and the second point indicates the ending the point. In order to create rectangle, the user has to select the rectangle tool from the toolbar and then click on the `Designer Window`, this click will become the first point of rectangle. In order to finish the rectangle creation, the user has to click again on the `Designer Window` where he/she wants to finish the rectangle. The second click will become the second point of rectangle.
- *Ellipse Tool* – Draws an ellipse. The ellipse is created in a similar way as a rectangle is created.
- *Text Tool* – Draws a text label.
- *Bitmap Tool* – Draws a bitmap container.

The shape tools are located at the top in the toolbar. See Figure 1-10.

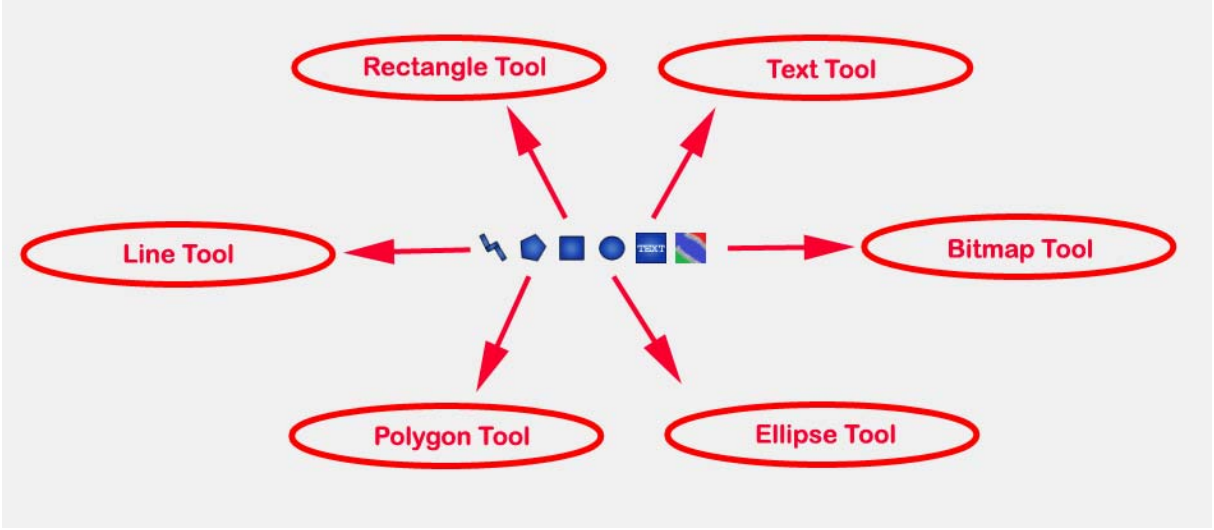

Figure 1-10: User defined shapes.

The user can select any of the shape tools and start drawing on the `Designer Window`. The shapes created on the `Diagram View` of `Designer Window` are part of the diagram and the shapes created on the `Icon View` will become the icon representative of the model.

For example, if a user creates a model with name `testModel` and add a rectangle using the rectangle tool and a polygon using the polygon tool, in the `Icon View` of the model. The model's `Modelica Text` will look like,

```
model testModel
annotation(Icon(graphics = {Rectangle(rotation = 0, lineColor = {0,0,255},
fillColor = {0,0,255}, pattern = LinePattern.Solid, fillPattern =
FillPattern.None, lineThickness = 0.25, extent = {{ -64.5,88},{63, -
22.5}}),Polygon(points = {{ -47.5, -29.5},{52.5, -29.5},{4.5, -86},{ -47.5, -
29.5}}, rotation = 0, lineColor = {0,0,255}, fillColor = {0,0,255}, pattern =
LinePattern.Solid, fillPattern = FillPattern.None, lineThickness = 0.25)}));
```

```
end testModel;
```

In the above code snippet of `testModel`, the rectangle and a polygon are added to the icon annotation of the model. Similarly, any user defined shape drawn on a `Diagram View` of the model will be added to the diagram annotation of the model.

Chapter 2 : OMEdit Views and Windows/Tabs

- ✓ Modeling, Plotting and Interactive Simulation Views.
- ✓ Library Window for Modelica Standard Library.
- ✓ Drawing interface in the form of Designer Window.
- ✓ Plot Variables Window contains the list of instance variables.
- ✓ Messages Window displays the informational, warning and error messages.
- ✓ Documentation Window displays the Modelica annotations based documentation in a QWebView.
- ✓ *Model Browser Window* displays the component hierarchy of the model.
- ✓ New Dialog for creating Modelica models.
- ✓ Simulation Dialog for simulating Modelica models.

2.1 Views

OMEdit has three kinds of views.

2.1.1 Modeling View

This is the default view. This view shows the `Designer Window` and allows users to create their models.

2.1.2 Plotting View

This view is used for showing plot graphs. The user can launch this view anytime by using the views button in the tool bar. This view also becomes active automatically when user simulates the model successfully.

2.1.3 Interactive Simulation View

This view is quite similar to `Plotting View`. One of the primary differences is that `Plotting View` is used to show graphs of pre-built models that cannot be changed. However, in the `Interactive Simulation View` the user can change the values of variables and parameters of the model on the runtime.

2.2 Windows

OMEdit consists of number of windows that shows different views to users.

2.2.1 Library Window

The Modelica Standard Library is automatically loaded in OMEdit and is located on the left dock window. Once a Modelica model has been created then the user can just drag and drop components into the model from the MSL, the `Library Window`. The available libraries in the MSL are:

- Blocks
- Constant
- Electric
- Icons
- Magnetic
- Math
- Mechanics
- SIunits
- Thermal
- UsersGuide
- Utilities

`Library Window` consists of two tabs one shows the Modelica Standard Library and is selected by default the other tab shows the Modelica files that user creates in OMEdit.

2.2.1.1 Viewing Models Description

In order to view the model details, double click the component and details will be opened in `Designer Window`. Alternative way is to right click on the component and press `Show Component`, it will do the same.

2.2.1.2 Viewing Models Documentation

Right click the model in the `Library Window` and select `View Documentation`; it will launch the `Documentation Window`. See Figure 2-1.

2.2.1.3 How to Open an Existing Model?

- Go to `File->Open`. An Open Model Dialog Appears. The user can go to the required file location and click on it to open the model in OMEdit.
- The user can also drag any model from an outside window and drop it into the Main Window of OMEdit to open that model.

2.2.1.4 How to create a Copy of an Existing Model?

To copy a model, the user can simply right click on the model in the `Library Window`, and then select `copy`. To paste the copied model the user can click on any empty space in the `Library Window` and click on the `paste` Option. A copy of the model will be created initialized with a new name.

2.2.1.5 How to Check a Model?

Right click the component in the library window and select `Check`; it will launch the `Check Dialog`. See Figure 2-1.

2.2.1.6 How to Instantiate a Model?

Right click the component in the library window and select `Instantiate Model`; it will launch the Instantiate `Model Dialog`. See Figure 2-1.

2.2.1.7 How to Rename a Model?

Right click the model in the `Library Window` and select `Rename`; it will launch the `Rename Dialog`. See Figure 2-1.

2.2.1.8 How to Delete a Model?

Right click the model in the `library window` and select `Delete`; a popup will appear asking “Are you sure you want to delete?”

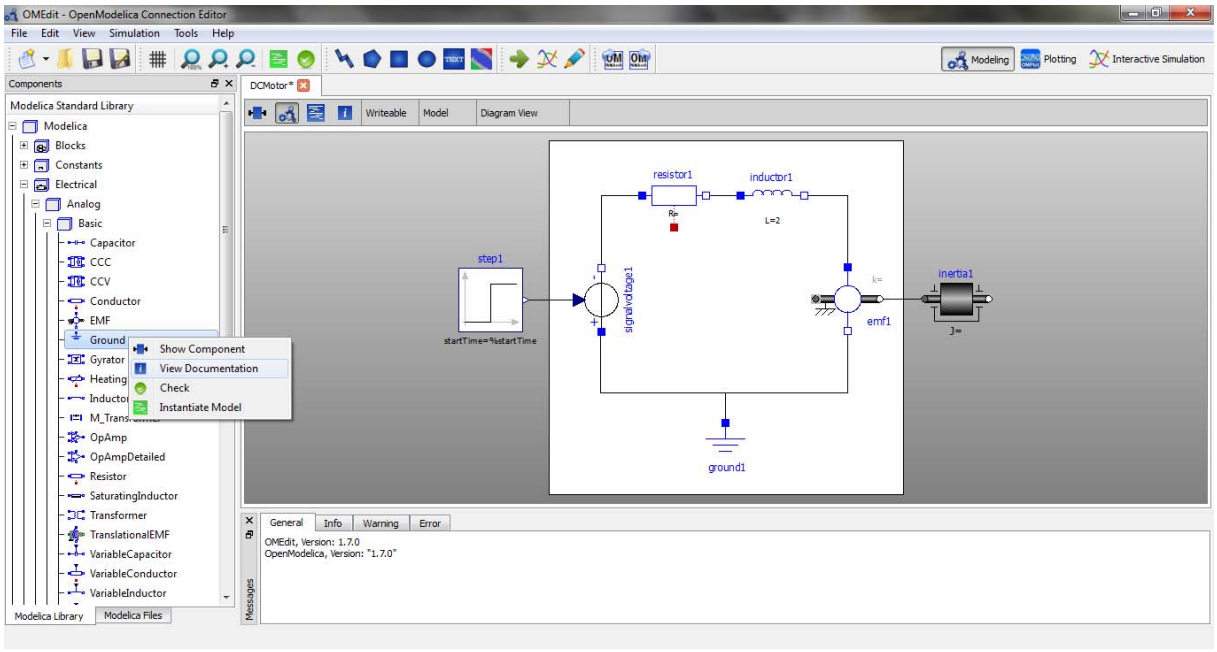

Figure 2-1: Context menu to view component model details.

2.2.2 Designer Window

Designer Window is the main window of OMEdit. It consists of three views,

- *Icon View* - Shows the model icon view.
- *Diagram View* - Shows the diagram of the model created by the user.
- *Modelica Text View* - Shows the Modelica text of the model.

2.2.3 Plot Variables Window

The right dock window represents the `Plot Variables Window`. It consists of a tree containing the list of instance variables that are extracted from the simulation result. Each item of the tree has a checkbox beside it. The user can click on the check box to launch the plot graph window. The user can add/remove the variables from the plot graph window by marking/unmarking the checkbox beside the plot variable.

2.2.4 Messages Window

`Messages Window` is located at the bottom of the application. The Messages Window consists of 4 types of messages,

- *General Messages* – Shown in black color.
- *Informational Messages* – Shown in green color.
- *Warning Messages* – Shown in orange color.
- *Error Messages* – Shown in red color.

2.2.5 Documentation Window

This window is shown when a user right clicks the component in the library window and selects `View Documentation`. This shows the OpenModelica documentation of components in a web view. All

externals links present in the documentation window are opened in the default browser of the user. All local links are opened in the same window. Figure 2-2 shows the `Documentation Window` view.

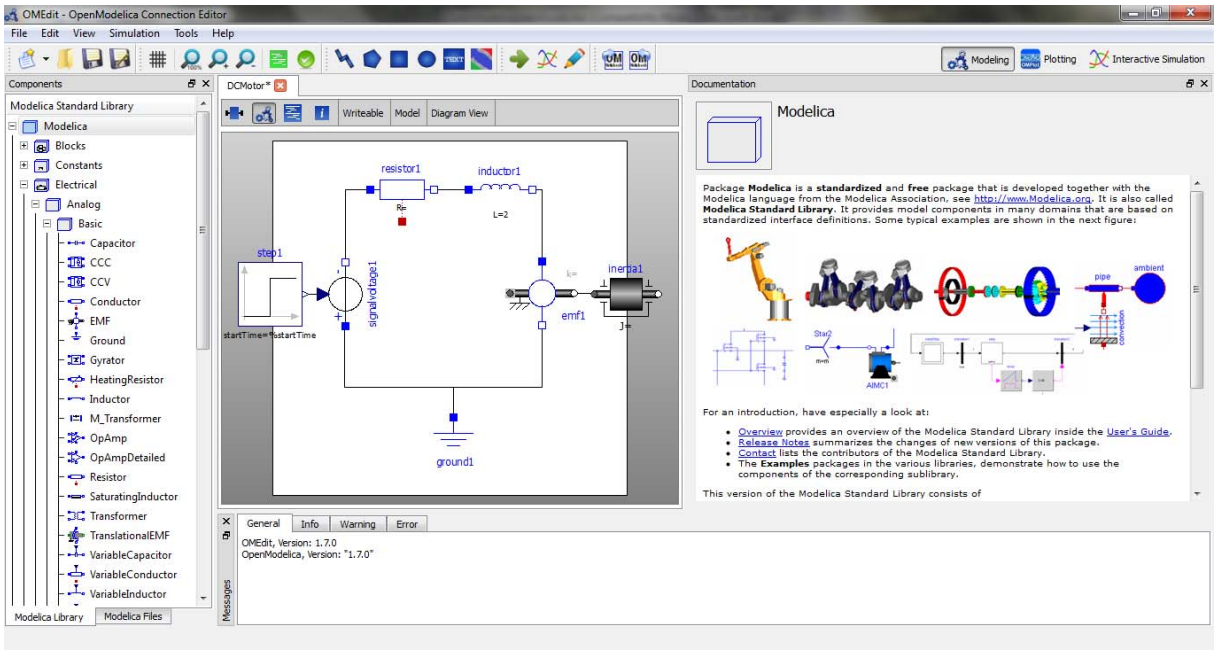

Figure 2-2: Documentation Window.

2.2.6 Model Browser Window

The Model Browser Window is located on the left bottom dock window below the Library Window. It lays the outline of the currently opened model and show all the component heirarchy in a tree format. See Figure 2-3

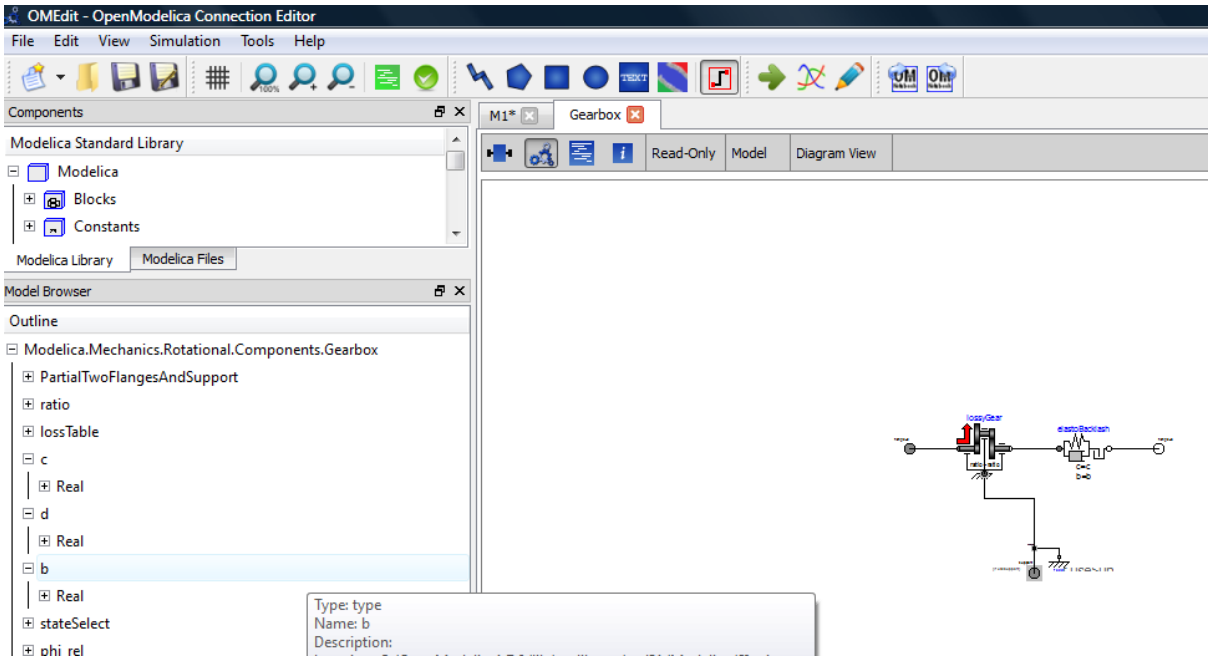

Figure 2-3: Model Browser Window.

2.3 Dialogs

Dialogs are the sub kind of windows that are not visible by default. The user has to launch them or they will automatically appear due to some user action.

2.3.1 New Dialog

The `New Dialog` can be launch from `File > New > Model Type`**.** Model type can be `model`, `class`, `connector`, `record`, `function` and `package`.

2.3.2 Simulation Dialog

`Simulation Dialog` can be launched either from `Simulation > Simulate` or by clicking on the `Simulate!` button in the toolbar. Figure 1-8 shows a simulation dialog. The simulation dialog consists of simulation variables. You can set the value of any variable, depending on the simulation requirement. Simulation variables are,

- Simulation Interval
  - Start Time
  - Stop Time
- Output Interval
  - Number of Intervals
  - Output Interval
- Integration
  - Method
  - Tolerance
  - Fixed Step Size
  - File Name
  - Compiler Flags

2.3.3 Model Properties Dialog

The models that are placed in the `Designer Window` can be modified by changing the properties. In order to launch the `Model Properties Dialog` of a particular model right click the model and select `Properties`. See Figure 2-4. The properties dialog contains the name of the model, class name the model belongs to and the list of parameters of the component.

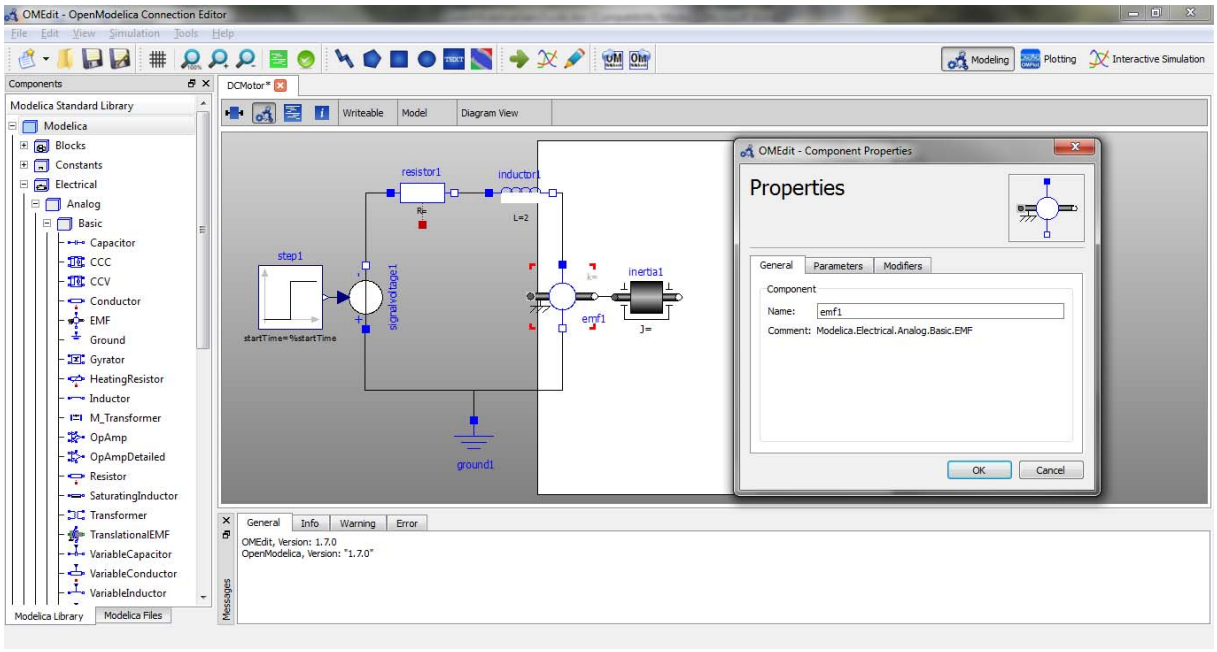

Figure 2-4: Properties Dialog.

2.3.4 Model Attributes Dialog

Right click the model placed in the `Designer Window` and select `Attributes`. It will launch the attributes dialog. Figure 2-5 shows the `Model Attributes Dialog`.

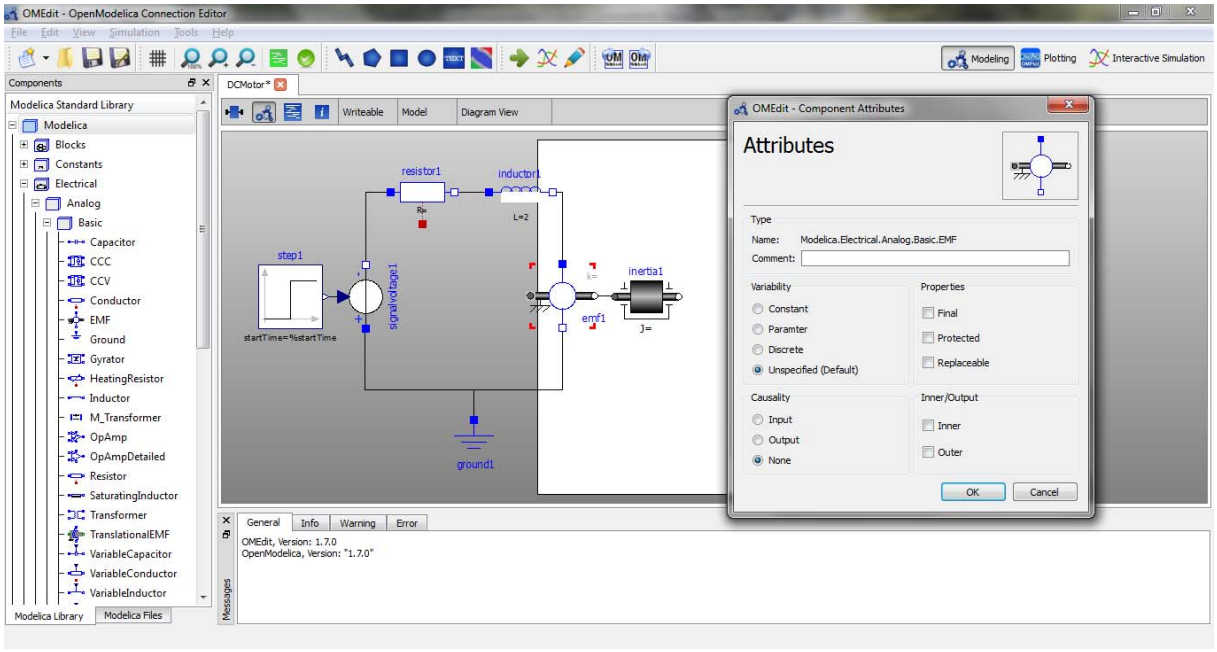

Figure 2-5: Attributes Dialog.

Chapter 3 : Interactive Simulation in OMEdit

- ✓ OpenModelica Interactive subsystem
- ✓ Starting Interactive Simulation.
- ✓ Interactive Simulation options.

3.1 OpenModelica Interactive

OMEdit uses the OpenModelica Interactive (OMI) subsystem to perform the interactive plotting. The OMI uses the TCP/IP technique to transfer data back and forth. OMEdit connects with OMI through TCP sockets.

3.2 Invoking Interactive Simulation

`Interactive Simulation Dialog` can be launched either from `Simulation > Interactive Simulation` or by clicking on the `Interactive Simulation!` button in the toolbar. Interactive Simulation Dialog looks similar to the Simulation Dialog but it differs in functionality, instead of performing normal pre-built simulation it performs online interactive simulation where simulation responds in real-time to user input and changes to parameters.

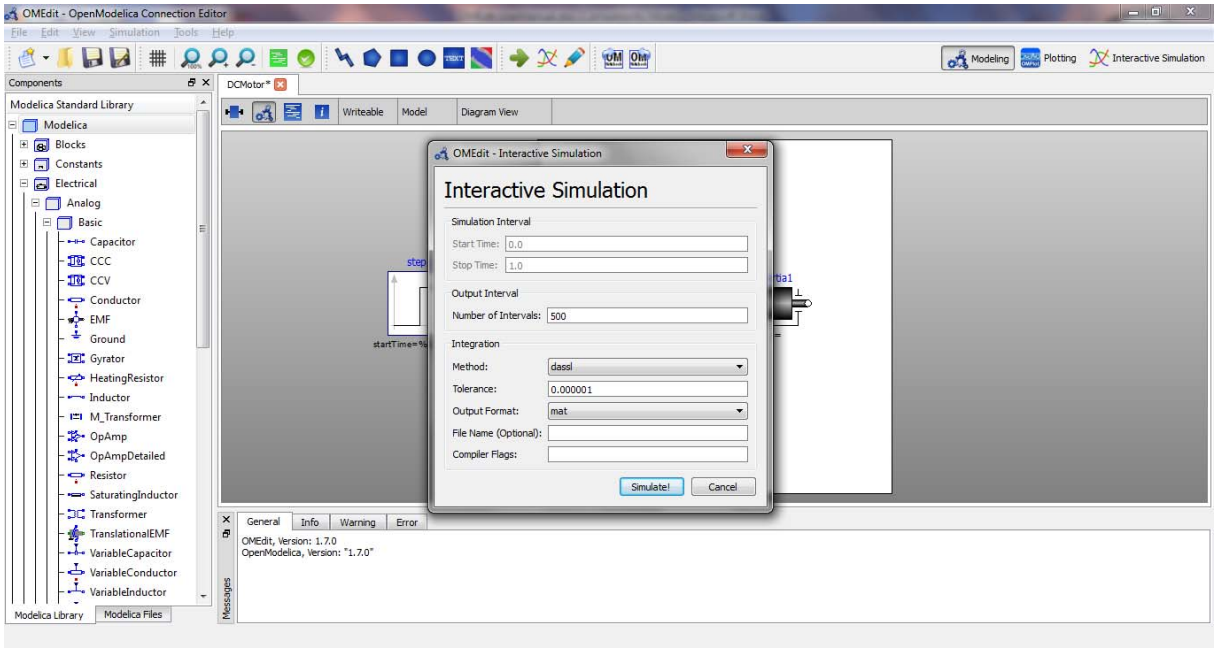

Figure 3-1: Interactive Simulation Dialog.

3.3 Interactive Simulation View

Once your model was successfully built using the `Interactive Simulation Dialog`, the `Interactive Simulation View` will become active automatically. `Interactive Simulation View` contains,

- *Graph* – It contains a graph which is used to display the values of selected variables over the time.
- *Parameters* – The parameters of the model are shown on the right top section with the default values.
- *Variables* – The right bottom section contains the list of variables that user can select for interactive plotting.
- *Initialize Button* – This button is used to send the information of changed parameters and checked variables to the OpenModelica Interactive subsystem.
- *Start Button* – Once the parameters and variables are initialized and sent to the OMI. Then the user can click on start button and start the interactive plotting.

- *Pause Button* – This button pauses the running interactive plotting.
- *Stop Button* – Clears everything but does not remove the connection with OMI. After clicking stop button user has to reinitialize everything and start the interactive plotting again.
- *Shut Down Button* – Disconnects from OMI and closes the interactive simulation session.
- *Show OMI Log Button* – Pops up a log window which displays the messages exchanged between OMEdit and OMI.

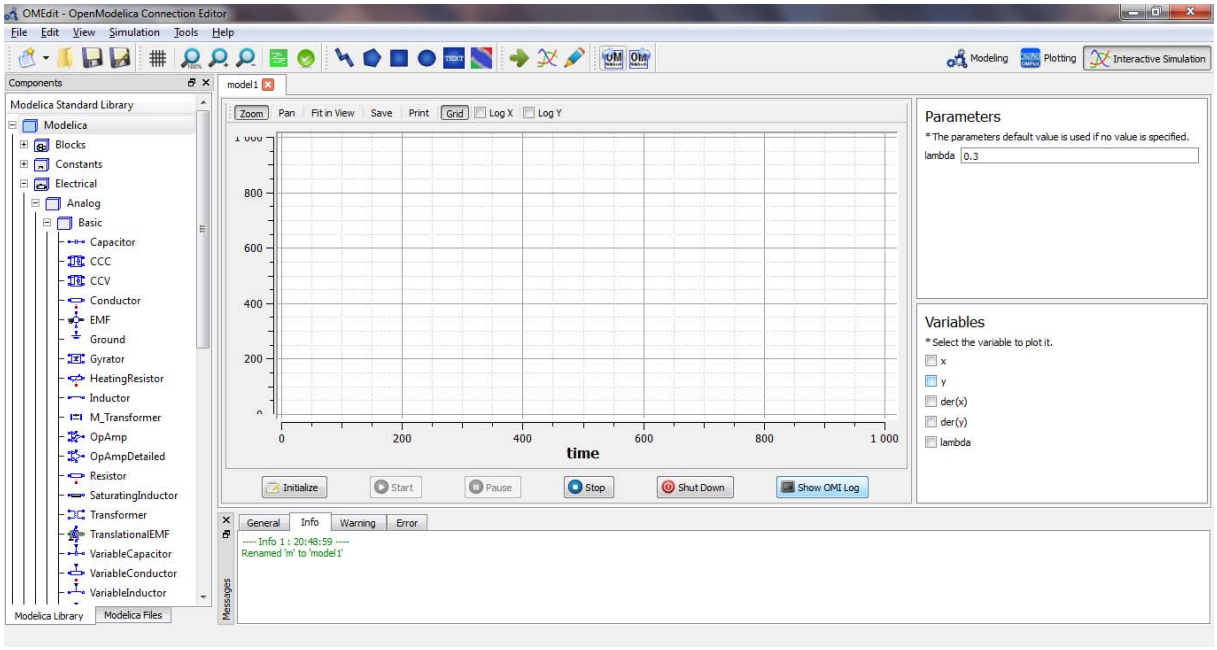

Figure 3-2: Interactive Simulation View.