# Coupling OpenModelica and GenOpt

Guidelines

*Version 2.1*

16th August 2021

**Vishak Dudhee and Dr Vladimir Vukovic**
v.dudhee@tees.ac.uk | v.vukovic@tees.ac.uk

School of Computing, Engineering & Digital Technologies
Teesside University, United Kingdom

# CONTENTS

The software version used in the coupling process, the Modelica library version and the operating system specifications are:

**Software:**

- OpenModelica v1.17.0 (64-bit)
  - Connected to OMSimulator v2.0.0.post284-gc8ec782-mingw
- GenOpt 3.1.1

**Modelica Library:**

- Modelica Standard Library V4.0.0 (Version Date: 2020-06-04)

**Operating System:**

- Windows 10 Home
  - Version: 20H2
  - System type: 64-bit operating system, x64-based processor
  - OS build: 19042.867

# 1 OpenModelica

## 1.1 Creating a Model

A model of a simple heating system, as illustrated in Figure 1, was created based on GenOpt's example `ObjectiveFunction.mo` to demonstrate the coupling of OpenModelica and GenOpt.

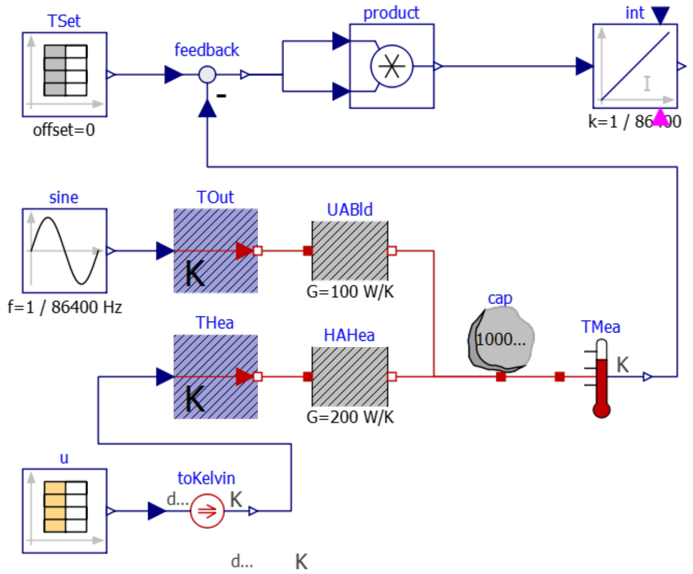


*Figure 1: Model in OpenModelica*

The models from the Modelica Standard Library (MSL) are the following:

```
Modelica.Blocks.Sources.TimeTable
Modelica.Blocks.Sources.Sine
Modelica.Blocks.Math.Feedback
Modelica.Blocks.Math.Product
Modelica.Blocks.Continuous.Integrator
Modelica.Blocks.Types.Init.InitialState
Modelica.Thermal.HeatTransfer.Sources.PrescribedTemperature
Modelica.Thermal.HeatTransfer.Components.ThermalConductor
Modelica.Thermal.HeatTransfer.Components.HeatCapacitor
Modelica.Thermal.HeatTransfer.Sensors.TemperatureSensor
Modelica.Thermal.HeatTransfer.Celsius.ToKelvin
Modelica.Blocks.Sources.CombiTimeTable
```

In the model, the `Modelica.Thermal.HeatTransfer.Components.HeatCapacitor` parameters were as illustrated in Figure 2.

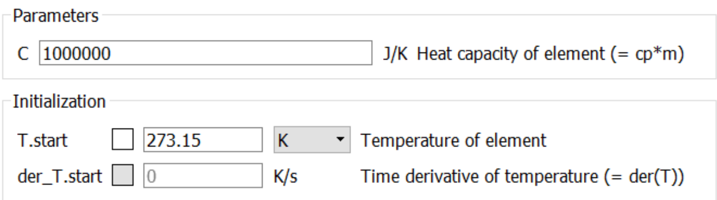


*Figure 2: HeatCapacitor Parameters*

Set `u.y` to `[1]`, while connecting `Modelica.Blocks.Sources.CombiTimeTable` to `Modelica.Thermal.HeatTransfer.Celsius.ToKelvin` as illustrated in Figure 3.

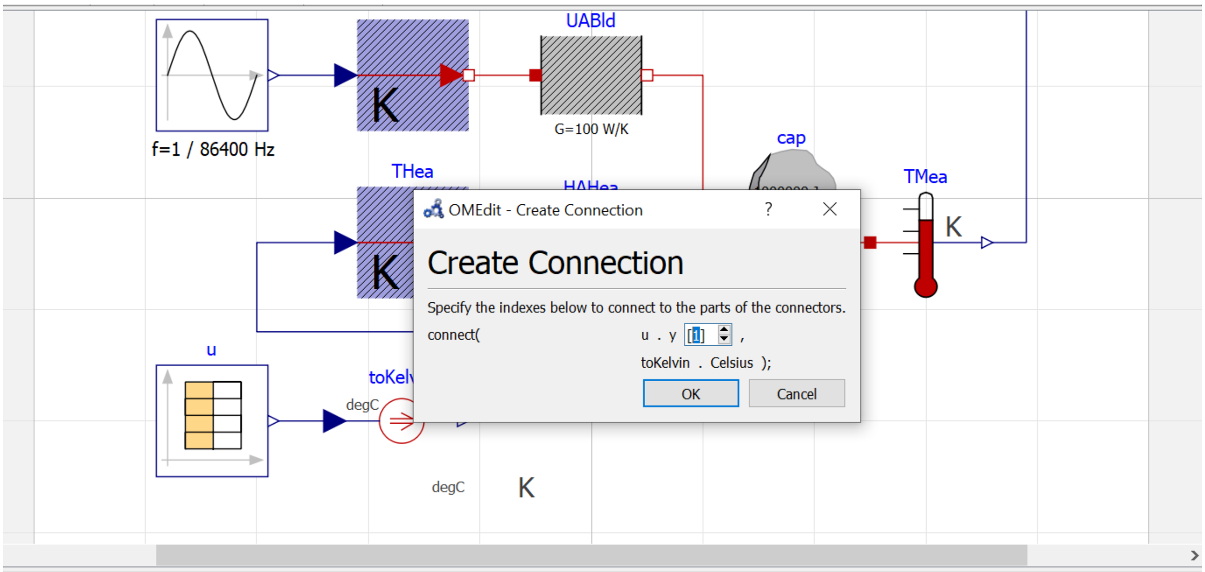


*Figure 3: Creating Connection*

To verify that there are no modelling errors and the model behaves as intended, please attempt to run the simulation.

**Additional Information**

The working directory path can be changed to the desired location. Click on `Tools> Options> General` and change the `Working Directory` path.

Set the simulation time to the desired simulation period. Click on `simulation> Simulation Setup> General` and set the `Start Time:` and `Stop Time:` of the simulation in seconds. For example, 24 hours is equivalent to 86400 seconds (secs).

The `Modelica.Blocks.Sources.CombiTimeTable` required a file with data to fill the table, and in the above context it requires a schedule file. In the model verification process, change the `fileName` path to the "modelicaSchedule.txt" file location.

## 1.2 Editing the Model

After having created the model, for the model to generate `results.txt` file the model code must be edited. To view the model code, press on 'view text' from the model option bar.

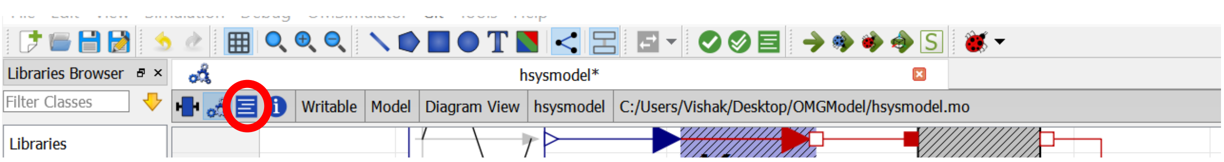


*Figure 4: Finding the 'view text' option*

### 1.2.1 Load File Command

The following command is then used to input the `modelicaSchedule.txt` and create the `result.txt` file. In this example, `modelicaSchedule.txt` is the file name of the schedule used, and the `result.txt` is the file name of the result file we wish to generate.

```
model hsysmodel
  parameter String inputFileName = "modelicaSchedule.txt" "File on which data
is present";
  parameter String resultFileName = "result.txt" "File on which data is
present";
  parameter String header = "Objective function value" "Header for result
file";
```

* `hsysmodel` is the model name.

### 1.2.2 Print Result File

After adding the load file command, the following command must be added before the `equation` to generate the result file.

```
initial algorithm
  if resultFileName <> "" then
    Modelica.Utilities.Files.removeFile(resultFileName);
  end if;
  Modelica.Utilities.Streams.print(fileName = resultFileName, string =
header);
equation
```

The following code is then added after the equation to add the objective function variable to the `result.txt` file in the required format.

```
equation
  when terminal() then
    Modelica.Utilities.Streams.print("f(x) = " + String(int.y, format =
"1.16f"), resultFileName);
  end when;
```

## 1.3 Create a Script

### 1.3.1 Script Creation Process

After the model is edited to generate a result file, a script is then created to run the model. The script file needs to be able to load the Modelica Standard Library and the model and then be able to simulate the model. The script file should have a `'.mos'` extension. The `'.mos'` extension file can be created by simply creating and blank text file and then altering the file extension from `'.txt'` to `'.mos'`.

For the model hsysmodel.mo the following commands were used in the script file. The simulation stop time was set to 86400 seconds (equivalent to 24 hours) and the number of intervals to 86400 so that the simulation interval is every second.

```
loadModel(Modelica);
loadModel(hsysmodel);
getErrorString();
simulate(hsysmodel,stopTime=86400.00,numberOfIntervals=86400);
getErrorString();
```

For the script file to successfully run once created, the model needs to be copied to the OpenModelica Library.

The file path of the OpenModelica Library depends on where the software was installed, but usually, it is:

```
C:/Program Files/OpenModelica1.17.0-64bit/lib/omlibrary
```

### 1.3.2 Run Script (Optional)

After having created the script and copied the model file to the OpenModelica Library, using the following command in the command prompter (cmd.exe), the script can be tested.

1. Open the command prompter.
2. Change the working directory to the path where the schedule and script files are located.
3. Run the following command:

```
C:\WorkingDirectory> C:\Progra~1\OpenModelica1.17.0-64bit\bin\omc script.mos
```

4. If the script is working, a result.txt file will be generated in the working directory.

Note: In some cases, writing `Program Files` sends an error message:

```
'C:\Program' is not recognised as an internal or external command,
operable program or batch file.
```

Therefore `Progra~1` should be used instead.

# 2 GenOpt

## 2.1 Coupling Files

For GenOpt to work, the following files are needed:

- Simulation input template file
- Initialisation file
- Configuration file
- Command file

The command file must be in the format as prescribed by GenOpt. The simulation input template file shows the input file structure, such as the combi table schedule. More details on creating the command file can be found in the GenOpt user manual. To allow the simulation file generated by OpenModelica to be optimised by GenOpt, the initialisation file and the configuration file require editing.

The `command.txt`, `initialisation.ini` and `SimulationInputTemplate.txt` files need to be in the same folder where the `model.mo` is located. The folder used in the example was called ‘coupOMG’; therefore, the following files were copied to the folder:

- Simulation input template – `modelicaSchedule.txt`
- Initialisation file – `GenoptWin.ini`
- Command file – `command.txt`

The configuration file location is specified in the initialisation file. Therefore, past the `configuration.cfg` file in the specified location.

- Configuration file - `OMGWin.cfg`

## 2.2 INITIALISATION FILE

The initialisation file contains the following codes:

```
/* GenOpt initialisation file
   Operating system: Windows 10
   Vladimir Vukovic and Vishak Dudhee
   Teesside University
   v.vukovic@tees.ac.uk, 24-03-2021
*/
Simulation {
  Files {
    Template {
      File1 = OpenModelicaScheduleTemplate.txt;
      File2 = script.mos;
          File3 = hsysmodel.mo;
    }
    Input {
      File1 = modelicaSchedule.txt;
      File2 = script.mos;
          File3 = hsysmodel.mo;
    }
    Log {
      File1 = hsysmodel.log;
          Path1 = Simulation.Files.Input.Path1;
          SavePath1 = Simulation.Files.Input.Path1;
    }
    Output {
      File1 = result.txt;
    }
    Configuration {
      File1 = "..\\..\\..\\cfg\\OMGWin.cfg";
    }
  }
  ObjectiveFunctionLocation{
        Delimiter1 = "f(x) =" ;
        Name1      = "f(x)";
  }

} // end of section Simulation
Optimisation {
  Files {
    Command {
      File1 = command.txt;
    }
  }
} // end of configuration file
```

## 2.3 CONFIGURATION FILE

The configuration file contains the following code:

```
/* GenOpt configuration file for
   OpenModelica on Windows 10
   Vladimir Vukovic and Vishak Dudhee
   Teesside University
   v.vukovic@tees.ac.uk, 24-03-2021
*/

// Error messages of the simulation program.
SimulationError
```

```
{
    ErrorMessage = "Integration terminated before reaching end point";
}

// Number format for writing the simulation input files.
IO
{
    NumberFormat = Double;
}

SimulationStart
{
  Command = "cmd /x /c \"start /D%Simulation.Files.Input.Path1% /WAIT /MIN
C:\\\"Progra~1\"\\OpenModelica1.17.0-64bit\\bin\\omc script.mos \" " ;
  WriteInputFileExtension = true;
}
```

# 3 EXAMPLE – HSYSMODEL

## 3.1 SCRIPT - COMMAND PROMPTER

The script file was created and run using command prompter to validate the script.

```
C:\Users\Vishak\Documents\GenOpt\coupOMG>   C:\Progra~1\OpenModelica1.17.0-
64bit\bin\omc script.mos
true
true
""
record SimulationResult
    resultFile                                                              =
"C:/Users/Vishak/Documents/GenOpt/coupOMG/hsysmodel_res.mat",
    simulationOptions    =    "startTime     =    0.0,    stopTime    =    86400.0,
numberOfIntervals   =   86400,   tolerance   =   1e-06,   method   =   'dassl',
fileNamePrefix  =  'hsysmodel',  options  =  '',  outputFormat  =  'mat',
variableFilter = '.*', cflags = '', simflags = ''",
    messages = "stdout              | info     | ... loading \"tab1\" from
\"modelicaSchedule.txt\"
LOG_SUCCESS          | info      | The initialization finished successfully
without homotopy method.
LOG_SUCCESS        | info     | The simulation finished successfully.
",
    timeFrontend = 0.1563498,
    timeBackend = 0.0179131,
    timeSimCode = 0.0411051,
    timeTemplates = 0.0315496,
    timeCompile = 4.1413164,
    timeSimulation = 0.1965352,
    timeTotal = 4.5850856
end SimulationResult;
"Warning: The model contains alias variables with redundant start and/or
conflicting nominal values. It is recommended to resolve the conflicts,
because otherwise the system could be hard to solve. To print the conflicting
alias sets and the chosen candidates please use -d=aliasConflicts.
Warning: The initial conditions are not fully specified. For more information
set -d=initialization. In OMEdit Tools->Options->Simulation->OMCFlags, in
OMNotebook call setCommandLineOptions(\"-d=initialization\").
"
```

## 3.2 OpenModelica Simulation

The hsysmodel.mo was simulated, and the variable int.y and u.y[1] against time were plotted as shown in Figure 5 to identify the simulation's objective function value.

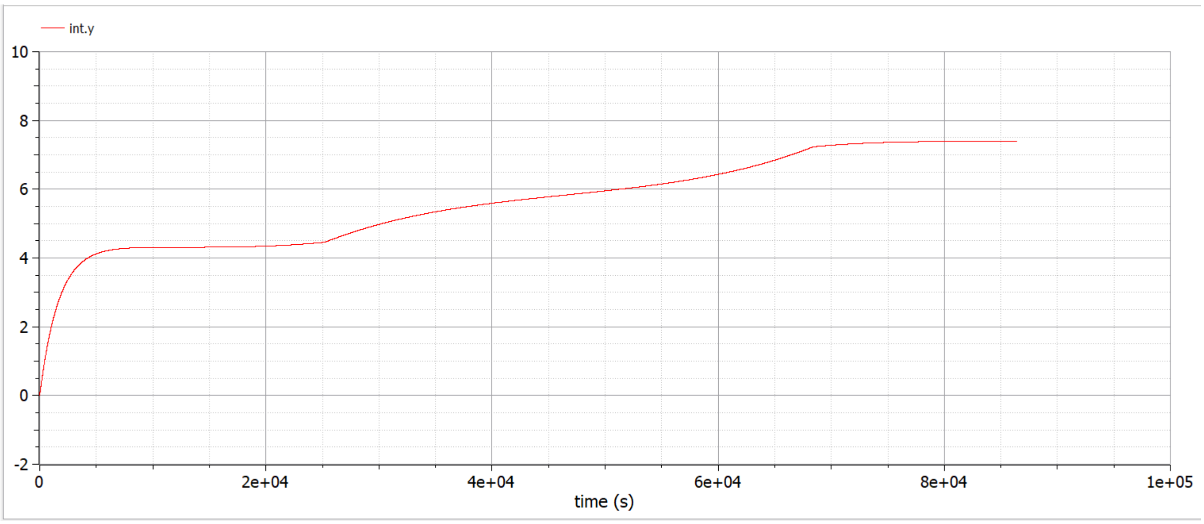


*Figure 5: Simulation output*

The final objective function value is: f(x) = 7.3941561989593971

## 3.3 GenOpt Optimisation

The model was then optimised through GenOpt, and the cost function decreased as illustrated in Figure 6.

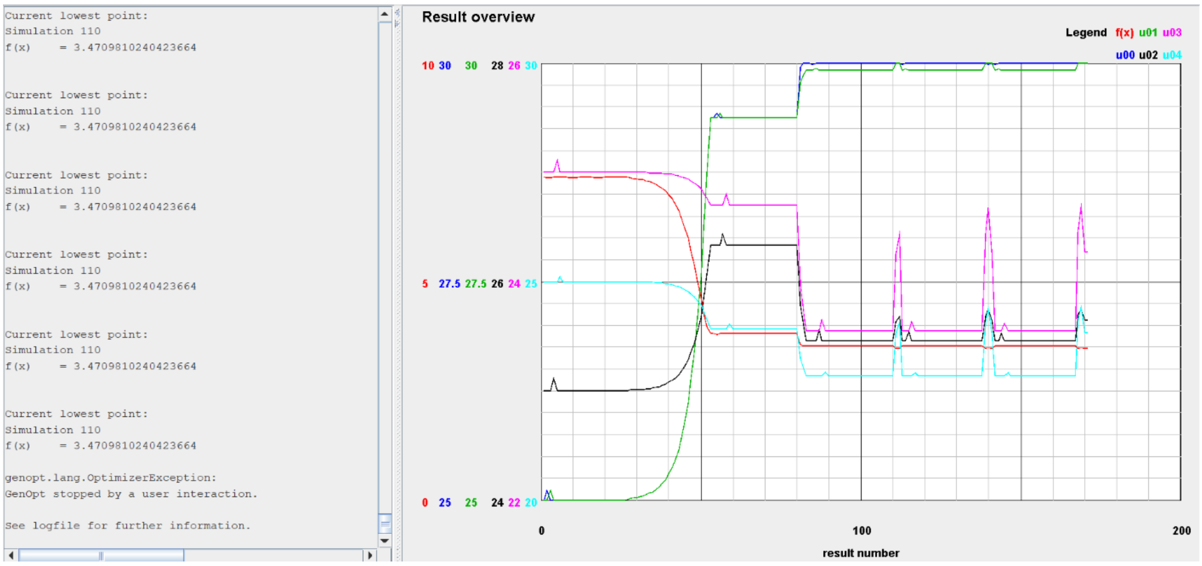


*Figure 6: Optimisation result*

Minimum: f(x*) = 3.4709810240423664

The Discrete Armijo Gradient optimisation algorithm was used with the following variable values:

```
Algorithm{
  Main = DiscreteArmijoGradient;
  Alpha = 0.5;
```

```
  Beta  = 0.8;
  Gamma = 0.1;
  K0    = 10;
  KStar = 0;
  LMax  = 50;
  Kappa = 50;
  EpsilonM = 0.05;
  EpsilonX = 0.05;
```

***Please Note:*** *The Antivirus installed on the computer was temporarily disabled throughout the process to prevent any interference during optimisation.*