Summer Fellowship Report

On

Modelling and Simulation in OpenModelica

Submitted by

Sarmitha V D

Under the guidance of

Prof. Kannan M. Moudgalya
Chemical Engineering Department
IIT Bombay

Acknowledgment

I would like to express my profound gratitude to Dr. Kannan M. Moudgalya, Professor, Department of Chemical Engineering, IIT Bombay, for providing me with this wonderful opportunity to work as an intern at the FOSSEE Summer Fellowship. I want to express my special thanks to my mentor, Mr. Rahul Paknikar, for the time and efforts he provided throughout the internship. Your useful advice and suggestions were beneficial to me. I am eternally grateful to the whole FOSSEE team.

Sarmitha V D

Contents

Chapter 1

Introduction

1.1 OpenModelica

OpenModelica is a free and open-source modelling and simulation environment, designed for both industrial and academic applications. Operating with the powerful "Modelica" modelling language, it adopts an equation-oriented approach, excelling in modelling, simulating, optimizing, and analysing complex steady-state and dynamic systems.

Modelica facilitates the expression of systems through equations, and OpenModelica takes this to the next level. It compiles expressions, equations, functions, and algorithms into efficient C code, seamlessly integrating them with utility functions, a run-time library, and a numerical Differential-Algebraic Equation (DAE) solver.

For a user-friendly experience, OpenModelica offers the OMEdit (OpenModelica Connection Editor), an integrated Graphical User Interface (GUI) that supports graphical modelling and editing. OMEdit has several libraries catering to domains like Electrical, Magnetic, Math, Thermal, and more. It enables the creation of custom models, editing connections be-

tween interfaces, and even plotting graphs for simulated model parameters.

In essence, OpenModelica isn't just a tool; it's an open gateway for expressing, optimizing, and simulating the intricate dance of complex systems.

1.2 Work Environment

Operating System: Ubuntu 22.04.2 LTS (Jammy Jellyfish)
Processor: Intel i7 11th gen
Ram: 16 GB
Other Software: Visual Studio Code, OpenModelica, OMEdit, Git

Chapter 2

OMChemSim

2.1 Description

The OMChemSim, short for OpenModelica Chemical Simulator, constitutes a collection of chemical system component models expressed in the Modelica language. These models find application in chemical process simulations, ranging from constructing binary phase envelopes to simulating entire process plant flow sheets.

2.2 Resolutions of the errors

2.2.1 Declarations of models as records

Defining as a record aligns, intending to encapsulate data related to the chemical compound without introducing unnecessary complexities associated with modelling dynamic behaviour. It improves code organization, readability, and maintainability.

The model files (within Simulator.Files.ChemsepDatabase) are GeneralProperties, Benzene, Toluene:

GeneralProperties.mo:

```
within Simulator.Files.ChemsepDatabase;

-    model GeneralProperties "Model to declare
   the variables for thermophysical properties"

+    record GeneralProperties "Model to declare
   the variables for thermophysical properties"

extends Modelica.Icons.Record;

    ...
```

Benzene.mo:

```
within Simulator.Files.ChemsepDatabase;

-    model Benzene

+    record Benzene

extends Modelica.Icons.Record;

    ...
```

Toluene.mo:

```
within Simulator.Files.ChemsepDatabase;

-    model Toluene
```

```
+    record Toluene

    ...

end Toluene;
```

2.2.2 Package name corrections in the import statements

The model files (within Simulator.Files.ThermodynamicPackages) are PoyntingCF, GrayonStreed, NRTL and (within Simulator.UnitOperations) HeatExchanger:

```
...

-    import Simulator.Files.
  Thermodynamic_Functions.*;

+    import Simulator.Files.
  ThermodynamicFunctions.*;

...
```

2.2.3 Unit corrections

Simulator/Files/ChemsepDatabase/GeneralProperties.mo:

The unit expression “Pa s” is invalid because it lacks a period to signify multiplication.

```
...

parameter Real VapCp[6] (each unit="J/kmol/K")
  "Ideal Gas Heat Capacity Coefficients";

-    parameter Real LiqVis[6] (each unit="Pa s")
    "Liquid Viscosity Coefficients";
-    parameter Real VapVis[6] (each unit="Pa s")
    "Vapor Viscosity Coefficients";

+    parameter Real LiqVis[6] (each unit="Pa.s")
    "Liquid Viscosity Coefficients";
+    parameter Real VapVis[6] (each unit="Pa.s")
    "Vapor Viscosity Coefficients";

parameter Real LiqK[6] (each unit="W/m/K") "
  Liquid Thermal Conductivity Coefficients";

...
```

Simulator/Streams/MaterialStream.mo:

```
...

Real Fm_pc[3, Nc](each unit = "kg/s", each min
```

```
   = 0, each start = Fg) "Component mass flow in
    phase";

-    Real Cp_p[3](each unit = "kJ/[kmol.K]",
   start={Hmixg,Hliqg,Hvapg}) "Phase molar
   specific heat";
-    Real Cp_pc[3, Nc](each unit = "kJ/[kmol.K
   ]") "Component molar specific heat in phase";

+    Real Cp_p[3](each unit = "kJ/kmol.K",start
   ={Hmixg,Hliqg,Hvapg}) "Phase molar specific
   heat";
+    Real Cp_pc[3, Nc](each unit = "kJ/kmol.K")
   "Component molar specific heat in phase";

Real H_p[3](each unit = "kJ/kmol",start={Hmixg,
   Hliqg,Hvapg}) "Phase molar enthalpy";
Real H_pc[3, Nc](each unit = "kJ/kmol") "
   Component molar enthalpy in phase";

-    Real S_p[3](each unit = "kJ/[kmol.K]") "
   Phase molar entropy";
-    Real S_pc[3, Nc](each unit = "kJ/[kmol.K]")
    "Component molar entropy in phase";

+    Real S_p[3](each unit = "kJ/kmol.K") "Phase
    molar entropy";
+    Real S_pc[3, Nc](each unit = "kJ/kmol.K") "
   Component molar entropy in phase";
```

```
 Simulator.Files.Interfaces.matConn In(Nc = Nc)
   annotation(Placement(visible = true,
   transformation(origin = {-100, 0}, extent =
   {{-10, -10}, {10, 10}}, rotation = 0),
   iconTransformation(origin = {-100, 0},
   extent = {{-10, -10}, {10, 10}}, rotation =
   0)));

...
```

2.2.4 Variable declarations

This is done as the variables were not found within the scope of the files.

Simulator/Files/Models/ReactionManager/ EquilibriumReaction.mo:

```
...

Real Hf_c[Nc];
Real Hr_r[Nr];

+    Integer BC_r[Nr];

//Equilibrium Constant
Real K[Nr](start=xliqg);

...
```

Simulator/Files/Models/ThermodynamicPackages/NRTL.mo:

```
...

model NRTL
import Simulator.Files.ThermodynamicFunctions
  .*;

+   import data = Simulator.Files.
  ChemsepDatabase;
+   parameter Integer Nc;
+   parameter Simulator.Files.ChemsepDatabase.
  GeneralProperties C[Nc];
+   Real x_pc[Nc, Nc];
+   Real Pdew;
+   Real T(unit = "K");
+   Real P;

Simulator.Files.Models.gammaNRTL Gma(Nc = Nc, C
   = C, x_c = x_pc[2, :], T = T), GmaDew(Nc =
  Nc, C = C, x_c = xliqdew_c, T = T), GmaBubl(
  Nc = Nc, C = C, x_c = x_pc[1, :], T = T);

...
```

Simulator/UnitOperations/ConversionReactor.mo:

```
...
```

```
parameter Integer Nc "Number of components"
  annotation(Dialog(tab = "Reactor
  Specifications", group = "Component
  Parameters"));

+    parameter Integer Nr "Number of Reactions";

parameter String CalcMode = "Isothermal" "
  Required mode of operation: ''Isothermal'', '
  Define_Out_Temperature'', ''Adiabatic''"
  annotation(Dialog(tab = "Reactor
  Specifications", group = "Calculation
  Parameters"));

...

Real Fout_cr[Nc, Nr](each unit = "mol/s") "
  Molar flor rate of components after each
  reaction";

+    Real Coef_cr[Nc, Nr];
+    Integer BC_r[Nr];
+    Real Hr_r[Nr];

//=============================================================

//Instanstiation of Connectors
Simulator.Files.Interfaces.matConn In(Nc = Nc)
  annotation(
```

```
...
```

Simulator/UnitOperations/ShortcutColumn.mo:

```
...

Real xvapcond_c[Nc](each unit = "-", each min =
    0, each max = 1, start = yg)"Component mole
   fraction in vapor phase in condenser";

+    Real K_c[Nc];
+    Real gmabubl_c[Nc];
+    Real philiqbubl_c[Nc];
+    Real gmadew_c[Nc];
+    Real phivapdew_c[Nc];
+    Real gma_c[Nc];
+    Real K[Nc];

Real Pdew(unit = "Pa", min = 0, start = Pmax)"
   Dew point pressure";

...
```

Simulator/UnitOperations/Flash.mo:

```
...
```

```
Real xvap(unit = "-", min = 0, max = 1, start =
    xvapg) "Vapor phase mole fraction";

+    Real gmabubl_c[Nc];
+    Real philiqbubl_c[Nc];
+    Real gmadew_c[Nc];
+    Real phivapdew_c[Nc];
+    Real K_c[Nc];
+    Real Cpres_p[Nc];
+    Real Hres_p[Nc];
+    Real Sres_p[Nc];

//=============================================================

//Instantiation of Connectors
Simulator.Files.Interfaces.matConn In(Nc = Nc)
  annotation(

...
```

Simulator/Files/Models/Flash.mo:

```
...

Real Pdew(start = Pmax, min = 0)"dew point
  pressure";

+    Real gmabubl_c[Nc];
+    Real philiqbubl_c[Nc];
```

```
+    Real gmadew_c[Nc];
+    Real phivapdew_c[Nc];
+    Real K_c[Nc];
+    Real Cpres_p[Nc];
+    Real Hres_p[Nc];
+    Real Sres_p[Nc];

extends GuessModels.InitialGuess;

...
```

Simulator/Files/Interfaces/enConn.mo:

```
...

Real Q;

+    Integer Nc "Number of components"
   annotation(Dialog(tab = "Stream
   Specifications", group = "Component
   Parameters"));

end enConn;

...
```

Simulator/UnitOperations/DistillationColumn/ DistCol.mo:

```
...

Dialog(tab = "Column Specifications", group = "
  Calculation Parameters"));

+    Simulator.Examples.Distillation.Condenser
  condenser;

Real RR(min = 0);

...
```

Simulator/Files/Models/ReactionManager/ ConversionReaction.mo:

```
...

import data = Simulator.Files.Chemsep_Database;

+    parameter ChemsepDatabase.GeneralProperties
    C[Nc] "Component instances array" annotation
  (Dialog(tab = "Flash Specifications", group =
    "Component Parameters"));
+    parameter Integer Nc "Number of components"
    annotation(Dialog(tab = "Flash
  Specifications", group = "Component
  Parameters"));
```

```
//Number of Reactions involved in the process
parameter Integer Nr "Number of reactions"
  annotation(
Dialog(tab = "Reactions", group = "Conversion
  Reaction Parameters"));

...
```

Simulator/UnitOperations/AbsorptionColumn/AbsCol.mo

```
...

Simulator.Files.Interfaces.matConn Out_Bot(Nc =
   Nc) annotation(Placement(visible = true,
  transformation(origin = {100, -80}, extent =
  {{-10, -10}, {10, 10}}, rotation = 0),
  iconTransformation(origin = {250, -300},
  extent = {{-10, -10}, {10, 10}}, rotation =
  0)));

+   Simulator.Examples.Absorption.Tray tray[Nt
  ];

equation
//connector equation
tray[1].Fliq_s[1] = In_Top.F;

...
```

Simulator/UnitOperations/AbsorptionColumn/ AbsTray.mo:

```
...

Real Pbubl(min = 0, start =Pmin);

+    Real gmabubl_c[Nc];
+    Real philiqbubl_c[Nc];
+    Real gmadew_c[Nc];
+    Real phivapdew_c[Nc];
+    Real K_c[Nc];
+    Real Hres_p[Nc];

Simulator.Files.Interfaces.trayConn In_Liq(Nc =
    Nc) annotation(Placement(visible = true,
  transformation(origin = {-50, 40}, extent =
  {{-10, -10}, {10, 10}}, rotation = 0),
  iconTransformation(origin = {-50, 40}, extent
   = {{-10, -10}, {10, 10}}, rotation = 0)));

...
```

Uncommented two lines of declarations

Simulator/Files/Models/ReactionManager/ KineticReaction.mo:

```
...
```

```
import data = Simulator.Files.Chemsep_Database;

-    // parameter ChemsepDatabase.
  GeneralProperties C[Nc];
-    // parameter Integer Nc;

+    parameter ChemsepDatabase.GeneralProperties
    C[Nc];
+    parameter Integer Nc;

parameter Integer Nr "Number of reactions"
  annotation (Dialog(tab = "Reactions", group =
   "Kinetic Reaction Parameters"));

...
```

2.2.5 Non-array modification for array components

This is done to set individual parameter values for each element in the array.

Simulator/Files/Models/ReactionManager/ EquilibriumReaction.mo:

```
...

Integer BC_r[Nr];
//Equilibrium Constant
```

```
-    Real K[Nr](start=xliqg);

+    Real K[Nr](each start=xliqg);

Real N[Nr](each start= Fg),D[Nr](each start=Fg)
   ;

...
```

Simulator/UnitOperations/HeatExchanger.mo:

```
...

Real Tcin(unit = "K", start=Tg) "Cold inlet
   stream temperature";

-    Real Fcin[1](unit = "mol/s", start=Fg) "
   Cold inlet stream molar flow rate";

+    Real Fcin[1](each unit = "mol/s", each
   start=Fg) "Cold inlet stream molar flow rate
   ";

Real Hcin(unit = "kJ/kmol", start=Htotg) "Cold
   inlet stream molar enthalpy";
Real Scin(unit = "kJ/[kmol.K]") "Cold inlet
   stream molar entropy";
```

```
-    Real xcin_pc[2, Nc](unit = "-") "Cold inlet
    stream component mole fraction";

+    Real xcin_pc[2, Nc](each unit = "-") "Cold
   inlet stream component mole fraction";

Real xvapcin(unit = "-", start=xvapg) "Cold
   inlet stream vapor phase mole fraction";
//Cold Stream Outlet
Real Pcout(unit = "Pa", start=Pg) "Cold outlet
   stream pressure";
Real Tcout(unit = "K", start=Tg)"Cold outlet
   stream temperature";
Real couttT(unit = "K", start=Tg) ;

-    Real Fcout[1](unit = "mol/s", start=Fg) "
   Cold outlet stream molar flow rate";

+    Real Fcout[1](each unit = "mol/s", each
   start=Fg) "Cold outlet stream molar flow rate
   ";

Real Hcout(unit = "kJ/kmol", start=Htotg) "Cold
    outlet stream molar enthalpy";

...
```

Simulator/UnitOperations/Mixer.mo:

```
...

parameter Integer NI = 6 "Number of inlet
  streams" annotation(Dialog(tab = "Mixer
  Specifications", group = "Calculation
  Parameters"));

-   Real Pin[NI](unit = "Pa", min = 0, start =
  Pg) "Inlet stream pressure";

+   Real Pin[NI](each unit = "Pa", each min =
  0, each start = Pg) "Inlet stream pressure";

Real xin_sc[NI, Nc](each unit = "-", each min =
    0, each max = 1) "Inlet stream component mol
    fraction";

...
```

SSimulator/UnitOperations/ShortcutColumn.mo:

```
...

Real RRmin(unit = "-", start = 1) "Minimum
  Reflux Ratio";

-   Real alpha_c[Nc](unit = "-") "Relative
  Volatility";
```

```
+    Real alpha_c[Nc](each unit = "-") "Relative
    Volatility";

Real theta(unit = "-", start = 1) "Fraction";

...
```

SSimulator/UnitOperations/Splitter.mo:

```
...

Real xout_sc[No, Nc](each unit = "-", each min
   = 0, each max = 1) "Outlet Mixture Molar
   Fraction";

-    Real Fout_c[No](each unit = "mol/s", each
   min = 0, start = Fg) "Outlet Mixture Molar
   Flow";
-    Real Fmout_c[No](each unit = "kg/s", each
   min = 0, start = Fg) "Outlet Mixture Mass
   Flow";

+    Real Fout_c[No](each unit = "mol/s", each
   min = 0, each start = Fg) "Outlet Mixture
   Molar Flow";
+    Real Fmout_c[No](each unit = "kg/s", each
   min = 0, each start = Fg) "Outlet Mixture
   Mass Flow";
```

```
//=================================================

//Instantiation of Connectors
Simulator.Files.Interfaces.matConn In(Nc = Nc)
  annotation(

...
```

Simulator/UnitOperations/AbsorptionColumn/ AbsTray.mo:

```
...

Real Fliq_s[2](each min = 0, start={Fliqg,Fliqg
  });

-   Real xvap_sc[2, Nc](each min = 0, each max
  = 1, start=xvapg);
-   Real xliq_sc[2, Nc](each min = 0, each max
  = 1, start=xliqg);

+   Real xvap_sc[2, Nc](each min = 0, each max
  = 1, each start=xvapg);
+   Real xliq_sc[2, Nc](each min = 0, each max
  = 1, each start=xliqg);

Real Hvap_s[2](start={Hvapg,Hvapg}), Hliq_s[2](
  start={Hliqg,Hliqg}), Hvapout_c[Nc],
  Hliqout_c[Nc];
```

```
-    Real x_pc[3, Nc](each min =0, each max = 0,
   start=xliqg);

+    Real x_pc[3, Nc](each min =0, each max = 0,
    each start=xliqg);

Real Pdew(min = 0, start =Pmax);

...
```

2.2.6 Declarations of variables as parameters

This is done as the dimensions of the arrays must be a parameter or a constant expression.

Simulator/Files/ThermodynamicFunctions/ DensityRacket.mo:

```
...

function DensityRacket
extends Modelica.Icons.Function;

-    input Integer Nc;

+    parameter Integer Nc;

input Real T;
```

```
input Real P;
input Real Pc_c[Nc];

...
```

Simulator/UnitOperations/PFR/Integral.mo:

```
...

extends Modelica.Math.Nonlinear.Interfaces.
  partialScalarFunction;

-    input Integer Nc;
-    input Integer Nr;

+    parameter Integer Nc;
+    parameter Integer Nr;

input Integer Base_comp;
input Real Co_dummy[Nc - 1];
input Real DO_dummy[Nc - 1, Nr];

...
```

2.2.7 Resolutions of type mismatch error

"zeros(3)" creates an integer array containing three zeros, [0, 0, 0], whereas the arrays Cpres_p, Hres_p, and Sres_p are

composed of real numbers.

Simulator/Files/ThermodynamicPackages/ RaoultsLaw.mo:

```
...

for j in 1:Nc loop
    K_c[j] = Pvap_c[j] / P;
end for;

-    Cpres_p[:] = zeros(3);
-    Hres_p[:] = zeros(3);
-    Sres_p[:] = zeros(3);

+    Cpres_p[:] = fill(0.0, 3);
+    Hres_p[:] = fill(0.0, 3);
+    Sres_p[:] = fill(0.0, 3);

end RaoultsLaw;

...
```

2.2.8 Accessibility mode modifications

The variables are declared as protected.

Simulator/Files/ThermodynamicFunctions/ DensityRacket.mo:

```
...

    output Real rho_c[Nc];

    -    parameter Integer Nc;
    -    parameter Real R = 83.14;

protected

    +    parameter Real R = 83.14;
    +    parameter Integer Nc;

    Real Tr_c[Nc], Pcbar_c[Nc], temp[Nc], Tcor_c
      [Nc], a, b, c_c[Nc], d, e_c[Nc], Beta_c[Nc
      ], f, g, h, j, k, RPnew_c[Nc];

...
```

Simulator/UnitOperations/PFR/Integral.mo:

```
...

extends Modelica.Math.Nonlinear.Interfaces.
  partialScalarFunction;

-    parameter Integer Nc;
-    parameter Integer Nr;
-    Real Rate;
```

```
input Integer Base_comp;

...

input Real k;

+    protected

+        Real Rate;
+        parameter Integer Nc;
+        parameter Integer Nr;

algorithm
    Rate := 1;

...
```

Chapter 3

Porting OMSched to Windows OS

3.1 Description

OMSched is a Python-based scheduler application with a PyQt interface designed to efficiently schedule simulations within the OMEdit environment.

3.2 Resolutions of the errors

3.2.1 Dependencies update

note: This error originates from a subprocess, and is likely not a problem with pip. ERROR: Failed building wheel for pyzmq Failed to build lxml PyQt5-sip pyzmq ERROR: Could not build wheels for lxml, PyQt5-sip, pyzmq, which is required to install pyproject.toml-based projects

The issue at hand involves constructing wheels for lxml, PyQt5-sip, and pyzmq, essential for installing projects based on pyproject.toml. The versions specified in the requirements.txt file are incompatible, leading to the need for an update to the latest versions.

requirements.txt:

```
@@ -1,28 +1,28 @@

-    certifi==2022.5.18
-    cffi==1.15.0
-    charset-normalizer==2.0.12
-    click==8.1.3
-    cryptography==36.0.1
-    future==0.18.2
-    idna==3.3
-    lxml==4.8.0
-    numpy==1.22.3
-    OMPython==3.3.0
-    opcua==0.98.13
-    path==16.4.0
-    psutil==5.9.0
-    pycparser==2.21
-    pyparsing==3.0.7
-    PyQt5==5.15.6
-    PyQt5-Qt5==5.15.2
-    PyQt5-sip==12.9.1
-    python-dateutil==2.8.2
-    pytz==2021.3
-    pyzmq==22.3.0
-    regex==2022.4.24
-    requests==2.27.1
-    six==1.16.0
-    sseclient-py==1.7.2
-    urllib3==1.26.9
-    scipy==1.9.3
```

```
-    DyMat==0.7

+    certifi
+    cffi
+    charset-normalizer
+    click
+    cryptography
+    future
+    idna
+    lxml
+    numpy
+    OMPython
+    opcua
+    path
+    psutil
+    pycparser
+    pyparsing
+    PyQt5
+    PyQt5-Qt5
+    PyQt5-sip
+    python-dateutil
+    pytz
+    pyzmq
+    regex
+    requests
+    six
+    sseclient-py
+    urllib3
+    scipy
+    DyMat
```

3.2.2 Simulation window title fix

Engine/SimulationUI/sim_output_widget.py:

```
@@ -16,6 +16,7 @@ def __init__(self, parent=
  None):

...

    self.parentUI = parent
super().__init__(parent)

+    self.setWindowTitle("OMScheduler")

self.resize(650, 450)
self.mpCancelButton = QPushButton(Helper.
  cancelSimulation)

...
```

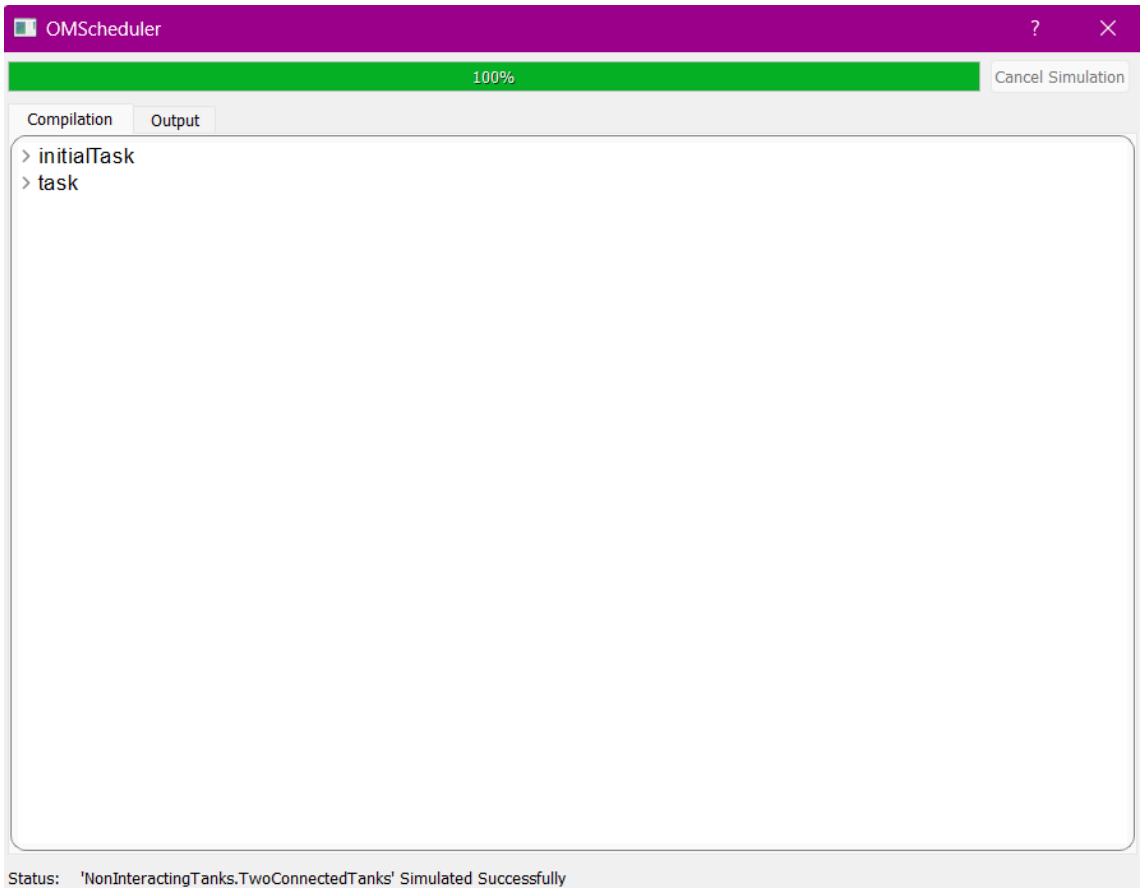

Figure 3.1: OMSched Simulation window

3.2.3 Simulation Window font fix

Engine/SimulationUI/sim_output_widget.py:

```
@@ -185,7 +185,11 @@ def __init__(self):

...

self.setSelectionMode(QAbstractItemView.
  ExtendedSelection)

self.setContextMenuPolicy(Qt.CustomContextMenu)
-    self.setFont(QFont(Helper.
  monospacedFontInfo.family()))
+    font = QFont("Arial", 10)
+    font.setLetterSpacing(QFont.
  PercentageSpacing, 110)
+    self.setFont(font)
+    self.setStyleSheet("QLineEdit { line-height
  : 150%; }")

self.setStyleSheet("""
    QTreeView {

...
```

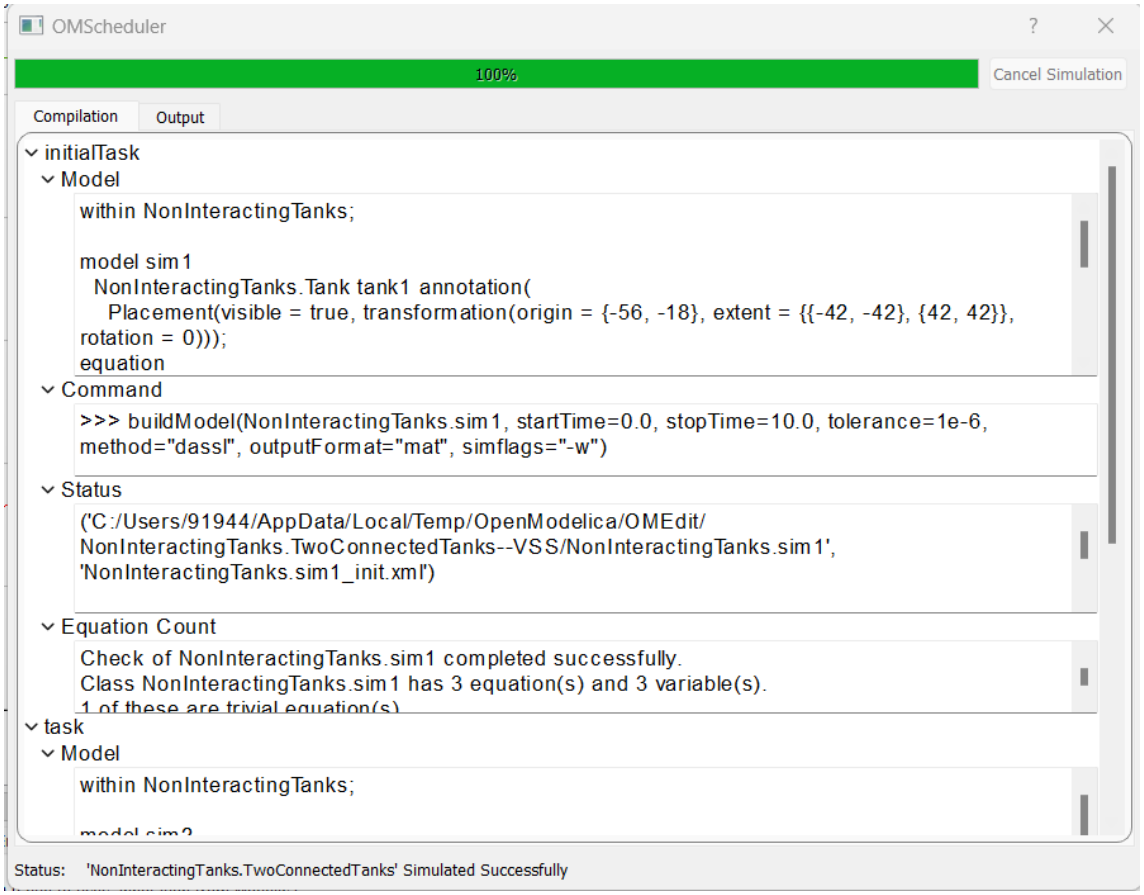

Figure 3.2: OMSched Simulation window

3.2.4 File path fix for Parallel Compilation method

Engine/schedule_model.py:

```
@@ -113,7 +113,7 @@ def loadPackage(self, pkg,
  isStdLib=False):

...

        if isStdLib:
            self.mosTemplate.append('loadModel("'
               + pkg + '");\n')
        else:
-               self.mosTemplate.append('
  loadFile("' + pkg + '");\n')
+               self.mosTemplate.append('
  loadFile("' + pkg.replace("\\", "/") + '");\n
  ')

def processSchedule(self, simName, topModel,
  topModelPkg):
    self.process = ProcessModel(self.DEBUG)

...
```

3.2.5 Cleaning up omc on closing for JIT Compilation method

Engine/schedule_model.py:

```
@@ -113,7 +113,7 @@ def loadPackage(self, pkg,
  isStdLib=False):

...

        if isStdLib:
            self.mosTemplate.append('loadModel("'
               + pkg + '");\n')
        else:
-               self.mosTemplate.append('
  loadFile("' + pkg + '");\n')
+               self.mosTemplate.append('
  loadFile("' + pkg.replace("\\", "/") + '");\n
  ')

def processSchedule(self, simName, topModel,
  topModelPkg):
    self.process = ProcessModel(self.DEBUG)

...
```

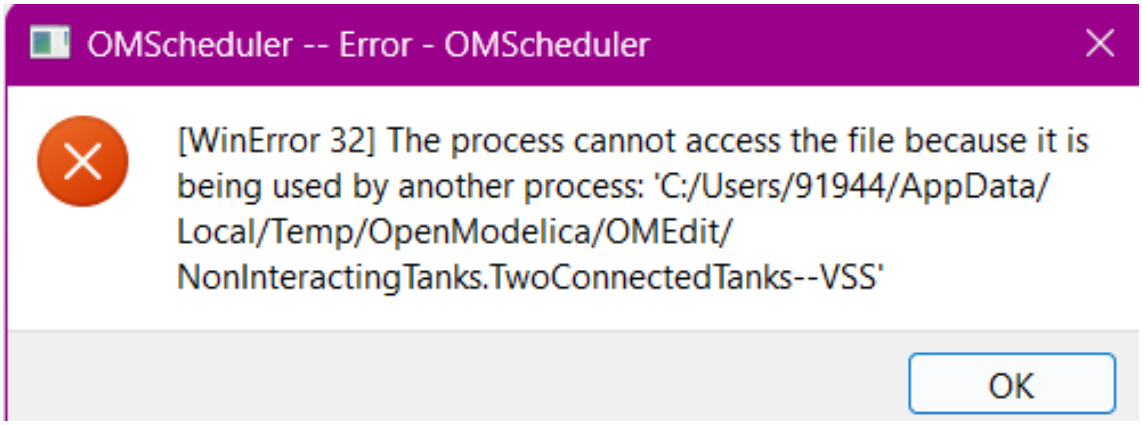

Figure 3.3: ERROR message

Chapter 4

Conclusion

In this report, I have presented my work as an intern at the FOSSEE Summer Fellowship, where I focused on modelling and simulation in OpenModelica. I have described the following tasks that I accomplished during the internship:
- Resolving errors and bugs in the OMChemSim library, which contains models for chemical process simulations.
- Part in porting OMSched, a scheduler application for OMEdit, to Windows OS and fixing a few issues.

Through these tasks, I have gained valuable experience and skills in using OpenModelica and its tools, as well as in debugging, testing, and version control.

The primary difficulty I encountered revolved around the building, construction, and installation phases, presenting numerous challenges that proved challenging to overcome. Despite being unfamiliar with the Modelica language initially, I managed to grasp it, thanks to the assistance provided by informative spoken tutorial videos.

Chapter 5

Reference

- Spoken Tutorials
- OpenModelica Documentation
- Stack overflow
- Wikipedia