

Library Structure of Dynamic Simulation for Combined Heat and Power Plant in Modelica Language

Amir A. Razak (*Author*)
Faculty of Mechanical Engineering
Universiti Malaysia Pahang
26600 Pekan, Pahang, Malaysia
e-mail: amirrazak@ump.edu.my

Abstract — The most common form of energy recycling system is Combined Heat and Power (CHP) plants. The CHP plant is a complex system and still under intensive development by many researchers. The system needs to be developed in quick and efficient manners with low resources based on modeling and simulation method. With the development of CHP library in open source Modelica language, it could be used as a base for further advancement of CHP technology. The aim of this work is to design a structure of initial version of a model library for the dynamic simulation of Combined Heat and Power plants (CHP). Modular approach and top-down design have been implemented in the model library development. A solid base for this work is defined which includes rules in modeling the components (e.g. robustness and reusability), default library structure arrangement and model documentation. By strictly follow the rules and concepts introduced in this work, the mistakes in modeling is minimized. The designed library in Modelica language will provide an organized environment in modeling a CHP plant.

Keywords – *Modelica language; Combined Heat and Power (CHP) plant, CHP library*

I. INTRODUCTION

Combined Heat and Power plant (CHP) or also known as cogeneration plant generates electricity (and/or mechanical energy) and thermal energy by accessing the respective energy flows from a prime mover. One main objective of cogeneration is to maximize the fuel utilization of the system and to save primary energy in comparison to separated generation of secondary energy [1]. To achieve this, a good plant topology with suitable components must be designed which require a thorough evaluation of the demand structure and location of the plant among others. For this purpose, quite a number of CHP plant variations can be seen nowadays as a result from the intensive applied research activities in the field of CHP.

A complete working CHP plant designing, without a doubt will consume tremendous and considerable amount of manpower and resources in terms of money and time. Because of these reasons it is paramount that designing process must be done efficiently with minimal mistakes in the plant design to ensure the CHP plant can work as it is intended to be. When designing a specific CHP plant many important aspects must be considered thoroughly, such as selection of plant topology, components types and capacities,

selection of sensors and actuators as well as plant control system among others. Besides the design aspect, the operation behavior of the target plant should be considered as well.

By using the CHP library with Modelica backbone, it could simplify the designing and dimensioning of CHP plant process and reduce significantly the design time if compared to conventional method. All scope of plant design can be analyzed and tested in Modelica tool with CHP library which is more effective and of course with much less resources. Other advantage is that some difficult analysis such as plant's operational behavior can be simulated by using this tool. One can evaluate the effect of component pitfalls to the overall plant as well as possible load demand change during the operational time.

The free Modelica language is developed by the non-profit Modelica Association. The Modelica Association also develops the free Modelica Standard Library that contains about 780 generic model components and 550 functions in various domains, as of version 3.0 [2].

II. MODELING APPROACH

The CHP library of this reported work is modeled according to the flow chart shown in Fig. 1. CHP system is first decomposed and analyzed. At this stage detail information about its structure and components are obtained. Then the initial modeling phase started where all equations, assumptions, variables and parameters as well as characteristics of the model are documented. A new Modelica library is then defined considering the structure and classes available in the Modelica Standard Library (MSL) and using Modelica modeling and simulation environment which in this case is referred to Dynasim [3]. Several models will then be sorted out from the Modelica library and continued by further works in modeling CHP components using the sub-models available in the library and followed by parameterization of the specific application's models.

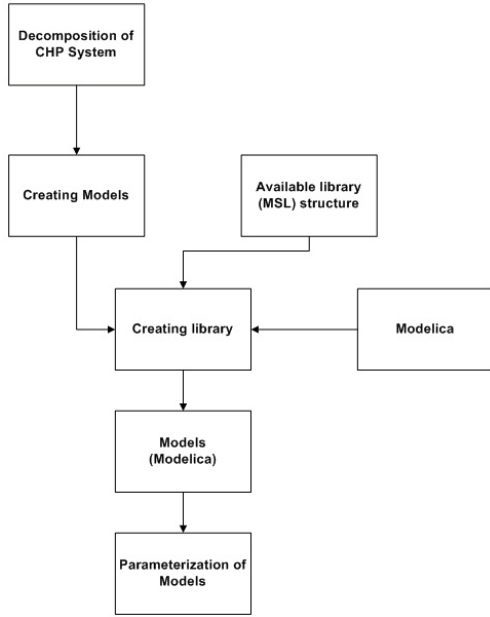


Figure 1. Approach in modeling of components using CHP library

III. LIBRARY STRUCTURE DESIGN

A top-down approach is implemented in most of the library structure design after some considerations have been taken into account as this approach is better than the bottom-up modeling [4]. A clear advantage of top-down approach in comparison to bottom-up approach is that most of basic problems in the initial model development phase can be tackled and solved phase by phase before a model evolving to be more complex.

However, model developer is not obliged to stick only to one approach, as it might be at some stage later that a mix between top-down and bottom-up design will be needed and suitable when dealing with ever growing complicated models. In order to provide a strong base for developing a modeling project, a good library structure must be provided. Any model should be developed according to the model type as specified in CHP library hierarchy presented in this paper.

From various literatures and sources of CHP plant for instance ASUE [5] and VDI 4608 Guideline [1], components commonly used in a CHP plant from various plant types can be identified. CHP library is designed in a hierarchical structure and divided into several top-level sub-libraries according to the type of models. A library or sub-library comprises several classes is called a package in Modelica nomenclature. Each top-level sub library may contain again sub-libraries with group classes that are even more specific.

The structure of the CHP library groups the different types of components of CHP according to their applications and specifications, e.g. thermal components, hydraulic

components, prime movers among others. Additionally, there are basic sub-libraries necessary to model a CHP plant such as media, load profile, and control libraries.

By identifying common interfaces and components used in several classes, a hierarchical structure of modular CHP library is built (Fig. 2).

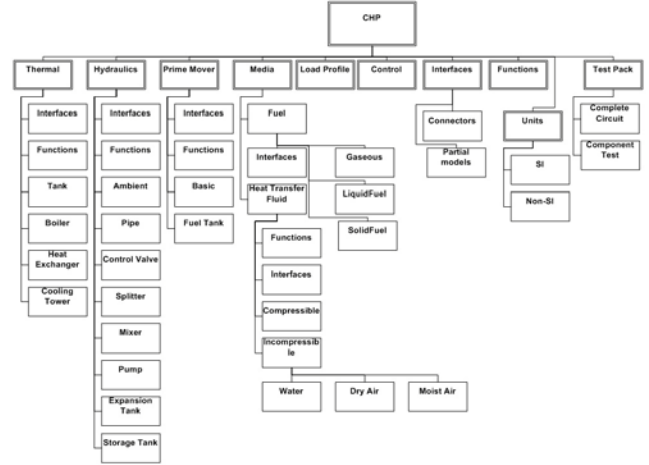


Figure 2. Hierarchical structure of a Modelica CHP library

A. Interfaces

Interfaces top-level package are at highest hierarchy because of its significant importance. In this package, all connectors and partial models used in the models are stored. Partial model is a model that is partially complete and cannot run and simulate by itself. It is used as a base model to build more complete model. In case these interfaces are needed in classes in specific top-level package, for instance pipe class in hydraulics package, then the suitable connectors and partial model available in top-level package interfaces are extended to the sub-package interfaces in the hydraulics package. The details of the CHP library structure is shown in Figure 2. The same method is applied for the use of functions in top-level package functions. Extending any record and class into other classes can be done by using extend command. Extend means that the specific class is “borrowed” by other classes that need to use the characteristics of the extended class in order to make a complete model. By separating interfaces and functions packages from other main packages, it will increase and maintain the modularity of models and functions in the library.

In each top level package, apart from the corresponding components, two more important packages namely interfaces and functions are included as a sub-package of each top-level package respectively. The sense of this design is that the classes and models inside the main package will have an access to these local (interfaces and functions) packages, which are the extension from the main functions and interfaces (I&F) packages. It is advisable that all new I&F

are defined at top-level and extended to the local I&F package. The purpose of this method is that to create a collection of I&F in main top-level I&F packages in order to centralize the control of these collections, to facilitate code maintainability in case the code or algorithm needed to be revised and to reduce the search time of I&F. In other respects, it will increase the reusability as well as reduce the code redundancy problem in the library. In case developers need anything from this package, they can easily extend it to local sub-library and use it directly without having to change any codes in it. This will localize the usage of specific I&F and reduce the time for finding a suitable I&F for a required model.

B. Media

Other important component of the library is media package. Almost all hydraulic and thermal classes, components used widely in the CHP plant require fluid properties to work. A lot of information about fluids depending on the level of details is needed. Media package is divided into several sub-libraries depending on the type of the fluids. Up to this point, two packages, fuel and heat transfer fluid are defined in media package. Fuel are introduced in here as it is necessary to represent primary energy supply to drive the prime mover and boiler in the CHP plant and it includes all fuel properties corresponding to each type of specialized fuel in this package, for instance diesel and natural gas. Fuel package branched out to physical type of fuel packages (solid, liquid and gaseous fuel) and in each of these packages; specific fuel (diesel, natural gas) properties are then defined.

In a hydraulic cycle of CHP plant, a fluid which serves as a heat transfer medium is required. Several types of working fluids have been defined in packages classed into physical fluids, which are incompressible, and compressible fluids sub-libraries. The working fluids are classed into their corresponding types similar to as in fuel packages, for instance, an incompressible fluid consists of water and other fluids. Working fluids should have relevant purposes to the models before including it in the package. For example, water is used as working fluid in the hydraulic cycle CHP plant as well as in the steam power plants, so that it is necessary to include it as it is an essential part of the model. The same case applied to combustion model. In this model, moist air or dry air can be used depends on the complexity of the model which means moist air and dry air models are required to be included and also for the purpose of future expansions of respective media models. The media package can be expanded in the same way as explained before.

If there are any new classes created, they should be placed in appropriate libraries in order to maintain the organizational structure. This should be done to make sure that no redundant models for similar components appear in different libraries hierarchy. If possible, the placement of the model should follow the “right the first time” concept so that it will reduce any complication with other dependent

models. Compared to other engineering modeling fields, such as electrical systems or multibody systems, the task of providing a “standard” library for thermo-fluid systems is much more difficult due to a greater variety of modeling assumptions that can be made, depending on the specific application needs [6]. It is necessary to refer to more established and tested fluid and media library as in [6] and [7] in order to minimize mistakes during modeling the media properties.

IV. SAMPLE MODEL

The following is a brief insight on the application of the library and the inner working of a Modelica model and Modelica functions from the library are discussed.

A. Friction factor equation, λ for circular pipe

In general, flow characteristics can be classed based on Reynolds Number into three different types as in Table I.

Table I. Friction factor equations for circular pipe

Flow type	Reynolds Number, Re	Friction factor equation, λ	
Laminar	Re < 2320	$\lambda = \frac{64}{Re}$	Haagen-Poiseuille [8]
Transition (laminar to turbulent)	2320 < Re < 105	$\lambda = \frac{0.3164}{\sqrt[4]{Re}}$	Blasius [8] (smooth pipe) or Colebrook-White [9]
		$\frac{1}{\sqrt{\lambda}} = -2.0 \log \left[\frac{\epsilon}{3.7d} + \frac{2.51}{Re\sqrt{\lambda}} \right]$	
Turbulent	Re > 105	$\frac{1}{\sqrt{\lambda}} = -2.0 \log \left[\frac{\epsilon}{3.7d} + \frac{2.51}{Re\sqrt{\lambda}} \right]$	Colebrook-White [9]

To complete the calculation of λ , variable Reynolds Number, Re, must be included in the equations.

Reynolds Number for flow in pipe:

$$Re = \frac{\rho \cdot c \cdot d}{\mu} = \frac{c \cdot d}{\nu}$$

Darcy-Weisbach [8] pressure loss equation in pressure loss form

$$\Delta p = \lambda \cdot \frac{l}{d} \cdot \frac{\rho \cdot c^2}{2}$$

and

$$\Delta p = p_{in} - p_{out}$$

where,

ε = absolute roughness of internal pipe wall (m)
 d = internal pipe diameter (m)
 ρ = density of fluid (kg/m³)
 c = mean fluid velocity (m/s)
 μ = dynamic viscosity (kg/m.s)
 ν = kinematic viscosity (m²/s)
 p_{in} = pressure at inlet pipe (Pa)
 p_{out} = pressure at outlet pipe (Pa)

As shown in the Colebrook-White (C-W) equation [9], the friction factor, λ , is implicitly expressed in the equation which has to be solved iteratively. In order to reduce the complexity of the equation while maintaining the accuracy of the results, explicit approximation equations of Haaland [10] and Chen [11] were chosen to be included in this work. Haaland approximation is simpler but sufficient for most turbulent cases while Chen has higher accuracy but more complex than the latter [9]. More detailed discussions regarding the accuracy of sets of explicit approximation equations to the implicit C-W equation can be found in [9].

Haaland explicit approximation:

$$\frac{1}{\sqrt{\lambda}} = -1.8 \log \left[\left[\frac{(\varepsilon/d)}{3.7} \right]^{1.11} + \left(\frac{6.9}{Re} \right) \right] \quad (1)$$

Chen explicit approximation:

$$\frac{1}{\sqrt{\lambda}} = -2 \log \left\{ \left[\frac{(\varepsilon/d)}{3.7065} \right] - \left(\frac{5.0452}{Re} \right) \log \left[\left[\frac{(\varepsilon/d)^{1.1098}}{2.8257} \right] + \left(\frac{5.8506}{Re^{0.8981}} \right) \right] \right\} \quad (2)$$

The abovementioned equations are then converted into Modelica code form.

B. Modelica function for friction factor, λ

```
function lambda_func
import Modelica.Math.log10;
input Velocity vel;
input Diameter pipe_d;
input Length E; //pipe roughness
output Real lambda_pipe;
output Real re;
protected
CHP.Media.HeatTransferFluid.IncompressibleFluid.Water.S
impleWater meda;
algorithm
```

```
if vel>0 then
//Calculate reynolds number
re:=vel*pipe_d/meda.nue;
if re<2320 then
lambda_pipe:=64/re; // Hagen-Poiseuille
elseif re>=2320 and re<10^5 then //and re<10^5
lambda_pipe:=0.3164/(re^(1/4)); // Blasius
else
lambda_pipe:=(1/(-
1.8*log10((6.9/re) + ((E/pipe_d)/3.7)^1.11)))^2; // Haaland
/* //Explicit Colebrook: Chen
else
lambda_pipe:=(1/(-2*log10(((E/pipe_d)/3.7065)-
(5.0452/re)*log10(((E/pipe_d)^1.1098)/2.8257)+(5.8506/(r
e^0.8981))))))^2;
*/
end if;
else
re:=0.001;
end if;
end lambda_func;
```

This function can be implemented in any circular pipe model using the same CHP library by extending it into a specific model and parameterize according to the user specification. Explicit approximation equation C-W: Chen is purposely disabled using comment form. Depending on demand of degree of accuracy and computation time, equation that best suit for a working model can be selected.

Modelica model for circular pipe

```
model pipe_func
Real lambda_pipe;
Real re;
.
.
equation
meda.t=ta;
medb.t=tb;
meda.h=fluida.h;
medb.h=fluidb.h;
medb.p=fluidb.p;
meda.p=fluida.p;
dp=fluida.p - fluidb.p;
vdot=fluida.mdot/meda.rho;
vel=vdot/pipe_ac;
(lambda_pipe,re)=CHP.Hydraulics.Functions.lambda_fu
nc(vel, pipe_d, E);
dp=c*vdot^2;
c=lambda_pipe*pipe_l*meda.rho/(pipe_d^2*pipe_ac^2);
0=fluida.mdot*fluida.h + fluidb.mdot*fluidb.h + qdot;
end pipe_func;
```

Friction factor, λ is then applied to the circular pipe model as shown in model pipe function above. This is an example of the reusability concept that has been implemented in the

library which has reduced significantly codes redundancy and increased modularity of the functions that help to reduce the complexity of a model.

V. CONCLUSION

Object-orientation of Modelica language can be utilized to produce a reusable component-based architecture that can be extended and customized to meet future application requirements. The modular individual components in the Modelica library for Combined Heat and Power plant (CHP) can be easily modified and extended so that new components models can be added conveniently. As a result, the library can be expanded, changed and updated freely by the model developers or users within boundary of the plant.

ACKNOWLEDGMENT

This research project was supported by University of Applied Sciences Offenburg, Germany. My thanks are due to Prof. Dr. -Ing. Peter Treffinger for his great support and advice to this project.

REFERENCES

- [1] VDI-Gesellschaft Energietechnik. VDI 4608, Part 1: Energy systems Combined heat and power. VDI-Handbuch Energietechnik. Düsseldorf : Beuth Verlag GmbH, 2005.
- [2] M. Tiller, Michael. Introduction to Physical Modeling with Modelica. s.l. : Kluwer Academic Publishers, 2001.
- [3] Dynasim A.B. Dymola-Dynamic Modeling Laboratory. [Online] <http://www.dynasim.se>. 2009.
- [4] Fritzsön, Peter. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. s.l. : Wiley & Sons, 2004.
- [5] ASUE- Arbeitsgemeinschaft für sparsamen und umweltfreundlichen Energieverbrauch e.V. Einbindung von kleinen und mittleren Blockheizkraftwerken / KWK-Anlagen. Kaiserslautern : Verlag Rationeller Erdgaseinsatz, 2007.
- [6] F. Casella, M. Otter, K. Proelss, C. Richter, H. Tummescheit, "The Modelica Fluid and Media Library for Modeling of Incompressible and Compressible Thermo-Fluid Pipe Networks", Modelica 2006 Conference, Vienna, Sept. 2006.
- [7] R. Franke, M. Sielemann, F. Casella, M. Otter, K. Proelss, M. Wetter, "Standardization of Thermo-Fluid Modeling in Modelica.Fluid", Modelica 2009 Conference, Como, Italy, Sept. 2009.
- [8] Zierep, Jürgen und Bühler, Karl. Stromungsmechanik. Berlin: Springer-Verlag, 1991.
- [9] G.Yıldırım. "Computer-based analysis of explicit approximations to the implicit Colebrook-White equation in turbulent in Advances in Engineering Software", Elsevier (2009). doi:10.1016/j.advengsoft.2009.04.004.
- [10] S.E. Haaland. "Simple and explicit formulas for the friction factor in turbulent." J. Fluids Eng. 105. 1983,
- [11] N.H. Chen, "An explicit equation for friction factor in pipe.", Bd. Ind Eng. Chem. Fund. vol.18(3), 1979, pp.296