Accepted at European Simulation Multiconference ESM'98, Manchester, United Kingdom

DESIGN OF A THERMO-HYDRAULIC MODEL LIBRARY IN MODELICA™

Hubertus Tummescheit, Jonas Eborn
Department of Automatic Control
Lund Institute of Technology
Box 118, SE-221 00 Lund, Sweden
Hubertus.Tummescheit@gmd.de, jonas@control.lth.se

ABSTRACT

A new language called Modelica™ for hierarchical physical systems modeling is developed in an international effort. The main objectives are to make it easy to exchange models and model libraries between tools and to use object-oriented constructs to facilitate the reuse of modeling knowledge. To promote the idea of a unified modeling language for continuous and hybrid systems, public domain model libraries for basic models are developed in various application domains. This paper deals with the design of a base library for thermo-hydraulic applications. Requirements for the library are derived from representative simulation examples and special care is taken to cover the basic physical phenomena broadly. Accurate and fast calculations for thermodynamic properties of fluids constitute a very important and intricate part of thermo-hydraulic calculations, therefore they are given much attention. The state equations for the transport of mass, momentum and energy are formulated in a very general way in order to allow application oriented simplifications in derived classes. Constitutive equation submodels (e. g., for heat transfer coefficients) complete the set of basic building blocks. Modelicas unique language features[1] of class parameters and multiple inheritance are extensively used in order to structure the library for intuitive use by model-users and maximum code reuse by model-developers. The library is under development now.

GENERAL CONCEPT

The general goal of the library is to provide a framework and basic building blocks for modeling thermo-hydraulic systems in Modelica™. For obvious reasons it is impossible to provide a complete library, so one of the main goals is extensibility. For the same reason, much more emphasis will be but on the basic parts of the library, such as medium models and essential control volumes, than on an exhaustive application library. The focus of the library is on models of homogeneous one- and two-phase flows, non-homogeneous and multi-phase flows are not taken into account yet. It is necessary to support bidirectional flow, because flow directions can change during simulation or are not known initially in networks. Flow splitters or junctions must also be modeled correctly for arbitrary flow directions in all branches.

These goals lead to the following guidelines:

1. one unified library both for lumped and distributed parameter models,
2. separation of the medium submodels, which can be selected through class parameters,
3. both bi- and unidirectional flows are supported,
4. assumptions (e. g., incompressible flow) can be selected by the user in the control volume level.

The first guideline puts a constraint on the discretization method that must be used in the distributed parameter models: only the multinodal or staggered grid method, where all fluxes are calculated on the border of a control volume and the intensive quantities are calculated in the center of a control volume, reduces to a useful model in the lumped parameter case. This model is very common for systems modeling and for one-dimensional discretizations, but it has the drawback that the approximation of the spatial derivatives is always only first order accurate. The staggered grid model is also used for heat conduction in solid structures.

Homogeneous fluid properties over the cross section are assumed in all models, but whenever suitable a property distribution along the main axis increase the model accuracy. In the case of combined convection-diffusion processes and heat transfer, the steady state solution temperature profile of the partial differential equation (PDE) combined with safeguards against unphysical behavior in fast transients results in a better approximation of real equipment and make it easier to compare result with stationary calculations.

Experiences from a library for the simulation of thermal power plants written in the SMILE language (Tummescheit and Pitz-Paal 1997) (mainly distributed models) and from a library written in OMOLA (Eborn 1998) (mainly lumped parameter models) are combined to form the basis for the Modelica library.

COMPONENT BUILDING BLOCKS

The main goal of a model library is to enable the user to quickly assemble complex plant topologies from the basic building blocks that result in a physically correct compound model and can be simulated efficiently. For thermo-hydraulic networks this can be achieved by structuring the user models in a specific way: all models have a design flow direction, also when the physical model correctly describes bidirectional flows. Many schemes of distributed and concentrated models can be realized in this way. It is also possible to model components, where the pressure drop is concentrated at the outlet and the thermal variables are distributed or neglect the pressure drop completely. These component blocks are at the same time the smallest unit, that "make sense" as a simulation experiment. Internally, these component blocks may be composed of several basic models. In general there is a basic energy and mass

[1] among physical systems modeling languages

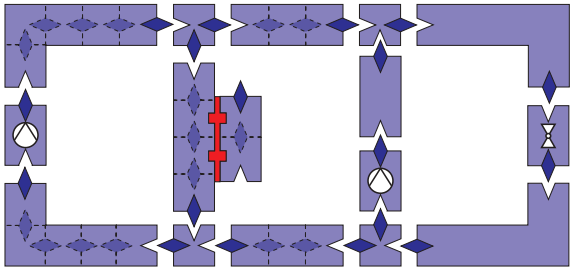

Figure 1: Simple example of plant topology involving lumped and distributed models.

storage block on the upstream side of the component and a relation between pressures and mass flows on the downstream side. In case of a dynamic momentum balance, this relation has the form of a differential equation, otherwise it is algebraic or implemented as an interpolated table (e. g., for pump characteristics). At the first look this may seem inflexible, but the concept of class parameters makes it possible for the user to exchange models on a lower level of the model composition hierarchy. The huge advantages of this concept are:

1. the resulting components are “natural”, like pumps, tubes and heat exchangers,

2. assembling such components always yields a well posed simulation problem, avoiding unnecessary high index problems,

3. a smaller number of different components is needed.

Figure 1 demonstrates a simple plant topology that is based on this principle. Models for heat transfer are composed in a similar way: models for heat storage and heat transfer coefficients are alternated.

The component building blocks are separated in application specific libraries, e. g., for power plant simulation, district heating networks, process engineering or building simulation.

CONTROL VOLUME MODELS

Fluid control volume models form the core of the library and define the assumptions and validity of the thermo-hydraulic applications that use them. They consist of at least one flow interface (most often two interfaces), one medium model, one set of state equations and a balance model, where all inflows and outflows and spatial derivatives of the system are calculated. This structure keeps the state equation model independent of the number of in- and outflows of a control volume and of the decision, whether or not bidirectional flow is permitted. Possible extensions to this minimal set comprise heat transfer models and pressure drop models from the constitutive equations and extended medium models for the transport properties λ, η and Pr. Control volumes for two-phase flow can be either of phase-separating type (a drum) or of phase-mixing type (two-phase flow in a tube). The standard library will contain control volume models for many standard cases, e. g., water and steam, air, flue gas etc.

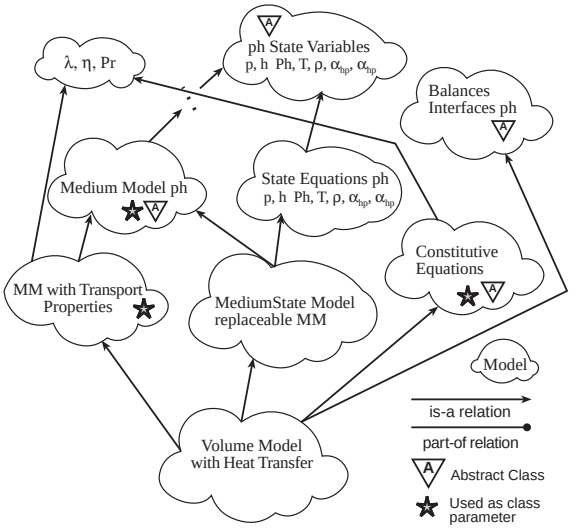

Figure 2: Multiple inheritance in control volume models.

Modelica™ offers two concepts for structuring model libraries that are unique among physical systems modeling languages: class parameters and multiple inheritance. Multiple inheritance is used in order to separate *orthogonal* concepts of the compound model in a way that merges the name spaces and thus allows to use variables from both submodels to be used within a compound model in a natural way. (Nilsson 1993) has investigated this possibility as an extension to the OMOLA language, though with slightly different semantics for multiple inheritance.

The usage of multiple inheritance in the composition of a control volume model is illustrated in Figure 2.

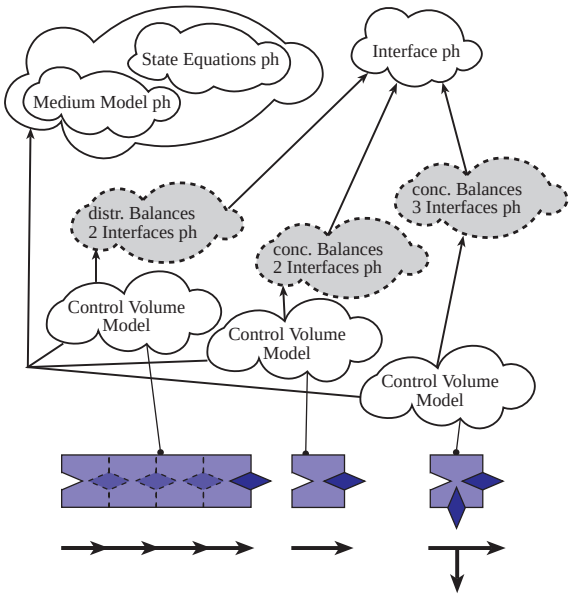

Figure 3: Exchanging the balance submodel in a control volume model.

One important feature of the library is the fact that the physical model of the control volume is independent of the number of inflows and outflows from the volume. All changes of mass and energy occur as a summation term in the state equations (1, 2). This term, the interfaces and,

when necessary, spatial derivatives are encapsulated in a balance submodel. The effect of using different balance submodels with the same set of medium- and state models is illustrated in Figure 3. Uniform interfaces for all control volumes enhance re-usability and allow mixing lumped and distributed models in the system.

The medium and the state submodels contain the physics part of the fluid model. The state equations define all phenomena that are part of the model and the medium model defines the complexity of the physical properties of the fluid. Therefore, the medium is implemented as a class parameter that can be exchanged by the user[2].

The following excerpt of Modelica code demonstrates, how multiple inheritance and class parameters are used together:

```
partial model BasicMediumState_ph
  extends BasicState_ph;
  replaceable model Medium = BasicMedium_ph;
  extends Medium;
  equation
  connect(BasicState_ph.state,Medium.state);
  ...
end BasicMediumState_ph;
```

The model first inherits (`extends`) from the class `BasicMediumState_ph`, which implements the state equations. Then it declares a class parameter, `Medium`, which is by default set to the base class for all medium models with pressure p and enthalpy h as states. Then the model inherits from this class parameter. The prefix partial indicates, that the model cannot be simulated as it is, because the abstract class `BasicMedium_ph` has to be replaced by a concrete class e. g., for water. It should be noted that in this example the parent class is changed via the class parameter, but this exchange is limited to classes that are *type compatible*[3]. The equation section establishes the interaction between the medium model and the state model with the `connect` statement. For a more thorough introduction to the Modelica™ language see (Mattsson, Elmqvist, and Broenink 1997).

An advantage of keeping control volume models general and using class parameters to adapt the behavior is, that less classes are needed in total. This results in a cleaner library design: instead of two inheritance branches in a library, only one branch and one class parameter with two possible replacement classes is needed.

Interfaces

The interfaces between control volumes are divided into two groups: one for onedirectional flow, the other for bidirectional flow. Each interface contains the mass flow, the downstream pressure in design flow direction and all transported properties. For bidirectional flow the transported properties are included twice, once for each flow direction. The interfaces in table 1 are provided.

[2]Currently this is done in textual form in the Modelica code. Soon, user interfaces will know about all suitable replacements for the class parameter e. g., in form of a pop-up menu.

[3]A class is type compatible if it contains at least the same components, possibly more.

one-component flows	
onedirectional	bidirectional
$\dot{m}, p, G, h, \varrho$	$\dot{m}, p, [G, h, \varrho]_l, [G, h, \varrho]_r$
$\dot{m}, p, G, T, \varrho$	$\dot{m}, p, [G, T, \varrho]_l, [G, T, \varrho]_r$
$\dot{m}, p, G, s, \varrho$	$\dot{m}, p, [G, s, \varrho]_l, [G, s, \varrho]_r$
multi-component flows	
onedirectional	bidirectional
$\dot{m}, p, c[:], G, h, \varrho$	$\dot{m}, p, [c[:], G, h, \varrho]_l, [c[:], G, h, \varrho]_r$
$\dot{m}, p, c[:], G, T, \varrho$	$\dot{m}, p, [c[:], G, T, \varrho]_l, [c[:], G, T, \varrho]_r$
$\dot{m}, p, c[:], G, s, \varrho$	$\dot{m}, p, [c[:], G, s, \varrho]_l, [c[:], G, s, \varrho]_r$

Table 1: Control Volume Interfaces.

G is the momentum flux ϱw, density times flow speed. The temperature T should only be used as interface variable, when the heat capacity c_p is assumed constant.

For component mixtures the above structure is extended by a dimensionless composition vector (kg/kg or mol/mol). Because the concentration is a transported property, two concentration vectors are needed in the interface for bidirectional flow.

Applications need specializations of the above interfaces, e. g., in building simulation temperature is used to model energy transport and two components of the total flow are modeled in a specific way: the water content in kg/kg and the CO_2 content in kg/kg. The concentration vector is then specialized to contain that 2 elements. Modelica supports two attributes for units: Unit and DisplayUnit. For CO_2 the DisplayUnit attribute is set to mg/kg, which is more usual in that domain.

For convective and radiative heat transfer a simple interface with temperature T and heat flux $\dot{Q}$ is sufficient. The mechanical power interface, consisting of torque t and angular velocity ω, needed in turbines, pumps and compressors is consistent with the interface from the mechanical domain base library.

Standardized interfaces are also needed between medium property submodels and flow models. The choosen state variables, e. g., $[p, h]$ or $[p, T, x]$, are always the input to the medium model. In the case of pressure and enthalpy as states $[p, h]$, the needed outputs are $[T, \varrho, (\partial\varrho/\partial h)_p, (\partial\varrho/\partial p)_h]$ and a discrete variable for the phase region (1 or 2 phases).

State Equations

The choice of the discretization method makes it necessary to calculate the momentum balance on the border of the control volume in contrast to the mass-, energy- or entropy balance, which are calculated in the center of the control volume. The choice of the state variables of these balance equations depends on numerical considerations and on the preferred input variables for the medium model. Most medium models, e. g., the cubic equations of state and also generalized Benedict-Webb-Rubin (BWR) models, describe the pvt-behavior of fluids, so that pressure p, specific volume v and temperature T are the most common choices for state variables with these simple medium models. A

convenient pair of state variables are p and T, except in the two-phase region. There T can be replaced by e. g., the steam quality x or the enthalpy h, which yields a model with switching states. Because of the close interconnection between medium models and choice of state variables and the need for efficiency in dynamic simulations, it is unreasonable to use one set of state variables for all applications. If the medium model is based on table interpolation, the choice of state variables depends only on numerical considerations. A common choice is pressure and enthalpy. In the case of constant-volume control volumes and a staggered grid discretization, the mass and energy balances can be transformed into two ODEs (Eborn 1998):

$$\frac{dp}{dt} = \frac{\varrho}{\alpha_p \varrho + \alpha_h} \left(\left(\frac{\varrho + \alpha_h h}{\varrho V} \right) \frac{dm}{dt} - \frac{\alpha_h}{\varrho V} \frac{de}{dt} \right) \quad (1)$$

$$\frac{dh}{dt} = \frac{\varrho}{\alpha_p \varrho + \alpha_h} \left(\left(\frac{1 - \alpha_p h}{\varrho V} \right) \frac{dm}{dt} + \frac{\alpha_p}{\varrho V} \frac{de}{dt} \right) \quad (2)$$

where $\alpha_p = \partial\varrho/\partial p|_h$ and $\alpha_h = \partial\varrho/\partial h|_p$. One of the important decisions when modeling thermo-hydraulic flow is if the momentum balance should be modeled dynamically or stationary. Due to the symbolic preprocessing of the equation system in Modelica, both variants can be combined in one model and a boolean compile-time parameter decides, which variant is actually used. The momentum balance for 1-phase or homogeneous 2-phase flows can be expressed as:

$$\frac{dG}{dt} = \underbrace{-w \left(\frac{\partial G}{\partial z} + \varrho \frac{\partial w}{\partial z} \right)}_{\Delta G} - \frac{\partial p}{\partial z} + \varrho g cos(\phi) + \left(\frac{\partial p}{\partial z} \right)_R \quad (3)$$

The right hand side of equation 3 without the gravitational term $\varrho g cos(\phi)$ is equivalent to one of the empirical stationary pressure drop correlations.

Looking at the momentum balance in detail we now define two flags $\mathbf{dy}$ and $\mathbf{gr}$ and introduce them in the corresponding terms:

$$\mathbf{dy} \left[\frac{dG}{dt} \right] = \mathbf{dy} \Delta G - \frac{\partial p}{\partial z} + \mathbf{gr}[\varrho g cos(\phi)] + \left(\frac{\partial p}{\partial z} \right)_R .$$

When both $dy = 0$ and $gr = 0$, this dynamic momentum balance reduces to the usual stationary pressure drop relation after equation sorting throws out all terms equal to 0:

$$\frac{\partial p}{\partial z} = \left(\frac{\partial p}{\partial z} \right)_R . \quad (4)$$

The variable G vanishes also, so that the equation $G = \varrho w$ is dropped completely.

This type of structural parameter allows to write state equations, where the influences of different forces or phenomena can be switched on or off via a parameter depending on the necessary range of validity of the model. State equations in fluid dynamics can always be structured like that: they consist of a sum of terms, each term accounting for the influence of one "force". It is thus possible to write only one state equation for all cases and adjust the number of included terms through parameters. Similar parameters as for the dynamic momentum balance account for

- the gravity term – not needed for gases
- the kinetic energy in the energy balance
- the dissipative term in the energy balance
- the diffusion term in the energy balance

When the state equations are used in a certain context, a component modification[4] sets these parameters to typical defaults, e. g., switch off the influence of gravity and the kinetic energy for ideal gas models in HVAC applications.

A similar approach can be applied to model differences between compressible and incompressible flow, which in that case would eliminate one equation completely (the mass balance) and alter another equation (the energy balance). At the same time, the medium model has to be exchanged via a class parameter.

Careful design is necessary in order to always end up with a well-defined set of equations. Application libraries using the base library set defaults for these parameters and type parameters.

Medium Models

The availability of fast, accurate medium models is a notorious problem in thermal hydraulic applications as soon as something more elaborate than ideal gas models are needed. In some areas there exist recommended formulations (IAPWS for water) or de-facto standards (NIST-routines for refrigerants) that have to be taken into account. External function call interfaces in Modelica make it possible to use these standards directly. Available routines and most medium property models in the literature (see, e. g., (Reid, Prausnitz, and Poling 1987)) are designed with stationary calculations in mind, therefore they have to be extended to include some extra derivatives for dynamic calculations. This has been done for the NIST routines for refrigerants, water and steam and cubic equations of state.

One important group of simple medium models are cubic equations of state, which only need a limited set of parameters for models of acceptable accuracy. They are implemented directly in Modelica and they are also valid for mixtures of (non-polar) media, for which critical data and binary interaction coefficients are available. Vapor-liquid equilibrium calculations (VLE) for cubic and other medium models have to be performed iteratively and numerically, either by using Maxwells criterium or calculating that Gibbs free enthalpy is equal for both phases. The numerical calculations are too inefficient to be performed at each time step during dynamic simulation. In order to calculate medium properties inside the two-phase region, it is sufficient to know the properties on the phase boundaries and interpolate with the vapor mass fraction x. An efficient implementation of medium properties for pure components requires that VLE are calculated before the simulation and that VLE data is approximated either with rational functions or with smooth spline interpolation. This approach is extendible to binary mixtures with well known properties.

[4] more information about Modelica grammar and semantics can be found on `http://www.dynasim.se/Modelica`

Many details of the medium model depend on the choice of the state equations. Medium models are separated into groups according to which states are used in the dynamic model.

The internal structure of medium models for the whole fluid region consists of at least two submodels, one for the one-phase part, another for the two-phase part. This division is useful for two reasons:

1. it allows to switch easily between different state variables in the one- and two phase regions.

2. the way of calculating medium properties is different in these two regions as explained above.

The direct usage of the available code implementations is often not advisable for the use in dynamic simulation, because of iterative routines within the medium property calls. The most efficient approach to high-accuracy medium property calculation are fast table interpolations, which are also independent from the choice of state variables. A suitable compromise between the size of the tables and the accuracy of the calculation can be found when the data sets for one- or two-dimensional smooth table interpolation are calculated on initialization of the model. For some applications and with a proper choice of state variables, cubic medium models are accurate enough and equally efficient.

Many applications need extensions to these medium models beyond the requirements of the state equations. Turbulent flow and heat transfer need the "transport properties" λ, η and Pr. The more common of these extensions will be included in expanded medium models which are subclasses of the simpler ones.

Constitutive Equations

Constitutive equations—heat transfer and pressure drop correlations, slip correlations for two-phase flow, etc—can be found in the literature in such an abundance, that it is frankly impossible to include more than a small selection. We will try to select some of the most common cases and give examples, so that a structure for extension exists. Models for laminar and turbulent convective, radiative, and two-phase heat transfer will be present. Often these constitutive equations are supplied by the equipment manufacturer, so that this part of the model needs to be user defined. In systems modeling there are two common approaches for heat transfer correlations: one is to supply a heat transfer coefficient for nominal conditions and scale it with the flow speed:

$$\frac{\alpha}{\alpha_0} = \left(\frac{Re}{Re_0}\right)^n \approx \left(\frac{c}{c_0}\right)^n \tag{5}$$

with $n = 0.8$ most often. One other approach is to calculate the Reynolds number Re, the Nusselt Number $Nu = f(Re, Pr)$ and use the definition of the Nusselt number $Nu := \alpha L_0/\lambda$ with the characteristic length L_0 and the heat conduction coefficient λ to calculate the heat transfer coefficient α. The second approach needs special medium property models and geometric information, the first one nominal values from stationary calculations. The constitutive equations are an orthogonal concept to the above mentioned parts of a control volume model and the structure and computational complexity of the two approaches are different. Therefore, the constitutive equations are collected in one class, which is one of the parent classes of the essential control volume model and can be exchanged via a class parameter in the same way as the medium model.

CONCLUSION AND OUTLOOK

Well validated yet flexible models for fluid control volumes constitute the basic building block of a model library for thermo-hydraulic processes. Reusability is assured by providing a fixed set of interfaces, flexibility is provided by implementing the medium models and constitutive equation parts of the model as replaceable *class parameters*. Modelicas unique language features of multiple inheritance and genericity are extensively used to create a library that consists of validated core models with high flexibility in the usage. This thermo-hydraulic base library forms an excellent basis for the development of application specific models. It is planned to realize an application library for heat exchangers in power plants: feedwater heaters, economizers, evaporators and superheaters, and to extend it later on with models for turbines, pumps, valves and controllers. Automatic translation tools from OMOLA to Modelica™ code will make it possible to reuse well validated existing models and speed up the library development.

References

Eborn, J. (1998, February). *Modelling and Simulation of Thermal Power Plants*. Lic Tech thesis ISRN LUTFD2/TFRT--3219--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Mattsson, S. E., H. Elmqvist, and J. F. Broenink (1997, September). Modelica: An international effort to design the next generation modelling language. *Journal A, Benelux Quarterly Journal on Automatic Control 38*(3), 16–19. Special issue on Computer Aided Control System Design, CACSD.

Nilsson, B. (1993, August). *Object-Oriented Modeling of Chemical Processes*. PhD thesis ISRN LUTFD2/TFRT--1041--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Reid, R. C., J. M. Prausnitz, and B. E. Poling (1987). *The Properties of Gases and Liquids*. Boston, Massachusetts: Mc Graw Hill.

Tummescheit, H. and R. Pitz-Paal (1997). Simulation of a solar thermal central receiver power plant. In A. Sydow (Ed.), *Proceedings of the 15th IMACS world congress on Scientific Computation, Modelling and Applied Mathematics*, Volume 6, Berlin, Germany, pp. 671–676. Wissenschaft und Technik Verlag.