Object-oriented Modeling of Thermo-hydraulical Processes

Hubertus Tummescheit, Lund Institute of Technology

This paper has been translated from German by other persons then the author. This translation process can have introduced errors in the paper. There are some untranslated text left in the paper but hopefully that will not hinder the understanding of the text.

1 Introduction

Thermohydraulics includes processes, which employ flow-mechanical and thermodynamical phenomena for system descriptions, for example in steam power plants, cooling facilities, air conditions and heat exchangers. The basics of modeling of such processes for the system simulation in Modelica is being shown.

Software for the dynamic simulation of thermohydraulical processes, like SINDA/FLUINT [3], evinces deciding disadvantages as comprehensive tools for modeling and simulating the complete development cycle of a product: it is tightly restricted to a subject, not combinable with other tools, and is hardly usable for regulation design. Models, based on Modelica, have a number of advantages compared to the traditional systems: the use of the object-oriented method speeds up the model design, maintenance and adaptation of model libraries is simplified and—increasingly important for the development of complex, heterogeneous systems—models of different application areas can be easily combined.

2 Basis

Models of thermohydraulical facilities consist of three basic elements which should be modeled separately, according to the object-oriented approach:

- Control volumes, consisting of
  - preservation equations for mass, energy and impulse,
  - descriptions of the fundamental thermohydraulical properties of fluids,
  - interfaces to bordering volumina,
  - empirical equations for the description of heat transmission and pressure losses.
- Heat storage and heat conduction in solid materials and
- description of device geometry.

On of the special problems of thermodynamical modeling of fluids is the selection of possible state variables which can describe the problem: two of the variables pressure p, density ρ or specific volume $v = 1/\rho$, temperature T, entalphy h, inner energy u or entropy s describe the thermodynamical state of a fluid unambiguously, with the exception of p and T, which are independent from each other in the two phase area. Two of this units are selected to be states, the others are calculated from a model of the material properties, as far as they are needed. Because of the numerical properties of the preservation equations, only the combination of pressure p with h, u or s over the complete state space is possible, in the one phase area additionally with T. Here models with pressure and entalphy as states are being presented, which are the most cited in literature (see overview in [8]). The preservation equations for mass, energy and impulse are being formulized in a very general form, the application-oriented simplifications are shifted into derived classes. Robust models require that flows in both directions are represented correctly at all positions of a network, because the flow direction is either not known in the beginning or can change while simulation. This also applies to connectors and branches[1].

The thermodynamical material models form an important basis for the simulation of thermohydraulical processes, precision and calculation speed are often determinative for the full simulation.

Empirical heat transition relations and pressure loss relations often have to be adapted, hence only the most important are included in the Modelica library.

One presumption for a library which allows for mixing models of distributed and concentrated parameters is a discretizing scheme which is valid in both situations. The "finite volume" method with displaces grids (detailed in [11]) fulfills this requirement. The basic idea of the finite volume method is to divide a flow up into a finite number of volume elements and to apply the preservation propositions to every control volume. Here the elements are being discretized through the definition of the "resulting" intensive units (T, p, h, s) of an element at an "inner point" of a volume and the extensive units (mass flow, impulse flow) at the surface. Intermediate values are being calculated with the help of interpolation formulas, mostly through linear interpolation. Putting up the preservation equations for a control volume requires the intensive units of the flow-upwards po-

[1] To handle arbitrary flow directions dynamically correct, the directions should not be modeled without mass but also consist of a control volume with three interfaces.

Modelle für Reinstoffe	
Eine Strömungsrichtung	Zwei Strömungsrichtungen
$\dot{m}, p, \rho, h$	$\dot{m}, p, \Delta G_l, \Delta G_r, \rho_l, \rho_r, h_l, h_r$
Modelle für n-Stoff Gemische variabler Zusammensetzung	
$\dot{m}, p, C[n], \rho, h$	$\dot{m}, p, C[n]_l, C[n]_r, \Delta G_l, \Delta G_r, \rho_l, \rho_r, h_l, h_r$

Table 1: Schnittstellen zwischen Volumenelementen

sitioned control volume h, ρ and also the convective part of the impulse transportation flow ΔG (see Gl.6), which makes it necessary to name those in the interface of a control volume representing model object (see figure 1).

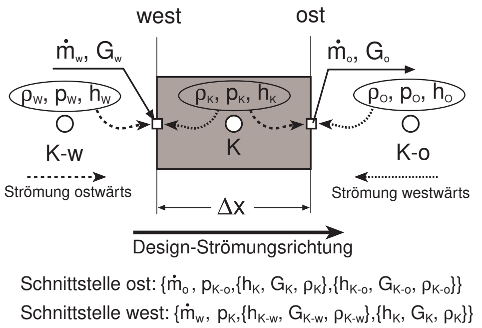

Figure 1: Diskretisierung der Kontrollvolumen.

3 Interfaces

The standardization of interfaces is an important presumption for the reuse and for the exchange of models between different libraries. The interfaces between the volume elements are divided into two groups: one for flows in which a flow direction can be defined everywhere, and another for arbitrary flow directions. All interfaces contain the mass flow, the pressure of the interface flow-upwards and all units which are transported with the flow, see table 1. For mixtures of fluids the vector $C[n]$ describes the dimensionless mixture composition in kg/kg.

An interface with the temperature T and the heat flow Q exists for the convective and radiative heat exchange. The mechanical interface to pumps, turbines and compressors is identical to the one of the Modelica library for rotatoric drives.

Standardized interfaces also exist between the material models and the models of preservation equations. Here the chose state variables, for example $[p,h]$ or $[p,s]$, are always input units for the material data model. In the case of pressure and entalphy as states, $[p,h]$, the required start values are $[T, \rho, \partial\rho/\partial h|_p, \partial\rho/\partial p|_h]$ and a variable for the phase (gas area, two phase area or fluid area).

4 Preservation Equations

The choice of variables which form the preservation equations for mass, energy and impulse, is very important for a fast simulation of thermohydraulic facilities, but such a choice unfortunately does not cover all cases in practice. Criteria are among other things the numerical stiffity and the availability of a fluid model fitting those preservation equations. Nearly all material models (see 5) build upon two of the units p, v or T, so that those are suitable to be used as state variables. But the mass balance cannot be formulated in liquid state in v respectively ρ, the resulting system is too stiff in case of pressure transients and thus inefficient to simulate. Outside the two phase area T and p form a suitable and often used pair, within the two phase area T and p are dependent on each other, thus one of the variables has to be replaced by, for example, entalphy h or the steam degree x.

An often selected choice for state variables are pressure and entalphy. In the case of the above motivated method of displaced grids the mass and energy balances can be transformed intro two differential equations for p and h (see [6],S. 185 ff):

$$\frac{dp}{dt} = \left(\frac{\partial p}{\partial m}\right)_{e,V} \Delta\dot{m} + \left(\frac{\partial p}{\partial e}\right)_{m,V} \Delta\dot{e} + \left(\frac{\partial p}{\partial V}\right)_{m,e} \frac{dV}{dt} \tag{1}$$

$$\frac{dh}{dt} = \left(\frac{\partial h}{\partial m}\right)_{e,V} \Delta\dot{m} + \left(\frac{\partial h}{\partial e}\right)_{m,V} \Delta\dot{e} + \left(\frac{\partial h}{\partial V}\right)_{m,e} \frac{dV}{dt} \tag{2}$$

$$\left(\frac{\partial p}{\partial m}\right)_{e,V} = \frac{\rho + \left(\frac{\partial \rho}{\partial h}\right)_p h}{\left(\frac{\partial \rho}{\partial p}\right)_h \rho + \left(\frac{\partial \rho}{\partial h}\right)_p} \quad \text{etc.} \tag{3}$$

All units in formula 1 and 2 refer only to the considered volume, except the energy and mass balances $\Delta\dot{m}$ and $\Delta\dot{e}$:

$$\Delta\dot{m}_K = \sum_i \dot{m}_i \tag{4}$$

$$\Delta\dot{e}_K = (\forall(\dot{m}_i < 0) \sum_i \dot{m}_i h_K) + (\forall(\dot{m}_j > 0) \sum_j \dot{m}_j h_{j,\uparrow}) + \sum_k \dot{q}_k \tag{5}$$

The mass flows are being calculated on the limits of the control volume and are directly summed. The energy flows transport in case of a turn of the flow direction the flow-upwards entalphy beyond the balance limit; $h_{j,\uparrow}$ stands for the flow-upwards existing entalphy at the interface. The equations 4 and 5 are encapsulated in an own class to program only a version of the preservation equations for control volumina with a differing number of interfaces only once.

This form of balance equations is valid for the entire state space of a fluid and is moreover in the fluid area not numerically rigid. The latter term in 1 and 2 is only needed for control volumina of variable size.

One of the most important and, up to now, very early in the modeling process made decisions is, whether the impulse balance shall be calculated static or dynamic. The symbolic equation translation of Modelica allows for a combination of the dynamic and static form of impulse balance in one model, a parameter which is defined at compilation time, decides which form to use. The impulse balance for one phase or homogeneous modeled two phase flows is:

$$\frac{dG}{dt} = \underbrace{-w\left(\frac{\partial G}{\partial z} + \rho\frac{\partial w}{\partial z}\right)}_{\Delta G\ konvektiv} - \frac{\partial p}{\partial z} + \rho g cos(\phi) + \underbrace{\left(\frac{\partial p}{\partial z}\right)_R}_{\Delta p\ Reibung} \quad (6)$$

with ϕ as angle to the gravitation direction. The velocity w is being calculated on the limit of the control volume with the flow-upwards density of the continuity equation $\dot{m} = \rho w A$ with the profile area A of the volume. The two parameters **dy** and **gr** which are only changeable during compilation time determine the assumptions of the impulse balance:

$$\mathbf{dy}\left[\frac{dG}{dt}\right] = \mathbf{dy}\Delta G - \frac{\partial p}{\partial z} + \mathbf{gr}[\rho g cos(\phi)] + \left(\frac{\partial p}{\partial z}\right)_R \quad (7)$$

If $\mathbf{dy} = 0$ as well as $\mathbf{gr} = 0$, then the dynamic impulse balance reduces itself to a static pressure loss relationship. After sorting the equations:

$$\frac{\partial p}{\partial z} = \left(\frac{\partial p}{\partial z}\right)_R \quad (8)$$

With the help of this parameterization one version of library models can be utilized for both, slow and by thermal events defined processes and for fast by hydraulic effects dominated processes with time constants in the range of milliseconds.

The Modelica library for thermohydraulics contains furthermore a so-called inhomogeneous model for two phase flows in which the velocities of the phases are different and calculated via two coupled impulse balances. This model delivers for fast processes in vaporizers clearly better results, because here the friction between the phases has an important influence of the entire dynamics.

5 Thermodynamical Material Properties

The availability of fast and precise models for the thermodynamical properties of fluids is a notorious problem of the dynamic simulation of thermohydraulical facilities, as soon as models with higher requirements as ideal gas or incompressible fluids are needed. In many areas standardized formulations (IAPWS-IF97 for water according to [13]) exists or de-facto standards (the routines of NIST[2]), whose accuracy is also expected for the dynamic model creation. Some of these routines can be directly programmed in Modelica (for example for the steam boards of the IAPWS-IF97), others, like the cooling material routines of NIST, can be used via the interface for calls of external functions in Modelica.

[2] USA National Institute of Standards and Technology.

The available routines and nearly all in the literature described models (see for example [12]) are tailored to be called from static calculations. In dynamic calculations additional direction derivations of the material properties are needed, so that the existing formulations have to be extended. In the Modelica library "thermohydraulics" this will be included for water [13], for cooling material according to the NIST REFPROP database [7], ideal gases [2] and for cubical state equations of the form $F(p,v,T) = 0$ like the Redlich-Kwong-Soave equation.

Two kinds of calculations within the material value calls are iterative, they have to be executed in every control volume at every time unit and are hence very time consuming:

- the calculation of phase equilibrium and
- iterations, to find with the help of dynamic states the according point in the state plane, created by other thermodynamical units

Both kinds of iterations have to be avoided at all costs for dynamic simulations, whereas a small loss of accuracy is allowed.

The calculation of the phase limit curves between the one and two phase area at a given base formulation is executed iterative on the basis of thermodynamical criteria such as the Maxwell criterion. The numerical break error of the iteration causes frequent switching[3] at the lines of boiling and melting (see figure 2) and maybe even to a "sliding mode" behavior of the simulation. To avoid this, the phase limits, which have numerical rounding errors, have to be replaced by unambiguous switch functions for dynamic simulations. For this cubic splines or rational approximation functions, which are calculated with iterative original routines before the simulation and which approximate the original function, are suitable. With the values on both phase limit curves all states within the two phase area are calculated non-iterative. Those approximations are very exact and of one to two grades faster than the iterative functions. For the steam boards of the IAPWS-IF97 such an approximation of the steam pressure curve already exists. In the one phase area it is more demanding to replace iterative formulations by explicit ones, because here all units have to be calculated as function of both states. Near the phase limits, see for example the dark shaded area in figure 2, the material values can be approximated quite exactly with Taylor-approximations of first or second order. The included partial derivations are being calculated with the help of a computer algebra program. For cooling material cycles which are always driven near the two phase area the faults of such approximations are always below 0.5 %.

A formulation which is based on state variables — here p and h — has hence major advantages in calculation speed. Of the technical interesting materials such a formulation only exists for water within IAPWS-IF97. This is realized as explicit inversion function $T(p,h)$ which then is used in the basic Gibbs-function $g(p,T)$.

For the calculation of the thermodynamical properties of pure fluids several possibilities exist, which only can be briefly handled here. The thermal state equation is a

[3] or: discontinuity sticking

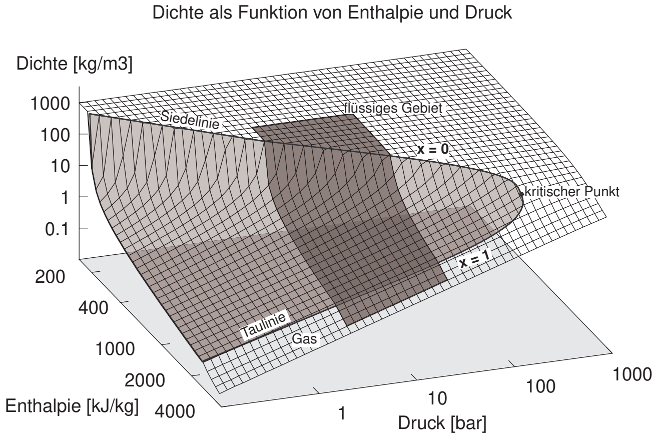

Figure 2: Zustandsfläche von Wasser, Dichte über Enthalpie und Druck. Nach [8].

function for the specific volume $v(T,p)$. For the calculation of enthalpy and entropy and additional function of the specific heat capacity in ideal state $c_p = c_p(T)$ is needed.

Further formulations are the canonical state equations for the *Helmholtz-function* $f = f(T,v)$ on which for example the REFPROP database is base and the *Gibbs-function* $g = g(T,p)$ which is used in IAPWS-IF97. The IAPWS-IF97 uses inverse functions $T = T(h,p)$ and $T = T(s,p)$, which can be even employed to explicitly calculate this combinations. Canonical state equations contain all information for the calculation of all thermodynamical relations. Those can be calculated from the derivations of the Helmholtz-function respectively the Gibbs-function, details see [1].

Another method to explicitly calculate the material values also in the one phase are is the two-dimensional spline interpolation based on the original functions.

Like figure 2 shows, the used derivations $(\partial\rho/\partial h)_p$ and $(\partial\rho/\partial p)_h$ are non-continuous at the phase limits: this has to be modeled as discrete switch functions for an efficient simulation. Whether if material values are modeled as external functions or are modeled in Modelica, whether they are calculated iteratively or explicitly, all material data models consist of the following components:

- A implicit defined interface (no connector, only via the attributes input and output) mit den dynamische Zuständen p, h als Eingangsgrößen und allen anderen benötigten Größen als Ausgänge.
- A discrete switch function for the phases.
- Two to three models, each of them modeling one phase.

All material data models which follow this scheme, are able to be exchanged randomly without influencing the other model parts. Because of the symbolic handling of equations before the simulation no redundant calculations are made, even for very simple material models like ideal gases or incompressible fluids.

5.1 Structure of the Thermohydraulics Library

In subject-specific simulation programs like APROS [5] the special knowledge about a field is hard-coded into the program code. In Modelica the knowledge is programmed as model libraries and thus also accessible for the user. The important abstract concepts of a field are suitable for a basic structure of the model library:

- abstract model parts (`partial model`), which cannot be simulated by themselves, encapsulate the equations and relations which shall be independently parameterizable in the final model.
- Typical components of thermohydraulic facilities are being put together of model parts. They can be directly assembled to facilities.

Within computer science so-called “design patterns” have been tried and tested for structuring object-oriented software (see for software in [4]): *Design Patterns* are informal described and frequently reappearing structural characteristics which increase the possibility for reuse of software components in certain applications. Also in modeling of physical systems *Design Patterns* can be found and are successfully used. The most known example for such a pattern within the field of process technology is the separation of machine and medium, for example described in [9]. With the help of class parameters of Modelica such a separation can be made even more consequent compared to the description in [9] of OMOLA.

In the Modelica™ library thermohydraulics the following “design patterns” are used:

Thermodynamical balance space: Basic physical assumptions of the model are parameterized on the level of a control volume, for example compressible/incompressible fluids. The exchangeability between pure and mixed material models is also realized on this level.

Fluid-machine separation: Consequent separation of the material data from the remaining description.

Performance profile module: Empirical data and device performance profiles, which can be described in tables or approximation functions, exist in different application areas, for example pump performance profiles.

Device parameter record: On this level all required parameters for one device type are summarized in one `record` and propagated further to the model parts, for example geometrical data.

The most important concept for the parameterization of thermodynamical models is the control volume, which also includes the interface between the elements. In a control volume a number of properties are summarized:

- One consistent set of preservation equations for a thermodynamical balance space is created. Basic assumptions like incompressibility are valid for one type of balance space.
- An exchangeable material data component which is compatible to this assumptions.
- Making the balance space one-dimensional discrete, one element modeling a balance space in concentrated parameters.

The discretization is based on the “finite volume method” and the request for consistent solvable equation systems at an arbitrary graphical connection of the library elements leads compulsively to a schema which again contains the “displaced grids” of the finite volumes: elements, which can store mass and energy, “stores” (calculation of the intensive units), are alternating with the elements in which the mass flow is calculated from pressure differences, “reducer”[4] (extensive units). To be able to simulate branches and joins with arbitrary flow direction, also the center of a branch is modeled as store element. For the simplification of the graphical joins a default direction is introduced in which the reducing element is aligned flow-downwards by definition. If all elements with uniform direction are connected along the default flow direction, an equal number of equations and unknown units are always the result of connecting arbitrary thermohydraulic networks, assuming that the conditions — mass flow and fluid state at open entries and pressure and fluid state at open exits — are set.

Similar *Design Patterns* like the thermodynamic balance space of thermohydraulics can be found in other areas of physical modeling and can be used for flexible parameterization: performance profiles and parameter sets for complex devices are usable in all applications.

5.2 Class Parameters and Multiple Inheritance

Modelica offers several possibilities for reusing once written program code respectively to shape models in a way that they are suitable for reuse.

1. Inheritance, also multiple inheritance with the keyword **extends**.
2. Class parameters, declared with **replaceable** and later on exchangeable with **redeclare**.

A class parameter is a placeholder for models whose **public** declared variables and components are equal to those of the default model, which is occupied by a default model. Furthermore a default model must not have **private** variables, thus it is a minimal declaration of the interfaces and visible parts which all have to contain instantiated models for class parameters. Class parameters can be combined in different ways with Modelica’s other language elements:

1. For one class parameter a component is declare whose class is then changeable.
2. One class parameter is used as base class for inheritance, so that the parent class contains a within certain conditions exchangeable class. The minimal demands on this class are defined by the standard class (in the example further down BaseFluid_ph).

Both possibilities are demonstrated using the same default model — one material model with pressure p and enthalpy h as dynamic states:

```
model BaseFluid_ph    // minimales
  input Pressure p;   // Vorgabe-Modell
  input Enthalpy h;
  output Temperature T;
  output Density d;
end BaseFluid_ph;
```

[4] Here a reducer is an arbitrary element in which pressure increase and decrease are measured)

This material model demonstrates both kinds of use in a control volume:

Material as component	Material as one of the parent classes

partial model CVolume1
replaceable class
Medium=Fluid_ph;
 	
Medium fluid;
equation
der(fluid.p) = ...;
der(fluid.h) = ...;
end CVolume1;
	partial model CVolume2
	replaceable class
	Medium=Fluid_ph;
	replaceable class
	States=States_ph;
	extends Medium;
	extends States;
	// States has equations
	// for the states
	end CVolume2;

Inherited models for water/steam	
model WaterVolume1 **extends** CVolume1(**redeclare model** Medium = IF97_ph); **connect**(fluid.T,T); **connect**(fluid.d,d); **end** WaterVolume1;	**model** WaterVolume2 **extends** CVolume2(**redeclare model** Medium = IF97_ph); **end** WaterVolume2;

Both models cannot be directly instantiated because the default class contains no equations for T and d. The essential difference of the application is that the variables of the components have to be accessed via the dot-notation. The method of combining class parameters and multiple inheritance offer additionally a higher flexibility because the material model can here be combined with different state models, for example for volumina of fixed or variable sizes. To complete those classes, the abstract base classes for an arbitrary medium are replaced by the one for water according to IAPWS/IF97 (see [13])

5.3 Basic Elements and Components

In table 2 the most important basic elements for thermohydraulic systems are shown. This basic elements are assembled from primitive components via multiple inheritance and class parameters, which alone make no model which can be simulated, for example material data functions or pressure loss relations. Thus the basic elements are very flexible. For technically important materials like water/steam, inherited libraries with a fixed fluid exist.

The following primitive components have been used:

Interfaces: Interfaces for convective transport of pure material and mixtures as well as heat transition.

Balance equations: Preservation equations for mass, energy and impulse in the variables pressure p, enthalpy h and mass flow density G.

Material data: Calculation of all of the fluid properties which are derived from the states: temperature T, density ρ and partial derivations of the density.

Pressure changes: Empirical equations for pressure loss, pump and valve performance profiles.

Heat transition: Empirical equations for heat transition.

Interfaces		
Reinstoffe (Mixtures)		Variable — calculated in class $\dot{m}$ — *States* p — *States* ρ_l, ρ_r — *Medium* h_l, h_r — *States* $(C[:]_l, C[:]_r)$ — *States*
Wärme-transition	[n]	Variable — calculated in class $\dot{q}$ — *HeatTransfer* T — *HeatStore*
Basic elements		
preservation equations	[n]	$\frac{dp}{dt} = \left(\frac{\partial p}{\partial m}\right)_{e,V} \Delta\dot{m} + \left(\frac{\partial p}{\partial e}\right)_{m,V} \Delta\dot{e}$ $\frac{dh}{dt} = \left(\frac{\partial h}{\partial m}\right)_{e,V} \Delta\dot{m} + \left(\frac{\partial h}{\partial e}\right)_{m,V} \Delta\dot{e}$ $\frac{dm}{dt} = \frac{1}{\Delta z}\left(\dot{I}_1 - \dot{I}_2 + \mathbf{F}_g + \mathbf{F}_p + \mathbf{F}_s\right)$
material properties	[n]	$\rho = f(p,h) \quad \left(\frac{\partial \rho}{\partial p}\right)_h = f(p,h)$ $T = f(p,h) \quad \left(\frac{\partial \rho}{\partial h}\right)_p = f(p,h)$
pressure loss	[n]	$\Delta p_{friction} = f(\dot{m}, \xi, \rho)$
heat storage	[n,1]	$mc_p \frac{dT}{dt} = \dot{q}_1 + \dot{q}_2$
adiabates controll volume	[n]	Assembled from: Preservation equations Material properties

Table 2: Selection from the library of thermohydraulic basic elements.

Components		
Pumpe	[1]	$\Delta p = f(n, \dot{V})$
Compressor	[1]	$\Delta p = f(n, \dot{V})$
multi-leveled turbine	ω,τ [n]	$\Delta p[i] = f(n, \dot{m}[i])$
Ventil	[1]	$\Delta p = f(A, \Delta p, \dot{V})$
T-Ventil 1	[1]	$\Delta p = f(\Delta p_0, \dot{V}_1^0, \dot{V}_2^0)$
T-Ventil 2	[1]	$\Delta p = f(\Delta p_0, \dot{V}_0)$
Rohr	[n]	Assembled from: n-discrete control volume pressure loss heat transition
steam drum	drum	Assembled from: 2-phase material properties drum model 4 interfaces
counterflow heat exchangers	WT-set [n]	Assembled from: 2-pipes heat storages WT-parameter set

Table 3: Selection from the library of thermohydraulic components.

Heat storage: Heat storages describe solid structures and walls which are in contact with the flow elements.

Except heat storages all elements contain the flow interfaces and are assembled in several steps through multiple inheritance to more comprehensive models: Interfaces, balance equations and material data form an adiabatic control volume (balance space), together with a pressure change and heat transition element a flow channel of a heat exchanges is being created.

Thus every element of the library consists of at least the following components:

- 2 or 3 flow interfaces,
- a control volume element with the preservation equations ("storage"),
- equations for pressure change ("reducer").

Furthermore there are thermal solid body storages and heat transition relations. The parameterization of the models is done via the *design patterns*: balance space, separate material description, performance profile and device parameter set, which are described in section 5.1

The realization of *design patterns* in Modelica is presented with an example of exchangeable interfaces for control volumes and their use in a universal applicable model for pumps.

```
record ThermalProps
  Density d;
  Enthalpy h;
end ThermalProps;
```

```
connector FlowPort2Dir_ph
  flow MassFlow mdot;
  Pressure p;
  ThermalProps up,down;
end NFlowPort2Dir_ph;

connector NFlowPort2Dir_ph
  extends FlowPort2Dir_ph;
  parameter Integer n;
  ChemicalProps Cup(n), Cdown(n);
end NFlowPort2Dir_ph;

partial model FlowTwoPort
  replaceable connector Ports=FlowPort2Dir_ph;
  Ports in,out;
end FlowTwoPort;

model BaseCV_ph
  extends  FlowTwoPort;
  ... // preservatio of mass, energy and impulse
end BaseCV_ph;

model NFlowCV_ph
  extends BaseCV_ph;
  redeclare connector Ports=NFlowPort2Dir_ph;
  ... //equations for chemical composition
end  NFlowCV_ph;
```

If the simplest model of a control volume is used as base class for pumps in the state variables P and h, here BaseCV_ph, a more complex, inherited control volume can be used in the derived classes which for example is valid even for mixtures of dynamic compositions.

The interfaces, *FlowPort2Dir_ph* for pure materials and *NFlowPort2Dir_ph* for mixtures are part of the control volume and are exchanged automatically.

```
model GenericPump
  replaceable model CV=BaseCV_ph;
  extends CV;
  replaceable model PumpEq=PumpEq1;
  extends Pumpeq;
end GenericPump;

model PumpEqMix
  extends PumpEq1;
  redeclare model Ports=NFlowPort2Dir_ph;
end PumpEqMix;

model MixPump
  extends GenericPump;
  redeclare CV=NFlowCV_ph;
  redeclare PumpEq=PumpEqMix;
end MixPump;
```

The equations which implement pump performance profiles are also exchangeable. They contain the same interfaces like the control volume classes. Through multiple inheritance those two sets of interfaces fuse together to one, e.g. that the interfaces of the model parts PumpEqMix and NFlowCV_ph have to be identical [5].

This kind of application of multiple inheritance offers high flexibility, because the kind of transported media and the pump performance profile are independently selectable, but it can also lead to error messages which are difficult to explain. If for example in the class *MixPump* it has been forgotten to declare *PumpEq*, the different interfaces for the parent classes *CV* and *PumpEq* collide. This problem can be addressed for normal users with the help of ready-made, consistent part-libraries. But the strong dependency between the models is being compensated in clearly structured and well-documented libraries through very compact and non-redundant program code. The thermohydraulics library will be freely available in spring 2000 from the homepage of the Modelica group [10].

5.4 Example: a cooling material cycle process

As example for the application possibilities of the library a Rankine-cycle of a refrigerator is being presented. Die structure of the model and the state changes in the $log(p) - h$ diagram is shown in figure 3. Model parts for evaporator and condensator can be chosen on the cooling materials behalf as either concentrated or distributed, on the rooms respectively cooling loads behalf concentrated parameters are always sufficient. Except the capillary pipe, which can be simplified described as a reducing valve, and (as the simulation shall be significantly speeded up) the material values of the cooling material, all other parts of the model are parts of the standard library: heat exchanger, thermal storages (environment, cooling room) and compressor. For the evaporator and the condensator an adapted device parameter set has to be modeled which projects the materials, surfaces and volumina of those components onto

[5] the model PumpEq1 implements the pump performance profile and is not shown here.

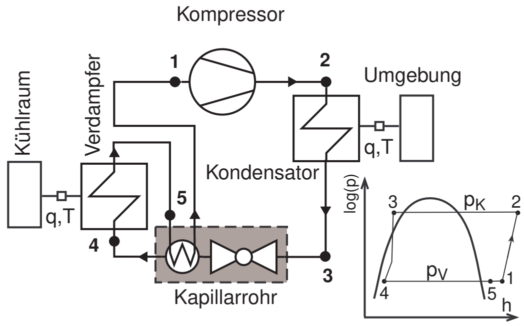

Figure 3: Simple model of a refrigerator

the parameters of the standard-one-pipe-model of a heat exchanger of the library. For the material data of cooling materials, an interface to the REFPROP-routines of NIST [7] is employed via the possibility to use external FORTRAN77 or C functions in Modelica. This is mostly the only adaptation work for a not existing variant of a device. This demonstrates the essential goal of the Modelica libraries: structuring models so that adaptation and maintenance work is reduced without changing the existing and tested program code.

Literatur

[1] BAEHR, HANS DIETER: *Thermodynamik*. Springer-Verlag, Berlin Heidelberg New York, 9. Auflage, 1996.

[2] BAEHR, H. D. und CH. DIEDERICHSEN: *Berechnungsgleichungen für Enthalpie und Entropie der Komponenten von Luft und Verbrennungsgasen*. BWK, 40(1/2):30–33, 1988.

[3] CULLIMORE, B., S. G. RING, R. G. GOBLE und C. L. JENSEN: *Sinda/Fluint Users Manual for Version 3.2*. Cullimore and Ring Technologies, Inc., Littleton, Colorado, 1996.

[4] GAMMA, ERICH, RICHARD HELM, RALPH JOHNSON und JOHN VLISSIDES: *Design Patterns*. Addison-Wesley, 1995.

[5] JUSLIN, K.: *Experience on mechanistic modelling of industrial processes with APROS*. Mathametics and Computers in Simulation, 39:505–511, 1995.

[6] KOLEV, NIKOLAY I.: *Transiente Zweiphasen-Strömung*. Springer-Verlag, Berlin, 1986.

[7] MCLINDEN, MARK O., SANFORD A. KLEIN, ERIC W. LEMMON und ADELE P. PESKIN: *NIST Thermodynamic and Transport Properties of Refrigerants and Refrigerant Mixtures—REFPROP, Version 6.0*. U. S. Department of Commerce, January 1998.

[8] MÜHLTHALER, GEORG: *Anwendung objektorientierter Simulationssprachen zur Modellierung von Kraftwerkskomponenten*. Doktorarbeit, Technische Universität Hamburg Harburg, 1999.

[9] NILSSON, BERNT: *Object-Oriented Modeling of Chemical Processes*. Doktorarbeit, August 1993.

[10] PACKAGE, MODELICA: *Homepage: http://www.modelica.org/*.

[11] PATANKAR, SUHAS V.: *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, 1980.

[12] REID, ROBERT C., JOHN M. PRAUSNITZ und BRUCE E. POLING: *The Properties of Gases and Liquids*. Mc Graw Hill, Boston, Massachusetts, 1987.

[13] WAGNER, W. und A. KRUSE: *Properties of water and steam; the industrial standard IAPWS-IF97 for the thermodynamic properties and supplementary equations for other properties*. Springer, Berlin, 1998.