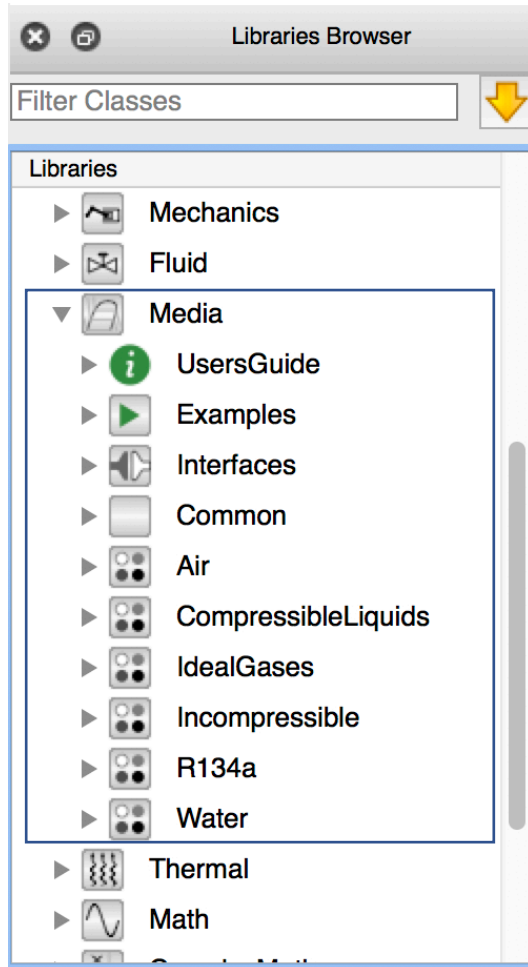


OpenModelica講習中級 Modelica.Fluidライブラリ解説

2. Modelica.Mediaライブラリ

2017年12月7日 田中 周(有限会社アマネ流研)

2. Modelica.Mediaライブラリ



Modelica.Media ライブラリ

物性に関する type, 定数、model、record、function を含む package を集めたものである。

- Air : 数種類の空気
- Water: 数種類の水
- IdealGases: 理想気体
 - SingleGases: 純物質 Ar, C2H4, CO2, ...
 - MixtureGases: 混合物質 天然ガスなど
- R134a: 冷媒などの 2 相流体
- Incompressible: 非圧縮性流体 潤滑油など
- CompressibleLiquids: 密度が温度・圧力と線形
- ...

Contents

Modelica.Media の全体的構成や構成要素について示す。

Modelica.Media ライブラリの構成

MediaExample1 水を温めて水蒸気にする（等圧過程）

MediaExample2 水を断熱圧縮する。水蒸気を断熱膨張させる。
（等エントロピ過程）

Modelica.Media ライブラリの全体構成(継承関係)



Modelica.Media の基本構成

物性値の型を表す type の宣言

```
<<package>> Interfaces.Types
...
<<type>> AbsolutePressure = ...
<<type>> Density = ...
<<type>> DynamicViscosity = ...
...
```

全ての物質の ベースパッケージ

```
<<partial package>> Interfaces.PartialMedium
<<constant>> ThermoStates
<<constant>> mediumName
<<constant>> singleState
...
}
```

②～⑤の実装

個々の物質の パッケージ

```
<<package>> XXX
<<constant>> ThermoStates=...
<<constant>> mediumName=...
<<constant>> singleState=...
...
```

①の再定義

```
<<replaceable>>
<<record>> FluidConstants
<<record>> ThermoDynamicState
<<partial model>> BaseProperties
<<partial function>> setState_pTX
...
<<partial function>> dynamicViscosity
...
```

交換可能な record, model,
partial function, ...

物性パッケージの主な構成要素

②ThermodynamicState
熱力学的状態を表すレコード

③BaseProperties
基本物性モデル

④setStateXXX
熱力学的状態
(ThermodynamicState)
を返す関数

⑤粘性率、熱伝導率
などの物性関数

①定数
物質名、成分名、参照状態
デフォルト状態などを示す定数

①定数 物質名、成分名、参照状態、デフォルト状態などを示す定数

PartialMedium の定数

個々の物質パッケージはこれらの定数をカスタマイズする

- 物質名と熱力学的変数特性
- 成分物質の情報
- 微小物質など付加的物質の情報
- reference state
- default state

定数名	型	デフォルト値	
mediumName	String	"unusablePartialMedium"	物質名
ThermoStates	IndependentVariables		T, pT, ph, phX, pTX, dTXのいずれか
singleState	Boolean		trueなら物性値が圧力に依存しない
substanceNames[:]	String	{mediumName}	成分名の配列
reducedX	Boolean	true	trueなら質量分率の和が1
fixedX	Boolean	false	trueならX=reference_X
nS	Integer	size(substanceNames, 1)	混合物の成分数
nX	Integer	nS	質量分率の要素数
nXi	Integer	if fixedX then 0 else if reducedX nS -1 else nS	独立な質量分率の要素数
extraPropertiesNames[:]	String	fill("",0)	付加的物質名(微小物質)
nC	Integer	size(extraPropertiesNames, 1)	付加的物質の成分数
C_nominal[nC]	Real	1.0e-6*ones(nC)	付加的物質の濃度
reference_p	AbsolutePressure	101325 [Pa]	圧力の参照値（基準値）
reference_T	Temperature	298.15 [K]	温度の参照値（基準値）
reference_X[nX]	MassFraction	fill(1/nX, nX)	質量分率の参照値
p_defalut	AbsolutePressure	101325 [Pa]	圧力のデフォルト値
T_default	Temperature	Conversions.from_degC(20)	温度のデフォルト値
h_default	SpecificEnthalpy	specifixEnthalpy_pTX(p_default, T_default, X_default)	比エンタルピのデフォルト値
X_default[nX]	MassFraction	reference_X	質量分率のデフォルト値

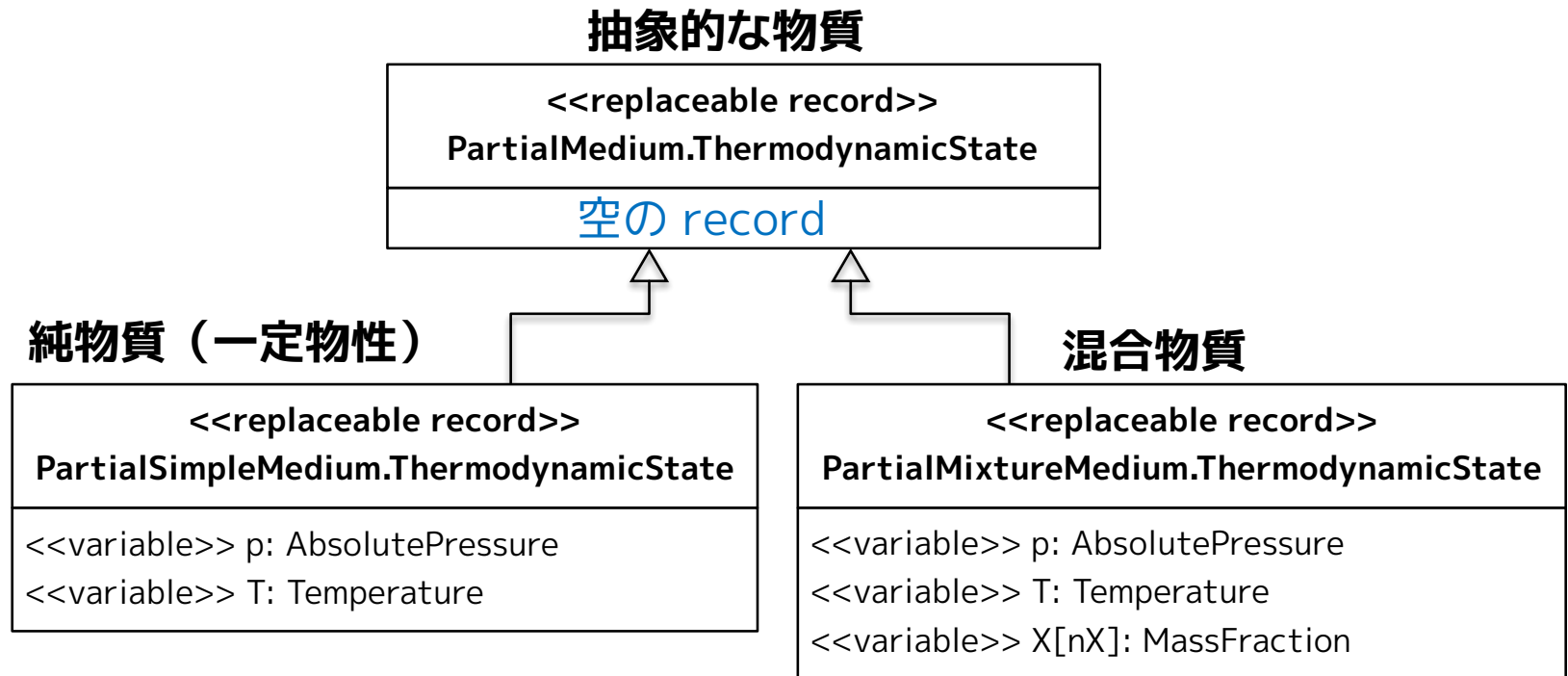
② ThermodynamicState

物質の熱力学的状態を決定するのに必要な変数の組を表す record

物質の熱力学的状態は、

2つの熱力学的状態変数と成分物質の質量分率で決定できる。

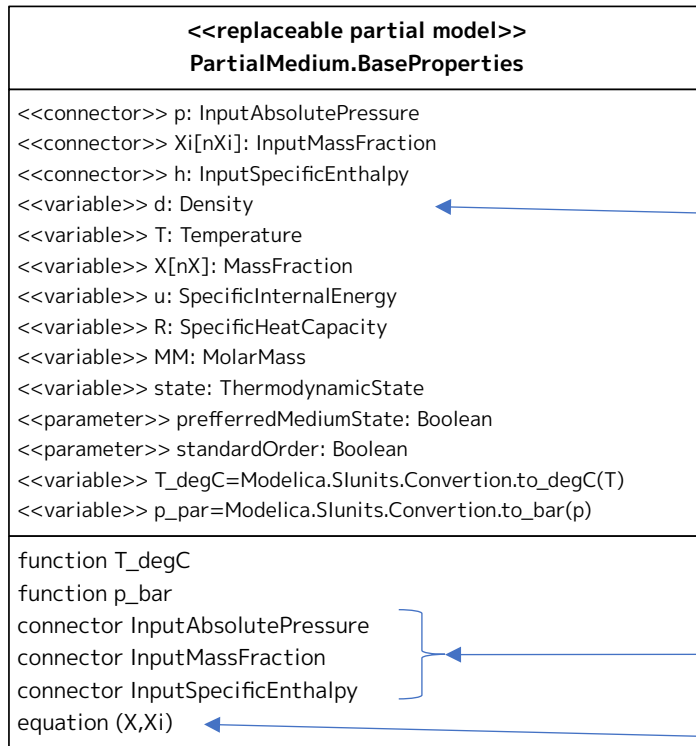
- 圧力 p 、温度 T 、密度 d 、比エンタルピー h 、内部エネルギー u のうち2つ
- 混合物質の成分の質量分率ベクトル $X[nX]$



物質の種類や使いやすさにより T , pT , ph , phX , pTX , dTX などを選ぶ。

③ BaseProperties

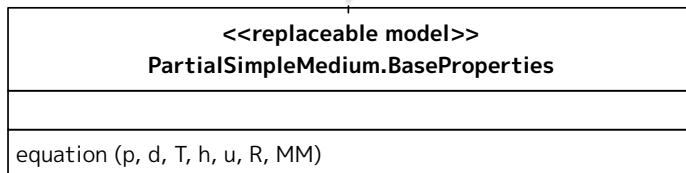
熱力学的状態変数、ガス定数、モル質量などの関係を表す model



p:圧力
d:密度
T:温度
h:比エンタルピー
u:内部エネルギー
R:ガス定数
MM:モル質量
X[nX]: 成分の質量分率
Xi[nXi]:独立な成分の質量分率
state: ThermodynamicState

p, h, X が内部的には入力コネクタになっている

XとXiの方程式が記述されている。



継承先のクラスで
7個の変数 p, d, T, h, u, R, MM に関する
7個の方程式が追加されている。
方程式の内容は物質の種類によって異なる。

④ setState_XXX と setSmoothState

異なる熱力学的状態変数の組み合わせから **ThermodynamicState** を返す関数

継承先のパッケージで **algorithm** を定義する。

- **setState_pTX(p, T, X)**
- **setState_phX(p, h, X)**
- **setState_psX(p, s, X)**
- **setState_dTX(d, T, X)**

p: 圧力
T: 温度
h: 比エンタルピ
s: 比エントロピ
d: 密度
X[nX]: 質量分率

setSmoothState(x, state_a, state_b, x_small)

2つの状態 state_a と state_b を $x \pm x_small$ の範囲で滑らかにつなぐ関数。x は 質量流量か圧力である。

⑤ 物性関数(その1)

record ThermodynamicState を引数として、物性値を返す関数

- *dynamicViscosity(state)*
- *thermalConductivity(state)*
- *prandtlNumber(state)*
- *pressure(state)*
- *temperature(state)*
- *density(state)*
- *specificEntalpy(state)*
- *specificInternalEnergy(state)*
- *specificEntropy(state)*
- *specificGibbsEnergy(state)*
- *specificHelmholtzEnergy(state)*
- *specificHeatCapacityCp(state)*
- *heatCapacity_cp(state)*
- *specificHeatCapacityCv(state)*
- *heatCapacity_cv(state)*
- *isentropicExponent(state)*
- *isentropicEnthalpy(p_sownstream, refState)*
- *velocityOfSount(state)*
- *isobaricExpansionCoefficinet(state)*
- *beta(state)*
- *isothermalCompressibility(state)*
- *kappa(state)*
- *molarMass(state)*

⑤ 物性関数(その2)

密度の偏微分を返す関数

- *density_derp_h(state)*
- *density_derh_p(state)*
- *density_derp_T(state)*
- *density_derT_p(state)*
- *density_derX(state)*

状態変数から他の状態変数を返す関数

- *specificEnthalpy_pTX(p,T,X)*
- *specificEntropy_pTX(p,T,X)*
- *density_pTX(p,T,X)*
- *temperature_phX(p,h,X)*
- *density_phX(p,h,X)*
- *temperature_psX(p,s,X)*
- *density_psX(p,s,X)*
- *specificEnthalpy_psX(p,s,X)*

- 斜線は、PartialMedium では algorithm が定義されていない partial function である。
- これらの物性関数は、オプション的な扱いであり、物質によっては実装不可能なものもある。全てが実装されていなければならないわけではない。

SimpleAir, DryAirNasa の継承関係

全ての物質モデルの
ベースパッケージ

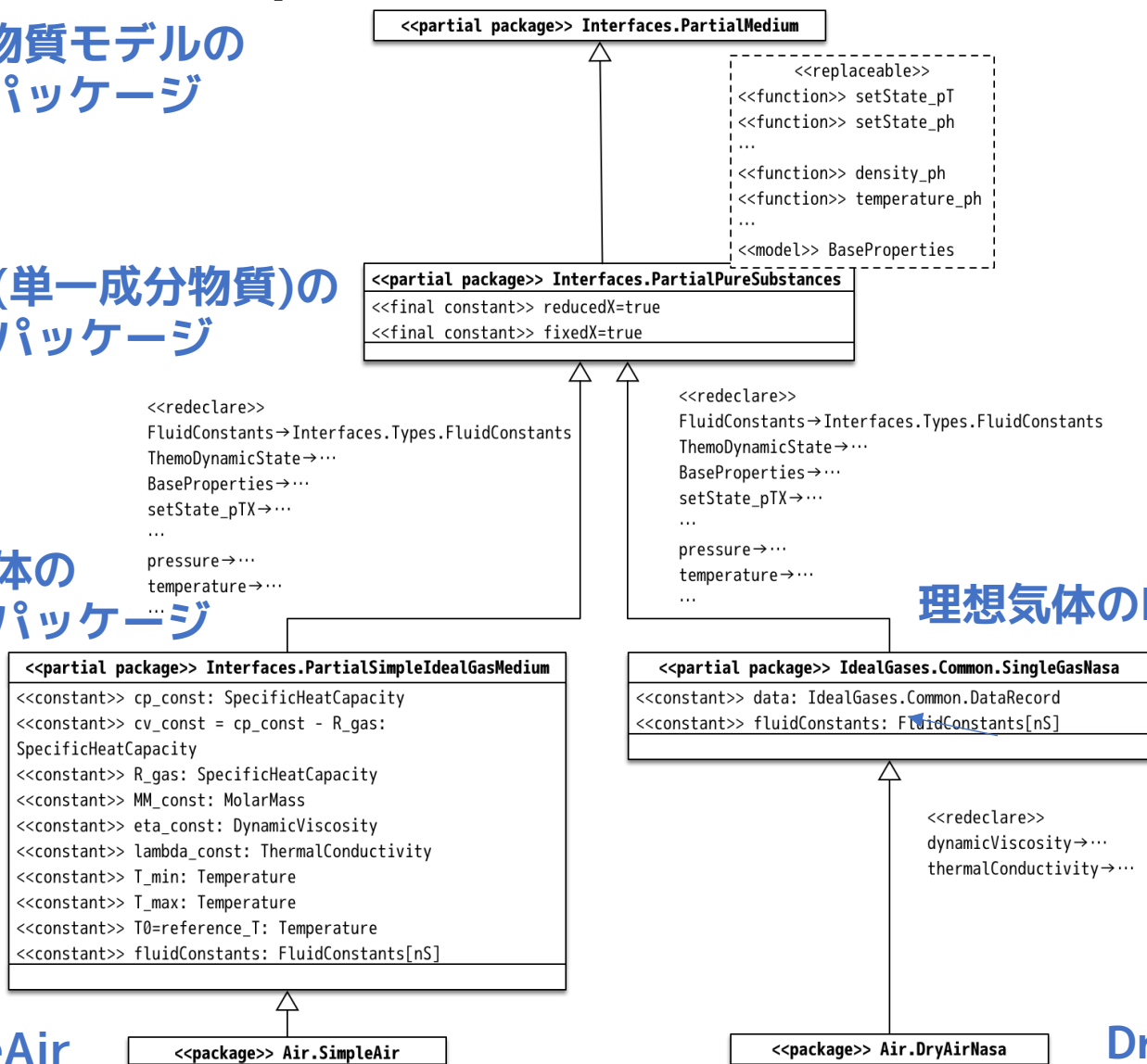
純物質(単一成分物質)の
ベースパッケージ

理想気体の
ベースパッケージ

理想気体のNASAモデル

SimpleAir

DryAirNasa

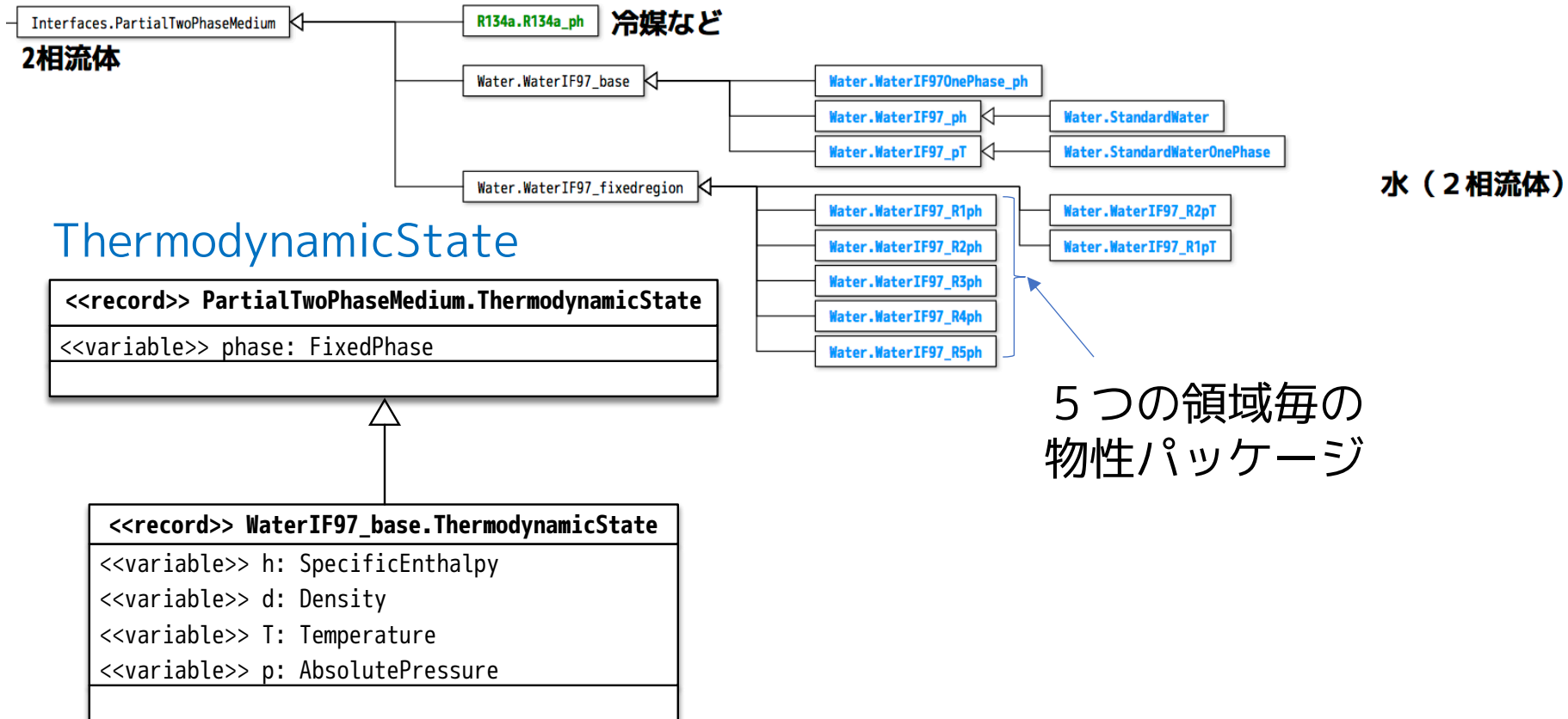


水（2相流体）のモデル

実用国際蒸気状態式(IAPWS-IF97)

The International Association for the Properties of Water and Steam <http://www.iapws.org/relguide/IF97-Rev.pdf>

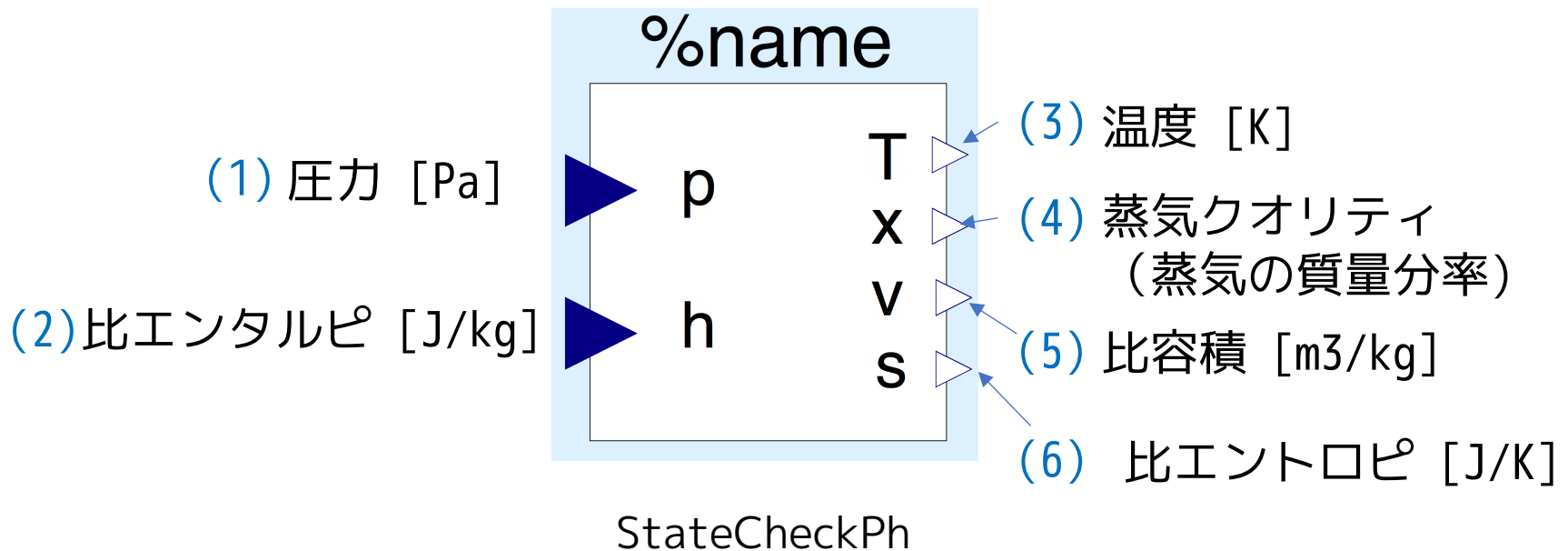
水の状態を 5 つの領域に分けて記述するモデル



MediaExample1

水を温めて水蒸気にする (等圧過程)

まず、圧力と比エンタルピを入力すると、温度、蒸気クオリティ、比容積、比エントロピを出力するモデルを作成する。



MediaExample1

StateCheckPh

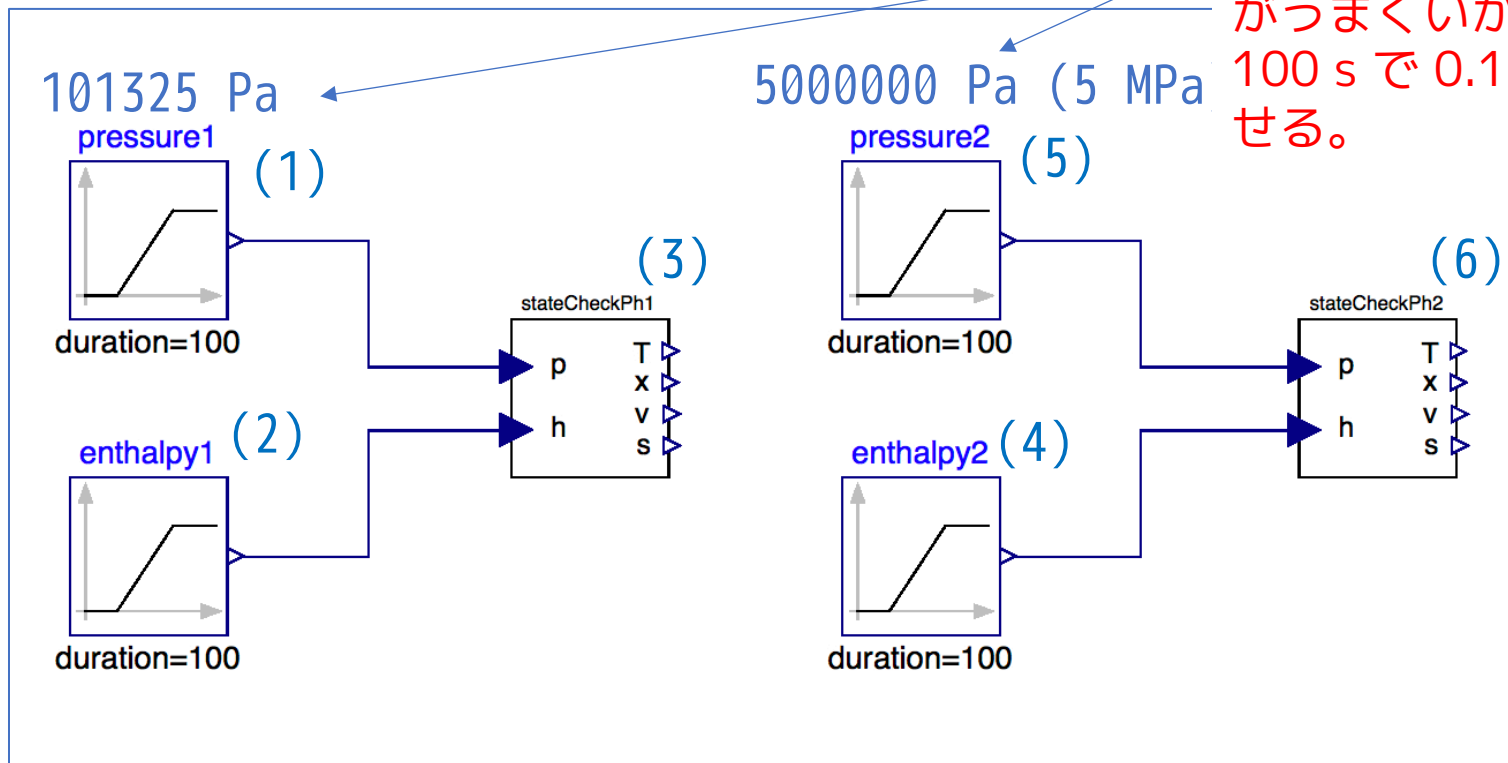
```
package MediaExample1

model StateCheckPh
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  Medium.ThermodynamicState state;
  Modelica.Blocks.Interfaces.RealInput p annotation( ...); (1)
  Modelica.Blocks.Interfaces.RealInput h annotation( ...); (2)
  Modelica.Blocks.Interfaces.RealOutput T annotation( ...); (3)
  Modelica.Blocks.Interfaces.RealOutput x annotation( ...); (4)
  Modelica.Blocks.Interfaces.RealOutput v annotation( ...); (5)
  Modelica.Blocks.Interfaces.RealOutput s annotation( ...); (6)
equation
  state = Medium.setState_ph(p,h);
  T = state.T;
  x = Medium.vapourQuality(state);
  v = 1/state.d;
  s = Medium.specificEntropy(state);
  annotation( ...);
end StateCheckPh;
```

MediaExample1

これを使って圧力を固定して比エンタルピを上げるモデルを作る
比エンタルピを100 s で 20000 J/kg から 3800000 J/kg まで変化させる。

本当は固定値だが結果のパラメトリックプロットがうまくいかないので100 s で 0.1 Pa 変化させる。



StateCheckTest1

MediaExample1

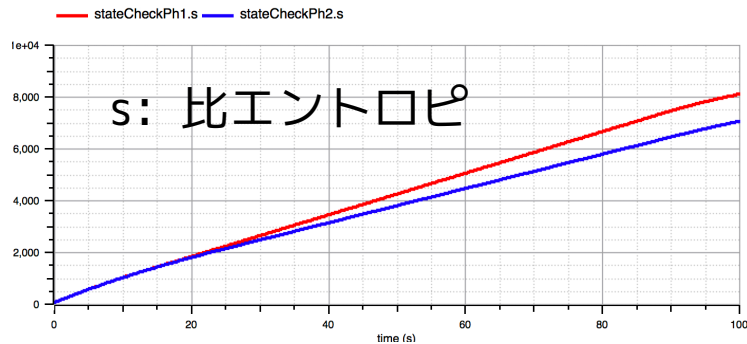
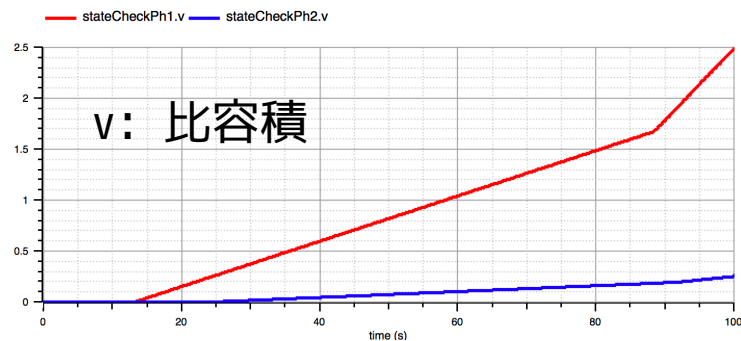
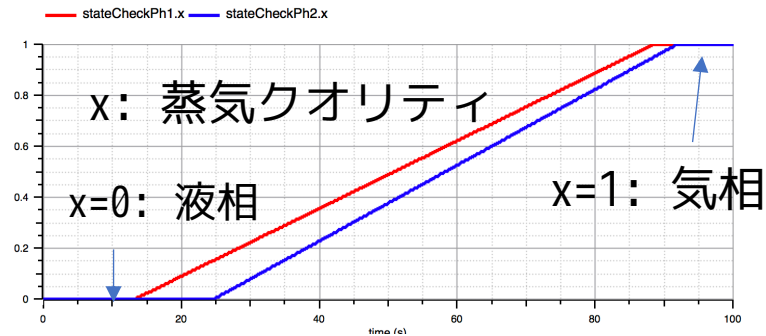
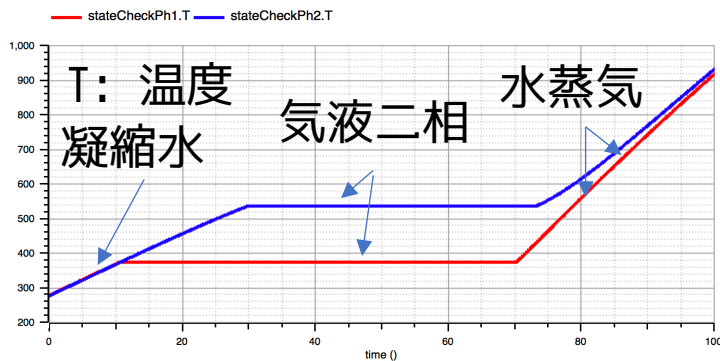
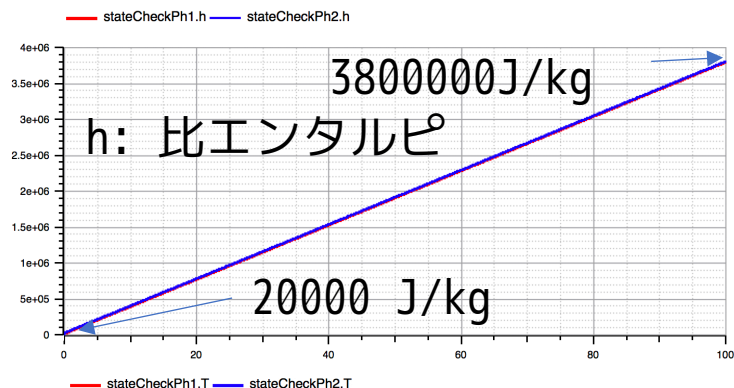
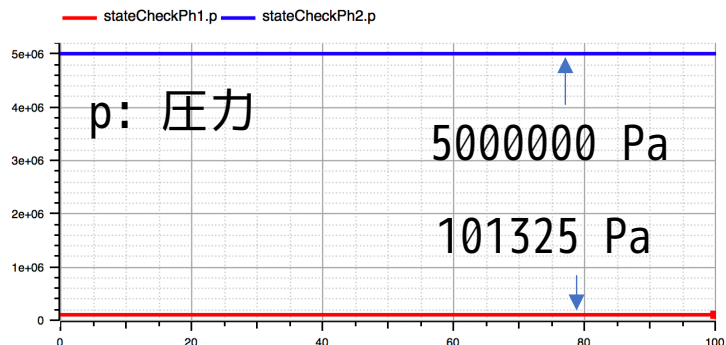
StateCheckTest1

```
model StateCheckTest1
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  Modelica.Blocks.Sources.Ramp pressure1(duration = 100, height = 0.1, offset = 101325, (1)
    startTime = 0) annotation( ...);
  Modelica.Blocks.Sources.Ramp enthalpy1(duration = 100, height = 3780000, offset = 20000, (2)
    startTime = 0) annotation( ...);
  MediaExample1.StateCheckPh stateCheckPh1(redeclare package Medium = Medium) annotation( ...); (3)
  Modelica.Blocks.Sources.Ramp enthalpy2(duration = 100, height = 3780000, offset = 20000, (4)
    startTime = 0) annotation( ...);
  Modelica.Blocks.Sources.Ramp pressure2(duration = 100, height = 0.1, offset = 5000000, (5)
    startTime = 0) annotation( ...); (6)
  MediaExample1.StateCheckPh stateCheckPh2(redeclare package Medium = Medium) annotation( ...);
equation
  connect(enthalpy2.y, stateCheckPh2.h) annotation( ...);
  connect(pressure2.y, stateCheckPh2.p) annotation( ...);
  connect(enthalpy1.y, stateCheckPh1.h) annotation( ...);
  connect(pressure1.y, stateCheckPh1.p) annotation( ...);
end StateCheckTest1;

annotation( ...);
end MediaExample1;
```

MediaExample1

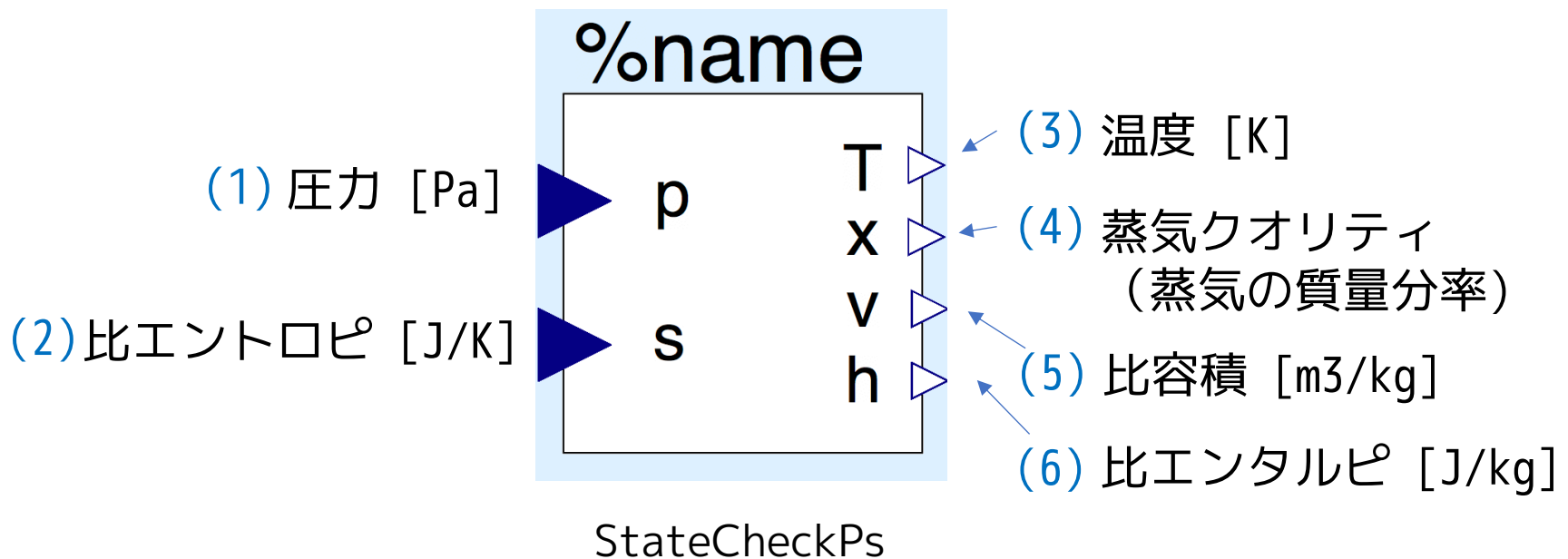
シミュレーション結果



MediaExample2

水を断熱圧縮する。水蒸気を断熱膨張させる。
(等エントロピ過程)

今度は、圧力と比エントロピを入力すると、温度、蒸気クオリティ、比容積、比エンタルピを出力するモデルを作成する。



MediaExample2

StateCheckPs

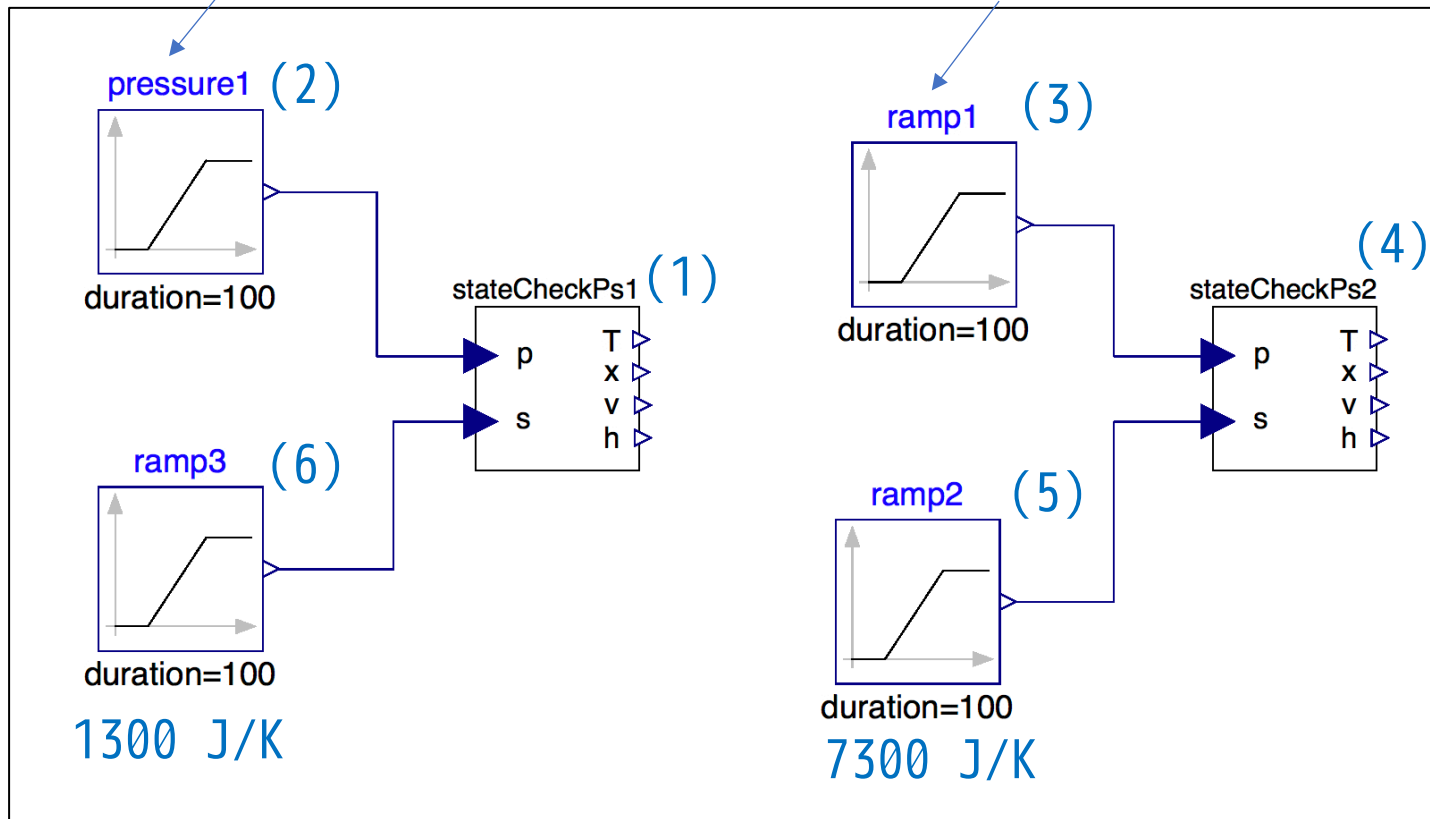
```
package MediaExample2
  model StateCheckPs
    replaceable package Medium = Modelica.Media.Water.StandardWater;
    Medium.ThermodynamicState state;
    Modelica.Blocks.Interfaces.RealInput p annotation( ...);      (1)
    Modelica.Blocks.Interfaces.RealInput s annotation( ...);      (2)
    Modelica.Blocks.Interfaces.RealOutput T annotation( ...);     (3)
    Modelica.Blocks.Interfaces.RealOutput x annotation( ...);     (4)
    Modelica.Blocks.Interfaces.RealOutput v annotation( ...);     (5)
    Modelica.Blocks.Interfaces.RealOutput h annotation( ...);     (6)
  equation
    state = Medium.setState_ps(p, s);
    T = state.T;
    x = Medium.vapourQuality(state);
    v = 1 / state.d;
    h = state.h;
    annotation( ...);
  end StateCheckPs;
```

MediaExample2

比エントロピを固定して圧力を変化させるモデルを作る。

100 kPa から 5 MPa
まで昇圧する。

5 MPa から 100 kPa
まで減圧する。



StateCheckTest2

MediaExample2

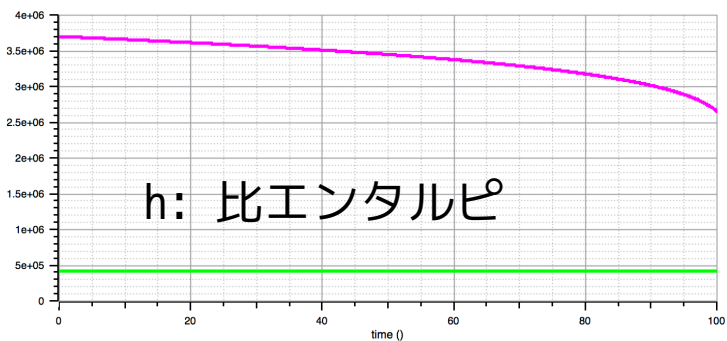
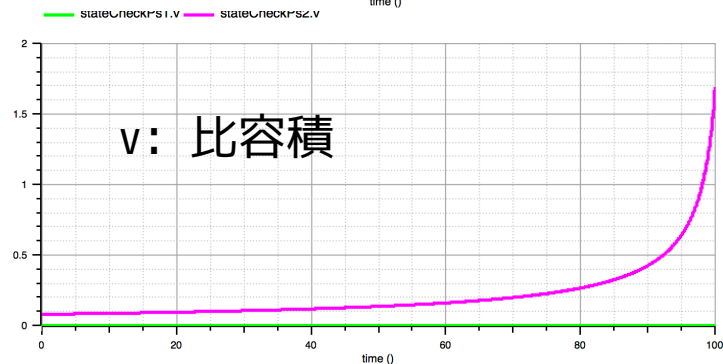
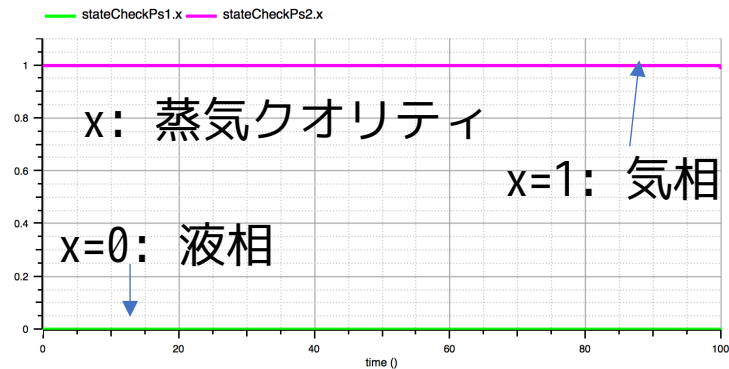
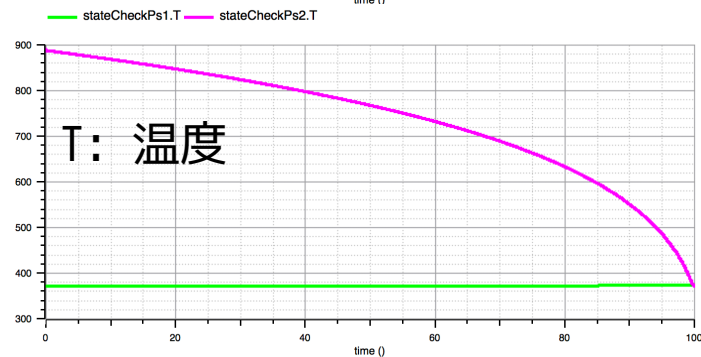
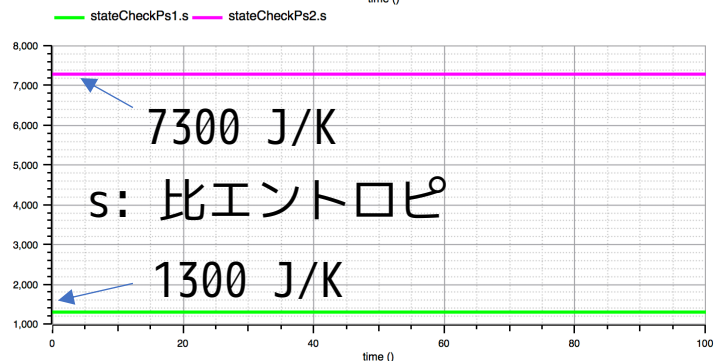
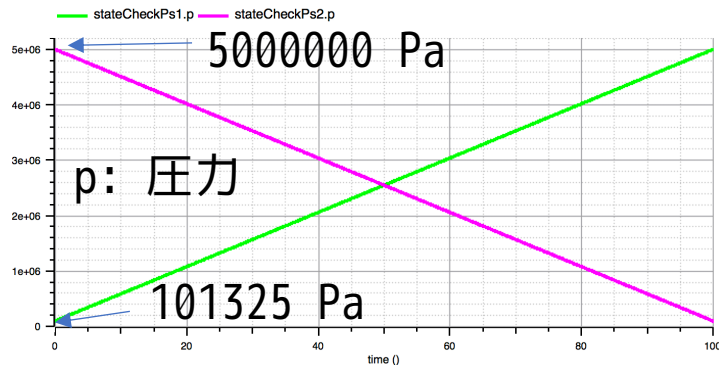
StateCheckTest2

```
model StateCheckTest2
  replaceable package Medium = Modelica.Media.Water.StandardWater;
  MediaExample2.StateCheckPs stateCheckPs1(redeclare package Medium = Medium) annotation( ...); (1)
  Modelica.Blocks.Sources.Ramp pressure1(duration = 100, height = 4900000, (2)
    offset = 100000, startTime = 0) annotation( ...);
  Modelica.Blocks.Sources.Ramp ramp1(duration = 100, height = -4900000, (3)
    offset = 5000000, startTime = 0) annotation( (4)
  MediaExample2.StateCheckPs stateCheckPs2(redeclare package Medium = Medium) annotation( ...);
  Modelica.Blocks.Sources.Ramp ramp2(duration = 100, height = 0.001, (5)
    offset = 7300, startTime = 0) annotation( ...);
  Modelica.Blocks.Sources.Ramp ramp3(duration = 100, height = 0.1, (6)
    offset = 1300, startTime = 0) annotation( ...);
equation
  connect(ramp3.y, stateCheckPs1.s) annotation( ...);
  connect(ramp2.y, stateCheckPs2.s) annotation( ...);
  connect(ramp1.y, stateCheckPs2.p) annotation( ...);
  connect(pressure1.y, stateCheckPs1.p) annotation( ...);
end StateCheckTest2;

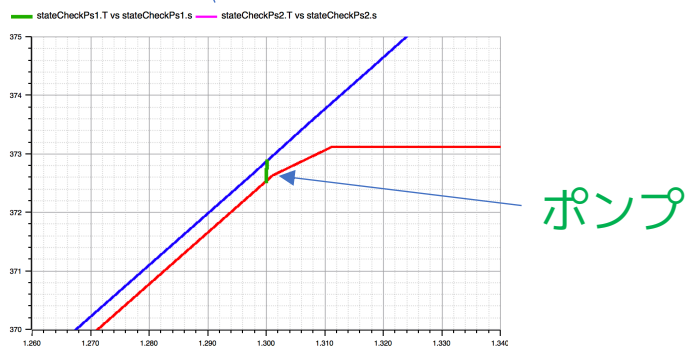
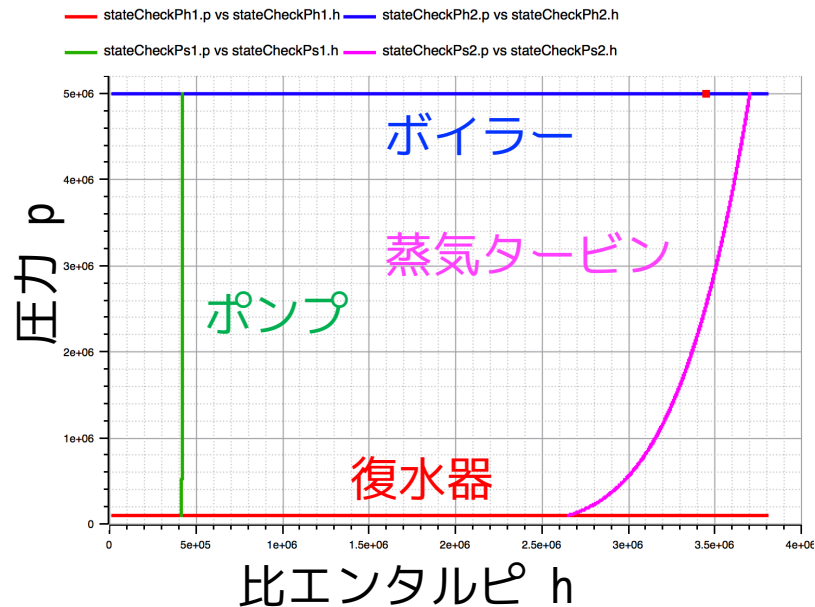
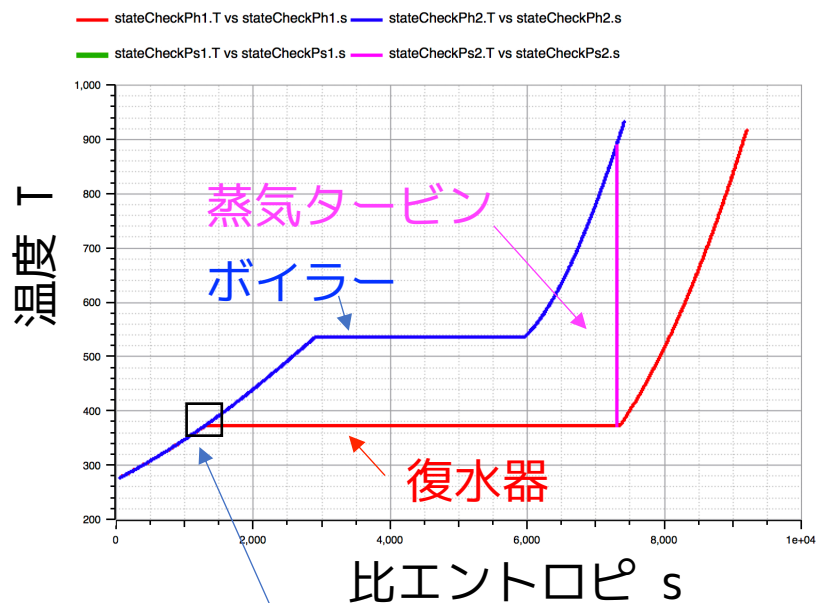
annotation( ...);
end MediaExample2;
```

ClassExample3

シミュレーション結果



MediaExample2



MediaExample1, MediaExample2 の結果をパラメトリックプロットしてみる。

閉じたループがランキンサイクルを表す。

まとめ

- Modelica.Media の全体構成や構成要素について述べた。
- 水（２相流体）について簡単なモデルを作り、物性値の参照方法を確認した。

- The purpose of this document is introducing Media and Fluid Libraries in the Modelica Standard Library (MSL). This document uses libraries, software, figures, and documents included in MSL and those modifications. Licenses and copyrights of those are written in next page.
- Copyright and License of this document are written in the last page.

Modelica Standard Library License

<https://github.com/modelica/ModelicaStandardLibrary/blob/master/LICENSE>

BSD 3-Clause License

Copyright (c) 1998-2018, ABB, Austrian Institute of Technology, T. Bödrich, DLR, Dassault Systèmes AB, ESI ITI, Fraunhofer, A. Haumer, C. Kral, Modelon, TU Hamburg-Harburg, Politecnico di Milano, and XRG Simulation
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



Copyright © 2017-2017 The Open CAE Society of Japan

This work is licensed under a Creative Commons
Attribution-NonCommercial 4.0 International License.

<http://creativecommons.org/licenses/by-nc/4.0/>

