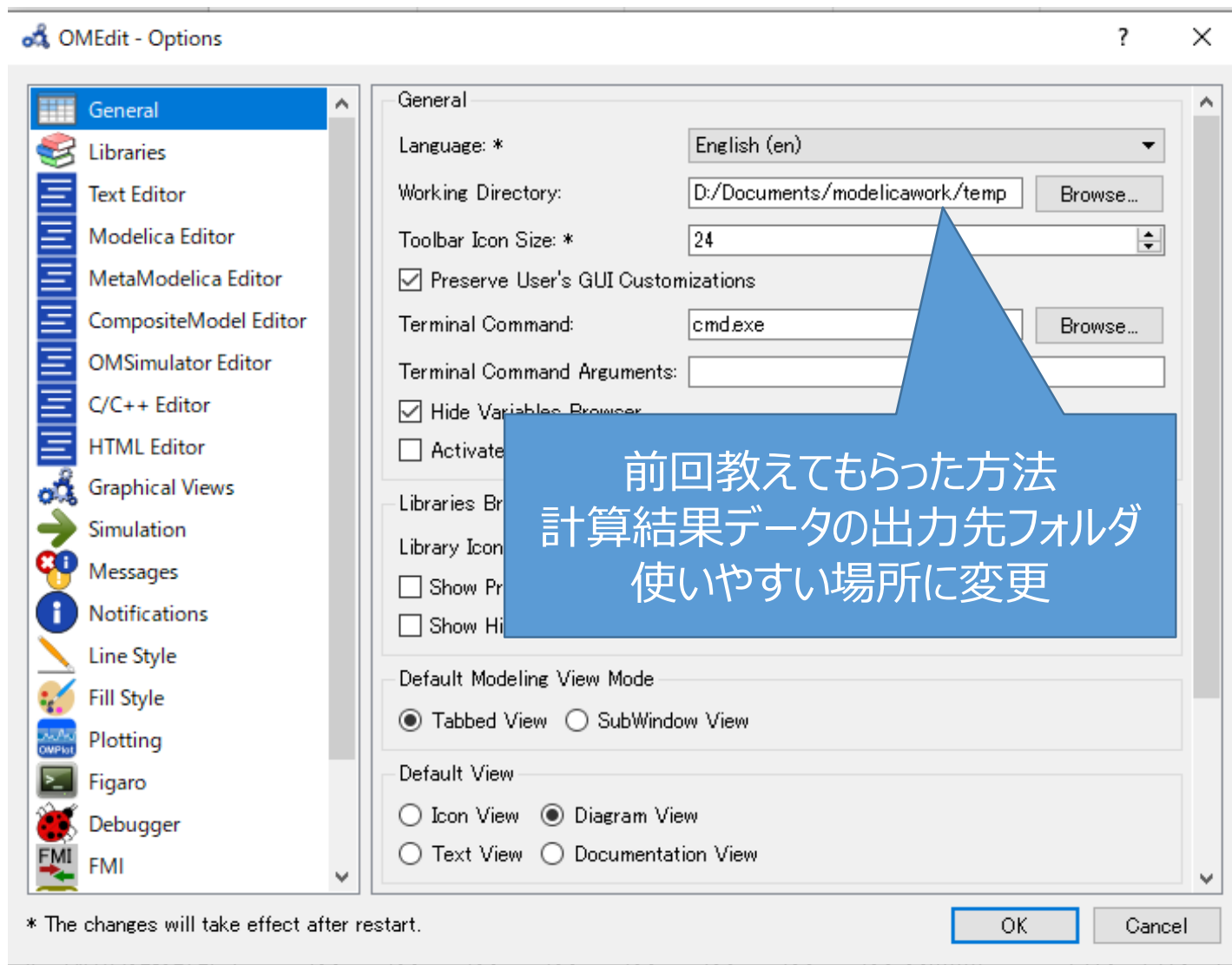


# Modelicaライブラリ勉強会 2019/2/23

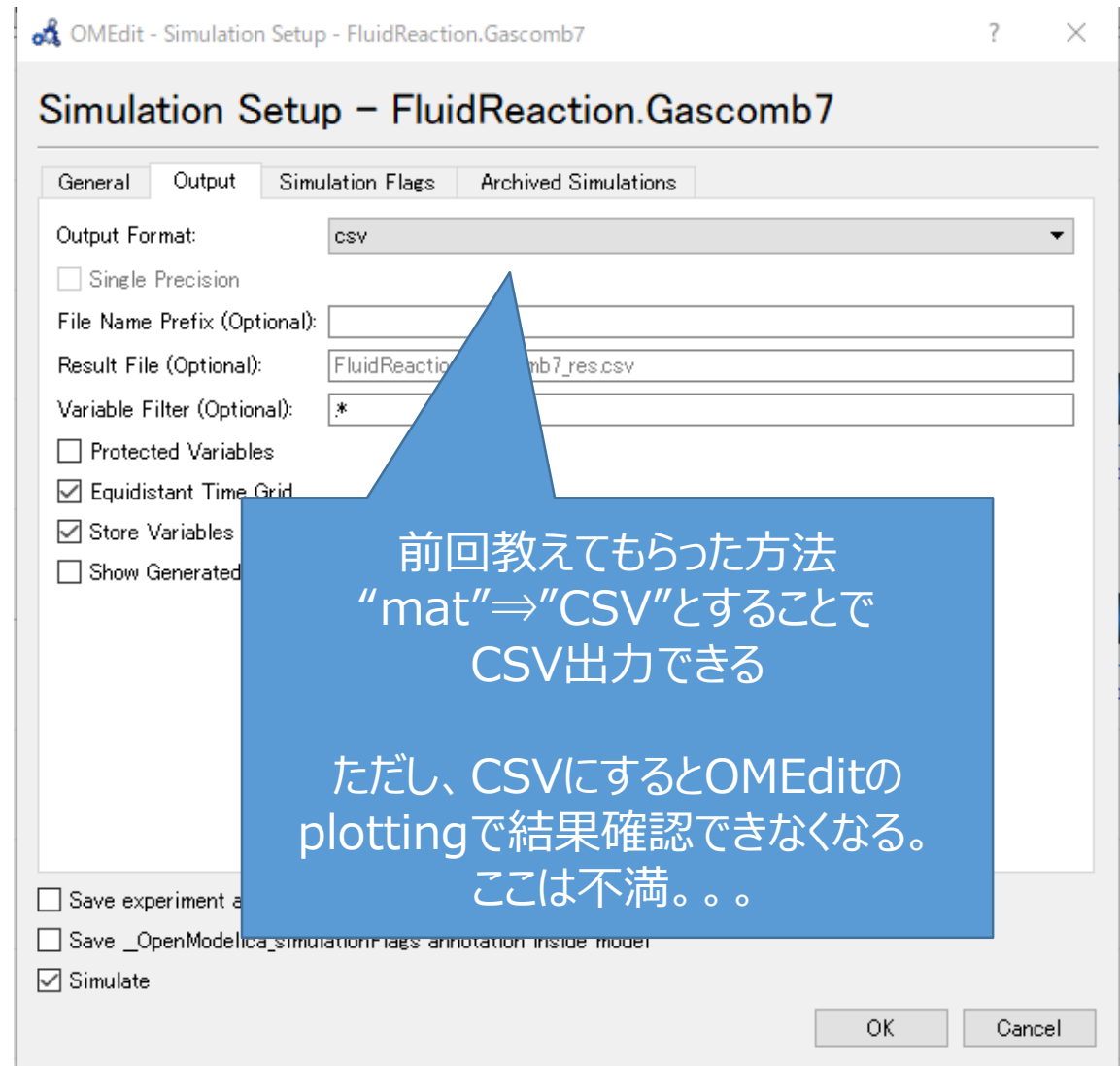
[k\\_takahashi](#)

# ご教授ありがとうございます：結果の出力（１）

## メニューバーの“Tools”-“options”

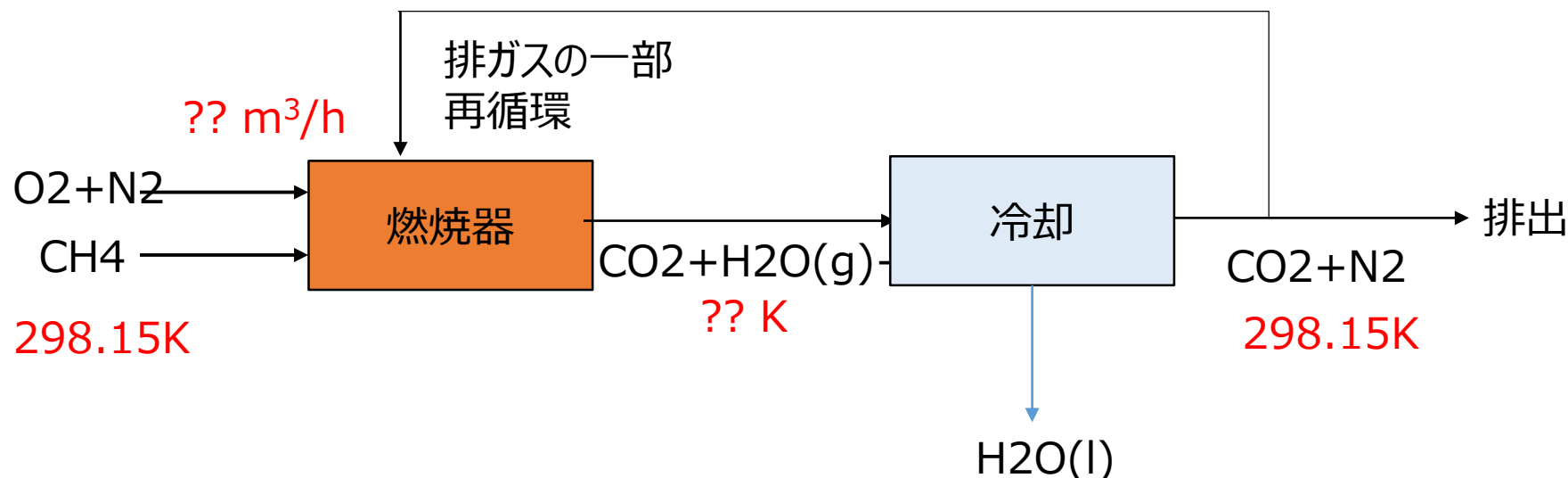


# ご教授ありがとうございます：結果の出力（２）



# やりたいこと：熱物質収支計算をOpenModelicaで

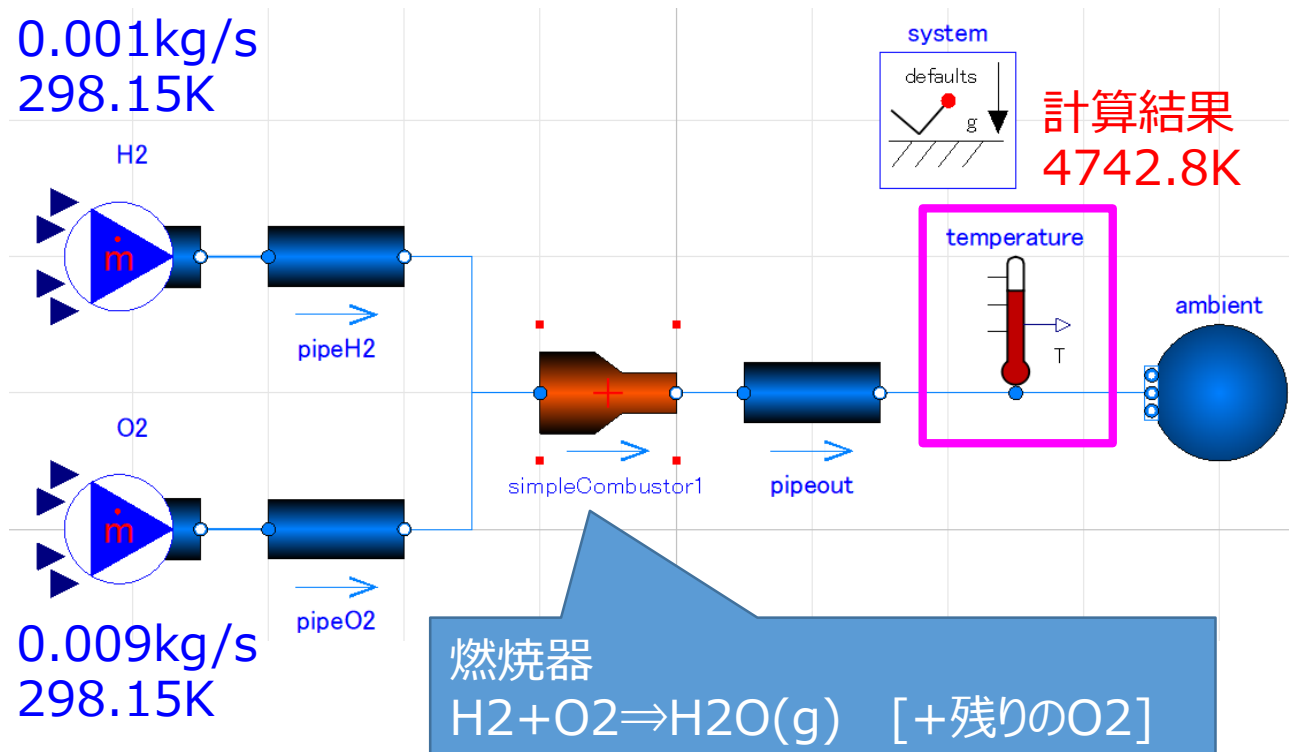
目標：打倒As●en！



## 【本日のお題目】

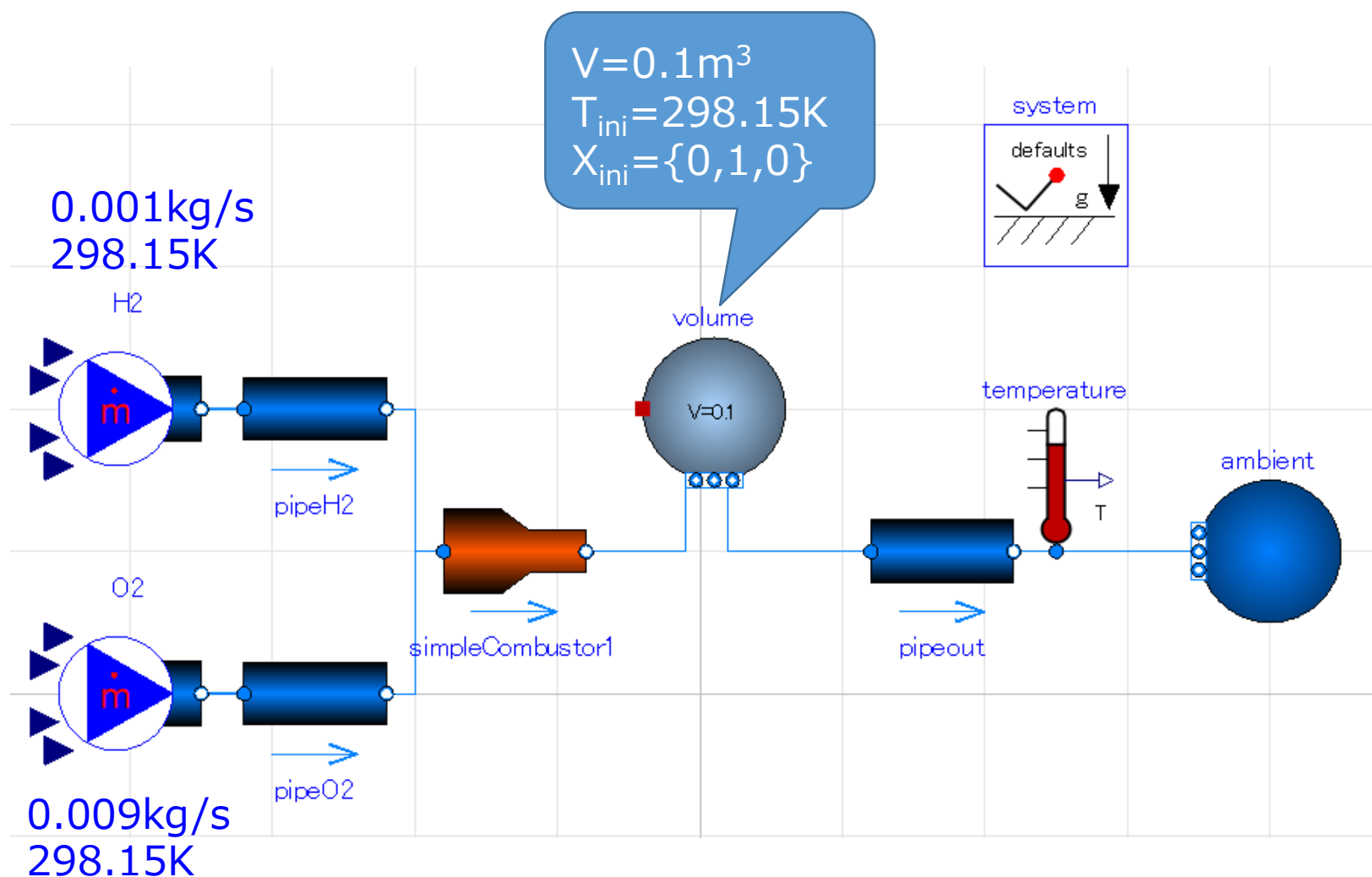
- ①収束性向上のため、体積要素を入れてみた
- ②組み合わせコンポーネントを作ってみた
- ③再循環作ってみた(流路途中の流量・温度制御器)
- ④わからないことリスト

# 前回報告：水素+酸素燃烧器



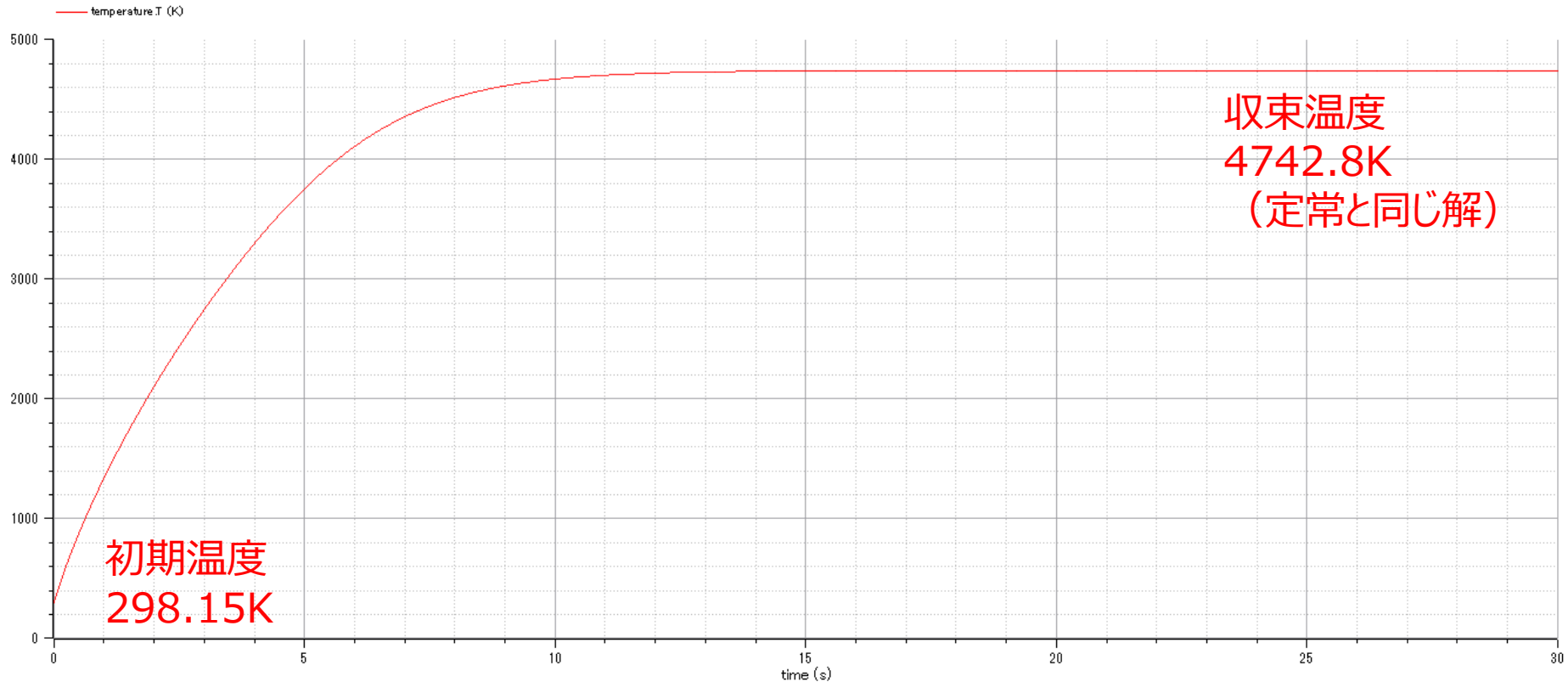
- この問題は解ける
- もう少し複雑にすると、計算できなくなる。(Singularityエラー)
- 体積要素**を加えて、あえて非定常化したほうが解きやすくなるのでは？とのアドバイスあり

# 体積要素を入れてみた：closedVolumeを付けてみる



中身ブラックボックスだが、とりあえずつけてみるか。  
(↑研究者としてはやってはいけない手順)

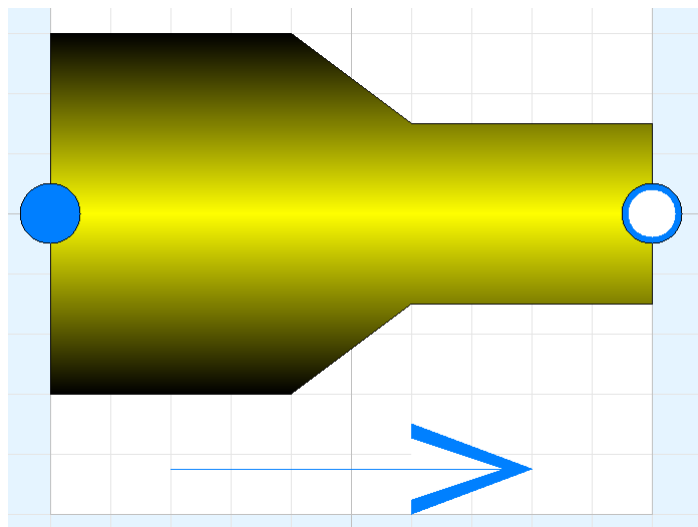
# 体積要素を入れてみた：まずは計算成功



(注) 上記グラフはプリントスクリーンではなく  
きちんとpngファイルを出力して作りました。

ClosedVolumeはおおよそ想定通りに動いた。  
中身はきちんと見ていないが、まあこれでよからう。

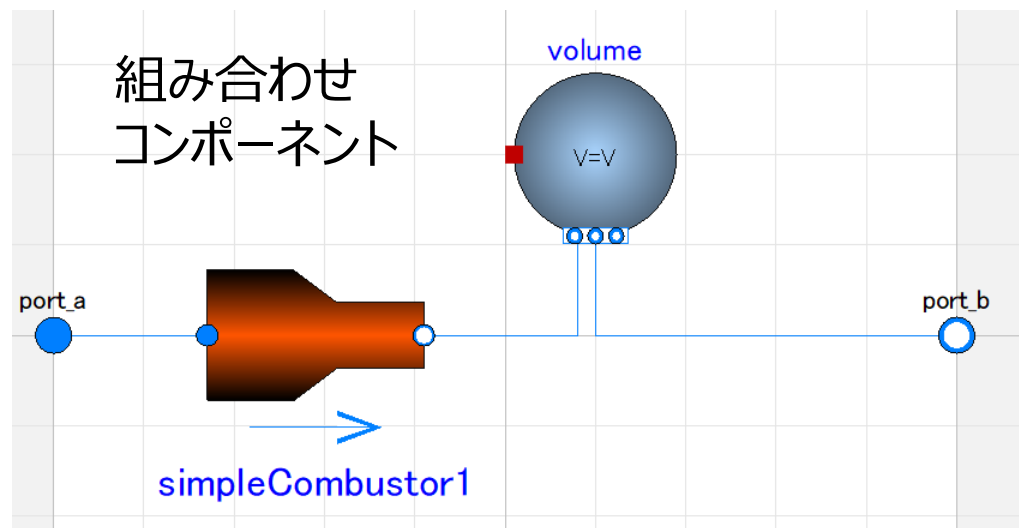
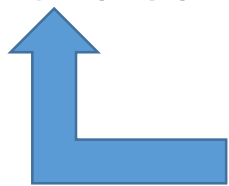
# 組み合わせコンポーネントを作ってみた



毎回燃焼器とVolumeをつないで作るのは**美しい**  
体積要素を含む燃焼器コンポーネントをつくるべき

組み合わせコンポーネントによる  
自作コンポーネント作成にトライ

simpleVolCombustor0





# 組み合わせコンポーネント：ソース中身

Fluidのコンポーネントを作る際は、  
PartialTwoPortから派生させる。

```
model simpleVolCombustor0 "simple combustor with fixed volume"
```

```
  extends Modelica.Fluid.Interfaces.PartialTwoPort(allowFlowReversal = false);
```

```
  FluidReaction.simpleCombustor simpleCombustor1(redeclare package  
  Medium = Medium) annotation(···);
```

```
  Modelica.Fluid.Vessels.ClosedVolume volume(redeclare package Medium =  
  Medium, nPorts = 2, use_portsData = false) annotation(···);
```

```
equation
```

```
  connect(simpleCombustor1.port_b, volume.ports[1]) annotation(···);
```

```
  connect(volume.ports[2], port_b) annotation(···);
```

```
  connect(port_a, simpleCombustor1.port_a) annotation(···);
```

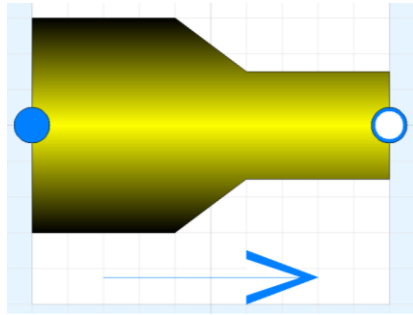
```
  annotation(···);
```

```
end simpleVolCombustor0;
```

内部コンポーネント  
(2個)

一応動くので成功だが。。。

# 組み合わせコンポーネント：困った



simpleVolCombustor0の  
GUIパネル



入力パラメータ(体積 $V$ 、初期温度 $T_{\text{start}}$ 等)がGUIから入れられなくなった。  
ソースに直書きすることで使えることは使えるが、これはあまりにも**美しくない**。

# 組み合わせコンポーネント：Parameterの再設定

```
model simpleVolCombustor "simple combustor with fixed volume"
  extends Modelica.Fluid.Interfaces.PartialTwoPort(allowFlowReversal = true)
  FluidReaction.simpleCombustor simpleCombustor1(redeclare package Medium = Medium)
  annotation(...);
  Modelica.Fluid.Vessels.ClosedVolume volume(redeclare package Medium = Medium, V = V,
  p_start=p_start, T_start = T_start, X_start = X_start, nPorts = 2, use_portsData = false)
  annotation(...);

  parameter Modelica.SIunits.Volume V "Volume";
  parameter Medium.AbsolutePressure p_start = system.p_start "Start value of pressure"
  annotation(...);
  parameter Medium.Temperature T_start=system.T_start "Start value of temperature"
  annotation(...);
  parameter Medium.MassFraction X_start[Medium.nX]=Medium.X_default "Start value of
  mass fractions m_i/m" annotation (...);

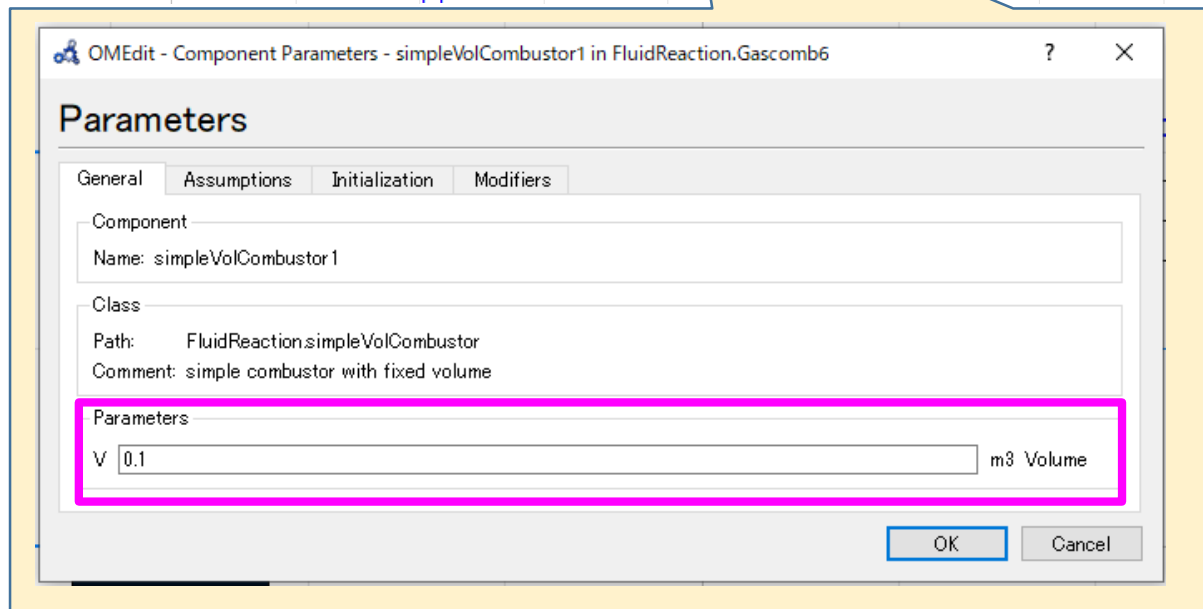
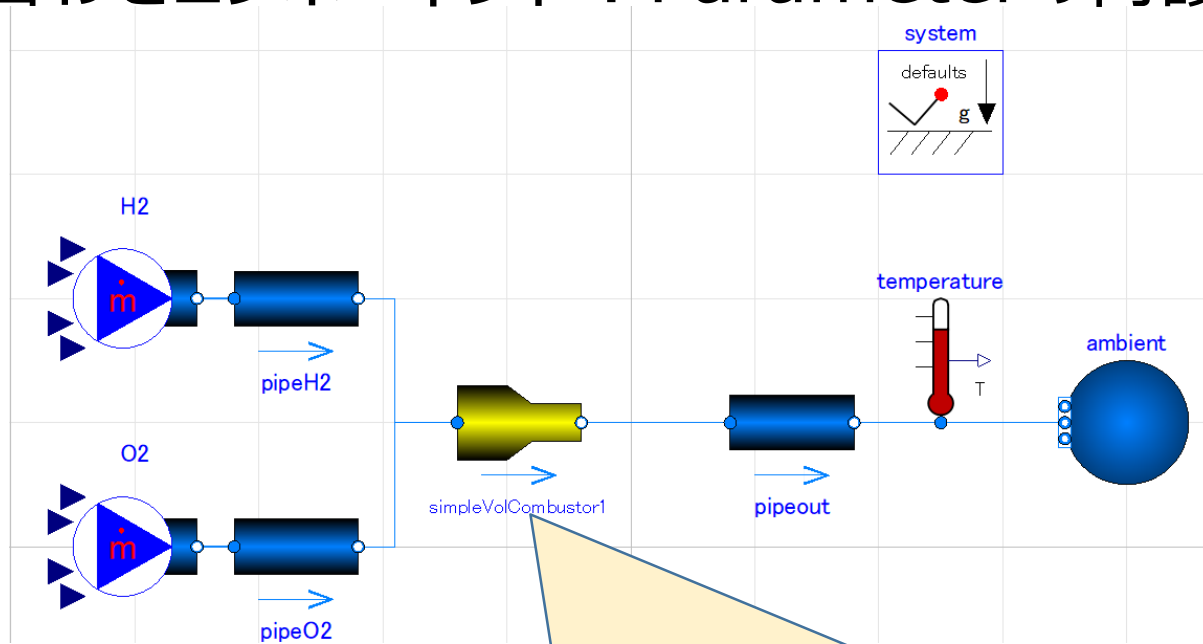
equation
  connect(simpleCombustor1.port_b, volume.ports[1]) annotation(...);
  connect(volume.ports[2], port_b) annotation(...);
  connect(port_a, simpleCombustor1.port_a) annotation(...);
  annotation(...);
end simpleVolCombustor;
```

左辺：ClosedVolume内のV

右辺：simpleVolCombustor内のV

内部コンポーネントClosedVolumeのparameter(V,T\_start等)をsimpleVolCombustorのGUI上から設定できるよう、parameterを再定義する。  
(ちょっと面倒。仕方ない。)

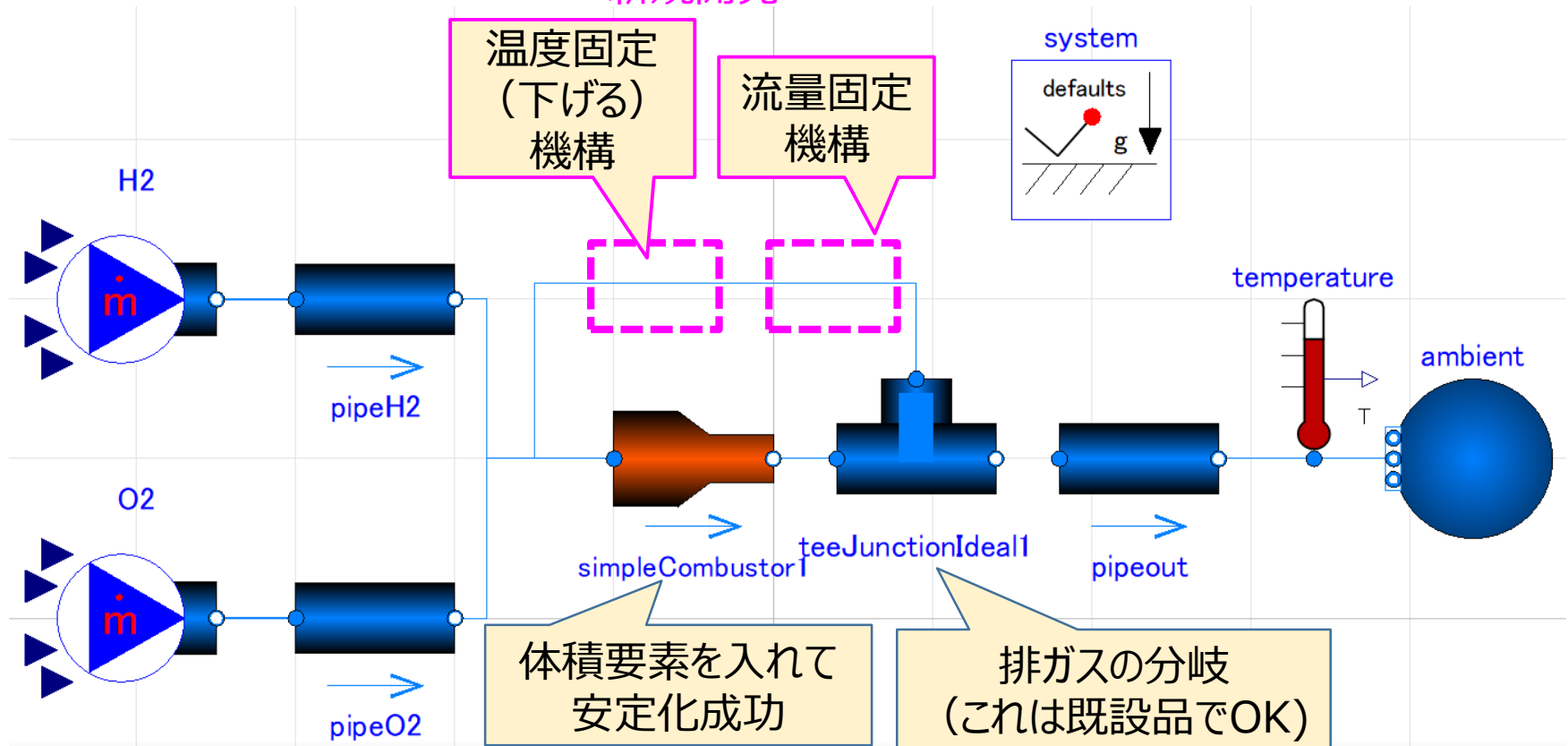
# 組み合わせコンポーネント：Parameterの再設定



体積をGUIから入力  
できるようになった  
美しい！！

# 排ガス再循環：前回報告

新規開発



長い間苦戦していた排ガス再循環に再トライ

- ・ガス流路分岐
- ・流量固定
- ・温度固定 (ガス温度を下げる)

の3つの気候が必要

# 排ガス再循環：流量固定コンポーネント

```
model mfCtl "Mass Flow Controller"  
  extends Modelica.Fluid.Interfaces.PartialTwoPort(allowFlowReversal = false);  
  parameter Real fix_m_flow(unit="kg/s") = 0.0 "Fixed mass flow rate";  
Equation  
  //port_a.p = port_b.p;    //流量制御に伴い両端で圧力同一にならないので、この項は削除  
  port_a.m_flow + port_b.m_flow = 0;  
  port_a.m_flow = fix_m_flow;  
  port_b.Xi_outflow[:] = inStream(port_a.Xi_outflow[:]);  
  port_b.h_outflow = inStream(port_a.h_outflow);  
  port_b.C_outflow = inStream(port_a.C_outflow);  
  port_a.Xi_outflow[:] = inStream(port_b.Xi_outflow[:]);  
  port_a.h_outflow = inStream(port_b.h_outflow);  
  port_a.C_outflow = inStream(port_b.C_outflow);  
annotation(...);  
end mfCtl;
```

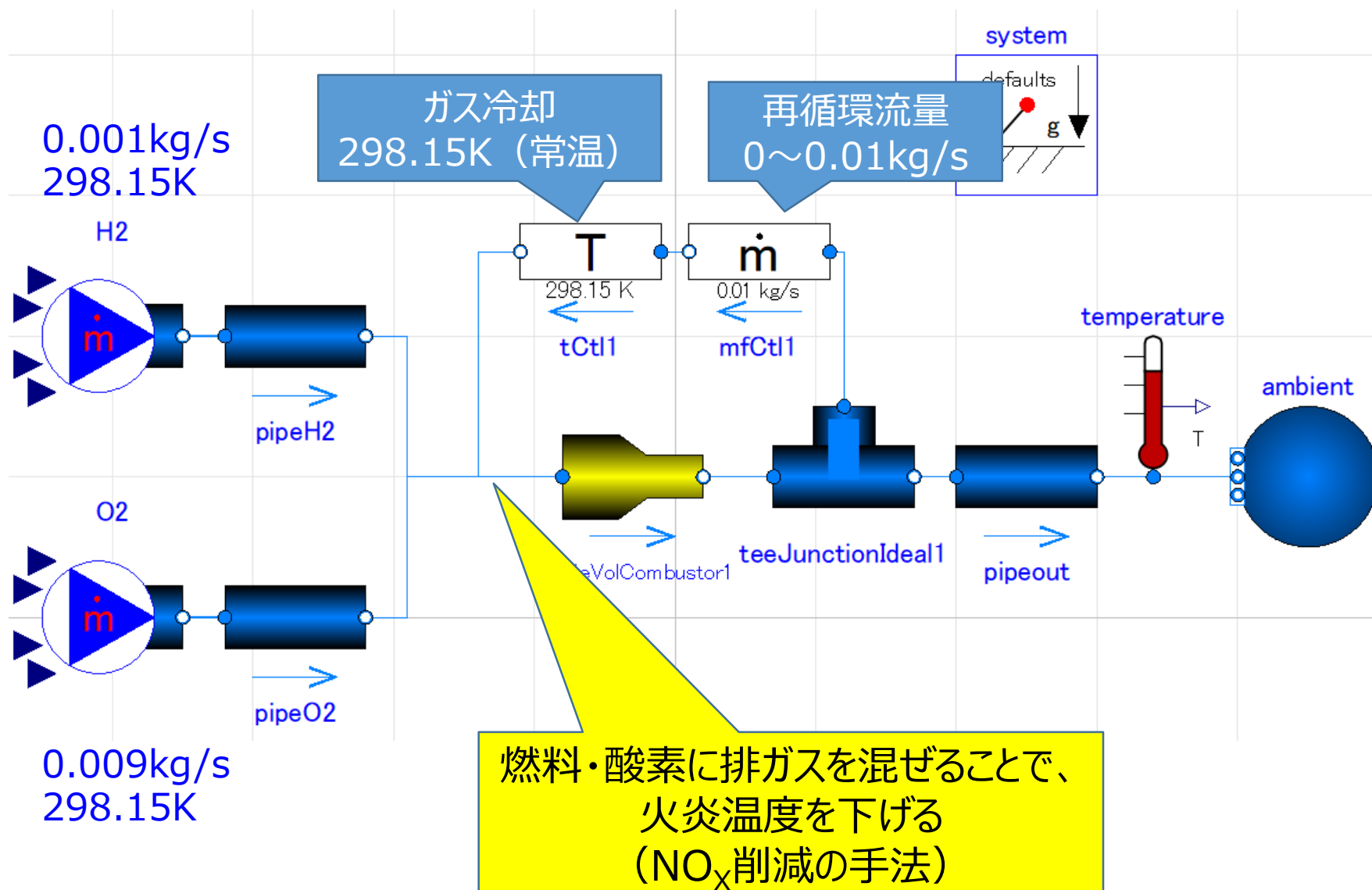
流量を固定するコンポーネント作成  
単に流量を指定値にするだけ。簡単。

# 排ガス再循環：温度固定コンポーネント

```
model tCtl "Temperature controller"
  extends Modelica.Fluid.Interfaces.PartialTwoPort(allowFlowReversal = false);
  parameter Real fix_T(unit="K") = 298.15 "Fixed temperature";
  Modelica.SIunits.SpecificEnthalpy dh "Heat supply(+) / loss(-)";
Equation
  port_a.p = port_b.p;
  port_a.m_flow + port_b.m_flow = 0;
  port_b.Xi_outflow[:] = inStream(port_a.Xi_outflow[:]);
  //port_b.h_outflow = inStream(port_a.h_outflow);
  port_b.h_outflow = Medium.specificEnthalpy(Medium.setState_pTX(p = port_a.p,
T = fix_T, X = inStream(port_a.Xi_outflow[:])));
  port_b.C_outflow = inStream(port_a.C_outflow);
  port_a.Xi_outflow[:] = inStream(port_b.Xi_outflow[:]);
  port_a.h_outflow = inStream(port_b.h_outflow);
  port_a.C_outflow = inStream(port_b.C_outflow);
  dh = port_b.h_outflow - port_a.h_outflow;
annotation(...);
end tCtl;
```

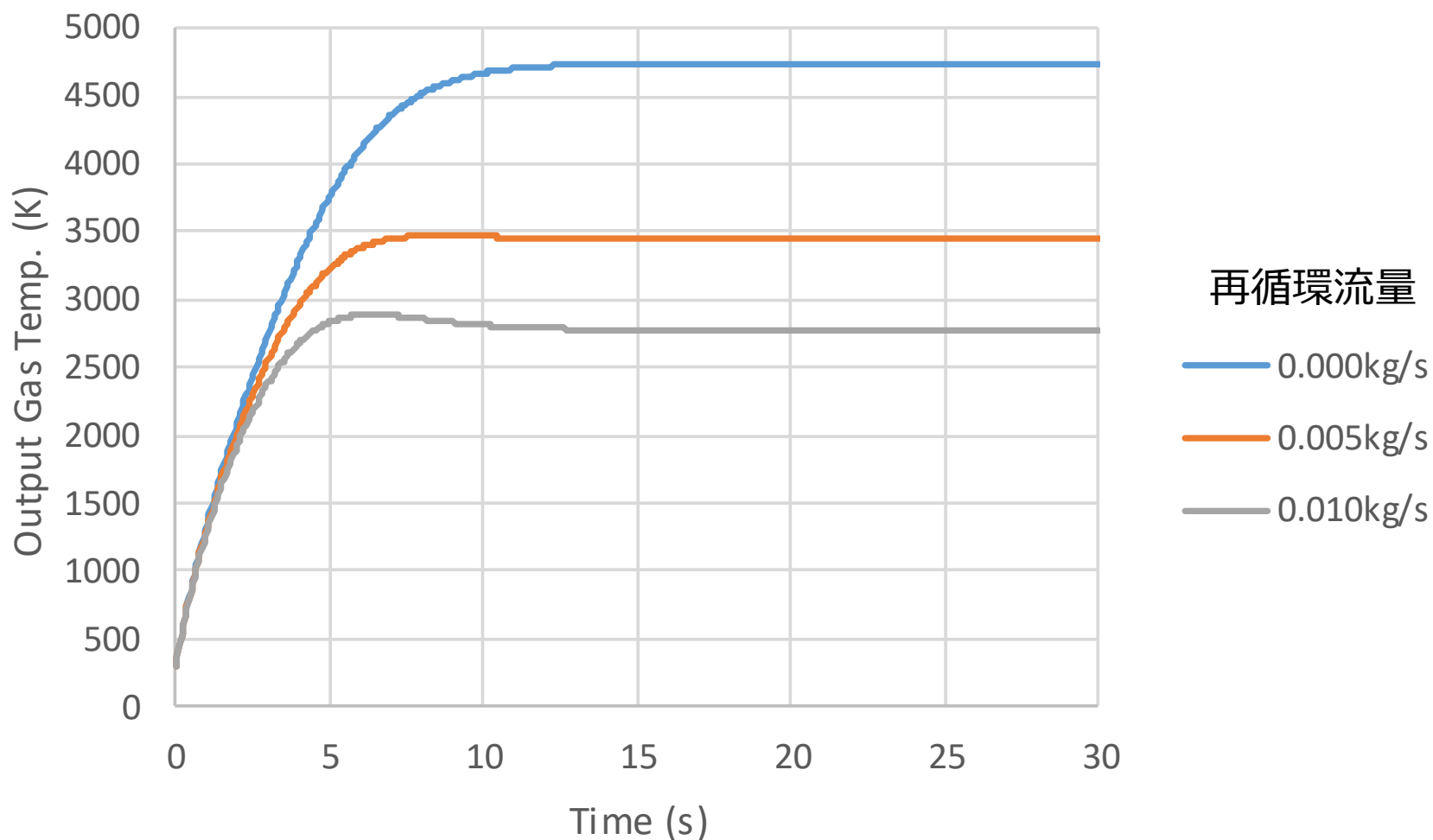
指定した温度に変更するコンポーネント作成。  
温度⇒エンタルピーの変換が必要なので、ちょっと面倒

# 排ガス再循環：モデル構成図





# 排ガス再循環：計算結果



排ガス再循環による火炎温度制御がやっと再現できるようになった。  
長き戦いであった。。。

# 今後の開発プラン（願望）

- 平衡計算ができるコンポーネント作成
  - かなり収束が難しい形態の数式なので、そのままequationで解かせるのは難しいかもしれない。

## • エクセルギーの解析機能追加

実際に使えるエネルギー < 理論エネルギー（エンタルピー）

実際に使えるエネルギーの理論上限  
=エクセルギー（Exergy）

$$E = H - T_0 S$$

↑

エンタルピー

↑


エントロピー

両方ともMediaの機能で計算できる  
なんとか作れそう

- その他、痒い所に手が届く微小改造
  - Massflow⇒VolFlow（kgではなくm3がいい）

# 分からないことリスト(1)

- Parameter , Constant, Input、無冠の違い（以下、高橋の独断解釈）
  - Parameter : GUI上でダブルクリックして設定値を入力できる



The screenshot shows the 'General' tab of a Simulink parameter editor for a component named 'volume'. The 'Class' is 'Modelica.Fluid.Vessels.ClosedVolume'. The 'Parameters' section shows a parameter 'V' with a value field and the unit 'm3 Volume'.

General	Assumptions	Initialization	Advanced	Modifiers
<b>Component</b>				
Name: volume				
<b>Class</b>				
Path: Modelica.Fluid.Vessels.ClosedVolume				
Comment: Volume of fixed size, closed to the ambient, with inlet/outlet ports				
<b>Parameters</b>				
V <input type="text"/> m3 Volume				

- Constant : 未来永劫、決して変更することがない神樣的な数値  
円周率 $\pi$ とか。
- Input : わからない。無冠と何が違うのか？

## 分からないことリスト(2)

simpleVolCombustorのGUI上で圧力p\_startの入力部に単位が出ない。しかしp\_start(unit="Pa")とつけると、エラー。なぜ？

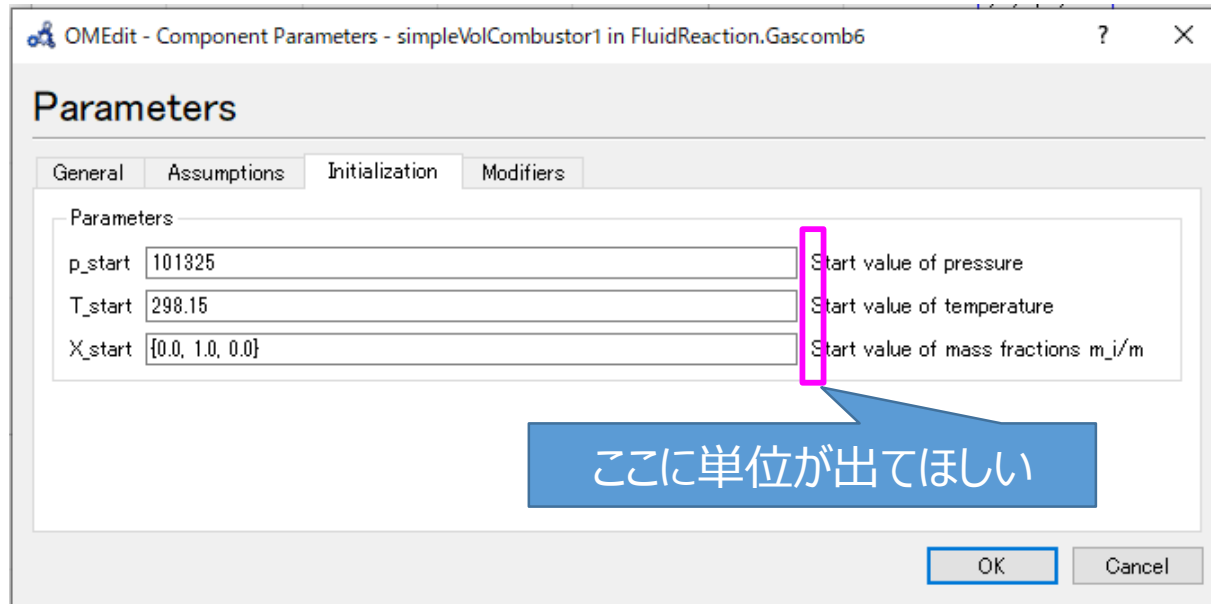
```
parameter Medium.AbsolutePressure p_start(unit="Pa") =  
system.p_start "Start value of pressure"
```

**[2] 00:38:08 Translation Error**

[FluidReaction: 481:47-481:56]:

Trying to override final element unit with modifier ' = "Pa"'.

すでに単位はfinalで定義済みということか？



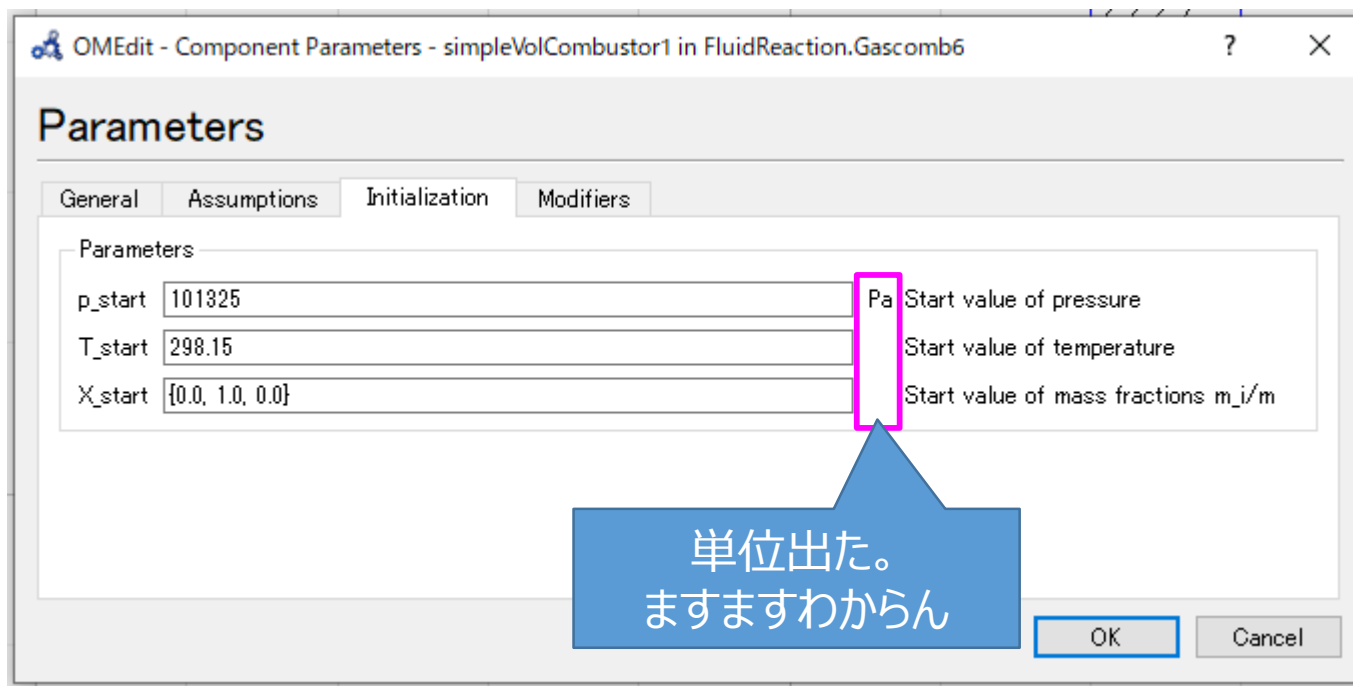
ここに単位が出てほしい

## 分からないことリスト(2)

```
model simpleVolCombustor "simple combustor with fixed volume"  
  extends Modelica.Fluid.Interfaces.PartialTwoPort(allowFlowReversal = false);  
  FluidReaction.simpleCombustor simpleCombustor1(redeclare package Medium =  
Medium) annotation(...);  
  Modelica.Fluid.Vessels.ClosedVolume volume(redeclare package Medium =  
Medium, V = V, p_start=p_start, T_start = T_start, X_start = X_start, nPorts =  
2, use_portsData = false) annotation(...);  
  parameter Modelica.SIunits.Volume V "Volume";  
  // parameter Medium.AbsolutePressure p_start = system.p_start "Start  
value of pressure" annotation(...);  
  parameter Real p_start(unit="Pa") = system.p_start "Start value of  
pressure" annotation(...);  
  parameter Medium.Temperature T_start=system.T_start "Start value of  
temperature" annotation(...);  
  parameter Medium.MassFraction X_start[Medium.nX]=Medium.X_default "Start  
value of mass fractions m_i/m" annotation (...);  
equation  
  connect(simpleCombustor1.port_b, volume.ports[1]) annotation(...);  
  connect(volume.ports[2], port_b) annotation(...);  
  connect(port_a, simpleCombustor1.port_a) annotation(...);  
  annotation(...);  
end simpleVolCombustor;
```

試しにp\_startをModelica.Medium内の型を使わず、自分でReal定義してみる

## 分からないことリスト(2)



入力変数を“Parameter Real ○○(unit=△△)”と定義して回避できたが。。。Modelica標準で定義されているtypeを使うと単位が出ない問題は解決できず。腑に落ちない。