**Chemical 2.0**

**(Free open-source Modelica library)**

Marek Mateják1,2

1Institute for Clinical and Experimental Medicine, Czech Republic

2First Faculty of Medicine, Charles University, Czech Republic

marek@matfyz.cz

# **Abstract**

Chemical2 (<https://github.com/MarekMatejak/Chemical>) provides: simpler chemical substance and process definitions; propagation of definitions and chemical solution state through connected components; working with unknown substances; typical chemical kinetics; working with be variable (as in previous versions) or constant chemical solution.   
The possibilities and performance of chemical pathways modeling are increased using new type of connectors based on electro-chemical potential inertia. Chemical processes can be directly connected without definition substance between them.

Keywords: Chemical 2.0, Modelica library, physical chemistry, thermodynamics equilibria, electrochemical potential, electrochemical cell, internal energy, semipermeable membrane, chemical kinetics, chemical pathways

# **Introduction**

TODO

# **Definition**

The previous version of Chemical library has various definitions of substances. In version 2.0 are all these structures for all type of substances and even for all type of chemical processes unified into just one type of operator record called Chemical.Interfaces.Definition. This record can define each substance and process used in previous version. In addition, user can set their values independently on internal representation using suitable constructors or function processData. Function processData create process definition from dissociation constant (molar-based) and consumed heat of the process (free molar enthalpy change). In the other hand user can easily evaluate new definition of new chemical substance or new chemical process using intuitive ‘\*’, ‘+’ and ‘-‘ operators.

In example, a definition of aqueous O2 can be set from tabulated Henry’s coefficient (0.0013, 1500K) or definition of H2O formation reaction from H2 and O2 is just an algebra between reaction products and substrates (**Listing 1**).

**Listing 1.** Example of definitions

import Chemical.Interfaces.Definition;

import Chemical.Substances.Gas;

import Chemical.Substances.Liquid;

import Chemical.Interfaces.processData;  
constant Real R = Modelica.Constants.R;

constant Definition **O2\_aq** =

Gas.O2 + processData(

K = 0.0013,

dH = -1500\*R);

constant Definition **H2O\_formation** =

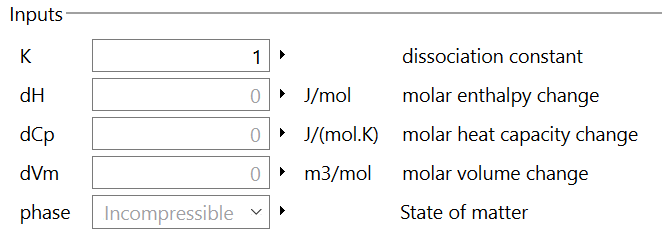
Gas.H2O - (Gas.H2 + 0.5\*Gas.O2);

constant Definition **Hemoglobin** =

Liquid.Unknown;

The default definition of Unknown substance is molar mass 1kg/mol and molar heat capacity 1 J/(mol.K) with zero free formation Gibbs energy and zero free formation enthalpy. These values are used if user does not set the substance. Because free formation energies are always defined in relative way unknown substances can be used such as any base substance with unknown formation process. If they play role in solution properties at least the adequate property should be set. E.g. molar mass and molar volume must be set if it take role in solution composition; molar heat capacity must be set if it take role in heat accumulation in chemical solution.

**Figure 1**, processData() inputs

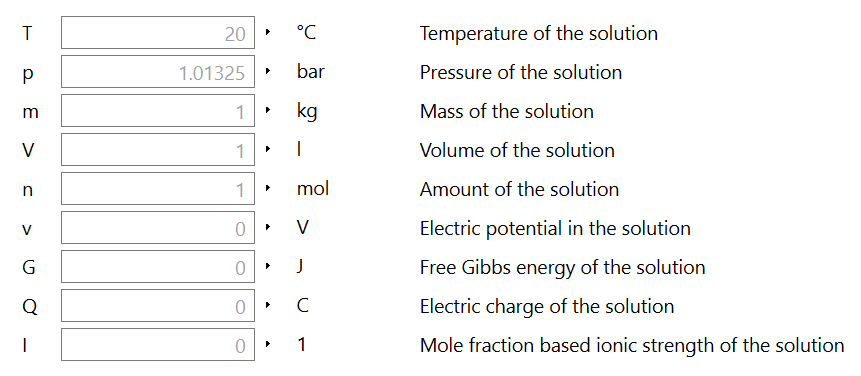


# **Chemical solution**

There are two options how to define chemical solution. First is using Chemical.Solution component and connecting all substances through its solution connector. This was the only option in previous version and it works in the same way also in this version. If a chemical solution is defined this way hen all its properties are dynamically calculated during simulation.

However not always there is a need to change solution properties. If solution properties such as temperature, pressure, total mass, total volume, total amount of particles and electric potential can be assumed as constant during simulation then a solution state can be set as parameter. If user did not set a solution property (**Listing 2**) then the default value (**Figure 1**.) will be used.

**Figure 2**. Default chemical solution properties



**Listing 2.** Example of solution

import Chemical.Interfaces.SolutionState;

import Chemical.Interfaces.Phase;

constant SolutionState SATP =

SolutionState(phase = Phase.Gas,

T = 298.15);

SolutionState heatingSolution =

SolutionState(phase = Phase.Gas,

T = 273.15+time);

# **Properties of chemical process**

There can be observed different properties from process definition in dependence on chemical solution. User just need to connect substance definition and solution to model Chemical.Interfaces.ProcessProperties to calculate these properties.

**Listing 3.** Example of process properties

import Interfaces.ProcessProperties;

ProcessProperties O2\_dissolving\_props

(definition = processData(0.0013,-1500\*R),

solutionState = heatingSolution);

ProcessProperties H2O\_formation\_props

(definition = H2O\_formation,

solutionState = heatingSolution);

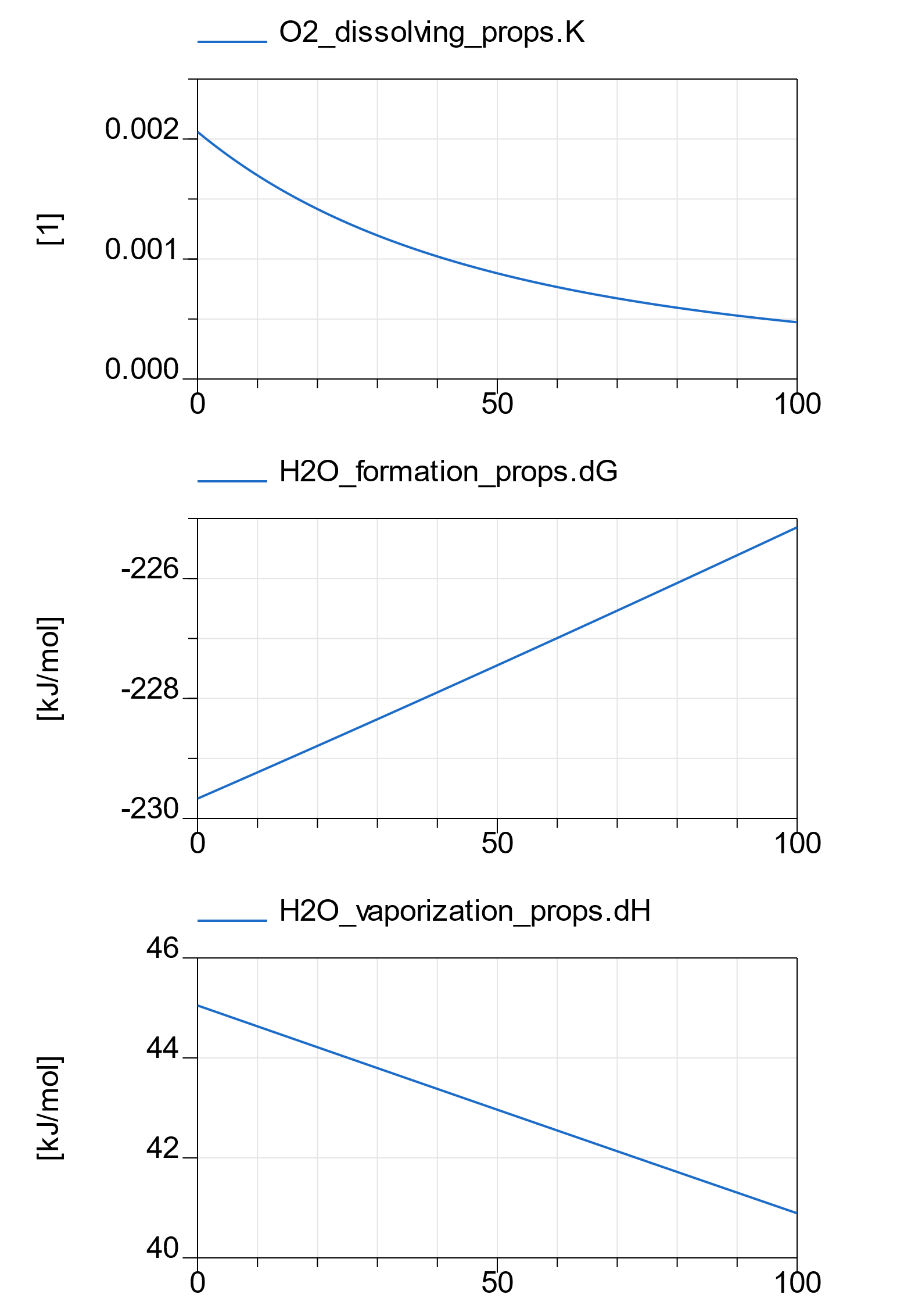
ProcessProperties H2O\_vaporization\_props

(definition = Gas.H2O - Liquid.H2O,

solutionState = heatingSolution);

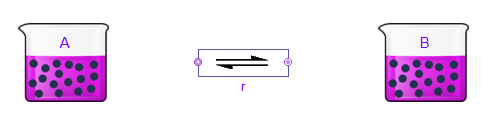
After run of the simulation during 100s the temperature of solution change from 0°C to 100°C and we can see the recalculated properties in dependence on temperature e.g. dissociation constants (Henry’s coefficient), free Gibbs energy change of reaction or consumed heat by process (free enthalpy change) – e.g. **Figure 2**.

**Figure 3**, Example of process properties

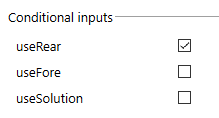


# **Model of a chemical reaction**

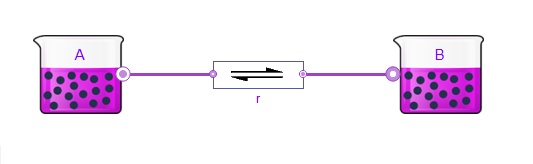
To create model of chemical reaction using Chemical 2.0 drag and drop substance component Substance from package Chemical.Boundaries as substance A and substance B. In the same way put into model Reaction component r from package Chemical.Processes.



In default setting the substances does not provides any connector, so it is necessary to setup which connectors will be used. This is done in parameter dialog just by click to the useFore of substance A and useRear of substance B.



Having all necessary connectors, it is possible to connect substances with chemical reaction r.

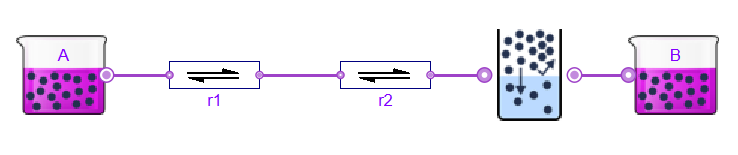


This model could be simulated, but it all remains constant because dissociation constant is one and both substances are Liquid.Unknown with amount of substance 1mmol by default. However, if we change the dissociation constant to 2 (r.K=2) then we can see dynamic behavior in constant default chemical solution. And after 5s of simulation time the reaction reaches almost the equilibrium where the amount of product B is two times higher as amount of substrate A. We can change reaction speed by setting forward reaction rate coefficient (r.k\_forward) lower than its default value. Backward reaction rate is not possible to set as reaction parameter because it is defined as K\*k\_forward. This reaction can have as many substrates and as many products as needed and its index is defined by in order of drown connection.

# **Data propagation**

The version 2.0 of the Chemical library is based on special type of connector firstly defined by ThermofluidStream library. These connectors propagate state of media without inertial part in flow direction and generates equality of inertial electro-chemical potential instead of direct equality of total electro-chemical potential. The equilibria and even the dynamic simulation reach almost the same values if inertia is fast enough. As proved by …… this type of connectors has physical background and have better performance because they broke non-linear sets of equations where typically solvers spend the most of time. Even more this design allows to connect more processes together without any accumulation (or just with a few) of substance. As a result, there is possible to define long chemical pathways as usual in physiology.

**Figure 4**, Example of possible pathway connection



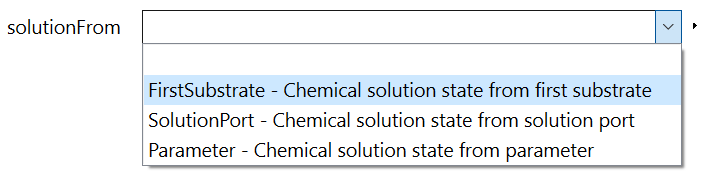
To have chemical pathway such as in ***Figure 3*** there is a need to have each substance definition between each chemical process. This is done using propagation pattern from forward connectors (called fore) into rearwards connectors (called rear). Please note that substances definition propagation is not dependent on flow direction. Each substance definition remains constant during simulation, so the direction of definitions propagation does not take any role during simulation and vice versa. In ***Figure 3*** can be defined only the substance A. Definition of a product of reaction r1 is evaluated from definition of this reaction and its substrate. So generally, each process can define first product from its other products, substrates and process definition. Even the substance B must not be explicitly defined because its definition can come from GasSolubility process.

In the same way can be propagated state of the chemical solution which is the same for each substance in the solution during each step of the simulation. In model of ***Figure 3*** it means that chemical solution is specified only in substance A and propagated trough connections to each other components as the same values.

If user prefer first product definition for a chemical process then is possible to set process option “firstProductFrom” into Substance selection. This enable the parameter firstProduct/product to set first product definition as process parameter. Because with knowledge of all products and all substrate definitions the process definition can be evaluated, so the parameter with process definition is disabled in this selection.

# 

To model the situation when process cross different solutions there is an option to break the solution propagation. In some processes such as Diffusion, Membrane or GasSolubility are not propagated solution state by default because their change the chemical solution between substrates and products. Also if chemical substance significantly change the chemical solution properties then it must be connected to Solution component using SolutionPort.



# **Examples**

TODO

# **Discussion**

TODO

# **Acknowledgements**

TODO

# **References**

TODO