Technische Universität Berlin

MOSAIC – A modeling and code generation tool

Gregor Tolksdorf, M.Sc. | Faculty of Process Sciences | CAPE-OPEN 2013 Annual Meeting

MOSAIC

Modular **MO**del **S**pecific**A**t**I**on on Do**C**umentation Level -

Application in a Web Based Modeling Environment.

www.mosaic-modeling.de

MOSAIC-Team

Scientific supervisor:

Prof. Dr.-Ing. G. Wozny ,

Head of Chair of Process Dynamics and Operation, TU Berlin

Current Developer:

Dipl.-Ing. S. Fillinger

Dipl.-Ing. R. Kraus

Dipl.-Ing. V.A. Merchan-Restrepo

M.Sc. G. Tolksdorf

Outline

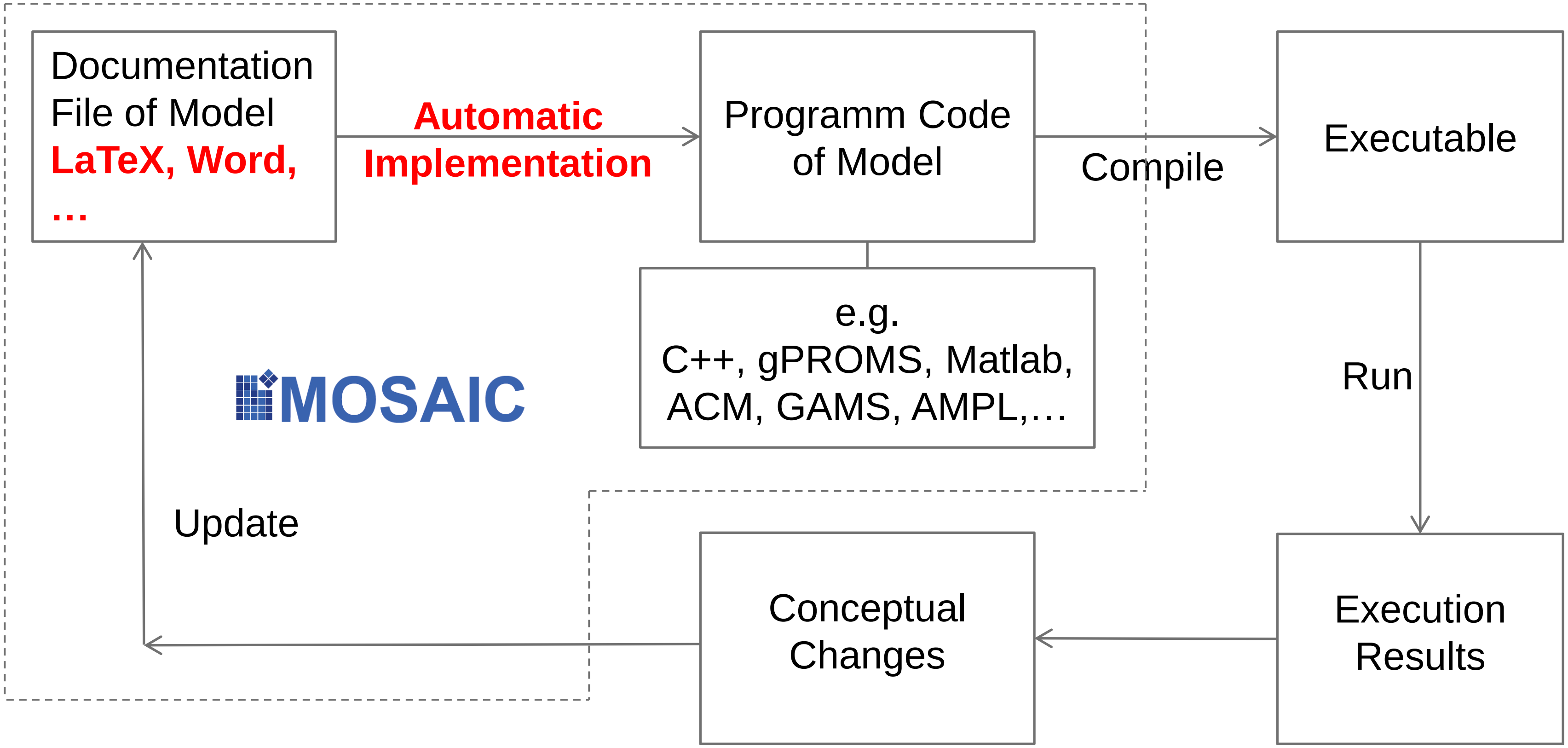
MOSAIC-Modeling

- Modular Concept
- Symbolic Notation
- Code Generation
- External Ports

CAPE-OPEN and MOSAIC

- Physical Properties
- Unit Operations

Modeling with MOSAIC



Documentation File of Model

LaTeX, Word, …

Automatic Implementation

Programm Code of Model

e.g.
C++, gPROMS, Matlab, ACM, GAMS, AMPL,…

Compile

Executable

Run

Execution Results

Conceptual Changes

Update

MOSAIC

Goals and Resulting Characteristics

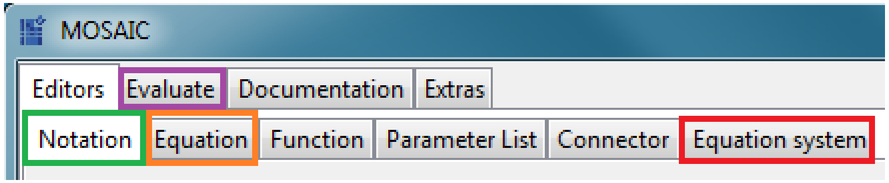
Main goals:

- Less errors
- Less effort
- More cooperative work
  - Improved reuse
  - Improved portability

Resulting characteristics:

- Highly **modular** modeling concept
- Define Platform Independent Models (PIM) in the documentation level using an **enhanced symbolic notation**
- Use of PIM and **code generation** to Platform Specific Models (PSM)
- Support web-cooperation
  - Store and share all model elements in a **web database**

Modular Modeling Concept – The Editors



MOSAIC editors and model elements:

Notation – What symbols/variables are allowed?

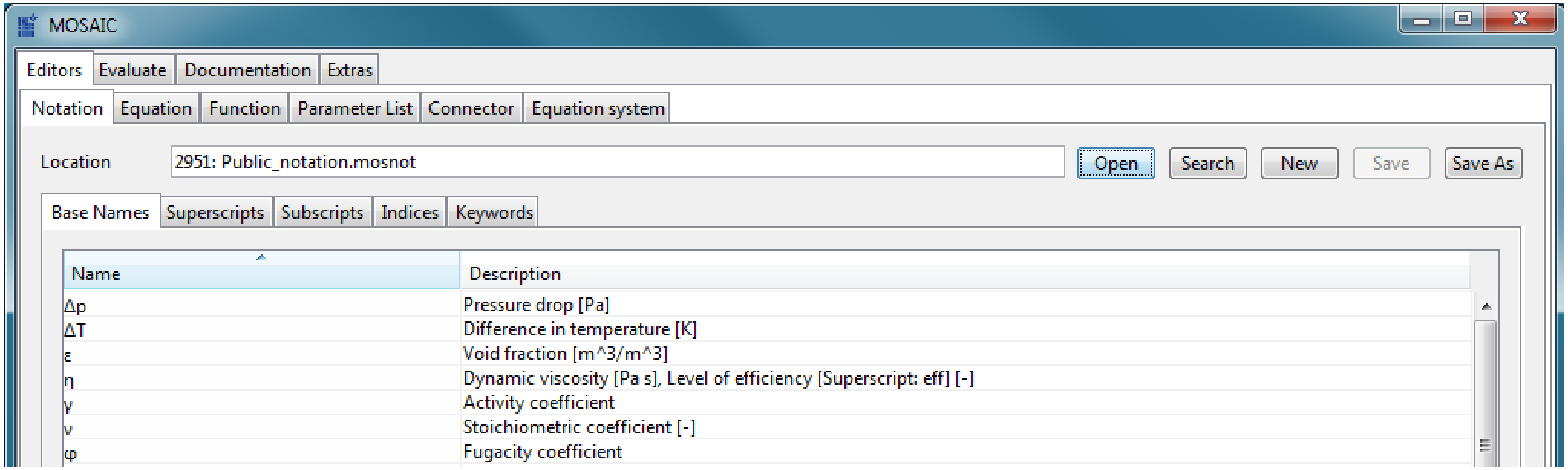
Equation – What equations will be used?

EquationSystem – How will the equations be combined? What functions will be used?

Evaluation – What are the design, state, and iteration values? How does the problem solving code look like?

Enhanced Symbolic Notation I - Variables

Notation editor:



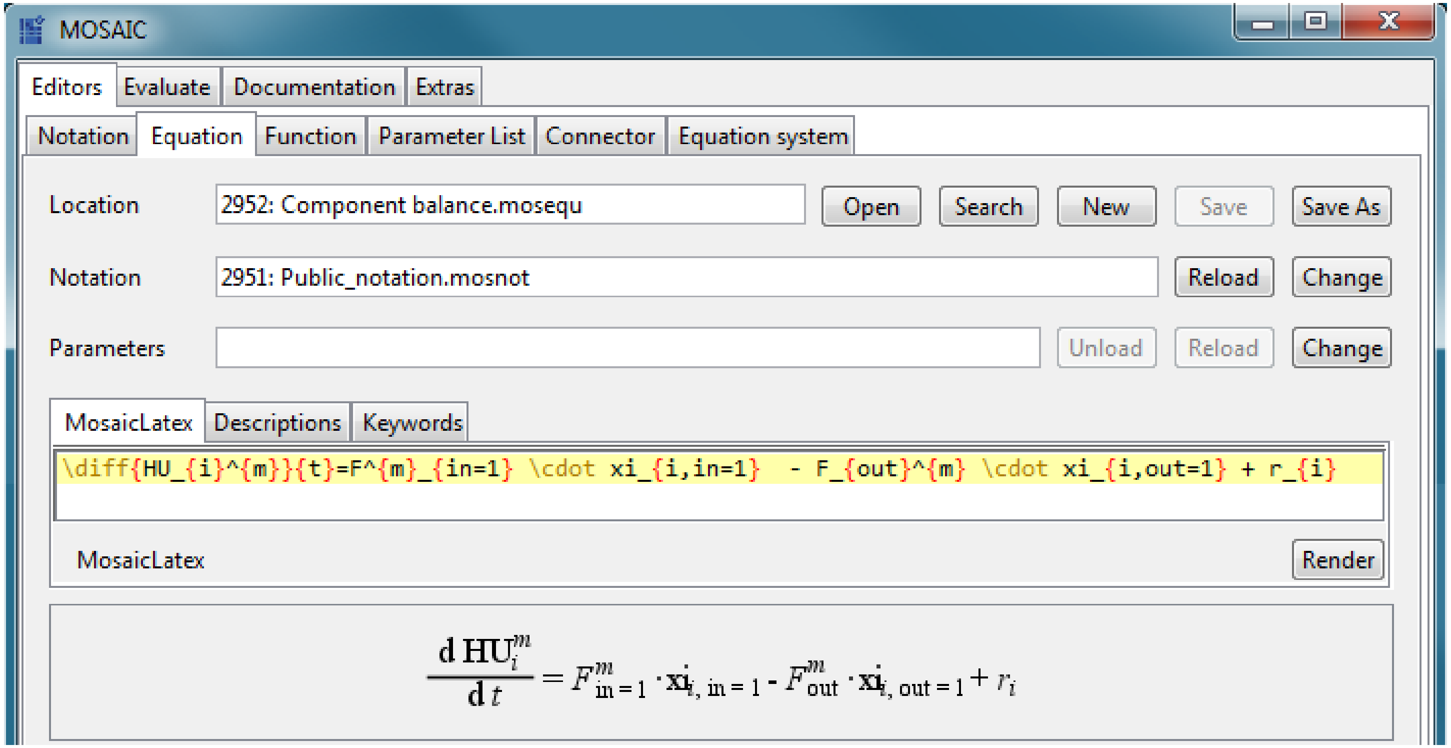
Location 2951: Public_notation.mosnot

Name	Description
Δp	Pressure drop [Pa]
ΔT	Difference in temperature [K]
ε	Void fraction [m^3/m^3]
η	Dynamic viscosity [Pa s], Level of efficiency [Superscript: eff] [-]
γ	Activity coefficient
ν	Stoichiometric coefficient [-]
φ	Fugacity coefficient

Example: $p^{LV,I}_{o,i=2,j=4}$

Enhanced Symbolic Notation II - Equations

Equation editor:



Location: 2952: Component balance.mosequ

Notation: 2951: Public_notation.mosnot

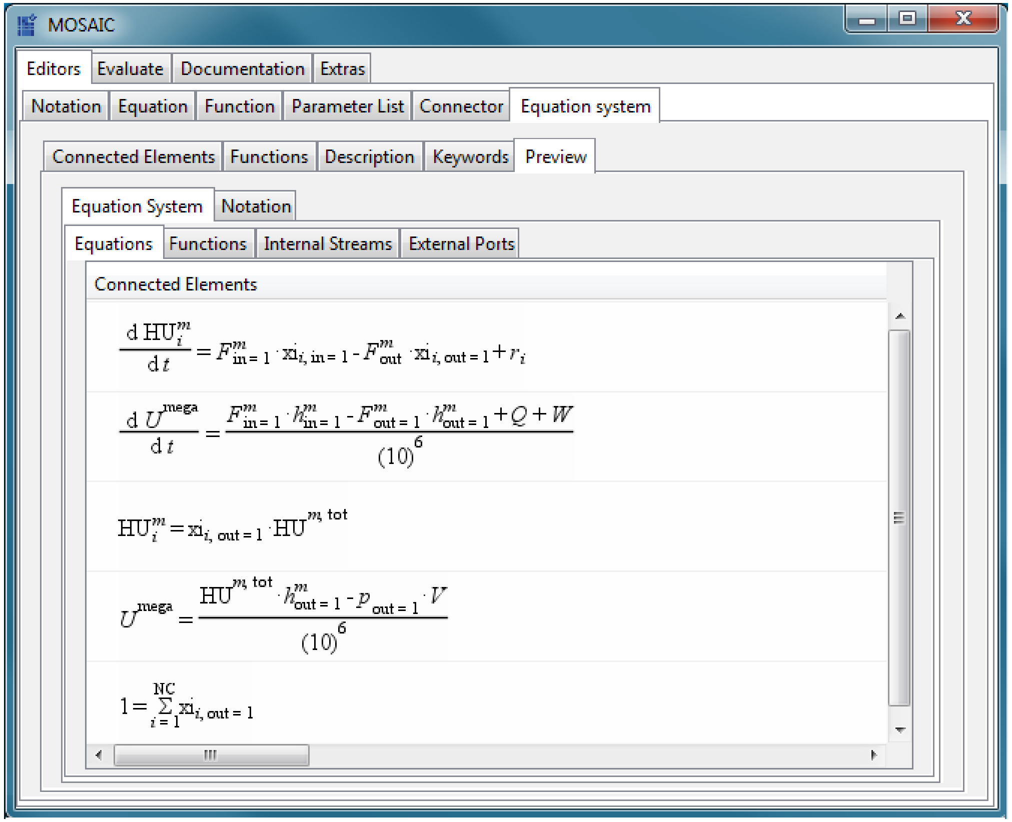
Parameters:

```
\diff{HU_{i}^{m}}{t}=F^{m}_{in=1} \cdot xi_{i,in=1}  - F_{out}^{m} \cdot xi_{i,out=1} + r_{i}
```

$$\frac{\mathrm{d}\,\mathrm{HU}_i^m}{\mathrm{d}\,t} = F^m_{\mathrm{in}=1} \cdot \mathrm{xi}_{i,\,\mathrm{in}=1} - F^m_{\mathrm{out}} \cdot \mathrm{xi}_{i,\,\mathrm{out}=1} + r_i$$

Enhanced Symbolic Notation III – Equation Systems

Equation system editor:



$$\frac{\mathrm{d}\,\mathrm{HU}_i^m}{\mathrm{d}\,t} = F^m_{\mathrm{in}=1} \cdot \mathrm{xi}_{i,\,\mathrm{in}=1} - F^m_{\mathrm{out}} \cdot \mathrm{xi}_{i,\,\mathrm{out}=1} + r_i$$

$$\frac{\mathrm{d}\,U^{\mathrm{mega}}}{\mathrm{d}\,t} = \frac{F^m_{\mathrm{in}=1} \cdot h^m_{\mathrm{in}=1} - F^m_{\mathrm{out}=1} \cdot h^m_{\mathrm{out}=1} + Q + W}{(10)^6}$$

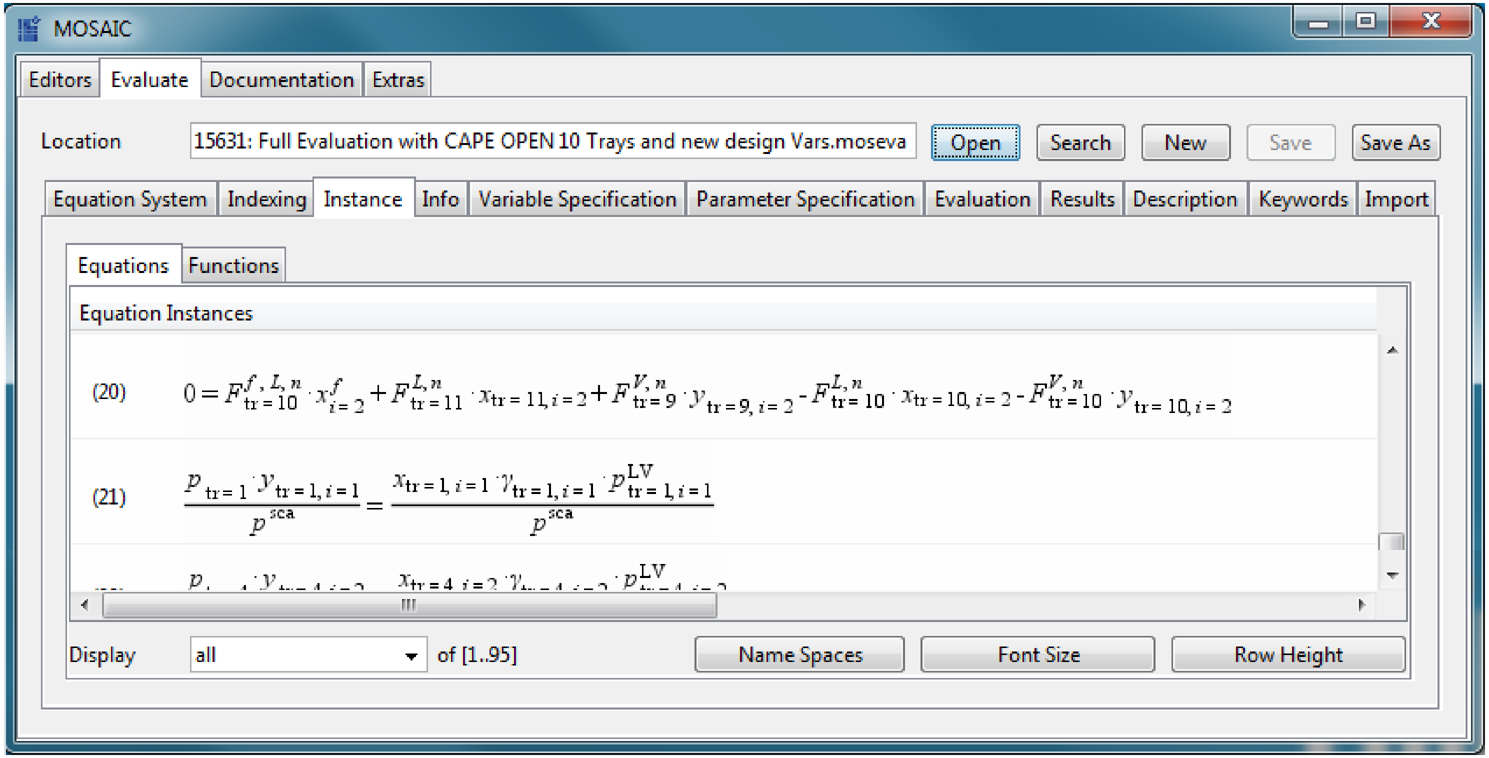
$$\mathrm{HU}_i^m = \mathrm{xi}_{i,\,\mathrm{out}=1} \cdot \mathrm{HU}^{m,\,\mathrm{tot}}$$

$$U^{\mathrm{mega}} = \frac{\mathrm{HU}^{m,\,\mathrm{tot}} \cdot h^m_{\mathrm{out}=1} - p_{\mathrm{out}=1} \cdot V}{(10)^6}$$

$$1 = \sum_{i=1}^{\mathrm{NC}} \mathrm{xi}_{i,\,\mathrm{out}=1}$$

Enhanced Symbolic Notation IV – Instantiated Equations

Evaluation editor:



Location: 15631: Full Evaluation with CAPE OPEN 10 Trays and new design Vars.moseva

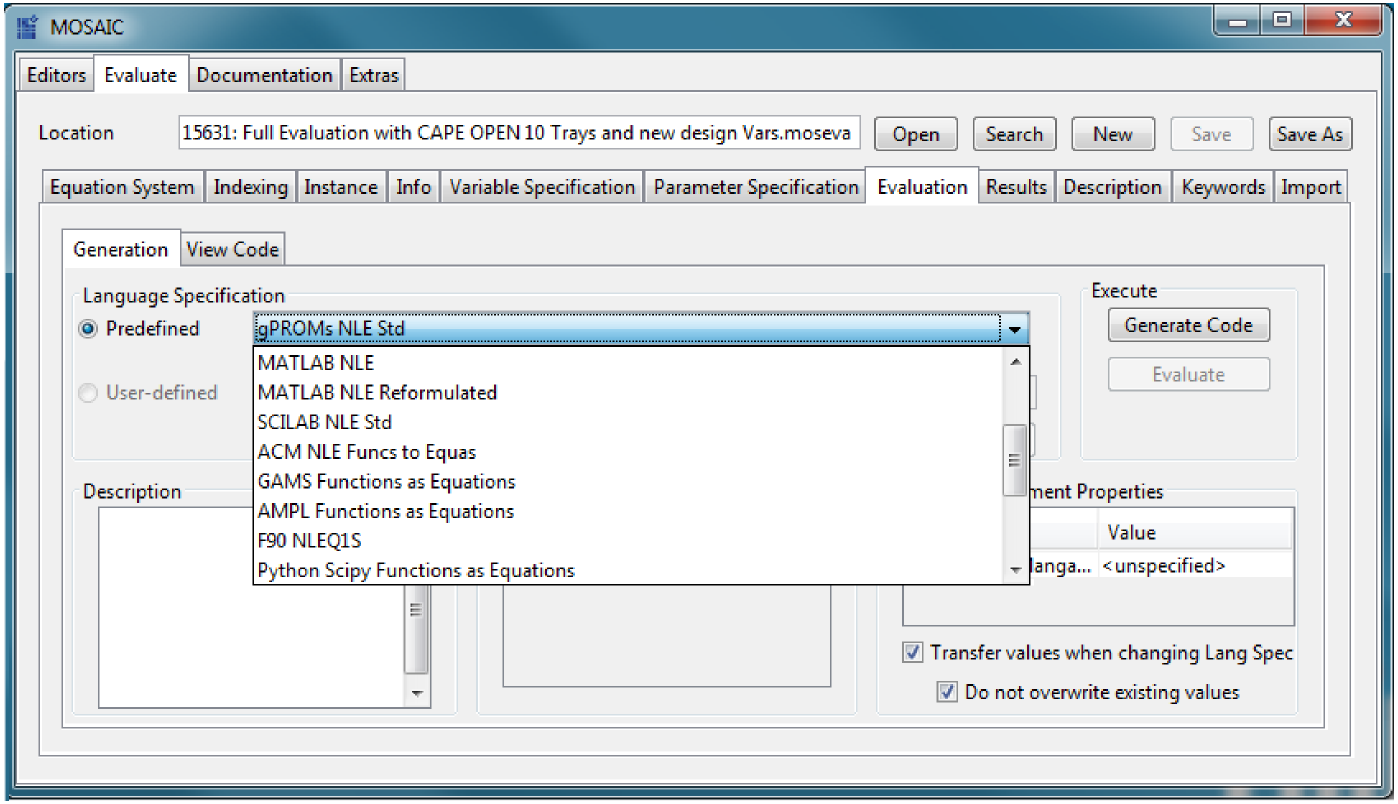
(20) $0=F^{f,L,n}_{tr=10}\cdot x^{f}_{i=2}+F^{L,n}_{tr=11}\cdot x_{tr=11,i=2}+F^{V,n}_{tr=9}\cdot y_{tr=9,i=2}-F^{L,n}_{tr=10}\cdot x_{tr=10,i=2}-F^{V,n}_{tr=10}\cdot y_{tr=10,i=2}$

(21) $\frac{p_{tr=1}\cdot y_{tr=1,i=1}}{p^{sca}}=\frac{x_{tr=1,i=1}\cdot \gamma_{tr=1,i=1}\cdot p^{LV}_{tr=1,i=1}}{p^{sca}}$

Display: all of [1..95]

Code Generation I – What language do you prefer?

Choose of a list of supported target platforms:



Location 15631: Full Evaluation with CAPE OPEN 10 Trays and new design Vars.moseva

Language Specification

● Predefined gPROMs NLE Std

○ User-defined

- MATLAB NLE
- MATLAB NLE Reformulated
- SCILAB NLE Std
- ACM NLE Funcs to Equas
- GAMS Functions as Equations
- AMPL Functions as Equations
- F90 NLEQ1S
- Python Scipy Functions as Equations

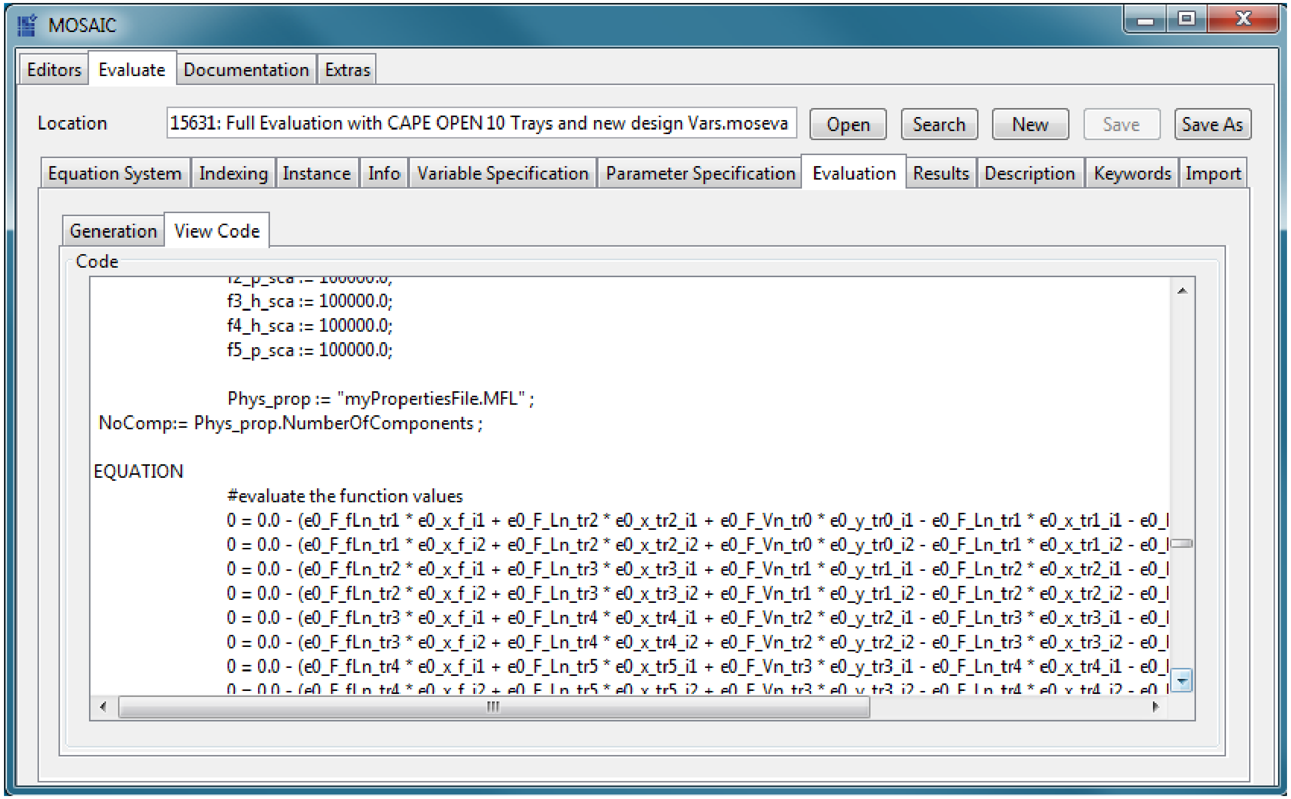
langa... <unspecified>

☑ Transfer values when changing Lang Spec

☑ Do not overwrite existing values

Code Generation II – Show me what you got!

Take a look at the generated code:



Location 15631: Full Evaluation with CAPE OPEN 10 Trays and new design Vars.moseva

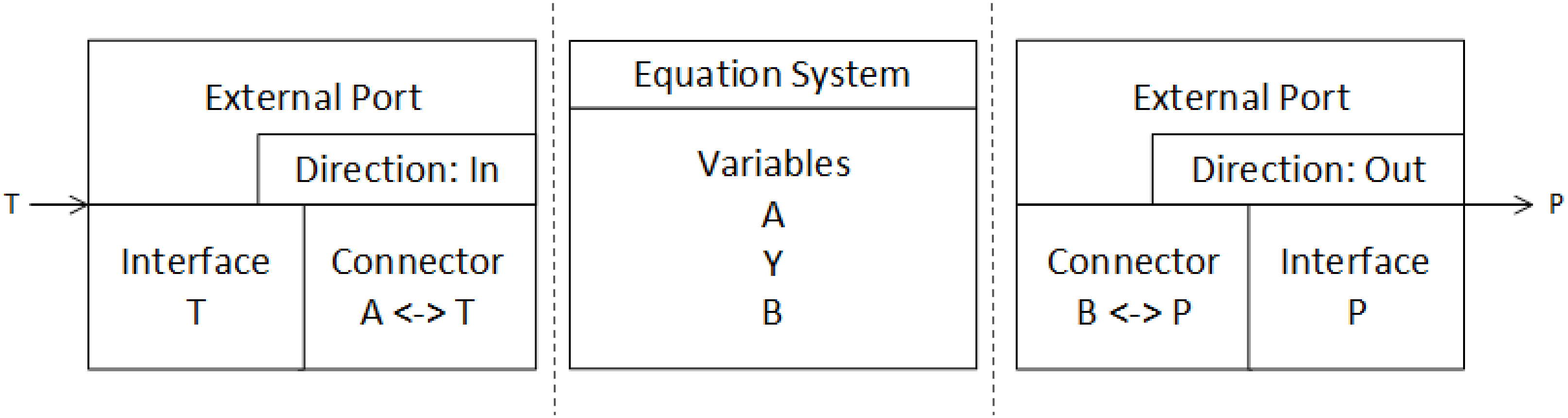
```
                    f2_p_sca := 100000.0;
                    f3_h_sca := 100000.0;
                    f4_h_sca := 100000.0;
                    f5_p_sca := 100000.0;

                    Phys_prop := "myPropertiesFile.MFL" ;
 NoComp:= Phys_prop.NumberOfComponents ;

EQUATION
                    #evaluate the function values
                    0 = 0.0 - (e0_F_fLn_tr1 * e0_x_f_i1 + e0_F_Ln_tr2 * e0_x_tr2_i1 + e0_F_Vn_tr0 * e0_y_tr0_i1 - e0_F_Ln_tr1 * e0_x_tr1_i1 - e0_
                    0 = 0.0 - (e0_F_fLn_tr1 * e0_x_f_i2 + e0_F_Ln_tr2 * e0_x_tr2_i2 + e0_F_Vn_tr0 * e0_y_tr0_i2 - e0_F_Ln_tr1 * e0_x_tr1_i2 - e0_
                    0 = 0.0 - (e0_F_fLn_tr2 * e0_x_f_i1 + e0_F_Ln_tr3 * e0_x_tr3_i1 + e0_F_Vn_tr1 * e0_y_tr1_i1 - e0_F_Ln_tr2 * e0_x_tr2_i1 - e0_
                    0 = 0.0 - (e0_F_fLn_tr2 * e0_x_f_i2 + e0_F_Ln_tr3 * e0_x_tr3_i2 + e0_F_Vn_tr1 * e0_y_tr1_i2 - e0_F_Ln_tr2 * e0_x_tr2_i2 - e0_
                    0 = 0.0 - (e0_F_fLn_tr3 * e0_x_f_i1 + e0_F_Ln_tr4 * e0_x_tr4_i1 + e0_F_Vn_tr2 * e0_y_tr2_i1 - e0_F_Ln_tr3 * e0_x_tr3_i1 - e0_
                    0 = 0.0 - (e0_F_fLn_tr3 * e0_x_f_i2 + e0_F_Ln_tr4 * e0_x_tr4_i2 + e0_F_Vn_tr2 * e0_y_tr2_i2 - e0_F_Ln_tr3 * e0_x_tr3_i2 - e0_
                    0 = 0.0 - (e0_F_fLn_tr4 * e0_x_f_i1 + e0_F_Ln_tr5 * e0_x_tr5_i1 + e0_F_Vn_tr3 * e0_y_tr3_i1 - e0_F_Ln_tr4 * e0_x_tr4_i1 - e0_
                    0 = 0.0 - (e0_F_fLn_tr4 * e0_x_f_i2 + e0_F_Ln_tr5 * e0_x_tr5_i2 + e0_F_Vn_tr3 * e0_y_tr3_i2 - e0_F_Ln_tr4 * e0_x_tr4_i2 - e0_
```

MOSAIC Ports – let‘s get connected

MOSAIC external Ports:



T

External Port

Direction: In

Interface
T

Connector
A <-> T

Equation System

Variables
A
Y
B

External Port

Direction: Out

Connector
B <-> P

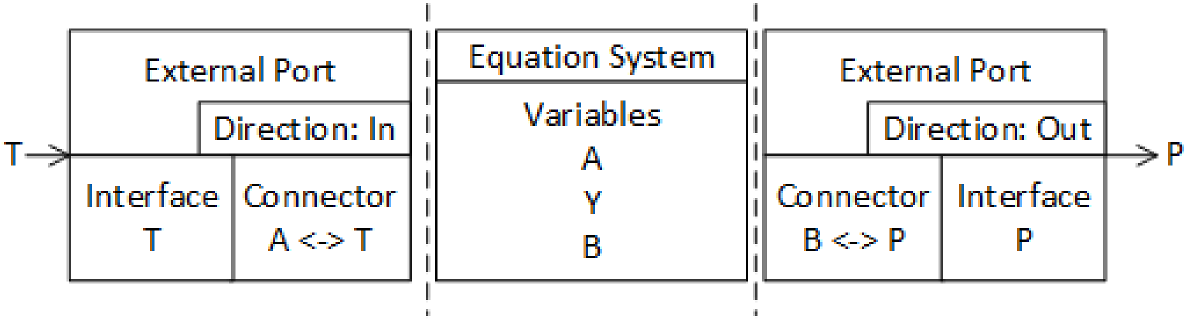
Interface
P

P

Y – internal variable
A – variable connected to input T
B – variable connected to output P

MOSAIC Ports – let‘s get connected

MOSAIC external Ports:



- Direction:

  In or Out

- Interface:

  Which variables will be presented?

  -> naming, dimension, engineering unit, direction

  e.g. p, scalar, bar, out

- Connector:

  How are internal variables and external interface variables connected?

  -> e.g. A <-> T , B <-> P

CAPE-OPEN and MOSAIC I – Physical Properties

Physical properties in MOSAIC:
Variables to be calculated by external functions, e.g. $p_i^{\mathrm{LV}}(T)$

Supported target platforms for „CO physical properties“ code generation:

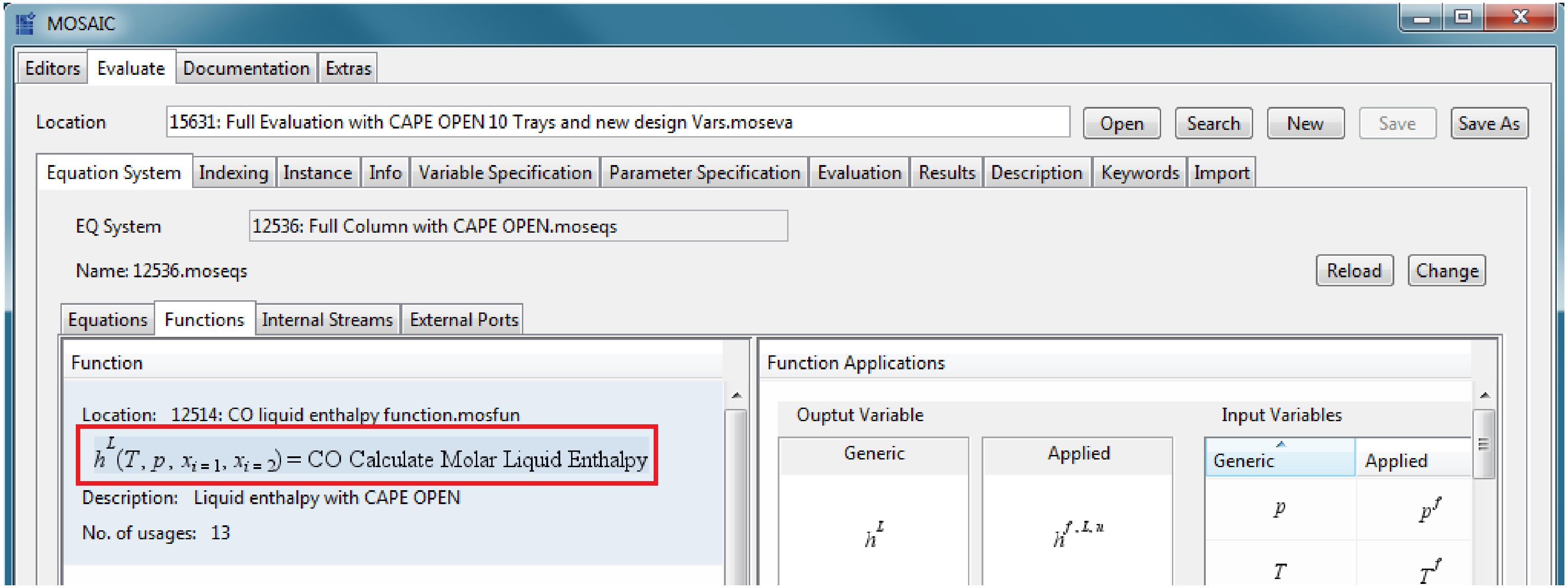
- Matlab

- gPROMS

Technische Universität Berlin

CAPE-OPEN and MOSAIC I – Physical Properties

MOSAIC Example – CO function:



Location 15631: Full Evaluation with CAPE OPEN 10 Trays and new design Vars.moseva

EQ System 12536: Full Column with CAPE OPEN.moseqs

Name: 12536.moseqs

Function

Location: 12514: CO liquid enthalpy function.mosfun

$h^{L}(T, p, x_{i=1}, x_{i=2}) =$ CO Calculate Molar Liquid Enthalpy

Description: Liquid enthalpy with CAPE OPEN

No. of usages: 13

Function Applications

Ouptut Variable

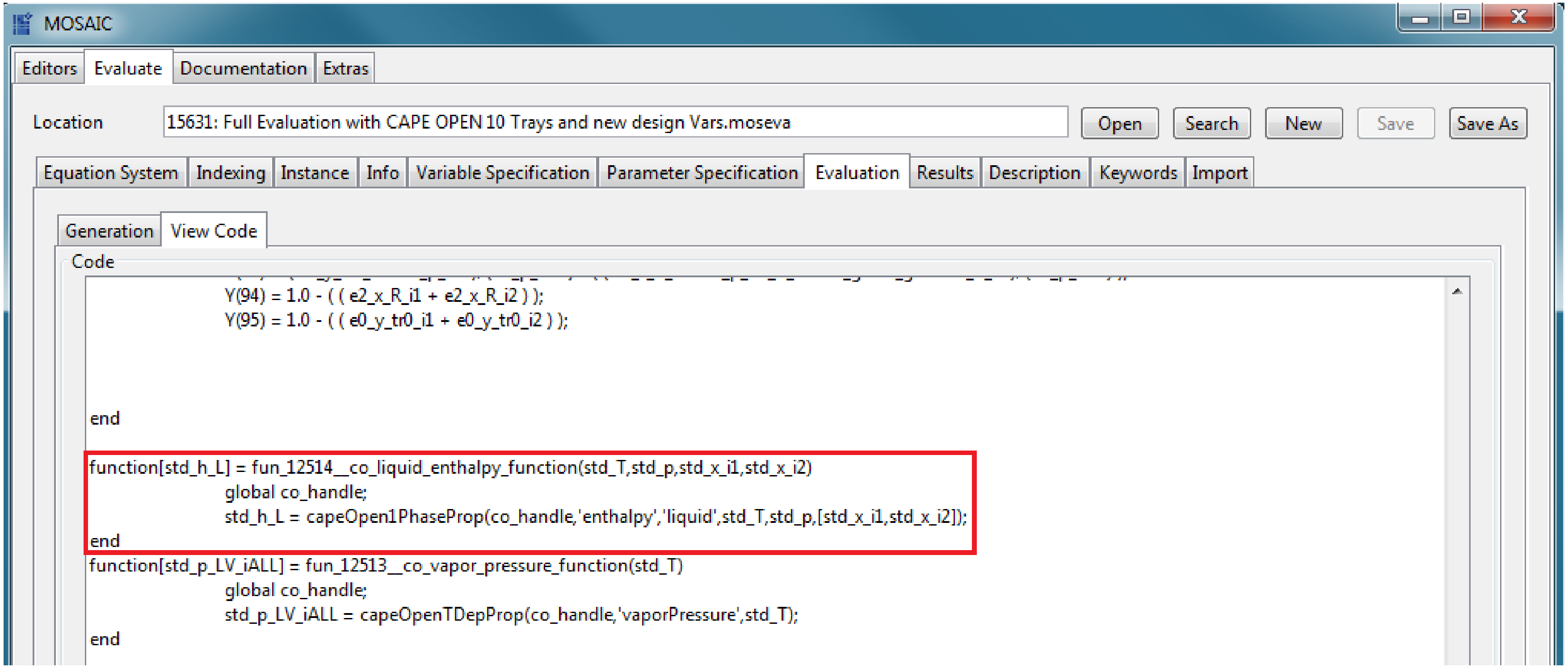
Generic	Applied
h^{L}	$h^{f,L,u}$

Input Variables

Generic	Applied
p	p^{f}
T	T^{f}

CAPE-OPEN and MOSAIC I – Physical Properties

MOSAIC Example – Matlab code:



Location: 15631: Full Evaluation with CAPE OPEN 10 Trays and new design Vars.moseva

```
                    Y(94) = 1.0 - ( ( e2_x_R_i1 + e2_x_R_i2 ) );
                    Y(95) = 1.0 - ( ( e0_y_tr0_i1 + e0_y_tr0_i2 ) );



end

function[std_h_L] = fun_12514__co_liquid_enthalpy_function(std_T,std_p,std_x_i1,std_x_i2)
                    global co_handle;
                    std_h_L = capeOpen1PhaseProp(co_handle,'enthalpy','liquid',std_T,std_p,[std_x_i1,std_x_i2]);
end
function[std_p_LV_iALL] = fun_12513__co_vapor_pressure_function(std_T)
                    global co_handle;
                    std_p_LV_iALL = capeOpenTDepProp(co_handle,'vaporPressure',std_T);
end
```

Summary

MOSAIC

- A **modular** equation based modeling tool
- Implemented in **Java**, using **XML/MathML**
- Provides **automatic code generation** for specific platforms (e.g. Matlab, C++)
- Can use the concept of **ports**
- Supports **CO physical properties** in code generation (Matlab, gPROMS)

MOSAIC is not

- Designed to be a full solver / process simulator
- A programming language
- A computer algebra system (CAS)

CAPE-OPEN and MOSAIC II – Next Steps:

- What has to be done to create CAPE-OPEN compliant Unit Operations with MOSAIC?
- What about CO Unit Operation Import and Export in MOSAIC?
- How can the MOSAIC xml models be converted into COM/CORBA objects?
- …

Thank you very much for your kind attention.

MOSAIC

Fachgebiet Dynamik und Betrieb technischer Anlagen

unicat Unifying Concepts in Catalysis

InPROMPT SFB/Transregio 63

www.mosaic-modeling.de

Acknowledgement:

unicat

This project is

- supported by the Cluster of Excellence 'Unifying Concepts in Catalysis'
- coordinated by the Technical University of Berlin and
- funded by the German Research Foundation.