



The MOSAIC Approach – Self-Made CO-UOs Without Programming Knowledge

Gregor Tolksdorf, M.Sc. | Faculty of Process Sciences | CAPE-OPEN 2015 Annual Meeting



Modular **MO**del **Sp**ecific**At**ion on Do**C**umentation Level - Application in a Web Based Modeling Environment.

www.mosaic-modeling.de

www.mosaic-modeling.de



MOSAIC



d|b|t|a
Fachgebiet
Dynamik und Betrieb technischer Anlagen

[News](#)
[MOSAIC ▾](#)
[Documentation ▾](#)
[Solvers & Environments ▾](#)
[Publications](#)
[Contact](#)
[Impressum](#)

[Forums ▾](#)

Home » Update to Version 1.8.1

≡ MOSAIC Hotlinks

- Download & Run MOSAIC
- Account and Feature Request

MOSAIC on YouTube



MOSAIC

MOSAIC is a free, web-based modeling, simulation, and optimization environment. Based on a LaTeX-style entry method for algebraic and differential equations, equation systems can be built and

Update to Version 1.8.1

18 Sep, 2015 in News by Gregor Tolksdorf

A new version of MOSAIC (1.8.1) is available now. New features, improvements, and fixes in MOSAIC 1.8.1: improvement: Nicer images for the flowsheet module improvement: User-defined langspec tests in advance if it is applicable for code generation of the selected evaluation new: code generation properties now available for user-defined language [...]





MOSAIC Workshop Bogota 2015

7 Sep, 2015 in News by Erik Esche

The latest MOSAIC workshop started on Tuesday, 8th of September 2015, in Bogota, Colombia. Access to the course material can be gained through TU Berlin's ISIS system: MOSAIC Workshop Bogota 2015 on ISIS 29 highly motivated students have worked their way through MOSAIC features such as function applications, user-defined discretisation of [...]



MOSAIC-Team

Scientific supervisor:

Prof. Dr.-Ing. G. **Wozny**,

Head of Chair of Process Dynamics and Operation, TU Berlin

d|b|t|a

Current Developer:

Dr.-Ing. E. **Esche**

Dipl.-Ing. S. **Fillinger**

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

M.Sc. G. **Tolksdorf**



Agenda

1. Motivation – The MOSAIC Approach
2. Unit Operations & Physical Properties
3. Code Generation
4. Application Example
5. Summary & Outlook



Self-Made CO-UOs Without Programming Knowledge

MOTIVATION – THE MOSAIC APPROACH



MOSAIC – Why?

Models and their application change over time.

Problems:

- Documentation is outdated
- Reimplementation is error-prone

Eutopia: Process Engineer == Programming Expert



MOSAIC offers:

- Transparent model documentation
- Automatic code generation

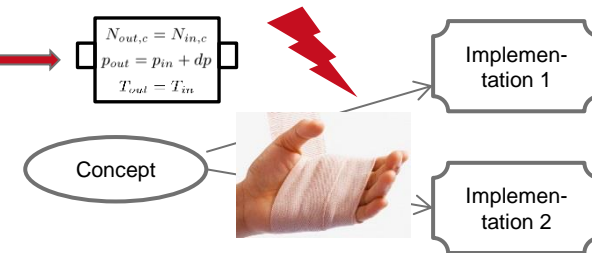
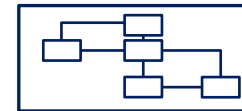


MOSAIC + CAPE-OPEN Advantages

- Encourages systematic modeling $N_{out,c} \rightarrow N_{out,c} = N_{in,c} \rightarrow \begin{matrix} N_{out,c} = N_{in,c} \\ p_{out} = p_{in} + dp \\ T_{out} = T_{in} \end{matrix} \rightarrow \begin{matrix} N_{out,c} = N_{in,c} \\ p_{out} = p_{in} + dp \\ T_{out} = T_{in} \end{matrix}$
- Eliminates redundant, error-prone manual implementation
- Enables usage of platform-independent models in virtually any simulation software based on

- Equations (z.B. PSE, ...)
- Flowsheeting (z.B. AspenTech, Pro Sim SA, Honeywell Process Solutions, Amsterchem, ...)

$$\begin{matrix} N_{out,c} = N_{in,c} \\ p_{out} = p_{in} + dp \\ T_{out} = T_{in} \end{matrix}$$



Vision: « Model once, simulate anywhere »

*Image "Hand being bandaged as injury" courtesy of Stuart Miles / FreeDigitalPhotos.net



Agenda

1. Motivation – The MOSAIC Approach
2. Unit Operations & Physical Properties
3. Code Generation
4. Application Example
5. Summary & Outlook



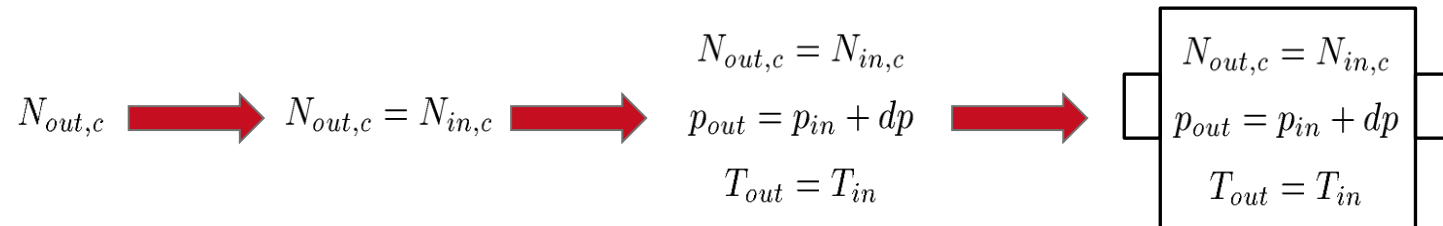
Self-Made CO-UOs Without Programming Knowledge

UNIT OPERATIONS & PHYSICAL PROPERTIES



MOSAIC – Unit Operations

A Unit Operation in MOSAIC is...



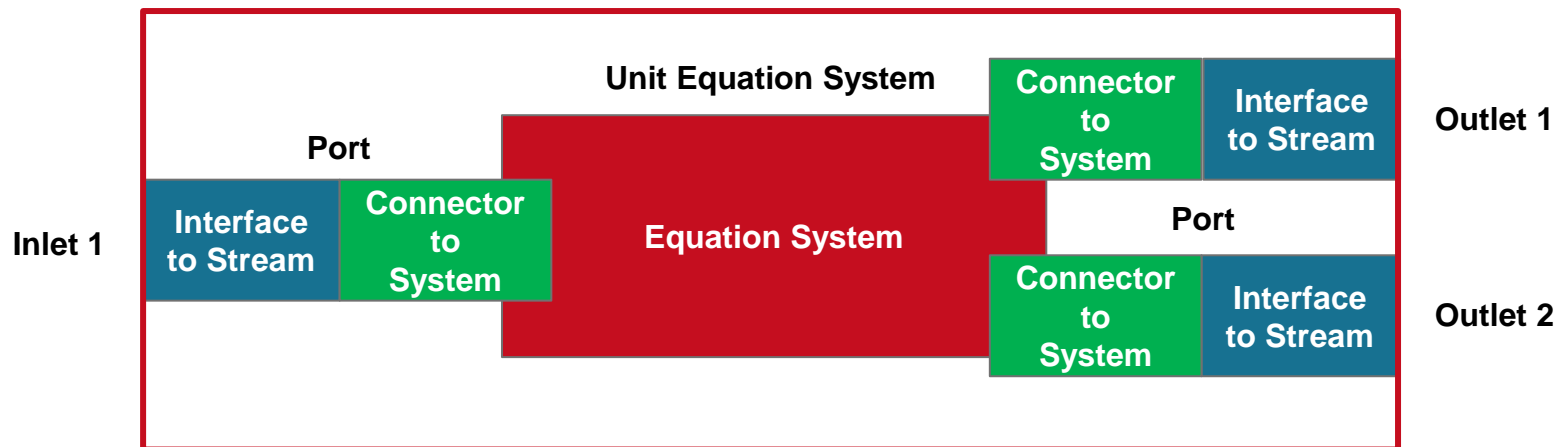
...an equation system with ports.

MOSAIC-Ports translate variables and are connection points for streams.



MOSAIC – Units and Flowsheets

The Definition of Ports:

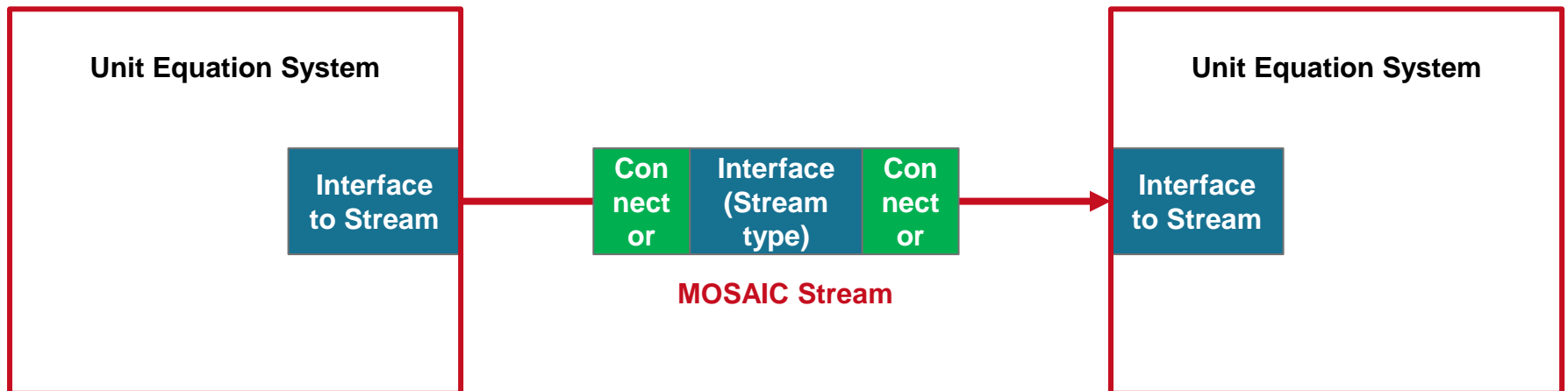


Connector connects variables from the System to the Interface

Interface defines the Stream Type (Material, Energy, etc.)



MOSAIC – Units and Flowsheets

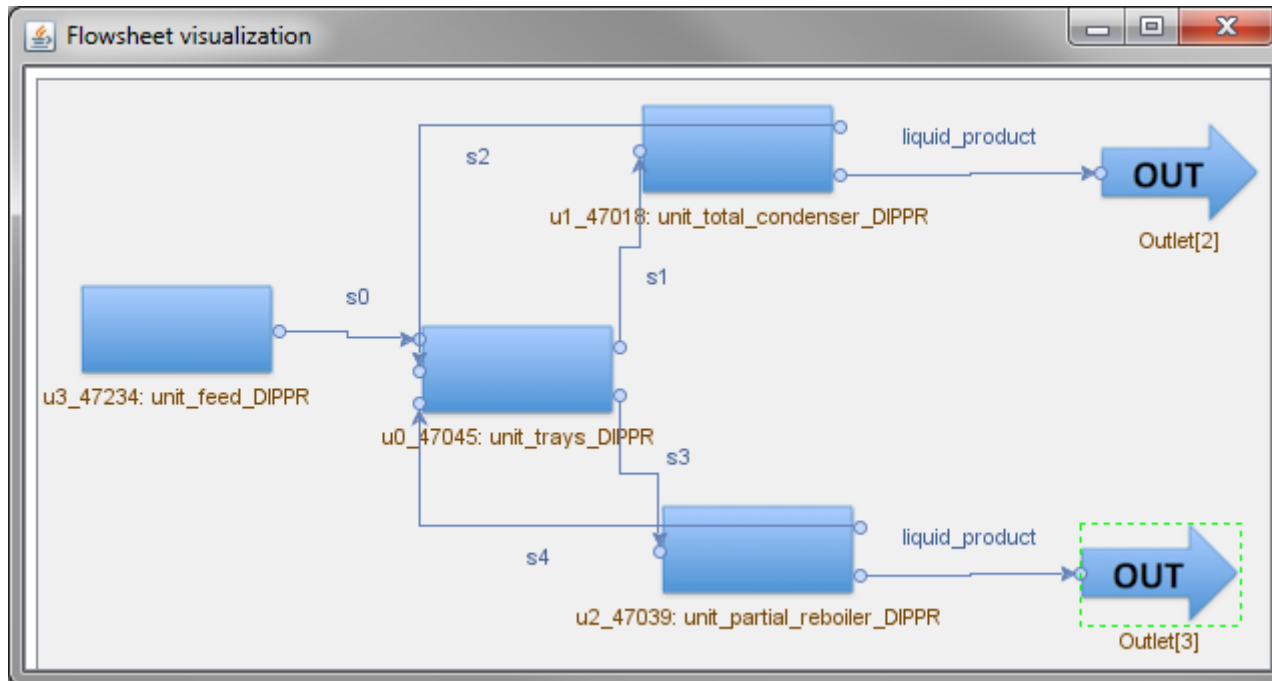


Optional Connectors for Conversion



MOSAIC – Units and Flowsheets

MOSAIC Unit Operations: „Equation-based flowsheeting“





MOSAIC – Physical Properties

(External) physical properties in MOSAIC are...

Function
Location: 12513: CO vapor pressure function.mosfun
$p_i^{LV}(T)$ = CO Calculate Vapor Pressure
Description: Vapour pressure with CAPE OPEN
No. of usages: 3

...variables that are calculated by external, language-specific functions.

Example code (using Amsterchem's „Matlab CAPE-OPEN Thermo Import“):

```
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
```




Agenda

1. Motivation – The MOSAIC Approach
2. Unit Operations & Physical Properties
3. Code Generation
4. Application Example
5. Summary & Outlook



Self-Made CO-UOs Without Programming Knowledge

CODE GENERATION



MOSAIC – Code Generation

MOSAIC offers code generation for several **predefined** languages:



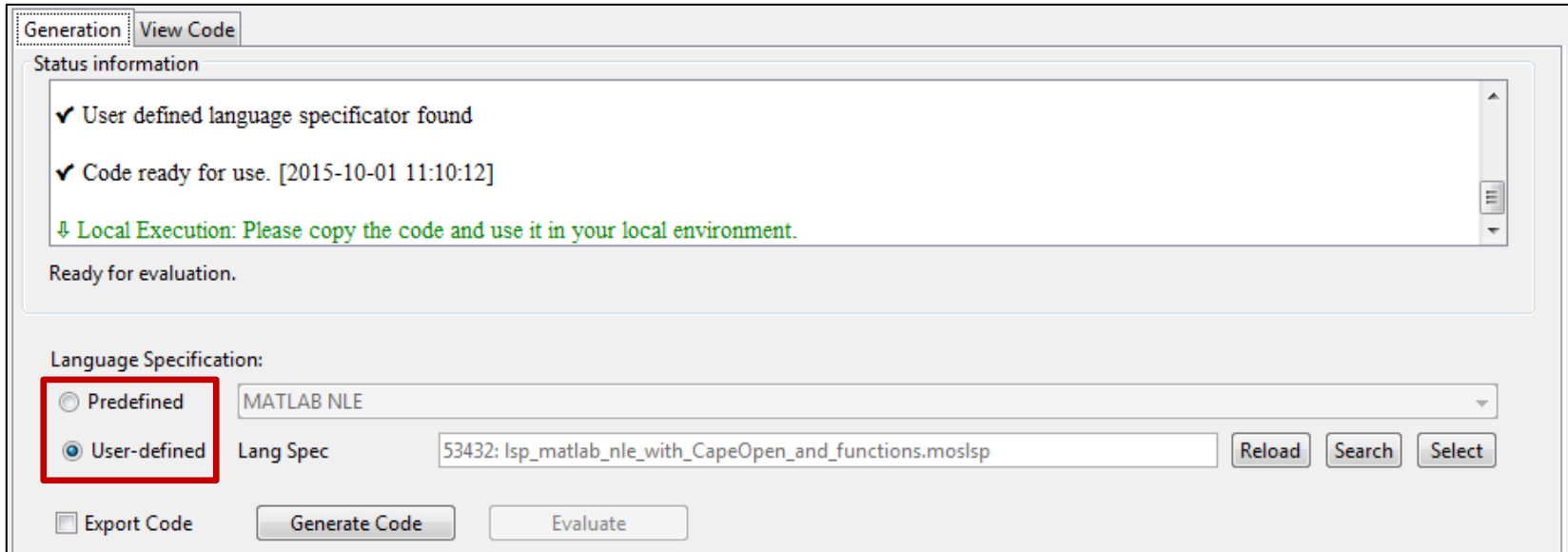
The predefined languages cover

- general purpose programming languages
- specific environments for simulation
- optimization languages



MOSAIC – Code Generation

MOSAIC users can define code generators for **new** languages.



The screenshot shows the MOSAIC code generation interface. It has two tabs: "Generation" (selected) and "View Code". Under the "Generation" tab, there is a "Status information" section with a list of messages: "✓ User defined language specifiicator found", "✓ Code ready for use. [2015-10-01 11:10:12]", and "↓ Local Execution: Please copy the code and use it in your local environment." Below this, it says "Ready for evaluation." The "Language Specification:" section has two radio buttons: "Predefined" and "User-defined". The "User-defined" button is selected and highlighted with a red box. To the right of the radio buttons is a dropdown menu showing "MATLAB NLE". Below the dropdown is a text field labeled "Lang Spec" containing the path "53432: lsp_matlab_nle_with_CapeOpen_and_functions.moslsp". To the right of the text field are three buttons: "Reload", "Search", and "Select". At the bottom, there is a checkbox labeled "Export Code" and two buttons: "Generate Code" and "Evaluate".

Example:

Code generator for Amsterchem's „Scilab CAPE-OPEN Unit Operation“



Agenda

1. Motivation – The MOSAIC Approach
2. Unit Operations & Physical Properties
3. Code Generation
4. Application Example
5. Summary & Outlook

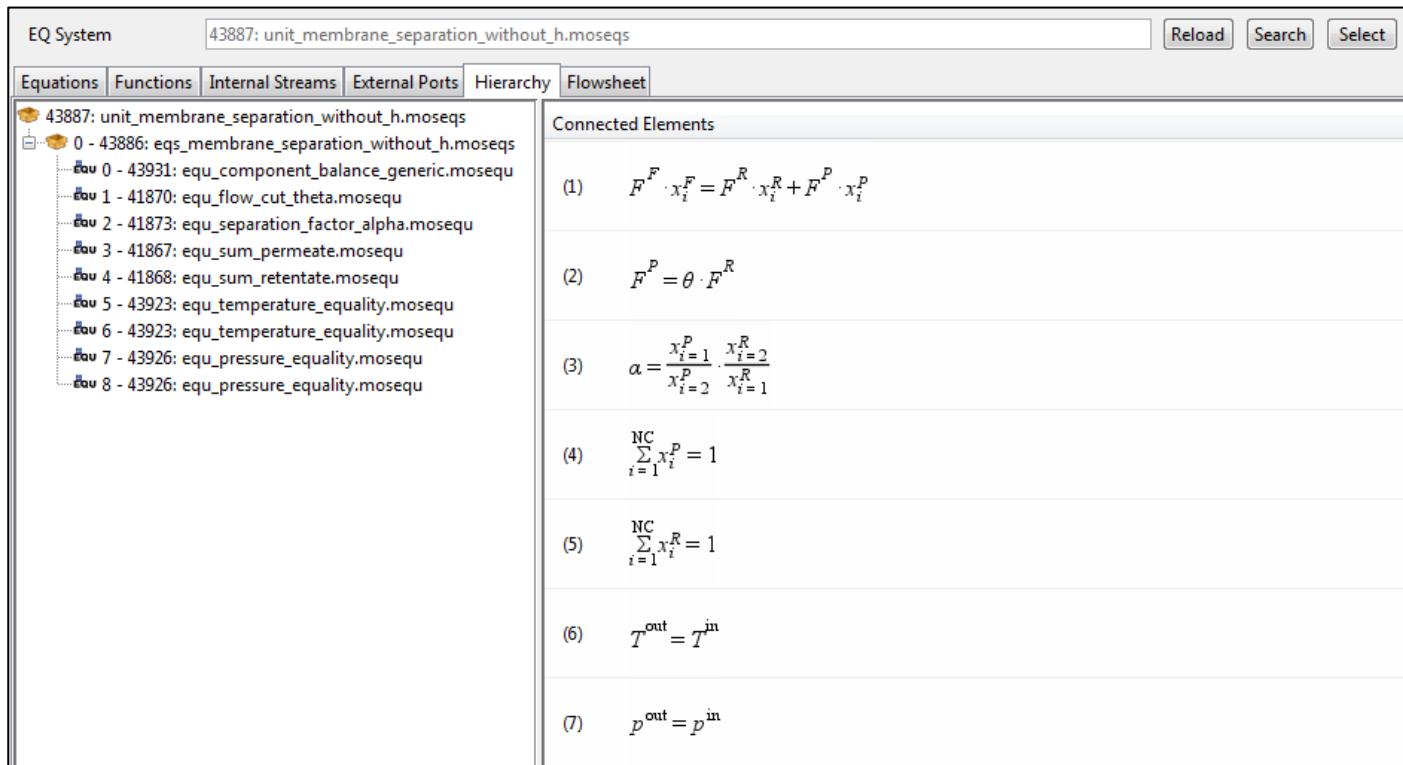


Self-Made CO-UOs Without Programming Knowledge

APPLICATION EXAMPLE

Application Example - Simple Membrane Separation

 MOSAIC:



The screenshot displays the MOSAIC software interface for a process model titled "43887: unit_membrane_separation_without_h.moseqs". The interface includes a tree view on the left showing the model hierarchy, with sub-models like "eqs_membrane_separation_without_h.moseqs" and various component balance and equality equations. The right pane, titled "Connected Elements", lists seven equations:

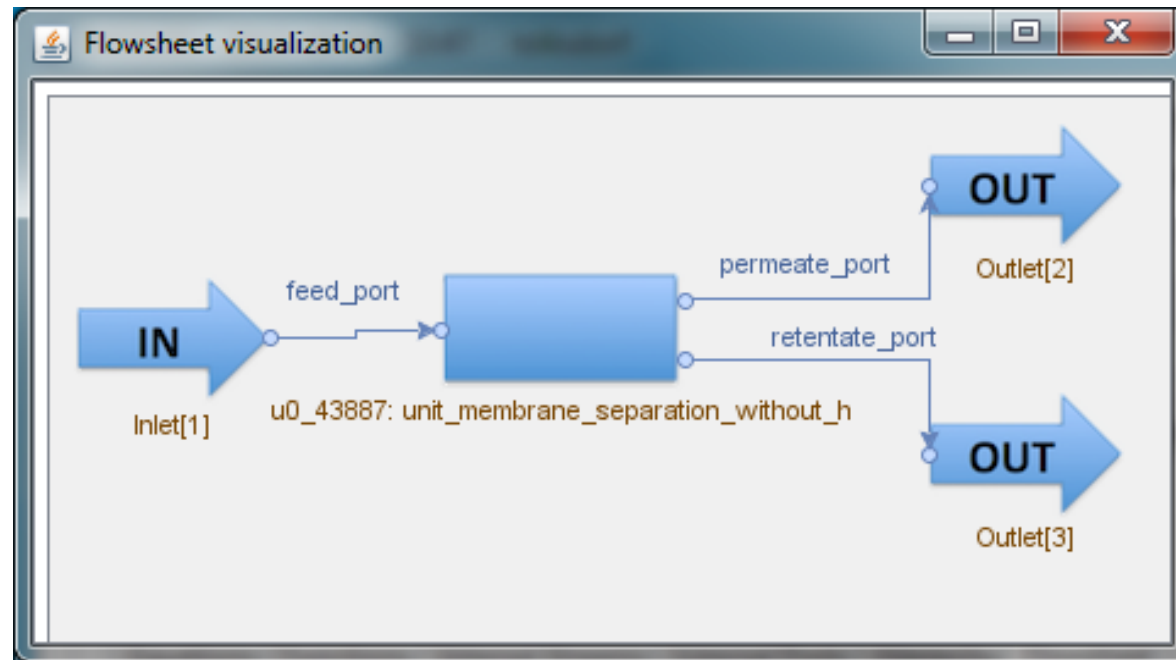
- (1) $F^F \cdot x_i^F = F^R \cdot x_i^R + F^P \cdot x_i^P$
- (2) $F^P = \theta \cdot F^R$
- (3) $\alpha = \frac{x_{i=1}^P}{x_{i=2}^P} \cdot \frac{x_{i=2}^R}{x_{i=1}^R}$
- (4) $\sum_{i=1}^{NC} x_i^P = 1$
- (5) $\sum_{i=1}^{NC} x_i^R = 1$
- (6) $T^{\text{out}} = T^{\text{in}}$
- (7) $p^{\text{out}} = p^{\text{in}}$

*van Baten, Taylor, Kooijman: „Using Chemsep, COCO and other modeling tools for versatility in custom process modeling“, AIChE 2010



Application Example - Simple Membrane Separation

 MOSAIC:

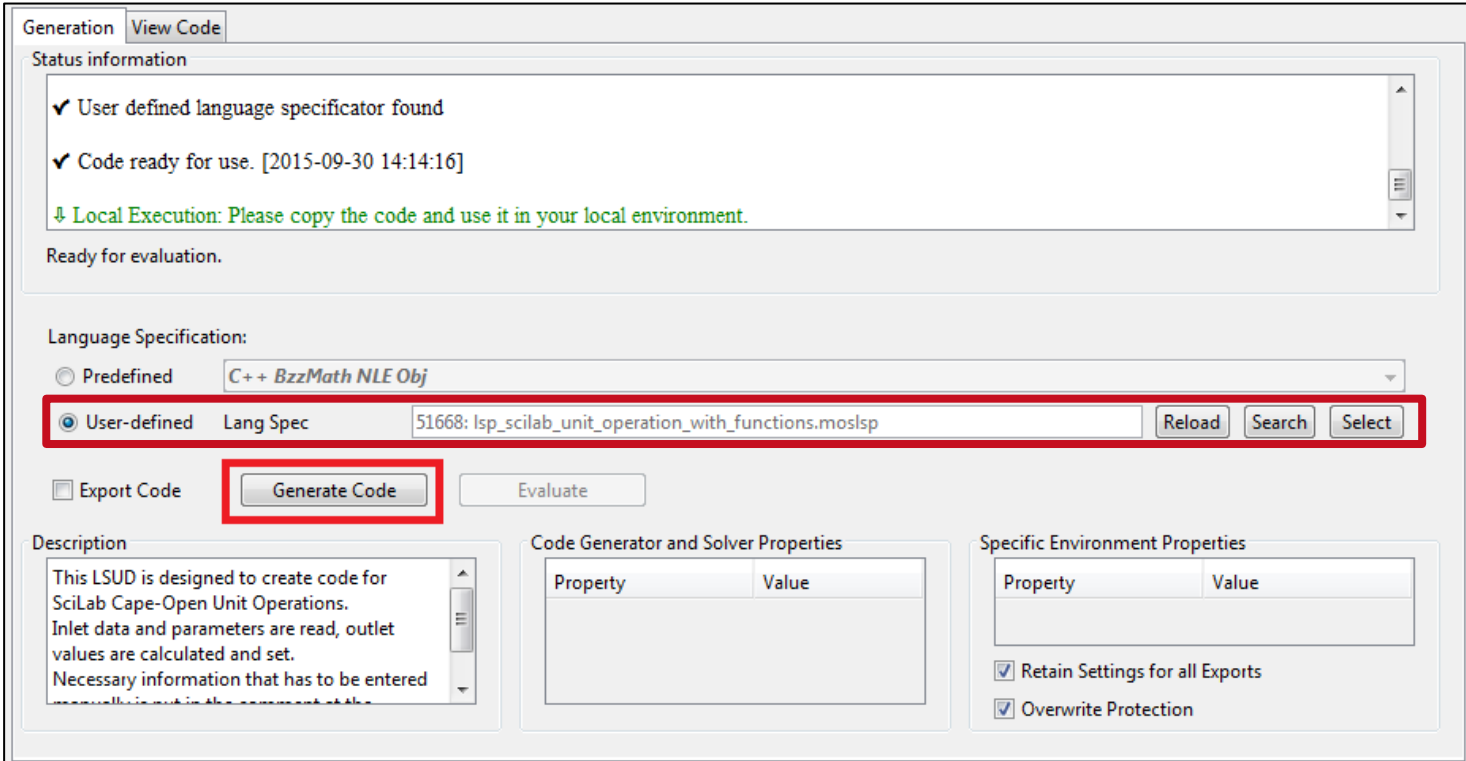


*van Baten, Taylor, Kooijman: „Using Chemsep, COCO and other modeling tools for versatility in custom process modeling“, AIChE 2010

Self-Made CO-UOs Without Programming Knowledge | G. Tolsdorf | CAPE-OPEN 2015 Annual Meeting

Application Example - Simple Membrane Separation

MOSAIC:



Generation View Code

Status information

- ✓ User defined language specifier found
- ✓ Code ready for use. [2015-09-30 14:14:16]
- ⚡ Local Execution: Please copy the code and use it in your local environment.

Ready for evaluation.

Language Specification:

☐ Predefined C++ BzzMath NLE Obj

☒ User-defined Lang Spec 51668: lsp_sciLab_unit_operation_with_functions.moslsp Reload Search Select

☐ Export Code **Generate Code** Evaluate

Description

This LSUD is designed to create code for SciLab Cape-Open Unit Operations. Inlet data and parameters are read, outlet values are calculated and set. Necessary information that has to be entered manually is put in the comment at the

Code Generator and Solver Properties

Property	Value

Specific Environment Properties

Property	Value

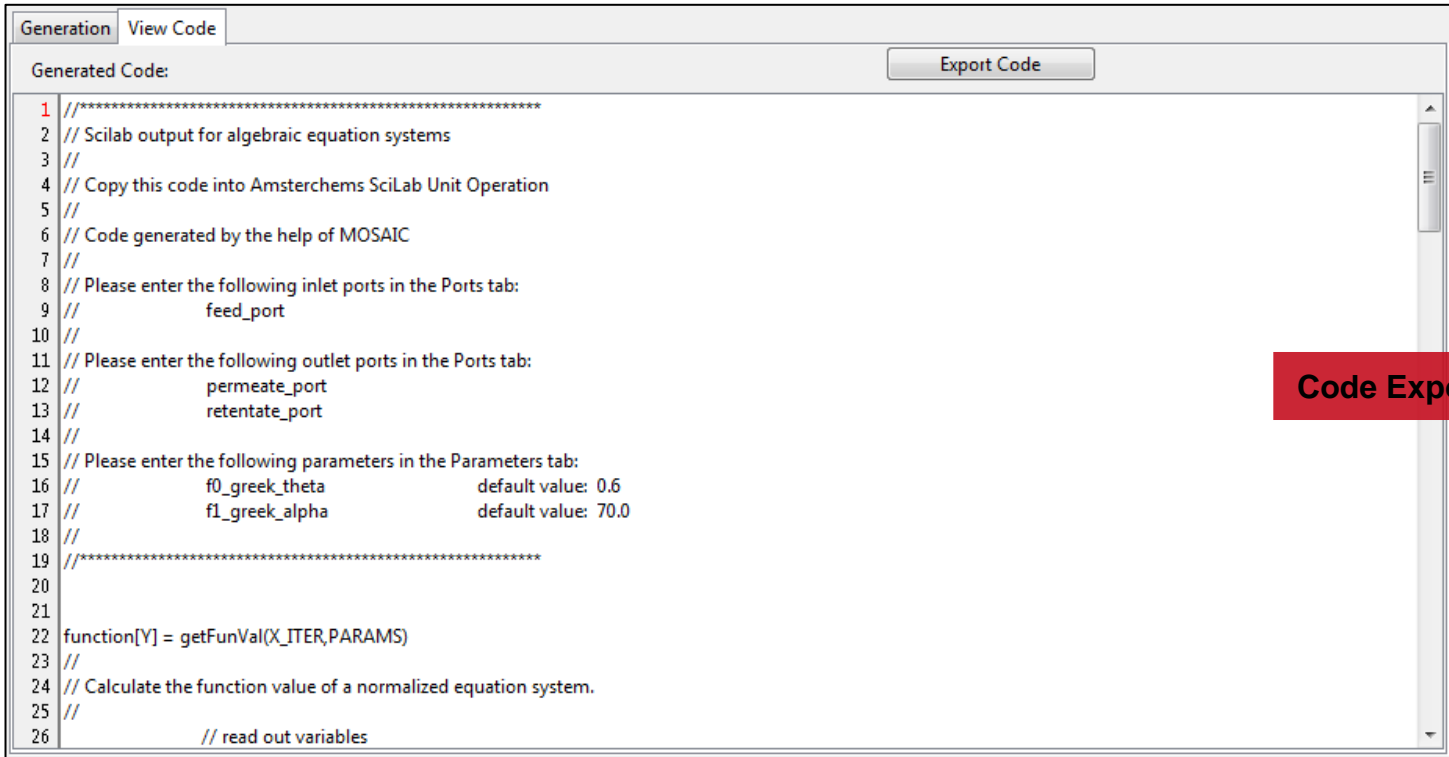
☒ Retain Settings for all Exports

☒ Overwrite Protection

*van Baten, Taylor, Kooijman: „Using Chemsep, COCO and other modeling tools for versatility in custom process modeling“, AIChE 2010

Application Example - Simple Membrane Separation

 MOSAIC:



The screenshot shows the MOSAIC software interface. It has two tabs: 'Generation' and 'View Code'. The 'View Code' tab is active, displaying a text area with generated code. The code is a Scilab script for algebraic equation systems, generated by MOSAIC. It includes comments for inlet and outlet ports and parameters. An 'Export Code' button is located in the top right corner of the code area.

```

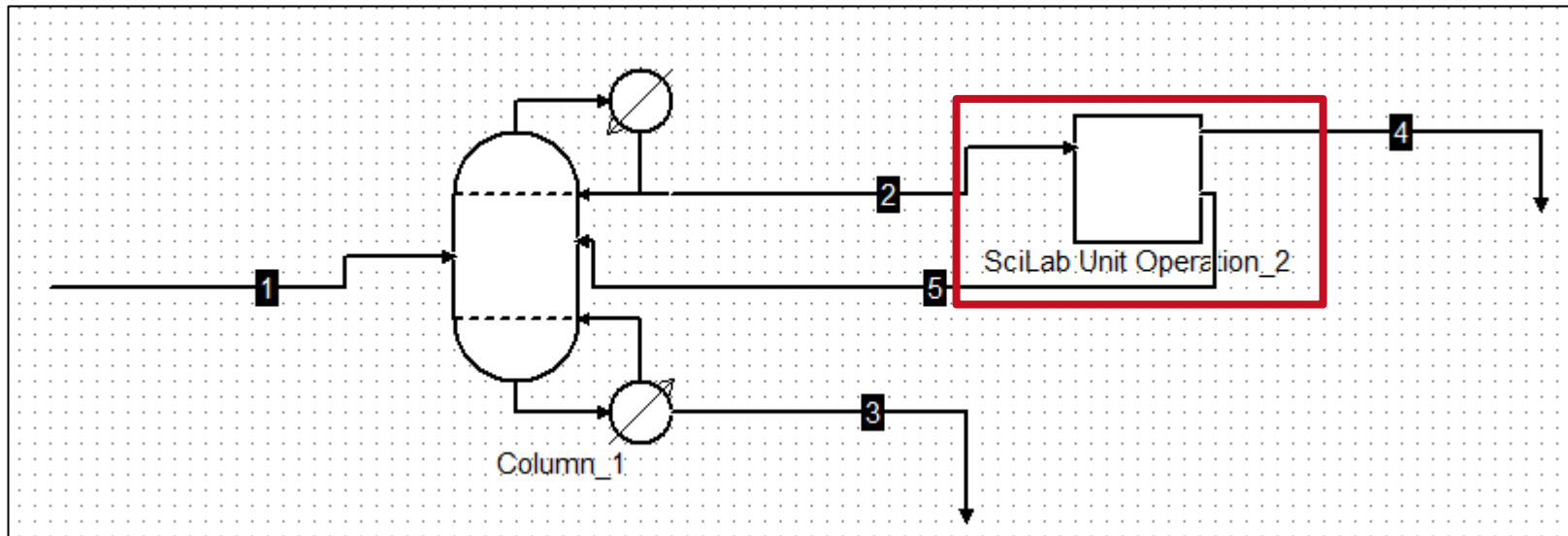
1 //*****
2 // Scilab output for algebraic equation systems
3 //
4 // Copy this code into Amsterchems SciLab Unit Operation
5 //
6 // Code generated by the help of MOSAIC
7 //
8 // Please enter the following inlet ports in the Ports tab:
9 //     feed_port
10 //
11 // Please enter the following outlet ports in the Ports tab:
12 //     permeate_port
13 //     retentate_port
14 //
15 // Please enter the following parameters in the Parameters tab:
16 //     f0_greek_theta      default value: 0.6
17 //     f1_greek_alpha      default value: 70.0
18 //
19 //*****
20
21
22 function[Y] = getFunVal(X_ITER,PARAMS)
23 //
24 // Calculate the function value of a normalized equation system.
25 //
26 // read out variables
  
```

Code Export

*van Baten, Taylor, Kooijman: „Using Chemsep, COCO and other modeling tools for versatility in custom process modeling“, AIChE 2010

Application Example - Simple Membrane Separation

 COCO/COFE:



*van Baten, Taylor, Kooijman: „Using Chemsep, COCO and other modeling tools for versatility in custom process modeling“, AIChE 2010

Self-Made CO-UOs Without Programming Knowledge | G. Tolsdorf | CAPE-OPEN 2015 Annual Meeting

Application Example - Simple Membrane Separation



Code Export

SciLab CAPE-OPEN Unit Operation:

Ports | Parameters | Reports | SciLab | Additional files | About

SciLab script info level: default echo

```

//*****
// Scilab output for algebraic equation systems
// Copy this code into Amsterchems SciLab Unit Operation
// Code generated by the help of MOSAIC
// Please enter the following inlet ports in the Ports tab:
//   feed_port
// Please enter the following outlet ports in the Ports tab:
//   permeate_port
//   retentate_port
// Please enter the following parameters in the Parameters tab:
//   f0_greek_theta      default value: 0.6
//   f1_greek_alpha      default value: 70.0
//*****

```

Script | Output

Test | Edit | SciLab Help

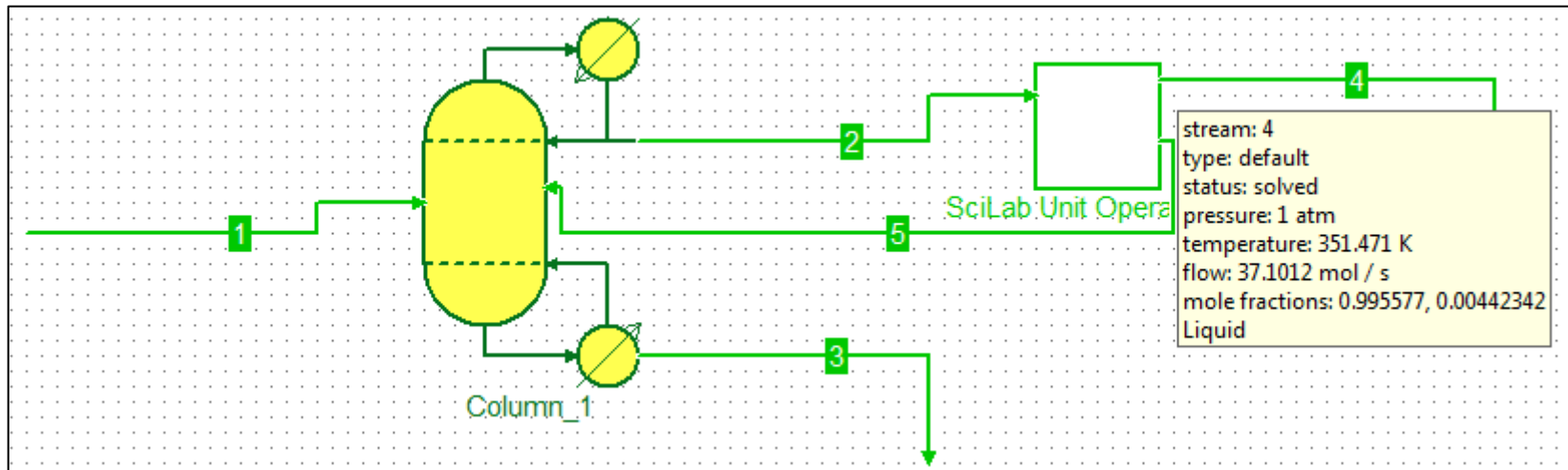
Save model | Load model | Help | Close

*van Baten, Taylor, Kooijman: „Using Chemsep, COCO and other modeling tools for versatility in custom process modeling“ ,AIChE 2010

Self-Made CO-UOs Without Programming Knowledge | G. Tolsdorf | CAPE-OPEN 2015 Annual Meeting

Application Example - Simple Membrane Separation

 COCO/COFE:



*van Baten, Taylor, Kooijman: „Using Chemsep, COCO and other modeling tools for versatility in custom process modeling“, AIChE 2010

Self-Made CO-UOs Without Programming Knowledge | G. Tolsdorf | CAPE-OPEN 2015 Annual Meeting



Agenda

1. Motivation – The MOSAIC Approach
2. Unit Operations & Physical Properties
3. Code Generation
4. Application Example
5. Summary & Outlook



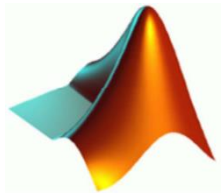
Self-Made CO-UOs Without Programming Knowledge

SUMMARY & OUTLOOK

Summary & Outlook

MOSAIC

- is an equation-based modeling and code generation tool
- covers unit operations and physical property calls
- automatically generates code for various programming languages, including
 - MatLab/SciLab Cape-Open UnitOperations by Amsterchem





Summary & Outlook

Vision:

« Model once, simulate anywhere »
by creating CO-UOs with MOSAIC

Next steps:

- C++ code generation for Cape-Open unit operations
- Direct delivery of a Cape-Open unit operation shared library (DLL)



Thank you very much for your kind attention.



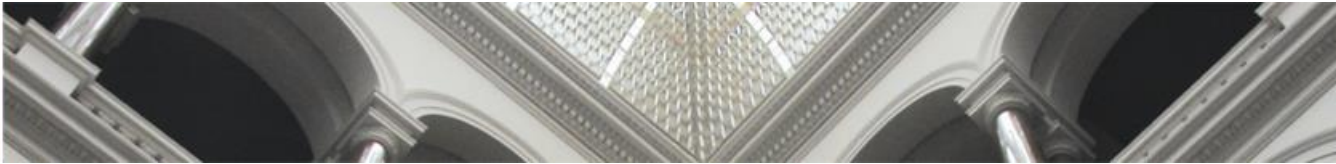
www.mosaic-modeling.de

Acknowledgement:



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.



Self-Made CO-UOs Without Programming Knowledge

BACKUP SLIDES



Goals and Resulting Characteristics

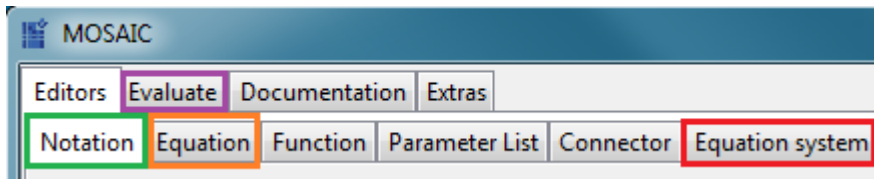
Main goals:

- 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

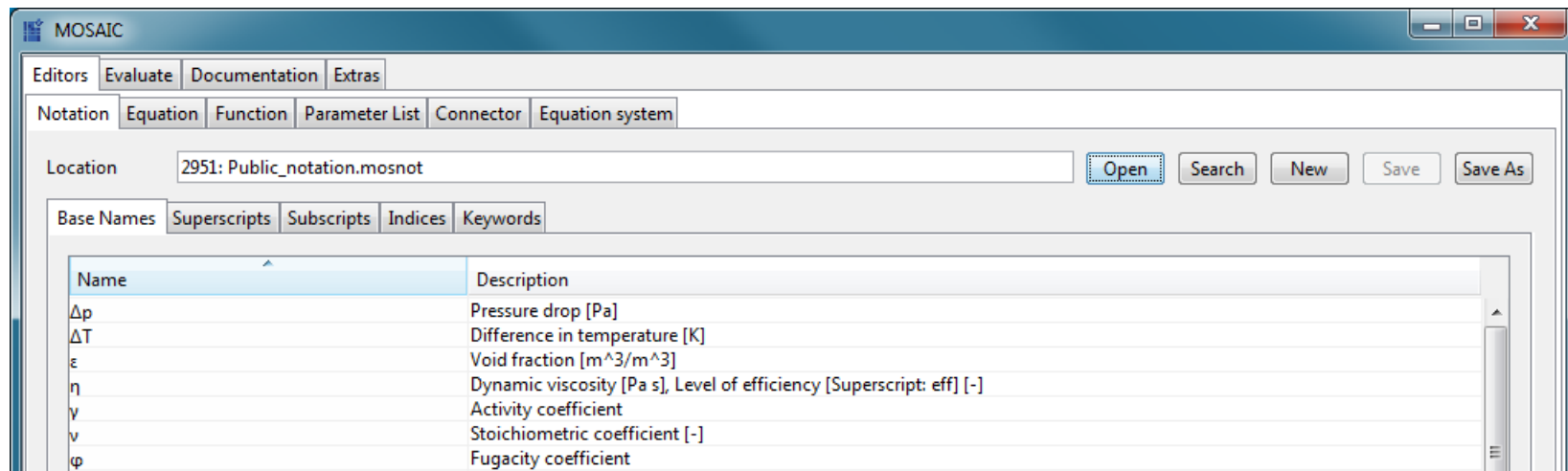
– How will the equations be combined? What functions will be used?

Evaluation

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

Enhanced Symbolic Notation I - Variables

Notation editor:

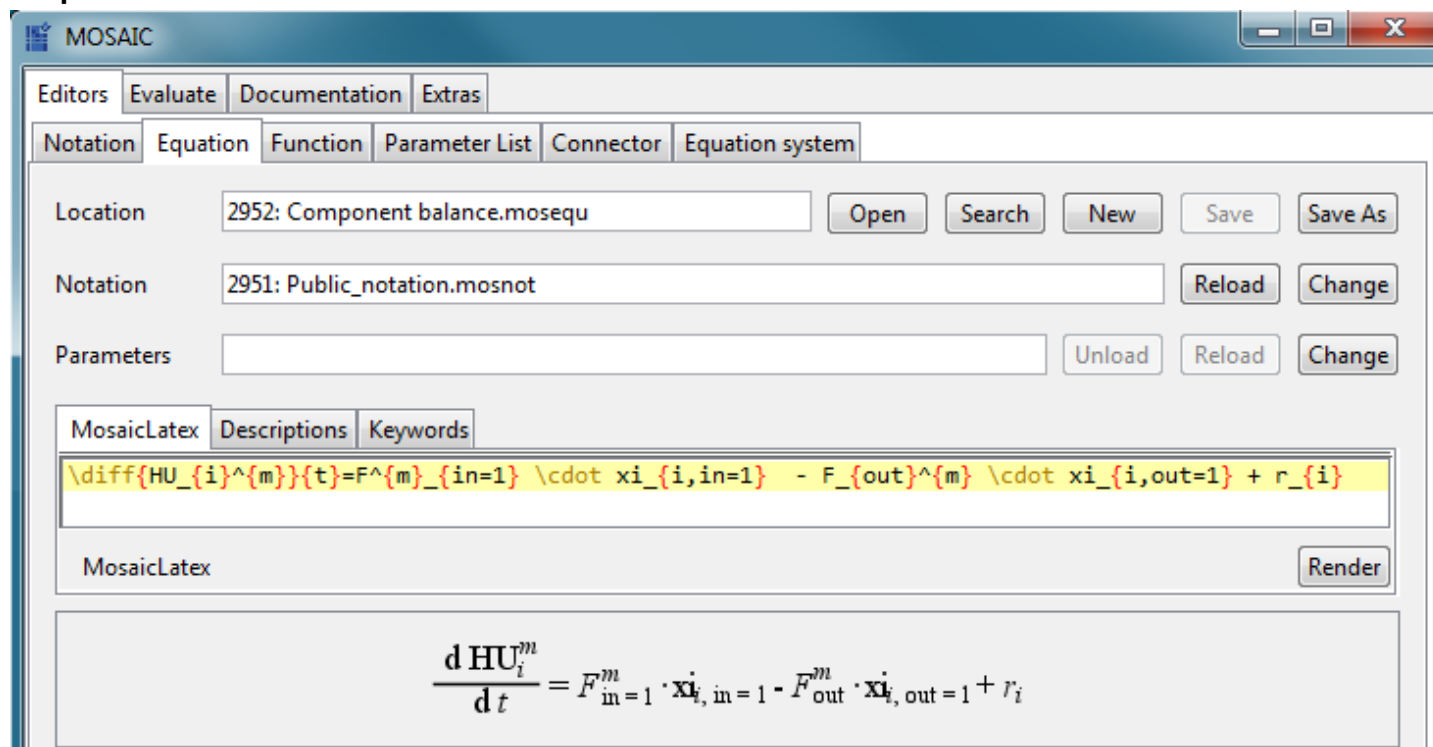


Example:

$$p_{o,i=2,j=4}^{LV,I}$$

Enhanced Symbolic Notation II - Equations

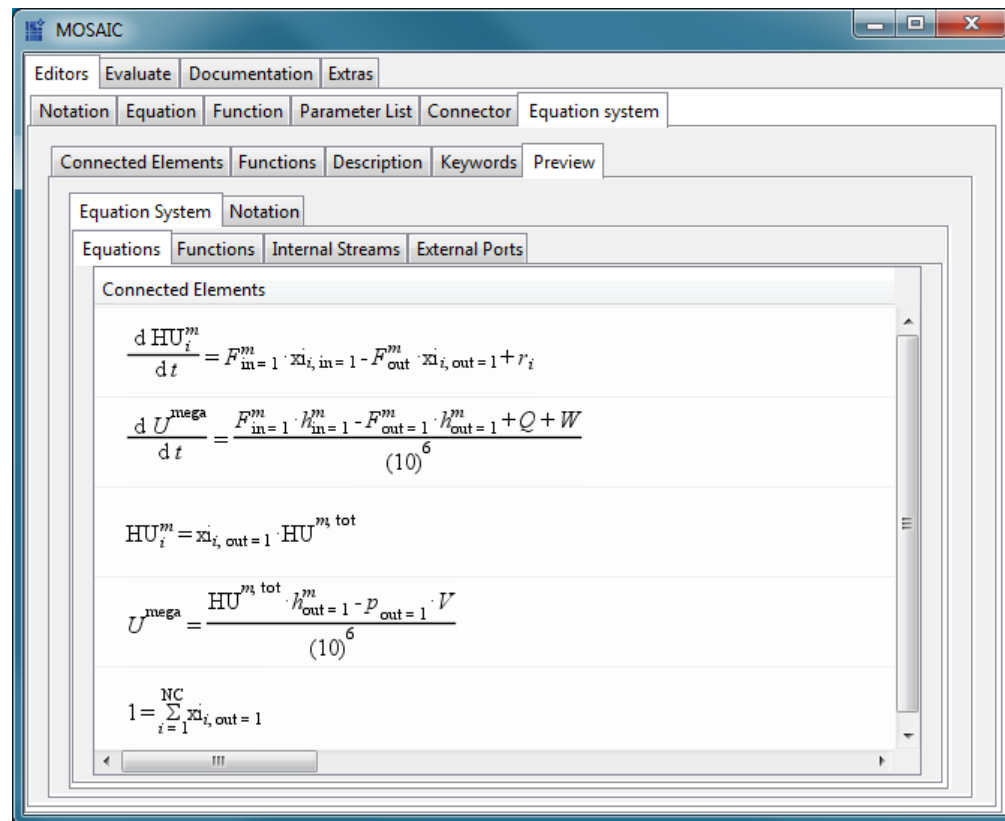
Equation editor:





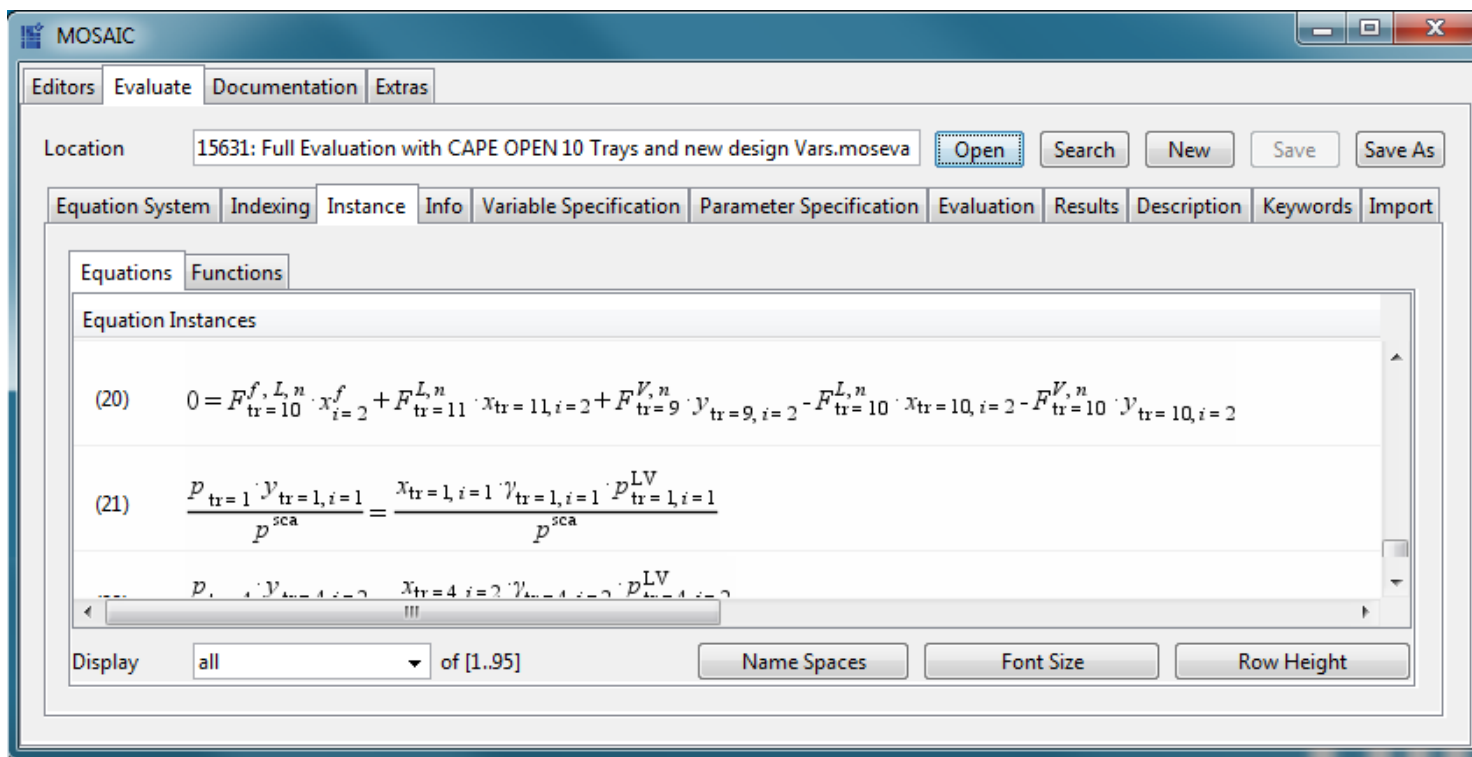
Enhanced Symbolic Notation III – Equation Systems

Equation system editor:



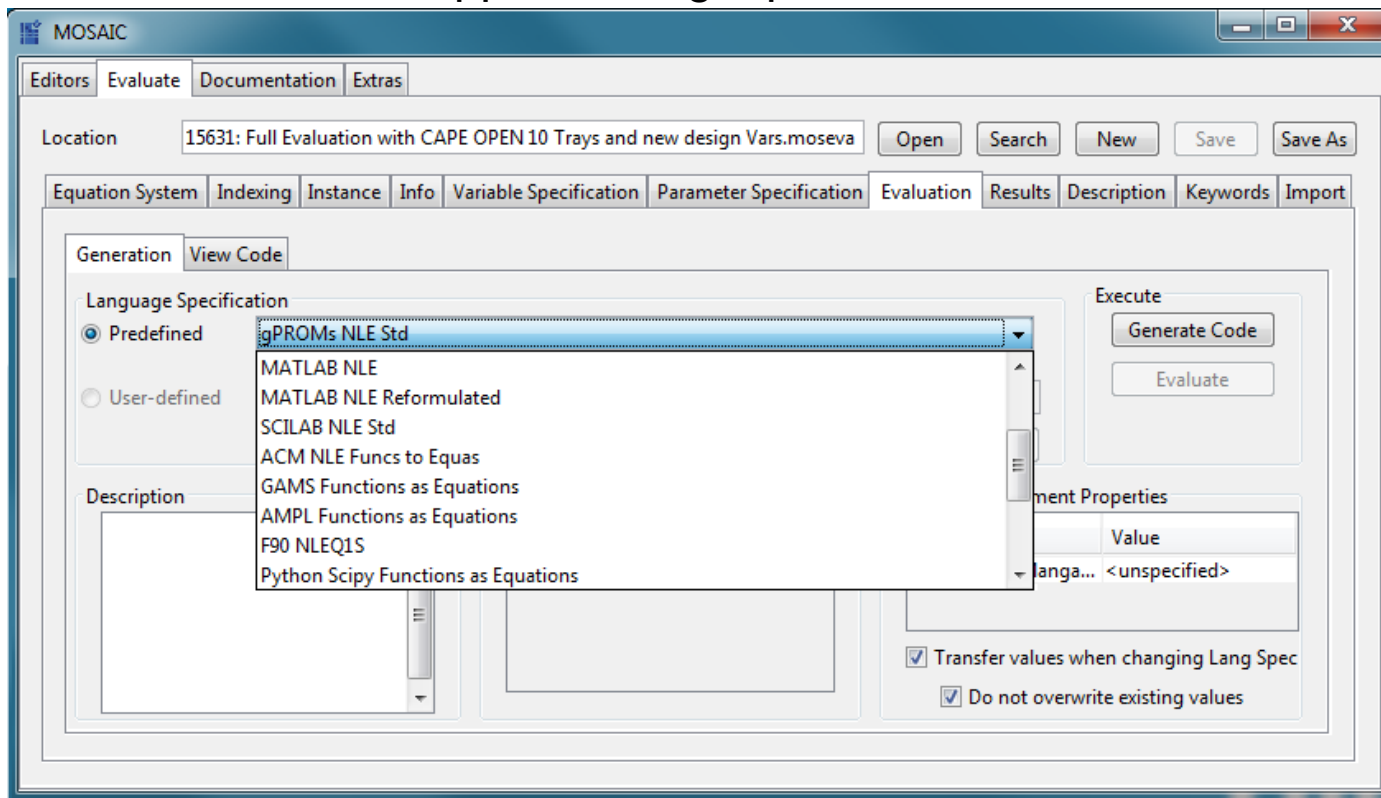
Enhanced Symbolic Notation IV – Instantiated Equations

Evaluation editor:



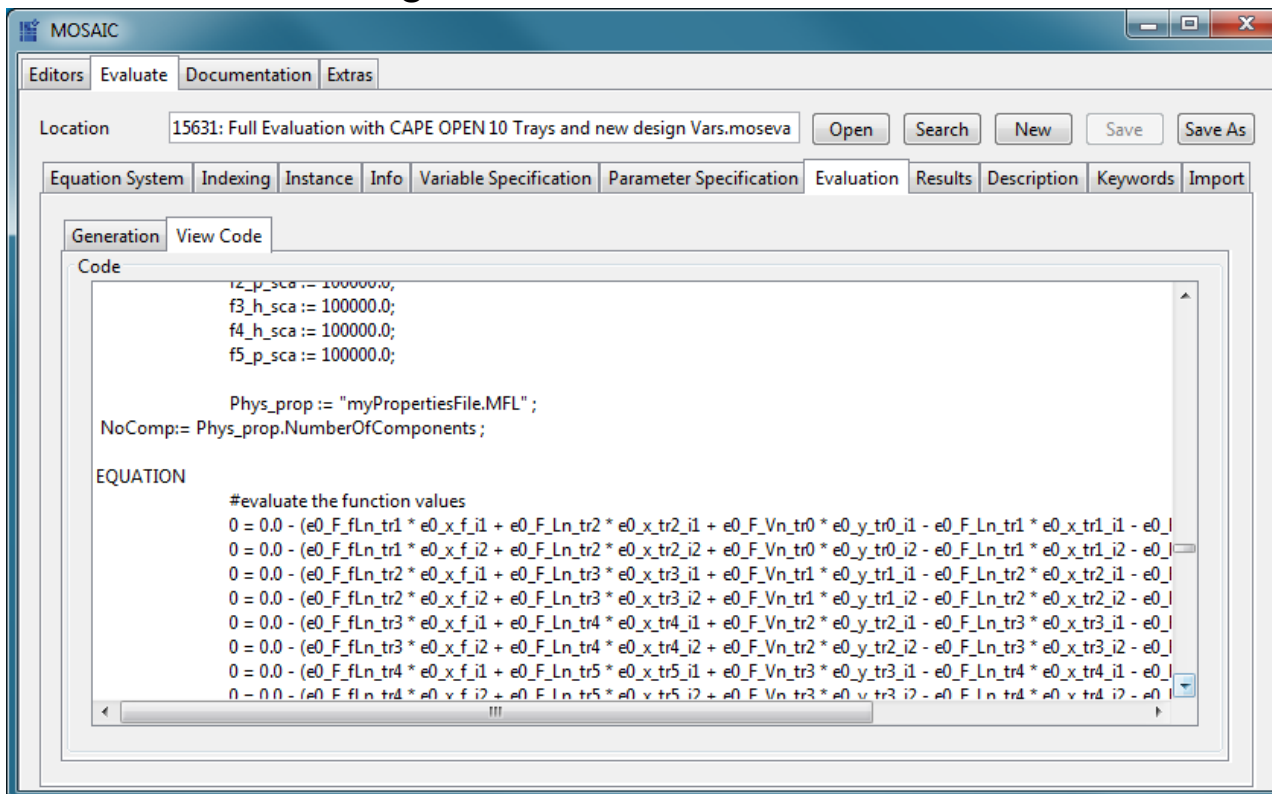
Code Generation I – What language do you prefer?

Choose of a list of supported target platforms:



Code Generation II – Show me what you got!

Take a look at the generated code:



The screenshot shows the MOSAIC software interface. The 'Generation' tab is active, and the 'View Code' sub-tab is selected. The code editor displays the following content:

```

Location 15631: Full Evaluation with CAPE OPEN 10 Trays and new design Vars.moseva [Open] [Search] [New] [Save] [Save As]

Equation System [Indexing] [Instance] [Info] [Variable Specification] [Parameter Specification] [Evaluation] [Results] [Description] [Keywords] [Import]

Generation View Code
Code
f2_p_sca := 100000.0;
f3_h_sca := 100000.0;
f4_h_sca := 100000.0;
f5_p_sca := 100000.0;

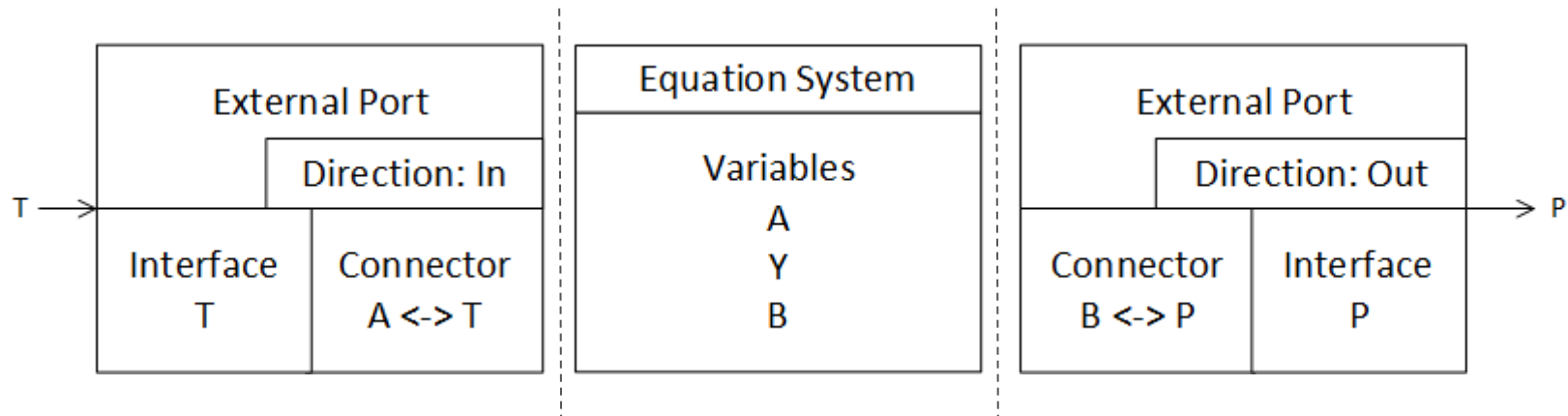
Phys_prop := "myPropertiesFile.MFL";
NoComp:= Phys_prop.NumberOfComponents;

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_L
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_L
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_L
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_L
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_L
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_L
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_L
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_L

```

MOSAIC Ports – let's get connected

MOSAIC external Ports:



Y – internal variable

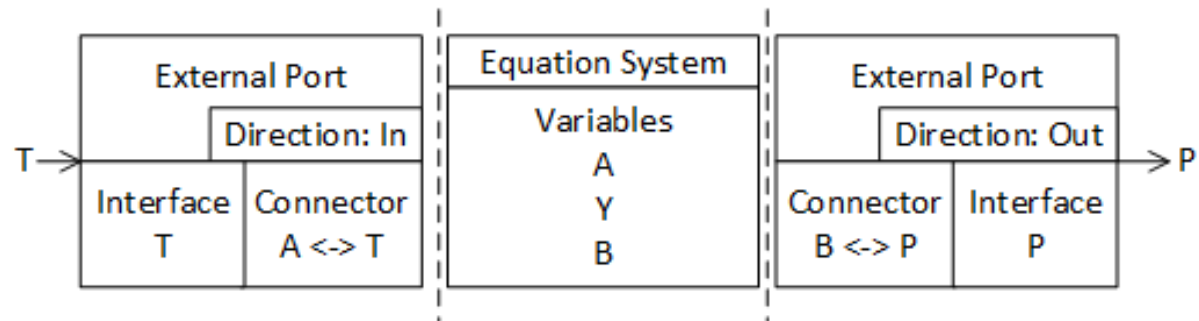
A – variable connected to input T

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. $A \leftrightarrow T$, $B \leftrightarrow P$



CAPE-OPEN and MOSAIC I – Physical Properties

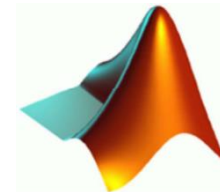
Physical properties in MOSAIC:

Variables to be calculated by external functions, e.g.

$$p_i^{LV}(T)$$

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

- Matlab

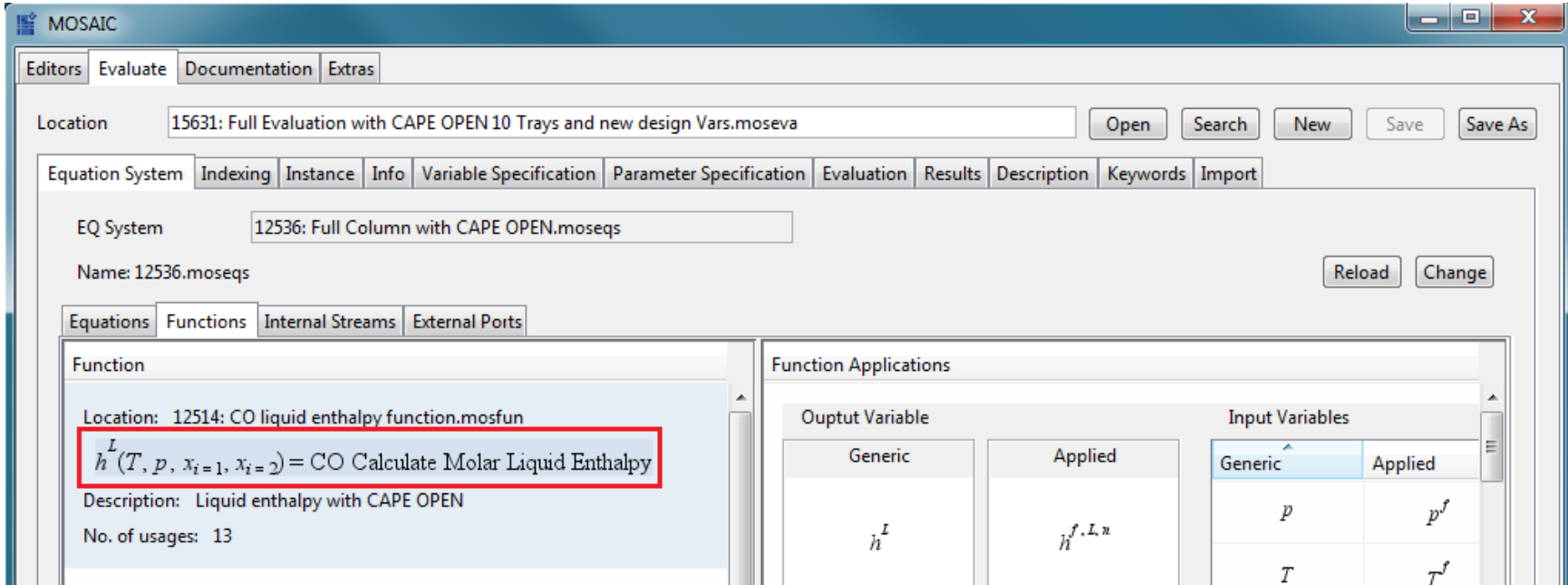


- gPROMS



CAPE-OPEN and MOSAIC I – Physical Properties

MOSAIC Example – CO function:



The screenshot shows the MOSAIC software interface. The 'Location' field displays '15631: Full Evaluation with CAPE OPEN 10 Trays and new design Vars.moseva'. The 'EQ System' field shows '12536: Full Column with CAPE OPEN.moseqs'. The 'Name' field is '12536.moseqs'. The 'Functions' tab is selected, showing a list of functions. The function 'CO liquid enthalpy function.mosfun' is highlighted, and its definition is shown in a red box:

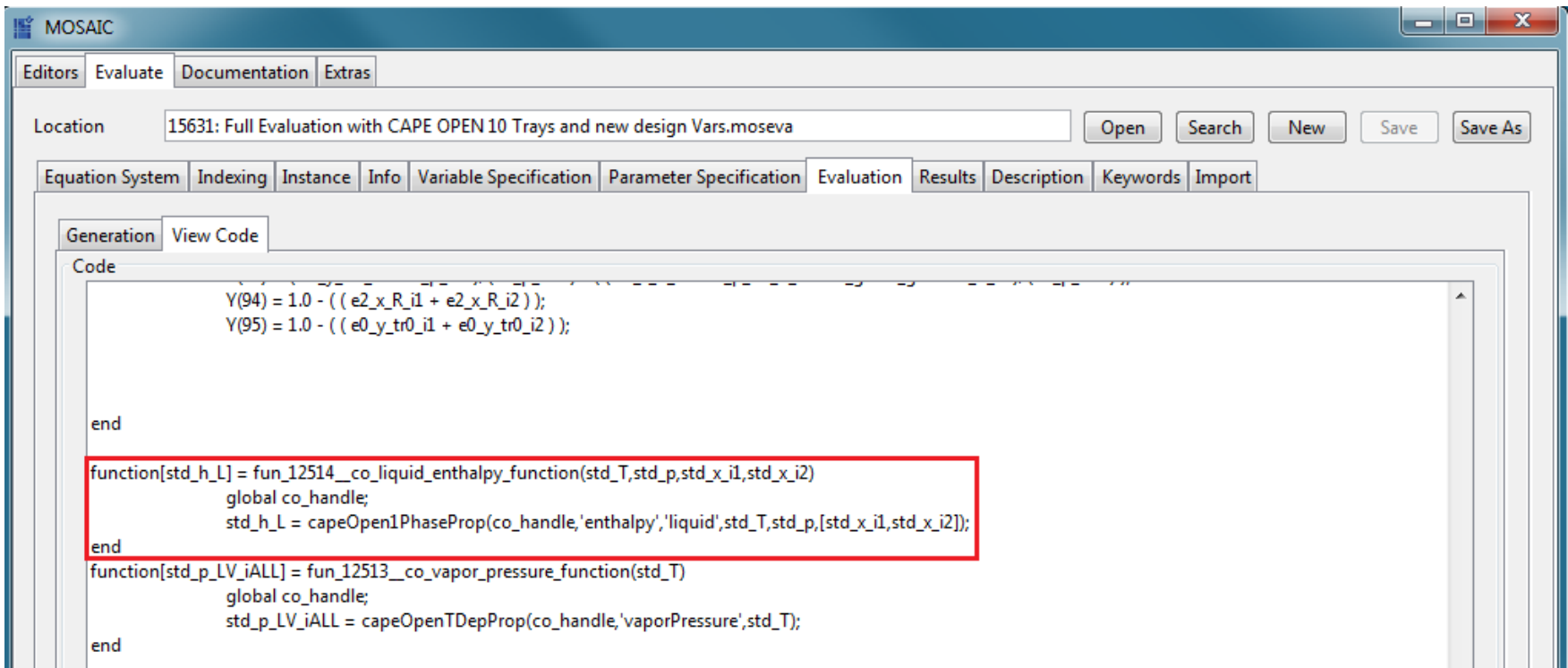
$$h^L(T, p, x_{i=1}, x_{i=2}) = \text{CO Calculate Molar Liquid Enthalpy}$$

The description is 'Liquid enthalpy with CAPE OPEN' and the number of usages is 13. The 'Function Applications' table is also visible, showing the output variable h^L and input variables p and T .

Output Variable		Input Variables	
Generic	Applied	Generic	Applied
h^L	$h^{f,L,u}$	p	p^f
		T	T^f

CAPE-OPEN and MOSAIC I – Physical Properties

MOSAIC Example – Matlab code:



The screenshot shows the MOSAIC software interface. The 'Evaluation' tab is selected, and the 'Code' window displays the following MATLAB code:

```

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
    
```

The function definitions for liquid enthalpy and vapor pressure are highlighted with a red box.



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)



Thank you very much for your kind attention.



www.mosaic-modeling.de

Acknowledgement:



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.