






## MOSAIC – A modeling and code generation tool

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

---


Presentation of „MOSAIC“ for the CAPE-OPEN Annual Meeting 2013. This slides from the TU Berlin (Faculty of Process Sciences) were presented by Gregor Tolksdorf, M.Sc.

The presentation focuses on the modeling and code generation aspects of MOSAIC, so the title is: „MOSAIC – A modeling and code generation tool“



[www.mosaic-modeling.de](http://www.mosaic-modeling.de)

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
2



[www.dbta.tu-berlin.de](http://www.dbta.tu-berlin.de)

MOSAIC emphasises the modeling on the documentation level, i.e. the model of a process is created by defining the descriptions of all model elements. This approach has been implemented using web based technologies.

Additional information, account requests and tutorials can be accessed using our main MOSAIC website „[www.mosaic-modeling.de](http://www.mosaic-modeling.de)“.



**MOSAIC**  
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

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
3

[www.dbta.tu-berlin.de](http://www.dbta.tu-berlin.de)

The development is supervised by Prof Günter Wozny who is the head of the chair of Process Dynamics and Operation (in german: „Fachgebiet für Dynamik und Betrieb technischer Anlagen“, dbta) at the TU Berlin. Currently four developers are working on MOSAIC improvements.



## Outline

**MOSAIC-Modeling**

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

**CAPE-OPEN and MOSAIC**

- Physical Properties
- Unit Operations

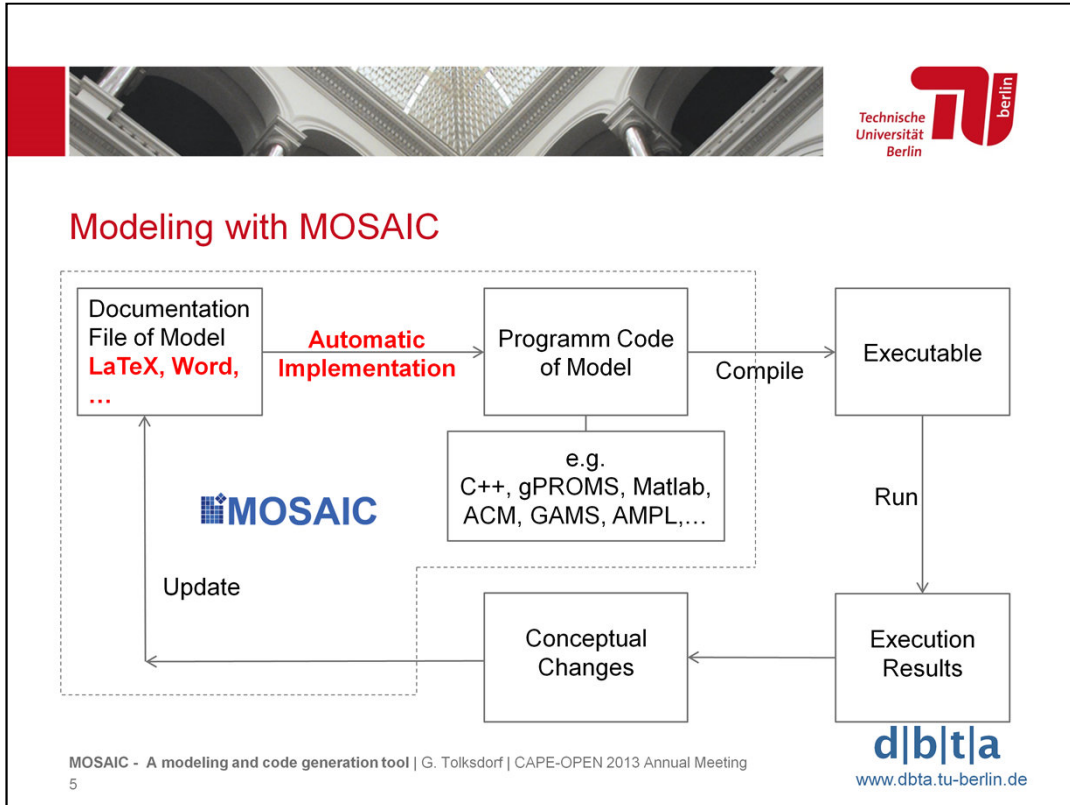
MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
4



[www.dbta.tu-berlin.de](http://www.dbta.tu-berlin.de)

This presentation covers two main topics:

1. The MOSAIC modeling approach containing the modular concept, the symbolic notation, the code generation, and the usage of external ports.
2. Integration of CAPE-OPEN functionality in MOSAIC, especially physical properties and unit operations.





### The idea of modeling with MOSAIC:

When creating a new equation based model, the user has more or less to define the documentation of all parts of the model. This includes mainly the equations and the notation for the used symbols (i.e. variables and parameters) of the model. To enter the symbols and equations LaTeX is preferred, but it is also possible to use e.g. the MS Word formular editor. After specifying the full model (such that the degree of freedom of the equation system is equal to zero) the user can use MOSAIC to generate the program code of the model. As the implementation is done automatically the user won't have to write code manually. MOSAIC supports code generation for different programming languages and target platforms, e.g. C++, gPROMS, and Matlab.

After compilation and running the user gets execution results that usually lead to conceptual changes of the model (e.g. different starting values or new initial values). To apply these changes to the program code the user shouldn't modify the program code manually, because it is an error prone task and will lead to an outdated documentation. When using MOSAIC the user just has to update the documentation of the model and the code once again will be created automatically to avoid errors occuring when coding manually.

MOSAIC is designed to cover the (update of the) platform independent documentation of the model and the generation of platform specific code.

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
6

www.dbta.tu-berlin.de

## 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**


The goals of the MOSAIC modeling approach are

- to avoid errors of manually implemented code,
- reduce the effort to for code generation, and
- facilitate the cooperative work by an improved reusability and portability

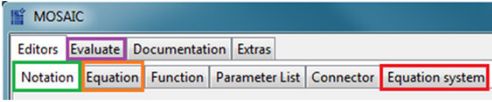
This approach results in

- a highly modular modeling concept
- the usage of an enhanced symbolic notation to define platform independent models (PIM) based on Java and XML in the documentation level
- automatic code generation to convert PIM to platform specific models (PSM)
- the support of web cooperation by storing and sharing all model elements in a web database

On the next slides the modular concept, the symbolic notation, and the code generation will be shown in more detail.




## 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?

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
7




www.dbta.tu-berlin.de

On this slide a part of a screenshot of the MOSAIC graphical user interface (GUI) is shown. The main editors are highlighted by colored squares. Each editor is an own module to create a distinct part of the whole model.

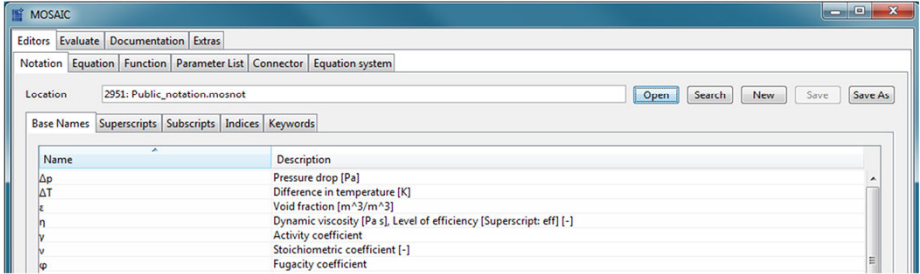
There are four main editors needed to define a fully specified model:

- The „Notation Editor“ defines which symbols / variables are allowed inside the model
- The „Equation Editor“ uses mainly LaTeX to enter each equation that shall be part of the model
- The „Equation system Editor“ combines equations and equation systems to new equation systems
- The „Evaluation Editor“ specifies not only the design, state, and iteration variables but also the start values, and initial values (depending on the type of the equation system, e.g. NLE or ODE). Additionally the code will be created inside this editor.




## Enhanced Symbolic Notation I - Variables

Notation editor:



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



MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
8



[www.dbta.tu-berlin.de](http://www.dbta.tu-berlin.de)

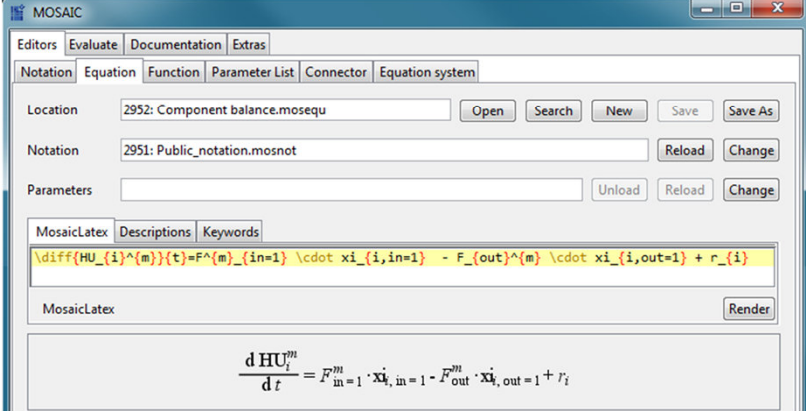
At the bottom of this slide an example for a variable in MOSAIC is shown. The user has to use the „Notation editor“ in order to define what the meaning of this variable is. Therefore all base names, superscripts, subscripts, and indices have to be specified including a human readable description to clarify the meaning of the symbols.




## Enhanced Symbolic Notation II - Equations

Equation editor:




MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting

9



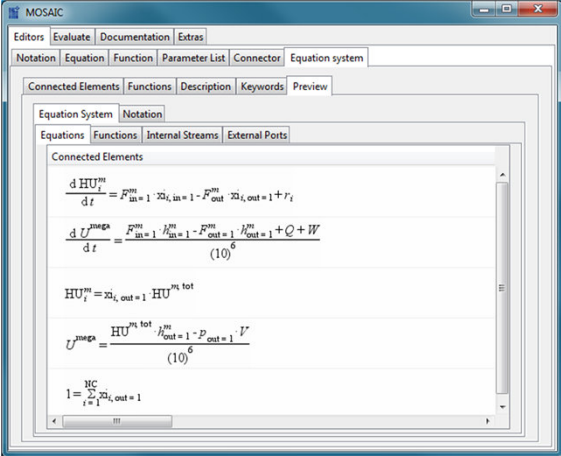
www.dbt.a.tu-berlin.de

Another example for the enhanced symbolic notation is shown on this slide. After entering the LaTeX-Code for the equation inside the „Equation editor“ the user can immediately see what the equation would look like on/in a paper by clicking the „Render“ button.



## Enhanced Symbolic Notation III – Equation Systems

Equation system editor:



The screenshot shows the MOSAIC software interface with the 'Equation System' editor open. The editor displays a list of equations under the 'Connected Elements' tab. The equations are:

$$\frac{d HU_i^m}{dt} = P_{in=1}^m \cdot x_{i, in=1} - P_{out}^m \cdot x_{i, out=1} + r_i$$


$$\frac{d U^{mega}}{dt} = \frac{P_{in=1}^m \cdot h_{in=1}^m - P_{out=1}^m \cdot h_{out=1}^m + Q + W}{(10)^6}$$

$$HU_i^m = x_{i, out=1} \cdot HU^{m, tot}$$

$$U^{mega} = \frac{HU^{m, tot} \cdot h_{out=1}^m - P_{out=1}^m \cdot V}{(10)^6}$$

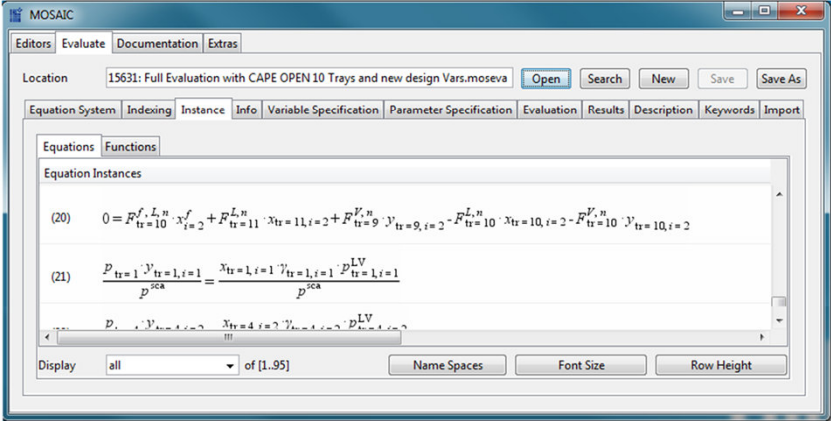
$$1 = \sum_{i=1}^{NC} x_{i, out=1}$$

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
10



[www.dbta.tu-berlin.de](http://www.dbta.tu-berlin.de)

In the „Equation system editor“ all combined equations of an equation system are displayed together.



MOSAIC

Editors Evaluate Documentation Extras

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

Equations Functions

Equation Instances

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

(21) 
$$\frac{P_{tr=1} \cdot y_{tr=1,i=1}}{P^{SCA}} = \frac{x_{tr=1,i=1} \cdot y_{tr=1,i=1} \cdot P_{tr=1,i=1}^{LV}}{P^{SCA}}$$

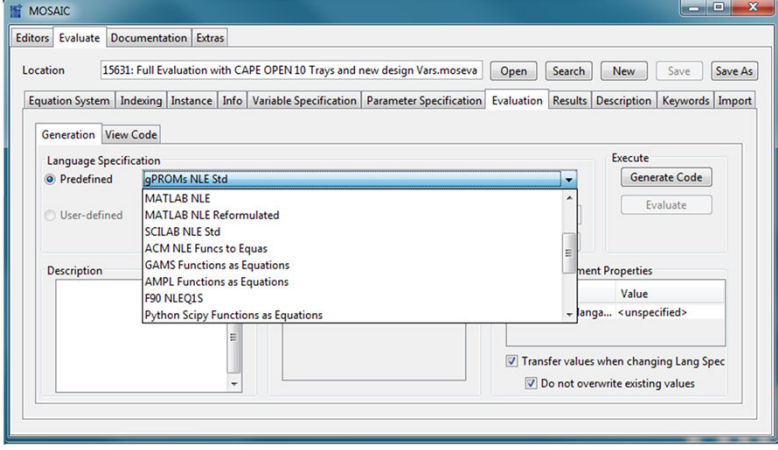
Display all of [1..95] Name Spaces Font Size Row Height

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
11

Technische Universität Berlin

d|b|t|a  
www.dbta.tu-berlin.de

In the „Evaluation editor“ MOSAIC provides a panel to take a look at all instantiated equations. In contrast to generic equations the instantiated equations use indices with specific values, e.g. generic:  $x_{\{i\}}$  vs. instantiated:  $x_{\{i=1\}}$ ,  $x_{\{i=2\}}$ .



The screenshot shows the MOSAIC software interface. At the top, there is a menu bar with 'Editors', 'Evaluate', 'Documentation', and 'Extras'. Below the menu bar is a toolbar with buttons for 'Open', 'Search', 'New', 'Save', and 'Save As'. The main window is titled 'MOSAIC' and contains several tabs: 'Equation System', 'Indexing', 'Instance', 'Info', 'Variable Specification', 'Parameter Specification', 'Evaluation', 'Results', 'Description', 'Keywords', and 'Import'. The 'Generation' tab is active, showing a 'Language Specification' section with a list of predefined languages: gPROMs NLE Std (selected), MATLAB NLE, MATLAB NLE Reformulated, SCILAB NLE Std, ACM NLE Funcs to Equas, GAMS Functions as Equations, AMPL Functions as Equations, F90 NLEQIS, and Python Scipy Functions as Equations. To the right of the list is an 'Execute' section with 'Generate Code' and 'Evaluate' buttons. Below the list is a 'Description' field and a 'Comment Properties' section with a 'Value' field and a dropdown menu. At the bottom right of the window are two checked checkboxes: 'Transfer values when changing Lang Spec' and 'Do not overwrite existing values'.



MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
12

Technische Universität Berlin

dbt|a  
www.dbta.tu-berlin.de

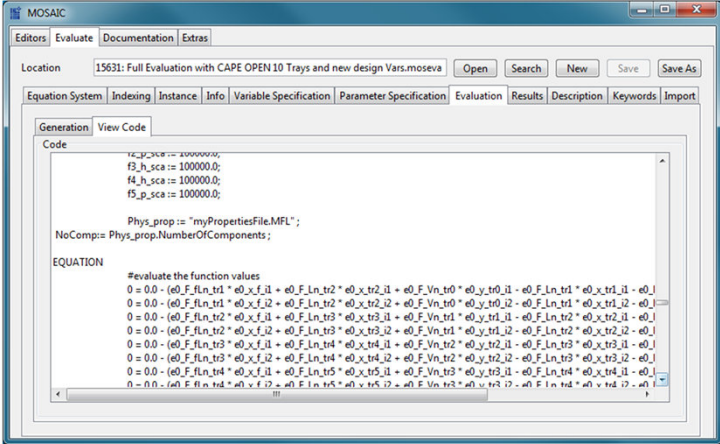
In order to generate the programming code for the model the user has to choose one of the programming languages provided by MOSAIC. The screenshot shows an example for an algebraic equation system. In this case MOSAIC supports e.g. gPROMS, Matlab, Scilab, ACM, GAMS, AMPL, Fortran and Python.

After selecting the preferred language/target platform the code will be created by clicking on the „Generate Code“ button in the MOSAIC GUI.

## Code Generation II – Show me what you got!

Take a look at the generated code:



The screenshot shows the MOSAIC software interface. The 'View Code' panel displays the following generated code:

```


1432308 := 2300000.0;
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_fl_n_tr1 * e0_x_f_11 + e0_F_l_n_tr2 * e0_x_tr2_11 + e0_F_Vn_tr0 * e0_y_tr0_11 - e0_F_l_n_tr1 * e0_x_tr1_11 - e0_l
0 = 0.0 - (e0_F_fl_n_tr1 * e0_x_f_12 + e0_F_l_n_tr2 * e0_x_tr2_12 + e0_F_Vn_tr0 * e0_y_tr0_12 - e0_F_l_n_tr1 * e0_x_tr1_12 - e0_l
0 = 0.0 - (e0_F_fl_n_tr2 * e0_x_f_11 + e0_F_l_n_tr3 * e0_x_tr3_11 + e0_F_Vn_tr1 * e0_y_tr1_11 - e0_F_l_n_tr2 * e0_x_tr2_11 - e0_l
0 = 0.0 - (e0_F_fl_n_tr2 * e0_x_f_12 + e0_F_l_n_tr3 * e0_x_tr3_12 + e0_F_Vn_tr1 * e0_y_tr1_12 - e0_F_l_n_tr2 * e0_x_tr2_12 - e0_l
0 = 0.0 - (e0_F_fl_n_tr3 * e0_x_f_11 + e0_F_l_n_tr4 * e0_x_tr4_11 + e0_F_Vn_tr2 * e0_y_tr2_11 - e0_F_l_n_tr3 * e0_x_tr3_11 - e0_l
0 = 0.0 - (e0_F_fl_n_tr3 * e0_x_f_12 + e0_F_l_n_tr4 * e0_x_tr4_12 + e0_F_Vn_tr2 * e0_y_tr2_12 - e0_F_l_n_tr3 * e0_x_tr3_12 - e0_l
0 = 0.0 - (e0_F_fl_n_tr4 * e0_x_f_11 + e0_F_l_n_tr5 * e0_x_tr5_11 + e0_F_Vn_tr3 * e0_y_tr3_11 - e0_F_l_n_tr4 * e0_x_tr4_11 - e0_l
0 = 0.0 - (e0_F_fl_n_tr4 * e0_x_f_12 + e0_F_l_n_tr5 * e0_x_tr5_12 + e0_F_Vn_tr3 * e0_y_tr3_12 - e0_F_l_n_tr4 * e0_x_tr4_12 - e0_l

```

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
13

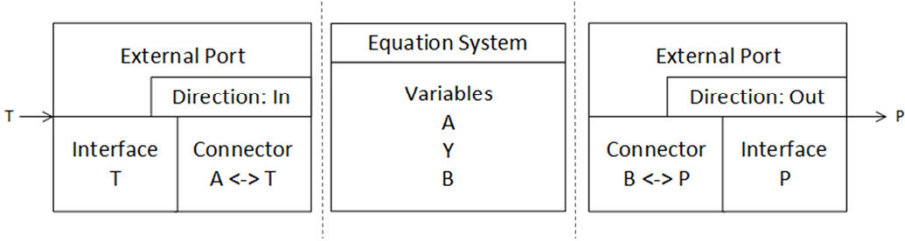


The „View Code“ panel of the „Evaluation editor“ is used to take a look at the generated code. By selecting, copying and pasting the code into the respective local environment (e.g. gPROMS, Matlab) the model can be evaluated. Of course the user has to have an installation of the preferred target platform as MOSAIC can not provide the specific software and it will otherwise not be possible to execute the code.




## 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 - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
 14



www.dbta.tu-berlin.de

MOSAIC can use the concept of „Ports“. Ports can be used to encapsulate an equation system meant to be a unit operation such that only specific variables are visible from the outside of this equation system. In the example shown on this slide the equation system contains three variables. The variable „Y“ is only used internally, whereas the variable „A“ is connected to the input „T“ and the variable „B“ is connected to the output „P“. From outside of the system boundaries the unit is a black box with in input „T“ going in and an output „P“ coming out. The variable „Y“ is hidden and can be directly accessed from inside the equation system describing the unit.




## 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

External Port		Equation System	External Port	
Direction: In		Variables	Direction: Out	
Interface	Connector	A	Connector	Interface
T	A <-> T	Y	B <-> P	P
		B		

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
15





www.dbta.tu-berlin.de

To specify a MOSAIC external Port three attributes have to be set:

1. Each port has to have a direction, i.e. either „In“ or „Out“.
2. Each port needs an interface describing the type of the variable presented to the outside. The example on the slide shows the mandatory fields „naming“, „dimension“, „engineering unit“, and „direction“.
3. Each port uses a connector to define the connection between an internal variable and an external interface variable. In the example shown on this slide the two connectors link „A“ with „T“ (input) and „B“ with „P“ (output).

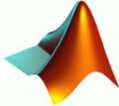

The concept of „Ports“ should be a basis to create or import CAPE-OPEN unit operations with MOSAIC.


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

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
16



[www.dbta.tu-berlin.de](http://www.dbta.tu-berlin.de)

MOSAIC supports code generation with CAPE\_OPEN physical properties for two target platforms so far:


Matlab (NLE,ODE,DAE) and gPROMS (NLE).

NLE – Non Linear Equation system

ODE – Ordinary Differential Equation system

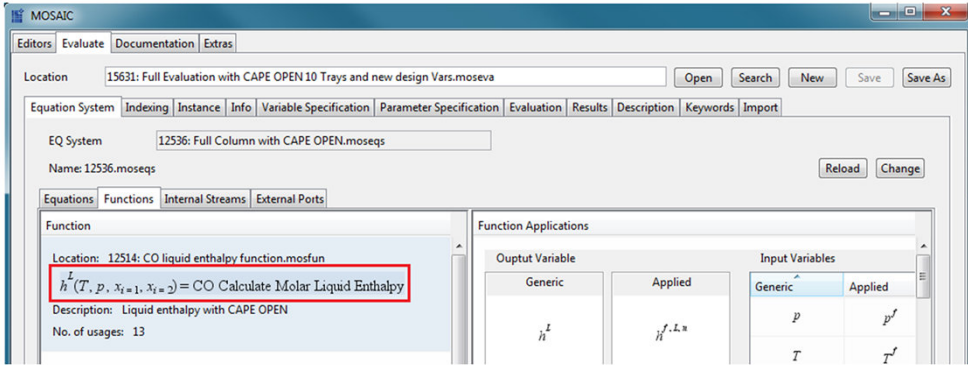
DAE – Differential Algebraic Equation system





## CAPE-OPEN and MOSAIC I – Physical Properties

MOSAIC Example – CO function:



The screenshot shows the MOSAIC software interface. The 'Function' tab is active, displaying the following information:

- Location: 12514: CO liquid enthalpy function.mosfun
- Function:  $h^L(T, p, \gamma_1, \gamma_2) = \text{CO Calculate Molar Liquid Enthalpy}$  (highlighted with a red box)
- Description: Liquid enthalpy with CAPE-OPEN
- No. of usages: 13


The 'Function Applications' section shows a table of variables:

Output Variable		Input Variables	
Generic	Applied	Generic	Applied
$h^L$		$p$	$p^f$
		$T$	$T^f$

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
17

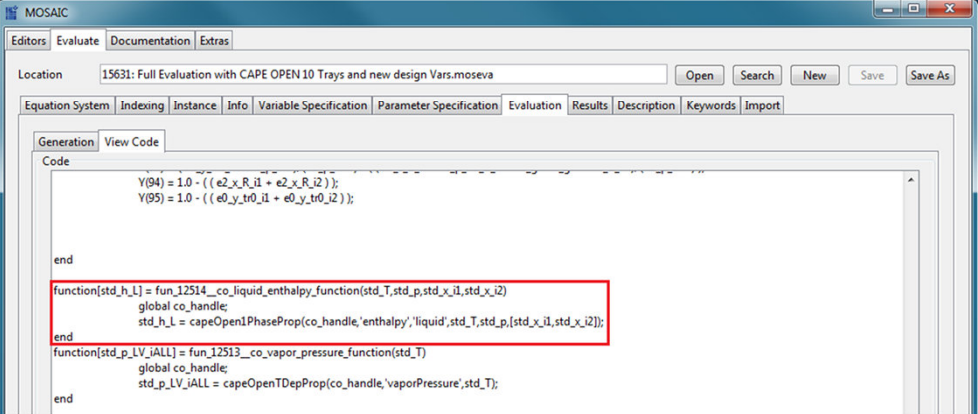
**d|b|t|a**  
www.dbta.tu-berlin.de

In order to use the CAPE-OPEN physical properties in MOSAIC the user has to create a MOSAIC function that implements the CO interface. In the example on this slide a function for liquid molar enthalpy is used inside an equation system.



## CAPE-OPEN and MOSAIC I – Physical Properties

MOSAIC Example – Matlab code:



```

Y(B4) = 1.0 * (( e2_x_R_i1 + e2_x_R_i2 ));
Y(B5) = 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 = capeOpenIPhaseProp(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

```

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
18



www.dbta.tu-berlin.de

In the generated code (in this case for Matlab) the MOSAIC function is converted into a platform specific function that uses the predefined property manager and property packages to calculate the physical property via the CAPE-OPEN interface.




## Summary



- 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)

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
19



[www.dbta.tu-berlin.de](http://www.dbta.tu-berlin.de)

Summarizing what has been mentioned so far in this presentation it can be said that MOSAIC

- Is a modular equation based modeling tool
- Is implemented in Java using XML/MathML to store the models
- Provides automatic code generation for specific platforms (C++, Fortran, Matlab, gPROMS, ACM, ...)
- Can use the „Ports“ concept
- Supports CO physical properties in code generation for Matlab and gPROMS

Additionally it is important to state that MOSAIC is neither a programming language nor a computer algebra system (CAS) . As MOSAIC is not designed to be a process simulator the term „Process Modeling Environment“ in the sense of CAPE-OPEN is not the appropriate notion. Referring to the title of this presentation: MOSAIC is modeling and code generation tool.



## 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?
- ...

Regarding the next steps of MOSAIC to include CAPE-OPEN unit operations many questions occur. Some of them are listed on this slide.





Thank you very much for your kind attention.



[www.mosaic-modeling.de](http://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.

MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting  
21


  
[www.dbta.tu-berlin.de](http://www.dbta.tu-berlin.de)

Additional information, account requests and tutorials can be accessed using our main MOSAIC website „[www.mosaic-modeling.de](http://www.mosaic-modeling.de)“.

The MOSAIC project is supported by the Cluster of Excellence 'Unifying Concepts in Catalysis' (unicat), coordinated by the Technical University of Berlin, and funded by the German Research Foundation.