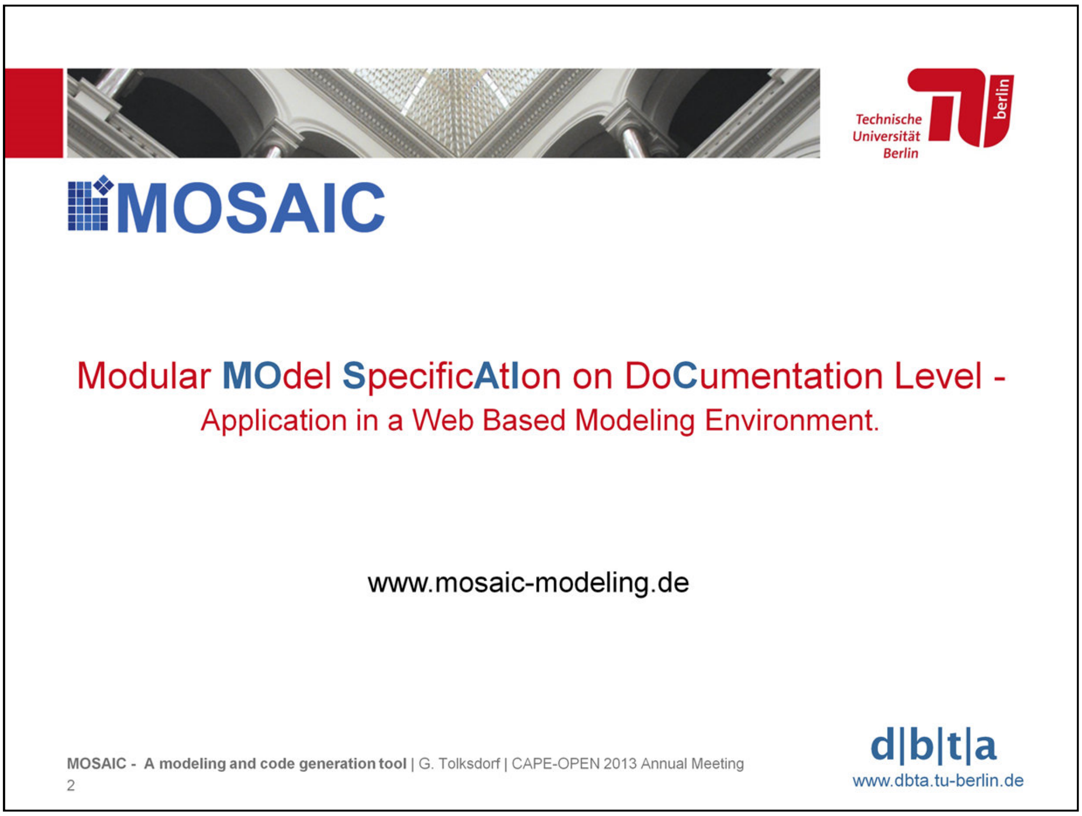
MOSAIC – A modeling and code generation tool

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“



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“.

Technische Universität Berlin

MOSAIC

MOSAIC-Team

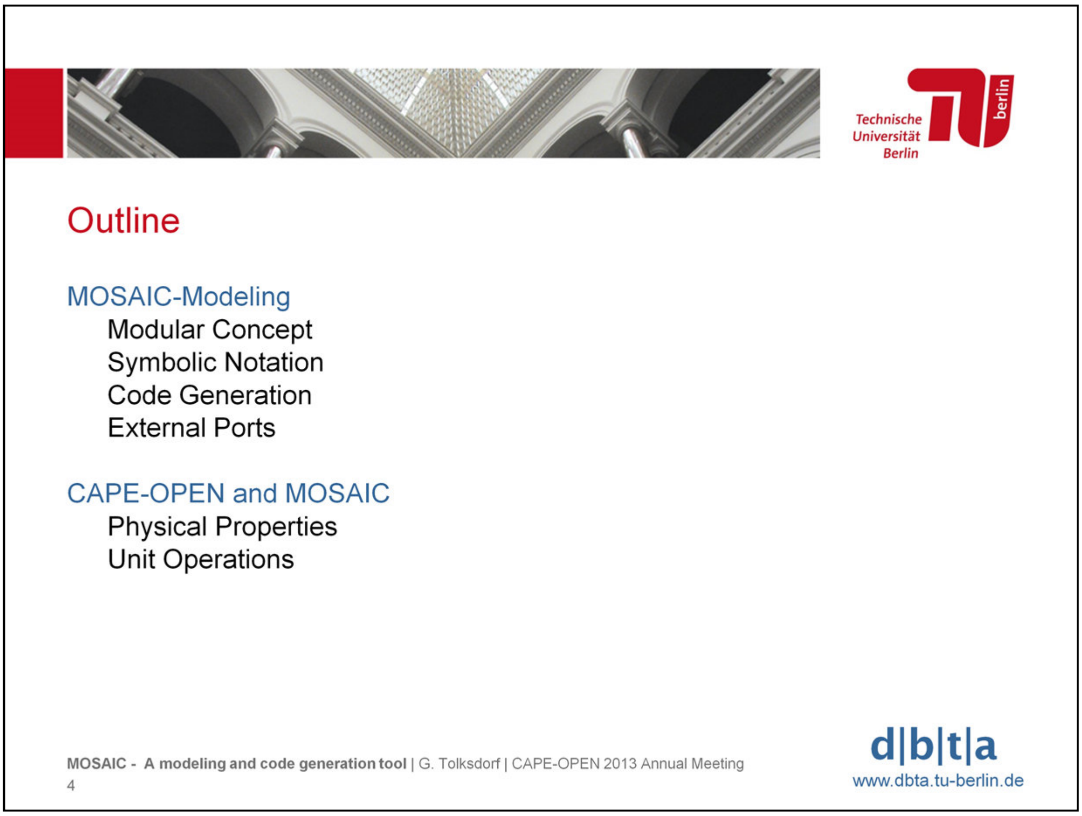
Scientific supervisor:
- Prof. Dr.-Ing. G. Wozny ,
  Head of Chair of Process Dynamics and Operation, TU Berlin

d|b|t|a

Current Developer:
- Dipl.-Ing. S. Fillinger
- Dipl.-Ing. R. Kraus
- Dipl.-Ing. V.A. Merchan-Restrepo
- M.Sc. G. Tolksdorf

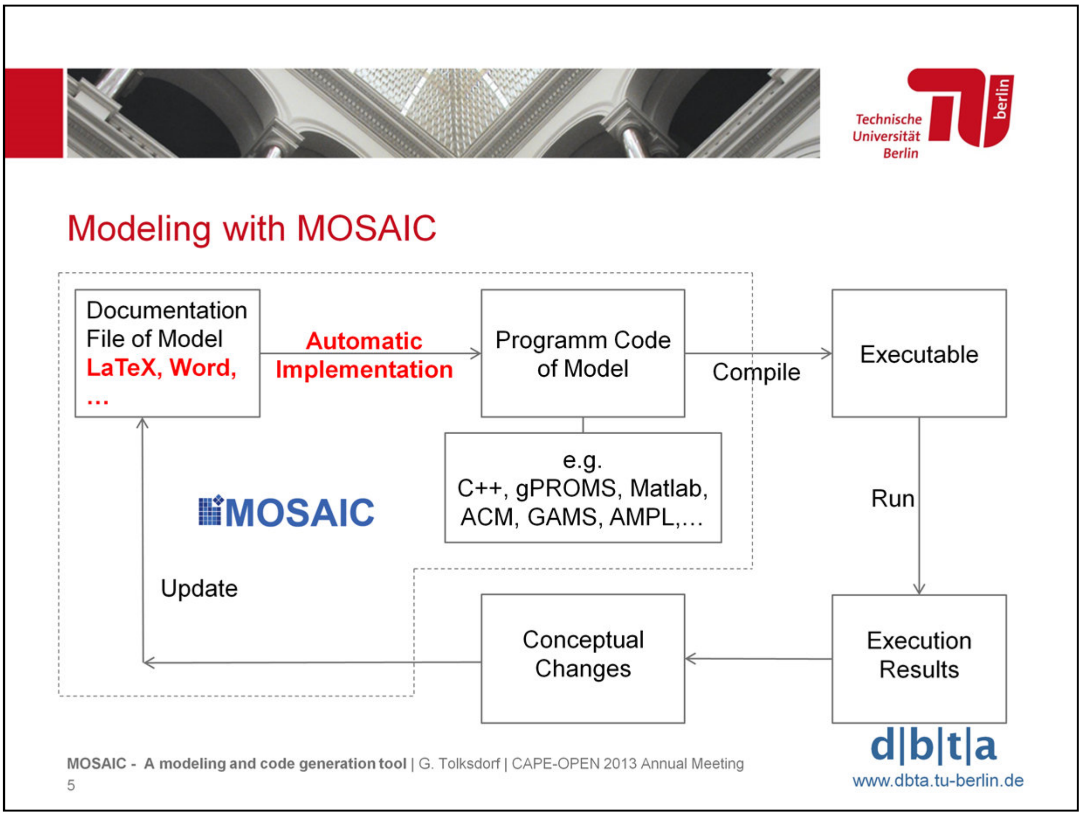
d|b|t|a
www.dbta.tu-berlin.de

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 inprovements.



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.

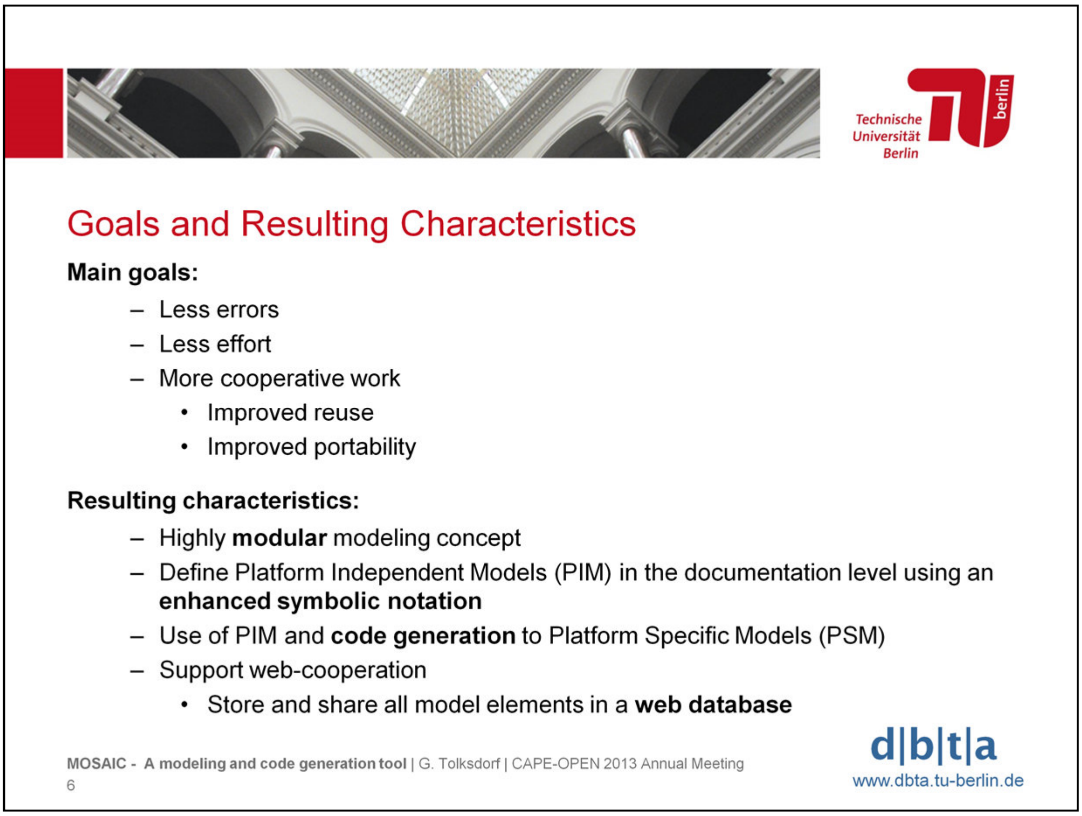


The idea of modeling with MOSAIC:

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 intial 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.



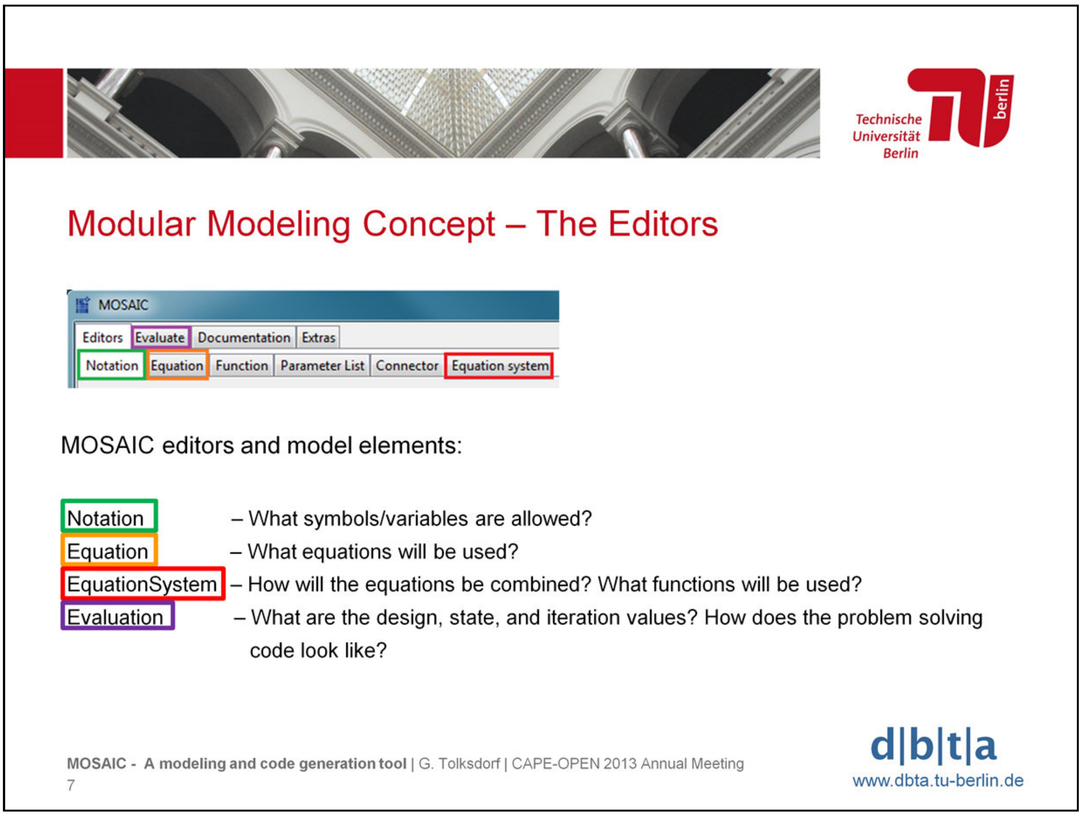
The goals of the MOSAIC modeling approach are

- 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.



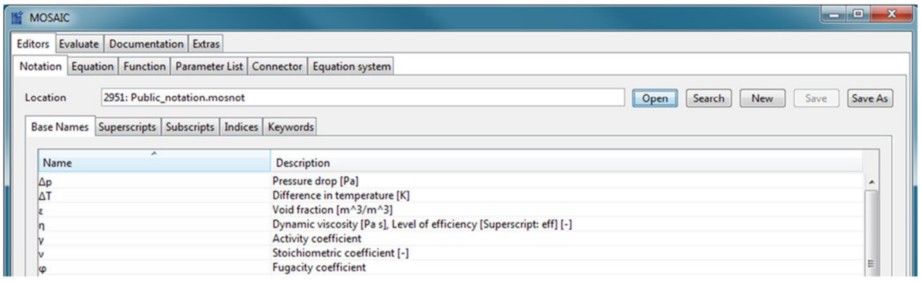
On this slide a part of a screenshot of the MOSAIC graphical user interface (GUI) is shown. The main editors are highlighted by colored squares. Each editor is an own module to create a distinct part of the whole model.

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:

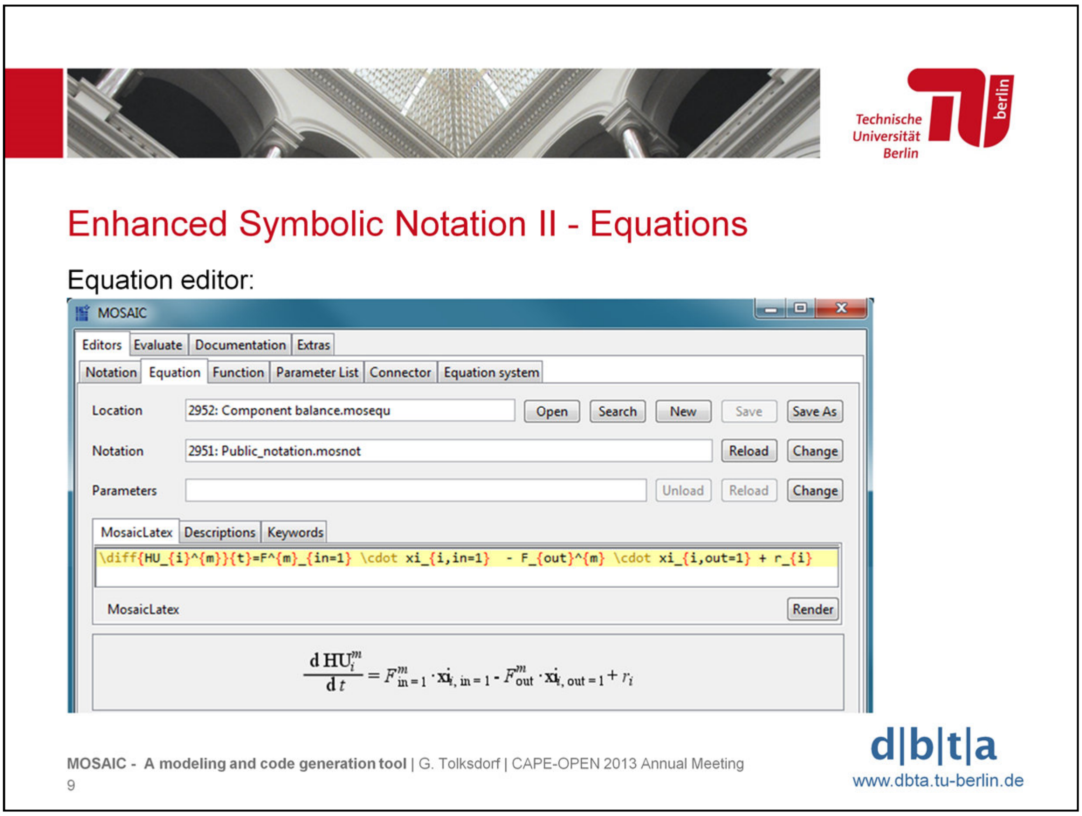


Example: $p_{o,i=2,j=4}^{LV,I}$

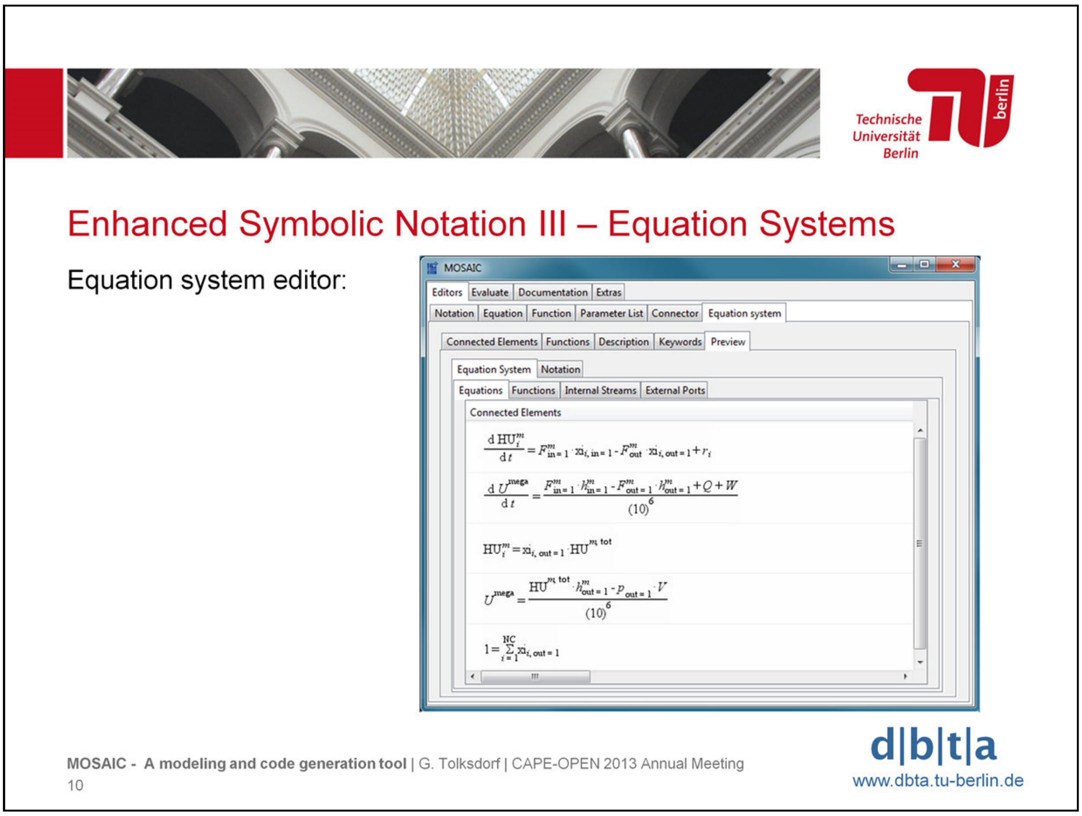
MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting
8

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

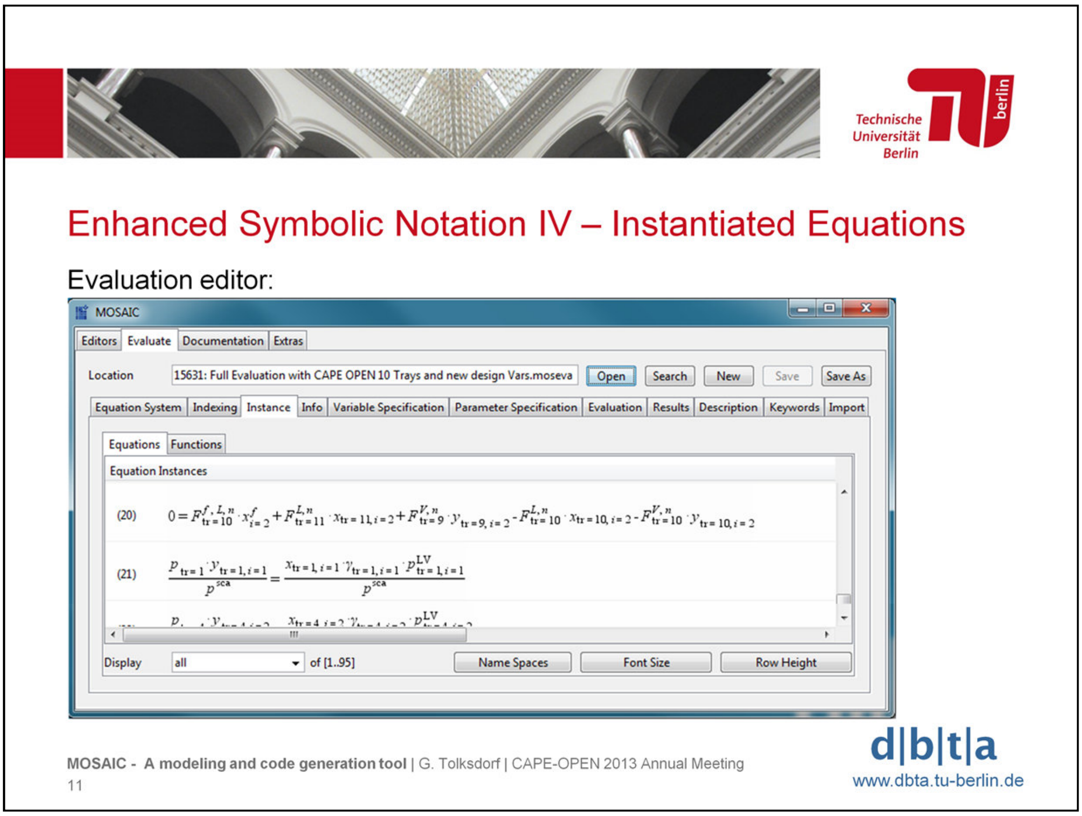
At the bottom of this slide an example for a variable in MOSAIC is shown. The user has to use the „Notation editor“ in order to define what the meaning of this variable is. Therefore all base names, superscripts, subscripts, and indices have to be specified including a human readable desciption to clearify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



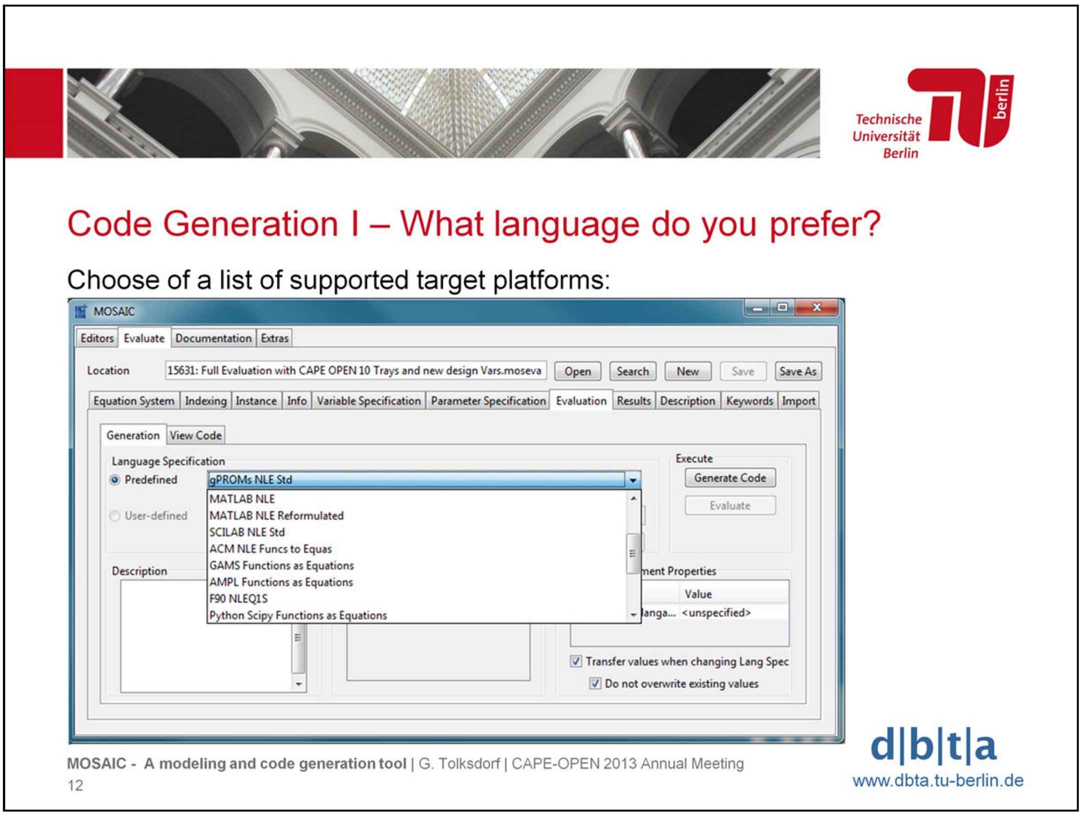
Another example for the enhanced symbolic notation is shown on this slide. After entering the LaTeX-Code for the equation inside the „Equation editor“ the user can immediately see what the equation would look like on/in a paper by clicking the „Render“ button.



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

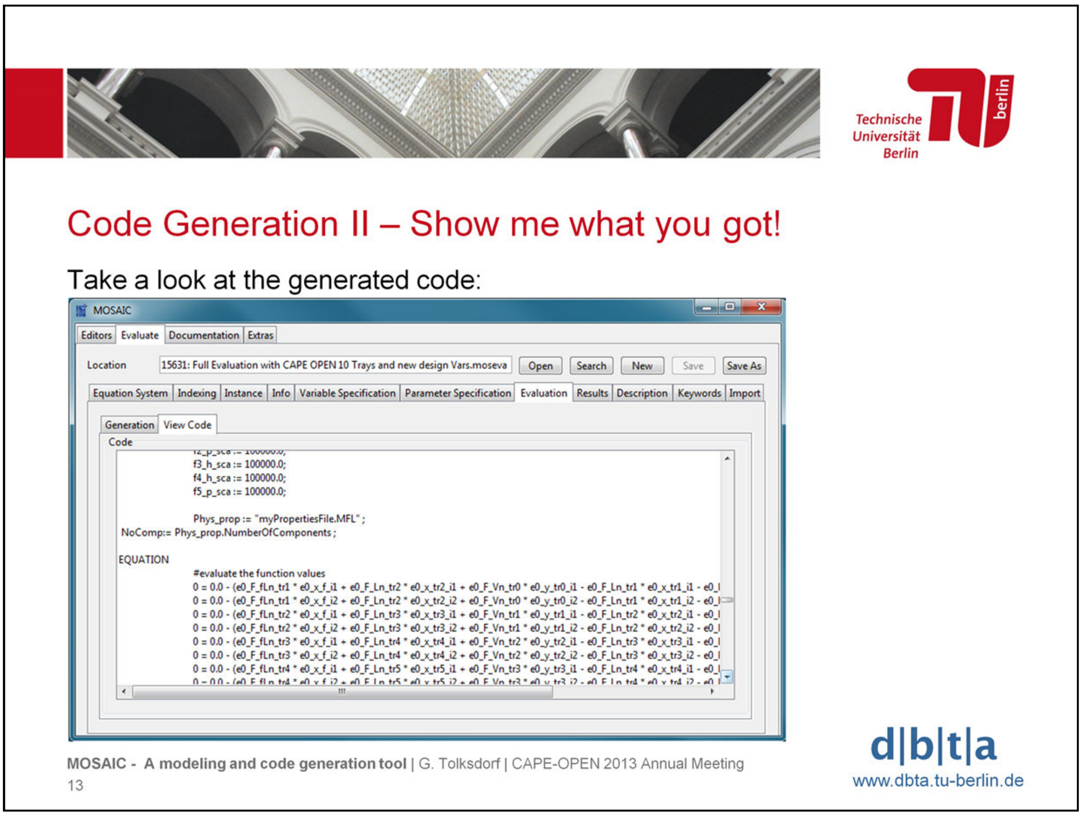


In the „Evaluation editor“ MOSAIC provides a panel to take a look at all instantiated equations. In contrast to generic equations the instantiated equations use indices with specific values, e.g. generic: x_{i} vs. instantiated: x_{i=1}, x_{i=2} .

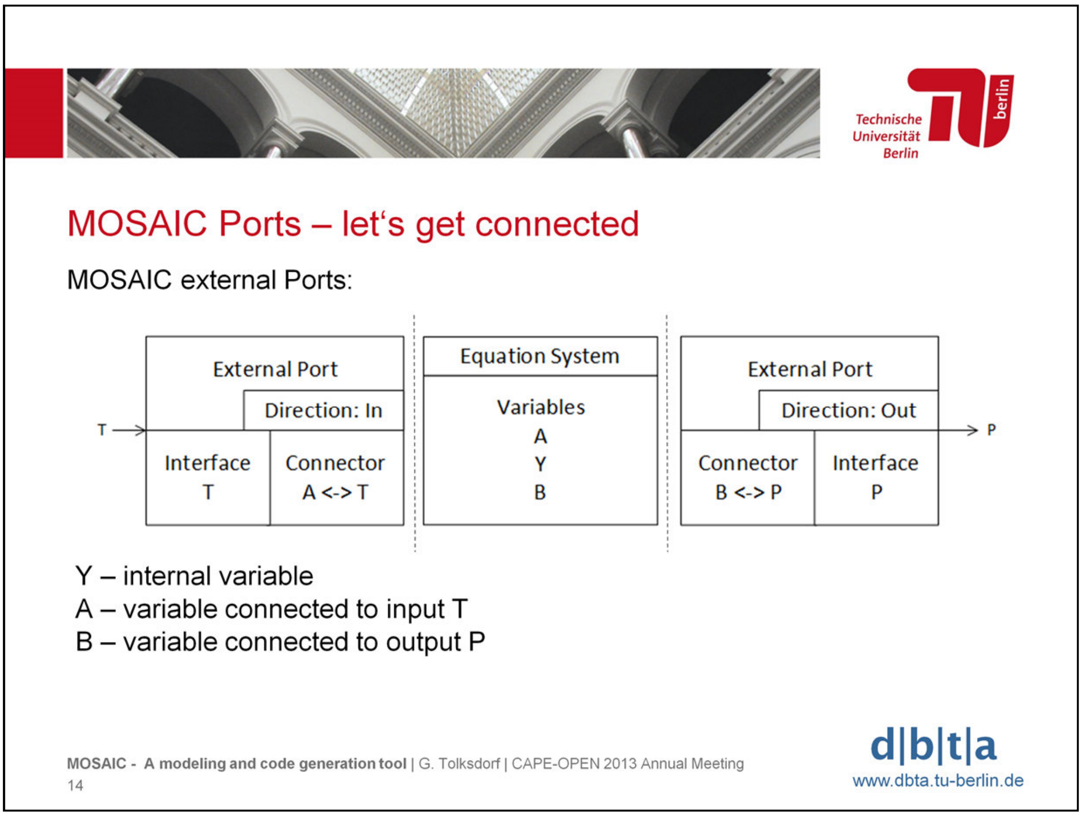


In order to generate the programming code for the model the user has to chose one of the programming languages provided by MOSAIC. The screenshot shows an example for an algebraic equation system. In this case MOSAIC supports e.g. gPROMS, Matlab, Scilab, ACM, GAMS, AMPL, Fortran and Python.

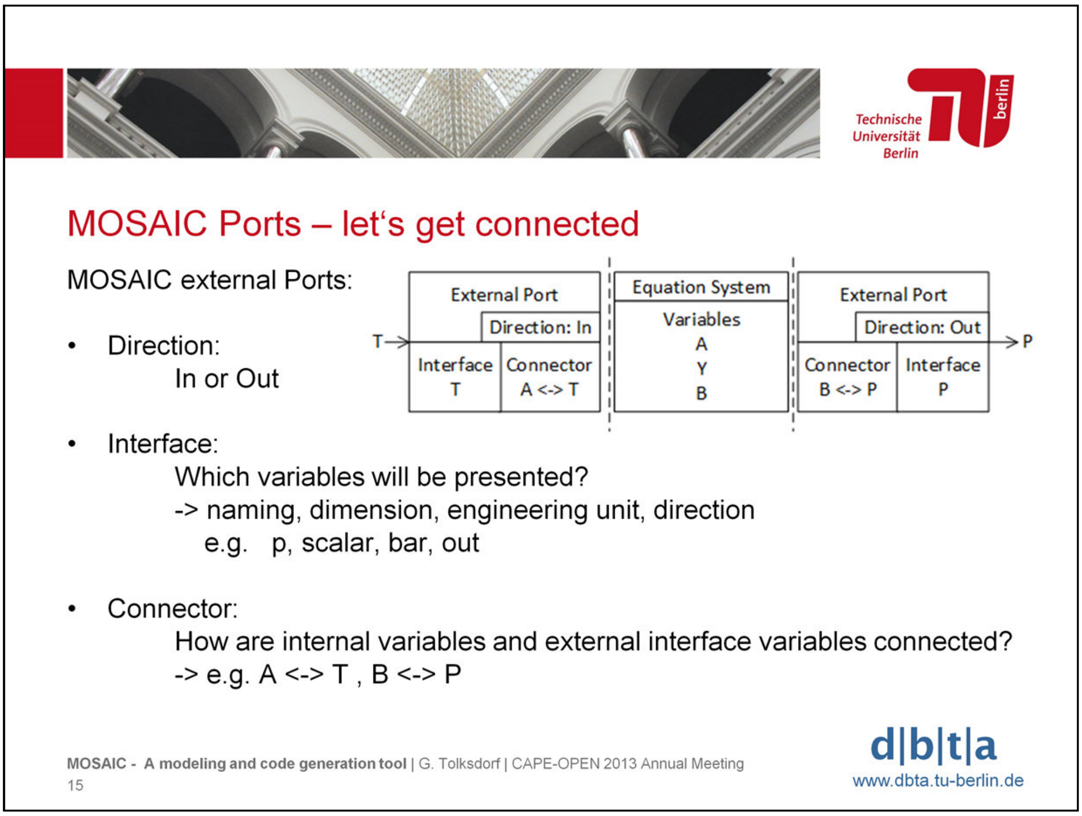
After selecting the preferred language/target platform the code will be created by clicking on the „Generate Code“ button in the MOSAIC GUI.



The „View Code“ panel of the „Evaluation editor“ is used to take a llok 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 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 ouput „P“ coming out. The variable „Y“ is hidden and can be directly accessed from inside the equation system describing the unit.



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.

Technische Universität Berlin

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

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

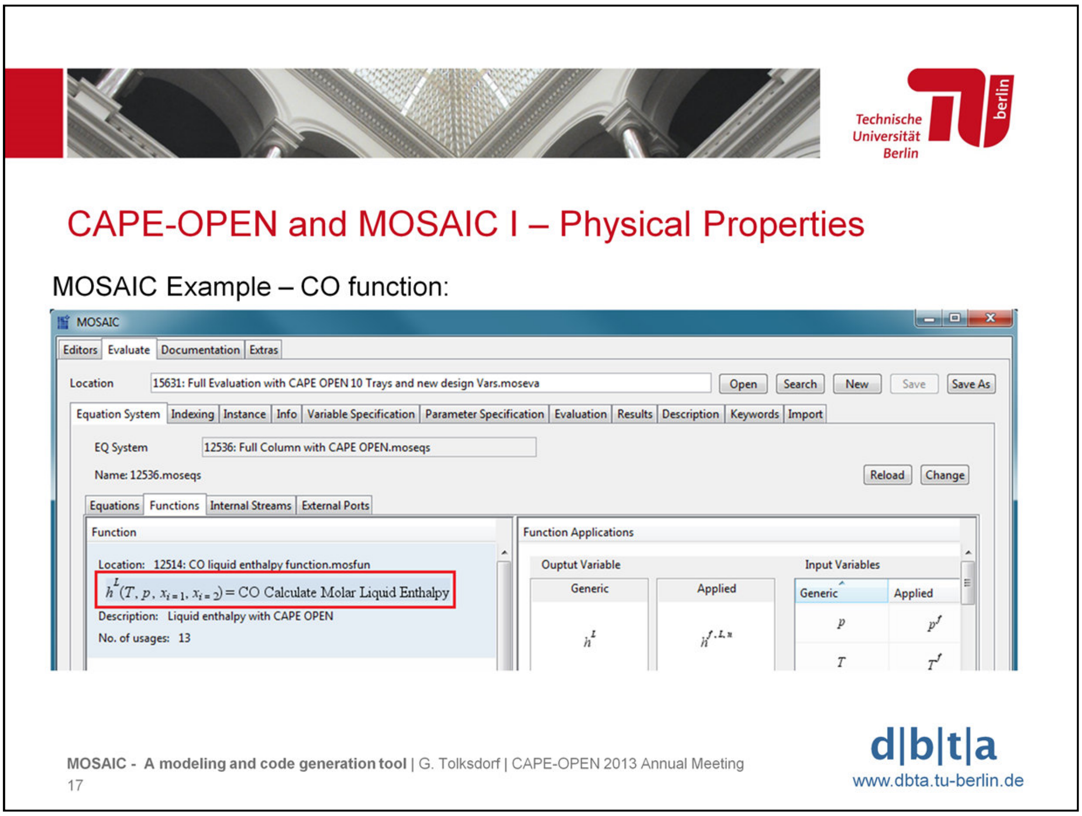
MOSAIC supports code generation with CAPE_OPEN physical properties for two target platforms so far:

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

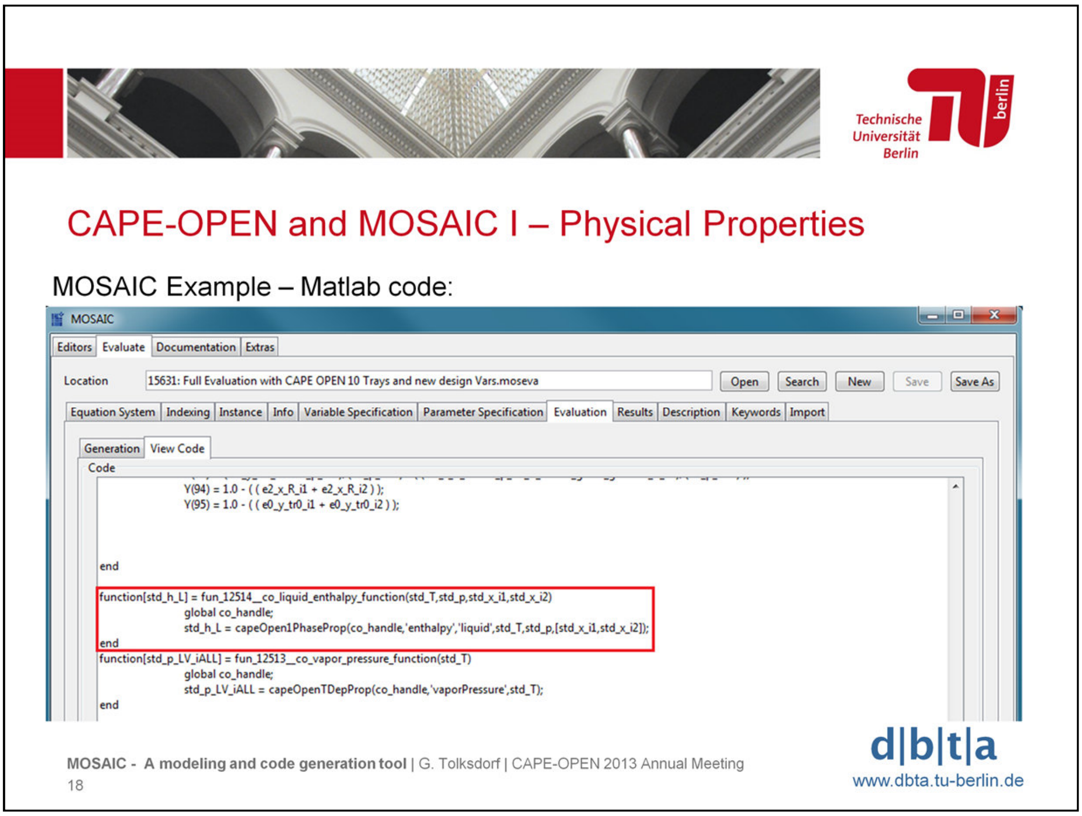
NLE – Non Linear Equation system

ODE – Ordinary Differential Equation system

DAE – Differential Algebraic Equation system



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.



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 calculat the physical property via the CAPE-OPEN interface.

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.

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

Additional information, account requests and tutorials can be accessed using our main MOSAIC website „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.