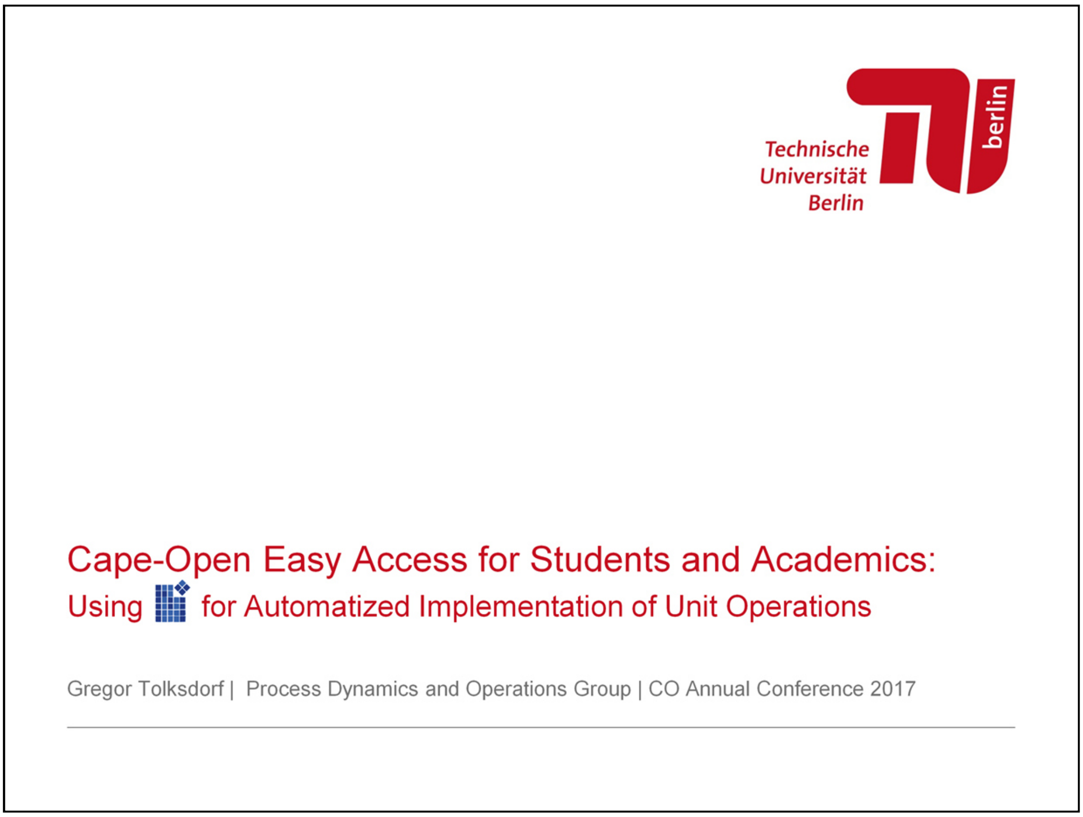


Technische
Universität
Berlin

Cape-Open Easy Access for Students and Academics:

Using for Automatized Implementation of Unit Operations

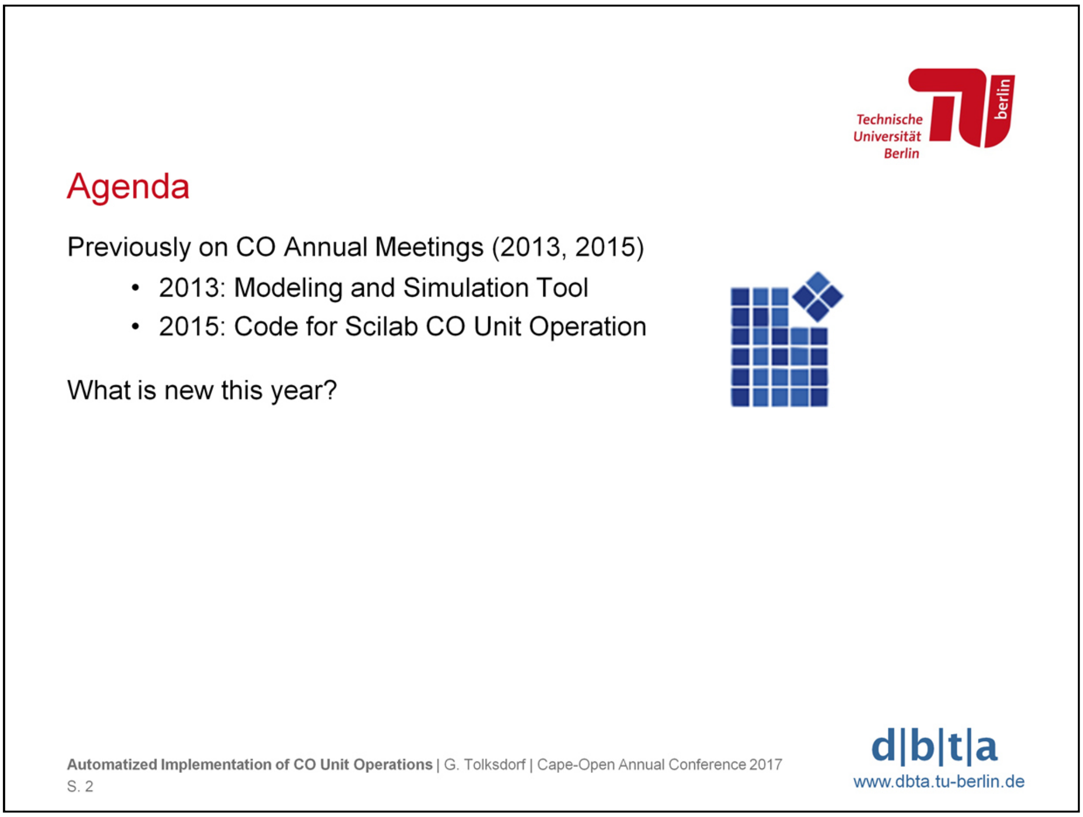
Gregor Tolksdorf | Process Dynamics and Operations Group | CO Annual Conference 2017

Presentation of „MOSAICmodeling“ for the CAPE-OPEN Annual Meeting 2017.

These slides from Technische Universität Berlin were presented by Gregor Tolksdorf, M.Sc.

Since 2013, Gregor works as a research assistant at the Process Dynamics and Operations Group in the Faculty of Process Sciences. He is lead developer of MOSAICmodeling, a modular tool for modelling and code generation, supporting simulation and optimization, based on platform-independent technologies (XML, MathML, MySQL, and Java).

Gregor regularly is participant of Cape-Open Annual Meetings and he previously gave two presentations: in Lyon (2013) and Amsterdam (2015).



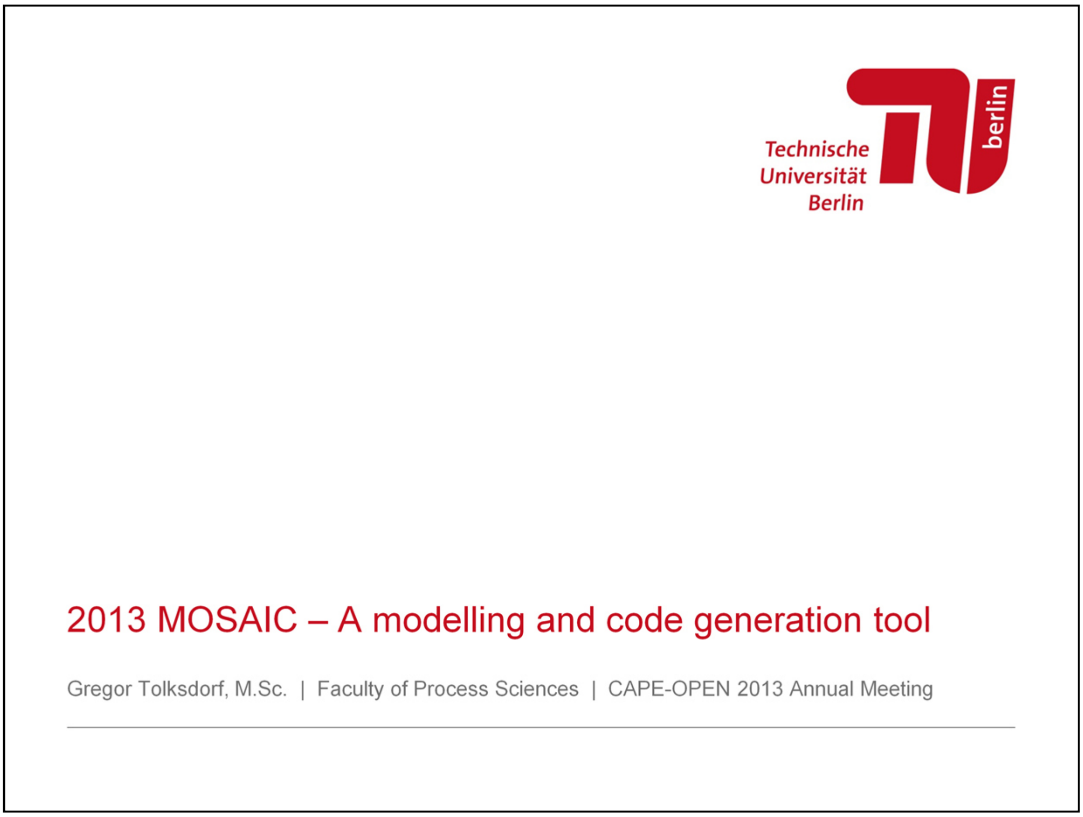
Agenda

Previously on CO Annual Meetings (2013, 2015)

- 2013: Modeling and Simulation Tool
- 2015: Code for Scilab CO Unit Operation

What is new this year?

This presentation gives a short review of the previous MOSAICmodeling talks at Cape-Open Annual Meetings and adds new developments regarding the export of ready-to-use Cape-Open Unit Operations.



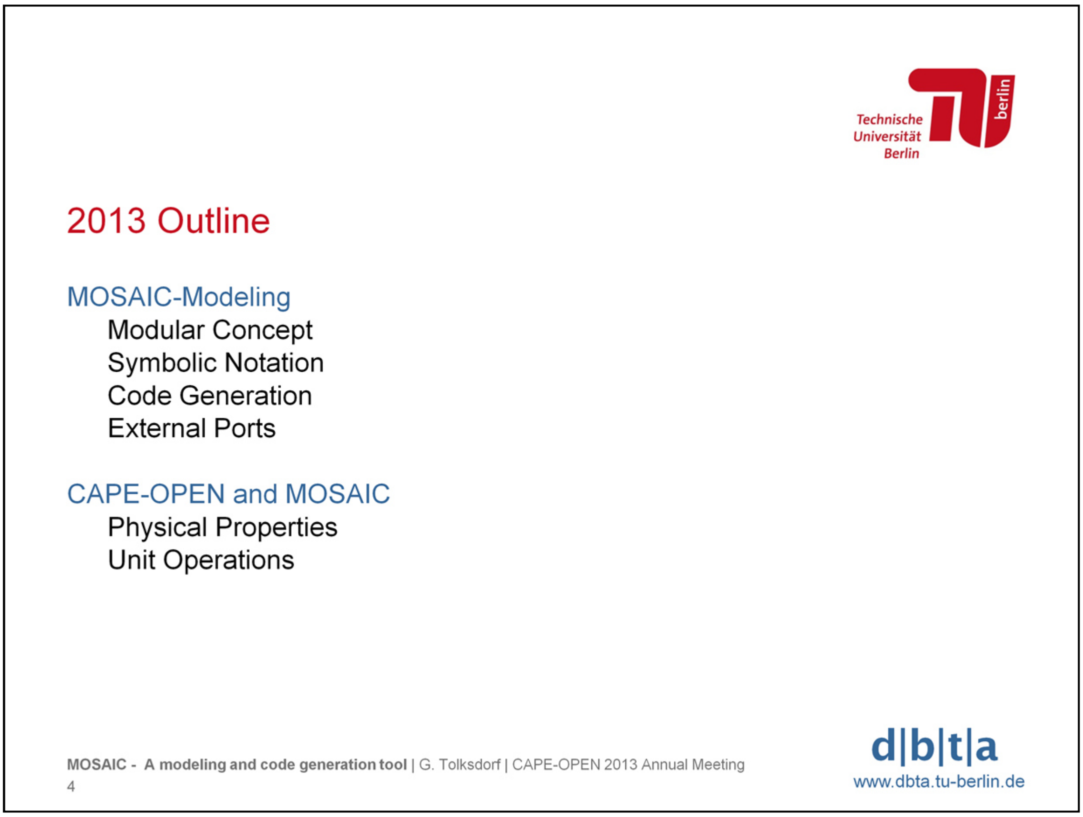
Technische Universität Berlin

2013 MOSAIC – A modelling and code generation tool

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

The presentation of „MOSAIC“ for the CAPE-OPEN Annual Meeting in 2013. These slides were presented by Gregor Tolksdorf, M.Sc.

The focus of this talk is on modelling and code generation aspects of MOSAIC, so the title is: „MOSAIC – A modelling and code generation tool“



2013 Outline

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

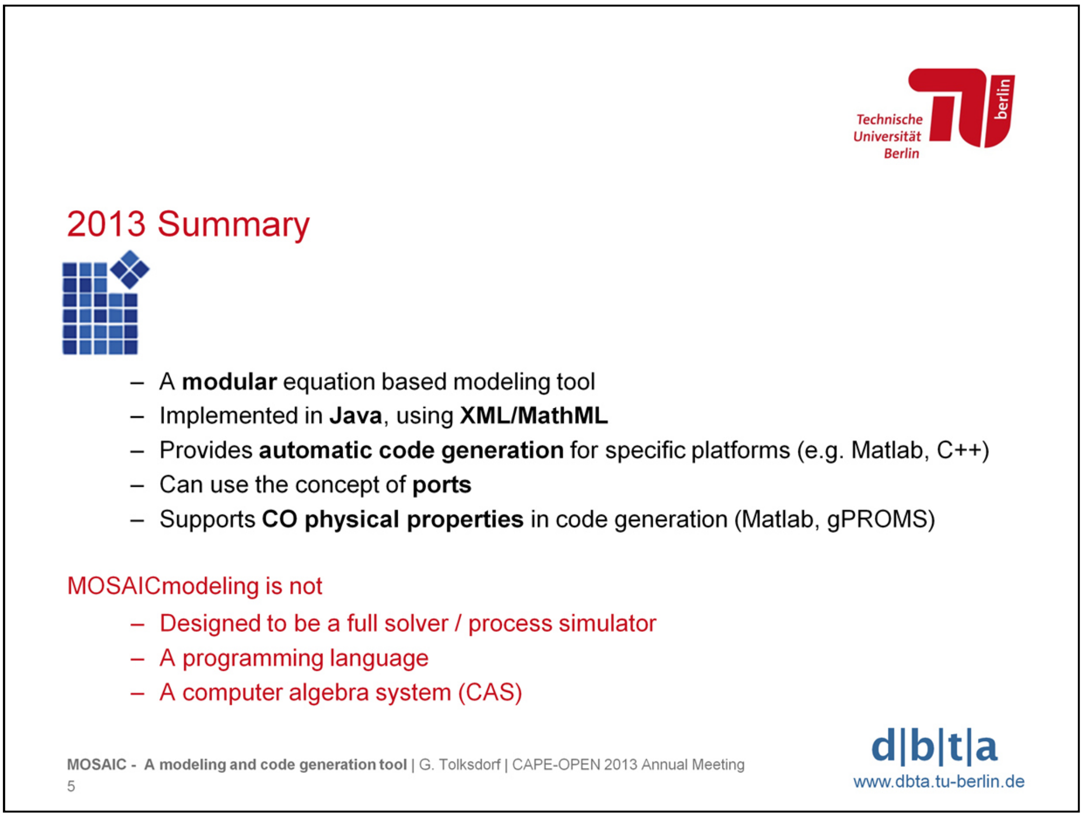
CAPE-OPEN and MOSAIC
- Physical Properties
- Unit Operations

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

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

This presentation covered 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.



Technische Universität Berlin

2013 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)

MOSAICmodeling is not

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

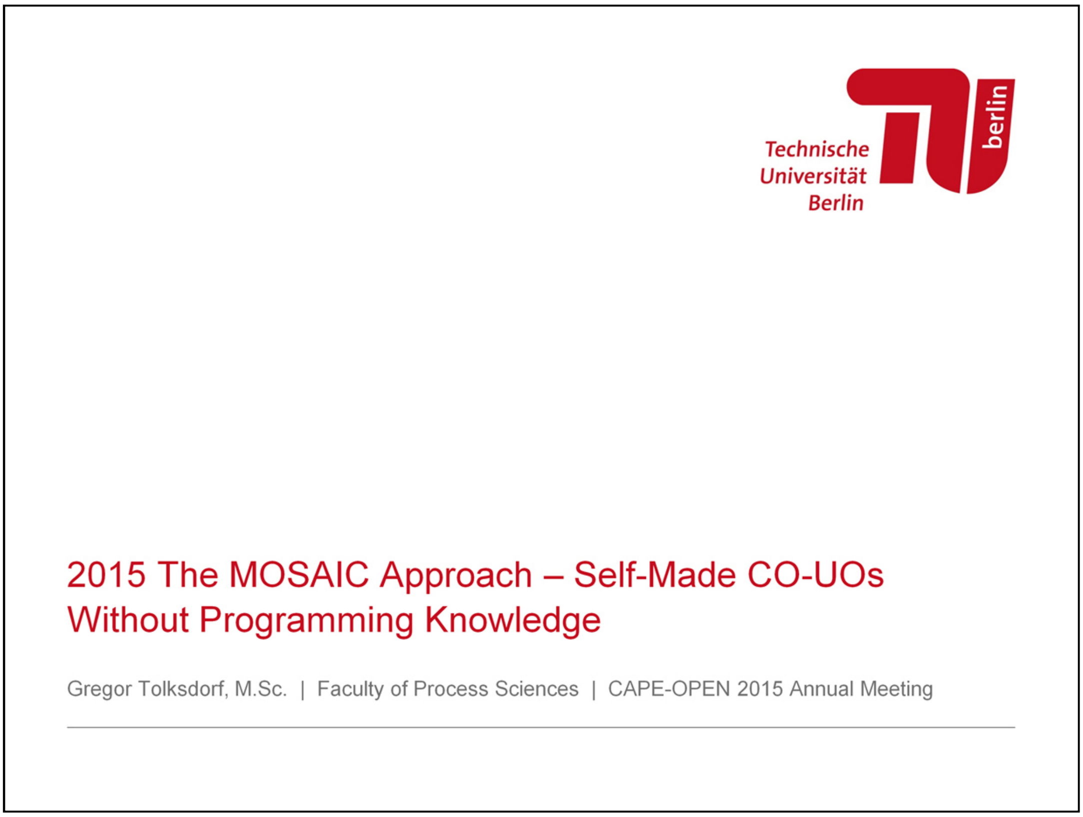
MOSAIC - A modeling and code generation tool | G. Tolksdorf | CAPE-OPEN 2013 Annual Meeting
5

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

Summarizing what has been mentioned in the 2013 presentation it can be said that MOSAICmodeling

- 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 MOSAICmodeling is neither a programming language nor a computer algebra system (CAS) . As it 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 a modelling and code generation tool.

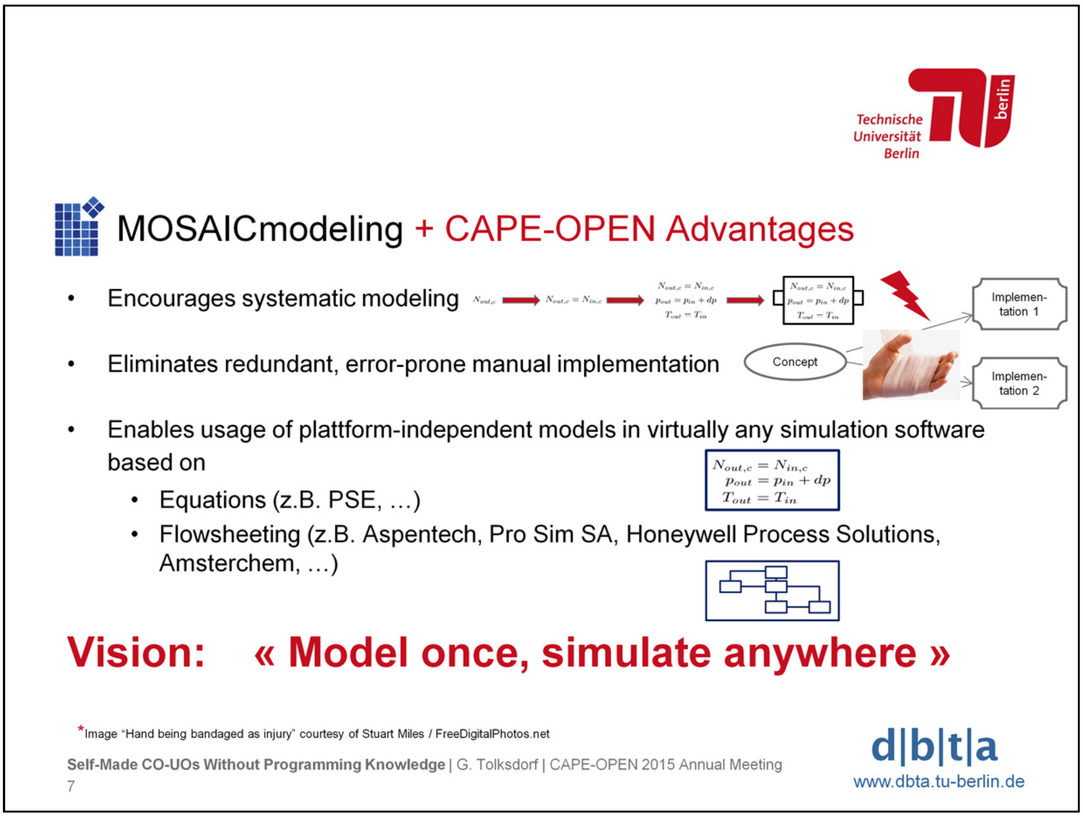


Technische Universität Berlin

2015 The MOSAIC Approach – Self-Made CO-UOs Without Programming Knowledge

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

The presentation of „The MOSAIC approach“ for the CAPE-OPEN Annual Meeting 2015 in Amsterdam was about the first steps to automatically generated code that can be used inside Cape-Open Unit Operations.



Technische Universität Berlin

MOSAICmodeling + CAPE-OPEN Advantages

- Encourages systematic modeling $N_{out,c} \rightarrow N_{out,c} = N_{in,c} \rightarrow \begin{matrix} N_{out,c} = N_{in,c} \\ p_{out} = p_{in} + dp \\ T_{out} = T_{in} \end{matrix} \rightarrow \boxed{\begin{matrix} N_{out,c} = N_{in,c} \\ p_{out} = p_{in} + dp \\ T_{out} = T_{in} \end{matrix}}$
- Eliminates redundant, error-prone manual implementation
- Enables usage of plattform-independent models in virtually any simulation software based on
  - Equations (z.B. PSE, …) $\boxed{\begin{matrix} N_{out,c} = N_{in,c} \\ p_{out} = p_{in} + dp \\ T_{out} = T_{in} \end{matrix}}$
  - Flowsheeting (z.B. Aspentech, Pro Sim SA, Honeywell Process Solutions, Amsterchem, …)

Implemen-tation 1

Concept

Implemen-tation 2

Vision: « Model once, simulate anywhere »

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

Self-Made CO-UOs Without Programming Knowledge | G. Tolksdorf | CAPE-OPEN 2015 Annual Meeting
7

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

In 2015, it was stated that combining the modular, equation based models with the standardized exchange of unit operations for simulation has advantages.

Aming at getting a exchangeable unti operations, users of MOSAICmodeling are encouraged to use a systematic modeling approach of several layers, including

- notation,
- composition of variable names,
- equations,
- systems of equations, and
- Encapsulation with ports.

The automatic code generation features of MOSAICmodeling prevent redundant, error-prone, manual implementation of one and the same concept/model for different simulation environments. By supporting Cape-Open, the platform-independent models can be used in virtually any simulation software, based on either equations or flowsheeting (unit operations).

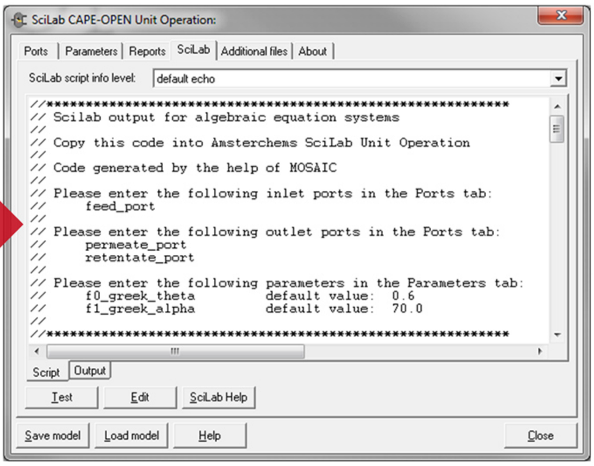
The vision of this MOSAICmodeling project can therefore be captured with „Model once, simulate anywhere", meaning that once you have entered your model in MOSAICmodeling, there is no need to manually implement it again in another language or modelling environment. The automatized code generation for equation based software products (e.g. Matlab scripts) or flowsheeting (using the Cape-Open standard) will do the rest.

Technische Universität Berlin

2015 Application Example - Simple Membrane Separation

COCO/COFE:

Code Export



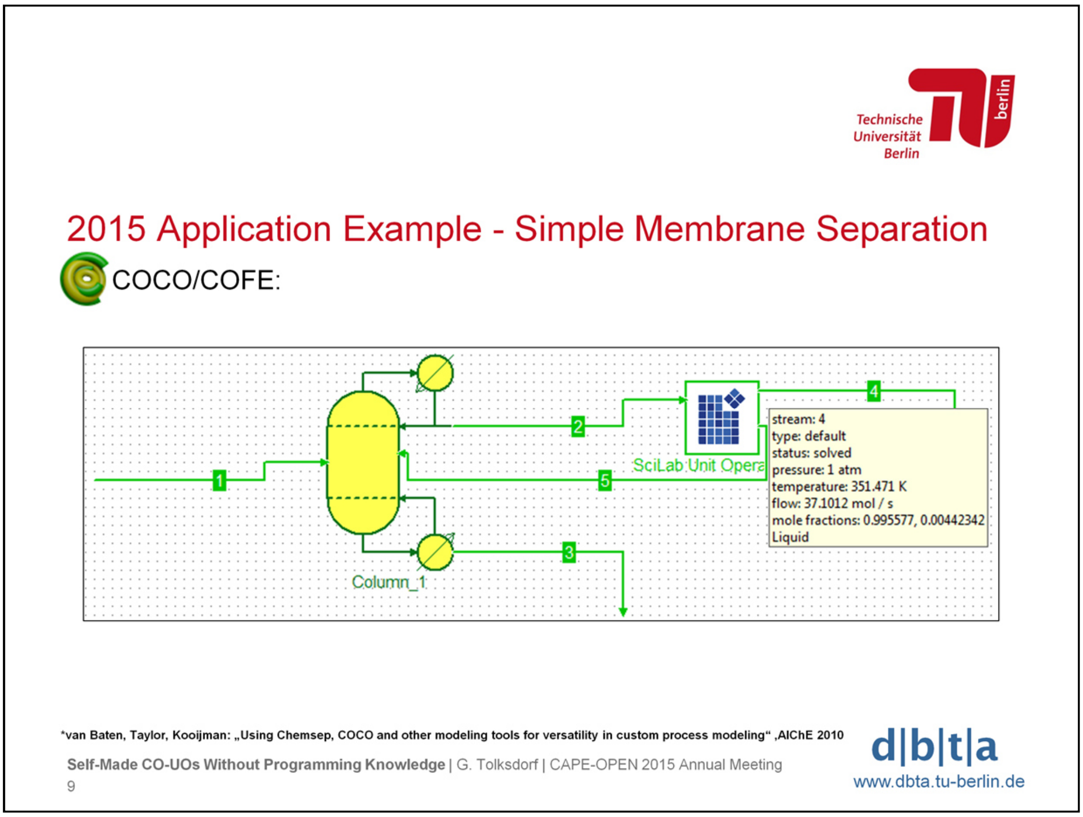
SciLab script info level: default echo

```
//************************************************************
// 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
//
//************************************************************
```

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

Self-Made CO-UOs Without Programming Knowledge | G. Tolksdorf | CAPE-OPEN 2015 Annual Meeting
8

The application example was the model of a membrane module for separation. It was shown that MOSAICmodeling could generate SciLab code to be directly inserted into the SciLab CAPE-OPEN Unit Operation from AmsterCHEM. After adding the necessary port and parameter information in the user interface, the unit operation can be connected to streams and the model is ready to be solved.



2015 Application Example - Simple Membrane Separation

COCO/COFE:

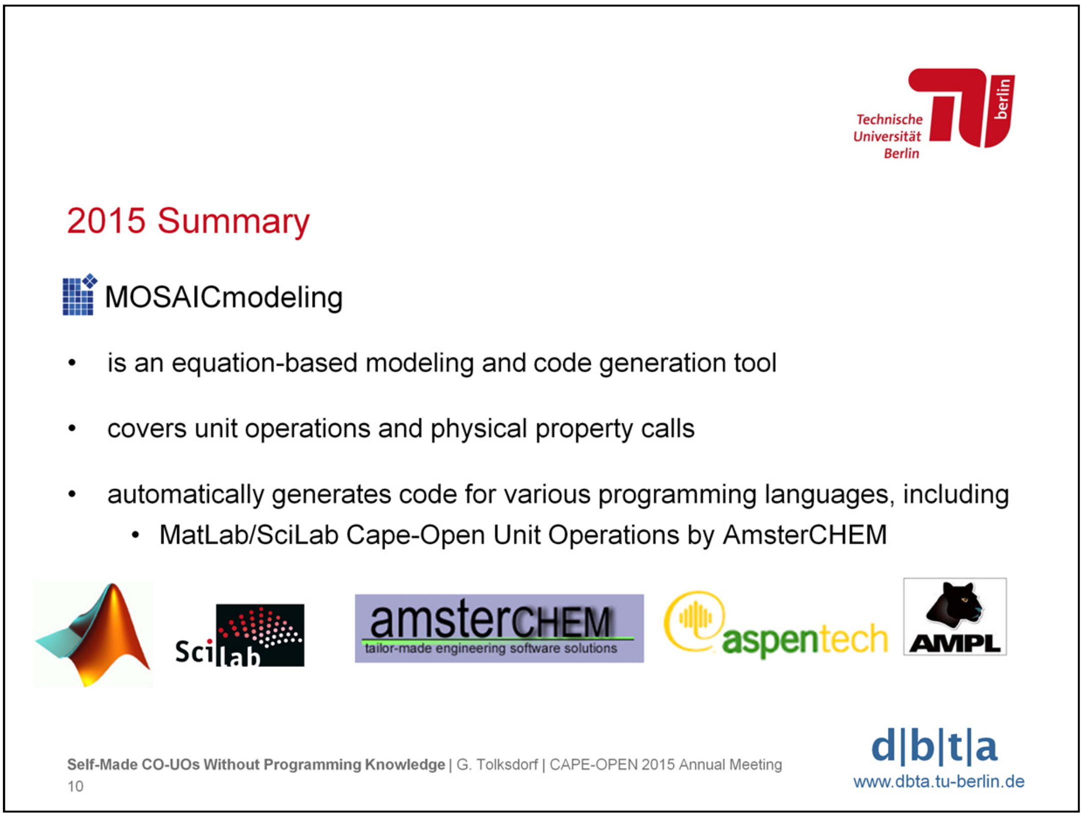
stream: 4
type: default
status: solved
pressure: 1 atm
temperature: 351.471 K
flow: 37.1012 mol / s
mole fractions: 0.995577, 0.00442342
Liquid

*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. Tolksdorf | CAPE-OPEN 2015 Annual Meeting
9

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

Finally the Cape-Open Unit Operation containing the code generated by MOSAICmodeling was succesfully integrated into a simple flowsheet consisting of a distillation column and this membrane module for separation of ethanol and water. For this example COCO/COFE was used as Cape-Open compliant Process Modelling Environment (PME).



Technische Universität Berlin

2015 Summary

MOSAICmodeling

- 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 Unit Operations by AmsterCHEM

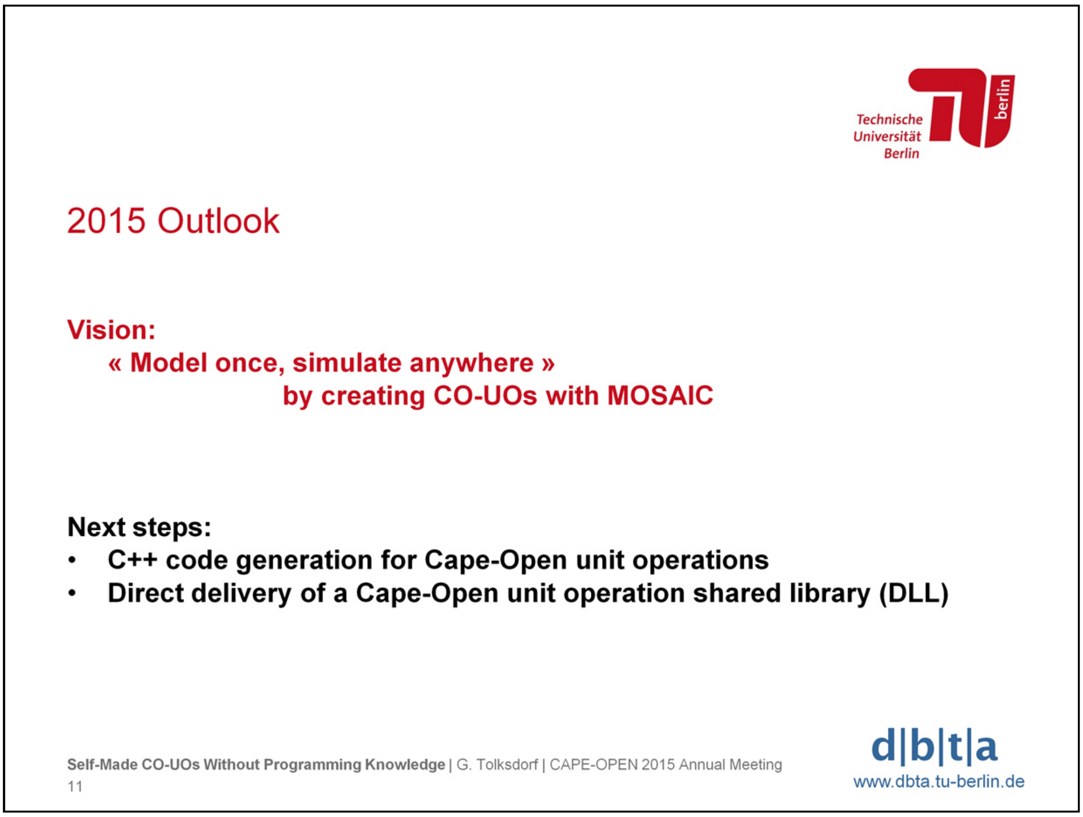
Scilab
amsterCHEM
tailor-made engineering software solutions
aspentech
AMPL

Self-Made CO-UOs Without Programming Knowledge | G. Tolksdorf | CAPE-OPEN 2015 Annual Meeting
10

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

In 2015 it was summarized that MOSAICmodeling is (still) an equation-based modelling and code generation tool, now also covering unit operations besides the physical property calls.

The automatic code generation comprises tools and languages for simulation and optimization, especially for Matlab/SciLab Cape-Open Unit Operations provided by AmsterCHEM.



Technische Universität Berlin

2015 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)**

Self-Made CO-UOs Without Programming Knowledge | G. Tolksdorf | CAPE-OPEN 2015 Annual Meeting
11

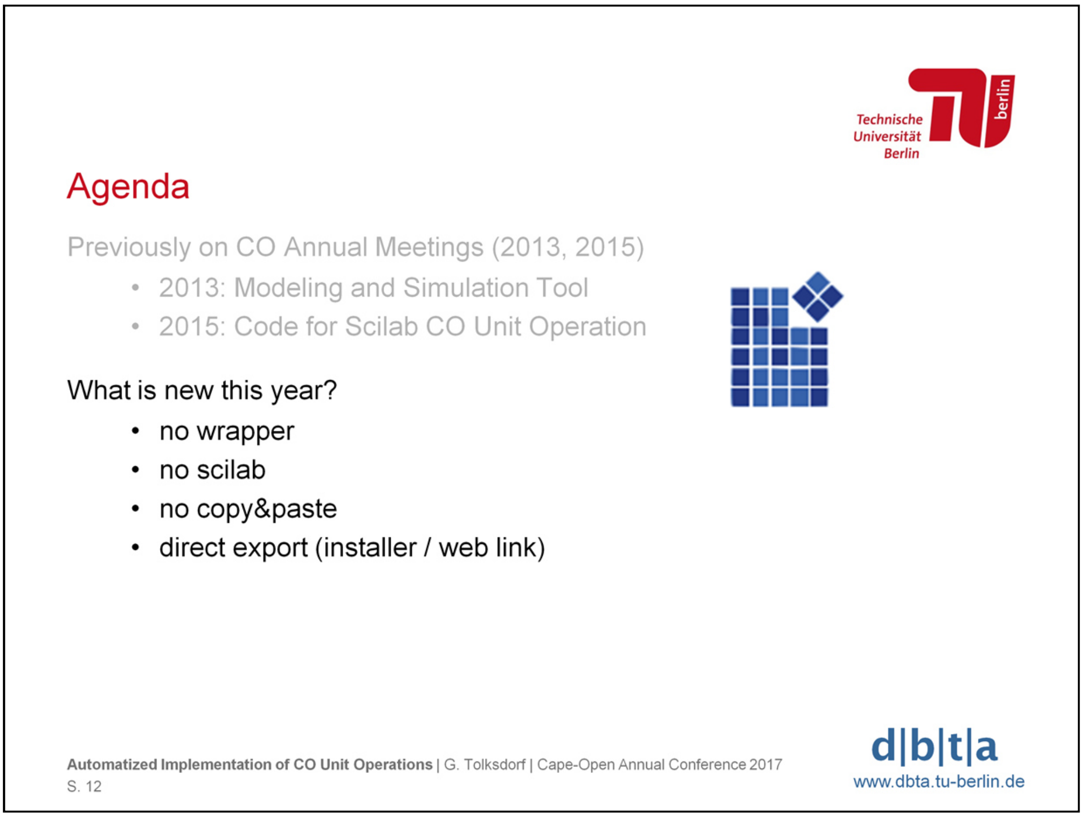
d|b|t|a
www.dbta.tu-berlin.de

The „mission“ of modelling once and executing anywhere is accomplished by creating Cape-Open Unit Operations with MOSAICmodeling.

Next steps:

Direct C++ code generation for Cape-Open Unit Operations including an appropriate solver.

In the second step, the whole code should be generated and afterwards automatically compiled to a dll file.



Technische Universität Berlin

Agenda

Previously on CO Annual Meetings (2013, 2015)

- 2013: Modeling and Simulation Tool
- 2015: Code for Scilab CO Unit Operation

What is new this year?

- no wrapper
- no scilab
- no copy&paste
- direct export (installer / web link)

Automatized Implementation of CO Unit Operations | G. Tolksdorf | Cape-Open Annual Conference 2017
S. 12

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

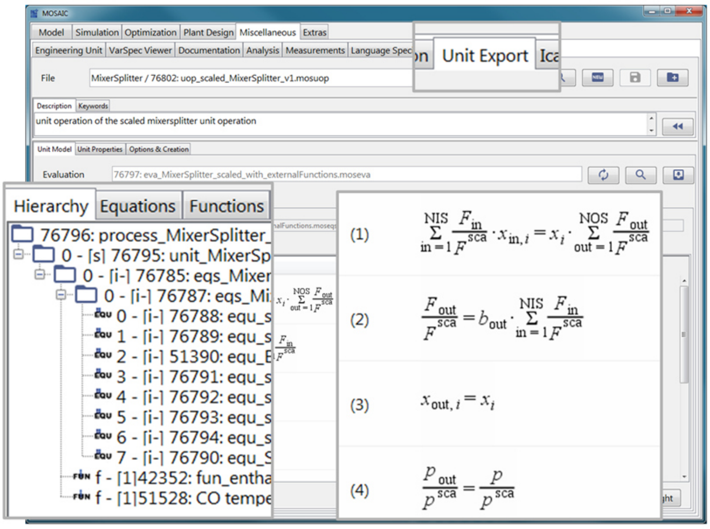
In the second part of this presentation the new developments are shown.

To sum it up: now there is no need for a wrapper (i.e. Matlab/SciLab Cape-Open Unit Operation) anymore, there is no need for a SciLab or Matlab installation, and there is no copy and paste of code necessary anymore. MOSAICmodeling now exports the model directly and delivers a web link to an installer including the user‘s unit operation.

Impressions and an example of this feature are given in the rest of this presentation.

Technische Universität Berlin

What is new 2017? – Export Editor



File MixerSplitter / 76802: uop_scaled_MixerSplitter_v1.mosuop

unit operation of the scaled mixersplitter unit operation

Evaluation 76797: eva_MixerSplitter_scaled_with_externalFunctions.moseva

76796: process_MixerSplitter_
0 - [s] 76795: unit_MixerSp
0 - [i-] 76785: eqs_Mixer
0 - [i-] 76787: eqs_Mi
0 - [i-] 76788: equ_s
1 - [i-] 76789: equ_s
2 - [i-] 51390: equ_E
3 - [i-] 76791: equ_s
4 - [i-] 76792: equ_s
5 - [i-] 76793: equ_s
6 - [i-] 76794: equ_s
7 - [i-] 76790: equ_S
f - [1]42352: fun_entha
f - [1]51528: CO tempe

$$\sum_{in=1}^{NIS} \frac{F_{in}}{F^{sca}} \cdot x_{in,i} = x_i \cdot \sum_{out=1}^{NOS} \frac{F_{out}}{F^{sca}} \quad (1)$$

$$\frac{F_{out}}{F^{sca}} = b_{out} \cdot \sum_{in=1}^{NIS} \frac{F_{in}}{F^{sca}} \quad (2)$$

$$x_{out,i} = x_i \quad (3)$$

$$\frac{p_{out}}{p^{sca}} = \frac{p}{p^{sca}} \quad (4)$$

New editor
for direct export
of
Unit Operations

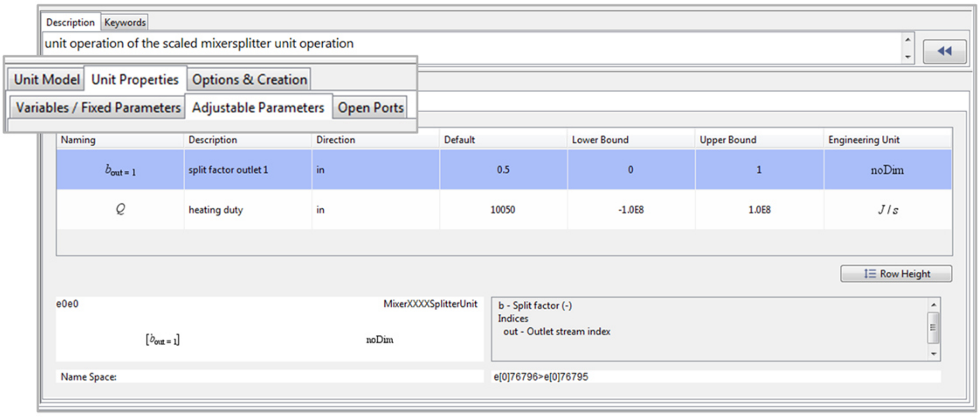
Automatized Implementation of CO Unit Operations | G. Tolksdorf | Cape-Open Annual Conference 2017
S. 13

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

In order to support direct export of Unit Operations a new editor has been added to the MOSAICmodeling user interface.

In one part of this editor the model (i.e. the underlying equations and variables) of the unit operation is shown, including the hierarchy of the (sub) models (on the left) and the explicit equations (on the right). This editor is meant for checking the existing model, not for editing or creating a new one.

What is new 2017? – Export Editor



unit operation of the scaled mixersplitter unit operation

Naming	Description	Direction	Default	Lower Bound	Upper Bound	Engineering Unit
$b_{out = 1}$	split factor outlet 1	in	0.5	0	1	noDim
Q	heating duty	in	10050	-1.0E8	1.0E8	J / s

e0e0 MixerXXXXSplitterUnit

$[b_{out = 1}]$ noDim

b - Split factor (-)
Indices
out - Outlet stream index

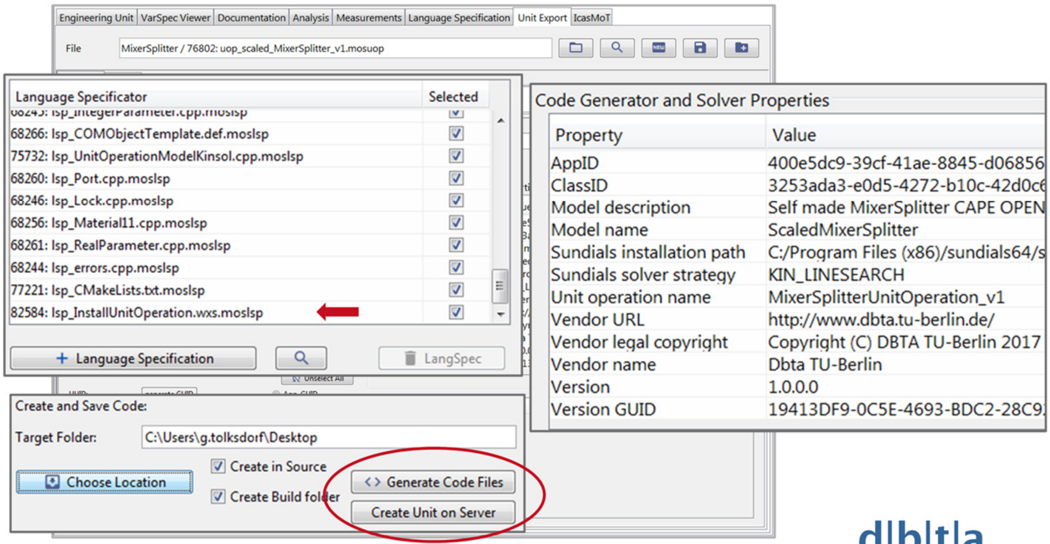
Name Space: e[0]76796>e[0]76795

Setting description and values of parameters

In another sub part of the new editor the public/adjustable parameters are available for setting description, direction, default value and value boundaries.

Technische Universität Berlin

What is new 2017? – Export Editor



File: MixerSplitter / 76802: uop_scaled_MixerSplitter_v1.mosuop

Language Specificator	Selected
68245: lsp_IntegerParameter.cpp.moslsp	☑
68266: lsp_COMObjectTemplate.def.moslsp	☑
75732: lsp_UnitOperationModelKinsol.cpp.moslsp	☑
68260: lsp_Port.cpp.moslsp	☑
68246: lsp_Lock.cpp.moslsp	☑
68256: lsp_Material11.cpp.moslsp	☑
68261: lsp_RealParameter.cpp.moslsp	☑
68244: lsp_errors.cpp.moslsp	☑
77221: lsp_CMakeLists.txt.moslsp	☑
82584: lsp_InstallUnitOperation.wxs.moslsp	☑

Code Generator and Solver Properties

Property	Value
AppID	400e5dc9-39cf-41ae-8845-d06856
ClassID	3253ada3-e0d5-4272-b10c-42d0c6
Model description	Self made MixerSplitter CAPE OPEN
Model name	ScaledMixerSplitter
Sundials installation path	C:/Program Files (x86)/sundials64/s
Sundials solver strategy	KIN_LINESEARCH
Unit operation name	MixerSplitterUnitOperation_v1
Vendor URL	http://www.dbta.tu-berlin.de/
Vendor legal copyright	Copyright (C) DBTA TU-Berlin 2017
Vendor name	Dbta TU-Berlin
Version	1.0.0.0
Version GUID	19413DF9-0C5E-4693-BDC2-28C9.

Create and Save Code:

Target Folder: C:\Users\g.tolksdorf\Desktop

☑ Create in Source

☑ Create Build folder

Automatized Implementation of CO Unit Operations | G. Tolksdorf | Cape-Open Annual Conference 2017

S. 15

d|b|t|a

www.dbta.tu-berlin.de

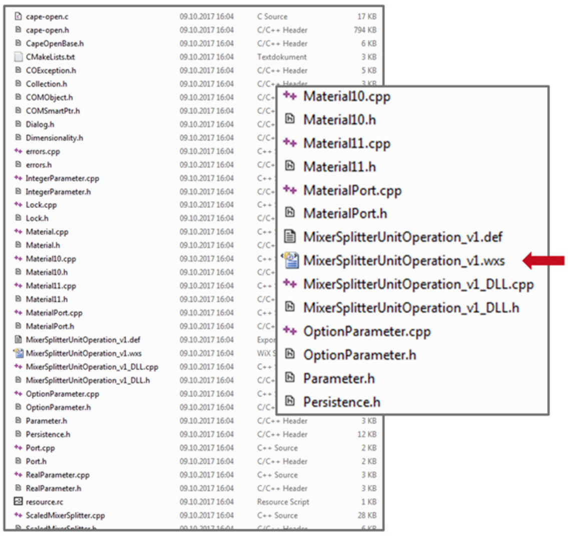
When it comes to the actual creation and export of code, several decisions have to be made.

1) The user has to select the code files that should be created for the unit operation. This is realized by so-called „Language Specificators“. Each specificator defines a single file to be created, e.g. cpp files oder header files for a c++ project, or a wxs file in order to create an installer file (highlighted with an arrow in the figure).

2) When the code files to be created have been selected, their options have to be set. On the right hand side of the figure the typical set of options is visible, e.g. model name, solver options, vendor information, and global unique identifier (GUIDs).

3) After choosing the specifications and setting the options the user can select a location where to store the code files locally, or decide to use the MOSAICmodeling server to create the unit operation online. This is represented by the encircled buttons in the figure saying „Generate Code Files“ (for the local creation), and „Create Unit on Server“, respectively.

What is new 2017? – Source Code Files



```
cape-open.c                                  09.10.2017 16:04   C Source          17 KB
cape-open.h                                  09.10.2017 16:04   C/C++ Header     794 KB
CapeOpenBase.h                               09.10.2017 16:04   C/C++ Header       6 KB
CMakeLists.txt                               09.10.2017 16:04   Textdokument       3 KB
COException.h                                09.10.2017 16:04   C/C++ Header       5 KB
Collection.h                                 09.10.2017 16:04   C/C++ Header       3 KB
COMObject.h                                  09.10.2017 16:04   C/C+
COMSmartPtr.h                                09.10.2017 16:04   C/C+
Dialog.h                                     09.10.2017 16:04   C/C+
Dimensionality.h                             09.10.2017 16:04   C/C+
errors.cpp                                   09.10.2017 16:04   C++
errors.h                                     09.10.2017 16:04   C/C+
IntegerParameter.cpp                         09.10.2017 16:04   C++
IntegerParameter.h                           09.10.2017 16:04   C/C+
Lock.cpp                                     09.10.2017 16:04   C++
Lock.h                                       09.10.2017 16:04   C/C+
Material.cpp                                 09.10.2017 16:04   C++
Material.h                                   09.10.2017 16:04   C/C+
Material10.cpp                               09.10.2017 16:04   C++
Material10.h                                 09.10.2017 16:04   C/C+
Material11.cpp                               09.10.2017 16:04   C++
Material11.h                                 09.10.2017 16:04   C/C+
MaterialPort.cpp                             09.10.2017 16:04   C++
MaterialPort.h                               09.10.2017 16:04   C/C+
MixerSplitterUnitOperation_v1.def            09.10.2017 16:04   Expor
MixerSplitterUnitOperation_v1.wxs            09.10.2017 16:04   WiX S
MixerSplitterUnitOperation_v1_DLL.cpp        09.10.2017 16:04   C++
MixerSplitterUnitOperation_v1_DLL.h          09.10.2017 16:04   C/C+
OptionParameter.cpp                          09.10.2017 16:04   C++
OptionParameter.h                            09.10.2017 16:04   C/C+
Parameter.h                                  09.10.2017 16:04   C/C++ Header       3 KB
Persistence.h                                09.10.2017 16:04   C/C++ Header      12 KB
Port.cpp                                     09.10.2017 16:04   C++ Source         2 KB
Port.h                                       09.10.2017 16:04   C/C++ Header       2 KB
RealParameter.cpp                            09.10.2017 16:04   C++ Source         3 KB
RealParameter.h                              09.10.2017 16:04   C/C++ Header       3 KB
resource.rc                                  09.10.2017 16:04   Resource Script    1 KB
ScaledMixerSplitter.cpp                      09.10.2017 16:04   C++ Source        28 KB
ScaledMixerSplitter.h                        09.10.2017 16:04   C/C++ Header       6 KB
```

```
Material10.cpp
Material10.h
Material11.cpp
Material11.h
MaterialPort.cpp
MaterialPort.h
MixerSplitterUnitOperation_v1.def
MixerSplitterUnitOperation_v1.wxs
MixerSplitterUnitOperation_v1_DLL.cpp
MixerSplitterUnitOperation_v1_DLL.h
OptionParameter.cpp
OptionParameter.h
Parameter.h
Persistence.h
```

Ca. 45 source code files generated

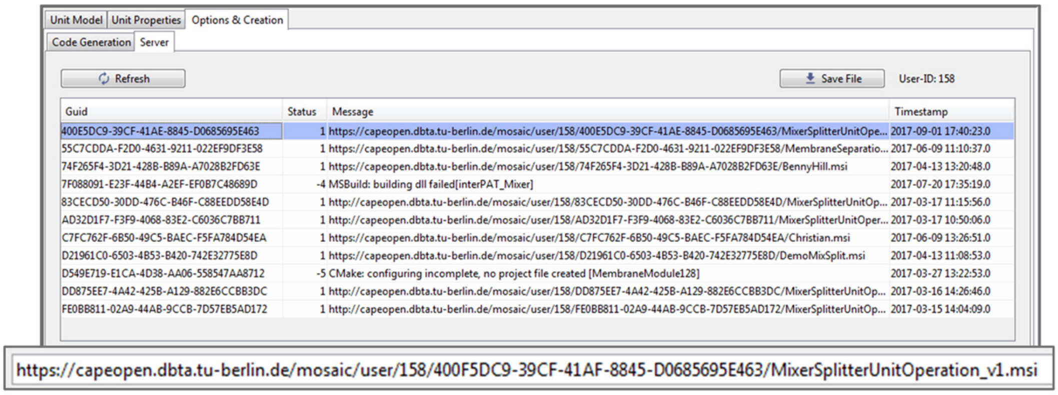
for creating the installer file

After local code creation the user can directly see all the generated files. In this example of a MixerSplitter system around 45 source code files have been stored, including the wxs file for creating the installer file that will include the actual unit operation dll file.

Technische Universität Berlin

What is new 2017? – Let the Server do it

Compressed code files are sent to the server



User-ID: 158

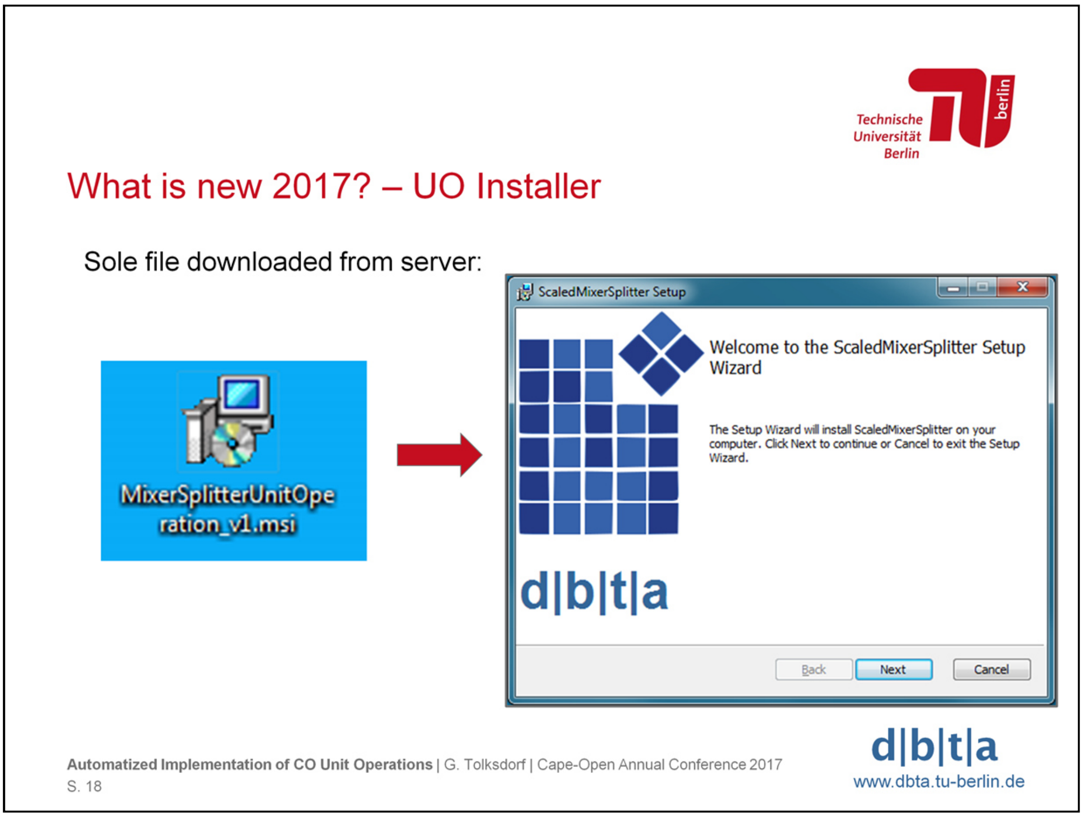
Guid	Status	Message	Timestamp
400E5DC9-39CF-41AE-8845-D0685695E463	1	https://capeopen.dbta.tu-berlin.de/mosaic/user/158/400E5DC9-39CF-41AE-8845-D0685695E463/MixerSplitterUnitOpe...	2017-09-01 17:40:23.0
55C7CDDA-F2D0-4631-9211-022EF9DF3E58	1	https://capeopen.dbta.tu-berlin.de/mosaic/user/158/55C7CDDA-F2D0-4631-9211-022EF9DF3E58/MembraneSeparatio...	2017-06-09 11:10:37.0
74F265F4-3D21-428B-B89A-A7028B2FD63E	1	https://capeopen.dbta.tu-berlin.de/mosaic/user/158/74F265F4-3D21-428B-B89A-A7028B2FD63E/BennyHill.msi	2017-04-13 13:20:48.0
7F088091-E23F-44B4-A2EF-EF0B7C48689D	-4	MSBuild: building dll failed[interPAT_Mixer]	2017-07-20 17:35:19.0
83CECD50-30DD-476C-B46F-C88EEDD58E4D	1	http://capeopen.dbta.tu-berlin.de/mosaic/user/158/83CECD50-30DD-476C-B46F-C88EEDD58E4D/MixerSplitterUnitOp...	2017-03-17 11:15:56.0
AD32D1F7-F3F9-4068-83E2-C6036C7BB711	1	http://capeopen.dbta.tu-berlin.de/mosaic/user/158/AD32D1F7-F3F9-4068-83E2-C6036C7BB711/MixerSplitterUnitOper...	2017-03-17 10:50:06.0
C7FC762F-6B50-49C5-BAEC-F5FA784D54EA	1	https://capeopen.dbta.tu-berlin.de/mosaic/user/158/C7FC762F-6B50-49C5-BAEC-F5FA784D54EA/Christian.msi	2017-06-09 13:26:51.0
D21961C0-6503-4B53-B420-742E32775E8D	1	https://capeopen.dbta.tu-berlin.de/mosaic/user/158/D21961C0-6503-4B53-B420-742E32775E8D/DemoMixSplit.msi	2017-04-13 11:08:53.0
D549E719-E1CA-4D38-AA06-558547AA8712	-5	CMake: configuring incomplete, no project file created [MembraneModule128]	2017-03-27 13:22:53.0
DD875EE7-4A42-425B-A129-882E6CCBB3DC	1	http://capeopen.dbta.tu-berlin.de/mosaic/user/158/DD875EE7-4A42-425B-A129-882E6CCBB3DC/MixerSplitterUnitOp...	2017-03-16 14:26:46.0
FE0BB811-02A9-44AB-9CCB-7D57EB5AD172	1	http://capeopen.dbta.tu-berlin.de/mosaic/user/158/FE0BB811-02A9-44AB-9CCB-7D57EB5AD172/MixerSplitterUnitOp...	2017-03-15 14:04:09.0

https://capeopen.dbta.tu-berlin.de/mosaic/user/158/400F5DC9-39CF-41AF-8845-D0685695E463/MixerSplitterUnitOperation_v1.msi

Direct https link for download

When choosing the online generation on the server, a list of already created unit operations is show and with a refresh button the status of the unit operation currently processed on the server can be checked. Status „1“ means that the unit operation installer file was succesfully created. Negative status numbers indicate errors during the configuration or compiling steps. In this case the user would have to check if the correct set of files („Language Specificators“) was selected in the previous steps.

By selecting a successfully created unit operation, the user can save the respective installer file on her local machine. Alternatively, the displayed link can be used to make the installer of this unit operation available via browser to colleagues and collaboration partners.


Technische Universität Berlin
berlin

What is new 2017? – UO Installer

Sole file downloaded from server:

MixerSplitterUnitOperation_v1.msi
ScaledMixerSplitter Setup
d|b|t|a
Back
Next
Cancel

Welcome to the ScaledMixerSplitter Setup Wizard

The Setup Wizard will install ScaledMixerSplitter on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

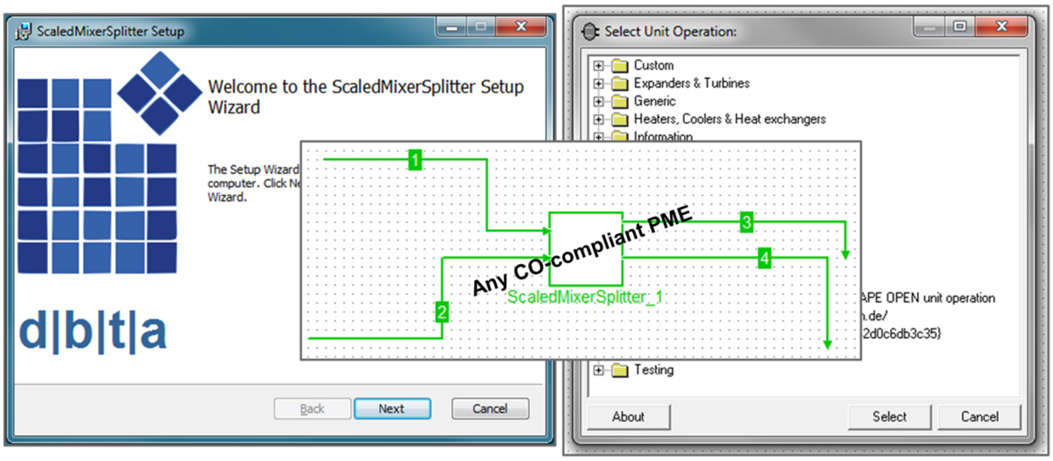
Automatized Implementation of CO Unit Operations | G. Tolksdorf | Cape-Open Annual Conference 2017
S. 18

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

The single file necessary to make the Cape-Open Unit Operation available for simulation is the downloadable installer file that starts the installer wizard.

Technische Universität Berlin

What is new 2017? – UO Installer



ScaledMixerSplitter Setup

Welcome to the ScaledMixerSplitter Setup Wizard

The Setup Wizard
computer. Click Ne
Wizard.

Select Unit Operation:

APE OPEN unit operation
.de/
2d0c6db3c35}

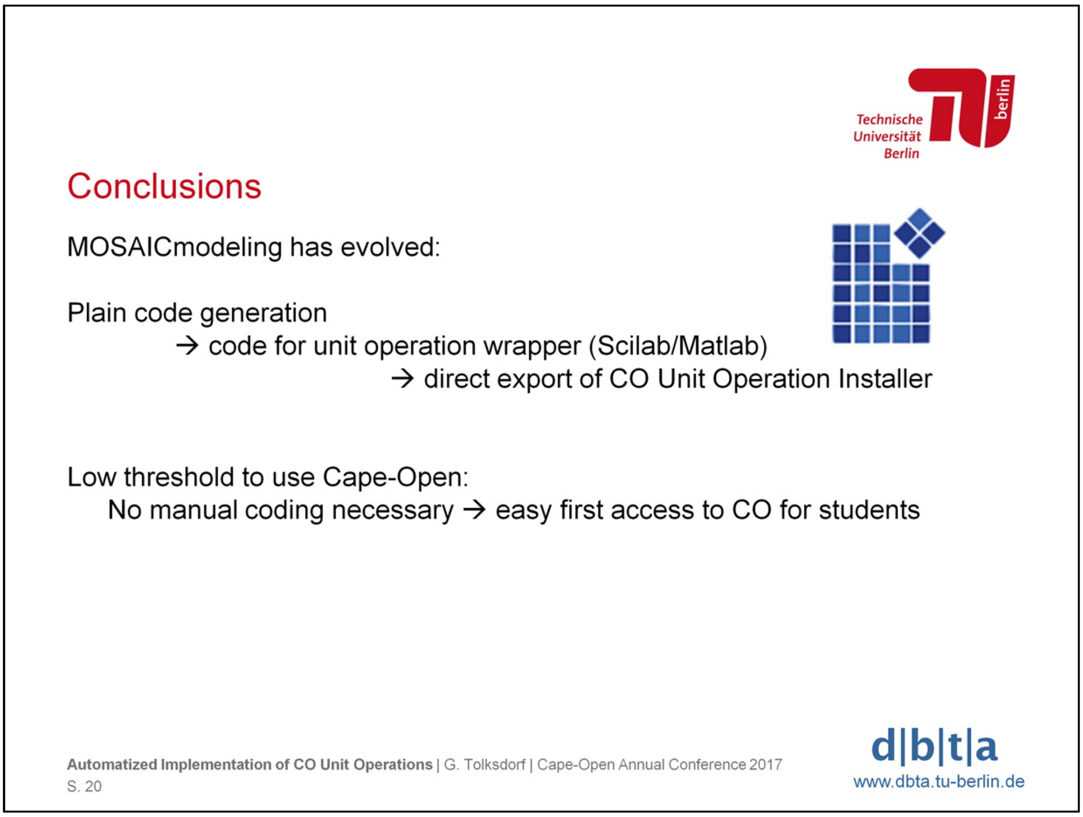
The Cape-Open Type Library is part of the installation!

Automatized Implementation of CO Unit Operations | G. Tolksdorf | Cape-Open Annual Conference 2017
S. 19

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

The installer takes care of registering the CAPE-OPEN type library and the particular unit operation, making the model available in any CAPE-OPEN-compliant process modelling environment.

In the figure the solved unit operation is shown in COCO/COFE as one example of a CO-compliant PME.



Technische Universität Berlin

Conclusions

MOSAICmodeling has evolved:

Plain code generation

→ code for unit operation wrapper (Scilab/Matlab)

→ direct export of CO Unit Operation Installer

Low threshold to use Cape-Open:

No manual coding necessary → easy first access to CO for students

Automatized Implementation of CO Unit Operations | G. Tolksdorf | Cape-Open Annual Conference 2017
S. 20

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

To conclude it can be stated that MOSAICmodeling has evolved in the past four years. Starting with plain code generation in 2013, code for unit operation wrapper relying on Matlab/SciLab was developed until 2015. Now (2017) a direct export of CO Unit Operation Installers is supported, thus reducing the user‘s need to do actual coding (in C++ or a similar programming language) to a minimum.

This way the threshold to use Cape-Open for creation and exchange of unit operations is lowered. By eliminating manual coding, it is now possible to bring process engineering students in contact with Cape-Open early on. This would not be possible if they had to learn a fully-fledged programming language and a lot more technical details including middleware aspects in advance.

Technische Universität Berlin

Thank you for your attention.

d|b|t|a

MOSAICmodeling

www.mosaic-modeling.de

Acknowledgement:

unicat

This project is

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

For more information visit ‘www.mosaic-modeling.de‘.

The MOSAICmodeling project is supported by the Cluster of Excellence 'Unifying Concepts in Catalysis‘, coordinated by Technische Universität Berlin, and funded by the German Research Foundation.