CHEMICAL ENGINEERING TRANSACTIONS
VOL. 43, 2015
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.aidic.it/cet
Chief Editors: Sauro Pierucci, Jiří J. Klemeš

**ISBN** 978-88-95608-34-1; **ISSN** 2283-9216
DOI: 10.3303/CET1543225

# A Posteriori Integration of University CAPE Software Developments

Gregor Tolksdorf*[a], Sandra Fillinger[a], Günter Wozny[a], Flavio Manenti[b], Francesco Rossi[b], Guido Buzzi-Ferraris[b], Sauro Pierucci[b], Marina Fedorova[c], Rafiqul Gani[c]

[a]Technische Universität Berlin, Chair of Process Dynamics and Operations, Straße des 17.Juni 135, 10623 Berlin, Germany
[b]Politecnico di Milano, CMIC Dept. "Giulio Natta", Piazza Leonardo Da Vinci 32, 20100 Milano, Italy
[c]Technical University of Denmark, CAPEC-PROCESS, Dept. of Chemical and Biochemical Engineering, Søltofts Plads 229, 2800 Kgs. Lyngby, Denmark

gregor.tolksdorf@tu-berlin.de

This contribution deals with the mutual integration of existing CAPE software products developed at different universities in Germany, Denmark, and Italy. After the motivation MOSAIC is presented as the bridge building the connection between the modelling tool ICAS-MoT and the numerical processing tool BzzMath. In the main part a case-study is presented, including descriptions of the benefits, challenges, implementation, and application. This paper is completed with aspects of current research activities and a conclusion summarizing the results.

## 1. Introduction

In the area of Computer Aided Process Engineering (CAPE), universities are developing software to support engineers in modelling, simulation, optimization, or process synthesis. Thereby modern aspects like web-based "cloud" technologies, open standards, integration of modelling, steady state and dynamic simulation, re-usability etc. have to be considered. These university research developments are usually not integrated in commercial products. On the other hand, universities pay for licences of proprietary CAPE software, e.g. AspenPlus®, gPROMS®, CHEMCAD®, PROII®, to quote some of them. Universities could cooperate with each other and incorporate their own developments to create synergies in CAPE research and education (Merchan et al., 2013). The synergies could be used to improve the solutions performance and the exchange of specialized process models. This way the number of used expensive proprietary software tools in a single department can be reduced. This leads to less payments for licences, and more money available to spend for the own developments and employees. Such a development puts some pressure on commercial vendors to improve their products and to integrate more of the up-to-date software technologies. MOSAIC, a freely available modelling software designed at TU Berlin, realizes most of the modern aspects mentioned above and integrates the developments of other universities: The robust numerical libraries "BzzMath" from Politecnico di Milano and the modelling tool "ICAS-MoT" from the Technical University of Denmark (DTU), the latter supported by an industrial consortium of CAPE software users.

## 2. The MOSAIC software

MOSAIC is a non-commercial, web based modelling, simulation, optimization, and code generation tool developed at the chair of process dynamics and operations at TU Berlin, Germany, and written in Java. Its development started ten years ago (Zerry et al., 2004). The software and its model databases are accessible by any computer with an internet connection (www.mosaic-modeling.de). The main server is located in Berlin and provides users with the latest software version when starting the program. Additional servers with model

Please cite this article as: Tolksdorf G., Fillinger S., Wozny G., Manenti F., Rossi F., Buzzi-Ferraris G., Pierucci S., Fedorova M., Gani R., 2015, A-posteriori integration of university cape software developments, Chemical Engineering Transactions, 43, 1345-1350
DOI: 10.3303/CET1543225

databases are located in Asia, North-, and South-America to enable faster access when residing in countries outside Europe.
The MOSAIC models are stored in a platform-independent way by using the XML/MathML standards and SQL-Databases. Users are supported to model in a well-documented way, as descriptions for all used symbols, variables and equations are mandatory. Design values, parameters, initial values, and starting values are individually set with so-called variable specification lists. All model elements (e.g. equations, notations, variable specifications) are exchangeable and reusable. It is an approach focusing on the problem description instead of algorithmic solution instructions. This way MOSAIC is designed to create code for various general or problem-specific programming languages (ACM®, AMPL, C++, GAMS, gPROMS®, Matlab®, Python, etc.). By using scientific libraries, freely available for academic purposes, MOSAIC is suitable to simulate problems directly on the MOSAIC server. In this case no local installation is needed and the results are directly imported into the user interface. In terms of simulations, algebraic equation systems (NLE), first order ordinary differential equation systems (ODE), and semi-explicit differential algebraic equation systems (DAE) are supported. In an additional module LP, NLP, MILP, and MINLP optimization problems can be solved by using the code generator and a connection to the NEOS-Optimization-Server (Czyzyk et al. 1998). The results of the optimization can be imported into the simulation to make full use of MOSAIC's versatility. A full documentation in LaTeX including descriptions of all the parts of the model can be generated immediately with the documentation module. A more detailed description of different aspects of MOSAIC has been published (Kuntsche et al., 2011).

## 3. Case study: A posteriori integration of different university CAPE developments

### 3.1 Exemplary developments ICAS-MoT and BzzMath

ICAS-MoT is based on in-depth work-flows for different generic modelling tasks such as model development, analysis, identification, discrimination, documentation and application. At each step of the work-flow MoT provides the required support and connections to libraries or other tools (Heitzig et al., 2010). In general it is straight forward to introduce a model in MoT, as the equations are presented through a very simple syntax close to a natural writing way. Therefore, no programming is necessary to develop, analyse and/or solve a model in MoT.
MoT is connected to a modelling templates library and the ICAS-tool ModDev, which can generate equations based on user specifications, forming the general modelling framework (Fedorova et al., 2013). Other internal connections include process simulator and thermodynamic databases of ICAS. The results obtained during the model development and application process in MoT can be exported in XML-format or as a COM-object, e.g., to MS Excel.
BzzMath (Buzzi-Ferraris & Manenti, 2012) is a numerical library of efficient and reliable computational tools, designed for engineering purposes. It is a framework organized in C++ classes, thus based on the idea of object-oriented programming. Its classes allow the user to solve a broad range of problems, including: (I) non-linear dense and sparse ODE/DAE systems; (II) mono-dimensional BVPs; (III) Unconstrained and constrained optimizations with efficient and robust algorithms (local and global minimum searches); (IV) Algebraic linear and non-linear systems; (V) Linear and non-linear regressions; (VI) Design of experiments; etc. In addition, the library contains several utilities for linear algebra operations and charts construction. BzzMath is based on cutting edge numerical methods, which have been successfully tested on several theoretical and real industrial problems over the last decade. The library algorithms have been employed in very computational demanding applications (NMPC, DRTO, etc.) where efficiency and reliability are the key targets. Consequently, BzzMath appears to be a well-established framework for both the academic and the industrial world. Finally, the BzzMath library is free of charge for academic users and can be downloaded from the web (Buzzi-Ferraris, 2014).

### 3.2 Targeted benefits

In the last decades a lot of models for different units and processes have been developed. Unfortunately the migration of models to new dedicated software or new powerful programming languages does not come for free. Time consuming redundant and manual reprogramming of already existing process models is error-prone, and should be avoided. The first targeted benefit of this cooperation project is to enable users of ICAS/MoT to easily migrate the models of their model library to bigger number of programming languages and modelling environments. Especially the members of the industrial consortium behind ICAS-MoT could benefit from MOSAIC's possibility of automatic code generation for special target platforms.
Automatic code generation is one of the main features of MOSAIC. On one hand, a local installation of proprietary software is necessary to execute the code generated for these tools. On the other hand, freely

available programming languages like C++ that could be compiled and executed directly on the server usually don't have built-in solution algorithm to solve large equation systems. So the second benefit to achieve is to simulate models immediately on the MOSAIC server by using the robust numerical C++ BzzMath libraries. It is also desired to show the capabilities of BzzMath in actual research applications.
A third targeted benefit from the academic research perspective is related to the costs of all the CAPE tools used in a single department. Universities would like to avoid paying for licences of commercial software products. One way to achieve this can be an extended cooperation with other universities and usage of their latest developments.

### 3.3 Challenges in the integration process

In this project, the products ICAS-MoT and BzzMath were not designed to be integrated either with each other or with MOSAIC. Neither was MOSAIC itself. ICAS-MoT and MOSAIC are both equation-oriented modelling tools what makes exchange of models naturally easier as if one of them was a flowsheeting software. Nevertheless some exchange format has to be defined, at best using an official open standard like XML (Extensible Markup Language).
Another challenge besides the general development of an exchange format and procedure is the loss of information, if a model is transferred from one modelling environment to the other. As long as each modelling tool has its own special features, it cannot be avoided to have some examples where models can only be transmitted incompletely. Being aware of this point it is an additional aim to have at least a clearly defined list of criteria to determine if a model can be exchanged without loss of information or not.
The BzzMath libraries are written in C++ and have to be compiled for the appropriate operating system and system configurations. The BzzMath developers are focusing on the Windows operating system whereas the MOSAIC server runs on a Linux machine. Therefore it cannot be avoided to invest some time to adapt parts of the BzzMath code and use a Linux compiler to generate the executables for the MOSAIC server.

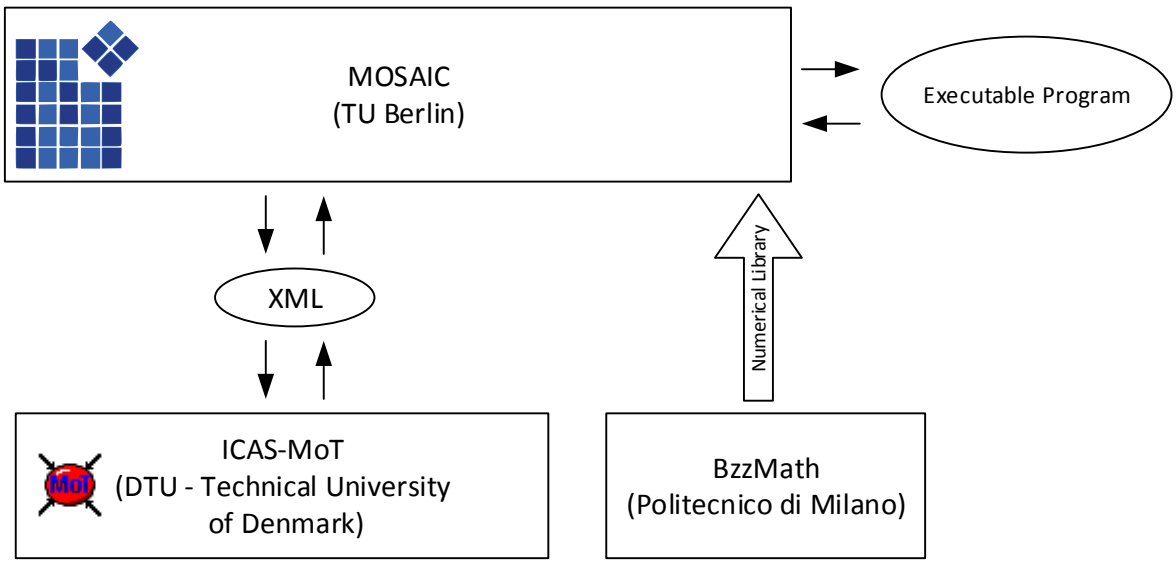


*Figure 1: CAPE software integration setting. MOSAIC is the main integration platform in this project, realizing model exchange via xml with ICAS-MoT and including the BzzMath libraries to create highly specific problem solving executables. Simulation results of these executables can be imported back to MOSAIC for visualization.*

### 3.4 Implementation

Implementing a connection between ICAS-MoT and MOSAIC basically meant developing the schema of a transfer file, collecting all necessary model information in a machine-readable format. After an analysis of similarities and differences, it could be stated that most properties of a model in one tool can be mapped directly to the properties in the other tool. The aspects of equations, indices, and variables regarding the models, and the technical aspects of the xml file are highlighted in the following paragraphs. At last, the

integration of the BzzMath libraries is described. The overall setting of the integration and the relations between ICAS-MoT, MOSAIC and BzzMath is shown in Figure 1.

**Equations and Indices**
Both modelling environments store equations in a generic respectively unexpanded way, e.g.

$$F \cdot z_i = D \cdot y_i + B \cdot x_i \,, \tag{1}$$

and use the information of indices, e.g.

$$i \in \{1,2,3\}$$

to instantiate respectively expand the equations to

$$F \cdot z_{i=1} = D \cdot y_{i=1} + B \cdot x_{i=1} \,, \tag{2}$$
$$F \cdot z_{i=2} = D \cdot y_{i=2} + B \cdot x_{i=2} \,, \tag{3}$$
$$F \cdot z_{i=3} = D \cdot y_{i=3} + B \cdot x_{i=3} \,. \tag{4}$$

The common structure resulted in three main parts in the xml transfer file, i.e. “equations”, “indexing”, and “variables”.

**Variable Types**
Variables inside the equations can be of different types. Each type is mapped on a type-id in the transfer file. The ids and type names in ICAS-MoT and MOSAIC, respectively, are given in Table 1. Variables describing the first derivative (type 6, Dependent Prime) of state variables (type 5, Dependent Variable) are not explicitly expressed in MOSAIC, but can be automatically determined by analysing the equations’ differential operators.

*Table 1: Mapping of variable types in ICAS-MoT and MOSAIC*

| Variable Type | ICAS-MoT | MOSAIC |
|---|---|---|
| 1 | Parameter | Parameter |
| 2 | Known | Design variable |
| 3 | Unknown | Iteration variable |
| 4 | Explicit variable | Calculated variable |
| 5 | Dependent variable | Iteration variable (state variable) |
| 6 | Dependent prime | --- |

**XML schema file**
The transfer file is an XML file containing all model information in a tree structure. XML schemas are special XML files containing a formal description of the tree structure and elements that are allowed in another XML file (W3C, 2012). With such a schema the transfer files can be automatically checked for formal correctness of the tree structure and the type of content in the XML elements. This helps to decrease the complexity of parsers to import the transfer files, because it can be assumed to have a valid model if the XML file is complying with the schema and further test for correctness in the parser are not necessary anymore. The particular schema for this project has been developed according to the so-called “Venetian Blind” design, having exactly one global element and defining all other xml elements as local elements.

**BzzMath**
In order to integrate the C++ BzzMath libraries, it was necessary to compile the source code fitting to the Linux server that is used in the MOSAIC environment. As the BzzMath libraries are currently developed mainly for Windows platforms, an additional revision of the BzzMath source code had to be done to include commands for Linux cases. Regarding MOSAIC, a new code generator had to be written to create C++ code that solves the model by including the compiled BzzMath libraries and using the functions of this numerical library. This generated code then has to be compiled again and can be executed directly on the MOSAIC server to calculate the solution.

### 3.5 Application and results

A generalized chemical processing model proposed at the beginning of the sixties (Williams & Otto, 1960) was used to show the interconnectivity between ICAS-MoT, MOSAIC and BzzMath. In this case the simplified model consisted of a set of fifteen algebraic equations describing the stationary operation of a reactor. The model is part of the collection of examples that comes with an ICAS-MoT installation and was stored as a mot-file: “WilliamsOtto Reactor ICAS.mot”. ICAS-MoT was used to open this model and to save it as an xml file according to the cooperatively developed xml-schema (see 3.4). In the next step MOSAIC loaded and translated the xml file containing the transferred model. The file collected all information necessary to create an internal platform-independent model that is used to generate platform-specific, problem-solving code. In this application MOSAIC had to generate C++ code including the BzzMath numerical libraries to solve the nonlinear equation system (see Figure 2). By applying all the variable information, i.e. fixed parameter values and initial guesses, contained in the xml transfer file, the code could be automatically compiled and executed directly on the MOSAIC server. The results were immediately written back into the MOSAIC user interface. A comparison with the results obtained with the original model file in ICAS-MoT eventually showed that the results were identical and therefore the model transfer including the code generation and execution was successful.

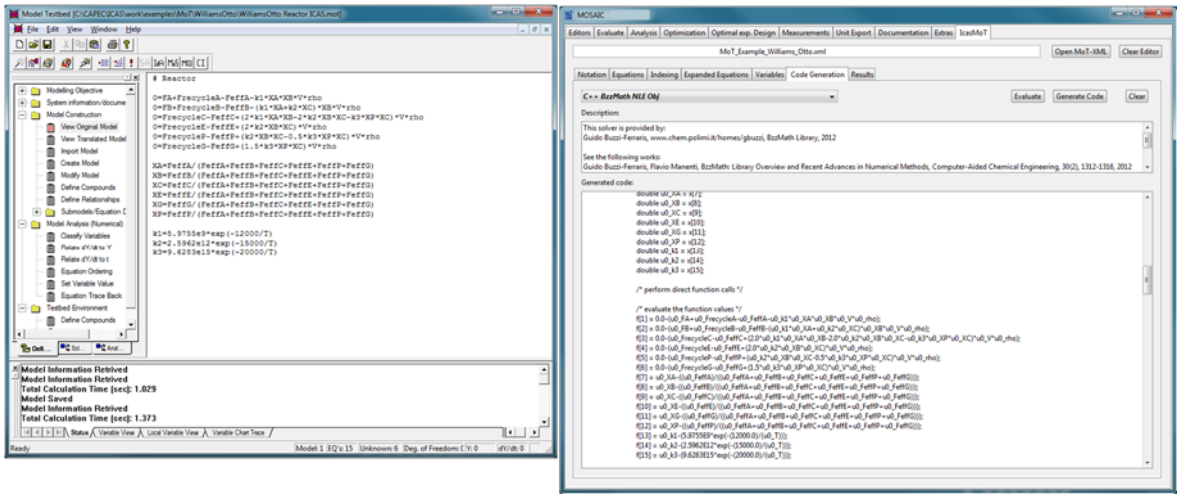

*Figure 2: Screenshots of the model in ICAS-MoT (on the left) and MOSAIC (on the right).The original ICAS-MoT model was transferred and C++ code, using the BzzMath libraries to solve the model, has been created automatically. The code can be compiled and executed directly on the MOSAIC server to get results immediately written back to the user interface.*

## 4. Ongoing developments

The number of MOSAIC users around the world is still increasing and it is planned to install more servers besides the ones in Germany, Brazil, Colombia, Malaysia and the United States. With decreasing distance to the next server, users will experience a faster connection. MOSAIC can be further used in university education (not only at TU Berlin) to demonstrate a systematic methodology of modelling and optimizing equation-based systems (Merchan et al., 2013). Current research is about flow sheeting on the documentation level and model export via automatically generated CAPE-OPEN Unit Operations. The information exchange between simulation and three-dimensional plant design software is another interesting topic under development. With automatic discretization of PDAEs and chance-constraint optimization support, two new powerful options are being prepared in MOSAIC. The combination of experimental data and plant configuration information together with a MOSAIC model in one database builds the basis for a full documentation of not only modelling but also operation of plants and processes (Kraus et al., 2014). The modular structure and usage of well-known standards makes MOSAIC comparatively easy to be extended in its functionality (Merchan et al., 2014) and the connection to an optimization server shows the power of flexible, automatic code generation based on well-documented, platform-independent models. MOSAIC is currently only used for research in universities and it is still a step to go into industrial application. A framework for local model storage in an intranet, in contrast to the centralized database available from around the world, is a need to protect industrial models containing valuable proprietary information. For a pilot study industrial partners are still to be found.

## 5. Conclusions

This article showed a way to a-posteriori combine different software tools, developed in different universities, to use their individual strengths in the CAPE modelling and simulation. MOSAIC (TU Berlin, Germany), the main integration platform in this project, realizes aspects of modern software products like web-based technologies, open standards, and re-usability. ICAS-MoT (DTU, Denmark), a modelling suite connected to industrial end-users, was delivering one selected model out of a wide range of models, that was transferred to MOSAIC via a jointly developed transfer file format. The code generation facility of MOSAIC was afterwards used to include the robust numerical libraries of BzzMath (Politecnico di Milano, Italy) in C++ code to solve the model. The example reported in this article was a case study building the basis for closer cooperation in model exchange (ICAS-MoT / MOSAIC) and enhanced problem-solving techniques (BzzMath / MOSAIC). Positive effects of the collaboration should be further investigated. The usage of the synergies in industrially productive applications is one main goal of ongoing developments. The collaboration leads to a reduction of licence fees by avoiding proprietary software.

**Acknowledgements**

This work is part of the Cluster of Excellence “Unifying Concepts in Catalysis” coordinated by the Technische Universität Berlin. Financial support by the Deutsche Forschungsgemeinschaft (DFG) within the framework of the German Initiative for Excellence as well as support from Alexander von Humboldt Stiftung is gratefully acknowledged.

**References**

Buzzi-Ferraris G., Manenti F., 2012, BzzMath: Library Overview and Recent Advances in Numerical Methods, Computer-Aided Chemical Engineering, 30, 1312-1316, 2012

Buzzi-Ferraris G., 2014, BzzMath library download <http://super.chem.polimi.it> accessed 20 November 2014

Czyzyk J., Mesnier M. P., Moré J. J. 1998. The NEOS Server. IEEE Journal on Computational Science and Engineering 5(3), 68-75.

Fedorova M., Sin G., Gani R., 2013, Computer-aided modeling framework – a generic modeling template for catalytic membrane fixed bed reactors, Computer Aided Chemical Engineering, 32, 775-780.

Heitzig M., Sin G., Glaborg P., Gani R., 2010, A computer-aided framework for regression and multi-scale-modelling needs in innovative product-process engineering, Computer Aided Chemical Engineering, 29, 379-384.

Kraus R., Fillinger S., Tolksdorf G., Hoang Minh D., Merchan V.A., Wozny G., 2014, Improving Model and Data Integration Using MOSAIC as Central Data Management Platform, Chem. Ing. Tech. 2014, 86, No.7, 1130-1136 DOI: 10.1002/cite.201400007

Kuntsche S., Barz T., Kraus R., Arellano-Garcia H., Wozny G., 2011, MOSAIC a web-based modelling environment for code generation, Comp. Chem. Eng. 35 (11), 2257-2273, DOI: 10.1016/j.compchemeng.2011.03.022

Merchan V.A., Kraus R., Wozny G., 2013, Use of a web-based modeling environment in the education of process engineers, Chemical Engineering Transactions, 35, 673-678 DOI:10.3303/CET1335112

Merchan V.A., Tolksdorf G., Kraus R., Wozny G., 2014, Extending Documentation-Based Models towards an Efficient Integration into Commercial Process Simulators, Chem. Ing. Tech. 2014, 86, No.7, 1117-1129 DOI: 10.1002/cite.201400014

W3C (World Wide Web Consortium), 2012, XML schema standard definition <http://www.w3.org/standards/techs/xmlschema> accessed 20 November 2014

Williams T.J., and Otto R.E., 1960, A generalized chemical processing model for the investigation of computer control. AIEE Trans. 79, P. 1, Communications and Electronics, (1960), 458-473

Zerry R., Gauss B., Urbas L., Wozny G., 2004, Web-based object oriented modelling and simulation using MathML, 2004, Comput. Aided Chem. Eng., 18, 1171-1176, DOI: 10.1016/S1570-7946(04)80261-X