CHEMICAL ENGINEERING TRANSACTIONS
VOL. 35, 2013

A publication of AIDIC
The Italian Association of Chemical Engineering
www.aidic.it/cet

Guest Editors: Petar Varbanov, Jiří Klemeš, Panos Seferlis, Athanasios I. Papadopoulos, Spyros Voutetakis

ISBN 978-88-95608-26-6; **ISSN** 1974-9791
DOI: 10.3303/CET1335112

Use of a Web-Based Modeling Environment in the Education of Process Engineers

Victor Alejandro Merchan*, Robert Kraus, Günter Wozny

Chair of Process Dynamics and Operation, Technische Universität Berlin, Sekr. KWT-9, Str. Des 17. Juni 135, D-10623 Berlin, Germany
v.merchanrestrepo@tu-berlin.de

The use of equation-oriented modeling and simulation software for the formulation and solution of model-based tasks has become an essential element in the education of process engineers. Computational exercises, offered normally as a supplement to traditional courses on process modeling, as well as special courses on process simulation, have proven to be an appropriate way of teaching the use of these modeling and simulation tools. The experience with such courses, however, shows that there is still a considerable room for improvement, when it comes to transferring the theoretical knowledge taught in the lecture into the computational exercises. When using conventional tools or modeling languages, actually unimportant software interface and syntax issues may play a central role, while fundamental aspects like a systematic model development and a degree of freedom analysis step into the background.
In this contribution we report on the use of the web-based modeling environment MOSAIC (Kuntsche et al. 2011) in the education of process engineers. Compared to conventional modeling and simulation tools MOSAIC combines several novel concepts that promote the application of systematic model development, while simultaneously facilitating the learning of further standard tools and modeling languages. The possibility to define and manage models over the internet and the provided automatic code generation for any equation-oriented software tool present themselves as an improvement over the traditional teaching methods and is accepted very well by the students.

1. Introduction

Following the successes resulting from the use of process simulation and further model-based tasks in the process and chemical industries, process modeling and its applications have become an indispensable part of the education of chemical and process engineers. Taking a look at job openings of popular companies in these fields, it is evident that besides knowledge and experience in the field of modeling a certain level of proficiency in the popular process modeling and simulation tools is highly desirable. This applies not only to the popular modular flow sheet simulation programs but also to the equation-oriented modeling and simulation environments that have increasingly gained on importance in the past years. The high flexibility given by the equation-oriented architecture regarding the development of custom user models and the possibility of using these models for several tasks as simulation, parameter estimation, optimization and model validation within one and the same modeling framework has significantly contributed to the further development and acceptance of advanced model based methods.
In order to target the needs of the industry and to educate future researches for the development of further efficient model based methods, universities offer several courses covering the range from process modeling to process optimization. In addition to mandatory basic lectures in the field of modeling, several courses in form of computational exercises or lectures with integrated computational exercises make nowadays part of the curriculum. Table 1 gives an overview of the courses related to process modeling consisting of lecture and computational exercises as offered by the Chair of Process Dynamics and Operation at the TU Berlin (Technische Universität Berlin) along with information regarding the used software tools. Similar courses are offered by comparable chairs in other universities. Besides the tools listed in Table 1 depending on the task to solve and experience of the people working on it further software

Please cite this article as: 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

tools are used within different projects as well. Current tasks at our Chair include modeling work with Aspen Custom Modeler and solution of large scale optimization problems using AMPL as modeling language.

*Table 1: Overview of the courses consisting of lecture and computational exercise offered at the Chair of Process Dynamics and Operation at the Technische Universität Berlin and the software tools used**

Course title	Used software tools
Computer Exercises to Process and Plant Dynamics	MATLAB
Process synthesis	Aspen Plus
Process simulation I/II	Aspen Plus/gPROMS
Computer aided plant design	CHEMCAD/PDMS
Process control and operation	MATLAB/Simulink
Process optimization	GAMS / MATLAB TOMLAB

* In contrary to the mandatory lecture on modeling "Process and Plant dynamics" all these courses are elective for the study program energy and process engineering. Further computational exercises with focus in other fields are offered by other chairs

In general, computational exercises have proven to be a proper complement for lectures on modeling principles and an appropriate way of teaching the use of modeling and simulation tools. The experience with such courses, however, shows that there is still a considerable room for improvement, when it comes to transferring the theoretical knowledge taught in the lecture into the computational exercises. In particular, based on our experience, we have recognized following significant problems:

- Many students show already deficient modeling skills at a basic level and have difficulties in applying the modeling workflow.
- Learning a new programming language or syntax is often seen as major obstacle. Because of this sometimes the computer aided solutions of relevant systems is not even attempted.
- Actually unimportant software interface and syntax issues may play a central role, while fundamental aspects like systematic model development and degree of freedom analysis step into the background.
- A big organizational effort is required in order to teach platform specific modeling and simulation.

It is noticeable that the first three problems listed above are related to the connection between the theoretical knowledge about modeling on the one hand and the use of modeling tools on the other hand.
While modeling by itself does not require knowledge on the use of modeling environments, the use of proper modeling environments, which is normally taught in separate courses, promotes the modeling skills and the learning of the modeling workflow. Therefore it has been found necessary to build a better connection between the contents of the different courses concerning modeling and its applications. Since a strong basis in modeling is necessary for further tasks discussed in the advanced courses, it is of particular interest to extend the basic modeling lecture in such a way, that the students have a better possibility of practicing the systematic model development while getting in contact with equation based modeling tools.

1.1 Experiences in promotion of modeling skills through the integration of modeling tools at early learning stages

In our chair, over the years several efforts have been done in order to promote the modeling skills through the integration of industry-relevant modeling tools in the courses, while keeping up to date with the developments in e-learning. More than ten years ago MathCAD files concerning examples of the lectures, were first made available in a CD along with further lecture materials. With the introduction of learning management systems, these files were made accessible over the internet and complemented with further examples of further tools such as MATLAB and gPROMS. The students were motivated to use these files for self-study using the available licenses of the respective software tools available in the institute's computer pool. Besides the developments in the lecture materials, the tutorial of the basic lecture has significantly evolved as well. Whereas at first the systematic model development was demonstrated "by hand" on the blackboard, nowadays the use of the possibilities opened by the electronic chalk and the live demonstration of the solution of modeling tasks with different modeling tools have become an integral part of the tutorials.
In addition to these attempts, concerned mainly to the learning materials, the very good learning outcomes in modeling principles and applications reached by students that took part in the course Computational Exercises to Process and Plant Dynamics, a course that promotes the solution of modeling tasks and

MATLAB syntax issues to the same extent, led to the introduction of a voluntary small semester project as a complement to the basic modeling lecture and its tutorials. Since organizational aspects prevent all students visiting the basic lecture on process modeling from attending the course addressed above, the voluntary semester project was conceived to give the students the possibility of practicing the modeling workflow by developing and implementing a model on their own. In order to keep the effort within limits, simple steady state models, such as a binary flash drum, or a Gibbs reactor were developed in pairs and implemented in MATLAB. The introduction of the voluntary semester project was a step in the right direction, since the students, who completed it, showed in general a better understanding of the modeling workflow in the oral exam. However it showed the big importance of the organizational problems discussed above. Even for the solution of the considered relatively simple steady state models, a significant part of the time was spent with platform dependent software and syntax issues that are neither related to the model development nor to the learning of the modeling workflow. Considering the fact, that the tasks are solved normally at an institute's computer with available software licenses and that a supervision may be required to help with the solution of software and syntax issues, it is evident that a very high organizational effort would be required in case of the introduction of a mandatory platform specific modeling project for all students visiting the basic lecture on modeling. Hence the used software tools need to fulfill further specific requirements.

1.2 Requirements concerning modeling tools for the early learning stages

It seems though, that an efficient integration of modeling tools at early learning stages would require the use of alternative software tools that reduce the potential language or syntax issues and promotes the systematic model development. For this purpose, one could think of computational systems that have been specially designed for the education, such as Polymath (Polymath, 2013) or Berkeley Madonna (Berkeley Madonna, 2013). A relevant disadvantage of this kind of systems lies on the fact, that they do not possess the industrial relevance and functionality that systems like MATLAB, Aspen Custom Modeler or gPROMS do possess and therefore the students will probably not profit of their experience with the software at later stages of their studies or careers.

Hence a suitable modeling environment should not only reduce the potential language and syntax issues so as to let the modeler focus in the systematic model development, but it also should equally give the user the possibility of learning the use of the most popular equation modeling environments. Considering the positive development in teaching and learning as a consequence of the digital revolution in the last years (Perry and Bulatov, 2009), a further important aspect is the suitability of a modeling environment for interactive e-learning as described by Grigorov et al. (2012).

Since the requirements described above are fulfilled by the web based modeling environment MOSAIC (Kuntsche et al. 2011), it was decided to embed it in the basic lecture on process modeling. In this contribution we describe how MOSAIC has been implemented in the education and report some of the insights gotten so far on how the possibilities of MOSAIC support beginners in the field of modeling.

2. MOSAIC and its features for the education of modelling principles

MOSAIC is a web-based equation-oriented modeling environment developed at the chair of process dynamics and operation of the TU Berlin. Motivated by the study of the possibilities of web-based object oriented modeling and simulation using MathML (Zerry et al. 2004), and taking into consideration the possibility of code generation in several languages based on symbolic equations as a key task (Kuntsche et al. 2009) MOSAIC as evolved to a web-based software realization of the concept of modular modeling on documentation level. The focus lies on the unification of model and model documentation in order to enhance the reusability and exchange between modelers working on different places. Further details on MOSAIC can be found on the website www.mosaic-modeling.de. There everyone can ask for an account and access MOSAIC. Further technical issues and features are discussed in detail by Kuntsche et al. (2011). The next subsections treat the main characteristics that make MOSAIC highly suitable for the use in the education.

2.1 Low amount of supervision and low organisational effort

Due to its implementation as a Java applet, the only requirements to run MOSAIC are an active internet connection and a Java-compatible computer, thus the use of MOSAIC is not limited by resource restriction in terms of space and time. Though the solution of problems using code generated by MOSAIC may be restricted by licensing issues, if the code generation for commercial tools like MATLAB or gPROMS is chosen, the students still have the possibility to solve their problems online, by choosing one of the available code generators for the numerical libraries installed in the MOSAIC for instance the BzzMath library by Buzzi-Ferraris (2010). Besides general algebraic systems, dynamic systems given by ordinary

differential equations (ODE) and differential algebraic equations (DAE) in the semi explicit form are supported.
Just as with any other software, new users first need to learn how to use MOSAIC. Herefor examples are given in the website. As reported by Kuntsche et al. (2011) students needed around 1.5 hours to master the use of MOSAIC. It is important to remark that the only thing that all new users need to learn in all cases is the workflow, since the major part of text inputs given by the user, limits to plain text used for the notation, comments and names for single MOSAIC elements that are saved in the database (equations, equation systems or evaluations). The only specific commands that may need to be learned concern the input of the equations. These are namely by default entered into MOSAIC with the document mark up language LaTeX. An advantage of using LaTeX syntax for the equation input is the possibility of writing two dimensional symbolic expressions, consisting of several combinations of base names, indices, subscripts and superscripts leading to highly readable models and model documentation. Further LaTeX is more or less a standard for writing scientific papers, so that a certain level of proficiency in it may be helpful in later stages of the studies.

2.2 Targeted promoting of the systematic model development

Support for a systematic model development is given at all stages of the modeling processes, starting from the definition of the modeling goal and an appropriate nomenclature, going through the formulation of the general model equations, the determination of the degree of freedom and selection of design variables, up to the numerical solution of the targeted modeling task. Figure 1 illustrates some aspects that emphasize the support on systematic model development given by MOSAIC by means of a model for a closed binary equilibrium stage. The figure section in the left side shows the rendered equations that build the model. It can be seen that the model equations appear in MOSAIC just as they had been written on a sheet of paper. At this point the model consists only of the universally valid equations for phase equilibrium of a binary mixture with the phi-gamma method in a closed system. No specifications related to design variables are done up to this point. This takes place in the next step shown in the middle. Here the definition of a specific simulation task takes place by fixing the design variables. The automatic counter of remaining degrees is of great use not only for modeling beginners. Finally in order to proof, whether the system is well defined, a structure analysis based on the Dulmage-Mendelsohn-decomposition (Duff, 1976) of the Jacobian incidence matrix of the system can be applied (Kraus et al. 2011). Among other things this provides graphical information on eventually underdetermined or overdetermined sections of the simulation problem, giving hints on wrong specifications or missing equations.

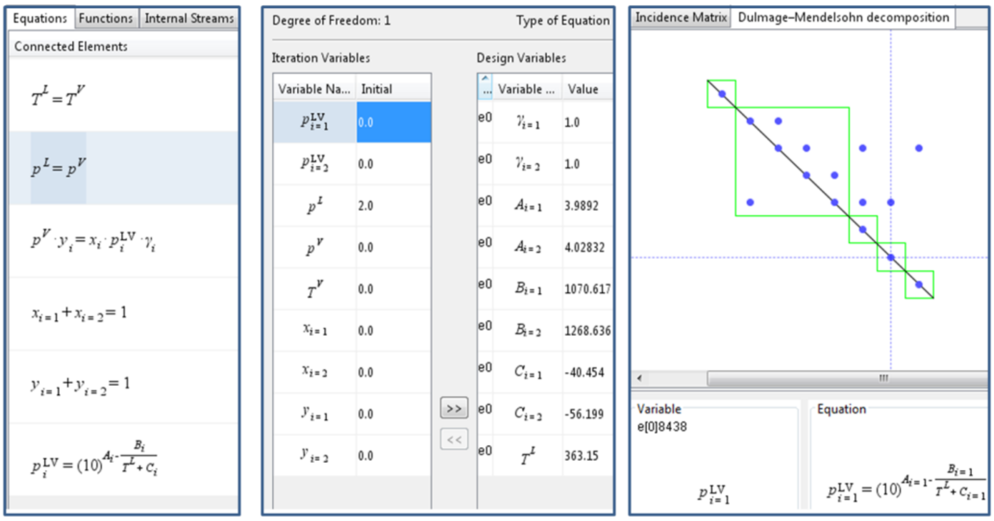

Figure 1: Some of the features that support the systematic model development. Left: Example of general (unspecified) equation systems. Middle: Graphic selection of design variables. Right: Structure analysis with Dulmage-Mendelsohn decomposition

2.3 Support in learning the use of further modeling languages and numerical libraries

This last point results as an obvious consequence of the code generation capabilities of MOSAIC. After the definition of the simulation task, by means of the selection of design variables, the modeler has the possibility to choose between several modeling languages or numerical libraries for the code generation.

Without requiring previous knowledge of popular tools like MATLAB, gPROMS, or further languages, the students obtain ready-to-run codes that can be either directly used for problem solutions, or in order to understand the problem implementation in the respective tools. But not only the learning of syntax issues enhances the learning of the selected modeling tools. Having a running model, as provided by the MOSAIC code generation, students can spend more time using and discovering the features supported by the different tools, such as the excellent debugging features of MATLAB, or the structure and index analysis for dynamic systems given by gPROMS.

3. Implementation of MOSAIC in the education

The facts discussed above motivated the decision of testing the integration of MOSAIC in the mandatory courses concerned with the basics of modeling. This integration took place in two phases. In the first phase the introduction consisted of live demonstrations in the lectures and tutorials, when discussions on the degree of freedom of “bigger” or composed systems took place. A good example was the live demonstration that the degree of freedom of a column under certain assumptions is two. Of greater interest was the second phase of the integration of MOSAIC. Based on the good experience with the previous attempts of voluntary platform dependent modeling work discussed in section 1.1, it was decided to introduce a mandatory modeling project using MOSAIC for students that study under the most actual examination regulations. While keeping the organizational efforts on the part of the course instructor low, the students got the chance of applying the modeling workflow by building a dynamic model and using it for simulation studies. For the simulation studies gPROMS was proposed, thus letting the students solve the dynamic simulation problems in a powerful commercial modeling environment, in which they have not had any previous experience. This way the feasibility of learning the use of an unknown equation oriented tool through the MOSAIC generated code, was tested on novices in the field of modeling. Experiences with more experienced modeler in the institute had already shown that the ready-to-run examples generated by MOSAIC provide a very helpful insight in syntax issues and the structure of modeling environments.

3.1 The semester modelling project

The mandatory semester project consists in an extension of an exercise treated in one of the tutorials during the semester. In the original exercise an isothermal dynamic flash drum with constant hold up (with ideal level and pressure control) describing the separation of water/ethanol mixture is modeled, thus resulting in a high index DAE system with a differential index of two. The goal of the modeling project is to further develop the model toward a more detailed modeling depth. In particular the holdup should be considered as a variable and further equations for controllers are considered as well, thus showing how an index problem can be avoided by enhancing the modeling depth. The workflow of this modeling project is shown in Figure 2 along with tools that are used at different stages of the problem solutions.

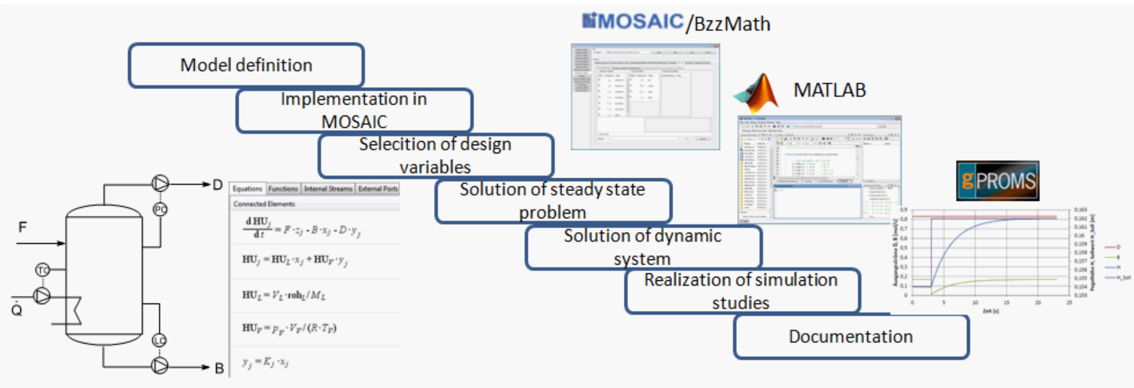

Figure 2: Workflow of the introduced MOSAIC modeling project and tools applied at the different stages

Since the simulation studies concern the reaction of the system to a deviation from the steady state, the steady state solution is required for the initialization of the problem. Here it has been interesting to see the solution strategies that single students develop on their own. We notice that many students make active use of the possibility given by the code generation to different tools. While some students solve the steady state problem online through the BzzMath library, others use MATLAB or gPROMS. Some students particularly interested in numerical aspects discuss convergence aspects through analysis in the MATLAB

debugger. Although the instructors had not put an emphasis on it, it is found that the MOSAIC models that are defined only once become the basis for a multitool modeling and simulation.
Since its introduction in Mai of 2012 around 50 students have finished the modeling project. The experience gained so far confirms that the integration MOSAIC in the education of modeling beginners was a step in the right direction. Students that have made the modeling project show in the exam in general better modeling skills than students that have not. Further many students have been introduced successfully into modeling tools such as MATLAB or gPROMS without a significant organizational effort.

4. Conclusions

The first experiences gotten so far show that the use of MOSAIC has led to an improvement in teaching of process modeling, since it has made possible to bring the modeling beginners in contact with several well developed standard modeling and simulation environments without considerable additional efforts on the part of students and instructors. The introduction of the mandatory modeling project has led not only to better modeling skills, especially concerning dynamic models, but has also brought interesting insights. In particular it is interesting to see, how modeling novices, based on the code generation capabilities by MOSAIC, develop own solution strategies based on multitool modeling and simulation.

Acknowledgement

The authors acknowledge the support from the Collaborative Research Centre SFB/TR 63 InPROMPT “Integrated Chemical Processes in Liquid Multiphase Systems” coordinated by the Berlin Institute of Technology and funded by the German Research Foundation

References

Berkeley Madonna – Modeling and Analysis of Dynamic Systems, <www.berkeleymadonna.com> accessed 21 March 2013

Buzzi-Ferraris G., 2010, BzzMath: Numerical Library in C++, Politecnico di Milano, <homes.chem.polimi.it/gbuzzi/> accessed 20 March 2013

Duff I.S., 1977, On Permutations to Block Triangular Form, IMA J. Appl. Math., 19 (3), 339-342, DOI: 10.1093/imamat/19.3.339

Grigorov A., Angelov A., Detcheva E., Varbanov P.S., 2012, Web technologies for interactive graphical e-learning tools in engineering education, Chemical Engineering Transactions, 29, 1597-1602, DOI: 10.3303/CET1229267

Kraus R., Merchan V.A., Kuntsche S., Manenti F., Buzzi-Ferraris, G., Wozny G., 2011, Modelling in the documentation level using mosaic and numerical libraries, AIDIC Conference Series, 10, 207-214, DOI: 10.3303/ACOS1110023

Kuntsche S., Arellano-Garcia H., Wozny G., 2009, Web-based object-oriented modeling environment for the simulation of chemical processes, Chemical Engineering Transactions, 18, 779-784, DOI: 10.3303/CET0918127

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

Perry S., Bulatov I., 2009, The digital revolution and the teaching and learning process in chemical engineering, Chemical Engineering Transactions, 18, 755-760, DOI:10.3303/CET0918123

Polymath Main Page Text, <www.polymath-software.com> accessed 21 March 2013

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