CHEMICAL ENGINEERING TRANSACTIONS Volume 24, 2011
Editor Sauro Pierucci

DOI: 10.3303/CET1124058

Modelling in the Documentation Level Using MOSAIC and Numerical Libraries

Stefan Kuntsche*[1], Flavio Manenti[2], Guido Buzzi-Ferraris[2], Günter Wozny[1]

[1] Technische Universität Berlin, Dynamics and Operation of Chemical Plants
Sekr. KWT-9, Str. d. 17. Juni 135, 10623 Berlin, Germany,
stefan.kuntsche@tu-berlin.de

[2] Politecnico di Milano, CMIC dept. "Giulio Natta"
Piazza Leonardo da Vinci 32, 20133 Milano, Italy

In the today's constellation of globally operating enterprises, the workforce is locally decentralized. Engineering tasks are increasingly divided up under several work groups that are situated at different premises. Information technology is used to coordinate the collaboration and to merge the results. Due to its versatility and adaptability, the cooperation on the Internet becomes increasingly important. One key issue in this development is the placement of applications in the Internet (cloud computing). Another important issue is the centralized access of cooperative results. Such centralized information may be stored in internet-based knowledge databases. With regard to the above developments, it seems to make sense to have a modelling environment working as a web application connected to a model database. Such a modelling environment should be linked to a strong numeric library. Such a library should contain reliable and very performing algorithms for solving ordinary differential equation (ODE) systems and differential and algebraic equation (DAE) systems. It should also be able to automatically recognize the sparsity and the structure of the Jacobians and to automatically select the best set of algorithms to integrate the differential system. By the help of these measures the computational time should be reduced significantly. In this contribution the MOSAIC modelling environment and the BzzMath library are presented and some main characteristics are described.

1. Introduction

Usually both the modelling software and the software to access company-wide shared databases are installed on the local computer. In the modelling environment MOSAIC (Kuntsche et al., 2009) both these software parts are integrated in one web application. Web applications and cloud computing have several organizational and economic advantages, cf. Linthicum (2009). With respect to modelling, the focus of MOSAIC lies on the creation and shared (re)use of customized models for process systems engineering. The modelling is done in the documentation level and code generation is used to create program code that is suitable for the environment the modeller needs. A considerable improvement of the workflow could be achieved by using the BzzMath numeric library (Buzzi-Ferraris, 2010a) to solve the models on the server. The BzzMath

Please cite this article as: Kuntsche S., Manenti F., Buzzi-Ferraris G. and Wozny G., 2011, Modelling in the documentation level using mosaic and numerical libraries, Chemical Engineering Transactions, 24, 343-348
DOI: 10.3303/CET1124058

Library is a comprehensive tool covering several fields of numerical analysis. For the specific case of interest, it includes a set of algorithms to integrate differential and differential-algebraic systems particularly efficient according to their stiffness, dimension, sparsity, and Jacobian structure, if any.

1.1 Related work

Symbolic programming of computers has a long tradition, see e.g. Klerer and May (1965). An interesting approach for documentation-level modelling in Microsoft Office documents is presented by Alloula et al. (2010). Code generation from symbolic expressions has been considered by e.g. Klerer (1992), see also Kajler and Soiffer (1998). Several projects have been dedicated to modelling in the meta level in combination with code generation. Bogusch et al. (2001) presented a non-internet modelling environment that provided many modelling assistance features and allowed code generation for the two tools gPROMS and Speedup. Westerweele and Laurens (2008) presented a non-internet modelling tool that provides code generation into many languages.

2. The MOSAIC modelling environment

2.1 The literature aspect of documentation-level modelling

One aspect of modelling in the documentation level is the orientation to the representation of equations in papers or documents for reference and communication. In such documents the equations are very often represented in two-dimensional symbolic form and the variable names follow a certain systematic, e.g. a letter 'x' on the base line might stand for molar fraction and the superscript letters 'L' and 'V' adjoined to a given 'x' specify if the variable is a liquid or a vapour molar fraction. This technique is widely used in the documentation of models since it allows for short formulations and clearly understandable variable names. Usually the meaning of the symbols is explained in a separate section that can be called 'notation' or 'nomenclature'. This 'notation' section is especially important since different naming conventions exist, e.g. with 'x' and 'y' for liquid and vapour molar fraction respectively. In MOSAIC the modelling is done in the spirit of this situation by three peculiarities: First the equations are entered in two-dimensional symbolic form. Second the variable names may contain superscripted and subscripted symbols (countable subscripted symbols represent indices). Third the ‘notation’ is introduced as a mandatory model element that works as distinguishing property during reuse. The first of the above items is well established (cf. Kajler and Soiffer, 1998) especially but not only in algebra tools. The second item is rarely provided and the third item is not at all considered in the current standards. The definition of equation systems using the above documentation-based approaches has the advantage that it is both computer-processable and human-readable with a very small visual and conceptual offset between published and implemented model equations. This does not only reduce the introduction of errors and the effort of error finding during the implementation. It also allows the generation of documentation in no time, since all necessary information is already contained in the model.

2.2 Reuse aspects in MOSAIC

The beneficial effects of reusing model parts are well known and are discussed e.g. by Eggersmann et al. (2004). Modelling in the documentation level as it is done in MOSAIC has several reuse aspects. The first of these reuse aspects is that equations and equation systems are created only once and then used to produce program code in several languages. Thus, program code can be produced in the language that is needed by the modeller. Creating the models in a meta level (such as the documentation level) in combination with adaptable code generation allows the user to reuse equations, and even large equation systems in different languages and for different purposes.

The second aspect of reuse is the modular and repeated use of the created model elements (such as equations, equation systems and notations) in many different contexts. The modelling concept in MOSAIC is based on creating modular model elements that are individually stored in XML files. Thus, each equation is stored in a separate XML file. An equation system is also stored in a separate XML file and contains references to the files containing the equations that should be used. Another class of such model elements is the notation. Both equations and equation systems must reference the XML file of the notation that shall be used to explain the contained symbols. Each equation or equation system can be (re)used in other equation systems, as shown in figure 1. There are other model elements, which cannot be covered here. Of course having to create every model element individually takes more time in the short term. However, the resulting high modularity, which notably supports the reuse model elements, has proven to speed up the modelling in the long term.

3. The BzzMath numeric library

The BzzMath numerical library is currently used in MOSAIC to solve non-linear algebraic equation systems and differential algebraic equation systems. Comprehensible output is generated that allows assessing the convergence quality and the reliability of the results and the library has proven to be robust and to provide reliable results not only on these topics. It provides algorithms and methods to tackle problems of different nature.

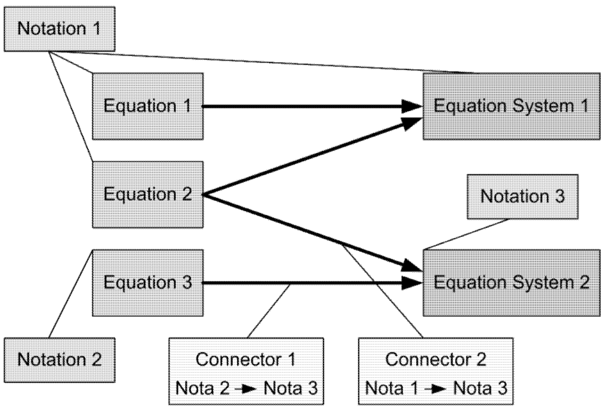

Fig. 1: Modular modelling and reuse in MOSAIC.

To do so, the library has a very performing kernel for linear algebra (Buzzi-Ferraris, 2010b) that "cannot be overstated in modern scientific computing" (Gill et al., 1991). The library is not only predisposed to solve linear systems, but even to automatically adopt the most performing algorithm in accordance with the type of system to be solved. Being thought in an object-oriented way, objects belonging to BzzMath can easily identify and, if possible, exploit matrix sparsity and structure of linear systems and openMP directives for shared memory machines, which results in a considerable gain of performance (Buzzi-Ferraris and Manenti, 2010a). The library includes specific algorithms and criteria for parameter estimations, outlier detections, rival model discriminations, and design of experiments based on very robust algorithms to solve linear and nonlinear regression problems (Buzzi-Ferraris and Manenti, 2010b). The same algorithms allow identifying with a certain accuracy the existence of common problems in the regression field such as masking/swamping effects, heteroscedasticity, multicollinearity, parameter correlations, gross errors, and influential observation. Numerically robust and efficient algorithms are implemented to tackle nonlinear systems and optimization problems. Optimizers are validated by the field in many industrial applications such as the online plant optimization of an oil refinery sulphur recovery unit (Signor et al., 2010) and the real-time dynamic optimization of a large-scale polymer plant (Manenti and Rovaglio, 2008). In addition, certain ad hoc solvers to tackle partially structured DAE systems, which are typical for process control and process systems engineering, have been recently introduced in the library and their superior performances have been validated on industrial cases (Manenti et al., 2009). In this context and looking forward to future developments, we plan to extend MOSAIC so that it is possible to describe optimization and parameter estimation problems. The BzzMath library offers the corresponding numerical algorithms and is thus predestined as standard background environment for calculations on the modelling server.

4. Example

The model of a distillation column has been created in MOSAIC. The equation system that was assembled from single reusable equations is presented in the MOSAIC user interface as shown in figure 2. The proximity to the presentation in the literature is obvious. To further demonstrate the aspects of program code generation and documentation output, the equation of the phase equilibrium (1) is included as an equation object below. Usually MathType (www.dessci.com/mathtype) would have been used to create the formula within the Microsoft Word document. However, equation (1) has just been copied as a latex string out of MOSAIC and directly pasted

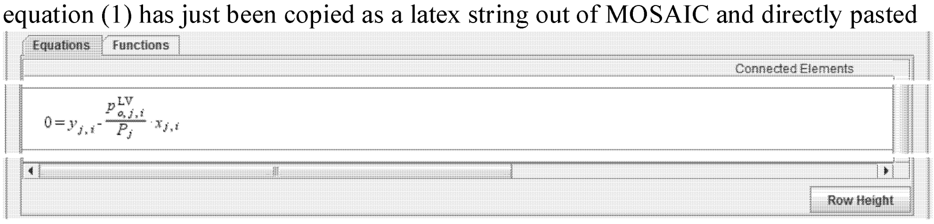

Fig. 2: Equation as it is represented in MOSAIC

into MathType, which resulted without further modifications into the expression displayed here.

$$0 = y_{j,i} - \frac{p_{o,j,i}^{LV}}{P_j} \cdot x_{j,i} \qquad (1)$$

The inverse process would also work. More precisely it would be possible to change the formula in MathType, obtain a latex string and hand it over to MOSAIC. The formula expression can be used directly if all contained symbols are described be the specified notation and if the expression follows the MOSAIC convention for mathematic expressions. Due to the indices ‘i’ and ‘j’ contained in (1) the equation is translated into many instances. The C++ code for the instance ‘i=1’ and ‘j=1’ is shown here:

```
f[15] = 0.0-(e0_y_j1_i1-(e0_p_LV_o_j1_i1)/(e0_P_j1)*e0_x_j1_i1);
```
(2)

The full generated code contains solver calls and file output specifications. It can be executed on the server upon a button click in the MOSAIC user interface. The results for non-linear algebraic system are presented in a table. The plot of the results obtained by the BzzMath library for the above example is given in figure 3.

5. Conclusion

The tool MOSAIC including the concepts presented in this contribution is not just an idea but an implemented software (www.mosaic-modeling.de), which has up to now been used in about 30 modeling projects at the TU-Berlin. Currently, ODE, DAE and AE systems are supported by MOSAIC. It is also possible to define functions. The BzzMath Library is used as a solver for all types of problems supported by MOSAIC. Code generation is used to create full C++ programs that include the BzzMath Library. These programs are compiled and executed on the modeling server machine. The results of these executed numeric programs are send back to the MOSAIC modeling environment (running on the client computer) to be viewed or used as numerical starting values for similar problems. The MOSAIC models can also be translated into other languages.

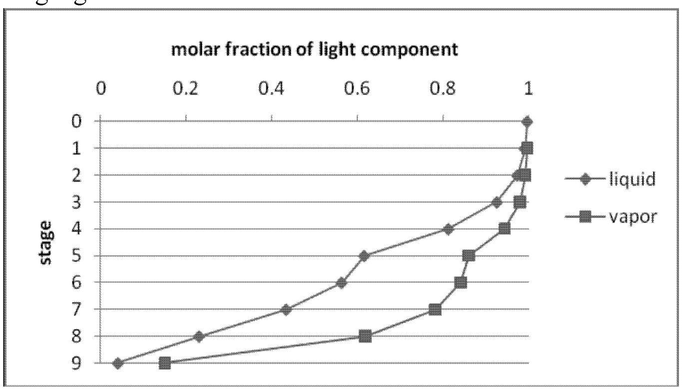

Fig. 3: Plot of results obtained by BzzMath Library with code generated by MOSAIC

This work is supported by the Cluster of Excellence 'Unifying Concepts in Catalysis' coordinated by the TU Berlin and funded by DFG (Deutsche Forschungsgemeinschaft).

References

Alloula, K., Belaud, J.-P. and Le Lann, J.-M., 2010, Solving CAPE models from Microsoft Office applications, Comp Aided Chem Eng, 28, 679-684

Bogusch, R., Lohmann, B., and Marquardt, W., 2001, Computer-aided process modeling with MODKIT, Comp Chem Eng, vol. 25, 963–995

Buzzi-Ferraris, G. 2010a. BzzMath: Numerical library in C++. Politecnico di Milano, http://chem.polimi.it/homes/gbuzzi

Buzzi-Ferraris, G., 2010b, New trends in building numerical programs. Computers and Chemical Engineering doi:10.1016/j.compchemeng.2010.07.004

Buzzi-Ferraris, G., & Manenti, F., 2010a, Fundamentals and Linear Algebra for the Chemical Engineer: Solving Numerical Problems. Wiley-VCH, Weinheim

Buzzi-Ferraris, G., & Manenti, F., 2010b, Interpolation and Regression Models for the Chemical Engineer: Solving Numerical Problems. Wiley-VCH, Weinheim

Eggersmann, M., v. Wedel, L. and Marquardt, W., 2004, Management and reuse of mathematical models in the industrial design process, Chem Eng Tech, vol. 27, 1, 13–22

Gill, P.E., Murray, W. and Wright, M.H. 1991, Numerical Linear Algebra and Optimization. Vol. 1. Addison-Wesley, Redwood City (CA), USA

Kajler, N. and Soiffer, N., 1998, A Survey of User Interfaces for Computer Algebra, J Symb Comp, 25, 127-159

Klerer, R. J., Klerer, M., and Grossman, F., 1992, A language for automated programming of mathematical applications., Computer Languages, 19, 3, 169–184

Klerer, M. and May, J., 1965, Two-dimensional programming, Proceedings of the 1965, fall joint computer conference, part I, 63-75

Kuntsche, S., Arellano-Garica, H. and Wozny, G., 2009, Web-based object oriented modelling environment for the simulation of chemical processes, Chem Eng Trans, 18, 779-784

Linthicum, D. S., 2009, Cloud computing and soa convergence in your enterprise: a step-by-step guide, Addison-Wesley Longman, Amsterdam

Manenti, F., & Rovaglio, M., 2008, Integrated multilevel optimization in large-scale poly(ethylene terephthalate) plants. Ind & Eng Chem Research 47(1), 92-104

Manenti, F., Dones, I., Buzzi-Ferraris, G., & Preisig, H.A., 2009, Efficient Numerical Solver for Partially Structured Differential and Algebraic Equation Systems. Ind & Eng Chem Res 48(22), 9979-9984

Signor, S., Manenti, F., Grottoli, M.G., Fabbri, P., & Pierucci, S., 2010, Sulfur Recovery Units: Adaptive Simulation and Model Validation on an Industrial Plant. Ind & Eng Chem Res 49(12), 5714-5724

Westerweele, M. and Laurens, J., 2008, Mobatec Modeller - A flexible and transparent tool for building dynamic process models, Comp Aided Chem Eng, vol. 25, 1045–1050