Status and Issues of Implementation of a Cape-Open Thermo Interface in the Polymer Kinetics Package Predici

Dr. Michael Wulkow, CiT GmbH,
Oldenburger Str. 200, Rastede, 26180, Germany, m.wulkow[at]cit-wulkow.de

Over the last decade the simulation package Predici [1] has evolved from a solver for population balances describing molecular weight distributions in polymer reaction kinetics to a comprehensive tool for nearly all kinds of polymer processes. Among various recent modeling issues and requirements the link of Predici simulations to thermodynamic computations has become an interesting topic. Especially for the modeling and simulation of polymerization reactions the incorporation of thermodynamic properties and equilibrium computations leads to particular challenges. The state equations can be more complex and the realization of a phase flash for full chain-length distributions is much more complicated than for standard chemicals, since hundreds of thousands of single species might be involved. In order to provide a general approach for all kind of models, Predici has been extended by the Cape-Open Thermodynamic & Physical Properties interface [2] recently. This interface has been designed to allow for all data and computations necessary within Predici simulations (i.e. selected pure component data, mixture properties and phase computations).

The talk will present the general concept and the results and difficulties arising from tests with two thermodynamic property packages (Multiflash [5] and COCO [6]). In particular the new, advanced but more complicated Cape-Open Thermo Interface 1.1 will be discussed.

In Predici thermodynamic data are mainly used in equations and balances related to

- densities
- molecular weights
- parameters of the heat balance
- phase equilibria.

Additionally modelers might use certain compound properties for the definition of special output written in terms of the built-in Predici script language. Usually only a relative small selection of all principally available data is required, where the state-based computation of phase fractions is the most complicated task.

Since the thermodynamic property tool Multiflash, developed and distributed by the Infochem Computer Services Ltd, London, U.K., includes a robust implementation of the PC-SAFT theory [3], at first Predici has made use of a direct access to Multiflash and the related computations by accessing a dynamic link library (dll) of the package. The advantages of such native calls to a third party program are efficiency, speed and a very comprehensive use of the features of the physical property tool. Among the disadvantages of the dll-approach we have to note the lack of portability to other products, a more complicated error handling and possibly long lists of arguments and parameters in function calls. Nevertheless, based on a dll-interface the results presented in [4] could successfully be obtained in an industrial project.
Obviously, an interface based on an open standard can lead to much more general implementations. Such a standard is provided by the Cape-Open Thermo 1.1 interface. However, some considerations regarding the general architecture had to be made in order to get an efficient and stable handling.

In the context of a complex solver for population balances, where single simulation runs might require minutes or even hours, it is crucial that all calls and results of a linked physical property tool can be checked outside the simulation context and before start-up. At implementation aspects of computation time have to be addressed as well as robust error handling. Since any error at preparation of the thermodynamic system, inside user scripts and during simulation are difficult to

trace back, we decided to write an independent configuration tool called CapeCon, which had to complete several tasks:

- Selection of the thermodynamic system and the actual property package based on a configuration already done and stored within the used physical property tool (e.g. by the Multiflash user interface, which exports all information to a configuration file)

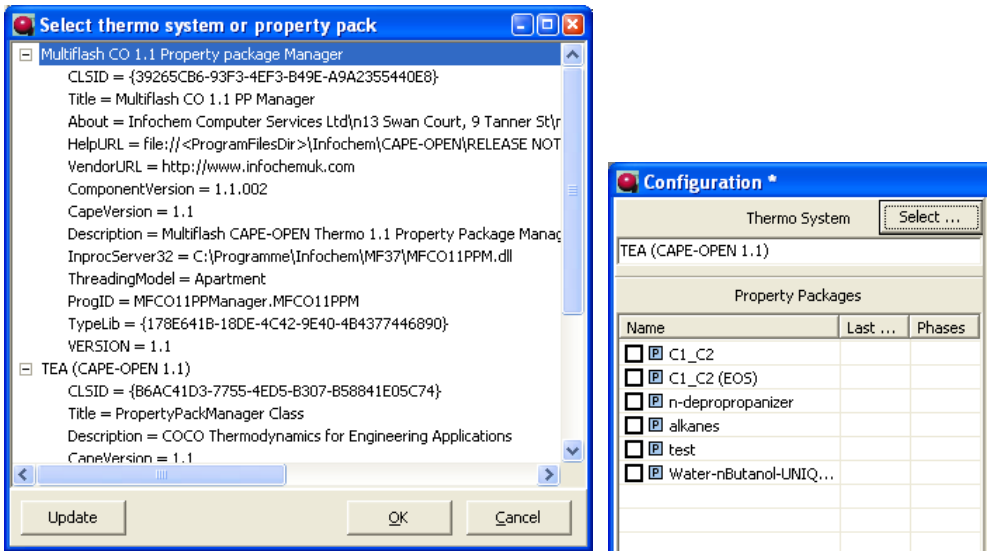

Figure 1: Selection of thermodynamic system in CapeCon

- Evaluation of exemplary flash computations for given temperature, pressure, enthalpy, mixture etc. and comparison to results obtained within the stand-alone property tool directly. This ensures that all settings, units etc. are stored as intended.

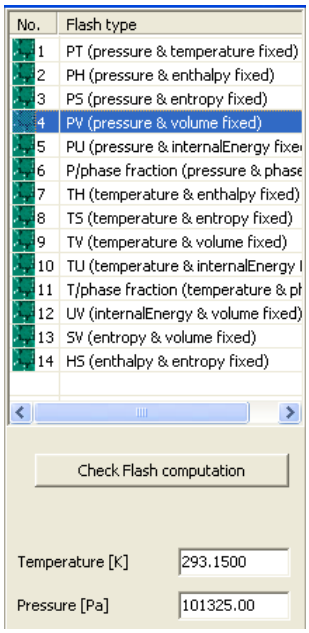

Figure 2: Selection of type of phase equilibrium in CapeCon

- Selection of all thermodynamic properties required for the simulation of a kinetic model. This is particularly important, since with Cape-Open a list of all such properties is transferred to the property tool as one list and then processed. The longer the list, the more effort and data transfer might occur. Therefore it makes sense to restrict the selection to the necessary calls beforehand.

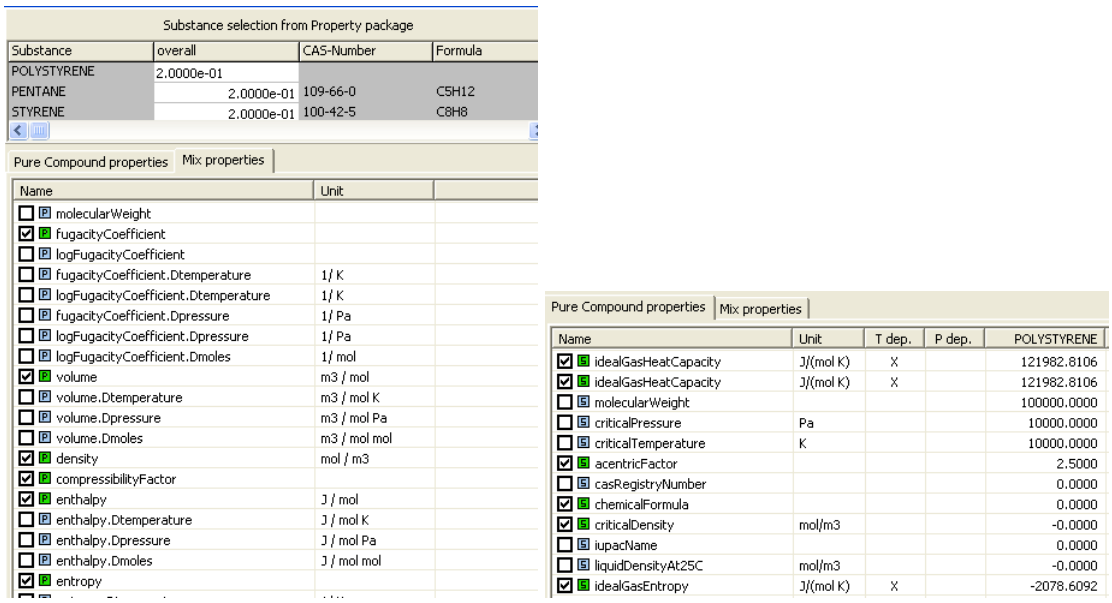

Figure 3: Selection of mixture properties (left) and pure compound properties (right) in CapeCon

- The selection of property system and actual package as well as flash type and all selected pure compound and mixture properties is stored in a small XML file, which will finally be assigned to a Predici model.
- CapeCon can always reload the configuration file and thus allows for changes and updates of the selections. A minimal configuration file for flash computations only has to include a few lines – the chosen system and package.

Due to these functionalities CapeCon might also serve as a Cape-Open tester.

The approach of an independent configuration and test tool can only ensure reliable results, if during the kinetic simulation itself the same code basis is used as for the configuration tool. All changes of the C++ code have to be effective both in CapeCon and in Predici. Therefore we used a project structure in the CodeGear C++-Builder 2007 such that the compilation of the source code alternatively leads to the CapeCon GUI or to a dll to be called by Predici. Updates are thus automatically available in both programs and usually do not require the delivery of a new Predici basis software.

The interaction between Cape-Open client (here: CapeCon and Predici) and server (the physical property tool, e.g. Multiflash) is based on a „material object“ (MO). The client program has to provide an object class and member functions derived from the interface type library. In order to leave Predici independent of the concrete implementation of the material object, we developed a so-called „storage object“. The storage object works as an interface class between the CapeCon dll and Predici. By that CapeCon is independent of native Predici C++ classes and vice versa. The setup and test of this chain of interfaces is a bit more complicated, but pays off when changes have to be made.

The first implementation of the Predici Cape-Open interface had been done two years ago based on interface specification 1.0. At this time only a few selected flash types were required (TV, TVF, TP) and no pure compound properties could be accessed. The flow of interface calls was relatively simple. After assignment of the material object and call of CalcEquilibrium the MO was asked for some properties and finally called to store back some results.

Packages based on version 1.1 could be used in the 1.0 style too, but it was apparent, that then the new, extended features of the interface could not be used. It took a while to recognize the need to implement a new interface using the “PropertyPackageManager” concept. Also for pure compound properties, the phase settings and the equilibrium computation (unexpected) additional effort compared to the much simpler version 1.0 was necessary. Tracing all interface calls also revealed much more intermediate calls from the property tools than before and a different order and chain of commands for e.g., Multiflash and COCO. Also the definition of some input data had

to be changed or extended. This is surely a drawback of the new interface and should be prevented in future updates of the interface specification.

Based on the configuration done by use of CapeCon, the Cape-Open interface of a thermodynamic property tool can principally be accessed by Predici. However, here another important design question arises: How to relate input and output of the thermodynamic computation to state-variables and parameters of a Predici model? In a Predici model all notations and names are up to the user. Of course, one could require that the compound names (or their aliases) of a Predici model have to match the names of a Cape-Open configuration. But this would be too restrictive. At first, a Predici model does not only consist of the basic substances, but there are intermediate products and so-called counter species (both types possibly present in a series of reactors). Moreover, „the“ polymer of a Predici reactor is usually split into types like radical, dead, living, dormant, cross-linked and/or distinguished by end-groups. For a flash computation the total polymer has to be summarized to one mass instead (or only selected fractions of the whole polymer distribution have to be chosen). Therefore we decided to avoid any kind of automatic assignment of Predici and Cape-Open compounds, but applied Predici's internal script language instead. The scripting allows access to all variables of a Predici system and is used for user-defined output, the definition of additional differential equations, state-dependent reaction rates (e.g. in gel effect models) and much more.

For the Cape-Open interface the script language in Predici has been extended by the commands

- co_get
- co_set
- co_action
- co_pp.

These calls are similar to the basic Cape-Open calls to the material object. For example, in order to set a certain fraction, using co_set(”property“, ”phase“, “component”, value) transfers a certain value to the storage object, which is then prepared to be called by the thermodynamic tool.

By that, inside Predici one can easily initialize and perform Cape-Open computations on a well-tested basis provided by the configuration in CapeCon.

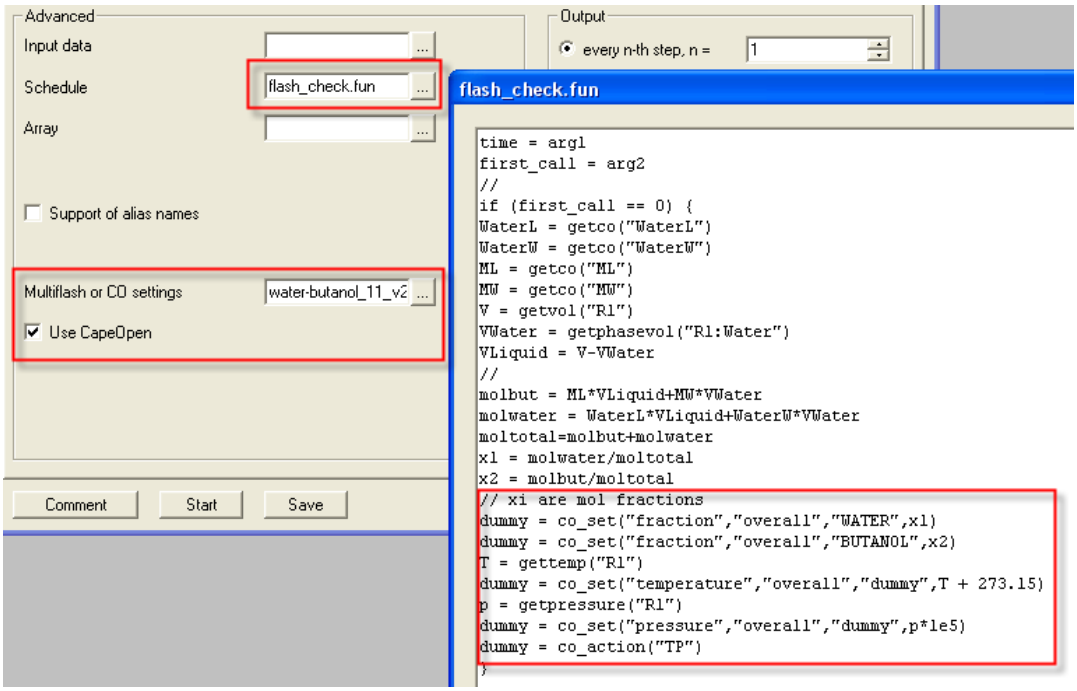

Figure 4: Assignment of configuration file (generated by CapeCon) and user-script executing the equilibrium computation in Predici

There are some extensions and planned activities, e.g.

- implementation of graphical output in CapeCon
- scripting in CapeCon in order to perform direct tests with the Predici script language
- more tests with different physical property tools.

References:

[1] M. Wulkow, *Computer Aided Modeling of Polymer Reaction Engineering - The Status of Predici, 1 – Simulation*, Macromol. React. Eng. 2008, 2 (Issue 6), in print; published online: 04.09.2008, DOI: 10.1002/mren.200800024

[2] *Thermodynamic and Physical Properties v1.1*, http://www.colan.org/Spec%2011/CO_Thermo_1.1_Specification_222.pdf

[3] Gross, J. and Sadowski, G, *Perturbed-Chain SAFT: An Equation of State Based on a Perturbation Theory for Chain Molecules*, *Ind.Eng.Chem.Res.,* **2001***, 40*, 1244

[4] Hungenberg, K.; Nieken, U.; Zöllner, K. & Szekely, *A. Modeling Safety Aspects of Styrene Polymerization Processes*, *Ind. Eng. Chem. Res.,* **2005***, 44*, 2518-2524

[5] Multiflash, http://www.infochemuk.com/product/products.htm

[6] COCO – Cape Open to Cape Open, http://www.amsterchem.com/coco.html