FLOWSHEET MONITORING – HOW, WHAT AND WHY?

Dr. Jasper van Baten, AmsterCHEM, Las Rozas, Spain

Summary

Flowsheet Monitoring is the ability to access all elements in a flowsheet without interfering with the flowsheet. Flowsheet Monitoring Components are software components that can be plugged into a flowsheet and have access to all thermodynamic property calculation methods, all streams and all Unit Operations. Requirements for the interface definitions are discussed and examples of applications are given. A CAPE-OPEN interface definition is proposed, and there is an evaluation of the implications of providing support for this interface in terms of implementation effort and complexity.

Introduction

Steady state flowsheet simulation is a computer technique used by process engineers for modeling entire (chemical) processes or parts thereof. Such flowsheet simulations typically consist of several process model units (unit operations) linked together by material and energy streams in combination with thermodynamic and physical property models. Flowsheets typically form a model of a complete process to - for example - assess process design, train process operators, and to close mass and energy balances. The unit operation models are combined in a Process Flow Diagram, showing connections between unit operations. An example of a flowsheet is presented in Figure 1[1,2]:
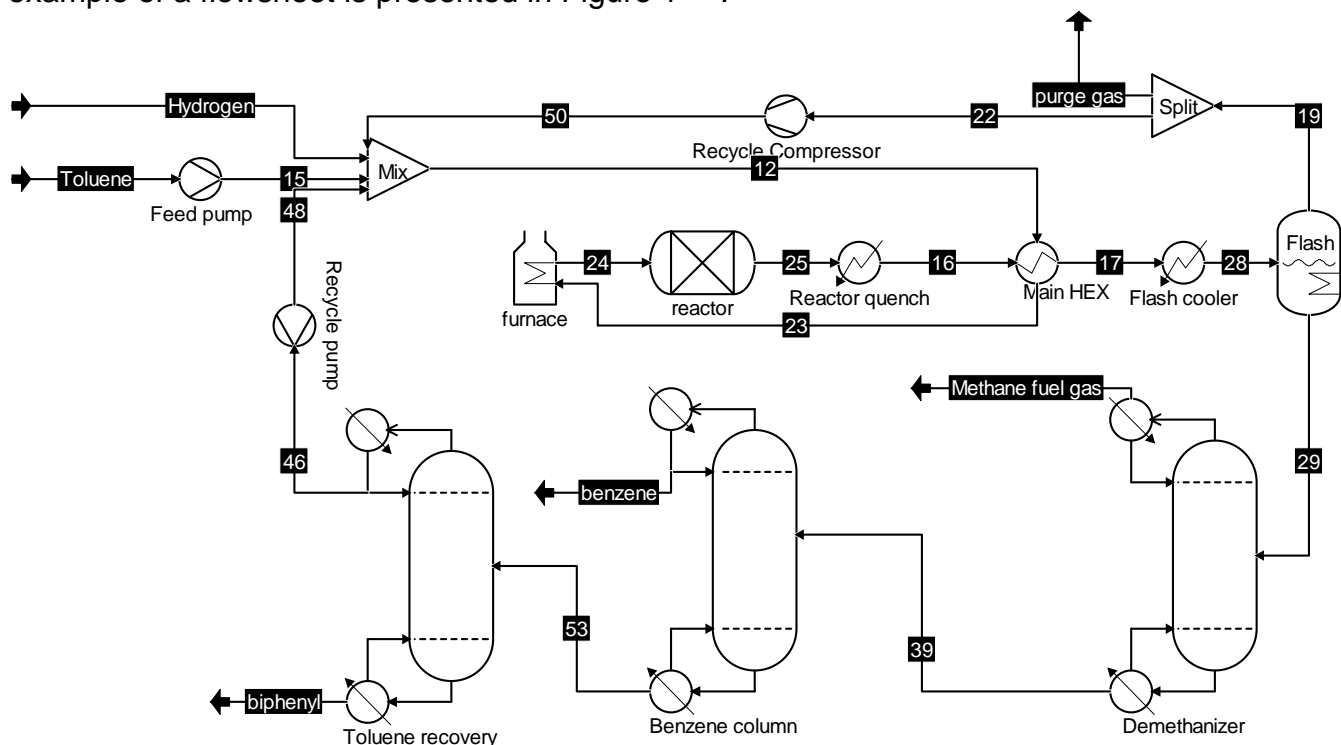

Figure 1: example of a process flowsheet, showing the dealkylation of toluene to benzene with hydrogen, as available from http://www.cocosimulator.org/index_sample.html.

CAPE-OPEN[3, 4] is a collection of interface standard definitions that facilitates exchange of thermodynamic and physical property models and unit operations (Process Modeling Clients, PMCs) between available simulation applications (Process Modeling Environments, or PMEs).

A typical simulation environment (PME) supporting Process Modeling Clients, or PMCs, is sketched in Figure 2:
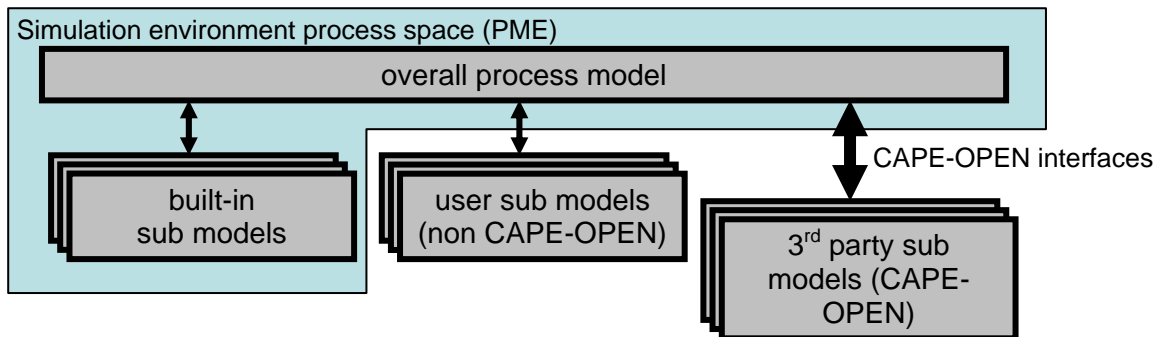

Figure 2: typical setup for simulation environments supporting CAPE-OPEN models.

Over the last years, CAPE-OPEN has already proven its effectiveness in allowing exchange of process sub models between different simulation environments; all major steady state flowsheet environments have CAPE-OPEN sockets to plug in third party models and many CAPE-OPEN compliant sub models have been designed to run in a variety of simulation environments.

For one important class of applications, no CAPE-OPEN interface is currently available. These are applications that require access to multiple parts of the flowsheet or its thermodynamic models, without interfering with the flowsheet itself. These flowsheet monitoring applications generally perform post-processing on flowsheet calculations, such as flowsheet evaluation and validation.

For some of these applications, solutions have been devised employing unit operations that have the mere tasks of collecting flowsheet information without actually performing operations on the flowsheet streams. This approach has severe drawbacks: if information is required from various places of the flowsheet this includes inserting multiple monitoring unit operations, convoluting the actual flow diagram. Moreover, each unit operation must perform an equilibrium calculation at each of the outlets. Unit operations performing monitoring inside recycles will as a result heavily contribute to the calculation load of the simulation.

A generic solution is presented for models to gain access to all parts of the flowsheet, without interfering with the flowsheet (in a read-only manner). These models will be called Flowsheet Monitoring Objects, and a proposal is made for a CAPE-OPEN interface definition to facilitate these models.

Flowsheet monitoring

For an Flowsheet Monitoring Object to be able to access all information in a given flowsheet, the following will have to be satisfied:

- access to the various available thermodynamic and physical property models;
- ability to perform physical and thermodynamic property calculations;

- access to a collection describing all streams (energy, information, …) in the flowsheet;
- for each of the streams, access to all its properties (pressure, temperature, composition, flow, phase equilibrium, …);
- access to a collection describing all unit operations present in the flowsheet;
- ability to determine connectivity of unit operations (determine which unit operations are connected by which streams);
- access to all public parameters exposed by a unit operation;
- ability to perform event driven calculations (*viz.* automatically performing a calculation when a new solution of the flowsheet is found).

From a modular design point of view arises the requirement to – where possible – reuse the existing interfaces already defined by CAPE-OPEN.

Example applications that benefit from monitoring interfaces are:
- applications that perform overall flowsheet calculations, including component-, mass-, and energy balances. This category includes environmental monitoring applications such as WAR[5, 6];
- applications that perform generic thermodynamic and physical property calculations in the context of a flowsheet, without specific flowsheet structure information. These applications benefit from the ability to perform calculations without having to set up the relatively complex CAPE-OPEN material object implementation, and can immediately use the chemical compounds that have already been selected by the flowsheet user. Example applications include phase envelope calculators[7] and residue curve calculators;
- applications that perform thermodynamic and physical property calculations in the context of specific flowsheet elements. These application perform post-processing and validation at the found solution. Example applications include wax formation calculations or hydrate deposition calculations in pipes represented by specific flowsheet streams[8];
- applications that conduct performance rating calculations for a particular class of unit operation from within the flowsheet by accessing stream data and public parameters[9];
- applications that analyse the structure of the flowsheet and the specific coupling of flowsheet elements, for example for heat integration by Pinch technology[10];
- archiving applications that store each found solution of a particular flowsheet to a database (goal could be generation of short-cut models by linearizing between previously found rigorous solutions);
- model-based control applications that perform a simulation of the current state of an actual physical process and want to report new control targets to external application. A separate mechanism is required for this class of applications to provide the simulation with measurements of the current state of the process (e.g. feed conditions),

Monitoring architecture

A Flowsheet Monitoring Object is a software component (PMC) that can be loaded into a flowsheet simulation environment (PME). As such, like unit operation PMCs, it will implement the generic CAPE-OPEN interfaces ICapeUtilities[11] and ICapeIdentification[12], and optionally interfaces allowing for storing and loading[13]. The ICapeIdentification interface allows the object to be identified by name and description. The ICapeUtilities interface allows for general operations such as user activation of the object (Edit method), initialization and termination,

exposing of a collection parameters[14], and, very important, access to the Simulation Context in which the monitoring object is running, via ICapeSimulationContext[15]. The Simulation Context represents the context in which the simulation takes place, and can be queried for simulation specific information.

Via the Simulation Context, the Flowsheet Monitoring Object already can have access to all available thermodynamic and physical property calculation methods, by obtaining the ICapeMaterialTemplateSystem interface, through which it can access the available ICapeThermoMaterialTemplate[16] objects that are available within the simulation.

The Simulation Context will expose a new interface: *ICapeFlowsheetMonitoring*. Apart from inquiring the status of the flowsheet (validation status and whether the flowsheet is currently in a solved status), the ICapeFlowsheetMonitoring interface will expose two collections[17]: the collection of streams (GetStreamCollection) and the collection of unit operations (GetUnitOperationCollection). After obtaining this interface, the Flowsheet Monitoring Object can enumerate all (material-, energy- and information-) streams available in the flowsheet. The regular interfaces that define streams[16-19] can now be used by the Flowsheet Monitoring Object not only to access all stream properties, but also to duplicate material objects and perform any physical and thermodynamic property calculations. The Flowsheet Monitoring Object can now also enumerate the Unit Operations[18], and access its public parameters. For each unit operation, it can enumerate the Unit Operation Ports, and obtain the attached stream. It can identify the attached stream by name, and by comparison with the names of the enumerated streams, it can determine the connectivity of the flowsheet objects.
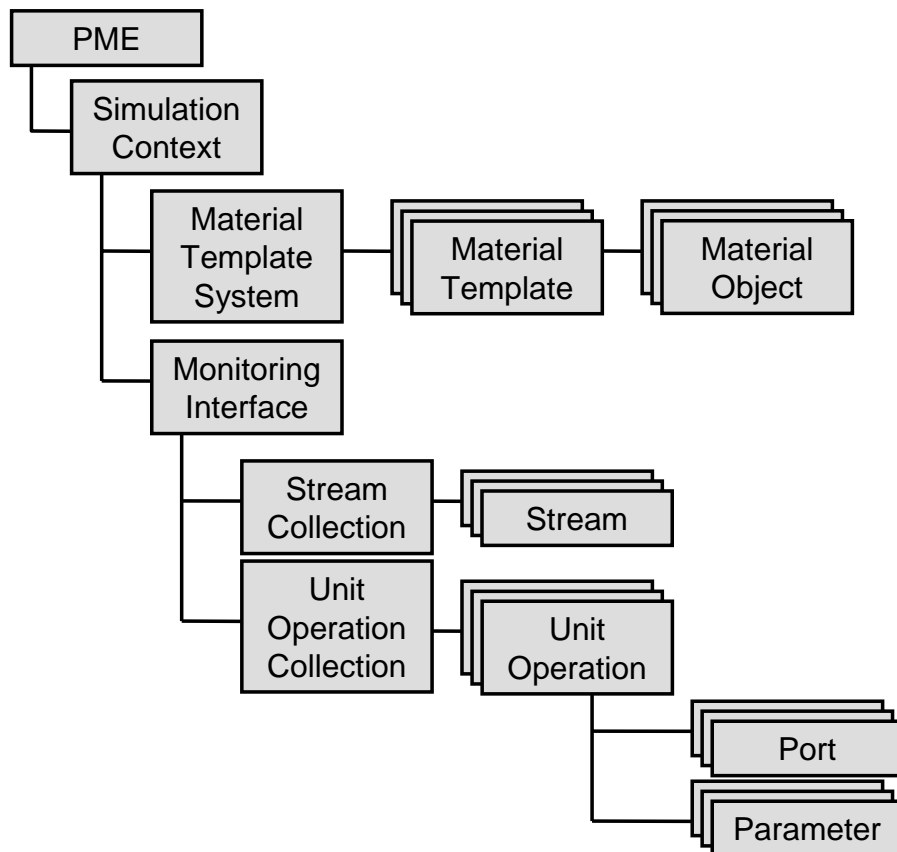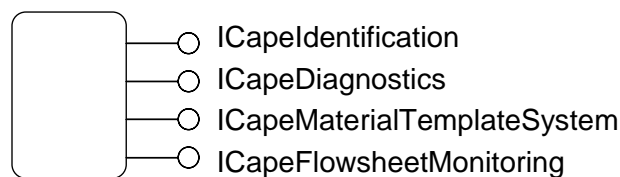

Figure 3: hierarchy of objects accessible by Flowsheet Monitoring Objects via the Simulation Context. Each of the objects shown is identified by name using ICapeIdentification.
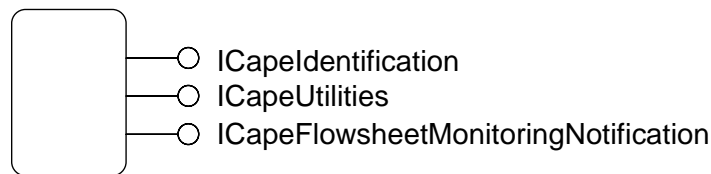
The above satisfies all requirements for flowsheet monitoring, as stated earlier, except for the requirement to perform event driven calculations. The only event available from the above discussion is user invocation by means of the Edit method (ICapeUtilities). The examples of online control and storage systems require events that are fired by the PME and handled by the Flowsheet Monitoring Object.

To this purpose, another new interface definition is required: *ICapeFlowsheetMonitoringNotification*. This interface is to be implemented by the Flowsheet Monitoring Object, only if it has a requirement to process notifications. For each of the available notifications, the ICapeFlowsheetMonitoringNotification interface has an entry point that is to be called by the PME. The notifications include changes to the flowsheet structure, changes with respect to the state of a flowsheet component, and changes in the solution state of the flowsheet. Suggested notifications are:

- a Unit Operation has been added to the flowsheet;
- a Unit Operation has been removed from the flowsheet;
- the state of a Unit Operation has changed (to be called when a parameter value changes, when a port is connected or disconnected, after calling Edit on a Unit Operation and upon completion of Calculate of the Unit Operation);
- a stream has been added to the flowsheet;
- a stream has been removed from the flowsheet;
- the state of a stream has changed (to be called when the user has modified the content of a stream or when the stream has been calculated);
- a Unit Operation was renamed (this notification is important as flowsheet elements are identified by name);
- a stream was renamed (this notification is important as flowsheet elements are identified by name);
- the flowsheet has been solved (a new solution is available).

ICapeIdentification
ICapeDiagnostics
ICapeMaterialTemplateSystem
ICapeFlowsheetMonitoring

Simulation Context

ICapeIdentification
ICapeUtilities
ICapeFlowsheetMonitoringNotification

Flowsheet Monitoring Object

Figure 4: Interfaces exposed by Simulation Context (PME)
and by Flowsheet Monitoring Object

The process of selecting and adding a Flowsheet Monitoring Object to the simulation is exactly analogous to adding a Unit Operation to a simulation. This requires definition of a new CAPE-OPEN Category ID for Flowsheet Monitoring Objects[32]. Whereas Unit Operations are generally present in the Process Flow Diagram, it is envisioned that Flowsheet Monitoring Objects are not. Depending on the simulation environment application front-end, Flowsheet Monitoring Objects could be available from a Plug-in or Add-in menu.

It is important to stress that although the Flowsheet Monitoring Objects will have access to interfaces that allow modification of the flowsheet elements, the access to these elements should be restricted to read-only. For example, parameter values can be obtained, but should not be set. Flowsheet Monitoring Objects should not connect or disconnect streams from Unit Operations. Thermodynamic and Physical Property calculations can be performed, but should not be performed on the original Material Objects, but rather on duplicates thereof. Modifications made by Flowsheet Monitoring Objects to the flowsheet elements cannot be foreseen by the PME, and may disrupt normal operations of the flowsheet.

Requirements on the PME

CAPE-OPEN is a collection of many interface definitions, some of which are more widely implemented than others. The success of a particular set of CAPE-OPEN interfaces can be measured by the amount of implementations that are around. Specifically the Unit Operation and Thermodynamics and Physical Property interfaces are very successful, and socket implementations are available in all main-stream process simulators.

For a new CAPE-OPEN interface to be successful depends on two factors: the demand by model builders and end-users for use of the interfaces, and the willingness of simulation software vendors to provide socket implementations. For the former, the example applications described earlier show that there is a demand. For the latter, it is important that the monitoring interfaces are easy to implement.

To support Flowsheet Monitoring Objects, the following is required for PMEs to implement:

- *a (menu-) structure for the users to load, unload and maintain Flowsheet Monitoring Objects*: this is considered an easy task. The objects that are used for selecting and loading Unit Operations and Physical Property Packages can readily be re-used for selecting and loading Flowsheet Monitoring Objects;
- *storing and loading the state of Flowsheet Monitoring Objects between sessions*: as this is exactly equivalent to Unit Operations and Physical Property Packages, also this is considered an easy task;
- *support for the ICapeSimulationContext interface*: this is present in most simulation environments already;
- *support for the ICapeFlowsheetMonitoring interface and the stream- and Unit Operation collections*: implementation of these interfaces is very straight-forward and it is not expected that this part will cause much effort;
- *exposing streams via the stream collection as CAPE-OPEN interfaces*: for material streams, most simulation environments already contain the proper interface implementations, as these are the same implementations that are used to connect material streams to Unit Operations. The implementation of energy and information streams is not yet wide-spread, but to support this functionality for Unit Operations means that the implementation can be re-used for Flowsheet Monitoring Objects;

- *exposing Unit Operations via the unit operation collection as CAPE-OPEN interfaces*: this part is straight-forward for Unit Operations that have been inserted as CAPE-OPEN components. For built-in or user Unit Operations (Figure 2), it requires a CAPE-OPEN shell around these Unit Operations. This task is not considered complex, but requires considerable implementation effort;
- *support for calling event notifications on ICapeFlowsheetMonitoringNotification*: this task is considered straight forward for simulation environments that already have an internal event notification hierarchy. For simulation environments that do not already keep track of changes in state of Unit Operations and streams, this task will be considerable effort and requires adaption of all parts of code that cause such changes to occur. Complexity of this task depends on the internal hierarchy of the simulation environment.

Current status

Support for the ICapeFlowsheetMonitoring interface is presently implemented in the COFE flowsheeting environment of COCO – the CAPE-OPEN to CAPE-OPEN simulation environment, starting from version 1.12, freely available from http://www.cocosimulator.org/. This allows for loading and testing of Flowsheet Monitoring Objects that do not require notifications. The installer will also provide a sample Flowsheet Monitoring Object TERNYP. The full CAPE-OPEN Flowsheet Monitoring interface proposal is available from http://www.cocosimulator.org/down.php?dl=Proposal_for_flowsheet_monitoring_interface.pdf. The preliminary IDL (Interface Definition Language) document describing the ICapeFlowsheetMonitoring interface (but not yet the ICapeFlowsheetMonitoringNotification interface) is available from http://www.cocosimulator.org/down.php?dl=CAPE-OPEN-Monitoring-interface.idl.

Conclusions

The presently proposed interface definitions for CAPE-OPEN Flowsheet Monitoring Objects will provide a useful context for a class of flowsheet simulation sub models that is not covered by the Thermodynamic and Physical Properties interfaces or the Unit Operations interfaces. A framework is presented that allows software components full (but read-only) access to all elements that make up the flowsheet. The framework is largely based on interface definitions that already exist and are implemented in most main-stream simulation environments. A partial implementation of the framework is available in COCO Simulator, and has successfully been tested. The example applications show that there is a demand for the proposed interfaces. The success will depend on the willingness of simulation software vendors to implement support. Here, most implementation effort is expected in exposing non-CAPE-OPEN unit operations through CAPE-OPEN interfaces, and providing support for the notification interface.

References

1. Douglas, J.M., "Conceptual Design of Chemical Processes", McGrawHill, 1988
2. Douglas, J.M. (1985), "A hierarchical decision procedure for process synthesis". AIChE Journal, vol. 31 no 3 pp 353-362.
3. Barrett, William M. and Yang, Jun (2005), "Development of a chemical process modeling environment based on CAPE-OPEN interface standards and the Microsoft .NET framework". *Computers and Chemical Engineering* 30, pp. 191-201
4. The CAPE-OPEN Laboratories Network, CO-LaN, http://www.colan.org/
5. Young, D., Scharp, R. and Cabezas, H. (2000), "The waste reduction (WAR) algorithm: environmental impacts, energy consumption, and engineering economics", Waste Management, Vol 20 no 8 pp 605-615.
6. Hilaly, A.K. and Sikdar, S.K. (1994), "A new method for minimizing waste production in manufacturing processes", J Air Waste Manag Assoc vol 44, pp 1303-1308.
7. TERNYP, available from http://www.cocosimulator.org/ as part of the COCO distribution
8. Envisioned by Infochem Computer Services Ltd (http://www.infochemuk.com/)
9. Column rating model, as envisioned by Chemsep (http://www.chemsep.com/)
10. Linnhoff B *et al* (1982), "User Guide on Process Integration for the Efficient Use of Energy", IChemE, Rugby, U.K.
11. Utilities Common Interface, available from http://www.colan.org/index-35.html
12. Identification Common Interface, available from http://www.colan.org/index-35.html
13. Persistence Common Interface, available from http://www.colan.org/index-35.html
14. Parameter Common Interface, available from http://www.colan.org/index-35.html
15. Simulation Context Interfaces, available from http://www.colan.org/index-34.html
16. Thermodynamics and Physical Properties Interfaces, version 1.0, available from http://www.colan.org/index-33.html
17. Collection Common Interfaces, available from http://www.colan.org/index-35.html
18. Unit Operations Interfaces, available from http://www.colan.org/index-33.html
19. Thermodynamics and Physical Properties Interfaces, version 1.1, available from http://www.colan.org/index-37.html
20. Methods & Tools Integrated Guidelines, available from http://www.colan.org/index-32.html

Acknowledgements