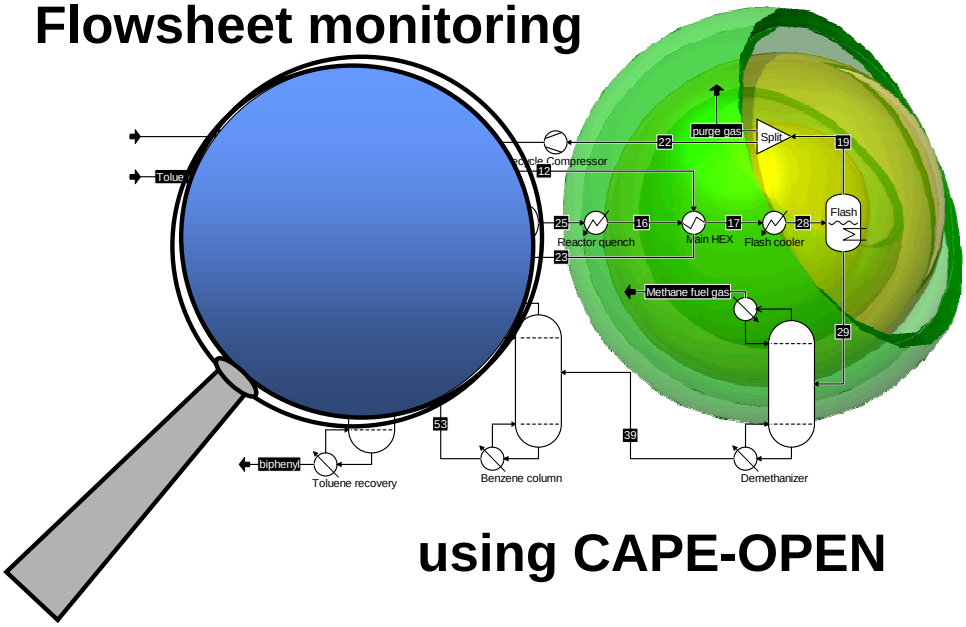
amsterCHEM
tailor-made engineering software solutions

Slide 1

Flowsheet monitoring



using CAPE-OPEN

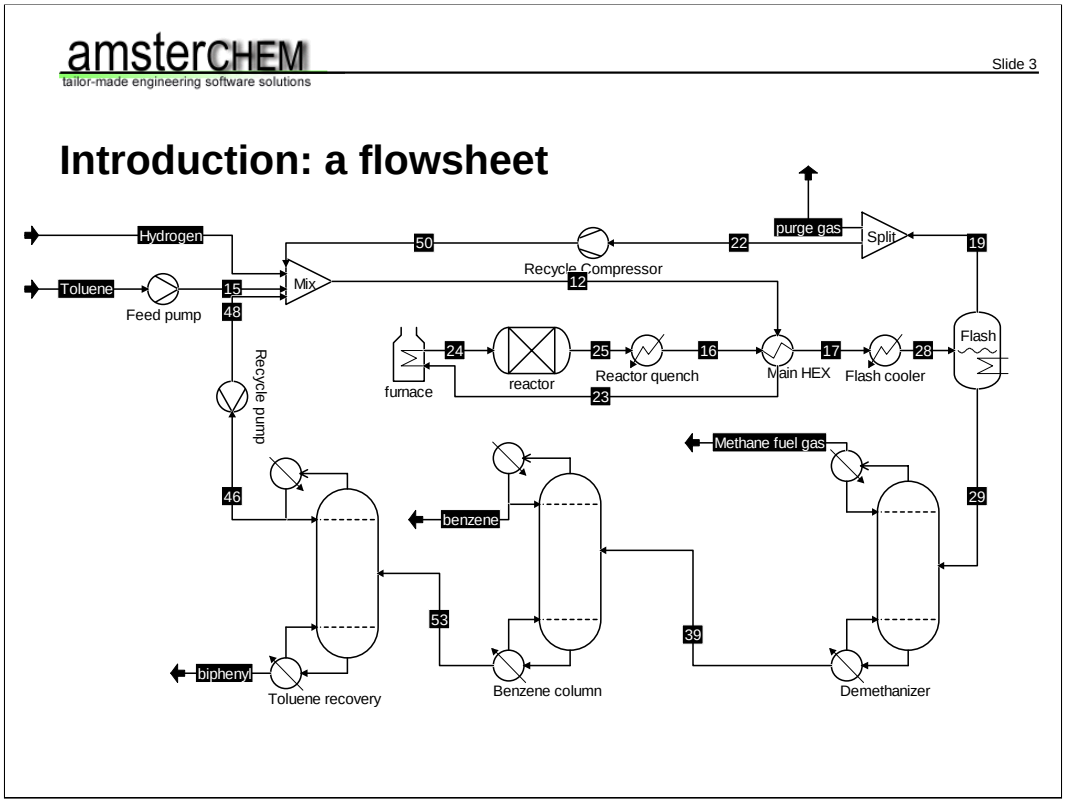
Jasper van Baten, AmsterCHEM

Good morning. My name is Jasper van Baten. I represent AmsterCHEM. Today I will talk about Flowsheet Monitoring. I will explain what I mean when I say Flowsheet Monitoring, what its applications are, and how to go about it in the framework of CAPE-OPEN simulation environments.

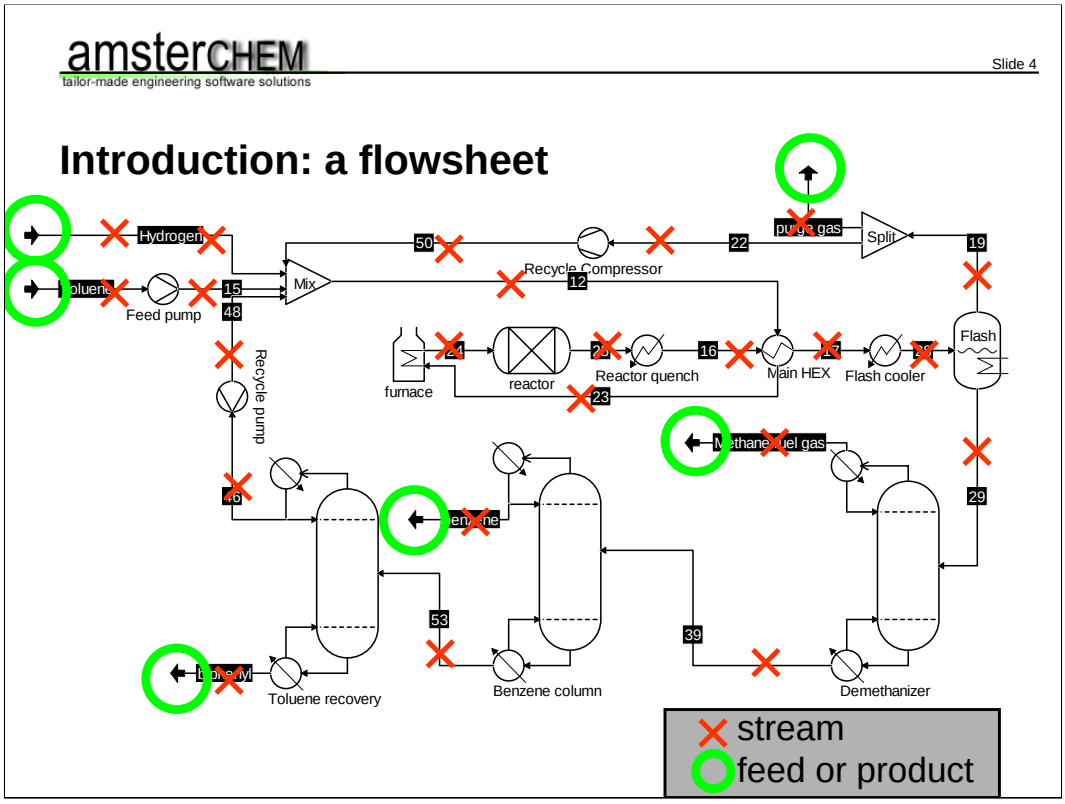
Presentation outline

- Introduction: a flowsheet and its elements
- What is flowsheet monitoring?
- Flowsheet monitoring applications
- Flowsheet monitoring and CAPE-OPEN
- Requirements on PME
- Current status

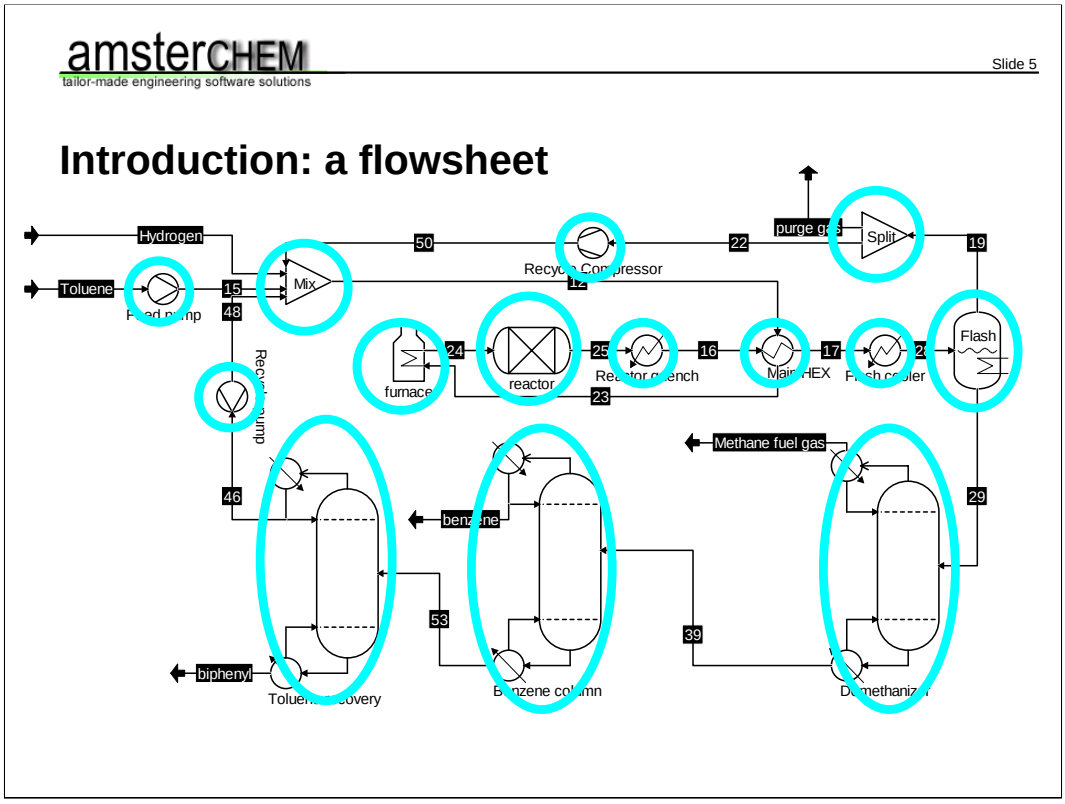
I will start off with an introduction. I will show a flowsheet example, and enumerate the types of modelling elements that are involved. Next, I will explain what flowsheet monitoring is. I will illustrate this with a number of example applications. Then, I will describe how to go about it, and why CAPE-OPEN provides a good platform for a generic implementation. I will proceed with showing what a Process Modelling Environment – or Simulation Environment – must do to allow for CAPE-OPEN flowsheet monitoring objects. I will finish off by showing the current implementation status in COCO's steady state flowsheeting engine COFE.



Let’s start with looking at a flowsheet. The example shown here is the HDA process for the dealkylation of toluene to benzene with hydrogen, available from the cocosimulator.org web site. This is a steady state modular flowsheet, but for the purpose of flowsheet monitoring this does not matter. Flowsheets are typically used to model a complete process or a part thereof. The aim of such a flowsheet is to link unit operations together in such as way that the overall mass and energy balances can be solved, all in a thermodynamically consistent manner. Therefore, underlying all calculations there are one or more thermodynamic software components that take care of thermophysical property calculations and thermodynamic equilibrium calculations. Having identified the thermodynamic system as a flowsheet element, we can proceed with identifying the other flowsheet elements.



We can see that all connections consist of streams, with as special cases: feeds and products to the entire flowsheet. Most streams will carry matter, and are referred to as material streams. In addition to matter, they also carry information about the physical state of the matter, such as temperature and pressure. Not all streams need to be material streams, energy streams and generic information streams are also possible.



Apart from the thermodynamic system and streams, we have unit operations. The unit operations represent physical equipment, and are connected by streams. Unit operation calculate the relation between streams that go in and streams that go out. This calculation may need to be done repeatedly if a unit operation is present in a recycle, in order to come to an overall solution.

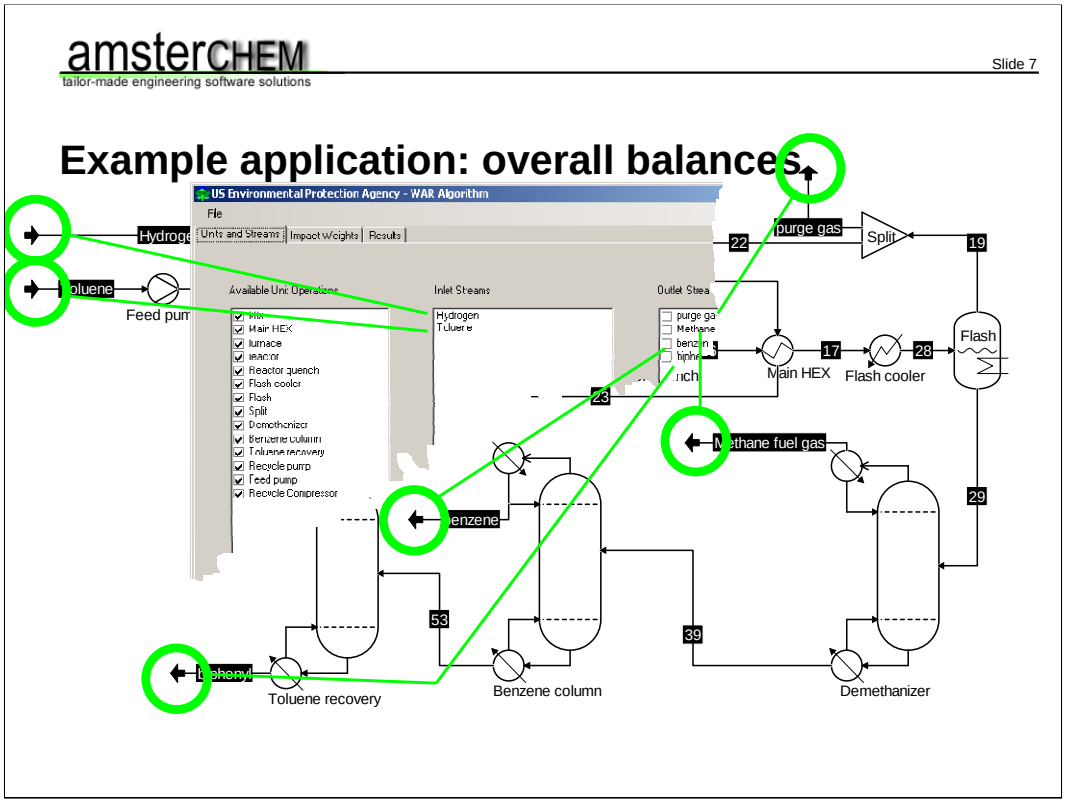
Flowsheet monitoring

Allow for an additional type of flowsheet component that

- has access to the underlying thermodynamic engine
- has access to all streams and stream data
- has access to all unit operations and -data
- has the ability to determine which streams are connected to which unit operations
- has the ability to perform event driven calculations

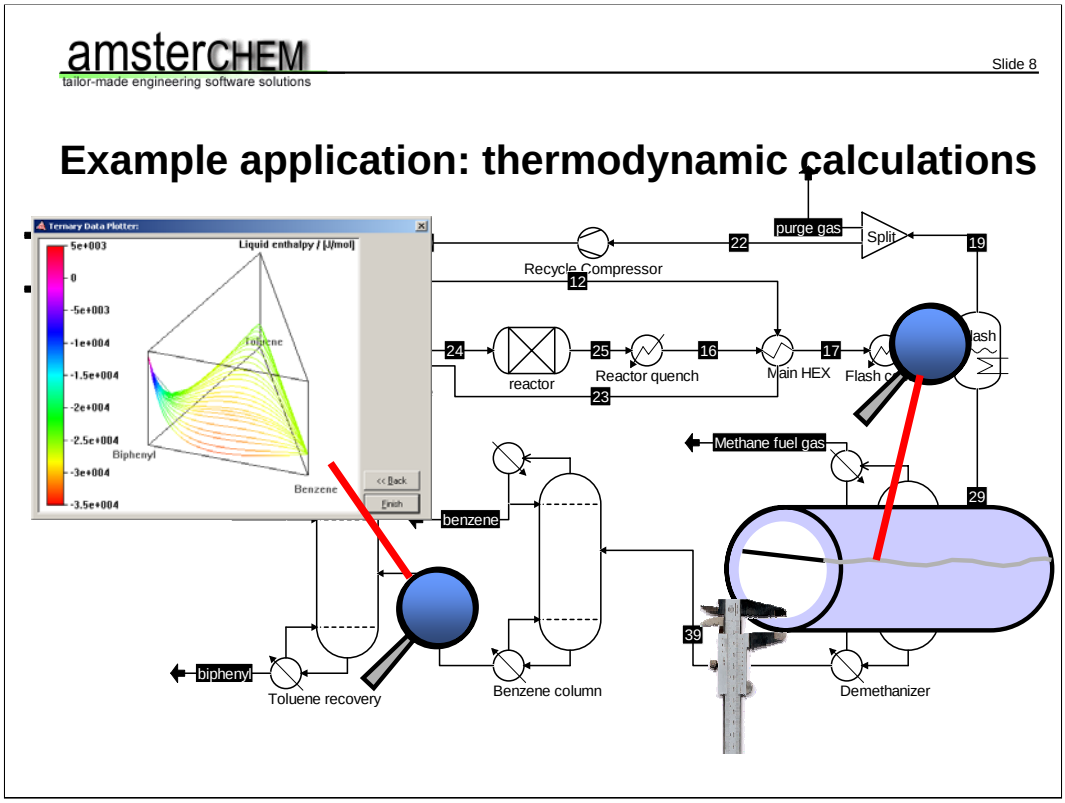
We have seen that streams connect to unit operations, unit operations only know about the streams that they are connected to, and the underlying thermodynamic system does not know anything about streams or unit operations that are present in the flowsheet. So the only software component that has access to all flowsheet elements is the simulation environment application. Now, what if you want to calculate something that applies to multiple flowsheet elements? Or even all flowsheet elements? This could hence only be done by the simulation environment. Having the possibility to do calculations that apply to multiple flowsheet elements – even if not supported by the simulation environment – would then be a useful feature. Hence, we need a new type of flowsheet element that has read-only access to all other flowsheet elements, for post-processing and other calculations. We will call such a software component a Flowsheet Monitioring Object. So we define flowsheet monitoring as allowing access to software components that can access all flowsheet elements: the thermodynamic system, the collection of streams, the collection of unit operations. It should be able to get access to all data of streams (e.g. pressure, temperature, flow, composition and other physical properties), it should have access to parameter values of unit operations. It should be able to figure out the connectivity between streams and unit operations. And – last but not least – it should be able to redo its calculations when something in the flowsheet has changed, for example when a new solution is found.

-

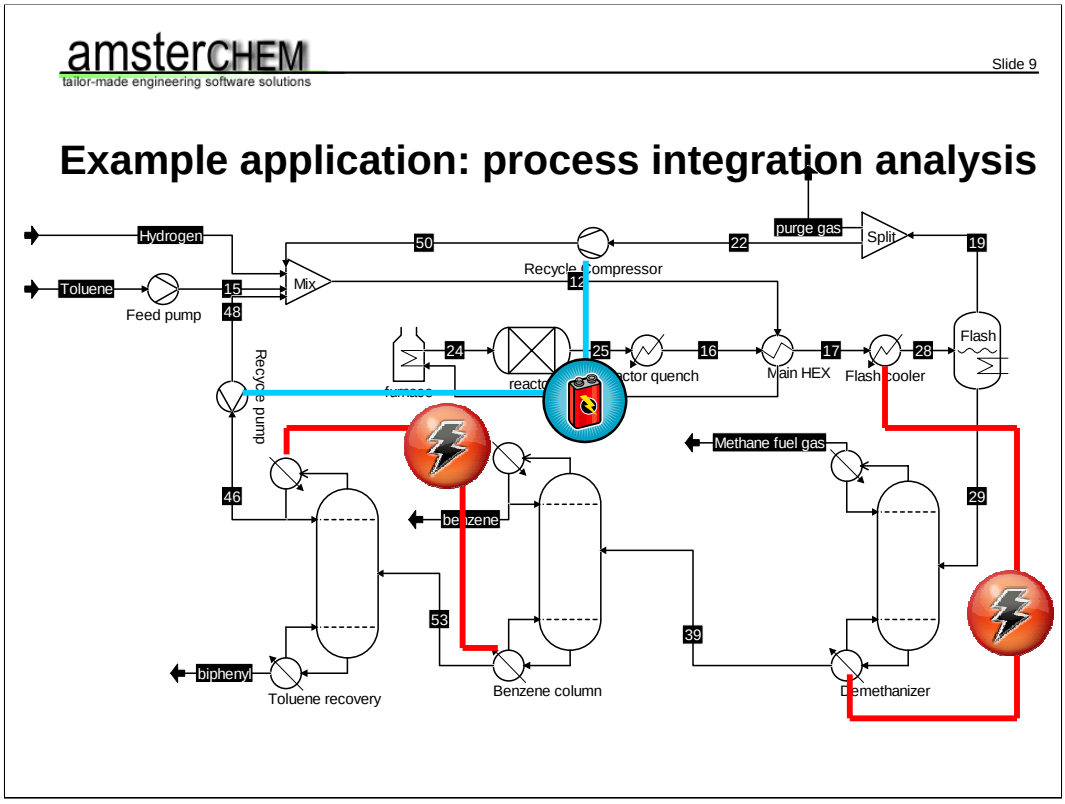


Let us look as some example applications where flowsheet monitoring is handy. First example, software component that want to access everything that goes into a process and everything that comes out. This starts with finding all global feeds and products. This step is rather straight forward. With the collection of streams and unit operations this is solved easily. You ask each unit operations for its collection of ports. For each inlet port and outlet port you see which streams is connected. This way, for each stream you know whether it is connected as inlet or outlet – or both – of a unit operations. Streams that are not connected as unit operation outlet, are feed to the entire process, and streams that do not go into a unit operations are process products. Or waste streams.

An example application in this category is the Waste Reduction Algorithm, the WAR application of the EPA, a tool to calculate environmental impact of the process. It is currently available from the EPA, and uses the monitoring interface.

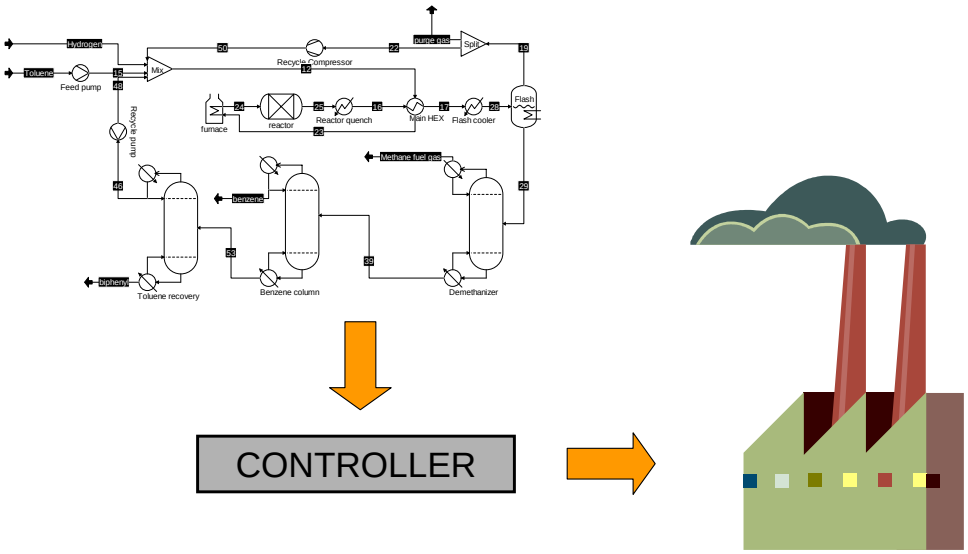


The next category of applications are those that want to do general thermodynamic calculations. This may or may not be at the conditions that are given by a stream in the flowsheet. For this, the Flowsheet Monitoring object requires access to underlying thermodynamics, and optionally to the collection of streams. An example is given by the Ternary Plugin TERNYP that ships with COCO; this will provide the user with ternary property plots, phase envelopes and residue curve calculations. Another example is determination of wax or hydrate formation in a pipe that is represented by a given stream, as envisioned by Infochem Computer Services Ltd that are specialists in the field.

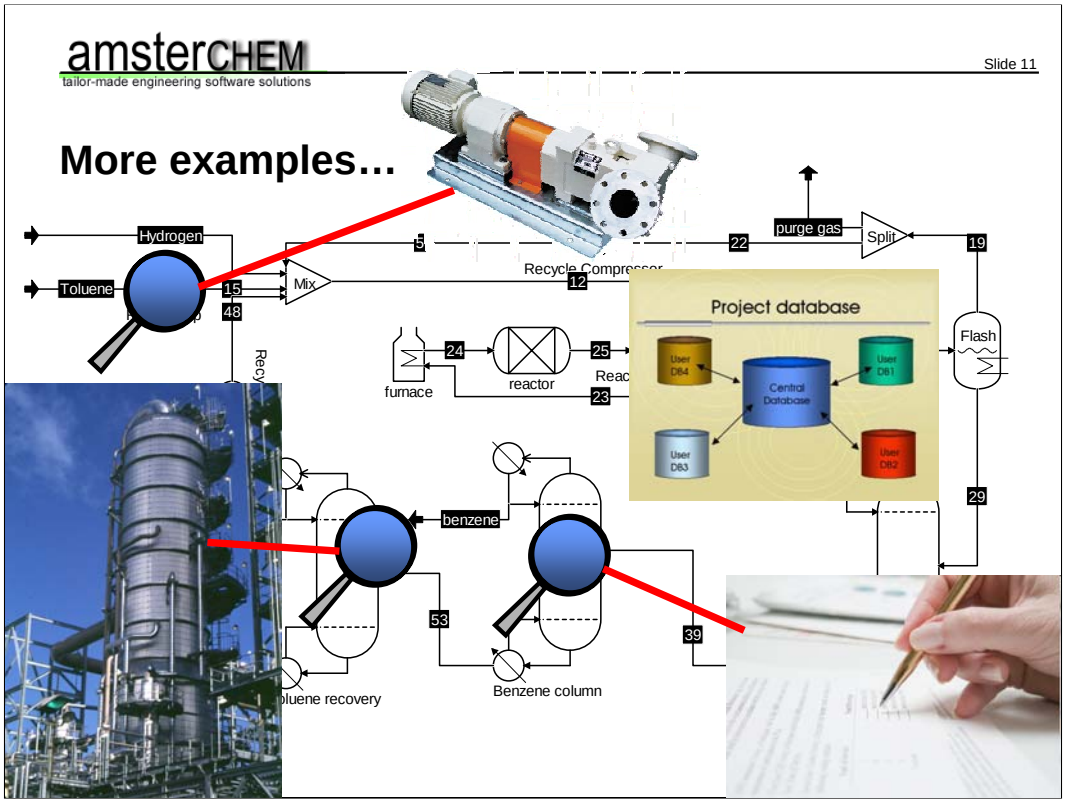


Then there are applications in process integration, where you need to analyse the complete flowsheet to find how to best use heat or work that you produce in one part of your process and apply it to another part of your process. An example for a heat integration approach would be a Flowsheet Monitoring Component that does a pinch analysis of your process.

Example application: real-time optimization



You can of course also use the solution of the complete flowsheet to monitor the status of an actual plant. The solution of your entire flowsheet can then serve to produce set points for controllers in the actual plant. Examples in this class require event driven operation; they need to be take action when a new flowsheet solution becomes available. As the proposed interface is equally applicable to dynamic simulations, on-line control is also possible.



There are more examples one can think of. The ChemSep authors plan on making a distillation column rating mode, which specifically needs access to a single unit operation of the flowsheet. One can directly use flowsheet data for equipment design, and even base an economical optimization on that. One could generate Data Requisition Sheets directly from the equipment data in the flowsheet. One can think of applications that collect all solutions to a flowsheet and store these in a data base for future reference, or for interpolation purposes, or to derive an initial guess for future solutions.

The general idea is clear at this point; we can dream up more applications that require access to multiple flowsheet elements than the simulation environment will be able to provide us with. The Flowsheet Monitoring approach therefore is justified, and the requirements are clear and straight-forward.

amsterCHEM
tailor-made engineering software solutions

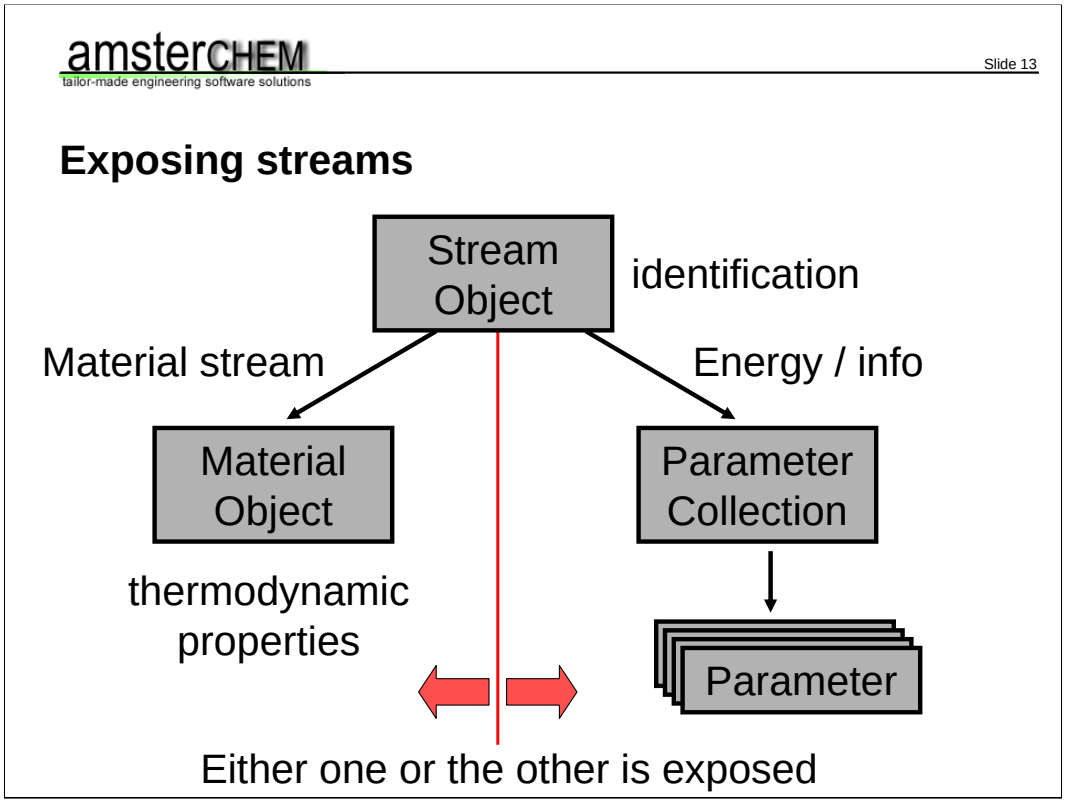
The architecture: CAPE-OPEN

CAPE-OPEN provides us with:

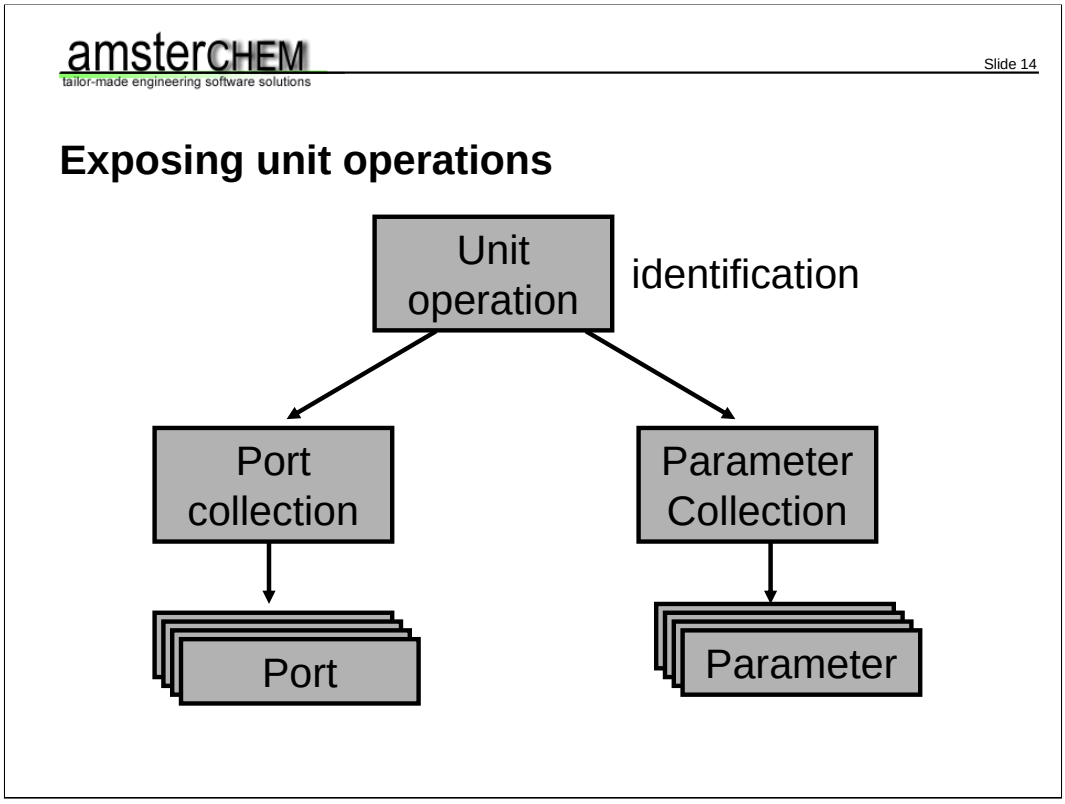
- definitions of a thermodynamic system
- definitions of streams
- definitions of unit operations
- common interfaces: utilities, collections, identification, errors, persistence

CAPE-OPEN available in all major simulation platforms

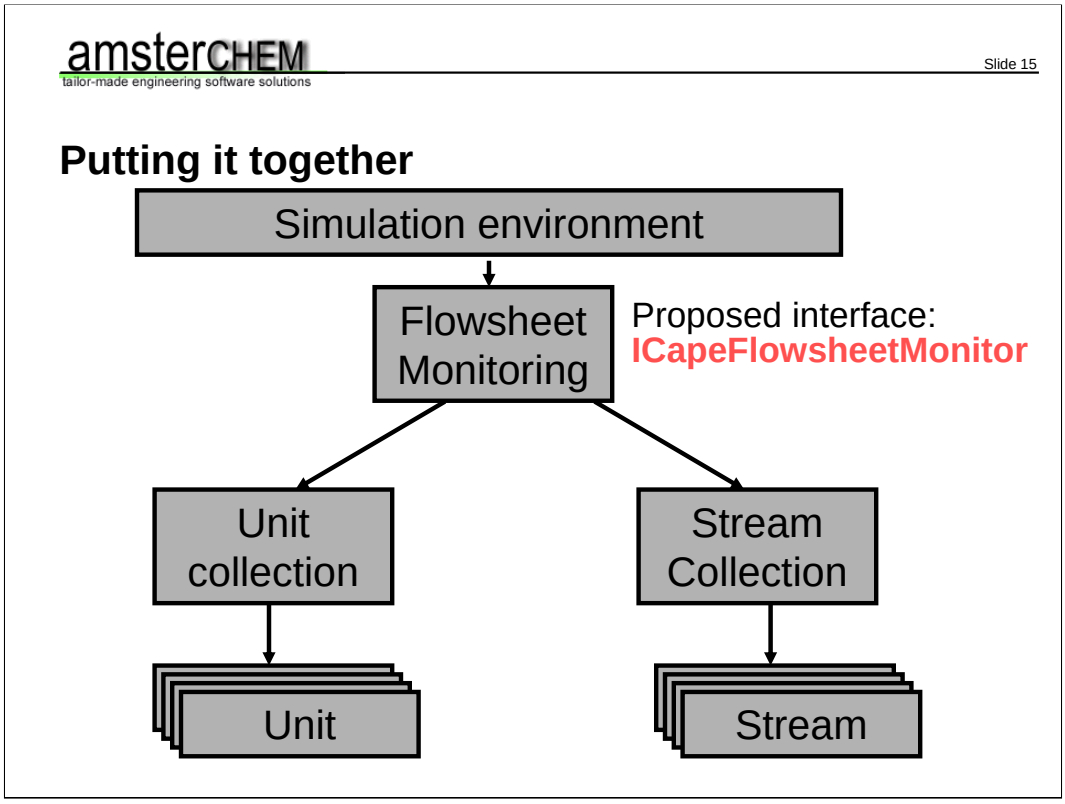
Now that we have set the scene and now what we require, let's have a look at what we have. If we are thinking of writing software components that work in multiple simulation environments, the first thing we should think of is CAPE-OPEN. This is why we are here today.
Conveniently, CAPE-OPEN defines interfaces to describe all of the flowsheet elements that we are interested in. The thermodynamic calculation system is well defined, material streams are defined by CAPE-OPEN material objects and energy or information streams are defined as collections of CAPE-OPEN parameters. There is a set of interfaces to describe unit operations, as well as one to describe collection. All CAPE-OPEN software components even identify themselves. And the best thing is support for all of these interfaces is present already in all major simulation environments. So let us build up a monitoring hierarchy from the ground up. I will show the elements that we need to access. Mind that we will work under the restriction that all flowsheet elements will be accessed by the flowsheet monitoring object in a read-only manner. This is because we cannot touch the data structure that is maintained by the simulation environment. We can therefore not modify – say – the pressure on a stream. We can however duplicate the material object that describes the stream and perform calculations on the duplicate. So…



If the monitioring object can access the streams, it has access to the material object in case of a material stream. This does not only allow us access to all thermodynamic properties, but also to all property calculations that can be done by the underlying thermodynamic system. We can after all duplicate the material object and populate the duplicate with the properties that we want to. If the stream is an energy or information stream, we have access to a collection of parameters describing the data on the stream. The stream also identifies itself.



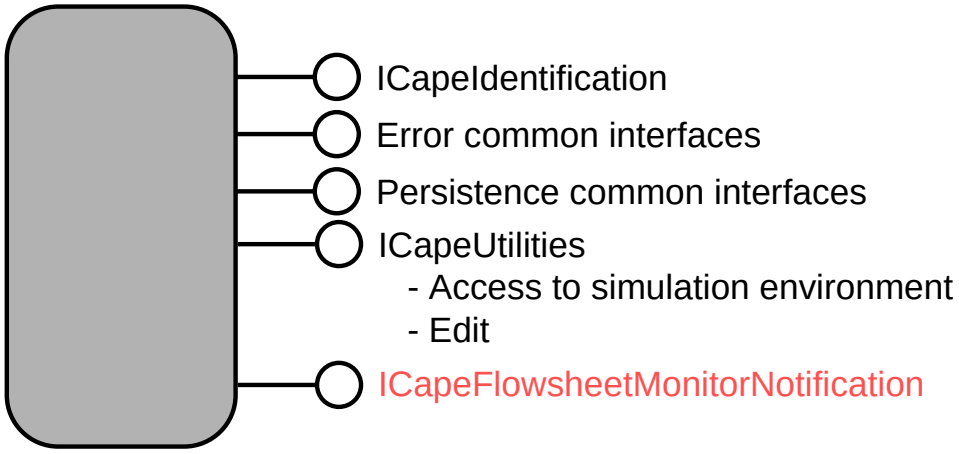
If the flowsheet monitoring object has access to the unit operations, we automatically have access to its ports and parameters. The ports tell us whether it is an inlet or outlet, what type it is (e.g. material, energy or information) and to what stream it is connected. This information, in combination with access to the streams, is sufficient to analyse the flowsheet connectivity. A unit operation also identifies itself. All CAPE-OPEN objects are identified by textual name, so it is up to the simulation environment or user to keep all names unique for proper operation of the monitoring object.



The flowsheet monitoring object thus needs to have access to a Flowsheet Monitoring interface. This proposed flowsheet monitoring interface will expose a collection of unit operations and a collection of streams. Not shown here is that it will provide some additional information methods, like whether or not the flowsheet is valid and solved.
The Flowsheet Monitoring interface here is a new interface. The collections are existing CAPE-OPEN interfaces however. Notice that the flowsheet monitoring interface is implemented by the simulation environment, and accessed by the Flowsheet monitoring object. Via the Flowsheet Monitoring interface, the Flowsheet Monitoring object can access all streams, the underlying thermodynamic data, and all unit operations. We have now fulfilled most of our requirements.

Flowsheet Monitoring Object

PMC software component: expose CAT-ID



The Flowsheet Monitoring Object itself a CAPE-OPEN client object, much like – say – a unit operation. To add a flowsheet monitoring object to our simulation is like adding a unit operation, we define a category ID for this category of CAPE-OPEN components.

As a CAPE-OPEN PMC software component, it will need to implement ICapeIdentification. For errors it will use the error common interface and if it wants to be saved between sessions also the persistence common interfaces.
It also needs to implement the utilities interface, through which it will get access to the simulation environment. The simulation environment will provide access to the Flowsheet Monitoring interface, and through there, all other flowsheet elements. Access to the simulation environment may also provide access to the material template system, if implemented by the simulation engine. Implementing the utilities interface also makes that we can manually activate our monitoring object, by calling its Edit routine.
We have now fulfilled all requirements that we have set, except for one: event driven operation. We can at this point only manually invoke the monitoring object, by calling its Edit functionality. For event driven operation, we need a new interface: ICapeFlowsheetMonitorNotification. This interface should be called by the simulation environment when a change in state has occurred.

amsterCHEM
tailor-made engineering software solutions

Proposed interface:

ICapeFlowsheetMonitorNotification

Only required for event driven monitoring objects

(not required for manually invoked monitoring objects)

Methods:

- Unit operation added / removed / modified / renamed
- Stream added / removed / modified / renamed
- Flowsheet solved
- Next time step (dynamic simulations only)

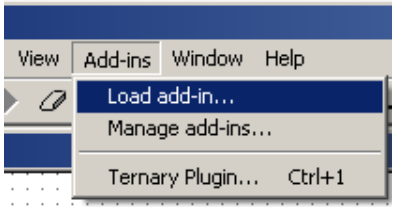
ICapeFlowsheetMonitorNotification and ICapeFlowsheetMonitor are the only newly defined interfaces required for the monitoring architecture. A flowsheet monitoring object can implement ICapeFlowsheetMonitorNotification, but does not need to. It only requires implementation for a flowsheet monitoring object that needs to perform event driven calculations. So for any flowsheet monitoring object that is invoked manually, this interface does not apply.

In this interface, methods should be present for when a unit operation or stream is added or removed, or when its state has changed. Also the rename event is important, as the name identifies a unit operation or stream. A notification also needs to be present for when a new solution is found to the flowsheet.

amsterCHEM
tailor-made engineering software solutions

Slide 18

Requirements on the PME



- Implement Flowsheet Monitor interface: little impact
- Ability to load and use monitoring objects
- Expose all streams as CAPE-OPEN MO: little impact
- Expose all unit operations as CAPE-OPEN: medium impact
- Notifications: considerable impact

A new CAPE-OPEN interface is only useful if it is used. And that can only be done if it is implemented. For the client side, I do not foresee this as an issue. There are two flowsheet monitoring objects already around and in use. And there are several on the planning, by various software vendors.

So let us have a quick look at the implementation requirements for the simulation environment. Most CAPE-OPEN efforts are after all limited in development speed and use by the time it takes for the major simulation vendors to implement support for it. Here, the smaller the task, the better the chance we can convince the software vendors to implement support.

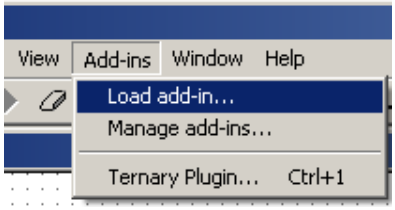
First of course, the simulation environment needs to provide the flowsheet monitoring interface, with the collections of streams and unit operations. This is not much work at all. Also, it needs to provide a user interface for loading and maintaining flosheet monitoring objects, as well as activating them. This is also not much work. In COFE, this is done by an Add-In menu, as shown. In the top of the sheet.

Through the stream collections, all streams need to be exposed. For material streams, in the form of an object that exposes ICapeIdentification and a CAPE-OPEN material object implementation. This would generally have significant impact, but the job is already done. All flowsheet environments that support CAPE-OPEN have material object implementations to represent the streams. Hence, little impact.

Through the unit operation collection, all unit operations need to be exposed as CAPE-OPEN objects. As opposed to the streams, this work is not already done. All flowsheet engines that support CAPE-OPEN can import CAPE-OPEN unit operations, but to expose internal unit operations as CAPE-OPEN unit operations is a different story. Mind that we need only read-access to the unit operations though.

amsterCHEM
tailor-made engineering software solutions

Requirements on the PME



- Implement Flowsheet Monitor interface: little impact
- Ability to load and use monitoring objects
- Expose all streams as CAPE-OPEN MO: little impact
- Expose all unit operations as CAPE-OPEN: medium impact
- Notifications: considerable impact

(ctd) … Mind that we need only read-access to the unit operations though. For a well-written simulation environment that can access its own unit operations in a uniform manner, this should not be too big a hassle. Classified as medium impact.

Finally, support for the notifications, that are required for event driven operation. This may be considerable impact. At every place in the code that causes the state of a unit operation or stream to change, an event may need to be fired. Whether or not this can be done easily depends on the internal organization of the flowsheeting engine. I foresee quite some trouble for non-object oriented legagy applications here. The good news is that this is the last and least required step. A very useful monitoring interface can already be supported if event driven operation is not part of the implemenentation. And support for that can always be provided in a later stage.

Current status:

- Proposal has been made
- IDL available (except for notifications)
- Implementation in COFE (COCO)
- Two client implementations: TERNYP (COCO) / WAR (EPA)
- Implementations have been tested
- More interested parties in writing client applications

Request to simulation vendors: support in PME!

A quick glance at the current status: a proposal has been made and posted to relevant interested parties inside and outside CO-LaN. Thiis proposal translates into a formal desciption in the form of an IDL and a describing document. This IDL does not yet include the notification structure for event driven operation. This part is fully implemented and functional in COFE, since version 1.12. There are currently two client implementations out there, that have been testeed and are both in use. Other client implementations are in the conceptual phase.

I therefore have a request to simulation vendors: please implement support for this interface, the socket side is very important.

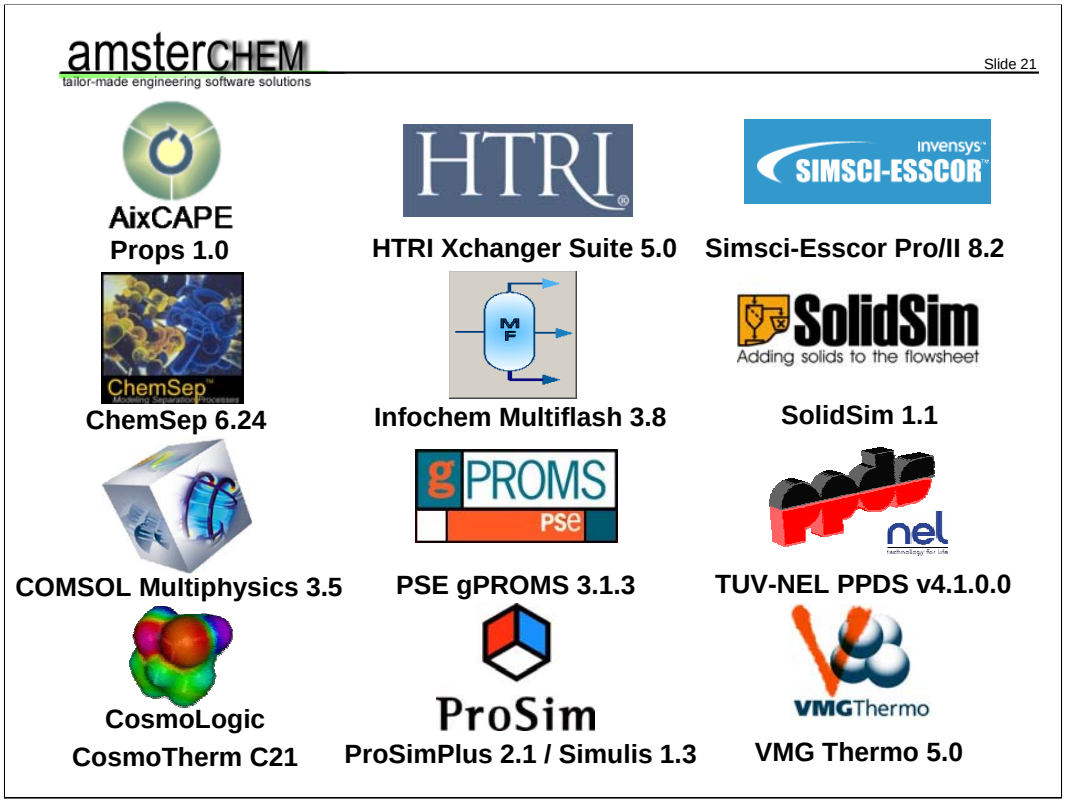
amsterCHEM
tailor-made engineering software solutions

- Download COCO: http://www.cocosimulator.org/ (or ask for a copy during the workshop)
- Forum: http://capeopen.19.forumer.com/viewforum.php?f=15
- Contact amster**CHEM** for CAPE-OPEN consulting and implementation
- Interoperability testing program: http://www.cocosimulator.org/index_compliancy.html

Acknowledgements:

- Richard Baur
- ChemSep: Ross Taylor, Harry Kooijman
- Cosmo*THERM*: Frank Eckert
- Michel Pons
- William Barrett

COCO is available for download from cocosimulator.org; the monitoring IDL is available on request. If you want my help in CAPE-OPEN software issues, please contact me at jasper@amsterchem.com. As always I call on people to join the interoperability programme. If you have CAPE-OPEN software or components, please see the compliancy testing page at cocosimulator.org for more information.



COCO would not be what it is today without continuous interoperability testing. So I would much like to thank all the people that provide me with the licenses to do so. Thank you.