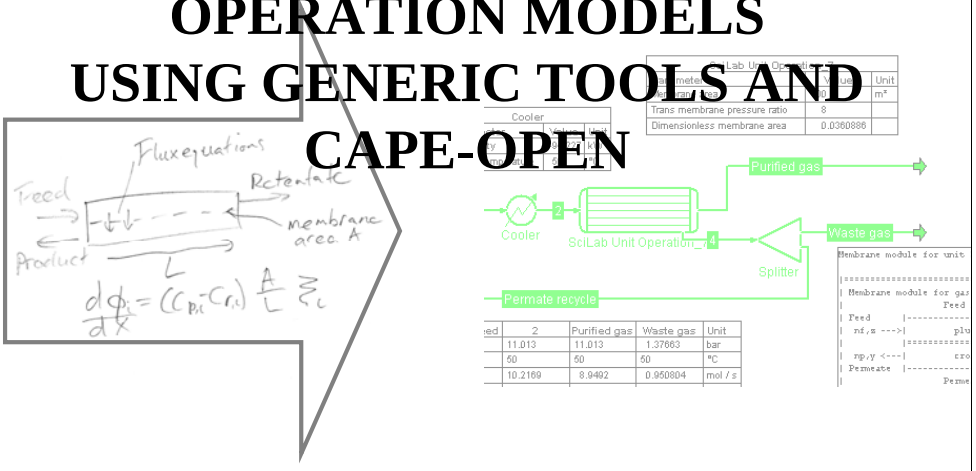
amsterCHEM
tailor-made engineering software solutions

Slide 1

RAPID PROTOTYPING OF UNIT OPERATION MODELS USING GENERIC TOOLS AND CAPE-OPEN



Jasper van Baten, AmsterCHEM **Michel Pons, CO-LaN**

Today we are surrounded by chemical engineers, or at least that is my presumption. Everybody here has experience, or is at least well familiar with the concepts of, steady state flowsheeting. We are all aware of the extensive use of flowsheeting in the fields of process design, plant monitoring, operator training and short term as well as long term planning. Those of you here today that come from industry may have contracts with one or two software vendors for specific simulation environments, such as Pro/II, Aspen Plus or others. Or you may be using in-house simulation tools. The success of simulation tools depends on both the accuracy and complexity of the underlying models, both the thermodynamic models and the unit operations. It is the unit operations that are addressed in this presentation. If the unit operation are not accurate enough, the flowsheet model predictions may be of no use beyond feasibility studies. The other extreme is that the unit operations are overly accurate, such as CFD based models, that practical use is limited by time-consumption. Each simulator comes with a number of built-in unit operations. For particular purposes, these may not be sufficient, and often unit operation models are developed specifically for use in a particular company or process.

amsterCHEM
tailor-made engineering software solutions

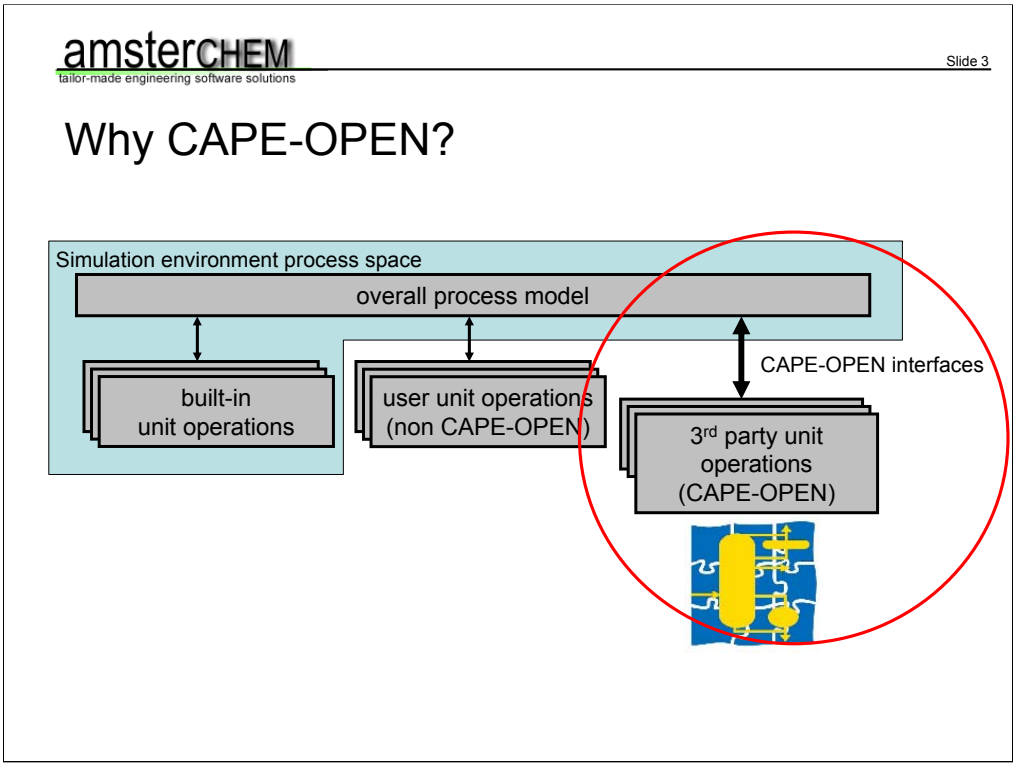
Slide 2

Presentation outline

- Why custom models?
- Why CAPE-OPEN?
- What is involved?
- Short-cut: Matlab and Scilab
- Short-cut: Excel Unit Operation
- Current status

In the foregoing quick introduction we have already indicated that custom unit operation models are widely used in practice, and touched upon the underlying reasons.

In the remainder of this presentation, we will quickly address the advantages of using CAPE-OPEN as a basis for custom unit operation model development. We will then take a peek at what is involved in CAPE-OPEN unit operation development, and see that for process engineers the threshold may be significant to go from the drawing board to a full-featured optimal implementation. We will then offer solutions that will allow you combine the advantage of CAPE-OPEN based development with generic modeling tools such as the commerical Matlab program, its freely available Scilab cousin, or Microsoft Excel without the requirement for the model developer to go into the depth of CAPE-OPEN programming. We will conclude with some notes on the current status and software availability.



CAPE-OPEN is a set of interface definitions that are supported by all main-stream steady state flowsheet engines. Its application with respect to unit operations is well-established, and implementation is at a status where for most simulation platforms, CAPE-OPEN will work. Typically, simulation environments offer, apart from the built-in unit operations, two ways of adding user-defined unit operations models. The non-CAPE-OPEN way, traditionally by means user FORTRAN code, will work fine for a particular simulation environment. The CAPE-OPEN way will work independently of the simulation environment. This statement makes two claims. The first one is that CAPE-OPEN will *work*. The second claim is that its working *does not depend* on the simulation environment. So the question: "why CAPE-OPEN?" is best answered by looking at the reasons not to depend on a particular simulation environment.

amsterCHEM
tailor-made engineering software solutions

Slide 4

Why CAPE-OPEN?

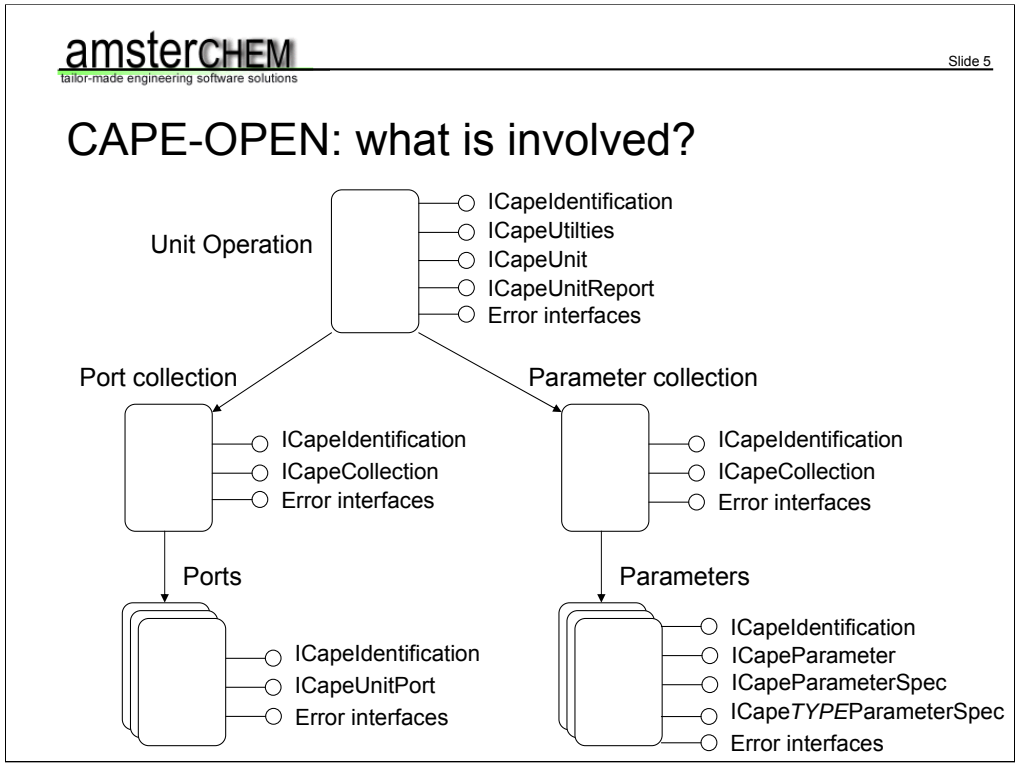
- Use of multiple simulation environment in company
- Validation of simulation results
- Bargaining position with respect to simulation vendor
- Model distribution

The first reason to write a unit operation that works in multiple simulation environments is the most obvious: multiple simulation environments may be used inside the company. Using CAPE-OPEN you can write code that will work in all of them, reducing the coding and software

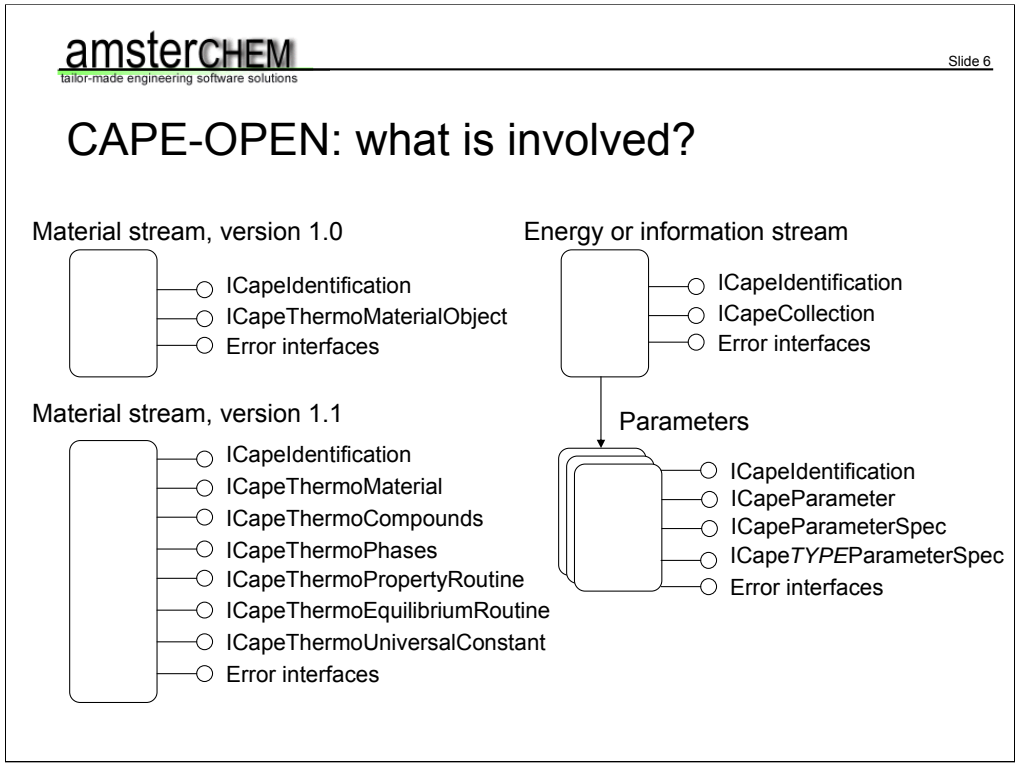
Secondly, this approach allows for validation of simulation results across simulation platforms. The simulation results ideally should not depend on what software we are using.

The next consideration is more a business strategic one: once we have developed our model to only run in a particular simulation environment, we are stuck to using that particular environment. Being flexible with respect to simulation environment gives us a better bargaining position when purchasing our simulation platform.

Finally, we may not only want to run the model inside the company, but we may also want to distribute the model for use by others. By using CAPE-OPEN we do not force our clients to use the same simulation environment we do.



CAPE-OPEN is a set of interface definitions. This means that we will need to implement a piece of code, usually a DLL-based COM-server, that implements these interfaces. This image shows an overview of what is expected from a unit operation: we need to implement not only the unit operation specific interfaces, but also interfaces that define its parameters, identification, generic error handing, generic utilties like editing, and persistence for loading and storing our model. This is covered in more detail in the CAPE-OPEN interace specifications, that are available from the CO-LaN web site. As unit operation developer, we need to implement all of these.



The ports of a unit operation connect to streams. These can be material streams, energy streams or information streams, all of which are represented by their own sets of CAPE-OPEN interfaces. Fortunately, as unit operation model developer, we do not need to implement these; the simulation environment does that. We do however need to be aware of how to use these objects.

amsterCHEM
tailor-made engineering software solutions

Slide 7

CAPE-OPEN: what is involved?

- COM server
- Implementation of the Unit Operation objects
- Deal with CAPE-OPEN interfaces exposed by streams
- Solution routines
- All in an efficient manner

A shortcut to model prototyping and testing is welcome

To summarize, for a CAPE-OPEN unit operation, we need to build a COM server that exposes a number of objects and implements a considerable amount of interfaces. Also, having our equations on paper does not automatically mean we have the algorithms to solve the equations. Often, we know how to solve our equations using generic tools, such as Microsoft Excel's equation solver, or the solution and integration routines of Matlab or Scilab.

Note that we do not want to discourage to eventually go through all the above steps: an efficient implementation of the final version of our unit operation may be very important, especially where a lot of thermodynamic requests are made upon the simulation environments.

We do however recognize that it would be helpful to first set up our model as a prototype, in tools that we are comfortable with, such as Excel or Matlab. Then we can validate that our model behaves as intended, possibly making changes to its design. The ability to do this in while running in a flowsheet gives us real insight how our unit operation model behaves in an actual process model. We want to do without necessarily first going through the learning curve of CAPE-OPEN programming. Once we are satisfied with the unit operation model, we can do its final implementation. This is a task we could outsource.

amsterCHEM
tailor-made engineering software solutions

Two-step development:

- evaluate our model equations
  → we require access to thermodynamics
- evaluate the final process model
  → requires running as Unit Operation inside simulation environment

If we have conceptually set up our unit operation's design equations on paper, we may first want to test how they behave. We will need access to thermodynamics, as more our unit operation's model equations generally are based on first principles like mass and heat balances, and use physical properties for mass, heat and momentum transport. We may not want to go straight from scratch to a unit operation model, as this would require us to be able to predict all the product streams of our unit operation; otherwise the unit operation will not run.

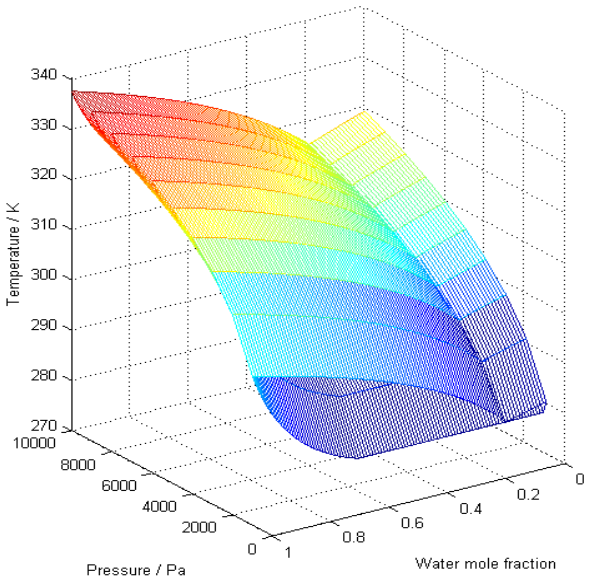
As a first step we may want to look at our model equations, preferably interactively, being able to access thermodynamics.As a second step, we want to take our validated model equations and put together the complete unit operation. This unit operation we can test inside a simulation environment to see its behaviour inside and actual process model.

amsterCHEM
tailor-made engineering software solutions

Interactive thermodynamic access:

- Matlab or Scilab: script based
- Interactive or batch script processing
- Load third party thermodynamics
- Access to:
  - ❖ Compounds and compound data
  - ❖ Phases and phase descriptions
  - ❖ Thermodynamic property calculations
  - ❖ Thermodynamic equilibrium (Flash) calculations

For the first step, we may use modeling environments such as Matlab or Scilab that allow us to enter commands to evaluate our equations. Both of these tools are script based: we can enter commands manually for interactive use of our equations, or put the commands in a script file that we can run. As we do not want to write our thermodynamics ourselves, we want to be able to import and use third party thermodynamic servers. Once loaded, we want to have access to the compound and phase definitions, and their properties. We want to be able to calculate thermodynamic properties, for example the enthalpy of a vapor phase at specified temperature, pressure and composition. We may also need equilibrium calculations, or flash calculations, for example, which phases exist for a given system at fixed composition, pressure and temperature or enthalpy. CAPE-OPEN Thermodynamic Import tools are available from amsterchem.com for both Matlab and Scilab. A quick example…



… we see here a phase envelope for the Water / n-Butanol system at various composition and temperature. Such a plot allows us to inspect the shape of the phase envelope within the region of interest for a quick analysis of what our process model may do under certain conditions. The code to generate the data is shown on the next slide.

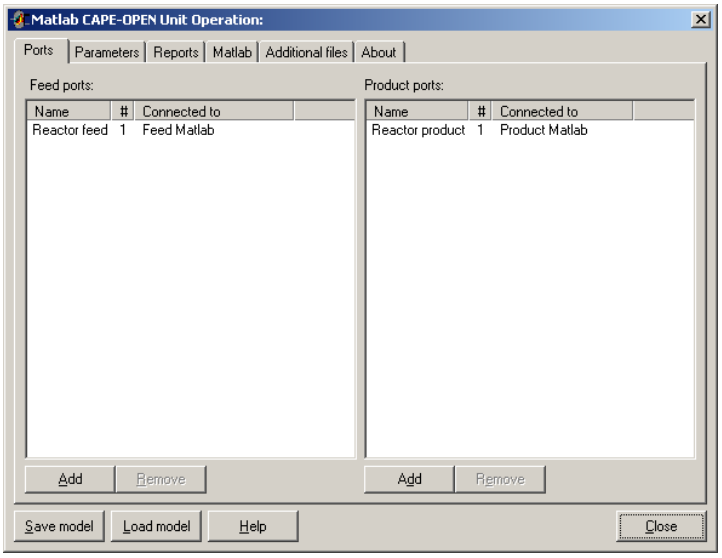
```
handle=capeOpenGetPackage …
('Multiflash Property Package Manager','WATER-N-BUTANOL');

frac=(0:0.01:1)';
X=[frac 1-frac];
P=[1e3:1e3:1e4];
tbub=[];
tdew=[];

for p=P

 tbub=[tbub capeOpenEquilibriumProp(handle,'temperature', ...
     X,'pressure',p,'vaporFraction',0)];
 tdew=[tdew capeOpenEquilibriumProp(handle,'temperature', ...
     X,'pressure', p,'vaporFraction',1)];

end;
```

First, we issue a command to create the CAPE-OPEN property package (shown in red). A handle is returned that we can use in the subsequent thermodynamic evaluation calls. Inside the for loop, the dew and bubble-point evaluations are done for a complete array of compositions, using a call to the capeOpenEquilibriumProp function (shown in red). For this, one does not need to know a lot about CAPE-OPEN.

amsterCHEM
tailor-made engineering software solutions

Slide 12

Implementation as a Unit Operation:



We thus have proceeded to use the thermodynamics as shown in our previous to validate and evaluate our model equations. Now it is time to set up the entire Unit Operation in the context of a process model. Feed conditions follow from material objects connected to feed ports.

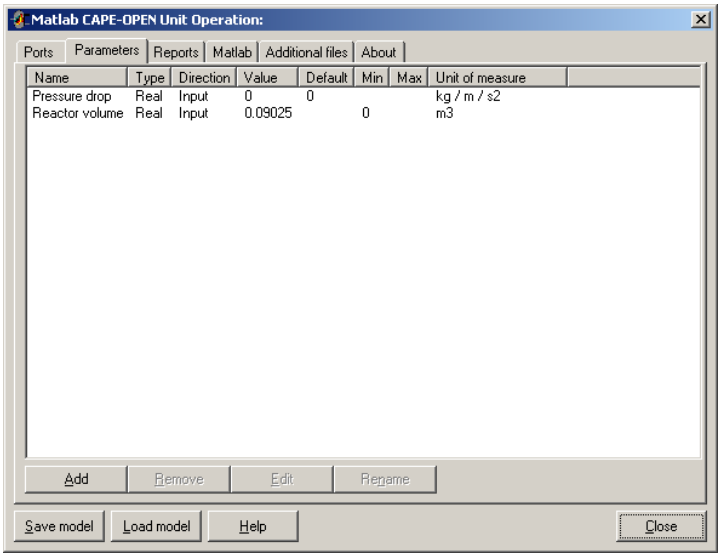
Thermodynamics of the simulation environment are used.

To get started, we insert a Matlab or Scilab Unit Operation into our flowsheet. Editing the unit operation results this dialog. We must tell the simulation environment some important things about our unit operation.

First we start by defining feed and product ports. A page is present in the dialog that allows for adding and removing feed and product ports. Once connected, this page will also show us what streams the ports are connected to.

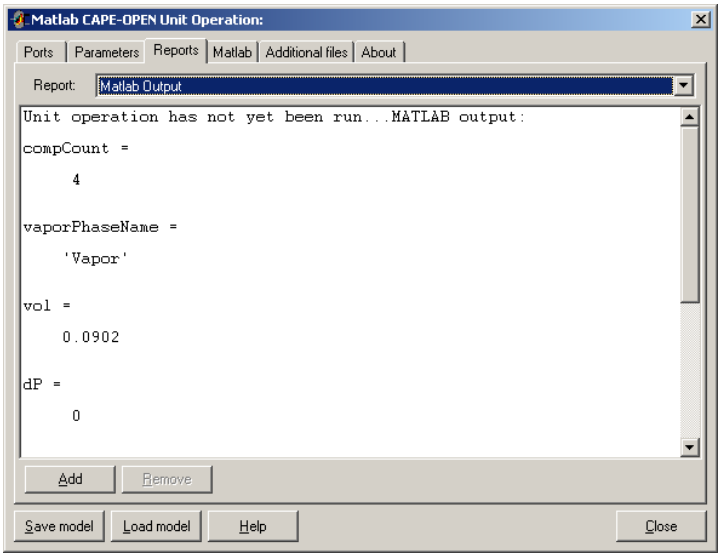
amsterCHEM
tailor-made engineering software solutions

Implementation as a Unit Operation:



Then we need to specify which in- and ouput parameters are present, on the next page of the dialog. For each parameter we can specify a name, data type, default value, minimum and maximum value and dimensionaly, which is converted by the simulation environment to a unit of measure. The values of input parameters we can access from our Matlab or Scilab script that calculates the unit operation. Similarly, the calculation script needs to set the values of output parameters.

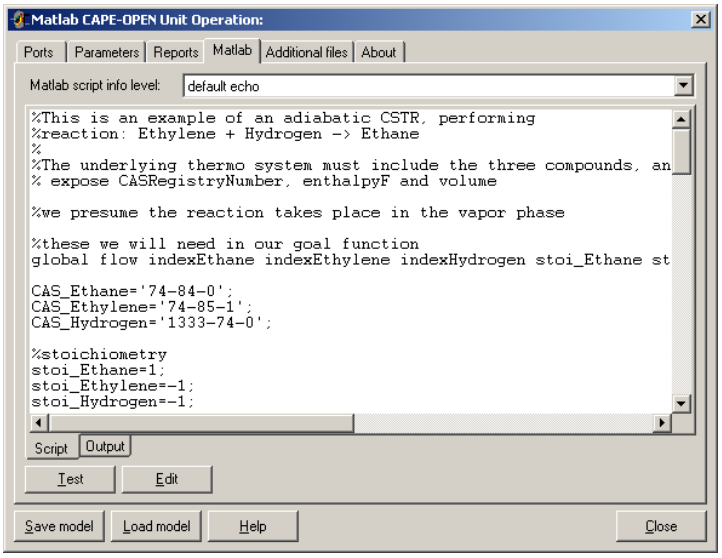
Implementation as a Unit Operation:



Textual reports can be specified at the next page. The calculation script will need to set the contents of each defined report, except for the default output report. The default output report shows the Matlab or Scilab calculation output.

amsterCHEM
tailor-made engineering software solutions

Implementation as a Unit Operation:



Finally we need to enter the script that performs the calculation of the unit operation. Additional files required by the calculation can be added as well.

The calculation script must: get the values from the feed ports and inlet parameters, set the values for each product port and for output parameters and set the contents of reports.

We can use thermodynamic calculations of the simulation environment, and of course all Matlab or Scilab routines such as integration methods, equations solvers, etc.

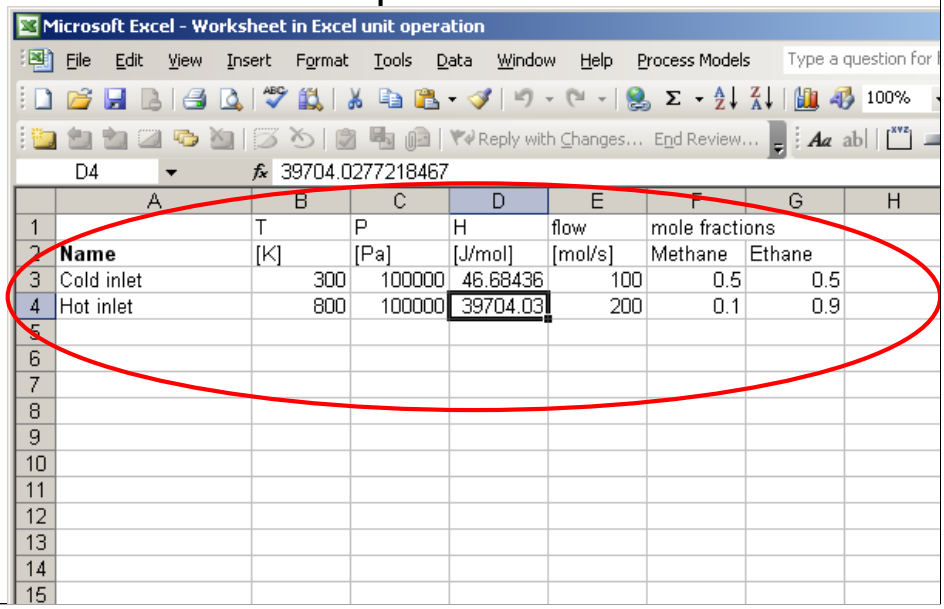
Once we have our calculation script, we can run the unit operaton in our simulation enviromnent and assess its behaviour.

Of course Matlab or Scilab needs to be installed to perform the actual calculations, so once the model has been validated and its design has converged, we still encourage a final version of the implementation that will run independent of Matlab or Scilab and will run more efficiently. With the prototype of the unit operation that we now have available, we have a good starting for doing the final implementation.

amsterCHEM
tailor-made engineering software solutions

Slide 16

Excel based Unit Operation:



	A	B	C	D	E	F	G	H
1		T	P	H	flow	mole fractions		
2	**Name**	[K]	[Pa]	[J/mol]	[mol/s]	Methane	Ethane	
3	Cold inlet	300	100000	46.68436	100	0.5	0.5	
4	Hot inlet	800	100000	39704.03	200	0.1	0.9	
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

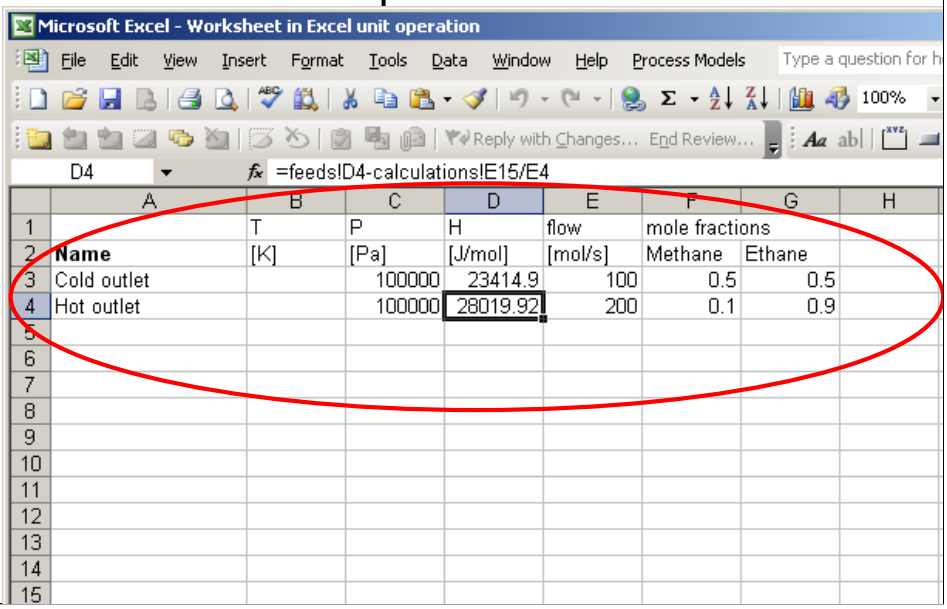
In script based unit operations, as shown until this point, the evaluation of equations is determined by the control flow of the script. Many engineers are more familiar wit h the Excel way of setting up equations. Here, formulas use inputs by referencing other cells in the Excel workbook. As this too is a common way of modeling, a Unit Operation implementation is available that uses Excel as the generic modeling tool. The setup is a bit different from before. We start again by inserting an Excel Unit Operation into a flowsheet. When editing the unit operation, Excel will pop up.

A workbook will be opened that has specific work-sheets for a number of tasks that we need to perform. First, we specify the feed ports of the unit operation. Specifying feed ports is as simple as adding rows to a table, as shown here. When the unit operation is evaluated, the feed conditions – pressure, temperature, enthalpy and composition – are automatically filled in. We can refer to these in out calculations.

amsterCHEM
tailor-made engineering software solutions

Slide 17

Excel based Unit Operation:



	A	B	C	D	E	F	G	H
1		T	P	H	flow	mole fractions		
2	**Name**	[K]	[Pa]	[J/mol]	[mol/s]	Methane	Ethane	
3	Cold outlet		100000	23414.9	100	0.5	0.5	
4	Hot outlet		100000	28019.92	200	0.1	0.9	
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

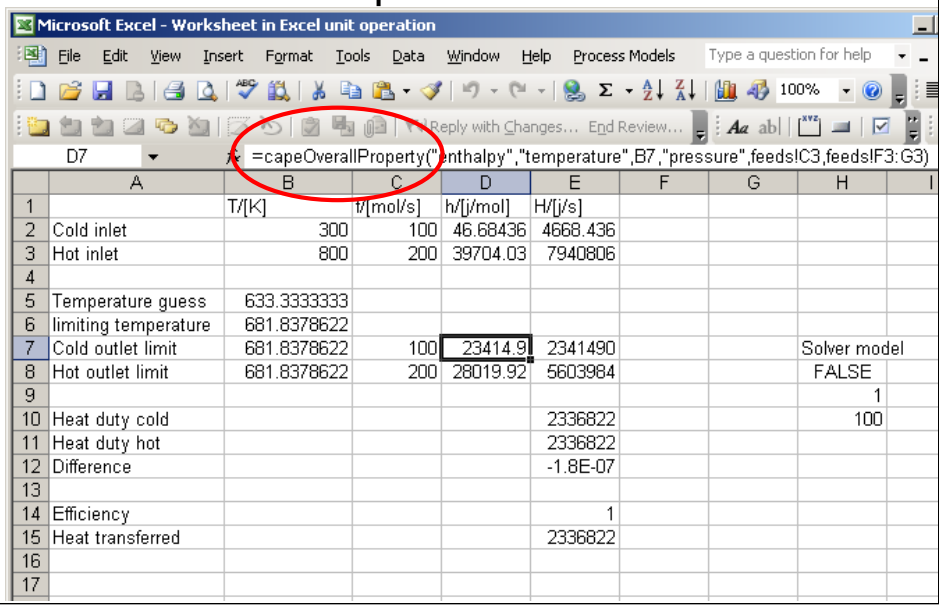
Similarly, we can add product ports to the unit operation, which is again as simple as adding rows to a table. It is the task of the unit operation to specify the product conditions. Hence, we must provide formulas for the overall composition of each of the compounds, as well as two out of three of pressure, temperature and enthalpy. The formula for enthalpy of a product stream shows here in the formula bar. It refers to the feed enthalpy and on an enthalpy difference that is calculated on another sheet.

In a manner very similar to feed and product ports, we can specify inlet and outlet parameters. Only double precision parameters are currently supported, for which we can supply a default value, minimum and maximum values and a dimensionality or unit of measure.

amsterCHEM
tailor-made engineering software solutions

Slide 18

Excel based Unit Operation:



	A	B	C	D	E	F	G	H	I
1		T/[K]	f/[mol/s]	h/[j/mol]	H/[j/s]				
2	Cold inlet	300	100	46.68436	4668.436				
3	Hot inlet	800	200	39704.03	7940806				
4									
5	Temperature guess	633.3333333							
6	limiting temperature	681.8378622							
7	Cold outlet limit	681.8378622	100	23414.9	2341490			Solver model	
8	Hot outlet limit	681.8378622	200	28019.92	5603984			FALSE	
9								1	
10	Heat duty cold				2336822			100	
11	Heat duty hot				2336822				
12	Difference				-1.8E-07				
13									
14	Efficiency				1				
15	Heat transferred				2336822				
16									
17									

Of course we also need to set up the equations for the unit operation. The formulas can refer to the feed conditions and to the input parameters. We can also use thermodynamic property calculations and flash calculations in our unit operation model as shown here. We can set up the Excel Solver Add-In to solve equations during calculation of the unit operation. The values calculated here should be used in the formulas that define product conditions and outlet parameters.

There is an additional options page that allows for configuration of the Excel solver, error messages, textual reports and so on.

amsterCHEM
tailor-made engineering software solutions

Slide 19

Current status:

- Matlab and Scilab Thermo Import utilities
- Matlab and Scilab Unit Operation utilities
- Excel Unit Operation utility
- Available free-of-charge for non-commercial use
- All use CAPE-OPEN thermo version 1.1
- Thermo tested with TEA / Multiflash / Simulis
- Unit operations tested in COFE / ProSimPlus
- Example are available online and in help
- http://www.amsterchem.com/

We have indicated that in practice, user-defined Unit Operations are commonly used. We have presented convincing reasons to use CAPE-OPEN as the basis for implementing Unit Operations. We have established that it can be useful to set up a working prototype of the Unit Operation before moving to the final implementation. Here are some finishing remarks on the current status:

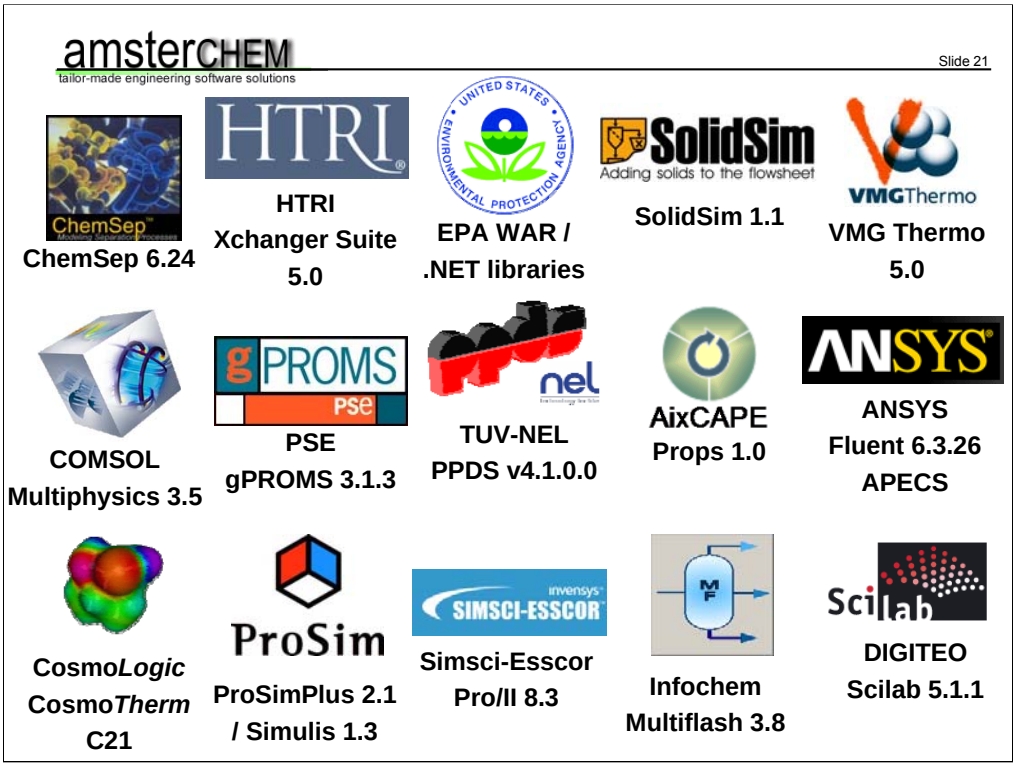
The thermodynamic import utilities are available for Matlab and Scilab. Matlab requires a licensed installation, its cousin Scilab is available free of charge. Unit operation implementations are available for Matlab and Scilab as well.

For Excel, CAPE-OPEN based thermodynamic calculations can already be performed with other tools, such as COCO's simulation environment COFE, or ProSim's Simulis Excel tools. In this context we have therefore only presented the Excel-based Unit Operation implementation. All of the above tools are available free-of-charge for non-commercial use. A small license fee is due for commercial use.

The tools presented here currently only use the CAPE-OPEN version 1.1 thermodynamic interfaces. This means they can at this point not be used in combination with software components that only implement the older version 1.0 standard. It is anticipated that all CAPE-OPEN implementations will upgrade to version 1.1 (if not, a COFE Flowsheet Unit Operation can be used as 'glue'). Both the thermodynamic import and unit operation tools have been tested against various third party products. More examples are available on-line and in the help files of each of the tools.

All tools are available from www.amsterchem.com.

COCO is freely available for download from cocosimulator.org; it can help you in your CAPE-OPEN development. Further CAPE-OPEN development resources can be found at the CO-LaN website, but also at the CAPE-OPEN forum. When developing CAPE-OPEN based models, you are encouraged to take part in the CAPE-OPEN Interoperability Testing programme. See cocosimulator.org for more details.



Finally, Jasper van Baten would like to thank all companies that generously provided software and licenses to AmsterCHEM for CAPE-OPEN development and testing.