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

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

Summary

Steady-state flowsheeting is widely used as a design and analysis tool in chemical process engineering. In a flowsheeting environment, various unit operation models are coupled together in order to close mass and energy balances. Existing simulation environments typically come with a built-in set of generic unit operation models, but for many purposes, more specialized unit operation models are required.

This presentation will show how generic modeling tools, like Matlab, SciLab and Excel, can be used to design prototype process models. To this extent, thermodynamic subroutines are required like those that are available in simulation environments. Via CAPE-OPEN it is possible to extend the generic modeling tools with third party thermodynamic engines.

Once the unit operation models have been created, the next step is validation and integration into process models. Generic modeling tools offer a way to quickly prototype the process models as CAPE-OPEN unit operations that can be used inside CAPE-OPEN aware process simulation environments.

Introduction

It is common practice in process design, plant monitoring and long-term planning to use steady state process simulations. Such simulations often are done in the form of flowsheet simulations, where process equipment is represented by unit operations; these unit operations are linked together to form a complete process model by means of material-, energy- and information-streams. The unit operation models are combined in a Process Flow Diagram, showing streams connecting the unit operations. An example of a flowsheet is presented in Figure 1[1,2].

A central facility for thermodynamic property calculations is underlying the individual unit operation calculations. This approach ensures thermodynamic consistency across each of the sub-models that make up the complete flowsheet model.

Flowsheet simulations are commonly performed in specific simulation applications or simulation environments. Typically, an industrial end-user (e.g. a chemical company) has contracts with one or more simulation software vendors, and use of simulation packages is standardized within the company to a few simulation packages. The primary reason that a company utilizes multiple simulation packages is that some packages specialize in a particular area of process simulation. Also, the company may need to validate simulation results across multiple simulation packages, to ensure that the simulation results do not depend on the software that is used.

Simulation environment independent unit operations (CAPE-OPEN)

Simulation environments typically come with a number of built-in equipment models, or unit operations. For example, there may be one or more models of a heat-exchanger in a given simulation environment. This model may not match the model of another simulation environment, and more importantly, it may not match the specific type of heat exchanger that the chemical company performing the simulations is using or intends to use. Hence, the end user, the chemical company, has a requirement for a specific unit operation that is not part of the standard unit operations that come with the simulation environment. This scenario is rather common for specific reactors and separation equipment.

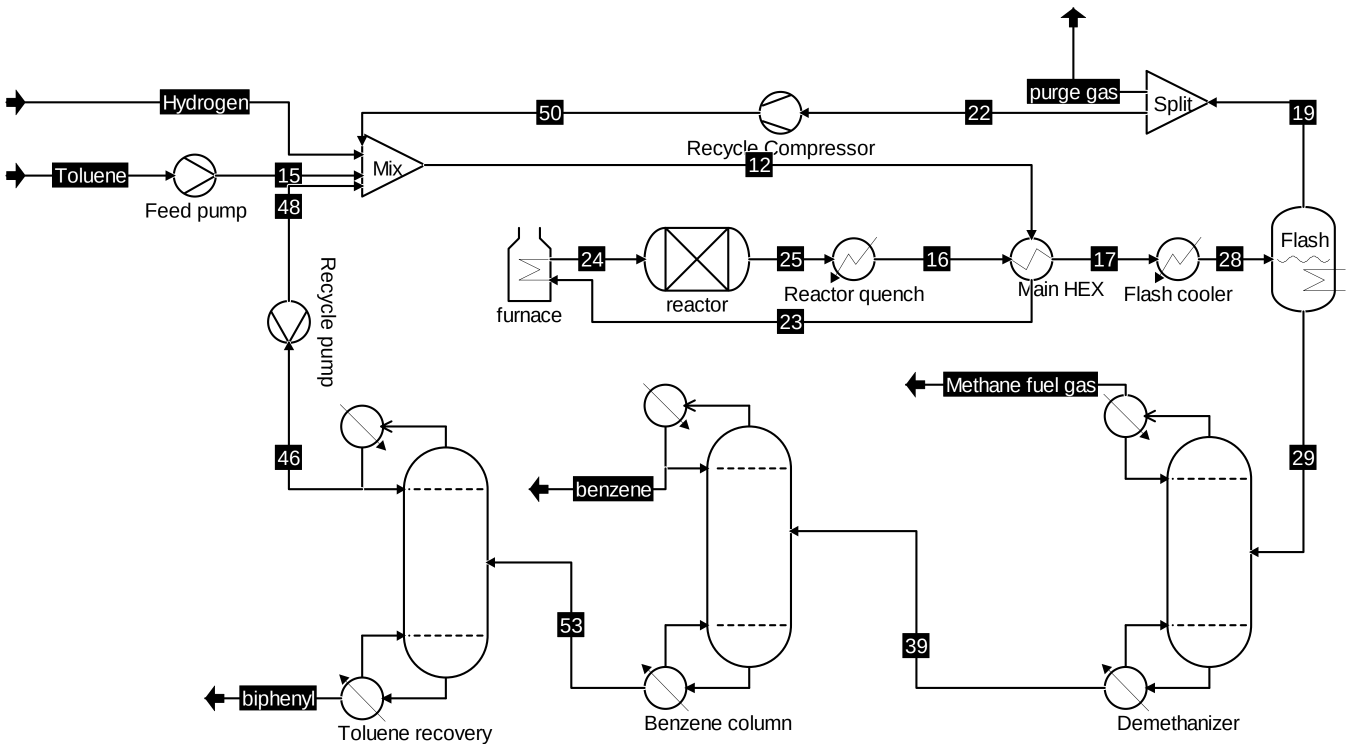


Figure 1: example of a process flowsheet, showing the dealkylation of toluene to benzene with hydrogen, as available from http://www.cocosimulator.org/index_sample.html.

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) supports built-in unit operations as well as user-defined unit operations. CAPE-OPEN aware simulation environments support inserting unit operation models that communicate with the simulation environment using CAPE-OPEN interfaces, as sketched in Figure 2.

Currently, most main-stream steady-state simulation environments support the CAPE-OPEN standards. CAPE-OPEN interfaces have been well established, validated and demonstrated. The CAPE-OPEN based unit operation connects to objects that describe the stream content in the flowsheet. The input streams can be queried; the output streams must be specified upon calculation of the unit operation. Through material stream objects, the unit operation has access to the underlying thermodynamic system of the simulation, and can thus calculate physical and thermodynamic properties, as well as perform thermodynamic phase equilibrium calculations (flash calculations).

It makes sense – when there is a need for a unit operation model that runs in multiple simulation environments – to only implement the unit operation once; CAPE-OPEN provides the means to do so.

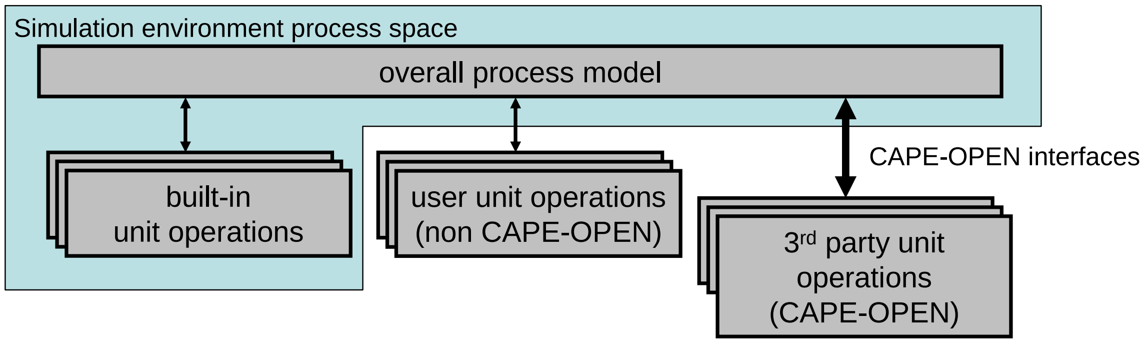

Simulation environment process space
overall process model
built-in unit operations
user unit operations (non CAPE-OPEN)
CAPE-OPEN interfaces
3rd party unit operations (CAPE-OPEN)

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

CAPE-OPEN architecture

This paragraph will show the power, but also the complexity, of unit operation development using CAPE-OPEN. Readers with a limited programming background may want to skip this paragraph; the next paragraph will demonstrate easier ways to use CAPE-OPEN.

The CAPE-OPEN interfaces are defined as a common object model structure, and there are two implementations available; COM and CORBA. COM[5] (Common Object Model) is integrated in Microsoft Windows. CORBA[6] (Common Object Request Broker Architecture) is in principle platform independent, but requires additional software to be installed to establish the connection (Object Request Broker (ORB) software). Of these two implementations, COM is currently the widely used one, and only very few CORBA CAPE-OPEN implementations are available. Thus, a COM implementation of a unit operation is what will meet the target of running in multiple simulation environments at the current state of technology.

The CAPE-OPEN unit operation interfaces are concise and intuitive[7]. In addition to the unit operation interfaces, a unit operation is expected to implement a number of CAPE-OPEN Common interfaces[8-12], as depicted in figure 3.

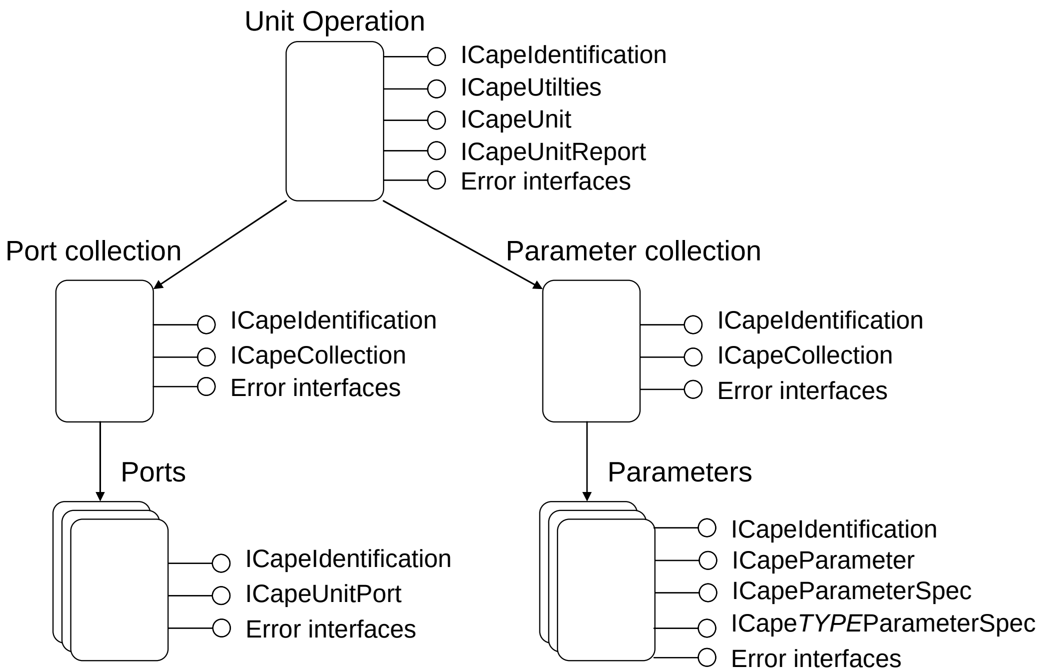


Figure 3: overview of object hierarchy and CAPE-OPEN interfaces to implement for a unit operation

The port objects that are exposed by a unit operation are connected to object describing streams of the flowsheet. It is from these stream objects that the unit operation can query feeds stream properties, and it is the unit operation's responsibility to specify all product stream objects' properties at calculation time. The stream objects are not implemented by the unit operation, but are implemented by the simulation environment.

The CAPE-OPEN thermodynamic standards at this point in time have two versions[13, 14]; Version 1.0 and Version 1.1; the former is still more widely supported, but Version 1.1 has a better design and solves a number of issues that were not well dealt with by the Version 1.0 specification. Figure 4 shows an overview of the stream objects that are connected to a unit operation's ports. The unit operation can access these objects for feed- and product port properties, but can also use the material stream objects to perform arbitrary thermodynamic or physical property calculations and flash calculations.

From Figures 3 and 4 we see that a substantial amount of knowledge of the various interfaces is required before one can commence to implement a CAPE-OPEN unit operation. More-over, additional constrains for the unit operation developer come from the COM-based implementation. The unit operation needs to be implemented as a COM server. For developers with a limited knowledge of COM programming, an option is to use a language that is native in COM, such as Microsoft's Visual Basic 6; however, Visual Basic 6 is at the end of its life cycle and no longer supported by Microsoft[15]. Another option is .NET based implementation, which is aided by CAPE-OPEN .NET class libraries[16]. However, for efficient interaction between the simulation environment and the unit operation, data marshalling by the COM framework should be avoided. This is accomplished by creating an in-process COM server (DLL) that is apartment threaded and native. Such a server can be created using programming languages such as C++. Hence, before making a final implementation of a unit operation, a developer should carefully decide what framework to use for the development.

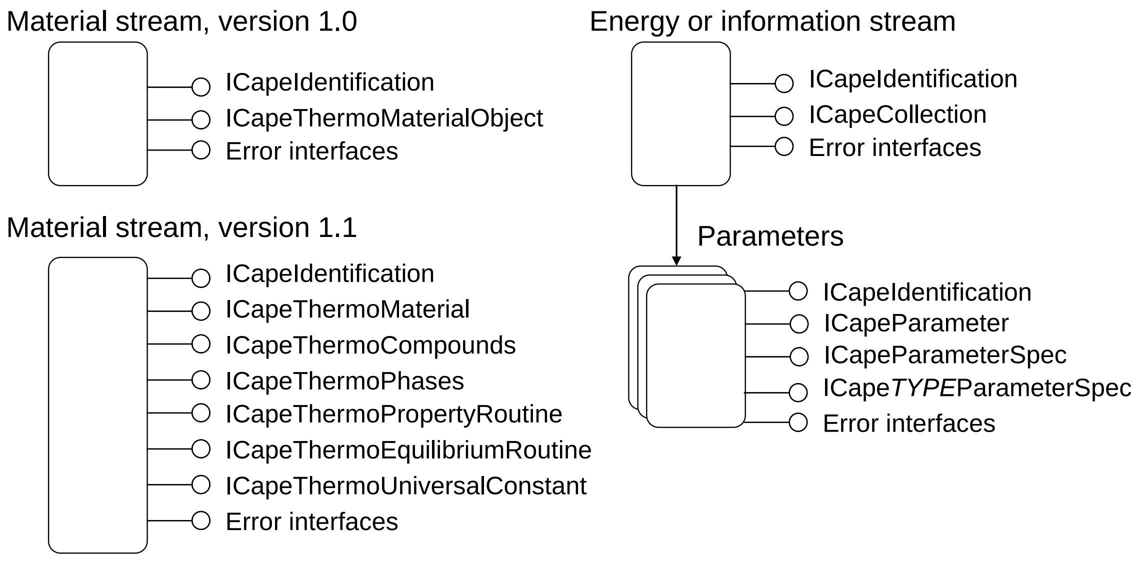


Figure 4: overview CAPE-OPEN interfaces to implemented by objects that represent streams

In addition to that, the developer may have a legacy version of the unit operation model already available, in for example Microsoft Excel, Scilab[17] or Matlab[18]. Such modeling environments have the advantage of being intuitive to the model developer, and come with a number of utilities to solve the model equations (such as the Excel Solver add-in). It would help to be able to prototype the model (build a working version of the model for testing whether the model performs as intended) using these generic modeling tools, before aiming for a final CAPE-OPEN based unit operation implementation.

Prototyping with Matlab or Scilab

Often, development of a process model is done outside of the context of a simulation environment to run in. Hence, all that is required to develop the model is access to thermodynamic and physical property calculations and flash calculations. AmsterCHEM has developed tools to do so in Matlab[19] and Scilab[20]. In addition to having the modeling environments installed, one should have an installation of the relevant thermodynamic import tools, as well as a valid CAPE-OPEN thermodynamic server. A number of such servers are available, for example the TEA server that comes with COCO[21]. End-user companies may also have their in-house thermodynamic servers available via CAPE-OPEN.

Access to thermodynamic calculations in Matlab or Scilab models therefore no longer require any knowledge of CAPE-OPEN programming. Functions are available for creation and maintenance of CAPE-OPEN thermodynamic property package. A property package handle is returned that can be used for further thermodynamic calculations. For example, in Matlab, we can create a “C1_C2” default property package of the TEA thermodynamic server, containing methane and ethane, by issuing the following command:

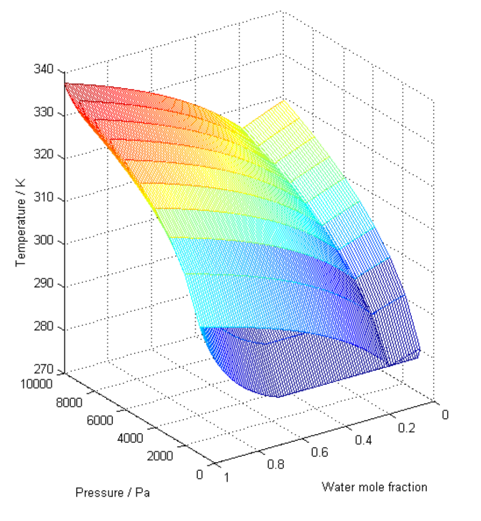
```
» handle=capeOpenGetPackage('TEA (CAPE-OPEN 1.1)','C1_C2');
```

The handle can be used in querying information about the property package that is created:

```
» capeOpenCompounds(handle)

ans =

    'Methane'
    'Ethane'

» capeOpenCompoundConstant(handle,'criticaltemperature')

ans =

  190.5600  305.3200
```

and functions are available for temperature dependent property calculations, single-phase mixture property calculations, and two-phase mixture property calculations. An example calculation of vapor enthalpy at equimolar composition, at 10^5 Pa pressure and a range of temperatures:

```
» capeOpen1PhaseProp(handle,'enthalpy','vapor',[300:20:400]',1e5,[0.5 0.5])

ans =

  1.0e+003 *

    0.0467
    0.9525
    1.8909
    2.8631
    3.8699
    4.9123
```



```
%script for calculation of phase envelope as shown on left

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;
%plot the results
mesh(P,frac,tbub);
hold on;
mesh(P,frac,tdew);
xlabel('Pressure / Pa')
ylabel('Water mole fraction')
zlabel('Temperature / K')
```

Figure 5: thermo example: calculation of a phase envelope of Water and n-Butanol using Multiflash[25]

Also methods for flash calculations are provided. A sample flash calculation at composition, 10^5 Pa and 300 K results a single vapor phase:

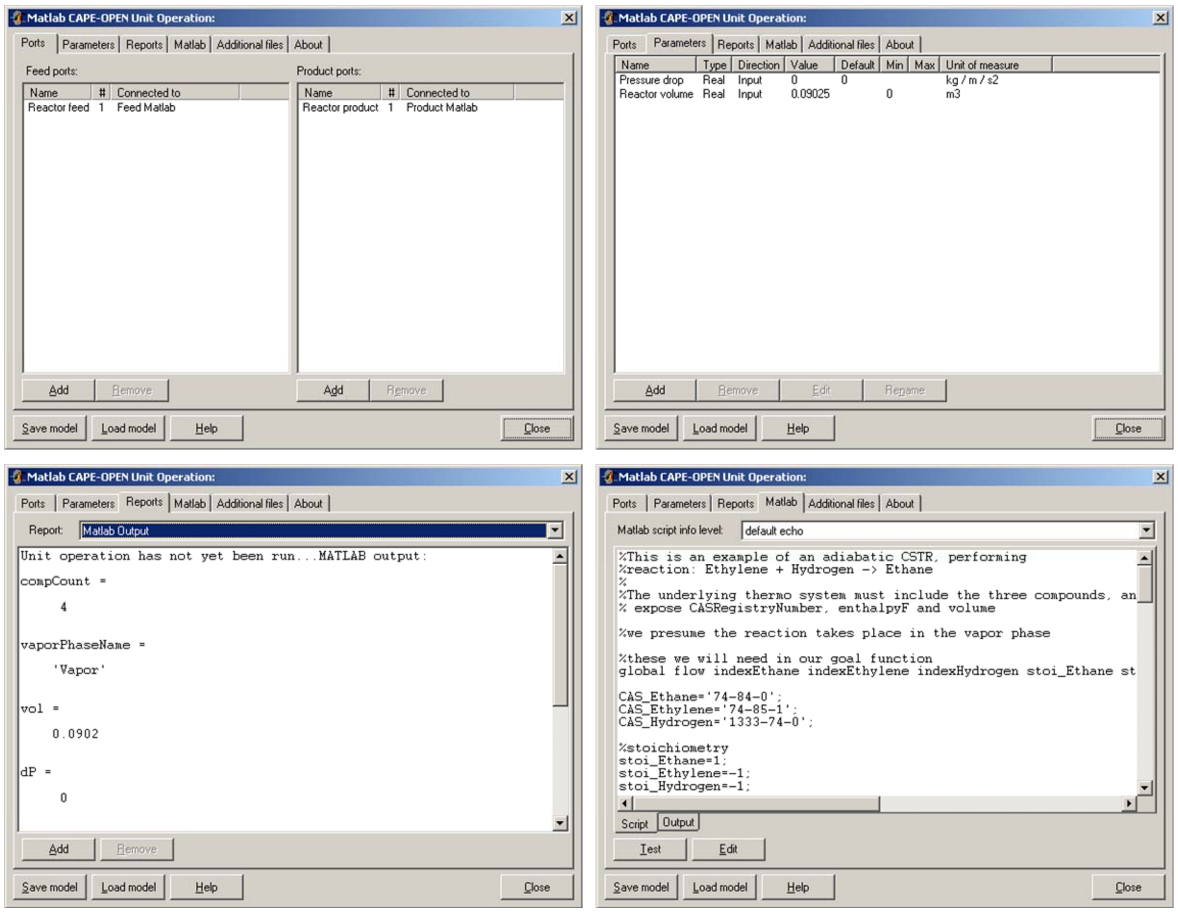
```
» [phases,phaseFracs,phaseComp]=capeOpenEquilibrium(handle,[0.5 0.5],'pressure',1e5,'temperature',300)

phases =

    'Vapor'

phaseFracs =

     1

phaseComp =

    0.5000
    0.5000
```

These commands can be issued from any script containing the model equations, or from the Matlab or Scilab command line, therefore allowing for script-based and interactive model development. Figure 5 shows an example of a phase envelope calculation.

The next step is to take the developed model and insert it into a flowsheet simulation as a CAPE-OPEN unit operation. Now, we no longer work with thermodynamic servers that are explicitly loaded, but we use the underlying thermodynamic of the simulation environment (whether CAPE-OPEN based or native to the simulation environment). Tools for CAPE-OPEN unit operation prototyping in Matlab[22] and Scilab[23] have been developed by AmsterCHEM.

Prototyping requires installation of a CAPE-OPEN compliant simulation environment, the Matlab or Scilab programs, and the unit operation tools. Definition of a unit operation starts with definition of its ports (feeds and products) and parameters (in- and outputs). This is accomplished by the unit operation configuration window, that is accessed by editing the unit operation that has been inserted into the simulation. If the unit operation will produce textual reports, these can also be configured. The output of the calculation is available as textual report by default. Finally, the script that calculates the model can be entered into the edit window. These four steps of the input are shown in Figure 6.

As example, table 1 shows a calculation script for a simple adiabatic mixer. The functions that access information via CAPE-OPEN have are highlighted.



Feed ports:

Name	#	Connected to
Reactor feed	1	Feed Matlab

Product ports:

Name	#	Connected to
Reactor product	1	Product Matlab

Name	Type	Direction	Value	Default	Min	Max	Unit of measure
Pressure drop	Real	Input	0	0			kg / m / s2
Reactor volume	Real	Input	0.09025		0		m3

Report: Matlab Output

```
Unit operation has not yet been run...MATLAB output:
compCount =

     4


vaporPhaseName =

    'Vapor'


vol =

    0.0902


dP =

     0
```

Matlab script info level: default echo

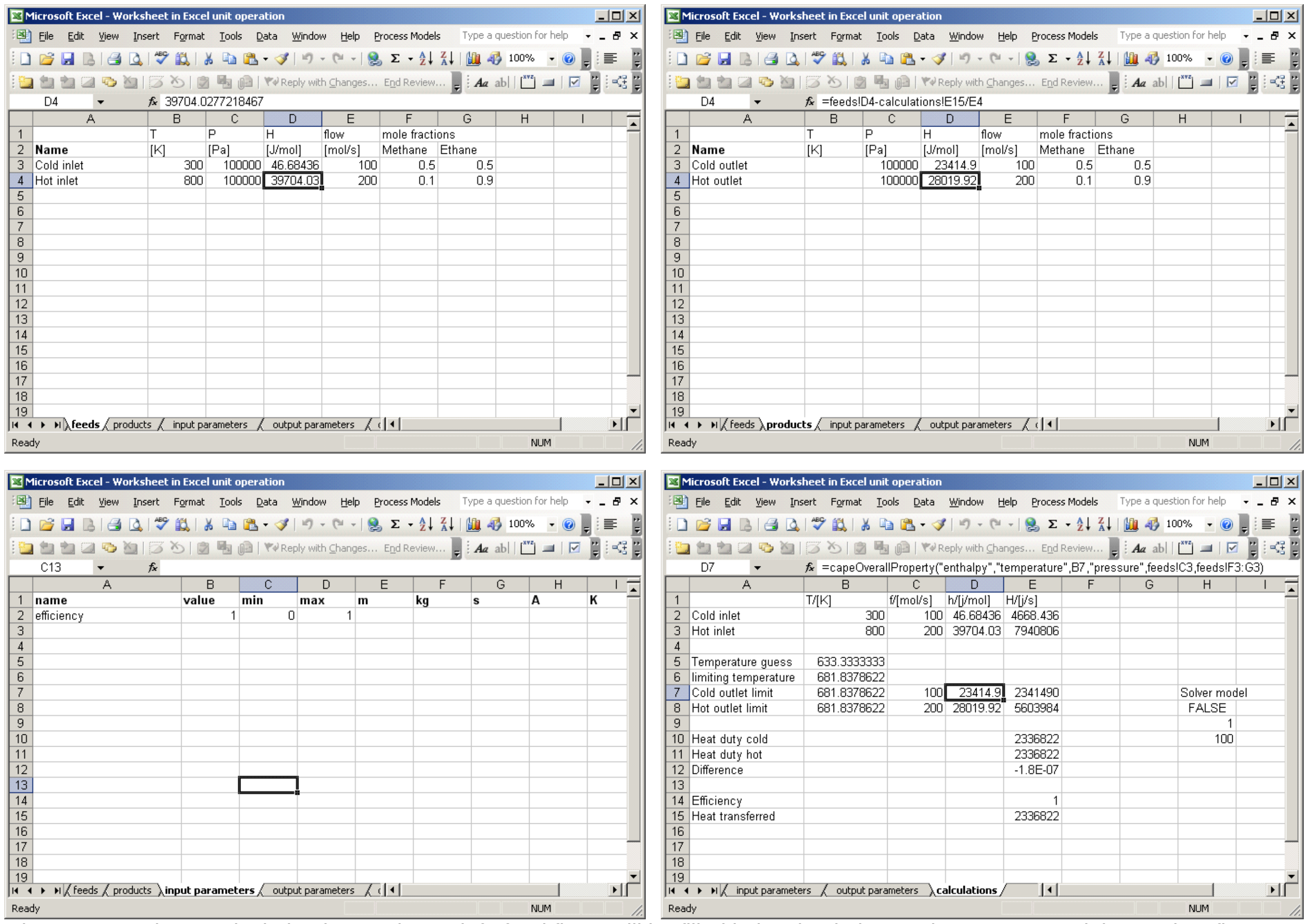
```
%This is an example of an adiabatic CSTR, performing
%reaction: Ethylene + Hydrogen -> Ethane
%
%The underlying thermo system must include the three compounds, an
% expose CASRegistryNumber, enthalpyF and volume

%we presume the reaction takes place in the vapor phase

%these we will need in our goal function
global flow indexEthane indexEthylene indexHydrogen stoi_Ethane st

CAS_Ethane='74-84-0';
CAS_Ethylene='74-85-1';
CAS_Hydrogen='1333-74-0';

%stoichiometry
stoi_Ethane=1;
stoi_Ethylene=-1;
stoi_Hydrogen=-1;
```

Figure 6: 4 steps of input required for configuring a Matlab based CAPE-OPEN unit operation. The full calculation script is available in the example that is installed with the Matlab CAPE-OPEN Unit Operation tool.

Prototyping with Excel

The prototyping in Excel is somewhat different than the script based prototyping that we have seen for Matlab and Scilab. Instead, formulas are entered in the traditional Excel approach of entering formulas that refer to cells containing the formula's inputs. The Excel Unit Operation does not have a dedicated user interface; instead, when editing, the Excel program will appear. The special Feeds and Products work-sheets contain a table of feed and products and their properties. Adding and removing feeds or products is as simple as adding rows to this table. The pressure, temperature, composition and enthalpy of each of the feeds is listed on the Feeds work-sheet. The task of the modeler is to present formulas in Excel that will provide values for each product stream, for composition and 2 out of 3 of pressure, temperature and enthalpy. Similarly, input and output parameters are configured simply by changing the parameter table in the Excel workbook that defines the unit operation.

Special facilities are available in the Control worksheet, allowing for specification of solution status, returning solution error messages and configuring the Excel Solver add-in to solve sets of equations.



D4 = 39704.0277218467

	A	B	C	D	E	F	G
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

D4 =feeds!D4-calculations!E15/E4

	A	B	C	D	E	F	G
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

C13

	A	B	C	D	E	F	G	H	I
1	**name**	**value**	**min**	**max**	**m**	**kg**	**s**	**A**	**K**
2	efficiency	1	0	1					

D7 =capeOverallProperty("enthalpy","temperature",B7,"pressure",feeds!C3,feeds!F3:G3)

	A	B	C	D	E	F	G	H
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			

Figure 7: entering a calculation in Excel. Top left: feed flows will be filled in by simulation environment. Top right: product flows require specification by formulas. Bottom left: parameters are defined in separate sheets; calculation can refer to the values that are filed in by the simulation environment. Bottom right, the calculation can use thermodynamic calculation functions.

Current status

The Matlab, Scilab and Excel unit operation prototyping tools are implemented with support for CAPE-OPEN Version 1.1 thermodynamics, and will run in simulation environments that are compliant with this standard. Operation has been tested in the COFE simulation environment of COCO[21] and in the ProSimPlus simulation environment of ProSim[24]. The thermodynamic import utilties for Matlab have been tested with COCO's TEA, Infochem's[25] Muliflash and ProSim's Simulis. All of the prototyping tools are available from http://www.amsterchem.com/ and are free for non-commercial, academic or evaluation use.

Conclusions

We have established in this article the need for unit operation model developers to base their implementations on CAPE-OPEN in order to have a simulation environment independent implementation. However, the threshold for implementing a model from scratch into a CAPE-OPEN framework is considerable. Tools for rapid prototyping and testing can aid in the development process of unit operation models. Software frameworks to do so have been introduced in this article. The modeling environments available to the developer are Microsoft Excel, Matlab and Scilab.

Table 1: calculation script for a simple adiabatic mixer

```
% Simple adiabatic mixer example
%   works for any number of feeds
%   special case for zero flow

numFeeds=getNumberOfFeeds
totH=0;
compFlows=[];
minP=[];
for i=1:numFeeds
 f=getFeedProp(i,'flow');   %component flows
 compFlows=([compFlows;f]); %make into a matrix of nfeed x ncomponent
 ftot=sum(f);               %total flow for this feed
 totH=totH+ftot*getFeedProp(i,'enthalpy');    %add enthalpy to total enthalpy
 minP=min([minP getFeedProp(i,'pressure')]);  %keep track of minimum feed pressure
end
if numFeeds>1; compFlows=sum(compFlows); end  %sum all component flows
totFlow=sum(compFlows)                        %total flow
if (totFlow==0) then
 %zero flow, use average feed composition and temperature
 x=[];
 T=0;
 for i=1:numFeeds
  x=sum([x;getFeedProp(i,'fraction')]); %add composition
  T=T+getFeedProp(i,'temperature');     %add temperature
 end
 x=x/numFeeds                           %average composition
 T=T/numFeeds                           %average temperature
 setProduct(1,0,x,'pressure',minP,'temperature',T)              %define product by T and P
else
 x=compFlows/totFlow
 setProduct(1,totFlow,x,'pressure',minP,'enthalpy',totH/totFlow) %define product by P and H
end
```

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. Rogerson, D. (1997). Inside COM. Redmond, Washington, Microsoft Press.
6. CORBA specifications: http://www.omg.org/technology/documents/spec_catalog.htm
7. Unit Operations Interfaces, available from http://www.colan.org/index-33.html
8. Utilities Common Interface, available from http://www.colan.org/index-35.html
9. Identification Common Interface, available from http://www.colan.org/index-35.html
10. Persistence Common Interface, available from http://www.colan.org/index-35.html
11. Parameter Common Interface, available from http://www.colan.org/index-35.html
12. Collection Common Interfaces, available from http://www.colan.org/index-35.html
13. Thermodynamics and Physical Properties Interfaces, version 1.0, available from http://www.colan.org/index-33.html
14. Thermodynamics and Physical Properties Interfaces, version 1.1, available from http://www.colan.org/index-37.html
15. http://msdn.microsoft.com/en-us/vbrun/ms788707.aspx
16. http://www.epa.gov/nrmrl/std/mtb/cape/ClassLibraryDocumentation.pdf

17. http://www.scilab.org/
18. http://www.mathworks.com/
19. http://www.amsterchem.com/matlabthermo.html
20. http://www.amsterchem.com/scilabthermo.html
21. COCO is freely available from http://www.cocosimulator.org/
22. http://www.amsterchem.com/matlabunitop.html
23. http://www.amsterchem.com/scilabunitop.html
24. http:/www.prosim.net/
25. http://www.infochemuk.com/