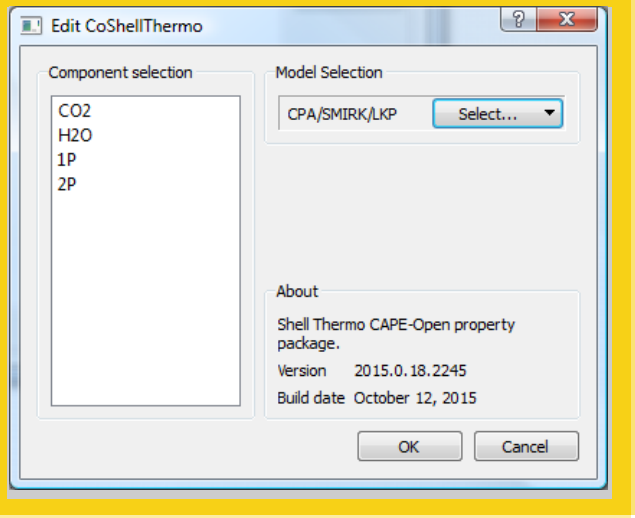
CHALLENGES REPLACING NATIVE EXTENSIONS WITH A CAPE-OPEN PROPERTY PACKAGE



Mark Stijnman
Developer Thermodynamics

OUTLINE

Background and motivation

Challenges:

- For us
- For the simulator vendors
- For the CAPE-Open standard

Conclusion

Questions/Discussion

BACKGROUND: WHO AM I?

Mark Stijnman

- Background in Scientific Computing
- With Shell since 2007 in thermodynamics team
- Learned thermodynamics on the job
- Work on Shell thermodynamics software
  - Shell thermo model library
  - Standalone applications
  - Extensions for 3rd party applications
  - Supporting tools
  - Etc

BACKGROUND: WHY HAVE SHELL THERMO?

Why not just rely on commercial software? Own software allows us to:

- Do better than market
  - Use Shell measured data
  - Use Shell-only models
- Experiment
- Understand the models and algorithms at a deep level
- Use commercial packages where they are better, and interface with them to provide
  - access to Shell models/data
  - functionality that isn't supported
  - consistency between packages

BACKGROUND: EXISTING INTERFACES

Shell Thermo is used in:

- Honeywell UniSim Design
  - Using extension interface
- Schneider Electric Pro/II
  - Using custom integration

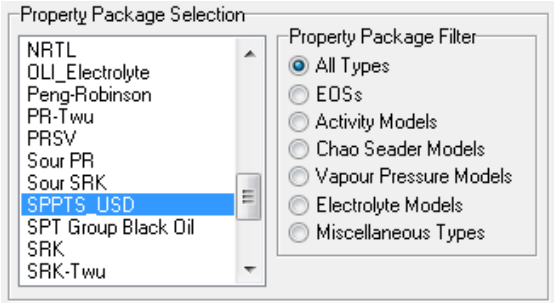
We would like to use it in:

- AspenTech Aspen Plus
- Matlab
- Others

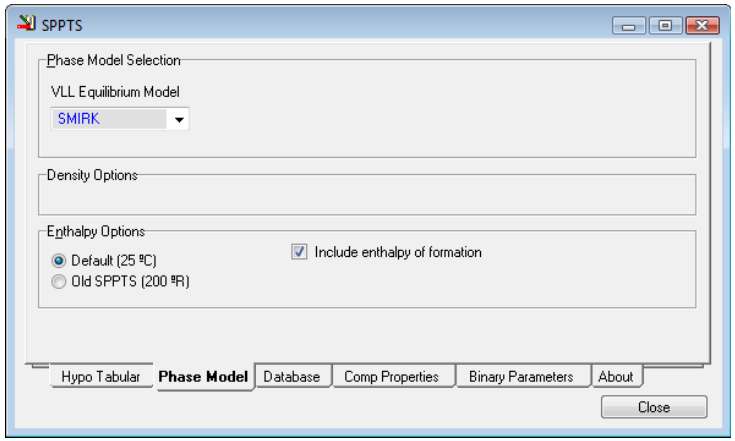
BACKGROUND: EXISTING INTERFACES

UniSim Design:

- Property package available among built-in packages



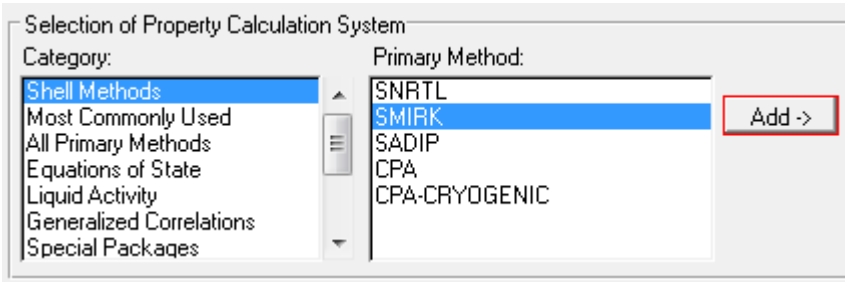
- Select Components using built-in component selection, including hypo/pseudo components
- Custom options dialog (uses USD user interface definition language)
- Distributed as part of a separate collection of add-ons



BACKGROUND: EXISTING INTERFACES

Pro/II:

- Property package available among built-in packages



- Select Components using built-in component selection, including hypo/pseudo components
- No custom dialog though
- Distributed by vendor

MOTIVATION

We want to:

- Reduce software maintenance costs
- Provide a more uniform experience across platforms
- More easily support other platforms
- Lower dependencies on vendors

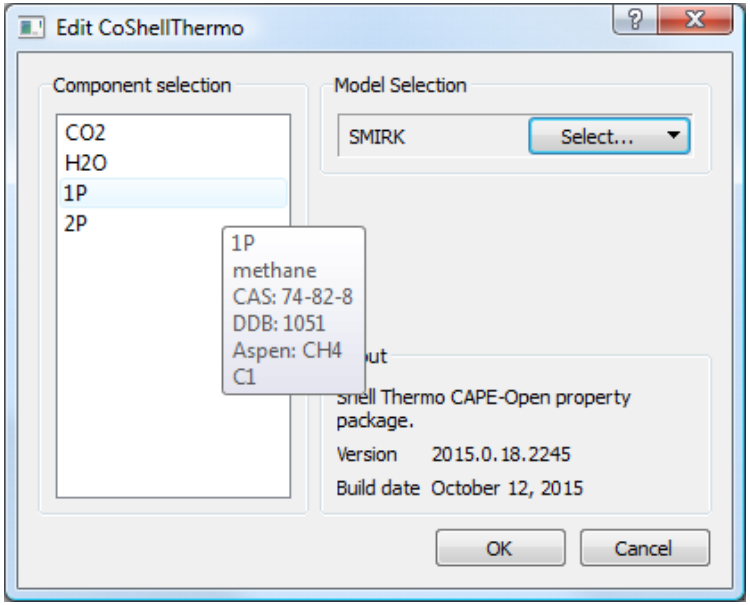
Solution:

- CAPE-Open

COSHELLTHERMO

So we made a CAPE-Open property package "CoShellThermo":

- Beta implementation available
- Works in COFE, USD, ProII, Matlab and others
- Supports CAPE-Open 1.0 and 1.1
- Custom user interface
  - Edit component list
  - Choose phase model
- Persistence support (Save/Load)



THE BIG QUESTION

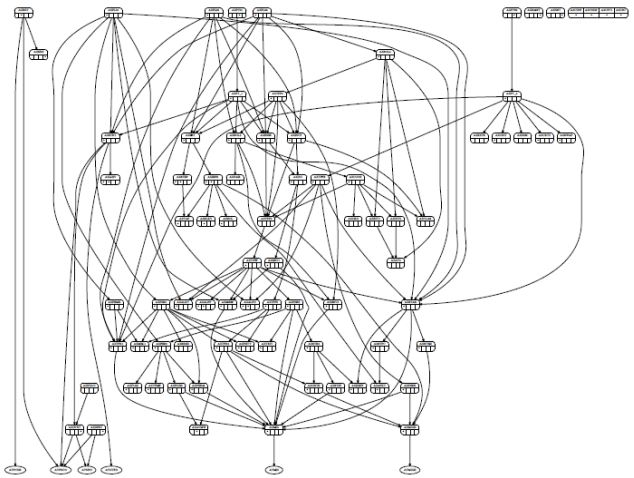
What is needed to make CoShellThermo a viable replacement to the vendor-specific property packages?

- Make CAPE-Open packages as easily available as native packages
- Allow workflow to be close to native workflow
- Make performance close to native packages
- Support custom dialogs

CHALLENGES: FOR US

We have a working property package CoShellThermo, but:

- Uses legacy code that is not re-entrant or thread-safe
  - Not safe to create multiple instances until this is fixed
  - Current extensions have workarounds, but now proper solution is needed
- Needs more user friendly interface
  - Including separate package editor



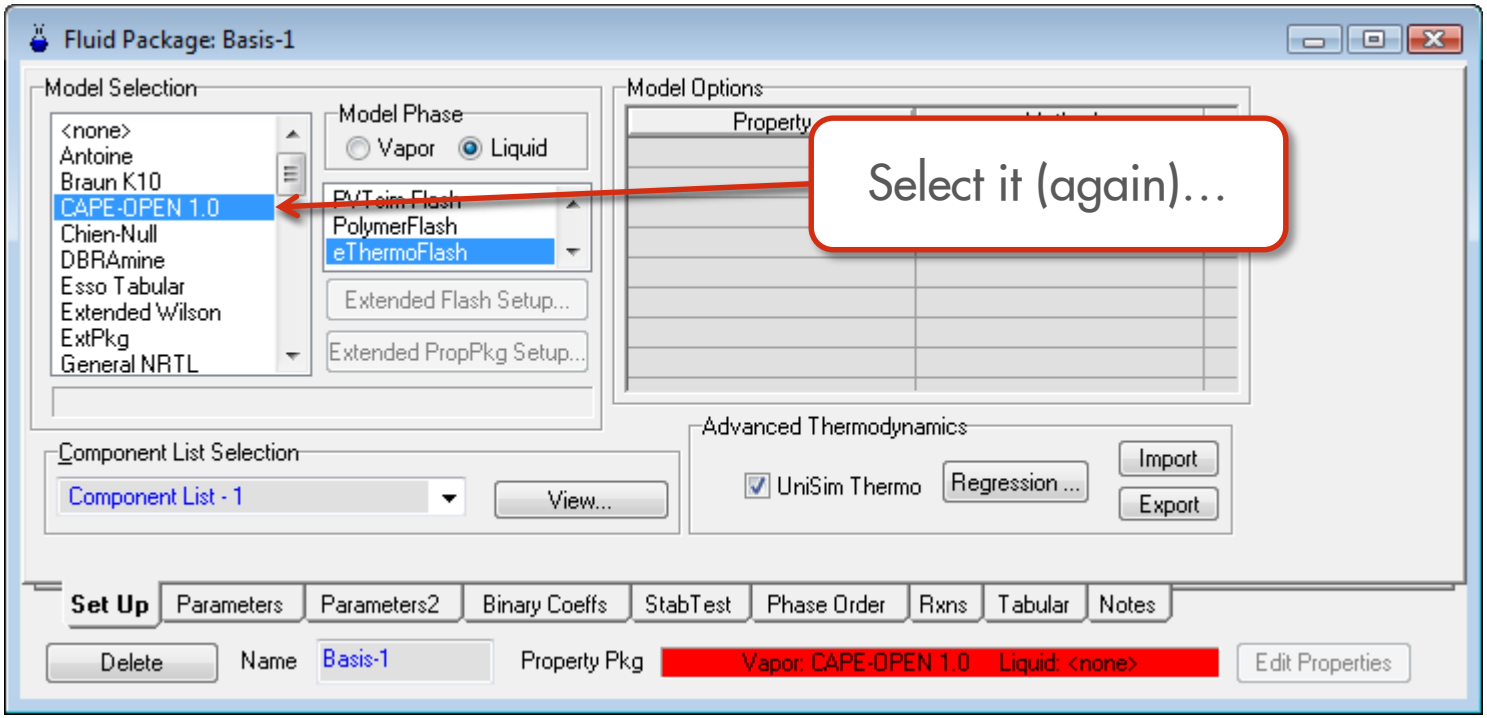
CHALLENGES: FOR SIMULATOR VENDORS

Unfortunately:

- CAPE-Open packages often work notably different from standard packages
- Performance usually considerably lower
- Often doesn't support custom dialogs and/or persistence

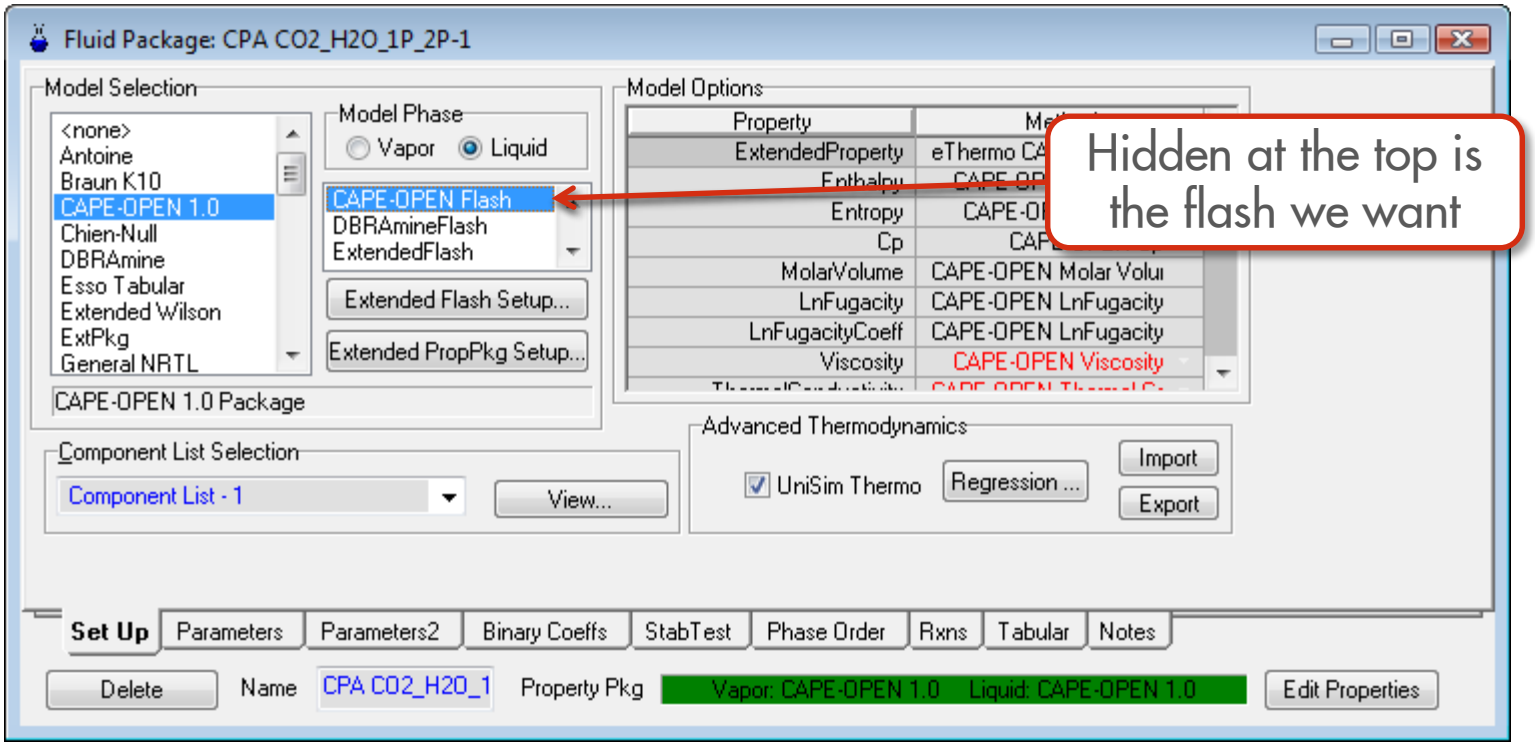
EXAMPLE: UNISIM DESIGN

How to add a CAPE-Open Property Package:



EXAMPLE: UNISIM DESIGN

Done! But… Why is it so slow?



CHALLENGES: FOR SIMULATOR VENDORS

- Performance:
  - In USD, CAPE-Open is ~30% slower compared to native extension
    - Worse when using external phase equilibrium solver
  - A lot of overhead in checking component list
  - V1.1 support is likely necessary to get the expected performance
- Absolutely need Edit and persistence support (Save/Load)

CHALLENGES: FOR THE CAPE-OPEN STANDARD

If CAPE-Open is to be an alternative for native interfaces, CO-LAN should:

- Set a stricter standard
- Change the standard to allow a workflow closer to the native workflow

CHALLENGES: A STRICTER STANDARD

CAPE-Open is often not specific enough, and too much is optional

- Edit/Save/Load support not mandatory, just recommended
  - No clear specification what should happen after an Edit
    - Reload component list?
    - Reload property list?
- Support V1.1 is still not mandatory to be considered CAPE-Open compliant
- What to do when a CAPE-Open property package offers components not supported by the simulator?
- What flash types should a Property Package offer? Should a simulator be required to provide the missing ones?
- Similarly with phase properties?

CHALLENGES: BEST PRACTICES

Alternatively, all of these issues have solutions that are in actual use, that could be captured in a set of official “Best Practices” guidelines.

CHALLENGES: WORKFLOW ISSUES

Common workflow in process simulator:

- Start a new case with a blank component list
- Select a package
- Interactively add and remove components using the component selection tools already available in the simulator
  - Pick from a list of available components
  - Create and edit pseudo/hypo/petroleum fraction components

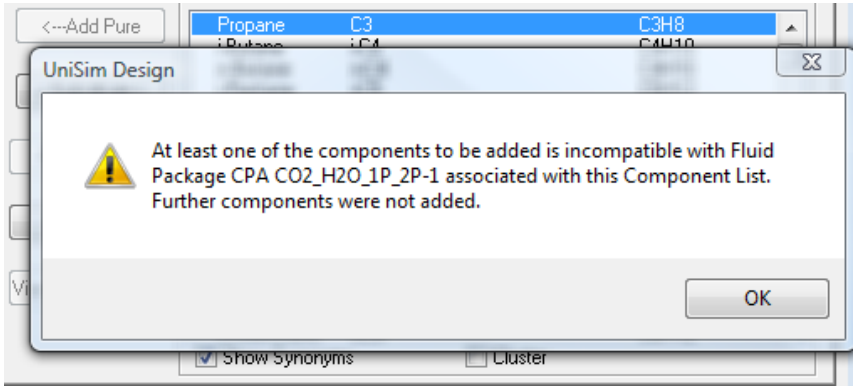
Could be done in custom edit dialog, of course, but:

- Not always supported
- Not always intuitive

CHALLENGES: WORKFLOW ISSUES

Instead, with a CAPE-Open property package:

- Must start with a pre-existing package
  - Or create one using an external application
- Component list now fixed
  - May create sub-selection, but can't expand selection
  - Unless Edit dialog has it (and is supported)
  - Native component selection interface no longer works as user expects



CHALLENGES: WORKFLOW ISSUES

Or what about this somewhat common workflow:

- Take an existing case, with existing component list, pseudo components, etc.
- Change the property package to something else.

This is absolutely required if we want to convert existing cases from the native extension to its CAPE-Open replacement

CHALLENGES: WORKFLOW ISSUES

This would likely require an interface extension:

- Some way to add components to a package:
  - AddComponent(…)
  - CreatePseudoComponent(…)?
- Some way to get information on supported components:
  - IsComponentSupported(…)?
  - GetSupportedComponents(…)?

…which would then of course require vendor support.

CONCLUSION

For now, CAPE-Open property package CoShellThermo will not replace the native extension interfaces, until

- CoShellThermo allows multiple instances
- Process simulators improve their CAPE-Open property package support
- CAPE-Open standard supports more interactive component addition and creation

So instead of replacing two interfaces with one, we now have three…

On the other hand:

- We can now use Shell thermo in all sorts of new contexts
- Future integration efforts may become easier

Q & A and Discussion