Towards CAPE-OPEN 2.0

Thinking about the future

Mark Stijnman
Thermodynamics Expert

Disclaimer

The opinions in this talk are my own and don't necessarily reflect those of Shell, CO-LaN or any of its SIGs.

Goal for this talk

- Identify changes in the computing landscape
- Identify changes in the process simulation landscape
- Identify pain points in the current standard
- Present ideas on how CAPE-OPEN could adapt

→Start a discussion of where CAPE-OPEN could/should be heading in the next years

- Plant seeds for a CAPE-OPEN 2.0
- Assumptions:
  - Allow breaking changes

Overview

1. Landscape changes
2. Thermo
3. Unit Operations
4. Cloud

Landscape changes

Computing landscape

Multi-core and cloud

- The future is multi-core
  - Computing power currently mostly grows by increasing the number of cores
    - Especially true for virtual machines in the cloud
  - Most process simulation software currently single-threaded
    - This is expected to change in next-generation simulators
- The future is cloudy
  - Simulators will move to the cloud
    - Even if only as desktop applications running on a virtual machine in the cloud
  - How to deal with plugins in a cloud environment?

Process simulation landscape

Dynamics and equation-oriented modeling

- The future is dynamic
  - Steady-state often no longer enough
  - Transient processes, startup, shutdown
- The future is equation-oriented
  - Can be used for steady-state, dynamics, optimization etc.
  - Sequential solving limits multi-core usage, while simultaneous allows better leverage of multi-core computing
  - Still in its infancy as compared to sequential modeling
  - Needs to support “black-box” models and/or “classic” unit operation modules

Thermo

2

Thermo: life cycle

Some pain points in property package life cycle:

- Property Package can be stand-alone
- Property Package can be created from a preset in the Package Manager
  - No way to create an empty or default package, to be configured or restored from a case file later
  - If a preset doesn't exist in the package manager, this is not an error
    - This is essentially a workaround for the above issue
- Package can be edited
  - Should the edits propagate back to the preset used?
    - And to other packages using the same preset?
    - Currently, the answer generally is no to both
- Package must be initialized
  - This is where you may get errors that originated way earlier in the process

Thermo: multi-threading

Some pain points in property packages with multi-threading:

- Property Packages are inherently unsafe for multi-threading
  - Even if underlying property calculations are thread-safe
- The cause: the context material
  - Simulator must call SetMaterial for each material object it wants properties for
    - The material object is used to store the outcome of the phase equilibrium and/or property calculations
    - This precludes using the same package on multiple material objects simultaneously
  - A Material may also define subsets of component lists and phase lists
    - May require partial or complete reload of the underlying model
  - A current workaround would be to create multiple copies of a property package
    - Use save/load functionality to initialize them to the same state

Thermo: proposed solution

Property packages become stateless

- Model a Property Package as a stateless, pure function object
  - It calculates phase properties as a function of T, P and phase composition
  - It calculates phase equilibrium as a function of input conditions
- Simplify life cycle
  - Once created, never changes
  - Is either created successfully, or not at all
    - No invalid property packages possible
- Move complexity to the Package Manager
  - Editing
  - Saving/Loading
  - Preset management

Thermo: proposed solution

Some additional upsides:

- All inputs and outputs are explicit
  - No side-effects
  - Possible performance improvements: no need to move data into and out of a material object if its not needed
- Define additional functionality on package manager:
  - Add/remove components
    - Integrate with native component selection user interface
    - Define pseudo components
- No cyclic references
  - A Material is associated with a property package, but a property package is no longer linked to a material.

Thermo: proposed solution

Some downsides (and possible mitigation):

- No more support for subsets of component lists or phase lists
  - Proposal is to move this to the package manager as well
- Might introduce a lot of extra complexity for the simplest cases
  - Stand-alone property packages might suddenly need a package manager
- Property packages may still have hidden state that will need to be protected
  - E.g. temperature-dependent that you want to store and only re-calculate when the temperature changes
    - Possible solutions: thread-specific storage, thread-safe least-recently used cache, etc
  - May be necessary to allow package authors to indicate thread-safety level
    - Simulators may apply workarounds as needed, like creating multiple copies with the same settings

Unit operations

3

Unit operations: equation-oriented modeling

Equation modeling

- There is currently no industry-standard equation modeling language
  - Do we want one?
  - Can CAPE-OPEN play a role here?
- Black-box modeling
  - Implement equations in own code, return residuals
    - Mostly for IP-protection reasons, but may also be more portable
    - Derivatives generally also needed
  - Need some way to support “classic” unit operations in an equation-oriented environment
    - For backwards compatibility or easy upgrade path
    - Who implements the wrapping code? Can we standardize this?

Unit operations: custom GUI

There is a need for custom user interfaces for unit operations

- Modal "Edit" functionality is often not enough
  - Real-time display of current conditions
  - Widget on the flowsheet showing graphs, tables, etc
- With COBIA, needs to be cross-platform
  - Suggestion: use HTML5/CSS/Javascript
    - Open standards
    - Many frameworks available (React, Angular, Vue)
    - Provide callback mechanism to push data to the widget (JSON)
    - CAPE-OPEN could standardize element class names so that host apps may style the GUI to fit their own style

Cloud

4

Cloud: Simulators are moving to the cloud

Simulator cloud strategies

- Provide a virtual machine to users that contains the software
  - Simple and quick solution: pretty much runs the desktop version on a remote desktop in the cloud
- Provide a (thin) client that talks to a web API or server that runs in the cloud
  - If the software doesn't have a client-server architecture yet, this can require a costly redesign

Cloud: But what to do with plugins?

In both cases, plugins are a problem

- The easy solution would be to simply install the plugin on the cloud instance
  - But not all users that access a particular virtual machine or web API will be allowed access to the same plugins
    - Licensing, IP protection
- Alternatively, for CAPE-OPEN we could establish a protocol where a CAPE-OPEN extension can be published to the cloud
  - Only publish meta-data to CAPE-OPEN, such as UUID and URL
  - Hosted on supplier's own cloud, so they can control their own authentication and authorization
  - Binaries (DLLs and data files) downloaded on the fly, no installation required
    - Might still need additional license protection
  - Could be implemented in CAPE-OPEN middleware (COBIA)

Cloud: A CAPE-OPEN web API?

A Web API for CAPE-OPEN could also be part of the solution

- Implement your plugin as a web API for your calculations
  - Instead of a DLL and data files that are downloaded to a potentially untrusted PC or virtual machine, the calculations safely take place in your own cloud
  - We could define a CAPE-OPEN standardized web API that mirrors the DLL-based API
    - Based on REST principles, or GraphQL, or similar open standards
    - As close to a 1:1 mapping as possible
  - Big downside of a web API is latency and overhead of message construction and parsing
    - Design the API to allow bundling multiple calculation requests in one message
      - The simulator may also have to be restructured to benefit from this

Questions and Answers