

**Methods and Tools
Special Interest Group Report
CAPE-OPEN 2018 Annual Meeting
Ludwigshafen**

**Michael Hlavinka
Bryan Research & Engineering, LLC**

10 October 2018

SIG Membership

[Bill Barrett](#)

US EPA

[Jasper van Baten](#)

AmsterCHEM

[Tony Garratt](#)

ANSYS

[Daniel Wagner](#)

DWSIM

[Mark Stijnman](#)

Shell Global Solutions

[Michael Hlavinka](#)

Bryan Research & Engineering, LLC

[Loic d'Anterroches](#)

Céondo GmbH

[David Jerome and](#)

[Krishna Penukonda](#)

AVEVA

M&T SIG Ongoing Activities

- ◆ **Common Interface conference calls**
 - ⇒ **First Wednesday of the month**
- ◆ **Threading conference call**
 - ⇒ **Third Wednesday of the month**
- ◆ **Object Model conference call**
 - ⇒ **Fourth Wednesday of the month**
- ◆ **Join? Please contact either SIG Leader or CTO**
 - ⇒ **Bill Barrett – barrett.williamm@epa.gov**
 - ⇒ **Michel Pons - technologyofficer@colan.org**

M&T SIG Charter

- ◆ Improve integration, and expand utilization of Computer-Aided Process Engineering (CAPE) applications within the enterprise through identification and resolution of existing cross-cutting issues with the CAPE-OPEN platform, develop mechanisms for use of CAPE within other application domains, and incorporate advances in information technology into the CAPE-OPEN platform.

- ◆ **Key responsibilities**
 - ⇒ Resolve issues with the common interface specifications.
 - ⇒ Develop and maintain standards and protocols for CAPE-OPEN implementations.
 - ⇒ Incorporate advances in information technology into the CAPE-OPEN protocols.
 - ⇒ Identify novel uses of CAPE and provide standards for utilizing CAPE within these applications.

No Change to vision and responsibilities.

M&T SIG 2017/2018 Activities

◆ COBIA Project

⇒ Specification:

- ⇒ COBIA Parameters Interfaces
- ⇒ COBIA Persistence Interfaces
- ⇒ COBIA Reporting Interface

⇒ Training

- ⇒ Training Session prior to this meeting.

⇒ Full native implementation of COBIA

- ⇒ Released for testing and evaluation.

⇒ Preparation for Distribution

- ⇒ Licensing Recommendation to Management Board.

◆ Interaction with Interoperability SIG:

⇒ Versioning

⇒ Support for Certification

⇒ Testing COBIA against its specifications

Flowsheet Monitoring Interface

- ◆ **New Interface Specification**
 - ⇒ Provides the ability to access all elements in a flowsheet without interfering with the flowsheet
 - ⇒ Can respond to event notifications for modification of flowsheet configuration.
- ◆ **Ready for Request for Comments (RFC).**
- ◆ **Prototype of preliminary version of interface COFE and WAR Algorithm can be updated following RFC.**
- ◆ **RFC starts on November 1.**

COBIA Project Roadmap

- ◆ Phase I – Proof of Concept **Completed**
 - ◆ Core technical components
 - ◆ Demonstrate COM/COBIA interoperability with Thermo 1.1 interface set
- ◆ Phase II – Full Windows Native **Currently Beta Testing**
 - ◆ Expanding COBIA to all interfaces of business value
 - ◆ Support for C/C++ development.
 - ◆ Allow development of fully functional COBIA-based PMEs and PMCs
- ◆ Phase III – Interoperability (**Future**)
 - ◆ Microsoft .NET is planned
 - ◆ Other platforms as identified by CO-LaN membership

COBIA Timeline

- ◆ **October 2016 - Phase I completed**
- ◆ **October 2017 – Phase II status presented and demonstrated**
- ◆ **October 2018 –Beta version of COBIA (Ready for release)**
 - ⇒ Testing of COBIA
 - ⇒ COBIA Training.
- ◆ **FUTURE**
 - ⇒ Verification of COBIA against specifications for release.
 - ⇒ Formalize COBIA Documentation
 - ⇒ .NET language bindings (Intend to develop first)
 - ⇒ Other language bindings (Develop as needed).
 - ⇒ CO-LaN will maintain COBIA codebase and provide updates as needed.
 - ⇒ Subject to business needs/Management Board Approval.

COBIA 2017/2018 Activities

- ◆ Query to Thermodynamics and Unit SIGs Regarding inclusion of Thermodynamics 1.0 Interfaces: *COBIA only Thermo 1.1+*
- ◆ Develop and Prototype Interfaces
 - ⇒ Persistence
 - ⇒ COM to COBIA Transitioning
 - ⇒ Parameters
 - ⇒ Reporting
 - ⇒ Error Handling
- ◆ Threading Model
- ◆ COBIA Training prior to the CAPE-OPEN 2018 Annual Meeting
 - ⇒ 8 Attendees
 - ⇒ Developed a COBIA-based unit operation PMC

CAPE-OPEN Interface Updates

- ◆ **COBIA Project includes updates of CAPE-OPEN interfaces.**
 - ◆ *Evaluation of existing interfaces for inclusion in COBIA*
- ◆ **Versioning Issue**
 - ⇒ Updating of interfaces as part of the COBIA project requires development of a versioning scheme.
 - ⇒ Interop SIG leading the versioning scheme definition with support from M&T SIG.
 - ⇒ Terminology – COBIA reflects revised interfaces, COM reflects CAPE-OPEN 1.1 interface set for this presentation.

Persistence Interfaces in COBIA

◆ Goals

- ⇒ Less ambiguity/fewer interfaces than COM persistence.
- ⇒ Use of platform-native serialization mechanisms.
 - No special developer knowledge required, e.g. .NET development should only require use of `Serializable` attribute.
 - Alternatively, developer can implement COBIA persistence interfaces.
- ⇒ Allow use of human-readable formats such as JSON or XML

◆ Design decision

- ⇒ Explicit separation of object serialization from storage to persistent media.
- ⇒ COMBIA handles COM persistence interoperability.
- ⇒ Limited number of persistence interfaces.

◆ Status – Implemented in the Beta version of COBIA.

Persistence Interfaces, Cont'd.

◆ *ICapePersist Methods:*

- ⇒ *Save* – Asks the PMC to save itself using the *ICapePersistWriter* interface.
- ⇒ *Restore* – Restores the PMC using the *ICapePersistReader* interface.

◆ *ICapePersistWriter*

- ⇒ Exposed by PME persistence Object.
- ⇒ Provides methods to write standard CAPE-OPEN data types and byte array.

◆ *ICapePersistReader*

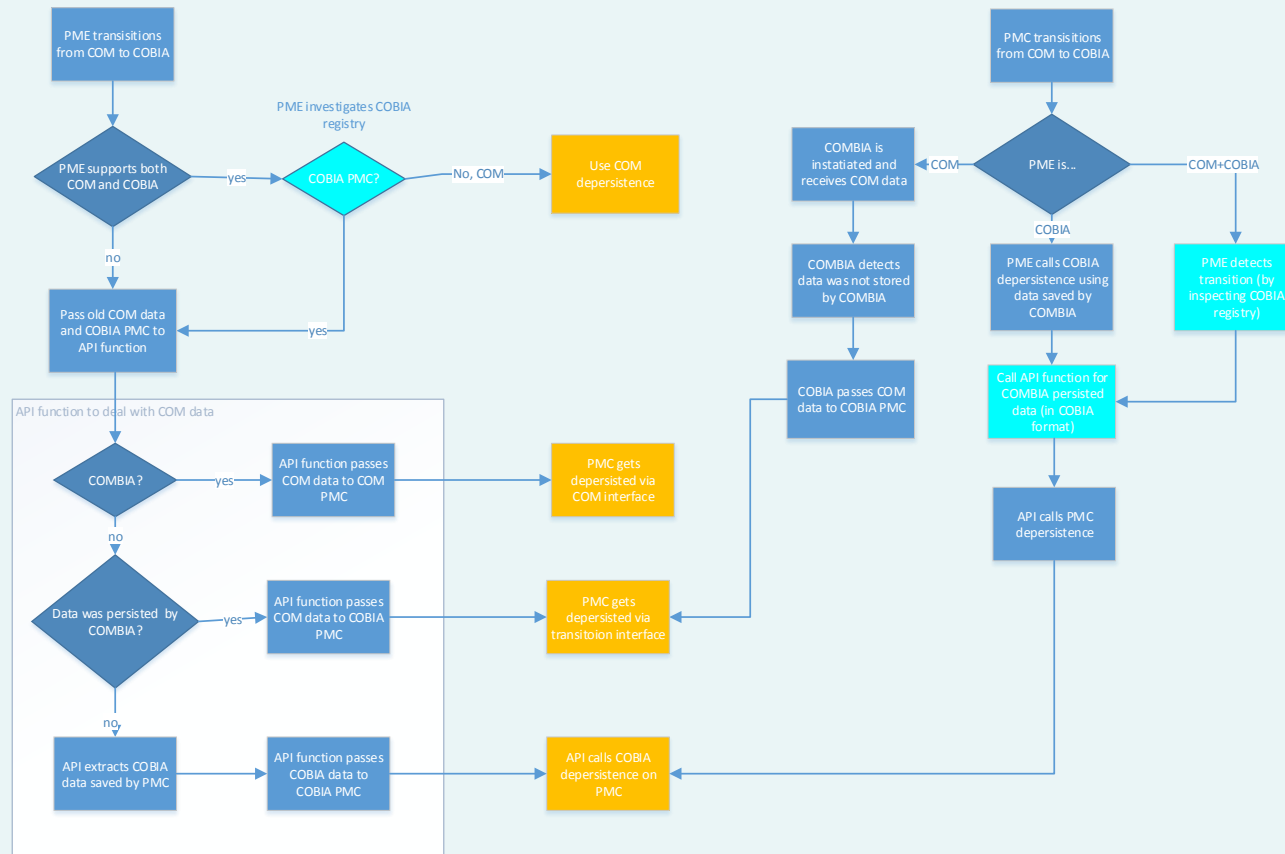
- ⇒ Exposed by PME persistence Object.
- ⇒ Provides methods to read standard CAPE-OPEN data types and byte array.

COM to COBIA Transitioning

◆ Support in COBIA/COMBIA for Transitioning PMCs and PME from COM to COBIA

⇒ Updating of persistence from COM to COBIA.

⇒ Developed Flowchart for analysis of transitioning steps.



Parameter Interfaces in COBIA

- ◆ **Goal - Strongly typed Value/Elimination of the VARIANT.**
- ◆ **Separation of common and type-specific aspects of the parameter.**
 - ⇒ **Common Parameter members:**
 - Mode
 - Type
 - Validate current value
 - Validation Status
 - ⇒ **Type-specific Parameter properties:**
 - Value
 - Upper/Lower Bounds
 - Default Value
 - Validate potential value
 - Units of Measure only provided for Real and Real Array parameters
- ◆ **Backwards compatible with existing COM-based CAPE-OPEN implementations provided by COMBIA interoperability.**
- ◆ **Parameter specification interfaces provided for information and energy streams.**

Parameter Interfaces, cont'd

- ◆ **Array Parameters:**
 - ⇒ Homogeneous array types.
 - ⇒ Parameter value will be **CapeArray(type)**
 - **CapeArrayInteger**
 - **CapeArrayReal**
 - **CapeArrayBoolean**
 - **CapeArrayString**
 - ⇒ Can obtain and set the value of the entire array at one time.
 - ⇒ Parameter values can be obtained/set by element using the **ICapeArray(Type)Parameter** interface.
- ◆ **ICapeArrayParameter** interface has two properties:
 - ⇒ **NumDimensions**
 - ⇒ **Size**
- ◆ **NOTE: All rows in a dimension have the same length, not a jagged array.**

Parameter Specification Interfaces

- ◆ **Energy and Information Streams are parameter collections.**
 - ⇒ Energy and Information Ports expose parameter specifications.
 - ⇒ PME uses parameter specifications to populate the energy and information ports with the required parameters.
- ◆ **Parameter Specification interfaces.**
 - ⇒ Information and Energy Ports expose a collection of these specification objects.
 - ⇒ ICapeParameterSpecification: one member, get_Type
 - ⇒ ICape(Type)ParameterSpecification: Contains all members of the ICape(Type)Parameter interface except Value, ValStatus.
- ◆ **COMBIA translates specifications.**
- ◆ **STATUS: Implemented in Beta version of COBIA.**

Parameter Economic UOM (Proposed)

- ◆ Queried CO-LaN membership about the interest in adding an economic dimension to parameters.
 - ⇒ Positive comments received on adding the dimension.

- ◆ Currency Issue
 - ⇒ Need the ability to convert currencies.
 - ⇒ Identified ISO 4217 – Currency Code Services
 - Widely used in banking, commerce, and trade.
 - Three letter code for currencies.
 - Web services available to acquire conversion factors using these codes.

- ◆ Two Interfaces proposed:
 - ⇒ ICapeCurrency exposed by the PMC.
 - One member, get_Currency that returns the currency code supported by the PMC.
 - ⇒ ICapeCurrencyExchange exposed by the PME.
 - Members include default currency for the PME, list of currencies known by the PME, and currency conversion methods.

- ◆ Currently not implemented in COBIA.

Reporting Interface (Proposed)

- ◆ **Issues leading to updating reporting interface :**
 - ⇒ **Currently Reporting is handled by ICapeUnitReport**
 - Reporting is only supported by unit operation PMCs.
 - ⇒ **Format of reports available is currently limited.**
 - Work around using Simulation Context.
- ◆ **Objective to develop a general reporting interface**
 - ⇒ **Usable by all types of PMCs.**
 - ⇒ **Expand the media types (formats) of reports that may be generated.**
 - Standard MIME types.
 - Potential for XML/JSON format that can be used in templated reporting.
 - Minimum reporting requirement is plain text format.
 - ⇒ **Provide means to create reports in different languages.**
- ◆ **Overview of interface design provided to membership for comment.**
- ◆ **This interface has not been implemented in COBIA, and is currently not anticipated to be available in COM-based CAPE-OPEN.**

Error Handling in COBIA

- ◆ **CAPE-OPEN uses (directly or indirectly) combination of error handling approaches**
 - ◆ **IErrorInfo defined by COM**
 - ◆ **CAPE-OPEN error interfaces**
 - ◆ **System and CAPE-OPEN errors returned as HRESULT's**
- ◆ **Developer must be prepared to handle all 3 types on each function call for robust products with good error reporting to end user**
 - ◆ **Result is poor error handling implementation in CAPE-OPEN**
- ◆ **New error model in COBIA:**
 - ◆ **Uses single error object**
 - ◆ **Built into base COBIA interface**
 - ◆ **Error text is provided for all errors**
 - ◆ **API routine for error formatting**
 - ◆ **C++ language binding wraps error handing into std::exception**

Development of COBIA Threading Model

- ◆ **Objective – Enable efficient use of COBIA in multi-threaded applications.**
- ◆ **Context:**
 - ◆ Availability of multi-threaded hardware is increasing as costs are decreasing.
 - ◆ Multiple threads may improve application performance.
 - ◆ Multi-threaded applications are becoming more prevalent.
 - ◆ Evolution of programming standards – standardization of thread safety methods is a relatively new feature.
 - ◆ Most CAPE applications are tightly coupled making thread safety difficult to implement.
 - ◆ Difficulty in developing thread safe software will ensure that some components will not be made thread safe in the future.

Scenario for Multi-Threaded PMC

- ◆ **Distillation column unit operation PMC**
 - ◆ Each tray has a material object representing the contents of the tray.
 - ◆ Calculation of PMC can be split amongst multiple threads.
 - ◆ This PMC updates the overall state of these internal Material Objects.
 - ◆ The PMC then invokes a flash on all the Material Objects using different threads.
 - ◆ The Material Objects then call the flash routine in the Equilibrium Servers.

- ◆ **Involves calls from Unit Operation PMC to the PME's Material Object, then to a third-party Property Package PMC.**
 - ◆ Could be three separate objects from three different vendors/developers.
 - ◆ Need to ensure that these calls do not block each other or cause race conditions in either the PME or PMCs.

Threading Considerations for COBIA

- ◆ **Current CAPE-OPEN approach: Microsoft COM Threading Apartments**
 - ◆ Single Threaded – Synchronized calling by COM in a single thread
 - ◆ Multithreaded – Must add synchronization code to component
 - ◆ Neutral – Call always occurs on caller's thread (may be through lightweight proxy)
- ◆ **None fully satisfy the needs of COBIA**
- ◆ **Design goal: Objects accessible by any thread, but not concurrently**
 - ◆ Caller must ensure this requirement
- ◆ **Persistence interfaces may be needed to allow changing thread**
- ◆ **COMBIA compatibility (restrictions) must be considered**
 - ◆ Legacy components were not designed with thread safety
 - ◆ Include modifiable global variables and shared resources
 - ◆ Majority of CAPE-OPEN objects are STA
 - ◆ Requires calls on a single thread
- ◆ **Design still under development as part of Phase III**

2018/9 Deliverables

◆ Flowsheet Monitoring interface

- ⇒ RFC starts on November 1

◆ COBIA

- ⇒ Phase II Beta version release.
- ⇒ Testing and evaluation of Phase II.
- ⇒ Develop interface specifications and RFC.
 - Parameters, Persistence, Reporting, Error Handling
- ⇒ Scoping of Phase III.
 - Threading model development, marshaling, language bindings

◆ Work with Interop SIG

- ⇒ Testing and Evaluation of COBIA
- ⇒ Certification tools
- ⇒ Versioning