

Method and Tools SIG Report

CAPE-OPEN 2015 Annual General Meeting

SIG Leader
Bill Barrett

Amsterdam, 13 October, 2015

SIG Membership

Bill Barrett

US EPA

Jasper van Baten

AmsterCHEM

Tony Garratt

Reaction Design

Daniel Wagner

Jorge Martinis and

Michael Hlavinka

Bryan Research and Engineering

Loic d'Anterroches

Céondo, Ltd.

David Jerome

Krishna Penukonda

SIMSCI/INVENSYS

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 from last year.

M&T SIG Current Projects

- ◆ Review M&T Integrated Guidelines and Common Interfaces
 - ❖ Identify issues exposed through implementation
 - ❖ Provide errata and clarification documents
 - ❖ Develop best practice guidance

- ◆ Currently working on:
 - ⇒ Parameters (Peer reviewed)
 - ⇒ Error
 - ⇒ Identification
 - ⇒ Persistence
 - ⇒ Utilities
 - ⇒ Flowsheet Monitoring

- ◆ CAPE-OPEN Binary Interop Architecture (COBIA) development

Type Library and .NET PIA

- ◆ **Digital Signing of CAPE-OPEN files**
 - ❖ **Developed guidelines:**
 - **Installers – Code Signed**
 - **Binaries – Code Signed**
 - **.NET PIA**
 - **Assembly will be Strong Name Signed with keys generated by sn tool**
 - **Signing with code signing certificate does not propagate back to Certification Authority**
 - **No Signing of pdf files**
 - ❖ **CO-LaN obtaining a digital code signing certificate.**
 - **Code Signing Certificate maintained by CTO**
 - **Private key distributed on an as-need basis**
- ◆ **Type Library and PIA Installers**
 - ❖ **distribution pending code signing certificates.**
 - ❖ **Need to create a current-user installer**
 - ❖ **Current version is available upon request**

PARAMETER Common Interface

- ◆ **STATUS: PEER REVIEW COMPLETED.**
- ◆ **The M&T SIG would like to *thank* reviewers**
- ◆ **Responses to review comments and actions taken are documented.**
- ◆ **Errata:**
 - ❖ Real parameter upper bound is read only
 - ❖ Integer parameter default value is read only
- ◆ **Clarifications:**
 - ❖ The roles of PME, Parameter Owners, Parameter Clients
 - ❖ Parameter specification and validation
 - ❖ Dimensionality
 - ❖ Array Parameters

Array Parameters

- **CAPE-OPEN allows Generic Array Parameter Structure**
 - **Not widely supported by PMEs:**
 - Multi-Dimensional Arrays
 - Nested Arrays
 - Elements of Mixed Data Types

- **Minimum required PME support:**
 - **One Dimensional**
 - **Homogeneous:**
 - All elements of the same type
 - All elements have same or equivalent specification
 - **Support means: inspect and modify value.**

Identification Common Interface

- ◆ Errata and Clarifications published on CO-LaN web site
- ◆ Clarifications:
 - ❖ Names within a collection must be unique: responsibility of collection owner
 - ❖ All CAPE-OPEN objects must implement *ICapeIdentification*
 - ❖ Restrictions on name:
 - Must be unique within collection (responsibility: collection owner)
 - Minimum: 1 character. No Maximum
 - First and last character must not be white space
 - No CR/LF/Tab or control characters
 - ❖ Character set follow rules of the implementing middleware
 - In COM, this is a UTF-16 encoded BSTR.
 - No Control Characters
 - ❖ *Don't call ICapeIdentification prior to ICapeUtilities.Initialize*

Persistence Common Interface

- ◆ Persistence Errata and Clarifications drafted
- ◆ Clarifications:
 - ❖ Minimum Level of Support for IPersistPropertyBag for VARIANT data types used in CAPE-OPEN
 - ❖ Persistence fall back:
 - PME can query PMC for support of IPersistPropertyBag, IPersistFile or IPersistStorage
 - Currently, if any of these fail, not clear what PME should do
 - Clarification: PME should fall back to IPersistStream or IPersistStreamInit
- ◆ Needs final SIG review and Peer Review.

Utilities Common Interface

- ◆ Requires final SIG review and Peer Review

- ◆ Clarifications:
 - ❖ Primary PMC Objects must implement *ICapeUtilities*
 - ❖ Edit Method Return Value:
 - 0 = S_OK: PMC was modified
 - 1 = S_FALSE: PMC remains unmodified
 - ❖ Simulation Context can be set prior to *ICapeUtilities.Initialize*
 - ❖ Object Life Cycle

Error Common Interface

◆ Issues identified:

⇒ Complexity

- CAPE-OPEN error handling not based on COM *ErrorInfo* API.
- CAPE-OPEN objects are required to expose all possible CAPE-OPEN error interfaces
- Most CAPE-OPEN objects only expose *ECapeRoot* and *ECapeUser* and return *ECapeUnknownHR*
- Error conditions are not transparent

⇒ Logging tools are required to identify the cause of problems.

◆ Error Handling Issues to be addressed under COBIA

- ⇒ Consistent error handling within and across platforms
- ⇒ Improving debugging capabilities

Flowsheet Monitoring Interface

- ◆ Currently under preparation by the M&T SIG
- ◆ Developed event notification strategy
 - ❖ Why event notification should be used
 - ❖ When can notifications be sent
 - ❖ How are notification subscribed to
- ◆ Additional requirements.
- ◆ Introduced *ICapeStream* interface:
 - ❖ Stream type
 - ❖ Upstream and downstream connected unit + port
 - ❖ Strongly typed stream collection object

CAPE-OPEN Binary Interop Architecture (COBIA)

- ◆ Formerly: CAPE-OPEN Object Model
- ◆ Development Timeline

⇒ Phase I - (3 weeks from Management Board approval)

- COBIA Registry - register components and create instances
- Data type standardization
- Limited set of interface stub codes
- Prototype property package and material object (CO-1.1)

⇒ Phase II - (3 weeks from acceptance of Phase I)

- IDL/Stub Code Parser
- Expansion of interfaces
- COBIA/COM Interop

CAPE-OPEN Binary Interop Architecture (COBIA)

◆ Phase III – Cross Platform Support

⇒ Port to other platforms

- Microsoft Visual Studio on Windows
- Intel C++ on Windows
- GCC (GNU Compiler Collection) C++ on both Windows and Linux
- Additional platforms can be added as need arises

⇒ Expand language binding and interop (.NET)

⇒ Marshaling between platforms

◆ Maintenance and support (CO-LaN)

⇒ Documentation

⇒ Distribution of source code and binaries

⇒ Ongoing support

Status of Interfaces

Review of interfaces for implementation

Business Specification Document	Responsible SIG	Proposed Action
Utilities Common	Method and Tools	Keep as is
Collection Common	Methods and Tools	Keep as is
Error Common	Methods and Tools	Replace
Flowsheet Monitoring	Methods and Tools	Finish Development
Identification Common	Methods and Tools	Keep as is
Parameter Common	Methods and Tools	Overhaul
Persistence Common	Methods and Tools	Replace
Simulation Context COSE	Methods and Tools	Keep as is
Optimisation	None	Deprecate
Parameter Estimation	None	Deprecate
Partial Differential Algebraic Equations	None	Deprecate
Planning and Scheduling	None	Deprecate
Sequential Modular Sequencing Tools	None	Deprecate
Solvers	None	Deprecate
Chemical Reactions	Thermodynamics	Finish Development and Replace
Petroleum Fractions	Unit Operations	Finish Development and Replace
Physical Properties Data Bases	Thermodynamics	Deprecate
Thermodynamics 1.0	Thermodynamics	Keep as is (*)
Thermodynamics 1.1	Thermodynamics	Keep as is
Unit Operation	Unit Operation	Keep as is (**)

- Interfaces without a SIG find limited or no use.
- Some interfaces need refactoring based upon implementation experience.

Interface Recommendations

- ◆ **Need a Numerics SIG to support**
 - ⊃ Solvers
 - ⊃ Optimization
- ◆ **Thermo SIG update and replace**
 - ⊃ Physical Property Database (PPDB)
 - ⊃ Chemical Reactions
- ◆ **Unit + Thermo SIGs update and replace**
 - ⊃ Petroleum Fractions
- ◆ **M&T SIG update**
 - ⊃ Parameters
 - ⊃ Persistence
 - ⊃ Error
- ◆ **Evaluate need for, and assign to SIG**
 - ⊃ Parameter Estimate and Data Reconciliation (PEDR)
 - ⊃ Planning and Scheduling
 - ⊃ Partial DAE (Solver?)
- ◆ **Deprecate unused interfaces**
 - ⊃ Sequential Modular Sequencing Tools (SMST)
 - Limited number of methods
 - Unique solution for partitioning.
 - Most PME's handle internally.

COBIA Roadmap

◆ 2015

- ❖ **Initiate Phase I development**

◆ 2016

- ❖ **Complete Phase I development**
- ❖ **Complete Phase II development**
- ❖ **Revise M&T SIG Common Interface Specifications to the COBIA**
 - This will incorporate issues raised in the Errata and Clarifications documents published.
- ❖ **Work with other SIGs to transition to COBIA**
 - Likely minor modifications to interface specifications documents
 - Will require COBIA IDL for the interfaces.

◆ 2017

- ❖ **Roll-out COBIA.**
- ❖ **CO-LaN will maintain the code and provide updates as needed.**

2015 Deliverables

- ◆ **Errata and Clarifications Documents**
 - ⇒ Identification published
 - ⇒ Parameters – peer review completed
 - ⇒ Persistence – initial draft created
 - ⇒ Utilities – minor edits remain, peer review this year

- ◆ **Flowsheet Monitoring interface:**
 - ⇒ Currently being revised

Ongoing Activities

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