Method and Tools SIG Report 2014

SIG Leader

Bill Barrett

SIG Membership

Bill Barrett
US EPA

Jasper van Baten
AmsterCHEM

Jorge Martinis and
Michael Hlavinka
Bryan Research & Engineering

Loïc d'Anterroches
Céondo, Ltd.

Javier Fuentes
Process Systems Enterprise Limited

David Jerome
Krishna Penukonda
Schneider Electric/SimSci

Tony Garratt
Reaction Design

Daniel Wagner

M&T SIG Charter

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

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

M&T SIG Current Projects

- Review M&T Integrated Guidelines and Common Interface Specifications
  - Identify issues exposed through implementation
  - Provide errata and clarification documents
  - Develop best practice guidance.

- The M&T SIG is currently working on the following interface specifications:
  - Parameter
  - Identification
  - Error handling
  - Persistence
  - Utilities
  - Flowsheet Monitoring

- **CAPE-OPEN Object Model development**

Parameter Common Interface

- Errata and clarification document under development
- Roles of Parameter Owners, Parameter Clients, and the PME
  - The Parameter Owner is the object that owns the Parameter Collection that contains the Parameter.
  - The Parameter Client is any software component accessing the Parameter.
  - Clarification made on when a Parameter Collection can change.
  - Clarification made on when a Parameter value may change
  - Limits the need for PMEs to check parameter states.

Parameter Common Interface, cont’d

- Parameter Specifications
  - Lower and upper bounds, default values, and the options list provide basic criteria for determining whether a Parameter value is valid.
  - Parameter Default Value need not be a valid value for the parameter.
    - This is needed for a case where there is no obvious default value and the user needs to set a value.
  - Lower and upper bounds, default value as well as the value itself, may be UNDEFINED.
  - UNDEFINED may or may not be a valid value, depending on the Parameter.

Parameter Common Interface, cont'd

- Parameter Validation
  - <u>**Parameters can have CAPE INVALID values!**</u>
  - Validation checks whether the Parameter's current value complies with the Parameter's specification and other applicable criteria. Examples:
    - Indicating that initial values are within a range where convergence is considered likely
    - Highlighting calculated results outside an acceptable range, e.g. pressures and temperatures outside a safety threshold.
  - After a successful call to Validate, the status must not be CAPE_NOT_VALIDATED.

Parameter Common Interface, cont'd

- **Dimensionality**
  - ➲ Formalizes the definition of the dimensionality object as a real-valued array.
- **Array Parameters**
  - ➲ Provides a structure for the Array Object
    - Value is a CapeArrayVariant, each element containing either a real, an integer, a string, a Boolean, or a nested array
    - Specification is a CapeArray of CapeObjects, each object supporting the ICapeParameterSpec and appropriate ICape<TYPE>ParameterSpec interface for the corresponding value element.
  - ➲ Provides minimum support requirements

Identification Common Interface

- Errata and clarification document peer review completed and submitted to Management Board for final approval.

- *ICapeIdentification.ComponentName* (section 3.5.1)
  - ➲ Minimum and maximum length
    - Minimum length is one alphanumeric character
    - No maximum length limit
  - ➲ White space in names is allowed.
    - First and last character of the name must not be whitespace.
  - ➲ Character sets – Issue for M&T guidelines clarification.
    - Character set dictated by middleware (COM: UTF16)
    - No control characters.

Identification Common Interface

- *ICapeIdentification.ComponentDescription*
  - Minimum and maximum length
    - No minimum or maximum length
  - Character sets – Issue for M&T guidelines clarification.
    - Character set dictated by middleware
    - Should control characters be allowed?
      - Line feed
      - Carriage return
      - Tab
      - Form feed, delete, escape, bell,
      - ….

Collection Common Interface

- Approved and posted on CO-LaN website

- Variant Value for *ICapeCollection.Item* method clarified

- Naming of Collection Members
  - Uniqueness is enforced by Collection Owner

Simulation Context COSE Interface

- Errata and Clarifications document approved and posted on CO-LaN website.

- New Named Values proposed
  - AbortCalculateRequested
  - DefaultThermoVersion
  - SimplifiedModelRequest

Utilities Common Interface

- Errata and Clarifications document near completion

- Requirement for PMCs to implement *ICapeUtilities*
  - PMC Primary Objects defined and identified.
  - Types of PMC Primary Objects that require *ICapeUtilities* tabulated.

- Edit Method Return Value
  - Created a *CapeEditResult* enumeration with two values:
    - *CapeModified* = 0 = S_OK
    - *CapeNotModified* = 1 = S_FALSE
  - Edit returns the appropriate *CapeEditResult* value.
  - Use HRESULT for COM implementations.

Utilities Common Interface, cont'd.

- Object Life Cycle clarified
  - Create or instantiate object
    - CoCreateInstance or from manager object
  - <u>Set Simulation Context</u>
  - Select persistence mechanism (see next slide)
  - InitNew (if appropriate)
  - Load (if appropriate)
  - ICapeUtilities.Initialize
  - … use the object …
  - ICapeUtilities.Terminate
    - <u>PMC releases all external references</u>
  - Release all COM references

Persistence Common Interface

- COM persistence is discussed as part of the Utilities Common Interface Errata and Clarification document.

- Clarify use of COM persistence
  - IPersistStream or IPersistStreamInit required
  - Additional COM interfaces can be implemented for various persistence options:
    - IPersistMemory - Persist to an allocated memory location (i.e., fixed-size byte array).
    - IPersistPropertyBag - Persist to a property bag container, such as an XML text file.
    - IPersistStorage - Persist to structured storage.
    - IPersistMoniker - Persist to a moniker.
  - Consistent use of persistence: InitNew

Flowsheet Monitoring Interface

- Currently being reviewed and edited by the M&T SIG.

- Interested parties:
  - Please request a copy of the current version.
  - Join the conference calls organized (2nd Wednesdays)

M&T Guidelines Issues

- .NET Primary Interop Assembly (PIA) provides a universal set of .NET-based CAPE-OPEN interfaces.
- CO-LaN recommends using Microsoft .NET Framework 4.5.2 when developing CAPE-OPEN PMEs.
  - .NET Framework is now a Windows Component and not an independent product.
  - .NET Framework is not a part of Visual Studio.
  - .NET 1.0 and 2.0 no longer supported by Microsoft.
  - .NET 3.5 SP1 is a Windows 7 and Windows 8.1 component and will be supported as part of those OSs (Windows 8 EOL January 2023).
  - .NET 3.5 SP1 is backward compatible with .NET 2.0.
- CAPE-OPEN development should be possible using free tools, such as Visual Studio Express.
- Development tools lifecycle remains an issue.

CO LaN

CAPE-OPEN Object Model

- Status
  - ➲ Discussion of platforms to support: priority to Windows
  - ➲ Identified user types for CAPE-OPEN
    - End-users of process simulation tools
    - Users that develop PMCs using tools such as gPROMS, MATLAB, Scilab, Excel or script (Python)
    - Software developers
      - Not necessarily with COM experience
  - ➲ Need to re-design a common interface definition language (for strong typing)
  - ➲ Reasons to use middleware:
    - Object registration
    - Object lifecycle
    - Memory management
    - Marshalling between processes, computers or platforms
  - ➲ Technical discussions on interface modifications (new error handling model, eliminate VARIANTs, data types, character sets, strong typing, …)
  - ➲ CO-LaN will distribute both source code and binaries for the Object Model.

CAPE-OPEN Object Model

- Deliverables
  - Revised Method and Tools Integrated Guidelines
  - IDL Syntax and Compiler
  - Registration Tool with specific registry component
  - Middleware including object creation, marshalling and data type management

- Requirements
  - Need to incorporate COM interoperation to ensure backwards compatibility.
  - Need bindings to different languages, plus stub generators.
  - Need to support 32- and 64-bits.

- Cross-Platform Issues
  - ISO-standard C++
  - Targeted operating systems: MS Windows, Linux and MacOS.
  - Compilers: MS Visual C++, GNU (gcc) C++ compiler.

CAPE-OPEN Object Model Roadmap

- 2014
  - Scoping of the Object Model
  - Revise M&T Integration Guidelines for Object Model
- 2015
  - Develop and test IDL compiler and registration tool for Windows and COM support
  - Prototype CAPE-OPEN Object Model middleware
  - Complete the M&T Integration Guidelines
- 2016
  - Revise M&T SIG Common Interface Specifications to the Object Model
    - This will incorporate issues raised in the Errata and Clarifications documents published.
  - Work with other SIGs to transition to the Object Model.
    - Likely minor modifications to Interface Specifications Documents
    - Will require Object Model IDL for the interfaces.
- 2017
  - Finished Object Model in use.
  - CO-LaN will maintain the code and provide updates as needed.

2014 Deliverables

- Errata and Clarifications Documents
  - COSE – Completed and on the web
  - Collection – Completed and on the web
  - Identification – Completed pending Management Board Approval
  - Parameters – Minor edits remain. Should be to peer review this year.
  - Utilities – Minor edits remain. Should be to peer reviewed this year.
- Flowsheet Monitoring Interface Specification
  - Currently being revised.

Ongoing Activities

- Common Interface monthly conference calls
  - First Wednesday at 11 AM Eastern US Time.
- Object Model monthly conference calls
  - Last Wednesday at 10 AM Eastern US Time.
- Please contact either SIG Leader or CTO if you are interested in participating:
  - Bill Barrett – barrett.williamm at epa.gov
  - Michel Pons – technologyofficer at colan.org