Methods and Tools Special Interest Group Report October 2023 – September 2024

Bill BARRETT

US Environmental Protection Agency

8 October 2024

SIG Information

Membership

Bill Barrett	Michael Hlavinka	Jasper van Baten
US EPA	Bryan Research & Engineering, LLC	AmsterCHEM

Meetings:

- ♦ Calls are held on Tuesdays at 1600 CET, 1000 US Eastern Time, 0900 US Central Time.
- ♦ Monthly Joint Conference call with Interop SIG. Typically, first Tuesdays at 1600 CET, 1000 US Eastern Time, 0900 US Central Time.

If you are interested in joining, please contact either SIG Leader or CTO

- **Bill Barrett – barrett.williamm@epa.gov**
- **Kyle Abrahams - kyle.abrahams@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.
  - Develop and Maintain Test Specifications for Common Interfaces.
  - Incorporate advances in information technology into the CAPE-OPEN protocols.
  - Identify novel uses of CAPE and provide standards for utilizing CAPE within these applications.

M&T SIG 2023/2024 Summary of Activities

- ❑ COBIA Maintenance, Development, and Testing
  - One maintenance release of COBIA Phase II
  - Released COBIA Phase III Work Package I
    - Released as beta in September 2024

- ❑ CAPE-OPEN Threading Guidance published on CO-LaN website.
- ❑ Updated Archival Information Template for documentation
- ❑ Interacted with Interop SIG
  - Support for Certification Testing
  - Versioning
- ❑ Developed Test Specifications for Common Interface Specifications
  - Utilities, Persistence

COBIA Project Roadmap

- ❑ Phase I – Proof of Concept Completed October 2016
  - Core technical components
  - Demonstrate COM/COBIA interoperability with Thermo 1.1 interface set
- ❑ Phase II Full Windows Native – Completed (October 2017)
  - ❑ Upcoming Phase III Versions are Backward Compatible
  - Expanding COBIA to all interfaces of business value
  - Support for C/C++ development.
  - Allow development of fully functional COBIA-based PMEs and PMCs
  - Maintenance – All new bug fix releases will be in Phase III.
- ❑ Phase III – Cross-Platform Interoperability Main Branch (October 2024)
  - WP I: Marshaling Beta Release
  - WP II: Language Bindings Justification Required
  - WP III: Remote Computing Justification Required
  - WP IV: Logging Proposed to Management Board
  - Maintenance - Ongoing
- ❑ Phase IV – Documentation Future

PHASE III - Work Package I (Marshaling) Completed

- ❑ COBIA Marshaler for CAPE-OPEN Interfaces
  - Marshals between threads/processes/machines.
  - Uses TCP/IP transport mechanism
    - Most general case, used in-process, out-of-process, remotely.
    - Further remote machine developments (*e.g.,* encryption) under WP III.
  - Available for testing by interested CO-LaN members.
- ❑ Optimized transport for specific marshaling scenarios Completed
  - Shared Memory Transport Mechanism
    - Improves performance on local machine compared to TCP/IP.
    - Designed for out-of-process marshaling.
    - Uses synchronized access to shared memory block.
  - C++ Heap Memory Transport
    - Optimized synchronization of in-process applications between threads.
- ❑ On-the-fly proxy generation released in Phase III (using libffi package)
  - For types for which precompiled proxies are not available. Completed
  - Enable custom (non-CAPE-OPEN) interfaces Completed
  - Enable COM interop on custom interfaces Completed

Release of COBIA Phase III

❑ Merged the Phase III Code Repository into the Main Branch.

- Phase II support will end with release of Phase III
- All updates/bug fixes will be provided for Phase III
- Phase III backwards compatible with Phase II

❑ Beta version released to CAPE-OPEN community for testing (September).

- In- and out-of-process marshaling
- Generic TCP/IP Marshaler
  - Can be expanded for use as the full-scale remoting solution
  - Secure transport mechanism is planned
- Pre-compiled proxies for COBIA interfaces
- On-the-fly proxy generation for custom interfaces

❑ Final release of full version planned by end of October 2024.

Moving Beyond COBIA Phases

- ❑ COBIA development largely complete.
- ❑ Cross-Platform Interoperability – Demonstrated.
  - Linux
    - COBIA has been compiled and run on Ubuntu.
    - Currently can be compiled from source (available to CO-LaN members).
    - Examples require GTK3 for GUI.
    - Planned development of a Debian & RPM package. This covers the majority of Linux distros and package managers.
  - macOS
    - Compiles and runs
    - Need strategy for distribution (macOS Framework)
- ❑ Open Tasks:
  - WP II: Language Bindings - Awaiting Business Case
  - WP III: Remote Computing - Awaiting Business Case
  - WP IV: Logging – Proposed to Management Board
- ❑ Maintenance - Ongoing
- ❑ Phase IV – Documentation Future

Path Forward Creating a Community Around COBIA

- ❑ **GOAL: Expand development and maintenance into the developer community.**
  - Provides long-term vehicle for support to users of COBIA
    - Maintenance/Improvements
    - Lower barriers to entry
  - Engaging a community…
    - Academics – Training/R&D
    - Industry -
    - Government
  - Identify and Debug Issues
  - Devise and Implement Useful Features
    - Marshaller transport mechanisms.
    - Language Bindings.
    - Additional Tools such as registration and code generation.

- ❑ **Pathways to Community Creation**
  - Dissemination
  - Training
  - Case Studies
  - Examples/How-to Documents

COBIA Threading Models

- ❑ **STATUS: Document available on CO-LaN website.**
  - https://www.colan.org/wp-content/uploads/2024/05/Guidelines-For-Threading-v1.0.pdf
- ❑ **Threading Models**
  - Default Threading Model – PMC can be used from multiple threads but not concurrently
  - Restricted Threading Model – PMC is restricted to thread in which it was created.
- ❑ **Considerations:**
  - Modern/flexible approach to multi-threaded applications.
  - Establish responsibility for synchronization.
  - Minimize need to marshal calls.
  - Interoperable with COM
- ❑ **Jasper to present details.**

Proposed COBIA Logging Overview

- ❑ **Logging capabilities will be built directly into COBIA.**
  - Different from the approach used for the CAPE-OPEN Logging and Testing Tool (COLTT).
  - Logging capabilities can be adapted to the needs of the CAPE-OPEN community.
- ❑ **Logging initiated by API call or environment variables.**
- ❑ **COBIA logging access to objects similar to marshaling.**
  - Utilizes a tear-off interface representation of the object being logged.
  - Interface is represented by proxies on both client side and server side.
  - Proxies can be precompiled or generated on-the-fly (just like marshalers).
  - All methods on COBIA interfaces can be logged.
- ❑ **Log output directed to a known file by default. File location can be changed via an environment variable.**
- ❑ **Proposed `ICobiaLogSink` interface allows other COBIA applications to subscribe to the logger, enabling, for instance, the PME to display log output in real time.**
- ❑ **Easy to implement – logger works just like a marshaler. Most of the code already exists.**

Test Specification Developments

- ❑ **M&T SIG developing Test Cases for the Common Interfaces.**
  - M&T SIG owns the Common Interfaces and is responsible for developing test cases for them.
  - These test cases serve as examples for other SIGs to develop Test Cases for their interfaces.

- ❑ **Test Cases**
  - Test cases include ensuring that the object supports required interfaces.
  - Test case format is based upon interface use case format.
  - Test cases are based on interface specification requirements, use cases, and method descriptions.
  - Test checks for correct return value types, where applicable.

Manager Interface

- ❑ Released December 2023
- ❑ Expands on the Factory Method Design Pattern used in Property Package Manager
  - Creates Primary PMC objects using a standard workflow.
  - Allows creation from persistence.
- ❑ Reduces the number of derived subclasses that need to be registered.
  - Template Based - Variations of a PMC can be made from a template for instantiation by the Manager.
  - Variations of a PMC can be instantiated by the factory without being registered.
  - Variation visibility independent of PME or Manager instance.
  - Example: Multiple variations of a property package, each having different compound and phase list.

Anticipated 2024/2025 Activities

❑ **Finalize COBIA Phase III**

- Beta version is available for testing
- Release COBIA with Marshaling and Multi-Thread Support and formalized threading model
- Implement Logging

❑ **Interface Specifications**

- CAPE-OPEN Native Persistence Interface Specifications (CAPE-OPEN Version 1.2+)
- Rewrite Parameters Interface Specification document (CAPE-OPEN Version 1.2+)
- Incorporate Errata and Clarifications into Identification Interface Specification (CAPE-OPEN Version 1.1+)
- Finalize Reporting (CAPE-OPEN Version 1.1+)

❑ **Develop Test Case Specifications for Common Interfaces**

❑ **Common Source Documentation with UNIT and THERMO (Kyle will present)**

❑ **Work with Other SIGs**

- Advise on Interop SIG on Certification Tools

Thank you For Your Attention

- Any Questions?

Backup Materials on Interface Update

Not presented unless questions raised.

Flowsheet Monitoring Interface

- Overview of Interface
  - Provides the FMC the ability to access all elements in a flowsheet without interfering with the flowsheet
  - Event notifications provided to the FMC for changes to the state of the flowsheet.
- TLB/PIA installers being distributed on CO-LAN website.
- Update COCO/USEPA WAR Algorithm plug.

Revising Persistence Common Interfaces

- ♦ **Motivation**
  - Use of CAPE-OPEN Data Types.
  - Allow use of human-readable formats such as JSON or XML
  - Required for COBIA.
    - COMBIA should handle COM/COBIA persistence interoperability.
    - Allow use of platform-native serialization mechanisms through COBIA language bindings.
  - Single set of interfaces defined to reduce complexity vs current use of COM Persistence Interfaces
- ♦ **Design decision**
  - Explicit separation of object serialization from storage to persistent media.
  - Generic Persistence Host and Persistence Client allows broader application.
  - Tree-based dictionary used to read/write object data as any CAPE-OPEN data type or a byte array.
- ♦ **Status**
  - Implemented in COBIA.
  - Textual specification under development.
  - Considering implications of adding these persistence interfaces to the COM Type Library. Looking for input from others.

Persistence Common Interfaces, Cont'd.

- ***ICapePersist***
  - *Save* – Asks the PMC to save itself using the *ICapePersistWriter* interface.
  - *Restore* – Restores the PMC using the *ICapePersistReader* interface.
  - Dirty status – indicates whether Persistable Object has been modified.

- ***ICapePersistWriter***
  - Exposed by Persistence Host to the Persistable Object.
  - Provides methods to write standard CAPE-OPEN data types and byte array.
  - Method to add nodes to tree.

- ***ICapePersistReader***
  - Exposed by Persistence Host to the Persistable Object.
  - Provides methods to read standard CAPE-OPEN data types and byte array.
  - Methods to determine the names of the nodes present, and their data type.

Revised Parameter Common Interfaces

Motivations

- Strongly typed Value/Elimination of the VARIANT.
- Eliminate the separation of static and dynamic elements of the Parameter.
- New Design:
  - Common Parameter members:
    - Mode
    - Type
    - Validate current value
    - Validation Status
  - Type-specific Parameter members:
    - 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.
  - Array compatibility limited to homogeneous arrays

Parameter Interfaces, cont’d

- Array Parameters:
  - Homogeneous array types.
  - Passed as a one-dimensional array
  - 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 (New Functionality).
- ICapeArrayParameter interface has two properties:
  - NumDimensions
  - Size
- NOTE: All rows in a dimension have the same length, not a jagged array.

Currency Interface (Proposed)

- Queried CO-LaN membership about the interest in adding an economic dimension to parameters.
  - Positive comments received on adding the dimension.
  - Applicable to more than Parameters.
- 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 property, 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: *ICapeReport*
  - 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.

Development of COBIA Threading Model

- Objective – Enable efficient use of COBIA in multi-threaded applications.

- Proposal – Components declare themselves safe for multi-threading
  - Restricted
    - ➲ Only called from a single thread
    - ➲ Only called from thread that created the object.
    - ➲ Can use persistence to move from one thread to another
    - ➲ Can be created in dedicated thread and calls marshaled – Similar to COM calling STA from MTA
  - Unrestricted
    - ➲ Can be called from any thread
    - ➲ Only one thread can call the object at a time (No re-entrancy)

- Documentation still under development.