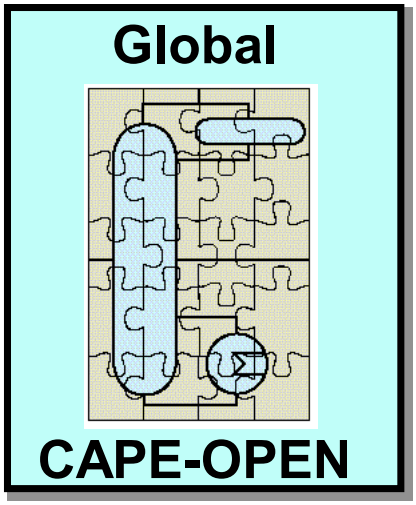
Global CAPE-OPEN

Delivering the power of component software
and open standard interfaces
in computer-aided process engineering



CAPE-OPEN Common Interfaces: Overview

Jean-Pierre Belaud, INPT
Daniel Piñol, HyproTech

GCO-M&T-20-CAPE-OPEN Common Interfaces: Overview Document

7 June 2001

Archival Information

Reference	GCO-M&T-20-CAPE-OPEN Common Interfaces: Overview Document
Filename (if different)	CO Interfaces Overview.doc
Authors	Jean-Pierre Belaud, INPT Daniel Piñol, HyproTech
Date	7 June 2001
Number of Pages	10
Version	Version 3
Reviewed by (date)	**Reviewed by (name) (greyed out means not yet)**
Distribution	GCO Project
Additional Material	
Location on BSCW	GCO/ WORK PACKAGES /MANAGEMENT /M&T/02 M&T Technical Documents

Important Notices

Disclaimer of Warranty

Global CAPE-OPEN documents and publications include software in the form of *sample code.* Any such software described or provided by Global CAPE-OPEN --- in whatever form --- is provided "as-is" without warranty of any kind. Global CAPE-OPEN and its partners and suppliers disclaim any warranties including without limitation an implied warrant or fitness for a particular purpose. The entire risk arising out of the use or performance of any sample code --- or any other software described by the Global CAPE-OPEN project --- remains with you.

Global CAPE-OPEN is a collaborative research project established under BE 3512 “Industrial and Materials Technologies” (Brite-EuRam III), under contract BPR-CT98-9005

Summary

This document gives an overview of the CAPE-OPEN (CO) Common Interfaces.

The Common Interfaces are interfaces and implementation models for handling concepts that may be required by any CO interface specification.

One of the objective of Methods & Tools group is to provide reusable interfaces for the CAPE-OPEN interface designers to be able to concentrate on engineering concepts and not on plumbing details. There is a set of simple unrelated functionalities that would be useful for any kind of PMC (Process Modelling Component), since it would allow maximum integration between them and the PME (Process Modelling Environment) to which they provide services.

Contents

1. Preface

This document sets out a proposal for a cross work package concept: the CO Common Interfaces. These interfaces will each have a separate interface specification document.

The CAPE-OPEN technology defines two categories of interfaces : Business Interfaces and Common Interfaces.

Business interfaces own vertical interface specifications. These interfaces are domain-specific interfaces for CAPE application domain. They define the CO components which can be involved in an execution of a CO process simulation application. Important classes of PMCs are identified such as physical properties, unit operation modules, numerical solvers, flowsheet analysis tools …

Common interfaces embody horizontal interface specifications for handling concepts that may be required by any business interfaces. This is a collection of interfaces that support basic functions and are always independent of business interfaces.

1.1 Recommendations to the intended audience

The Common Interfaces specifications are aimed at designers of CO interfaces and developers of CO components.

1.1.1 Designers of CO interfaces

They design the business interfaces belonging to the CO standard and write the open interface specification document.

Methods & Tools group recommends Common Interface(s) to be part of future CAPE-OPEN interface specifications, if these specifications need functionalities which can be supplied by already existing Common Interface(s). If the specification requires further functionalities than the ones provided by Common Interface(s), the M&T group will consider enhancing the Common Interface(s).

The designers of a CO interface specification remain free to use or not one or several Common Interfaces. Somehow the specification document must show clearly the dependencies between this specification and any Common Interfaces; basically for instance through the UML model by drawing a diagram of packages dependencies.

1.1.2 Developers of CO components

They develop applications/components which are compliant with the CO standard.

The Common Interfaces are general purpose interfaces that can be mandatory for developing CO-based components. The CO component developer has to implement Common Interface(s) if the corresponding specification depends on Common Interface(s). For instance the CO Unit developer should provide the Identification Interface since the Unit Interface Specifications requires the use of this interface.

1.2 Needs for CO Common Interfaces

The Common Interfaces are a collection of interfaces that support basic functions and are always independent of Business Interfaces. Within CAPE-OPEN there have been currently few opportunities and initiatives to reuse design concepts across the work packages. In some instances this design reuse could be facilitated to provide one means of achieving consistency across the main deliverables of the project (i.e. the

interface specifications). In addition to design reuse, it may be possible to go further and produce implementations of these designs, which are also reusable across the CO interface specifications.

1.3 General design principles

- The Common Interfaces are built in the same manner as any Business Interface specification. Therefore the content and the syntax used to specify the Common Interfaces are similar to the ones used to specify any Business Interface specification. Each Common Interface is specified by a separate document. So as specified by the M&T group at present, the Template for Interface Specification Document is the reference document.

- The textual requirements and the UML model should make no reference to any other Business Interfaces, since the Common Interfaces are of general purpose. However, use cases and diagrams can be added as concrete examples of how Common Interfaces are used by some Business Interfaces and corresponding CO components; for instance how the Unit interfaces deal with the Identification Common Interface.

- The design of Common Interfaces leads to interfaces which a priori are integrated directly within the design of any Business Interfaces. That means that there is no client/server relation between Business Interfaces and Common Interfaces (from a CO point of view since the implementation can always distribute subsets of a PMC). The resulting CO component will implement not only CO Business interface specifications, but also if required Common Interface specifications. For instance, the interface from Identification Common Interface is part of the Unit interfaces (see the Unit interface specification document which requires the Identification functionality). The Unit PMC will provide the implementation of Unit and Identification interfaces. The interface of Identification Common Interface is designed in order to provide interfaces to the client of the Unit component and not to the Unit component itself. The links between Identification Common Interface and CO Unit interface are proprietary and implementation dependant.

- The designers of Business Interfaces have the responsibility to integrate the Common Interface(s) in their design and to specify within the interface specification document how their specification is related to Common Interfaces; for instance using inheritance or association. The SMST interface specification document shows that the ICapeNumericGATComponent interface inherits from the ICapeIdentification interface.

1.4 Versioning aspect

The Implementation Specifications under the IDL form are represented by a single library; one for the COM platform and one for the CORBA platform. One version number corresponds to the whole library.

The Implementation Specifications enclose the Common Interfaces. The Common Interfaces belong to the standard versioning system and so don't have any specific version number. Only a version number for internal use is used. The designers of the CO interfaces could refer to it but that has no interest for developers of CO components.

The CO components only need to be compliant with a specific version of the Implementation Specifications, for instance version 0.9.3 using the CAPE-OPENv0.9.3.idl which involves some Common Interfaces.

1.5 Associated documents

The CO Common Interfaces involve the following documents:

- CAPE-OPEN Common Interfaces: Overview (this manual);

- Open Interface Specification: Identification Common Interface;
- Open Interface Specification: Parameter Common Interface;
- Error Handling Strategy: Error Common Interface;

2. Summary of key features

2.1 Error Common Interface

- This report gives guidelines for managing errors within any Business and Common Interfaces.
- By definition an error is an abnormal termination. It represents a binary status; either there is no error or an error occurs. When a request is made, if this request is successful it raises no error otherwise it raises an error. When an error occurs, the execution is immediately aborted.
- The error strategy is first defined from a conceptual view. Thus the strategy is independent from any architecture, system and implementation language. The result uses the UML notation. Then the error strategy is applied to the COM and CORBA platform.
- This document describes a classification and a hierarchy of potential errors occurring in the CO standard. These errors are common to all CO interfaces, which can easily reuse them.

2.2 Identification Common Interface

- This specification will be used by a CO component that wishes to expose its name and description. This information refers to an instance of the component, not to the software class.
- In a particular situation, a system may contain several CO components of the same class. The user should be able to assign different names and descriptions to each instance in order to refer to them unambiguously and in a user-friendly way.
- The Unit Operations interface specification has for instance the following requirements: If a flowsheet contains two instances of a Unit Operation of a particular class, the CO Simulator Executive needs to provide the user a textual identifier to distinguish each of the instances. For instance, when the CO Simulator Executive requires to report about an error that occurred in one of the Unit Operations.
- The interface contains a straightforward interface called ICapeIdentification.

2.3 Parameter Common Interface

- This specification will be used by a CO component that wishes to expose some of its own internal data to its clients, so that the latter may utilise it through standard interfaces.
- The interface is made up of two different parts, each corresponding to a different client need:

    1. The first part is a fixed, static aspect that describes the Parameter, such as a type, name, description, dimensionality and so on. This is proposed to be used to assist the human users in deciding what value to give to the Parameter.
    2. The second part deals with the value of the Parameter itself. It is expected that the parameter values will change quite frequently both within and outside of the Component that needs it.

    Additionally, several parameters of a system may share the same parameter description.