Open Software Architecture For Process Simulation: The Current Status of CAPE-OPEN Standard

Jean-Pierre Belaud
Laboratoire de Génie Chimique (LGC, UMR CNRS 5503), Equipe Analyse Fonctionnelle des Procédés, INPT-ENSIACET
118 route de Narbonne, 31077 Toulouse Cedex 4, France
JeanPierre.Belaud@ensiacet.fr

Michel Pons
ATOFINA Centre Technique de Lyon
BP 32 F-69492 Pierre-Benite Cedex, France
Michel.PONS@atofina.com

Abstract

Traditionally simulation environments have been closed monolithic systems; and the resulting bottlenecks in interoperability, reuse and innovation have led to the search for a more open and interoperable solution. The CAPE-OPEN (CO) effort, launched in January, 1997, is a standardisation process for achieving true plug and play of process industry simulation software components. The resulting CO standard is now being widely disseminated to the chemical engineering community. It relies on a technology that integrates up to date concepts from the software field such as a component-based approach. A number of software components based on this technology have been developed and are already available. Thanks to this new generation of CAPE tools, it is expected to reach cheaper, better and faster design, operation and control of processes. The CAPE-OPEN Laboratories Network (CO-LaN) consortium is in charge of managing the lifecycle of the CO standard.

1. Introduction and Objectives

The integration of external know-how in traditional simulation environments has to deal with proprietary code, and implies painful and expensive activities, while open software architectures are the way forward for the next generation of CAPE tools as demonstrated by (Braunschweig et al., 1999). The CO standard is the result of a world-wide collaboration between chemical and petroleum refining industries, academics, and CAPE software suppliers; with a common goal of defining a standard for a component-based approach to process simulation. Thus any compliant component can be integrated instantly in any compliant environment. The standard is open, multiplatform, uniform and available free of charge.

Version 1.0 of the CO standard was released in January 2002. This paper explains how it will be subsequently managed by the CO-LaN. The associated technology and the scope of this standard are then described. Finally we present some CO-compliant commercial and non-commercial implementation of this technology.

2. CAPE-OPEN Laboratories Network

The (CO-LaN, 2001), a non for profit organisation, maintains the CO standard, disseminates about the standard, releases software tools to test CO compliance and publish interoperability test results. With more than 20 members (operating companies, vendors and academics), it provides a service to the CAPE community in all aspects of the CO standard. The bylaws of the CO-LaN insure that it is open to the entire CAPE community for access and membership. Membership is definitely not required to use CO compliant software components or to develop such components. However the CO-LaN organises the CAPE community with respect to the CAPE-OPEN standard.

The CO-LaN operates through a web portal where visitors find or will soon find the CAPE-OPEN documentation, as well as additional resources for implementing and using CO compliant components (FAQ's, discussion board, how-to's, software migration support, etc.). Members of the CO-LaN find the services that help them to develop new standard interfaces or improve existing ones, as well as dedicated help for implementing CO compliant software components.
Special Interest Groups will be created and maintained by the CO-LaN for refining or extending the CO standard, following a careful analysis of the value creation brought to users and to vendors by each development and improvement. The SIGs will be managed in such a way as to bring these developments quickly to the market, updating as necessary the documentation and the tools delivered to the CAPE community.

3. CAPE-OPEN Technology

Key elements of CO technology are openness, interoperability, standardisation process and service. This technology integrates today's concepts in software domain, development tools and web enabled skills.
It embodies the *CO formal documentation set*, the *CO architecture*, and *the CO system model*. We focus below on the current potential for process simulation development engineers, so the *CO system model* is detailed while the *CO formal documentation set* and *the CO architecture* are only introduced.

3.1 CO formal documentation set

The CO standard follows a specific versioning system using a unique and global version number and is composed of a set of documents. These documents are organised according to a *CO formal documentation set*. This documentation set includes six blocks: *General Vision, Technical Architecture, Business Interfaces, COSE Interfaces, Common Interfaces and Implementation Specifications*.

- *General vision* contains documents that should be read first to get the standard general information, such as general requirements and needs.
- *Technical architecture* integrates the horizontal technical materials and defines an infrastructure for a process simulation based on the CO standard.

- *Business interfaces* contain all vertical interface specification documents. These interfaces are domain-specific interfaces for the CAPE application domain. They define CO components involved in a CO process simulation application.
- *COSE Interfaces* own horizontal interface specifications. They are interfaces for simulation environment such as simulator executive. Within this category, services of general use are defined such as *diagnostics* and *material template system* in order to be called by any CO components through a call back usage.
- *Common interfaces* enclose horizontal interface specification documents for handling concepts that may be required by any *Business* and *COSE interfaces*. This is a collection of interfaces that support basic functions and are always independent of *Business* and *COSE Interfaces*.
- *Implementation Specifications* contain the implementation of the *Business*, *COSE* and *Common Interfaces* specifications for a given distributed computing platform. All documents from *Business*, *COSE* and *Common Interfaces* are abstract specifications which create and document a conceptual model in an implementation neutral manner. Thus the design of CO is independent from any computing platform. It has the ability to be extended to any platform. The *Implementation Specifications* are available for (D)COM and CORBA through the Interface Definition Language libraries. In order to produce CO compliant software components any software developer has to use these official libraries.

3.2 CO architecture

The *CO architecture* elements describe technical objectives and terminology and provide the infrastructure upon which supporting *Business*, *COSE* and *Common Interfaces* are based. This identifies the technologies associated with the CO standard, includes the object model which defines common semantics, and shows the reference model which embodies the CO interfaces categories, CO (compliant) software components and communication mode. That is based on the distributed component (heterogeneous) system and the object-oriented paradigm. The involved technologies are the UML notation (Rumbaugh et al., 1997), the OMG CORBA (OMG, 2001) and Microsoft (D)COM (Microsoft, 2001) middleware, as well as the Unified Processes and object-oriented programming languages (Meyer, 2001). The wide-scale industry adoption of this CO architecture provides application developers and end-users with the means to build web-enabled interoperable simulation software systems distributed across all major hardware, operating system and programming language environments.

3.3 CO system model

The *CO system model* represents the UML design model of the standard. It defines the scope. The physical view of this model allows extraction of the CO software components and shows their dependency relationships. The logical view of this model organises the services, identifies the CO packages and CO interfaces, and designs the related structural organisation. The standard distinguishes two kinds of software components: *Process Modelling Components* (*PMCs*) and *Process Modelling Environments* (*PMEs*), the latter making use of the services provided by the *PMCs*. Typically the *PMEs* are environments that support the construction of a process model and that allow the end-user to perform a variety of different tasks, such as process simulation or optimisation (Pantelides et al., 1995). Among the standardised *PMCs* there are: *Thermodynamic and Physical Properties*, *Physical Properties DataBases*, *Unit Operations*, *Numerical Solvers* and *Sequential Modular Tools*.

- *Thermodynamic and Physical Properties* component: In the area of physical properties, CO focuses on uniform fluids that are mixtures of pure components or pseudo-components, and whose quality can be described in terms of molar composition. The physical properties operations that have been provided with standardised interfaces are those required for the calculation of vapour-liquid or liquid-solid equilibria or subsets thereof, as well as other commonly used thermodynamic and transport properties. A key concept is that of a Material Object. Typically, each distinct material appearing in a process (in streams flowing between unit operations, as well as within individual unit operations) is characterised by one such object. Each unit operation module may interact with one or more Material Objects. To support the implementation of the above framework, the CO standard defines interfaces for Material Objects as well as for thermodynamic property packages, calculation routines and equilibrium servers.
- *Unit Operation* component: CO defines a comprehensive set of standard interfaces for unit operation modules being used within modular and steady-state *PMEs*. A unit operation module may have several ports that allow it to be connected to other modules and to exchange material, energy or information with them. In the material case (which is also the most common), the port is associated with a Material Object. Ports also have directions (input, output, or input-output). Unit operation modules also have sets of parameters. These represent information that is not associated with the ports but that the modules wish to expose to their clients. Typical examples include equipment design parameters (e.g. the geometry of a reactor) and important quantities computed by the module (e.g. the capital and operating cost of a reactor).
- *Numerical Solvers* component: As explained by (Belaud et al., 2001a) the CO standard focuses on the solution algorithms that are necessary for carrying out steady-state and dynamic simulation of lumped systems. In particular, this includes algorithms for the solution of large, sparse systems of non-linear algebraic equations (NLAEs) and mixed (ordinary) differential and algebraic equations (DAEs). Algorithms for the solution of the large sparse systems of linear algebraic equations (LAEs) that often arise as sub-problems in the solution of NLAEs and DAEs are also considered. The CO standard introduces new concepts, such as models and the equation set object (ESO), which is a software abstraction of a set of non-linear algebraic or mixed (ordinary) differential and algebraic equations. The standard ESO interface enables access to the structure of the system, as well as to information on the variables involved. The equations in any model may involve discontinuities. Discontinuous equations in a models are represented as state-transition networks (Avraam et al., 1998).
- *Sequential Modular Specific Tools* component: A key part of the operation of sequential modular simulation systems is the analysis of the process flowsheet in order to determine a suitable sequence of calculation of the unit operation modules (Westerberg et. al., 1979). Thus, typically the set of units in the flowsheet is partitioned into one or more disjoint subsets (maximal cyclic

networks, MCNs) which may then be solved in sequence rather than simultaneously ("ordering"). The units within each MCN are linked by one or more recycle loops which are converged iteratively via the identification of appropriate "tear streams". The above tasks are typically carried out using a set of tools that operate on the directed graph representation of the flowsheet. The CO standard defines standard interfaces for the construction of these directed graphs, and for carrying out partitioning, ordering, tearing and sequencing operations on them.

- *Physical Properties DataBases* component: CO defines how connecting a data base with recorded physical property values and with model parameters to flowsheeting and other engineering programs. This interface deals with physical property data at discrete values of the variables of state (temperature, pressure, composition), as far as measured, correlated or estimated values are concerned. There will be no access methods that deliver recorded thermophysical property values exactly at a given state.

4. Delivering Components

Software component developers are working either to bring new CO compliant products to the market place or to make existing software components CO compliant. In either case, these software components can be for commercial sale, for proprietary use within an organisation, or for proprietary delivery to a specific client. Commercial software such as detailed by (Belaud et al., 2001b) and non-commercial software based on the CO standard are already available. Only a few are listed in order to illustrate results and potentials of the CO standardisation effort.

4.2 Process modelling environment development

The CO technology is now delivered in commercial process simulation software: Hyprotech has developed the HYSYS Unit and Thermodynamic CO sockets, which make HYSYS.Process/Plant version 2.2 (and subsequent versions) a CO compliant *PME*. AspenTech has implemented a socket for CO *Thermodynamic and Physical Properties* components in Aspen Plus 10.2 and Aspen Properties 10.2 (and subsequent versions i.e. 11.1). Aspen Plus 10.2 also implements a socket for CO *Unit Operation* components. Process System Enterprise (PSE) has released a new version of their gProms tool with a CO Thermodynamic socket. BELSIM SA has done the same for their VALI III data reconciliation tool.

4.3 Process modelling components development

Several *PMCs* are already implemented and many more are being developed. Some are only prototypes or for internal use only (for example ProSim SA delivered, exclusively to TotalFinaElf, a CO compliant thermodynamic server) while others are releases to the CAPE marketplace. In its CAPE-OPEN kit, for demonstration purposes, Hyprotech is distributing one Property Package and two Unit Operations which are CO compliant. AspenPlus can be used to create new CO physical property packages, which can be integrated in any CO compliant *PME*. Infochem has made its MultiFlash tool CO compliant and has successfully tested it with Hysys.Process, AspenPlus and gProms.

From the academic side, (Belaud et al., 2001a) from LGC-INP Toulouse institute have developed a CO-compliant *Numerical Solvers* component called Numerical Services Provider which supplies LAE, NLAE and DAE objects and acts as a real framework that makes up a reusable design for disseminating any solver algorithm through the CO standard. Furthermore, INPT has demonstrated a *Sequential Modular Specific Tools* PMC.

5. Conclusion

The CO standard gives process engineers more flexible process modelling tools by allowing simulation with software components from multiple sources, assembled easily in a simulation environment. Any CO compliant software component can be integrated in any CO compliant simulation environment by "plug and play". The CO standard benefits software component developers by increasing the usage of CAPE tools and reducing the development time thanks to the CO technology. This technology is based on the distributed component heterogeneous system and modern software development techniques. A non for profit organisation, CO-LaN, promotes and maintains the standard.

References

Avraam M., N. Shah and C.C. Pantelides, 1988, Modelling and. Optimisation of General Hybrid Systems in the Continuous Time Domain, Comput. Chem. Engng., 22S, S221-S228, 1998

Belaud J.P., K. Alloula, J.M. Le Lann and X. Joulia, Open software architecture for numerical solvers, 2001, Proceedings of European Symposium on Computer Aided Process Engineering-11, Elsevier, pp 967-972, 2001.

Belaud J.P., B. L. Braunschweig, M. Halloran, K. Irons and D. Piñol, 2001, New generation simulation environment: Technical vision of the CAPE-OPEN standard, Presentation at the AIChE Annual Meeting, Reno, NV, Nov 4-9, Modelling and Computations for Process Design, 2001.

Braunschweig B. L., C. C. Pantelides, H. I. Britt and S. Sama, Open software architectures for process modelling: current status and futures perspectives 1999, FOCAPD’99 Conference, Breckenridge, Colorado, 1999.

CO-LaN, 2001, CAPE-OPEN Laboratory Network web site: www.colan.org

Meyer B., 1997, Object-Oriented Software Construction, 2nde edition, Prentice Hall, 1997

Microsoft, 2001, Microsoft web site about COM: www.microsoft.com/com

OMG, 2001, Object Management Group web site about CORBA: www.corba.org

Pantelides C.C. and H. I. Britt, 1995, Multipurpose Process Modeling Environments. In L.T. Biegler and M.F. Doherty (Eds.), Proc. Conf. on Foundations of Computer-Aided Process Design '94. CACHE Publications, Austin, Texas. pp.128-141, 1995.

Rumbaugh J., I. Jacobsen and G. Booch, 1997, Unified Modeling Language Reference Manual, Addison Wesley, 1997.

Westerberg A.W, H.P. Hutchinson, R.L. Motard and P. Winter, 1979, Process Flowsheeting. Cambridge University Press, Cambridge, U.K, 1979.