17th European Symposium on Computer Aided Process Engineering – ESCAPE17
V. Plesu and P.S. Agachi (Editors)

A deliverable from CO-LaN to CAPE-OPEN developers and users: the CAPE-OPEN Logging and Testing Tool (COLTT)

Michel Pons,[a] Peter Banks,[b] Bertrand Braunschweig[c]

[a]*CAPE-OPEN Laboratories Network, 32 rue Raulin, F-69007 Lyon, France, technologyofficer@colan.org*
[b]*Peter Banks Associates, 9 Emmets Park, Binfield, Berkshire RG42 4HQ UK, psbanks@btinternet.com*
[c]*IFP, 1&4 avenue de Bois Préau, F-92500 Rueil-Malmaison, France, bertrand.braunschweig@ifp.fr*

Abstract

CAPE-OPEN is a set of standards that define interfaces to allow the integration of process modelling software components from diverse vendors. Since the delivery in 2002 of version 1.0 of the CAPE-OPEN standards [1], CAPE-OPEN interfaces have been widely implemented in commercial, academic and in-house software tools [2].

Achieving interoperability between complex software components from different vendors has the potential to throw up errors at run-time that are difficult to track down. In the case of CAPE-OPEN interoperability, the CAPE-OPEN Laboratories Network (CO-LaN) has provided a tool, the CAPE-OPEN Logging and Testing Tool (COLTT), to assist developers and end-users with this problem. COLTT is freely available to the CO-LaN membership.

Keywords

CAPE-OPEN, component technology, interoperability

1. Introduction

The suggestion to develop a plug and play standard in process simulation software was raised in 1994 by BP during discussions at a FOCAPD meeting held in Snowmass, Colorado. Subsequently, within the European Union (EU) funded PRIMA project (PRocess Industries Manufacturing Advantage), BP found support, particularly from Elf, BASF AG and Bayer AG, for submitting to the EU a proposal for a project based on process simulation software interoperability. This eventually led to a series of EU-funded projects starting with CAPE-OPEN and followed by Global CAPE-OPEN and GCO Support projects, which also had IMS funding. These projects developed a set of interface standards that enabled interoperability between pieces of software making up process simulation tools. The projects were all led by IFP, which continues to play a leading role in CAPE-OPEN activities. More than ten years later, software developers have implemented CAPE-OPEN interfaces in many codes and end-users are taking advantage of the interoperability provided by CAPE-OPEN. Following the end of the EU projects, the CAPE-OPEN standards are now maintained and developed by the CAPE-OPEN Laboratories Network (CO-LaN), which is funded by process industries end user companies.

2. Problem Statement, background

The CAPE-OPEN set of interface specifications makes up an impressive collection of documents that need to be read and understood precisely in order to achieve interoperability. Consequently the implementation process can lead to errors that might create interoperability mismatchs between software compnents. Some examples of implementation are available from CO-LaN but CAPE-OPEN interfaces are only specified: a developer has to code them.

Users of process simulation tools are expressing the need for reliable and seamless interoperability through CAPE-OPEN technology. Developers want to reduce the learning curve, meaning they aim at reducing the cost of adopting CAPE-OPEN. The CAPE-OPEN Laboratories Network (CO-LaN), the organization in charge of maintaining and developing the CAPE-OPEN standards [3], wants to accelerate adoption of CAPE-OPEN: getting more components and environments available with this technology implemented.

Training is a solution to ease up understanding and reducing difficulties in implementing the CAPE-OPEN technology. For example a short course on CAPE-OPEN main concepts and implementation has been delivered on November 12, 2006 as a Short Course within the AIChE 2006 Annual Meeting. However there is a need for support tools always available both to developers and to users. For end-users, using CAPE-OPEN compliant tools should not be more difficult than using components from within the generic libraries attached to any process simulator. No understanding of CAPE-OPEN technology is required to manipulate CAPE-OPEN compliant tools. However if interoperability fails, end-users need to be able to document adequately and easily what goes wrong, especially since they are often the only ones having access to all the software components involved.

3. Description of solution

The CAPE-OPEN Logging and Testing tool (COLTT) is one of the tools provided by CO-LaN to help with the above described issues.

3.1. Product overview

COLTT works with CAPE-OPEN Process Modeling Components (PMCs) and a CAPE-OPEN Process Modeling Environments (PMEs) on Windows platforms. Its role is to capture and record information about the interaction between a PME and a PMC (or combination of PMCs) in a form that makes it easy to detect problems or potential problems and to document them.

COLTT intercepts the creation of objects by modifying the Windows registry so that the appropriate logger (Unit Operation, Thermo) gets constructed instead of the requested object. The logger then constructs the requested object and forwards all calls to it. As a consequence CAPE-OPEN PMCs are not loaded differently when they are logged. The logger intercepts method calls by implementing all the interfaces that each type of PMC is expected to support.

The tool generates two types of information:

- a trace of the sequence of calls made between two components, showing arguments, results and error codes; and,
- the results of checks executed by the tool to validate that calls are in the right sequence, that the arguments are correctly implemented and that the results of the call are correctly implemented.

The tool provides the user with the ability to control which combination of components is logged, how much information is logged and where the information is logged. COLTT works only with Microsoft COM implementations of the CAPE-OPEN standards.

3.2. Development process

A COLTT prototype was developed at the end of 2004 and beginning of 2005 to demonstrate that the concept chosen for logging was adequate. This development was undertaken by Michael Halloran from AspenTech with support from SHMA Ltd and the prototype was demonstrated at the Interoperability Showcase organized by CO-LaN on February 17, 2005 in Como, Italy. Then a specification document for a complete tool was created by AspenTech under the auspices of the Interoperability Support Special Interest Group of CO-LaN.

In a subsequent development phase (Phase I), also financed by CO-LaN, forty-four combinations of PMCs and PMEs were tried out with COLTT, pinpointing a number of problems. In Phase II, the code was modified to overcome the problems found in Phase I. Phase III was begun in November 2006 to implement in COLTT the complete specification defined. These three phases were subcontracted by CO-LaN to SHMA Ltd in Pakistan.

3.3. Main product features

The starting point for the interaction between a PMC and a PME is the user selecting a PMC as part of configuring a problem in a PME. The selection available to the user is determined by the operation the user is performing – adding and using a unit operation to a flowsheet, specifying the configuration of physical properties, or selecting a numerical solver for example – and the set of installed components of the appropriate type. COLTT allows the user to configure logging for PMCs installed on the local machine only. It is possible to enable and disable logging for a particular PMC. It is also possible to find out which PMCs are being logged. The above features are available through a dedicated Graphical User Interface. The PMC components that COLTT presents for logging are the primary CAPE-OPEN components that a user can select within a PME. Secondary CAPE-OPEN components such as errors, ports, parameters and Material Objects are logged automatically as a consequence of logging a primary PMC.

By default, COLTT logs all calls made in both directions, via CAPE-OPEN interfaces between a PME and a PMC. Each call to any method from any of the interfaces generates a log entry showing:

- Which object made the call
- Which call was made
- The values for the input arguments that were passed
- The return values that were passed back
- Whether the call generated an error and what the error was – error codes are explained by a message where possible, or at least translated to a Windows or CAPE-OPEN error name such as E_FAIL or ECapeLimitedImpl.

Log files use a human-readable text format so that they can be viewed easily. For long simulations this generates very large log files. Consequently it may be difficult to identify problems due to the volume of information. In order to resolve this issue it is possible to filter out calls which are of no interest so that the log file is focused on interactions involving particular interfaces.

Configuring COLTT so that a PMC is being logged does not change the behaviour of the PMC or the PME: using a logged PMC is no different for the end-user from using a non-logged PMC. In particular:

- The lifetime of a logged component is the same whether logged or not.
- COLTT determines which of the Microsoft persistence interfaces a PMC supports and behaves accordingly when a PME requests one of the interfaces:

if the PMC does not support the requested interface then at runtime COLTT does not support it either; if the PMC does support the requested interface then at runtime COLTT also supports it and logs calls to it.

- The rules of COM identity are preserved when a PMC is logged: two requests for the same interface from the same PMC returns the same pointer.
- The set of CAPE-OPEN interfaces available to a PME from a PMC is not changed when the PMC is being logged. So for example, if a PMC does not support CAPE-OPEN error interfaces then a PME will not be able to access CAPE-OPEN error interfaces when the PMC is being logged.

3.4. Example of use

Consider in Aspen Plus 2004.2 (from Aspentech, Inc.), a process model has been developed where three Xist (from Heat Transfer Research, Inc.) Unit Operations are simulating a heat exchanger train with material recyles. Xist Unit Operations are used to simulate precisely shell&tube heat exchangers and are plugged into Aspen Plus 2004.2 using CAPE-OPEN technology. This is typically the kind of use described recently in [4].

```
MaterialObject 2 : Call to CalcEquilibrium
MaterialObject 2 : Return from CalcEquilibrium - 0x0
MaterialObject 2 : Call to get_PhaseIds
MaterialObject 2 : Return from get_PhaseIds - 0x0
 Property       Phase--- Basis--- Calc---- Value returned
 temperature    overall  (null)   (null)   338.160000000
 Property    Phase--- Basis--- Calc---- Value returned
 pressure    overall  (null)   (null)   443479.790662282
MaterialObject 2 : Call to CalcProp
 Return from CalcProp with
 CAPE-OPEN error:
 Error Name : ECapeSolvingErrorHR
 Error Code : 0x80040510
 Object Name : Anonymous
 Interface : ICapeThermoPackage
 Method : AspenCalcProp()
 Description : Unable to calculate Material Object properties in o phase.
```

All the instances of Xist UOs are logged. A sequence of calls logged is as shown above. Each call of Get/SetProp methods for example is logged with the property used as well as values of arguments such as phase, basis and property value. This enables an exact trace of the pieces of information exchanged. “Material Object 2” relates to the material stream named “2” on the flowsheet.
A call to a temperature – pressure flash is logged. Method CalcEquilibrium is invoked by a PMC that requests the Thermodynamic Server, via the Material Object, to perform such a calculation. Results of the flash calculation are saved

by the Thermodynamic Server within the Material Object. Then there is property calculation requested that ends up with a solving error.

COLTT provides the necessary information to follow closely the computational steps within both a PME and a PMC.

4. Conclusions

COLTT is typically used by individual users working on desktop or laptop computers within end-user organizations. COLTT users have some combination of CAPE-OPEN PMEs and PMCs installed on their computers. They are using COLTT to generate information when a PMC fails to interoperate with a PME or with another PMC. The information gathered by COLTT is forwarded to both PMC and PME vendors so that they can diagnose the cause of the failure. COLTT provide an objective piece of information shareable between vendors.

COLTT is also used by PMC and PME vendors to detect errors in their implementations of the CAPE-OPEN standards. COLTT is then used to log many different runs. CO-LaN Technical representatives are using COLTT in the same way as PME and PMC vendors. They perform compatibility tests between combinations of PMEs and PMCs that the vendors do not have access to.

Version 1.0 of the CAPE-OPEN standards is supported by the current COLTT version for Thermodynamic and Unit Operations Primary Objects. Expanding COLTT applicability to version 1.1 of the Thermodynamic & Physical Property interface specification is under consideration by CO-LaN. All versions of COLTT are freely available to CO-LaN members.

Acknowledgements

Contributions from Michael Halloran, formerly from AspenTech, and from SHMA Ltd in Pakistan are gratefully acknowledged.

References

1. J.P. Belaud and M. Pons, Computer-Aided Chemical Engineering, vol. 10: Proceedings of the 12th European Symposium on Computer Aided Process Engineering, pp 847-852, J. Grievink and J. van Schijndel (eds.), Elsevier Science B.V., Amsterdam, 2002.
2. M. Pons, AIDIC Conference Series, vol. 6: Selected papers of the Sixth Italian Conference on Chemical and Process engineering, pp. 253-262. S. Pierucci (ed.), Milan, Reed Business Information S.p.A., Milan, 2003.
3. J.P. Belaud et al., Proceedings of Foundations Of Computer-Aided Process Operations, Coral Springs, Florida, USA, 2003.
4. J. W. Holmes, D. L. Johnson, S. Ogundiran and M. Lichtenauer, 06'AIChE, Paper 535c, in CDRom, San Francisco, CA, 2006.