Interoperability concept in a COM thermodynamic server architecture. Example of integration in Microsoft® Excel™.

SIMO
24-25th of October 2002
Toulouse, France

Alain Vacher, Philippe Guittard

ProSim SA
Stratège Bâtiment A
BP 2738
F-31312 LABEGE Cedex
France

Alain.Vacher@prosim.net, Philippe.Guittard@prosim.bet

Summary

From their origin, simulation tools have got to answer a more and more demanding market and to adapt to the new hardware and software technologies. Today, they have to satisfy, among others, criteria such as interoperability, integration and reusability. Our StarDust® Research and Development project, centred round a new software architecture, aims to propose such an environment. Although founded on an owner specification, this project integrates CAPE-OPEN standard and rests on the Microsoft® COM/DCOM middleware, widely diffused within its Windows® operating system. Within StarDust® framework, a great care was taken in the thermodynamic server development, rightly considered as the cornerstone of any simulation application in process engineering. These server concepts make it possible to well apprehend the new architecture capacities, which can be illustrated by an example of integration in Microsoft® Excel™.

1. Introduction

In order to better determine current requirements, we will start by shortly recalling the simulation tools evolution. Even if it seems obvious, it is important not to forget that new systems we propose tomorrow must go on satisfying yesterday requirements. We will then present StarDust® Research and Development project broad outline, which defines a component oriented new software architecture that is to be adopted by all our products. We will explain our expectations about such a system and its positioning compared to the standards as well on the field of activity level, with the implementation of CAPE-OPEN interfaces, as on the purely data-processing level, with the selection of the COM/DCOM Microsoft® middleware faced with CORBA®. We will show the advantages offered by such an architecture by detailing the thermodynamic server developed within the StarDust® framework. Lastly, we will finish with an example of integration of this server in Microsoft® Excel™ to show how these two software systems can be of use to each other.

2. Simulation tools evolution

Throughout their history, process simulation tools underwent periods of change imposing to them to adapt to the current hardware and software technologies and to answer a more and more demanding market. To summarize this evolution, such as we can apprehend it today, we will describe three great generations of environment that have followed one another in the course of time:

Years 1970-80:

Tools were monolithic systems used on mainframe or minicomputers. Most of these applications, written in languages such as FORTRAN, were under the form of batch processing (input files / output files) or in the best case interactive (sequential unfolding of questions/answers). The major concern was then to have robust general tools and to maintain acceptable performances in computing times.

Years 1990:

Practices of works have been completely changed when workstations and micro-computers arrived in the market. With the increase in computers power, the performance criterion, which remains even still today a significant factor, was not any more the major concern of users and, therefore, of developers. Tools had to be convivial and easy to use. Graphical interfaces then brought an answer.

Today:

Practices are once again disturbed today by networks and co-operative work advent. In addition to the performance and user-friendliness criteria, the various tools must now be able to be substituted and to exchange services, they must be interoperable and integrable. New technologies based on components developed with object-oriented languages should make it possible to meet this new market requirement.

To conclude this short review, we can also wonder where is the future of process simulation environments in Internet expansion. As it already occurs in other fields of activity, we can imagine to find, within specialized directories, business components with standardized functions that will be able to dynamically expose their services exchange protocol in order to allow their integration in a client application.

3. StarDust® project

Considering the situation, we undertook to change the architecture of our tools to answer not only our own customers' expectations, but also the whole simulation process software users community's. This is StarDust® project ambition.

StarDust® project is built around interfaces specifications (or contracts) that have to be respected by the components representing the various objects (physicochemical properties, thermodynamic calculations, chemical reactions, unit operations...) met in our field of activity. Obviously, these interfaces are governing the various components functioning but also the way they can communicate with each other and with external entities that can be strongly heterogeneous, from a structural point of view.

Building a new architecture does not automatically imply to completely rewrite all our existing codes. One of StarDust® challenges is indeed the keeping or, at least, the minimum modification of these codes that, today, have the enormous advantage of having being tested and validated during many years, by multiple uses and in varied contexts. Developments carried out within StarDust® framework thus consist successively in a breaking and a wrapping of existing codes.

This splitting thus enables us to have a palette of components that can be assembled together and to external systems. The components assembly can thus enable us to have new general products

such as a steady state process simulator but also to have lighter systems, conceived for more specific purposes. Some components can even be viable out of StarDust® environment.

StarDust® components must answer rigorous quality criteria. Their performances have to be comparable to those recorded today in traditional codes. In addition to the technical services that are entrusted to them, components must answer user-friendliness and data persistence requirements. In other words, they have, when necessary, a man-machine interface that makes it possible to handle their data. A mechanism of data persistence that allows to store the state of the component at a given moment is also implemented. The client application is thus released of these loads by these two services, which allows it to concentrate on its own purpose. This kind of services is one of the great contributions of a components approach that is not limited to a simple modular approach.

4. StarDust® and CAPE-OPEN

The architecture imagined in StarDust®, although based on an owner specification that is distinct from CAPE-OPEN standard, ensures a total compliance to its specifications. It thus results in a perfect opening to CAPE-OPEN components that can offer their specific services within our architecture, but also offer, from their definitions, derived services such as sources of data required by StarDust® components. In addition to this interoperability placed under the inputs point of view, StarDust® components, which implement obviously their own set of interfaces, also propose, when justified, a standard CAPE-OPEN interfaces implementation.

To sum up, even if the architecture conceived within the StarDust® framework project is not based, in native, on CAPE-OPEN standard (mainly related to the keeping of existing codes), it proposes a nearly total interoperability (input and output side) with the components based on this standard.

5. COM middleware

The software components constituting the StarDust® framework must be able to be deployed on one or more computers of a local area network and to be able, nevertheless, to communicate with each other. This constraint thus requires that the components are developed on the basis of a middleware ensuring this management. In comparison with the de facto market standards, the choice is restricted to Microsoft® COM/DCOM and OMG CORBA®. They are besides the only middleware for which a CAPE-OPEN standard has been specified. CORBA®'s main asset, authorizing a platforms heterogeneous management, loses part of its interest when considering the importance of the total amount of computers operating under Microsoft® Windows®, which remains our privileged target. Another essential argument that led us to select COM/DCOM as StarDust® project basic technology is expressed by the fact that this middleware is native within Microsoft® Windows® and thus requires, for the end-user, neither installation, nor cost or administration specifically related to the middleware.

6. StarDust® thermodynamic server

Thermophysical data knowledge and estimation prove to be fundamental and essential in any process simulation tool. It thus appears natural that the first developments relate to a thermodynamic server that can be called, obviously, from all the future StarDust® components but also from any other external system.

For a better understanding of the services StarDust® thermodynamic server offers, an extract of its architecture is presented hereafter:

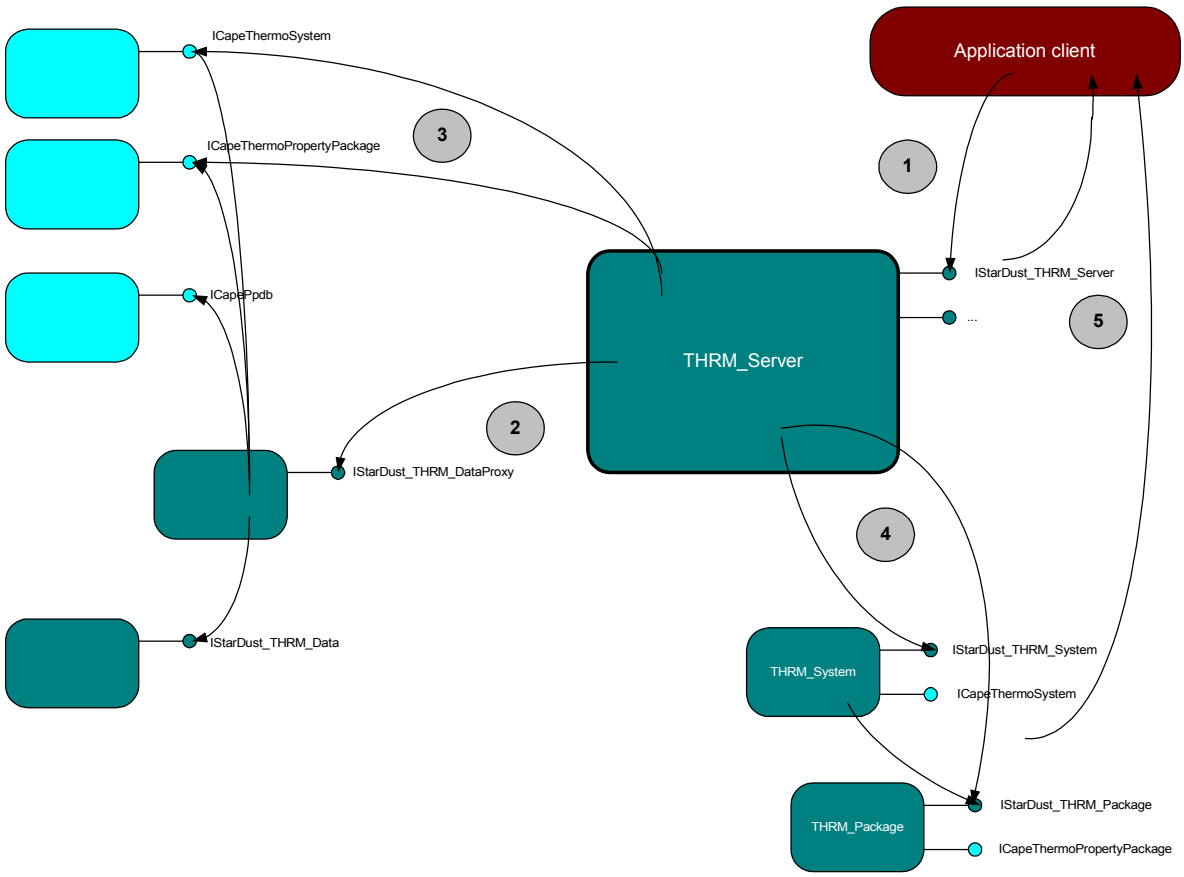

Figure 1 : Thermodynamic server archtitecture

An usual scenario of use of the thermodynamic server unfolds as follows:

1°) The client application obtains an instance from the thermodynamic server thus authorizing it to use its services thanks to the IStarDust_THRM_Server interface.

2°) Any thermodynamic server requires the knowledge of various data such as, for example, constant properties of the pure components in presence in the treated mixture. Standard interfaces (IStarDust_THRM_DataProxy) of access to these data have been created in StarDust® in order to connect to heterogeneous sources such as components internal to our architecture (IStarDust_THRM_Data) or external such as CAPE-OPEN compliant components (ICapePpdb and even ICapeThermoSystem or ICapeThermoPropertyPackage that are also properties containers). Other data necessary to the thermodynamic server are not described here.

3°) The thermodynamic server also proposes a CAPE-OPEN thermodynamic socket making it possible to use directly within our system external components implementing ICapeThermoSystem or ICapeThermoPropertyPackage interfaces.

4°) The thermodynamic server is able to create reusable components such as a thermodynamic system (IStarDust_THRM_System interface) and a thermodynamic package (IStarDust_THRM_Package interface). These last two concepts are identical to those of CAPE-OPEN and it is thus logical that these components respectively implement the ICapeThermoSystem and ICapeThermoPropertyPackage interfaces.

5°) Lastly, the client application can obtain, either directly from the thermodynamic server (THRM_Server) or systems (THRM_System) or packages (THRM_Package), the properties it is waiting for (pure components or mixture properties, fluid phases equilibria...).

In conclusion, it can be noted that StarDust® thermodynamic server authorizes the client application to connect in a transparent and centralized way to several types of components resulting from different environments. In front of the range of suggested possibilities, the client application can then choose the best option to enable it to achieve its specific goal.

7. Example of integration in Microsoft® Excel

As an illustration, we will finish with an integration of StarDust® thermodynamic server in Microsoft® Excel™ example. We will first detail an example of Visual BASIC (VBA) code carrying out a bubble temperature calculation that should allow to evaluate the implementation simplicity and the nature of the offered services:

```
Dim ThrmServer As ThrmStarDust.StarDustThrmServer
Dim nbObject As Long
Dim Composition(1 To NCMAX) As Double
Dim Pressure As Double
Dim BubbleTemperature As Double
Dim TypeOfComposition As Long
Dim IThermodynamic As Long
…
'Thermodynamic server creation…
'…storage of the state of the thermodynamic server in the XLS file
ActiveSheet.OleObjects.Add ClassType:= "ThrmStarDust.StarDustThrmServer" _
     Link:=False, DisplayAsIcon:=False, Left:=0, Top:=0, _
     Width:=0, Height:=0
'Set to a variable
nbObject = ActiveSheet.OleObjects.Count
Set ThrmServer = ActiveSheet.OleObjects(nbObject)
'Server initialization
Call ThrmServer.InitializeSvr
'Parameters input :
'1- selection of pure components (specific dialog)
Call ThrmServer.SelectCompounds
'2- selection of thermodynamic profile (specific dialog)
Call ThrmServer.SelectThermodynamics
'3- set of calculation parameters
'…thermodynamic
IThermodynamic = 1
'…pressure
Pressure = 1
'…composition (2 pure components)
Composition(1) = 0.5
Composition(2) = 0.5
'…type of composition (molar)
TypeOfComposition = 0
'Bubble temperature calculation
Call ThrmServer.ymTb(IThermodynamic,Pressure,Composition(1),_
                     TypeOfComposition,BubbleTemperature)
…
```

Figure 2 : VBA Code for bubble temperature calculation

In addition, we have developed on this basis a Microsoft® Excel™ add-in, called BibPhy Add-In, which makes it possible to carry out within an Excel™ worksheet any pure component or mixture property or fluid phase equilibria calculation. An example of a phase envelop calculation result obtained thanks to various functions available through this macro is exposed in the figure below:

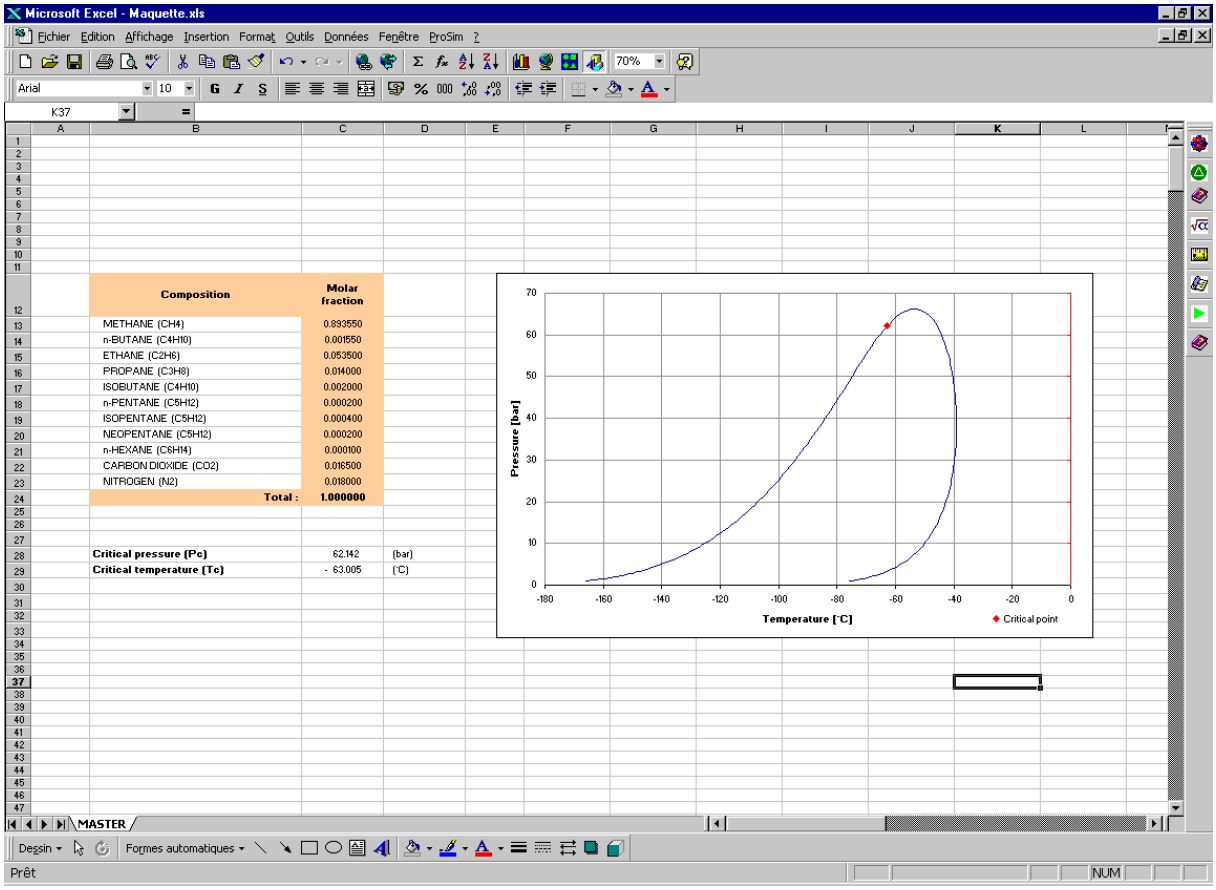

Figure 3 : Phase envelop calculated with BibPhy AddIn

8. Conclusion

From now on, components based simulation environments open new horizons to study processes. They allow process engineers, by a judicious assembly of ready to use components, to improve their developments quality while reducing associated times. Thanks to CAPE-OPEN standard, components interchangeability becomes possible. This flexibility makes it possible to select the component considered to be most powerful for a given case of figure. The component approach thus incontestably brings of gains in productivity as well to the users as to the developers of these systems.

Web sites

CO-LaN, CAPE-OPEN Network Laboratory, information relating to the standard: www.colan.org

Microsoft® information relating to COM: www.microsoft.com/com

OMG, Object Group Management, information relating to CORBA® www.corba.org

ProSim SA, Internet site: www.prosim.net