Designing a Component Store for Chemical Engineering Software Solutions

Jörg Köller¹, Thomas List¹, Matthias Jarke¹,²
¹ RWTH Aachen, Informatik V (Information Systems), Ahornstr. 55, 52056 Aachen, Germany
² GMD-FIT, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{koeller,list,jarke}@informatik.rwth-aachen.de

Abstract

In the Global Cape Open project of the G7 Intelligent Manufacturing Systems (IMS) program, the worldwide chemical industries have defined an open standard for chemical engineering simulation software. This opens up a market for simulation components representing the structure and behavior of specialized processing units for small enterprises and research institutions. This paper describes the requirements and design of a web-based infrastructure, called the Cape-Open Laboratory Network, through which such components can be certified and then integrated into simulation studies which can be bought or rented as electronic application services. A kernel of this infrastructure is being implemented.

1. Introduction

Taking a look at the current trends in software development we can observe two contradictory processes: On the one hand modern applications tend to grow more and more complex with respect to resource consumption, maintenance and functionality. Very detailed knowledge is needed to run them and large investments have to be made for providing a suitable hardware infrastructure. On the other hand, modern software systems should be designed to run from any networked computer without the need of installing or administrating a complex software. Therefore, Application Service Providers (ASPs) offer an alternative to the classical model of buying a software and installing it locally. ASPs own one or more applications and offer access to this software product via the internet.

As these companies also host these applications on their hardware the cost for using this software are dramatically reduced on the customers side because he does not have to employ a special hardware or to maintain the system himself [23,24]. Therefore, several major software vendors (e.g. Microsoft, Oracle and HP) are planning to use this new business model in forthcoming products [3]. The step beyond ASPs are ASP marketplaces where different ASPs offer their services to the customer.

This situation of continuously growing complexity is quite similar in the field of computer aided process engineering (CAPE), especially in process simulation. Process simulators are tools designed for creating mathematical models of manufacturing facilities for processing and/or transforming materials. Chemical manufacturing through continuous or batch processing, polymer processing, and oil refining are examples of such processes. Process simulators are central for designing new processes; they are also used extensively to predict behavior of existing or proposed processes. [16] and [19] give an overview of trends and state-of-the-art approaches for simulation based process engineering. Designing and simulating such processes is a very complex task which typically involves lots of different tools. Most of these tools are highly specialized, expensive, and require much expertise to run and maintain them. This would make process simulation a perfect candidate for using the ASP approach to save time and money.

But the current situation in software for CAPE application prohibits the use of the ASP approach. The tools used for process simulation are closed, monolithic applications which makes it almost impossible to include new components from other vendors or to combine these tools. But this would be very desirable, as manual exchange of data between those application is tedious and error prone. Additionally, these tools are highly heterogeneous because they may run on simple PCs using Windows or mainframes using Unix. To combine these tools each must be divided up into standardized components with defined interfaces.

This problem was addressed by the European CAPE-OPEN initiative in which the chemical industries have accomplished to standardize open interfaces for process simulation software [5,6]. The results of this initiative form the basis for bringing the ASP approach into process simulation. However, the situation differs from the classi-