Computer Aided Process and Product Engineering
Luis Puigianer and Georges Heyen

5
Emergent Standards

Jean-Pierre Belaud and Bertrand Braunschweig

5.1
Introduction

Software standards in computer-aided process and product engineering are needed in order to facilitate application and software components interoperability. In the past, end-user organizations, software companies, governmental organizations and universities have spent hundreds of thousands, if not millions of euros, dollars and yens to develop bridges between software systems such as for transferring simulation data to an engineering database in order to provide the values for basic design; for integrating real time data coming from several process control systems into a common information network for the operators; for allowing a process simulation tool to use pure component data from a physical properties data bank; for using a specialized unit operation simulation model within a commercial process simulation environment, etc.

This question has been a subject of concern for years, as a source of unnecessary costs, delays and moreover of inconsistencies between data produced and consumed by different nonintegrated systems using different bases, different calculation principles, different units of measurements, running on different computers under different operating systems and written in different languages. This need in the domain of computer-aided process engineering has been described elsewhere; see, for example, Braunschweig and Gani (2002).

Software standards remove this problem by providing the desired interoperability between software tools, platforms and databases. With appropriate machine-to-machine interface standards, using the best available tools together becomes a matter of plug-and-play, supposedly as easy as connecting USB devices or hi-fi systems[1]. Moreover, not only do these standards enable several software pieces available on your local PC to be put together, but they allow, thanks to the use of *middleware*, heterogeneous software modules available on your organizations' intranet, or on the

1 Assuming that there is one commonly agreed standard and not several, e.g., see the problems of the multiple standards for writable DVDs and the lack of interoperability that this multiplicity generates.

Computer Aided Process and Product Engineering. Edited by Luis Puigjaner and Georges Heyen

ISBN: 3-527-30804-0

internet to interoperate, e.g., thanks to *Web services* technologies. Of course, such a facility has significant organizational, economic and technical consequences. We will briefly examine these consequences at the end of this chapter.

However, our main focus will be on technologies, starting with a discussion on the concepts of openness and of open standards development. Then, we will examine some of the most significant operational standards in the domain of computer-aided process and product engineering, namely the CAPE-OPEN standard for process modeling tools, the OPC standard for process control systems. Following this, we will look at some of the current software interoperability technologies that we think will power future systems, i.e., XML and Web services technologies, leading to what is now called service-oriented architectures. Further on, we will shortly address standards for multiagent systems and the emerging Semantic Web standards, which should play a major role in the longer term, moving from *syntactic* to *semantic* interoperability of CAPE systems and services. We will conclude with a brief look at the organizational and economic consequences of the trend towards interoperability and standards.

This chapter deals essentially with software-oriented standards, i.e., standards related to the use of one piece of software from within another piece of software. *Data-oriented standards* allowing to exchange data (from databases, files, etc.) between many software applications are only marginally addressed, e.g., in the POSC section.

5.1.1 Open Concepts

There is a clear fact that the emergence of the World-Wide Web was done with concepts of common development and usage. These concepts called here open concepts commonly encompass open standards, open computing, standardization processes and open software. In the first years of e-business, (open) standards were essential to the development of the Web, to e-commerce and to inter/intra-organizational integration. Standardized information technologies such as TCP/IP, HTTP, HTML, XML, CORBA-IIOP, Web services-SOAP, etc., achieve interaction and information exchange with external or internal, homogeneous or heterogeneous, and remote or nearby applications. These technologies are now core technologies of our networked environment. For the next generation of information systems and of computer technologies, open concepts should again play a key role for emergent information technologies (IT) standards introduced in Section 5.3. Heintzman (2003) gives a good introduction to open concepts for the domain of IT, through formal definitions, a brief history from the 1970s to the modern day battle of openness, and addresses commercial challenges of open projects from an IBM perspective. There is no reason why process engineering would escape from this trend, even if this field is a niche business and therefore more restricted and less global. Section 5.2 illustrates concrete technologies using open concepts in the field of CAPE. For example, CAPE-OPEN (CO) is a significant technology for interoperability and integration of process engineering software components allowing engineering based on *off-the-shelves components*.

5.1.2
Open Standards and Standardization Process

In order to develop modern software applications and systems, technology selection involves many criteria. One main issue is to know if the technology is an (open) standard technology or a proprietary technology. Open standard technologies are freely distributed data models or software interfaces. They provide a basis for communication, common approaches and enable consistency (Fay 2003), resulting in improvements of developments, investments and maintenance. Clearly the common effort to develop an IT or a CAPE standard and its world-wide adoption by the community can be a source of cost reduction, because not only is the development cost shared but also the investment is expected to be more future-proof.

Open standards are developed by software and/or business partners who collaborate within neutral organizations (such as W3C, OASIS, OMG, etc., for IT and CO-LaN, POSC, etc., for process engineering) in accordance with a standardization process. Such organizations represent a new kind of actor additional to more traditional actors, i.e., academics, software/hardware services suppliers and end-user companies. In the information and communication industry Warner (2003) calls this standardization process *block-alliance* in committee-based standard setting and examines it with block-alliance in market-based standard battle. The latter, which is beyond our scope, leads to *de facto* standards if the resulting technology successfully matches the market. However, both approaches are not so distinct since a standardization process can be a means in a business strategy. For example the Java platform and UML mix

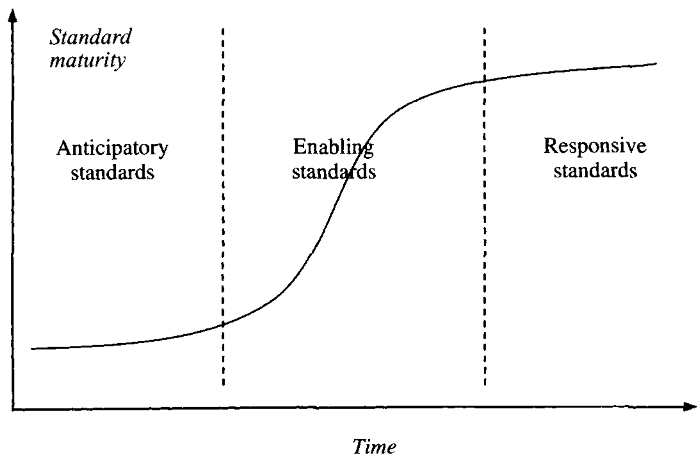

Figure 5.1 Timing of standard

committee-based and market-based processes. If we consider the S-curve lifecycle of a simple technology, Sherif (2003) classifies the technological innovations in terms of market innovation and of technological competencies with radical, platform, incremental and *architectural innovations.* Weiss and Cargill (1992) show the ideal relationship between these types of innovation and the standardization process timing with the type of standards needed at each phase (Fig. 5.1). As an illustration we would say that the CO standard is in the second phase: initial products are commercialized; CO technology is now well disseminated; there is a well-established organization releasing formal specifications; development tools, labeling process and promotion actions support the CO standard.

5.1.3
Open Computing, Open Systems, and Open/Free Software

By extension of the open standards paradigm, building modern software solutions can be based on an open computing paradigm. Open computing means that there is a standardization of information exchange. Then the resulting open system is a system whose characteristics comply with standards made available throughout the industry and therefore that can be connected to other systems complying with the same standards (IBM Glossary 2004). Open computing promises many benefits: flexibility/agility, integration capability, software editor independence, development cost and adoption of technological innovation. While always giving priority to the quality of business models available in a specific CAPE tool, process engineers can now privilege open CAPE systems, ensuring the exchange of information between CAPE solutions of distinct editors thus making it possible to benefit from various fields of expertise. This communication can be done statically with *data models* or dynamically with application programming interface (API). Open computing in CAPE is illustrated in Section 5.2.

The tools for application engineering or for software development can be open source software tools or commercial software tools. Heintzman (2003) identifies several types of projects for the development and management of open source software: academic projects (especially viewed as a new media for collaboration, innovation promotion and dissemination), foundation projects (for base software such as Linux, Apache, Eclipse, Mozilla, etc.), middleware projects (advanced software such as JBoss, MySQL, etc.), niche projects (very specific software available on the Internet[2]). Open source software projects in the CAPE field are not significant at present but they could occur in academic or niche projects, the only known example at the time of writing being SIM42 project (Sim42 Foundation 2004), which develops an open source chemical engineering simulator.

2 For example SourceForge.net is the largest repository of open source software projects with more than 118 000 projects at the beginning of 2006.

5.2 Current CAPE Standards

For several decades, experts and process engineers concentrated on the creation, evolution and improvement of models of thermodynamic and physical properties, unit operation, numerical methods, etc. Thus many CAPE software solutions allowing a more or less rigorous representation were developed. Each one is unique and dependent on the know-how of its author or editor. Particularly, in addition to the specific modeling activity, each one is characterized by selected computing technologies, i.e., supporting environment, implementation languages, persistence system, logical architecture, etc. This results in heterogeneity of available solutions and an impossibility of exchanging information between the different tools. Dual bridges between certain tools exist but this option remains proprietary and only operational for a limited number of associations of tools. Now the demand of users of CAPE tools turn to open systems, ensuring process, model and data exchange with third-party tools. In the same way, process engineers wish to be able to integrate their know-how easily and thus to deploy a final solution specific to their needs from best-in-class software components. Open computing and its related IT and CAPE standards allow to build a user-centered modeling and simulation environment from enterprise internal components and selected off-the-shelf components. Several initiatives that promote a standard for process information exchange can be identified, according to two types of techniques[3], data models and API:

- data models such as pdXML, energy eStandards from POSC and Physical Property Data eXchange from DECHEMA;
- APIs such as OPC from OPC Foundation, Physical Properties Package from IK-CAPE and CAPE-OPEN from the CO-LaN.

Open software architectures can now be exploited by the new generations of CAPE software solutions in order to provide better enterprise process applications integration. As an illustration of interest, Fieg et al. (1995), Mahalec (1998), Braunschweig et al. (2000), White (2000), Braunschweig and Gani (2002) and Belaud et al. (2002) discuss open computing, its resulting and its expected benefits. The next sections introduce CAPE-OPEN, OPC and energy eStandards.

5.2.1 CAPE-OPEN Standard for Modeling and Simulation

To solve problems, process engineers typically use a collection of in-house, commercial and/or academic software. Each user requires a broader access to available information and models to fit with the demand on the one hand, and has the constraint to match easily the old and the new, on the other hand. Information technologies play a predominant role to improve CAPE tools in supporting process engineers who

3 In some cases this distinction is not so obvious as some work both ways. Moreover, the XML technology adopted by some standards does not really comply with this classification.

face these new challenges of interoperability. It is quite obvious that work is needed to develop and establish open systems for CAPE related software. Development of open systems requires the establishment of open standards. The CAPE-OPEN standard, through which a host tool and any external tool can communicate, is the answer to this question, as it provides an *open communication system for process simulation*, allowing the final users to employ various elements within any other. Specifically, since 1995, an international group of operating companies, software suppliers and academics, developed, through the CAPE-OPEN initiative, an open communication system for key simulation elements, and demonstrated its effectiveness on numerous examples. Through this it also promoted the adoption of the open system by the major providers and users of process simulation.

The CAPE-OPEN standard (Belaud and Pons 2002, present version 1.0) consists in a technical architecture, interface specifications and implementation specifications. The technical architecture relies on modern development tools and up-to-date information technologies such as object-oriented paradigm, component-based approach, Web-enabled distributed architecture, middleware technology and uses the Unified Modeling Language (UML) notation. The interface specifications identify a conceptual model and the implementation specifications give the corresponding platform specific model for COM and CORBA. The specifications cover major application areas, e.g., unit operation, thermodynamic and physical properties, numerical solvers, optimization, planning and scheduling, chemical reactions systems, etc. CAPE-

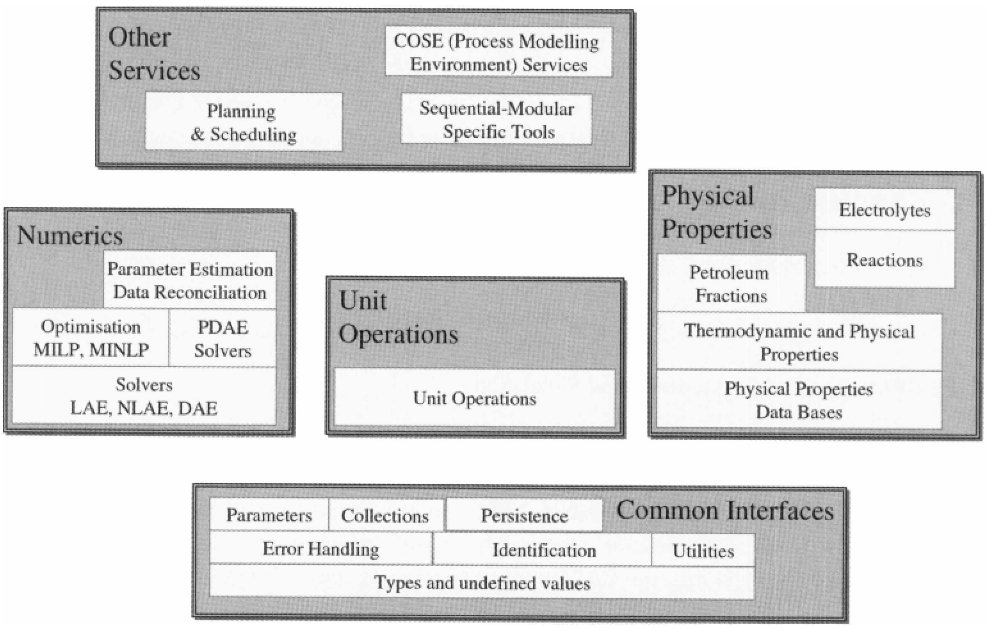

Figure 5.2 CAPE-OPEN version 1.0 specifications

OPEN compliant software environments and components are now available on the market. Belaud et al. (2003) deal with the unit operation interface and show an example for a fixed bed reactor for butane isomerization. The CAPE-OPEN standard is free of charge and is managed by the CO-LaN consortium (www.colan.org, and Pons et al. 2003), which gathers operating companies, software suppliers and academic institutes.

In addition to publishing the standard specifications, CO-LaN provides tools for supporting the transition to CAPE-OPEN technology:

- migration tools, that is, software that automate the migration of existing components to CAPE-OPEN compliance,
- code examples for re-use,
- software testers that check compliance with the standard,
- guidelines and other helpful documents.

Recent announcements from software suppliers, end-users and research institutions demonstrate that CAPE-OPEN is increasingly accepted by the CAPE community. Its main technological benefits are:

- for suppliers: increased usage of CAPE tools and reduced development and integration costs,
- for users: "develop your expertise once, plug and run everywhere" and access to best-in-class solutions,
- for academics: improved dissemination of research results and better matching with industrial needs.

Organizations who adopt the CAPE-OPEN standard, and possibly become members of the CO-LaN, will be the first ones to harvest the benefits of open standard interfaces in process modeling and simulation.

5.2.2
Extensions to the CAPE-OPEN Standard

The 1.0 version of the CAPE-OPEN standard offers the following interface specifications as shown in Fig. 5.2. Details on these specifications are available elsewhere and on CO-LaN's Web site. Although addressing a broad range of applications of CAPE modeling and simulation, the specifications are subject to improvements and extensions. At the time of writing this chapter, two such projects are active:

- Improvement and refactoring of the *thermodynamic and physical properties* specifications. This work will eventually deliver version 1.1 of the specification which should be restructured in a more logical way, better documented, and therefore easier to use.
- Extension of the *unit operation* (UO) specification. The UO CAPE-OPEN standard, in version 1.0, does only address steady state simulation; although several tests have shown that CAPE-OPEN unit operations could be used, with limitations, in

dynamic simulation, work is going on to provide a specification fully compliant with all possible uses in dynamic simulation. A new version of the UO standard will be released after sufficient testing in a number of dynamic process modeling environments.

The decision to launch a new improvement/extension project is taken by CO-LaN's board of directors following proposals presented by special interest groups or by CO-LaN members.

5.2.3 OPC for Process Control and Automation

Since 1996 the OPC Foundation (OPC Foundation 1998) has been a nonprofit organization which ensures the definition and the use of interfaces for applications in control and automation of processes. It is dedicated to ensuring interoperability in automation by creating and maintaining open specifications that standardize the communication of acquired process data, alarm and event records, historical data, and batch data to multi-supplier enterprise systems and between production devices. The vision of OPC is to be the foundation for interoperability for moving information vertically from the factory floor through the enterprise of multi-vendor systems, as well as providing interoperability between devices on different industrial networks from different vendors. The foundation gathers more than 300 members, suppliers and users of control systems, instrumentation, and process control systems. It is worth noting that Microsoft is a member and acts as a technology advisor.

The OPC-OLE for process control standard (Iwanitz and Lange 2002) is based on Microsoft OLE-ActiveX/(D)COM technology and standardizes the communication of OPC compliant data sources[4] and OPC compliant applications[5] through different connections (radio, serial, Ethernet and others) on different operating systems (Windows, Unix, VMS, DOS and others). Many specifications are available:

- OPC Data Access provides access to real-time process data,
- OPC Historical Data access is used to retrieve process data for analysis,
- OPC Alarms and Events is used to exchange and acknowledge process alarms and events,
- OPC Data eXchange defines how OPC servers exchange data with other OPC servers;
- OPC XML encapsulates process control data making it available across all operating systems.

As for CAPE-OPEN, the OPC foundation provides several tools and technologies supporting application and migration to the OPC standard, including self-testing software.

4 programmable logic controllers, distributed control systems, databases and other devices

5 human machine interface, trending subsystems, alarm subsystems, spreadsheet, historians, enterprise resource planning, etc.

5.2.4 Energy eStandards for Oil and Gas Processes

POSC is an international not-for-profit membership corporation. It unites industry people, issues and ideas to facilitate *exploration and production information sharing and business process integration* in the petroleum industry. Since 1990, membership has grown to over 100 companies. The membership includes world-wide representation of major and national oil companies, suppliers of petroleum exploration and production software and services, government agencies, computing and consulting companies, and research and academic institutions.

POSC provides open specifications for information modeling, information management, and data and application integration over the life cycle. These specifications are gathered in the energy eStandards project that relies principally on XML technologies (DTD, XML, Schema, etc.) for leveraging Internet technologies in the integration of oil and gas business processes. The set of standards are classified according to POSC areas: internet data exchange standards, practical exploration and production standards, data management standards, standards usability and application interoperability standards. For example, in the data management standards area the Epicentre standard provides a logical data model for upstream information. Also in the Internet data exchange standards area, ChemicalUsageML is a specification for the transfer of information about potential chemical hazards, and WellLogML is an XML DTD and a XML schema for well log data representation.

These standards are not directly related to CAPE applications. However, the scope of POSC encompasses both underground applications (geology, geophysics, reservoir, drilling) and offshore applications (production, transportation). The second application area has many similarities with downstream areas such as petroleum refining, as it essentially involves the design, operation and monitoring of continuous processes. Some of the POSC projects such as POSC-CAESAR delivered technologies applicable to CAPE in general. Since these are data-oriented standards we do not address them in this chapter. Commonalities can also be found with a number of data modeling projects undertaken by the chemical engineering community such as PI-STEP, PDXI or pdXML (Teague 2002 and Teague 2002b).

5.3 Emergent Information Technology Standards

Although not yet fully exploited by the CAPE community, a number of emergent IT standards will become important for our applications in the near future. Complementing some of the technologies presented in the previous section, these new IT standards support Internet-based computing and take advantage of Web technologies. We will first look at Web services together with their newly developed business standards, leading to service-oriented architectures; then we will go a step further and introduce IT standards for multi-agents architectures and the recently published[6] Semantic Web standards.

5.3.1
Web Services and Business Standards

Web technologies are being used more and more for application to application communication. Before the twenty-first century, software suppliers and IT experts promised this interconnected world thanks to the technology of Web services. Web services propose a *new paradigm for distributed* computing (Bloomberg 2001) and are one of today's most advanced application integration solutions (Linthicum 2003). They help business applications to contact a service broker, to find and to integrate the service from the selected service provider.

For example, during a simulation, the simulation environment, in need of an external thermodynamic service, contacts a UDDI directory in order to take advantage of a particular thermodynamic model (yellow page function). Once the producer of such services (a company) is selected, the simulation environment recovers the signatures of all available services using the associated WSDL descriptions[7]. These phases of discovery and description can be carried out dynamically or statically during the development process. Then the simulation environment connects to the specific thermodynamic service and uses it with SOAP[8] communication protocol. This scenario can take place on the Internet or on company intranets or extranets; it uses a set of technologies: UDDI, WSDL and SOAP, proposed by the Web services community to ensure interworking and integration of Web services.

However, even if the idea of Web services has generated too many promises[9], Web services should be viewed for now as a part of a global enterprise software solution and not as a global technical solution. In a project, Web services can be used within a general architecture relying on Java EJB or on Microsoft's .NET framework. Many projects already utilize Web services, sometimes with nonstandard technologies, particularly for noncritical intranet applications. Even if Web services miss advanced functionalities, many advantages like lower integration costs, the re-use of legacy applications, the associated standardization processes and Web connectivity can plead in favor of this new concept for software interoperability and integration (Manes 2003).

5.3.1.1
Definition

A Web service is a standardized concept of functions invocation relying on Web protocols, independent of any technological platform (operating system, application

6 at the time of writing this section (early 2004)

7 Web Service Description Language, somewhat equivalent to OMG's CORBA and to Microsoft's COM IDL

8 simple object access protocol, known as the "piping" between Web services

9 Early standards, security, orchestration, transaction, reliability, performance, ethic and economic models are the main concerns.

server, programming language, database, and component model). BearingPoint et al. (2003) focus on the evolution from software components to Web services and write: "a Web service is autonomous and modular application component, whose interfaces can be published, sought and called through Internet open standards." We see the introduction of Web services as a move from component architectures towards *internet awareness,* this context implying the use of associated technologies, i.e., HTTP and XML, and an e-business economic model. Current component technology based on EJB, .NET and CCM being not fully suitable, Web services provide a new middleware for providing functionality anywhere, anytime and to any device.

5.3.1.2
Key Principles

IBM and Microsoft's initial view of Web services, first published in 2000, identified three kinds of roles (Fig. 5.3):

- A service provider publishes the availability of its services and responds to requests to use its services.
- A service broker registers and categorizes published service providers and offers search capabilities.
- A service requester uses service brokers to find a needed service and then employs that service.

These three roles make use of proposed standard technologies: UDDI from the OASIS consortium, WSDL and SOAP from the World-Wide Web consortium (W3C). UDDI acts as a directory of available services and service providers; WSDL is an XML vocabulary to describe service interfaces. SOAP is an XML-based transfer protocol that allows you to send requests to services on through HTTP. Further domain-specific technologies related to Web services are being developed, e.g., the following proposed by the OASIS consortium, a consortium of companies interested in the development of e-business standards ebXML, supported by Sun Microsystems, is a global framework for e-business data exchange; BPEL (formerly called BPEL4WS), is a proposed standard for the management and execution of business processes based on Web services; SAML aims at exchanging authentication and authorization information; WS-Reliable Messaging is for ensuring reliable message delivery for Web services; WS-Security aims at forming the necessary technical foundation for higher-level security services, etc. A recent glossary of technologies related to Web services, each one defined by only a few lines of text, is 16 pages long (Cutter Consortium 2003).

Simply stated, the interface of a Web service is documented in a file written in WSDL and the data transmission is carried out through HTTP with SOAP. SOAP can also be used to query UDDI for services. The functions defined within the interface can be implemented with any programming language and be deployed on any platform. In fact any function can become a Web service if it can handle XML-based calls. The interoperability of Web services is similar to distributed architectures based on standard middleware such as CORBA, RMI or (D)COM but Web services offer a loose coupling, a nonintrusive link between the provider and the requester,

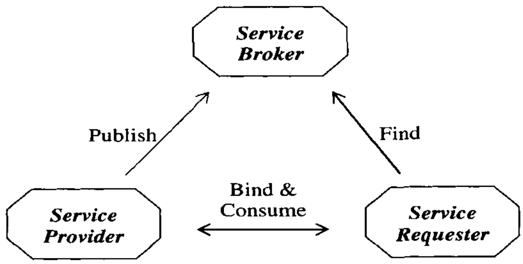

Figure 5.3 Key principles of Web services

due to the loosely-coupled SOAP middleware. Bloomberg (2001) compares these different architectures.

Oellermann (2002) discusses the creation of enterprise Web services with real business value. Basically he reminds that a Web service must provide the user with a service and needs to offer a business value. The technically faultless but closed .NET "my services" project from Microsoft demonstrates that this is always challenging to convince final users. With Google Web API beta (2004), software developers can query the Google search engine using Web services technology. Indeed Google search engine is available as a Web service since mid-2002. Search requests submit a query string and a set of parameters to the Google Web APIs service and receive in return a set of search results. A developer's kit provides documentation and example code (Java, C# and Visual Basic) for using this Web service from any platform that supports it.

5.3.1.3
SOAP: a Loosely-coupled Middleware Technology

HTML-HTTP act as loosely-coupled middleware technology between the Web client (navigator) and the business logic layer (Web server). Around the year 2000 Microsoft and IBM proposed to use the XML data format over the Internet protocols: HTTP as *transport layer* and XML as *encoding format* now constitute the key underlying technologies for Web services.

On top of these, SOAP (currently in version 1.2) was delivered in June 2003, as a lightweight protocol for exchange of information in a decentralized and distributed environment. SOAP can handle both the synchronous request/response pattern of RPC architectures and the asynchronous messages of messaging architectures. An example of SOAP request message in a synchronous manner can be found in Google Web APIs beta (2004). A SOAP request is sent as a HTTP POST. The XML content consists in three main parts:

- The envelope defines the namespaces used.
- The header is an optional element for handling supplementary information such as authentification, transactions, etc.
- The body performs the RPC call, detailing the method name, its arguments and service target.

Whereas OMG CORBA, Java RMI, Microsoft (D)COM and .NET Remoting try to adapt to the Web, SOAP middleware ensures a native connectivity with it since it builds on HTTP, SMTP and FTP and exploits the XML Web-friendly data format. The many reasons for the success of SOAP are its native Web architecture compliancy, its modular design, its simplicity and extensibility, its text-based model[10], its error handling mechanism, its ability for being the common messaging layer of Web services, its standardization process and its support from major software editors.

With so many advantages for integration and interoperability one could expect a massive adoption by software solutions architects. However the deployment of Web services still remains limited. In addition to technical issues, three main reasons can be noted:

- Web services are associated to SOAP, WSDL and UDDI. The UDDI directory of Web services launched in 2000 by IBM, Microsoft, Ariba, HP, Oracle, BEA and SAP, was operational at the end of 2001 with three functions (white, yellow and green pages). However due to technical and commercial reasons this world-wide repository that meets an initial need (to allow occasional, interactive and direct interoperability) founded on the euphoria of e-business years does not match the requirements of enterprise systems. Entrusted to OASIS in 2002, UDDI version 3.0 proposes improvements in particular for intranet applications.
- The simplicity and interoperability claimed by Web services are not so obvious. Different versions of SOAP and incompatibilities of editors' implementations are source of difficulties, to such a degree that editors created the WS-I consortium to check implementations of standards of Web services across platforms, applications, and programming languages.
- The concept was initially supported by a small group of editors (with Microsoft and IBM leading); now the "standards battle"[11] and the multiplication of proposed standards weaken the message of Web services (Koch 2003).

5.3.1.4
Service-oriented Architecture

In order to better integrate the concept of Web services in enterprise systems, IT editors now propose the service-oriented architecture (SOA) approach (Sprott and Wilkes 2004). Beyond the marketing hype, a consensus is established on the concept of service as an autonomous process, which communicates by message within an architecture that identifies applications as services. This design is based on coarse-grained, loosely-coupled services interconnected by asynchronous or synchronous communication and XML-based standards. The definition and elements of SOA are not well established yet. Sessions (2003) wonders whether a SOA is (1) a collection of components over the Internet, (2) the next release of CORBA or (3) an architecture for publishing and finding services.

An SOA is only an evolution of Web-distributed component-based architectures to get applications integration easier, faster, cheaper and more flexible, improving

10 In contrast to binary and not self-describing CORBA, RMI, (D)COM, .NET protocols.

11 with BEA, IBM and Microsoft on one side and Iona, Oracle and Sun from the other side

return on investment. In fact the main innovations are in the massive adoption of Web services[12] by the industry and in the use of the XML language to describe services, processes, security and exchanges of messages. This promises more future-proof IT projects than in the past.

Despite limitations of Web services, the technology now appears to be complementary to solutions based on classic middleware bus, as well as to enterprise application integration solutions. Its loose coupling brings increased flexibility and facilitates the re-use of legacy systems. Moreover Web services can be used like low-cost connectors between distinct technological platforms like COM, .NET and J2EE. The next release of Microsoft's Windows Vista operating system will include Indigo, a new interoperability technology based on Web services, for unifying Microsoft's proprietary communication mode; Abitboul, research director at INRIA, estimates that Web services will represent, in the long run, the natural protocol for accessing information systems. Thus it seems that we are only at the start of Web services and SOA. Andrews (2004) predicts dramatic changes in the Web services market for 2006, and announces a new class of business applications called service-oriented business applications. The merging of Web, IT and object/component technologies to form SOA and Web services is announced as the next stage of evolution for e-business (knowing that grid computing and autonomic computing will add their contributions too, but this is another story).

There is no doubt that the scientific field will get many benefits from this trend. As for CAPE, one can foresee several applications of SOA and Web services. However, it is sure that innovations will probably go beyond what is predictable at this stage of development. Here are a few examples:

- Sama et al. (2003) presents a Web-based process engineering architecture where simulator components can be executed over the Web.
- Many front-end engineering companies share design data over communications network. Access to this design data could be made easier through an SOA.
- Physical properties databases can be made available through Web services; a good example of such a service is Dechema's "DETHERM ... on the Web" on-line service (Westhaus 2004). This service is currently available through conventional technology (PhP requests on database) and could be made into a Web service, therefore directly interoperable with other programs.
- In the long run, process engineering software could interoperate with equipment manufacturers services not only to develop better simulation models by using the manufacturer's specific unit operation model, but as well to link into manufacturers' supply chain when moving into detailed design, procurement and commissioning.

As can be seen from these examples, the advent of service-oriented architectures brings many opportunities to the CAPE professional. Now let us move even further and come to semantic interoperability.

12 Even if a SOA does not imply the use of web services technology and vice versa.

5.3.2
W3C's Semantic Web Standards

The current World-Wide Web is very rich in terms of content, but is essentially *syntactic* or even *lexical*. Looking for information on the Web, using search engines, is done by finding groups of terms in the pages and in the documents, without taking consideration of the meaning of those terms.

For example, using the most popular search engine, Google, to look for information about the ESCAPE-15 conference, the first page brings the results seen in Table 5.1.

Thanks to the referencing work done by the conference organizers, the first hit is the conference's Web site. However, in the first page, together with the correct hit, Google reports a ski bag, a motor racing wheel, and a tour in New Zealand. One might wish to go to the "advanced search" page and specify that only Web sites about conferences should be returned. This is not possible, since Google does not allow this restriction. As a matter of fact, none of the most popular search engines currently used could restrict the search to a category of pages, as the *semantics* of the pages are unknown to them.

Supported by the W3C, of which it is a priority action, many projects aim at developing the semantic level, where information is annotated by its meaning. A necessary stage is to define consensual representations of the terms and objects used in the applications-these consensual representations are called *ontologies*. These ontologies will be expressed in OWL (Ontology Web Language), which itself is based on XML and RDF (Resource Description Language), a specialization of XML. Programs in the whole world support this movement towards the semantization of informa-

Table 5.1 ESCAPE-15 search with Google on 18 July 2004

ESCAPE 15 The ESCAPE (European Symposium on Computer Aided Process Engineering) series brings the latest innovations and achievements ... *www.ub.es/escape15/escape15.htm*
Thule Escape 15 Cubic Foot Rooftop Cargo Bag Buy Thule Escape 15 Cubic Foot Rooftop Cargo Bag here, one of many top quality Ski Rooftop Storage products ... *www.sportsensation.com/skiing/r/Ski-Rooftop-Storage/Thule-Escape-15-Cubic-Foot-Rooftop-Cargo--Bag-1330418.htm*
Motegi Racing **Escape**, **15**" Wheels 01-On *info.product-finder.net/motegi/ Escape--15--Wheels-01-On-154.html*
Grand **Escape** **15** Days Auckland to Christchurch This morning we journey across the Auckland Harbour Bridge traveling through small rural farming communities. Visit the Matakohe Pioneer Museum ... *www.newzealandtours.net.nz/auckland/guided/akguid66x.html*

tion. In Europe, the EC strongly supports through the Information Society Technologies (IST) program. A few ontology development projects have taken chemical engineering as their application domain. A good definition of ontologies is provided in the *Web Ontology Language Use Cases and Requirements* document published by W3C (2004):

> *Ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information. Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable.*
>
> *The word ontology has been used to describe artifacts with different degrees of structure. These range from simple taxonomies to metadata schemes, to logical theories. The Semantic Web needs ontologies with a significant degree of structure. These need to specify descriptions for the following kinds of concepts:*
>
> - *classes (general things) in the many domains of interest,*
> - *the relationships that can exist among things,*
> - *the properties (or attributes) those things may have.*

The definition of ontologies is a multidisciplinary work, which requires competence (1) in the application area: processes, chemistry, environment, etc., (2) in the modeling of knowledge into a form exploitable by machines. It is also an important stake for the actors of the field, who will use the standards defined to annotate and index their documents, their data, their codes, in order to facilitate the semantic retrieval.

Applications of the Semantic Web are many. The last section of this chapter presents an example in intelligent reconfiguration of process simulations using software agents. Before this, it is worth listing the main use cases selected by the W3C working group on the definition of OWL that have guided its development before its official release as a standard:

- *Web portals.* A Web portal powered by ontologies will bring more relevant content by applying inferences on its content (e.g., a distillation column is a separation process, therefore information about distillation would be useful to readers interested in separation).
- *Multimedia collections.* Semantic annotation of large multimedia collections will help in the retrieval among these collections, e.g., a section of a video presentation about operating special equipment.
- *Corporate Web site management.* This is the same as above, with specific functionality for company personnel, such as finding competences among employee directories etc.
- *Design documentation.* The problem of documenting designs has been identified in the chemical engineering field as in other fields where design is a key phase; it is interesting to note that this problem has been outlined by the W3C as one which could most benefit of semantic annotations, allowing to retrieve design chunks in a structured manner.

- *Agents and services.* Ontologies will be used by software agents to discover and analyze service offers and select the most relevant one; the next section presents such a system developed in the COGents EC-funded project.
- *Ubiquitous computing.* New information and technical systems will be configured at runtime by appropriate selections of services in *unchoreographed* ways, that it, in configurations which were not predicted at the time of setting up the services; annotation of ubiquitous services by ontologies will help in interoperating such combinations.

5.3.3
Use of Ontologies by Software Agents

The IST COGents developed an agent-based architecture for numerical simulation, with a concrete implementation in the process simulation domain relying on the CAPE-OPEN interoperability standard. The project, which lasted two years (April 2002–March 2004), proposed and implemented a framework, designed the OntoCAPE domain ontology of modeling knowledge, and demonstrated its benefits through case studies. COGents was funded by the European Community under the Information Society Technologies program, contract IST-2001-34431.

As before, the CAPE-OPEN standard facilitates process simulation software interoperability and can be the foundation for Web services in this domain. The COGents project pushed the technology further: we used cognitive agents to support the dynamic and opportunistic interoperability of CAPE-OPEN compliant process modeling components over the Internet. The result is an environment which provides automatic access to best-of-breed CAPE tools when required wherever situated.

For this purpose the COGents project:

- defined a framework allowing simulation components to be distributed and referenced on the Internet and intranets,
- defined representations of requirements and services in form of an ontology of process modeling, "OntoCAPE",
- designed facilities for supporting the dynamic matchmaking of modeling components,
- demonstrated the concepts through software prototypes and test cases.

The project was supported by case studies serving as examples: nylon-6 process modeling; HDA process synthesis and simulation. The nylon-6 process case study poses challenges to the component set-up and configuration: the choice on how a simulation shall be performed depends on the availability of solvers and discretization methods. The HDA process has been used as a case study in process design, process optimization and heat exchanger network synthesis. The availability of published results provides a benchmark for the agent-based design and optimization tools.

The architecture of the COGents framework is illustrated in Fig. 5.4.

The extended functionality of COGents is provided by a multi-agent system (MAS), represented by the DIMA block in the above figure. MAS aims to model com-

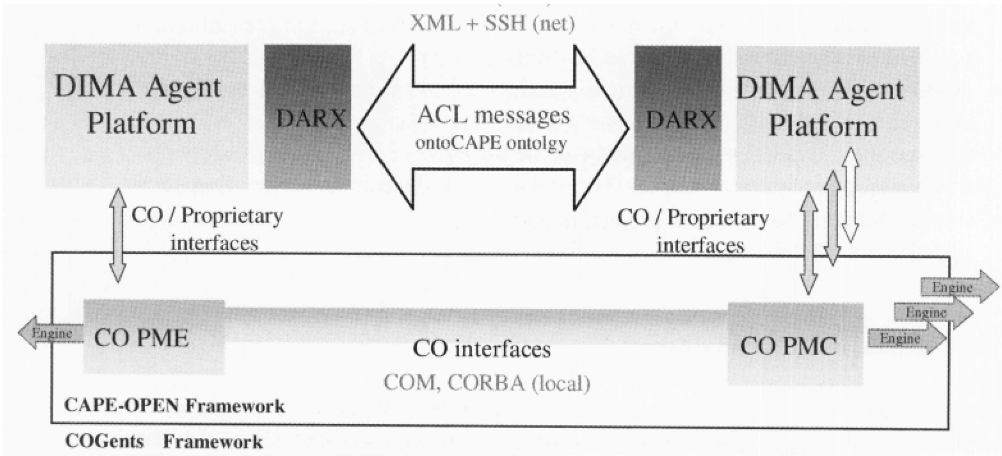

Figure 5.4 The COGents framework

plex systems as collections of interactive entities called agents. Each agent is autonomous and proactive and can interact with others and act upon its environment, applying its individual knowledge, skills, and other resources to accomplish goals. In COGents the key role of the MAS is to conduct negotiation mechanisms for composing the simulation during the design phase, as well as providing runtime facilities such as diagnostics and guidance to the users. The communication between individual agents is done with messages exchanged using an Agent Communication Language (ACL), whose content is expressed using the OntoCAPE ontology. DIMA is complemented with DARX, which provides a global naming and location service on a network. COGents integrates a security layer based on SSH, which provides strong authentication and secure communications over the Internet.

The advantages of agent-oriented approach are as follows:

- *Openness.* New Agentscan can be dynamically and easily added and/or removed.
- *Heterogeneity.* The various components can be developed with different programming languages, they can be executed on different platforms.
- *Flexibility.* Interactions between the various entities are not rigidly defined.
- *Distribution/Mobility.* The agents can be executed on a set of distributed machines and can move from one machine to another.

In COGents, agents are used to improve the dynamic of simulations and to facilitate the design and development of distributed large-scale simulations. These distributed interactive simulations are built from a set of independent simulation components linked together by a network. They provide rich adaptive simulations with agents that can interact with humans and each other.

As any application where domain knowledge has to be explicitly represented, COGents calls for an ontology to support the knowledge representation and inter-agent communication. More specifically, this ontology of the process modeling domain defines concepts indispensable for describing process modeling tasks, modeling strategies as well as software resources, and is the foundation of a matchmaking between requirements of users (i.e., process engineers) and suitable software components. OntoCAPE supports reasoning for mapping user's requests into modeling strategies and for locating software resources to implement the identified modeling strategies. OntoCAPE was developed in DAML + OIL, a predecessor of the OWL language.

More details on the COGents project, including full access to OntoCAPE, can be obtained from COGents (2004).

5.4
Conclusion (Economic, Organizational, Technical, QA)

Interoperability standards such as CAPE-OPEN, OPC, Web Services and the Semantic Web's OWL supporting reference ontologies, open new opportunities for the process industries. Once these ideas gain wide acceptance by the process engineering community, we will find ourselves facing some very major changes in the ways process engineering software are designed, developed, marketed, distributed and used, for the mutual benefit of users and vendors.

The market now has access to robust, reliable, commercial simulators that have standard software component interfaces. Process industries will be able to enjoy the lower cost and lower maintenance of commercial software, but this will be combined with an abundant flexibility. This combination will allow those companies to predict and manage process performance as never before. The number of potentially affected products is in the hundreds, due to the numerous application areas, components and suppliers. We will see many innovative combinations of process modeling components and services from large and small suppliers, used in opportunistic and changing ways depending on the modeling task at hand.

This new collaboration framework is called "co-opetition" as defined by Brandenburger and Nalebuff (1996): "Business is cooperation when it comes to creating a pie and competition when it comes to dividing it up." Plug-and-play capacity stimulates the market and creates new opportunities that could never have happened before. New value nets will be created with one supplier being another supplier's competitor, and at the same time the supplier's complement, as assembling components (or Web services in SOA) from several sources will provide more than just summing up the parts by operating them separately.

Be prepared for further innovations and business benefits in process and product engineering thanks to the increasing role of interoperability standards and to emerging information technologies.

Abbreviations

AIChE	American Institute of Chemical Engineering
API	Application Programming Interface
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Services
BPML	Business Process Markup Language
BPMI	Business Process Management Initiative
CAPE	Computer-aided process engineering
CCM	CORBA Component Model
CO	CAPE-OPEN
CO-LaN	CAPE-OPEN Laboratory Network
CORBA	Common Object Request Broker Architecture
(D)COM	(Distributed) Component Object Model
DTD	Document type definition
EAI	Enterprise Application Integration
ebXML	Electronic business XML
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDL	Interface Description Language
IIOP	Internet InterOrb Protocol
IS	Information system
IT	Information technologies
J2EE	Java 2 Platform Enterprise edition
MAS	Multi-agent system
OASIS	Organization for the Advancement of Structured Information Standards
OLE	Object linking and embedding
OPC	OLE for process control
OWL	Ontology Web Language
pdXI	Process Data Exchange Institute
pdXML	PlantData XML
OMG	Object Management group
RDF	Resource description framework
RPC	Remote procedure call
SAML	Security Assertions Markup Language
SQL	Structured Query Language
SOA	Service-oriented architecture
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery, Integration
UML	Unified Modeling Language
UO	Unit operation
WSDL	Web Services Description Language
WS-I	Web Services Interoperability Association
W3C	World-Wide Web Consortium
XML	Extensible Markup Language

Acknowledgements

The authors wish to express their thanks to colleagues of the CAPE-OPEN, Global CAPE-OPEN and COGents projects.

References

1 *Andrews W.* (2004) Predicts 2004, Gartner's predictions, www3.gartner.com/research/spotlight/asset-55117-895.jsp
2 BearingPoint, SAP and Sun Microsystems (2003), Livre blanc, Les services Web, Pourquoi? www.bearingpoint.fr/content/library/138-731.htm
3 *Bloomberg J.* Web services: A New Paradigm for Distributed Computing, The Rational Edge, September 2001, www-106.ibm.com/developerworks/rational/library/content/RationalEdge/archives/sep01.html
4 *Belaud J. P. Pons M.* Open Software Architecture for Process Simulation, Computer-Aided Chemical Engineering, 10, May 2002, Elsevier, Amsterdam, pp 847–852
5 *Belaud J. P. Braunschweig B. L. Halloran M. Irons K. Pi˜nol D. Von Wedel L.* Processus de standardisation pour l'interopérabilité des composants logiciels de l'industrie des procédés, Système d'information modélisation, optimisation commande en génie des procédés, October Toulouse, France 2002
6 *Belaud J. P. Roux P. Pons M.* Opening Unit Operations for Process Engineering Software Solutions, AIDIC Conference Series, Vol. 6, AIDIC & Reed Business Information S.p.A. (2003) pp 35–44
7 *Brandenburger A. Nalebuff B.* Co-opetition, Currency Doubleday, New York 1996
8 *Braunschweig B. L. Britt H. Pantelides C. C. Sama S.* Process Modeling: the Promise of Open Software Architectures, Chemical Engineering Progress, September (2000) pp. 65–76
9 *Braunschweig B. L. Gani R. (eds.)* Software Architectures and Tools for Computer-Aided Process Engineering, Elsevier, Amsterdam 2002
10 COGents (2004), COGents project Web site, www.cogents.org
11 Cutter (2003) Consortium Web Services Terminology, Web Services Strategies, Vol. 2, No. 12, December 2003
12 *Fay S.* (2003), Standards and Re-Use, The Rational Edge, May 2003, www-106.ibm.com/developerworks/rational/library/2277.html
13 *Fieg G. Gutermuth W. Kothe W. Mayer H. H. Nagel S. Wendeler H. Wozny G.* A Standard Interface for Use of Thermodynamics in Process Simulation, Computers and Chemical Engineering, Vol. 19, Suppl., (2002) pp. S317–S320
14 Google Web APIs Beta (2004), www.google.fr/apis/index.html
15 *Heintzman D.* (2003), An Introduction to Open Computing, Open Standards, and Open Source, The Rational Edge, July 2003, www-106.ibm.com/developerworks/rational/library/content/RationalEdge/archives/july03.html
16 IBM Glossary (2004), Glossary of Computing Terms, www-306.ibm.com/ibm/terminology/goc/gocmain.htm
17 *Koch C.* (2003), The Battle for Web Services, CIO magazine, October 2003, www.cio.com/archive/100103/standards.html
18 *Iwanitz F. Lange J.* OPC-Fundamentals, Implementation and Application, Hüthig Fachverlag 2002
19 *Linthicum D. S.* Next Generation Application Integration: From Simple Information to Web services, September 2003, Addison Wesley, Boston 2003
20 *Mahalec V.* Open System Architectures for Process Simulation and Optimization, AspenTech Speech, ESCAPE'8 Conf., Belgium, 25 May 1998
21 *Manes A. T.* Web Services: a Manager's Guide, September 2003, Addison Wesley, Boston 2003
22 *Oellermann W.* Create Web Services with Business Value, .NET Magazine, November 2002, Vol. 2, Number 10, www.ftponline.com/wss/2002-11/magazine/features/wollermann/default.aspx
23 OPC foundation (1998), OPC Technical Overview, www.opcfoundation.org/01-about/OPCOverview.pdf

24 *Pons M. Belaud J. P. Banks P. Irons K. Merk W.* Missions of the CAPE-OPEN Laboratories Network, Proceedings of Foundations of Computer-Aided Process Operations, 2003, Coral Springs, Florida 2003

25 *Sama S. Piñol D. Serra M.* Web-based Process Engineering, Petroleum Technology Quarterly, 2003

26 *Sessions R.* What is a Service-Oriented Architecture (SOA)? ObjectWatch Newsletter, Number 45, October 2003, www.objectwatch.com/issue-45.htm

27 *Sherif M. H.* When is Standardization Slow ?, International Journal of IT Standards and Standardization Research, Vol. 1, Number 1, March 2003

28 Sim42 Foundation Simulator 42 Open Source Chemical Engineering Process Simulator, 2004 www.virtualmaterials.com/sim42

29 *Sprott D. Wilkes L.* Understanding Service-Oriented Architecture, Microsoft Architects Journal, EMEA edition, January 2004, www.thearchitectjournal.com/Journal/issue1/article2.html

30 *Teague T. L.* Electronic Data Exchange Using PlantData XML, AIChE Spring National Meeting, 10–14 March 2002, New Orleans Riverside, New Orleans 2002

31 *Teague T. L.* PlantData XML, Section 4.3 of Software Architectures and Tools for Computer-Aided Process Engineering, Elsevier, Amsterdam 2002b

32 W3C (2004) W3C, World-Wide Web Consortium, Web Ontology Language Use Cases and Requirements, www.w3.org/TR/2004/REC-webont-req-20040210/

33 *Warner A. G.* Block Alliances in Formal Standard Setting Environments, International Journal of IT Standards and Standardization Research, Vol. 1, Number 1, March 2003

34 *Weiss M. Cargill C.* Consortia in the Standards Development Process, Journal of the American Society for Information Science, Vol. 43, Number 8 (1992) pp. 559–565

35 *Westhaus U.* DETHERM ... on the Web, an On-line Service from DECHEMA, http://i-systems.dechema.de/detherm/ 2004

36 *White M.* Working Together: Collaborative Competition Creates New Markets, Cap Gemini Ernst & Young Center for Business Innovation E-journal, Issue 5 (2000) pp. 33–35, www.cbi.cgey.com/journal/issue5/index.html