Designing Workflow-Enabled Business-to-Business Infrastructures

Diogo Ferreira[1], J. J. Pinto Ferreira[2]

[1]*INESC Porto, Rua José Falcão 110, 4050-315 Porto, Portugal, dmf@inescporto.pt*

[2]*Faculty of Engineering U. P., Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal, jjpf@fe.up.pt*

Abstract

Companies are facing unprecedented challenges as improved connectivity opens new markets and news ways of conducting business. There are new possibilities of increasing intimacy with business partners and of co-ordinating business processes with them. As a result, several Business to Business (B2B) infrastructures are being proposed from a wide range of approaches, leveraging on Internet technologies. But many of these initiatives overlook the fact that deploying sophisticated technologies is not enough to produce a B2B success; companies must be able to align their business processes in order to benefit from the potential of e-business. Workflow management is precisely the paradigm which, built into B2B infrastructures, will provide the desired business process alignment.

Keywords

E-commerce, Workflow management, Business-to-business frameworks

1 Introduction

E-commerce has usually been divided into two main types: B2C and B2B. However, the development of both types has unveiled different requirements. B2C e-commerce was the first step, targeting the individual consumer and focusing in content presentation, order processing, and security. Then B2B introduced additional requirements, involving the engineering of interactions of diverse, complex enterprises, where interoperability plays the key role. This paper is about designing B2B infrastructures supporting that interoperability, but introducing workflow management as the paradigm that enables business process alignment between trading partners.

2 B2B Trends

Recent trends in B2B e-commerce have led to the appearance of B2B marketplaces. These e-marketplaces are Web portals where suppliers, buyers or both get together to explore new opportunities and establish trading communities. E-marketplaces are typically vertical, meaning that they focus on a particular type of business. The most common type of e-marketplaces is the neutral exchange marketplace, set up by an independent third party and where both buyers and suppliers within a particular industry sector can be found [Forbes, 2000]. They usually comprise two main areas: one dedicated to the member suppliers advertising selling offers and another to member buyers where they advertise purchase needs. Contact between buyer and supplier is mediated through the independent third party, which charges a service fee that may take the form of a transaction value percentage. Ironically, this prevents buyers and suppliers from trading directly, which raises the question of whether these marketplaces classify as a form of B2B e-commerce connectivity or not.

A different case is the mySAP.com initiative from SAP. The mySAP.com Marketplace [SAP, 2000] is an open portal, which everyone can join and showcase or deliver products and services. This marketplace, which is hosted by SAP, not only allows for finding suppliers and customers but also provides the functionality to actually attain B2B trading. The marketplace is also a horizontal one, providing access to a wide range of industry sectors. This form of marketplace is more open than most other portals, but the technology enabling B2B transactions is a proprietary solution from SAP.

Aiming at the same B2B interoperability, several companies have formed consortia to develop B2B frameworks, which can provide generic functions enabling businesses to communicate over the Internet [Shim, Pendyala, Sundaram, Gao, 2000]. Instead of building proprietary solutions, these consortia aim at developing industry-wide standards that companies can easily adopt.

One of the first standards was proposed by the Open Buying on the Internet (OBI) Consortium [OBI Consortium, 1997], which is based on a particular model of business-to-business interaction, basically dealing with orders placed by a buying organisation on a selling organisation. Orders are mapped into EDI-compatible order requests (and transmitted over HTTP). In this respect, OBI is more the exception than the rule since most of the current B2B frameworks are more favourable to the use of XML than EDI. Apart from the discussion of whether XML will replace or complement EDI, the OBI framework actually specifies in detail the content of each order request. On another perspective, OBI also specifies each step of the business logic, i.e. the complete b-to-b interaction under which orders are transmitted and processed. This interaction comprises a *requisitioner* belonging to a buying organisation who places orders on a selling organisation, and a payment authority which issues invoices and receives payments. The OBI standard specifies the implementation of most of the interaction between these entities, according to its own B2B commerce model.

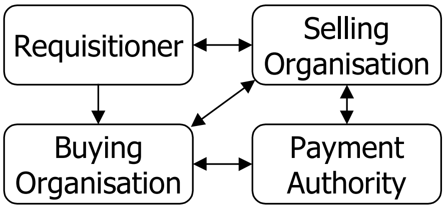

Figure 1: OBI e-commerce model

Other frameworks have put fewer requirements on either the document specification or the business logic, or both. The cXML framework [Ariba, 1999] is such a framework, since it relies on similar roles for the buying and selling organisations but it places fewer assumptions on their interaction. The cXML framework requires the specification of catalogue and purchase order XML documents that trading partners will exchange during procurement transactions. It does not assume a rigid B2B interaction model as OBI does, and relies on the trading partners to effectively define those documents, given some introductory examples. The framework is actually more focused on the application level by specifying how two to set up a sales site for interactive catalogue presentation, known as *punchout*. The content on this site is then provided to procurement applications on buying organisations, by way of an e-commerce hub (or *punchout dispatcher*) which mediates the B2B interaction. The presence of this go-between can be better understood when considering that this is akin to the way Ariba provides some of its B2B solutions (an example can be found at [ForumB2B.com, 2000]).

Another framework that puts strong emphasis on the application level is BizTalk™ [Microsoft, 1999]. BizTalk promotes the exchange of XML documents through an application architecture where a specific product – the BizTalk Server – plays a fundamental role. Moreover, the BizTalk framework specifies only the document headers, or message envelope, necessary for exchanging XML documents between servers. Regarding the actual message content, the framework relies on the contribution of industry groups or developers in order to achieve a comprehensive set of reusable document types. However, the framework cannot be taken under consideration without the server, which implements the relevant features, but is available as a commercial product, interoperable only with servers of its own kind.

At the opposite end of the spectrum, we would find the eCo framework [CommerceNet, 1999]. The eCo framework is aware that e-commerce systems implementation varies widely, and claims that confining those systems to specific architectural models hinders the desired B2B interoperability. As a result, eCo does not represent any particular e-commerce system; instead, it represents those aspects of an e-commerce system which contributes to its interaction with a prospective trading partner. The eCo framework actually works as a conceptual framework through which other e-commerce frameworks or systems can be applied. The implementation of eCo is straightforward: a company should place a document called “eco.xml” at the root level of their site. This document describes the interoperability capabilities of that site according to a set of interoperability layers, for which there are particular XML formats. The framework further defines how to discover and query eCo compliant sites through an URL mechanism, with an ingenious way of leveraging existing Web infrastructures.

Targeting business level, application level and data level integration, the RosettaNet framework [RosettaNet, 1998] is one of the most comprehensive B2B frameworks, which comprises a methodology for the design of interoperable e-commerce systems. A central focus of the RosettaNet framework is the definition of Partner Interface Processes (PIPs) that describe business connectivity between supply chain partners. In addition, RosettaNet makes use of dictionaries that set the semantics of the business and data models. RosettaNet already provides most of these elements, such as a comprehensive set of PIPs and the data dictionaries, targeting the Information Technology and Electronic Components sectors. Therefore, these are the industrial sectors that can better benefit from the use of this framework.

Other frameworks have also been devised with particular industries or services in mind. Especially for financial-related services we would find the FIX Protocol [FIX Protocol, 2000], which defines message format and session-level interaction between two applications. However, most initiatives in this sector (and others) usually target document structure alone, as illustrated by markup languages such as FinXML [FinXML.org, 1999]. For the banking and logistics sectors bolero$_{\text{XML}}$ [Bolero, 2000] provides, against a service fee, a common messaging platform for securely exchanging pre-defined XML documents over the Internet.

3 Findings

The previous section has presented some of the current approaches towards B2B interoperability. Though not exhaustive, the most representative initiatives have been briefly described. From that it can be seen that current B2B frameworks concentrate on providing architectures for the exchange of information between business partners. Some of these architectures are open and may be freely implemented while others rely on a specific product or service. Emphasis is put on XML [W3C, 1998] as the enabling technology for defining and exchanging common business documents.

These frameworks comprise basically two different approaches for tackling interoperability. On one hand, there are initiatives which aim at setting a common data format enabling the exchange of information between trading partners within the same industry or service sectors (e.g. FinXML). This is especially the case of several XML-based specifications that define the document structure to be used by applications.

On the other hand, there are frameworks trying to cope with particular B2B interaction models, either defined by the same standard (e.g. OBI) or to be defined by the target companies (e.g. eCo). These initiatives are confronted with a wide diversity and complexity of B2B interactions, and therefore must find a way to keep the standardisation effort under a manageable scope. For example, the cXML framework targets procurement activities alone, regardless of the industry sector. However, the RosettaNet standard is dedicated mainly to the electronic components sector, while supporting an extensive set of processes ranging from marketing to inventory and order management.

These approaches actually hamper the applicability of such frameworks. A narrow focus on a specific industry sector or business area confines the approach to a limited use and precludes the implementation of the same framework in other business environments. This also means that other solutions will have to be worked out for other sectors, increasing the number of different – and therefore incompatible – solutions being implemented.

This is a consequence of the assumptions that underlie the development of such frameworks. When assuming a particular and detailed B2B interaction model, the whole framework becomes dependent upon that model, and can be hardly reused when another interaction model takes place. When assuming no interaction model at all, the framework has limited usefulness as it becomes confined to the scope of application integration. Defining data formats is essential, but should not be confused with the actual business transactions taking place. Therefore, an interaction model is surely needed for implementing a B2B infrastructure, but then no particular model should be taken for granted. Designing a B2B framework should leave room for subsequently defining the interaction model – in effect, the business process – that the trading partners will perform.

Then it will become clear that a generic B2B infrastructure must be workflow-enabled, i.e., it must support the definition and enactment of business processes that are different among industry sectors, or from company to company. A B2B infrastructure should be, in fact, a workflow management system (WfMS) which implements business processes that are performed by resources belonging to different organisations. These business processes describe the exchange of certain types of data between those resources. Those data types or formats are to be defined independently, possibly according to an existing XML-based standard, but regardless of the interaction effectively taking place.

This approach has two important characteristics:

- the infrastructure is able to support the exchange of any XML-based data format, be it financial, logistic, marketing, production, etc.;
- the infrastructure is able to support any interaction model that can be described as sequence of information exchange steps between trading partners.

4 Design Principles

To design a B2B infrastructure exhibiting these characteristics, there are two main guidelines that should be followed: (1) clearly separate data requirements from interaction behaviour, as a way to support exchange of any kind of data, and (2) use a workflow-enabled system to set up behaviour, as a way to support any kind of B2B interaction. These two guidelines lay down the foundation for a workflow-based approach to the design of B2B infrastructures, which can be highlighted through the following set of principles.

4.1 Identify information exchange requirements between organisations

In B2B transactions, companies need to exchange product info, stock levels, orders, invoices, shipping info, and other logistic data. But they might as well need to exchange marketing data, sales reports, budgets, predictions, etc. In essence, companies exchange certain types of data according to given protocols, which the B2B infrastructure will support. Identifying these data and protocols is the first step to tackle the problem at hand, and to ascertain the role the infrastructure will play in the business trade.

4.2 Identify data formats to be standardised

Once the B2B scenario becomes perceptible, application requirements will arise. In fact, applications, and even users, will expect those data to be presented in a certain format, while others will be able to present maybe a similar kind of information in a completely different way. Most probably neither format is the most advantageous, and the usual approach is to define a neutral one that can be easily transferred and parsed through the infrastructure. The solution is then to translate all documents to be exchanged into an XML-based format. That translation should prove to be useful regardless of the underlying infrastructure, since most B2B frameworks are based on XML, and it is also a step towards enabling information to be presented through the Web.

4.3 Describe interaction behaviour between applications

Supposing that all information items can be standardised to a common format, now we are left with sorting out the correct sequence for exchanging them. This input comes from how the companies already do, or want to do, their information exchange. If administrative approval is needed for an order request or not, or if there is payment authority issuing the invoice, are details to be figured out on a case-by-case basis. Beyond the data, the exact sequence and actors taking part in the exchange must be unambiguously described.

4.4 Application integration is declarative

In former times, the challenge of application integration has been trying to programmatically connect applications, since each of them was developed within a proprietary environment. But now that several interoperability solutions exist, application integration concentrates on defining the data formats that applications will interchange, despite of how they will reach that information. Therefore, sharing information between applications should not be attempted by defining programming interfaces. Instead, it consists on defining those XML-based documents previously identified. Thus, connection constraints arise not from technological limitations but from semantic mismatches between applications. One way of illustrating connection possibilities is through the use of a publish-subscribe matrix showing, for each data format generated by one application, which applications are able to consume it.

Publish Subscribe	**App1**	**App2**	**App3**
App1		DTD (A)	
App2	DTD (B)		DTD (C)
App3	DTD (D)		

Figure 2: Document definitions enabling applications to communicate

4.5 Integrate applications through asynchronous messaging

The effective way to make unknown applications communicate is through Message Oriented Middleware (MOM). On one hand, the applications may not be online at the same time or may not be able to handle incoming data at any moment. On the other hand, applications will deal with a single programming interface, which is the MOM interface, through which they will be able to reach any other application connected to the MOM platform. The MOM platform works as a message queuing system where each message will contain a document following one of the previously specified XML-based formats. Furthermore, a MOM platform is able to provide addressed communication, where the recipient for a message is specified, or multicasting as in a publish/subscribe paradigm.

4.6 Business integration is procedural

After having defined data formats and having provided a communication platform, it would seem that the infrastructure is ready to work. Some B2B frameworks stop here, but actually this is the point where the alignment of business processes between companies begins. In fact, when a piece of information moves from one company to another it is not a transaction between two applications on different locations; it is a business interaction between two business processes that are implemented by two different trading partners. Then the description of the interaction between applications (section 4.3) comes into play. This interaction logic, which is procedural in nature, should not be *hard-coded* inside the applications; instead, it should be implemented by a co-ordination facility working on top of the MOM platform. That co-ordination facility, which takes the interaction logic as input, will control the routing of information between applications, according to the desired B2B processes.

4.7 Integrate businesses through workflow processes

The co-ordination facility just mentioned is a workflow management system (WfMS), and the interaction logic is a set of workflow process definitions. The WfMS will create an instance of a given process definition whenever the triggering conditions for that process happen to be true. A triggering condition may be a certain application publishing a specific message on the MOM platform, which results in the corresponding process being instantiated. This process instance will then enforce a sequence of message exchanges, according to its definition. At each step of that sequence, the WfMS will send a message as input data to an application, and expect a result that will be forwarded to the next step. When all steps become complete, the process is finished.

4.8 Use existing standards whenever possible

Several B2B frameworks have already undertaken significant data standardisation efforts in establishing document formats, so instead of defining those documents from scratch, B2B infrastructure developers can leverage on several results from previous work, such as the frameworks mentioned on section 3.

When defining workflow processes or designing workflow functionality, developers can also rely on standardised approaches that will provide the highest degree of interoperability with other workflow systems. In this respect, the Workflow Management Coalition (WfMC) hosts most of the workflow standardisation initiatives [WfMC, 1999], and the Object Management Group (OMG) also provides valuable contributions [OMG, 2000].

Concerning the MOM platform, several solutions exist such as IBM MQSeries (http://www.ibm.com/mqseries), Microsoft MSMQ (http://www.microsoft.com/msmq), or implementations of the standard CORBA Notification Service specification [OOC, 2001].

5 Case Study

The DAMASCOS project (IST-1999-11850) aims at providing a set of IT modules and mechanisms for supply chain management, focusing especially on small and medium enterprises (SMEs) working within Customised Supply Networks [DAMASCOS, 2000]. As key features, DAMASCOS employs forecasting techniques for seasonal sales prediction, and workflow management technologies for co-operative information management. The set of IT modules, called the DAMASCOS suite, comprises functionality supporting sales, distribution, production orders, forecasting, and co-ordination; this suite is to be integrated with existing legacy systems on the target companies.

Concerning B2B infrastructures, the interesting feature is that the DAMASCOS suite comprises an integration infrastructure – the Workflow Backbone (WfB) – whose purpose is to integrate the other modules, both within the company and across the supply chain. The WfB is a workflow-driven infrastructure where messages are routed between applications according to workflow models that describe the intended interactions. At its lowest level, the WfB is a persistent messaging platform that allows applications to exchange XML documents. This messaging platform is based on a CORBA Notification Service implementation [OOC, 2001], although a lighter, simpler CORBA and COM interface was devised providing three main methods for applications to use:

- push(), allowing an application to send or publish a message into the infrastructure;
- pull(), which allows an application to check for new messages, and can work on a blocking or non-blocking mode; the message brings the identifier of the sender;
- and reply(), allowing to send a message to a specific application, in reply to a previously received message.

Despite having access to these methods, applications never communicate directly because, connected to this same messaging platform, there is a workflow facility that governs all message exchanges. This workflow facility receives all messages submitted with push(), and triggers the appropriate process according to each message received. For each activity in the triggered process, the workflow facility sends an input message to the application assigned to that activity, and waits for an output response. This response will be returned with a call to reply().

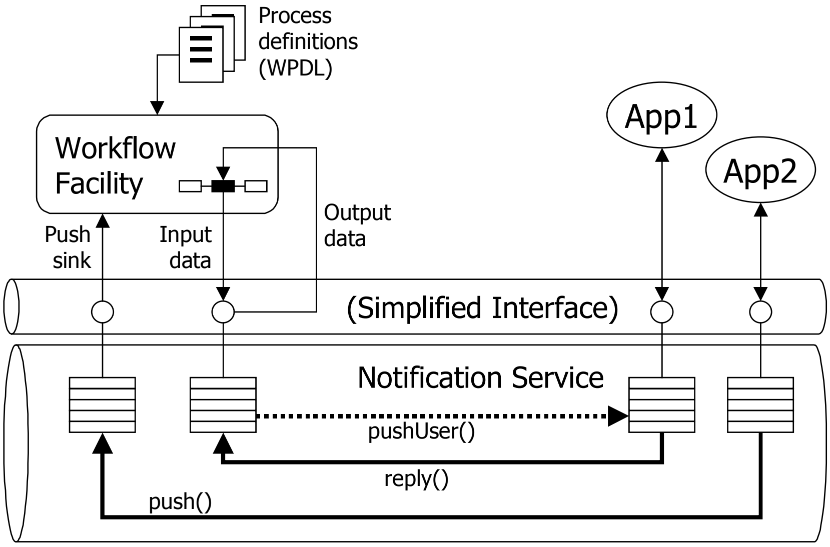

Figure 3: The DAMASCOS Workflow Backbone

This workflow facility has been implemented according to the OMG Workflow Management Facility [OMG, 2000], and processes are defined according to the standard WPDL language [WfMC, 1999]. This means that the infrastructure is interoperable with other workflow systems, either through a standardised interface, or through the exchange of process definitions.

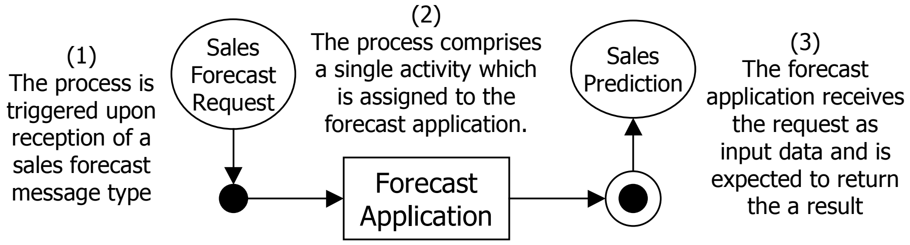
(1)
The process is triggered upon reception of a sales forecast message type
Sales Forecast Request
(2)
The process comprises a single activity which is assigned to the forecast application.
Forecast Application
Sales Prediction
(3)
The forecast application receives the request as input data and is expected to return the a result

Figure 4: The sales forecast process definition example

The two main challenges that the infrastructure now faces are (1) federation and (2) security. As far as security is concerned, a CORBA over SSL implementation seems mandatory, together with application authentication rules, some of which the WfB already enforces. But in the case there are two instances of the infrastructure installed in two companies, an application must be developed in order to transfer messages from one Notification Service into the other. This cannot be done with CORBA, unless the ORBs are working on a specific port that is open throughout the network. As this might be difficult to achieve, the solution could be a TCP/IP socket application that would transfer those events on a peer-to-peer basis. But then, ironically, precisely the issues that the Notification Service handles, such as persistence, guaranteed delivery, no message duplication, etc., would arise.

6 Conclusion

Current B2B infrastructure developments fall short of their objectives, either because they assume particular B2B interaction models or because they assume none at all. Workflow management brings flexibility to the B2B arena by modelling B2B interactions as business processes, and using those processes to orchestrate information exchange. While providing the semantic and communication interfaces between applications is certainly essential, the interaction logic should be left to a B2B infrastructure, which will implement it according to changing business requirements.

Acknowledgement

The second author wishes to acknowledge the European Commission, which is funding the DAMASCOS project (No. IST-1999-11850) under Key Action II. The first author would like to acknowledge the Foundation for Science and Technology (http://www.fct.mct.pt), which is funding his research.

References

Ariba, Inc.: cXML User‘s Guide. WWW page. http://www.cxml.org/home/index.htm, accessed 22.2.2001.

Bolero International Ltd.: Guide to bolero$_{XML}$. WWW page. http://www.bolero.net/downloads/boleroXML_standards.pdf, accessed 25.2.2001

CommerceNet, Inc.: eCo Architecture for Electronic Commerce Interoperability. WWW page. http://eco.commerce.net/rsrc/eCoSpec.pdf, accessed 23.2.2001

DAMASCOS: Dynamic Forecast for Master Production Planning with Stock and Capacity Constraints. WWW page. http://www.damascos.com, accessed 21.2.2001.

FinXML.org: FinXML™ Solutions. WWW page. http://www.finxml.org/Solution/Solutions.asp, accessed 25.2.2001

FIX Protocol Ltd.: Financial Information Exchange Protocol. WWW page. http://www.fixprotocol.org/specification/fix-42.html, accessed 25.2.2001

Forbes.com: Best of the Web > B2B. WWW page. http://www.forbesbest.com/b2b/, accessed 21.2.2001.

ForumB2B.com: Demo para compradores (in portuguese). WWW page. http://www.forumb2b.com/asp/tour_compradores/tour_main_log.asp, accessed 22.2.2001

Microsoft Corp.: The Biztalk Framework. WWW page. http://www.microsoft.com/biztalk/techinfo/framework20.htm, accessed 22.2.2001

OBI Consortium: Open Buying on the Internet Technical Specifications. WWW page. http://www.openbuy.org/obi/specs/, accessed 21.2.2001.

OMG: Workflow Management Facility Specification V1.2. Available by FTP. ftp://ftp.omg.org/pub/docs/formal/00-05-02.pdf, accessed 21.2.2001.

OOC: ORBacus Notify. WWW page. http://www.ooc.com/notify/, accessed 11.03.2001

RosettaNet: Lingua Franca for eBusiness. WWW page. http://www.rosettanet.org, accessed 23.2.2001

SAP AG: mySAP.com Marketplace. WWW page. http://www.mysap.com/solutions/marketplace/, accessed 21.2.2001.

Shim, Simon; Pendyala, Vishnu; Sundaram, Meera; Gao, Jerry: Business-to-Business E-Commerce Frameworks. IEEE Computer, Vol 33, No 10, October 2000, p. 40-47.

W3C: Extensible Markup Language (XML). WWW page. http://www.w3.org/XML/, accessed 21.2.2001.

WfMC: Interface 1: Process Definition Interchange V1.1. WWW page. http://www.aiim.org/wfmc/standards/docs/if19910v11.pdf, accessed 11.03.2001