

A focus on a b2b e-commerce architecture

David Basanta Gutiérrez, Lourdes Tajés Martínez

Computer Science Department, University of Oviedo
Oviedo, Spain

basanta@ieee.org; tajes-martinez@acm.org }

Abstract

The term electronic commerce is generally applied to any form of business transaction by means of telecommunication networks. This involves the use of information technology (such as electronic data interchange (EDI), electronic mail, transfer of money, etc) to link the functions provided by participants in this kind of commerce, rather than physical exchanges or direct physical contact. It also refers to the procedures, policies and strategies required to support the incorporation of electronic interaction to the business environment.

In this paper we use a clean focus on business to business frameworks for developing new e-commerce systems. We think that the new frameworks that can be designed using this approach, have the potential to solve some of the most important lacks that traditional systems reveal. We also feel that this level of indirection (focus-framework-implementation) is necessary given the complexity and diversity of current and future b2b e-commerce systems.

Keywords: E-commerce, business to business, development framework, distribution, security, traceability, extensibility, information integrity

1 Introduction

We have witnessed the introduction of a new business channel between consumers and companies. In some sense this way of doing business is revolutionary, but we can see it like a refinement of the traditional catalogue-based commerce. Of course it's not the same.

E-commerce introduces amazing concepts like information and prices always updated, commerce 24h a day and potential clients in every corner of the world, but indeed that's not that revolutionary in the way of doing business.

We believe that the promise of a real revolution in the way commerce will be fulfilled by the introduction and development of business to business (b2b) e-commerce. Its effects won't be perceived by the general public in a direct way but it will have a profound effect in the economy making it more efficient and dynamic.

B2b means the commerce established by two or more business (in most cases between business and its providers) bypassing the bureaucracy needed to effectively doing business transactions in the traditional off-line way.

B2b e-commerce can be seen as a kind of third wave of e-commerce, following the first wave that consisted only in a web site where the company offered a catalogue of its products, and a second wave where the consumer could buy those products by a link established between the web site and the company's back-end. In this third wave the company is so focused on the Internet that not only does it offer its services to its clients online but also it does business with its providers online too.

In this paper a new approach on b2b frameworks is presented. We think this new focus on frameworks can help solve the traditional wrongs in these systems as the complexity or the inflexibility of the browser – http server that is the usual support to these systems.

In the following section we describe the actual lacks of development. Next, the field of activity is presented. In next sections, some relevant concepts of the b2b field are described as well as the set of requirements we impose to the solution of the b2b commerce. Following, we

describe a generic high-level framework and then the integration of this framework with the distributed platforms is shown. Finally, our conclusions are presented.

2 Actual lack of development in the b2b commerce

There are some issues that should be addressed before we can see a real boom in the development of electronic commerce between companies. Among the most important of those issues are the **lack of accepted standards** [Ber00] and the **complexity** involved in getting the current business applications of the companies to operate in a networked multi-tiered environment.

The complexity mentioned before doesn't affect B2C e-commerce now because the entire infrastructure was developed with that kind of commerce in mind. On the other hand, the lack of well-accepted standards in the Business side of the Business to Consumer commerce is minimised by the existence of a universal client: the web browser, a program every e-commerce site can rely on his clients will have.

But it's precisely the web browser and the rigidity of the browser/http server combination what creates serious problems that make it ill-suited for b2b e-commerce.

On one hand, we lose control of the end-to-end communications, so if we want to operate in some particular way in the client-side, we'll need to use Java applets[Tho98], which still don't operate flawlessly and depend on which browser we are using. On the other hand, the web browser is a tool that depends on the user feeding it some input so it's not easy to provide the automation needed for unattended installations.

3 Field of activity

Before proceeding to describe our model of a b2b e-commerce framework, we have to bring in the context in which we think this model will be advantageous to a company.

Firstly, nowadays business to business e-commerce is more suitable for big companies. A small enterprise such as a little bookshop is unlikely to find it easy and profitable to establish an electronic link like the one proposed, with the few providers it may have.

Secondly, the companies involved in this kind of electronic commerce already have computer information and financial systems in place and are not willing to throw all the efforts and know-how away, just for another revolutionary idea that will change the way business is done.

For these two reasons, the IT departments of these companies are more willing to deploy a in-house designed b2b solution than a all-in-one e-commerce option like Commerce One Buysite, Broadvision (<http://www.broadvision.com>) or Ariba [Rob00].

4 Relevant concepts in B2B

Nowadays most of the e-commerce between business is done by using a traditional client/server architecture with a web browser and a HTTP server. This architecture has some clear advantages. It is already very popular in client to business e-commerce and it makes very easy for anyone to begin using the system as a client because web browsers are very popular applications and are available in almost any computer system.

As we mentioned before, there are important disadvantages in this architecture mainly the **lack of automation** and the **inflexibility**.

For the first problem, there are not solutions due to the interactive nature of the web browser.

The second problem, however, could be addressed by using some client-side/server-stored

programs like those developed as Java Applets or ActiveX controls. These are not perfect solutions, mainly because applets have lots of restrictions that limit the kind of operations they can do and ActiveX controls only work in win32 platforms. Furthermore, these solutions lack reliability and fail to operate in some situations.

If we want b2b to succeed, we need more control over the client side, even if that means losing the ability of having clients in virtually any internet-capable device in the world.

This is not as bad as it may seem. While having clients already deployed is something desirable, it is not as important as it is in consumer to business e-commerce.

5 Requirements of a B2B solution

There are some requirements that have to be met by any solution that pretends to deliver the promise of b2b e-commerce, some of these are outlined in:

1. **Security.** We have to make sure that no one will get into our system and disrupt the service it offers or steal valuable data.
2. **Scalability.** The system should begin offering its service to a small number of business and be capable of being upgraded to a large number of business.
3. **Robustness.** The system should be able to operate even if some of its nodes are not able to be running.
4. **Extensibility.** The system should be able to increase the number of services without disrupting the rest of the services offered.
5. **Traceability.** Business have to know when and where and transaction took place.
6. **Integrity of information.** The data contained in the nodes of the system must be preserved against malicious or no-deliberated attacks.

7. **Flexibility.** The system should be able to adapt to the ever-changing requirements of a dynamic environment.

6 What's already in place

What are companies doing to change from doing business to doing e-business?

Most of times, these companies just change from delivering orders by mail to delivering those same orders by e-mail. This change presents some advantages while at the same time being easy to adopt even for small sized companies but it doesn't unleash all the potential gains that could be achieved by a full adoption of a b2b e-commerce system.

Companies and business that are more aware of the benefits of a complete b2b e-commerce system normally integrate into their information systems pre-built packages from such vendors as Ariba Corp (<http://www.ariba.com>) or Broadvision Inc. (<http://www.broadvision.com>).

But this way of doing b2b has some important drawbacks. First, the packages are hard to customise and in most situations the company has to adapt its way of doing operations to what the b2b package is designed to do and no the other way around.

Also, by using these kind of software packages, we are linking the future of our business to that of the package vendor. These packages are so specific of the vendor, the task they perform is so important for business and they are so dependant of it's technical support that if the vendors go out of the market, or only if the quality of the support they provide is not good enough, that would surely affect the bottom line of our company operations (<http://lwn.net/2000/0518/press.phtml>).

Another important factor is that those packages don't have an accepted standard as a reference model, which means that two companies using different packages from different

vendors won't be able to inter-operate. That severely reduces the number of companies we can make business with.

7 Description of the new focus

It's important to remark that as this is just a focus on b2b frameworks and it's not intended to provide all the details needed to deliver a useful system. It's mission is to give some guidelines of how a b2b framework could or should be designed in order to comply with a series of requirements we have previously detailed. The task of doing the framework design relays on the business that wants to use it, or in third companies specialised in software developing.

So let's suppose that we have four companies (business A, B, C and D) all with their systems connected to the Internet. We can picture the situation this way:

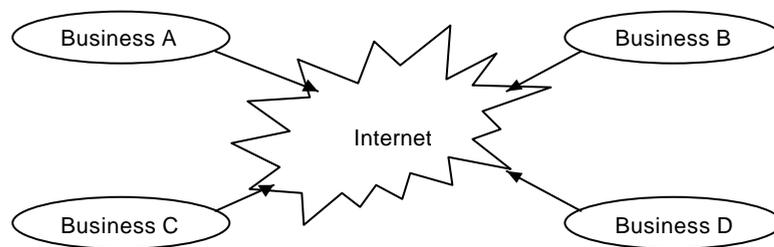


Figure 1. General Situation in the e-commerce

So far, none of these companies have a direct connection between them, they are just connected to Internet. When two of this companies (eg. A and C) use a framework based in our approach to do business, the system creates a link between them. This link is not a low level one in the sense of the ISO OSI model, we don't need to think about how to put the

information into the wire, we already have the Internet so the link will be established over it. The result is shown in the next figure.

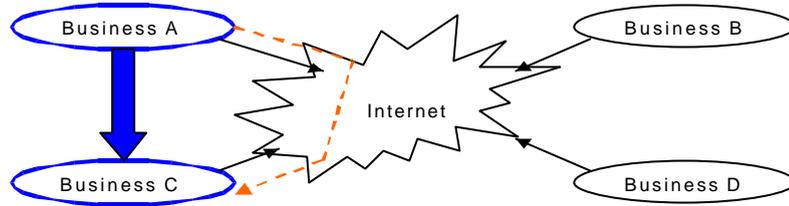


Figure 2. Modified environment with a framework

By now, only two of the four companies are able to do business under our model system. If company B wants to do business with either of these companies it just has to apply the system so a link is established with the rest of companies involved:

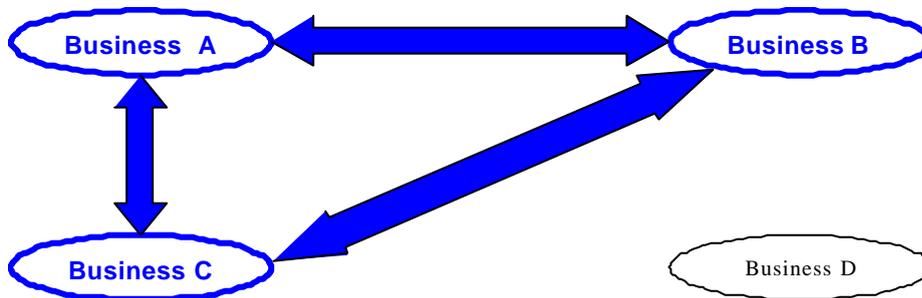


Figure 3. Environment with traditional and proposed e-commerce vias

What this picture shows is that for every two companies we need a point to point bi-directional link. Company D has not installed the system, or if it has, is no interested in using it to do business with companies A, B or C. Anyway it is connected to internet so there may be some other kind of operations that D could do with the rest of companies that is not contemplated in our system.

Obviously, if for instance company B only wanted to do business with company A and not with company C, this wouldn't be a problem as establishing a link is just the first step needed to inter operate.

For two companies to inter operate, someone in each of the two companies needs to give permissions to the computer systems of the other company. These permissions are not a everything or nothing choice, the IT staff of a company should take care of what kind of operations are available to any of the companies involved. By default, a company should be allowed to do nothing until some other company gives it any kind of permission.

7.1 General Architecture

We don't want to provide strict rules of how the computer systems of a company should be organised because every company is a different world on its own. Anyway, some sort of general rules should do no harm.

The architecture should be organised in a multi-tier fashion. When we say multi-tier in fact we mean that at least it should be 3-tier, but it could be more so as to be able to accommodate to the complexity of the company's computer infrastructure.

The first tier, the client layer, is installed in the computer systems of the rest of the companies. If a company has a client tier is only to operate with it's business partners The client is the same for all the companies in our system as it would be very complex for a company to have all the clients of every company it makes business with. This client will provide a number of features that make it more customisable and business-specific than a web browser, but as general as to accommodate the needs of different companies (that nonetheless operate in the same business sector).

The third tier is in fact the existing software that constitutes the business back-end of the company. This software is already able to support the business operations that the company does with other companies.

The middle tier should be a kind of driver that allows the first tier, which is company independent to do business transactions with the back-end of the company which is different for each company. This tier also takes care of the security (which will be explained further in this text), the transactions management and granting and controlling permissions.

We could also think about a fourth tier sitting between the middle-tier and the company's business back-end that takes care of the business rules. The number of tiers depends upon the complexity of the company and it's computer systems and business back-end.

7.2 Security

Security is a great concern in this kind of systems, even more than in the case of consumer to company e-business because of the fear of losing trade secrets, the sheer number of transactions, and the importance of those transactions.

The architecture discussed previously shows three places where security could be compromised:

7.2.1. Security within the site

The security in the own sites is quite difficult to solve completely. The reason is that the security model behind most commercial platforms is based on ACLs (Access Control Lists) and this model is basically flawed. A better idea would be to use a capabilities-based operating system in which software running on top of it is constrained to do what it says it does, based on the principle of the minimum privileges needed to do an intended operation.

Some operating systems as Amoeba or Mungi are capability-based but they are not very extended in the commercial world.

The lack of systems like that one means that we need a very competent system administrator that takes care of seeking and applying the latest security patches to avoid buffer overflows and DoS (*Denial of Service*) attacks [Neu00] and that makes sure that the system is doing all what is supposed to do, but no more than this

7.2.2. Security in the network

Our system cannot be called secure if the link between two of the sites in the system can be compromised. We can imagine an eavesdropper getting into the network and watching the flow of information as it travels over the Internet, or even changing that information with disastrous consequences.

Our proposal to address this concern is to use an implementation of the SSL v3 (Secure Sockets Layer) like the one provided by the OpenSSL project (<http://www.openssl.org>).

With SSL we can establish a secure communication between a client and a server using public key cryptography. SSL stands between the TCP layer and the applications layer allowing software systems to make it near impossible for an outsider to get information from the transactions.

7.2.3. Security in the identification of nodes

We can imagine a more disastrous situation than the one described in the previous point. What if a stranger pretends to be a legitimate partner making transactions on its behalf? . We need a method to assure that one node asking us to do some transaction is really what it pretends to be.

We will not reinvent the wheel here, we will relay in one or more nodes in our system that will act as CAs (Certification Authority). These network of machines will have a copy of the keys of every node in the system so when some client node wants to begin a transaction, it will give the AC a copy of it's public key so the AC can verify if he is what he says, then it sends the result to the server node and lets it act accordingly.

7.3 Data exchange

Most of the data of a company is stored in its back-end databases, but not all the info can be stored that way. The information stored in commercial relational databases is structured in a very defined way, but there is some kind of information like some types of transactions, reports and non-structured data that is not suited to be stored in a database.

To solve this problem, we will use XML[GTM99]. XML is a W3C (World Wide Web Consortium) standard that defines a common framework for mark-up languages. Using this framework, particular mark-up languages can be created.

To create a new language based on XML (what is called a XML application), the author has to create a DTD (Document Type Definition) describing the grammar and then making that DTD public. From that moment on, any XML capable system is able to understand data expressed in that new language.

There are a number of organisations devoted to e-commerce XML applications including Biztalk (<http://www.biztalk.com>) sponsored by Microsoft, and XML.org (<http://www.xml.org>) which includes such companies as IBM Corp and Sun Microsystems. These consortiums offer several DTDs specialized in different types of business situations, like tpaML (Trading Partner

Agreements Mark-up Language)[Gon00], that could be used in an environment that uses a framework based in our model as a reference.

8 Integration with distributed systems.

There are obvious advantages of using distributed and component-based systems concepts when developing the software needed to provide the infrastructure of the proposed system. As David Sprott says: "*... componentized products exhibit vastly superior adaptability characteristics demanded by fast moving ebusiness and e-commerce environments*" [Spr00].

At this stage we will not discuss the advantages of using a particular flavour of distributed or component architectures. We have several of them, including COM+/ActiveX, CORBA/OMA and RMI/EJB. All of them provide a common set of features like transactions management and a certain degree of transparency. So from now on we will take all these technologies as a whole.

Previously, we have described the requirements a b2b e-commerce system must fulfil. Now, we will talk about how these requirements can be eased by using distributed technologies.

At this point, whenever we talk about distributed systems, we are also referring about component technologies. The concepts are quite different, but in this case we lose almost nothing and we gain simplicity if we do that way.

8.1 Scalability

Component-based systems like those built using distributed systems technologies are easily scalable. Those systems usually have a pool of components giving service to incoming request by clients, if there's a need of more components giving service, it's easy to create more of them

in another machine. This way is easy to augment the number of transactions being processed at any given time.

8.2 Robustness

Most of today's e-commerce sites (mainly consumer-to-business but also business-to-business) use not very reliable technologies like CGI scripts. There's nothing fundamentally flawed with these way of doing things, it's just that it doesn't make it easy to use modern OO software engineering techniques.

By the own nature of software production that involves human processing, mistakes are made and bugs appear. While we don't find a more formal way of doing it, we have to resort to a number of techniques that in the past have shown to produce more reliable, reusable and maintainable software.

Component and distributed technologies provide an abstraction which facilitates to apply the results of an analysis and design made using object oriented techniques without having to worry about the underlying circumstances of a particular hardware platform or the location of some particular part of the system.

8.3 Extensibility

When a system is developed around a well-defined framework it's easy to increase it making new components that can use code and knowledge of components from this same framework[Gil00]. Reuse reduces the time and complexity to develop a new component which will provide new services or that will extend the services that are already being provided.

8.4 Flexibility

Flexibility, as defined previously, is the capability of a system to adapt to changing requirements. As distributed systems built using distributed technologies are based on the notion of objects, which are taken from its counterparts of the real world, these systems target the real requirements of the system, so when these requirements change it's easy to see and it's also easy to do these changes.

9 Future Work

So far, the purpose of our work was only to get a new insight of the world of e-commerce given our expertise in distributed systems and systems architecture. An analysis of the business requirements given by industry publications and magazines provided a foundation from which to lay the ideas for an analysis in b2b e-commerce.

This analysis still hasn't got an implementation that is necessary to prove its validity. We feel that the analysis can be useful for anyone interested in developing a framework for b2b e-commerce. Nonetheless is our intention to implement a b2b framework designed to work in a specific sector of the e-commerce. This implementation will be tested in a simulated environment and the results of this test will be incorporated to the metamodel.

10 Conclusions

There are a number of reasons that lead us to believe that the third stage of the e-commerce revolution, the b2b e-commerce, is going to be the one that will make a more profound impact in the way business is done by making it a lot of more dynamic.

But for business to embrace this change, a sort of standard must be set and security concerns must be addressed.

As for the second problem, steps are being taken to minimise its effects, but as for the first one, nothing appears on the horizon. What we have got is a set of commercial products delivered by powerful software companies that are basically incompatible between them.

In this text, we introduce a new focus on frameworks for b2b e-commerce in which the details of design, implementation and integration with legacy systems is somehow supported, but left for software companies to develop. On the other hand, a common architecture provides the ground to allow interoperability between different implementations working under this framework. Thanks to this, implementations can be adjusted to work with the specific details of certain organisations while at the same time being able to operate with the financial systems of other companies that have another implementation of these same frameworks.

11 Bibliography

- [Ber00] K. Berg-Painter. *Defining the second wave of e-business*. White paper from Nortel Networks. March 2000.
- [Coy00] F.P. Coyle. *Legacy Integration: Changing Perspectives*. IEEE Software, March/April 2000.
- [Gil00] D. Gillbrand. *Essential Business Object Design*. Communications of the ACM. February 2000.
- [NS98] Netscape Corporation. *Netscape proxy server administrator's guide*. Available via HTTP by: developer.netscape.com/docs/manuals/proxy/adminux/. Feb, 1998.
- [GTM99] R.J. Glushko, J.M. Tenenbaum, B. Meltzer. *An XML framework for agent-based e-commerce*. Communications of the ACM. March 1999.

- [Gon00] Antone Gonsalves. *IBM Pumps up XML for B2B*. ZDNet Eweek, February 2000.
- [Neu00] P.G. Neumann. *Denial of service attacks*. Communications of the ACM. April 2000.
- [Rob00] Roberts-Witt. *B2B Apps are Hot Item On Menu*. Sarah L. Interactive Week April, 2000.
- [Spr00] D. Sprott. *Componentizing the enterprise applications packages*. Communications of the ACM. April, 2000.
- [Tho98] A. Thomas. *Enterprise Java Beans: Server Component Model for the Java Platform*. Patricia Seybold Group for Sun Microsystems. December 1998.