

An XML information server for advanced B2B architectures

Jean-Jacques Dubray Ph.D.
eXcelon Corp., Burlington, USA
jjdubray@exceloncorp.com
<http://www.exceloncorp.com>

Jayaram (Jay) Valliyur
SupplierMarket.com, Inc., Burlington, USA
jvalliyur@suppliermarket.com
<http://www.suppliermarket.com>

Abstract:

Applications of XML are often limited to transient data exchange and messaging between systems across corporation boundaries. Persistent XML-based information models can greatly simplify the design, implementation, and evolution of highly connected eBusiness systems because the entities of such models can flow readily into enterprise systems, be viewed and edited by users, or participate in business partner message interchanges. These models are also key enablers to a new generation of electronic commerce architecture.

Introduction

A lot of corporations from B2B marketplaces to global-2000 companies are currently implementing a B2B infrastructure that will ultimately allow them to interact with several thousands of business partners that include suppliers, distributors, retailers, marketplaces, financial institutions, and corporate customers. The goal is to develop high-value, dynamic electronic relationships with all their business partners to support:

- Interactive and collaborative processes
- Shared planning and decision making
- Broader, richer reference information sharing
- Simple, infrequent or transient relationships
- Legally binding transactions

A significant part of the architecture is dedicated to the exchange of business documents to support interactions between business partners. These documents can be as simple as signals, or part of a long running transaction that represents a business process. XML is viewed by many as a key enabler to encode the messages and information that support all these interactions in a way that can be understood by all partners. This view is reflected by today's B2B architectures and products. Such technology implements the common services to securely exchange XML documents, and is often referred to as Internet Middleware, or XML-RPC.

However, XML itself does not guarantee that thousands of business partners will understand a particular message. Each partner has to share the same vocabulary and meaning before a message can be exchanged. Many people expect that the development of corporate vocabularies could create a translation nightmare across entire industries, forcing suppliers or buyers to map to and from dozens of proprietary syntaxes.

After the Internet and XML, B2B specifications are the third key enablers of business-to-business electronic commerce. Specifications like RosettaNet, ebXML, OAG, and commercial organization such as

CommerceOne and Ariba, define an abstraction layer on top of the Internet core standards that allows corporations to establish non-negotiated relationships with their business partners over the Internet. It would be almost impossible for any corporation, regardless of its size, to negotiate the nature of the relationships (message formats and sequences) with hundreds or thousands of partners. For instance, RosettaNet's trading partner agreement includes message formats, sequences of messages, and an implementation framework that specifies the physical means of exchanging message. Despite perceived benefits, in all these cases, the role of XML seems limited to formatting data during interchanges of business documents such as purchase orders, invoices, request for quotes, quotes and so on.

This article examines the lifecycle of inbound and outbound business messages inside a given IT infrastructure, and describes the benefits of making business information persistent as XML documents. The first part presents a typical business-to-business interchange - starting with a request for quote, quote, business process in engineer-to-order or make-to-order scenarios. The second part shows the benefits of managing the information contained in these messages as XML documents within an XML data server.

B2B integration

Let us take the example of a request for quote business process used by buyers to select suppliers. At a high level, the business process follows these steps:

Activities	Description	Documents
Start	Buyers issue a request for quote (RFQ)	RFQ
Nomination	Suppliers are selected by the buyer	RFQ
Discovery	Suppliers ask questions, buyer provide answers	Append the RFQ
Bid	Suppliers submit quotes	Quotes
Transaction	Buyers select supplier and issue purchase order, supplier acknowledges, sends an advanced shipping notice followed by an invoice	Purchase order, acknowledge purchase order, advanced shipping notice, invoice

Table 1

From the buyer, the marketplace, or the supplier point of view, a typical B2B architecture combines enterprise systems and a B2B integration server responsible for message interchanges, as shown in Figure 1. Enterprise systems of a marketplace business engine are often web-enabled or complete web applications. This allows access by business partner employees to perform self-service activities. The B2B integration server is used for activities that involve system-to-system communications.

Incoming XML messages (RFQ, quotes, and so on) go through an authentication and decryption layer. Messages are then validated against their schema and business rules. Sometimes the messages go through a translation layer in case the infrastructure has to deal with heterogeneous partners who might conform to different standards. In such a case, the incoming XML document is transformed into an internal format. At

this point, the XML document is ready to be parsed and consumed by the enterprise systems such as ERP or eMarket business engines. The use of an intermediate internal format avoids the "many B2B formats to many enterprise systems" problem .

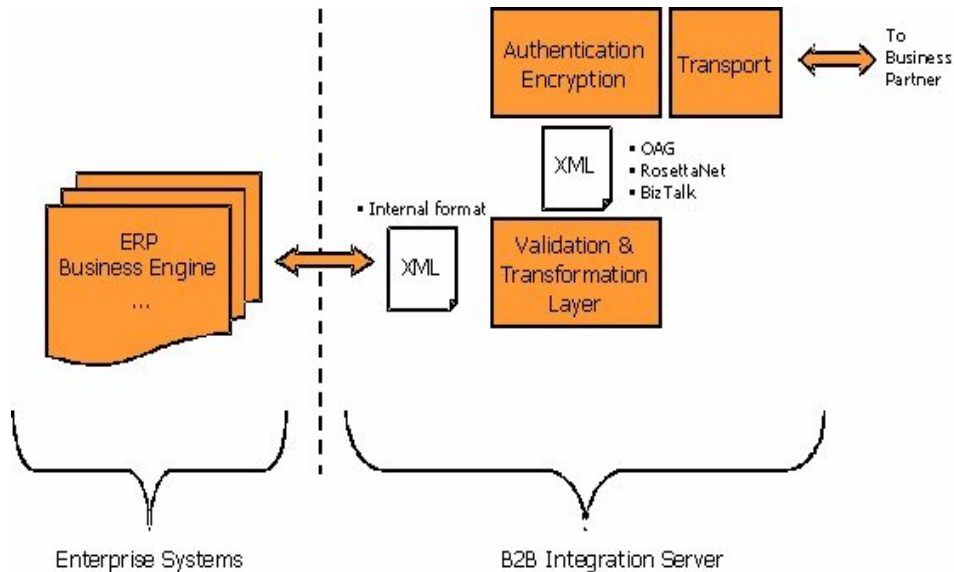


Figure 1. Typical B2B architecture

One issue with this type of architecture is that the information carried by various B2B-standard XML messages might not fit readily into your information systems. It is likely that there will be extraneous elements, parts of a transaction that must be persistent separately because they participate in subsequent message interchange. The B2B documents might also be needed for auditing and non-repudiation purposes.

Similarly, information might be missing from the document to continue its lifecycle within the enterprise systems or at a business partner system. These properties might be captured manually via a user interface or completed automatically by a system or a business component.

Another issue exists when there are some manual steps needed to further validate or complete incoming or outgoing documents. If the XML document is a purchase order and the enterprise system is an order entry system, you might want to introduce manual steps that could be used if the incoming (or outgoing) order exceeds a certain threshold (For example, more than a 1000 parts). If the elements of a document are stored in a relational store, it is often difficult to append the content with context information that may vary extensively based on the properties of the document itself, the business unit currently handling this document or the workflow in which this document participates. It becomes rapidly difficult to maintain a clean audit trail of what happened to this document within the organization or when the document participates in B2B interchanges. The level of complexity increases further if the audit trail involve digital signatures.

An information model optimized for data exchange

The information that participates in business message interchange is expected to interact with a handful of external and internal systems. The catch, however, is that external systems could be one of many thousand different partner systems. For large corporations, the number of internal systems might also be large. Furthermore, some of these documents must be reviewed or edited by various employees interested in different parts of the documents. As a consequence, B2B architectures are required to optimize business

data models both for the exchange of data, and for capturing the context of increasingly varied business transactions.

Let us examine a new kind of data server - an XML Information Server - within the architecture of a B2B integration server where business objects persist as well-formed XML documents. The purpose of this server is to manage business documents (purchase orders, invoices, and so on), profiles (customers, partners, employees, systems and so on) and reference information (such as product description).

Because well-formed XML documents contain both data structure (metadata) and data, they can be semantically accessible without the need for an interface. You can use relative XPath predicates to access the content of any element without a complete knowledge of document structure. You can build a transformation engine to transform a given XML document to the specification of an information consumer - also known as service - without any interface associated to the XML business object. We define services as units of application logic that accept XML documents as input and either update these documents, or generate new documents as output. Categories of services include:

- B2B services hosted at partner systems, often compliant with a B2B standard such as RosettaNet
- Enterprise Application Integration services
- User Interface services
- Component based services (COM, EJB, CORBA)

In addition, unlike valid documents with respect to a schema, well-formed XML documents are extensible, that is, data structure and data can be added without compromising the integrity of the initial document. This approach offers several benefits. One of the most innovative is the ability to evolve a business object instance. Unlike object-oriented or component-based implementations, XML instances can evolve from one schema to another or even hold structured data they were not specifically designed to hold, and can share any part of it. This makes them both flexible and adaptive.

Semantic access and extensibility are critical for enabling the true separation of the application logic wrapped behind services from the information model. These two characteristics are also critical when you are developing information models based on the concept of a business document and managing information as a lifecycle. XML business objects avoid integration problems since services can embed traces of interactions within the object itself (as shown in Figure 2) without breaking the relationship between the document and the other services. In a traditional model, the service would have to keep this information in a private store. In the event that another service wanted to access not only the information contained in the business object, but this particular trace, the services would have to be integrated. While this scenario is feasible in a departmental infrastructure, it would be virtually impossible to integrate with thousands of business partner systems that could potentially require access to this information. With an XML information server, other services (within or outside the enterprise) can access this information, as long as they share the semantics as opposed to just being aware of a specific interface.

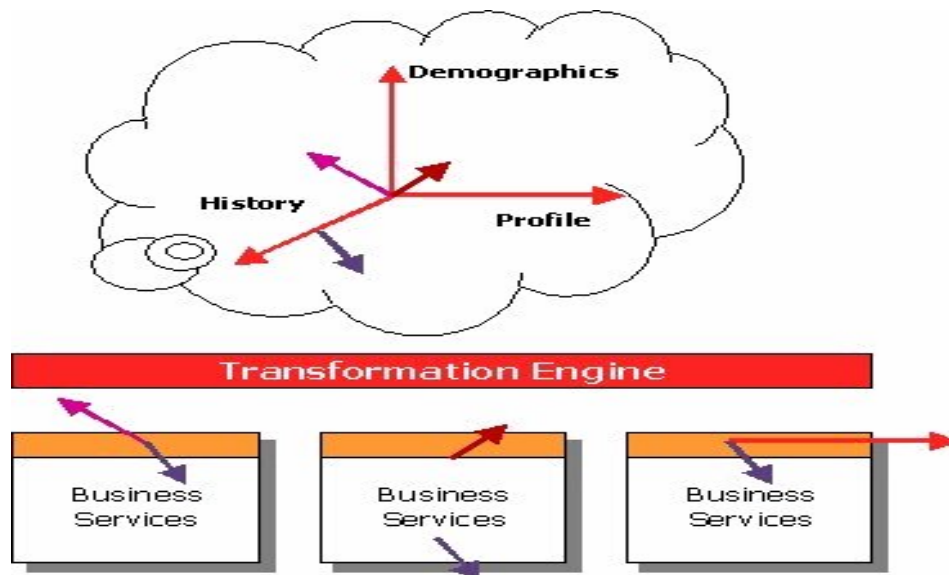


Figure 2. XML-based information model

An XML information server within a B2B architecture

The use of an XML information server between a B2B gateway and the enterprise systems as shown in Figure 3, has some major design advantages.

XML business objects can be associated with an XSL stylesheet, thereby transforming the user into a message receiver and message sender. You can develop intranet or extranet self-service activities simply by adding a stylesheet to an existing document.

State-captured XML business objects can be augmented, as they interact with various enterprise, B2B, or user interface services. You can design the internal format from an existing B2B standard by extending it to capture information about the internal business processes. You can transform the same document back into a RosettaNet-compliant document by trimming these extensions.

The internal format managed by the XML information server facilitates the interaction with business partners who are compliant with different B2B standards. Documents can be transformed just-in-time, once the identity of the partner is validated. For that matter, the XML information server can even support partners that can only accept flat-file data formats.

Elements of business documents that cannot fit in enterprise systems are made natively persistent.

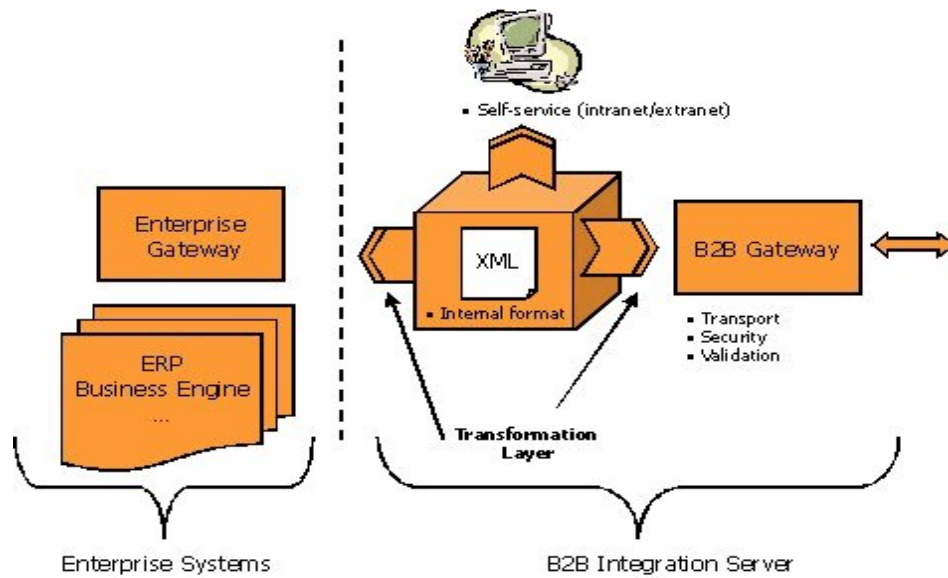


Figure 3. B2B integration server with an XML information server

Moreover, the information server and the concept of information entities that behave like business documents help divide the business logic into three very distinct categories:

Methods (such as data access methods) provide an encapsulation layer on top of XML documents. This layer is implemented as part of the transformation engine.

Services represent units of work.

Business processes specify the sequence and definition of service invocations and business documents.

When the number of business processes and their respective instance is become large, a workflow engine can easily be added into the architecture to manage thousands of concurrent business processes in a very dynamic environment where partners are added, removed or changed daily.

It is difficult to estimate the performance penalty imposed by the introduction of such information server versus a relational store from which XML documents are assembled on demand. The more complex the document structure becomes, the more advantageous it is to keep it assembled. It is also difficult to assert the impact of using digital signatures. Should performance be an issue, the mitigation factor is that most B2B interactions are handled asynchronously. In this case sub-second responses are hardly necessary.

Conclusion

The concept of an XML Information Server with a transformation layer ensures the smallest coupling between the application logic and the information model. It enables the data model to interact with a large number of systems and their services, while limiting the need for integrating these systems together. It also simplifies greatly the development of self-service applications that are required to allow internal and external employees to review or edit business objects.

Bibliography

- [1] R. Worden " XML E-Business Standards: promises and pitfalls," <http://www.XML.com> ,
January 5 2000.
- [2] The RosettaNet consortium, <http://www.rosettanet.org>
- [3] The ebXML consortium, UN/CEFACT, OASIS, <http://www.ebxml.org>
- [4] The Open Applications Group consortium, <http://www.openapplications.org>
- [5] SupplierMarket.com, Inc., <http://www.suppliermarket.com>
- [6] T. Berners-Lee et al "Web Architecture: Describing and Exchanging Data" W3C Note, 7
June 1999
- [7] J. Clark et al "XML Path Language (XPath)," Version 1.0, W3C Recommendation 16
November 1999
- [8] J.J. Dubray et al "Business Object Modeling: an XML-based approach," Accepted for
publication in the Journal of Markup Languages: Theory & Practice, 1999
- [9] J.J. Dubray "An eBusiness Solution Framework," XML 99 conference, Philadelphia,
December 1999

Authors

Jean-Jacques Dubray Ph.D.

Chief-Architect
eXcelon Corp.
Postal Address:
25 Mall Rd
01803 Burlington
MA
USA
Telephone: 781-402-7220
Fax: 781-402-7320
E-mail: jjdubray@exceloncorp.com
Web: www.exceloncorp.com

Jean-Jacques Dubray, Ph.D. — Jean-Jacques Dubray has nine years of experience in building large scale distributed object-oriented systems for the semiconductor and telecommunication industries, and more recently for business intelligence and e-Commerce applications. He is currently chief architect at eXcelon Corp. He is also the project leader of the eCommerce initiative of the Open Applications Group.

Jayaram (Jay) Valliyur

Director of Engineering
SupplierMarket.com, Inc.
Postal Address:
10 Mall Rd
01803 Burlington
MA
USA
Telephone: 781-273-6744
Fax: 781-273-6803
E-mail: jvalliyur@suppliermarket.com
Web: www.suppliermarket.com

Jayaram (Jay) Valliyur — Jayaram (Jay) Valliyur has over twelve years of experience in building large scale distributed systems in the areas of retail banking and marketing automation, and more recently for e-marketplaces. He is currently Director of Engineering at Suppliermarket.com - a leading online marketplace for manufactured direct materials. He is responsible for building the XML based B2B gateway for the marketplace.