A Survey and Comparison of Business-to-Business E-Commerce Frameworks

Asuman Dogac
and
Ibrahim Cingil
Sofware Research and Development Center (SRDC)
Dept. of Computer Engineering
Middle East Technical University (METU)
Ankara, Turkey
(asuman,ibrahim)@srdc.metu.edu.tr

Business-to-business (B2B) e-commerce is booming thanks to the ubiquity of the Internet, which dwarfs the coverage of EDI VANs. B2B e-commerce needs to seamlessly and dynamically handle the interactions among a vast variety of organizations without ad hoc and proprietary integrations. A number of B2B e-commerce frameworks have emerged as a result. In this paper we present a brief survey and a comparison of some of these frameworks.

1. INTRODUCTION

International Data Corp. (IDC) forecasts worldwide B2B e-commerce revenues to be of USD 2.2 trillion by 2004 (http://www.idc.com/eBusiness/press/EBIZ082200 pr.stm). The Gartner Group predicts that e-marketplaces will handle 2.71 trillion Dollars by 2004 and that this will account for 37 % of all B2B e-commerce (http://www.gartner.com/public /static/aboutgg/pressrel/ pr20001004b.html).

However the lack of standard business interfaces puts a huge burden on business participants ultimately creating inefficiencies and inhibiting the ability to leverage the Internet as a business-to-business commerce tool. That is, the interoperability has become the key issue where operations across business boundaries necessitate industry-wide agreements on common ontology, formats, content, and system interfaces.

These needs are being addressed by International consortiums and quite a number of B2B interoperability frameworks have emerged, some of which are eCo framework (http://eco.commerce.net), RosettaNet (http://www.rosettanet.org), Commerce XML (http://www.cxml.org), and BizTalk (http://www.microsoft.com/Biz

talk). We also cite MESChain (http://www.srdc.metu.edu.tr/MESChain/) which additionally adresses how to handle meta-data in B2B e-commerce.

All of these frameworks are document-centric and based on XML. It seems the alternative, which can be an interface-centric framework like CORBA IDL, is not preferred because most information exchanged in e-business is declarative in nature and not procedural. Also, although there are many distributed programming paradigms that can be used to implement a distributed architecture over the Internet (e.g. HTTP with SSL, CORBA, DCOM and RMI), all of the frameworks support HTTP since it builds upon a prolific base of WWW servers and browsers. These systems with open, extensible formats capture the essentials of business processes in dynamic virtual organizations while allowing flexible internal implementations. It is clear that by making the technology available over the Internet in this way in a cost-effective manner compared to EDI applications, creates an advantage especially for small and medium enterprises. This in turn can effectively enable mass-market B2B interactions and trading, increasing alternatives and lowering costs for all involved.

The paper is organized as follows: Section 2 describes eCo framework, Section 3 briefly introduces RosettaNet and Section 4 summarizes Microsoft's BizTalk initiative. Section 5 briefly discusses Ariba's cXML. Section 6 covers a work on supply chain integration and automation. Section 7 presents a comparison of the frameworks described. Finally Section 8 concludes the paper.

2. ECO FRAMEWORK

CommerceNet's eCo architecture (http://eco.commerce.net) provides an interoperability framework. In the eCo architecture, businesses agree on a common method of describing what they do rather than agreeing on standards of what they do and how they do it. The eCo Framework consists of an architectural specification and a semantic specification. The Architectural Specification presents information about an e-commerce system in seven different categories (layers) as shown in Figure 1 where "networks" (layer 1) contain "markets" (layer 2) where "businesses" (layer 3) provide and use "services" (layer 4) which conduct "interactions" (layer 5) that exchange "documents" (layer 6) containing "information items" (layer 7). The eCo Semantic Specification, on the other hand, provides a sample set of business documents that can be used inside the eCo framework. These can be used as is, or extended and modified to meet specific needs. The network layer contains various eCo compliant markets for providing or obtaining specific goods and services like computers, phones, or books. In the market layer, for a specific market like computers, their participating businesses are listed like Dell, or IBM. In the business layer, the services provided by a business are listed, for example catalog browsing, ordering products, making payment, or checking order status. At the service layer, the possible interactions of the services are listed in terms of input and output documents and an optional execution URI.

A service may invoke other services in order to complete that service. These relationships among services are described in the interactions layer. In other words, this layer describes a "choreography" of interactions that may take place when a service is invoked and the types of messages which are exchanged during each interaction. The document types exchanged in an interaction are described by the

document layer which lists its data elements, if any. For example, an interaction for purchasing a computer can take an order document as input, and produce an invoice as output defined in XML. At the data element layer, details of data elements are presented. One of the recommendations to be used at the document layer and the data element layer of the eCo architecture is Common Business Library (CBL) (http://www.commerceone.com/xml/ cbl/index.html). CBL contains a set of building blocks common to many business domains such as address (location.dtd), price (value.dtd), purchase order (order.dtd) and standard measurements (measures.dtd), and thus provides the much needed basis to ensure interoperability among XML applications. When this is complemented by a set of DTDs common for specific industries, i.e. for vertical domains, the open electronic commerce infrastructure will be achieved. As a summary each layer of an eCo-compliant e-commerce system presents information about itself. By examining this information, the users can:

—locate the system

—understand what it is for

—identify protocols the system uses to communicate

—discover what documents the system uses to conduct business

—learn how to interoperate with the system

This information is provided through published interfaces for each layer that involve specific sets of queries that must be implemented for a system to be eCo compliant. These queries return documents (e.g. property sheets) describing each layer.

A brief description of the interfaces and the documents returned is given in the following:

Network Layer. Provides a way to categorise and locate markets or communities of commerce on a network. The layer consists of a set of interfaces that can be used to find markets and relate those markets to a set of defined market types. The published interface defines the following standard queries that can be requested from a specific Network:

— **NetworkGetProperties** (returns a set of properties that describe a network),

— **NetworkGetMarkets** (this query returns a list of markets that participate in this network),

—**NetworkGetMarketByType** (a list of markets is returned filtered by type).

The document types, that is, **NetworkPropertySheet, EcoInterfaceList, EcoTypeList, etc.** are returned in response to these queries, respectively.

Market Layer. Allows a set of businesses to group together through a common portal or access point. By participating in an eCo Market Environment, a business can offer its goods or services in a shared context with other related businesses. The eCo Market Layer is conceptually similar to the physical markets in which businesses often group together. The published interface to a Market defines a set of queries that can be requested of that Market. The standard queries include:

— **MarketGetProperties**,

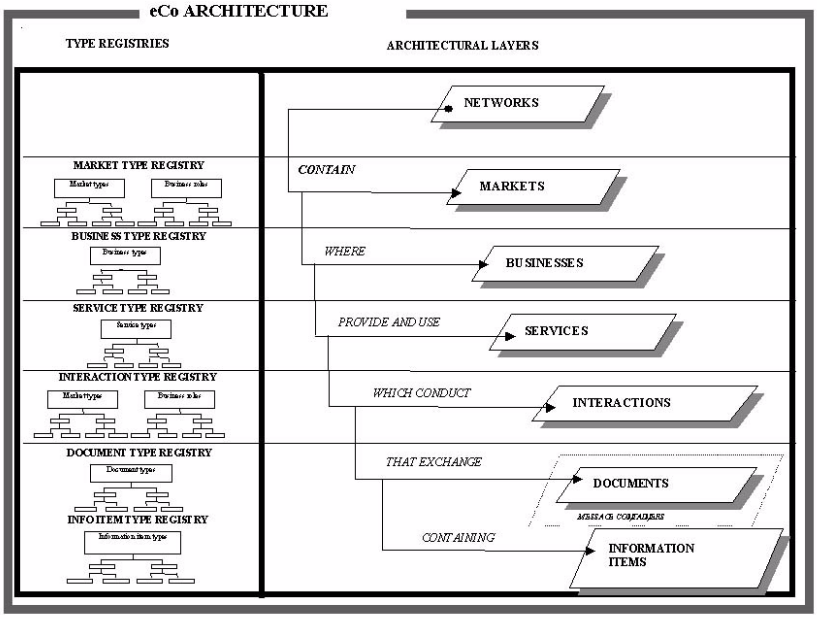

Fig. 1. eCo arcitecture

— **MarketGetBusinesses** and

—**MarketGetBusinessTypes**.

MarketPropertySheet, EcoInterfaceList, EcoTypeList, etc. are returned in response to these queries, respectively.

Business Layer. Provides identifying information about members of a trading community and lists the services they offer or consume. This layer can represent a single individual, a business, or a business unit of a larger organisation. The published interface to a business defines a set of queries that can be requested of that business. The standard queries include **BusinessGetProperties, BusinessGetServices,** etc. **BusinessPropertySheet, EcoInterfaceList, EcoTypeList, etc.** that returned in response to these queries, respectively.

Service Layer. Represents that part of an e-commerce system that is responsible for providing meta- data about business services and the exchange of commercial documents. "Services" in the eCo Specification are defined as interfaces to a business process. Each Service offered by a business provides the ability for a trading partner to interact with that business in some way. The published interface to a service defines a set of queries that can be requested of that service. The standard queries include, **ServiceGetProperties, ServiceGetInteractions,** etc. **ServicePropertySheet, EcoInterfaceList, EcoTypeList, etc.** are returned in response to these queries.

Interaction Layer. Interactions represent the fundamental building blocks from

which Services are built. The published interface to an interaction defines a set of queries that can be requested of that interaction. The standard queries include, **InteractionGetProperties, InteractionGetServices,** etc. **InteractionPropertySheet and EcoInterfaceList**, are returned in response to these queries.

Document Layer. Documents represent a physical encapsulation of information items. Documents are in the form of XML, with Document Type Definitions (DTDs) and XML Schemas (DTDs as instances with other information) according to the W3C recommendations. The published interface to a document defines a set of queries that can be requested of that document. The standard queries include, **DocumentGetProperties, DocumentGetDataElements,** etc. **DocumentPropertySheet, EcoInterfaceList, EcoTypeList, etc.** are returned in response to these queries.

Data Element Layer. As the fundamental mark-up elements within a document, data elements serve to encapsulate data or code to be used in a particular interaction. The published interface to a data element defines a set of queries that can be requested of that data element. The standard queries include, **DataElementGetQueries, DataElementGetSubDataElement,** etc. **DataElementPropertySheet, EcoInterfaceList, EcoTypeList, etc.** are returned in response to these queries. It should be noted that Data Element layer defines smaller chunks of information like date or price to be used by the document layer.

3. ROSETTANET FRAMEWORK

Founded in 1998, RosettaNet is an independent, self-funded, non-profit consortium dedicated to the development of XML-based standard electronic commerce interfaces to align the processes between IT supply chain partners on a global basis. The RosettaNet consortium includes IT companies like IBM, Microsoft, EDS, Netscape, Oracle, SAP, Cisco systems, Compaq and Intel which represents 400 billion USD out of the industry's 700 billion USD.

RosettaNet Framework (http://www.rosettanet.org) consists of Partner Interface Processes (PIPs), a master dictionary and an implementation framework, the relationship among which can be expressed with the following analogies: RosettaNet dictionaries provide the words, the RosettaNet Implementation Framework (RNIF) acts as the grammar and RosettaNet Partner Interface Processes (PIP) form the dialog as shown in Figure 2. This master dictionary, coupled with an established implementation framework for exchange protocols, is used to support the eBusiness dialog known as the PIP.

The RosettaNet model enables supply chain business partners to execute interoperable e-business processes by developing, maintaining and distributing partner interface process implementation guidelines. RosettaNet dictionaries provide a common platform for conducting business within the trading network. The RosettaNet Business Dictionary designates the properties for defining business transactions between trading partners, and the RosettaNet Technical Dictionaries provide properties for defining products and services. RosettaNet distributes PIPs to the trading partners, who use these guidelines as a road map to develop their software applications.

The RNIF is an open, common networked-application which defines the protocols for exchanging messages for quick and efficient implementation of RosettaNet

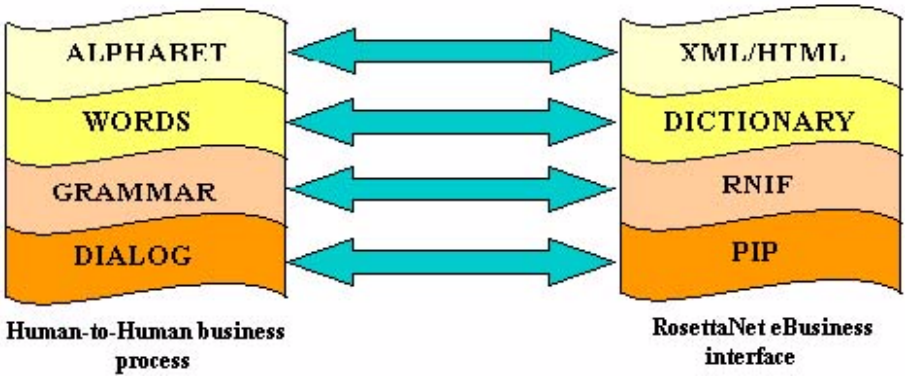

Fig. 2. RosettaNet eBusiness Interface

standards so that RosettaNet Supply Chain and Solution Partners can collaboratively execute RosettaNet PIPs. It specifies message format, message content, network architecture and security mechanism. The PIP specification model enables RosettaNet to specify partner-to-partner electronic business processes in terms of "actions," "transactions" and "execution processes". The implementation framework specification enables RosettaNet partners and solution providers to create networked applications that can execute these electronic business processes by communicating according to strictly defined protocols. These protocols specify application message formats and message exchange sequences. Also, this specification includes authentication, authorization, encryption and non-repudiation implementation aspects that are necessary for conducting secure electronic business over the Internet. To implement the Technical Dictionary, an organization must categorize all saleable products according to the product classes and class properties outlined in the Technical Dictionary. The RosettaNet IT Technical Dictionary can then be used for searching electronic catalogs and for maintaining technical information databases. For example RosettaNet PIP2A5, "Query Technical Information," describes how to query an online product.

As of October 2000, RosettaNet members delivered real-world examples of how standards are facilitating supply-chain collaboration. Companies such as Arrow Electronics Inc., Intel Corp., Lucent Technologies Inc., reported that they are using RosettaNet standards associated with acknowledging a receipt of a purchase order and providing order status, advance shipment notices and new product information. Furthermore first wireless implementation of RosettaNet PIPs are available from Epic Data International Inc. which will make it possible for mobile users to instantly receive event notifications and quickly respond to exception situations, like checking product availability, obtaining price quotes, or monitoring trading partner requests.

4. BIZTALK

BizTalk provides an XML based application integration framework. Application integration is achieved through a message passing mechanism. For applications that do not have native XML support, BizTalk framework provides facilities that

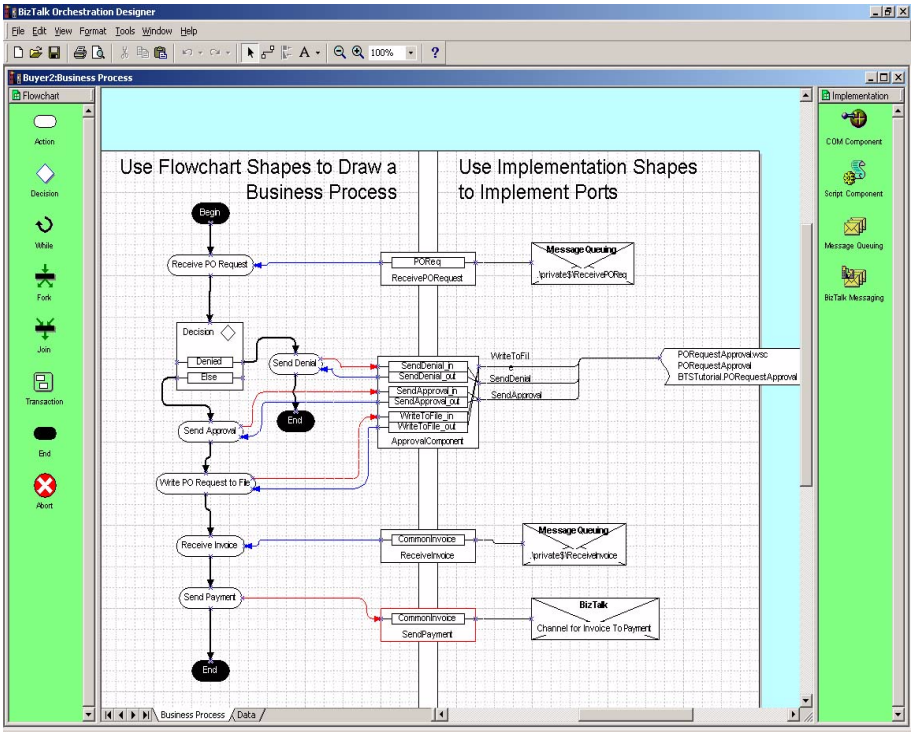

Fig. 3. Visio:BizTalk 2000 Server Orchestration service's GUI

transform native data types to XML and then performs the XML document routing.

The BizTalk framework contains a technical specification, a set of XML documents, the http://www.biztalk.org/BizTalk/default.asp Web portal and BizTalk 2000 Server.

The biztalk.org Web portal contain the XML schemas suggested by the businesses, that is,it is a globally accessible document library. These schemas are validated, versioned, registered and stored in a repository at the Web portal Web site. In other words, BizTalk does not enforce a common dictionary; rather the businesses make their dictionary and the document structures publicly available at the BizTalk portal and expect other businesses to conform to them for conducting business. One BizTalk-driven online service is eShop, a shopping portal accessible at http://eShop.msn.com. eShop allows consumers to find and compare products in more than 100 product categories, then find out where to purchase those products either online or in retail stores. Any merchant who wants to offer his goods and services on eShop online can download "eShop Offers" schema from BizTalk.org and built a new application which would allow him to automate sending electronic offers to eShop to be presented to consumers online.

The Microsoft product BizTalk 2000 Server has two services: Orchestration service and Messaging service. BizTalk 2000 Server Orchestration service allows the users to model, deploy and maintain distributed business processes within and be-

tween organizations. It provides a GUI tool based on Visio, to graphically build business processes as shown in the Figure 3. It is possible to integrate previously defined distributed, heterogeneous applications to the process. From the visual description of the process, a process definition is generated in XLANG which is a textual process definition language in XML. XLANG provides the classical workflow definition features like sequencing, branching, parallel execution as well as transactional support, persistence and monitoring. XLANG is a message based system, that is, the activity invocation is through passing messages. The message reliability is provided by the message queue product of Microsoft, MSMQ. BizTalk 2000 Server Messaging service support XML, X12, EDIFACT formats. BizTalk mapper helps to map one document format to the other and can apply transformations to data. For example BizTalk mapper facilitates transforming EDI documents to SAP iDOC documents by utilizing XSL technology. BizTalk Management Desk administers agreements between trading partners and applications. Content based routing of the documents is also possible.

5. COMMERCE XML

Commerce XML (cXML) (http://www.cxml.org) is a standard for facilitating exchange of catalog content and transaction information between trading partners and proposed as a joint effort of more than forty companies spearheaded by Ariba Inc.

The cXML catalog definitions consist of three main elements:

—*Supplier:* Contains basic data about the supplier, such as address, contact, and ordering information.

—*Index:* Describes data about the supplier's inventory of goods and services, such as description, part numbers, and classification codes. Index uses several subelements to describe line items in supplier's inventories.

—*Contract:* Describes data about flexible aspects of the inventory negotiated between the buyer and supplier, such as price.

Seller organizations create catalogs so that buying organizations' procurement applications can see their products and service offerings. Procurement applications read these catalogs and store them internally in their databases. After buyer organization approves the seller organization's catalog, the content becomes available to the users. In other words, procurement applications allow communities of users to buy contract products and services from vendors approved by their purchasing managers. Requested purchases are first approved by managers in the communities, and approved purchase orders are transmitted to suppliers.

In addition to these static catalogs, cXML allows for dynamic catalogs called, PunchOut Catalogs, which are Web server applications coded in a programming language like ASP, JavaScript, or CGI to manage buyers' PunchOut sessions. In this way, buying organizations can either store content locally within the organization, or they can access it remotely on the Internet, through PunchOut catalogs.

For PunchOut sites, procurement applications display a button instead of product or pricing details. When users click this button, their Web browsers display pages from seller organization's Website. Depending on how the these pages are implemented, users can browse product options, specify configurations, and select

delivery methods. When users are done selecting items, they click a button that returns the order information to the procurement application. The fully configured products and their prices appear within users' purchase requisitions. Seller organization's Website can offer previously agreed-upon contract products and prices. When this step is completed buying organizations send purchase orders to suppliers to request fulfillment of a contract. Order-receiving systems are applications at supplier sites that accept and process purchase orders sent by buying organizations. Order-receiving systems can be any automated system, such as inventory management systems, order-fulfillment systems, or order-processing systems. Because it is simple to extract information from cXML purchase orders, it is relatively easy to create the adapters that enable existing order-receiving systems to accept them.

There are two communications models for cXML transactions: Request-Response, and One- Way. In the Request-Response mode, the requestor creates a cXML message and sends it to another domain over an HTTP connection. The receiver, upon request, translates the message, parses the content of the request, understands the context of the request, and invokes the corresponding back-end system. The back-end system, serves the request and delivers the results to the receiver. In the sequel, the receiver formulates the results in terms of a cXML message and sends it back to original sender of the message.

In the One-Way model, the sender sends a cXML message to a receiver by describing the type of message. The receiver, on the other business domain, parses and understands the message, and invokes the corresponding service on a back-end system. However, no response is generated and sent back to the sender.

6. MESCHAIN

MESChain (http://www.srdc.metu.edu.tr/MESChain/) is an XML-based and message oriented supply chain architecture developed at METU. An overview of system architecture is shown in Figure 4. MESChain uses the CBL's catalog DTD as the canonical data model for catalog integration and allows participants either to maintain their catalogs directly in XML conforming to CBL or keep them in legacy applications. Legacy application support is provided by introducing an <?EXECUTE ... > instruction (http://www.acm.org/sigecom/exchanges/issue-1.1/volume-1.1.html)
which makes it possible to specify in an XML document where to get external data and how to integrate it into the document. In this way, XML documents dynamically generated from a number of possibly heterogeneous resources can be flexibly integrated.

Each catalog is associated with a catalog agent which handles XML-QL or XSLT queries against the catalog. Producing customized catalogs is possible through integrating the results of queries. A component based workflow system is provided to automate the business processes down the supply chain. When the catalog agent receives a customer order, it automatically starts a customer order handling process. The catalog agent is capable of differentiating the messages according to different purchase protocols like OBI or OTP and thus invoking different workflow processes correspondingly. MESChain provides facilities to monitor the status of a customer order over the entire chain. In MESChain, the taxonomy of the products and an "Added_Value" property are expressed in Resource Description Framework

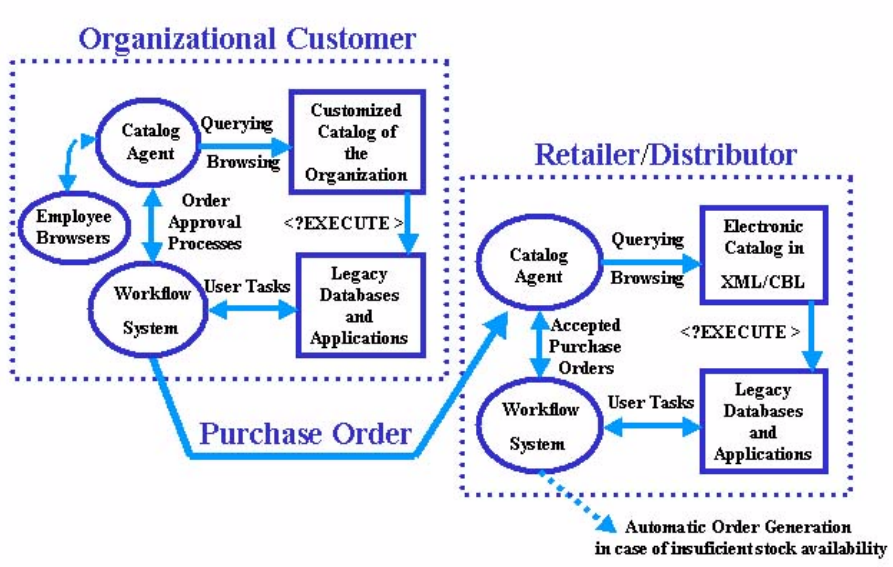

Fig. 4. General Arcitectur of MESChain

(RDF) (http://www.w3.org/RDF) which facilitate finding complementary and/or add-on products for a given anchor product. The relationships among the catalogs on the supply chain, in other words participant roles, are specified through the "up_link" and "down_link" descriptions in RDF which facilitate the bi-directional catalog search and integration.

MESChain provides automated inventory control by associating a stock danger level with each item in stock. When an availability of an item falls below the danger level, an order is automatically generated which realizes the automation of the business processes down the supply chain recursively.

On-the-fly supply chain integration is greatly facilitated in MESChain. Once a buyer search agent locates an electronic catalog of a potential supplier, it informs its own catalog agent which in turn contacts and negotiates with the catalog agent located. If a deal is reached, each catalog agent starts an approval process to evaluate the new candidate partner for its own company. If both parties approve each other as 'new partners', then the catalog agents exchange messages to automatically insert "up_link" and "down_link" property values pointing to each other in the RDF descriptions of the catalogs. They also exchange the necessary CBL data such as market participant information.

7. A COMPARISON OF THE FRAMEWORKS

In Figure 5, a comparison of the discussed frameworks is presented. The first issue we investigate is the level of implementation. Although particular implementations exist, eCo is a specification. RosettaNet and cXML provide implementation guidelines whereas BizTalk framework includes commercial off-the-shelf products like BizTalk 2000 Server and MSMQ to realize the framework functionality.

Features	eCo Framework	RosettaNet	BizTalk	cXML	MESChain
Purpose	Development of standard service descriptions in XML	Development of XML based standard electronic commerce interfaces	XML based application integration framework	XML based standard to exchange catalog content and transaction information	Automating supply chain integration on the Internet based on XML
Specification/ Implementation /COTS	Specification	Implementation Guidelines	Commercial off-the-shelf product set	Implementation Guidelines	Implementation
Product taxonomy Support	It is possible to define product taxonomies in type registries	RosettaNet is developing product descriptions for IT industry	Not addressed	Not addressed	Product taxonomy and value added products are described through RDF
Catalog support	Catalog support comes indirectly from using CBL at the document level	Provides catalog functions by defining PIPs for new product introductions, updates, etc.	Not addressed	Support for both static and dynamic catalogs (punchout catalogs)	Supports catalog integration and customization
Service discovery	Provide extensive querying interfaces to the documents in the market place	Not addressed	Provided through Server Orchestration service's Visio tool	Not addressed	Not addressed
Document conversion	Not addressed	Not addressed	BizTalk mapper helps to map one document format to the other and can apply transformations to data	Not addressed	Not addressed
Automated business process support	A specification for business process automation is mentioned but not specified	Although RosettaNet allows to define business processes; there is no tool to automate them	BizTalk 2000 Server Orchestration service provides business process automation support	Not addressed	The WFMS and the catalog agent provided allows for business process automation

Fig. 5. Comparison of B2B Frameworks

Electronic catalogs are indirectly supported in eCo framework since it conforms to CBL's catalog DTD at the document level. RosettaNet provides catalog functions by defining related PIPs (like PIP2A5 "Query Technical Information") and there is support both for static and dynamic catalogs in cXML. BizTalk does not address this issue; MESChain provides catalog integration by conforming to CBL catalog DTD as the canonical data model and through an RPC mechanism imbedded in XML. MESChain also provides customized catalog support through standardized queries. To provide catalog integration, it is important to also have product taxonomy. In eCo it is possible to specify product taxonomies in types registries; RosettaNet is already providing product descriptions for the IT industry an essen-

tial part of which is a product taxonomy and MESChain uses RDF in this respect. In BizTalk and cXML this issue is not addressed.

Forming business partnership on the fly where participants can transact business spontaneously is a very important part of B2B e-commerce. This necessitates the discovery of businesses, services and related documents on the fly and this issue is addressed in eCo through its querying interfaces. MESchain provides facilities for this through its search and catalog agents. This issue is not addressed in the other frameworks.

Another important issue is providing support for business process integration. Service composition requires the need to organize the order of service invocation, to manage data and data transfer between services. It should be noted that execution of composite services typically span organizational boundaries and the capability of interacting with heterogeneous applications is necessary. Organizing interactions into services is mentioned in eCo but workflow specification is left open. There is no off-the-shelf tool to automate business processes in RosettaNet. BizTalk provides a GUI tool based on Visio to graphically build business processes. MESChain provides a workflow management system and a catalog agent to automate business processes down the supply chain.

Finally, it may not be realistic to expect all the documents exchanged to conform to the same standard, and mapping documents from one format to the other is a useful functionality. BizTalk is unique among the framework described; BizTalk mapper helps to map one document format to the other and can apply transformations to data.

8. CONCLUSIONS

B2B e-commerce frameworks isolate the back-end systems of the trading partners from the external world. In this way, a trading partner does not have to implement as many formats as there are back-end systems at its trading partner's site.

The B2B frameworks summarized in this article are in no way exhaustive: there are other frameworks like ebXML (http://www.ebxml.org/). Furthermore there are very many B2B e-commerce standards addressing different aspects of e-commerce. For example there are transport binding standards like SOAP (http://www.w3.org/TR/2000/NOTE-SOAP-20000508/), Web service description standards like WSDL (http://www.w3.org/TR/wsdl) or e-service publishing and discovery standards like UDDI (http://uddi.org/).

These standards sometimes overlap or complement one another. As an example, SOAP complements WSDL, since WSDL allows for transport bindings to be specified and this can be SOAP. OAGIS (http://www.openapplications.org/) can be said to complement BizTalk with the OAGIS as the business content and BizTalk as a part of the underlying messaging infrastructure.

However e-commerce standards compete most of the time and this creates the following problem: unless all trading parties conform to the same standard there is still a need to develop interoperability mechanisms among these standards. That is B2B integration engines should be develop to make it possible to dynamically add standards over time required by trading partners being added.