

# Ontology and XML-based Specifications for Collaborative B2B Relationships

Ma. Laura Caliusco  
Gidsatd - UTN - FRSF  
Lavaise 610 – Santa Fe  
Santa Fe - Argentina  
+54 0342 4608585 – Int 222

[mcaliusc@frsf.utn.edu.ar](mailto:mcaliusc@frsf.utn.edu.ar)

Ma. Rosa Galli  
INGAR-CONICET  
Avellaneda 3657  
Santa Fe - Argentina  
+54 0342 4534451

[mrgalli@ceride.gov.ar](mailto:mrgalli@ceride.gov.ar)

Omar Chiotti  
INGAR-CONICET  
Avellaneda 3657  
Santa Fe - Argentina  
+54 0342 4534451

[chiotti@ceride.gov.ar](mailto:chiotti@ceride.gov.ar)

## ABSTRACT

A collaborative B2B relationship implies jointly executing business processes. This relationship demands a complete access to available information and knowledge to support decision-making activities between trading partners. To support information interchange between enterprises in collaborative B2B e-commerce there are some XML-based standards technologies, like RosettaNet, ebXML and OAGIS. However, XML does not express semantics by itself. So, these standards only provide an infrastructure to support the information interchange. They are suitable to integrate information but not to support decision-making activities where a common understanding of the information is needed. In this paper we analyze the integration of these standards with ontology to describe the meaning of the information and knowledge interchanged between trading partners to jointly execute business processes. Furthermore, we define the main components of an ontology development environment to support the entire ontology lifecycle.

## Keywords

Ontology, XML specifications, collaborative B2B.

## 1. INTRODUCTION

Traditional business-to-business (B2B) relationships occur when systems of two or more enterprises exchange information that, directly or indirectly, results in a transaction. A transaction includes traditional purchases and other events such as fulfillment, bids in a real-time auction, inventory replenishment, establishing new accounts, account lookup, and account maintenance. Historically, this integration has been accomplished through manual tasks, via phone and fax, and/or via EDI (Electronic Data Interchange).

Due to the growing globalization and pressures of competitiveness on businesses, companies need to work closely with their partners to collaborate and bring new and better products into market sooner than their competitors [34]. That is, companies and their trading partners are moving to collaborative B2B relationships. This is more than simply sharing forecasts or interchange transactions. In a collaborative B2B relationship, the involved enterprises integrate their information systems to jointly execute business processes, like production and resources planning, sharing decision-making.

Web-based applications provide new opportunities for aligning inter-enterprise processes that allow for collaborative B2B e-commerce. In this context, the success of collaborative B2B e-commerce depends on effective collaborative information systems that help to access the information and knowledge to support collaborative decision-making processes.

After the web-based applications, B2B specifications are the second key enablers of collaborative B2B relationships. These specifications, like ebXML [44], RosettaNet [34] and OAGIS [38] are based on XML (eXtensible Markup Language). Taking into account that XML itself does not guarantee that thousands of business partners will understand a particular message, these specifications propose to share the same vocabulary and meaning before a message can be exchanged. These specifications add semantics to the documents but only to help someone to implement it. However, these documents only provide an infrastructure to interchange information. On the one hand, they do not provide semantics associated to the information in order to facilitate the mapping between different specifications. That is useful because enterprises information systems may use different specifications. On the other hand, they do not provide support to manage the knowledge latent in their content. And that is important in order to support decision-making activities.

Initiatives to solve the semantic problem associated with the information were presented by Klein [25], who proposes to associate the labels of an XML document with something that carries meaning using RDF Schema Specification [4]. However, this only solves a part of the problem, because the knowledge management problem to support decision-making activities in a collaborative B2B relationship is not solved.

The objective of this paper is to analyze the role of the ontology in information and knowledge management between trading partners in a collaborative B2B relationship. Furthermore, we define the main components of an ontology development environment to support the entire ontology lifecycle.

In Section 2, we describe different XML specifications for B2B e-commerce. In Section 3, we present a B2B integration architecture and following, by means of an example, we analyze the semantic problem associated to the information and knowledge contained in business documents interchanged between trading partners. In Section 4, we introduce the ontology concept and we define the components to be integrated within the B2B integration architecture in order to manage the semantics. Finally, in Section 5 we present our conclusions and future work.

## **2. XML-BASED SPECIFICATIONS AND SEMANTICS**

XML is the Extensible Markup Language defined by W3C consortium [3]. It was originally designed to meet the challenges of large-scale electronic publishing, but now it is playing an important role in B2B integration and Enterprise Application Integration (EAI). An interesting analysis on XML-standards was performed by the EDS's XML Innovation Center [47]. This document analyzes XML-standards from eight perspectives: human resource management (HRM), finance, content syndication and synchronization, industrial automation, law, e-commerce, purchasing and logistics. Our interest is on B2B e-commerce, where there are different specifications for describing business documents based on the XML standard.

However, it is not sufficient to adopt and implement an XML-based standard to effectively carry out a collaborative B2B relationship. In this section, we introduce the three most used standards: RosettaNet, ebXML and OAGIS. Then, we analyze these standards from the semantics point of view.

### **2.1 XML-Based Specifications for B2B**

According to Kotinurmi [26] XML-based specifications for B2B can be divided into:

- (1) standard schemas for messages, like cXML (commerce eXtensible Markup Language) [6] and xCBL (XML Common Business Library) [46]; and
- (2) standards defining both the processes and the associated message schemas, like ebXML [44], RosettaNet [34] and OAGIS [38].

The adoption of an XML standard to interchange business documents between trading partners depends on the business model that both trading partners want to implement. In order to support a collaborative B2B relationship it is necessary to manage messages and processes jointly executed by both trading partners. So, the most suitable standards are those that define both the processes and the associated message schemas. Following, we analyze each of these standards.

### 2.1.1 EbXML

The ebXML (Electronic Business using eXtensible Markup Language) [44] goal is to develop an internationally accepted technical framework that will enable XML to be used in a consistent manner to interchange all electronic business data in order to facilitate the trade. EbXML is sponsored by United Nations/ECE agency CEFAC and a not-for-profit, global consortium called OASIS (Organization for the Advancement of Structured Information Standards). It is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. By using ebXML, companies have a standard method to interchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes.

Rather than focusing just on documents, ebXML emphasizes business processes that define data and actions. The ebXML *Business Process Specification Schema* provides the semantics, elements, and properties necessary to define business collaborations. A business collaboration consists of a set of roles collaborating through a set of choreographed transactions by exchanging business documents. Business processes are expressed as process models and encoded in XML DTD (Data Type Definition) and XML Schema. Taking into account the fragment of ebXML Business Process Definition [2] showed in Figure 1, we can see that a Business Process Specification will reference, but not define, a set of required Business Documents.

```
<xsd:schema
targetNamespace="http://www.ebxml.org/BusinessProcess"
xmlns="http://www.ebxml.org/BusinessProcess"
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
elementFormDefault="qualified">
<xsd:element name="Attachment">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="Documentation" minOccurs="0"
maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="name" type="xsd:string"
use="required"/>
<xsd:attribute name="nameID" type="xsd:ID"/>
<xsd:attribute name="businessDocument" type="xsd:string"/>
<xsd:attribute name="specification"
type="xsd:uriReference"/>
<xsd:attribute name="mimeType" type="xsd:string"
.....
</xsd:complexType>
</xsd:element>
```

**Figure 1. A fragment of ebXML Business Process Definition**

All ebXML-developed messages are encoded in XML as well. However, ebXML may transport any type of data such as binary content or EDI transactions. Trading partner agreements are also expressed in XML as well as a business service interface to implement these agreements. A transport and delivery layer moves information among partners, and a formal registry and repository acts as a container for these process definitions, vocabularies, and partner profiles.

### 2.1.2 RosettaNet

RosettaNet is a non-profit organization supported by the major Information Technology, Electronic Components and Semiconductor Manufacturing companies. Its aim is to create and implement industry-wide, open e-business process standards aligning processes between supply chain partners on a global basis.

RosettaNet not only defines data formats, but also transport methods. It developed common Partner Interface Processes (PIPs) to define the business processes between trading partners. PIPs are specialized system-to-system XML-based guidelines that define how business processes are conducted between IT manufacturers, software publishers, distributors, resellers and corporate end-users. Each PIP contains specification documents, Document Type Definition (DTD) and messages guidelines documents. PIPs can be classified into eight core groups of business processes: RosettaNet Support, Partner Profile Management, Product Information, Order Management, Inventory Management, Marketing Information Management, Service and Support, Manufacturing. Each business process between supply chain companies is defined by a specialized system-to-system XML-based dialog.

A fragment of the "Request Purchase Order" Partner Interface Process is presented in Figure 2. This PIP enables a buyer to issue a purchase order, and a provider to acknowledge if the order is accepted, rejected, or pending. In order to facilitate the implementation of this PIP, RosettaNet defines a process model which relate this PIP with others defined by the standard.

```
<!ELEMENT PurchaseOrder
(AccountDescription?, comments?, ContractInformation*,
DocumentReference*, FinancingTerms*,
generalServicesAdministrationNumber?,
GlobalGovernmentPriorityRatingCode?,
GlobalPurchaseOrderFillPriorityCode?,
GlobalPurchaseOrderTypeCode+,
governmentContractIdentifier?,
installAt?, isDropShip, OrderShippingInformation?,
ProductLineItem+, proprietaryInformation?,
requestedEvent?, requestedShipFrom*,
SecondaryBuyer?, shipTo?, TaxExemptStatus?,
totalAmount?)>

<!ELEMENT AccountDescription
(accountName, AccountNumber?, billTo?, CreditCard?,
financedBy?, GlobalAccountClassificationCode?,
PaymentCheckNumber?, WireTransferIdentifier?)>
```

**Figure 2. A Fragment of "Request Purchase Order" PIP**

### 2.1.3 OAGIS

The Open Applications Group Integration Specification (OAGIS) defines Business Object Documents (BODs) as a common message architecture needed to achieve interoperability between disparate systems, disparate companies and disparate supply chains [38]. BODs are the business messages or documents that are interchanged between software applications of collaborating partners.

BODs use metadata to describe what kind of message is to be expected. BODs have the concept "Noun" for common business objects like "Planning Schedule" and the concept "Verb" to name actions like "Get" or "Show" to be applied to a certain business object. OAGIS does not define messaging itself. BODs can be transported via protocols like HTTP or SOAP, or they can be used with more complex messaging frameworks like ebXML messaging service [12] or RosettaNet Implementation Framework 2.0 [20].

The BODs are defined using XML Schema [36]. A BOD contains the framework necessary to convey its two primary components: the business service request and the business data area. Each business service request (BSR) contains a unique verb/noun combination such as *ShowPlanningSchedule* that drives the contents of the business data area (BDA). This BSR and BDA combination corresponds to the object name, method, and arguments model of a procedure call or method-invocation model.

A fragment of XML schema that defines the *ShowPlanningSchedule* BOD is shown in Figure 3 (a). This BOD can be used by trading partners to communicate their current requirement or supply schedule in response to what they received from the other party. Figure 3 (b) shows a fragment of the *ShowShipmentPlanningSchedule* BOD. This BOD defines the element ShipmentPlanningScheduleLine that is reused in the *ShowPlanningSchedule* BOD. This element is reused in others BODs and its meaning depend on the BOD that is reused. That is, "ItemQuantity" means the customer's actual requested amount of the item to be shipped, in the shipment context and to be planning, in the planning context. This fact is only human-understandable but not computer-processable.

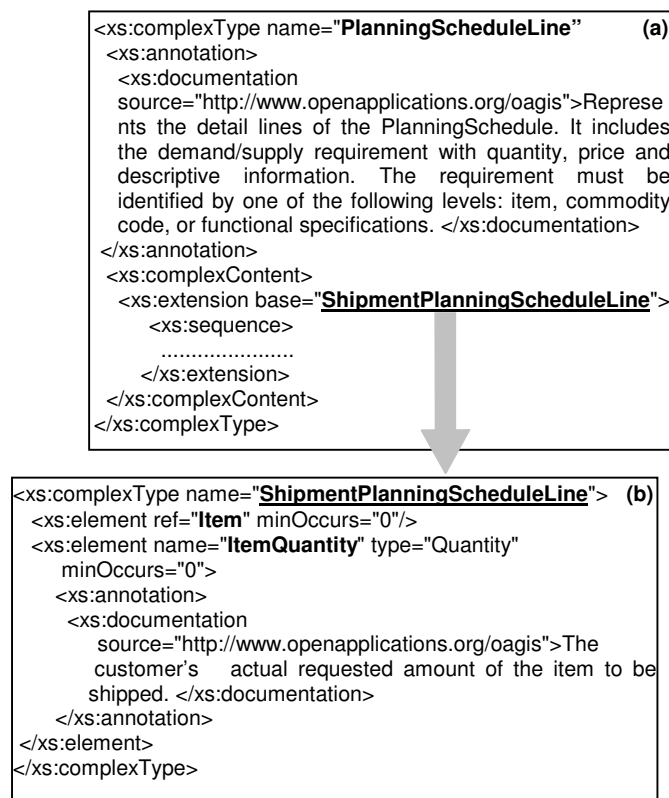


Figure 3. Fragment of BODs

## 2.2 Semantics Associated to XML Standards

XML [3] is a meta-markup language defined by the World Wide Web Consortium (W3C) that provides a way to create extensible formats for describing structured data in electronic documents and to express rules about those data. All XML documents have to be associated to a definition. This definition could be expressed using a DTD (Document Type Definition) or an XML Schema [13]. So, an XML document is then valid if it has an associated definition and complies with the grammatical constraints of that definition. An XML processor carries out that validation.

An XML schema provides a similar service to DTDs but adds important aspects: XML schemas provide a rich set of elementary data types that can be used to define the values of elementary tags; XML schemas provide much richer means for defining nested tags; they also provide the namespace mechanism to combine XML documents with heterogeneous vocabulary. DTDs are insufficient to support the features required for B2B e-commerce [45]. So, some B2B initiatives are defining their specifications using the XML Schema because of its expressive power, e.g. ebXML and OAGIS.

In order to have application programs talking to each other, they need not only a way to know how the data they receive are structured but also what the vocabulary means, its semantics. In an XML document, the elements are delimited by start and end tags. Furthermore, it has a type and may have a set of attribute specifications consisting of a name and a value. XML allows us to create our own markup, e.g. `<PurchaseOrder>` or `<ItemQuantity>`, which seems to carry some semantics. However, from a computational perspective a tag like `<ItemQuantity>` carries as much semantics as a tag like `<H1>`. A computer simply does not know what an *ItemQuantity* is and how the concept of *ItemQuantity* is related to e.g. the concept of *PurchaseOrder*.

A way to include semantics in an XML schema is to provide information on the element meaning within the `<annotation>` element and to adopt the convention that every element and attribute has an *annotation* [9]. Considering the OAGIS standard, the meaning of the elements in the BODs is described by using the `<annotation>` element. One problem is that these semantics are described using the natural language which could be ambiguous and it is not machine-processable. Another problem that we have detected is that some elements have the same descriptions. Furthermore, XML processors have no means of validating semantics even if they are declared informally in an XML schema. Semantics validation, however, can be important in several situations, such as in case two XML servers communicate with each other without human intervention.

In fact, XML syntax is designed for representing an encoded serialization, and thus it has a very limited range of expression for modelling complex object semantics, where "semantics" fundamentally means an intricate web of constrained relationships and properties [10]. So, into XML semantics can be expressed only informally (e.g., in comments). For this reason, other languages have to be used to express and validate semantics integrity constraints.

In the following section, we analyse the problem associated to the lack of semantics in XML-based specifications for a collaborative B2B relationship. Particularly, we show an example using OAGIS specification.

### 3. DISCOVERING SEMANTIC PROBLEMS

We are focused on the development of an information system to support a Partner-to-Partner Collaborative Model that proposes a decentralized management of the supply chain [42]. This model proposes three collaborative processes: collaborative production and resource planning, collaborative master production scheduling and rough-cut capacity planning, and collaborative material requirement planning and capacity requirement planning.

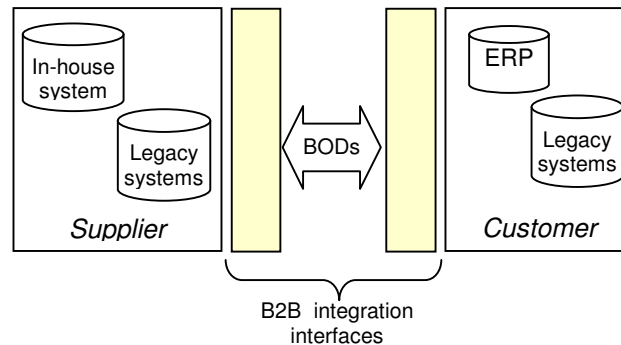
Taking into account this model, firstly we introduce a B2B integration architecture and then we analyze potential problems associated with the XML-based specifications due to the lack of semantics.

#### 3.1 Collaborative B2B Scenario

Let us consider a B2B relationship between middle-size enterprises, as an example. One of the enterprises, the *Customer*, manufactures bicycles, and the other, the *Supplier*, supplies the pneumatics for the bicycles' wheels. Both enterprises want to jointly execute a Production and Resource Planning Process in order to achieve a consensus in the pneumatic supply schedule. During the Collaborative

Partner Agreement, both enterprises define the information needed to execute this business process and its format. They decide to use Business Object Documents defined by OAGIS due to their maturity.

From the supplier or customer point of view, a typical architecture to support a collaborative B2B relationship combines intra-enterprise systems and a B2B integration interface responsible for business documents interchanges. The document interchanging can be done directly between these interfaces, implementing a peer-to-peer system, or indirectly by a third party, via an e-marketplace. After an analysis we concluded that to support a Partner-to-Partner collaborative model the most suitable technology is the one associated with Peer-to-Peer systems [41]. Particularly because of the autonomy. Figure 4 shows the general architecture to support a collaborative peer-to-peer relationship. This technology does not require a mediator between enterprises as e.g. the e-marketplace. So, document management has to be done in a decentralized way.



**Figure 4. Collaborative B2B Relationship**

Once the general architecture is defined, the layer-based architecture that composes the B2B integration interfaces has to be defined. The interface, schematically represented in Figure 5, is defined by the customer that has to send the requirement schedule information to the supplier in order to start the Collaborative Production and Resource Planning Process.

### 3.1.1 Extraction Layer

The information to be exchanged, which in this case is the requirement schedule information, could be stored in different data sources with different formats. So, on the customer side this information has to be extracted from these information sources and formatted to the XML standard. Taking into account the business rules defined in the Partner Agreement an “Extraction layer” performs this task. Furthermore, as regards incoming information, this layer has to transform it into the internal format. This is a complex layer because sometimes manual steps are needed to complete the information.

### 3.1.2 Translation Layer

Then, this information has to be translated into the corresponding standard. That is, the customer could establish more than one collaborative relationship with their suppliers. Therefore, this process is responsible for dealing with heterogeneous partners that might conform different standards. In this case, let us suppose that in this relationship the *ShowPlanningSchedule* BOD, shown in Figure 3 (a) has to be instantiated. In this case, a “Translation Layer” takes the information that has been extracted and instantiates the corresponding BOD.

### 3.1.3 Transport Layer

Finally, BODs are sent to the supplier through the “Transport Layer”. This layer is responsible for validation and security. In this paper we do not focus on this layer. We assume that BODs will be transported using a protocol or a messaging framework.

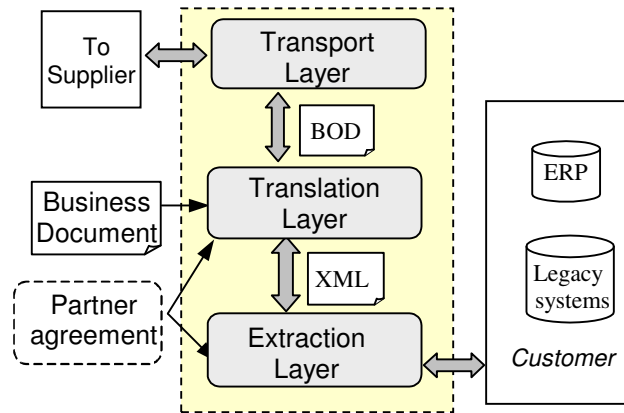


Figure 5. The architecture of B2B integration interface

## 3.2 Semantics Associated with Information

Following the example, let us suppose that the “Extraction Layer” generates an XML document, which is shown in Figure 6, that contains the requirement information. The problem arises when, on the customer side, the programmer wants to define the “Translation Layer” to translate the requirement information shown in Figure 6 to the BOD shown in Figure 3 (b). That is because:

- (1) descriptions contain some elements with no evident semantics, e.g. “**IdReference**” on the customer side;
- (2) the XML schema defined on the customer side provides more information than the BOD fragment, e.g. “**UnitOfMeasure**”;
- (3) both specifications use different terms to specify the same object, e.g. “**Quantity**” and “**ItemQuantity**”.

```

<xs: IdReference> 8564 </xs: IdReference>
<xs:OrderItem> 20 inch Flat-proof bicycle tires </xs: OrderItem>
<xs: UnitOfMeasure > Unit </xs: UnitOfMeasure >
<xs: Quantity> 24 </xs: Quantity>

```

Figure 6. An instance of Requirement schedule information

In order to solve this problem Klein [25] proposes to associate the labels of an XML document with something that carries meaning using Resource Description Framework (RDF) Schema Specification [4]. The RDF data model defines a simple model for describing interrelationships among resources in terms of named properties and values. The RDF data model, however, provides no mechanisms for saying



certain things about certain kinds of resources (attribute), nor does it provide any mechanisms for defining the relationships between these attributes and other resources. That is the role of RDF Schema.

A schema defines not only the properties of the resource (Title, Author, Subject, Size, Color) but may also define the kinds of resources being described (books, Web pages, people, companies).

The RDF Schema document does not specify a vocabulary of descriptive elements such as "author". Instead, it specifies the mechanisms needed to define such elements, to define the classes of resources they may be used with, to restrict possible combinations of classes and relationships, and to detect violations of those restrictions. RDF Schema defines resources and properties such as *Class* and *subClassOf* that are used in specifying application-specific schemas. However, in order to achieve interoperation between numerous, autonomously developed and managed schemas, richer semantics are needed. For example, RDF Schema cannot specify that *Person* and *Bicycle* classes are disjoint, or that a *Bicycle* has exactly two *Wheels*.

Furthermore, adding meaning to the elements contained in an XML document is not enough to solve the semantics problem. Let's us consider the following situation. On the supplier side, the requirement information has to be integrated into the ERP system in order to generate a supply schedule that can satisfy this demand. A problem arises because the ERP system cannot interpret the meaning of "20 inch Flat-proof bicycle tires" because it uses different terminology to refer to the same product. So, the semantics problem is not only associated to the definition of each element contained in the XML Schema document but also with the meaning of the content of each element.

### 3.3 Semantics Associated with Knowledge

Knowledge is not a collection of information. From XML documents without semantics we do not infer knowledge because the knowledge exists when we can recognize patterns amidst data and information [1]. This inference can be done if we know what exactly the information and data mean.

Knowledge is valuable in a collaborative B2B relationship because decision-making activities depend on it. To make a good decision, reliable information and knowledge are required.

In this paper we do not talk about sharing knowledge between trading partners. We are only focused on the relationship between semantics and latent knowledge present in XML documents. We explain this concept following the example above presented. Let us suppose that the customer sends to the supplier information related to the sale data specifying product, quantity and sale date. On the one hand, these data with the corresponding semantics allow the supplier to generate appropriate information to make decisions. On the other hand, by saving these data and their semantics over time, the supplier will have a set of historic data and their semantics. Then, they could be analyzed to detect sale behavior patterns such as tendency or seasonally. That is, to infer knowledge. Furthermore, if the supplier also receives data and their semantics about the consumers of these products, it could "mine" them to infer knowledge about behavior patterns of consumers.

For the purpose of sharing information and knowledge (i. e. for interoperability) between different enterprises a shared set of terms describing the application domain with a common understanding is needed.

As pointed out above, the use of XML is insufficient for determining the resources semantics. This problem can be solved using ontologies to describe the meaning of the tags included into an XML document in order to process this document according to the intention of each tag and to facilitate the discovering of the knowledge latent in its content.

In the next section, we introduce the ontology concept and we define the components that have to be integrated within the B2B integration interface in order to manage the semantics associated to the XML documents.

## 4. ONTOLOGY-BASED APPROACH

The use of ontologies is a straightforward and promising approach in order to explain contextual information and to make semantics-preserving translation possible [37]. The use of ontologies to solve the semantics problem in business integration is not new. However, quite often ontologies are used as simple or structured vocabularies and in this role they do not provide any substantial benefit when compared to existing techniques [32]. This is so because the term “ontology” has been used to describe artifacts with different degrees of structure. In this paper we consider an ontology as “a vocabulary of terms and some specifications of their meaning through axioms and properties using a logic-based representation language”. Properties define the attributes of terms. Axioms are properties of the relation between ontology terms.

Some ontology-based approaches are trying to solve the semantics problem related to B2B relationships but most of them are focused on solving the document integration problem related to an e-marketplace B2B relationship. An e-marketplace can operate as an Internet-based intermediary that focuses on specific industry verticals or specific business processes and use various market-making mechanisms to mediate any-to-any transactions among enterprises [11]. One of these initiatives is presented by Omelayenko and Fensel [33]. In this initiative they suppose that each document has a counterpart to be translated, or there exists well-specified dependency between the documents. Furthermore, they suppose that there is a single ontology maintained by the e-marketplace. In this context, the Customer enterprise sends a *Purchase order* to the Supplier enterprise following some document standard, like for example cXML, via the e-marketplace. This document is described using the single ontology and then translated into the document standard used by the supplier side, like xCBL.

In Peer-to-Peer systems, as required to support Partner-to-Partner Collaborative model, there is no mediator between enterprises, so document management has to be done in a decentralized way. Taking this into account, the objective of this section is to briefly introduce the main components needed to support the semantics interoperability analyzing the requirements associated to the information system.

### 4.1 Collaborative Ontology Development

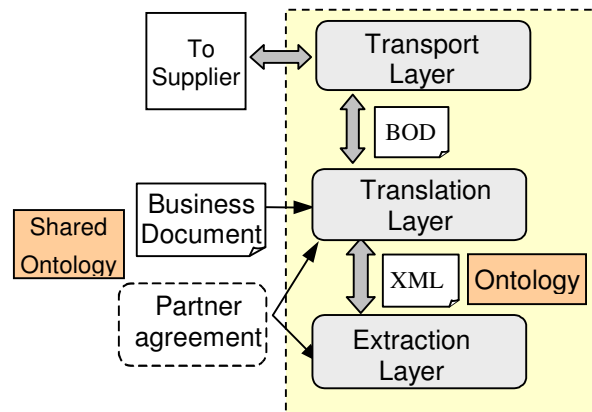
Our focus is on the analyzes of the ontologies that support the process that takes place in the B2B integration interface. These ontologies have to describe the information and document that support the collaborative relationship.

During the collaborative agreement the Partner Agreement and the documents to be interchanged are defined. During this process the formats of these documents and their semantics are defined. That is, an ontology. We refer to this ontology as *shared ontology*, as shown in Figure 7. The creation and maintenance of this ontology have to be performed in a collaborative way. That is, the *shared ontology* has to be constructed in a collaborative effort of domain experts of both trading partners. This shared ontology is not a common ontology for all collaborative relationships. That is, if an enterprise has more than one collaborative relationship then it can have a shared ontology for each relationship.

Furthermore, the information generated by the Extraction Layer has to be described by an ontology. This ontology helps the Translation layer to map this information into the standard documents that are interchanged between trading partners. To develop this ontology experts of the enterprise need to work jointly. The objective is not to define a common ontology of the enterprise but to define semantics of the information to be exchanged. This ontology is showed in Figure 7 associated to the intermediate XML documents.

The main characteristic of the ontologies described above is that they are a product of a joint effort of a group of people. This joint effort requires (1) the use of a methodology that guides the ontology development process and (2) tools to inspect, browse, codify, modify, and download the ontology.

On the one hand, at present the construction of ontologies is more an art than a science; however, some methodologies have been developed to support the development of ontologies. Examples of such methodologies include METHONTOLOGY [15], a methodology based on experiences in the development of TOVE (Toronto Virtual Enterprise) [39] and the methodology based on the experience of developing the ENTERPRISE ONTOLOGY [40]. What these methodologies have in common is that they start from the identification of the purpose of the ontology and the need for domain knowledge acquisition. In most cases, these methodologies are very general and they only provide basic guidance for the development of an ontology infrastructure [18]. Furthermore, these methodologies do not consider collaborative and distributed construction of ontologies [8].



**Figure 7. Ontology within B2B architecture**

On the other hand, both ontologies are the result of agreement processes that take a lot of effort and time. Agreeing on the ontology definition and later on maintaining this ontology within a worldwide-distributed group of people is a challenging task. Currently almost no tools are available to support this task.

Briefly, to support the semantics interoperability between enterprises a collaborative environment and methodology to guide the ontology development is necessary. So, the first component to be taken into account is a collaborative development environment that supports the agreement processes to develop an ontology.

## 4.2 Ontology Languages Interoperability

Ontologies can be built in different ways using different languages. At the beginning of the 1990s, a set of AI-based ontology implementation languages was created. These traditional languages were based on frames combined with order logic, like Ontolingua [14], OKBC [5] and Flogic [22] and on description logic, like LOOM [28]. Some traditional languages, like OKBC, have become the bases of other ontology languages. An interesting comparison of ontologies specification languages is presented in [7].

Furthermore, there are other languages created in the context of Internet that exploits the characteristics of the Web. Such languages are usually called *web-based ontology languages* or *ontology markup languages*. These languages are still in a development phase: they are continuously evolving [8].

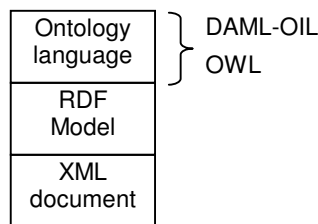
RDF [26] and RDF Schema [4] were developed by the W3C. An RDF description is a list of triples: an object (a resource), an attribute (a property), and a value (a resource or free text). The RDF Schema is an extension of RDF. With RDF Schema, one can define classes that may have multiple subclasses and super classes, and can define properties, which may have sub properties, domains, and ranges. In this

sense, RDF Schema is a simple ontology language. This language is useful to describe the taxonomy of the ontology in order to support the translation process. This was demonstrated in the work done by Omelayenko [33].

Other web-based ontology languages used to represent the knowledge contained in an ontology in a simple and human-readable way, as well as to allow for the interchange of ontologies across the web, are XOL [21], SHOE [17] and DAML+OIL [29]. The latter is being evolved into the OWL Web Ontology Language standard [30], which is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL provides three increasingly expressive sublanguages designed for use by specific communities or users and implementators: OWL Lite, OWL DL and OWL Full.

The RDF and RDF schema are suitable to represent the ontology objects but another language to represent the semantics in order to infer knowledge is needed. So, different layers represent the ontology, as presented in Figure 8. The layer approach to data modeling [31] has been successfully tried in practice.

The existence of different ontology specification languages leads us to conclude that an ontology development environment has to include an approach for language interoperability. That is, the interoperability between different ontologies at language specification level. Furthermore, this system has to deal with different layers of ontology representation.



**Figure 8. Relationship Between Ontology Languages**

### 4.3 Ontology Mappings

For information integration, mappings between ontologies and the information they describe, and mappings between ontologies used in the system are required [43].

On the one hand, the shared ontology may relate to the elements contained in the Business Document and its content. Furthermore, there will be an ontology associated to the information that is the output of the “Extraction Layer”.

On the other hand, given the shared ontology and the ontology associated with the intermediate XML documents, one should be able to map concepts found in one ontology into the ones found in the other. That is useful in order to translate the information into the BOD and vice versa.

The problem of mapping different ontologies is a well-known problem in knowledge engineering. There are systems that perform ontology mapping but they are often either embedded in an integrated environment for ontology editing or attached to a specific formalism. Furthermore, in most cases mappings are based on heuristics that mostly use syntactic clues to determine correspondence or equivalence between ontology concepts, but rarely use the meaning of those concepts [19].

So, in order to achieve semantics interoperability we need a mapping process based on mapping between terms taking into account their meaning. Figure 9 tries to show that the mapping between both the terms of the shared ontology and the enterprise ontologies is necessary. Furthermore, Ontology mapping

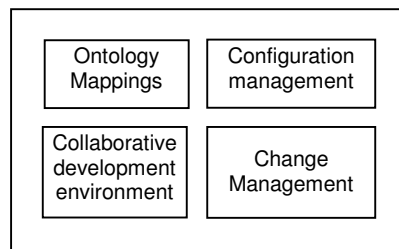
component is separated from the collaborative development environment in order to represent that this component should not be embedded in the environment to facilitate its reuse.

#### 4.4 Ontology Versioning and Change Detection

Ontologies are often developed by several persons and continue to evolve over time. Enterprise domain changes, adaptations to different tasks, or changes in the conceptualization might cause modifications of the ontology. Therefore, both the development and the maintenance of ontologies require advanced versioning methods. Configuration management component, which takes care of the identification, relations and interpretation of ontology versions, is necessary [23]. This component is represented in Figure 9.

Furthermore, we have to take into account that the data stored in XML documents, which are the output of the “Extraction Layer”, could change. This will probably cause incompatibilities in the “Translation Layer” and ontologies that refer to them. So, change detection is important to avoid wrong data interpretations and data inaccessibility [24]. Changes that affect the ontology taxonomy are sometimes easy to discover. The problem of schema evolution has been extensively studied in the database field and can be applied to this context. But changes that affect the conceptualization depend on humans being and can be detected only by analyzing the user’s interaction with the system. The same problem can be analyzed inside the enterprise and affects the “Extraction Layer”. Therefore, a component to detect changes is necessary.

So, an efficient ontology management system to support collaborative B2B relationships has to support ontology versioning and change detection related to enterprise ontologies and shared ontology.



**Figure 9. Ontology Management System Components.**

## 5. CONCLUSIONS

Web-based applications provide new opportunities for aligning inter-enterprise processes allowing for collaborative B2B e-commerce. In this context, the success of collaborative B2B e-commerce depends on effective collaborative information systems that help with the access of information and knowledge to support collaborative decision-making processes.

In order to enable collaborative B2B relationships some XML-based specifications, like RosettaNet, ebXML and OAGIS were defined. Taking into account that XML itself does not guarantee that thousands of business partners will understand a particular message, these specifications propose to share the same vocabulary and meaning before a message can be interchanged. However, the use of XML is insufficient for determining the resources semantics. This problem can be solved using ontologies to describe the meaning of the tags included into an XML document in order to process this document according to the intention of each tag and to facilitate the discovering of the knowledge latent in its content.

Some ontology-based approaches are trying to solve the semantics problem related to B2B relationships but most of them are focused on solving the document integration problem related to an e-marketplace B2B relationship. In contrast, our initiative is focused on the development of a ontology-based approach to solve the semantics problem in Peer-to-Peer systems. The main difference to e-marketplace is that there is no mediator between enterprises.

If we focus on ontology tools, the main problem to be solved in this area is the lack of integrated environments for ontology development. We have defined four components to support the semantics interoperability analyzing the requirements associated to the ontology development environment. Furthermore, we have defined some characteristics of them. The first component, the collaborative development environment, is responsible for the entire ontology lifecycle supporting agreement processes and language interoperability. This component should support collaborative and distributed construction of ontologies. The second component, Ontology mappings, is responsible for maintaining the mappings between ontologies and the information they describe, and mappings between different ontologies used in the system. The third component, Ontology versioning, manages different versions of the different ontologies that compose the B2B integration interface. The fourth component, Change Management, has to detect changes in the data and structure of the information in order to update the ontology that describes them to avoid misunderstandings.

Our future work is to develop a methodology and a system that support the ontology creation from B2B XML-based specifications. Furthermore, this system has to support the maintenance of this ontology in a collaborative way.

## 6. REFERENCES

- [1] Bellinger, G. Knowledge Management - Emerging Perspectives. Retrieved from <http://www.outsights.com/systems/kmgmt/kmgmt.htm>.
- [2] Business Process Project Team (May 2001). EbXML Business Process Specification Schema Version 1.01. Retrieve from <http://www.ebxml.org/specs/ebBPSS.pdf>
- [3] Bray, T.; Paoli, J.; Sperberg-McQueen, C. M., Maler E and Yergeau, F. Extensible Markup Language (XML) 1.0 (3<sup>o</sup> Edition). February, 2004. W3C Recommendation. Available on-line <http://www.w3.org/TR/2004/REC-xml-20040204/>
- [4] Brickley, D and Guha R.V., RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft, January 2003. Retrieved from <http://www.w3.org/TR/PR-rdf-schema>.
- [5] Chaudhri, V., Farquhar, A, Fikes, R., Karp, P., Rice, J. The Generic Frame Protocol 2.0. (July, 1997). Retrieve from <http://www.ai.sri.com/~gfp/gfp2/gfp2.html>
- [6] Commerce XML. Retrieved from <http://www.cxml.org/>
- [7] Corcho O. and Gómez-Pérez, A. A Roadmap to Ontology Specification Languages. Lecture Notes on artificial Intelligence. 2nd International Conference on Knowledge Engineering and Knowledge Management. Springer-Verlag. October 2000. 80-96.
- [8] Corcho, O.; Fernández-López, M.; Gómez-Pérez, A. Methodologies, tools and Languages for building ontologies. Where is the meeting point? Data and Knowledge Engineering. 46 (2003) 41–64.
- [9] Costello, R. L. (2002). XML Schema Tutorial. Available on-line <http://www.xfront.com/xml-schema.html>
- [10] Cover, R.: The XML Cover Pages, XML and Semantics Transparency, Last modified: November 24, 1998. Available on-line <http://www.oasis-open.org/cover/xmlAndSemantics.html>

- [11] Dai, Q and Kauffman, R. J. (2001). Business models for Internet-based e-procurement systems and B2B electronic markets: an exploratory assessment. Proceeding of the 34<sup>th</sup> Hawaii International Conference on System Sciences.
- [12] Dubray, Jean-Jacques (2001). OAGIS with ebXML's (CPP, CPA and BPSS specifications). White Paper.
- [13] Fallside, D.C. (May, 2001) XML Schema Part 0: Primer W3C Recommendation. Retrieved from <http://www.w3.org/TR/xmlschema-0/>
- [14] Farquhar, A., Fikes, R., Rice, J. The Ontolingua Server: A Tool for Collaborative Ontology Construction. Proceedings of KAW96. Banff, Canada, 1996.
- [15] Fernández, M; Gomez-Perez, A and Juristo, N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering Workshop on Ontological Engineering. Spring Symposium Series. AAAI97 Stanford, USA. <http://delicias.dia.fi.upm.es/miembros/ASUN/SSS97.ps>
- [16] Gruber, T.R. A translation approach to portable ontology specification, Knowledge Acquisition 5 (1993) pp. 199–220.
- [17] Heflin, J.; Hendler, James and Sean Luke. SHOE: A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078 (UMIACS TR-99-71). 1999.
- [18] Jones, D., Bench-Capon, T., and Visser, P. Methodologies for Ontology Development. In Proceeding of IT&KNOWS Conference of the 15th IFIP World Computer Congress, Budapest, Chapman-Hall, 1998.
- [19] Kalfoglou, Y. and Schorlemmer, M. Information Flow based ontology mapping. In Proceedings of the 1st International Conference on Ontologies, Databases and Application of Semantics (ODBASE'02), Irvine, CA, USA, October 2002.
- [20] Kanaskie, Kurt (2001). OAGIS over RosettaNet's RNIF. White Paper.
- [21] Karp, P. XOL: An XML-Based Ontology Exchange Language. June, 2003. Retrieved from <http://www.ai.sri.com/~pkarp/xol/>
- [22] Kifer, Michael; Lausen, Georg and Wu, James. Logical Foundations of Object-Oriented and Frame-Based Languages. JACM 42 (4): 741-843 (1995).
- [23] Klein, M. (2002). Supporting evolving ontologies on the internet. In Lindner, W. and Stuller, J., editors, Proceedings of the EDBT 2002 PhD Workshop, pages 597-606, Prague, Czech Republic. Retrieved from <http://www.cs.vu.nl/~mcaklein/papers>
- [24] Klein, M. and Fensel, D. Ontology versioning for the Semantics Web. In Proceedings of the International Semantics Web Working Symposium (SWWS), pages 75 – 91, Stanford University, California, USA, July 30 – Aug. 1, 2001. Retrieved from <http://www.cs.vu.nl/~mcaklein/papers>
- [25] Klein, Michel: Interpreting XML via an RDF schema. In *ECAI workshop on Semantics Authoring, Annotation & Knowledge Markup (SAAKM 2002)*, Lyon, France, July 23, 2002.
- [26] Kotinurmi, P. Comparing XML-based B2B frameworks, XML Finland 2002, Helsinki, Finland, October 2002.
- [27] Manola, F and Miller, E. Resource Description Framework (RDF) Premier. W3C Recommendation. February 2004. Available on-line <http://www.w3.org/TR/REC-rdf-syntax/>.
- [28] MacGregor, Robert M.: Inside the LOOM Description Classifier. SIGART Bulletin 2(3): 88-92 (1991).
- [29] McGuinness, D.; L., Fikes, R.; Stein, L. A. and Hendler, J. DAML-ONT: An Ontology Language for the Semantics Web. In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. The Semantics Web: Why, What, and How, MIT Press, 2001.

- [30] McGuinness, D. L. and Frank van Harmelen. OWL Web Ontology Language: Overview. W3C Recommendation February, 2004. <http://www.w3.org/TR/owl-features/>
- [31] Melnik S. and Decker, S. A Layered Approach to Information Modeling and Interoperability on the Web. In Proceeding of the ECDL'00 Workshop on the Semantics Web, 2000. Retrieved from <http://www-db.stanford.edu/~melnik/pub/sw00/>
- [32] Omelayenko B. Ontology-Mediated Business Integration. In Proceedings of the 13-th EKAW 2002 Conference, Siguenza, Spain, October 1-4, LNAI 2473, 2002, pp. 264-269. Springer-Verlag.
- [33] Omelayenko B., and Fensel D. (2001): Scalable Document Integration for B2B Electronic Commerce. Special Issue of ELECTRONIC COMMERCE RESEARCH JOURNAL on B2B RESEARCH, 2001.
- [34] RosettaNet. Retrieved from <http://www.rosettanel.org>
- [35] Sahay, B. S and Mohan, R. Emerging issues in supply chain management. VISION: The Journal of Business Perspective, Special Issue on Supply Chain Management, Vol. 7, pp 1-12, 2003.
- [36] Stuart, I. XML Schema, a brief introduction. December, 2003. Retrieved from <http://lucas.ucs.edu.ac.uk/xml-schema/>
- [37] Stuckenschmidt, H. Ontology-Based Information Sharing in Weakly-Structure Environments. Ph.D. Thesis, Faculty of Sciences, Vrije Universiteit Amsterdam, January 2003. <http://www.cs.vu.nl/~heiner/publications.html>
- [38] The Open Applications Group XML Project Team. OAGIS Release 8.0. (May, 2002).
- [39] Uschold, M. and Gruninger, M. (1996). Ontologies: principles, methods, and applications. Knowledge Engineering Review, 11(2):93-155.
- [40] Uschold, M. and King, M. (1995). Towards A Methodology for Building Ontologies. In Proceeding of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada.
- [41] Villarreal, P; Caliusco, M. L.; Galli, M. R.; Salomone, H.; Chiotti, O. (2003) Decentralized Process Management For Interenterprise Collaboration. VISION: The Journal of Business Perspective, Special Issue on Supply Chain Management, Vol 7. pp 69-80.
- [42] Villarreal,P., Caliusco, M L., Zucchini, D., Arredondo, F., Zanel, C., Galli, M.R., and Chiotti, O. (2003). Integrated Production Planning and Control in a Collaborative Partner-to-Partner Relationship. In S. Sharma & J. Gupta (eds.), Managing E-Business in the 21<sup>st</sup> Century . pp 91-110. Heidelberg Press, Australia.
- [43] Wache, H.; Vögele, T.; Visser, U.; Stuckenschmidt, H.; Schuster, G.; Neumann, H.; and. Hübner, S. Ontology-Based Integration of Information - A Survey of Existing Approaches. Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing, Retrieved from <http://www.cs.vu.nl/~heiner/publications.html>
- [44] Walddt, D. and Drummond, R. EbXML - The Global Standard for Electronic Business. Retrieved from <http://www.ebxml.org>.
- [45] Willaert, Frederik. XML-based frameworks and standards for B2B e-commerce. Katholieke Universiteit Leuven, 2001.
- [46] XML Common Business Library, version 4.0 (March, 2003). Retrieved from <http://www.xcbl.org/xcbl40/xcbl40.html>
- [47] XML Standards Profiles (November, 2000). Prepared for the US Postal Service. Retrieved from <http://www.eds.com>



