

Adding Semantics to Electronic Business Documents Exchanged in Collaborative Commerce Relations

Mariela Rico¹, Ma. Laura Taverna², Ma. Laura Caliusco³,
Omar Chiotti⁴ and Ma. Rosa Galli⁵

¹ Universidad Tecnológica Nacional, CIDISI R&D Center, mrico@frsf.utn.edu.ar

² Universidad Tecnológica Nacional, CIDISI R&D Center, ltaverna@fiq.unl.edu.ar

³ UTN-CONICET, CIDISI R&D Center, Santa Fe, Argentina, mcaliusc@frsf.utn.edu.ar

⁴ INGAR-CONICET-UTN, Santa Fe, Argentina, chiotti@santafe-conicet.gov.ar

⁵ INGAR-CONICET-UTN, Santa Fe, Argentina, mrgalli@santafe-conicet.gov.ar

Received 2 April 2008; received in revised form 2 November 2008; accepted 30 January 2009

Abstract

In order to support the execution of collaborative business processes, trading partners have to exchange electronic business documents. Since each partner has its own systems and culture, they could use different terms and metadata structures to represent their data, even when referring to the same domain of interest. Then, an appropriate approach for defining the semantics of these documents is required to achieve semantic interoperability. To this aim, a number of approaches have been proposed based on the use of a single domain ontology. These approaches imply the imposition of a global meaning of terms, which are used to represent the entities of a domain. This goes in opposition to one of the aims of trading partners, which is to participate in collaborative business processes without losing their autonomy and privacy. In this scenario, it is necessary to define a framework for allowing trading partners to exchange information without imposing a global meaning of it. In this paper, a novel approach is proposed in order to define the semantics of the electronic business documents interchanged between trading partners in a collaborative relationship. This approach is based on the idea of making some domain features, which are generally implicit, explicit in the domain ontologies. An application example of this approach is also presented.

Key words: Collaborative Commerce, Semantic Interoperability, Context, Ontology, Electronic Business Document

1 Introduction

In the new competitive global market, an increasing number of organizations are exploiting the potential of modern technologies to conduct business over the Internet. These technologies allow implementing business models, which were unthinkable a few years ago, such as the collaborative commerce between faraway enterprises. Collaborative Commerce can be defined as the use of Internet technologies to integrate core business processes of an enterprise with those of its customers, suppliers and trading partners. Using collaborative commerce capabilities, trading partners can operate as a single business entity. They can make joint decisions and focus on adding value to their customers. In order to implement a collaborative commerce, enterprises have to decide on both the business models and the information technologies to be implemented.

As regards business models, several centralized and decentralized models for implementing collaborative relations between enterprises have been proposed. Collaborative Planning, Forecasting and Replenishment (CPFR) [33]; and Demand Activated Manufacturing Architecture (DAMA) [8] models can be mentioned as examples of centralized models. On the other hand, examples of decentralized models are Partner-to-Partner Collaborative Model [31], and Virtual Collaborative Forecasting Management (V-CFM) [12].

With regard to information technologies, collaborative commerce entails sharing heterogeneous and distributed information resulting from decision-making activities involved in inter-enterprise and intra-enterprise processes. To this aim, trading partners have to exchange Electronic Business Documents (EBD). To support collaborative commerce, EBDs based on XML (eXtensible Markup Language) [4] were defined. Examples of these specifications are xCBL [35], ebXML [34], RosettaNet [24], and OAGIS [28]. They propose to share the same vocabulary and meaning before exchanging a message. They provide the syntax to exchange information, but they do not provide the semantics associated with this information in order to facilitate the mapping between different specifications. As a result, XML does not guarantee by itself that two trading partners will understand a particular message since they could use different terms and metadata structures to represent their data, even when referring to the same domain of interest. Despite the fact that the specifications of XML-based EBD have been a great advance in the business area, they do not define the semantics necessary to achieve a real communication between trading partners. Moreover, they lack the ability of managing the knowledge latent in the EBD content, which is important in order to support joint decision-making activities.

In order to represent the information semantics, different representational artifacts have been proposed from which the most prevalent one is ontology [16]. With the aim of solving semantic problems associated with XML-based EBDs, several ontology-based initiatives have been developed [19, 5]. Most of them propose to define a global ontology, which means that all trading partners have to modify their internal data structures to convey the global ontology. Collaborative commerce is characterized by enterprises that want to exchange only *the necessary* information with their trading partners to achieve common goals, without losing their autonomy and privacy. In order to achieve semantic interoperability, it is not appropriate to define a global meaning of the private information of each enterprise exchanged in the EBDs. Then, it is necessary to define an approach which allows trading partners to exchange information without imposing a global meaning.

Motivating scenario: The following motivating scenario briefly introduces the problem. A complete analysis can be found in Section 4.1.

Web-based applications provide new opportunities to align inter-enterprise business processes allowing collaborative commerce between faraway enterprises. This fact has led to the definition of different business models. This paper is focused on the Partner-to-Partner Collaborative Model [31], which proposes a decentralized management of long-term relations among enterprises of a supply chain.

In order to implement this model, it is not only necessary to define the collaborative processes and the EBDs associated with them, but also the semantics of the information involved. In this scenario, there are two kinds of ontologies: the ones that describe the semantics of the private information of each enterprise, and the ones that describe the semantics of the EBDs. With the aim of achieving data integration at a semantic level, each trading partner has to define a translation process capable of converting the instances of one ontology into another. That is, the instances of an EBD ontology can be translated into the instances of a private ontology. This translation process has to face the problem of how to execute an effective translation via conversion rules, considering that the involved ontologies could be very heterogeneous.

When trying to define conversion rules, it is necessary, in the first place, to identify the correspondences between the ontology entities. To this aim, the ontology-matching techniques could be used. In a data integration process, a wide set of correspondences is crucial as a previous step to the definition of conversion rules. In this paper, it is assumed that the set of correspondences can be improved if the input of the matching process is improved.

Our Approach and Contributions: With the aim of improving the input of the matching process, the idea of making explicit contextual features generally implicit in an ontology is proposed. Next, a brief discussion is given.

Each domain has a set of particular features, named here context, which affects the semantics of concepts associated with it. Therefore, an ontology is composed by a set of elements (axioms, relations, and terms) which describes the inherent features of a real-world object, and another set of elements representing the context of the domain which affects its semantics. In an ontology, this context is implicit in its definition, which is behind an ontology. The context of each domain has to be made explicit in the ontology definition to allow the semantic interpretation of a term in different domains.

The objective of this paper is to present an approach based on the idea of disaggregating the ontology elements making explicit the set of domain features that affects the semantics of each concept. In addition, an application example of this approach is presented.

The remainder of the paper is organized as follows: Section 2 presents the background and definitions to the terminology used in this paper. Section 3 presents an approach for adding semantics to EBDs by making explicit the context behind an ontology. Section 4 presents a reference architecture for supporting the approach previously presented. Section 5 discusses an application example. Section 6 discusses related work. Finally, Section 7 is devoted to discussion and future work.

2 Background

The aim of this section is to provide an introduction of some concepts that are used along the paper, regarding business models for supporting collaborative commerce, the necessity for making the semantics explicit in the business documents, and ontology and context.

2.1 Business Models for Collaborative Commerce

In literature, some models that propose a centralized management of the supply chain can be found in order to implement collaboration between enterprises, such as CPFR [33] and DAMA [8]. The CPFR model has been successfully implemented by some enterprises like Wal-Mart and Procter & Gamble. It focuses on the manufacturer/retailer partnership while the DAMA model assumes a supply chain with multiple trading partners collaboratively working to meet consumer demands. The latter proposes a centralized coordination of all trading partners of the supply chain. It may consist of one to four centralized processes (defining products, forecasting and planning capacity commitments; scheduling products and product delivery; expediting production; and delivery exceptions).

A centralized collaborative model is mainly based on the concept of visibility of the whole supply chain, which enables inter-enterprise optimization of the operation variables involved in the logistics of the supply chain. Then, centralized models are appropriate for collaborative management of a unique and entire supply chain, where each enterprise participates at a high level. However, manufacturing enterprises usually join in two or more supply chains resulting in non-linear relationships. In this case, a centralized collaborative model is not appropriate.

In order to deal with this situation, decentralized management models have been proposed, like V-CFM [12] and the Partner-to-Partner Collaborative Model [31]. The former establishes an extended enterprise for supply chains focusing only on the modeling of a collaborative forecasting. The latter defines a collaborative business process divided into three subprocesses:

- Consensus at the level of Production Aggregated Planning (PAP). Its objective is to make enterprises reach a consensus about a material provision plan at PAP level. Here, trading partners agree on the products they will collaborate (at the product family level), the periods of that collaboration (horizon is between 6 and 18 months), and approximate quantities of these products.
- Consensus at the level of Production Master Program (PMP). In this subprocess, enterprises have to arrive at a consensus about a material provision scheduling at PMP level. At this level, products are defined at the highest detail level required by the manufacturer enterprise. In addition, periods and quantities are specified at a higher level of detail. The customer specifies the required material quantity and due date. The supplier defines the date and the size of provision orders, considering the frame agreement. This is "probable" information.
- Consensus at the level of Provision Order Program (POP). Its objective is to make trading partners reach a consensus about the definition of a provision order scheduling. At this level, "true" information is managed. Periods indicate an accurate day in which products will be available to be dispatched. A provision and a production order schedule for both trading partners are defined.

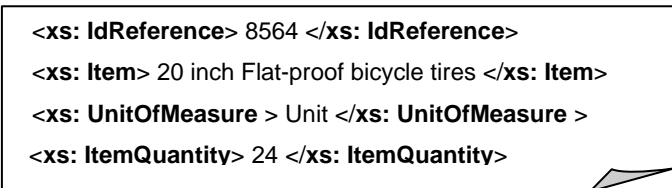
To support the execution of the subprocesses mentioned above, EBDs have to be exchanged between trading partners. These EBDs are standardized data structures, which on the one hand, replace traditional business documents such as invoice and dispatch notes, and on the other hand, support the exchange of information generated by the subprocesses such as sale point data and order forecasting.

2.2 Electronic Business Documents: The Need for Explicit and Formal Semantics

In order to exchange business information with their trading partners, an enterprise has to face two issues: how to access it, and how to understand its meaning [36]. This is a challenging task since a collaborative relationship implies the integration of heterogeneous information sources that belong to different enterprises with their own systems and culture.

The problem of accessing business information from different sources, that run over different technologies, has been solved with the advent of standards, such as the adoption of the eXtensible Markup Language (XML) [4]. XML was created to facilitate data integration using XML documents. All XML documents have to be associated with a definition. This definition is expressed using the XML Schema language [14]. Therefore, an XML document is then valid if it has an associated definition and complies with the grammatical constraints of that definition. That validation is carried out by a XML processor. In literature, different standards for describing EBDs based on the XML Schema language can be found. These standards can be divided into [21]: (1) standards schemas for messages, like cXML [9], and xCBL [35]; and (2) standards for defining both the processes and the associated message schemas, like RosettaNet [24], ebXML [34], and OAGIS [28].

In order to have application programs that talk to each other, they need to know not only how the data that they receive are structured, but also what the vocabulary means, i.e., their semantics. Without semantics, nothing is known about what the data means. Semantics can be defined as anything that helps to determine the meaning of the data [30]. In a XML document, the elements are delimited by start and end tags, as shown in Figure 1. XML allows to create an own markup, e.g. <Item> or <ItemQuantity>, which carry some implicit semantics for people. From a computational perspective, tags like <ItemQuantity> carry as much semantics as a tag like <H1>. A computer simply does not know, what an ItemQuantity is and how the concept ItemQuantity is related to e.g. the concept UnitOfMeasure. So an explicit semantics is needed. XML syntax is designed for representing an encoded serialization, and thus has a very limited range of expression for modeling complex object semantics, where "semantics" fundamentally means an intricate web of constrained relationships and properties [10]. In addition, in XML, semantics can be expressed only informally. Other languages may be used to express and validate semantic integrity constraints.



```
<xs: IdReference> 8564 </xs: IdReference>
<xs: Item> 20 inch Flat-proof bicycle tires </xs: Item>
<xs: UnitOfMeasure > Unit </xs: UnitOfMeasure >
<xs: ItemQuantity> 24 </xs: ItemQuantity>
```

Figure 1: An example of an XML document.

As pointed out above, the use of XML is insufficient for determining the resource information semantics. One way to overcome this problem is the use of an explicit model that can be used 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 describe the content of each document. In this sense, some initiatives based on the use of ontologies were proposed [5]. These initiatives were focused on semantically describing the tags or labels of the EBDs without considering their content. However, in order to achieve a real communication, the information interchanged in the EBDs (content) has to be semantically described, too. In addition, the previous initiatives propose to define a global meaning. In spite of these initiatives were a great advance in the area, a lot of work remains to be done. Collaborative commerce enterprises want to exchange only the necessary information to achieve common goals with their trading partners, without losing their autonomy and privacy. This fact implies that each enterprise manages its own information with particular characteristics. In addition, the information and its semantics have to be maintained locally, allowing only to the access of the authorized application or process. Then, in this scenario, it is necessary to define a new approach, which allows trading partners to exchange information without imposing a global meaning. To this aim, the idea of context is applied.

2.3 Context and Ontology

Since ontology and context have been used in different disciplines for different purposes, several definitions can be found in literature. The objective of this section is to give precise definitions to the terminology around these concepts as used in this paper.

In the area of collaborative commerce, trading partners have to deal mainly with particular entities, i.e. with individual clients, suppliers, manufactured and traded products, and their characteristics. These particulars (individuals, tokens) are simply located entities, bound to a specific (normally topologically connected) location in space and time [25]. Examples of particulars are this packaging industry, this collaborative relation, this product commercialized.

In order to refer to particulars, it is possible to use general terms such as “enterprise”, “process”, and “product”, which represent structures or characteristics in reality, which are exemplified. These general terms refer to universals. A universal is something that is shared in common by all those particulars, which are its instances [26].

Particulars and universals are connected to each other in various relations. Thus, a particular is connected to the corresponding universal in the relation of instantiation. Other relations may be between universals, such as “skim milk is_a milk”, “whole milk is_a milk”, and “product has_size size”. In these relations, “skim milk”, “whole milk”, “milk”, “product”, and “size” refer to universals, and “is_a” and “has_size” are the names of the relations. Based on the is_a relation, universals can be organized into trees or taxonomies. In addition, a universal may be connected to other universals representing its essential features by means of a relation of characterization. “has_size” is an example of this type of relation. Finally, other relations may be between particulars, such as “product1 has_size s196”, where “product1” and “s196” refer to particulars and “has_size” is an instance of the relation of the same name.

In a collaborative relation, trading partners exchange information to each other. In order to achieve real communication, they have to agree on the meaning of the interchanged information. To this aim, they can use domain ontologies. A domain is a portion of reality [26]. A domain ontology is a representational artifact used to render the cognitive representations of a domain. In this way, the cognitive representations can be shared and agreed upon by people that have to try with them, and can be made available to machine agents, too. For sake of simplicity, the word “ontology” will be used to refer to the term “domain ontology”.

In this paper, there are identified the following representational units: terms, relations between terms, and formal axioms. A *term* is a word or group of words intended to designate universals existing in a given domain whatever position the universals occupy in the tree. Examples are *Enterprise*, *Product*, *Process*, *Milk*, *SkimMilk*, *WholeMilk*, among others. By convention, each word designing a term begins in upper case. For compound names, the second and following words begin in upper case without spaces between them. Relations between terms refer to relations between universals in a given domain, that is to say, relations between the universals designated by those terms. *Relations* are formally defined as any subset of a product of n sets, that is: $R \subset T_1 \times T_2 \times \dots \times T_n$. Ontologies usually contain binary relations. The first argument is known as the domain of the relation, and the second argument is the range. For instance, the *isA* relation between *SkimMilk* and *Milk* terms states that “skim milk” is a “milk” too. In the same way, it is possible to use a binary relation *hasSize* between *Product* and *Size* terms to state that the size of a “product” is “size”. By convention, the names of relations begin in lower case. For compound names, the second and following words begin in upper case without spaces between them. *Formal axioms* are representational units that serve to model sentences or propositions that are always true in a given domain. They are normally used to represent knowledge that cannot be formally defined by the other representational units. For example, the following axiom “ \forall hasSize.Size” states that the range of the relation *hasSize* only is the set *Size*. As can be seen, axioms constrain the interpretation of other representational units in a given domain.

There is another possible type of representational unit for an ontology, called instance. An instance is used to represent a certain particular of a corresponding universal in the domain of discourse. By convention, the names of instances begin in lowercase and underlined. In this paper, it is called ontology with instances to an ontology that includes instances.

A term representing a universal and its instances are related by the relation of instantiation. For example, “product1 instanceOf Product” states that the particular “product1” is an instance of the universal “product”.

There are also relations that involve only instances. These relations are instances of relations between the corresponding terms. For example, the relation “product1 hasSize s196” is an instance of the relation “Product hasSize Size” between the *Product* and *Size* terms, where product1 is an instance of *Product* and s196 is an instance of *Size*.

As already mentioned, a domain is a portion of reality. This portion of reality comprehends both single universals and particulars and their more or less complex combinations [26]. In this paper, the set of essential features that characterize a domain, and whose meaning is dependent on that domain, is called the context of the domain. For example in the domain of collaborative relations, the regions in which companies are situated have their own economy. The features characterizing the economy of regions constitute a context.

3 Semantic Representation: Extending Domain Ontologies

The objective of this section is to present a way of how domain ontologies can be extended by making context explicit. To this aim, a motivating scenario is presented.

3.1 Motivating scenario

The motivating scenario is based on the Partner-to-Partner Collaborative Model [31], which proposes a decentralized management of long-term collaborations between pairs of enterprises from a supply chain. The Partner-To-Partner Collaborative Model involves coordinating two business process types: private processes, which are independently executed by each partner, and collaborative processes, which are jointly executed by both trading partners. The collaborative processes are defined as abstract ones. In order to implement a collaborative process, each trading partner has to define a business interface process (IP). The IP is responsible for the role that a partner plays in a

collaborative process and for the invocation and execution of those private processes required for the fulfilling of that collaborative process. For executing collaborative processes, EBDs are sent as part of each message interchanged between trading partners.

A UML Profile for Collaborative Business Processes based on Interaction Protocols (UP-CoIBPIP) has been proposed as a language for modeling collaborative processes [32]. UP-CoIBPIP allows the association of a speech act with business messages representing the intention that a trading partner has with regard to Electronic Business Documents (EBDs) sent in those messages. An EBD is a standardized data structure that replaces traditional business documents such as sale point data and order forecasting among others. In addition, they support the exchange of information required to execute collaborative processes [6]. Figure 2 shows an inter-enterprise collaboration between a supplier (e.g. a packaging industry) and one of its clients (e.g. a dairy industry). In this figure, the interaction protocol *Replenishment Plan Request* is shown. The purpose of this protocol is to agree on the replenishment plan to be defined by the supplier and its client. To this aim, an EBD is sent as part of each message interchanged between them.

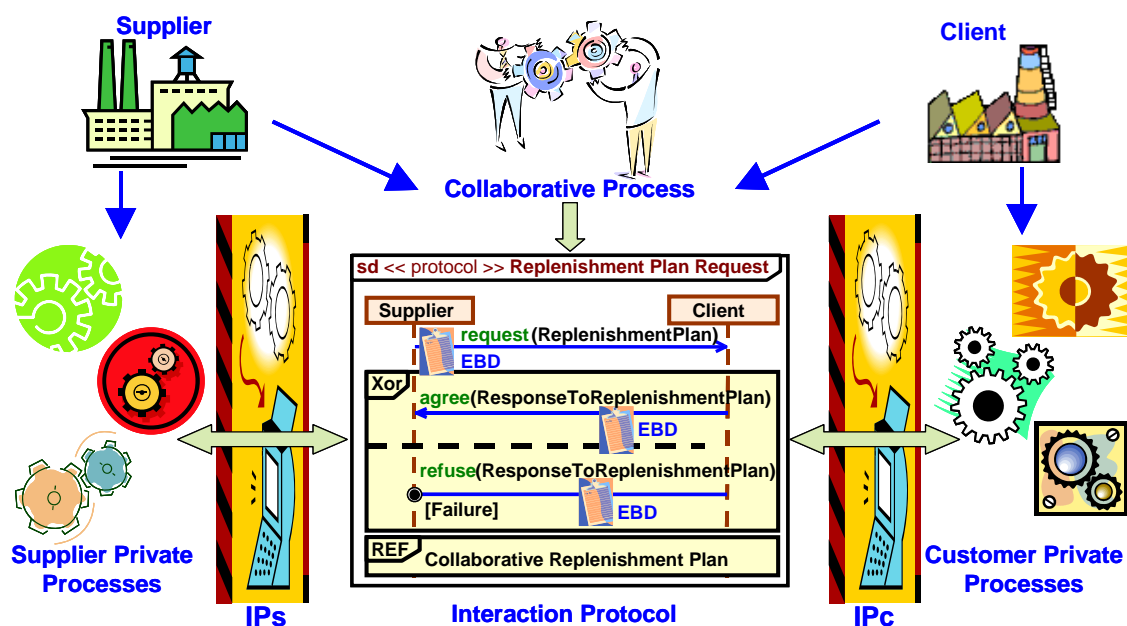
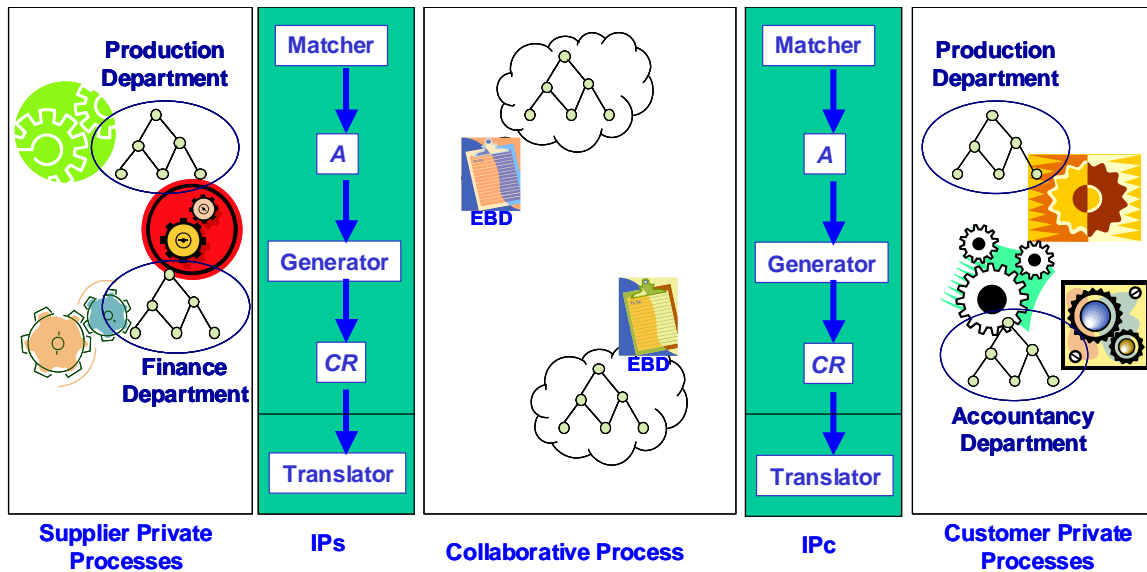


Figure 2: Elements of a collaborative relation between a supplier and one of its clients.

The supplier and its client have both their own information systems and culture. Then, to execute a collaborative process without misunderstanding, trading partners have to agree on the meaning of the information interchanged in the EBDs. When the IP of a trading partner receives an EBD, it has to be able to translate the information from that EBD to the internal processes according to the semantics of corresponding enterprise domains. When the IP has to send an EBD, first, it has to populate that EBD with the information generated by the corresponding enterprise domain according to the collaborative process semantics.

In a centrally managed relationship, the problem of misinterpretation would be solved by defining a global ontology administrated by the entity that manages the supply chain. A global ontology can be defined as an ontology, which describes general terms of a domain and it is agreed by a group of entities that belong to the domain. Nevertheless, in a relationship managed in a decentralized way, although it is technically viable to implement a global ontology, this one is not a proposal that is in tune with the objectives of the decentralized management. That is because each enterprise in a supply chain has its own interests; and its information systems and data structures have been designed to achieve those interests. In addition, when these enterprises decide to join one another in a collaborative process, they do it with a common interest but keeping their individuality and privacy. Privacy might be used not only to keep "intellectual property" hidden from the trading partners, but also decision support methodologies, thus enabling to obtain competitive advantages.

In a decentralized relationship, both enterprises should agree on the interchanged information semantics, which can be represented by means of ontologies. Figure 3 shows different ontologies participating in a collaborative relation between a supplier and one of its clients. Both trading partners can decide on an ontology for each EBD used to share information (center area of Figure 3). Private processes are the ones that have to deal with the EBD information. These private processes may affect different departments of a trading partner, which usually belong to different domains such as Accountancy, Production, Finance, etc., and have their own vision of the enterprise. Each department may have its own ontologies (left and right areas of Figure 3). These ones are different from the ontologies of the EBDs.



IPs: Business interface process of the supplier.
 IPC: Business interface process of the client.

Figure 3: Different ontologies in a collaborative relation.

Therefore, the problem here is how to transform the instances from the agreed ontology of an EBD to the corresponding instances of ontologies of the enterprise departments and vice versa. To this aim, at design time of collaborative relationships, the IP has to: (1) *match* relevant parts of internal ontologies and EBD ontologies, thus resulting in alignment A, (2) *generate* the conversion rules (CR) for transforming instances from one ontology into corresponding instances expressed in another ontology. Then, at run time, the IP has to execute a process for making the data *translation*.

Then, at design time, one important issue to fulfill is the definition of conversion rules to allow EBDs interoperability at semantic level. Before defining the conversion rules, the correspondences between the entities of the reality as represented in ontologies have to be identified. Following, an example of the problems to be faced during this process is showed.

Example. Let us suppose a collaborative relationship between a packaging industry (supplier) and a dairy industry (client). In order to achieve an agreement about a replenishment plan, both trading partners interchange the information shown in Table 1.

Table 1: Example of the information to decide on a replenishment plan

Horizon: 7/04-1/06				
Period	Product			Quantity
	Trademark	Type	Size	
7/04-13/04	yy	carton	1500	4400
			2900	2880
		plastic	196	1600
	250		1800	
	1000		6500	
	zz	carton	1500	2200
2900			8064	
plastic		250	1800	
14/04-20/04
...

For sake of simplicity, this paper is focused on one trading partner: the client. The analysis is similar to the other trading partner. Even though a trading partner can have several ontologies to describe its private information semantics, this example is focused on one ontology. Figure 4.a shows a portion of an ontology (O_{EBD}), agreed by both trading partners, which describes the semantics of an EBD used to interchange the information of Table 1 between them. Figure 4.b shows a portion of an ontology for the Production Department of the client (O_c).

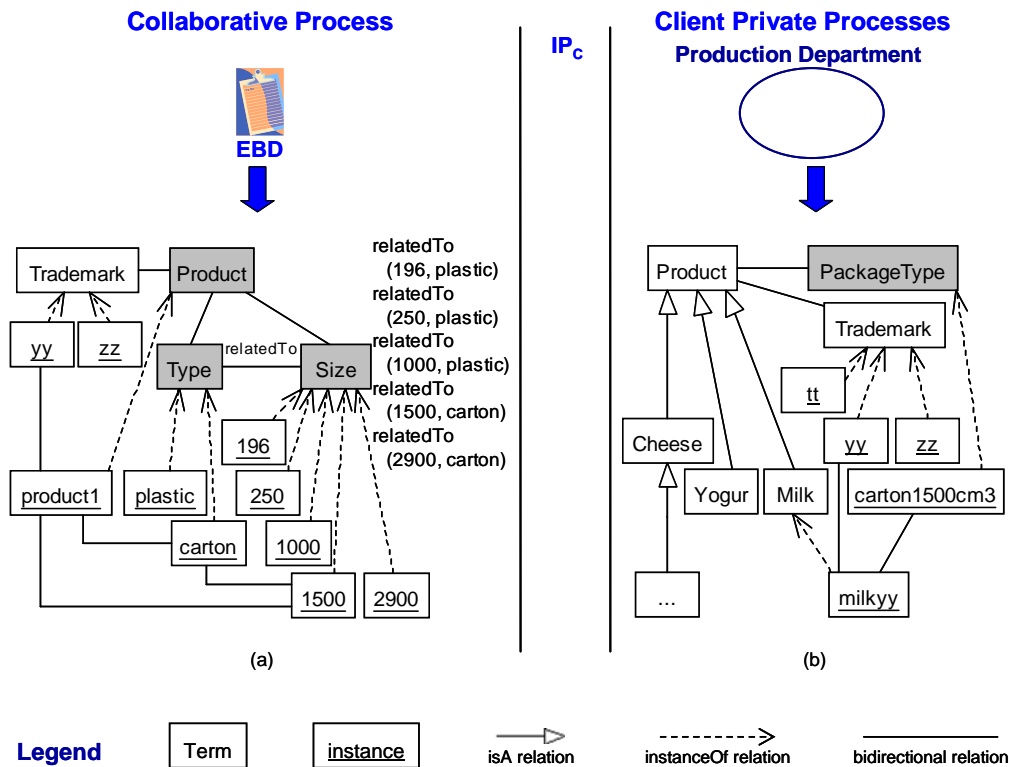


Figure 4: Client business Interface Process (IP_c) and ontologies that it deals with: (a) Portion of the O_{EBD} ontology. (b) Portion of one of the O_C ontologies.

In these ontologies, it is important to observe that the entities of reality are represented in different ways. This is because each trading partner perceives reality from a certain perspective and manages information from a certain granularity of focus. In this case, the entities of reality are the goods traded between partners. For the client, the term *Product* refers to the dairy products manufactured by him. In the collaborative process domain, the term *Product* refers to the goods traded between the supplier (the packaging industry) and the client (the dairy industry), packages in this case. Therefore, the translation process of the business interface process of the client has to manage conversion rules between the entity of reality “package” as represented in O_{EBD} and the same entity of reality “package” as represented in O_C.

Comparing O_{EBD} and O_C (Figure 4.a – 4.b), the entity of reality “package” is represented by the term *PackageType* in O_C, and by the terms *Size*, *Type*, and *Product* plus their relations in O_{EBD}. So, *PackageType* is related to *Product*, *Type*, and *Size* plus their relations. Analyzing the instances, carton1500cm3 from O_C has to be translated into O_{EBD} as:

```

product1 instanceOf Product and
carton instanceOf Type and
1500 instanceOf Size and
1500 relatedTo carton and
carton relatedTo 1500 and
carton typeOf product1 and
product1 hasType carton and
1500 sizeOf product1 and
product1 hasSize 1500
    
```

This means that, in order to translate this information from an ontology to another, the translation process of the client’s business interface process has to face the problem that a certain entity of reality is represented by a term in a particular ontology, but it is represented as a set of different representational units (terms, relations, axioms) in the other ontology. In addition, some entities of reality are represented in such a way that the representational units, which refer to them, have implicit interpretation features that are not necessary to do explicit when the ontology is used to specify the private information semantics within a given domain. For example in the domain of the collaborative process (Figure 4.a), it is assumed that the unit of measurement associated with sizes of product (i.e., the sizes of the entity of reality “package”) is expressed in cm³. Nevertheless, private processes of the client (Figure 4.b) could assume that measurement in liters. This fact could cause the definition of wrong conversion rules. Then,

any process dealing with the translation of instances of entities of reality between ontologies needs to identify discrepancies like the ones mentioned before.

3.2 Extending the Ontology: Making explicit context

With the aim of establishing effective conversion rules between ontologies from different domains, it is necessary to represent the entities of reality, whose instances must be translated, with the necessary degree of detail. In this paper, it is proposed to make explicit features which are usually implicit in an ontology and whose meaning is dependent on a given domain. The set of these features define the context of the domain. The conversion rules are managed by the business interface processes, which are responsible for making translations. For this, the IPs use extended ontologies obtained by taking the original ones and adding the necessary features of a given collaborative process. In this way, original ontologies are not affected, keeping the autonomy of the private information systems that use them, and favoring the reusability of the same ones in different domains.

In extended ontologies, representational units can be divided in two categories: content and context. The content representational units are those that already existed in the original ontologies, whereas the context representational units are those that are added in the extended ontologies. For example, the entity of reality “package” represented by the term `PackageType` in O_C (Figure 4.b) has implicit an association with a representation dimension. In this case, the dimension is not metric, but an enumeration of possible values. In order to make this dimension explicit, a term representing it has to be added to the original O_C . Figure 5 shows how the portion of interest of the extended ontology would be like (white terms belong to the set of content elements, and shaded terms belong to the set of context elements). In addition, this figure shows how the instance `carton1500cm3` would be represented. As can be seen, the instance `carton1500cm3` (Figure 4.b) has been divided in two parts in the extended ontology, `packagetype1` instanceOf `PackageType` and `carton1500cm3` instanceOf `PackageDimension`, which represents the value of the `packagetype1` instance in the domain of the private processes of the client. These two new instances are related by the `associatedWith` relation.

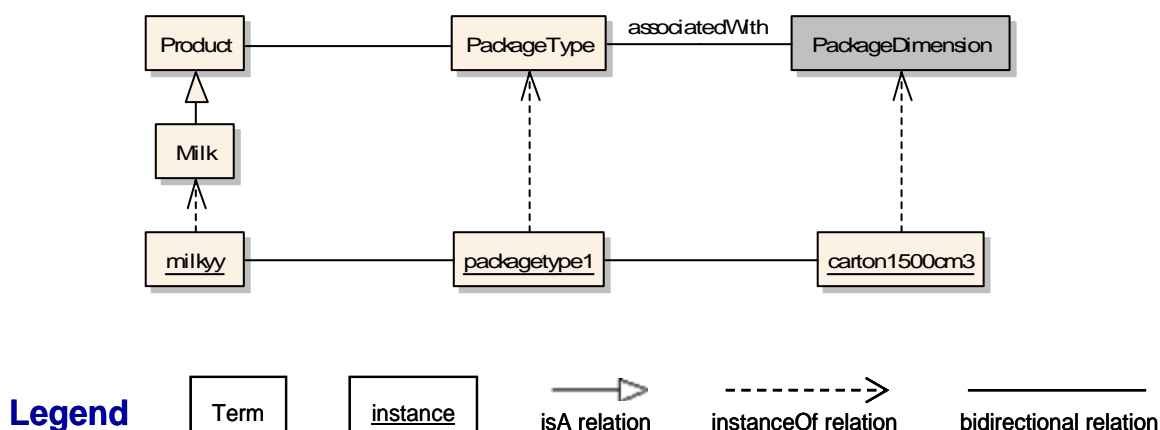


Figure 5: A portion of an extended O_C , and the `carton1500cm3` instance representation.

Similarly, the entity of reality “package” represented by the terms `Product`, `Type` and `Size` plus their relations in O_{EBD} (Figure 4.a) has implicit an association with a set of representation dimensions, named multi-dimension. These dimensions are qualities that cannot be assigned a value on one dimension without giving it a value on the other; they are called integral dimensions. In this case (Figure 6), the terms `TypeDimension` and `SizeDimension` represent integral dimensions that are used to define the product multi-dimension, represented by the term `ProductMultiDimension`. In its definition, this product multi-dimension has a set of rules constraining the relations between its constituting dimensions represented by a note related to the `ProductMultiDimension` term (upper right corner of Figure 6).

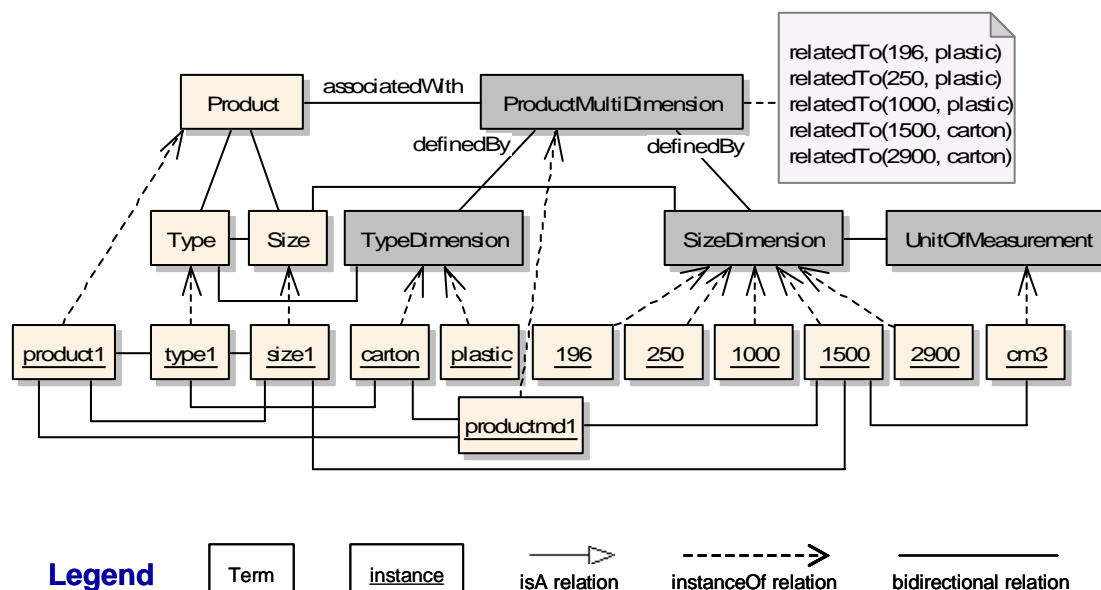


Figure 6: A portion of an extended O_{EBD} , and the equivalent of the `carton1500cm3` instance representation.

The Type Dimension is an enumeration of possible values (such as “carton” and “plastic”), and the Size Dimension is metric, i.e. its possible values are in the set of nonnegative numbers (196, 250, etc.). Additionally, Size Dimension has an associated metric unit, represented by the term `UnitOfMeasurement`, indicating that all the possible values of Size Dimension are expressed in that metric unit, cm^3 in this case. The `Type` term in O_{EBD} is implicitly associated with the Type Dimension, and the `Size` term in O_{EBD} is implicitly associated with the Size Dimension. The lower part of Figure 6 shows how the equivalent of the instance `carton1500cm3` would be represented in the extended O_{EBD} .

The presence of features inherent to entities of reality, whose representation and interpretation depends on the domain in which they are considered, can be observed in Figure 5 and Figure 6. In the domain of the private processes of the client, the entity “package” requires a representation so simple that it can be represented by a dimension, whereas in the domain of the collaborative process it requires of a set of dimensions.

In order to facilitate the generation of conversion rules, it is necessary to represent the entities, whose instances must be translated, with the necessary degree of detail. However, this does not guarantee that the translation of instances of an entity of reality from a domain to another one can be done by itself. First, the set of representational units and its semantics in a given domain have to be correctly identified. For example, in the extended O_C (Figure 5) the absence of a representational unit that designates the entity of reality represented by the `PackageType` term can be noticed. The same happens in the extended O_{EBD} (Figure 6) with the terms `Product`, `Type`, and `Size` plus their relations. The missing representational unit in each of these figures is the term `Package`. Figure 7 shows the two resulting extended ontologies, where the term `ProductMultiDimension` from the extended O_{EBD} was renamed `PackageMultiDimension` with the aim of properly refer to the semantics of the entity of reality.

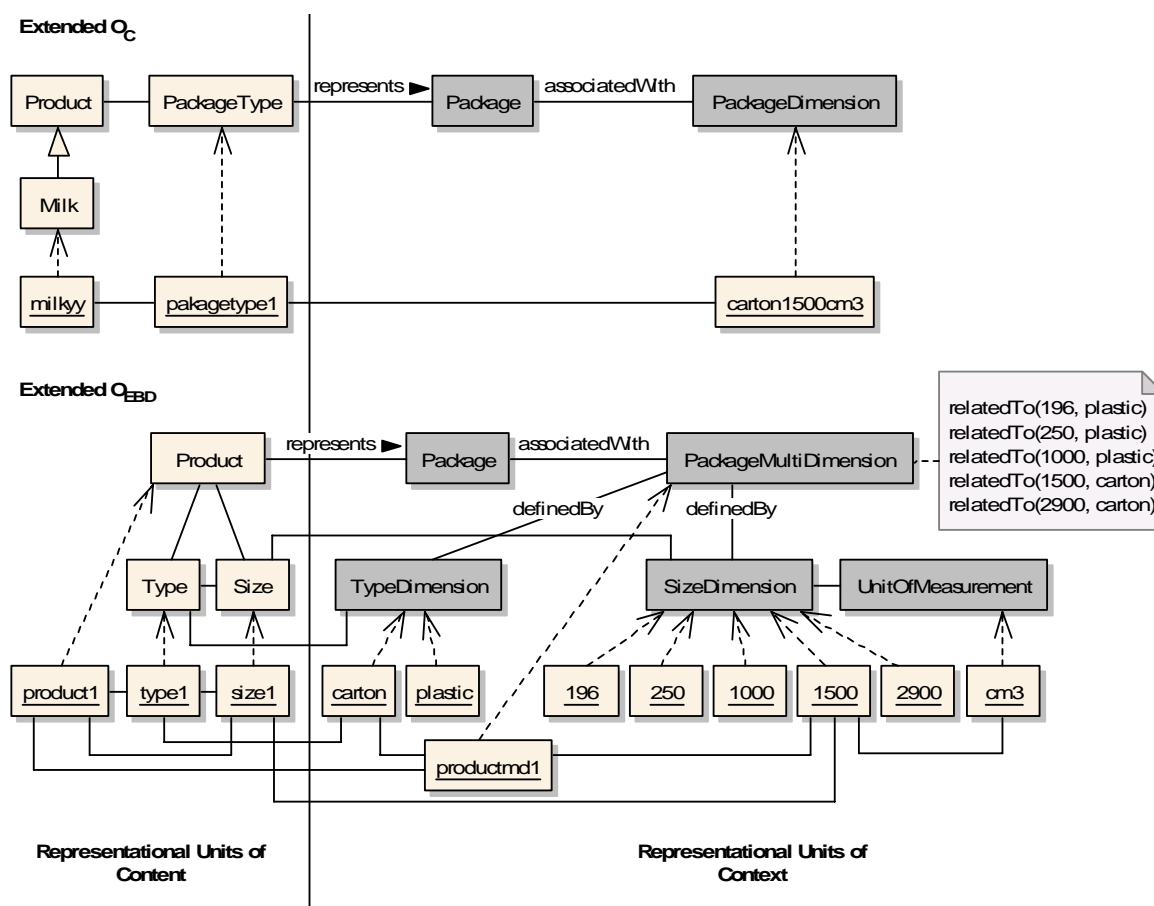


Figure 7: Extended O_C and extended O_{EBD} with instances.

The term *Package* refers to an entity, whose representation and interpretation depend on the domain in which it is considered. In the domain of the private processes of the client, the term *Package* is associated with a dimension, whereas in the domain of the collaborative process it is associated with a multi-dimension. Thus, the term *Package* represents a common feature to both domains, whose instances of ontologies have to be translated to each other. This kind of term will be used in order to support the generation of conversion rules, which will be automatically executed by the translation process to allow semantic interoperability between trading partners. Furthermore, considering that the conversion rules are generated from an alignment, entities represented in this way are a step toward the complete identification of the elements to translate and its features by the matching process. Moreover, extending ontologies like those of Figure 7, a conversion rule that associate the term *Product* in O_{EBD} (Figure 4.a) with the term *Product* in O_C (Figure 4.b) would not be erroneously defined.

4 An Ontology-based Reference Architecture

Previously, it was showed that, many different ontologies could take part in a collaborative commerce relationship. Some of them already existed when the collaborative commerce relationship was established. An example of these pre-existing ontologies is O_C (Figure 4.b) that describe private information of an enterprise from a given department viewpoint. Other ontologies are developed when the relationship is initiated, as O_{EBD} (Figure 4.a) that describes the semantics of the EBD agreed by both trading partners.

With the aim of defining effective conversion rules between these ontologies, while maintaining the autonomy of the private information systems, the use of extended ontologies by the IPs is proposed. In the case of pre-existing ontologies, so called original ontologies, the necessity of making explicit features usually implicit in these ontologies was showed. In other cases like the ontology that describes the semantics of the EBD (O_{EBD}) in the collaborative process, the ontology should be developed extended, that is to say, the entities of reality should be represented with the necessary degree of detail, making explicit all the features whose meaning is dependent on domain. Because the set of these features define the context of a domain, the extended ontologies will be hereinafter referred to as "contextualized ontologies".

In the previous subsection, representational units of contextualized ontologies were categorized as representational units of content and representational units of context. Representational units of content are those that already existed in the original ontologies. In the case of an ontology describing the semantics of an EBD, i.e. this is a new

ontology, representational units of content are those that describe semantically the syntactic structure of an EBD, and can be grouped in a structural ontology.

Representational units of context are those that are added to the contextualized ontologies in order to do explicit features that describe a given domain, and whose meaning is dependent on that domain. All the representational units of context that refer to the same domain can be grouped in an ontology called context specification ontology. In turn, there are features that are common to more than one domain; for example, features that describe temporal entities, such as instants or intervals; or a feature that describes spatial entities, like location. These features can be grouped by the type of entities they describe in a primitive feature ontology, and can be referenced from the context specification ontologies that require them.

All these different types of ontologies are combined in a comprehensive framework to facilitate the construction of some of them and its relationship to existing ones. Figure 8 shows the above-mentioned framework, which is presented considering one trading partner. Four types of ontology are defined: structural ontology (STO), context specification ontology (CSO), primitive feature ontology (PFO), and contextualized ontology (CO). These ontologies are described following.

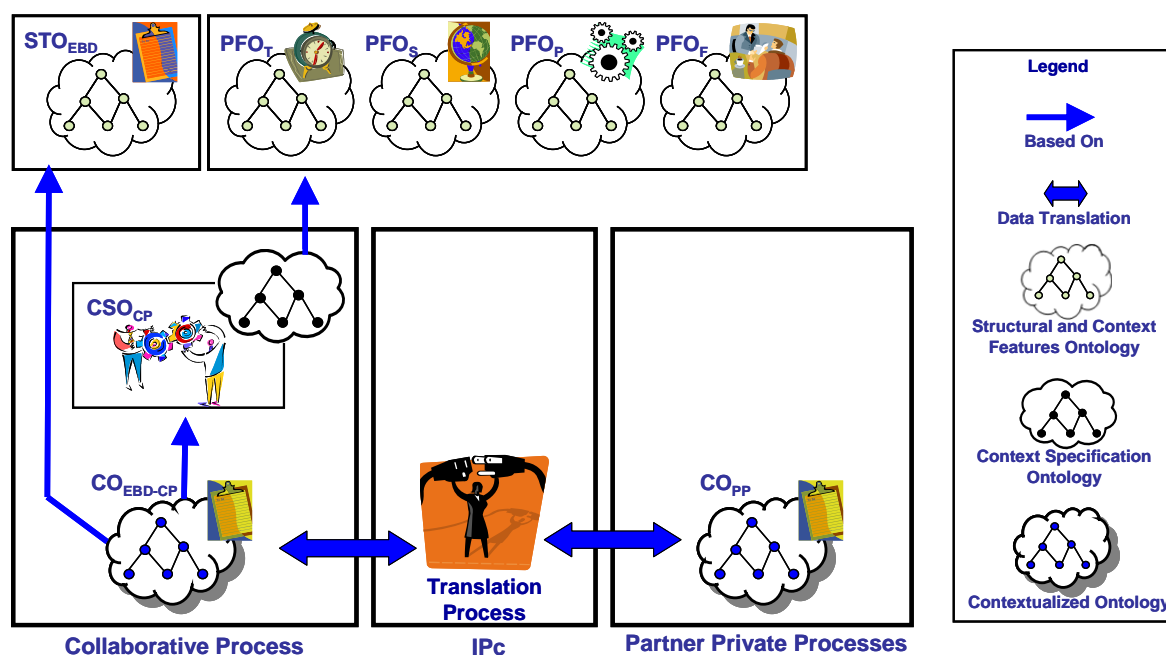


Figure 8: An ontology-based Reference Architecture

4.1 Structural Ontology (STO)

Trading partners in a collaborative relationship agree on a number of EBDs to interchange information. These EBDs can be seen as having two parts: a syntactic structure and content (the information to be interchanged). The syntactic structure is represented by a XML-schema document and the content is represented by a XML document. Then, a structural ontology is used to describe semantically the XML-schema document of an EBD. Then, a structural ontology consists of generic terms that refer to the representational units of the structure of an EBD, and it does not contain instances (see the left upper corner of Figure 8, STO_{EBD}). In this way, a structural ontology could be reused in different domains and with different objectives.

The structural ontology may be built ad hoc or may be taken from standards. Many XML-based EBDs for supporting collaborative commerce have been defined, as mentioned in Section 2.2. In addition, different mechanisms for extracting an ontology from XML-based documents can be found in literature [5].

4.2 Context Specification Ontology (CSO)

A context specification ontology is proposed to refer the particular features that describe a given domain. Since there are primitive features that appear more frequently in the collaborative commerce area, it is possible to specify a generic structure to describe such features; i.e., a primitive feature ontology (PFO) described next. Then, the context specification ontology may be built taking terms from these structures, and appending other needed terms that describe typical features of the given domain, or may be built ad hoc. The context specification ontology includes

instances. An example of this type of ontology is the CSO_{CP} (see the center-left area of Figure 8), which objective is to represent the features that describe the main characteristics of the Collaborative Process domain.

4.3 Primitive Feature Ontology (PFO)

This ontology has only representational units to describe the semantics of primitive features of domains. These primitive features can be classified into different types according to the perspective of the domain they describe. For example, Acker and Porter [1] have identified the following types of features: behavioral, structural, functional, temporal, and procedural among others. Meanwhile, Theodorakis [29] considers a context as a reference environment on which descriptions of real world objects are given. For him, a context captures different perspectives or records different levels of detail of a particular situation, and thus, represents partitions of real world. In his study of the use of context has identified the following partitions: temporal, spatial, functional, and viewpoints among others. From these perspectives or partitions, four appear more frequently in the collaborative commerce area: temporal, spatial, procedural, and functional. Each of them may have a primitive feature ontology (see the right upper corner of Figure 8, PFO_T , PFO_S , PFO_P , PFO_F).

- A Temporal PFO (PFO_T) describes the primitive characteristics of a temporal context. A temporal context is the one in which the temporal dimension takes to consider the information according to the time moment to which this one refers. For example, there is a demand for a product before the Argentine economic crisis in 2001, and another demand for that product after that moment. Both periods, before and after crisis, constitute different contexts.
- A Spatial PFO (PFO_S) describes the primitive characteristics of a spatial context. A spatial context is the one that allows grouping the information by place or region of origin. For example, the regions, in which companies have a collaborative relationship, have their own economy. Each region constitutes a different context.
- A Process PFO (PFO_P) describes the primitive characteristics of a process context. Considering the collaborative relationships, there is a need to define a type of process context that allows grouping activities, resources, and restrictions of processes. For example, a collaborative production-planning model defines three levels of collaboration: Consensus at level of Production Aggregated Planning (PAP), Consensus at level of Production Master Program (PMP), and Consensus at level of Provision Orders Program (POP). The EBDs used to interchange information will have a different context according to the level in which they are used.
- A Functional PFO (PFO_F) describes the primitive characteristics of a functional context. A functional context is the one that is created with the purpose of representing the different roles people or concepts can play. For example, in a collaborative relationship, participating companies assume different roles (client or supplier). Each role constitutes a different context.

Other PFOs may exist to describe the characteristics of other context types. The PFOs can be standards, that is to say, ontologies agreed by the whole community; or they can be agreed by both trading partners in the collaborative process. Additionally, these ontologies do not have instances.

4.4 Contextualized Ontology (CO)

Contextualized ontologies are directly created when these relate to EBDs, or are extensions of pre-existing ontologies when these relate to the private information of an enterprise. In this paper, the focus is on the contextualized ontologies associated with the EBDs.

When a collaborative commerce relationship is initiated, both trading partners have to agree on the semantic interpretation of the content of the EBDs interchanged between them. This semantic interpretation depends on the context (the set of features that describe a domain) in which the EBD is considered. Then, each EBD agreed by trading partners has an associated contextualized ontology, which is based on the structural ontology for the EBD (STO_{EBD}) and the context specification ontology for the given domain, the collaborative process in this case (CSO_{CP}) (see the bottom-left area of Figure 8, CO_{EBD-CP}). This contextualized ontology allows relating terms from the structural ontology (STO) to terms from the context specification ontology (CSO).

In this way, the semantics of the entities of a domain can be specified referring appropriate instances of the set of representational units representing the domain features. Making explicit these features, not only the semantics of the interchanged information in an EBD is clarified, but also the reusability of the same EBD in different domains is possible, since different features dependent on the domain can be associated with the same structure of an EBD. For example, one XML-based EBD could be used to interchange the information needed to execute the three subprocesses proposed by the Partner-to-Partner collaborative model (PAP, PMP, POP).

On the other hand, the content of the EBDs has to be interpreted according to the semantics of each enterprise department that has to deal with these EBDs. This semantics is represented by contextualized ontologies, which are extensions of original ontologies. This extension is out of the scope of this paper and will be presented in future work.

Then, in Figure 8 (see the center-left area), the semantics of an enterprise department is represented by the CO_{ED} ontology.

The translation process of each business interface process is the one that has to deal with all these contextualized ontologies in order to convert the instances from the agreed ontology of an EBD (CO_{EBD-CP}) to the corresponding instances of the ontologies of each enterprise department (CO_{ED}) and vice versa. These contextualized ontologies have the information interchanged between trading partners as instances.

5 Applying the Approach to the Example

The aim of this section is to show how the ontology-based reference architecture (see Figure 8 – Section 4) and the idea of making explicit the context behind a domain ontology (Section 3.2) can be used to define the semantics of an EBD interchanged between a packaging industry and a dairy industry (Section 3.1).

When two partners agree to collaborate, they have to define a collaborative process. This process is the domain in which collaborative relations take place. The characteristics of this domain are described by a CSO of Collaborative Process (CSO_{CP}), as explained in Section 4.2. Terms in this ontology may describe things like currency in which prices are expressed, unit of measure in which volumes are indicated, time horizon for the collaboration, and products involved in the collaboration, among others. These terms may be created based on PFO_T , PFO_S , PFO_P , and PFO_F terms or not. Figure 9 shows an example of a CSO for the PMP collaborative process where the $TimeHorizon$ term is created based on a term of PFO_T (e.g. $Interval$ from the “entry” sub-ontology of Hobbs and Pan [18]). In the same way, $Currency$, $Unit$, and $CapacityMeasurement$ are terms created based on terms of PFO_S (e.g. $Currency$ from the $Currency$ ontology of DAML [11], and $UMUni$ and $VolumeMeasurement$ from an ontology of visible object on the Earth’s surface derived from the Visual Objects Taxonomy (VOT) developed at the University of West Florida - etd.lib.fsu.edu). Additionally, observing Table 1, the CSO_{PMPCP} have to append terms that describe typical characteristics of the collaborative domain that do not derive from these PFOs, such as the term $Package$ that refers to the products that are commercialized between the partners.

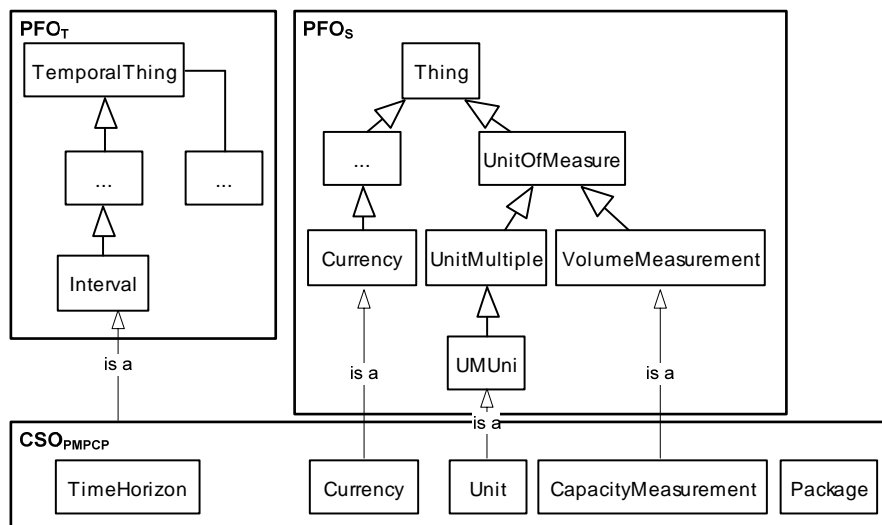


Figure 9: An example of a Context Specification Ontology (CSO) for the Production Master Program (PMP) collaborative process

In order to interchange information, the packaging industry and the dairy industry have to define a number of EBDs. To this aim, they can define their own EBDs or they can use a standard. Supposing that the partners decide to base their collaboration on the OAGIS [28] standard, the information of Table 1 can be interchanged between partners by using the ShipmentPlanningSchedule XML schema. This schema has an STO (STO_{EBD}) associated that describes its syntactic structure as explained in Section 4.1. A portion of the ShipmentPlanningSchedule document and its STO_{EBD} is shown in Figure 10.

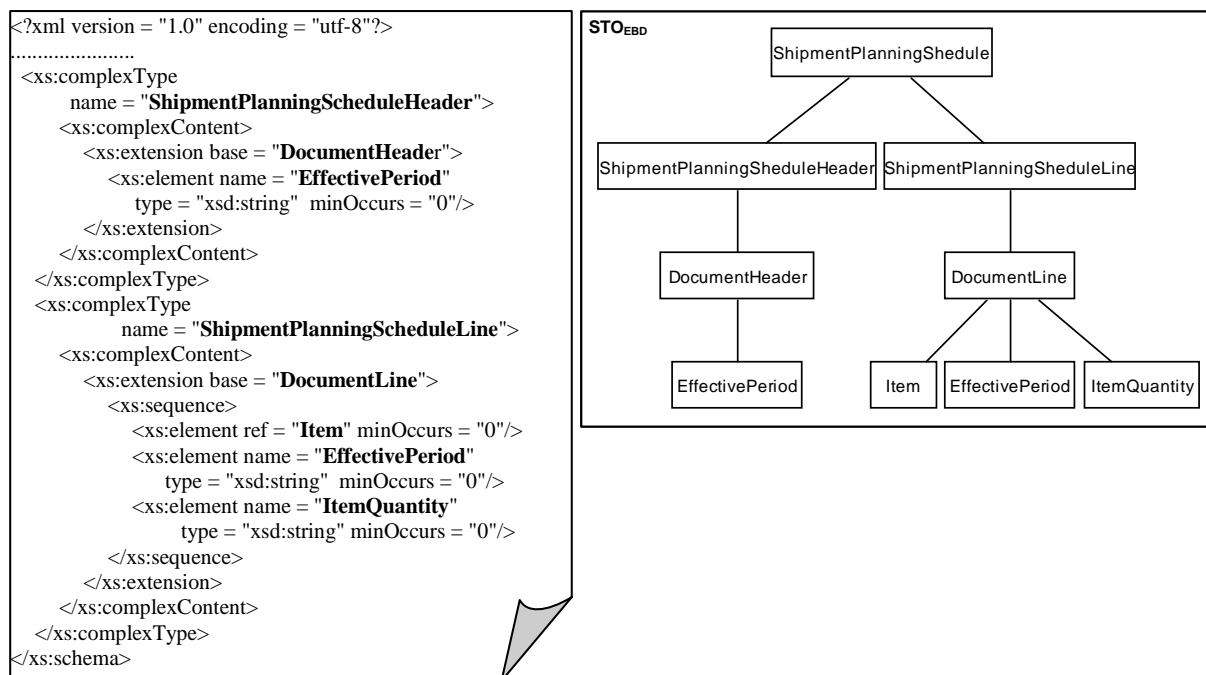


Figure 10: A portion of the ShipmentPlanningSchedule XML schema [28] and its STO associated.

In addition, in order to achieve an effective communication, partners have to agree on the semantics of the content of EBDs. Then, each agreed EBD has an associated CO_{EBD-CP} , which is based on a STO for the EBD (STO_{EBD}) and the CSO_{CP} as mentioned in Section 4.4. Figure 11 shows an example of CO_{EBD-CP} for the Production Master Program (PMP) collaborative process, $CO_{EBD-PMPCP}$. In this figure, it can be seen all terms from the STO_{EBD} , which describes the structure of the EBD, and how these terms are related to terms from CSO_{PMPCP} , which describes the characteristics of the PMP collaborative process domain. As indicated by thick arrows, the STO_{EBD} and the CSO_{PMPCP} ontologies are imported, i.e., are reutilized, and thus, are not defined whenever these ontologies are needed.

As can be seen in Figure 11, the use of an EBD in the domain of PMP collaborative process implies that the instances of *EffectivePeriod* term must be interpreted as weeks, instances of *ItemQuantity* term must be interpreted as units of packs, and instances of *Item* term must be interpreted as packages with the structure defined by the corresponding context specification ontology (CSO_{PMPCP}). This is thus due to the relations "interpretedAs" between the STO terms and the CSO terms, and because of the values of the CSO instances.

Nevertheless, the same EBD could be used in another domain, e.g. the Production Aggregated Planning (PAP) collaborative process. Then, the same STO_{EBD} would be associated with another CSO, CSO_{PAPCP} , where the instance of the *TimeHorizon* term is month, and the instance of the *Unit* term is liters. Then, the agreed EBD would have an associated $CO_{EBD-PAPCP}$, which instances of *EffectivePeriod* term would be interpreted as months, and instances of *ItemQuantity* term would be interpreted as liters.

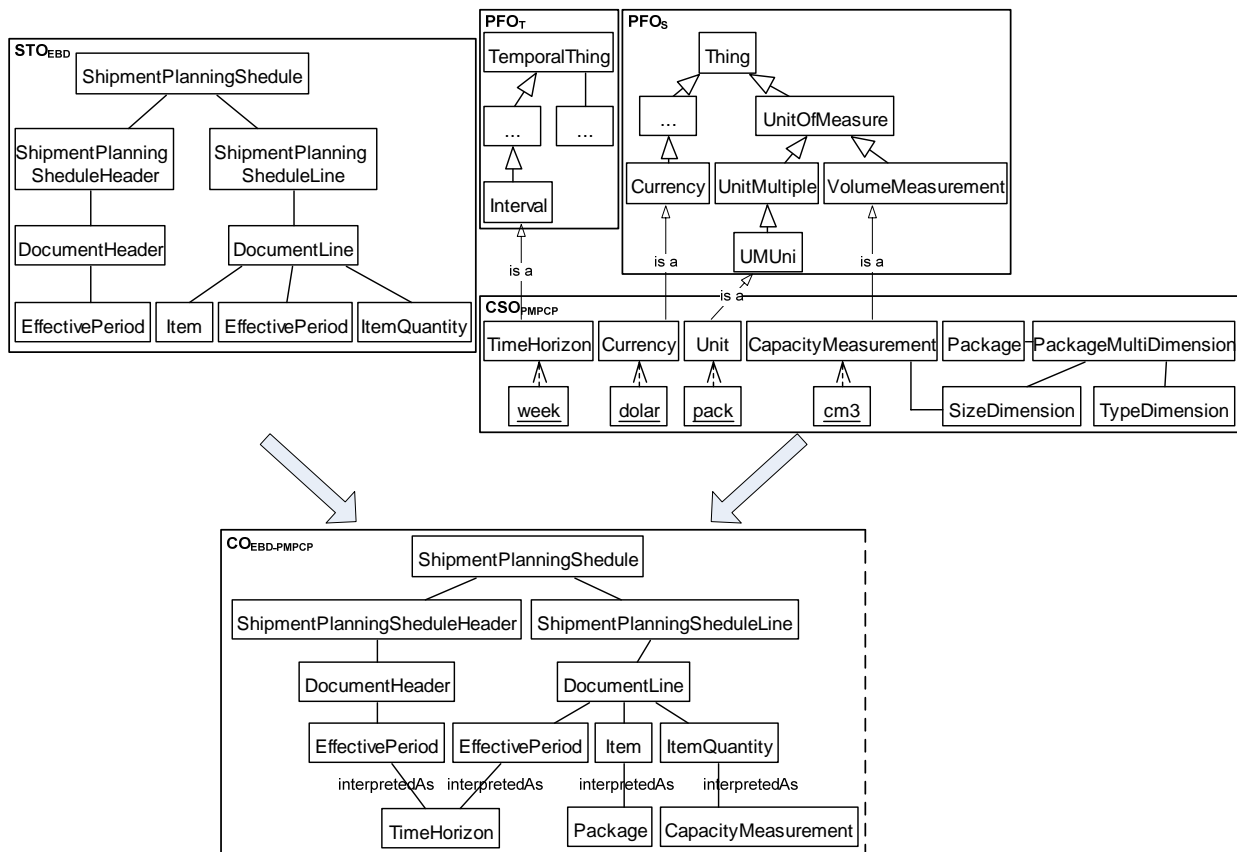


Figure 11: An example of a Contextualized Ontology (CO) for an EBD used in the Production Master Program (PMP) collaborative process domain.

6 Related Work

This section compares the approach presented in this paper with related work, including other ways of representing semantics via context and ontology, and other researches that have faced the ontology interoperability problem.

6.1 Representing Semantics via Context and Ontology

Information semantics could be represented by means of ontologies. Several definitions of ontology were stated [16, 29]. However, they agree on the fact that ontologies aim at capturing a consensual meaning of a domain entity in a generic way, and that ontologies may be reused and shared across software applications and by groups of people [16].

Another strategy for representing semantics is to use contexts. In computer science, a number of formal and informal definitions of context have appeared in several areas, such as information modeling [29], conceptual graphs [23], and artificial intelligence [22]. In [29], a context is viewed as a non-shared reference environment in which descriptions of real world objects are given.

The strengths of ontologies are the weaknesses of contexts and vice versa [3]. Ontologies define a common understanding of specific terms, and thus make the semantic interoperability between systems possible, but they can only be used after reaching consensus about their content. Contexts encode non-shared individual interpretation schemas that are easy to define and maintain since they can be created with a limited consensus among parties. However, communication between systems can be achieved only by constructing explicit mappings.

The idea of combining context and ontology was presented by Kashyap and Sheth [20], among others. The authors proposed to use terms in domain specific ontologies to characterize contextual descriptions. Therefore, ontology terms are contextual descriptors. In [15], the ECOIN semantic interoperability framework has been defined. ECOIN proposes to define a single ontology just consisting of generic terms without specifying their exact semantics. Then, the ontology terms are specialized in local contexts to express specific meanings. A context model coupled with an ontology-shared model explicitly specifies possible modification dimensions of an ontological term. ECOIN defines mappings based on lifting axioms [17] that are structured according to a context model and defined for each modification dimension as a conversion function network.

The ideas described in the previous paragraph result in a simpler context model, which works very well in a domain where it is possible to define a single ontology and to relate it to multiple contexts. These approaches are suitable for integrating different databases describing its content using descriptors based on a global shared ontology.

In [3], the C-OWL language to represent contextual ontologies, has been presented. C-OWL allows a user to define ontologies alignment where it is inappropriate to define a global shared ontology. An alignment is the set of correspondences between two or more ontologies. Ontology is contextualized when its contents are kept local and mapped with the contents of other ontologies via explicit mappings using bridge rules. A mapping is the oriented version of an alignment: it maps the entities of one ontology to at most one entity of another ontology. The bridge rules defined by [3] represent the following mappings: equivalent to, more general than, less general than, compatible, and incompatible. As regards context, the authors assume that a context is a concrete domain viewed from a description logic perspective.

6.2 Addressing Ontology Interoperability

In open or evolving systems, different parties could adopt, in general, different ontologies. Thus, merely using ontologies does not reduce heterogeneity: it raises heterogeneity problems at a higher level [13].

Ontology-matching is a plausible solution to the semantic heterogeneity problem [13]. Ontology-matching aims at finding correspondences between semantically related entities of different ontologies. These correspondences may stand for equivalence as well as other relations, such as consequence, subsumption, or disjointness, between ontology entities. Ontology-matching results, called alignments, can thus express with various degrees of precision the relations between the ontologies under consideration [13].

There are different algorithms for implementing the match process, which can be classified in schema-based and instance-based matching. A schema-based matcher considers the structural conflicts and uses some similarity measure to determine correspondence. An example of these matchers is the H-Match algorithm, which dynamically performs ontology matching at different levels of depth [7]. On the other hand, an instance-based matcher considers the instances conflicts and terminological conflicts. An example of these matchers is the FCA-Merge [27]. These matchers can also be combined like in the IF-Map algorithm [19]. A complete classification of these algorithms can be found in [13].

The improvement of the alignment, a set of correspondences between ontologies, is crucial to facilitate the generation of conversion rules. To this aim, the matching process as well as the input to it could be improved. Contrary to the approaches previously presented, it was proposed to improve the input to the matching process, i.e. the development of the ontologies to be matched.

7 Conclusions and Future Work

Decentralized collaborative relationship allows each trading partner to maintain privacy and autonomy of their processes and data structure while exchanging information between them. Moreover, the decentralized management of the supply chain allows a partner to manage simultaneous relations with different partners avoiding, in this way, possible conflicting situations. Trading partners who participate in these supply chains are in most cases geographically distributed and belong to different domains. The completely distributed nature and high degree of autonomy of individual peers in this collaborative schema comes with new challenges to achieve interoperability at semantic level between electronic business documents.

In this paper, an approach for representing the semantics of the electronic business documents interchanged between trading partners for supporting a decentralized collaborative relationship was presented. The approach presented in this paper defines a strategy based on ontology and context, avoiding defining a global meaning, which is sometimes difficult even an impossible solution in a decentralized relationship. Using the idea of context, this approach makes some characteristics that are generally implicit in an ontology explicit. The main advantage of this approach is that it facilitates the semantic interoperability between completely autonomous entities belonging to different domains, while maintaining the autonomy and privacy requirements of the participating entities. The main disadvantage of this approach is the complexity of the ontology maintenance tasks. That means, the approach proposed to define a set of ontologies related each other, some of them agreed by different stakeholders, which make the maintenance task difficult.

In addition, an application example was shown. Through this application example, it is possible to see how a same electronic business document could be used in different collaborative processes without misunderstanding following the ontology-based reference architecture. This is possible due to the semantics of the electronic business documents is clearly specified extending the domain ontologies by making explicit the characteristics of the context domain in which the electronic business document is used.

Despite the disadvantage of the required time to develop the extended ontologies, this approach improves significantly the output of the matching process, and facilitates the generation of conversion rules that are the core of the data translation process.

The definition of this approach is the first step in the development of a tool for solving semantic interoperability problem between trading partners that belong to a decentralized collaborative relationship. Since adding semantics to data interchanged between trading partners without affecting its privacy and autonomy is an emerging issue, a lot of work remains to be done. Future work will be focused on studying both methodology for extending ontologies and how conversion rules could be defined from the alignment.

Acknowledgments

The authors are grateful to "Universidad Tecnológica Nacional (UTN)", "Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)", and "Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT)" for their financial support.

References

- [1] L. Acker and B. Porter, Extracting Viewpoints from Knowledge Bases, in Proceedings of 20th National Conference on Artificial Intelligence. Seattle, WA, USA. AAAI Press, 1994, pp. 547-552.
- [2] B. Beran, A. Saiful Islam, L. Bermudez, S. Fella and M. Piasecki. (2004, August) Drexel University. [Online]. Available: <http://loki.cae.drexel.edu/~wbs/ontology/2004/09/iso-19103.owl>
- [3] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini and H. Stuckenschmidt, Contextualizing Ontologies. Journal of Web Semantics: Science, Services and Agents on the World Wide Web 1(4), 2004, pp. 325-343.
- [4] T. Bray, J. Paoli, C. Sperberg-McQueen and E. Maler (2006, August) Extensible Markup Language (XML) 1.0 (Fourth Edition), W3C Recommendation. [Online]. Available: <http://www.w3.org/TR/REC-xml/>
- [5] M. L. Caliusco, M. R. Galli and O. Chiotti, Ontology and XML-Based Specifications for Collaborative B2B Relationships, CLEI Electronic Journal, vol. 7, no. 1, 2004.
- [6] M. L. Caliusco, M. R. Galli and O. Chiotti, Ontology role in B2B Relationships. XXIX Conferencia Latinoamericana de Informática (CLEI 2003) La Paz, Bolivia. September, 2003.
- [7] S. Castano, A. Ferrara, S. Montanelli and G. Racca. Semantic Information Interoperability in Open Networked Systems. In: Semantics for Grid Databases, First International IFIP Conference, ICSNW 2004, Paris, France, June 17-19, 2004, Revised Selected Papers. LNCS 3226. pp 215-230.
- [8] L. Chapman and M. Petersen, Demand Activated Manufacturing Architecture (DAMA). Model for Supply Chain Collaboration, in Proceedings of Conference on Modeling and Analysis of Semiconductor Manufacturing. Arizona State University, Tempe, 2000.
- [9] Commerce XML (2007, June). [Online]. Available: <http://www.cxml.org>
- [10] R. Cover (1998, November) The XML Cover Pages, XML and Semantics Transparency. [Online] Available: <http://www.oasis-open.org/cover/xmlAndSemantics.html>
- [11] DAML Currency ontology (2004, March) [Online]. Available: <http://www.daml.ecs.soton.ac.uk/ont/currency.owl>
- [12] F. Esteban, A. Bas, R. Escoto and D. Perales, Supply Chain Management. Modelling Collaborative Decision, in Proceedings of 9th Emerging Technologies and Factory Automation. Lisbon, Portugal. IEEE Conference, 2003, pp. 137-141.
- [13] J. Euzenat, P. Shvaiko. Ontology Matching, London: Springer, 2007.
- [14] D. C. Fallside (2001, May) XML Schema Part 0: Primer W3C Recommendation. [Online]. Available: <http://www.w3.org/TR/xmlschema-0/>
- [15] A. Firat, S. Madnick, and F. Manola, Multi-dimensional Ontology Views via Contexts in the ECOIN Semantic Interoperability Framework. In: Contexts and Ontologies: Theory, Practice and Applications. AAAI Workshop. AAAI Press, 2005, pp. 1-8.
- [16] A. Gómez-Pérez, M. Fernández-López and O. Corcho, Ontological Engineering. 2nd edn. London: Springer-Verlag, 2004.
- [17] R. Guha, Contexts: A Formalization and Some Applications. PhD Thesis. Computer Science Dep., Stanford University, 1995.
- [18] J. Hobbs and F. Pan, An Ontology of Time for the Semantic Web. ACM Transactions on Asian Language Information Processing (TALIP), vol. 3, no 1, pp. 66-85, 2004.
- [19] Y. Kalfoglou and M. Schorlemmer. IF-Map: an ontology mapping method based on information flow theory. Journal on Data Semantics, pp. 98-127, 2003.
- [20] V. Kashyap and A. Sheth, Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context and Ontologies. In: Papazoglou, M., Schlageter, G. (eds.): Cooperative Information Systems: Current Trends and Applications. Academic Press, London, 1996, pp. 139-178.
- [21] P. Kotinurmi, Comparing XML-based B2B frameworks, in Proceedings. XML Finland 2002, Helsinki, Finland.
- [22] J. McCarthy and S. Buvac, Formalizing Context (Expanded Notes). In: Aliseda, A., Glabbeek, R.V., Westersth, D. (eds.): Computing Natural Language. CSLI Lecture Notes 81, 1998, pp. 13-50.

- [23] G. Mineau and O. G., Gerbé, Contexts: A Formal Definition of Worlds of Assertions. In: Lukose, D., Delugach, H., Keeler, M., Searle, L., Sowa, J. (eds.): Conceptual Structures: Fulfilling Peirce's Dream, LNAI Vol. 1257. Springer-Verlag, 1997, pp. 80–94.
- [24] RosettaNet Standards (2007, June). [Online]. Available: <http://www.rosettanet.org>
- [25] B. Smith, Ontology. In L. Floridi (ed.), Blackwell Guide to the Philosophy of Computing and Information, Oxford: Blackwell, 2003, pp. 155–166.
- [26] B. Smith, W. Kusnierczyk, D. Schober and W. Ceusters, Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain. In Proc. of KR-MED 2006, Biomedical Ontology in Action, November 8, 2006, Baltimore MD, USA.
- [27] The Open Applications Group XML Project Team. OAGIS Release 8.0. (2002, May). [Online]. Available: <http://www.openapplications.org/downloads/oagidownloads.htm>
- [28] M. Theodorakis, Contextualization: An Abstraction Mechanism for Information Modeling. PhD Thesis. University of Crete, Greece, 2001.
- [29] M. Uschold, Where is the Semantics in the Semantic Web? AI Magazine, vol. 24, no 3, pp. 25–36, 2003.
- [30] G. Stumme and A. Mäedche. FCA-Merge: Bottom-up merging of ontologies. In: Proc. 17th International Joint Conference on Artificial Intelligence, pp 225-234, Seattle (WA US), 2001.
- [31] P. Villarreal, M. L. Caliusco, M. R. Galli, H. Salomone and O. Chiotti, Decentralized Process Management for Inter-Enterprise Collaboration. In: Sahay, B.S. (ed.): Emerging Issues in Supply Chain Management. Macmillan, India, 2004, pp. 293–310.
- [32] P. Villarreal, E. Salomone and O. Chiotti, Modeling and specifications of collaborative business processes using a MDA approach and a UML profile. In: Rittgen P, editor. Enterprise modeling and computing with UML. Idea Group Inc.; 2006.
- [33] Voluntary Interdisciplinary Commerce Standards Association (VICS) (2007, June) Collaborative Planning, Forecasting, and Replenishment (CPFR®) Version 2.0 (2002). [Online]. Available: <http://www.cpfr.org>
- [34] D. Waldt and R. Drummond, EbXML - The Global Standard for Electronic Business. [Online]. Available: http://www.ebxml.org/presentations/global_standard.htm
- [35] XML Common Business Library, version 4.0 (2003, March). [Online]. Available: <http://www.xcbl.org>
- [36] Y. Zhao and K. Sandahl, Potential Advantage of Semantic Web for Internet Commerce, in Proceedings of 5th International Conference on Enterprise Information Systems 4, 2003, pp. 151–160.